Algorithms for the solution of flow networks containing open channels

This item was submitted to Loughborough University's Institutional Repository by the/an author.

Additional Information:

• A Doctoral Thesis. Submitted in partial fulfilment of the requirements for the award of Doctor of Philosophy at Loughborough University.

Metadata Record: https://dspace.lboro.ac.uk/2134/33253

Publisher: © Newton Fan Young Ho

Rights: This work is made available according to the conditions of the Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International (CC BY-NC-ND 4.0) licence. Full details of this licence are available at: https://creativecommons.org/licenses/by-nc-nd/4.0/

Please cite the published version.
<table>
<thead>
<tr>
<th>AUTHOR/FILING TITLE</th>
<th>Ho, N.F.Y.</th>
</tr>
</thead>
<tbody>
<tr>
<td>VOL. NO.</td>
<td>0400910+3</td>
</tr>
<tr>
<td>CLASS MARK</td>
<td>Loan Copy</td>
</tr>
</tbody>
</table>
ALGORITHMS FOR THE SOLUTION OF FLOW NETWORKS CONTAINING OPEN CHANNELS

by

Newton Fan Young Ho

A Doctoral Thesis submitted in partial fulfillment of the requirements for the award of Doctor of Philosophy of Loughborough University of Technology

February 1994

© Newton Fan Young Ho 1994
DEDICATIONS

I would like to dedicate this work to

*Ho On Wai*

and

*Ho Young Kwan Tee*
ACKNOWLEDGEMENTS

The author wishes to extend his gratitude to his supervisor Dr Andrew Cotton for his valuable guidance, advice, support, patience and proof reading of this thesis. Otherwise, this work would not have been possible to complete.

Special thanks to Mrs Eva Tse, Mrs Tina Chan, Miss Anita Ho and Mr Ivan Ko for their hospitality, encouragement and support in preparing this thesis.

Thanks also to the staffs of Applications In Cadd Ltd., especially Dr John Strodachs and Mr Michael Case, for their financial support and allowing me to use their office facilities throughout the course of this research.
SYNOPSIS

Hydraulic networks involving free surface open channel flows have many applications, such as water and wastewater treatment plants, irrigation and drainage systems. In such systems, networks of open channels are interlinked with hydraulic control structures in a complex fashion, and determination of the resulting flow distribution and head-losses in such systems is far more complicated and time consuming than the analogous problem of pipe networks. This is because the relationship between flow and head-loss for free surface flow systems also depends upon the water levels and controls operating within the network.

The objective of the research is to develop a model which enables the flow distribution throughout such networks to be determined. Networks are simulated by building up 'strings' from a number of discrete 'hydraulic modules'. The strings are interlinked to define the network. The method of 'loop formulation' is used to relate discharge and head-loss along the strings. This results in a set of simultaneous non-linear equations which require solution. The basis of the derivation for the equations is energy and continuity of flow. As a first stage, flow profile computation for open channels is investigated, and the fourth-order Runge-Kutta procedure is found to be the most satisfactory. The research then investigates and compares the effectiveness of different methods of network analysis based on the Newton-Raphson method and quasi-Newton methods. Efficiency is measured in terms of function evaluations and accuracy rather than the actual computational time spent.

The performance of the algorithms is verified by simulating various flow conditions. These include parallel and nested hydraulic networks; 'switching' of hydraulic control; lateral discharge; and the formation of hydraulic jump. The method of BFGS (Broyden, Fletcher, Goldfarb and Shanno) is found to be the most effective in that it converges relatively rapidly to a final solution over a wide range of hydraulic conditions.

The method of 'line searching' is investigated to improve convergence in situations where the algorithm does not converge to a solution. This procedure can be embedded within the overall algorithm and was found to ensure convergence in the examples studied.

The sensitivity of flow distribution to network parameters was investigated theoretically and used examples. It was found that errors propagated during water surface profile computation are more significant than changes due to fluctuations in downstream control water levels.
# TABLE OF CONTENTS

**ACKNOWLEDGEMENTS**

**SYNOPSIS**

**LIST OF TABLES**

**LIST OF FIGURES**

**LIST OF SYMBOLS**

**CHAPTER 1:**

**INTRODUCTION**

1.1 Background  2
1.2 Aims and Objectives  7

**CHAPTER 2:**

**COMPUTATION OF WATER SURFACE PROFILE**

2.1 Introduction  11
2.2 Dynamic Equation of Gradually Varied Flow  12
2.3 Standard Step Method  18
2.4 Numerical Error Definitions  24
2.5 Numerical Integration Method  27
2.6 Trapezoidal Method  30
   2.6.1 Local Truncation Error in Trapezoidal Method  33
   2.6.2 Global Truncation Error in Trapezoidal Method  33
2.7 Runge-Kutta Method  35
   2.7.1 Local Truncation Error in Runge-Kutta Method  37
   2.7.2 Global Truncation Error in Runge-Kutta Method  40
2.8 Adaptive Step Size Control  42
2.9 Comparison of Numerical Procedures  45
   2.7.1 Magnitude of Local Truncation Error  47
   2.7.2 Predictions of Local Truncation Error  40
   2.9.3 Upper Bounds on Global Error  63
   2.9.4 Investigation of Adaptive Step Size Control  67
2.10 Concluding Remarks  74
CHAPTER 3: NETWORK ANALYSIS

3.1 Introduction 77
3.2 Conceptual Model 78
3.3 Governing Equations 80
  3.3.1 Nodal Principle 83
  3.3.2 Loop Principle 85
3.4 Applications of the Nodal Method 86
  3.4.1 Wylie's Approach 86
  3.4.2 Chaudhry and Schultes' Approach 90
3.5 Applications of the Loop Method 94
3.6 Concluding Remarks 97

CHAPTER 4: METHODS OF SOLUTION OF NETWORKS 99

4.1 Introduction 100
4.2 Newton-Raphson Method 102
4.3 Partial Derivatives for Newton-Raphson Method 104
  4.3.1 Analytical Derivation 104
  4.3.2 Numerical Derivation 111
4.4 Shortcomings of Newton-Raphson Method 114
  4.4.1 Initial Approximation 115
  4.4.2 Jacobian Matrix 115
4.5 Quasi-Newton Method 116
4.6 Line Search 123
  4.6.1 Bracketing Stage 127
  4.6.2 Sectioning Stage 132
4.7 Concluding Remarks 137

CHAPTER 5: APPLICATIONS AND TESTING OF METHODS OF SOLUTION OF NETWORKS 140

5.1 Introduction 141
5.2 Determination of Hydraulic Controls 143
  5.2.1 Artificial Controls 143
  5.2.2 Critical Depth Controls 146
# Chapter 8: Conclusions and Recommendations

## 8.1 Conclusions

## 8.2 Recommendations for Future Work

## References

## Appendices

### Appendix A:

**Detailed Calculation Procedures for Solution Algorithms**

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>A.1 Systems of Equations</td>
<td>A-2</td>
</tr>
<tr>
<td>A.2 Newton-Raphson Method</td>
<td>A-3</td>
</tr>
<tr>
<td>A.2.1 Partial Derivatives for Newton-Raphson Method</td>
<td>A-7</td>
</tr>
<tr>
<td>A.3 Broyden Method</td>
<td>A-9</td>
</tr>
<tr>
<td>A.4 Davidon-Fletcher-Powell Method</td>
<td>A-12</td>
</tr>
<tr>
<td>A.5 Broyden-Fletcher-Goldfarb-Shanno Method</td>
<td>A-15</td>
</tr>
</tbody>
</table>

### Appendix B:

**Program for Analysing Flow Distribution in Hydraulic Networks**

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>B.1 Description of Files</td>
<td>B-2</td>
</tr>
<tr>
<td>B.2 List of Programs</td>
<td>B-5</td>
</tr>
<tr>
<td>B.3 Input of Data</td>
<td>B-39</td>
</tr>
<tr>
<td>B.3.1 Input of Modules</td>
<td>B-39</td>
</tr>
<tr>
<td>B.2.1 Input of Strings</td>
<td>B-40</td>
</tr>
<tr>
<td>B.3.1 Input of Initial Flow Distribution</td>
<td>B-40</td>
</tr>
<tr>
<td>B.3 Execution of NETWORK</td>
<td>B-41</td>
</tr>
</tbody>
</table>
### LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1.</td>
<td>Water surface profile classification.</td>
<td>13</td>
</tr>
<tr>
<td>2.2.</td>
<td>Average friction slope.</td>
<td>20</td>
</tr>
<tr>
<td>2.3.</td>
<td>Physical characteristics of the open channel.</td>
<td>48</td>
</tr>
<tr>
<td>2.4a.</td>
<td>Relative truncation error, $h = 0.25$ (m).</td>
<td>49</td>
</tr>
<tr>
<td>2.4b.</td>
<td>Relative truncation error, $h = 2.5$ (m).</td>
<td>50</td>
</tr>
<tr>
<td>2.4c.</td>
<td>Relative truncation error, $h = 25$ (m).</td>
<td>51</td>
</tr>
<tr>
<td>2.4d.</td>
<td>Relative truncation error, $h = 250$ (m).</td>
<td>52</td>
</tr>
<tr>
<td>2.5a.</td>
<td>Fourth-order Runge-Kutta - Predictions of truncation error, $h = 250$ (m).</td>
<td>59</td>
</tr>
<tr>
<td>2.5b.</td>
<td>Fourth-order Merson - Predictions of truncation error, $h = 250$ (m).</td>
<td>60</td>
</tr>
<tr>
<td>2.6.</td>
<td>Accuracy of a water surface profile with adaptive step size control.</td>
<td>70</td>
</tr>
<tr>
<td>4.1.</td>
<td>List of methods for solving a set of simultaneous non-linear equations.</td>
<td>139</td>
</tr>
<tr>
<td>5.1.</td>
<td>Equations for thin-plate full-width weir.</td>
<td>144</td>
</tr>
<tr>
<td>5.2.</td>
<td>Flow classification for two connected channels.</td>
<td>148</td>
</tr>
<tr>
<td>5.3a.</td>
<td>Culvert flow characteristics.</td>
<td>154</td>
</tr>
<tr>
<td>5.3b.</td>
<td>Culvert flow equations.</td>
<td>154</td>
</tr>
<tr>
<td>5.4.</td>
<td>Parallel Network - Physical characteristics.</td>
<td>164</td>
</tr>
<tr>
<td>5.5a.</td>
<td>Parallel Network - Initial flow distribution.</td>
<td>165</td>
</tr>
<tr>
<td>5.5b.</td>
<td>Parallel Network - Final flow distribution.</td>
<td>165</td>
</tr>
<tr>
<td>5.6a.</td>
<td>Parallel Network - Convergence characteristics of the NR method.</td>
<td>166</td>
</tr>
<tr>
<td>5.6b.</td>
<td>Parallel Network - Convergence characteristics of the BR method.</td>
<td>167</td>
</tr>
<tr>
<td>5.6c.</td>
<td>Parallel Network - Convergence characteristics of the DFP method.</td>
<td>168</td>
</tr>
<tr>
<td>5.6d.</td>
<td>Parallel Network - Convergence characteristics of the BFGS method.</td>
<td>169</td>
</tr>
<tr>
<td>5.7a.</td>
<td>Parallel Network - Overall performance of various methods, $H_0 = I$.</td>
<td>170</td>
</tr>
<tr>
<td>5.7b.</td>
<td>Parallel Network - Overall performance of various methods, $H_0 = J_0^{-1}$.</td>
<td>170</td>
</tr>
<tr>
<td>5.8.</td>
<td>Nested Network - Physical characteristics.</td>
<td>174</td>
</tr>
</tbody>
</table>
Table 5.9a. Nested Network - Initial flow distribution. 175
Table 5.9b. Nested Network - Final flow distribution. 175
Table 5.10a. Nested Network - Convergence characteristics of the NR method. 176
Table 5.10b. Nested Network - Convergence characteristics of the BR method. 178
Table 5.10c. Nested Network - Convergence characteristics of the DFP method. 180
Table 5.10d. Nested Network - Convergence characteristics of the BFGS method. 182
Table 5.11a. Nested Network - Overall performance of various methods, $H_0 = I$. 184
Table 5.11b. Nested Network - Overall performance of various methods, $H_0 = J^{-1}$. 184
Table 5.12a. Parallel Network - Initial flow distribution, (5 metres). 185
Table 5.12b. Parallel Network - Final flow distribution, (5 metres). 185
Table 5.13. Parallel Network - Overall performance of various methods, $H_0 = I$. 185
Table 5.14a. Parallel Network - Initial flow distribution, (1000 metres). 186
Table 5.14b. Parallel Network - Final flow distribution, (1000 metres). 186
Table 5.15. Parallel Network - Overall performance of various methods, $H_0 = I$. 186
Table 5.16. Parallel Network - Initial flow distribution. 189
Table 5.17a. Parallel Network - Overall performance of various methods, $H_0 = I$. 189
Table 5.17b. Parallel Network - Overall performance of various methods, $H_0 = J^{-1}$. 189
Table 5.18. Nested Network - Initial flow distribution. 190
Table 5.19a. Nested Network - Overall performance of various methods, $H_0 = I$. 190
Table 5.19b. Nested Network - Overall performance of various methods, $H_0 = J^{-1}$. 190
Table 5.20. Parallel Network - Differencing interval, $h$, for partial derivatives. 192
Table 5.21a. Overall performance evaluated with near initial approximation, $H_0 = I$. 195
Table 5.21b. Overall performance evaluated with near initial approximation, $H_0 = J^{-1}$. 195
Table 5.21c. Overall performance evaluated with poor initial approximation, $H_0 = I$. 196
Table 5.21d. Overall performance evaluated with poor initial approximation, $H_0 = J_0^{-1}$.

Table 5.22. Culvert Network - Physical characteristics.

Table 5.23. Culvert Network - Flow distribution.

Table 5.24. Culvert Network - Convergence characteristics, $H_r = 1.50$.

Table 5.25a. Culvert Network - Convergence characteristics with line searches, $H_r = 1.50$.

Table 5.25b. Culvert Network - Convergence characteristics with line searches, $H_r = 1.50$.

Table 5.25c. Culvert Network - Convergence characteristics with line searches, $H_r = 1.50$.

Table 5.25d. Culvert Network - Convergence characteristics with line searches, $H_r = 1.50$.

Table 5.26. Culvert Network - Total number of function evaluations, $H_r = 1.50$.

Table 5.27. Culvert Network - Convergence characteristics, $H_r = 1.65$.

Table 5.28a. Culvert Network - Convergence characteristics with line searches, $H_r = 1.65$.

Table 5.28b. Culvert Network - Convergence characteristics with line searches, $H_r = 1.65$.

Table 5.28c. Culvert Network - Convergence characteristics with line searches, $H_r = 1.65$.

Table 5.28d. Culvert Network - Convergence characteristics with line searches, $H_r = 1.65$.

Table 6.1. Classification of side-weir flow.

Table 6.2. Equations for the analysis of side-weir flow.

Table 6.3. Equations for estimating longitudinal velocity component, $U$.

Table 6.4. Known Inflow Network - Physical characteristics, Type-1 flow.

Table 6.5a. Known Inflow Network - Initial flow distribution, Type-1 flow.

Table 6.5b. Known Inflow Network - Final flow distribution, Type-1 flow.

Table 6.6. Known Inflow Network - Convergence characteristics, Type-1 flow.
Table 6.7. Overall performance of various solution methods. 252
Table 6.8. Known Outflow Network - Physical characteristics, Type-1 flow. 256
Table 6.9a. Known Outflow Network - Initial flow distribution, Type-1 flow. 257
Table 6.9b. Known Outflow Network - Final flow distribution, Type-1 flow. 257
Table 6.10. Known Outflow Network - Convergence characteristics, Type-1 flow. 258
Table 6.11. Overall performance of various solution methods. 259
Table 6.12. Known Outflow Network - Physical characteristics, Type-2 flow. 264
Table 6.13a. Known Outflow Network - Initial flow distribution, Type-2 flow. 265
Table 6.13b. Known Outflow Network - Final flow distribution, Type-2 flow. 265
Table 6.14. Known Outflow Network - Convergence characteristics, Type-2 flow. 266
Table 6.15a. Known Inflow Network - Mean convergence rate, Type-1 flow. 269
Table 6.15b. Known Outflow Network - Mean convergence rate, Type-1 flow. 269
Table 7.1. Sensitivity of a channel flow with depths. 281
Table 7.2. Physical characteristics for the parallel open channel network. 284
Table 7.3. Final flow distribution for $H_{outfall} = 0.750$ (m). 294
Table 7.3b. Critical and normal depths for $H_{outfall} = 0.750$ (m). 294
Table 7.4a. Final flow distribution with 0.001 (m) increased in $H_{outfall} = 0.750$ (m). 295
Table 7.4b. Final flow distribution with 0.100 (m) increased in $H_{outfall} = 0.750$ (m). 295
Table 7.5a. Sensitivity of flow re-distribution with 0.001 (m) increased in $H_{outfall} = 0.750$ (m). 296
Table 7.5b. Sensitivity of flow re-distribution with 0.100 (m) increased in $H_{outfall} = 0.750$ (m). 296
Table 7.6a. Theoretical sensitivity of flow re-distribution with various $H_{outfall}$. 297
Table 7.6b. Actual sensitivity of flow re-distribution with various $H_{outfall}$. 297
Table 7.7a. Sensitivity of flow re-distribution, $\varepsilon_r = 10^{-6}$ (m) (all channels). 306
Table 7.8a. Sensitivity of flow re-distribution, $\varepsilon_r = 10^{-6}$ (m) (Channel-1 only). 306
### LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 1.1a</td>
<td>A converging network.</td>
<td>4</td>
</tr>
<tr>
<td>Figure 1.1b</td>
<td>A diverging network.</td>
<td>4</td>
</tr>
<tr>
<td>Figure 1.2a</td>
<td>A parallel open channel network.</td>
<td>5</td>
</tr>
<tr>
<td>Figure 1.2b</td>
<td>A nested open channel network.</td>
<td>5</td>
</tr>
<tr>
<td>Figure 2.1</td>
<td>Section of a water surface profile.</td>
<td>13</td>
</tr>
<tr>
<td>Figure 2.2</td>
<td>A generalised algorithm for standard step method.</td>
<td>22</td>
</tr>
<tr>
<td>Figure 2.3</td>
<td>A generalised algorithm for numerical integration method.</td>
<td>29</td>
</tr>
<tr>
<td>Figure 2.4</td>
<td>A generalised algorithm for adaptive step size control.</td>
<td>44</td>
</tr>
<tr>
<td>Figure 2.5a</td>
<td>Relative truncation error, $h = 0.25$ (m).</td>
<td>53</td>
</tr>
<tr>
<td>Figure 2.5b</td>
<td>Relative truncation error, $h = 2.5$ (m).</td>
<td>53</td>
</tr>
<tr>
<td>Figure 2.5c</td>
<td>Relative truncation error, $h = 25$ (m).</td>
<td>54</td>
</tr>
<tr>
<td>Figure 2.5d</td>
<td>Relative truncation error, $h = 250$ (m).</td>
<td>54</td>
</tr>
<tr>
<td>Figure 2.6a</td>
<td>Fourth-order Runge-Kutta - Goodness of fit of relative error, $h = 250$ (m).</td>
<td>53</td>
</tr>
<tr>
<td>Figure 2.6b</td>
<td>Fourth-order Merson - Goodness of fit of relative error, $h = 250$ (m).</td>
<td>53</td>
</tr>
<tr>
<td>Figure 2.7</td>
<td>A typical plot of $f(x, y)$ against $y/y_e$.</td>
<td>64</td>
</tr>
<tr>
<td>Figure 2.8a</td>
<td>Error bounds on global error, $h = 2.5$ (m).</td>
<td>65</td>
</tr>
<tr>
<td>Figure 2.8b</td>
<td>Error bounds on global error, $h = 25$ (m).</td>
<td>65</td>
</tr>
<tr>
<td>Figure 2.9</td>
<td>Accuracy of a water surface profile with adaptive step size control.</td>
<td>71</td>
</tr>
<tr>
<td>Figure 2.10</td>
<td>Error incurred in the water surface profile with adaptive step size control.</td>
<td>71</td>
</tr>
<tr>
<td>Figure 2.11</td>
<td>Growth in global error with respect to the direction of computation.</td>
<td>73</td>
</tr>
<tr>
<td>Figure 3.1a</td>
<td>Composition of a network with strings.</td>
<td>79</td>
</tr>
<tr>
<td>Figure 3.1b</td>
<td>Composition of a string with modules.</td>
<td>79</td>
</tr>
<tr>
<td>Figure 3.2</td>
<td>A typical hydraulic network.</td>
<td>84</td>
</tr>
<tr>
<td>Figure 4.1</td>
<td>Flow chart for the Newton-Raphson Method.</td>
<td>105</td>
</tr>
<tr>
<td>Figure 4.2</td>
<td>Flow chart for the modified Newton-Raphson Method.</td>
<td>117</td>
</tr>
<tr>
<td>Figure 4.3</td>
<td>Flow chart for the quasi-Newton Methods (BR, DFP and BFGS).</td>
<td>122</td>
</tr>
</tbody>
</table>
Figure 4.4. Line searches in two dimensional space. 125
Figure 4.5. Goldstein conditions (1965). 128
Figure 4.6. Flow chart for a triplet of points. 131
Figure 4.7. Convergence to a minimum by parabolic interpolation. 133
Figure 4.8. Convergence to a minimum by golden section search. 135
Figure 4.9. Flow chart for Golden Section search. 137
Figure 5.1a. Free discharge of a thin-plate weir. 145
Figure 5.1b. Submerged discharge of a thin-plate weir. 145
Figure 5.2a. A critical depth control is formed at upstream. 149
Figure 5.2b. The critical depth control is overridden due to high tailwater level. 149
Figure 5.3a. A hydraulic jump is formed at downstream due to upstream supercritical flow. 150
Figure 5.3b. The hydraulic jump is washed downstream due to low tailwater level. 150
Figure 5.3c. The hydraulic jump is drowned due to high tailwater level. 150
Figure 5.3d. The supercritical flow is completely submerged due to high tailwater level. 150
Figure 5.4. Classification of culvert flow. 153
Figure 5.5. Algorithm for identifying culvert flow. 155
Figure 5.6. A parallel open channel network. 164
Figure 5.7. A nested open channel network. 174
Figure 5.8. A culvert network. 202
Figure 5.9. Culvert Network - A plot of function evaluations against \(|e|\), \(H_r = 1.50\). 202
Figure 5.10a. Culvert Network - Illustration of convergence trend, \(H_r = 1.50\). 213
Figure 5.10b. Culvert Network - Illustration of difficulties in convergence, \(H_r = 1.65\). 213
Figure 6.1a. Network comprises open channels. 222
Figure 6.1b. Network comprises a side-weir. 222
Figure 6.2. Section of a water surface profile with side discharge. 223
Figure 6.3a. A side-weir network. 235
Figure 6.3b. Equivalent network for the side-weir system. 235
Figure 6.4. Flow chart for the external trial-and-error procedure. 226
Figure 6.5a. A known constant inflow network. 240
Figure 6.5b. Equivalent network for the known constant inflow system. 240
Figure 6.6a. A known constant outflow network. 242
Figure 6.6b. Equivalent network for the known constant outflow system. 242
Figure 6.7. A hydraulic jump is formed at downstream because of Type-2 flow. 267
Figure 6.8. Exponent $e$ for virtual string characteristics. 270
Figure 7.1. Non-Dimensional plot of water surface profile, $B/2y_e = 1.0$. 278
Figure 7.2. A parallel open channel network. 284
Figure A.1. Head-losses evaluated with the initial approximation $Q_o$. A-4
LIST OF SYMBOLS

A  Cross sectional area.
B  Channel bed width.
C  Chézy friction coefficient.
C_d Coefficient of discharge.
E  Specific energy.
Fr Froude number.
H  Total energy head.
H_c Height of a culvert.
H_r Head ratio.
H_w Height of a weir.
N_f Number of function evaluations.
P  Wetted perimeter.
Q  Discharge or flow rate.
R  Hydraulic radius.
Re Reynolds number.
S_o Bed slope.
S_f Friction slope.
\bar{S}_f Mean friction slope.
S_w Water surface slope.
T  Top width.
V  Mean flow velocity.
U  Longitudinal velocity component.
e_g Global error.
e_m Machine precision.
e_p Propagated truncation error.
e_r Round-off error.
e_t Truncation error.
f_x Partial derivative with respect to x.
f_y Partial derivative with respect to y.
g  Acceleration due to gravity.
h  Step length for finite differencing.
h_f Head-loss due to friction.
h_v Velocity head.
k, Sand grain roughness.
m Channel side slope.
n Manning friction coefficient.
q Loop flow correction.
r Correlation coefficient.
t Step parameter.
x Distance along channel bed.
y Depth of flow.
y_c Critical depth.
y_n Normal depth.
y_w Crest height.
z Potential head.
\( \mathbb{R} \) Golden ratio.
\( \alpha \) Kinetic energy (Coriolis) coefficient.
\( \Delta \) Finite difference.
\( \delta x \) Finite increment in x.
\( \delta y \) Finite increment in y.
\( \varepsilon_l \) Lower error bound.
\( \varepsilon_u \) Upper error bound.
\( \varepsilon_g \) Error bound on global error.
\( \chi \) Convergence criteria.
\( \eta \) Kinematics viscosity.
\( \eta_s \) Overall performance.
\( \lambda \) Darcy friction coefficient.
\( \mu \) Dynamic viscosity.
\( \rho \) Density of a fluid.
\( \xi \) Prescribed tolerance.

\( H \) Inverted Jacobian matrix.
I Identity matrix.
J Jacobian matrix.
\( \hat{J} \) Jacobian matrix evaluated using finite differencing.
\( J^{-1} \) Inverted Jacobian matrix.
f A function vector.
x An independent vector.
\( \delta \) A vector to hold loop flow correction.
If \(|f|\) Absolute value of \(f\).
\(|f|\) Euclidean norm or length of \(f\).

\(O(\cdot)\) Order of magnitude.
\(f(\cdot)\) A function of.
\(f_j(\cdot)\) Partial derivative respect to \(y\).
\(\vartheta(\cdot)\) Sensitivity of water depth.

Superscript 'T' refers to transposition.
Superscript '-1' refers to matrix inversion.
Subscript '1' refers to upstream section.
Subscript '2' refers to downstream section.
Subscript 'i' refers to the variable associated with section.
Subscript 'j' refers to the variable associated with nodal junction.
Subscript 'l' refers to the variable associated with loop.
Subscript 'm' refers to the variable associated with module.
Subscript 's' refers to the variable associated with string.

\(\zeta\) an ordinate lies somewhere between the interval \([x_l, x_u]\) or \([y_l, y_u]\).
\(\gamma_r\) the coefficient of \(h^{r+1}\) for the \(r\) th-order integration.
CHAPTER 1

INTRODUCTION
INTRODUCTION

1.1 Background

The hydraulic analysis of 'networks' which comprise open channels, pipes and other hydraulic structures linked together in a variety ways have numerous important applications in the field of civil engineering. For example,

- braided river systems
- potable and waste water treatment works
- irrigation water distribution systems
- storm-water drainage systems

The research presented in this thesis was prompted by the earlier development of software for the hydraulic analysis of wastewater works (CHAT, 1987). The objective of the software was to determine the flow distribution within complex networks containing a variety of hydraulic structures, pipes and channels. This work developed a number of important concepts in building up the model of a treatment plant using hydraulic 'building blocks' or 'modules', as defined in detail in Chapter 3. The particular difficulty encountered during the development of the software was in forcing the network analysis to converge to a solution for complex networks. The difficulties were assigned to two main areas.

i) The problems of dealing with open channels flows where head-loss is a function of both discharge and water level; this is explained in more detail below.
ii) The calculation of 'updating' flow correction factors to improve initial assumptions about flow distribution.
The research therefore focuses on these broad areas, namely:

i) An exploration of water surface profile analysis for open channel flow.

Networks which are dendritic, either converging or diverging (see Figures 1.1a and 1.1b), do not present undue difficulties of analysis. However, those (see Figures 1.2a and 1.2b) which contain channels in parallel and/or loops give rise to certain problems. Comparison with a pipe-network is useful in helping to illustrate these problems; for a pipe, the relationship between head-loss and discharge is uniquely defined by an equation of the general form

\[ h_f = kQ^n \]

in which \( k \) and \( n \) are constants. The problem with open channels is that the flow is characterised by its free surface, subject to atmospheric pressure. It means that the equation describing the condition of the flow involves both discharge and water depth

\[ h_f = f(Q, y) \]

This additional water depth gives rise to an extra degree of freedom in the governing equation meaning that the head-loss cannot be generalised in terms of physical characteristics of the channel alone. As a result of this, the determination of the resulting flow distribution and head-loss in such systems are far more complex than the analogous problem for pipe networks. This is because the head-loss is not only related to both discharge and water depth, but also depends upon the controls operating within the network.

Network models usually generate a set of simultaneous equations to characterise the flow behaviour of the network. The steady state (independent of time) principles of flow continuity and conservation of energy can be used to obtain a solution pertaining to the distribution of flows and nodal heads in a complex network. If the equations are formulated in terms of head-losses (flow resistance), then the derived set of equations is known as node equations; if the equations are formulated in terms of flow rates, then the derived set of equations is known as loop equations.
Figure 1.1a. A converging network.

Figure 1.1b. A diverging network.
Figure 1.2a. A parallel open channel network.

Figure 1.2b. A nested open channel network.
Regardless of which principle is adopted for the formulation, the system equations derived are usually non-linear; they cannot be solved directly, and must be solved iteratively. The approach to the solution usually involves an iterative scheme in which successive linearisation of the problem leads to a series of the approximations of increasing accuracy as the final solution is approached. Traditional non-linear solutions using loop formulation have produced better conditioned equations than the node formulation. This means that the loop formulation exhibits accelerated convergent behaviour which requires comparatively fewer iterative steps to achieve the solution. However, a thorough understanding of the properties of loop formulation is necessary in order to exploit the most effective way of solving a hydraulic network.

Once the system of equations have been set up, a solution needs to be obtained. A method which is commonly used for solving a set of simultaneous non-linear equations is the Newton-Raphson method. The concept of the method is simple, and it has a desirable rate of convergence which is classified as second-order. This tremendous convergent potential of the Newton-Raphson method has led to wide acceptance of its applications to many network problems, including electrical and hydraulic systems. However, the method suffers from three disadvantages; firstly, it requires partial derivatives of the objective functions; secondly, it requires the initial approximation to be close to the final solution; and thirdly, it requires the inversion of the Jacobian matrix.

The extent to which solution methods will cope with flow equations which are ill-behaved owing to the complexities of open channel flows is not known. During the derivation of the flow equations, one important assumption made is that the equations must be smooth and continuous and have a second or higher-order partial derivatives. This assumption can be easily violated in case of open channel networks. Specifically, it occurs when the flow in channel is updated during the course of iterations if the flow require changes from subcritical to supercritical or vice versa. This change in flow regime creates those unwanted discontinuities in the flow equation, termed 'switching' of hydraulic controls. It is envisaged that this phenomenon will create a serious problem in convergence and, at worst, may not converge at all.

More complications arise in the analysis due to variable net inflow and outflow which occur in treatment plants and irrigation systems. Inclusion of side-weir structures in network analysis leads to the problem that part of the design flow is diverted and leaves the network. This diverted flow is a function of flow conditions upstream and
downstream of the side weir, implying that an additional variable is introduced in the system equations. This gives rise to the 'one equation and two unknowns' situation. The way by which the equations are formulated cannot immediately analyse this extra degree of complexity.

1.2 Aims and Objectives

The main objective of the research is to develop a generalised algorithm which enables the final flow distribution throughout a hydraulic network containing open channels and other hydraulic structures to be determined. The algorithm should converge rapidly to a solution. The research methodology involves investigation and comparison of the effectiveness of different methods of analysis and their application to computerised solution. The research is divided into six separate areas which are outlined below:

i) Examine the accuracy of the water surface profile calculated from the gradually varied flow equation for open channels. Computation of a water surface profile involves numerical procedures, meaning that the profile computed will not be exact. Numerical errors will be introduced in the computation, resulting in an incorrect water surface profile. It is envisaged that this will affect the final flow distribution in a network. Therefore, special attention is given to the numerical methods available. Error analysis is included to investigate different types of numerical errors and their significance and ultimately the error bounds. This work is described in Chapter 2.

ii) Derive a methodology for generalised network analysis based on the loop principle. The formulation is developed through the definition of basic hydraulic 'building blocks'. Any hydraulic feature which has a definable relationship between head-loss, flow and other parameters is defined as a 'module'. Modules may be linked together to form 'strings'; for example, a string may be recognisable process unit, such as settlement tank. The resulting formulation gives rise to a set of simultaneous non-linear equations which require solution. Steady state energy principles and flow continuity principles are used to set up a solution procedure. This work is described in Chapter 3.

iii) Investigate solution methods for different networks which will converge rapidly to a solution. The methods investigated are based on Newton-Raphson type procedures.
The Newton-Raphson method (Marquardt, 1963) itself is investigated, along with a number of derivative methods presented by Broyden (Broyden, 1965) for the solution of a set of simultaneous non-linear equations. The approach is based on the iterative scheme of Newton-Raphson method, but it does not requires derivatives to be calculated analytically. The derivatives are approximated and are imbedded in the method. Two further methods chosen for the investigation are the Davidon-Fletcher-Powell method (Fletcher and Powell, 1963) and the Broyden-Fletcher-Goldfarb-Shanno method (Fletcher, 1987). The efficiency of different methods is measured in terms of function evaluations and accuracy rather than the actual computational time. This work is described in Chapters 4 and 5.

iv) Investigate the phenomenon of 'switching' of hydraulic control which may cause discontinuities in the procedures, and the ways in which it can be accommodated. If the methods adopted in iii) above cannot cope with this phenomenon, then a subsidiary method imbedded in the solution in order to enforce convergence may be necessary. In a sense, the subsidiary method optimises the correction to the channel flow (assuming loop formulation is used) and avoids the discontinuities which exist in the flow equations. The effect of enforcing accelerated convergence on the computational effort is an important consideration. This work is described in Chapters 4 and 5.

v) Develop procedures to overcome the variable inflow and outflow problems. For example, such problems are caused by the inclusion of side discharge structures in the network. This will introduce an additional variable to the system of equations. The aim is to derive a method to enable this lateral discharge problem to be solved directly without recourse to a subsidiary trial-and-error procedure. This work is described in Chapter 6.

vi) Investigate the sensitivity of the final distribution of flow with respect to the fluctuation of water levels in the network. A sensitivity analysis yields the rate of change of selected variables with respect to the changes in a single variable of interest. For example, what effect will be imposed on the final flow distribution when the downstream water level is adjusted. This includes the investigation of the effect on the final flow distribution when the water surface profile is incorrectly computed. This work is described in Chapter 7.
Numerical experiments will be described to illustrate the capabilities of various algorithms tested.
CHAPTER 2

COMPUTATION OF WATER SURFACE PROFILE
2 COMPUTATION OF WATER SURFACE PROFILE

2.1 Introduction

Many civil engineering applications such as water treatment works incorporate open channels which require the determination of the water surface profile along the channel length. The water surface profile computations entail determining the water depth along the channel. The basic relationship between depth and distance, under steady state, is described by the dynamic equation of gradually varied flow. There is no exact solution of the equation. Various techniques have been developed in attempting to solve the problem. These techniques include graphical procedures, integration methods with the aid of tables and step-by-step numerical integration methods. Of these numerical procedures, one would find that the direct step method and the standard step method are most commonly used.

With the advent of computer technology and more accessible computers, the computation of gradually varied flow has undergone extensive computerisation. Some of the computational procedures have evolved directly from existing hand calculation methods such as those mentioned above while the other methods have utilised numerical analysis. An important limitation with the numerical methods is their stability, since numerical errors are introduced in the computation procedures. Small errors accumulate and grow into sizeable errors which may result in an erratic water surface profile. If such results are incorporated in an analysis to determine the flow distribution throughout a hydraulic network, it can be shown that a small error in profile depths would incur a significant amount of flow incorrectly distributed throughout the network. This phenomenon will be described later in Chapter 7.
The work presented herein is three fold. Firstly, it examines various numerical solutions to generalise the water surface profile computations; secondly, it investigates various types of numerical errors, their significance and ultimately the error bounds; and finally, it investigates the technique of adaptive step size control to aid computational efficiency.

2.2 Dynamic Equation of Gradually Varied Flow

Computation of water surface profiles based on an integration of the one-dimensional dynamic equation of gradually flow are found in standard hydraulic texts (Chow, 1959; French, 1986). The equation is derived based on the assumption that the head-loss is entirely due to frictional resistance. In order to simplify the derivation of the equation, the following assumptions are also made:

i) The channel is prismatic.
ii) The slope of the channel is relatively small even for a steep channel; that is, the water depth is the same whether it is measured vertically or normal to the bed slope.
iii) The pressure distribution is hydrostatic.
iv) The velocity distribution is constant over the cross section.
v) No abrupt transition occurs either in plan or elevation; that is, the streamlines are assumed parallel.

Consider a section of water flowing along a channel. The total energy head, $H$, at the upstream $i$th section shown in Figure 2.1 is given by

$$H = z + y + \frac{\alpha V^2}{2g}$$  \hspace{1cm} (2.1)

where

$z$ = channel invert specified as datum
$y$ = water depth
$V$ = mean velocity
$\alpha$ = energy coefficient
$g$ = acceleration due to gravity
COMPUTATION OF WATER SURFACE PROFILE

Figure 2.1. Section of a water surface profile.

<table>
<thead>
<tr>
<th>Slope</th>
<th>Zone 1</th>
<th>Zone 2</th>
<th>Zone 3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( y &gt; y_a &gt; y_c )</td>
<td>( y_a &gt; y &gt; y_c )</td>
<td>( y &gt; y_a &gt; y_c )</td>
</tr>
<tr>
<td>Mild Slope</td>
<td>M1</td>
<td>M2</td>
<td>M3</td>
</tr>
<tr>
<td>Steep Slope</td>
<td>S1</td>
<td>S2</td>
<td>S3</td>
</tr>
<tr>
<td>Critical Slope</td>
<td>C1</td>
<td>C2</td>
<td>C3</td>
</tr>
<tr>
<td>Horizontal Slope</td>
<td>-</td>
<td>H2</td>
<td>H3</td>
</tr>
<tr>
<td>Adverse Slope</td>
<td>-</td>
<td>A2</td>
<td>A3</td>
</tr>
</tbody>
</table>

Table 2.1. Water surface profile classification.
Differentiating (2.1) with respect to $x$, the distance measured along the bed, gives

$$\frac{dH}{dx} = \frac{dz}{dx} + \cos(\theta) \frac{dy}{dx} + \frac{\alpha V dV}{g} \frac{dx}{dx}$$

(2.2)

in which the $dH/dx$ describes the change in energy head with respect to distance. If condition ii) holds, $\cos(\theta) = 1$ is substituted into (2.2) and solving for $dy/dx$ gives,

$$\frac{dy}{dx} = \frac{S_o - S_f}{1 - Fr^2}$$

(2.3)

in which $S_o$, $S_f$ and $Fr$ denote bed slope, friction slope and Froude number respectively, and are defined as:

$$S_o = \frac{dz}{dx}$$

$$S_f = \frac{dH}{dx}$$

$$Fr = \sqrt{\frac{\alpha V^2 T}{gA}}$$

where

$T$ = top width

$A$ = cross-sectional area

Equation (2.3) is known as the dynamic equation of gradually varied flow. It describes the longitudinal slope of the water surface with respect to the channel invert; hence, it can be used to characterise the water surface profile for a particular flow. A total of thirteen flow profiles may be classified according to the nature of the channel slope and the zone in which the flow surface lies, and those are summarised in Table 2.1.

One exceptional case which is not included in the table is when $S_f \to S_o$; as a consequence $dy/dx \to 0$. This occurs only when the water depth is approaching the
normal depth. In such a case, the flow is known as *uniform flow*. For uniform flow, the water surface is parallel to the invert implying that the specific energy

\[ E = y + \frac{\alpha V^2}{2g} \]  

(2.4)

is preserved within the flow. Head-loss is balanced by the difference in bed level in the direction of the flow.

Since \( S_o \) and \( S_r \) are coincident for uniform flow, \( S_r \) may be evaluated by a number of methods: i) *Chézy friction formula* (1768)

\[ S_r = \frac{V^2}{C^2 R} \]  

(2.5a)

in which \( C \) is known as *Chézy coefficient*; ii) *Manning friction formula* (1889)

\[ S_r = \frac{n V^2}{R^{4/3}} \]  

(2.5b)

in which \( n \) is known as *Manning coefficient*; iii) *Darcy-Weisbach friction formula* (1850)

\[ S_r = \frac{\lambda V^2}{8gR} \]  

(2.5c)

in which \( \lambda \) is given by the *Colebrook-White transition formula* (1939)

\[ \frac{1}{\sqrt{\lambda}} = -2\log\left( \frac{k_r}{14.8R} + \frac{2.51}{Re\sqrt{\lambda}} \right) \]  

(2.5d)

where

- \( k_r \) = surface roughness
- \( Re \) = Reynolds number
- \( R \) = hydraulic radius
A fuller discussion of $S_f$ may found in standard hydraulic texts (Chow, 1959; Chadwick and Morfett, 1986; French, 1986).

Note that (2.3) is an ordinary differential equation for which there is no general explicit solution. However, some explicit solutions do exist for a very restrictive circumstance. The first solution known to exist dates back many years to the Bresse method (1860). It was developed for wide rectangular channels, and utilised the Chézy formula. With these assumptions, (2.3) can be re-arranged in terms of normal depth $y_n$ and critical depth $y_c$ to give

$$\frac{dy}{dx} = S_o \left(1 - \frac{y_n}{y} \right)^3 \left(1 - \frac{y_c}{y} \right)^3$$

(2.6)

The direct integration of the differential equation (2.6) was first performed by Bresse and the solution is stated below:

$$x = \frac{y_n}{S_o} \left[ u - \frac{1}{6} \ln \left( \frac{u^2 + u + 1}{(u - 1)^2} \right) - \frac{1}{\sqrt{3}} \tan^{-1} \left( \frac{\sqrt{3}}{2u + 1} \right) \right] + \text{constant}$$

(2.7)

where

$$u = \frac{y}{y_n}$$

This analytical solution is not without its criticisms from a practical point of view. Firstly, the $C$ in Chézy formula is not constant as assumed in Bresse's solution but is a function of water depth; and secondly, the solution is valid for very wide rectangular channels only.

Chow (1959) presented a solution based on the same assumptions as Bresse to solve the gradually varied flow problem. Unlike the Bresse's solution, Chow's solution is not restricted to the use of the Chézy formula, but the Manning formula also be used. From a
practical point of view, the Manning formula is preferred to the Chézy formula because it is more readily applicable than the Chézy formula. The solution of the integration is

\[
x = \frac{y_n}{S_o} \left[ u - \int_0^u \frac{1}{1-u''} \, du + \left( \frac{y_e}{y_n} \right)^M J \int_0^v \frac{1}{1-v'} \, dv \right] + \text{constant} \tag{2.8}
\]

where

\[
\begin{align*}
M &= \text{hydraulic exponent for critical-flow computation} \\
N &= \text{hydraulic exponent for uniform-flow computation} \\
J &= \frac{N}{N-M+1} \\
\nu &= u^{\nu/3}
\end{align*}
\]

When Manning's formula is used, \( M \) and \( N \) are equal to \( \frac{10}{3} \) and 3 respectively; when Chézy formula is used, both \( M \) and \( N \) are equal to 3, and the solution given by (2.8) is identical to that of (2.7). Therefore, (2.8) may be treated as the generalised solution of gradually varied flow problem providing the integrals in (2.8) can be evaluated readily. Unfortunately, the solution of (2.8) relies on numerical integration of the integrals which are tabulated in suitable intervals for a range of values. Whilst these are useful for manual calculation they are not useful for computer applications. First of all, a set of tables must be prepared which are stored in the computer memory for later referencing. Secondly, it is tedious to interpolate \( u \) and \( v \) with the hydraulic exponents in order to find a suitable solution for the integrals.

Pickard (1963) presented an integration which essentially involved a solution of the same mathematical series implied by Chow's method. The series were derived based on a finite series of polylogarithms and polynomials. Although sets of tables are no longer required for the solution of (2.8), evaluation of the finite series during program execution can be very time consuming especially for those slowly converging series.

Gill (1976) presented an integration method entitled to 'the complex conjugate roots' method. The solution is again found to be some form of a series and relies largely on the preparation of coefficients for the complex conjugate roots.
As seen, a number of analytical solutions exist for the gradually varied flow equation. Although the solution is no longer restricted to Chézy friction formula, a wide rectangular channel is still generally assumed. Because of this limitation, the application of numerical procedures is explored in detail.

2.3 Standard Step Method

Instead of directly solving (2.3) for the gradually varied flow problem, the common approach is to apply the principle of energy conservation or the so called the standard step method (Chow, 1959). The standard step method is one of the most widely accepted procedures for finding water depth at a specified position along the channel length. Because of its simplicity it can be adapted and implemented for machine computation, and a number of computer programs have been developed (Chaudhry and Schulte, 1986; Paine, 1992). Unfortunately, the method developed generates an implicit equation meaning that the equation cannot be solved directly, and a trial-and-error procedure is required to achieve a solution.

Consider a subreach of channel with length $\delta x$ as shown in Figure 2.1, over which the total energies are related by

$$H_i = H_{i+1} + h_f + h_e$$  \hspace{1cm} (2.9)

where

- $H_i$ = total energy head at upstream $i$ th section
- $H_{i+1}$ = total energy head at downstream $i+1$th section
- $h_f$ = friction loss
- $h_e$ = eddy loss

With prismatic channels, the term $h_e$ may be neglected in the equation above because the energy loss $h_e$ is usually much smaller than $h_f$. For sufficiently small $\delta x$, the $h_f$ is usually approximated by

$$h_f = 3_f \delta x$$  \hspace{1cm} (2.10)
in which \( \overline{S_f} \) denotes the average friction slope. According to French (1986), a number of formulae can be used to estimate the \( \overline{S_f} \) for the subreach, and those are tabulated in Table 2.2. With definitions of (2.1) and (2.10) and neglecting eddy loss \( h_e \), (2.9) becomes

\[
z_i + y_i + \frac{V_i^2}{2g} = z_{i+1} + y_{i+1} + \frac{V_{i+1}^2}{2g} + \overline{S_f} \delta x
\]

(2.11)

For a given discharge, two unknowns, namely \( y_i \) and \( y_{i+1} \), are involved in the equation above. To solve this equation, a boundary condition is required; when computing the flow conditions from one subreach to another, the water depth must be provided at one end of the subreach in order to complete the computations. For instance, when \( y_{i+1} \) is provided as a boundary condition, then (2.11) can be solved for \( y_i \) or vice-versa.

Because the equation is non-linear in nature, it must be solved iteratively. Graphical procedures and trial-and-error procedures are available (Chow, 1959; French, 1986). These procedures were developed mainly to facilitate hand computations, and hence may not be suitable or even applicable to computer applications.

Equation (2.11) is a non-linear equation of single variable. It can readily be solved iteratively using numerical techniques. Numerical algorithms which are suitable for computer applications are found in Bajpai et al (1974), and Chapra and Canale (1989). The most direct approach is by re-arranging (2.11) into

\[
y_i = y_{i+1} + (z_{i+1} - z_i) + \left( \frac{V_{i+1}^2}{2g} - \frac{V_i^2}{2g} \right) + \overline{S_f} \delta x
\]

(2.12)

The iteration formula above implies that \( y_{i+1} \) is given as boundary condition. Since \( \overline{S_f} \) involves both \( y_i \) and \( y_{i+1} \), a direction solution is impossible, and an iterative scheme such as the method of successive substitution may be used. However, the (2.12) can converge very slowly or even may not converge at all if the value of \( \delta x \) is not chosen sufficiently small. Due to this weakness, other solution methods are needed to accelerate and enforce convergence. Those methods require (2.12) to be re-arranged into a function like

\[
f(y) = H_i - H_{i+1} - \overline{S_f} \delta x = 0
\]

(2.13)
### Average friction slope

<table>
<thead>
<tr>
<th>Methods</th>
<th>Average friction slope</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average Conveyance</td>
<td>( \overline{s}_f = \left( \frac{Q_1 + Q_2}{K_1 + K_2} \right)^2 )</td>
</tr>
<tr>
<td>Average friction slope</td>
<td>( \overline{s}<em>f = \frac{S</em>{f1} + S_{f2}}{2} )</td>
</tr>
<tr>
<td>Geometric mean</td>
<td>( \overline{s}<em>f = \sqrt{S</em>{f1}S_{f2}} )</td>
</tr>
<tr>
<td>Harmonic mean</td>
<td>( \overline{s}<em>f = \frac{2S</em>{f1}S_{f2}}{S_{f1} + S_{f2}} )</td>
</tr>
</tbody>
</table>

Table 2.2. Average friction slope.
in which $f(y)$ denotes the function depending on water depth only. Paine (1992) applied the Newton-Raphson method to solve (2.13) and reported that the method is suitable for all gradually varied flow regimes. The use of the Newton-Raphson method is preferred because it converges considerably faster than its counterpart successive bisection algorithm (Cheney and Kincaid, 1985).

Cautiously executed, the standard step method is an effective approach in computing water surface profiles. This is accomplished by dividing the channel into a number of small subreach sections, and then carrying out the standard step computation described above on each consecutive subreach separately until the entire channel length is covered. The section (or subreach) at which the computation starts is dependent upon the boundary condition given to the problem. When a boundary condition is provided, say at the downstream end of the channel, the computation is started from that end and then proceeds step-by-step in the upstream direction. The general approach to profile computation is summarised in Figure 2.2.

Chaudhry and Schulte (1986) presented a procedure for water surface profile computation which essentially applies the standard step principle and balances the energy between sections. Consequently, equations identical to (2.13) are formed and solved iteratively. Unlike Paine's method, Chaudhry and Schulte's method iterates on the whole set of equations (all subreach sections) simultaneously whereas the Paine's method iterates on separate equations, one at a time.

Consider a channel which is divided in $N$ number of subreach sections. The standard step for each subreach be written as

$$f_i = H_i - H_{i+1} - h_i = 0, \quad i = 1, 2, \ldots, N \tag{2.14}$$

A total of $N$ equations are formed. Equation (2.14) represents a system of $N$ equations but $N+1$ unknowns, namely $y_i$ to $y_{N+1}$. This necessitates an additional equation to obtain a solution of the system. This additional equation is provided by initialising the flow condition at one end of the channel to a particular boundary condition $y_0$. That is to equate
COMPUTATION OF WATER SURFACE PROFILE

Figure 2.2. A generalised algorithm for standard step method.
COMPUTATION OF WATER SURFACE PROFILE

\[ y_0 = y_o \quad (2.15a) \]

or

\[ y_{n+1} = y_o \quad (2.15b) \]

depending upon which end is the control section. With this boundary condition given, (2.14) can be solved readily. Since the equations are non-linear, Chaudhry and Schulte applied the solution system of the Newton-Raphson method, and reported that the solution method is accurate, efficient and can be easily adapted for real-life applications. A full description of the Newton-Raphson method is deferred to Chapter 4.

Conceptually, Chaudhry and Schultes' method is far more efficient than the Paine's method. It is because solving the equations simultaneously can eliminate unnecessary intermediate iterative steps, and thereby reduce computational time considerably. In contrast to the step-by-step approach (Paine's method), it is not without its disadvantages mainly:

i) Partial derivatives of \( f \) are required and needed to assemble in the Jacobian matrix \( J \),

\[
\begin{bmatrix}
    \frac{\partial f_i}{\partial y_j}
\end{bmatrix}, \quad \begin{cases}
    i = 1, 2, \ldots, N \\
    j = 1, 2, \ldots, N
\end{cases}
\]

A total of \( N^2 \) entries are necessary in the matrix. Apparently, evaluation of this matrix can be time consuming especially when the channel is divided into many finite subreach sections in order to obtain an accurate water surface profile. The computational time required to assemble the matrix is proportional to \( N^2 \).

ii) Large computer storage requirements are needed to operate on a large matrix. The matrix is sparse, diagonally banded and its size equals the number of subreach sections generated, that is \( N^2 \) as shown above.

iii) Matrix inversion is needed to operate on the Jacobian matrix. An \( O(N^3) \) of multiplications and additions are needed to perform the inversion. Moreover, the inverted matrix can be singular if the initial approximation far from the solution.
When solving a standard step, there are numerical uncertainties involved in the solution process, specifically the friction loss within the standard step itself. As mentioned, the friction loss is only an approximation for which the exact value cannot be determined. As seen in (2.12), if $S_f$ contains errors, then these errors will be magnified by $\delta x$ and then propagate and grow as the computation proceeds. In fact, it can be proved that the error involved is not simply factored by $\delta x$ but proportional to the order of $\delta x^2$ (Traver and Chadderton, 1990).

The only approach to maintain numerical accuracy is to minimise the error involved in $S_f$ and then chooses sufficiently small $\delta x$ for computations. Unfortunately, this may not be possible. Firstly, it is extremely difficult to minimise the error involved in $S_f$ because it involves an error inherent in the mathematical formulation of the problem which may not be describing the physical situation realistically. Secondly, it is difficult to quantify the size of $\delta x$ without an accurate measure of $S_f$. Although investigations have been carried out by Traver and Chadderton trying to establish the significance of $S_f$ in relation to the type of water surface profile, little has been discussed about the choice of $\delta x$ in relation to $S_f$. Thus, the size of $\delta x$ for computation remains unresolved.

In view of this inadequacy, an alternative approach to the gradually varied flow problem is to solve the gradually varied flow equation directly using standard numerical integration techniques. The solution given is by no means exact but it can be shown that it is very close to exact if the size of $\delta x$ is cautiously chosen. These techniques allow numerical errors to be estimated and then give indication of the size of $\delta x$ so that the numerical errors can be kept under proper control.

2.4 Numerical Error Definitions

Before considering these techniques in detail, it is useful to define the nature of the likely errors. Small errors may grow as the integration proceeds, resulting in errors in the final water surface profiles. If such a profile is incorporated in a network analysis, it will constitute to errors in calculating the distribution of flow. It is therefore relevant to study the behaviour of the error growth as the integration proceeds.
If the solution \( y \) is an approximation to the exact solution \( \hat{y} \), then the difference between \( y \) and \( \hat{y} \) is the error in \( \hat{y} \); that is,

\[
\text{Exact} = \text{Approximation} + \text{Error}
\]

This error, apart from those inherent in mathematical formulation of the problem, arises from the use of an approximate process in finding the solution numerically. Examples include approximating an integral of a function by finite summation of functional values; and finding the solution of a differential equation by replacing the derivatives by approximation to them. These approximate processes contribute various errors to the solution which are described below:

i) **Round-off errors, \( e_r \):** Round-off errors originate from the fact that computing machines only retain a fixed number significant figures. Therefore, arithmetic calculations can never be performed with complete accuracy. Round-off errors increase with the number of arithmetic operations involved in an analysis. If the solution of a problem requires extensive arithmetic operations, each of which is performed using rounded numbers, it is envisaged that the accumulated round-off errors may significantly affect the final result.

Round-off error has a random character which makes it difficult to estimate realistically. In general, the only way to minimise the round-off errors is to increase the number of significant figures of the computing machine. According to Ralston and Rabinowitz (1978), if \( \hat{y} \) is the exact solution and is correctly rounded to \( d \)-decimal-place number \( y^{(d)} \), then the round-off error is

\[
|e_r| = |\hat{y} - y^{(d)}| \leq \frac{1}{2} \times 10^{-d}
\]

ii) **Local truncation errors, \( e_t \):** Local truncation errors arise from the replacement of an infinite or infinitesimal process by a finite mathematical formula whereby the exact solution is approximated. In order to gain insight into the properties as well as behaviour of such errors, Taylor series expansion is widely used in numerical methods to express functions in an approximation fashion.
If the function $f(x_{i+1})$ describing the behaviour of the problem has continuous derivatives, then it can be represented by the Taylor series expansion about $x_i$ the independent variable, to give

$$f(x_{i+1}) = f(x_i) + h_i f'(x_i) + \frac{h_i^2}{2!} f''(x_i) + \cdots + \frac{h_i^r}{r!} f^{(r)}(x_i) + e_r, \quad r = 0, 1, \ldots \quad (2.17)$$

in which $h_i = x_{i+1} - x_i$; and $e_r$ denotes the local truncation error which is defined by

$$e_r = \frac{h_i^{r+1}}{(r+1)!} f^{(r+1)}(\zeta) \quad (2.18)$$

in which $\zeta$ lies somewhere between the interval $[x_i, x_{i+1}]$. When the series is expanded up to and including the term $h_i^r f^{(r)}(x_i)$, the expansion is said to be of order-$r$. It follows that the truncation error due to the $r+1$ and higher-order derivative terms are left out in the expansion.

If the approximate formula is derived up to and including the $r$th-order term corresponding to the Taylor series, then the formula is also said to be of order-$r$. If such formula is used for an analysis, then a truncation error of order $r+1$ is introduced into the numerical solution. In general, a $r$th-order Taylor series expansion will be exact for a $r$th-order polynomial. For other differential and continuous functions, a finite number of terms will usually not provide an exact estimate. Each additional term will contribute some improvement, however slight, to the approximation. This leads to the assessment of how many terms are needed to approximate the solution, and this requires the estimate of local truncation error.

iii) Propagated truncation error, $e_p$: Propagated truncation errors arise from the approximation produced during the previous steps. At each step of an integration, a local truncation error will be produced. The accumulated truncation error is not simply the sum of the local truncation error because each local truncation error is propagated and may either grow or decay as the computation proceeds. Loosely speaking, an approximate method is unstable if errors introduced into the integration grow at an exponential rate as the computation proceeds; whereas if the errors do not grow as the computation proceeds, then the method is said to be stable. Although the
integration may be stabilised by using sufficiently small $h$, it might have a tendency of increasing round-off error.

iv) *Global truncation error, $e_g$: Global truncation error is the sum of local truncation error and the propagated truncation error, that is,

$$e_g = e_l + e_p$$  \hspace{1cm} (2.19)

It is possible to take this error into account as the computation proceeds in order to establish the bounds on the error incurred in the final solution. However, the resulting bounds are often very conservative; that is, the actual error incurred is unlikely to be near the bounds. Although the bounds on numerical error are of no particular practical use to describe the actual error incurred, they provide insight into the stability of the method.

2.5 Numerical Integration Methods

As the gradually varied flow equation is an ordinary differential equation, a closed form solution is generally impossible. However, a semi-analytical scheme like the Taylor series expansion can be used to integrate the equation providing the higher-order derivative terms of the equation are continuous.

Consider the gradually varied flow equation in its most general form

$$\frac{dy}{dx} = f(x, y)$$  \hspace{1cm} (2.20)

The equation can be solved by integrating between the limits at sections $i$ and $i+1$:

$$\int_{y_i}^{y_{i+1}} dy = \int_{x_i}^{x_{i+1}} f(x, y) \, dx$$  \hspace{1cm} (2.21)
The left-hand side of the above integral can be integrated and evaluated to give

\[ y_{i+1} = y_i + \int_{x_i}^{x_{i+1}} f(x, y) \, dx \]  

(2.22)

Denoting

\[ \Phi(x_i, y_i) h_i = \int_{x_i}^{x_{i+1}} f(x, y) \, dx \]

equation (2.22) becomes

\[ y_{i+1} = y_i + \Phi(x_i, y_i) h_i \]  

(2.23)

Equation (2.23) represents a solution of the (2.20). It is linearised and \( \Phi(x_i, y_i) \) corresponds to the slope of that equation. The idea is to predict \( y_{i+1} \) over the distance \( h_i \) on the basis of a prior value \( y_i \) and the slope estimate of \( \Phi(x_i, y_i) \) at the point \((x_i, y_i)\) of interest. As shown, \( y \) appears on both side of the equation. It implies that (2.23) is recursive and must be solved repetitively (step-by-step) in order to trace out the trajectory of the solution.

The basic algorithmic structure of using (2.23) for the computation of a water surface profile is shown in Figure 2.3. Note that the similarity between this and the standard step method (see Figure 2.2); the only difference being the choice of formulation.

Conte and de Boor (1980) showed that if \( f(x, y) \) is sufficiently differentiable with respect to \( x \) and \( y \), then (2.20) possesses a unique solution. Suppose \( y_{i+1} = y(x_{i+1}) \) is the exact solution of (2.20), the Taylor series expansion of which about \( x_i \) is

\[ y(x_{i+1}) = y(x_i) + h_i y'(x_i) + \frac{h_i^2}{2!} y''(x_i) + \cdots + \frac{h_i^r}{r!} y^{(r)}(x_i), \quad r = 1, 2, \ldots \]  

(2.24)

where

\[ h_i = x_{i+1} - x_i. \]
COMPUTATION OF WATER SURFACE PROFILE

Figure 2.3. A generalised algorithm for numerical integration method.

Start

Divide the Channel into N number of subreach sections

\[ i = 0 \]

\[ y_0 = y_p \]

\[ y_{i+1} = y_i + \Phi(x_i, y_i)h \]

\[ i = i + 1 \]

Is \( i < N \)?

End

Yes

No
Note that \( y \) itself is a function of \( x \), and the total derivatives of (2.20) are given by

\[
y' = f(x, y) \\
y'' = f_x + f_y f
\]

in which \( f_x \) and \( f_y \) denotes the partial derivatives with respect to \( x \) and \( y \) respectively.

By comparing the (2.24) with (2.23), \( \Phi(x_i, y_i) \) is found to be

\[
\Phi(x_i, y_i) = \sum_{i=1}^{r} \frac{h_{i-1}^{(i-1)}}{m!} f^{(i-1)}(x_i, y_i) + O(h')
\]

(2.25)

It has already been noted that the higher-order derivatives can be laboriously difficult to derive unless the \( f(x, y) \) is very simple. In such a case, one would limit the number of derivative terms in the expansion to a reasonable number in order to yield a reasonable approximation of \( y_{i+1} \). When higher-order derivative terms are neglected from the expansion, it is necessary to take smaller integration steps over the interval \([x_i, x_{i+1}]\) for which \( y_{i+1} \) is reasonable approximated.

Apparently, the method relies largely on \( \Phi(x_i, y_i) \) to predict where the solution lies. For sufficiently small \( h \), the errors involved in the prediction usually decrease as the order-\( r \) in \( \Phi(x_i, y_i) \) increases. Due to the complexity of the total derivatives, it would be impractical to evaluate those derivatives analytically. In order to achieve a greater accuracy, and at the same time avoid the need to find the total derivatives, numerical methods to estimate \( \Phi(x_i, y_i) \) can be adopted. The underlying strategy is to evaluate \( f(x, y) \) at selective points within each interval \([x_i, x_{i+1}]\). Adopting this strategy, a whole family of methods conforming to (2.23) can be derived. The only difference between methods is the way in which \( \Phi(x_i, y_i) \) is estimated.

### 2.6 Trapezoidal Method

Prasad (1970) adopted the *Trapezoidal method* for the computation of water surface profiles. The method was reported to be efficient because no initial computation was required to determine the shape of the profile and the 'proper' direction of computation.
The method was thoroughly tested and found to have no limitation on its use within the realm of applicability of the gradually varied flow problem.

The Trapezoidal method averages the slopes at points \((x_{i+1}, y_{i+1})\) and \((x_i, y_i)\) is

\[
y_{i+1} = y_i + \frac{h_i}{2} \left[ f(x_{i+1}, y_{i+1}) + f(x_i, y_i) \right]
\]

(2.26)

in which \(h_i = x_{i+1} - x_i\). Note that the equation is implicit because two unknowns, \(y_{i+1}\) and \(f(x_{i+1}, y_{i+1})\), appear on either side of the equation. It means that an iterative procedure must be incorporated to obtain a solution. Solution of the equation by the method of successive substitutions proceeds from an initial approximation of \(y_i^{(0)}\). The estimated \(y_i^{(0)}\) (left-hand side) is subsequently substituted into the right-hand side of the equation resulting in a series of estimations. Every successive estimate is closer to its final value as \(n \to \infty\). The procedure continues until two successive estimates agree within the prescribed tolerance.

Ideally, such a procedure should converge under all likely conditions; more frequently, it is convergent only when certain conditions are satisfied. Specifically, the step size \(h_i\) needs to be chosen sufficiently small. To establish the range of (or ultimately the bounds on) \(h_i\), McBean and Perkins (1975a) established the following convergence test.

\[
\frac{h_i L_i}{2} < 1
\]

(2.27)

where

\[
L_i = \left| f_y(x_{i+1}, y_{i+1}^{(n)}) \right|
\]

(2.28)

for some constant \(L_i > 0\). Its value can be found by direct differentiation of (2.28) with respect to \(y\). When the gradually varied flow equation is used, (2.28) becomes

\[
L \geq \left| \frac{1}{1 - Fr^2} \left( 2fFr \frac{\partial Fr}{\partial y} - \frac{\partial S_f}{\partial y} \right) \right|
\]

(2.29)
in which \( f = f(x, y) \). It follows that (2.29) is an upper bound on the values of \( h \). This means that (2.26) is convergent providing (2.27) and (2.29) are satisfied.

Although (2.26) is guaranteed to converge, it has two criticisms from a practical point of view. Firstly, the rate of convergence of the procedure can be intolerably slow. This occurs when \( \frac{1}{2} h L_i \to 1 \) as \( y \to y_e \), implying that the convergence scheme is unstable near such a water depth. Secondly, the procedure may converge to an incorrect water surface profile because the prediction can fall on the opposite side of the \( y_e \) or \( y_a \) when operating unrestrictedly close to them.

In order to overcome these problems simultaneously, McBean and Perkins (1975a) presented a convergence scheme to ensure the Trapezoidal method is convergent even though the criteria (2.27) is violated. The approach is to re-arrange the (2.26) into a function like

\[
f(y_{i+1}) = y_{i+1} - y_i - \frac{h_i}{2} [f(x_{i+1}, y_{i+1}) + f(x_i, y_i)] = 0
\]

in which \( f(y_{i+1}) \) denotes a function of \( y_{i+1} \). Note the similarity between (2.30) and (2.13). The idea is to repetitively sample the values of \( y_{i+1} \) using the technique of successive bisection until \( f(y_{i+1}) = 0 \) or some prescribed tolerance is satisfied. Apparently, the scheme guarantees to convergence; the solution is definitely bounded so that the possibility of transition to fictitious profiles is excluded. Unfortunately, the rate of convergence can be slow, since the approximations are only improved by a factor of two for each successive iteration. This means that the number of iterations \( n \) required to reduce the initial span, \( y_{i+1} - y_{i+1} \), to the prescribed tolerance \( \xi \) is

\[
n > \frac{1}{\ln 2} \ln \left( \frac{\bar{y}_{i+1} - \bar{y}_{i+1}}{\xi} \right)
\]

(Press et al, 1988) in which \( \bar{y}_{i+1} \) and \( \bar{y}_{i+1} \) denote the upper and lower bounds on the solution \( y_{i+1} \). This also corresponds to the number of evaluations of \( f(x_{i+1}, y_{i+1}) \) to converge. Although faster convergence schemes such as the Newton-Raphson method may be used, they require derivative measurements of \( f(x_{i+1}, y_{i+1}) \) with respect to \( y_{i+1} \) in order to converge rapidly.
2.6.1 Local Truncation Error in Trapezoidal Method

As defined in Section 2.4, local truncation error is the difference between the exact solution and the approximation solution. Therefore, the local truncation error \( e_t \) for the Trapezoidal method can be found by subtracting (2.22) from (2.26), and expanding the integral into a Taylor series. From which, the local truncation error can be shown to be

\[
e_n = -\frac{h^3}{12} f''(\zeta, y(\zeta))
\]

(Bajpai et al, 1974) in which \( f'' \) is the second-order total derivative of \( f \). Thus, the local truncation error is said to be of order three. Unfortunately, the exact position of \( \zeta \) is not precisely known but it is known that \( \zeta \) lies somewhere in the interval \([x_i, x_{i+1}]\). Since the exact value of the \( e_n \) is impossible to obtain, one would seek a value of \( \zeta \) to maximise \( e_n \) in order to provide an upper bound on the truncation error.

2.6.2 Global Truncation Error in Trapezoidal Method

Local truncation error is estimated at discrete points only. No information is given about the error which accumulates as the computation proceeds over a number of steps. This means that the accuracy of the solution is not only affected by the local truncation error \( e_t \) but also the errors propagated \( e_p \) from previous steps. As defined in Section 2.4, the sum of these two errors is known as the global truncation error \( e_g \). McBean and Perkins (1975b) showed that the bounds on this global error is

\[
e_{g+i} = \frac{1 + \frac{hL_i}{2}}{1 - \frac{hL_i}{2}} e_{gi} + \frac{\varepsilon \eta}{1 - \frac{hL_i}{2}}
\]

(2.33)
in which \( \varepsilon \eta \) is the bound on the global error at the \( i \)th step; given that the following conditions are met

\[
e_n > |e_n|, \quad L_i > \left| f_y(x_i, \zeta_i) \right|, \quad 1 > \frac{1}{2} h_i L_i
\]
COMPUTATION OF WATER SURFACE PROFILE

Note that round-off error is not included in the equation above because it is usually comparatively much smaller than the truncation error. If necessary, the round-off error can be included in \( e_n \) in the inequality above. Equation (2.33) is known as the recursive error bound which defines the growth of the global error at the \( i+1 \)th step due to the contributions of the propagated errors and the local truncation error at the \( i \)th step. Intuitively, the coefficient in (2.33)

\[
\left[ 1 + \frac{h_i L_i}{2} \right] > 1 \\
\left[ 1 - \frac{h_i L_i}{2} \right]
\]

is always greater than unity meaning that the error will accumulate and grow with step \( i \). The larger the value of the coefficient, the quicker the error will grow. Furthermore, it is realised that (2.33) is a first-order non-homogeneous difference equation\(^\dagger\) (Bajpai et al., 1977), and its solution is

\[
e_{i+1} = \left( e_{i0} + \frac{e_L}{hL} \right) - \frac{e_L}{hL}
\]

(2.34)
given that the following conditions are met

\[ e_r \geq e_n \hspace{10pt} L \geq L_i \hspace{10pt} h = h_i \]

The parameter \( e_{i0} \) in (2.34) denotes the error incurred at the beginning of the integration. It is usually taken to be zero. This is true if the given boundary condition is exact. Equation (2.34) is the upper bound on the global error. It would be found more

\(\dagger\) The solution of the first-order homogeneous equation

\[
\xi_{i+1} = C_1 \xi_i + C_0
\]

is

\[
\xi_t = C_1 \xi_0 + C_0 \frac{(C_1 - 1)}{(C_1 - 1)} \hspace{10pt} C_1 \neq 0 \\
\xi_t = \xi_0 + C_0 t \hspace{10pt} C_1 = 0
\]

where \( C_0 \) and \( C_1 \) are constants.
COMPUTATION OF WATER SURFACE PROFILE

conservative that the recursive bound since $\varepsilon_i$ and $L$ as derived bound the values of $\varepsilon_{i+1}$ and $L_{i+1}$ respectively for all steps $i = 1, 2, \ldots$. In such a case, the bound will grow beyond realistic values and serve little use in describing the magnitude of the actual error. However, the implication of this is to establish the stability of the trapezoidal method rather than to provide a realistic a priori error estimate.

2.7 Runge-Kutta Method

Humpidge and Moss (1971) adopted the fourth-order Runge-Kutta method to solve the gradually varied flow problem instead of using McBean and Perkins' scheme (the Trapezoidal method). The mode of operation of the two methods is identical except for the formula used for the integration. Therefore, the algorithm presented in Figure 2.3 is applicable to the fourth-order Runge-Kutta method too.

In a sense, the fourth-order Runge-Kutta method is preferred because firstly, it is expected to provide more accurate results than the second-order trapezoidal method for the same integration with the same step size $h_i$; and secondly, the formula is explicit so that it does not require intermediate iterative steps to converge to the solution. One single application of the fourth-order Runge-Kutta method requires four function evaluations rather than $n$ function evaluations (see (2.31)) required by the McBean and Perkins' scheme. However, as in Prasad's method, the prediction may fall on the opposite side of the normal depth if $h_i$ is chosen too large. This problem can be easily be overcome by reducing the size of $h_i$. An increase in function evaluations is therefore inevitable.

The Runge-Kutta method exists in many forms. The basis of all Runge-Kutta methods is to express the difference between the values of $y$ at $x_{i+1}$ and $x_i$ as

$$y_{i+1} = y_i + h_i \sum_{i=1}^{\rho} w_i k_i$$ \hfill (2.35)

in which $w_i$ is constant and

$$k_i = f\left(x_i + \alpha_i h_i, y_i + h_i \sum_{m=1}^{l-1} \beta_{im} k_m\right)$$ \hfill (2.36)
with $\alpha_i = 0$. The equations above represent a whole family of methods to which the combinations of $w_l$, $\alpha_l$ and $\beta_m$ are infinite. The objective is to determine $w_l$, $\alpha_l$ and $\beta_m$ so that (2.35) has the desired properties which make the coefficients of $h^p$ in the Taylor series expansion of both side of (2.35) about $(x_l, y_l)$ identical for $p = 1, 2, \ldots, r$. The detailed derivation of particular formula can be found in numerical texts such as Ralston and Rabinowitz (1978). The classic fourth-order Runge-Kutta formula is

$$y_{i+1} = y_i + \frac{h}{6} (k_1 + 2k_2 + 2k_3 + k_4)$$

(2.37)

where

$$
\begin{align*}
  k_1 &= f(x_i, y_i) \\
  k_2 &= f(x_i + \frac{1}{2}h_i, y_i + \frac{1}{2}h_i k_1) \\
  k_3 &= f(x_i + \frac{1}{2}h_i, y_i + \frac{1}{2}h_i k_2) \\
  k_4 &= f(x_i + h_i, y_i + k_3)
\end{align*}
$$

If higher accuracy of the approximated solution is required, a higher-order Runge-Kutta formula can be used, such as the fifth-order Butcher method (Chapra and Canale, 1989)

$$y_{i+1} = y_i + \frac{h}{90} (7k_1 + 32k_2 + 12k_4 + 32k_3 + 7k_6)$$

(2.38)

where

$$
\begin{align*}
  k_1 &= f(x_i, y_i) \\
  k_2 &= f(x_i + \frac{1}{2}h_i, y_i + \frac{1}{2}h_i k_1) \\
  k_3 &= f(x_i + \frac{1}{2}h_i, y_i + \frac{1}{2}h_i k_2) \\
  k_4 &= f(x_i + \frac{1}{2}h_i, y_i - \frac{1}{2}h_i k_2 + h_i k_3) \\
  k_5 &= f(x_i + \frac{3}{8}h_i, y_i + \frac{9}{32}h_i k_1 + \frac{3}{32}h_i k_4) \\
  k_6 &= f(x_i + h_i, y_i - \frac{9}{32}h_i k_1 + \frac{1}{32}h_i k_2 + \frac{1}{12}h_i k_3 - \frac{1}{32}h_i k_4 + \frac{5}{16}h_i k_5)
\end{align*}
$$

or the sixth-order Verner method (Conte and de Boor, 1980), which requires eight function evaluations. However, in general, beyond fourth-order schemes the gain in accuracy is offset by additional function evaluations and complexity.
2.7.1 Local Truncation Error in Runge-Kutta Method

Like the trapezoidal method, the fourth-order Runge-Kutta method does not provide an immediate estimate of the local truncation error. However, Romanelli (1958) (Ralston and Wilf, 1960) showed that the theoretical local truncation error in the process is

\[ e_n = \frac{h^5}{1440} \left[ \frac{T^4}{2} - 3(s^2t - st^2 + f_x^2t + 3f_yp + 2f_yST - 4f_x^2T) + 2f_yT^3 \right] \]  

(2.39)

where

\[ P = (Df)^2 \]
\[ T^i = D^i f \]
\[ S^i = D^i f_y \]
\[ D = \frac{\partial}{\partial x} + f \frac{\partial}{\partial y} \]

From which, it can be stated that the local truncation error is \( O(h^4) \). Equation (2.39) involves various partial derivatives which are very tedious to obtain analytically making the equation impractical to use. On the other hand, Lotkin (1951) obtained the bound on \( e_n \)

\[ e_n \geq |e_n| = \frac{73}{720} M_l L_t h^3 \]  

(2.40)

given that the following conditions are met

\[ M_l \geq |f(x, y)|, \quad \frac{L_t^{i+m}}{M_{k-l}^{i+m}} \geq \frac{\partial^{i+m}}{\partial x^i \partial y^m}, \quad 4 \geq (l + m) \]

\( L_t \) and \( M_l \) are some positive constants independent of \( f(x, y) \). Although a bound on the \( e_n \) is available, it serves little use from a practical point of view. Firstly, it is very difficult to find values of \( L_t \) and \( M_l \) for which the conditions hold. Secondly, the bound on the \( e_n \)
can be largely over-estimated since the integration is bounded within the domain of interest.

Instead of using (2.40), another possibility is to use the technique of step halving or doubling (Ralston and Rabinowitz, 1978) in order to obtain a good estimate of the truncation error; this can be applied to any order of Runge-Kutta method. The basis of the technique is to integrate the interval of interest \([x_1, x_{i+1}]\) with one step of size \(h_i\), and then integrate the interval with two steps of size \(\frac{1}{2}h_i\). The difference between the two integration cases provides an estimate of the truncation error.

Consider the truncation error for the general case

\[
e_n = \gamma \cdot h_i^{r+1} + O(h_i^{r+2})
\]  

(2.41)

in which \(\gamma\) is the coefficient of \(h_i^{r+1}\) dependent on \(f(x,y)\); and \(r\) is previously defined as the order of the integration. Assume \(\gamma\) varies little over the interval \([x_i, x_{i+1}]\); the integrations corresponding to the two cases, \(h_i\) and \(\frac{1}{2}h_i\), can be written as

\[
y_h(x_{i+1}) = y(x_{i+1}) + \gamma \cdot h_i^{r+1}
\]

(2.42a)

\[
y_{\frac{h}{2}}(x_{i+1}) = y(x_{i+1}) + 2\gamma \cdot \left(\frac{1}{2}h_i\right)^{r+1}
\]

(2.42b)

in which \(y(x_{i+1})\) is the exact value at \(x_{i+1}\); \(y_h(x_{i+1})\) and \(y_{\frac{h}{2}}(x_{i+1})\) are the approximation solutions calculated with \(h_i\) and \(\frac{1}{2}h_i\) respectively. The term \(2\gamma \cdot \left(\frac{1}{2}h_i\right)^{r+1}\) in (2.42b) is the error accumulated in two steps; this implies that \(\gamma\) varies little over the interval \([x_i, x_{i+1}]\). Subtracting (2.42a) from (2.42b) and solving for the \(2\gamma \cdot \left(\frac{1}{2}h_i\right)^{r+1}\) term in (2.42b) yields

\[
2\gamma \cdot \left(\frac{h_i}{2}\right)^{r+1} = \frac{y_h(x_{i+1}) - y_{\frac{h}{2}}(x_{i+1})}{1 - 2^{r}}
\]

(2.43)

Thus, (2.43) provides a computable estimate of the truncation error in the approximation \(y_{\frac{h}{2}}(x_{i+1})\). With the case of the fourth-order Runge-Kutta method, \(r = 4\), the truncation error approximates to
\[
\varepsilon_n = \left| \frac{y_h(x_{i+1}) - y_{h/2}(x_{i+1})}{15} \right| \quad (2.44)
\]

 Apparently, this is a very time consuming way to obtain an error estimate because it requires one full-step \(h\) and two half-steps \(\frac{1}{2}h\) in order to estimate the local truncation error. Because the \(y_{h/2}(x_{i+1})\) obtained is more accurate than that of \(y_h(x_{i+1})\), the former will be used instead of the latter. Therefore, for the fourth-order Runge-Kutta method, it requires four additional function evaluations per step, that is a 50% increase in computational time.

Aside from the technique of step halving or doubling, an alternative approach which allows a direct estimate of the truncation error exists. This approach requires two Runge-Kutta methods of order \(r\) and \(r+1\) to predict two approximate solutions. The difference between the two approximations provides an estimate of the truncation error. However, this approach relies largely on the careful choice of \(\beta_{in}\) when deriving the \(r\)th-order method and the method of order \(r+1\). A variety of methods following this approach such as the Fehlberg method are available (Chapra and Canale, 1989). This method requires six function evaluations and the resulting estimate is only accurate to the order of four, which is same as the fourth-order Runge-Kutta method. However, if certain assumptions are made on the form of the \(f(x, y_i)\), then the Merson method (1957) and the Scraton method (1964) may be used. These methods achieve the same accuracy \(O(h^4)\) but require five function evaluations only. For illustration, the Merson method is stated below:

\[
y_{i+1} = y_i + \frac{h}{6}(k_1 + 4k_3 + k_2) \quad (2.45)
\]

where

\[
k_1 = f(x_i, y_i)
k_2 = f(x_i + \frac{1}{2}h, y_i + \frac{1}{2}hk_1)
k_3 = f(x_i + \frac{1}{2}h, y_i + \frac{1}{2}hk_1 + \frac{1}{2}hk_2)
k_4 = f(x_i + \frac{1}{2}h, y_i + \frac{1}{2}hk_1 + \frac{1}{2}hk_3)
k_5 = f(x_i + h, y_i + \frac{1}{2}hk_1 - \frac{1}{2}h k_2 + \frac{1}{2}h k_3 + 2hk_4)
\]
and the truncation error in this process is given by

\[ e_n = \frac{h}{30} (2k_1 - 9k_2 + 8k_3 - k_4) \]  

(2.46)

The error estimated by (2.46) will be exact if \( f(x, y) \) is of the form

\[ f(x, y) = ax + by + c \]  

(2.47)

in which \( a, b \) and \( c \) are constants. A point of interest in connection with these processes is that \( e_n \) can be added to the \( y_{i+1} \) after each step to improve the order of accuracy of the solution by one.

It is worth noting that if the differential equation is of the form (2.47), then the immediate estimate of truncation error of the fourth-order Runge-Kutta method is

\[ e_n = \frac{8h_1 (k_3 - k_s)^3}{15(k_4 - k_3)^2} \]  

(2.48)

Of these error predicting equations, it is important to know about their limitations, applicabilities and most important whether the equations provide useful predictions of the actual errors, even though the differential equation does not strictly conform to (2.47). This must be resolved through numerical experiments.

### 2.7.2 Global Truncation Error in Runge-Kutta Method

As with the Trapezoidal method, bounds on the propagated error will be proved important. With regard to the stability of the fourth-order Runge-Kutta method, Carr (1958) and Romanelli (Ralston and Wilf, 1960) proved the following theorem.
Theorem: Let \( f \) be continuous, negative, and bounded from above and below throughout a region \( D \) of the \((x, y)\) plane, i.e.,

\[-M_2 < f < M_1, \quad M_2 > M_1 > 0\]

\( \varepsilon_i \) be the absolute value of the maximum error (truncation, round-off, or both) introduced at each step, i.e., the bound of the local error; \( D' \) be a region in which the solution of the difference equation approaches no closer than \( Qh + |\varepsilon_p| \) to \( y \) boundary of \( D \), where

\[
Q \geq \max_{x, y \in D} |f(x, y)| \geq \max \left( |\frac{1}{2}k_1|, |\frac{1}{2}k_2|, |k_3| \right)
\]

and \( \varepsilon_p \) is the error in \( y \) at the \( i \)th step. Then, the Runge-Kutta fourth-order numerical integration procedure has a bound on the total error at the \( i \)th step of

\[
|\varepsilon_p| \leq \frac{2\varepsilon_i}{hM_1}
\]

(2.49)

in the region \( D' \), when the step size

\[
h < \min \left( \frac{M_1}{M_2}, \frac{4M_1^3}{M_2^4} \right)
\]

If \( f \) is allowed to be zero or positive (the unstable case), but bounded throughout \( D \), \( 0 < f < M \), then the error in \( D' \) at the \((i+1)\)th step is

\[
|\varepsilon_p| \leq \varepsilon_i \left( \frac{e^{hM} - 1}{e^{hM} - 1} \right) e^{AM}
\]

(2.50)

and \( \varepsilon_i \), and \( h \) are as given in the stable case \((f < 0)\).
Two errors bounds, the stable bound (2.49) and unstable bound (2.50), are presented in the above theorem. The unstable bound denotes that the integration is so unstable that a sizeable error, including local and propagated truncation errors, will be introduced in the computation as the integration proceeds with step $i$. In fact, the error will grow exponentially as step $i$ increases. Moreover, the larger the size of $h$, the quicker the error will grow. On the other hand, the stable bound indicates that the integration is so stable, the error will not grow and might decay as the integration proceeds. Therefore, it is seen that the bound is independent of the integration step $i$, but dependent of the size of $h$ only. Hence, a constant bound on the error results.

The importance of the error bounds is that they relate the step size and propagated error. Hence, if the partial derivative is known, then the step size can be determined. However, as mentioned previously, it is very difficult to find values of $M_1$ and $M_2$ throughout a region $D$. Even though they are available, the error bounds are expected to be very conservative because the values of $M_1$ and $M_2$ are derived to be applicable to all integration steps.

### 2.8 Adaptive Step Size Control

One important aspect underlying all numerical methods is the choice of the step size $h$ for computation. For sufficiently small $h$, it is believed that negligible errors will be introduced into the final result, although this statement has yet to be justified.

Adaptive step size control is a process of selecting the appropriate step size $h$ for a particular integration. The objective of this process is to find the appropriate step size $h$ for the subsequent integration steps based on numerical errors available at the present stage, in order to achieve certain accuracy in the final solution. A suitable approach to the numerical integration of the gradually varied flow equation is to adapt the step size $h$ as the computations proceed, through investigating the magnitude of the likely errors, as described in the previous sections.

In general, the strategy is to increase the step size if the truncation error is found to be small, and decrease it if the error is found to be large. If, however, the chosen step size is too small, not only will the computational effort be increased considerably, but there is also a possibility that round-off error will dominate the issue.
If the truncation error is estimated for a particular integration, it can be used to decide whether to accept or reject the computed value. It has been shown that the truncation error for a particular numerical method is a function of $h$; this relationship can be used to decide on a new value of $h$ either to repeat the computation or to continue. One such scheme has been suggested by Conte and de Boor (1980) which provides the lower and upper error bounds $e_l$ and $e_u$ respectively. Three outcomes are possible, those are:

i) $e_r > e_u$: In this case the error is too large so that the step size must be reduced, say by a factor of two, and retry the present step.

ii) $e_l < e_r < e_u$: In this case the error is within bounds so that the step size remains unchanged for subsequent steps.

iii) $e_r < e_l$: In this case the computed value is more accurate than required so that the step size may be increased, say by a factor of two, for subsequent steps.

According to Conte and de Boor, if the step size changes are restricted to halving or doubling, then the lower error bound $e_l$ can be set to

$$e_l = \frac{e_u}{2^r/1}$$

(2.51)

in which $r$ is the order of the integration method. The main disadvantage of this approach is that if $e_r > e_u$ is found for the current step, then the size of $h$ for the current step has to be reduced and the integration is repeated. This may not be a problem when the technique is applied to the gradually varied flow equation, because this repetition may occur once or twice at most per integration step, providing the direction of computation being correctly exercised. Thus, the algorithm shown in Figure 2.4 is devised illustrating the mechanism of the adaptive step size control. It must be pointed out that firstly, the algorithm will automatically adjust the appropriate step size for integration even though an initial step size is largely chosen; and secondly, that the algorithm is generalised so that it is suitable for use with any order of Runge-Kutta methods.

One suggestion would be more efficient to use the error from the current step to predict the step size for the subsequent integration steps. One such scheme according to Press et al (1988) is stated below:
Figure 2.4. A generalised algorithm for adaptive step size control.
COMPUTATION OF WATER SURFACE PROFILE

\[ h_{\text{new}} = h_{\text{old}} \frac{\varepsilon_d}{\varepsilon_e} \]  

(2.52)

where

\[ h_{\text{new}} = \text{estimated new step size} \]
\[ h_{\text{old}} = \text{present step size} \]
\[ \varepsilon_d = \text{desired accuracy} \]
\[ \varepsilon_e = \text{estimated error} \]

This expression relates the change in step size to the information given by the current evaluation. Although the implementation of variable step sizes in a computer program incurs considerable complexity and lead to a water surface profile calculated at a set of non-uniformly spaced points, it is important because

i) the computed values are kept within the prescribed error bounds;
ii) the step size is adjusted automatically which is of particular importance in regions of abrupt changes; and
iii) the correct step size can be automatically determined when it is not certain in the beginning of the integration process.

2.9 Comparison of Numerical Procedures

Computation of a water surface profile requires the solution of the gradually varied flow equation. The solution of the equation describes the water depth along the channel length. Since the equation cannot be directly integrated, the solution of the equation must be obtained by numerical integration.

One way to achieve an approximate solution is to simulate a Taylor series expansion. The greater the number of derivative terms included in the expansion, the higher the accuracy of the solution will be. However, this may not find true in practice since the chosen method is generally developed, which may not suit to certain forms of function. Besides, all numerical methods depend on choosing the proper step size for integration. If the step size is not chosen carefully, then numerical error will grow and may lead to instability of
the integration. The work presented herein examines the magnitude of numerical errors incurred at each integration step and how this affects the results in subsequent integration steps. From this, it is possible to make a judgement and then decide on which integration method is best suited for the gradually varied flow problem.

In order to make meaningful comparison, the various numerical procedures have to be compared with an analytical solution. The explicit solution (2.7) of the gradually varied flow equation derived by Bresse is used for this. If the step increments in Bresse's method are chosen to coincide with those used in the numerical solution, then the difference between the two gives the error introduced during the integration processes. Although the explicit solution of Bresse is restrictive in terms of general application, it provides an opportunity for studying the behaviour of the various methods of solution as well as finding out whether the higher-order integration methods produce more accurate solution. It can also be used to test whether the technique of 'halving or doubling' provides realistic error predictions in contrast to the simpler approach of using error prediction equations. The solution process can then make use of this information to adjust the step size accordingly so as to keep the error under proper control.

A numerical solution is time consuming to perform. The effectiveness of a particular method of solution is directly related to the computational time required to perform an integration. In a sense, it is not meaningful to record the time directly since the time taken varies with the processing speed of the computing machine and depends upon programming efficiency. On the other hand, the computational time taken is directly proportional to the number of function evaluations $f$ that are necessary to achieve a solution. Thus, it is possible to express the computational time in terms of the computational effort $\eta$, as follows

$$\eta = N_f \frac{x_{i+1} - x_i}{h}$$  \hspace{1cm} (2.53)

A high value of $\eta$ implies inefficient performance. The $x_{i+1}$ and $x_i$ are limits of a particular integration; and $h$ is the step size for integrations. The $N_f$ denotes the number of function evaluations involved in the particular integration step; for example, the fourth-order Runge-Kutta method requires four function evaluations ($N_f = 4$) whereas the Butcher method requires six function evaluations ($N_f = 6$).
To conduct this investigation, a rectangular channel of having the physical characteristics tabulated in Table 2.3 is explored. The reason for using mild slopes in this investigation is that it is the one usually encountered in practice. Therefore, it is important to understand the likely errors when the gradually varied flow equation is numerically integrated. Even though mild slopes are used, it is impossible to test for all situations because there offers infinite combinations, hence a mild slope of $S_o = 0.0005$ is chosen for the investigation. Because of using mild slopes, M1 and M2 are the water surface profiles included in the investigation. It is believed that the use of such slope can give indications of the behaviour of various integration methods.

The generalisation of the algorithms presented in Figure 2.2 and Figure 2.3 are adopted for testing various numerical integration methods. These include the Standard Step method (SS), the second-order Trapezoidal method (TM2), the fourth-order Runge-Kutta method (RK4) (and its variants) and the fifth-order Butcher method (BM5). These methods are programmed under the identical conditions. All computations are executed in double precision (approximately 15 significant figures) so that round-off error is essentially negligible.

2.9.1 Magnitude of Local Truncation Error

In order to investigate the truncation error involved in a particular integration step, the gradually varied flow equation is numerically integrated using the methods described above. The water depth at the control section is given and is allowed to vary between 0.55 (m) and 1.60 (m). The intention is to observe the significance of the truncation error with respect to the slope of the water surface. It is suspected that the steeper the water surface slope, the greater the truncation error. However, water depths very close to critical depth are avoided because the gradually varied flow equation becomes indeterminate or infinite. For subcritical flow computations, the integrations were performed in the upstream direction. The difference between the numerical solution and the analytical Bresse solution is used as the indicator of the accuracy of the methods.

Four set of computer runs were conducted separately with different step sizes $h \in \{0.25, 2.50, 25.0, 250.0\}$ (m) for the methods of SS, TM2, RK4, BM5 against the Bresse's solution. The results shown in Table 2.4a to 2.4d give the computed upstream water depth after a single step of size $h$, where $h$ is varied between 0.25 (m) and
Physical Characteristics | Numerical Values
--- | ---
Bed width | 10 (m)\(^{\circ}\)
Bed slope | 0.0005 (m)
Chézy friction coefficient | 750 (m/s\(^2\))
Discharge | 1.0 (m\(^3\)/s)
Critical depth | 0.4671354294 (m)
Normal depth | 1.0280828476 (m)

Table 2.3. Physical characteristics of the open channel.

\(^{\circ}\) The value chosen for bed width in this instance is not strictly 'wide', in accordance with the requirements of the Bresse method. Further numerical experiments with hydraulically wide channel would be useful in confirming these results.
## COMPUTATION OF WATER SURFACE PROFILE

### Downstream Water Depth (m)

<table>
<thead>
<tr>
<th>Downstream Water Depth (m)</th>
<th>Computed Upstream Water Depth (m)</th>
<th>Bresse</th>
<th>Standard Step</th>
<th>Trapezoidal</th>
<th>Runge-Kutta</th>
<th>Butcher</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.55</td>
<td>0.55176082</td>
<td>0.55052617</td>
<td>0.55176195</td>
<td>0.55176169</td>
<td>0.55176169</td>
<td></td>
</tr>
<tr>
<td>0.65</td>
<td>0.65058628</td>
<td>0.65015785</td>
<td>0.65058624</td>
<td>0.65058624</td>
<td>0.65058624</td>
<td></td>
</tr>
<tr>
<td>0.75</td>
<td>0.75025937</td>
<td>0.75005648</td>
<td>0.75025946</td>
<td>0.75025946</td>
<td>0.75025946</td>
<td></td>
</tr>
<tr>
<td>0.85</td>
<td>0.85011481</td>
<td>0.85011835</td>
<td>0.85011526</td>
<td>0.85011526</td>
<td>0.85011526</td>
<td></td>
</tr>
<tr>
<td>0.95</td>
<td>0.95003753</td>
<td>0.94998790</td>
<td>0.95003792</td>
<td>0.95003792</td>
<td>0.95003792</td>
<td></td>
</tr>
<tr>
<td>1.05</td>
<td>1.04999130</td>
<td>1.04997357</td>
<td>1.04999160</td>
<td>1.04999160</td>
<td>1.04999160</td>
<td></td>
</tr>
<tr>
<td>1.15</td>
<td>1.14996186</td>
<td>1.14996433</td>
<td>1.14996175</td>
<td>1.14996175</td>
<td>1.14996175</td>
<td></td>
</tr>
<tr>
<td>1.25</td>
<td>1.24994159</td>
<td>1.24995807</td>
<td>1.24994150</td>
<td>1.24994150</td>
<td>1.24994150</td>
<td></td>
</tr>
<tr>
<td>1.35</td>
<td>1.34992693</td>
<td>1.34995364</td>
<td>1.34992719</td>
<td>1.34992719</td>
<td>1.34992719</td>
<td></td>
</tr>
<tr>
<td>1.45</td>
<td>1.44991711</td>
<td>1.44995042</td>
<td>1.44991678</td>
<td>1.44991678</td>
<td>1.44991678</td>
<td></td>
</tr>
<tr>
<td>1.55</td>
<td>1.54990941</td>
<td>1.54994801</td>
<td>1.54990899</td>
<td>1.54990899</td>
<td>1.54990899</td>
<td></td>
</tr>
</tbody>
</table>

### Normalised Water Depth

<table>
<thead>
<tr>
<th>Downstream Water Depth (m)</th>
<th>Normalised Water Depth</th>
<th>Relative Truncation Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard Step</td>
<td>Trapezoidal</td>
<td>Runge-Kutta</td>
</tr>
<tr>
<td>0.55</td>
<td>1.17738875</td>
<td>2.23765E-03</td>
</tr>
<tr>
<td>0.65</td>
<td>1.39145943</td>
<td>6.58539E-04</td>
</tr>
<tr>
<td>0.75</td>
<td>1.60553012</td>
<td>2.70415E-04</td>
</tr>
<tr>
<td>0.85</td>
<td>1.81960080</td>
<td>1.21129E-04</td>
</tr>
<tr>
<td>0.95</td>
<td>2.03367148</td>
<td>5.22401E-05</td>
</tr>
<tr>
<td>1.05</td>
<td>2.24774216</td>
<td>1.68938E-05</td>
</tr>
<tr>
<td>1.15</td>
<td>2.46181284</td>
<td>2.14784E-06</td>
</tr>
<tr>
<td>1.25</td>
<td>2.67588353</td>
<td>1.31846E-05</td>
</tr>
<tr>
<td>1.35</td>
<td>2.88995421</td>
<td>1.97874E-05</td>
</tr>
<tr>
<td>1.45</td>
<td>3.10402489</td>
<td>2.29734E-05</td>
</tr>
<tr>
<td>1.55</td>
<td>3.31809557</td>
<td>2.49053E-05</td>
</tr>
</tbody>
</table>

Table 2.4a. Relative truncation error, \( h = 0.25 \) (m).
## COMPUTATION OF WATER SURFACE PROFILE

### Computed Upstream Water Depth (m)

<table>
<thead>
<tr>
<th>Downstream Water Depth (m)</th>
<th>Bresse</th>
<th>Standard Step</th>
<th>Trapezoidal</th>
<th>Runge-Kutta</th>
<th>Butcher</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.55</td>
<td>0.56589117</td>
<td>0.55507973</td>
<td>0.56604870</td>
<td>0.56589153</td>
<td>0.56589089</td>
</tr>
<tr>
<td>0.65</td>
<td>0.65572942</td>
<td>0.65156715</td>
<td>0.65573259</td>
<td>0.65572985</td>
<td>0.65572985</td>
</tr>
<tr>
<td>0.75</td>
<td>0.75257085</td>
<td>0.75056326</td>
<td>0.75257139</td>
<td>0.75257122</td>
<td>0.75257122</td>
</tr>
<tr>
<td>0.85</td>
<td>0.85114739</td>
<td>0.85011816</td>
<td>0.85114735</td>
<td>0.85114734</td>
<td>0.85114734</td>
</tr>
<tr>
<td>0.95</td>
<td>0.95037829</td>
<td>0.94987910</td>
<td>0.95037825</td>
<td>0.95037825</td>
<td>0.95037825</td>
</tr>
<tr>
<td>1.05</td>
<td>1.04991639</td>
<td>1.04973580</td>
<td>1.04991610</td>
<td>1.04991610</td>
<td>1.04991610</td>
</tr>
<tr>
<td>1.15</td>
<td>1.14961771</td>
<td>1.14964346</td>
<td>1.14961793</td>
<td>1.14961793</td>
<td>1.14961793</td>
</tr>
<tr>
<td>1.25</td>
<td>1.24941504</td>
<td>1.24958078</td>
<td>1.24941540</td>
<td>1.24941540</td>
<td>1.24941540</td>
</tr>
<tr>
<td>1.35</td>
<td>1.34927240</td>
<td>1.34953652</td>
<td>1.34927234</td>
<td>1.34927234</td>
<td>1.34927234</td>
</tr>
<tr>
<td>1.45</td>
<td>1.44916870</td>
<td>1.44950428</td>
<td>1.44916809</td>
<td>1.44916809</td>
<td>1.44916809</td>
</tr>
<tr>
<td>1.55</td>
<td>1.54909113</td>
<td>1.54948017</td>
<td>1.54909015</td>
<td>1.54909015</td>
<td>1.54909015</td>
</tr>
</tbody>
</table>

### Normalised Water Depth

<table>
<thead>
<tr>
<th>Downstream Water Depth (m)</th>
<th>Normalised Water depth</th>
<th>Relative Truncation Error</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Standard Step</td>
<td>Trapezoidal</td>
</tr>
<tr>
<td>0.55</td>
<td>1.17738875</td>
<td>1.91052E-02</td>
</tr>
<tr>
<td>0.65</td>
<td>1.39145943</td>
<td>6.34755E-03</td>
</tr>
<tr>
<td>0.75</td>
<td>1.60553012</td>
<td>2.66765E-03</td>
</tr>
<tr>
<td>0.85</td>
<td>1.81960080</td>
<td>1.20923E-03</td>
</tr>
<tr>
<td>0.95</td>
<td>2.03367148</td>
<td>5.25249E-04</td>
</tr>
<tr>
<td>1.05</td>
<td>2.24774216</td>
<td>1.72007E-04</td>
</tr>
<tr>
<td>1.15</td>
<td>2.46181284</td>
<td>2.23990E-05</td>
</tr>
<tr>
<td>1.25</td>
<td>2.67588353</td>
<td>1.32656E-04</td>
</tr>
<tr>
<td>1.35</td>
<td>2.88995421</td>
<td>1.95751E-04</td>
</tr>
</tbody>
</table>

Table 2.4b. Relative truncation error, $h = 2.5$ (m).
### Computation of Water Surface Profile

<table>
<thead>
<tr>
<th>Control Water Depth (m)</th>
<th>Bresse</th>
<th>Standard Step</th>
<th>Trapezoidal</th>
<th>Runge-Kutta</th>
<th>Butcher</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.55</td>
<td>0.64547195</td>
<td>0.58999740</td>
<td>0.66503067</td>
<td>0.64859749</td>
<td>0.64552191</td>
</tr>
<tr>
<td>0.65</td>
<td>0.69760001</td>
<td>0.66467870</td>
<td>0.69884881</td>
<td>0.69761341</td>
<td>0.69759984</td>
</tr>
<tr>
<td>0.75</td>
<td>0.77364580</td>
<td>0.75548119</td>
<td>0.77368555</td>
<td>0.77364581</td>
<td>0.77364568</td>
</tr>
<tr>
<td>0.85</td>
<td>0.86097733</td>
<td>0.85116521</td>
<td>0.86099101</td>
<td>0.86097867</td>
<td>0.86097687</td>
</tr>
<tr>
<td>0.95</td>
<td>0.95368575</td>
<td>0.94880077</td>
<td>0.95368686</td>
<td>0.95368554</td>
<td>0.95368554</td>
</tr>
<tr>
<td>1.05</td>
<td>1.04917530</td>
<td>1.04737143</td>
<td>1.04917456</td>
<td>1.04917464</td>
<td>1.04917464</td>
</tr>
<tr>
<td>1.15</td>
<td>1.14622078</td>
<td>1.14644665</td>
<td>1.14620799</td>
<td>1.14622086</td>
<td>1.14622086</td>
</tr>
<tr>
<td>1.25</td>
<td>1.24419862</td>
<td>1.24581767</td>
<td>1.24419844</td>
<td>1.24419842</td>
<td>1.24419842</td>
</tr>
<tr>
<td>1.35</td>
<td>1.34276268</td>
<td>1.34537307</td>
<td>1.34276331</td>
<td>1.34276323</td>
<td>1.34276323</td>
</tr>
<tr>
<td>1.45</td>
<td>1.44171517</td>
<td>1.44504897</td>
<td>1.44171464</td>
<td>1.44171455</td>
<td>1.44171455</td>
</tr>
<tr>
<td>1.55</td>
<td>1.54093020</td>
<td>1.54480666</td>
<td>1.54092944</td>
<td>1.54092935</td>
<td>1.54092935</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Control Water Depth (m)</th>
<th>Normalised Water depth</th>
<th>Standard Step</th>
<th>Trapezoidal</th>
<th>Runge-Kutta</th>
<th>Butcher</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.55</td>
<td>1.17738875</td>
<td>8.59442E-02</td>
<td>3.03014E-02</td>
<td>4.84227E-03</td>
<td>7.74101E-05</td>
</tr>
<tr>
<td>0.65</td>
<td>1.39145943</td>
<td>4.71922E-02</td>
<td>1.79014E-03</td>
<td>1.92006E-05</td>
<td>2.39973E-07</td>
</tr>
<tr>
<td>0.75</td>
<td>1.60553012</td>
<td>2.34792E-02</td>
<td>1.58653E-04</td>
<td>1.75694E-09</td>
<td>1.62150E-07</td>
</tr>
<tr>
<td>0.85</td>
<td>1.81960080</td>
<td>1.13965E-02</td>
<td>1.58905E-05</td>
<td>5.32914E-07</td>
<td>5.32956E-07</td>
</tr>
<tr>
<td>0.95</td>
<td>2.03367148</td>
<td>5.12221E-03</td>
<td>1.16919E-06</td>
<td>2.15702E-07</td>
<td>2.15510E-07</td>
</tr>
<tr>
<td>1.05</td>
<td>2.24774216</td>
<td>1.71932E-03</td>
<td>7.04949E-07</td>
<td>6.27207E-07</td>
<td>6.27220E-07</td>
</tr>
<tr>
<td>1.15</td>
<td>2.46181284</td>
<td>1.97053E-04</td>
<td>4.12337E-09</td>
<td>6.94004E-08</td>
<td>6.93810E-08</td>
</tr>
<tr>
<td>1.25</td>
<td>2.67588353</td>
<td>1.30128E-03</td>
<td>1.38417E-07</td>
<td>1.59040E-07</td>
<td>1.59050E-07</td>
</tr>
<tr>
<td>1.35</td>
<td>2.88995421</td>
<td>1.94404E-03</td>
<td>4.65790E-07</td>
<td>4.07524E-07</td>
<td>4.07519E-07</td>
</tr>
<tr>
<td>1.45</td>
<td>3.10402489</td>
<td>2.31239E-03</td>
<td>3.68092E-07</td>
<td>4.33118E-07</td>
<td>4.33120E-07</td>
</tr>
<tr>
<td>1.55</td>
<td>3.31809557</td>
<td>2.51566E-03</td>
<td>4.95257E-07</td>
<td>5.54261E-07</td>
<td>5.54262E-07</td>
</tr>
</tbody>
</table>

Table 2.4c. Relative truncation error, $h = 25$ (m).
### Computation of Water Surface Profile

#### Computed Upstream Water Depth (m)

<table>
<thead>
<tr>
<th>Downstream Water Depth (m)</th>
<th>Bresse</th>
<th>Standard Step</th>
<th>Trapezoidal</th>
<th>Runge-Kutta</th>
<th>Butcher</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.55</td>
<td>0.85431769</td>
<td>0.75167406</td>
<td>1.40320187</td>
<td>2.13826420</td>
<td>0.70912934</td>
</tr>
<tr>
<td>0.65</td>
<td>0.86565206</td>
<td>0.75115389</td>
<td>0.96000432</td>
<td>0.91942112</td>
<td>0.86682894</td>
</tr>
<tr>
<td>0.75</td>
<td>0.88934004</td>
<td>0.79434571</td>
<td>0.91132935</td>
<td>0.89176614</td>
<td>0.88937094</td>
</tr>
<tr>
<td>0.85</td>
<td>0.92725705</td>
<td>0.86026386</td>
<td>0.93209933</td>
<td>0.92726196</td>
<td>0.92725863</td>
</tr>
<tr>
<td>0.95</td>
<td>0.97912320</td>
<td>0.93892573</td>
<td>0.97983940</td>
<td>0.97910899</td>
<td>0.97912259</td>
</tr>
<tr>
<td>1.05</td>
<td>1.04297230</td>
<td>1.02505107</td>
<td>1.04290984</td>
<td>1.04297337</td>
<td>1.04297197</td>
</tr>
<tr>
<td>1.15</td>
<td>1.11621773</td>
<td>1.11570654</td>
<td>1.11613684</td>
<td>1.11622083</td>
<td>1.11621832</td>
</tr>
<tr>
<td>1.25</td>
<td>1.19644999</td>
<td>1.20920837</td>
<td>1.19645285</td>
<td>1.19645075</td>
<td>1.19644920</td>
</tr>
<tr>
<td>1.35</td>
<td>1.28174255</td>
<td>1.30455898</td>
<td>1.28180413</td>
<td>1.28174305</td>
<td>1.28174222</td>
</tr>
<tr>
<td>1.45</td>
<td>1.37067889</td>
<td>1.40114796</td>
<td>1.37076473</td>
<td>1.37067894</td>
<td>1.37067848</td>
</tr>
<tr>
<td>1.55</td>
<td>1.46224563</td>
<td>1.49858993</td>
<td>1.46233591</td>
<td>1.46224648</td>
<td>1.46224623</td>
</tr>
</tbody>
</table>

#### Normalised Water Depth

<table>
<thead>
<tr>
<th>Downstream Water Depth (m)</th>
<th>Normalised Water depth</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.55</td>
<td>1.17738875</td>
</tr>
<tr>
<td>0.65</td>
<td>1.39145943</td>
</tr>
<tr>
<td>0.75</td>
<td>1.60553012</td>
</tr>
<tr>
<td>0.85</td>
<td>1.81960080</td>
</tr>
<tr>
<td>0.95</td>
<td>2.03367148</td>
</tr>
<tr>
<td>1.05</td>
<td>2.24774216</td>
</tr>
<tr>
<td>1.15</td>
<td>2.46181284</td>
</tr>
<tr>
<td>1.25</td>
<td>2.67588353</td>
</tr>
<tr>
<td>1.35</td>
<td>2.88995421</td>
</tr>
<tr>
<td>1.45</td>
<td>3.10402489</td>
</tr>
<tr>
<td>1.55</td>
<td>3.31809557</td>
</tr>
</tbody>
</table>

#### Relative Truncation Error

<table>
<thead>
<tr>
<th>Downstream Water Depth (m)</th>
<th>Normalised Water depth</th>
<th>Standard Step</th>
<th>Trapezoidal</th>
<th>Runge-Kutta</th>
<th>Butcher</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.55</td>
<td>1.17738875</td>
<td>1.20147E-01</td>
<td>6.42483E-01</td>
<td>1.50289E+00</td>
<td>1.69947E-01</td>
</tr>
<tr>
<td>0.65</td>
<td>1.39145943</td>
<td>1.32266E-01</td>
<td>1.08996E-01</td>
<td>6.21139E-02</td>
<td>1.35954E-03</td>
</tr>
<tr>
<td>0.75</td>
<td>1.60553012</td>
<td>1.06814E-01</td>
<td>2.47254E-02</td>
<td>2.72797E-03</td>
<td>3.47442E-05</td>
</tr>
<tr>
<td>0.85</td>
<td>1.81960080</td>
<td>7.22488E-02</td>
<td>5.22215E-03</td>
<td>5.30162E-06</td>
<td>1.70573E-06</td>
</tr>
<tr>
<td>0.95</td>
<td>2.03367148</td>
<td>4.10546E-02</td>
<td>7.31473E-04</td>
<td>1.45129E-05</td>
<td>6.25112E-07</td>
</tr>
<tr>
<td>1.05</td>
<td>2.24774216</td>
<td>1.71829E-02</td>
<td>5.98955E-05</td>
<td>1.01800E-06</td>
<td>3.24489E-07</td>
</tr>
<tr>
<td>1.15</td>
<td>2.46181284</td>
<td>4.57974E-04</td>
<td>7.24679E-05</td>
<td>2.77606E-06</td>
<td>5.22020E-07</td>
</tr>
<tr>
<td>1.25</td>
<td>2.67588353</td>
<td>1.06635E-02</td>
<td>2.39161E-06</td>
<td>6.32849E-07</td>
<td>6.62482E-07</td>
</tr>
<tr>
<td>1.35</td>
<td>2.88995421</td>
<td>1.78011E-02</td>
<td>4.80481E-05</td>
<td>3.94680E-07</td>
<td>2.56989E-07</td>
</tr>
<tr>
<td>1.45</td>
<td>3.10402489</td>
<td>2.22292E-02</td>
<td>6.26268E-05</td>
<td>3.54011E-08</td>
<td>2.94036E-07</td>
</tr>
<tr>
<td>1.55</td>
<td>3.31809557</td>
<td>2.48551E-02</td>
<td>6.17386E-05</td>
<td>5.82813E-07</td>
<td>4.08658E-07</td>
</tr>
</tbody>
</table>

Table 2.4d. Relative truncation error, $h = 250$ (m).
Figure 2.5a. Relative truncation error, $h = 0.25$ (m).

Figure 2.5b. Relative truncation error, $h = 2.5$ (m).
Figure 2.5c. Relative truncation error, \( h = 25 \) (m).

Figure 2.5d. Relative truncation error, \( h = 250 \) (m).
250.0 (m) as previously described. The range of values selected for the downstream water depth (control depth) are such that both M1 and M2 profiles are investigated. The relative truncation errors are calculated as follows:

\[
\text{Relative Truncation Error} = \left| \frac{y_{\text{method}} - y_{\text{Bresse}}}{y_{\text{Bresse}}} \right|
\] (2.54)

in which \( y_{\text{method}} \) denotes depth as calculated by the numerical method being used; \( y_{\text{Bresse}} \) denotes the water depth calculated by the analytical Bresse method. The water depths are normalised as shown below:

\[
\text{Normalised Water Depth} = \frac{y}{y_e}
\] (2.55)

The intention is to relate a particular water depth to the critical depth so that the findings could be generalised. Four logarithmic plots of the corresponding tabulated results are also presented in Figures 2.5a to 2.5d respectively.

The SS method was the least accurate by a considerable margin when compared with other methods of solution. It may be due to the inherent error in the formulation, so that the head-losses could not be predicted accurately even though a very small step size was used. The kink (near \( y/y_e = 2.5 \)) which appears in the curves was caused by the nature (positive or negative) of the relative errors. Before the relative errors are taken to be absolute values, they can be numerically positive or negative. When the relative errors are taken absolute values, the negative values (left of the kink) turned into positive values so that each curve appears to be discontinuous.

Consider the case where \( y_2 = 0.55 \) (m) and \( h = 0.25 \) (m); the average head-loss for the SS method is calculated as

\[
h\bar{S}_f = \frac{h}{2}(S_{f1} + S_{f2}) = 2.67 \times 10^{-4} \text{ (m)}
\]

in which \( S_{f1} \) and \( S_{f2} \) were calculated using the Chézy formula. Compare this with the actual head-loss from the Bresse solution
in which $H_1$ and $H_2$ were evaluated using the figures highlighted in Table 2.4a. The head-loss under-estimated by 67% but only a 0.22% exists in the computed upstream water depth (see highlighted areas in Table 2.4). This implies that a slight change in water depth can cause a considerable change in head-loss. In order to achieve a balanced situation between $H_1$ and $H_2$, the $\gamma_1$ in SS method must be decreased accordingly, thereby introducing an error of 0.22% in the $\gamma_1$. As shown in Figure 2.5a, this error exists near critical depth, and then decays as the water depth increases. This implies that the formulation used above provides a more accurate estimation of the head-loss for large water depths than for those computed in the vicinity of critical depth. This is generally observed for other test cases too.

With $h = 0.25$ (m), there was no difference observed between the TM2, RK4 and BM5 methods in obtaining the solution. The size of $h$ was so small that the truncation errors incurred in the integrations were essentially negligible. The random distribution of the relative error dictates the threshold limits of the integration methods. A common bound on the relative error is found approximately to be of order $10^{-6}$ to $10^{-7}$, meaning that the solution can only be obtained to that accuracy even though smaller size of $h$ be used for the integrations. This threshold limit is also obtained for other test cases with larger step sizes.

With step size $h = 250$ (m), the difference between the TM2, RK4 and BM5 methods becomes more apparent. The graphs show that the higher-order methods agreed best with the theory and provide more accurate solutions for lower normalised water depths. In general, about a factor of ten of improvement in truncation errors were obtained between methods. The sizeable error obtained is due to the sharp curvature of the water surface; an accurate prediction of the upstream water depth cannot be made with such a large step size. It implies that a smaller step size must be used near regions of sharp curvature in order to control the truncation error in a particular integration step.

The error decayed as the water depth increased (that is, water surface slope decreased). The error incurred in the method of TM2 stabilised below $O(10^{-4})$ whereas the error incurred in the methods of RK4 and BM5 decayed further to between $O(10^{-6})$ and $O(10^{-7})$. These figures appear to define the threshold limits and suggest that they have

\[ H_1 - H_2 = 8.12 \times 10^{-4} \text{ (m)} \]
reached their limitations. Based on these examples, the accuracy of the solution does not noticeably improved even though a smaller size of \( h \) is used. Moreover, the error decayed at a faster rate, which also indicates the effectiveness of the method. More accurate solutions were obtained with the higher-order methods.

Regarding the effectiveness of the SS, TM2, RK4 and BM5 methods, the computational effort \( \eta \) required for the methods are found to be 19, 19, 4 and 6 respectively. The methods of SS and TM2 are rejected for further consideration because they required more function evaluations but achieved less accurate solutions than the other two methods. Hence, the only choice is between the methods of RK4 and BM5. The BM5 method is the best choice as far as accuracy of the solution is concerned. However, the RK4 method has the following important advantages:

1) As shown in Figures 2.5a to 2.5d, the difference between the methods of RK4 and BM5 is indistinguishable when \( h < 2.5 \) (m). The only evidence of difference in performance is when \( h > 25 \) (m). This is observed only when the integrations were performed near to the critical depth. In such a situation, one would consider decreasing the step size in order to stabilise the integration.

2) The BM5 method does not provide a direct estimate of the truncation error, so that it is unable to describe the accuracy of the solution. Although the technique of step halving or doubling may be used to estimate the truncation, it increases the computational effort by as much as 50%.

3) The RK4 method can provide a direct estimate on the truncation error if some assumptions are made on the form of the differential equation. It is shown in the following section that this error estimate gives a good description of the actual truncation error and its bounds which are incurred in a particular integration step.

### 2.9.2 Predictions of Local Truncation Error

None of the methods investigated provide an immediate estimate of the truncation error. It is therefore difficult to choose the appropriate step size for integration. This problem can be resolved using the technique of step halving or doubling described in Section 2.7.1, but it has an important disadvantage in that it requires a 50% increase in computational
effort. As a consequence, a variant of RK4 method is investigated. Such variants have the advantage of providing an immediate error estimate, but they are specifically designed in comply with certain specific forms of differential equation. If such methods are used in conjunction with other forms of equations, then the general validity of the error estimate may no longer hold. However, since these methods are variations of the RK4 method, it may be reasonable to assume that their general application to other forms of equation is acceptable providing that the equations are well behaved and utilise sufficiently small step size of \( h \) for integrations.

In order to investigate the applicability of those methods to gradually varied flow problems, the RK4 method and its variant the Merson method (MM4) (see Section 2.7.1) are chosen for illustration. The channel used in the previous section is again used herein; and a step size \( h = 250 \text{(m)} \) is adopted throughout the investigation. Such step size is selected because sizeable errors will be introduced in the computation process. Two sets of computer runs were conducted on each method, and the results are tabulated in Tables 2.5a and 2.5b. The results give the computed upstream water depth after a single step of size \( h = 250 \text{(m)} \). The range of values selected for the downstream water depth (control depth) are such that both M1 and M2 profiles are investigated. The headings 'Single' and 'Double' denote that the integrations were performed using one full-step \( h \) and two half-steps \( \frac{1}{2}h \) for the technique of step halving or doubling respectively, while the heading 'Predictor' denotes the truncation error estimated directly by equation (2.46) for MM4 method and (2.48) for RK4 method.

Two corresponding logarithmic plots of the tabulated results are shown in Figure 2.6a and Figure 2.6b. The graphs show the estimated relative errors against the actual relative errors incurred. The idea was to correlate the estimated errors with the actual error so that the accuracy of a particular error prediction can be ascertained. If a data point lies above the Actual-Error-Line, then the error prediction is said to be over-estimated; otherwise, the prediction is under-estimated.

A regression analysis (Bajpai et al, 1977) can also be carried out to investigate the distribution of data points. The idea is to create a mathematical relationship between the estimated error \( y \) and the actual error \( x \) so that the relationship might be used to represent an overall average estimation of the \( y \)-values that correspond to the actual \( x \)-values.
COMPUTATION OF WATER SURFACE PROFILE

<table>
<thead>
<tr>
<th>Normalised Depth (m)</th>
<th>Computed Upstream Water Depth (m)</th>
<th>Relative Truncation Error</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Bresse</td>
<td>Single</td>
</tr>
<tr>
<td>1.17738875</td>
<td>0.85431769</td>
<td>2.13826420</td>
</tr>
<tr>
<td>1.39145943</td>
<td>0.86565206</td>
<td>0.91942112</td>
</tr>
<tr>
<td>1.60553012</td>
<td>0.88934004</td>
<td>0.89176614</td>
</tr>
<tr>
<td>1.81960080</td>
<td>0.92725705</td>
<td>0.92726196</td>
</tr>
<tr>
<td>2.03367148</td>
<td>0.97912320</td>
<td>0.97910899</td>
</tr>
<tr>
<td>2.24774216</td>
<td>1.04297230</td>
<td>1.04297337</td>
</tr>
<tr>
<td>2.46181284</td>
<td>1.11621773</td>
<td>1.11622083</td>
</tr>
<tr>
<td>2.67588353</td>
<td>1.19644999</td>
<td>1.19645075</td>
</tr>
<tr>
<td>2.88995421</td>
<td>1.28174255</td>
<td>1.28174305</td>
</tr>
<tr>
<td>3.1042489</td>
<td>1.37067889</td>
<td>1.37067894</td>
</tr>
<tr>
<td>3.31809557</td>
<td>1.46224563</td>
<td>1.46224648</td>
</tr>
</tbody>
</table>

Table 2.5a. Fourth-order Runge-Kutta - Predictions of truncation error, \( h = 250 \) (m).

Figure 2.6a. Fourth-order Runge-Kutta - Goodness of fit of relative error, \( h = 250 \) (m).

59
### Table 2.5b. Fourth-order Merson - Predictions of truncation error, $h = 250$ (m).

<table>
<thead>
<tr>
<th>Normalised Depth (m)</th>
<th>Computed Upstream Water depth (m)</th>
<th>Relative Truncation Error</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Bresse</td>
<td>Single</td>
</tr>
<tr>
<td>1.17738875</td>
<td>0.85431769</td>
<td>0.93174008</td>
</tr>
<tr>
<td>1.39145943</td>
<td>0.86565206</td>
<td>0.86533748</td>
</tr>
<tr>
<td>1.60553012</td>
<td>0.88934004</td>
<td>0.88916006</td>
</tr>
<tr>
<td>1.81960080</td>
<td>0.92725705</td>
<td>0.92719496</td>
</tr>
<tr>
<td>2.03367148</td>
<td>0.97912320</td>
<td>0.97911827</td>
</tr>
<tr>
<td>2.24774216</td>
<td>1.04297230</td>
<td>1.04297205</td>
</tr>
<tr>
<td>2.46181284</td>
<td>1.11621773</td>
<td>1.11621809</td>
</tr>
<tr>
<td>2.67588353</td>
<td>1.19644999</td>
<td>1.19644892</td>
</tr>
<tr>
<td>2.88954211</td>
<td>1.28174255</td>
<td>1.28174204</td>
</tr>
<tr>
<td>3.10402489</td>
<td>1.37067889</td>
<td>1.37067840</td>
</tr>
<tr>
<td>3.31809557</td>
<td>1.46224563</td>
<td>1.46224620</td>
</tr>
</tbody>
</table>

---

### Figure 2.6b. Fourth-order Merson - Goodness of fit of relative error, $h = 250$ (m).
A linear relationship between $y$ and $x$ is assumed.

$$\log(y) = a_0 + a_1 \log(x)$$

Using the technique of least squares (Bajpai et al, 1974), both coefficients $a_0$ and $a_1$ can be evaluated.

As shown in Figures 2.6a and 2.6b, there emerges a common pattern in which the data points are distributed. Beneath the region where the relative error is less than $O(10^{-6})$ (for both horizontal and vertical axes), the data points are scattered above and below the Actual-Error-Line. This would be expected if the methods have reached their threshold limit of $O(10^{-6})$ (see Figures 2.5a to 2.5d). Neither the technique of halving or doubling, nor the error equations can provide a good description or a reliable prediction of the actual truncation error. However, it appears that the actual relative error incurred in a particular integration is about or less than $O(10^{-5})$, when the relative error predicted by the methods is less than $O(10^{-6})$.

In the region beyond $O(10^{-6})$, the truncation errors (triangular data points) predicted by the technique of step halving or doubling agree well with the theory. The estimated truncation errors are very close to the actual errors, and are marginally under-estimated. The predictions of the error are sufficiently accurate to enable the step size of $h$ for integration to be confidently interpolated using the method of adaptive size control described in Section 2.8.

On the other hand, the truncation errors (circular data points) predicted by the error equations were consistently over-estimated. The errors predicted were greater than those obtained by the technique of halving or doubling. It is because the assumption made about the form of equation (2.47) is not satisfied. Owing to this non-linearity, the error equations are seen to provide certain bounds on the truncation error as shown in Figures 2.6a and 2.6b. The use of the error equations gives an over-estimate of the actual truncation errors; approximately 66% in these examples.

The technique of step halving or doubling is the best choice as far as numerical accuracy is concerned. However, the 50% increase in computational effort required is a serious disadvantage, and for the reason the use of the error equation may be preferred. Although
the predictions are over-estimated, it has been shown that the actual errors incurred are reasonably well predicted. Hence, it is reasonable to choose $h$ based on the estimate given by the error equation.

On comparing the data in Tables 2.5a and 2.5b, the errors incurred in the MM4 method were smaller than those incurred in the RK4 method, implying that the MM4 method approximated the solution more accurately than the RK4 method. The regression analysis yields

$$
\text{RK4 method: } \log(y) = 0.668 + 1.022 \log(x) \\
\text{MM4 method: } \log(y) = 0.660 + 1.021 \log(x)
$$

It must be noted that the data points lying beneath the region of $O(10^{-6})$ (on both axes) were excluded from the above analysis because they were randomly scattered, making any predictions unreliable. The coefficients calculated are very close; the coefficient $a_0$ is greater than zero, indicating that the errors were over-estimated. The slopes $a_1$ are marginally greater than unity suggesting that the error predicted will increase at a very slow rate away from the Actual-Error-Line as the actual error increases. Because $a_1$ is so close to unity, the regression line could reasonably be used to dictate the average error incurred in a particular integration.

The regression analysis also shows that the correlation coefficients obtained for both investigations are

$$
\text{RK4 method: } r = 0.569 \\
\text{MM4 method: } r = 0.408
$$

If $r \to \pm 1$, then the fit is said to be perfect; whereas if $r \to 0$, it represents the data points are randomly scattered. The data points obtained for the RK4 method were better correlated that those obtained for the MM4 method. The MM4 method has the additional disadvantage of requiring five function evaluations rather than the four required by the RK4 method. This increase in function evaluations may prove significant in terms of overall computational effort, particularly if the integration is performed near the critical depth, where smaller step sizes are needed in order to keep the error under proper control. Therefore, the RK4 is the preferred method based on the foregoing analysis of the truncation error.
2.9.3 Upper Bounds on Global Error

As shown in previous sections, numerical errors always be introduced in a particular numerical integration regardless of choice of the step sizes. It is therefore relevant to study the behaviour of the error growth as the integration proceeds. The effect of errors accumulated over previous integration steps is referred to as global error. As defined in Section 2.4, it is the sum of local truncation error and the error propagated from previous integration steps. The global error will grow with increasing distance along the channel length so that an inaccurate water surface profile may be obtained. Therefore, the error bounds describing the growth of the errors are investigated. This is important because it enables the step size to be related to the global error so that ultimately appropriate step size can be determined in order to keep the error under proper control. Further numerical experiments are carried out to investigate the application of adaptive step size control.

With RK4 method, the sufficient conditions for developing the bounds on global error are described in Section 2.7.2. The $M_i$ in (2.49) can be found by direct differentiating the gradually varied flow equation (2.6)

$$M_i \geq |f_s(x, y)| = \left| -\frac{3}{y} \left( \frac{S_y(y/y_c)^3 - f(x, y)}{(y/y_c)^3 - 1} \right) \right|$$

Over the domain of interest, a typical plot of the $M_i$ is shown in Figure 2.7. $M_i$ increases asymptotically to infinity as the water depth approaches the critical depth; conversely, $M_i$ decreases asymptotically to zero as the water depth approaches infinity. Hence, the bounds on $M_i$ are defined and the global error can be established in the following fashion.

Using the channel characteristics in Table 2.3, consider the integration starting with an initial water depth $y_i = 0.55$ (m). The step size of $h = 2.5$ (m) is chosen because the intention is to investigate the effects of the global error when the local truncation error is small. The results obtained are plotted in Figure 2.8a; the 'Stable Bound' and 'Unstable Bound' denote the upper error bounds of RK4 method obtained by (2.49) and (2.50) respectively. Based on the previous numerical experiments, the calculations on the bounds (2.49) and (2.50) assume that the local truncation error is confidently bounded by $|e_i| < 10^{-6}$ (m).
Figure 2.7. A typical plot of $f_y(x, y)$ against $y/y_c$. 
Figure 2.8a. Error bounds on global error, \( h = 2.5 \) (m).

Figure 2.8b. Error bounds on global error, \( h = 25 \) (m).
According to the Figure 2.7, the closer to critical depth, the larger the value of $M_t$. Since the integration started from downstream, it follows that $M_t$ must be bounded at $y_2 = 0.55 \text{ (m)}$ to give

$$M_t \geq 0.10760$$

Hence, the stable bound (2.49) is evaluated

$$|\epsilon_g| \leq \frac{2(10^{-6})}{(2.5)(0.10760)} = 7.43 \times 10^{-6} \text{ (m)}$$

The unstable bound (2.50) initially starts from $i = 0, |\epsilon_g| = 0$. When $i = 1, |\epsilon_g|$ becomes

$$|\epsilon_g| \leq 10^{-6} \left( e^{\left(\frac{2.5(0.10760)}{2.5(0.10760)} - 1\right)} e^{(2.5)(0.10760)} - 1 \right) = 1.31 \times 10^{-6} \text{ (m)}$$

Intuitively, other values of $|\epsilon_g|$ for $i = 1, 2, \ldots$ can be evaluated in the same fashion as $i = 1$, and the computation proceeds in the upstream direction until the channel length is fully covered. It must be noted that the bounds derived above are theoretical values. The actual error involved may not reach even close to the bound derived above. They represent the worst situation of possible growth in error.

Three points to observed from Figure 2.8a are:

i) The actual errors incurred neither grew nor decayed with distance. The errors oscillate indefinitely in the vicinity of error line $10^{-6}$. This oscillatory effect was due to the RK4 method reaching its threshold limit. Because the local truncation error incurred was so small, the growth in global error was essential negligible. Hence, the integration is very stable.

ii) The unstable bound grew beyond realistic values with distance. Therefore, it does not provide an useful description of the size of the actual error incurred, and hence, it serves little use for adaptive step size control.
iii) The stable bound not only completely bounded the global error but also provided a good description of the actual error incurred so that it may be used to determine the step size prior to computations providing that \( e_g \) and \( M_t \) are obtainable.

Since the local truncation error incurred was so small, a further investigation involving sizeable truncation error on the growth of the global error was carried by increasing the step size to \( h = 25 \) (m). The results obtained are plotted in Figure 2.8b; the bounds on local truncation error and global error are calculated to be \( |e_g| < 10^{-2} \) (m) and \( |e_g| \leq 7.43 \times 10^{-3} \) (m) (stable case) respectively.

The actual error decayed rather than grew; this may be due to the truncation error (and \( M_t \)) decreasing with increasing water depth as the computation proceeded in the upstream direction. Although the errors obtained are larger than those obtained with the smaller step size \( h = 2.5 \) (m), these errors appear to be predictable even though a larger step size is used. For this example, the global error still does not grow beyond unrealistic values and confines beneath the stable bound \( |e_g| \leq 7.43 \times 10^{-3} \) (m), suggesting that the integrations are stable and well behaved. Therefore, it can be stated that the application of the RK4 method to the gradually varied flow problems is suitable, and hence provides a good method of analysis.

### 2.9.4 Investigation of Adaptive Step Size Control

The importance of adaptive step size control is that it will automatically select the size of \( h \) for a particular integration step so that the solution can be maintained within the prescribed accuracy. The approach to adaptive step size control is based on the magnitude of the local truncation error. If the error is found too large, then the size of \( h \) must be reduced; if the error is found sufficiently small, then the size of \( h \) may be increased. The objective of this is aiming at reducing computational effort, and hence improve overall efficiency.

In order to investigate the effectiveness of the adaptive step size algorithm shown in Figure 2.4, the channel characteristics shown in Table 2.3 is used. Because of the channel slope involved happens to be mild, the water surface profile obtained is likely to be M1 or M2, implying that the computation is started from the downstream and proceed in an
upstream direction. A control water depth of $h = 0.55$ (m) is chosen for the investigation because such depth is closed to the critical depth so that sizeable error will be introduced in the computation. Therefore, small $h$ must need to be chosen to stabilise the computation. Initially, a step size of $h = 2.5$ (m) is selected, and the resulting water surface profile obtained is shown in Table 2.6, which gives the computed water depths along the channel length and the step sizes used for integrations. The actual error introduced in the computation is given by the difference between the 'Bresse Profile' and the 'Runge-Kutta Profile'; while the local truncation error predicted by (2.48) for each integration step is also tabulated. The highlighted areas denote the local truncation error $e_t$ is sufficiently small ($e_t < \varepsilon_{th}$) so that the size of $h$ can be doubled for the subsequent computations.

Before the computation is carried out, it is determined to obtain a water surface profile as accurate as possible. Therefore, the error of magnitude of $e_t < 10^{-6}$ (m) (the threshold limit of the integration method) is expected to be introduced in a particular integration step providing that the size of $h$ is chosen sufficiently small. Hence, the upper limit $\varepsilon_u$ for the adaptive step size control is assumed to be

$$\varepsilon_u = 10^{-6} \text{ (m)}$$

meaning that when $e_t$ is found greater than $\varepsilon_u$, the size of $h$ is halved; while the lower limit $\varepsilon_l$ is calculated according to the method described in Section 2.8 as

$$\varepsilon_l = \frac{\varepsilon_u}{2^{\varepsilon_t}} = 3.125 \times 10^{-8} \text{ (m)}$$

meaning that when $e_t$ is found less than $\varepsilon_l$, the size of $h$ is doubled. For example, the $e_t$ described in the 1st integration step of Table 2.6 is found to be $5.68704 \times 10^{-7}$ (m). Because it is $e_t < \varepsilon_t < \varepsilon_u$, therefore the size of $h = 2.5$ (m) for the subsequent computations remains unchanged. However in the 4th step, $e_t = 1.61441 \times 10^{-8}$ (m) is found less than $\varepsilon_{th}$, suggesting that the current step size can be doubled to $h = 5.0$ (m) for the subsequent computations.

A corresponding plot of the results is also shown in Figure 2.9, which includes the analytical water profile (Bresse) and the approximated water profile (RK4). It appears
that the RK4 profile obtained coincides with the Bresse profile, suggesting that the RK4 profile is well predicted, even though various sizes of $h$ are used for the computation. In fact, the size of $h$ is gradually increased from $h=2.5\text{ (m)}$ to $h=20\text{ (m)}$ as the computation proceed upstream, but no signs of growth in global errors. A total of 17 steps are used for the 170 (m) long channel as compared with 68 steps when $h=2.5\text{ (m)}$ is used without implementing the adaptive step size control. As a consequence, a considerable 75% reduction in computational effort is achieved. Not only this has proved the effectiveness of the adaptive step size control, but also proved that the errors being introduced in the integration is generally decreasing as the water depth approaching to the normal depth. However, these results obtained are meaningful only when the errors are within the errors bounds. Therefore, a further error analysis is mandatory.

Figure 2.10 shows the actual errors incurred together with the stable and unstable error bounds. These error bounds are obtained from (2.49) and (2.50) respectively based on previous numerical experiments. It is seen that the global error is confidently bounded by $|\varepsilon_p| \leq 7.43 \times 10^{-6}\text{ (m)}$ (stable case). The actual errors obtained although oscillating randomly towards the upstream, they appear to be lying in about the line of $10^{-6}\text{ (m)}$. This random character has been described previously is due to the threshold limit of the method is being reached. However, the most important aspect shown in the graph is that the errors are self-contained well beneath the stable bound $|\varepsilon_p| \leq 7.43 \times 10^{-6}\text{ (m)}$. There exhibits no signs of error growth as the computation proceed upstream, even though increasingly large step sizes are used. Not only these results have proved that the RK4 method is very stable (see Section 2.7.2), but also justified the effectiveness of the presented adaptive step size algorithm, for the reason of automation while maintaining numerical accuracy.

Because the results show the behaviour of the RK4 method is predictable, it can be stated that the application of the RK4 method along with its error equation in conjunction with the adaptive step size control to gradually varied flow problems provides an effective method of analysis. The results show that the procedure is very stable and guarantees that the global error will not grow beyond unrealistic values even though the step size is initially chosen inconsiderately large.
<table>
<thead>
<tr>
<th>Step</th>
<th>Distance (m)</th>
<th>Size of ( h ) (m)</th>
<th>Computed U/S Water Depth (m)</th>
<th>Numerical Errors (m)</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Analytical Bresse Profile</td>
<td>Fourth-order</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Runge-Kutta Profile</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Actual Total Error</td>
<td>Error Predicted by</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(2.48)</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>0.0</td>
<td>20.0</td>
<td>0.81191023</td>
<td>0.81190995</td>
<td>2.76397E-07</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1.94009E-08</td>
</tr>
<tr>
<td>16</td>
<td>20.0</td>
<td>20.0</td>
<td>0.79847837</td>
<td>0.79847799</td>
<td>3.84448E-07</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3.30506E-08</td>
</tr>
<tr>
<td>15</td>
<td>40.0</td>
<td>20.0</td>
<td>0.78342339</td>
<td>0.78342476</td>
<td>1.37041E-06</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>6.15391E-08</td>
</tr>
<tr>
<td>14</td>
<td>60.0</td>
<td>20.0</td>
<td>0.76633676</td>
<td>0.76633658</td>
<td>1.80362E-07</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>6.15391E-08</td>
</tr>
<tr>
<td>13</td>
<td>80.0</td>
<td>20.0</td>
<td>0.74660205</td>
<td>0.74660260</td>
<td>5.47077E-07</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3.25308E-07</td>
</tr>
<tr>
<td>12</td>
<td>100.0</td>
<td>10.0</td>
<td>0.72325451</td>
<td>0.72325522</td>
<td>7.06893E-07</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1.89484E-08</td>
</tr>
<tr>
<td>11</td>
<td>110.0</td>
<td>10.0</td>
<td>0.70974424</td>
<td>0.70974424</td>
<td>1.38393E-10</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3.69099E-08</td>
</tr>
<tr>
<td>10</td>
<td>120.0</td>
<td>10.0</td>
<td>0.69459442</td>
<td>0.69459568</td>
<td>1.26341E-06</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>8.21506E-08</td>
</tr>
<tr>
<td>9</td>
<td>130.0</td>
<td>10.0</td>
<td>0.67731082</td>
<td>0.67731163</td>
<td>8.06411E-07</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2.22722E-07</td>
</tr>
<tr>
<td>8</td>
<td>140.0</td>
<td>5.0</td>
<td>0.65709100</td>
<td>0.65709102</td>
<td>1.35997E-08</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1.39770E-08</td>
</tr>
<tr>
<td>7</td>
<td>145.0</td>
<td>5.0</td>
<td>0.64547195</td>
<td>0.64547230</td>
<td>3.52379E-07</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3.18460E-08</td>
</tr>
<tr>
<td>6</td>
<td>150.0</td>
<td>5.0</td>
<td>0.63249056</td>
<td>0.63249060</td>
<td>4.05806E-08</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>7.14431E-08</td>
</tr>
<tr>
<td>5</td>
<td>155.0</td>
<td>5.0</td>
<td>0.61770001</td>
<td>0.61770057</td>
<td>5.52202E-07</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2.23557E-07</td>
</tr>
<tr>
<td>4</td>
<td>160.0</td>
<td>2.5</td>
<td>0.60035441</td>
<td>0.60035435</td>
<td>6.07967E-08</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1.61141E-08</td>
</tr>
<tr>
<td>3</td>
<td>162.5</td>
<td>2.5</td>
<td>0.59032019</td>
<td>0.59032159</td>
<td>1.40213E-06</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3.93269E-08</td>
</tr>
<tr>
<td>2</td>
<td>165.0</td>
<td>2.5</td>
<td>0.57900387</td>
<td>0.57900337</td>
<td>1.49637E-06</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1.21839E-07</td>
</tr>
<tr>
<td>1</td>
<td>167.5</td>
<td>2.5</td>
<td>0.56589117</td>
<td>0.56589153</td>
<td>3.64190E-07</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>5.68704E-07</td>
</tr>
<tr>
<td>0</td>
<td>170.0</td>
<td>-</td>
<td>0.55000000</td>
<td>0.55000000</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0</td>
</tr>
</tbody>
</table>

Table 2.6. Accuracy of a water surface profile with adaptive step size control.
Figure 2.9. Accuracy of a water surface profile with adaptive step size control.

Figure 2.10. Errors incurred in the water surface profile with adaptive step size control.
2.9.5 Direction of Computations

It is common practice that the direction of computation always proceeds in the upstream direction for subcritical flow and downstream for supercritical flow. However, there is no mathematical proof of this. Under identical flow conditions, it is envisaged that two different water surface profiles will be obtained when the computation is carried out in opposite directions. In order to investigate this phenomenon, the gradually varied flow equation is numerically integrated both upstream and downstream directions. The difference between the numerical solutions and the analytical Bresse solution is used to dictate the appropriateness of the direction of integration.

The channel characteristics shown in Table 2.3 is used, and a step size \( h = 25 \) (m) is chosen throughout the investigation. Two computer runs were conducted under identical conditions. The first water surface profile was computed starting from the downstream end of the channel with a control water depth of \( y_2 = 0.55 \) (m) and proceeded in an upstream direction. The second water surface profile was computed starting from the upstream end of the channel with a control water depth of \( y_1 = 0.929217 \) (m) (calculated from the analytical Bresse solution) and proceeded in a downstream direction. The results are plotted in Figure 2.11; the curves U/S and D/S denote the computations proceeding in the upstream direction and downstream directions respectively.

With the U/S case, the global error decayed with distance and oscillated towards the upstream end of the channel. This phenomenon has been observed in previous examples. With the D/S case, the global error grew with increasing distance. This can be explained using the theorem given by Carr (see Section 2.7.2). Briefly, if \( f_\gamma < 0 \), then the numerical error will not grow and the integration is said to be stable; if \( 0 \leq f_\gamma \), then the numerical error will grow exponentially and the integration is said to be unstable. In fact, \( f_\gamma \) was found to be positive with the value of \( f_\gamma = 0.10760 \). This demonstrates that if the integration proceeds in an inappropriate direction, the integration can be unstable and caused a sizeable error in the final solution. Thus, for subcritical computations, the integration should proceed in an upstream direction, whereas for supercritical computations, the integration should proceed in a downstream direction if sizeable numerical errors are to be avoided.
Figure 2.11. Growth in global error with respect to the direction of computation.
2.10 Concluding Remarks

A general explicit solution of the gradually varied flow equation does not exist, and therefore numerical methods are used. These include the Standard Step method, the second-order Trapezoidal method, the fourth-order Runge-Kutta method and its variants fourth-order Merson and the fifth-order Butcher method.

The primary concern with all numerical methods is their accuracy. Apart from round-off error, two factors which affect the accuracy of the solution are the local truncation error and propagated truncation errors. It is suspected that, during the course of integration, these errors will grow in size if they are not kept under proper control. Thus, a number of experiments were conducted. The basis for comparison of the various numerical methods is the accuracy of the final solution, the total number of function evaluations and their stability. The upper bounds on the numerical errors have also been investigated in order to find out whether they will provide useful information for adaptive step size control.

It was found that the accuracy which the methods can achieve varies consistently and differently, but none of them are found to be unstable. In all test cases, sizeable local truncation was obtained in the vicinity of the critical depth but this error appeared to be predictable. It is due to very sharp curvature of the water surface, meaning that an accurate prediction of the nearby water depth cannot be obtained. It is suggested that the integration must be performed in conjunction with a very small step size in order to keep the truncation error under proper control in this region.

As implied, the higher-order methods provide a more accurate solution. Both Standard Step method and Trapezoidal method require intermediate iterations to achieve the solution, and the solution obtained by them were found to be considerably less accurate than other methods. Therefore, these methods were rejected for further investigations. This leaves the fourth-order Runge-Kutta, fourth-order Merson and fifth-order Butcher for further investigations. With small step size $h < 2.5$ (m), the results obtained by the Runge-Kutta method were almost indistinguishable from those obtained by the Butcher method. Because the latter requires six function evaluations and does not provide an immediate error estimate, the Butcher method is not considered further. For the reason of increased computational effort, the Merson method is rejected because it requires five function evaluations rather than four for the Runge-Kutta method. Besides, regression analysis has shown that the error predicted by the Runge-Kutta method ($r = 0.569$) were
more consistent than the Merson method \((r=0.408)\) although the average of the predictions was almost identical. It is concluded that the Runge-Kutta method is the preferred numerical procedure for flow profile determination.

Numerical experiments showed that the technique of step halving or doubling provides a very good estimate of the local truncation error. The difference between the actual error and the estimated error is insignificant. Unfortunately, the technique requires a 50% increase in computation effort so that, in a sense, the technique is less efficient. An alternative approach to error prediction is to use an error equation. The equation requires no additional function evaluations but it provides an immediate measure of the truncation error. However, the error estimate can be unreliable since the equation is designed for application to certain specific forms of differential equation. If the step size is sufficiently small, then it is reasonable to assume that the application of the error equation to gradually varied flow is acceptable. The prediction provided by the equation not only bounds the actual error but also provides a reasonable means to determine the step size prior to the integration.

Both the unstable bound and the stable bound were found to completely bound the actual errors incurred. However, the unstable bound is extremely conservative. The unstable bound does not provide a useful description of the size of the actual error. Therefore, it serves little practical use. On the other hand, the stable bound provides a reasonable description of the actual error so that it can be used to decide the accuracy of the solution and hence to choose the step size prior to the integration.

Finally, numerical examples showed that the direction of the integration is vital to error growth. If the appropriate direction is being exercised, numerical error will decay. This phenomenon has been illustrated and shown to agree very well with the error theory. Consequently, this provides the explanation why the water surface profile computation must be started from downstream and proceeds in the upstream direction for subcritical flow. The reverse is true for the supercritical flow computation.
CHAPTER 3

NETWORK ANALYSIS
NETWORK ANALYSIS

3.1 Introduction

The analysis of the distribution of flows in networks containing inter-connected channels is far more complicated than the analogous problems of pipe networks, due to the existence of the free water surface of open channel flows. However, procedures currently available for modelling such problems are available. Of these procedures, Smith and Ashenhurst (1986) have developed a computer program called RIVER4 which concerns storm water management systems, but also includes the analysis of bifurcating island flows. The Hardy-Cross method (1936) was used as the basis of the formulation. The software CHAT (Computerised Hydraulic Analysis of Treatment) developed by Cotton, Faulkner, Case and Mortimer (1987) allows networks comprising modules in wastewater treatment works to be analysis. This also used an adaptation of the Hardy-Cross method to force the network to a solution which satisfied both continuity and head-loss criteria. Problems of convergence were encountered using this approach. A more systematic approach to the bifurcating river problem has been addressed by Wylie (1972) and Chaudhry and Schulte. (1987). Their approach is essentially based on the principles of flow continuities and energy conservation. It appears that these schemes generate a very large set of equations resulting in large sparse matrices, meaning that considerable computational power is required to manipulate them.

The works presented herein is aimed principally at applications to the hydraulic analysis of potable water and waste water treatment works. The model proposed is generalised and allows systems to be analysed in the broadest sense. This includes hydraulic structures like
NETWORK ANALYSIS

- open channels and channel transitions
- weir and flumes
- penstocks and control gates
- tanks
- pipes and pipe fittings
- overflow structures
- screen and proprietary equipment

The method is developed using 'loop formulation' resulting in a set of simultaneous non-linear equations which requires solution. The basis of the formulation is head-loss and continuity of flow. Using these relationships, one can predict the flow distribution and tailwater conditions for different head-losses within the network.

3.2 Conceptual Model

The geometrical representation of an hydraulic system, referred to as topology, can be described by an equivalent network which can be broken down into constituent components as defined below:

i) A 'network' can be considered as a collection of branches interconnected in a predetermined manner. The interconnected branches are called 'strings'.

ii) A 'string' is a collection of 'modules' which are arranged in a series manner. A string has two ends called 'nodes', at which it is connected to the network. A string can be considered as a module or vice versa if and only if the string contains one single module.

iii) A 'module' can be considered as a hydraulic structure which has a unique mathematical relationship between the head-loss, discharge and water depth variables.

By way of example, consider part of a wastewater treatment plant as forming a network which has three strings as shown in Figure 3.1a. The network represents three parallel primary settlement tanks. Each primary settlement tank constitutes a 'string', and each string is made up of a series 'modules'. The individual hydraulic modules which made up each string are shown in Figure 3.1b. The importance of this string construction is two fold.
Figure 3.1a. Composition of a network with strings.

Figure 3.1b. Composition of a string with modules.
i) It allows individual modules to be changed without necessarily modifying the overall configuration (layout) of the network. This, in turn, enables the basic system equations to remain unchanged.

ii) Solution of the network system equations can be considered with respect to strings rather than modules, because the combined hydraulic characteristics of the individual modules enable the hydraulic characteristics of the string to be determined. This has considerable advantages in terms of efficiency in solving the system of equations.

The modelling process proceeds as follows:

i) Assemble the modules into strings.

ii) Assume an initial flow distribution between strings which satisfies the continuity at all nodes.

iii) Calculate the head-loss along each string.

iv) Check that the sum of head-losses around each loop in the network is zero.

v) Calculate a correction factor based on the 'out-of-balance head-loss' to apply to the initial assumed flow distribution.

vi) Repeat the head-loss calculation/flow correction procedure until a prescribed tolerance is reached.

The main focus of this research is to explore means by which flow correction factors can be predicted to enable the calculation procedure to converge rapidly to a solution.

The rest of this Chapter is devoted to an explanation of how the governing system equations are set up, and to previous work undertaken in this field. Chapter 4 then explores the different methods available to predict flow correction factors with particular regard for rapid convergence to a network solution.

3.3 Governing Equations

Regardless of the pattern of interconnection and complexity of a network, the flow in any network must obey the fundamental laws of physics. Those are:

i) Conservation of energy - Energy within the system is neither created nor destroyed.

ii) Continuity of flow - The total flow into a nodal junction must be equal to the flow out.
iii) The algebraic sum of the head-losses around any closed loop must be zero.

iv) The flow in each hydraulic element must satisfy friction laws; that is, the head versus discharge relationship.

Firstly, consider the law of conservation of energy. If energy is neither created nor destroyed, at any point along a string, then the flow along that string can be described by

\[ H_{r,1} = H_{r,2} + h_f \]

where

- \( H_{r,1} \) = total energy head at upstream end of string \( s \)
- \( H_{r,2} \) = total energy head at downstream end of string \( s \)
- \( h_f \) = head-loss due to frictional resistance across string \( s \)

Since a string is comprised of a number of modules, the \( h_f \) in the equation above may not be explicitly expressible. If, however, the equation used to calculate the head-loss across each module is known, then the sum of all the module head-losses defines the \( h_f \). That is,

\[ h_f = \sum_{m=1}^{M_s} h_{f,m} \]  

where

- \( M_s \) = number of modules associated with string \( s \)
- \( h_{f,m} \) = head-loss across module \( m \) associated with string \( s \)

The above equation assumes that the flow across each module is constant and flowing in the same direction.

Secondly, if the flow is assumed incompressible, then the total flow into a junction must equal the total flow out of it. Therefore, the law of continuity of flow at nodal junctions can be written as
\[ \sum_{s=1}^{S_j} Q_{j,s} + C_j = 0 \]  
(3.3)

where

\[ C_j = \text{constant inflow or outflow at junction} \ j \]

\[ S_j = \text{number of strings associated with nodal point or junction} \ j \]

\[ Q_{j,s} = \text{flow of string} \ s \ \text{associated with junction} \ j \]

and

\[ Q_{j,s}, C_j = \begin{cases} 
+\text{ve}, & \text{if a flow is directed towards junction} \ j \\
0, & \text{if no flow is directed towards or away from junction} \ j \\
-\text{ve}, & \text{if a flow is directed away from junction} \ j 
\end{cases} \]

Finally, the sum of head-losses around any closed loop must equal to zero and that be written as

\[ \sum_{s=1}^{S_l} h_{f,s} = 0 \]  
(3.4)

where

\[ S_l = \text{number of strings associated with loop} \ l \]

\[ h_{f,s} = \text{head-loss of string} \ s \ \text{associated with loop} \ l \]

and

\[ h_{f,s} = \begin{cases} 
+\text{ve}, & \text{if the flow direction of string} \ s \ \text{is as same as the direction of loop} \ l \\
-\text{ve}, & \text{if the flow direction of string} \ s \ \text{is opposite to the direction of loop} \ l 
\end{cases} \]

Modelling a hydraulic network is equivalent to setting up a system of simultaneous equations that describes the response of that network. A number of approaches to
formulating a system of equations are possible. Two approaches commonly adopted in pipe network problems are the nodal principle and the loop principle.

3.3.1 Nodal Principle

With the nodal principle, the formulation is concentrated on flow continuity at nodal junctions. The flows at each junction can be written as

\[ f_j = \sum_{s=1}^{s_j} Q_{j,s} + C_j = 0, \quad j = 1, 2, \ldots, J_N \]  

(3.5)

in which \( f_j \) denotes the unbalanced flow at the \( j \)th nodal junction and which must be zero at the solution; \( J_N \) is the total number of nodal junctions within the system. The equation above is readily solvable but the solution is trivial. It is because \( f_j \) is only a linear function of \( Q_{j,s} \) so that an infinite combination of flows in the strings that satisfies (3.5) are possible. To develop a solution, the flow behaviour or characteristic of the strings are necessary. It is known from point iv) that the flow across a string is related to the energy head. It follows that it is valid to write the function

\[ Q_{j,s} = \phi(H_j, H_{j+1}) \]  

(3.6)

in which \( H_j \) refers to the energy head at the \( j \)th nodal junction. This definition is essential because the \( H_j \) is the energy head common to all the heads in the strings which adjoin to the \( j \)th nodal junction. Consider the example shown in Figure 3.2, which comprises four strings and four nodal junctions; the energy heads at the nodal junctions common to all strings are defined such that

\[
\begin{align*}
H_j &= H_{s,1} \\
H_{j+1} &= H_{s,2} = H_{s+1,1} = H_{s+1,2} \\
H_{j+2} &= H_{s+1,2} = H_{s+2,2} = H_{s+3,1} \\
H_{j+3} &= H_{s+3,2}
\end{align*}
\]
Figure 3.2. A typical hydraulic network.
Hence, (3.6) determines the flow distribution throughout the system. Substituting \( Q_{j,s} \) into \( f_j \) yields

\[
f_j = \sum_{k=1}^{J_N} \phi(H_{j_k}, H_{s}) + C_j = 0, \quad j = 1, 2, \ldots, J_N
\]  

Equation (3.7) is known as a node equation. With \( J_N \) equations and \( J_N \) unknowns, the system is readily solved providing the flow across the system is given a priori.

### 3.3.2 Loop Principle

With the loop principle, the formulation is concentrated on the head-losses around any closed loop. That is,

\[
f_l = \sum_{s=1}^{S_l} h_{l,s} = 0, \quad l = 1, 2, \ldots, L_N
\]  

in which \( f_l \) denotes the unbalanced head around the closed loop and which must equal zero at the solution; \( L_N \) is the total number of closed loops throughout the system. For pipe networks, since \( h_{l,s} \) is a function of \( Q_{l,s} \), it is valid to write

\[
h_{l,s} = \varphi(Q_{l,s})
\]  

Substituting \( h_{l,s} \) into \( f_l \) yields

\[
f_l = \sum_{s=1}^{S_l} \varphi(Q_{l,s}) = 0, \quad l = 1, 2, \ldots, L_N
\]  

To maintain flow continuity at nodal junctions, it is necessary to introduce a variable \( q \) to the equation above. The \( q \) is known as the loop flow correction and is incorporated into (3.10) as follows:
Equation (3.11) is known as a *loop equation*. The sign of $\pm q_i$ follows the sense of the associated loop. Note that $Q_{i,s}$ can be taken as the initial approximation to the flow; it is updated by applying the flow correction factor $q$. It follows that whenever the value of $q_i$ is changed, continuity of flow at nodal junctions is always maintained and satisfied. With $L_N$ equations and $L_N$ unknowns, the system can be solved.

With the nodal and loop principles, formulation of the system equations for analysing open channel networks can be derived in a similar fashion. Unlike pipe network problems, the flow in open channels is complicated by an additional variable, namely water depth, hence, $h_f = f(Q, y)$. The most direct approach is to adopt the standard step principle in conjunction with either of the two principles outlined above. Although the standard step principle is suggested for formulating the system equations, it needs not be used to compute a water surface profile. The details are described in the following sections.

3.4 Applications of the Nodal Method

3.4.1 Wylie's Approach

Wylie (1972) adopted the nodal principle in conjunction with standard step principle to derive a system of equations for analysing island flow. Firstly, the method requires a standard step to be re-arranged in terms of energy heads so that the flows across the strings are the dependent variable. Secondly, the flows are eliminated by substituting the corresponding flow equations into the equation of flow continuity at nodal junctions. Consequently, the resulting equations consist of a system of unknown energy heads only. With appropriate boundary conditions, such as total system flow, the system is solved for the nodal heads, and thence the flow distribution for the given problem.
Consider a standard step which is applied to a string

\[ H_{s1} = H_{s2} + \overline{S_f} \delta x_s, \quad s = 1, 2, \ldots, S_N \]  

(3.12)

where

\[ \overline{S_f} = \text{average friction slope associated with string } s \]
\[ \delta x_s = \text{length of the channel} \]
\[ S_N = \text{total number of strings} \]

If \( \delta x_s \) is considered too long to be accurately modelled in one reach, it can be sub-divided into a number of subreaches. Each sub-division boundary has a common total head and water depth associated with it.

Wylie assumed the \( \overline{S_f} \) is averaged by the friction slopes at the sections under consideration, that is,

\[ \overline{S_f} = \frac{1}{2} (S_{f,1} + S_{f,2}) \]  

(3.13)

and used Manning equation

\[ S_f = \frac{n^2 Q^2}{A^2 R^{4/3}} \]  

(3.14)

to calculate \( S_{f,1} \) and \( S_{f,2} \). Substituting (3.14) into (3.12), and solving for \( Q_s \) yields

\[ Q_s = \sqrt{\frac{2(H_{s1} - H_{s2})}{\delta x_s}} \sqrt{\frac{n_{s1}^2}{A_{x1}^2 R_{x1}^{4/3}} + \frac{n_{s2}^2}{A_{x2}^2 R_{x2}^{4/3}}}, \quad s = 1, 2, \ldots, S_N \]  

(3.15)

To develop the system of equations for a solution, the \( Q_s \) in the equation above is eliminated by using the equation of continuity of flow at nodal junctions. Hence, substituting (3.15) into (3.5) yields
\[ f_j = \sum_{s=1}^{s_1} \pm \sqrt{\frac{2(H_{s_1} - H_{s_2})}{\delta x_s}} \sqrt{\frac{n^2_{s_1}}{A^2_{s_1}} + \frac{n^2_{s_2}}{A^2_{s_2}}} + C_j = 0, \quad j = 1, 2, \ldots, J_N \] (3.16)

or in terms of nodal heads

\[ f_j = \sum_{k=1}^{s_k} \pm \sqrt{\frac{2(H_j - H_k)}{\delta x_s}} \sqrt{\frac{n^2_{s_1}}{A^2_{s_1}} + \frac{n^2_{s_2}}{A^2_{s_2}}} + C_j = 0, \quad j = 1, 2, \ldots, J_N \] (3.17)

It is important that the sign ± is positive if the flow is directed towards a nodal junction; otherwise, the sign is assumed negative. Equation (3.17) is the system of equations which is derived to determine the energy heads at nodal junctions. When the nodal heads are found, the flow distribution throughout a network can be obtained.

If minor head-loss is of particular importance, it can be formulated as follows:

\[ H_{s_1} = H_{s_2} + \frac{Q^2}{2g} \left( \frac{\alpha_{s_1}}{A^2_{s_1}} - \frac{\alpha_{s_2}}{A^2_{s_2}} \right) \] (3.18)

According to Wylie, the value of \( \alpha \) is positive for a flow area expansion and negative for a contraction. If it is further assumed that the water stage at some point at each nodal junction is also common to all branches, then (3.18) may be used to represent losses at branch flow junctions.

In order to illustrate the mechanism of the above analysis, the network system shown in Figure 3.2 is used. For simplicity, each string in the network comprises a single open channel module. Since the system consists of four strings and four nodal junctions, it must give rise to a system of four node equations. For brevity, the minor losses are neglected in the illustration, and the node equations are:

\[ f_j = C_j - \sqrt{\frac{2(H_j - H_{s_1})}{\delta x_s}} \sqrt{\frac{n^2_{s_1}}{A^2_{s_1}} + \frac{n^2_{s_2}}{A^2_{s_2}}} \]
For subcritical flow computations, the primary control is situated downstream of the network, that is at the $j+3$th nodal junction. With the water depth and the total flow at the $j+3$th nodal junction given, the total energy head at that point can be calculated. Hence, the $f_{j+3}$ is effectively eliminated from the system above; and thus, $f_j$, $f_{j+1}$ and $f_{j+2}$ form a set of simultaneous non-linear equations in three unknowns $H_j$, $H_{j+1}$ and $H_{j+2}$. When these energy heads are known the flows across the channels can be determined from (3.15).

Because the system of equations is non-linear, a direct solution is not possible, and it must be solved iteratively. Wylie adopted the simultaneous solution of the Newton-Raphson method and found that the procedure converged very fast to a solution. Normal prismatic or non-prismatic channels may be handled as well as concentrated effects such as bridge...
restrictions, eddy losses, etc. Although the method may be used to determine a water surface profile, it is not recommended because various procedures such as those described in Chapter 2 are generally available and far more efficient.

If determination of the flow distribution is the prime objective, then Wylie's method requires an additional stage to first determine the nodal heads and then use the nodal heads to determine the flow distribution throughout the network. It is more straightforward to calculate the energy head using water depth and discharge, rather than the other way round, using the energy head to calculate the water depth and discharge. This results in increasing the computational time and unnecessary complications.

3.4.2 Chaudhry and Schultes' Approach

Unlike Wylie's method, Chaudhry and Schulte (1987) derived a solution which involves a system of equations in terms of commonly used variables, such as water depths and discharges, in place of energy heads. This approach to formulation is, in a sense, more desirable because it allows a direct access to information such as water depth for practical design.

For solving bifurcating river flow problems, Chaudhry and Schulte also adopted the standard step principle to formulate a system of equations. The way in which the equations are formulated allows the equations to be simultaneously solved for both water depth and flow across the strings. However, the method leads to a very large set of equations even for a very small network problem. Using Figure 3.2 as an illustrative example, the method will generate twelve equations and twelve unknowns. The latter include the water depth and flow across the strings. In order to show the mechanism of the formulation, a brief description of the procedure is given below.

Firstly, identify all the strings within the network, and then consider balancing the energy within each of the strings. That is, write the standard step to ensure the law of conservation of energy holds. Four strings (channels) are found in the network so that the standard step for each of the strings can be written as
\[ f_s = H_{s,1} - H_{s,2} - h_p = 0 \]
\[ f_{s+1} = H_{s+1,1} - H_{s+1,2} - h_{p+1} = 0 \]
\[ f_{s+2} = H_{s+2,1} - H_{s+2,2} - h_{p+2} = 0 \]
\[ f_{s+3} = H_{s+3,1} - H_{s+3,2} - h_{p+3} = 0 \]  
\tag{3.19}

in which \( f_s \) denotes the unbalanced head of the \( s \)th string. On expanding the (3.19) in terms of water depths and discharges, (3.19) becomes

\[ f_s = (z_{s,1} - z_{s,2}) + (y_{s,1} - y_{s,2}) + \left( \frac{\alpha_{s,1} Q_{s,1}^2}{2 g A_{s,1}^2} - \frac{\alpha_{s,2} Q_{s,2}^2}{2 g A_{s,2}^2} \right) - h_p = 0 \]
\[ f_{s+1} = (z_{s+1,1} - z_{s+1,2}) + (y_{s+1,1} - y_{s+1,2}) + \left( \frac{\alpha_{s+1,1} Q_{s+1,1}^2}{2 g A_{s+1,1}^2} - \frac{\alpha_{s+1,2} Q_{s+1,2}^2}{2 g A_{s+1,2}^2} \right) - h_{p+1} = 0 \]  
\tag{3.20}

\[ f_{s+2} = (z_{s+2,1} - z_{s+2,2}) + (y_{s+2,1} - y_{s+2,2}) + \left( \frac{\alpha_{s+2,1} Q_{s+2,1}^2}{2 g A_{s+2,1}^2} - \frac{\alpha_{s+2,2} Q_{s+2,2}^2}{2 g A_{s+2,2}^2} \right) - h_{p+2} = 0 \]
\[ f_{s+3} = (z_{s+3,1} - z_{s+3,2}) + (y_{s+3,1} - y_{s+3,2}) + \left( \frac{\alpha_{s+3,1} Q_{s+3,1}^2}{2 g A_{s+3,1}^2} - \frac{\alpha_{s+3,2} Q_{s+3,2}^2}{2 g A_{s+3,2}^2} \right) - h_{p+3} = 0 \]

The head-loss \( h_p \) terms in the equations above can be evaluated in a number of ways (see Chapter 2). Like Wylie's method, Chaudhry and Schultz chose \( h_p \) in the form of

\[ h_p = \frac{8x}{2L}(S_{p,1} + S_{p,2}) \]  
\tag{3.21}

The Manning formula was used to evaluate the friction slopes \( S_f \) in the equation above.

Secondly, identify the nodal junctions throughout the system, and then consider balancing the energy between the upstream and downstream strings which adjoin a common junction. This is essential because it ensures that the law of conservation of energy at nodal junctions always holds. Apart from the two end nodal junctions, \( j \) and \( j + 3 \), two other nodal junctions are found in the network, therefore, two sets of equations are needed. At the \( j + 1 \)th nodal junction, the energy equations can be written as
\[
\begin{align*}
\dot{f}_{s,s+1} &= (z_{s,2} - z_{s+1,2}) + (y_{s,2} - y_{s+1,2}) = 0 \\
\dot{f}_{s,s+2} &= (z_{s,2} - z_{s+2,2}) + (y_{s,2} - y_{s+2,2}) = 0
\end{align*}
\]

(3.22)

in which subscripts \(s\) and \(s+1\) denote the upstream and downstream strings respectively. Chaudhry and Schulte assumed that the velocity head is small enough to be negligible in comparison with the water depth so that it is omitted from the (3.22). If necessary, the velocity head component can be included in the equation above. It implies that the equations derived above are aiming to match the water levels rather than energy heads.

Similar to (3.22), the energy equations at the \(j+2\)th nodal junction are written as

\[
\begin{align*}
\dot{f}_{s+1,j+3} &= (z_{s+1,2} - z_{s+3,2}) + (y_{s+1,2} - Y_{s+3,2}) = 0 \\
\dot{f}_{s+2,j+3} &= (z_{s+2,2} - z_{s+3,2}) + (y_{s+2,2} - y_{s+3,2}) = 0
\end{align*}
\]

(3.23)

For subcritical flow computation, the minor head loss is usually small. If it is of particular importance, then it may also be included in the (3.22) and (3.23). This minor loss takes the form of

\[
h_{fr,s+1} = \frac{Q^2}{2g} \left( \frac{\alpha_s}{A^2} - \frac{\alpha_{s+1}}{A^2} \right)
\]

(3.24)

in which subscripts \(s\) and \(s+1\) in \(h_{fr,s+1}\) denote the upstream and downstream channels respectively. The value of \(\alpha\) is positive for a flow area expansion and negative for a contraction.

Finally, write down the continuity of flow at nodal junctions, that is,

\[
\begin{align*}
\dot{f}_{s+1} &= Q_s - Q_{s+1} - Q_{s+2} = 0 \\
\dot{f}_{s+2} &= Q_{s+1} + Q_{s+2} - Q_{s+3} = 0
\end{align*}
\]

(3.25)

A total of twelve unknowns \(Q_s, Q_{s+1}, Q_{s+2}, Q_{s+3}; y_{s,1}, y_{s+1,1}, y_{s+2,1}, y_{s+3,1}; \) and \(y_{s,2}, y_{s+1,2}, y_{s+2,2}, y_{s+3,2}\) appear in ten equations \(f_s, f_{s+1}, f_{s+2}, f_{s+3}; f_{s,s+1}, f_{s,s+2}, f_{s,s+3}, f_{s+1,s+2}, f_{s+1,s+3}; \) and \(f_{j+1}, f_{j+2}.\) This implies that two boundary conditions or additional equations are necessary to obtain a unique solution. For subcritical flow computations, the primary
control is situated downstream of the network, that is at the \( j+3 \)th nodal junction. When the water depth and the total flow at the \( j+3 \)th nodal junction are given as boundary conditions, then the additional equations describing the water stage at the control section can be derived. That is, write the equations in the following form

\[
\begin{align*}
    f_{b1} &= y_{s+3,2} - y_o = 0 \\
    f_{b2} &= Q_{s+3} - C_{j+s3} = 0
\end{align*}
\]  

(3.26)

in which the \( y_o \) and \( C_{j+s3} \) denotes the control water depth and the total system flow respectively.

If the channel is considered too long to be accurately modelled in one reach, it can be subdivided into a number of subreaches. Between two consecutive sub-division boundaries, the energy equation for each subreach can be written as

\[
\begin{align*}
    f_i = (z_{i,i} - z_{i,i+2}) + (y_{i,i} - y_{i+1,i}) + \left( \frac{\alpha_{i,i}Q_{i,i}^2}{2gA_{i,i}^2} - \frac{\alpha_{i,i+1}Q_{i+1,i}^2}{2gA_{i+1,i}^2} \right) - h_f = 0
\end{align*}
\]

(3.27)

in which subscript \( i \) in \( f_i \) denotes the subreach section. When the channel is sub-divided into \( N_s \), an additional \( 2N_s \) equations will be formed. Similar to (3.27), the energy equation and the continuity equation can be written for each of the \( N_s \) subreaches, resulting in the following set of equations:

\[
\begin{align*}
    f_i = (z_{i,i} - z_{i,i+2}) + (y_{i,i} - y_{i+1,i}) + \left( \frac{\alpha_{i,i}Q_{i,i}^2}{2gA_{i,i}^2} - \frac{\alpha_{i,i+1}Q_{i+1,i}^2}{2gA_{i+1,i}^2} \right) - h_f = 0, \quad i = 1, 2, ..., N_s \tag{3.28a}
\end{align*}
\]

and

\[
\begin{align*}
    f_i &= Q_{i,i} - Q_{i+1,i} = 0 \quad i = 1, 2, ..., N_s \\
    f_i &= y_{i,i} - y_{i+1,i} = 0
\end{align*}
\]

(3.28b)

With the system of equations formed, equations of (3.28) are solved simultaneously with (3.20) to (3.26). Chaudhry and Schulte adopted the simultaneous solution of the
Newton-Raphson method to obtain a solution. It was reported that the method is accurate and efficient and can be used for water surface profile computations (see Chapter 2).

3.6 Applications of the Loop Method

The network solution proposed by Chaudhry and Schulte is direct and easy to formulate but it suffers from a serious problem of producing a very large set of node equations even though the problem in hand is very small. To solve a large set of equations can become time consuming and require a large amount of computer storage space for data manipulations. Since the equations are non-linear, the solution must be obtained iteratively. In general, the size of the domain of convergence is inversely related to the degree and number of equations. Depending on the configuration of the network, the larger the set of equations, the smaller the size of the domain of convergence. Consequently, the solution process might exhibit certain difficulties in convergence and might not converge at all if the initial approximation were inappropriately chosen.

In order to reduce the set of equations to a more manageable size, 'loop formulation' is proposed. The idea of the loop formulation is to eliminate the string flows by assuming a flow distribution which must satisfy the continuity of flow throughout the system. The head-loss for each string is then calculated separately from the point of separation to the point of union. If there exists a discrepancy in energy head at the point of union, then the flow distribution is said to be unbalanced. Should this be the case, the flow distribution will be corrected and the entire procedure continued until the discrepancy is zero. In order to show the mechanism of loop formulation, Figure 3.2 is used for illustration and the corresponding procedure is described below.

Firstly, identify how many closed loops exist in the network. According to Figure 3.2, there exists only one loop, which comprises two strings $s+1$ and $s+2$. With the loop formulation, string $s$ and $s+3$ are redundant strings which are not considered in the formulation because they are not part of a 'loop' within the network. The loop is then designated with the loop flow correction $q_l$. This example assumes that anti-clockwise is positive so that if the flow across the string follows the anti-clockwise sense, then $q_l$ will be designated positive; otherwise, $q_l$ will be designated negative (see Figure 3.2). This step is to ensure that continuity of flow at nodal junctions is satisfied. Once all the loops are traced and designated, the formulation can begin.
Secondly, it is necessary to ensure that the algebraic sum of head-losses around a closed loop is zero. Hence, the loop equation be written as

\[ f_l = h_{j+1} - h_{j+2} = 0 \]  

(3.29)

in which \( f_l \) denotes the discrepancy of head-loss around the closed loop and which must equal zero at the solution; \( h_{j+1} \) and \( h_{j+2} \) are the head-losses:

\[
\begin{align*}
    h_{j+1} &= z_{s+1,1} - z_{s+1,2} + y_{s+1,1} - y_{s+1,2} + \frac{\alpha_{s+1,1}(Q_{s1} - q_l)^2}{2gA_{s+1,1}^2} - \frac{\alpha_{s+1,2}(Q_{s1} - q_l)^2}{2gA_{s+1,2}^2} \\
    h_{j+2} &= z_{s+2,1} - z_{s2,2} + y_{s+2,1} - y_{s+2,2} + \frac{\alpha_{s+2,1}(Q_{s2} + q_l)^2}{2gA_{s+2,1}^2} - \frac{\alpha_{s+2,2}(Q_{s2} + q_l)^2}{2gA_{s+2,2}^2} 
\end{align*}
\]  

(3.29a)

respectively. Note that \( Q_{s1} \) and \( Q_{s2} \) are no longer independent variables but constants, and their values can be arbitrary assumed providing that continuity of flow throughout the system is satisfied.

The one equation (3.29) contains five unknowns, \( y_{s+1,1}, y_{s+1,2}, y_{s+2,1}, y_{s+2,2}; \) and \( q_l \). This implies that there is not enough information to obtain a solution. However, there is enough information to evaluate the unknowns, \( y_{s+1,1}, y_{s+1,2}, y_{s+2,1}, y_{s+2,2}, \) providing the boundary conditions are given. For subcritical computation, the control section is situated at the downstream end of the network. If the system discharge and the water depth at \( j+3 \)th nodal junction are given as boundary conditions, then the \( y_{s+1,1}, y_{s+1,2}, y_{s+2,1}, y_{s+2,2}, \) can be evaluated as follows.

When the boundary conditions are given at \( j+3 \), the water surface profile along the \( s+3 \)th string can be calculated and therefore the energy head \( H_{s+3,1} \) at the upstream end of that string is found. By equating

\[ H_{s+3,1} = H_{s+2,1} = H_{s+2,2} \]  

(3.30)

\( y_{s+2,1} \) and \( y_{s+2,2} \) can therefore be calculated. It follows that the \( y_{s+1,1} \) and \( y_{s+1,2} \) can be found too. Hence, \( y_{s+1,1}, y_{s+1,2}, y_{s+2,1}, y_{s+2,2} \) are effectively eliminated from the (3.29a).
The implication of (3.30) is that it matches the energy heads strictly at nodal junctions rather than matches the water depths (see (3.22) and (3.23)). If, however, the velocity head is considered negligible compared with the water depth, then the following expression may be used in place of (3.30):

\[ y_{s+3,1} = y_{s+1,2} = y_{s+2,2} \]  

(3.31)

The minor head-losses can also be included in (3.30) and (3.31) in order to perform a realistic analysis on the practical situation.

Proceeding to this point, the formulation above involves not one but two procedures. The first involves finding the water depths to calculate the sum of the head-loss around the closed loop. The second involves finding the appropriate value of \( q \) to update the flows across the strings in order to achieve a better approximation to (3.29). This approach provides a flexibility that other methods cannot match.

i) The computation of a water surface profile is not restricted to the standard step method as in the case of Chaudhry and Schulte's' method but some other methods (see Chapter 2) which are far more accurate and efficient may be used in place. This implies that the water surface profile can be determined independently from the system solution.

ii) The loop formulation can effectively eliminate one set of unknowns, namely the flows across the strings, while still maintaining continuity of flow throughout the system. With fewer unknowns, the number of equations required to obtain a solution is reduced. It is a matter of fact that, using Figure 3.2 as an illustrative example, the final set of equations contains only one equation compared with twelve required by Chaudhry and Schulters' method. In general, one loop will give rise to one equation only. Hence, the data storage requirement is significantly reduced and manipulations of data becomes more efficient.

iii) The loop formulation has better convergent properties than the node formulation. It has been shown (Brameller et al, 1976) in pipe network problems that methods based on loop formulation, such as the Hardy-Cross method, are superior in convergence to node formulation. From this point alone, it is evident that the loop formulation has a distinct advantage over the node formulation in this type of iterative problem.
iv) With the loop formulation, the entire set of loop equations will remain unchanged unless the layout of the network is modified. Any changes which involve altering parameters within strings are easily accommodated. This does not affect the overall formulation of the network because strings are treated as discrete entities. For example, in Chaudhry and Schulters' method, if a channel is subsequently split into two subreaches, matrices defining the another equation is added and the whole network have to be re-assembled; the solution matrix has to be increased in size accordingly. In the loop method, addition of an extra channel simply increases the number of modules within a given string. All that is affected is the actual head-loss calculated along that string. The network model itself does not have to be modified.

With the water depths found, the only unknown in the loop equation is \( q_i \). With one equation in one unknown, the loop equation is solvable. The loop equation can be solved iteratively using the Newton-Raphson method as in the case of Wylie and Chaudhry and Schulte solutions. Chapter 4 contains detailed consideration of different solution procedures.

### 3.7 Concluding Remarks

The concept of modules and strings is fundamental and is introduced as the basic 'building blocks' to represent the physical characteristics of network. This enables a wide range of hydraulic modules to be incorporated in the network analysis in conjunction with the loop principle. This formulation disregards the composition or even the complexity of the strings as soon as the head-flow characteristics for each string exist for a network solution. The idea is to formulate a set of system of equations which solely depend on the loop flow correction factors. The method predicts the head-losses for different flow distributions, and then calculates the correction to the flow along each string in order to obtain a solution.

This formulation has several advantages over node formulation, in particular with regard to Wylie's method and Chaudhry and Schulters' method.

i) It will not create a large set of equations which need solving; only one per loop equation.
ii) Individual modules can be inserted into strings without the need to reformulate the whole solution procedure for the network. All that is affected is the head-loss calculation for the strings concerned.

iii) The concept is general; it allows any hydraulic structures to be incorporated in the analysis providing they have a unique flow to head-loss relationship.

The solution of the loop equation is analogous to that of Hardy-Cross; it assumes a flow distribution between strings, and then solves each equation separately in order to obtain a flow correction for each string. Instead of solving the equations separately, the loop equations can be solved simultaneously based on a Newton-Raphson method. The method by which this is achieved uses the relationship between head-loss and discharge of each of the individual modules in a string. With this approach, there is no limitations in the concept from which the hydraulic networks can be analysed.
CHAPTER 4

METHODS OF SOLUTION OF NETWORKS
4 METHODS OF SOLUTION OF NETWORKS

4.1 Introduction

Two fundamental principles have been described to formulate the system equations for analysing hydraulic networks. If the equations are formulated in terms of head-losses, then the set of equations is known as node equations. If the equations are formulated in terms of flow rates, then the set of equations is known as loop equations. Since both sets of equations are non-linear in nature, they cannot be solved directly, and must be solved iteratively.

An early method for solving pipe networks is the Hardy-Cross method (1936). Because of its simplicity it can be easily implemented for machine computations, and computer programs have been developed (Dillingham, 1967). Unfortunately, the method suffers from a serious problem of convergence even though some enhancement schemes have been developed to improve the convergence rate (Williams, 1973).

The Newton-Raphson method was subsequently applied to water distribution systems. Unlike the Hardy-Cross method, the Newton-Raphson method iterates on the whole set of equations simultaneously whereas the Hardy-Cross method iterates on separate equations, one at a time. It was reported that the Newton-Raphson method converges rapidly to a solution providing certain conditions are satisfied (Warga, 1954). This rapid converging property has generated much interest in the field of solving pipe network problems (Sharmir and Howard, 1968; Lam and Wolla, 1972a). Unfortunately, the method may not converge at all if the initial approximation is not sufficiently close to the root; the method also requires first-order partial derivatives. To overcome the first problem, a number of subsidiary methods were developed to estimate the set of initial approximations (Freudenstein and Roth, 1963). In theory, they converge to the solution with arbitrary initial conditions; however, they may be prohibitively slow. The second
problem of estimating partial derivatives may be overcome by numerical means but it may affect the rate of convergence of Newton-Raphson method.

In order to tackle these problems, Broyden (1965) presented a class of methods based on the Newton-Raphson method for solving a set of simultaneous non-linear equations without requiring partial derivatives. It was reported that the method works very well with pipe network problems even though the method converges slower than the Newton-Raphson method; the requirement for the initial approximation is more relaxed than for the Newton-Raphson method.

Regarding applications to open channel network problems, two existing computer programs, CHAT (Cotton et al, 1987) and RIVER4 (Smith and Ashenhurst, 1986), adapted the Hardy-Cross principle to solve bifurcating systems. Experience shows that these programs suffer from convergence problems because the methods iterate on separate loop equations, one at a time. On the other hand, Wylie (1972) and Chaudhry and Schulter (1987) used the Newton-Raphson method for solving island-flow problems (open channels only) and reported no difficulties in obtaining a solution. The way by which the system was formulated generated a very large set of node equations even though the problem in hand was small. In general, the size of the domain of convergence is inversely related to the degree and number of equations. The larger the set of equations, the smaller the size of the domain of convergence. Consequently, it can be very difficult to obtain an initial approximation from which the iteration converges.

In a sense, neither approaches as they stand are effective in terms of computer storage and computational time. One suggestion is to use the loop principle to formulate the system of equations, and then use the Newton-Raphson method to solve the equations. It may be more desirable to use the Broyden method, or optimisation techniques, in place of the Newton-Raphson method.

The work presented herein is two-fold. Firstly, it proposes a method of solution and investigates its applicability to networks. Secondly, it examines the convergence behaviour of the Newton-Raphson method and Broyden method in conjunction with optimisation techniques to solve the network equations. The principle objective is to develop a method which both converges and requires the minimal computational effort. The various methods are explored in details.
4.2 Newton-Raphson Method

Referring to (3.11), the equations for analysing an open channel network can be written in the general form as shown below:

\[ f_i(q_1, q_2, ..., q_{L_N}) = 0, \quad i = 1, 2, ..., L_N \]  \hspace{1cm} (4.1)

in which \( q_j \) for \( j = 1, 2, ..., L_N \) are loop correction factor. The solution of a network requires that all the non-linear equations in (4.1) equal zero simultaneously. Equation (4.1) may be written more concisely in matrix notation

\[ \mathbf{f}(\hat{\mathbf{q}}) = 0 \]  \hspace{1cm} (4.2)

in which \( \hat{\mathbf{q}} \) denotes the column vector of independent variables \( q_j \) at the solution, and \( \mathbf{f} \) is the column vector of functions \( f_i \). Expanding \( \mathbf{f} \) into a Taylor's series about \( \mathbf{q} \)

\[ \mathbf{f}(\hat{\mathbf{q}}) = \mathbf{f}(\mathbf{q}) + \frac{\partial \mathbf{f}(\mathbf{q})}{\partial \mathbf{q}} (\hat{\mathbf{q}} - \mathbf{q}) + \cdots \]  \hspace{1cm} (4.3)

Neglecting the second and higher-order derivative terms and hence reducing \( \mathbf{f}(\hat{\mathbf{q}}) \) into a linearised problem

\[ \mathbf{f}(\hat{\mathbf{q}}) = \mathbf{f}(\mathbf{q}) + \frac{\partial \mathbf{f}(\mathbf{q})}{\partial \mathbf{q}} (\hat{\mathbf{q}} - \mathbf{q}) \]  \hspace{1cm} (4.4)

At the solution, \( \mathbf{f}(\hat{\mathbf{q}}) \) equals zero

\[ 0 = \mathbf{f}(\mathbf{q}) + \frac{\partial \mathbf{f}(\mathbf{q})}{\partial \mathbf{q}} (\hat{\mathbf{q}} - \mathbf{q}) \]  \hspace{1cm} (4.5)

which solves to

\[ \hat{\mathbf{q}} = \mathbf{q} - \left( \frac{\partial \mathbf{f}(\mathbf{q})}{\partial \mathbf{q}} \right)^{-1} \mathbf{f}(\mathbf{q}) \]  \hspace{1cm} (4.6)
If \(q_n\) is the \(n\)th approximation to the solution of (4.2) and \(f_n\) is written for \(f(q_n)\), then the method of Newton-Raphson (NR) is defined by the iterative formula

\[
q_{n+1} = q_n + \delta_n
\]

(4.7)

where

\[
\delta_n = H_n f_n
\]

(4.8)

and the inversion of \(J_n\) is

\[
H_n = -J_n^{-1}
\]

(4.9)

The term \(J\) is known as the Jacobian matrix whose elements are given by

\[
[J_{i,j}] = \left[ \frac{\partial f_i}{\partial x_j} \right], \quad \{i = 1, 2, \ldots, L_N\}, \quad \{j = 1, 2, \ldots, L_N\}
\]

(4.10)

The conditions for convergence require firstly, that the initial approximation \(q_0\) is sufficiently close to the solution \(\hat{q}\); secondly, that \(f_n\) is continuous and at least has first-order partial derivatives available at \(q_n\); and finally, that \(J_n\) is invertible. Otherwise, the method of NR would perform poorly. If the second-order partial derivatives of \(f_n\) are continuous, then it can be shown that

\[
\frac{\|q_{n+1} - \hat{q}\|}{\|q_n - \hat{q}\|} \rightarrow \kappa
\]

(4.11)

(Pike, 1986) for some constant \(\kappa > 0\) and sufficiently large \(n\). This means that the rate of convergence is quadratic or second-order; the number of correct decimal places approximately doubles with each iteration. The proof of such convergence characteristic can be found in Pike (1986) and Fletcher (1987). This strong convergence property makes NR an attractive method.

\(\oplus\) \(\| \cdot \|\) denotes the Euclidean length or norm.
Given a system of $L_N$ non-linear equations with $L_N$ unknowns, and an initial approximation $q_o$ for a solution of the system, the steps to obtain a solution are shown in Figure 4.1. The procedure is applied repetitively until $\|f_{n+1}\|$ satisfies the prescribed tolerance, $\xi$.

4.3 Partial Derivatives for Newton-Raphson Method

As aforementioned, the method of NR relies on the $J$ to be invertible. Each entry in the $J$ is the partial derivative of $f$ with respect to $q$. It can be shown that even if the $f$ is sufficiently simple, then the partial derivatives of the $J$ can be extremely complicated to obtain analytically. This offers many opportunities for making mistakes, both in the derivation and in the coding of the entries of $J$.

An alternative approach is to obtain the derivatives by numerical differentiation. Although the method gives an immediate estimation of the derivatives, it suffers from two serious problems. Firstly, it requires the $f$ to be evaluated by at least $L_N + 1$ sets of $q$. This can be excessive as far as computational time is concerned. Secondly, it exhibits some numerical instability if the step size is not cautiously chosen. This instability usually cause less rapid convergence.

4.3.1 Analytical Derivation

Expanding $f_i$ in (4.1) in term of head-loss

$$f_i = \sum_{r=1}^{S_i} \pm h_r = 0, \quad i = 1, 2, ..., L_N$$

(4.12)

in which $S_i$ denotes the total number of strings associated with the $i$ th loop. The sign $\pm$ designates whether the string flow $Q_s$ follows the predefined flow direction of $q_i$; a positive $+h_r$ means $Q_s$ has the same direction as $q_i$; a negative $-h_r$ means $Q_s$ has a direction opposite to $q_i$. 

104
Figure 4.1. Flow chart for the Newton-Raphson method.
Differentiating \( f_i \) with respect to \( q_j \) yields

\[
\frac{\partial f_i}{\partial q_j} = \sum_{s=1}^{S} \pm \frac{\partial h_i}{\partial q_j}, \quad \left\{ \begin{array}{l} i = 1, 2, \ldots, L_N \\ j = 1, 2, \ldots, L_N \end{array} \right. \tag{4.13}
\]

or in terms of energy heads

\[
\frac{\partial f_i}{\partial q_j} = \sum_{s=1}^{S} \pm \left( \frac{\partial H_{s,1}}{\partial q_j} - \frac{\partial H_{s,2}}{\partial q_j} \right), \quad \left\{ \begin{array}{l} i = 1, 2, \ldots, L_N \\ j = 1, 2, \ldots, L_N \end{array} \right. \tag{4.14}
\]

in which the subscripts in \( H_{s,1} \) and \( H_{s,2} \) denote the upstream and downstream sections of the \( s \) th string respectively.

Consider the upstream energy head, \( H_{s,1} \), for the \( s \) th string separately

\[
H_{s,1} = z_{s,1} + \frac{\alpha_{s,1} \left( Q_s + \sum_{j=1}^{L} \pm q_j \right)^2}{2 g A_{s,1}^2} \tag{4.15}
\]

The derivative with respect to \( q_j \) is

\[
\frac{\partial H_{s,1}}{\partial q_j} = \frac{\alpha_{s,1} (Q_s \pm q_j)}{g A_{s,1}^2} + \left( 1 - \frac{\alpha_{s,1} (Q_s \pm q_j)^2}{g A_{s,1}^2} \right) \frac{\partial A_{s,1}}{\partial q_j} \frac{\partial y_{s,1}}{\partial q_j} \tag{4.16}
\]

The inclusion of water depth in the above differentiation is necessary because the water depth in channel flow will change accordingly as the flow changes. Equating (4.16) in terms of the Froude number gives

\[
\frac{\partial H_{s,1}}{\partial q_j} = \frac{\alpha_{s,1} (Q_s \pm q_j)}{g A_{s,1}^2} + \left( 1 - F_{s,1}^2 \right) \frac{\partial y_{s,1}}{\partial q_j} \tag{4.17}
\]
In the first term of (4.17), the rate of change in $H_{s,1}$ is caused by the change in $q_j$; in the second term, the change in $q_j$ causes a change in $y_{s,1}$, which consequently causes a second change in $H_{s,1}$. The sum of these two components gives the total change in $H_{s,1}$ with respect to a unit change in $q_j$.

With the same approach, the derivative for the downstream energy head, $H_{s,2}$, is

$$
\frac{\partial H_{s,2}}{\partial q_j} = \pm \frac{\alpha_{s,2}(Q_s \pm q_j)}{g A_{s,1}^2} + (1 - Fr_{s,2}^2) \frac{\partial y_{s,2}}{\partial q_j} \quad (4.18)
$$

Assuming that the $H_{s,1}$ is independent of $H_{s,2}$, the difference between the two derivatives gives the change in head loss caused by the $q_j$. Subtracting (4.16) from (4.18) yields

$$
\frac{\partial h_{h}}{\partial q_j} = \mp \left( \frac{Q_s \pm q_j}{g} \right) \left( \alpha_{s,1} A_{s,1}^2 - \alpha_{s,2} A_{s,2}^2 \right) + \left[ (1 - Fr_{s,1}^2) \frac{\partial y_{s,1}}{\partial q_j} - (1 - Fr_{s,2}^2) \frac{\partial y_{s,2}}{\partial q_j} \right] 
$$

Note that the resulting equation involves two derivative terms $\partial y_{s,1}/\partial q_j$ and $\partial y_{s,2}/\partial q_j$, which are describing the response rate of the water depths with respect to a unit change in $q_j$. Unfortunately, none of them can be evaluated immediately because the flow characteristic for the $s$th string is undefined at present. However, one will be available when the $s$th string is defined during the execution of the program.

However, the $H_{s,1}$ is not completely independent of $H_{s,2}$, since $y_{s,1}$ is a function of $y_{s,2}$ and is related by the gradually varied flow equation. Thus,

$$
\frac{\partial y_{s,1}}{\partial q_j} = f \left( \frac{\partial y_{s,2}}{\partial q_j} \right) \quad (4.20)
$$

This situation arises frequently in open channel analysis. To evaluate (4.20), all the modules associated with the $s$th string and the flow characteristics of each module and its derivatives must be known. All these flow characteristics are then combined to form a composite mathematical expression to characterise the flow for the $s$th string. With some modules the implementations of the derivatives can become exceedingly complicated.
METHODS OF SOLUTION OF NETWORKS

Taking a typical case where the $s$th string is an open channel, the relationship between $y_{s,1}$ and $y_{s,2}$ is governed by the gradually varied flow equation

$$
S_w = \frac{dy}{dx} = \frac{S_e - S_f}{1 - Fr^2}
$$

(4.21)

in which $S_w$ denotes the slope of water surface relative to bed at the point of interest.

Integrating $S_w$ for the $s$th string from distance $x_2$ to distance $x_1$ gives

$$
y_{s,1} = y_{s,2} + \int_{x_2}^{x_1} S_w \, dx
$$

(4.22)

which defines the water surface profile. Consider the same integration but with the rate of flow along the channel increased by $\delta q$

$$
\tilde{y}_{s,1} = \tilde{y}_{s,2} + \int_{x_2}^{x_1} \tilde{S}_w \, dx
$$

(4.23)

in which $\tilde{y}_{s,1}$ and $\tilde{y}_{s,2}$ denote the new water depths caused by the increased in flow; and the $\tilde{S}_w$ denotes the new water surface slope. The difference of the two integrations (4.22) and (4.23) provides a way to estimate the sensitivity of the water surface profile with respect to the flow. With this approach, differences in initial water-depth can be included. Subtracting (4.22) from (4.23) yields

$$
\delta y_{s,1} = \delta y_{s,2} + \int_{x_2}^{x_1} \delta S_w \, dx
$$

(4.24)

where

$$
\delta y_{s,1} = \tilde{y}_{s,1} - y_{s,1}
$$

$$
\delta y_{s,2} = \tilde{y}_{s,2} - y_{s,2}
$$

$$
\delta S_w = \tilde{S}_w - S_w
$$

Expanding $\delta S_w$ into a Taylor's series and neglecting the second and higher-order derivative terms yields
\[ \delta S_{wr} = \frac{\partial S_{wr}}{\partial q} \delta q + \frac{\partial S_{wr}}{\partial y} \delta y \]  

\text{(4.25)}

The \( \delta S_{wr} \) describes the change of water surface slope. In the first term of (4.25), the change in \( S_{wr} \) is caused by the change in \( q \) with the size of \( \delta q \); in the second term, the change in \( S_{wr} \) is caused by the change of \( y \) with the size of \( \delta y \). The total change in \( S_{wr} \) with the size of \( \delta S_{ns} \) is the sum of these terms. The \( \delta y \) term is included to provide a more realistic estimation of the total change of \( \delta S_{wr} \). Substituting (4.25) into (4.24), and dividing the resulting expression by \( \delta q \), and taking the limit as \( \delta q \to 0 \),

\[ \frac{\partial y_{z,2}}{\partial q} = \frac{\partial y_{z,2}}{\partial q} + \int_{y_{z,1}}^{y_{z,2}} \left( \frac{\partial S_{ws}}{\partial q} + \frac{\partial S_{wr}}{\partial y} \right) dx \]  

\text{(4.26)}

The integration of (4.26) may be performed numerically using the methods described in Chapter 2. The first term describes the rate of change of water surface slope caused by the change in \( q \). The second term describes the rate of change of water surface slope caused by the change in \( y \). These terms can be evaluated directly by differentiating \( S_{wr} \) with respect to \( q \) and \( y \). Thus, the only unknown in the (4.26) is the term \( \partial y_{z,2}/\partial q \). It can be obtained if one of the following conditions is met.

Firstly, if the downstream end of the \( s \)th string is directly connected to the outfall whose energy head \( H_{\text{outfall}} \) is provided as a boundary condition, then \( \partial y_{z,2}/\partial q \) can be obtained from (4.18). Because \( H_{z,2} \) is connected to \( H_{\text{outfall}} \) and which is a constant, it follows that

\[ \frac{\partial H_{z,2}}{\partial q} = \frac{\partial H_{\text{outfall}}}{\partial q} = 0 \]  

\text{(4.27)}

Hence, (4.18) becomes

\[ 0 = \pm \frac{\alpha_{z,2} (Q_{z} \pm q_{j})}{2 A_{z,1}^2} + (1 - Fr_{z,2}^2) \frac{\partial y_{z,2}}{\partial q_{j}} \]  

\text{(4.28)}
and solving (4.28) for $\partial y_{s,2}/\partial q_j$

$$\frac{\partial y_{s,2}}{\partial q_j} = \pm \frac{\alpha_{s,2} (Q_{s,2} \pm q_j)}{gA_{s,2}^2 (1 - Fr_{s,2}^2)} \qquad (4.29)$$

in which the sign $\pm$ denotes that the value of $y_{s,2}$ is decreasing with increasing $q_j$. This result is substituted into (4.26) to compute $\partial y_{s,1}/\partial q_j$. Since the values of both $\partial y_{s,1}/\partial q_j$ and $\partial y_{s,2}/\partial q_j$ are obtained, the value of $\partial h_{s}/\partial q_j$ can be found by (4.19).

Secondly, if downstream end of the $s$th string is connected to the $s+1$th string (other than the outfall) and whose direction of flow is as same as the $s$th string, then

$$\frac{\partial H_{s,2}}{\partial q_j} = \frac{\partial H_{s+1,1}}{\partial q_j} \quad (4.30)$$

Equation (4.29) assumes that the rate of change in $H_{s,2}$ is as same as that of $H_{s+1,1}$. It is known from (4.18) that both $\partial H_{s,2}/\partial q_j$ and $\partial H_{s+1,1}/\partial q_j$ exist. Hence, equating both terms and solving for $\partial y_{s,1}/\partial q_j$ yields

$$\frac{\partial y_{s,2}}{\partial q_j} = \pm \left( \frac{\alpha_{s+1,1} (Q_{s+1,1} \pm q_j)}{gA_{s+1,1}^2 (1 - Fr_{s+1,1}^2)} - \frac{\alpha_{s,2} (Q_{s,2} \pm q_j)}{gA_{s,2}^2 (1 - Fr_{s,2}^2)} \right) + \frac{(1 - Fr_{s,2}^2)}{(1 - Fr_{s+1,1}^2)} \frac{\partial y_{s+1,1}}{\partial q_j} \quad (4.31)$$

The recursion property of (4.30) requires that the value of $\partial y_{s+1,1}/\partial q_j$ to be available, meaning that the computation must start from the point of outfall and proceeds in the direction to upstream.

It is seen that the partial derivatives for the strings containing open channel are very complex in contrast to the pipe network problem. Because the flow in a pipe does not involve water-depth, the flow characteristics can be written in the form

$$h_j = kQ^e \quad (4.32)$$
in which \( k \) and \( e \) are constants depending upon the friction equation chosen. For a pipe network, the partial derivative for each loop can be shown to be

\[
\frac{\partial f_i}{\partial q_j} = \sum_{r=1}^{s} \pm ek(Q_{i,r} \pm q_r)^{-1}, \quad \{i = 1, 2, \ldots, L_N \}, \quad \{j = 1, 2, \ldots, L_N \}
\]

(4.33)

The additional complexity of the channel flow networks is apparent if (4.33) is compared with (4.14), (4.17), (4.18), (4.29) and (4.31). The evaluation of the Jacobian matrix in the case of open channel networks is extremely complicated.

4.3.2 Numerical Derivation

Construction of the \( J \) is exceedingly complicated if partial derivation for channel flow are derived analytically. It is therefore worthwhile exploring alternative ways of obtaining these partial derivatives. One approach is to neglect the derivative terms in (4.19). This implies that the terms involved are sufficiently small not to impair convergence. This is difficult to quantify given the large number of possible entries to \( J \).

An alternative approach to the problem is to use the technique of finite difference, (Bajpai et al, 1974). Two formuli commonly used for approximating the derivatives are the forward difference formula

\[
\frac{df}{dx} = \frac{f(x+h) - f(x)}{h} - \frac{h}{2} f''(\xi)
\]

(4.34)

and the central difference formula

\[
\frac{df}{dx} = \frac{f(x+h) - f(x-h)}{2h} - \frac{h^2}{6} f'''(\xi)
\]

(4.35)

in which \( h \) denotes the differencing interval. The numerical accuracy of these formuli differ for a given \( h \). The former has the accuracy of \( O(h) \) whereas the latter has the accuracy of \( O(h^2) \). However, this extra degree of accuracy is offset by an additional evaluation of the function. Although the approach gives an immediate estimation of the partial derivatives, it is not without its disadvantages.
Firstly, the amount of computational effort required to evaluate $f$ is proportional to $L_n + 1$ sets of $q_j$, since the $f$ is evaluated using two different values of $q_j$ while holding the remaining $L_n - 1$ number of $q_j$ constant.

Secondly, since both formulae were derived by Taylor's series expansion, the higher-order derivative terms were truncated. If those terms converge slowly because of the nature of the objective function, then the truncation error involved could be comparatively high even though a small value of $h$ is chosen for differentiation. This implies that the differentiation may be unstable.

Thirdly, if $h$ is chosen too small, then the significant digits (limited by machine precision) would be wiped out by round-off error. If the $h$ is chosen too large, then the approximation would not be sufficiently accurate to represent its analytical value. With both cases, a loss in quadratic convergence results. If, however, the $h$ is carefully chosen, then the numerical errors incurred will be minimal but cannot be fully eliminated.

A value of $h$ which minimises the total error, $e_s$, incurred in the computation is required. This can be accomplished by the following expression

$$|e_s| = |e_t| + |e_r|$$

(4.36)

where

- $e_s = \text{total error}$;
- $e_t = \text{truncation error}$;
- $e_r = \text{round-off error}$.

It can be shown that, when the forward difference formula is used, the $e_s$ is

$$|e_s| = \frac{|h|}{2} + \frac{2e_m}{|h|}$$

(4.37)

(Conte and de Boor, 1980) in which $e_m$ denotes the machine precision or the error involved in computing the function, whichever is larger. To find the value of $h$ for which
\( e_s \) is a minimum, the \( e_s \) is differentiated with respect to \( h \) and the resulting expression equates to zero. Thus, the positive solution of

\[
h_{\text{optimum}} = \sqrt{4e_m}
\]  

(4.38)

gives the optimum value of \( h \). A similar expression for the central difference formula is

\[
h_{\text{optimum}} = \sqrt[3]{3e_m}
\]  

(4.39)

A fuller description on this step size control may find in Conte and de Boor (1980).

Assuming that reasonable decisions are made on the \( h \), a generalised procedure to construct the approximate Jacobian matrix \( \mathbf{J} \) is as follows. Although the forward difference formula is chosen for illustration, the procedure can be easily adapted for the central difference formula.

The principle is to evaluate the \( f \) for two different values of \( q_j \), say \( q_j \) and \( q_j + h \), while holding the remaining \( L_N - 1 \) number of \( q_j \) constant. Each \( q_j + h \) is considered separately and substituted into \( f \) one at a time. Having covered all the \( q_j + h \), the reduction process can begin. The difference of \( f_i \) provides a way to approximate the partial derivatives, and the steps to approximate the \( J \) is illustrated below:

Firstly, evaluate

\[
f_i^{(0)} = f_i(q_1, q_2, \ldots, q_L), \quad i = 1, 2, \ldots, L_N
\]  

(4.40)

in which the superscript in \( f_i^{(0)} \) denotes the evaluation involving all the \( q_j \) values. This step corresponds to the \( f(x) \) in (4.34).

Secondly, evaluate

\[
f_i^{(j)} = f_i(q_j + h), \quad j = 1, 2, \ldots, L_N
\]  

(4.41)
METHODS OF SOLUTION OF NETWORKS

in which the superscript in \( f_i^{(j)} \) denotes the evaluation involving one \( q_j + h \) at a time while holding the remaining \( L_N - 1 \) number of \( q_j \) constant. This step corresponds to the \( f(x+h) \) in (4.34).

Finally, substitute the computed \( f_i^{(0)} \) and \( f_i^{(n)} \) into the forward difference formula, and thence the Jacobian matrix is approximated to

\[
\begin{bmatrix}
\tilde{J}_{ij}
\end{bmatrix} = \begin{bmatrix}
\frac{\partial f_i}{\partial q_j}
\end{bmatrix} = \begin{bmatrix}
\frac{f_i^{(n)} - f_i^{(0)}}{h}
\end{bmatrix}, \quad \{ i = 1, 2, \ldots, L_N \}
\]

\[
\{ j = 1, 2, \ldots, L_N \}
\]  

(4.42)

Note that the number of evaluations for \( \frac{\partial f_i}{\partial q_j} \) per iteration is \( L_N^2 \). If, however, the Jacobian matrix is known to be symmetrical about its diagonal, then the number of evaluations of \( \frac{\partial f_i}{\partial q_j} \) may be reduced to \( \frac{1}{2} L_N (L_N + 1) \).

The concept of the numerical method is more simple and direct as it replaces the step of analytically deriving the partial derivative for a particular module. It reduces the source for errors, the computer codings will be simpler, and the storage requirements reduced. However, it is offset by a greater number of evaluation of functions especially when a network involves many loops. It has been mentioned above that the time taken to successfully evaluate the Jacobian matrix is directly proportional to \( L_N^2 \). Moreover, if the step size is not carefully chosen, then the rate of convergence of the Newton-Raphson method might be impaired.

4.4 Shortcomings of Newton-Raphson Method

Although the method of NR has a very strong convergence characteristic, it suffers the following drawbacks.

i) It lacks convergence when the initial set of approximation is far from the actual solution, since, the inversion of the Jacobian matrix may be singular.

ii) It requires the Jacobian matrix to be constructed with partial differentials, which is only practicable when the functions are sufficiently simple; and it requires the inversion of \( J_n \) to solve for \( \delta_n \), this may require considerable computational effort.
These points are discussed in turn.

### 4.4.1 Initial Approximation

The initial set of approximations to be sufficiently close to the actual solution, since, it generates a correction $\delta_n$ without consideration of the significance of the magnitude of $\delta_n$. In practice, the actual solution is not known and it is very difficult to choose an initial set of approximations that is sufficiently close to the actual solution. In order to overcome this difficulty, several ideas have been put forward.

The simplest idea is to attempt is to improve the initial set of approximations. Once the approximation is considered sufficiently close to the final solution, then the method of NR is switched into operation to accelerate to the solution. This approach has been applied successfully in pipe network problems. To improve the initial set of approximations, Warga (1954) presented a method similar to that of the Hardy-Cross method with which the improvements are determined one equation at a time. Nielsen (1989) adopted the linear theory method (LT) as described in Wood and Charles (1952). The method is reported to have poor convergence properties close to the solution. On the other hand, when the initial set of approximations is far from the solution, then the LT performed far superior to the NR. Nielsen's experimental results suggested that a robust and efficient procedure can be obtained if the LT method is used in conjunction with the NR method; one step of method of LT and then one or two steps of method of NR.

### 4.4.2 Jacobian Matrix

The NR method requires the partial derivatives of $f_n$ to be assemble in the $J_n$. It has been shown in Section 4.3.1 that the derivatives of $f_n$ can be extremely complicated to derive analytically even though the $f_n$ is sufficiently simple. In view of this difficulty, the technique of finite difference shown in the Section 4.3.2 may be applied to approximate the $J_n$ in a more direct manner. However, care must be exercised in choosing the appropriate value of $h$ in order to maximise numerical accuracy.
The method of NR requires the $J_n$ and its inversion to be evaluated at each iteration. If the $J_n$ is of order $L_N^N$ by $L_N$, then a total of $L_N^2$ entries are required. Making an entries in the $J_n$ is laborious; it requires $O(L_N^2)$ multiplications and additions to perform the inversion of $J_n$. Thus, the bulk of the computational time is solely spent on the $J_n$.

The inversion of $J_n$ can be unstable especially when the iterates moving close to a minimum, because the entries of partial derivatives in the $J_n$ will tend to zero and singularity of the $J_n^{-1}$ may occur. Instead of inverting the $J_n$ directly, some factorisation methods such as the $LU$-decomposition (Golub and Van Loan, 1983) may be incorporated to solve for $\delta_n$. Once the $J_n$ is factorised, the additional operations required to solve for $\delta_n$ is of order $L_N^2$.

If there is convergence and $J_{n+1}$ differs little from $J_n$, then it is reasonable to consider evaluating the $J_{n+1}$ once for a few iterations instead of at every iteration as strictly required. When a slowdown in convergence is detected, then the $J_{n+k} = J_n$ for $k = 1, 2, ...$ is renewed and the iterative process is resumed.

4.5 Quasi-Newton Methods

To overcome the aforementioned difficulties in the NR method, Broyden (1965) suggested to modify the classical NR method. This leads to the following modification of (4.7)

$$q_{n+1} = q_n + t_n \delta_n$$

in which $\delta_n$ is the correction vector, and $t_n$ is known as the line search parameter. When $t_n$ is unity, then (4.43) is equivalent to (4.7). The importance of $t_n$ is such that it enables the NR method does not diverge when the initial set of approximation far from the final solution, and it can be used to accelerate convergence of the NR method. With the $t_n$ involved in (4.43), the iterative algorithm of the NR method can be modified as shown in Figure 4.2. The only difference between Figure 4.1 and Figure 4.2 is the additional step to select an appropriate $t_n$ value for updating $q_n$, and the details of performing line searches are described in Section 4.6.
Figure 4.2. Flow chart for the modified Newton-Raphson method.
In order to derive a method without needing to derive the partial derivatives analytically, Broyden also suggested to use an approximate Jacobian matrix $B_n$ in place of the $J_n$, that is,

$$B_n = J_n$$

(4.44)

or

$$H_n = -B_n^{-1}$$

(4.45)

By doing this, the steps for evaluating and inverting the $J_n$ at each iteration are eliminated, and much of the interest lies in finding the appropriate non-singular matrix $H_n$ for solving $\delta_n$. Such procedures are known as *quasi-Newton methods*.

The basic idea is to use the information one step ahead, that is the $n+1$th iteration, and at the $n$th iteration to approximate the Jacobian matrix. It follows that four quantities, $q_n$, $q_{n+1}$, $f_n$ and $f_{n+1}$, are available and they can be related in such a way that the following equation holds:

$$B_{n+1}(q_{n+1} - q_n) = (f_{n+1} - f_n)$$

(4.50)

On comparing the above equation with the NR iterative formula

Newton–Raphson method: $J_n(q_{n+1} - q_n) = (0 - f_n)$

quasi–Newton method: $B_{n+1}(q_{n+1} - q_n) = (f_{n+1} - f_n)$

the similarities are clear; the approximate matrix $B_{n+1}$ is one step ahead of $J_n$. By virtue of the *mean value theorem*, the $B_{n+1}$ is evaluated at some point $\zeta$ between $q_{n+1}$ and $q_n$. Unfortunately, the precise location of the $\zeta$ is not known, but it is seen that $B_{n+1} = B(q_{n+1})$ offers some predictions to points closer to the solution than those offered by $B_n = B(q_n)$, since for a converging procedure the $q_{n+1}$ is closer to the solution than $q_n$.

\[ f'(\zeta) = \frac{f(b) - f(a)}{b-a}, \quad a < \zeta < b. \]
Instead of performing the inversion of $B_{n+1}$ at each iteration, the $B_{n+1}$ is substituted by (4.45), and the following result is obtained

$$H_{n+1} \gamma_n = -t_n \delta_n$$

(4.47)

where

$$\delta_n = (x_{n+1} - x_n)/t_n$$

$$\gamma_n = f_{n+1} - f_n$$

in which $t_n$ is previously described as the line search parameter. Equation (4.47) is known as the *quasi-Newton condition*, which defines a class of matrices based upon the NR method for solving a set of non-linear simultaneous equations.

Fletcher (1987) described a method of achieving such a condition (4.47). The idea is to construct an approximation to the inversion of the Jacobian matrix by adding some corrections, $\Sigma_n$, gained on the $n$th iteration. That is to generate a sequence of matrices

$$H_{n+1} = H_n + \Sigma_n, \quad n = 0, 1, \ldots, \infty$$

(4.48)

with $H_n$ always preserved symmetrical and positive definite, whose termination property is

$$\lim_{n \to \infty} H_n = J^{-1}$$

(4.49)

The initial $H_0$ may be any arbitrary positive definite matrix; a unit matrix $I$ is used in the absence of any better estimate.

A particular solution

$$H_{n+1} = H_n - \frac{(t_n \delta_n + H_n \gamma_n) \delta_n^T H_n}{\delta_n^T H_n \gamma_n}$$

(4.50)
which satisfies the condition (4.47) was proposed by Broyden (1965), which is subsequently referred to as the BR updating formula. The method has a number of important properties such as

i) it terminates in at most \( n \) iteration with \( H_{n+1} = J^{-1} \);

ii) previous quasi-Newton conditions (4.51) are preserved.

iii) quadratic (second-order) termination for quadratic functions with positive definite Jacobian matrices;

For linear systems, property ii) is destroyed, and hence, the result \( H_{n+1} = J \) implies termination.

According to Broyden, if in particular, the \( f_n \) contains the first-order derivative information of a convex function, then solving the equations is a way of minimising the function. Under these conditions, it is known that at the solution the Jacobian matrix is both symmetrical and positive definite, so that \( H_{n+1} \) may be chosen to fulfil these requirements, thereby defining \( H_{n+1} \) as

\[
H_{n+1} = H_n - \frac{H_n \gamma_n \gamma_n^T H_n}{\gamma_n^T H_n \gamma_n} - t_n \frac{\delta_n \delta_n^T}{\delta_n^T \gamma_n}
\]  \hspace{1cm} (4.51)

This formula was proposed by Davidon (1959) and Fletcher and Powell (1963) and is known as the DFP updating formula (Fletcher and Powell, 1963). Another formula was proposed by Broyden, Fletcher, Goldfarb and Shanno (1970) and is known as BFGS updating formula (Fletcher, 1987).

\[
H_{n+1} = H_n - \frac{\delta_n \gamma_n^T H_n + H_n \gamma_n \delta_n^T}{\delta_n^T \gamma_n} - t_n \frac{\gamma_n^T H_n \gamma_n}{\delta_n^T \gamma_n} \frac{\delta_n \delta_n^T}{\delta_n^T \gamma_n}
\]  \hspace{1cm} (4.52)
According to Fletcher, the BFGS updating formula is an extension to the DFP updating formula. Therefore, the following properties for the DFP also hold for the BFGS:

i) it terminates in at most \( n \) iteration with \( H_{n+1} = J^{-1} \);
ii) previous quasi-Newton conditions (4.47) are preserved;
iii) it preserves positive definite \( H_{n+1} \) matrices, hence the descent property holds;
iv) global convergence for strictly convex functions (with exact line searches);
v) superlinear order of convergence, that is

\[
\frac{\|x_{n+1} - \hat{x}\|}{\|x_n - \hat{x}\|} \to 0
\]  

(4.53)

However, one important aspect separates the two in as much as the method of BFGS will converge to a global minimum of a strictly convex function even when inexact line searches are used. This convergence property has not been proved for the DFP method. In addition, Dixon (Press et al, 1988) has shown that the BFGS and DFP methods differ only in details such as round-off errors.

Apart from slower rate of convergence, the quasi-Newton methods have the following important features:

i) The basic algorithmic structure for the quasi-Newton methods is as same as the NR. Hence, the corresponding flow diagram for the quasi-Newton methods is constructed and shown in Figure 4.3.
ii) The quasi-Newton methods require no additional function evaluations even though the updating formulæ are far more elaborate.
iii) The steps for evaluating and inverting the Jacobian matrix are eliminated.
iv) The positive definite character of \( H_n \) is preserved. It implies that downhill minimisation of the function is asserted.
v) Because of iv), the requirements for initial set of approximations are considerably less restrictive than those required by the NR.
Figure 4.3. Flow chart for the quasi-Newton methods (BR, DFP and BFGS).
Lam and Wolla (1972b) applied the BR method for analysing piped water distribution systems. They reported that with a good initial set of approximations, the NR method converged faster, in terms of number of iterations, than the BR method. However, the computational time required by the BR method per iteration was significantly reduced by a factor of seven. With a very close initial approximation, the NR method converges quadratically whereas the BR method converges faster in term of computational time. The requirements of the initial set of approximations in the NR method are far more restrictive than those of the BR method, although the BR method still requires reasonable initial approximations; otherwise, it may not converge even though a solution exists.

For general optimisation problems, Fletcher (1987) conducted an experiment to compare the performance of the methods of the DFP and BFGS. It was found that, for small problems (less than 10 independent variables), the performance between the two methods varied little. The BFGS method converged slightly faster than the DFP method. However, for large and difficult problems (trigonometry functions with 50 independent variables), the DFP method required considerably extra effort, and has tendency to fail whereas the BFGS method did not exhibit these problems.

4.6 Line Search

All the solution methods described previously are iterative methods that generate a sequence of points, \( q_n \) for \( n = 1, 2, \ldots, L_N \), hopefully converging to a point \( \hat{q} \) that is the solution of the equations \( f(\hat{q}) = 0 \). The sequence is defined by the linearised model (4.7). The generalisation of (4.7) accommodates a class of methods to be used for finding the direction correction \( \delta_n \) which is used to correct \( q_n \). However, the methods provide only the direction \( \delta_n \) for correction without further examination of the significance of the magnitude of \( \delta_n \). When \( \delta_n \) is found too large, divergence may occur; whereas when \( \delta_n \) is too small, a very slow rate of convergence may result.

As with Broyden, Lemieux (1972) also suggested to scale the \( \delta_n \) by some factor \( t_n \) so that the iterative sequence of points, \( q_n \) for \( n = 1, 2, \ldots, L_N \), is predicted by (4.43) instead of (4.7). It must be noted that \( t_n \) is only a scalar to the direction vector \( \delta_n \). Hence, the inclusion of \( t_n \) in (4.7) becomes a sub-problem of optimisation which is used to find the optimal value of \( \delta_n \) in order to bring the new approximation closer to the solution. This
METHODS OF SOLUTION OF NETWORKS

Sub-problem is imbedded in the solution methods and operated at each iterative step. As seen in Figures 4.2 and 4.3, the step for selecting \( t_n \) is inserted just after the \( \delta_n \) is estimated (compare with Figure 4.1). This is the additional step which is used to find a \( t_n \) which optimises the value of \( \delta_n \) for subsequent updating.

The importance of the idea of using \( t_n \) to scale \( \delta_n \) becomes more apparent when the principle is illustrated graphically. Figure 4.4 shows a two dimensional problem where the iterate \( q_n = \begin{bmatrix} q_n^{(0)} & q_n^{(2)} \end{bmatrix}^T \) moves towards the solution \( \hat{q} \). The arrow denotes the direction \( \delta_n \) to which the \( q_{n+1} \) is approximated from \( q_n \). The \( q_n \) is a fixed point which corresponds to \( t_n = 0 \) while the \( q_n + \delta_n \) is the point which corresponds to \( t_n = 1 \). It follows that some points \( q(t_n) \) along the line \( q_n + t_n \delta_n \) are closer to the solution \( \hat{q} \) than the point \( q_n + \delta_n \) (a full step Newton-Raphson method). This means that the \( t_n \) can be used to bring the iterate \( q_n \) closer to the solution \( \hat{q} \); this is the sole objective of the \( t_n \). It follows that the \( q(t_n) \) is determined by the \( t_n \) and the process to find an appropriate \( t_n \) to predict \( q(t_n) \) is known as a line search.

A line search may be applied when the \( \delta_n \) leads to an increase in \( \|f_{n+1}\| \); that is,

\[
\|f(+\delta_n)\| < \|f(q_n)\| \quad (4.54)
\]

In such a case, \( q_{n+1} = q_n + \delta_n \) is unacceptable and a value of \( t_n \) to satisfy the following condition is required

\[
\|f(q_n + t_n \delta_n)\| < \|f(q_n)\| \quad (4.55)
\]

The sufficient condition for (4.55) to hold is that the \( \delta_n \) be a descent direction towards the global minimum. This requirement is essential because it means that the iterations are moving towards the solution while the \( \|f(q_n + t_n \delta_n)\| \) is decreasing. It can be shown that the methods (NR, BR, DFP and BFGS) used to find the \( \delta_n \) are providing a descent direction, and hence, the (4.55) is well defined (Ralston and Rabinowitz, 1978). This result leads to a very important aspect of keeping the iteration within bounds, and hence, preventing divergence.
Figure 4.4. Line searches in two dimensional space.
To find a value of \( t_n \) to satisfy (4.55) is a problem of single variable minimisation (Pike, 1986). The objective function involved can be represented by

\[
\begin{align*}
\text{minimise} & \quad \| f(q_n + t_n \delta_n) \| \\
\text{subject to} & \quad t_n
\end{align*}
\]  

(4.56)

The search along the line \( q_n + t_n \delta_n \) is carried out by repetitively sampling the size of \( \| f(q_n + t_n \delta_n) \| \) for different values of \( t_n \) until the condition (4.56) or some other termination conditions are satisfied.

Given the system of \( L_n \) equations and \( N \) unknown, the basic algorithmic structure for implementing the technique of line search is as follows:

i) Determine a direction of search \( \delta_n \).

ii) Minimise \( \| f(q_n + t_n \delta_n) \| \) with respect to \( t_n \) until the condition (4.56) is satisfied.

iii) Update \( q_{n+1} = q_n + t_n \delta_n \).

iv) Repeat Step i) until \( \| f_{n+1} \| < \xi \).

Note that Step ii) is only a sub-problem of achieving a solution, which is further divided into two separate stages. First comes the bracketing stage followed by the sectioning stage. The bracketing stage involves a procedure of finding an interval \([a_i, b_i]\) for such that

\[
\| f(q_n + t_n \delta_n) \| \leq \min\{\| f(q_n + a_i \delta_n) \|, \| f(q_n + b_i \delta_n) \|\}, \quad i = 1, 2, ...
\]  

(4.57)

The sectioning stage involves a procedure of generating a sequence of intervals of \([a_i, b_i]\) such that

\[
|a_j - b_j| \to 0, \quad j = i, i+1, ...
\]  

(4.58)

This is analogous to the algorithm of finding a root of an equation without providing the search interval a priori. With this analogy, one set of procedures for finding a minimum is described below.
4.6.1 Bracketing Stage

A number of algorithms for bracketing the minimum of $\|f(q_n + t_n \delta_n)\|$ with respect to $t_n$ were found in the literatures (Pike, 1986; Press et al, 1988). The availability or not of first-order derivatives is a primary consideration. If derivatives are available, then some convergence tests described below may be used to terminate the line search. If, however, derivatives are not available then the search must rely on the information provided by a triplet of points $f(a), f(b), f(c)$ for $a < b < c$ to determine how the line search is to be terminated. These two cases will be described in turn.

Two conditions which may be used to bracket the minimum of $\|f(q_n + t_n \delta_n)\|$ with respect to $t_n$ are given by Goldstein (Fletcher, 1987). In stating these conditions, the following notation is used

$$f(t_n) = \|f(q_n + t_n \delta_n)\|$$

For clarity only, the subscript $n$ is omitted from the following description. The conditions are

$$f(t) \leq f(0) + tf'(0) \quad (4.60a)$$

$$f(t) \geq f(0) + t(1 - \rho)f'(0) \quad (4.60b)$$

in which $0 \leq \rho \leq 0.5$ is a selectable parameter and may be regarded as a modifier to $f'(0)$. To satisfy the descent condition, the condition $f'(0) < 0$ must hold. These conditions are stated graphically in Figure 4.5.

In the light of Figure 4.5, a particular value of $t$ can significantly reduce the value of the function $f(t)$. Conversely, a negligible reduction in $f(t)$ can be obtained when either $t \rightarrow 0$ or $t \rightarrow \mu$. This is equivalent to drawing a horizontal line across the graph starting from $t = 0$. The $\mu$ is the least positive value of $t$ for which

$$f(\mu) = f(0) \quad (4.61)$$
Figure 4.5. Goldstein conditions (1965).
This implies that the search is confined to the interval where

\[ 0 < t < \mu \]  

(4.62)

This is necessary because the search may continue indefinitely when no suitable \( t \) is found. However, the graph of \( f(t) \) may never intersect the line of \( \rho = 0 \) so that the \( \mu \) must somehow be redefined. A sufficiently large constant value can be used but Fletcher (1987) suggested that the \( \mu \) can be estimated by

\[ \mu = \frac{\tilde{f} - f(0)}{\rho f'(0)} \]  

(4.63)

in which \( \tilde{f} \) is the lower bound of \( f(t) \). A value of \( \tilde{f} = 0 \) would be appropriate for the problems of sums of squares.

Referring to Figure 4.5, the intersection points \( a \) and \( b \) between the curve \( f(t) \) and the conditions given in (4.60) define the bracketing interval \([a_i, b_i]\) for \( i = 0, 1, \ldots \) and \( a_i < b_i \) in which the acceptable points lie. Points lying to the left of \( a \) to the right \( b \) are excluded. Initially, the \( t_i \) for \( i = 0, 1, \ldots \) is started with \( t_0 = 0 \) and then continuously moved out to the right until the conditions (4.60) are satisfied. The point at which \( t_i \) satisfies (4.60) defines the left-hand interval \( a_i \) of the bracket. The search is continued until either the (4.60a) is violated or the \( \mu \) is reached. Hence, the bracketing interval \([a_i, b_i]\) for a minimum is located. The choice of step for the subsequent searches \( t_{i+1} \) can be made arbitrary, but a direct choice would be to scale the initial step \( |t_i - t_0| \) by some factor.

Conceptually, the conditions (4.60) provide a powerful way to bracket a minimum, but it is not without its disadvantages, namely:

i) The requirement that \( 0 \leq \rho \leq 0.5 \) allows for the property that the minimising value of a quadratic function is acceptable. In general, the graph of \( f(t) \) is non-quadratic so that the minimum of \( f(t) \) is usually not bounded by the conditions (4.60); this is illustrated in the Figure 4.5.

ii) The conditions (4.60) require the first-order derivative measurement of \( f(t = 0) \). It creates the difficulty of deriving an analytical expression for \( f'(0) \), insofar which is
just what this development is trying to avoid. Although the technique of finite
differencing may be used to approximate \( f'(0) \), an additional evaluation of the \( f(h) \)
is required as a consequence.

Instead of using the Goldstein conditions to bracket the minimum, a test by means of a
triplet of points is considered more suitable for the purpose because it does not require
derivative measurements but a series of function evaluations only. The main features of a
triplet of points are similar to those described earlier in this section. Instead of using two
ordinate points to form a bracket interval \([a, b]\), the triplet of points requires three
ordinate points to form a bracket \([a, b, c]\).

Within the bracketing interval \([a_i, b_i, c_i]\) for \( i = 0, 1, \ldots \) and \( a_i < b_i < c_i \) there exists a
minimum if the following inequality is satisfied

\[
f(b_i) \leq \min(f(a_i), f(c_i))
\]  

(4.64)

To conduct the above test, the initial ordinate points of the bracket interval \([a_0, b_0, c_0]\) have
to be provided a priori. The sensible choice for the initial ordinates \( a_0 \) and \( b_0 \) would be
zero and unity respectively since these values arise naturally in the NR method. If the
condition \( f(a_0) = 0 < f(b_0) = 0 \) is satisfied, then there is a likelihood that the interval
\([a_0, b_0]\) contains a minimum. Otherwise, the minimum must lie outside the interval
\([a_0, b_0]\). To complete the test, the ordinate of \( c_0 \) is required which can be found by

\[
c_i = (1 + \mathcal{R})(b_i - a_i), \quad i = 0, 1, \ldots
\]  

(4.65)

in which \( \mathcal{R} = \frac{1}{2}(\sqrt{5} - 1) \) is known as golden ratio (Press et al, 1988). If the test does not
satisfy (4.64), then exchange of variables is needed, that is,

\[
a_{i+1} \leftarrow b_i, \quad b_{i+1} \leftarrow c_i, \quad i = 0, 1, \ldots
\]  

(4.66)

and the ordinate \( c_{i+1} \) is calculated according to (4.65). The test is continued until the
conditions (4.64) are satisfied. It is advisable to enforce the condition \( c_{i+1} \leq \mu \) so as to
prevent digression. The basic algorithmic structure for a triplet of points is shown in
Figure 4.6.
Figure 4.6. Flow chart for a triplet of points.
4.6.2 Sectioning Stage

Once the bracket interval \([a, b, c]\) is initially located, the objective is to reduce this interval down to a single point, namely the location of the minimum value of \(f(t)\) which gives the optimum value of \(t\) to be used in a particular step. Two principles commonly used in practice are parabolic interpolation and successive sectioning, which are described below.

If the function is parabolic near to the minimum, then a parabola fitted through any three points ought to pass very close to the minimum. This idea can be directly depicted from Figure 4.7. A parabola (dashed line) is fitted through the initial three points \(\odot, \ominus\) and \(\bigcirc\) on the given function (solid line). The minimum of the function is then evaluated to give point \(\odot\). Since point \(\odot\) is between points \(\odot\) and \(\ominus\), point \(\odot\) is used in place of point \(\odot\), and hence, three new points \(\odot, \ominus\) and \(\bigcirc\) are obtained. A second parabola is then fitted through points \(\odot, \ominus\) and \(\bigcirc\) to give point \(\ominus\). Subsequently, another three new points \(\odot, \ominus\) and \(\bigcirc\) are obtained. The process is repeated until the minimum is reached or some convergence conditions are satisfied.

Many interpolation formulae to locate the minimum of a parabola exist. One such formula suggested by Brent (Press et al (1988) is given below:

\[
\tau_j = b_j + \frac{1}{2} \left[ \frac{f(b_j) - f(c_j)}{f(b_j) - f(a_j)} \right] \left[ b_j - a_j \right] - \frac{1}{2} \left[ \frac{f(b_j) - f(c_j)}{f(b_j) - f(a_j)} \right] \left[ b_j - c_j \right] \left[ f(b_j) - f(a_j) \right], \quad j = i, i+1, \ldots (4.67)
\]

However, the Brent's formula may fail if the three points are colinear, in which case, the denominator is zero, which means that the minimum of the parabola is infinitely far away. Another useful formula suggested by Broyden (1965) which does not require three points for interpolation is stated below:

\[
\tau_2 = \frac{\sqrt{1+6\theta} - 1}{3\theta} \quad (4.68)
\]

where

\[
\theta = f(1)/f(0)
\]
Figure 4.7. Convergence to a minimum by parabolic interpolation (Press et al, 1988).
Unfortunately, parabolic interpolation may not be applicable to all situations. Firstly, the method has a tendency of giving wrong interpolations especially for functions that have a very sharp valley. Secondly, if the given function has discontinuous first or second-order derivatives, then the parabolic interpolation is of no particular advantage. This is because information about derivatives is used to derive the interpolation formula.

In view of these limitations, a simple sectioning algorithm called the golden section search (Press et al, 1988) is explored. Although it performs less efficiently than the parabolic interpolation, it can perform very high accuracy searches, and it does not require derivative measurements.

By analogy with the method of successive bisection, the first step is to choose a point between \( a \) and \( b \) for such that the following condition holds

\[
\frac{(t_2 - t_3)}{(t_3 - t_1)} = \xi
\]  

(4.69)

in which \( t_1 \) and \( t_2 \) correspond to \( a \) and \( b \) (see Figure 4.8) and \( t_3 \) is the newly inserted point. The requirement is to maintain the same ratio for the subsequently determined bracketing intervals. When an additional point \( t_4 \) is inserted to contract the bracket, it is also required that \( (t_4 - t_1) = (t_2 - t_3) \). By symmetry, it follows that the brackets \([t_1, t_4, t_3]\) and \([t_4, t_3, t_2]\) have intervals in the same ratio

\[
\frac{(t_3 - t_4)}{(t_4 - t_1)} = \frac{(t_2 - t_3)}{(t_2 - t_1)} = \xi
\]  

(4.70)

Once four points are available, the test based on the value of \( f(t) \) can begin. Figure 4.8 shows that the \( f(t_4) \) has the lowest value so that the \( t_2 \) is excluded from the subsequent tests and hence the bracketing interval \([t_1, t_4, t_3]\) is formed. If, however, the \( f(t_3) \) has the lowest value, then \( t_1 \) is excluded and the bracketing interval \([t_4, t_3, t_2]\) is formed. Another new trial point \( t_5 \) is inserted into the bracketing interval \([t_1, t_4, t_3]\) based on the calculations in (4.70). The process is then repeated until convergence tests are satisfied. The basic algorithmic structure of golden section search is shown in Figure 4.9.
Figure 4.8. Convergence to a minimum by golden section search.
Figure 4.9. Flow chart for Golden Section Search.
Thus, the line search methods described above in Sections 4.6.1 and 4.6.2 enable a suitable value of the parameter $t_n$ to be obtained for use in (4.43). It must be stressed that line search is a sub-problem and operated at each iterative step. This is used to find the optimal value of $\delta_n$ in order to accelerate convergence. Compare Figure 4.1 with Figures 4.2 and 4.3, the only difference is the additional step in solution algorithms for selecting an appropriate $t_n$ for subsequent updating.

4.7 Concluding Remarks

The modelling of a hydraulic network containing open channels results in a set of non-linear equations. A method commonly applied to solve a set of simultaneously non-linear equations is the Newton-Raphson method. The concept of the method is simple and it has a desirable rate of convergence. However, the Newton-Raphson method is not easy to apply to open channel networks because it requires partial derivatives of the governing equations to be worked out. In open channel networks, the string functions involved can be extremely complex, which gives rise to serious problems when deriving the partial derivatives analytically. The derivatives can be obtained by numerical means, but if the step size is not carefully chosen, it is suspected that it will affect the convergence of the method.

The Newton-Raphson method requires the partial derivatives to be assembled in the Jacobian matrix, which is inverted to obtain a solution. Inversion of the Jacobian matrix $J^{-1}$ can be singular even though a solution exists. The method can fail to converge if a poor set of initial approximations is chosen. Although a line search may rectify this situation, it is offset by considerable additional computational effort. However, the use of line search is advantageous because it will enforce convergence and it enables the method in which it is imbedded to benefit from accelerated convergence.

In order to overcome the drawbacks in the Newton-Raphson method, Broyden presented a class of methods for solving a set of simultaneously non-linear equations. The methods do not require partial derivatives, since the derivatives are approximated by the method. The inversion of the approximated Jacobian matrix is also incorporated in the formulation of the methods. Despite using more elaborate updating formula to approximate the inversion of the Jacobian matrix, the method is in fact simpler to apply to open channel networks than the Newton-Raphson method. Other methods which are similar in
derivation to Broyden method include the DFP and BFGS procedures. The updating formulii used by the methods examined in the research are listed Table 4.1.

As with the Newton-Raphson method, line searches are also important to the methods of Broyden, DFP and BFGS, because it accelerates the convergence of the solution methods in which it is imbedded. Moreover, the incorporation of line searches also ensures global convergence, which is not guaranteed by the Newton-Raphson method. In general, the requirement of the initial approximation is more relaxed, although the methods still require reasonable initial approximations.

It is important to carry out an investigation regarding solvability and applicability of the various solution procedures to open channel networks. Whether these methods are more versatile than the Newton-Raphson method is resolved through numerical experiments, the details of which are resorted in Chapter 5.
### Table 4.1. List of methods for solving a set of simultaneous non-linear equations.

<table>
<thead>
<tr>
<th>Methods of Solution</th>
<th>Updating Formula$^\circ$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Newton-Raphson</td>
<td>$H_{n+1} = -J_{n+1}^{-1}$</td>
</tr>
<tr>
<td>Broyden (1965)</td>
<td>$H_{n+1} = H_n - \frac{(t_n \delta_n + H_n y_n) \delta_n^T H_n}{\delta_n^T H_n y_n}$</td>
</tr>
<tr>
<td>Davidon-Fletcher-Powell (1963)</td>
<td>$H_{n+1} = H_n - \frac{H_n y_n \gamma_n^T H_n}{\gamma_n^T H_n y_n} - \frac{t_n \delta_n}{\delta_n y_n}$</td>
</tr>
<tr>
<td>Broyden-Fletcher-Goldfarb-Shanno (1970)</td>
<td>$H_{n+1} = H_n - \frac{\delta_n y_n \gamma_n^T H_n + H_n y_n \delta_n}{\delta_n y_n} - \left( I_n - \frac{\gamma_n^T H_n y_n}{\delta_n y_n} \right) \frac{\delta_n}{\delta_n y_n}$</td>
</tr>
</tbody>
</table>

$^\circ \gamma_n = f_{n+1} - f_n$;

$\delta_n = (x_{n+1} - x_n)/t_n$
CHAPTER 5

APPLICATIONS AND TESTING OF METHODS OF SOLUTION OF NETWORKS
5

APPLICATIONS AND TESTING OF METHODS OF SOLUTION OF NETWORKS

5.1 Introduction

The development of a network solution based on string and loop formulation makes it possible to analyse many practical situations. The basic requirement for the analysis is that each module associated with a particular string must have a relationship between discharge and head-loss. The relationship for each module is then combined in turn to characterise that string. An understanding of these relationships and how to apply them is essential. The procedures adopted depend upon the identification of hydraulic controls. A hydraulic control is a point at which the relationship between the depth of flow and discharge is uniquely defined. From this definition, it follows that if any feature that acts as a control, then the depth of flow can be calculated once the discharge is given, or vice versa. Although a control serves as a starting point for calculating water depth at other points, the nature of the control and what distance it is influencing is an important issue in network analysis.

There are two aspects of particular concern.

i) Whether the module acts as a control. If so, it prevents the transmission of the effects of change in flow conditions either in an upstream or in a downstream direction depending upon the precise nature of the control. If not, the flow conditions for the module are influenced by controls further upstream or downstream.

ii) Whether the tailwater situation will override the control situation. If so, the effects of change in water surface elevation downstream are transmitted upstream by a backwater profile. Thus, the control is overridden and no longer served as a control.
Certain modules, for example weirs, require different equations to describe the discharge and head-loss relationship depending upon the actual conditions under which the weir is operating. In mathematical term, the relationship between discharge and head-loss is discontinuous. Apparently, this has invalidated the assumption that objective functions are smooth which was made earlier in achieving a network solution (see Section 4.2). Other examples of this include a change from subcritical to supercritical flow in channel, or vice versa, and control changes at any point hydraulic structure.

A hydraulic control exists along a particular string depending upon the type of installation and the influence of tailwater conditions. The problem in network analysis is not the determination of the existence of hydraulic controls, but the phenomenon of 'switching of hydraulic control' during the iterative process. The control may switch (change) as a result of the string flow being altered. This may happen when the loop flow corrections are obtained and changes made to the flow in a particular string during the iterative process. For example, a flow correction made to a particular string could cause the state of flow to change from subcritical to supercritical, or the flow over a hydraulic structure to become submerged. This change in flow regime causes a switch in equations that govern the flow conditions. If this switching happens so frequently, then it can cause some difficulties in achieving a network solution.

Thus, two main issues are explored. Firstly, the work investigates the applicability and solvability of the loop formulation method to open channel networks. This involves the investigation of the performance of the convergent NR, BR, DFP and BFGS solution algorithms as described in Chapter 4. Numerical examples illustrate the performance of the methods, and thence propose the most suitable method for solving the open channel networks. Secondly, the work investigates the effects on convergence caused by the phenomenon of switching of hydraulic control. It is envisaged that this will cause some difficulties in convergence, and in order to accelerate to the solution, the theory of the line search as described in Chapter 4 is investigated. The advantage of using a line search is that it is a sub-problem and can be imbedded in any of the iterative method mentioned above.
5.2 Determination of Hydraulic Controls

A network comprises a number of strings, and a string comprises a series of modules with different physical characteristics. The very first step in network flow analysis is to decompose the network into strings. Once the strings are found, it is necessary to establish those points along the strings that can serve as hydraulic controls. The purpose behind the determination of controls is to provide some starting water depths for water surface profile computations. This is analogous to the boundary value problem of differential equation that requires an initial value in order to proceed with the computations. Hydraulic controls are generally classified into two types, namely artificial controls and critical depth controls as described below.

5.2.1 Artificial Controls

Artificial controls such as weirs, flumes and sluice gates are generally designed for flow regulation or measurement and act as controls in the channel in which they are situated. They are not affected by the flow condition downstream under normal operation. This means that the flow is governed only by the physical characteristics of the structure and the flow condition of the approaching channel. Table 5.1 (Acker et al, 1978) gives examples of discharge equations for thin-plate full-width weirs.

In certain circumstances, for example rising outfall water levels, the tailwater depth will increase and may rise beyond the modular limit so that the performance of the structure will be affected. In this situation, the discharge is said to be submerged, since the relationship between head and discharge is governed both by the physical characteristics and the tailwater level. Under this circumstance, the upstream head needs to increase (see Figures 5.1a and 5.1b) to maintain the flow; the submerged discharge may be estimated by Villemonte's formula (Ackers et al, 1978)

\[ Q_s = Q_f \left[ 1 - \left( \frac{h_2}{h_1} \right) \right]^{0.385} \]  (5.1)
<table>
<thead>
<tr>
<th>Reference</th>
<th>Limitations</th>
<th>Discharge Equations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Francis</td>
<td></td>
<td>$Q = 0.587B\sqrt{gh_1^3}$</td>
</tr>
<tr>
<td>Swiss S I.A</td>
<td>$0.025 &lt; h_t &lt; 0.8$</td>
<td>$Q = \frac{2\sqrt{2}}{3} \left( 0.615 + \frac{0.000615}{h_t + 0.0016} \left( \frac{h_t}{h_t + H_w} \right)^2 \right) B\sqrt{gh_1^3}$</td>
</tr>
<tr>
<td>Rehbock</td>
<td>$0.03 &lt; h_t &lt; 0.75$</td>
<td>$Q = \frac{2\sqrt{2}}{3} \left( 0.602 + 0.0832 \frac{h_t}{H_w} \right) B\sqrt{g(h_t + 0.00125)^3}$</td>
</tr>
<tr>
<td>Kindsvater and Carter</td>
<td>$h_t &gt; 0.03$</td>
<td>$Q = \frac{2\sqrt{2}}{3} \left( 0.602 + 0.0075 \frac{h_t}{H_w} \right) (B - 0.001)\sqrt{g(h_t + 0.001)^3}$</td>
</tr>
<tr>
<td>I.M.F.T</td>
<td>$h_t &gt; 0.03$</td>
<td>$Q = \frac{2\sqrt{2}}{3} \left( 0.627 + 0.0180 \frac{H_t}{H_w} \right) B\sqrt{gh_1^3}$</td>
</tr>
</tbody>
</table>

Table 5.1. Equations for thin-plate full-width weir.
Figure 5.1a. Free discharge of a thin-plate weir.

Figure 5.1b. Submerged discharge of a thin-plate weir.
where

\[ Q_s = \text{discharge when submerged} \]
\[ Q_f = \text{discharge under free-flow condition at the same upstream head } h_t \]
\[ h_z = \text{tailwater level relative to crest elevation} \]
\[ e = \text{exponent of the flow equation} \]

At the modular limit, which is the condition where the tailwater level is just high enough to begin affecting the performance of the structure, there co-exists two discharges (free and submerged). The composite discharge equation thus has a discontinuity at this transition point. This, in turn, constitutes discontinuity in the system of equations, so that the performance of the solution algorithms may be affected. Detailed descriptions of the most common hydraulic structures and their mode of operation are given in Chow (1959) and Acker et al (1978).

### 5.2.2 Critical Depth Controls

A critical depth control will exist at a point where the state of flow changes from subcritical to supercritical or vice versa. This occurs when a channel bed slope is abruptly steepened or flattened. For example, if the downstream slope of a channel is abruptly steepened beyond the critical slope \( S_c \) then a flow with supercritical velocity will develop. To sustain the flow, a depth reduction of flow must occur in the downstream direction to conserve specific energy. The water surface profile passes through the critical depth at a point known as the critical control section, where the Froude number \( Fr \) is

\[ Fr = \sqrt{\frac{Q^2 T}{gA^3}} = 1 \tag{5.2} \]

Flow profile classifications for abrupt changes of slope are shown in Table 5.2. However, the type of profile developed in a steep channel may also be governed by the tailwater situation.

In Case-1, if the tailwater level just downstream of the critical section is less than or equal to critical depth, the flow condition upstream is unaffected as shown in Figure 5.2a. If the
tailwater level just downstream of the critical section rises beyond the critical depth level, then the supercritical flow will be drowned and the flow will be directly governed by the control downstream as shown in Figure 5.2b. This implies that the effect of the control at the point under consideration has been overridden by the action of control farther downstream.

In Case-2, a hydraulic jump will be formed, if the sequent depth and the tailwater depth are exactly equal as shown in Figure 5.3a. The actual position of the jump depends on the local Froude number. The larger the value of Froude number, the farther downstream the jump begins. If the tailwater depth is less than the sequent depth, then the hydraulic jump will be washed downstream as shown in Figure 5.3b. Conversely, if tailwater depth rises above the sequent depth, then the supercritical flow will be drowned as shown in Figure 5.3c. As the tailwater level rises further, the jump will move farther upstream until it reaches the upstream critical section. Beyond this point, the incoming flow will be directly affected by tailwater. In this situation, the control upstream is completely overridden and controlled by the tailwater conditions as shown in Figure 5.3d.

It is possible to have several points along a string which are potential hydraulic controls. Under this circumstance, the entire string must first be examined and potential controls identified. The tailwater level then determines what distance is under the influence of a particular control, and it is then possible to decide which control will dominate and override other potential controls.

The matter is somewhat more complicated if the bed slope and/or dimensions of a channel vary with distance. In this case, the critical control section may not be situated at either upstream or downstream ends of the channel, but at some distance along its length. At a critical section, a singularity of the gradually varied flow equation will occur, that is,

\[
\frac{dy}{dx} = \frac{S_o - S_f}{1 - Fr^2} = 0
\]

To determine the depth and position of the control section requires the real and positive solution of the simultaneous equations

\[
\begin{align*}
S_o - S_f &= 0 \\
1 - Fr^2 &= 0
\end{align*}
\]
Table 5.2. Flow classification for two connected channels.
Figure 5.2a. A critical depth control is formed at upstream.

Figure 5.2b. The critical depth control is overridden due to high tailwater level.
Figure 5.3a. A hydraulic jump is formed at downstream due upstream supercritical flow.

Figure 5.3b. The hydraulic jump is washed downstream due to low tailwater level.
Hydraulic Jump is Drowned since Tailwater Depth > Sequent Depth

Figure 5.3c. The hydraulic jump is drowned due to high tailwater level.

Tailwater Depth Rises Above Upstream Critical Section

Figure 5.3d. The supercritical flow is completely submerged due to high tailwater level.
A further requirement is that $S_x - S_f$ must change from negative to positive with increasing distance. A direct solution of the above equations is not possible, and an iterative procedure must be incorporated to obtain the solution.

It must be stressed that a normal depth may also be used as a control depth since it has a unique depth to discharge relationship. However, due to numerical instability of the integration, the normal depth will normally not serve as a control but rather as the terminal value to which the calculated depths will tend asymptotically in a long reach. The potential danger of profile computation starting from an inappropriate direction has been illustrated in Chapter 2.

Further complication arises when the flow along a channel has a height restriction, for example the flow through a culvert. The characteristics of culvert flow are complicated because the flow is governed by both physical characteristics, inlet and outlet geometry, headwater and tailwater conditions. Culverts are considered in more detail in the following section, because their flow characteristics are particular appropriate for exploring the performance of different algorithm for network solution. In particular, culvert flow will be used to investigate the problem of control switching during the numerical calculation procedures for network analysis.

A culvert shown in Figure 5.4 will flow full either when the outlet is submerged or when the outlet is not submerged but the headwater reaches beyond a critical value $H_r$. Otherwise, the culvert acts as an open channel. The value of $H_r$ varies from $1.2H_b$ to $1.5H_b$ (Chow, 1959; French, 1986) in which $H_b$ denotes the height of the culvert. In the vicinity of the culvert entrance, there occurs a rapid contraction of the flow and a subsequently rapid expansion of the flow within culvert barrel. If the culvert is not sufficiently long to allow the expanding depth to rise and fill the barrel, then the culvert will never flow full even though the headwater is greater than $H_r$. Such a culvert is hydraulically short. Otherwise, the culvert is hydraulically long. It must be stressed that whether a culvert is hydraulically short or hydraulically long cannot be determined by the length of the barrel alone. It depends on other physical characteristics as well as headwater and tailwater conditions. Consequently, investigations have been carried out by Carter (Chow, 1959) and Bodhaine (French, 1986) to classify various types of flow.
<table>
<thead>
<tr>
<th>Culvert Flow</th>
<th>Flow Regimes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type - 1</td>
<td><img src="Type-1-Diagram" alt="Diagram" /></td>
</tr>
<tr>
<td>Type - 2</td>
<td><img src="Type-2-Diagram" alt="Diagram" /></td>
</tr>
<tr>
<td>Type - 3</td>
<td><img src="Type-3-Diagram" alt="Diagram" /></td>
</tr>
<tr>
<td>Type - 4</td>
<td><img src="Type-4-Diagram" alt="Diagram" /></td>
</tr>
<tr>
<td>Type - 5</td>
<td><img src="Type-5-Diagram" alt="Diagram" /></td>
</tr>
<tr>
<td>Type - 6</td>
<td><img src="Type-6-Diagram" alt="Diagram" /></td>
</tr>
</tbody>
</table>

Figure 5.4. Classification of culvert flow.
Table 5.3a. Culvert flow characteristics.

<table>
<thead>
<tr>
<th>Culvert Flow</th>
<th>Barrel</th>
<th>Flow</th>
<th>Slope, ( S_o )</th>
<th>( y_1/H_b )</th>
<th>( y_4/y_c )</th>
<th>( y_4/H_b )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type -1</td>
<td>Partly Full</td>
<td>Subcritical</td>
<td>Mild</td>
<td>&lt;1.5</td>
<td>&gt;1.0</td>
<td>≤1.0</td>
</tr>
<tr>
<td>Type -2</td>
<td>Partly Full</td>
<td>Subcritical</td>
<td>Mild</td>
<td>&lt;1.5</td>
<td>&lt;1.0</td>
<td>≤1.0</td>
</tr>
<tr>
<td>Type -3</td>
<td>Partly Full</td>
<td>Supercritical</td>
<td>Steep</td>
<td>&lt;1.5</td>
<td>&lt;1.0</td>
<td>≤1.0</td>
</tr>
<tr>
<td>Type -4</td>
<td>Full</td>
<td>Submerged</td>
<td>Any</td>
<td>&gt;1.0</td>
<td>-</td>
<td>&gt;1.0</td>
</tr>
<tr>
<td>Type -5</td>
<td>Full</td>
<td>Hydraul' Long</td>
<td>Any</td>
<td>≥1.5</td>
<td>-</td>
<td>≤1.0</td>
</tr>
<tr>
<td>Type -6</td>
<td>Partly Full</td>
<td>Hydraul' Short</td>
<td>Any</td>
<td>≥1.5</td>
<td>-</td>
<td>≤1.0</td>
</tr>
</tbody>
</table>

Table 5.3b. Culvert flow equations.

<table>
<thead>
<tr>
<th>Culvert Flow</th>
<th>Control Type</th>
<th>Control Section</th>
<th>Discharge Equation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type -1</td>
<td>Backwater</td>
<td>Outlet</td>
<td>( Q = C_d A_c \sqrt{2g \left(E_1 + z_2 - z_3 - y_3 - h_{f2} - h_{f3} \right)^2} )</td>
</tr>
<tr>
<td>Type -2</td>
<td>Critical Depth</td>
<td>Outlet</td>
<td>( Q = C_d A_c \sqrt{2g \left(E_1 + z_2 - z_3 - y_3 - h_{f2} - h_{f3} \right)^2} )</td>
</tr>
<tr>
<td>Type -3</td>
<td>Critical Depth</td>
<td>Inlet</td>
<td>( Q = C_d A_c \sqrt{2g \left(E_1 - y_3 - h_{f2} \right)^2} )</td>
</tr>
<tr>
<td>Type -4</td>
<td>Backwater</td>
<td>Inlet/Outlet</td>
<td>( Q = C_d A_b \sqrt{2g \left(z_4 - z_1 + y_1 - y_3 - h_{f2} \right)^2} )</td>
</tr>
<tr>
<td>Type -5</td>
<td>Entrance &amp; Barrel Geometry</td>
<td>Inlet</td>
<td>( Q = C_d A_b \sqrt{2g \left(z_4 - z_3 + y_1 - y_3 - h_{f2} \right)^2} )</td>
</tr>
<tr>
<td>Type -6</td>
<td>Entrance Geometry</td>
<td>Inlet</td>
<td>( Q = C_d A_b \sqrt{2g \left(y_1 \right)^2} )</td>
</tr>
</tbody>
</table>
Figure 5.5. Algorithm for identifying culvert flow.
When the slope of the culvert is steep, a critical control section at the culvert entrance will develop and the flow within the barrel is supercritical. A hydraulic jump may occur within the barrel if the tailwater conditions are suitable. If the sequent depth rises and reaches the barrel soffit, then the culvert may prime itself automatically and thus flow full.

The loss in energy in a culvert is determined by the application of the continuity and energy equations between the culvert entrance and outlet sections; it is related to the rapid contraction and expansion of the flow and the boundary friction on the culvert walls. For practical purposes and computational convenience, culvert flow may be divided into six categories and which are summarised in Tables 5.3a and 5.3b. The computation of flow profile along a string that involves a culvert is summarised in Figure 5.5 and outlined below:

i) If $H_r < 1.5$, $S_o < S_e$, and the tailwater level is greater than the critical depth, the culvert will flow like a channel, then the flow will be of Type-1.

ii) If $H_r < 1.5$, $S_o < S_e$, and the tailwater level is less than the critical depth, the culvert will flow like a channel, then the flow will be of Type-2.

iii) If $H_r < 1.5$, and $S_o \geq S_e$, the culvert will flow like a channel, then the flow will be of Type-3.

iv) If $H_r \geq 1.5$, and the outlet is submerged, the culvert will flow full like a pipe, then the flow will be of Type-4.

v) If $H_r \geq 1.5$, and the outlet is not submerged, then the flow will be either Type-5 (hydraulically long) or Type-6 (hydraulically short). To distinguish between these two types of flow, Carter's charts are needed, which is described in details in Chow (1959).

In the above flow classification, controls appear at culvert entrance and outlet. Whenever there is a transition of culvert flow, because of changing headwater or tailwater conditions, controls will migrate either in an upstream or downstream direction, and thus an appropriate equation must be applied to analyse the corresponding flow.
5.3 Testing Network Solution Algorithms

The analysis of an open channel network involves defining a set of simultaneous non-linear equations and finding their solution. The root of the equations represents the steady state flow distribution throughout the network. Using the approaches described in Chapter 3 and Chapter 4, the algorithm used to analyse a hydraulic network is as follows:

i) Trace all the independent loops throughout the system. This step is equivalent to setting up the loop equations with the loop flow correction vector, \( q \), as the independent variable; that is, formulating the objective vector function \( f(q) \).

ii) Obtain an initial set of flow distribution. It may be arbitrarily chosen but must satisfy continuity of flows at nodal junctions. In free surface flow problems where the flow direction is generally known a priori, reasonable initial guesses of the unknown flow distribution should be possible.

iii) Use either the NR, BR, DFP or BFGS algorithm to solve for \( f(q) = 0 \), until \( \| f_n \| \leq \xi \).

The generalisation of the algorithm above allows various methods of solution to be tested. Subsequently, the methods are programmed under identical conditions to solve two network problems. One of the main difficulties in testing these algorithms is that there is an infinite number of combinations of modules and strings to form networks. Therefore, the numerical experiments have been carried out on two classes of networks which are encountered in practice in water and wastewater treatment plants. The first is a parallel network, where strings are linked in parallel from common start and end nodes, as illustrated in Figure 5.6. Whilst four parallel strings are considered in the following experiments, it is hoped that the outcome will provide sufficient information to enable the performance of a large number of parallel to be gauged.

The second type of network is more complex, and contains 'nested' loops, as illustrated in Figure 5.7. The strings do not have common start and end nodes. A particular problem in analysis is that the flow distribution may be very uneven. The network illustrated contains three level of 'nesting' which is as complex as is likely to be encountered in practice. Although only open channels are included in the investigations, they will provide some insight into performance, applicability, capability as well as flexibility of various solution algorithms.
The primary factor of concern is the rate of convergence. This is not only related to the number of iterations required to converge to the prescribed tolerance but also the computational time consumed at each iteration. The secondary factor is the initial set of approximations. The methods may perform poorly if the initial set of approximations is far away from the final solution, since the rate of convergence is a function of the initial assumption of the flow distribution.

The basis for comparison is to assess the accuracy of the solution and the total computational time required to reach this solution. However, it is not satisfactory to directly record the computational time because the time taken to reach a solution can be affected by numerous external factors. To ensure that the computer program works, numerous time-consuming housekeeping routines have to be imbedded in the program. Although the time recorded gives some hints on the efficiency of a particular method, it does not provide an accurate measure on the actual time taken by a particular method. Thus, it seems more appropriate to use number of evaluations of the vector function $f_n$ in place of computational time, because the actual time taken is directly proportional to the number of evaluations of the vector function $f_n$ that are necessary to obtain a solution.

The time taken to perform the inversion of the Jacobian matrix $J_n$ is likely to be minimal in comparison with the time taken to evaluate the highly complex vector function $f_n$ and its partial derivatives.

Although the comparisons start from an identical set of initial conditions, it is impossible to terminate the computations at the same point because the course of computation varies from method to method. The iterative process terminates once the size of $\|f_n\|$ is found less than the prescribed tolerance $\xi$. In order to include the discrepancy due to different points of termination, Broyden (1965) suggested the following formula

$$\eta_r = \frac{1}{N_r} \ln \left( \frac{\|f_0\|}{\|f_n\|} \right)$$

(5.5)

to measure the performance of a particular method. The $\eta_r$ is known as the mean convergence rate, and $N_r$ is the total number of function evaluations. The larger the value of $\eta_r$, the higher the efficiency of a particular method. The $\|f_n\|$ denotes the residual of the system of equations in the beginning of the solution process, while $\|f_n\|$ is the
residual of the system of equations after $n$ iterations. Unfortunately, the formula does not relate the performance of a particular method to the accuracy of the solution finally obtained. Whilst it does provide some indications of the accuracy of the final solution obtained, it does not provide sufficient information to describe the actual error involved in the final solution. Therefore, a modification to (4.5) is proposed. This is achieved by directly substituting the $f_n$ in (4.5) with $x_n - \hat{x}$, and hence,

$$\eta_s = \frac{1}{N_f} \ln \left( \frac{\|x_0 - \hat{x}\|}{\|x_n - \hat{x}\|} \right)$$  \hspace{1cm} (5.6)$$

in which $x_n$ and $\hat{x}$ denote the approximate solution and the exact final solution respectively. The importance of the above equation is that it enables the performance (in terms of number of iterations) to be directly related to the accuracy of the approximate solution $x_n$ that can be achieved by a particular method.

The following assumptions are included in the examples used.

i) Minor head losses such as the eddy losses at nodal junctions are neglected.

ii) No lateral inflow or outflow occurs at any point in the network system.

iii) Zero flow does not occur at any point in the network system.

iv) Flow conditions in strings are strictly subcritical throughout.

v) The vector function $f$ is so complicated that analytical evaluation of the Jacobian matrix $J$ is impracticable. Therefore, the $J$ is evaluated by means of finite differencing as described in Section 4.3. A value of $h = 10^{-5}$ is chosen for the differentiations.

vi) The line search parameter $t_n$ is taken to be unity because $t_n = 1$ arises naturally in the classical NR. The significance of $t_n$ will be considered later in Section 5.4.

5.3.1 A Parallel Network

The first network comprises four open channels is shown in Figure 5.6; the physical characteristics are shown in Table 5.4. The initial set of flow distributions for the computer run was obtained by splitting the system outflow equally at nodal junctions, as shown in Table 5.5a. In this case, the bed slopes selected were small, in order to simulate
APPLICATIONS AND TESTING OF METHODS OF SOLUTION OF NETWORKS

subcritical flow. Four computer runs for the network were conducted separately under identical conditions using the NR, BR, DFP and BFGS methods. Double precision (15 significant figures) was used throughout the computations, and the results for the four runs were obtained. The detailed calculations for the corresponding runs are also included in Appendix 'A' to illustrate the differences and similarities of the four methods. It was found that the solutions obtained by the NR, BR, DFP and BFGS methods are identical. The results, along with the water depths and energy heads are shown in Table 5.5b. It is noted that there exists some discrepancies, approximately one millimetre difference in water depths at the ends of the channels. This discrepancy in water depths were computed due to the difference in velocity heads at nodal junctions as explained below.

Consider the given specific energy at the point of outfall, $E_{\text{outfall}}$. Because the minor losses are neglected in the network analysis above, the specific energy at the downstream end of each channel must equal $E_{\text{outfall}}$ by the law of energy conservation; that is,

$$E_{\text{outfall}} = E_{s,2}, \quad s = 1, 2, 3, 4$$

Expanding the equation above in terms of discharge and water depth, it becomes

$$E_{\text{outfall}} = y_{s,2} + \frac{Q^2}{2gA_{s,2}^2}, \quad s = 1, 2, 3, 4$$

Two variables in the equation above are $y_{s,2}$ and $Q_{s,2}$. If the velocity head is considered negligible in compared with $y_{s,2}$, then it may be neglected from the equation above, and hence the equation may be written as

$$E_{\text{outfall}} = y_{s,2}, \quad s = 1, 2, 3, 4$$

Thus, no discrepancy of $y_{s,2}$ is found between channels. If, however, the velocity head is significant and must incorporate in the analysis, then the velocity heads obtained will be determined by the discharge and channel characteristics. An increase of velocity head will lead to a decrease of $y_{s,2}$, in order to attain the balanced situation. As a result of this, different values of $y_{s,2}$ are obtained, because of difference values of velocity heads in channels.
Because the head-loss along each channel must equal, it follows that the upstream specific energy for each channel must be identical and equal to the energy at the point of inflow. For the same reason as the discrepancy in $y_{11}$ is caused by the difference in velocity heads.

The convergence characteristics of the NR, BR, DFP and BFGS methods can be judged by studying Tables 5.6a, 5.6b, 5.6c and 5.6d respectively. The reason for going to $\|f_n\| < 10^{-15}$ is that the detailed performance can be compared. Whilst this accuracy is not of practical relevance, it is important to understand the likely problems with the methods; when the methods are applied, there is an infinite combination of modules and layouts to form a network. Therefore, it is impossible to test all of these and must choose the solution method which it is believed to be most robust on limited tests. Hence, the tolerance $\|f_n\| < 10^{-15}$ is chosen in order to obtain in-depth analysis of performance. An entry '0' under the heading $\|f_n\|$ denotes that the iterative procedure has terminated with an exact solution, whereas an entry other than '0' denotes the process iterated indefinitely.

It is seen that the NR method converged rapidly in the beginning and then oscillate indefinitely towards the solution. In fact, the NR method converged to the order of $10^{-10}$ on $\|f_n\|$ in just 4 iterations. By studying the size of $\|f_n\|$ over these 4 iterations, the property of quadratic termination is observed. The oscillations in the results was caused by instability in $\tilde{J}_n$ (Jacobian matrix approximated by finite differencing). When the $\tilde{J}_n$ was evaluated near the solution, the entries in the matrix elements became very small owing to the $\sim 0$ as the iteration moved closer to the solution. It was found that the matrix

$$
\tilde{J}_n = \begin{bmatrix}
-0.02686 & 0.00972 & 0.00000 \\
0.00972 & -0.01632 & 0.00661 \\
0.00000 & 0.00661 & 0.01155
\end{bmatrix}
$$

approximated using the technique of finite difference varied very little after the 5th iteration (or $\|f_n\|$ reaches the order of $10^{-12}$). It implies that the method always provides the same correction to which the string flows were updated. This suggests that the use of the technique of finite difference to approximate the partial derivative introduces sizeable numerical errors such as round-off error into the approximation. The closer to the
solution, the greater the effects of the round-off error. Consequently, the inversion of the \( \mathbf{J}_n \) led to an incorrect \( \mathbf{\delta}_n \) for updating. This explains why the NR method wandered around the solution indefinitely.

On the other hand, the quasi-Newton methods converged slowly, but, eventually, they converged to an exact solution. A great diversity in performance between methods were clearly shown in Tables 5.6b to 5.6d. It is evident that the BFGS method converged to the solution as much as 3 and 6 times faster than the BR and DFP methods respectively. Furthermore, the property of super-linear convergence was generally observed for the BFGS method.

The slow convergence in the beginning was found to be the matrix \( \mathbf{H}_n \) approximated insufficient accuracy to the inversion of the \( \mathbf{J}_n \). The matrix \( \mathbf{H}_0 \) shown below is obtained using the technique of finite difference for \( n = 0 \)

\[
\mathbf{H}_0 = \mathbf{J}_0^{-1} = \begin{bmatrix}
24.35719 & 20.96013 & 13.48712 \\
20.96013 & 88.08728 & 56.67909 \\
13.48712 & 56.67909 & 151.69624
\end{bmatrix}
\]

as compared this with \( \mathbf{H}_0 = \mathbf{I} \), the significance is apparent. If a matrix is diagonally dominated such as that of \( \mathbf{H}_0 = \mathbf{I} \), the matrix provided the direction approaching to the steepest descent rather than the direction offered by \( \mathbf{J}_n^{-1} \) to improve the solution. A descent direction does not guarantee the member function \( f \) of the vector function \( \mathbf{f}_n \) decreases simultaneously and with the size of \( \|\mathbf{f}_n\| \). As a result of this, divergence was obtained on occasions.

Once the iterative processes settled down, it is evident that the rate of convergence was improved towards the solution (see later iterations). The \( \mathbf{H}_n \) had accumulated sufficient information from the previous iterations to approximate the \( \mathbf{J}_n^{-1} \) accurately. Subsequently, a faster rate of convergence was obtained.

Table 5.7a shows the performance of various methods. The values are calculated using those highlighted data presented in Tables 5.6a to 5.6d. The criteria
was chosen for the calculations because a tolerance of $0.1 \, (1/s)$ is considered sufficient accurate for most applications including those analysis for the water treatment works.

On comparing the figures shown in Table 5.7a under the heading $N_f$, it appears that the NR method out-performed the quasi-Newton methods by a big margin. The NR method required $N_f = 16$ function evaluations, whereas the DFP method converged so slowly that it required $N_f = 185$ to achieve $\chi_{f+1}$ as compared to $N_f = 48$ and $N_f = 18$ for the BR and BFGS methods respectively. The reason was mentioned previously, namely that the initial $H_0 = I$ was not a sufficiently close approximation to the $J_0^{-1}$. The proof of this can be found in Table 5.7b. The results were obtained using $H_0 = J_0^{-1}$ rather than $H_0 = I$. Consequently, the convergence of the quasi-Newton methods were substantially improved. In fact, the BR, DFP and BFGS methods converged to the $\chi_{f+1}$ in just 7 ($N_f = 7$), 6 ($N_f = 6$) and 6 ($N_f = 6$) iterations respectively. With $H_0 = J_0^{-1}$ used, the quasi-Newton methods out-performed the NR method by as much as 60% (on average) reduction in function evaluations.

Likewise, on comparing the performance $\eta_8$ between methods, similar findings were obtained. With $H_0 = I$, the NR method out-performed the quasi-Newton methods; however, with $H_0 = J_0^{-1}$, the quasi-Newton methods out-performed the NR method by as much as 50% (on average). It is interested to note that the performance deviated little between the DFP method ($\eta_8 = 1.904$) and the BFGS method ($\eta_8 = 1.930$) methods.

Indeed, the use of $H_0 = J_0^{-1}$ has defeated the object of using quasi-Newton methods (namely to avoid setting up $J_s^{-1}$ using partial derivatives), but it has shown the significance of $H_0$ on the performance of the quasi-Newton methods especially the DFP method. When $H_0 = I$ is used, the NR method out-performs the others according to the criteria on convergence described above.
Figure 5.6. A parallel open channel network.

<table>
<thead>
<tr>
<th>Strings</th>
<th>Width, $B$, (m)</th>
<th>Length, $L$, (m)</th>
<th>Bed Slope, $S_b$</th>
<th>Bank Slope, $m$</th>
<th>Manning, $n$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inflow</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Channel - 1</td>
<td>1.00</td>
<td>500</td>
<td>0.0005</td>
<td>0.0</td>
<td>0.013</td>
</tr>
<tr>
<td>Channel - 2</td>
<td>1.50</td>
<td>500</td>
<td>0.0005</td>
<td>0.0</td>
<td>0.013</td>
</tr>
<tr>
<td>Channel - 3</td>
<td>2.00</td>
<td>500</td>
<td>0.0005</td>
<td>0.0</td>
<td>0.013</td>
</tr>
<tr>
<td>Channel - 4</td>
<td>2.50</td>
<td>500</td>
<td>0.0005</td>
<td>0.0</td>
<td>0.013</td>
</tr>
<tr>
<td>Outfall</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 5.4. Parallel Network - Physical characteristics.
### Table 5.5a. Parallel Network - Initial flow distribution.

<table>
<thead>
<tr>
<th>Strings</th>
<th>Initial Energy Head, $H$, (m)</th>
<th>Initial Water Depth, $y$, (m)</th>
<th>Initial Discharge, $Q$, (m$^3$/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Upstream</strong></td>
<td><strong>Downstream</strong></td>
<td><strong>Upstream</strong></td>
</tr>
<tr>
<td>Inflow</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Channel -1</td>
<td>1.00963</td>
<td>1.00000</td>
<td>0.97112</td>
</tr>
<tr>
<td>Channel -2</td>
<td>1.00304</td>
<td>1.00000</td>
<td>0.97204</td>
</tr>
<tr>
<td>Channel -3</td>
<td>1.00139</td>
<td>1.00000</td>
<td>0.97302</td>
</tr>
<tr>
<td>Channel -4</td>
<td>1.00077</td>
<td>1.00000</td>
<td>0.97362</td>
</tr>
<tr>
<td>Outfall</td>
<td>1.00000</td>
<td>1.00000</td>
<td>-</td>
</tr>
</tbody>
</table>

### Table 5.5b. Parallel Network - Final flow distribution.

<table>
<thead>
<tr>
<th>Strings</th>
<th>Final Energy Head, $H$, (m)</th>
<th>Final Water Depth, $y$, (m)</th>
<th>Final Discharge, $Q$, (m$^3$/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Upstream</strong></td>
<td><strong>Downstream</strong></td>
<td><strong>Upstream</strong></td>
</tr>
<tr>
<td>Inflow</td>
<td>1.00192</td>
<td>1.00192</td>
<td>-</td>
</tr>
<tr>
<td>Channel -1</td>
<td>1.00192</td>
<td>1.00000</td>
<td>0.97420</td>
</tr>
<tr>
<td>Channel -2</td>
<td>1.00192</td>
<td>1.00000</td>
<td>0.97312</td>
</tr>
<tr>
<td>Channel -3</td>
<td>1.00192</td>
<td>1.00000</td>
<td>0.97226</td>
</tr>
<tr>
<td>Channel -4</td>
<td>1.00192</td>
<td>1.00000</td>
<td>0.97157</td>
</tr>
<tr>
<td>Outflow</td>
<td>1.00000</td>
<td>1.00000</td>
<td>-</td>
</tr>
</tbody>
</table>
Table 5.6a. Parallel Network - Convergence characteristics of the NR method.
### Table 5.6b. Parallel Network - Convergence characteristics of the BR method.

<table>
<thead>
<tr>
<th>Iteration, $n$</th>
<th>$Q_1$, (m$^3$/s)</th>
<th>$Q_2$, (m$^3$/s)</th>
<th>$Q_3$, (m$^3$/s)</th>
<th>$Q_4$, (m$^3$/s)</th>
<th>$|f_s|$, (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.50000</td>
<td>0.50000</td>
<td>0.50000</td>
<td>0.50000</td>
<td>6.82379E-03</td>
</tr>
<tr>
<td>1</td>
<td>0.49341</td>
<td>0.50494</td>
<td>0.50103</td>
<td>0.50062</td>
<td>6.53114E-03</td>
</tr>
<tr>
<td>2</td>
<td>0.34781</td>
<td>0.61097</td>
<td>0.52621</td>
<td>0.51501</td>
<td>3.10764E-03</td>
</tr>
<tr>
<td>3</td>
<td>0.32708</td>
<td>0.62260</td>
<td>0.53243</td>
<td>0.51788</td>
<td>3.31915E-03</td>
</tr>
<tr>
<td>4</td>
<td>0.32186</td>
<td>0.62185</td>
<td>0.53679</td>
<td>0.51950</td>
<td>3.31413E-03</td>
</tr>
<tr>
<td>5</td>
<td>0.31896</td>
<td>0.61593</td>
<td>0.54330</td>
<td>0.52181</td>
<td>3.19422E-03</td>
</tr>
<tr>
<td>6</td>
<td>0.28830</td>
<td>0.54632</td>
<td>0.61659</td>
<td>0.54879</td>
<td>1.98469E-03</td>
</tr>
<tr>
<td>7</td>
<td>0.26553</td>
<td>0.49176</td>
<td>0.67175</td>
<td>0.57097</td>
<td>1.58981E-03</td>
</tr>
<tr>
<td>8</td>
<td>0.26348</td>
<td>0.48418</td>
<td>0.67715</td>
<td>0.57520</td>
<td>1.57812E-03</td>
</tr>
<tr>
<td>9</td>
<td>0.26766</td>
<td>0.49468</td>
<td>0.66751</td>
<td>0.57015</td>
<td>1.58026E-03</td>
</tr>
<tr>
<td>10</td>
<td>0.25275</td>
<td>0.45666</td>
<td>0.70044</td>
<td>0.59015</td>
<td>1.68124E-03</td>
</tr>
<tr>
<td>11</td>
<td>0.25152</td>
<td>0.45309</td>
<td>0.70159</td>
<td>0.59379</td>
<td>1.68296E-03</td>
</tr>
<tr>
<td>12</td>
<td>0.25518</td>
<td>0.46241</td>
<td>0.69635</td>
<td>0.58607</td>
<td>1.65572E-03</td>
</tr>
<tr>
<td>13</td>
<td>0.24824</td>
<td>0.44480</td>
<td>0.70416</td>
<td>0.60280</td>
<td>1.68981E-03</td>
</tr>
<tr>
<td>14</td>
<td>0.24735</td>
<td>0.44269</td>
<td>0.70295</td>
<td>0.60701</td>
<td>1.66778E-03</td>
</tr>
<tr>
<td>15</td>
<td>0.24197</td>
<td>0.43060</td>
<td>0.69296</td>
<td>0.63446</td>
<td>1.50151E-03</td>
</tr>
<tr>
<td>16</td>
<td>0.23309</td>
<td>0.41301</td>
<td>0.65949</td>
<td>0.69441</td>
<td>9.99841E-04</td>
</tr>
<tr>
<td>17</td>
<td>0.22498</td>
<td>0.39806</td>
<td>0.58524</td>
<td>0.79172</td>
<td>4.21355E-05</td>
</tr>
<tr>
<td>18</td>
<td>0.22457</td>
<td>0.39724</td>
<td>0.58230</td>
<td>0.79589</td>
<td>8.36878E-05</td>
</tr>
<tr>
<td>19</td>
<td>0.22539</td>
<td>0.39885</td>
<td>0.58813</td>
<td>0.78762</td>
<td>1.29659E-06</td>
</tr>
<tr>
<td>20</td>
<td>0.22541</td>
<td>0.39888</td>
<td>0.58823</td>
<td>0.78748</td>
<td>6.33855E-07</td>
</tr>
<tr>
<td>21</td>
<td>0.22540</td>
<td>0.39886</td>
<td>0.58817</td>
<td>0.78757</td>
<td>8.25654E-07</td>
</tr>
<tr>
<td>22</td>
<td>0.22539</td>
<td>0.39883</td>
<td>0.58822</td>
<td>0.78756</td>
<td>1.77636E-15</td>
</tr>
<tr>
<td>23</td>
<td>0.22539</td>
<td>0.39883</td>
<td>0.58822</td>
<td>0.78756</td>
<td>0</td>
</tr>
</tbody>
</table>
### Table 5.6c. Parallel Network - Convergence characteristics of the DFP method.

| Iteration, $n$ | $Q_1$, (m$^3$/s) | $Q_2$, (m$^3$/s) | $Q_3$, (m$^3$/s) | $Q_4$, (m$^3$/s) | $||r_n||$, (m) |
|---------------|------------------|------------------|------------------|------------------|----------------|
| 0             | 0.50000          | 0.50000          | 0.50000          | 0.50000          | 6.82379E-03    |
| 1             | 0.49341          | 0.50494          | 0.50103          | 0.50062          | 6.53114E-03    |
| 2             | 0.34653          | 0.61247          | 0.52601          | 0.51499          | 3.13037E-03    |
| 3             | 0.32405          | 0.62608          | 0.53201          | 0.51786          | 3.40066E-03    |
| 4             | 0.30553          | 0.63453          | 0.53906          | 0.52088          | 3.66944E-03    |
| 5             | 0.28563          | 0.64096          | 0.54864          | 0.52477          | 3.95679E-03    |
| 6             | 0.26584          | 0.64478          | 0.56007          | 0.52931          | 4.21412E-03    |
| 7             | 0.24511          | 0.64624          | 0.57390          | 0.53475          | 4.43866E-03    |
| 8             | 0.22487          | 0.64516          | 0.58921          | 0.54076          | 4.60177E-03    |
| 9             | 0.20449          | 0.64155          | 0.60639          | 0.54757          | 4.70410E-03    |
| 10            | 0.18524          | 0.63557          | 0.62440          | 0.55480          | 4.73716E-03    |
| 11            | 0.16682          | 0.62719          | 0.64341          | 0.56258          | 4.70644E-03    |
| 12            | 0.15022          | 0.61684          | 0.66238          | 0.57056          | 4.62004E-03    |
| 13            | 0.13527          | 0.60452          | 0.68141          | 0.57881          | 4.48854E-03    |
| 14            | 0.12256          | 0.59075          | 0.69965          | 0.58703          | 4.32837E-03    |
| 15            | 0.11190          | 0.57547          | 0.71729          | 0.59534          | 4.15058E-03    |
| ...           | ...              | ...              | ...              | ...              | ...            |
| 180           | 0.22528          | 0.39935          | 0.58861          | 0.78676          | 9 80067E-06    |
| 181           | 0.22522          | 0.39918          | 0.58873          | 0.78687          | 9 18507E-06    |
| 182           | 0.22522          | 0.39901          | 0.58881          | 0.78696          | 8 49371E-06    |
| 183           | 0.22526          | 0.39887          | 0.58884          | 0.78704          | 8 02923E-06    |
| 184           | 0.22531          | 0.39876          | 0.58883          | 0.78711          | 7 84882E-06    |
| 185           | 0.22536          | 0.39867          | 0.58879          | 0.78718          | 7 85920E-06    |
| ...           | ...              | ...              | ...              | ...              | ...            |
| 232           | 0.22539          | 0.39883          | 0.58822          | 0.78756          | 2 51215E-15    |
| 233           | 0.22539          | 0.39883          | 0.58822          | 0.78756          | 0              |
### Broyden-Fletcher-Goldfarb-Shanno

<table>
<thead>
<tr>
<th>Iteration, ( n )</th>
<th>( Q_1 ), (m³/s)</th>
<th>( Q_2 ), (m³/s)</th>
<th>( Q_3 ), (m³/s)</th>
<th>( Q_4 ), (m³/s)</th>
<th>( | r_n | ), (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.50000</td>
<td>0.50000</td>
<td>0.50000</td>
<td>0.50000</td>
<td>6.82379E-03</td>
</tr>
<tr>
<td>1</td>
<td>0.49341</td>
<td>0.50494</td>
<td>0.50103</td>
<td>0.50062</td>
<td>6.53114E-03</td>
</tr>
<tr>
<td>2</td>
<td>0.34643</td>
<td>0.61200</td>
<td>0.52643</td>
<td>0.51514</td>
<td>3.12150E-03</td>
</tr>
<tr>
<td>3</td>
<td>0.32342</td>
<td>0.62530</td>
<td>0.53035</td>
<td>0.51823</td>
<td>3.38509E-03</td>
</tr>
<tr>
<td>4</td>
<td>0.30138</td>
<td>0.63270</td>
<td>0.54343</td>
<td>0.52250</td>
<td>3.65023E-03</td>
</tr>
<tr>
<td>5</td>
<td>0.26801</td>
<td>0.63257</td>
<td>0.56760</td>
<td>0.53182</td>
<td>3.89593E-03</td>
</tr>
<tr>
<td>6</td>
<td>0.22368</td>
<td>0.60973</td>
<td>0.61638</td>
<td>0.55021</td>
<td>3.77923E-03</td>
</tr>
<tr>
<td>7</td>
<td>0.19194</td>
<td>0.55507</td>
<td>0.67887</td>
<td>0.57412</td>
<td>3.06133E-03</td>
</tr>
<tr>
<td>8</td>
<td>0.19712</td>
<td>0.49447</td>
<td>0.71783</td>
<td>0.59058</td>
<td>2.34604E-03</td>
</tr>
<tr>
<td>9</td>
<td>0.22905</td>
<td>0.44725</td>
<td>0.72620</td>
<td>0.59750</td>
<td>1.96758E-03</td>
</tr>
<tr>
<td>10</td>
<td>0.25115</td>
<td>0.43282</td>
<td>0.71745</td>
<td>0.59858</td>
<td>1.87534E-03</td>
</tr>
<tr>
<td>11</td>
<td>0.25850</td>
<td>0.43061</td>
<td>0.71012</td>
<td>0.60078</td>
<td>1.81889E-03</td>
</tr>
<tr>
<td>12</td>
<td>0.27636</td>
<td>0.42556</td>
<td>0.68060</td>
<td>0.61749</td>
<td>1.62214E-03</td>
</tr>
<tr>
<td>13</td>
<td>0.28631</td>
<td>0.41885</td>
<td>0.63953</td>
<td>0.65531</td>
<td>1.38092E-03</td>
</tr>
<tr>
<td>14</td>
<td>0.27742</td>
<td>0.40784</td>
<td>0.58515</td>
<td>0.72959</td>
<td>9.50066E-04</td>
</tr>
<tr>
<td>15</td>
<td>0.25128</td>
<td>0.40085</td>
<td>0.56676</td>
<td>0.78110</td>
<td>4.89314E-04</td>
</tr>
<tr>
<td>16</td>
<td>0.23219</td>
<td>0.39928</td>
<td>0.57700</td>
<td>0.79153</td>
<td>1.66476E-04</td>
</tr>
<tr>
<td>17</td>
<td>0.22558</td>
<td>0.39910</td>
<td>0.58672</td>
<td>0.78860</td>
<td>1.96252E-05</td>
</tr>
<tr>
<td>18</td>
<td>0.22537</td>
<td>0.39986</td>
<td>0.58811</td>
<td>0.78764</td>
<td>3.23210E-06</td>
</tr>
<tr>
<td>19</td>
<td>0.22537</td>
<td>0.39886</td>
<td>0.58822</td>
<td>0.78755</td>
<td>6.79740E-07</td>
</tr>
<tr>
<td>20</td>
<td>0.22539</td>
<td>0.39883</td>
<td>0.58822</td>
<td>0.78755</td>
<td>4.66906E-08</td>
</tr>
<tr>
<td>21</td>
<td>0.22539</td>
<td>0.39883</td>
<td>0.58822</td>
<td>0.78756</td>
<td>3.72889E-09</td>
</tr>
<tr>
<td>22</td>
<td>0.22539</td>
<td>0.39883</td>
<td>0.58822</td>
<td>0.78756</td>
<td>2.52010E-10</td>
</tr>
<tr>
<td>23</td>
<td>0.22539</td>
<td>0.39883</td>
<td>0.58822</td>
<td>0.78756</td>
<td>2.51215E-15</td>
</tr>
</tbody>
</table>

Table 5.6d. Parallel Network - Convergence characteristics of the BFGS method.
### APPLICATIONS AND TESTING OF METHODS OF SOLUTION OF NETWORKS

#### Table 5.7a. Parallel Network - Overall performance of various methods, $H_0 = I$.

| Methods | $n$ | $N_f$ | $|q_0 - \hat{q}|$ | $|q_n - \hat{q}|$ | $\eta_8$ |
|---------|-----|-------|--------------------|--------------------|---------|
| NR      | 4   | 16    | 4.19666E-01        | 1.00000E-07        | 0.953   |
| BR      | 48  | 48    | 4.19666E-01        | 9.29068E-05        | 0.175   |
| DFP     | 185 | 185   | 4.19666E-01        | 7.03376E-04        | 0.035   |
| BFGS    | 18  | 18    | 4.19666E-01        | 1.92595E-04        | 0.427   |

#### Table 5.7b. Parallel Network - Overall performance of various methods, $H_0 = J_0^{-1}$.

| Methods | $n$ | $N_f$ | $|q_0 - \hat{q}|$ | $|q_n - \hat{q}|$ | $\eta_8$ |
|---------|-----|-------|--------------------|--------------------|---------|
| NR      | 4   | 16    | 4.19666E-01        | 1.00000E-07        | 0.953   |
| BR      | 7   | 7     | 4.19666E-01        | 1.47648E-06        | 1.794   |
| DFP     | 6   | 6     | 4.19666E-01        | 4.57712E-06        | 1.904   |
| BFGS    | 6   | 6     | 4.19666E-01        | 3.93065E-06        | 1.930   |
5.3.2 A Nested Network

A second network that comprises eight open channels is shown in Figure 5.7. This configuration is chosen to simulate the situation where comparatively high flows occur in one part of the network while very low flows occur in other parts of the network. The aim is to find out whether the solution methods are capable of achieving a solution of such a difficult network. It is suspected that this configuration may cause oscillations which, in turn, may retard the rate of convergence owing to substantial differences in sensitivity (water depth against discharge) in the channels.

The physical characteristics for the network are shown in Table 5.8. The initial set of flow distribution are shown in Table 5.9a; these were obtained by splitting the incoming flows into equal numbers of outgoing flows. Four computer runs for the network were conducted separately under conditions identical to the parallel network. The results were obtained and are shown in Table 5.9b. Furthermore, the convergence characteristics of the NR, BR, DFP and BFGS methods are shown in Tables 5.10a, 5.10b, 5.10c and 5.10d respectively.

The solution obtained by all of the methods were identical. All the points discussed in the previous example were generally applied in this example too. The property of quadratic convergence of the NR method was also observed in the first four iterations in Table 5.10a. The oscillatory behaviour towards the solution was obtained owing to the instability of the approximate Jacobian matrix $\tilde{J}$ as described in previous example.

Again, the quasi-Newton methods converged comparatively slower than the NR method owing to the $H_0 = I$ being insufficiently accurate to represent the inversion of the Jacobian matrix. Tables 5.11a shows the computational effort as well as performance between methods. In this test case, the quasi-Newton methods required fewer function evaluations than the NR method $N_r = 12$, compared with the previous example. This may be due to the sensitivity of the network, which will be described later in this section. Although the $N_r$ required by the quasi-Newton methods were less than the NR method, the performance of the NR method surpassed the quasi-Newton methods owing to the NR method converging faster and incurring a smaller residue in $\|q - \hat{q}\|$. 
When $H_0 = J_0^{-1}$ is used, the $N_r$ required by the quasi-Newton methods were expected to reduce. The $N_r = 5$ was obtained for the quasi-Newton methods indicates that the overall performance of the methods were improved. The residues in $\|q_n - \hat{q}\|$ were also found to reduce because of the faster convergence rate. With both reduction in $N_r$ and $\|q_n - \hat{q}\|$, the performance of quasi-Newton methods were better than the NR method by an average of 50%. Apparently, the success of the methods hinges on the approximation $H_0 = J_0^{-1}$ for subsequent iterations.

An examination between Tables 5.5b and 5.9b also reveals that a very small head-loss, approximately 0.002 (m), and a comparatively high head-loss, approximately 0.050 (m), were obtained for the parallel network and the nested network respectively. This considerable difference in head-losses suggests that the rate of convergence may be affected by differing sensitivity in the networks. An investigation, therefore, is in order. This can be accomplished by calculating the finite ratio

$$|\varphi_{n+1}| = \frac{\|f_{n+1} - f_n\|}{\|q_{n+1}\|}$$

between two consecutive iterations. It describes the response of $\|f_n\|$ with respect to a unit change in $\|q_{n+1}\|$. Hence, the sensitivities of the two systems can be examined. Using the highlighted figures shown in Tables 5.6a and 5.10a, the values of $\varphi_{n+1}$ for both networks are calculated as follows:

Parallel Network: $|\varphi_4| = \frac{\|f_4 - f_3\|}{\|q_4\|} = \frac{2.22600 \times 10^{-6}}{1.39284 \times 10^{-4}} = 1.59817 \times 10^{-2}$

Nested Network: $|\varphi_4| = \frac{\|f_4 - f_3\|}{\|q_4\|} = \frac{2.63026 \times 10^{-6}}{1.86279 \times 10^{-4}} = 3.34458 \times 10^{-2}$

Iterations 3 and 4 were chosen for the calculations because they satisfied the condition $\chi_{n+1}$. It is seen that the value of $\varphi_4$ calculated for the parallel network is less than that for the nested network. A small value of $\varphi_4$ means that the $\|f_n\|$ is insensitive to the
adjustments made in the $\|q_n\|$. Thus, in a network of low sensitivity, the order of magnitude of the $\|f_n\|$, to certain extent, is little affected by the variations of the $\|q_n\|$. With a slowly decreasing magnitude of $\|f_n\|$ to calculate $\Sigma_n$ and then update the $H_n$, the consequence was the slow rate of convergence obtained.

To bring some further support to this argument, two further runs on the parallel network were conducted separately under the same conditions using different lengths of channel, with the intention of simulating both low head-loss and high head-loss situations. The first run comprised channels of 5 (m) in length while the second runs comprised channels 1000 (m) in length. The head-losses found in Tables 5.12b and 5.14b for the two runs are approximately 0.0002 (m) and 0.0843 (m), and the performance of the methods are shown in Tables 5.13 and 5.15.

With the 5-metre network, both BR and DFP methods were unable to converge to $\chi_{j+1}$ owing to the sensitivity of the network being so low that the system of equations are possibly ill-conditioned, and seemed to oscillate indefinitely; with the 1000-metre network, both methods converged to the solution. From this example alone, it is seen both methods BR and DFP methods are very sensitive to the form of the system of equations so that these methods cannot be used with full confidence for general applications. On the other hand, the performance of the NR method deviated little from previous examples; whereas the performance of the BFGS method, as expected, gradually improved as the sensitivity of the network increased. With the high sensitivity network, although the BFGS method ($N_f = 8$) converged 2.5 times slower than the NR method ($N_f = 3$), it appears that its performance ($\eta_b = 1.427$) is comparable to NR method ($\eta_b = 1.525$) owing to the fact that BFGS method required fewer function evaluations.

The results in Tables 5.7 and 5.11, and Tables 5.13 and 5.15 support the argument that the performance of the quasi-Newton methods is affected by the sensitivity of the network. The BR and DFP methods are worst affected in the test cases. With a high sensitivity network, the performance of the BFGS method is comparable to the NR method, but the best performance is obtained by the NR method.
Figure 5.7. A nested open channel network.

Table 5.8. Nested Network - Physical characteristics.
## Applications and Testing of Methods of Solution of Networks

### Initial Energy Head, \( H \), (m)  
<table>
<thead>
<tr>
<th>Strings</th>
<th>Initial Water Depth, ( y ), (m)</th>
<th>Initial Discharge, ( Q ), (m³/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Upstream</td>
<td>Downstream</td>
</tr>
<tr>
<td>Inflow</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Channel - 1</td>
<td>1.09588</td>
<td>1.07173</td>
</tr>
<tr>
<td>Channel - 2</td>
<td>1.02041</td>
<td>1.00501</td>
</tr>
<tr>
<td>Channel - 3</td>
<td>1.07460</td>
<td>1.07173</td>
</tr>
<tr>
<td>Channel - 4</td>
<td>1.00501</td>
<td>1.00155</td>
</tr>
<tr>
<td>Channel - 5</td>
<td>1.07173</td>
<td>1.04308</td>
</tr>
<tr>
<td>Channel - 6</td>
<td>1.04380</td>
<td>1.04308</td>
</tr>
<tr>
<td>Channel - 7</td>
<td>1.00155</td>
<td>1.00000</td>
</tr>
<tr>
<td>Channel - 8</td>
<td>1.04308</td>
<td>1.00000</td>
</tr>
<tr>
<td>Outfall</td>
<td>1.00000</td>
<td>1.00000</td>
</tr>
</tbody>
</table>

### Final Energy Head, \( H \), (m)  
<table>
<thead>
<tr>
<th>Strings</th>
<th>Final Water Depth, ( y ), (m)</th>
<th>Final Discharge, ( Q ), (m³/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Upstream</td>
<td>Downstream</td>
</tr>
<tr>
<td>Inflow</td>
<td>1.05012</td>
<td>1.05012</td>
</tr>
<tr>
<td>Channel - 1</td>
<td>1.05012</td>
<td>1.03067</td>
</tr>
<tr>
<td>Channel - 2</td>
<td>1.05012</td>
<td>1.03100</td>
</tr>
<tr>
<td>Channel - 3</td>
<td>1.03100</td>
<td>1.03067</td>
</tr>
<tr>
<td>Channel - 4</td>
<td>1.03100</td>
<td>1.01767</td>
</tr>
<tr>
<td>Channel - 5</td>
<td>1.03067</td>
<td>1.01739</td>
</tr>
<tr>
<td>Channel - 6</td>
<td>1.01767</td>
<td>1.01739</td>
</tr>
<tr>
<td>Channel - 7</td>
<td>1.01767</td>
<td>1.00000</td>
</tr>
<tr>
<td>Channel - 8</td>
<td>1.01739</td>
<td>1.00000</td>
</tr>
<tr>
<td>Outfall</td>
<td>1.00000</td>
<td>1.00000</td>
</tr>
</tbody>
</table>

Table 5.9a. Nested Network - Initial flow distribution.

Table 5.9b. Nested Network - Final flow distribution.
### Applications and Testing of Methods of Solution of Networks

<table>
<thead>
<tr>
<th>Iteration, ( n )</th>
<th>( Q_1 ) ((\text{m}^3/\text{s}))</th>
<th>( Q_2 ) ((\text{m}^3/\text{s}))</th>
<th>( Q_3 ) ((\text{m}^3/\text{s}))</th>
<th>( Q_4 ) ((\text{m}^3/\text{s}))</th>
<th>( |f_n|_2 ) ((\text{m}))</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1.00000</td>
<td>1.00000</td>
<td>0.50000</td>
<td>0.50000</td>
<td>4.89372E-02</td>
</tr>
<tr>
<td>1</td>
<td>0.82520</td>
<td>1.17480</td>
<td>0.18893</td>
<td>0.98587</td>
<td>2.59986E-03</td>
</tr>
<tr>
<td>2</td>
<td>0.85246</td>
<td>1.14754</td>
<td>0.16313</td>
<td>0.98441</td>
<td>6.23043E-06</td>
</tr>
<tr>
<td>3</td>
<td>0.85243</td>
<td>1.14757</td>
<td>0.16308</td>
<td>0.98449</td>
<td>1.65225E-10</td>
</tr>
<tr>
<td>4</td>
<td>0.85243</td>
<td>1.14757</td>
<td>0.16308</td>
<td>0.98449</td>
<td>4.47568E-12</td>
</tr>
<tr>
<td>5</td>
<td>0.85243</td>
<td>1.14757</td>
<td>0.16308</td>
<td>0.98449</td>
<td>3.51432E-12</td>
</tr>
<tr>
<td>6</td>
<td>0.85243</td>
<td>1.14757</td>
<td>0.16308</td>
<td>0.98449</td>
<td>2.42795E-12</td>
</tr>
<tr>
<td>7</td>
<td>0.85243</td>
<td>1.14757</td>
<td>0.16308</td>
<td>0.98449</td>
<td>8.53540E-13</td>
</tr>
<tr>
<td>8</td>
<td>0.85243</td>
<td>1.14757</td>
<td>0.16308</td>
<td>0.98449</td>
<td>1.67013E-12</td>
</tr>
<tr>
<td>9</td>
<td>0.85243</td>
<td>1.14757</td>
<td>0.16308</td>
<td>0.98449</td>
<td>4.00125E-13</td>
</tr>
<tr>
<td>10</td>
<td>0.85243</td>
<td>1.14757</td>
<td>0.16308</td>
<td>0.98449</td>
<td>1.90859E-12</td>
</tr>
<tr>
<td>11</td>
<td>0.85243</td>
<td>1.14757</td>
<td>0.16308</td>
<td>0.98449</td>
<td>1.80928E-12</td>
</tr>
<tr>
<td>12</td>
<td>0.85243</td>
<td>1.14757</td>
<td>0.16308</td>
<td>0.98449</td>
<td>1.71576E-12</td>
</tr>
<tr>
<td>13</td>
<td>0.85243</td>
<td>1.14757</td>
<td>0.16308</td>
<td>0.98449</td>
<td>5.23582E-13</td>
</tr>
<tr>
<td>14</td>
<td>0.85243</td>
<td>1.14757</td>
<td>0.16308</td>
<td>0.98449</td>
<td>1.83424E-12</td>
</tr>
<tr>
<td>15</td>
<td>0.85243</td>
<td>1.14757</td>
<td>0.16308</td>
<td>0.98449</td>
<td>7.89147E-13</td>
</tr>
<tr>
<td>16</td>
<td>0.85243</td>
<td>1.14757</td>
<td>0.16308</td>
<td>0.98449</td>
<td>1.69707E-12</td>
</tr>
<tr>
<td>17</td>
<td>0.85243</td>
<td>1.14757</td>
<td>0.16308</td>
<td>0.98449</td>
<td>4.75621E-13</td>
</tr>
<tr>
<td>18</td>
<td>0.85243</td>
<td>1.14757</td>
<td>0.16308</td>
<td>0.98449</td>
<td>1.86224E-12</td>
</tr>
<tr>
<td>19</td>
<td>0.85243</td>
<td>1.14757</td>
<td>0.16308</td>
<td>0.98449</td>
<td>8.44880E-13</td>
</tr>
<tr>
<td>20</td>
<td>0.85243</td>
<td>1.14757</td>
<td>0.16308</td>
<td>0.98449</td>
<td>1.67326E-12</td>
</tr>
<tr>
<td>21</td>
<td>0.85243</td>
<td>1.14757</td>
<td>0.16308</td>
<td>0.98449</td>
<td>4.09674E-13</td>
</tr>
<tr>
<td>22</td>
<td>0.85243</td>
<td>1.14757</td>
<td>0.16308</td>
<td>0.98449</td>
<td>1.90318E-12</td>
</tr>
<tr>
<td>\ldots</td>
<td>\vdots</td>
<td>\vdots</td>
<td>\vdots</td>
<td>\vdots</td>
<td>\vdots</td>
</tr>
<tr>
<td>100</td>
<td>0.85243</td>
<td>1.14757</td>
<td>0.16308</td>
<td>0.98449</td>
<td>1.59032E-12</td>
</tr>
<tr>
<td>101</td>
<td>0.85243</td>
<td>1.14757</td>
<td>0.16308</td>
<td>0.98449</td>
<td>2.27270E-12</td>
</tr>
</tbody>
</table>

Table 5.10a. Nested Network - Convergence characteristics of the NR method.

176
Table 5.10a (cont). Nested Network - Convergence characteristics of the NR method.

<table>
<thead>
<tr>
<th>Iteration, n</th>
<th>$Q_5$, (m³/s)</th>
<th>$Q_6$, (m³/s)</th>
<th>$Q_7$, (m³/s)</th>
<th>$Q_8$, (m³/s)</th>
<th>$|r_n|$, (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1.50000</td>
<td>0.25000</td>
<td>0.25000</td>
<td>1.75000</td>
<td>4.89372E-02</td>
</tr>
<tr>
<td>1</td>
<td>1.01413</td>
<td>0.16690</td>
<td>0.81897</td>
<td>1.18103</td>
<td>2.59986E-03</td>
</tr>
<tr>
<td>2</td>
<td>1.01559</td>
<td>0.14975</td>
<td>0.83466</td>
<td>1.16534</td>
<td>6.23043E-06</td>
</tr>
<tr>
<td>3</td>
<td>1.01551</td>
<td>0.14987</td>
<td>0.83462</td>
<td>1.16538</td>
<td>1.65225E-10</td>
</tr>
<tr>
<td>4</td>
<td>1.01551</td>
<td>0.14987</td>
<td>0.83462</td>
<td>1.16538</td>
<td>4.47568E-12</td>
</tr>
<tr>
<td>5</td>
<td>1.01551</td>
<td>0.14987</td>
<td>0.83462</td>
<td>1.16538</td>
<td>3.51432E-12</td>
</tr>
<tr>
<td>6</td>
<td>1.01551</td>
<td>0.14987</td>
<td>0.83462</td>
<td>1.16538</td>
<td>2.42795E-12</td>
</tr>
<tr>
<td>7</td>
<td>1.01551</td>
<td>0.14987</td>
<td>0.83462</td>
<td>1.16538</td>
<td>8.53540E-13</td>
</tr>
<tr>
<td>8</td>
<td>1.01551</td>
<td>0.14987</td>
<td>0.83462</td>
<td>1.16538</td>
<td>1.67013E-12</td>
</tr>
<tr>
<td>9</td>
<td>1.01551</td>
<td>0.14987</td>
<td>0.83462</td>
<td>1.16538</td>
<td>4.00125E-13</td>
</tr>
<tr>
<td>10</td>
<td>1.01551</td>
<td>0.14987</td>
<td>0.83462</td>
<td>1.16538</td>
<td>1.90859E-12</td>
</tr>
<tr>
<td>11</td>
<td>1.01551</td>
<td>0.14987</td>
<td>0.83462</td>
<td>1.16538</td>
<td>1.80928E-12</td>
</tr>
<tr>
<td>12</td>
<td>1.01551</td>
<td>0.14987</td>
<td>0.83462</td>
<td>1.16538</td>
<td>1.71576E-12</td>
</tr>
<tr>
<td>13</td>
<td>1.01551</td>
<td>0.14987</td>
<td>0.83462</td>
<td>1.16538</td>
<td>5.23582E-13</td>
</tr>
<tr>
<td>14</td>
<td>1.01551</td>
<td>0.14987</td>
<td>0.83462</td>
<td>1.16538</td>
<td>1.83424E-12</td>
</tr>
<tr>
<td>15</td>
<td>1.01551</td>
<td>0.14987</td>
<td>0.83462</td>
<td>1.16538</td>
<td>7.89147E-13</td>
</tr>
<tr>
<td>16</td>
<td>1.01551</td>
<td>0.14987</td>
<td>0.83462</td>
<td>1.16538</td>
<td>1.69707E-12</td>
</tr>
<tr>
<td>17</td>
<td>1.01551</td>
<td>0.14987</td>
<td>0.83462</td>
<td>1.16538</td>
<td>4.75621E-13</td>
</tr>
<tr>
<td>18</td>
<td>1.01551</td>
<td>0.14987</td>
<td>0.83462</td>
<td>1.16538</td>
<td>1.86224E-12</td>
</tr>
<tr>
<td>19</td>
<td>1.01551</td>
<td>0.14987</td>
<td>0.83462</td>
<td>1.16538</td>
<td>8.44880E-13</td>
</tr>
<tr>
<td>20</td>
<td>1.01551</td>
<td>0.14987</td>
<td>0.83462</td>
<td>1.16538</td>
<td>1.67326E-12</td>
</tr>
<tr>
<td>21</td>
<td>1.01551</td>
<td>0.14987</td>
<td>0.83462</td>
<td>1.16538</td>
<td>4.09674E-13</td>
</tr>
<tr>
<td>22</td>
<td>1.01551</td>
<td>0.14987</td>
<td>0.83462</td>
<td>1.16538</td>
<td>1.90318E-12</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>100</td>
<td>1.01551</td>
<td>0.14987</td>
<td>0.83462</td>
<td>1.16538</td>
<td>1.59032E-12</td>
</tr>
<tr>
<td>101</td>
<td>1.01551</td>
<td>0.14987</td>
<td>0.83462</td>
<td>1.16538</td>
<td>2.27270E-12</td>
</tr>
</tbody>
</table>

Newton-Raphson
### Applications and Testing of Methods of Solution of Networks

**Broyden**

<table>
<thead>
<tr>
<th>Iteration, n</th>
<th>$Q_1$, (m³/s)</th>
<th>$Q_2$, (m³/s)</th>
<th>$Q_3$, (m³/s)</th>
<th>$Q_4$, (m³/s)</th>
<th>$|r_n|_2$ (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1.00000</td>
<td>1.00000</td>
<td>0.50000</td>
<td>0.50000</td>
<td>4.89372E-02</td>
</tr>
<tr>
<td>1</td>
<td>0.99131</td>
<td>1.00869</td>
<td>0.48199</td>
<td>0.52669</td>
<td>4.57919E-02</td>
</tr>
<tr>
<td>2</td>
<td>0.86237</td>
<td>1.13763</td>
<td>0.21755</td>
<td>0.92008</td>
<td>4.71713E-03</td>
</tr>
<tr>
<td>3</td>
<td>0.86160</td>
<td>1.13840</td>
<td>0.21383</td>
<td>0.92457</td>
<td>4.42212E-03</td>
</tr>
<tr>
<td>4</td>
<td>0.85539</td>
<td>1.14461</td>
<td>0.18055</td>
<td>0.96407</td>
<td>1.50565E-03</td>
</tr>
<tr>
<td>5</td>
<td>0.85187</td>
<td>1.14813</td>
<td>0.16188</td>
<td>0.98625</td>
<td>1.45996E-04</td>
</tr>
<tr>
<td>6</td>
<td>0.85210</td>
<td>1.14790</td>
<td>0.16304</td>
<td>0.98486</td>
<td>3.98819E-05</td>
</tr>
<tr>
<td>7</td>
<td>0.85223</td>
<td>1.14777</td>
<td>0.16352</td>
<td>0.98425</td>
<td>2.44111E-05</td>
</tr>
<tr>
<td>8</td>
<td>0.85222</td>
<td>1.14778</td>
<td>0.16335</td>
<td>0.98444</td>
<td>1.92389E-05</td>
</tr>
<tr>
<td>9</td>
<td>0.85223</td>
<td>1.14777</td>
<td>0.16332</td>
<td>0.98444</td>
<td>1.75528E-05</td>
</tr>
<tr>
<td>10</td>
<td>0.85238</td>
<td>1.14762</td>
<td>0.16314</td>
<td>0.98448</td>
<td>4.05413E-06</td>
</tr>
<tr>
<td>11</td>
<td>0.85243</td>
<td>1.14757</td>
<td>0.16308</td>
<td>0.98449</td>
<td>5.35614E-08</td>
</tr>
<tr>
<td>12</td>
<td>0.85243</td>
<td>1.14757</td>
<td>0.16308</td>
<td>0.98449</td>
<td>5.28527E-10</td>
</tr>
<tr>
<td>13</td>
<td>0.85243</td>
<td>1.14757</td>
<td>0.16308</td>
<td>0.98449</td>
<td>5.40637E-10</td>
</tr>
<tr>
<td>14</td>
<td>0.85243</td>
<td>1.14757</td>
<td>0.16308</td>
<td>0.98449</td>
<td>1.58080E-10</td>
</tr>
<tr>
<td>15</td>
<td>0.85243</td>
<td>1.14757</td>
<td>0.16308</td>
<td>0.98449</td>
<td>1.20627E-11</td>
</tr>
<tr>
<td>16</td>
<td>0.85243</td>
<td>1.14757</td>
<td>0.16308</td>
<td>0.98449</td>
<td>7.18467E-12</td>
</tr>
<tr>
<td>17</td>
<td>0.85243</td>
<td>1.14757</td>
<td>0.16308</td>
<td>0.98449</td>
<td>3.47313E-12</td>
</tr>
<tr>
<td>18</td>
<td>0.85243</td>
<td>1.14757</td>
<td>0.16308</td>
<td>0.98449</td>
<td>7.53638E-12</td>
</tr>
<tr>
<td>19</td>
<td>0.85243</td>
<td>1.14757</td>
<td>0.16308</td>
<td>0.98449</td>
<td>4.65821E-12</td>
</tr>
<tr>
<td>20</td>
<td>0.85243</td>
<td>1.14757</td>
<td>0.16308</td>
<td>0.98449</td>
<td>3.68039E-12</td>
</tr>
<tr>
<td>21</td>
<td>0.85243</td>
<td>1.14757</td>
<td>0.16308</td>
<td>0.98449</td>
<td>3.56239E-12</td>
</tr>
<tr>
<td>22</td>
<td>0.85243</td>
<td>1.14757</td>
<td>0.16308</td>
<td>0.98449</td>
<td>2.72970E-11</td>
</tr>
<tr>
<td>23</td>
<td>0.85243</td>
<td>1.14757</td>
<td>0.16308</td>
<td>0.98449</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>0.85243</td>
<td>1.14757</td>
<td>0.16308</td>
<td>0.98449</td>
<td>3.14019E-16</td>
</tr>
<tr>
<td>25</td>
<td>0.85243</td>
<td>1.14757</td>
<td>0.16308</td>
<td>0.98449</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 5.10b. Nested Network - Convergence characteristics of the BR method.
| Iteration, n | $Q_{3r}$ (m³/s) | $Q_{4r}$ (m³/s) | $Q_{7r}$ (m³/s) | $Q_{8r}$ (m³/s) | $|r_n|$, (m) |
|------------|----------------|----------------|----------------|----------------|----------------|
| 0          | 1.50000        | 0.25000        | 0.25000        | 1.75000        | 4.89372E-02   |
| 1          | 1.47331        | 0.23661        | 0.29008        | 1.70992        | 4.57919E-02   |
| 2          | 1.07992        | 0.05071        | 0.86937        | 1.13063        | 4.71713E-03   |
| 3          | 1.07543        | 0.05693        | 0.86764        | 1.13236        | 4.42212E-03   |
| 4          | 1.03593        | 0.11867        | 0.84540        | 1.15460        | 1.50565E-03   |
| 5          | 1.01375        | 0.15280        | 0.83345        | 1.16555        | 1.45996E-04   |
| 6          | 1.01514        | 0.15050        | 0.83436        | 1.16564        | 3.98819E-05   |
| 7          | 1.01575        | 0.14955        | 0.83470        | 1.16530        | 2.44111E-05   |
| 8          | 1.01556        | 0.14984        | 0.83460        | 1.16540        | 1.92389E-05   |
| 9          | 1.01556        | 0.14985        | 0.83460        | 1.16540        | 1.75528E-05   |
| 10         | 1.01552        | 0.14986        | 0.83462        | 1.16538        | 4.05413E-06   |
| 11         | 1.01551        | 0.14987        | 0.83462        | 1.16538        | 5.35614E-08   |
| 12         | 1.01551        | 0.14987        | 0.83462        | 1.16538        | 5.28527E-10   |
| 13         | 1.01551        | 0.14987        | 0.83462        | 1.16538        | 5.40637E-10   |
| 14         | 1.01551        | 0.14987        | 0.83462        | 1.16538        | 1.58080E-10   |
| 15         | 1.01551        | 0.14987        | 0.83462        | 1.16538        | 1.20627E-11   |
| 16         | 1.01551        | 0.14987        | 0.83462        | 1.16538        | 7.18467E-12   |
| 17         | 1.01551        | 0.14987        | 0.83462        | 1.16538        | 3.47313E-12   |
| 18         | 1.01551        | 0.14987        | 0.83462        | 1.16538        | 7.53638E-12   |
| 19         | 1.01551        | 0.14987        | 0.83462        | 1.16538        | 4.65821E-12   |
| 20         | 1.01551        | 0.14987        | 0.83462        | 1.16538        | 3.68039E-12   |
| 21         | 1.01551        | 0.14987        | 0.83462        | 1.16538        | 3.56239E-12   |
| 22         | 1.01551        | 0.14987        | 0.83462        | 1.16538        | 2.72970E-11   |
| 3          | 1.01551        | 0.14987        | 0.83462        | 1.16538        | 3.14019E-16   |
| 44         | 1.01551        | 0.14987        | 0.83462        | 1.16538        | 0             |

Table 5.10b (cont). Nested Network - Convergence characteristics of the BR method.
### Table 5.10c. Nested Network - Convergence characteristics of the DFP method.

<table>
<thead>
<tr>
<th>Iteration, n</th>
<th>$Q_1$ (m$^3$/s)</th>
<th>$Q_2$ (m$^3$/s)</th>
<th>$Q_3$ (m$^3$/s)</th>
<th>$Q_4$ (m$^3$/s)</th>
<th>$|r_n|$ (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1.00000</td>
<td>1.00000</td>
<td>0.50000</td>
<td>0.50000</td>
<td>4.89372E-02</td>
</tr>
<tr>
<td>1</td>
<td>0.99131</td>
<td>1.00869</td>
<td>0.48199</td>
<td>0.52669</td>
<td>4.57919E-02</td>
</tr>
<tr>
<td>2</td>
<td>0.86226</td>
<td>1.13774</td>
<td>0.21725</td>
<td>0.92049</td>
<td>4.73030E-03</td>
</tr>
<tr>
<td>3</td>
<td>0.86149</td>
<td>1.13851</td>
<td>0.21351</td>
<td>0.92500</td>
<td>4.44204E-03</td>
</tr>
<tr>
<td>4</td>
<td>0.85163</td>
<td>1.14837</td>
<td>0.16099</td>
<td>0.98738</td>
<td>2.89207E-04</td>
</tr>
<tr>
<td>5</td>
<td>0.85223</td>
<td>1.14777</td>
<td>0.16246</td>
<td>0.95311</td>
<td>1.05549E-04</td>
</tr>
<tr>
<td>6</td>
<td>0.85251</td>
<td>1.14749</td>
<td>0.16299</td>
<td>0.98450</td>
<td>6.67407E-06</td>
</tr>
<tr>
<td>7</td>
<td>0.85250</td>
<td>1.14750</td>
<td>0.16300</td>
<td>0.98450</td>
<td>6.08369E-06</td>
</tr>
<tr>
<td>8</td>
<td>0.85245</td>
<td>1.14755</td>
<td>0.16303</td>
<td>0.98452</td>
<td>3.04764E-06</td>
</tr>
<tr>
<td>9</td>
<td>0.85244</td>
<td>1.14756</td>
<td>0.16304</td>
<td>0.98451</td>
<td>2.41094E-06</td>
</tr>
<tr>
<td>10</td>
<td>0.85243</td>
<td>1.14757</td>
<td>0.16308</td>
<td>0.98450</td>
<td>9.79484E-07</td>
</tr>
<tr>
<td>11</td>
<td>0.85243</td>
<td>1.14757</td>
<td>0.16308</td>
<td>0.98449</td>
<td>6.03826E-07</td>
</tr>
<tr>
<td>12</td>
<td>0.85243</td>
<td>1.14757</td>
<td>0.16308</td>
<td>0.98449</td>
<td>1.59667E-07</td>
</tr>
<tr>
<td>13</td>
<td>0.85243</td>
<td>1.14757</td>
<td>0.16308</td>
<td>0.98449</td>
<td>4.89408E-08</td>
</tr>
<tr>
<td>14</td>
<td>0.85243</td>
<td>1.14757</td>
<td>0.16308</td>
<td>0.98449</td>
<td>4.15082E-09</td>
</tr>
<tr>
<td>15</td>
<td>0.85243</td>
<td>1.14757</td>
<td>0.16308</td>
<td>0.98449</td>
<td>4.19198E-10</td>
</tr>
<tr>
<td>16</td>
<td>0.85243</td>
<td>1.14757</td>
<td>0.16308</td>
<td>0.98449</td>
<td>1.07133E-11</td>
</tr>
<tr>
<td>17</td>
<td>0.85243</td>
<td>1.14757</td>
<td>0.16308</td>
<td>0.98449</td>
<td>1.55200E-12</td>
</tr>
<tr>
<td>18</td>
<td>0.85243</td>
<td>1.14757</td>
<td>0.16308</td>
<td>0.98449</td>
<td>1.68244E-12</td>
</tr>
<tr>
<td>19</td>
<td>0.85243</td>
<td>1.14757</td>
<td>0.16308</td>
<td>0.98449</td>
<td>2.20498E-12</td>
</tr>
<tr>
<td>20</td>
<td>0.85243</td>
<td>1.14757</td>
<td>0.16308</td>
<td>0.98449</td>
<td>5.12143E-12</td>
</tr>
<tr>
<td>21</td>
<td>0.85243</td>
<td>1.14757</td>
<td>0.16308</td>
<td>0.98449</td>
<td>1.78740E-12</td>
</tr>
<tr>
<td>22</td>
<td>0.85243</td>
<td>1.14757</td>
<td>0.16308</td>
<td>0.98449</td>
<td>3.48963E-12</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>100</td>
<td>0.85243</td>
<td>1.14757</td>
<td>0.16308</td>
<td>0.98449</td>
<td>7.03881E-14</td>
</tr>
<tr>
<td>101</td>
<td>0.85243</td>
<td>1.14757</td>
<td>0.16308</td>
<td>0.98449</td>
<td>6.26170E-14</td>
</tr>
<tr>
<td>Iteration, n</td>
<td>$Q_{5}$ (m$^3$/s)</td>
<td>$Q_{6}$ (m$^3$/s)</td>
<td>$Q_{7}$ (m$^3$/s)</td>
<td>$Q_{8}$ (m$^3$/s)</td>
<td>$|f_n|_b$ (m)</td>
</tr>
<tr>
<td>-------------</td>
<td>----------------</td>
<td>----------------</td>
<td>----------------</td>
<td>----------------</td>
<td>----------------</td>
</tr>
<tr>
<td>0</td>
<td>1.50000</td>
<td>0.25000</td>
<td>0.25000</td>
<td>1.75000</td>
<td>4.89372E-02</td>
</tr>
<tr>
<td>1</td>
<td>1.47331</td>
<td>0.23661</td>
<td>0.29008</td>
<td>1.70992</td>
<td>4.57919E-02</td>
</tr>
<tr>
<td>2</td>
<td>1.07951</td>
<td>0.05028</td>
<td>0.87021</td>
<td>1.12979</td>
<td>4.73030E-03</td>
</tr>
<tr>
<td>3</td>
<td>1.07500</td>
<td>0.05636</td>
<td>0.86864</td>
<td>1.13136</td>
<td>4.44204E-03</td>
</tr>
<tr>
<td>4</td>
<td>1.01262</td>
<td>0.14935</td>
<td>0.83803</td>
<td>1.16197</td>
<td>2.89207E-04</td>
</tr>
<tr>
<td>5</td>
<td>1.01469</td>
<td>0.14933</td>
<td>0.83599</td>
<td>1.16402</td>
<td>1.05549E-04</td>
</tr>
<tr>
<td>6</td>
<td>1.01550</td>
<td>0.14987</td>
<td>0.83463</td>
<td>1.16537</td>
<td>6.67407E-06</td>
</tr>
<tr>
<td>7</td>
<td>1.01550</td>
<td>0.14987</td>
<td>0.83463</td>
<td>1.16537</td>
<td>6.08369E-06</td>
</tr>
<tr>
<td>8</td>
<td>1.01548</td>
<td>0.14991</td>
<td>0.83460</td>
<td>1.16540</td>
<td>3.04764E-06</td>
</tr>
<tr>
<td>9</td>
<td>1.01549</td>
<td>0.14991</td>
<td>0.83460</td>
<td>1.16540</td>
<td>2.41094E-06</td>
</tr>
<tr>
<td>10</td>
<td>1.01550</td>
<td>0.14987</td>
<td>0.83461</td>
<td>1.16539</td>
<td>9.79484E-07</td>
</tr>
<tr>
<td>11</td>
<td>1.01551</td>
<td>0.14987</td>
<td>0.83462</td>
<td>1.16538</td>
<td>6.03826E-07</td>
</tr>
<tr>
<td>12</td>
<td>1.01551</td>
<td>0.14987</td>
<td>0.83462</td>
<td>1.16538</td>
<td>1.59667E-07</td>
</tr>
<tr>
<td>13</td>
<td>1.01551</td>
<td>0.14987</td>
<td>0.83462</td>
<td>1.16538</td>
<td>4.89408E-08</td>
</tr>
<tr>
<td>14</td>
<td>1.01551</td>
<td>0.14987</td>
<td>0.83462</td>
<td>1.16538</td>
<td>4.15082E-09</td>
</tr>
<tr>
<td>15</td>
<td>1.01551</td>
<td>0.14987</td>
<td>0.83462</td>
<td>1.16538</td>
<td>4.19198E-10</td>
</tr>
<tr>
<td>16</td>
<td>1.01551</td>
<td>0.14987</td>
<td>0.83462</td>
<td>1.16538</td>
<td>1.07133E-11</td>
</tr>
<tr>
<td>17</td>
<td>1.01551</td>
<td>0.14987</td>
<td>0.83462</td>
<td>1.16538</td>
<td>1.55200E-12</td>
</tr>
<tr>
<td>18</td>
<td>1.01551</td>
<td>0.14987</td>
<td>0.83462</td>
<td>1.16538</td>
<td>1.68244E-12</td>
</tr>
<tr>
<td>19</td>
<td>1.01551</td>
<td>0.14987</td>
<td>0.83462</td>
<td>1.16538</td>
<td>2.20498E-12</td>
</tr>
<tr>
<td>20</td>
<td>1.01551</td>
<td>0.14987</td>
<td>0.83462</td>
<td>1.16538</td>
<td>5.12143E-12</td>
</tr>
<tr>
<td>21</td>
<td>1.01551</td>
<td>0.14987</td>
<td>0.83462</td>
<td>1.16538</td>
<td>1.78740E-12</td>
</tr>
<tr>
<td>22</td>
<td>1.01551</td>
<td>0.14987</td>
<td>0.83462</td>
<td>1.16538</td>
<td>3.48963E-12</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>100</td>
<td>1.01551</td>
<td>0.14987</td>
<td>0.83462</td>
<td>1.16538</td>
<td>7.03881E-14</td>
</tr>
<tr>
<td>101</td>
<td>1.01551</td>
<td>0.14987</td>
<td>0.83462</td>
<td>1.16538</td>
<td>6.26170E-14</td>
</tr>
</tbody>
</table>

Table 5.10c (cont). Nested Network - Convergence characteristics of the DFP method.
## Broyden-Fletcher-Goldfarb-Shanno

<table>
<thead>
<tr>
<th>Iteration, n</th>
<th>( Q_1 ) (m³/s)</th>
<th>( Q_2 ) (m³/s)</th>
<th>( Q_3 ) (m³/s)</th>
<th>( Q_4 ) (m³/s)</th>
<th>( |f_n| ) (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1.00000</td>
<td>1.00000</td>
<td>0.50000</td>
<td>0.50000</td>
<td>4.89372E-02</td>
</tr>
<tr>
<td>1</td>
<td>0.99131</td>
<td>1.00869</td>
<td>0.48199</td>
<td>0.52669</td>
<td>4.57919E-02</td>
</tr>
<tr>
<td>2</td>
<td>0.86220</td>
<td>1.13780</td>
<td>0.21720</td>
<td>0.92060</td>
<td>4.72237E-03</td>
</tr>
<tr>
<td>3</td>
<td>0.86141</td>
<td>1.13860</td>
<td>0.21342</td>
<td>0.92517</td>
<td>4.42908E-03</td>
</tr>
<tr>
<td>4</td>
<td>0.85202</td>
<td>1.14798</td>
<td>0.16241</td>
<td>0.98557</td>
<td>9.64361E-05</td>
</tr>
<tr>
<td>5</td>
<td>0.85221</td>
<td>1.14779</td>
<td>0.16338</td>
<td>0.98441</td>
<td>1.99133E-05</td>
</tr>
<tr>
<td>6</td>
<td>0.85223</td>
<td>1.14777</td>
<td>0.16334</td>
<td>0.98444</td>
<td>1.81424E-05</td>
</tr>
<tr>
<td>7</td>
<td>0.85237</td>
<td>1.14763</td>
<td>0.16311</td>
<td>0.98452</td>
<td>5.80093E-06</td>
</tr>
<tr>
<td>8</td>
<td>0.85242</td>
<td>1.14758</td>
<td>0.16307</td>
<td>0.98451</td>
<td>1.74008E-06</td>
</tr>
<tr>
<td>9</td>
<td>0.85243</td>
<td>1.14757</td>
<td>0.16308</td>
<td>0.98449</td>
<td>1.76332E-07</td>
</tr>
<tr>
<td>10</td>
<td>0.85243</td>
<td>1.14757</td>
<td>0.16308</td>
<td>0.98449</td>
<td>1.54365E-08</td>
</tr>
<tr>
<td>11</td>
<td>0.85243</td>
<td>1.14757</td>
<td>0.16308</td>
<td>0.98449</td>
<td>3.78054E-10</td>
</tr>
<tr>
<td>12</td>
<td>0.85243</td>
<td>1.14757</td>
<td>0.16308</td>
<td>0.98449</td>
<td>6.43587E-12</td>
</tr>
<tr>
<td>13</td>
<td>0.85243</td>
<td>1.14757</td>
<td>0.16308</td>
<td>0.98449</td>
<td>2.05183E-12</td>
</tr>
<tr>
<td>14</td>
<td>0.85243</td>
<td>1.14757</td>
<td>0.16308</td>
<td>0.98449</td>
<td>2.19172E-12</td>
</tr>
<tr>
<td>15</td>
<td>0.85243</td>
<td>1.14757</td>
<td>0.16308</td>
<td>0.98449</td>
<td>1.76784E-12</td>
</tr>
<tr>
<td>16</td>
<td>0.85243</td>
<td>1.14757</td>
<td>0.16308</td>
<td>0.98449</td>
<td>1.55723E-12</td>
</tr>
<tr>
<td>17</td>
<td>0.85243</td>
<td>1.14757</td>
<td>0.16308</td>
<td>0.98449</td>
<td>7.72026E-12</td>
</tr>
<tr>
<td>18</td>
<td>0.85243</td>
<td>1.14757</td>
<td>0.16308</td>
<td>0.98449</td>
<td>2.22358E-12</td>
</tr>
<tr>
<td>19</td>
<td>0.85243</td>
<td>1.14757</td>
<td>0.16308</td>
<td>0.98449</td>
<td>1.57048E-12</td>
</tr>
<tr>
<td>20</td>
<td>0.85243</td>
<td>1.14757</td>
<td>0.16308</td>
<td>0.98449</td>
<td>7.76513E-12</td>
</tr>
<tr>
<td>21</td>
<td>0.85243</td>
<td>1.14757</td>
<td>0.16308</td>
<td>0.98449</td>
<td>1.61116E-12</td>
</tr>
<tr>
<td>22</td>
<td>0.85243</td>
<td>1.14757</td>
<td>0.16308</td>
<td>0.98449</td>
<td>1.56343E-12</td>
</tr>
<tr>
<td>29</td>
<td>0.85243</td>
<td>1.14757</td>
<td>0.16308</td>
<td>0.98449</td>
<td>3.14019E-16</td>
</tr>
<tr>
<td>30</td>
<td>0.85243</td>
<td>1.14757</td>
<td>0.16308</td>
<td>0.98449</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 5.10d. Nested Network - Convergence characteristics of the BFGS method.
### Broyden-Fletcher-Goldfarb-Shanno

<table>
<thead>
<tr>
<th>Iteration, ( n )</th>
<th>( Q_3 ) (m³/s)</th>
<th>( Q_4 ) (m³/s)</th>
<th>( Q_7 ) (m³/s)</th>
<th>( Q_9 ) (m³/s)</th>
<th>( | f_n | ) (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1.50000</td>
<td>0.25000</td>
<td>0.25000</td>
<td>1.75000</td>
<td>4.89372E-02</td>
</tr>
<tr>
<td>1</td>
<td>1.47331</td>
<td>0.23661</td>
<td>0.29008</td>
<td>1.70992</td>
<td>4.57919E-02</td>
</tr>
<tr>
<td>2</td>
<td>1.07940</td>
<td>0.05045</td>
<td>0.87015</td>
<td>1.12985</td>
<td>4.72237E-03</td>
</tr>
<tr>
<td>3</td>
<td>1.07483</td>
<td>0.05663</td>
<td>0.86854</td>
<td>1.13146</td>
<td>4.42908E-03</td>
</tr>
<tr>
<td>4</td>
<td>1.01443</td>
<td>0.15178</td>
<td>0.83379</td>
<td>1.16621</td>
<td>9.64361E-05</td>
</tr>
<tr>
<td>5</td>
<td>1.01559</td>
<td>0.14980</td>
<td>0.83461</td>
<td>1.16539</td>
<td>1.99133E-05</td>
</tr>
<tr>
<td>6</td>
<td>1.01556</td>
<td>0.14984</td>
<td>0.83460</td>
<td>1.16540</td>
<td>1.81424E-05</td>
</tr>
<tr>
<td>7</td>
<td>1.01548</td>
<td>0.14994</td>
<td>0.83458</td>
<td>1.16542</td>
<td>5.80093E-06</td>
</tr>
<tr>
<td>8</td>
<td>1.01549</td>
<td>0.14990</td>
<td>0.83461</td>
<td>1.16539</td>
<td>1.74008E-06</td>
</tr>
<tr>
<td>9</td>
<td>1.01551</td>
<td>0.14987</td>
<td>0.83462</td>
<td>1.16538</td>
<td>1.76332E-07</td>
</tr>
<tr>
<td>10</td>
<td>1.01551</td>
<td>0.14987</td>
<td>0.83462</td>
<td>1.16538</td>
<td>1.54365E-08</td>
</tr>
<tr>
<td>11</td>
<td>1.01551</td>
<td>0.14987</td>
<td>0.83462</td>
<td>1.16538</td>
<td>3.78054E-10</td>
</tr>
<tr>
<td>12</td>
<td>1.01551</td>
<td>0.14987</td>
<td>0.83462</td>
<td>1.16538</td>
<td>6.43587E-12</td>
</tr>
<tr>
<td>13</td>
<td>1.01551</td>
<td>0.14987</td>
<td>0.83462</td>
<td>1.16538</td>
<td>2.05183E-12</td>
</tr>
<tr>
<td>14</td>
<td>1.01551</td>
<td>0.14987</td>
<td>0.83462</td>
<td>1.16538</td>
<td>2.19172E-12</td>
</tr>
<tr>
<td>15</td>
<td>1.01551</td>
<td>0.14987</td>
<td>0.83462</td>
<td>1.16538</td>
<td>1.76784E-12</td>
</tr>
<tr>
<td>16</td>
<td>1.01551</td>
<td>0.14987</td>
<td>0.83462</td>
<td>1.16538</td>
<td>1.55723E-12</td>
</tr>
<tr>
<td>17</td>
<td>1.01551</td>
<td>0.14987</td>
<td>0.83462</td>
<td>1.16538</td>
<td>7.72026E-12</td>
</tr>
<tr>
<td>18</td>
<td>1.01551</td>
<td>0.14987</td>
<td>0.83462</td>
<td>1.16538</td>
<td>2.22358E-12</td>
</tr>
<tr>
<td>19</td>
<td>1.01551</td>
<td>0.14987</td>
<td>0.83462</td>
<td>1.16538</td>
<td>1.57048E-12</td>
</tr>
<tr>
<td>20</td>
<td>1.01551</td>
<td>0.14987</td>
<td>0.83462</td>
<td>1.16538</td>
<td>7.76513E-12</td>
</tr>
<tr>
<td>21</td>
<td>1.01551</td>
<td>0.14987</td>
<td>0.83462</td>
<td>1.16538</td>
<td>1.61116E-12</td>
</tr>
<tr>
<td>22</td>
<td>1.01551</td>
<td>0.14987</td>
<td>0.83462</td>
<td>1.16538</td>
<td>1.56343E-12</td>
</tr>
<tr>
<td>23</td>
<td>:</td>
<td>:</td>
<td>:</td>
<td>:</td>
<td>:</td>
</tr>
<tr>
<td>24</td>
<td>1.01551</td>
<td>0.14987</td>
<td>0.83462</td>
<td>1.16538</td>
<td>3.14019E-16</td>
</tr>
<tr>
<td>25</td>
<td>1.01551</td>
<td>0.14987</td>
<td>0.83462</td>
<td>1.16538</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 5.10d (cont). Nested Network - Convergence characteristics of the BFGS method.
## Applications and Testing of Methods of Solution of Networks

### Table 5.11a. Nested Network - Overall performance of various methods, $H_0 = I$.

| Methods | $n$ | $N_f$ | $|q_0 - \hat{q}|$ | $|q_n - \hat{q}|$ | $\eta_5$ |
|---------|-----|-------|----------------|----------------|--------|
| NR      | 3   | 12    | 1.14896E+00    | 1.00000E-07    | 1.355  |
| BR      | 9   | 9     | 1.14896E+00    | 3.75256E-04    | 0.892  |
| DFP     | 7   | 7     | 1.14896E+00    | 1.30287E-04    | 1.298  |
| BFGS    | 8   | 8     | 1.14896E+00    | 4.92371E-05    | 1.257  |

### Table 5.11b. Nested Network - Overall performance of various methods, $H_0 = J_0^{-1}$.

| Methods | $n$ | $N_f$ | $|q_0 - \hat{q}|$ | $|q_n - \hat{q}|$ | $\eta_5$ |
|---------|-----|-------|----------------|----------------|--------|
| NR      | 3   | 12    | 1.14896E+00    | 1.00000E-07    | 1.355  |
| BR      | 5   | 5     | 1.14896E+00    | 1.07238E-06    | 2.777  |
| DFP     | 5   | 5     | 1.14896E+00    | 3.74166E-07    | 2.987  |
| BFGS    | 5   | 5     | 1.14896E+00    | 2.82843E-07    | 3.043  |
Applications and Testing of Methods of Solution of Networks

<table>
<thead>
<tr>
<th>Strings</th>
<th>Initial Energy Head, $H_i$ (m)</th>
<th>Initial Water Depth, $y_i$ (m)</th>
<th>Initial Discharge, $Q_i$ (m$^3$/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Upstream</td>
<td>Downstream</td>
<td>Upstream</td>
</tr>
<tr>
<td>Inflow</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Channel -1</td>
<td>1.00094</td>
<td>1.00000</td>
<td>0.98532</td>
</tr>
<tr>
<td>Channel -2</td>
<td>1.00029</td>
<td>1.00000</td>
<td>0.99204</td>
</tr>
<tr>
<td>Channel -3</td>
<td>1.00013</td>
<td>1.00000</td>
<td>0.99441</td>
</tr>
<tr>
<td>Channel -4</td>
<td>1.00007</td>
<td>1.00000</td>
<td>0.99551</td>
</tr>
<tr>
<td>Outfall</td>
<td>1.00000</td>
<td>1.00000</td>
<td></td>
</tr>
</tbody>
</table>

Table 5.12a. Parallel Network - Initial flow distribution, (5 metres).

<table>
<thead>
<tr>
<th>Strings</th>
<th>Final Energy Head, $H_i$ (m)</th>
<th>Final Water Depth, $y_i$ (m)</th>
<th>Final Discharge, $Q_i$ (m$^3$/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Upstream</td>
<td>Downstream</td>
<td>Upstream</td>
</tr>
<tr>
<td>Inflow</td>
<td>1.00018</td>
<td>1.00018</td>
<td></td>
</tr>
<tr>
<td>Channel -1</td>
<td>1.00018</td>
<td>1.00000</td>
<td>0.99506</td>
</tr>
<tr>
<td>Channel -2</td>
<td>1.00018</td>
<td>1.00000</td>
<td>0.99404</td>
</tr>
<tr>
<td>Channel -3</td>
<td>1.00018</td>
<td>1.00000</td>
<td>0.99321</td>
</tr>
<tr>
<td>Channel -4</td>
<td>1.00018</td>
<td>1.00000</td>
<td>0.99255</td>
</tr>
<tr>
<td>Outfall</td>
<td>1.00000</td>
<td>1.00000</td>
<td></td>
</tr>
</tbody>
</table>

Table 5.12b. Parallel Network - Final flow distribution, (5 metres).

| Methods | $n$ | $N_r$ | $|q_0 - q|$ | $|q_n - q|$ | $\eta_\delta$ |
|---------|-----|-------|-------------|-------------|----------------|
| NR      | 4   | 16    | 4.19657E-01 | 6.54599E-08 | 0.980          |
| BR      | -   | -     | -           | -           | -              |
| DFP     | -   | -     | -           | -           | -              |
| BFGS    | 25  | 25    | 4.19657E-01 | 8.16299E-06 | 0.434          |

Table 5.13. Parallel Network - Overall performance of various methods, $H_0 = 1$. 

185
### Applications and Testing of Methods of Solution of Networks

<table>
<thead>
<tr>
<th>Strings</th>
<th>Initial Energy Head, $H$, (m)</th>
<th>Initial Water Depth, $y$, (m)</th>
<th>Final Discharge, $Q$, (m$^3$/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Upstream</td>
<td>Downstream</td>
<td>Upstream</td>
</tr>
<tr>
<td>Inflow</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Channel-1</td>
<td>1.27650</td>
<td>1.00000</td>
<td>0.75409</td>
</tr>
<tr>
<td>Channel-2</td>
<td>1.12046</td>
<td>1.00000</td>
<td>0.60499</td>
</tr>
<tr>
<td>Channel-3</td>
<td>1.06371</td>
<td>1.00000</td>
<td>0.55330</td>
</tr>
<tr>
<td>Channel-4</td>
<td>1.03831</td>
<td>1.00000</td>
<td>0.53108</td>
</tr>
<tr>
<td>Outfall</td>
<td>1.00000</td>
<td>1.00000</td>
<td></td>
</tr>
</tbody>
</table>

**Table 5.14a.** Parallel Network - Initial flow distribution, (1000 metres).

<table>
<thead>
<tr>
<th>Strings</th>
<th>Final Energy Head, $H$, (m)</th>
<th>Final Water Depth, $y$, (m)</th>
<th>Final Discharge, $Q$, (m$^3$/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Upstream</td>
<td>Downstream</td>
<td>Upstream</td>
</tr>
<tr>
<td>Inflow</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Channel-1</td>
<td>1.08434</td>
<td>1.08434</td>
<td>0.57590</td>
</tr>
<tr>
<td>Channel-2</td>
<td>1.08434</td>
<td>1.00000</td>
<td>0.57304</td>
</tr>
<tr>
<td>Channel-3</td>
<td>1.08434</td>
<td>1.00000</td>
<td>0.57089</td>
</tr>
<tr>
<td>Channel-4</td>
<td>1.08434</td>
<td>1.00000</td>
<td>0.56925</td>
</tr>
<tr>
<td>Outflow</td>
<td>1.00000</td>
<td>1.00000</td>
<td></td>
</tr>
</tbody>
</table>

**Table 5.14b.** Parallel Network - Final flow distribution, (1000 metres).

<table>
<thead>
<tr>
<th>Methods</th>
<th>$n$</th>
<th>$N_f$</th>
<th>$|\mathbf{q}_0 - \hat{\mathbf{q}}|$</th>
<th>$|\mathbf{q}_s - \hat{\mathbf{q}}|$</th>
<th>$\eta_8$</th>
</tr>
</thead>
<tbody>
<tr>
<td>NR</td>
<td>3</td>
<td>12</td>
<td>3.07498E-01</td>
<td>3.46410E-09</td>
<td>1.525</td>
</tr>
<tr>
<td>BR</td>
<td>8</td>
<td>8</td>
<td>3.07498E-01</td>
<td>2.09673E-05</td>
<td>1.199</td>
</tr>
<tr>
<td>DFP</td>
<td>11</td>
<td>11</td>
<td>3.07498E-01</td>
<td>8.14259E-07</td>
<td>1.167</td>
</tr>
<tr>
<td>BFGS</td>
<td>8</td>
<td>8</td>
<td>3.07498E-01</td>
<td>3.37832E-06</td>
<td>1.427</td>
</tr>
</tbody>
</table>

**Table 5.15.** Parallel Network - Overall performance of various methods, $H_0 = I$. 

---

186
5.3.3 Initial Conditions

A very important practical limitation of the NR method is that when the initial set of approximations is not sufficiently close to the final solution, the Jacobian matrix may not be fully ranked. Consequently, the NR method fails to converge. The closer they are to the solution, the faster the methods converge.

To investigate the effects of initial approximation on rate of convergence, two computer runs were conducted using the networks (parallel and nested) identical to those described in previous Tables 5.4 and 5.8. Tables 5.16 and 5.18 show the assumed initial flow distribution of the networks. These sets of the data were deliberately chosen to examine with some extreme cases.

Consider the parallel network first. The methods converged to the solution identical to that shown in Table 5.5b. Tables 5.17a and 5.17b show the performance of various methods with the $H_0 = I$ and $H_0 = J_0^{-1}$ respectively. The quasi-Newton methods were converged slowly in the beginning (as in previous examples) and then gradually increased in speed as the iteration moved towards the solution. The methods required more iterations to reach to $\chi_{n+1}$ except for the BR and DFP methods shown in Table 5.17a (also see Tables 5.7a). The reason was that the methods required fewer iterations to attain a good approximation to $H_n = J_n^{-1}$. As expected, the performance of the NR method deteriorated from $\eta_0 = 0.953$ to $\eta_0 = 0.727$; whereas the quasi-Newton methods showed improvement. However, it is also found that the performance of the quasi-Newton methods has deteriorated when $H_0 = J_0^{-1}$ was used. This implies that the $H_0 = J_0^{-1}$ was a better choice than the $H_0 = I$ when the initial approximation far from the final solution. Indeed, the use of $H_0 = J_0^{-1}$ has defeated the object of using the quasi-Newton methods, but it has shown the significance of $H_0$ on the performance of the methods.

Tables 5.19a and 5.19b show the results of the second run on the nested network. The distribution of flow obtained was identical to that shown in Table 5.9b. The points observed above also apply here.
i) The NR method converges considerably faster than the quasi-Newton methods. It only requires 4 iterations to satisfy the criteria $\chi_{ij+1}$.

ii) The NR method oscillates towards the final solution, owing to the instability of $J_n^{-1}$ evaluated using the technique of finite differencing.

iii) The slow convergence rate occurs in the quasi-Newton methods is caused by $H_o = I$ is not a good approximation to $J_o^{-1}$. Once $H_n$ has accumulated sufficient information to approximate $J_n^{-1}$ as the iteration proceeds, a faster rate of convergence is obtained.

iv) The BFGS method appears to be insensitive to the sensitivity of the network, but the BR and DFP methods are highly sensitive to the sensitivity of the network.

In addition to the foregoing observations, the performance of all methods deteriorated. Interestingly, the methods performed better with the $H_o = I$ rather than $H_o = J_o^{-1}$ implying that the $H_o = I$ was the better approximation than when the approximation was far from the final solution. It would seem that the $J_o^{-1}$ may not necessarily always provide better predictions than the $I$.

The following general observation can be made from these results.

i) The overall performance NR method deteriorated as the approximation was farther away from the final solution.

ii) The quasi-Newton methods performed less well with $H_o = J_o^{-1}$ when the approximation was farther away from the final solution. The performance of the quasi-Newton methods were worst with $H_o = J_o^{-1}$ in the nested case.

iii) The BR performed better with $H_o = I$ when the approximation farther away from the final solution.

iv) The DFP was very sensitive to the network sensitivity (discharge and head-loss relationship) and the choice of $H_o$.

v) The BFGS, in general, performed better than the other quasi-Newton methods; it was comparatively stable with either $H_o = I$ or $H_o = J_o^{-1}$.

vi) With a 'low sensitivity' network, the choice of $H_o$ for the quasi-Newton methods would be $J_o^{-1}$. With a 'high sensitivity' network the choice of $H_o$ would be $I$. 

188
APPLICATIONS AND TESTING OF METHODS OF SOLUTION OF NETWORKS

Table 5.16. Parallel Network - Initial flow distribution.

<table>
<thead>
<tr>
<th>Strings</th>
<th>Initial Energy Head, $H$, (m)</th>
<th>Initial Water Depth, $y$, (m)</th>
<th>Initial Discharge, $Q$, (m$^3$/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Upstream</td>
<td>Downstream</td>
<td>Upstream</td>
</tr>
<tr>
<td>Inflow</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Channel-1</td>
<td>1.27388</td>
<td>1.11118</td>
<td>1.07202</td>
</tr>
<tr>
<td>Channel-2</td>
<td>1.00000</td>
<td>1.00000</td>
<td>0.97500</td>
</tr>
<tr>
<td>Channel-3</td>
<td>1.00000</td>
<td>1.00000</td>
<td>0.97500</td>
</tr>
<tr>
<td>Channel-4</td>
<td>1.00000</td>
<td>1.00000</td>
<td>0.97500</td>
</tr>
<tr>
<td>Outfall</td>
<td>1.00000</td>
<td>1.00000</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 5.17a. Parallel Network - Overall performance of various methods, $H_0 = I$.

<table>
<thead>
<tr>
<th>Methods</th>
<th>$n$</th>
<th>$N_f$</th>
<th>$|q_0 - \hat{q}|$</th>
<th>$|q_n - \hat{q}|$</th>
<th>$\eta_5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>NR</td>
<td>7</td>
<td>28</td>
<td>2.0641E+00</td>
<td>2.95098E-09</td>
<td>0.727</td>
</tr>
<tr>
<td>BR</td>
<td>33</td>
<td>33</td>
<td>2.0641E+00</td>
<td>5.26213E-06</td>
<td>0.390</td>
</tr>
<tr>
<td>DFP</td>
<td>164</td>
<td>164</td>
<td>2.0641E+00</td>
<td>2.13747E-04</td>
<td>0.056</td>
</tr>
<tr>
<td>BFGS</td>
<td>21</td>
<td>21</td>
<td>2.0641E+00</td>
<td>2.73496E-06</td>
<td>0.644</td>
</tr>
</tbody>
</table>

Table 5.17b. Parallel Network - Overall performance of various methods, $H_0 = J_0^{-1}$.

<table>
<thead>
<tr>
<th>Methods</th>
<th>$n$</th>
<th>$N_f$</th>
<th>$|q_0 - \hat{q}|$</th>
<th>$|q_n - \hat{q}|$</th>
<th>$\eta_5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>NR</td>
<td>7</td>
<td>28</td>
<td>2.0641E+00</td>
<td>2.95098E-09</td>
<td>0.727</td>
</tr>
<tr>
<td>BR</td>
<td>19</td>
<td>19</td>
<td>2.0641E+00</td>
<td>1.02122E-05</td>
<td>0.643</td>
</tr>
<tr>
<td>DFP</td>
<td>12</td>
<td>12</td>
<td>2.0641E+00</td>
<td>8.17007E-06</td>
<td>1.037</td>
</tr>
<tr>
<td>BFGS</td>
<td>13</td>
<td>13</td>
<td>2.0641E+00</td>
<td>6.10655E-06</td>
<td>0.979</td>
</tr>
</tbody>
</table>
### Applications and Testing of Methods of Solution of Networks

<table>
<thead>
<tr>
<th>Strings</th>
<th>Initial Energy Head, $H$, (m)</th>
<th>Initial Water Depth, $y$, (m)</th>
<th>Initial Discharge, $Q$, ($m^3/s$)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Upstream</td>
<td>Downstream</td>
<td>Upstream</td>
</tr>
<tr>
<td>Inflow</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Channel - 1</td>
<td>1.05965</td>
<td>1.05965</td>
<td>0.95965</td>
</tr>
<tr>
<td>Channel - 2</td>
<td>1.13168</td>
<td>1.07126</td>
<td>0.92614</td>
</tr>
<tr>
<td>Channel - 3</td>
<td>1.05965</td>
<td>1.05965</td>
<td>0.98465</td>
</tr>
<tr>
<td>Channel - 4</td>
<td>1.07126</td>
<td>1.00000</td>
<td>0.87930</td>
</tr>
<tr>
<td>Channel - 5</td>
<td>1.05965</td>
<td>1.05965</td>
<td>1.00965</td>
</tr>
<tr>
<td>Channel - 6</td>
<td>1.11277</td>
<td>1.05965</td>
<td>0.96595</td>
</tr>
<tr>
<td>Channel - 7</td>
<td>1.00000</td>
<td>1.00000</td>
<td>0.95000</td>
</tr>
<tr>
<td>Channel - 8</td>
<td>1.05965</td>
<td>1.00000</td>
<td>0.92999</td>
</tr>
<tr>
<td>Outfall</td>
<td>1.00000</td>
<td>1.00000</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 5.18. Nested Network - Initial flow distribution.

<table>
<thead>
<tr>
<th>Methods</th>
<th>$n$</th>
<th>$N_f$</th>
<th>$|q_0 - \hat{q}|$</th>
<th>$|q_s - \hat{q}|$</th>
<th>$\eta_s$</th>
</tr>
</thead>
<tbody>
<tr>
<td>NR</td>
<td>7</td>
<td>28</td>
<td>2.88657E+00</td>
<td>8.54810E-10</td>
<td>0.784</td>
</tr>
<tr>
<td>BR</td>
<td>15</td>
<td>15</td>
<td>2.88657E+00</td>
<td>1.62481E-06</td>
<td>0.959</td>
</tr>
<tr>
<td>DFP</td>
<td>35</td>
<td>35</td>
<td>2.88657E+00</td>
<td>1.33515E-04</td>
<td>0.285</td>
</tr>
<tr>
<td>BFGS</td>
<td>16</td>
<td>16</td>
<td>2.88657E+00</td>
<td>1.50333E-06</td>
<td>0.926</td>
</tr>
</tbody>
</table>

Table 5.19a. Nested Network - Overall performance of various methods, $H_0 = I$.

<table>
<thead>
<tr>
<th>Methods</th>
<th>$n$</th>
<th>$N_f$</th>
<th>$|q_0 - \hat{q}|$</th>
<th>$|q_s - \hat{q}|$</th>
<th>$\eta_s$</th>
</tr>
</thead>
<tbody>
<tr>
<td>NR</td>
<td>7</td>
<td>28</td>
<td>2.88657E+00</td>
<td>8.54810E-10</td>
<td>0.784</td>
</tr>
<tr>
<td>BR</td>
<td>21</td>
<td>21</td>
<td>2.88657E+00</td>
<td>8.20488E-06</td>
<td>0.608</td>
</tr>
<tr>
<td>DFP</td>
<td>28</td>
<td>28</td>
<td>2.88657E+00</td>
<td>3.48642E-05</td>
<td>0.404</td>
</tr>
<tr>
<td>BFGS</td>
<td>19</td>
<td>19</td>
<td>2.88657E+00</td>
<td>8.36660E-06</td>
<td>0.671</td>
</tr>
</tbody>
</table>

Table 5.19b. Nested Network - Overall performance of various methods, $H_0 = J_0^{-1}$.  

190
5.3.4 The Choice of $h$

Since the $\tilde{J}_n$ is required to approximate as closely as possible to the $J_n$, it implies that the step size $h$ for differentiation (see Section 4.3.2) should be chosen as small as possible, whilst keeping the round-off as small as possible too. The problem in applying the numerical methods is to choose an appropriate size of $h$ for differentiation. It must be noted that the smallest size (lower bound) of $h$ is limited by machine precision. Although the formulae described in Section 4.3.2. may be used to estimate $h$, it is important to explore the significance of $h$ on the NR procedure through numerical experiments.

For simplicity, the parallel network shown in the Figure 5.3 previous section was used; the physical characteristics of the network are shown in Table 5.4. With the exception of channel length which was set to 5 (m) in order to investigate the effects of $h$ on a low head-loss system. Five computer runs were conducted separately using various sizes of $h \in \{10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}\}$, and the results were obtained and are shown in Table 5.20.

The convergence characteristics obtained for the five runs varied with different sizes of $h$. The smaller the size of $h$, the quicker the NR method converged to, say $\|\epsilon_n\| < 10^{-10}$. An examination of the figures shows that there exists a well defined trend converging asymptotically towards the smaller sizes of $h$; this is highlighted in Table 5.20. It implies that the optimum size of $h$ is reached at about $h_{\text{optimal}} = 10^{-4}$, meaning that there would be little or no improvement on the rate of convergence for this example even though smaller sizes of $h < 10^{-4}$ were chosen to approximate $\tilde{J}_n$. Besides, the effects of $h$ on the rate of convergence gradually diminished after $\|\epsilon_n\| < 10^{-10}$. This may be explained by the following reasons:

i) When $h$ is chosen too small, the round-off errors will dominate the total errors incurred in the computations.

ii) When $h$ is chosen too large, the $\delta_n$ predicted by $\tilde{J}_n$ will be increasingly over-estimated as the approximations approaching to a solution.

There are believed to be the reasons why the NR method oscillates infinitely near a solution for the example chosen.
### Applications and Testing of Methods of Solution of Networks

#### Table 5.20. Parallel Network - Differencing interval, $h$, for partial derivatives.

<table>
<thead>
<tr>
<th>Iteration, $n$</th>
<th>$h = 0.10000$</th>
<th>$h = 0.01000$</th>
<th>$h = 0.00100$</th>
<th>$h = 0.00010$</th>
<th>$h = 0.00001$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.65649E-04</td>
<td>1.57594E-04</td>
<td>1.57763E-04</td>
<td>1.57790E-04</td>
<td>1.57802E-04</td>
</tr>
<tr>
<td>2</td>
<td>1.25716E-05</td>
<td>1.48505E-05</td>
<td>1.60100E-05</td>
<td>1.61275E-05</td>
<td>1.61352E-05</td>
</tr>
<tr>
<td>3</td>
<td>1.76229E-06</td>
<td>1.07445E-07</td>
<td>2.26244E-07</td>
<td>2.44508E-07</td>
<td>2.48266E-07</td>
</tr>
<tr>
<td>4</td>
<td>2.35222E-07</td>
<td>1.21557E-09</td>
<td>2.34236E-10</td>
<td>4.91321E-11</td>
<td>8.77641E-11</td>
</tr>
<tr>
<td>5</td>
<td>2.45037E-08</td>
<td>1.28732E-11</td>
<td>7.25898E-13</td>
<td>3.39767E-13</td>
<td>2.54256E-12</td>
</tr>
<tr>
<td>6</td>
<td>2.74111E-09</td>
<td>9.42951E-13</td>
<td>1.52214E-12</td>
<td>7.83510E-13</td>
<td>3.50815E-12</td>
</tr>
<tr>
<td>7</td>
<td>4.53514E-10</td>
<td>1.02686E-12</td>
<td>9.77140E-13</td>
<td>1.23751E-12</td>
<td>2.05030E-12</td>
</tr>
<tr>
<td>8</td>
<td>6.89515E-11</td>
<td>1.28341E-12</td>
<td>4.34841E-13</td>
<td>2.15690E-12</td>
<td>8.84692E-13</td>
</tr>
<tr>
<td>9</td>
<td>8.17337E-12</td>
<td>1.69156E-12</td>
<td>1.26304E-12</td>
<td>2.25930E-12</td>
<td>7.38839E-13</td>
</tr>
<tr>
<td>10</td>
<td>1.36651E-12</td>
<td>9.11275E-13</td>
<td>1.15794E-12</td>
<td>3.01167E-12</td>
<td>1.20424E-12</td>
</tr>
<tr>
<td>11</td>
<td>1.64301E-13</td>
<td>1.45974E-12</td>
<td>1.25856E-12</td>
<td>6.32323E-13</td>
<td>1.05248E-12</td>
</tr>
<tr>
<td>12</td>
<td>1.20977E-12</td>
<td>2.43370E-12</td>
<td>1.31281E-12</td>
<td>9.14929E-13</td>
<td>1.40037E-12</td>
</tr>
<tr>
<td>13</td>
<td>1.36736E-12</td>
<td>1.47689E-12</td>
<td>1.88701E-12</td>
<td>2.74913E-12</td>
<td>9.19750E-13</td>
</tr>
<tr>
<td>14</td>
<td>1.67693E-12</td>
<td>1.97970E-12</td>
<td>9.04999E-13</td>
<td>1.33528E-12</td>
<td>9.26290E-13</td>
</tr>
<tr>
<td>15</td>
<td>4.0946E-13</td>
<td>2.87877E-12</td>
<td>5.08535E-13</td>
<td>1.37088E-12</td>
<td>1.99015E-12</td>
</tr>
<tr>
<td>16</td>
<td>2.62015E-12</td>
<td>1.40074E-12</td>
<td>3.11977E-13</td>
<td>2.04588E-12</td>
<td>1.55022E-12</td>
</tr>
<tr>
<td>17</td>
<td>1.76023E-12</td>
<td>1.66926E-12</td>
<td>3.71068E-12</td>
<td>2.20561E-12</td>
<td>1.50234E-12</td>
</tr>
<tr>
<td>18</td>
<td>1.87490E-12</td>
<td>1.60471E-12</td>
<td>1.05693E-12</td>
<td>1.15303E-12</td>
<td>2.63677E-12</td>
</tr>
<tr>
<td>19</td>
<td>1.75909E-12</td>
<td>1.07904E-12</td>
<td>9.37975E-13</td>
<td>1.53821E-12</td>
<td>3.27997E-12</td>
</tr>
<tr>
<td>20</td>
<td>7.01373E-13</td>
<td>5.92255E-13</td>
<td>1.86989E-12</td>
<td>6.31991E-13</td>
<td>1.79156E-12</td>
</tr>
<tr>
<td>22</td>
<td>6.39145E-13</td>
<td>9.95321E-13</td>
<td>1.10280E-12</td>
<td>2.08903E-12</td>
<td>2.39659E-12</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>99</td>
<td>2.54585E-12</td>
<td>3.84884E-13</td>
<td>1.19429E-12</td>
<td>1.53782E-12</td>
<td>1.84868E-12</td>
</tr>
<tr>
<td>100</td>
<td>1.13143E-12</td>
<td>1.58132E-12</td>
<td>1.32384E-12</td>
<td>1.09708E-12</td>
<td>1.21573E-13</td>
</tr>
</tbody>
</table>
Suppose the given machine has precision accurate to 15 digits, and the computations are expected to have the same degree of accuracy, then the size of $h$ may be found by (4.39), and hence

$$h_{\text{optimum}} = \sqrt{4 \times 10^{-15}}$$

or approximating it to $h \approx 10^{-8}$. This figure does not agree with the $h_{\text{optimum}} = 10^{-4}$ found in the example. As aforementioned, equation (4.39) is unlikely to provide a realistic prediction of $h$, due to the fact that the derivation of (4.39) did not take into account the form of the function involved as well as the complexity of the calculation, such as how many additions and multiplications are involved to evaluate the function. These calculations will contribute round-off error in the solution so that a good estimate of $h$ is difficult to obtain. In practice, it is unlikely that a value $h < 10^{-8}$ will be chosen even when working in double precision. It is because the least significant decimal places are easily lost, especially when the numbers are raised to a power. Instead of using $h \approx 10^{-8}$ for differentiation, it would be more appropriate to regard $h \approx 10^{-8}$ as the lower bound.

With a 'low sensitivity' network, such as this given example, the choice of $h$ may be less restrictive because the member function $f_i$ of $f_*$ is insensitive to the variations of $q_j$. Conversely, with a 'high sensitivity' network, the $f_i$ may be very sensitive to the variations of $q_j$ so that the choice of $h \rightarrow 10^{-8}$ is usually required with increasing network sensitivity. It implies that the upper bound of $h$ may be estimated from the 'low sensitivity' network. Hence, the size of $h$ may be bounded by $10^{-2} < h < 10^{-8}$.

### 5.3.5 The Choice of Method

As shown in previous sections, the methods performed differently. As far as rate of convergence (number of iterations) is concerned, the NR method converged fastest for all test cases. However, the method is offset by the need to evaluate partial derivatives. Although the derivatives can be approximated by finite differencing, this increases the computational effort in proportion to $L^3$, and it has been shown that the choice of $h$ is critical to the rate of convergence. It appears to be impractical to finalise a particular value of $h$ because of the dependency on the specific problem involved. However, it is recommended that the size of $h$ be bounded by $10^{-2} < h < 10^{-8}$.
With the $H_0 = I$, and with the initial set of approximation close to the final solution, the NR method converged to $\chi_{j+1}$ faster than the BR, DFP and BFGS methods. The difference between BR, DFP and BFGS methods was highly dependent on the sensitivity of the network. If the network was relatively insensitive, then the BR method and especially the DFP method were very inefficient. If the network was relatively sensitive, then the difference between the three was small. In fact, the difference between the DFP method and the BFGS method was also inseparable.

With the initial set of approximation far away from the final solution, the performance of the NR method has deteriorated from $\eta_5 = 0.953$ to $\eta_5 = 0.727$ (parallel network), but still it out-performed the quasi-Newton methods. With the 'low sensitivity' network the NR method performed superior to the BR and BFGS methods; with the 'high sensitivity' network, the NR method performed less well. In both case, the DFP method performed poorly. Besides, the quasi-Newton methods performed less well with $H_0 = J_0^{-1}$ when the approximation was farther away from the final solution. In fact, the performance of the quasi-Newton methods were worst with $H_0 = J_0^{-1}$ in the nested case. Therefore, the choice of $H_0$ would be $I$ rather than $J_0^{-1}$.

With the $H_0 = J_0^{-1}$, and with the initial set of approximation close to the final solution, the BR, DFP and BFGS methods performed superior to the NR method by as much as 50%. The difference between the three was very little, but the BFGS method was found to be most stable and performed best.

The performance of the methods are summarised in Tables 5.21. The highlighted areas in the tables denote the most efficient method. This has been quantified by summing the values of $\eta_5$ obtained for the networks which have been examined. However, adoption of the NR method involves complex evaluation and coding of partial derivative as explained in Section 4.3.2. For this reason, and given the superior performance of the BFGS method is other situations, it is recommended that the BFGS algorithm be adopted for solving the types of network considered herein. When the initial set of approximations is known to be close to the final solution, then the NR method is the best method. Otherwise, the BFGS method is the best method regardless of initial conditions. In summary, it is recommended that the method of BFGS method be adopted for hydraulic network analysis involving open channels and other hydraulic modules.
<table>
<thead>
<tr>
<th>Network</th>
<th>NR</th>
<th>BR</th>
<th>DFP</th>
<th>BFGS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parallel Network</td>
<td>0.953</td>
<td>0.175</td>
<td>0.035</td>
<td>0.427</td>
</tr>
<tr>
<td>Nested Network</td>
<td>1.355</td>
<td>0.892</td>
<td>1.298</td>
<td>1.257</td>
</tr>
<tr>
<td>Sum</td>
<td>2.308</td>
<td>1.067</td>
<td>1.333</td>
<td>1.684</td>
</tr>
</tbody>
</table>

Table 5.21a. Overall performance evaluated with near initial approximation, $H_0 = I$.

<table>
<thead>
<tr>
<th>Network</th>
<th>NR</th>
<th>BR</th>
<th>DFP</th>
<th>BFGS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parallel Network</td>
<td>0.953</td>
<td>1.794</td>
<td>1.904</td>
<td>1.930</td>
</tr>
<tr>
<td>Nested Network</td>
<td>1.355</td>
<td>2.777</td>
<td>2.987</td>
<td>3.043</td>
</tr>
<tr>
<td>Sum</td>
<td>2.308</td>
<td>4.571</td>
<td>4.891</td>
<td>4.973</td>
</tr>
</tbody>
</table>

Table 5.21b. Overall performance evaluated with near initial approximation, $H_0 = J_0^{-1}$. 
## Applications and Testing of Methods of Solution of Networks

<table>
<thead>
<tr>
<th>Network</th>
<th>NR</th>
<th>BR</th>
<th>DFP</th>
<th>BFGS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parallel Network</td>
<td>0.727</td>
<td>0.390</td>
<td>0.056</td>
<td>0.644</td>
</tr>
<tr>
<td>Nested Network</td>
<td>0.784</td>
<td>0.959</td>
<td>0.285</td>
<td>0.926</td>
</tr>
<tr>
<td>Sum</td>
<td>1.511</td>
<td>1.349</td>
<td>0.341</td>
<td>1.570</td>
</tr>
</tbody>
</table>

Table 5.21c. Overall performance evaluated with poor initial approximation, \( H_0 = I \).

<table>
<thead>
<tr>
<th>Network</th>
<th>NR</th>
<th>BR</th>
<th>DFP</th>
<th>BFGS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parallel Network</td>
<td>0.727</td>
<td>0.643</td>
<td>1.037</td>
<td>0.979</td>
</tr>
<tr>
<td>Nested Network</td>
<td>0.784</td>
<td>0.608</td>
<td>0.404</td>
<td>0.671</td>
</tr>
<tr>
<td>Sum</td>
<td>1.511</td>
<td>1.251</td>
<td>1.441</td>
<td>1.650</td>
</tr>
</tbody>
</table>

Table 5.21d. Overall performance evaluated with poor initial approximation, \( H_0 = J_0^{-1} \).
5.4 Investigations of Switching of Hydraulic Control

An assumption made earlier in analysing open channel networks is that the state of flow is strictly subcritical. In a sense, this assumption is very restrictive, and it is advantageous to include other types of flow. It is not a problem to include various flow conditions for the analysis; the potential difficulty is with the phenomenon known as 'switching' of hydraulic control, for example when the flow changes from subcritical to supercritical or vice versa. This phenomenon could keep the state of flow changing from iteration to iteration during iterative process, which could cause some difficulties regarding convergence. To overcome these difficulties, the line search (see Chapter 4) is investigated. It is understood that the line search merely provides a way to enforce convergence and, in a sense, is operated independently of the iterative method adopted. The basic structure to implement the line search for a network solution is as follows:

i) Set up the system equation \( f(q) \) with respect to \( q \).

ii) Obtain an initial set of flow distribution.

iii) Use either NR, BR, DFP or BFGS method to determine the direction of search \( \delta_n \).

iv) Minimise \( \|f(q_n + t_n \delta_n)\| \) with respect to \( t_n \). The initial values of \( t_n \in [0, 1] \) may be used for each iteration. Use the technique of a triplet of points to bracket a minimum and then use the golden section search to locate the minimum. That is, obtain the optimum of \( t_n \) for the particular iteration of the algorithm.

v) Update \( q_{n+1} = q_n + t_n \delta_n \).

vi) Repeat Step iii) until \( \|f_{n+1}\| < \xi \).

The main factor concerned with line searches is that whilst it will enforce convergence, additional computational efforts are required to achieve a solution. Line searching is time consuming to operate because it requires additional function evaluations to enforce convergence. However, an 'exact line search' is important because it can affect the performance of the method in which it is embedded. The requirement for 'exactness' can be described by the following inequalities

\[
\left| \frac{f(t_j)}{f(t_{i+\varepsilon})} \right| < \varepsilon < 1, \quad j = i, i+1, \ldots
\]  

(5.7)
in which $n$ denotes the number of iterations. The condition (5.7) implies that the expectation of improvement on $f$ over two consecutive iterations $n$ and $n+1$ must be less than the expectation value $\varepsilon$. This means that when the search is terminated, the following inequality must hold:

$$|f_n| - |f_{n+1}| = |f_{n+1}| < |f_n|$$  \hspace{1cm} (5.8)

In certain situations, it is not possible to terminate the line searches with (5.7) satisfied. It is because sometimes a reduction of $\varepsilon$ cannot be achieved, especially when the value of $\varepsilon$ is chosen to be very small. The direction of search may not contain a 'deep' minimum. Instead of using (5.7) to terminate the line searches, the following arbitrary condition is used

$$\left| \frac{b_j - a_j}{b_i - a_i} \right| < 0.001, \hspace{0.5cm} j = i, i+1, \ldots$$  \hspace{1cm} (5.9)

in which $a$ and $b$ are the left-handed and right-handed bounds of a bracketing interval $[a, b]$. It must be stressed that conditions (5.7) and (5.9) are used to terminate the process of line searching for the $n$th iteration, whichever is being satisfied first.

To investigate the effects of line searches in combating potential problems switching of hydraulic control, the work presented herein sets up a network which is known to have difficulties in achieving a solution. It is envisaged that hydraulic controls will be switched (meaning that the flow regimes will keep changing), from iteration to iteration during the solution process. Two examples are provided for the investigation. The first provides the numerical data for a network problem of having no difficulties in achieving a solution. The second deals with the same network with different controls, giving rise to difficulties in convergence. The line search is incorporated in both examples to illustrate its significance.

The purpose of the investigation is to assess the convergence characteristics of the method in which it is imbedded. This includes the additional computational efforts required to achieve a solution, and the effects of $\varepsilon$ on the overall performance. Before the investigation can begin, a number of assumptions are included:
i) Minor headlosses such as eddy losses at nodal junctions are neglected.

ii) No lateral inflow or outflow occurs at any point in the network system.

iii) Zero flow does not occur at any point in the network system.

iv) Although the problem in hand is solving a set of non-smooth simultaneous equations, they are being treated as smooth to investigate the switching phenomenon.

v) In analysing open channel network problems, the flow direction is generally known a priori. It follows that the $\mu_n$ for $n = 1, 2, \ldots$ (see (4.67)) could be set to a value where a reverse flow takes place within the network. Hence, the scope of line searching is restricted.

### 5.4.1 Culvert Network with a Solution Obtained

The first example comprises three channels and a culvert as shown in Figure 5.8. The system was designed to illustrate the effects of control switching when the tailwater fluctuates in the vicinity of culvert the outlet. If the upstream water level of Channel-2 rises because of a gradual increase of flow, then the culvert outlet will eventually be submerged, and the flow will be classified as Type-5. Otherwise, the culvert flow can be either Type-1, Type-2 or Type-4. Type-3 and Type-6 will not occur in this particular example because the flow cannot be supercritical or hydraulically short. Although the configuration is simple, it provides some insight into applicability and capability of line searches.

A word about the headwater ratio $H_r$ is essential. The $H_r$ is not constant, but is a function of culvert entrance, barrel characteristics and approach condition. According to laboratory investigations, a value between $1.2 < H_r < 1.5$ (Chow, 1959; French, 1986) may be used. For illustration purposes, the value of investigation of $H_r = 1.5$ was chosen.

The physical characteristics of the network shown in Table 5.22 were chosen to ensure that switching of hydraulic controls would occur. It must be noted that the coefficient of discharge $C_d$ is also not a constant but a function of entrance geometry and depth of headwater. As $C_d$ will not affect the investigation of control switching, a value of $C_d = 0.85$ was chosen for the illustrations.

A computer run without incorporating a line search was conducted. The results shown in Tables 5.23 were obtained using the BFGS iterative formula because it is recommended...
in Section 5.3.5 as the method of choice. This initial flow distribution was deliberately chosen to be highly uneven because the intention was to investigate the convergence characteristics of the method under severe conditions. The convergence characteristics of the BFGS method are shown in Table 5.24 along with the various types of culvert flow occurring during the iterative process.

The convergence characteristics of the BFGS method can be judged by examining Table 5.23. The rapid convergence to the solution was caused in part by the high head-loss 1.3 (m) (see Table 5.23) incurred in the network, that is, the head-loss is sensitive to the flow correction made to the network. Hence, it also supports the findings in Section 5.3 that the BFGS method performs more efficiently with network of high sensitivity.

Table 5.24 shows that the BFGS method diverged in the 1st iteration and thereafter converged rapidly towards the solution. The divergence was due to the initial flow distribution being far away from the solution. The initial $H_0 = I$ was not a good approximation to the inversion of the Jacobian matrix, so that the loop flow correction $q_0$ was largely over-estimated to correct the string flows. Consequently, a slow rate of convergence was obtained but, eventually, accelerated to the final solution.

Switching of hydraulic control occurred in the first 6 iterations as the flow distribution was progressively updated. After the 6th iteration, the approximated string flows were sufficiently close to the final solution so that the same flow requires were maintained until a solution was obtained.

In order to investigate the effects of the line search on this network problem, a further eight separate computer runs were conducted using various degrees of exactness $0.001 \leq |\epsilon| \leq 0.700$ to indicate the accuracy of the line searches. The $|\epsilon| = 0.001$ denotes a high accuracy search whereas the $|\epsilon| = 0.700$ denotes a low accuracy search. All computer runs converged to the same solution as shown in Table 5.23, and the effects of $|\epsilon|$ on convergence are shown in Tables 5.25a to 5.25d; the highlighted areas denote where the line searches terminated with the condition satisfied (5.9) rather than (5.7). The heading 'Bracketing Interval' denotes the search interval where the leftmost and rightmost figures denote the left-handed and right-handed bounds of a bracket, whereas the middle figure denotes the ordinate of a desirable minimum. From the given results, two important points are drawn.
Firstly, divergence (1st iteration) observed previously, without incorporating the line search (see Table 5.24) did not occur in this case (see Tables 5.25). This is because $t$ restricts the iterations from moving too far away from the final solution. Therefore, strict convergence was obtained for all eight cases. The effects of $t$ on the rate of convergence are clearly shown in the 1st iteration in the tables. Different values of $t$ produced different values of $\|f\|$. It implies that the $t$ not only overcame the problem of divergence, but also to accelerated convergence.

It is seen that the number of iterations required to obtain the solution rose consistently with the coarseness of $|e|$. With high accuracy line searches ($|e|<0.001$), the rate of convergence was improved by as much as 64% whereas the coarser searches ($|e|>0.300$) exhibited no improvements at all. This is because the coarser searches could not locate the minimum accurately.

Secondly, the right-handed interval of $t$ seldom exceeded 3.000. Typical values of $t$ are found in Lemieux (1972) ranging from 0.5 to 1.1. These values are commonly applied to pipe network analysis when the Newton-Raphson method is used, and could provide a guideline for the range of $t$ in open channel analysis. According to these values, only two to three function evaluations were required to bracket a particular minimum, suggesting that the bulk of time was spent on the sectioning process. This can be justified by examining Table 5.26, in which $N_t$ denotes the total number of function evaluations required by line searches; and $N_t/n$ denotes the mean number of function evaluations per iteration. A corresponding plot of the table is also shown in Figure 5.9. It is seen that there exists a broad trend where the total computational efforts required rose with the coarseness of the searches. With smaller values of $|e|$, the searches required more function evaluations (see $N_t/n$) to locate a particular minimum but required less iteration to reach the solution so that the total number of function evaluations appeared to be less. On the other hand, the coarser searches required less function evaluations during the line searches but required more iterations to reach the solution so that a total number of function evaluations appeared to be more. It follows that a point exists which balances the coarseness of the search (function evaluations) and the number of iterations required to converge to the solution. Based on this example (see Figure 5.9), about $|e|=0.010$ is such a point where minimum function evaluations was attained.
Figure 5.8. A culvert network.

<table>
<thead>
<tr>
<th>Strings</th>
<th>Height, $H_b$ (m)</th>
<th>Width, $B_1$ (m)</th>
<th>Length, $L_1$ (m)</th>
<th>Bed Slope, $S_b$</th>
<th>Bank Slope, $m$</th>
<th>Manning, $n$</th>
<th>Coeffic'nt, $C_d$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inflow</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Channel - 1</td>
<td>-</td>
<td>1.00</td>
<td>1000.0</td>
<td>0.00050</td>
<td>0.0</td>
<td>0.0130</td>
<td>-</td>
</tr>
<tr>
<td>Channel - 2</td>
<td>-</td>
<td>1.00</td>
<td>1000.0</td>
<td>0.00050</td>
<td>0.0</td>
<td>0.0130</td>
<td>-</td>
</tr>
<tr>
<td>Channel - 3</td>
<td>-</td>
<td>1.00</td>
<td>3000.0</td>
<td>0.00050</td>
<td>0.0</td>
<td>0.0130</td>
<td>-</td>
</tr>
<tr>
<td>Culvert</td>
<td>0.70</td>
<td>1.00</td>
<td>2000.0</td>
<td>0.00025</td>
<td>0.0</td>
<td>0.0130</td>
<td>0.85</td>
</tr>
<tr>
<td>Outfall</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 5.22. Culvert Network - Physical characteristics.
<table>
<thead>
<tr>
<th>Strings</th>
<th>Energy Head, $H$, (m)</th>
<th>Water Depth, $y$, (m)</th>
<th>Discharge, $Q$, (m$^3$/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Upstream</td>
<td>Downstr'm</td>
<td>Upstream</td>
</tr>
<tr>
<td>Inflow</td>
<td>2.29796</td>
<td>2.29796</td>
<td>-</td>
</tr>
<tr>
<td>Channel - 1</td>
<td>2.29796</td>
<td>2.15551</td>
<td>0.78426</td>
</tr>
<tr>
<td>Channel - 2</td>
<td>1.20533</td>
<td>1.00000</td>
<td>0.68751</td>
</tr>
<tr>
<td>Channel - 3</td>
<td>2.29796</td>
<td>1.00000</td>
<td>0.76749</td>
</tr>
<tr>
<td>Culvert</td>
<td>2.15551</td>
<td>1.20533</td>
<td>1.14913</td>
</tr>
<tr>
<td>Outfall</td>
<td>1.00000</td>
<td>1.00000</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 5.23. Culvert Network - Flow distribution, $H_r = 1.50$. 

203
<table>
<thead>
<tr>
<th>Iteration, $n$</th>
<th>Type of Flow</th>
<th>$Q_1$, (m$^3$/s)</th>
<th>$Q_2$, (m$^3$/s)</th>
<th>$Q_3$, (m$^3$/s)</th>
<th>$Q_4$, (m$^3$/s)</th>
<th>$|f_n|_I$, (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Type-1</td>
<td>0.00100</td>
<td>0.00100</td>
<td>0.00100</td>
<td>0.99900</td>
<td>1.18877E+00</td>
</tr>
<tr>
<td>1</td>
<td>Type-4</td>
<td>0.81023</td>
<td>0.81023</td>
<td>0.81023</td>
<td>0.18977</td>
<td>8.14460E+00</td>
</tr>
<tr>
<td>2</td>
<td>Type-1</td>
<td>0.15241</td>
<td>0.15241</td>
<td>0.15241</td>
<td>0.84759</td>
<td>7.60226E-01</td>
</tr>
<tr>
<td>3</td>
<td>Type-1</td>
<td>0.24097</td>
<td>0.24097</td>
<td>0.24097</td>
<td>0.75903</td>
<td>5.57105E-01</td>
</tr>
<tr>
<td>4</td>
<td>Type-4</td>
<td>0.48387</td>
<td>0.48387</td>
<td>0.48387</td>
<td>0.51613</td>
<td>4.73278E-01</td>
</tr>
<tr>
<td>5</td>
<td>Type-1</td>
<td>0.37230</td>
<td>0.37230</td>
<td>0.37230</td>
<td>0.62770</td>
<td>2.53643E-01</td>
</tr>
<tr>
<td>6</td>
<td>Type-5</td>
<td>0.41123</td>
<td>0.41123</td>
<td>0.41123</td>
<td>0.58877</td>
<td>2.49798E-02</td>
</tr>
<tr>
<td>7</td>
<td>Type-5</td>
<td>0.40774</td>
<td>0.40774</td>
<td>0.40774</td>
<td>0.59226</td>
<td>6.32022E-03</td>
</tr>
<tr>
<td>8</td>
<td>Type-5</td>
<td>0.40656</td>
<td>0.40656</td>
<td>0.40656</td>
<td>0.59344</td>
<td>6.73956E-05</td>
</tr>
<tr>
<td>9</td>
<td>Type-5</td>
<td>0.40655</td>
<td>0.40655</td>
<td>0.40655</td>
<td>0.59345</td>
<td>1.87155E-07</td>
</tr>
<tr>
<td>10</td>
<td>Type-5</td>
<td>0.40655</td>
<td>0.40655</td>
<td>0.40655</td>
<td>0.59345</td>
<td>5.23137E-12</td>
</tr>
<tr>
<td>11</td>
<td>Type-5</td>
<td>0.40655</td>
<td>0.40655</td>
<td>0.40655</td>
<td>0.59345</td>
<td>1.42109E-14</td>
</tr>
<tr>
<td>12</td>
<td>Type-5</td>
<td>0.40655</td>
<td>0.40655</td>
<td>0.40655</td>
<td>0.59345</td>
<td>8.88178E-15</td>
</tr>
<tr>
<td>13</td>
<td>Type-5</td>
<td>0.40655</td>
<td>0.40655</td>
<td>0.40655</td>
<td>0.59345</td>
<td>1.77636E-15</td>
</tr>
<tr>
<td>14</td>
<td>Type-5</td>
<td>0.40655</td>
<td>0.40655</td>
<td>0.40655</td>
<td>0.59345</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 5.24. Culvert Network - Convergence characteristics, $H_r = 1.50$. 
### Table 5.25a. Culvert Network - Convergence characteristics with line searches, $H_r = 1.50$.

| $n$ | $|e| = 0.001$ | Bracketing Interval, $t_n$ | $|e| = 0.005$ | Bracketing Interval, $t_n$ |
|-----|---------------|-----------------------------|---------------|-----------------------------|
| 0   | 1.18877E+00   | -                           | 1.18877E+00   | -                           |
| 1   | 8.49335E-03   | [0.00000, 0.34249, 1.00000] | 8.49335E-03   | [0.00000, 0.34249, 1.00000] |
| 2   | 1.93679E-05   | [0.00000, 0.55417, 1.00000] | 1.93679E-05   | [0.00000, 0.55417, 1.00000] |
| 3   | 1.21758E-08   | [0.61803, 1.00310, 1.23606] | 2.48600E-08   | [0.61803, 1.00502, 1.23606] |
| 4   | 9.23705E-14   | [0.61803, 1.00000, 1.23606] | 7.69908E-11   | [0.61803, 0.99689, 3.00000] |
| 5   | 0             | [1.00000, 3.19720, 4.01197] | 1.11910E-13   | [0.61803, 0.99689, 3.00000] |
| 6   | 0             | [1.00000, 3.18034, 4.23606] |               |                             |

### Table 5.25b. Culvert Network - Convergence characteristics with line searches, $H_r = 1.50$.

| $n$ | $|e| = 0.010$ | Bracketing Interval, $t_n$ | $|e| = 0.050$ | Bracketing Interval, $t_n$ |
|-----|---------------|-----------------------------|---------------|-----------------------------|
| 0   | 1.18877E+00   | -                           | 1.18877E+00   | -                           |
| 1   | 1.09563E-02   | [0.00000, 0.33939, 1.00000] | 4.06676E-02   | [0.00000, 0.34752, 1.00000] |
| 2   | 2.10973E-05   | [0.00000, 0.55728, 1.00000] | 1.92342E-03   | [0.00000, 0.52786, 1.00000] |
| 3   | 6.82704E-08   | [0.61803, 0.99186, 1.23606] | 3.29915E-05   | [0.61803, 1.00000, 1.23606] |
| 4   | 1.75859E-12   | [0.61803, 1.00000, 1.23606] | 2.80244E-08   | [0.61803, 1.00000, 1.23606] |
| 5   | 0             | [0.61803, 1.05572, 1.23606] | 2.07833E-13   | [0.61803, 1.00000, 1.23606] |
| 6   |               |                             | 1.20792E-13   | [1.00000, 1.32623, 1.38196] |
| 7   |               |                             | 0             | [0.61803, 1.19005, 2.03655] |

205
| n  | $|\epsilon| = 0.100$ | Bracketing Interval, $t_i$ | $|\epsilon| = 0.300$ | Bracketing Interval, $t_i$ |
|----|----------------|-----------------|----------------|----------------|
| 0  | 1.18877E+00    | -               | 1.18877E+00    | -               |
| 1  | 8.95309E-02    | [0.00000, 0.32623, 1.00000] | 2.82824E-01    | [0.00000, 0.38196, 1.00000] |
| 2  | 5.53889E-03    | [0.00000, 0.52786, 1.00000] | 1.30284E-02    | [0.00000, 0.52786, 1.00000] |
| 3  | 7.64875E-05    | [0.61803, 0.94427, 1.23606] | 1.63806E-04    | [0.61803, 1.09017, 1.23606] |
| 4  | 1.91988E-07    | [0.61803, 1.00000, 1.23606] | 1.37561E-05    | [0.61803, 1.09017, 1.23606] |
| 5  | 6.81765E-12    | [0.61803, 1.00000, 1.23606] | 2.47962E-06    | [0.61803, 1.18034, 3.00000] |
| 6  | 6.21724E-14    | [0.61803, 0.91337, 2.64235] | 2.23603E-07    | [0.61803, 1.09017, 1.23606] |
| 7  | 2.30926E-14    | [1.97213, 2.14080, 2.27210] | 4.03243E-08    | [0.61803, 1.18034, 3.00000] |
| 8  | 0              | [0.61803, 1.04759, 3.00000] | 7.27210E-09    | [0.61803, 1.18034, 3.00000] |
| 9  |                |                 | 6.55688E-10    | [0.61803, 1.09017, 1.23606] |
| 10 |                |                 | 5.92450E-11    | [0.61803, 1.09017, 1.23606] |
| 11 |                |                 | 5.39124E-12    | [0.61803, 1.09017, 1.23606] |
| 12 |                |                 | 2.84217E-14    | [0.61803, 1.09017, 1.23606] |
| 13 |                |                 | 1.42108E-14    | [1.00000, 1.18879, 1.32918] |
| 14 |                |                 | 1.77635E-15    | [0.61803, 1.47213, 1.73606] |
| 15 |                |                 | 0              | [0.00000, 0.38196, 1.00000] |

Table 5.25c. Culvert Network - Convergence characteristics with line searches, $H_r = 1.50$. 

206
<table>
<thead>
<tr>
<th>$n$</th>
<th>${c_{n}}$</th>
<th>Bracketing Interval, $t_n$</th>
<th>${c_{n}}$</th>
<th>Bracketing Interval, $t_n$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1.18877E+00</td>
<td>-</td>
<td>1.18877E+00</td>
<td>-</td>
</tr>
<tr>
<td>1</td>
<td>2.82824E-01</td>
<td>[0.00000, 0.38196, 1.00000]</td>
<td>2.82824E-01</td>
<td>[0.00000, 0.38196, 1.00000]</td>
</tr>
<tr>
<td>2</td>
<td>1.30284E-02</td>
<td>[0.00000, 0.52786, 1.00000]</td>
<td>8.28848E-02</td>
<td>[0.00000, 0.38196, 1.00000]</td>
</tr>
<tr>
<td>3</td>
<td>1.63806E-04</td>
<td>[0.61803, 1.09017, 1.23606]</td>
<td>6.54471E-04</td>
<td>[0.61803, 1.09017, 1.23606]</td>
</tr>
<tr>
<td>4</td>
<td>1.37561E-05</td>
<td>[0.61803, 1.09017, 1.23606]</td>
<td>3.51145E-05</td>
<td>[0.61803, 1.09017, 1.23606]</td>
</tr>
<tr>
<td>5</td>
<td>2.47962E-06</td>
<td>[0.61803, 1.18034, 3.00000]</td>
<td>6.32067E-06</td>
<td>[0.61803, 1.18034, 3.00000]</td>
</tr>
<tr>
<td>6</td>
<td>2.23603E-07</td>
<td>[0.61803, 1.09017, 1.23606]</td>
<td>5.70040E-07</td>
<td>[0.61803, 1.09017, 1.23606]</td>
</tr>
<tr>
<td>7</td>
<td>4.03243E-08</td>
<td>[0.61803, 1.18034, 3.00000]</td>
<td>1.02799E-07</td>
<td>[0.61803, 1.18034, 3.00000]</td>
</tr>
<tr>
<td>8</td>
<td>7.27210E-09</td>
<td>[0.61803, 1.18034, 3.00000]</td>
<td>9.26997E-09</td>
<td>[0.61803, 1.09017, 1.23606]</td>
</tr>
<tr>
<td>9</td>
<td>6.55688E-10</td>
<td>[0.61803, 1.09017, 1.23606]</td>
<td>8.35951E-10</td>
<td>[0.61803, 1.09017, 1.23606]</td>
</tr>
<tr>
<td>10</td>
<td>5.92450E-11</td>
<td>[0.61803, 1.09017, 1.23606]</td>
<td>1.50222E-10</td>
<td>[0.61803, 1.18034, 3.00000]</td>
</tr>
<tr>
<td>11</td>
<td>5.39124E-12</td>
<td>[0.61803, 1.09017, 1.23606]</td>
<td>2.77857E-11</td>
<td>[0.61803, 1.18034, 3.00000]</td>
</tr>
<tr>
<td>12</td>
<td>2.84217E-14</td>
<td>[0.61803, 1.09017, 1.23606]</td>
<td>3.26316E-12</td>
<td>[0.61803, 1.09017, 1.23606]</td>
</tr>
<tr>
<td>13</td>
<td>1.42108E-14</td>
<td>[1.00000, 1.18879, 1.32918]</td>
<td>9.76996E-14</td>
<td>[0.61803, 0.96555, 3.00000]</td>
</tr>
<tr>
<td>14</td>
<td>1.77635E-15</td>
<td>[0.61803, 1.47213, 1.73606]</td>
<td>1.77635E-15</td>
<td>[1.00000, 3.47213, 4.23606]</td>
</tr>
<tr>
<td>15</td>
<td>0</td>
<td>[0.00000, 0.38196, 1.00000]</td>
<td>0</td>
<td>[0.00000, 0.76393, 1.00000]</td>
</tr>
</tbody>
</table>

Table 5.25d. Culvert Network - Convergence characteristics with line searches,

$H_r=1.50$.  

\[ \% \]
APPLICATIONS AND TESTING OF METHODS OF SOLUTION OF NETWORKS

| $|\varepsilon|$ | Iterations, $n$ | $N_t$ | Average $N_t/n$ | Total $N_t + n$ |
|----------------|----------------|-------|----------------|----------------|
| 0.001          | 5              | 47    | 9.4            | 52             |
| 0.005          | 6              | 59    | 9.8            | 65             |
| 0.010          | 5              | 36    | 7.2            | 41             |
| 0.050          | 7              | 42    | 6.0            | 49             |
| 0.100          | 8              | 44    | 5.5            | 52             |
| 0.300          | 15             | 51    | 3.4            | 66             |
| 0.500          | 15             | 55    | 3.6            | 70             |
| 0.700          | 15             | 63    | 4.2            | 78             |

Table 5.26. Culvert Network - Total number of function evaluations, $H_r = 1.50$.

Figure 5.9. Culvert Network - A plot of function evaluations against $|\varepsilon|$, $H_r = 1.50$. 

208
According to these results, the main advantages of line searching are

i) to stabilise the behaviour of the method in which it is imbedded;
ii) to increase the rate of convergence.

However, a line search is expensive to carry out since increasingly accurate line searches require more computational effort but with decreasing returns of overall efficiency. One would suggest to use line searches only when divergence was encountered. The accuracy of the searches required varies with the specific problem so that it is difficult to predefine a specific value for the \( \epsilon \). As a guideline, the values between \( 0.010 < |\epsilon| < 0.100 \) may be used based on these examples.

### 5.4.2 Culvert Network without a Solution Obtained

To test whether the line search will enforce convergence, the network shown in Figure 5.5 was re-analysed. A different value of \( H_r \) was selected so as to deliberately cause difficulties with the BFGS method. The role of \( H_r \) is to differentiate the culvert flows between Types 1-3 and Types 4-6. For the purpose of investigating the stability of the procedures, a value of \( H_r = 1.65 \) was chosen. It implies that switching of hydraulic control occurs from \( H_r = 1.65 \) rather than \( H_r = 1.50 \). It must be stressed that it is not intended to redefine the operating rules for the culvert flow, but merely to demonstrate certain numerical problems associated with composite flow characteristics where potential discontinuities occur.

A computer run was conducted under the conditions identical to the previous example. Table 5.27 shows the convergence characteristics of the BFGS method without incorporating line searches. It is seen that the method failed to converge to a solution; it oscillated about a point indefinitely. This was perhaps caused by the search along the direction \( q \) overshooting a very steep valley where a solution lies. This situation can occur especially where a discontinuity occurs in hydraulic operating rules (for example, as defined by the \( H_r \) value). To verify this point, two graphs corresponding to \( H_r = 1.50 \) and \( H_r = 1.65 \) were plotted in Figures 5.10a and 5.10b respectively; they are described below.
The graphs of $\|f(q)\|$ shown in Figures 5.10a and 5.10b were plotted with finite incremental steps of $q$, and each consists a composite curve (appears to be a serious of straight lines), and the curves denote sum of head-losses around the closed loop. The Point R, denotes where the solution lies. The intersection Points C and D denote the positions where switching of hydraulic controls occured. Curves $\overline{AB}$, $\overline{CD}$ and $\overline{DE}$ correspond to culvert flows Type-1, Type-5 and Type-4 respectively. The section $\overline{BC}$ in the graphs denotes the discontinuity whose position is defined by $H_r$. The larger the value of $H_r$, the further $\overline{BC}$ travels to the right of Point A.

Firstly, a solution was obtained in Figure 5.10a but not in Figure 5.10b. This can be explained directly from the graphs. A solution exists only when the curve $\overline{CD}$ intersects the $q$-axis such as at Point R in Figure 5.10a, that is when $\|f(q)\|=0$. On the other hand, the curve $\overline{CD}$ in Figure 5.10b did not intersect the $q$-axis suggesting that no solution can be attained. A minimum situated at Point C is found instead, because the discontinuity $\overline{BC}$ ($H_r = 1.65$) is too far to the right and passed the point where the solution lies. The position of the solution is defined by $H_r$, for example, the Point R is the solution when $H_r = 1.50$ is used. Due to the fact that the position of the discontinuity $\overline{BC}$ is predefined by the condition $H_r = 1.65$ as mentioned above. When iterating close to such a condition, switching of flow equations from Type-1 to Type-5 or vice versa will occur, and it becomes impossible for the solution process to converge to the solution; hence it iterates indefinitely.

Secondly, the oscillatory behaviour of this example can also be depicted from Figures 5.10a and 5.10b. The arrows shown on the graphs describe the behaviour of the converging process, in which the numbers denote the $n$th iteration. Referring to Figure 5.10b, since the vertical slope $\overline{BC}$ lies just to the right of Point 7 (same point in Figure 5.10a), the prediction from the 6th iteration to the 7th iteration (same prediction shown in Figure 5.10a) appeared to be over-estimated and hence caused switching of flow equations. Therefore, the resulting prediction is seen to be Point 8 in Figure 5.10b (compared with Point 8 in Figure 5.10a). The subsequent predictions from the 8th iteration can be anywhere along the curve $\overline{ABCDE}$. It follows that the oscillatory behaviour of this example was due to the vertical slope of $\overline{BC}$ either coinciding with or being in the close vicinity of the minimum C (or the solution if Point C intersects the $q$-axis).
One suggestion to overcome the above problem is to use line searches. As in the previous example, eight separate computer runs were conducted with different values of \(|e|\), and the convergence characteristics were obtained in Tables 5.28a to 5.28d.

Firstly, with \(|e| \in \{0.001, 0.005, 0.010, 0.050, 0.100, 0.300, 0.500\}\), the convergence characteristics were identical. This was due to the searches terminating for the same reason (5.9) so that the \(t_n\) predicted the same point \(q(t_n)\) for different coarseness of line search. It implies that the higher accuracy searches provided no improvement in the rate of convergence.

Secondly, it was found that all eight runs converged to the same minimum \(C\) but exhibited no oscillatory behaviour. The line searches enforced convergence so that the condition \(||f_{n+1}|| < ||f_n||\) always held. This means that the predictions were restricted to the right-handed side of the vertical slope \(BC\) of the graphs. There exists a convex curve (a requirement for convergence) and appears to have no discontinuities within the domain of interest. When \(t_n = 0\) was found, it signified either a minimum or a solution was obtained. The following conditions differentiate between the two conditions:

A solution: \(||f_{n+1}|| = 0\) and \(t_n = 0\)

A minimum: \(||f_{n+1}|| > 0\) and \(t_n = 0\)

It this example, a minimum has been located, whilst it is not a true solution, it does enable the network analysis to proceed. However, the overall accuracy will be affected because a true solution to this part of the network has not been found. The effect on the flow distribution can be investigated by comparing Table 5.24 (14th iteration) and Table 5.27 (10th iteration). The iterations 7 and 14 are chosen because they represent the approximate solution where the minimum \(C\) lies and the exact solution respectively. As mentioned previously, this network is a relatively high head-loss network and its rate of change of \(||f_n||\) with respect to a unit change of \(||q_n||\) be calculated as follows:

\[
\frac{||f_7|| - ||f_{14}||}{||q_7||} = \frac{6.32022 \times 10^{-3}}{1.19000 \times 10^{-3}} = 5.311
\]
### Broyden-Fletcher-Goldfarb-Shanno

<table>
<thead>
<tr>
<th>Iteration, ( n )</th>
<th>Type of Flow</th>
<th>( Q_1, (m^3/s) )</th>
<th>( Q_2, (m^3/s) )</th>
<th>( Q_3, (m^3/s) )</th>
<th>( Q_4, (m^3/s) )</th>
<th>( | r_n |, (m) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Type-1</td>
<td>0.00100</td>
<td>0.00100</td>
<td>0.00100</td>
<td>0.99900</td>
<td>1.18877E+00</td>
</tr>
<tr>
<td>1</td>
<td>Type-4</td>
<td>1.18977</td>
<td>1.18977</td>
<td>1.18977</td>
<td>0.18977</td>
<td>8.14460E+00</td>
</tr>
<tr>
<td>2</td>
<td>Type-1</td>
<td>0.15241</td>
<td>0.15241</td>
<td>0.15241</td>
<td>0.84759</td>
<td>7.60226E-01</td>
</tr>
<tr>
<td>3</td>
<td>Type-1</td>
<td>0.24097</td>
<td>0.24097</td>
<td>0.24097</td>
<td>0.75903</td>
<td>5.57105E-01</td>
</tr>
<tr>
<td>4</td>
<td>Type-4</td>
<td>0.48387</td>
<td>0.48387</td>
<td>0.48387</td>
<td>0.51613</td>
<td>4.73278E-01</td>
</tr>
<tr>
<td>5</td>
<td>Type-1</td>
<td>0.37230</td>
<td>0.37230</td>
<td>0.37230</td>
<td>0.62770</td>
<td>2.53634E-01</td>
</tr>
<tr>
<td>6</td>
<td>Type-5</td>
<td>0.41123</td>
<td>0.41123</td>
<td>0.41123</td>
<td>0.58877</td>
<td>2.49798E-02</td>
</tr>
<tr>
<td>7</td>
<td>Type-5</td>
<td>0.40774</td>
<td>0.40774</td>
<td>0.40774</td>
<td>0.59226</td>
<td>6.32022E-03</td>
</tr>
<tr>
<td>8</td>
<td>Type-1</td>
<td>0.40656</td>
<td>0.40656</td>
<td>0.40656</td>
<td>0.59344</td>
<td>1.70897E-01</td>
</tr>
<tr>
<td>9</td>
<td>Type-5</td>
<td>0.40770</td>
<td>0.40770</td>
<td>0.40770</td>
<td>0.59230</td>
<td>6.09663E-03</td>
</tr>
<tr>
<td>10</td>
<td>Type-5</td>
<td>0.40766</td>
<td>0.40766</td>
<td>0.40766</td>
<td>0.59234</td>
<td>5.88842E-03</td>
</tr>
<tr>
<td>11</td>
<td>Type-1</td>
<td>0.40655</td>
<td>0.40655</td>
<td>0.40655</td>
<td>0.59345</td>
<td>1.70919E-01</td>
</tr>
<tr>
<td>12</td>
<td>Type-1</td>
<td>0.40763</td>
<td>0.40763</td>
<td>0.40763</td>
<td>0.59237</td>
<td>1.68477E-01</td>
</tr>
<tr>
<td>13</td>
<td>Type-4</td>
<td>0.48169</td>
<td>0.48169</td>
<td>0.48169</td>
<td>0.51831</td>
<td>4.58337E-01</td>
</tr>
<tr>
<td>14</td>
<td>Type-4</td>
<td>0.42753</td>
<td>0.42753</td>
<td>0.42753</td>
<td>0.57247</td>
<td>1.15935E-01</td>
</tr>
<tr>
<td>15</td>
<td>Type-5</td>
<td>0.40920</td>
<td>0.40920</td>
<td>0.40920</td>
<td>0.59080</td>
<td>1.40454E-02</td>
</tr>
<tr>
<td>16</td>
<td>Type-1</td>
<td>0.40667</td>
<td>0.40667</td>
<td>0.40667</td>
<td>0.59333</td>
<td>1.70656E-01</td>
</tr>
<tr>
<td>17</td>
<td>Type-5</td>
<td>0.40900</td>
<td>0.40900</td>
<td>0.40900</td>
<td>0.59100</td>
<td>1.30195E-02</td>
</tr>
<tr>
<td>18</td>
<td>Type-5</td>
<td>0.40884</td>
<td>0.40884</td>
<td>0.40884</td>
<td>0.59116</td>
<td>1.21368E-02</td>
</tr>
<tr>
<td>19</td>
<td>Type-1</td>
<td>0.40656</td>
<td>0.40656</td>
<td>0.40656</td>
<td>0.59344</td>
<td>1.70897E-01</td>
</tr>
<tr>
<td>20</td>
<td>Type-5</td>
<td>0.40869</td>
<td>0.40869</td>
<td>0.40869</td>
<td>0.59131</td>
<td>1.13325E-02</td>
</tr>
<tr>
<td>21</td>
<td>Type-5</td>
<td>0.40856</td>
<td>0.40856</td>
<td>0.40856</td>
<td>0.59144</td>
<td>1.06287E-02</td>
</tr>
<tr>
<td>22</td>
<td>Type-1</td>
<td>0.40656</td>
<td>0.40656</td>
<td>0.40656</td>
<td>0.59344</td>
<td>1.70903E-01</td>
</tr>
<tr>
<td><strong>99</strong></td>
<td><strong>Type-1</strong></td>
<td><strong>0.40655</strong></td>
<td><strong>0.40655</strong></td>
<td><strong>0.40655</strong></td>
<td><strong>0.59345</strong></td>
<td><strong>1.70914E-01</strong></td>
</tr>
<tr>
<td>100</td>
<td>Type-5</td>
<td>0.40795</td>
<td>0.40795</td>
<td>0.40795</td>
<td>0.59205</td>
<td>7.40629E-03</td>
</tr>
</tbody>
</table>

Table 5.27. Culvert Network - Convergence characteristics, \( H_r = 1.65 \).
Figure 5.10a. Culvert Network - Illustration of convergence trend, $H_r = 1.50$.

Figure 5.10b. Culvert Network - Illustration of difficulties in convergence, $H_r = 1.65$. 
### Table 5.28a. Culvert Network - Convergence characteristics with line searches, $H_r = 1.65$.  

| n | $|e| = 0.001$ | $|e| = 0.005$ |
|---|---|---|
| | $|t_n|$ | Bracketing Interval, $t_n$ | $|t_n|$ | Bracketing Interval, $t_n$ |
| 0 | 1.18877E+00 | - | 1.18877E+00 | - |
| 1 | 8.49335E-03 | [0.00000, 0.34249, 1.00000] | 8.49335E-03 | [0.00000, 0.34249, 1.00000] |
| 2 | 5.72884E-03 | [0.00000, 0.18034, 1.00000] | 5.72884E-03 | [0.00000, 0.18034, 1.00000] |
| 3 | 5.71107E-03 | [0.00000, 0.00310, 1.00000] | 5.71107E-03 | [0.00000, 0.00310, 1.00000] |
| 4 | 5.71107E-03 | [0.00000, 0.00000, 1.00000] | 5.71107E-03 | [0.00000, 0.00000, 1.00000] |

### Table 5.28b. Culvert Network - Convergence characteristics with line searches, $H_r = 1.65$.  

| n | $|e| = 0.010$ | $|e| = 0.050$ |
|---|---|---|
| | $|t_n|$ | Bracketing Interval, $t_n$ | $|t_n|$ | Bracketing Interval, $t_n$ |
| 0 | 1.18877E+00 | - | 1.18877E+00 | - |
| 1 | 8.49335E-03 | [0.00000, 0.34249, 1.00000] | 4.06676E-02 | [0.00000, 0.34752, 1.00000] |
| 2 | 5.72884E-03 | [0.00000, 0.18034, 1.00000] | 5.72908E-03 | [0.00000, 0.47524, 1.00000] |
| 3 | 5.71107E-03 | [0.00000, 0.00310, 1.00000] | 5.71107E-03 | [0.00000, 0.00310, 1.00000] |
| 4 | 5.71107E-03 | [0.00000, 0.00000, 1.00000] | 5.71107E-03 | [0.00000, 0.00000, 1.00000] |
### Table 5.28c. Culvert Network - Convergence characteristics with line searches, $H_r = 1.65$.

| n  | $|\epsilon| = 0.100$ | $|\epsilon| = 0.300$ |
|----|------------------|------------------|
|    | $\|r_n\|$  | Bracketing Interval, $t_n$ | $\|r_n\|$ | Bracketing Interval, $t_n$ |
| 0  | 1.18877E+00 | - | 1.18877E+00 | - |
| 1  | 4.06676E-02 | [0.00000, 0.34752, 1.00000] | 2.09580E-01 | [0.00000, 0.32623, 1.00000] |
| 2  | 5.72908E-03 | [0.00000, 0.47524, 1.00000] | 9.89952E-03 | [0.00000, 0.23606, 1.00000] |
| 3  | 5.71107E-03 | [0.00000, 0.00310, 1.00000] | 5.71107E-03 | [0.00000, 0.89164, 1.00000] |
| 4  | 5.71107E-03 | [0.00000, 0.00000, 1.00000] | 5.71107E-03 | [0.00000, 0.00000, 1.00000] |

### Table 5.28d. Culvert Network - Convergence characteristics with line searches, $H_r = 1.65$.

| n  | $|\epsilon| = 0.500$ | $|\epsilon| = 0.700$ |
|----|------------------|------------------|
|    | $\|r_n\|$  | Bracketing Interval, $t_n$ | $\|r_n\|$ | Bracketing Interval, $t_n$ |
| 0  | 1.18877E+00 | - | 1.18877E+00 | - |
| 1  | 2.82824E-01 | [0.00000, 0.38196, 1.00000] | 2.82824E-01 | [0.00000, 0.38196, 1.00000] |
| 2  | 1.30284E-02 | [0.00000, 0.52786, 1.00000] | 8.28848E-02 | [0.00000, 0.38196, 1.00000] |
| 3  | 5.71107E-03 | [0.00000, 0.61803, 1.00000] | 1.79998E-02 | [0.00000, 0.85410, 1.23606] |
| 4  | 5.71107E-03 | [0.00000, 0.00000, 1.00000] | 8.77775E-03 | [0.00000, 0.52786, 1.00000] |
| 5  | 5.92892E-03 | [0.00000, 0.32623, 1.00000] | 5.71107E-03 | [0.00000, 0.03444, 1.00000] |
| 6  | 5.71107E-03 | [0.00000, 0.00000, 1.00000] | 5.71107E-03 | [0.00000, 0.00000, 1.00000] |
Hence, $\|f_n\| = 5.71007 \times 10^{-3}$ (see Tables 5.28) incurs an apparent error in flow of

\[
\frac{5.71007 \times 10^{-3}}{5.311} = 1.07514 \times 10^{-3} \text{ (m}^3\text{/s)}
\]

or approximately 0.26% of flow is incorrectly distributed. In terms of practical applications, this error in flow distribution is relatively small, for example in wastewater treatment works.

It must be stressed that the line search is used to enforce convergence but it does not guarantee global convergence to a solution. It is due to the fact that the search directions may not contain the solution, but only some local minima. Once the procedure has converged to a local minimum, it is then very difficult to find a path to converge to the actual solution. The only approach remedy is to select the initial set of approximation to be close to the solution but this is not usually feasible. It is not known in general terms whether complex system such as these have an exact solution.

5.6 Concluding Remarks

Determination of the flow distribution in an open channel network requires solving a set of non-linear simultaneous equations. In order to solve these equations, the NR, BR, DFP and BFGS procedures have been investigated. Two network problems were designed for these investigations are one parallel and one nested network systems. The basis for comparison is to assess the accuracy of the final solution, the total number of function evaluations and the total number of iterations required to reach a prescribed tolerance $\chi_{r+1} < 10^{-4} \text{ (m}^3\text{/s)}$. Based on these factors, the mean convergence rate $\eta_8$ for a particular method can be calculated to describe overall performance.

It was found that the NR method oscillated indefinitely in the neighbourhood of the solution because the approximation to the Jacobian matrix was unstable; either the size of $h$ was chosen too small or the size of $h$ was chosen too large so that the flow corrections were incorrectly estimated. An investigation was carried out to determine the effects of $h$ on convergence. With larger $h$, convergence was slower. The optimum size of $h$ was investigated and found to be bounded by $10^{-2} < h < 10^{-8}$ for the test cases.
As far as rate of convergence is concerned, the NR method converged considerably faster than the quasi-Newton methods in all test cases. Since the NR method required more function evaluations than the quasi-Newton methods, it was found that the overall performance of BFGS method was comparable with the NR method. On the other hand, the BR and DFP methods were shown to fail to converge in one test case (very low sensitivity network). It implies that the BR and DFP methods are not always stable depending upon the form of the equations and sensitivity of the network, whereas the BFGS method was found to be stable in this respect for the situations investigated.

When the initial approximation was far from the solution, the performance of the NR method deteriorated by as much as 50%, whereas the quasi-Newton methods were found to improve in performance. A summary of the performance of the methods is found in Tables 5.21. The highlighting areas denote the most efficient method for the specific investigation. When the initial set of approximation is known to be close to the final solution, the NR is the best method; otherwise, the BFGS is the method of choice regardless of initial conditions. Therefore, it is recommended that the BFGS method be adopted for hydraulic network analysis involving open channels and other modules.

Switching of controls can occur when the string flows are updated during the iterative process. An updating in string flow can cause a change in flow condition, for example changing the flow from subcritical to supercritical or the flow over a particular structure becomes submerged. If switching of control happens frequently, serious problems with network analysis might be expected.

Thus, two network problems were designed to investigate this. The main factor of concern was convergence. It was found that switching of controls did cause convergence problems with indefinite oscillations. This oscillatory behaviour was due to discontinuities in the flow equations in the vicinity of a minimum or a solution.

A technique suggested to overcome this is the line search. Principally, the technique is a sub-problem of minimisation which is designed to be imbedded in any iterative procedure. Although the method does not require derivative measurements, it requires a number of function evaluations to locate a particular minimum depending on the accuracy of the search. The technique was applied successfully to the network examples and the following points are drawn:
i) It prevents divergence when the condition \( \| f_{n+1} \| < \| f_n \| \) is enforced.

ii) It stabilises the iterative method in which it is imbedded, and increases the rate of convergence. A 64% increase in rate of convergence was obtained with \( |e| < 0.001 \) for the example used.

iii) In general, higher accuracy line searches \( |e| < 0.010 \) require more function evaluations but less iterations to reach a solution. On the other hand, the coarser searches \( |e| > 0.100 \) require less function evaluations but more iterations. Values of \( |e| < 0.001 \) appear unlikely to result in benefits commensurate with the computational effort required. The range of \( 0.100 < |e| < 0.010 \) is recommended for applications based on the tests carried out.

iv) With line searches imbedded, the additional function evaluations required to achieve the solution were found to vary between 6 and 8 per iteration (corresponding to \( 0.100 < |e| < 0.010 \)) implying that 6 to 8 times the computational effort is required.

As seen in iv), line searches are expensive to carry out. It is suggested that line searches are brought in when divergence is encountered. If there is a certain need to prevent divergence or enforce convergence, then the use of line searches is essential.
CHAPTER 6

PROBLEMS ARISING
FROM NET INFLOW AND OUTFLOW
6 PROBLEMS ARISING FROM NET INFLOW AND OUTFLOW

6.1 Introduction

Previous developments of a network solution have assumed that the net inflow and outflow must be given a priori as a boundary condition. This requirement is necessary to maintain continuity of flow throughout the network during the solution process. The initial approximation is estimated based on the net inflows and outflows, and thence an iterative algorithm is used, such as the BFGS algorithm, to improve the initial approximation, and eventually to achieve a final solution. However, this requirement might not be met in practice in particular when, say, the outflow is restricted so that the excess flow through the network must be diverted out of the system. If the exact quantity of this diverted flow is known, then it will not give rise to any problem in achieving a solution. The problem which arises from the diverted flow is that the exact quantity may be unknown in practice when certain common hydraulic structures are used, because it is governed by several factors such as the flow capacity of the network; upstream and downstream water depths and the total quantity of flow entering the network. This, in turn, leads to the problem of unknown or variable inflow through the network. It is, therefore, difficult to make an estimate of the initial flow distribution which satisfies continuity of flow, and hence the solution process cannot begin. side-weir structures are a particular case where these problems arise, which are shown in Figures 6.1a and 6.1b.

The overflow from a side-weir is dependent upon the location of a control point which may be at either upstream or downstream end of the side-weir structure. The conditions at the control are determined by the water depth and discharge as well as the physical characteristics of the side-weir. This means that the quantity of overflow water leaving the system cannot be determined independently of what is happening in the rest of the network. The result of this is that the side-weir overflow cannot be treated as an
independent variable in the network analysis. Therefore, for each iterative step in solving
the network problem, the actual value of side-weir overflow has to be determined based
on the network flow conditions which exist during the course of the iteration.

A side-weir is a free overflow weir structure which is set into the side of a channel to
allow part of the flow to spill over the side when the surface of the flow in the channel
rises above the side-weir crest. Side-weirs have been used extensively for water level
control in irrigation and in flood relief schemes on rivers for flood protection works
Another common use is as storm overflow structures in wastewater treatment plants for
diverting excess flood water when the inflow exceeds the plant capacity. The diverted
flow is, in general, discharged into storage tanks or into relief channels for later treatment.

Inclusion of side-weir structures in network analysis leads to the problem that part of the
design flow is diverted and leaves the network. If the diverted flow is constant, then it
may be treated as an outflow leaving at the nodal junction. Due to the fact that the
diverted flow is a function of flow conditions upstream and downstream of the side-weir,
an additional variable is introduced in the system equations. This gives rise to the
situation of 'two unknowns and one equation' so that the network cannot be solved
readily. One approach is to incorporate some trial-and-error procedures, but this is likely
to be laborious and time consuming.

The way by which the loop equations are formulated cannot take into account this extra
degree of complexity, and it implies that a new technique for a network solution is
mandatory. However, in order to take full advantage of the analytical tools being
developed, the concept of virtual string is introduced. This concept deserves particular
attention as it requires no special modification to the loop equations while maintaining
flow continuity throughout the network. Moreover, this concept enables the solution to be
obtained directly without recourse to some subsidiary trial-and-error procedures. The
concept is generalised and applicable to other side discharge situations such as that of
siphon flows, and is described in detail in Section 6.3.
PROBLEMS ARISING FROM NET INFLOW AND OUTFLOW

Figure 6.1a. Network comprises open channels.

\[ Q_2 = Q_3 \]
\[ \therefore \text{network can be solved directly.} \]

Figure 6.1b. Network comprises a side-weir.

\[ Q_2 = Q_3 + Q_{\text{Overflow}} \]
\[ \therefore Q_{\text{Overflow}} \text{ is a variable depending upon } Q_2 \text{ or } Q_3, \]
\[ \text{and a water level condition not known a prior,} \]
\[ \therefore \text{network cannot be solved readily.} \]
Figure 6.2. Section of a water surface profile with side discharge.
6.2 Dynamic Equation for Spatially Varied Flow

In hydraulic terms, a long side-weir passing flood water from the main stream into a relief channel is governed by the dynamic equation of spatially varied flow. Figure 6.2 shows a small section of spatially varied flow, which describes the depth of flow in relation to the varying discharge in the main stream. Two fundamental principles frequently used for the derivation are constant energy principle or the momentum principle. The former ignores frictional losses in the direction of the main flow. When there exists a strong secondary current in the flow there exists a longitudinal component of velocity of the flow over the side-weir which deviates appreciably from the mean velocity in the main channel. In this case, the above assumption is invalidated. The use of the momentum principle which includes the longitudinal velocity component of the spill flow is more appropriate in these circumstances.

6.2.1 Constant Energy Principle

The application of energy principle for the derivation of spatially varied flow can be found in standard hydraulic texts (Chow, 1959; French, 1986). In the derivation, the longitudinal component of velocity of the water flowing over the side-weir at any section is treated the same as the mean velocity of the flow in the channel at that section. That is, the total energy of a unit mass of water remaining in the channel is constant. It means that the total energy line is parallel to the side-weir crest.

The total energy head, $H$, at any section along the side-weir length (see Figure 6.2) is given by

$$H = z + y + \frac{\alpha Q^2}{2gA^2}$$  \hspace{1cm} (6.1)$$

Differentiating (6.1) with respect to $x$, the distance measured in the direction of flow along the bed slope, gives

$$\frac{dH}{dx} = \frac{dz}{dx} + \frac{dy}{dx} + \frac{\alpha}{2g} \left( \frac{2Q}{A^2} \frac{dQ}{dx} - \frac{2Q^2}{A^3} \frac{dA}{dx} \right)$$  \hspace{1cm} (6.2)$$
in which the $dH/dx$ describes the change in energy head with respect to distance. Substituting $S_o$, $S_f$ and $Fr$ for a rectangular channel into (6.2) and solving for $dy/dx$ gives,

$$\frac{dy}{dx} = \frac{S_o - S_f - \frac{\alpha Q}{gA^2} \frac{dQ}{dx}}{1 - \frac{\alpha Q^2 T}{gA^3}}$$

where

$$\frac{dQ}{dx} = -\frac{2}{3} C_d \sqrt{2gh^3}$$

The above expression is known as the dynamic equation of spatially varied flow, and forms the basis of many analyses of side-weir discharge.

### 6.2.2 Momentum Principle

The assumption made with the constant energy method is that at any section the longitudinal component of velocity of the spill flow is equal to the velocity in the channel at that section. However, El-Khashab and Smith (1976) disagreed and found that

i) the longitudinal velocity component of the spill flow was invariably higher than that in the channel;

ii) the velocity distribution in the channel was very asymmetrical; the flow in the immediate neighbourhood of the side-weir was accelerating towards the upstream end of the side-weir;

iii) the observed total energy line deviated considerably from the constant energy assumption.

El-Khashab and Smith felt that the longitudinal velocity component of the spill flow cannot be ignored. They investigated the application of the momentum principle in order to bring their experimental observations and theoretical computations more into line. Hence, the spatially flow equation developed by El-Khashab and Smith is stated below:
\[
\frac{dy}{dx} = \frac{S_o - S_f - (2\beta V - U) \frac{1}{gA} \frac{dQ}{dx}}{1 - \frac{\beta Q^2 T}{gA^3}}
\]

(6.4)

where

\(U\) = longitudinal velocity component of the spill flow
\(V\) = mean velocity in the main stream
\(\beta\) = momentum coefficient

Note the similarity between (6.3) and (6.4). Taking \(U = V\) (that is, reflecting the longitudinal velocity component of the spill) and \(\alpha = \beta = 1\), the two equations become identical.

### 6.2.3 Classification of Side-Weir Flow

In broad terms, the state of flow in the main stream can be depicted directly either from (6.3) or (6.4). Consider the numerator in the equations; the sum is positive since \(dQ/dx\) is negative and its absolute value is much larger than that of \(S_f\). Therefore, a rising profile will occur if the denominator \(1 - Fr^2 > 0\) (subcritical); whereas a falling profile will occur if the denominator \(1 - Fr^2 < 0\) (supercritical).

For computational convenience, Balmforth and Sarginson (1978) have identified five different types of flow corresponding to various weir heights and bed slopes. These are summarised in Table 6.1 and described briefly below.

In Type-1 flow (mild slope, weir height greater than critical depth, downstream throttle), a sluice gate or throttle pipe is positioned at the downstream end of the weir to control the flow. The flow in the approach channel is subcritical, and the water surface is drawn down as it approaches the weir entrance, but it is impossible for the flow to draw down to the critical depth. Therefore, the flow along the weir will be subcritical throughout with a rising water profile. Computations commence at the weir exit, moving in the upstream direction.
In Type-2 flow (mild slope, weir height less than critical depth), the flow in the approach channel is subcritical, and the water surface is drawn down approximately to critical depth as it approaches the weir entrance. The approaching depth will be \(0.8y_c < y < 1.0y_c\) depending on the bed slope and the height of the weir. Balmforth and Sarginson have developed a chart to establish such a depth for computations. The flow along the weir will be supercritical throughout with a falling water surface profile. This type of flow will only occur when the upstream specific energy exceeds twice the height of the weir. That is,

\[ E_1 = y_1 + \frac{\alpha v^2}{2g} \geq 2H_w \]  

(6.5)

Computations commence at the critical depth point, moving in the downstream direction. However, the precise location of the critical depth point is not fully established but is known to be some distance downstream of the weir entrance. For computational convenience, it is assumed that the critical depth is located at the side-weir entrance.

In Type-3 flow (mild slope, weir height less than critical depth, downstream throttle), a sluice gate or throttle pipe is positioned at the downstream end of the weir to control the flow. With a very high rate of flow, \(E_1 \geq 2H_w\), the flow along the weir is first supercritical, but the flow farther downstream will be subcritical, since a hydraulic jump is caused by the downstream throttle. The position of the jump is unstable, and analysis of this type of flow is difficult.

In Type-4 flow (steep slope, weir height less than critical depth), the flow in the approach channel is supercritical, and the flow along the weir is supercritical throughout with a falling water profile. Computations commence at the weir entrance in the downstream direction.

In Type-5 flow (steep slope, weir height less than critical depth, downstream throttle), the flow regime is similar to that of Type-4, but a sluice gate or throttle pipe is positioned at the downstream end of the weir to change the flow to subcritical. The flow along the weir is first supercritical, but the flow in farther downstream will be subcritical, since a hydraulic jump is caused by the downstream throttle. The position of the jump is unstable, and analysis of this type of flow is difficult.
Table 6.1. Classification of side-weir flow.

<table>
<thead>
<tr>
<th>Side-Weir Flow</th>
<th>Slope, $S_o$</th>
<th>Height, $H_w$</th>
<th>Upstream Condition</th>
<th>Weir Condition</th>
<th>Control Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type-1</td>
<td>Mild</td>
<td>$H_w &gt; y_c$</td>
<td>$y_1 &gt; y_c$</td>
<td>Subcritical</td>
<td>Downstream Throttle</td>
</tr>
<tr>
<td>Type-2</td>
<td>Mild</td>
<td>$H_w &lt; y_c$</td>
<td>$y_1 &gt; y_c$</td>
<td>Supercritical</td>
<td>Upstream</td>
</tr>
<tr>
<td>Type-3</td>
<td>Mild</td>
<td>$H_w &lt; y_c$</td>
<td>$y_1 &gt; y_c$</td>
<td>Supercritical to Subcritical</td>
<td>Upstream and Throttle</td>
</tr>
<tr>
<td>Type-4</td>
<td>Steep</td>
<td>$H_w &lt; y_c$</td>
<td>$y_1 &gt; y_c$</td>
<td>Supercritical</td>
<td>Upstream</td>
</tr>
<tr>
<td>Type-5</td>
<td>Steep</td>
<td>$H_w &lt; y_c$</td>
<td>$y_1 &gt; y_c$</td>
<td>Supercritical to Subcritical</td>
<td>Upstream and Throttle</td>
</tr>
</tbody>
</table>
According to El-Khashab and Smiths' findings, there was a clear-cut difference in behaviour between subcritical flow and supercritical flow along the side-weir. The differences were not only in relation to the water surface profile, but also the flow passing over the side-weir. A relatively high ratio of $Q_s/Q_i$, in which $Q_i$ and $Q_s$ denote the upstream flow and spill flow along the weir respectively, was obtained for the subcritical conditions due to the rising water profile. Moreover, they illustrated the effects of side discharge on the value of $\alpha$ along the weir. For the subcritical flow, the value of $\alpha$ rose gradually and became very high towards the end of the weir because of the non-uniform velocity distribution mentioned previously; whereas, for the supercritical flow, the value of $\alpha$ did not rise far from unity.

Balmforth and Sarginson observed a linear relationship between $Q_i$ and $Q_s$ for the subcritical flow. A similar relationship was also drawn for the supercritical flow, but there was a relatively low increase in spill flow as the upstream flow increased.

### 6.2.4 Method of Solution

Direct integration of the spatially varied flow equation is not possible. Therefore, various other methods have been developed for analysing side-weir flows. The methods generally fall into three categories, namely empirical, analytical and numerical, some of which are described in Table 6.2.

One of the first analytical solutions for analysing side-weir flow is due to De Marchi (1934). It was developed for rectangular, horizontal and frictionless side-weir channels. These restrictions limit the direct application of the method to many practical problems. Chow (1959) applied the momentum principle and developed a solution based on the technique of finite differencing. The method is very attractive for hand computation and can be facilitated for machine computations. Unfortunately, the method suffers a serious problem of numerical accuracy. Experience shows that the calculation is very sensitive to the choice of the step sizes. El-Khashab and Smith (1976) adopted Prasad's procedure (1963) (trapezoidal method) whereas the Balmforth and Sarginson (1978) relied on the simultaneous solution of the fourth-order Runge-Kutta procedure.
### Problems Arising from Net Inflow and Outflow

<table>
<thead>
<tr>
<th>Methods</th>
<th>Solution</th>
<th>Equations</th>
</tr>
</thead>
</table>
| De Marchi (1934)             | Analytical Solution for Horizontal, Frictionless, Rectangular Section | \[
L = \frac{3B}{2 \omega C_d} \left[ \Phi \left( \frac{y_2}{E} \right) - \Phi \left( \frac{y_1}{E} \right) \right] \\
\Phi \left( \frac{y}{E} \right) = \frac{2E - 3H_w}{E - H_w} \sqrt{\frac{E - y}{y - H_w}} - 3 \sin^{-1} \sqrt{\frac{E - y}{y - H_w}}
\] |
| Chow (1959)                  | Method of Finite Difference                    | \[
\Delta y = \frac{\alpha Q_1 (V_1 + V_2)}{\varepsilon (Q_1 + Q_2)} \Delta V \left[ 1 - \frac{\Delta Q}{2Q_1} \right] - S_f \Delta x \\
\Delta V = V_1 - V_2 \\
\Delta Q = Q_1 - Q_2
\] |
| El-Khashab-Smith (1976)      | Second-Order Trapezoidal Method                | \[
\frac{dy}{dx} = \left( S_o - S_f - \frac{\alpha Q}{\varepsilon A^2} \frac{dQ}{dx} \right) \left/ \left( 1 - \frac{\alpha Q^2 T}{\varepsilon A^3} \right) \right.
\] |
| Balmforth-Sargenson (1978)   | Fourth-Order Runge-Kutta Method               | \[
\frac{dV}{dx} = -\frac{1}{A} \frac{dQ}{dx} \\
\frac{dy}{dx} = \left[ S_o - S_f - (2\beta - 1) \frac{V}{gA} \frac{dQ}{dx} \right] \sqrt{1 - \frac{\beta V^2 T}{gA}}
\] |

Table 6.2. Equations for the analysis of side-weir flow.
In order to derive a meaningful functional relationship for evaluating $U$, El Khashab and Smith have categorised the flow based on the ratio of $Q_t/Q_1$ as shown in Table 6.3. It is noted with the Category-2 flow that the equation for evaluating $U$ is in terms of $y_i$ and $V_i$. Because the water surface profile computations are started from the downstream end of the side-weir in an upstream direction, the values of $y_i$ and $V_i$ are not readily available. It implies that an iterative procedure must be incorporated to determine the exact solution.

With an initial approximate longitudinal velocity component, $U = V$, a water surface profile is computed to the upstream end of the side-weir. The computed value of $U$ (with the appropriate equation for the ratio of $Q_t/Q_1$) is then compared with the assumed value of $U$. If the two values agree, then the process terminates; otherwise, a new value of $U$ must be re-established by the technique of successive bisection. The method, as pointed out by Balmforth-Sarginson, involves

i) the use of empirical correlations to evaluate $U$ that cannot be extrapolated with certainty beyond the range of their tests;

ii) for subcritical flow, a preliminary computation to estimate $U$;

iii) an iterative solution at each step to compute the water surface profile.

To overcome the above drawbacks, Balmforth and Sarginson have developed an alternative method which assumes $U = V$. At any cross section the value of $U$ corresponds to the mean velocity some distance upstream is generally greater than the mean velocity at that section.

Consider the mean velocity $V$ of flow at any section along the channel; $V$ can be written as

$$V = \frac{Q_t - \int_0^x \frac{dQ}{dx}}{A}$$  \hspace{1cm} (6.6)$$

and

$$\frac{dV}{dx} = \frac{1}{A} \frac{dQ}{dx}$$  \hspace{1cm} (6.7)$$
### Table 6.3. Equations for estimating longitudinal velocity component, $U$.

<table>
<thead>
<tr>
<th>Category</th>
<th>Type of Flow</th>
<th>$Q_s/Q_t$</th>
<th>Equations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Category-1</td>
<td>Subcritical</td>
<td>$&lt; 0.5$</td>
<td>$U/V = \frac{y}{H_w}$</td>
</tr>
<tr>
<td>Category-2</td>
<td>Subcritical</td>
<td>$&gt; 0.5$</td>
<td>$V_t/V = \frac{y-H_w}{y_t-H_w}$</td>
</tr>
<tr>
<td>Category-3</td>
<td>Supercritical</td>
<td>-</td>
<td>$U/V = Fr_l$</td>
</tr>
</tbody>
</table>
With the assumption $U = V$ and the above modification to $V$, (6.4) becomes

$$\frac{dy}{dx} = \frac{S_e - S_f - (2\beta - 1) \frac{V}{gA} \frac{dQ}{dx}}{1 - \beta V^2 T} \frac{dQ}{dx}$$

(6.8)

Numerical methods will not produce an exact solution. If the accuracy of the solution is the prime factor, then the procedure developed by Balmforth and Sarginson is recommended. Balmforth and Sarginson adopted the fourth-order Runge-Kutta procedure to solve the simultaneous differential equations (6.7) and (6.8). A fuller description on the simultaneous solution of (6.7) and (6.8) can be found in Chapra and Canale (1989).

6.3 Networks Solution Procedures

6.3.1 Trial-And-Error Approach

The inclusion of spatially varied flow into network analysis is not readily achievable because the lateral discharge incurs an additional variable in the loop equations which cannot be solved readily. Therefore, it is necessary to find an additional equation to make the set complete.

Consider a typical single loop network shown in Figure 6.3a. It comprises three channels and a side-weir. The loop equations be written as

$$f_i = h_{f_1} - h_{f_2} - h_{f_3} - h_{fw}$$

(6.9)

in which subscripts 1, 2, 3 and $w$ denote strings of channels and side-weir respectively. The head-loss for each of the strings be written as

$$h_{f_1} = h_f(Q_1 + q_1, y_{1,1}, y_{1,2})$$
$$h_{f_2} = h_f(Q_2 - q_1, y_{2,1}, y_{2,2})$$
$$h_{f_3} = h_f(Q_3 - q_1, y_{3,1}, y_{3,2})$$
$$h_{fw} = h_f(Q_{w,1} - q_1, Q_{w,2} - q_1, y_{w,1}, y_{w,2})$$

(6.10)
in which subscripts \(w,1\) and \(w,2\) denote upstream and downstream of the side-weir respectively. The problem which now arises from the spatially varied flow situation is that the overflow discharge is initially unknown, and is only determined once the side-weir computations are complete. The side-weir module is split into three conceptual components in order to analyse the problem. These are:

- upstream end, \([U/S]\)
- downstream end, \([D/S]\)
- lateral discharge, \(\perp\)

as shown in Figure 6.3b. The flows just upstream end and downstream end are \(Q_{w,1}\) and \(Q_{w,2}\) respectively. By the law of continuity of flows

\[
\begin{align*}
Q_2 &= Q_{w,1} \\
Q_3 &= Q_{w,1} - Q_{w,o} \\
Q_{w,o} &= Q_{w,1} - Q_{w,o}
\end{align*}
\]

in which \(Q_{w,o}\) denotes the lateral discharge. Substituting (6.11) into (6.10) yields

\[
\begin{align*}
h_{f1} &= h_f((Q_1 + q_1, y_{1,1}, y_{1,2}) \\
h_{f2} &= h_f((Q_2 - q_1), y_{2,1}, y_{2,2}) \\
h_{f3} &= h_f((Q_3 - q_1) - Q_{w,o}, y_{3,1}, y_{3,2}) \\
h_{fu} &= h_f((Q_{w,1} - q_1), (Q_{w,1} - q_1) - Q_{w,o}, y_{w,1}, y_{w,2})
\end{align*}
\]

in which \(q\) denotes the loop flow correction and \(+ q_1\) denotes clockwise is positive. The \(Q_1\) and \(Q_{w,1}\) are known once the design flow (total incoming or outgoing flow), \(Q_d\), is given. If \(Q_{w,o}\) is constant, then (6.9) can be solved readily. Unfortunately, \(Q_{w,o}\) is not constant for spatially varied flow. Due to the fact that \(Q_{w,o}\) is not related to \(q_1\), but depends on the flow conditions upstream and downstream of the side-weir, it implies that \(Q_{w,o}\) is an independent variable in addition to \(q_1\). Apparently, there exist two unknowns \(q_1\) and \(Q_{w,o}\) but only one equation (6.9) is available, hence (6.9) cannot be solved directly.
PROBLEMS ARISING FROM NET INFLOW AND OUTFLOW

Figure 6.3a. A side-weir network.

Figure 6.3b. Equivalent network for the side-weir system.
PROBLEMS ARISING FROM NET INFLOW AND OUTFLOW

Figure 6.4. Flow chart for the external trial-and-error procedure.
The solution of (6.9) could be obtained by using an external trial-and-error procedure as shown in Figure 6.4. The procedure is to start with an assumed value of $Q_{w,o}$ and then solve (6.9) for $q_i$. When the solution of (6.9) is found, the assumed $Q_{w,o}$ is compared with the $Q_{w,o}$ obtained from solution. If there exists any discrepancy between the two values, then a new value of $Q_{w,o}$ is chosen. The process is repeated until the discrepancy is within the prescribed tolerance. This process will be extremely tedious, involving lengthy operations. The above approach is applicable providing only one side-weir is incorporated in the analysis. If more than one side-weir is incorporated in a particular network, then the task becomes increasingly impossible.

6.3.2 Virtual Strings Approach

Solving networks that incorporate side-weirs is complicated by the existence of lateral discharge. It has been shown that the lateral discharge incurs an additional variable in the system of loop equations. To find an unique network solution, it is necessary to find the additional equation. An equation is formulated once a closed loop is formed. A closed loop can be formed by inserting a virtual string into the existing network. This unique property of loop formation has motivated the investigation into the use of virtual strings, rather than methods which require external procedures to achieve a solution. With a virtual string, no special modification to the network is required, and the network is solved directly without recourse to external procedures, such as that trial-and-error approach shown in Figure 6.4. It must be stressed that virtual string is not related to anything physical, but simply provides a mechanism of closing loops. A virtual string is inserted into a network to provide

i) a necessary linkage to form an additional loop in order to maintain flow continuity within the network as a whole;

ii) sufficient number of equations to complete the set for a network solution;

iii) an indication by how much the final flow distribution is out-of-balance.

Before the virtual string is inserted into the network, it is necessary to ensure whether the flow over the side-weir will be redirected to points within the network or discharged independently of the existing network. The former case implies that the virtual string is not necessary because the additional loop (or equation) is automatically provided. The latter case implies that the virtual string is necessary because part of the design flow is
leaving the network so that continuity of flow within the network is no longer achieved. Thus, the main role of the virtual string can be considered to be the maintenance of flow continuity. The virtual string can be inserted into the network in one of two ways:

i) **Constant Inflow** - If the total incoming flow passing through the system inflow point is assumed constant, then a virtual string must be inserted in between the side-weir and the point of system outfall.

ii) **Constant Outflow** - If the total outgoing flow passing through the system outfall point is assumed constant, then a virtual string must be inserted in between the side-weir and the point of system inflow.

These two cases are considered in turn.

### 6.3.3 Constant Inflow

Consider the network shown in Figure 6.5a where the flow over the side-weir is discharged independently. It is also assumed that the system inflow (design flow) is constant. To model such a situation, a virtual string is required to connect between the overflow arrow and the point of unknown outflow. Thus, two loops are formed as shown in Figure 6.5b. The requirements are:

i) As indicated by (6.12), the $h_{f3}$ and $h_{fw}$ are dependent both on $q_1$ and $Q_{w,o}$. However, it is advantageous to express the equations in terms of the appropriate loop flow correction factors $q_1$ and $q_2$ because the whole set of loop equations can then be solved readily. Thus, the equivalent set of equations to (6.12) is rewritten and shown below:

\[
\begin{align*}
    h_{f1} &= h_f((Q_o + q_1), y_{1,1}, y_{1,2}) \\
    h_{f2} &= h_f((Q_{w,1} - q_1), y_{2,1}, y_{2,2}) \\
    h_{f3} &= h_f((Q_{w,1} - q_1) - (Q_{w,o} + q_2), y_{3,1}, y_{3,2}) \\
    h_{fw} &= h_f((Q_{w,1} - q_1) - (Q_{w,o} + q_2), y_{w,1}, y_{w,2})
\end{align*}
\]  

(6.13)

in which $q_2$ denotes the loop flow correction and which is used to correct $Q_{w,o}$ (the terms $Q_o$, $Q_{w,1}$ and $Q_{w,o}$ are constant in (6.13)).
ii) As the term implies, the quantity of flow passing through the Unknown Outflow is an unknown, but dependent on the flow over the side-weir. In order to eliminate this unknown, one end of the virtual string is connected to the point of unknown outflow so that flow continuity holds:

\[ Q_d = Q_1 + Q_3 + Q_v \]  

(6.14)

where

\[ Q_d = \text{the discharge leaving the unknown outflow} \]
\[ Q_v = \text{the discharge through the virtual string} \]

It seems that the flow passing through the point of unknown outflow is \( Q_d \) but, in reality, the actual flow passing through is \( Q_1 + Q_3 \). This can be shown by subtracting \( Q_{w,o} \) from both sides of the equation

\[ Q_d - Q_{w,o} = Q_1 + Q_3 + Q_v - Q_{w,o} \]  

(6.15)

or

\[ Q_{\text{Outflow}} = (Q_1 + Q_3) + (Q_v - Q_{w,o}) \]  

(6.16)

in which \( Q_{\text{Outflow}} \) denotes the actual flow passing through the point of unknown outflow. The \( Q_{\text{Outflow}} \) is found if the following condition holds:

\[ Q_v = Q_{w,o} \]  

(6.17)

Thus, the virtual string connection between the arrow and the point of unknown outflow enforces flow continuity throughout the system.
PROBLEMS ARISING FROM NET INFLOW AND OUTFLOW

Figure 6.5a. A known constant inflow network.

Figure 6.5b. Equivalent network for the known constant inflow system.
With the virtual string inserted in the network, two closed loops are formed. The corresponding loop equations that satisfy both head-loss and flow continuity criterion are:

\[ f_1 = h_{f1} - h_{f2} - h_{f3} - h_{fw} \]
\[ f_2 = Q_v - Q_{w,o} \]

(6.18)

in which the \( f_2 \) is the virtual string characteristic, obtained by re-arranging (6.17). The \( f_2 \) measures the discrepancies between the actual flow \( Q_{w,o} \) over the side-weir and the assumed flow \( Q_v \). It follows that any value other than zero indicates the network is out-of-balance. However, it is considered more appropriate to rewrite (6.18) in the following form

\[ f_1 = h_{f1} - h_{f2} - h_{f3} - h_{fw} \]
\[ f_2 = \phi_v - \phi_{w,o} \]

(6.19)

where

\[ \phi_v = (Q_v - q_2)^e \]
\[ \phi_{w,o} = (Q_{w,o} - q_2)^e \]

for some exponent \( e \) to compensate certain degrees of convexity of the function (Fletcher, 1987). There are now two equations (6.19) with two unknowns \( q_1 \) and \( q_2 \), which is sufficient to obtain a unique network solution. The only problem associated with (6.19) is the choice of \( e \); and this will be investigated later in the chapter.

6.3.4 Constant Outflow

Consider the network problem shown in Figure 6.6a where the flow passing through to the outflow is constant. The flow over the side-weir is discharged independently of the network; this means that the total flow entering the network is unknown. The concept of virtual strings is once again applied in order to solve the problem. The approach to the solution is similar to that previously described in Section 6.3.3. A virtual string is inserted between the arrow and the point of unknown inflow. Consequently, two closed loops are formed as shown in Figure 6.6b.
PROBLEMS ARISING FROM NET INFLOW AND OUTFLOW

Figure 6.6a. A known constant outflow network.

Figure 6.6b. Equivalent network for the known constant outflow system.
Considering the head-losses around the closed loop, the loop equations can be written as

\begin{align}
  h_{f1} &= h_f((Q_1 + q_1), y_{1,1}, y_{1,2}) \\
  h_{f2} &= h_f((Q_{w,2} - q_1) + Q_{w,o}, y_{2,1}, y_{2,2}) \\
  h_{f3} &= h_f((Q_{w,2} - q_1), y_{3,1}, y_{3,2}) \\
  h_{fw} &= h_f((Q_{w,2} - q_1) + Q_{w,o}, (Q_{w,2} - q_1), y_{w,1}, y_{w,2}) \\
\end{align}

(6.20)

The (6.20) is written in terms of $Q_{w,2}$ rather than $Q_{w,1}$ because the flow passing through the point of constant outflow is fixed so that the $Q_{w,2}$ is regarded as a reference point ($Q_{w,1}$ is a reference point in the problem of constant inflow). The appearance of $Q_{w,o}$ in $h_{f2}$ and $h_{fw}$ must mean an additional loop is required to close both Channel-2 and the Side-Weir. With the additional loop, the above equations are re-written as

\begin{align}
  h_{f1} &= h_f((Q_i + q_1), y_{i,1}, y_{i,2}) \\
  h_{f2} &= h_f((Q_{w,2} - q_1) + (Q_{w,o} + q_2), y_{2,1}, y_{2,2}) \\
  h_{f3} &= h_f((Q_{w,2} - q_1), y_{3,1}, y_{3,2}) \\
  h_{fw} &= h_f((Q_{w,2} - q_1) + (Q_{w,o} + q_2), (Q_{w,2} - q_1), y_{w,1}, y_{w,2}) \\
\end{align}

(6.21)

(compare with (6.13)). By analogy to the problem of constant outflow, the unknown inflow is eliminated by writing the following equation

\begin{align}
  Q_i &= Q_1 + Q_2 - Q_v \\
\end{align}

(6.22)

where

- $Q_i$ = the discharge entering through the unknown inflow
- $Q_v$ = the discharge through the virtual string

$Q_{w,o}$ is added to both sides of (6.22) to give:

\begin{align}
  Q_{\text{inflow}} &= (Q_1 + Q_2) - (Q_v - Q_{w,o}) \\
\end{align}

(6.23)
in which $Q_{\text{inflow}} = Q_i + Q_{w,o}$ denotes the actual flow passing through the point of unknown inflow. Again, the $Q_{\text{inflow}}$ is found once the following condition holds

$$Q_v = Q_{w,o} \quad (6.24)$$

It follows that the loop equations can be written as:

$$f_1 = h_f_1 - h_f_2 - h_f_3 - h_f_w$$
$$f_2 = -(\phi_v - \phi_{w,o}) \quad (6.25)$$

The only difference between the (6.19b) and (6.25b) is the overall negative sign. It implies that the flow in the virtual string is clockwise as $q_2$; thus the flow in the virtual string increases with increasing flow in Channel-2 (see Figure 6.6b).

Despite the additional variable incurred in the side-weir, the insertion of a virtual string enables the additional variable to be eliminated. However, care must be taken about where to insert the virtual string within the network. Once the virtual string is inserted, the solution to the problem is direct and does not require any subsidiary external procedures to be incorporated.

### 6.4 Applications of Virtual String to Hydraulic Networks

A network which incorporates a side-weir can be analysed directly by inserting an appropriate virtual string. The solution process is identical to that described in Chapter 4 and Chapter 5. The basic structure for solving a particular network problem is outlined below:

i) Identify the problem (constant inflow or constant outflow) and insert the virtual string accordingly. For the problem of constant inflow, the virtual string is inserted between the side-weir and the point of system outflow. For the problems of constant outflow, the virtual string is inserted between side-weir and the constant inflow. If the flow over the side-weir is redirected elsewhere in the network, then a virtual string is not required.

ii) Trace all the independent loops throughout the system to set up the function $f(q)$. 

iii) Obtain an initial set of flow distributions which must satisfy continuity of flows at nodal junctions.

iv) Use BFGS method to solve for \( f(q) = 0 \), until \( \|e\| = 0 \). Line searches may also be used if necessary.

Note that the only step preceding the algorithm presented in Section 4.6 is Step i). The generalisation of the algorithm above allows various flow conditions to be tested. Three examples are investigated. The first provides numerical data for the known total inflow situation; the second deals with the opposite situation where the total outflow is given; and the third deals with the more complicated Type-2 flow where a hydraulic jump is formed downstream of the side-weir.

The basis of the investigation is to assess the convergence characteristics of network solution algorithm in which a virtual string is inserted. This includes the computational efforts required to achieve a solution and the effects of the exponent \( e \) on the overall performance. Before the investigation can begin, a number of assumptions are stated:

i) Minor head-losses such as eddy losses at nodal junctions are neglected.

ii) Zero flow does not occur at any point in the network.

iii) The Balmforth and Sarginson solution is used for side-weir calculations.

iv) The momentum coefficient \( \beta \) is assumed unity throughout.

v) Exponent \( e = 1 \) is used throughout.

vi) Line search parameter \( t_s = 1 \) is used throughout.

6.4.1 Known Constant Inflow

To illustrate the principle of using the virtual string to analyse the known constant inflow situation, the network shown in Figures 6.5 is used; its physical characteristics are shown in Table 6.4. It is understood that \( C_d \) is not a constant (Ackers et al, 1978) but a function of upstream headwater. As the \( C_d \) is unlikely to affect the investigation, a value of \( C_d = 0.75 \) is chosen for this example.

This example assumes a constant inflow of 4.0 \((\text{m}^3/\text{s})\) which enables side-weir flow Type-1 to be analysed. An initial distribution of flow is assumed as shown in Table 6.5a.
PROBLEMS ARISING FROM NET INFLOW AND OUTFLOW

This is obtained by splitting the constant inflow from the point of known constant inflow and proceeding downstream towards the point of unknown outflow. For example,

\[
\begin{align*}
Q_1 &= 2.0, \quad Q_2 = 4.0 - Q_1 = 2.0 \quad \text{and} \quad Q_{w1} = Q_2 = 2.0 \\
Q_3 &= 1.0, \quad Q_s = Q_2 - Q_3 = 1.0 \quad \text{and} \quad Q_{w2} = Q_3 = 1.0 \quad (\text{m}^3/\text{s})
\end{align*}
\]

The \(Q_{w,o} = 0.58638\) (m³/s) is the actual flow over the side weir calculated by (6.8) from \(Q_{w2} = 1.0\) (m³/s) and \(y_{w2} = 0.95135\) (m) (see below). As the initial distribution of flow is assumed, it is considered more appropriate to express the flow distribution in vector form below

\[
Q_{n=0} = \begin{bmatrix}
Q_1 \\
Q_2 \\
Q_3 \\
Q_{w2} \\
Q_s
\end{bmatrix} = \begin{bmatrix}
2.00000 \\
2.00000 \\
1.00000 \\
1.00000 \\
1.00000
\end{bmatrix}
\]

in which \(Q_n\) denotes the flow distribution for the \(n\)th iteration; and the \(n = 0\) denotes the initial distribution of flow.

With the \(Q_0\) assumed, the head-losses around the closed loops can be calculated. This is accomplished by calculating the water surface profile for each string starting from the point of unknown outflow and then proceeding upstream towards the point of known constant inflow. Once the water surface profiles are determined, it follows that the energy heads as well as the flow depths at nodal junctions are also obtained, and those are tabulated in Table 6.5a. Hence, the sum of head-losses around the closed loop are calculated as follows:

\[
f_0 = \begin{bmatrix}
H_1 - H_2 \\
Q_{w,o} - Q_s
\end{bmatrix} = \begin{bmatrix}
1.15313 - 1.09059 \\
0.58638 - 1.00000
\end{bmatrix} = \begin{bmatrix}
0.06253 \\
-0.41362
\end{bmatrix}
\]

Both \(Q_{w,o}\) and \(Q_s\) are involved in the above evaluation instead of energy heads, because it is assumed that the \(e\) in (6.19) is unity. Therefore, a direct substitution of \(Q_{w,o}\) and \(Q_s\) in (6.19) is possible. This implies that the relationship between \(Q_{w,o}\), \(Q_s\) and energy heads
are linear. Since the BFGS method described in Chapter 5 is the recommended solution method, it is used to solve the problem. This example assumes $H_0 = I$, and substituting $f_0$ calculated above into

$$q_0 = t_0 H_0 f_0$$

yields the loop flow correction for $n = 0$

$$q_0 = t_0 \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0.06253 \\ -0.41362 \end{bmatrix}$$

Putting $t_0 = 1$, the value of $q_0$ is evaluated to give

$$q_0 = \begin{bmatrix} 0.06253 \\ -0.41362 \end{bmatrix}$$

and hence a new set of flow distribution

$$Q_1 = \begin{bmatrix} 2.00000 - 0.06253 \\ 2.00000 + 0.06253 \\ 1.00000 + 0.06253 - 0.41362 \\ 1.00000 + 0.06253 + 0.41362 \end{bmatrix} = \begin{bmatrix} 1.93747 \\ 2.06253 \\ 1.47616 \\ 1.47676 \\ 0.58638 \end{bmatrix}$$

The procedure is continued for $n = 1$. With $Q_1$ calculated, the calculation procedure for $f_1$ is identical to that for $f_0$, and hence

$$q_1 = t_1 \begin{bmatrix} 1.02318 & -0.07882 & 0.03221 \\ -0.07882 & 1.02878 & -0.01390 \end{bmatrix}$$

Putting $t_1 = 1$,

$$q_1 = \begin{bmatrix} 0.03405 \\ -0.01683 \end{bmatrix}$$
and thence,

\[
Q_2 = \begin{bmatrix}
1.90341 \\
2.09659 \\
1.52704 \\
1.52704 \\
0.56954
\end{bmatrix}
\]

The procedure is continued until the condition \( \|f\| < \xi \) is met. These results together with those from subsequent iterations are tabulated in Tables 6.6.

It is seen from Table 6.6 that the solution algorithm converged reasonably uniformly during the course of iteration. This may be due to the \( H_o \) being a sufficiently accurate approximation of the inversion of the Jacobian matrix. Calculations for this example were also performed using NR, BR and DFP methods; the trends in convergence were similar to those described in Section 4.6 when the performance of each method was analysed in details. In fact, the NR, BR, DFP and BFGS methods converged to \( \chi_{j+1} < 10^{-4} \ (m^3/s) \) for \( j = 1, 2, \ldots, L \) in 3, 7, 8 and 7 iterations respectively. A summary of performance is given in Table 6.7.

Tables 6.5b shows the final flow distribution in the network along with energy heads and flow depths at the nodal junctions. It was found that the flow along the side-weir belongs to Type-1. This can be verified by calculating the specific energy at the upstream end of the side-weir, that is

\[
E_{w,1} = 1.023 < 2H_w \ (m).
\]

This implies that the flow is subcritical and has a falling water surface profile in the upstream direction. The flow in the virtual string obtained is identical to the actual flow over the side-weir which has a value of

\[
Q_v = Q_{w,s} = 0.57138 \ (m^3/s).
\]
### Table 6.4. Known Inflow Network - Physical characteristics, Type-1 flow.

<table>
<thead>
<tr>
<th>Strings</th>
<th>Height, $H_w$ (m)</th>
<th>Width, $B$ (m)</th>
<th>Length, $L$ (m)</th>
<th>Bed Slope, $S_b$</th>
<th>Bank Slope, $m$</th>
<th>Manning, $n$</th>
<th>Coeffic't, $C_d$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inflow</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Channel-1</td>
<td>-</td>
<td>2.00</td>
<td>250.0</td>
<td>0.00042</td>
<td>0.0</td>
<td>0.0130</td>
<td>-</td>
</tr>
<tr>
<td>Channel-2</td>
<td>-</td>
<td>2.00</td>
<td>100.0</td>
<td>0.00050</td>
<td>0.0</td>
<td>0.0130</td>
<td>-</td>
</tr>
<tr>
<td>Channel-3</td>
<td>-</td>
<td>2.00</td>
<td>100.0</td>
<td>0.00050</td>
<td>0.0</td>
<td>0.0130</td>
<td>-</td>
</tr>
<tr>
<td>Side-Weir</td>
<td>0.85</td>
<td>2.00</td>
<td>10.0</td>
<td>0.00050</td>
<td>0.0</td>
<td>0.0130</td>
<td>0.75</td>
</tr>
<tr>
<td>Outfall</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
### PROBLEMS ARISING FROM NET INFLOW AND OUTFLOW

<table>
<thead>
<tr>
<th>Strings</th>
<th>Initial Energy Head, $H$, (m)</th>
<th>Initial Water Depth, $y$, (m)</th>
<th>Initial Discharge, $Q$, (m³/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Upstream</td>
<td>Downstream</td>
<td>Upstream</td>
</tr>
<tr>
<td>Inflow</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Channel-1</td>
<td>1.15313</td>
<td>1.00000</td>
<td>0.99683</td>
</tr>
<tr>
<td>Channel-2</td>
<td>1.09059</td>
<td>1.01839</td>
<td>0.92618</td>
</tr>
<tr>
<td>Channel-3</td>
<td>1.01543</td>
<td>1.00000</td>
<td>0.95135</td>
</tr>
<tr>
<td>Side-Weir</td>
<td>1.01839</td>
<td>1.01543</td>
<td>0.92599</td>
</tr>
<tr>
<td>Overflow</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Outflow</td>
<td>1.00000</td>
<td>1.00000</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 6.5a. Known Inflow Network - Initial flow distribution, Type-1 flow.

<table>
<thead>
<tr>
<th>Strings</th>
<th>Final Energy Head, $H$, (m)</th>
<th>Final Water Depth, $y$, (m)</th>
<th>Final Discharge, $Q$, (m³/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Upstream</td>
<td>Downstream</td>
<td>Upstream</td>
</tr>
<tr>
<td>Inflow</td>
<td>1.12765</td>
<td>1.12765</td>
<td>-</td>
</tr>
<tr>
<td>Channel-1</td>
<td>1.12765</td>
<td>1.00000</td>
<td>0.97884</td>
</tr>
<tr>
<td>Channel-2</td>
<td>1.12765</td>
<td>1.04757</td>
<td>0.95609</td>
</tr>
<tr>
<td>Channel-3</td>
<td>1.04123</td>
<td>1.00000</td>
<td>0.95483</td>
</tr>
<tr>
<td>Side-Weir</td>
<td>1.04757</td>
<td>1.04123</td>
<td>0.92082</td>
</tr>
<tr>
<td>Overflow</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Outflow</td>
<td>1.00000</td>
<td>1.00000</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 6.5b. Known Inflow Network - Final flow distribution, Type-1 flow.
### Table 6.6. Known Inflow Network - Convergence characteristics, Type-1 flow.

| n  | $Q_1$, (m$^3$/s) | $Q_2$, (m$^3$/s) | $Q_3$, (m$^3$/s) | $Q_{o-o}$, (m$^3$/s) | $Q_r$, (m$^3$/s) | $||r_n||$, (m) |
|----|------------------|------------------|------------------|----------------------|------------------|----------------|
| 0  | 2.00000          | 2.00000          | 1.00000          | 0.58638              | 1.00000          | 4.18324E-01   |
| 1  | 1.93747          | 2.06253          | 1.47616          | 0.57248              | 0.58638          | 3.50794E-02   |
| 2  | 1.90341          | 2.09659          | 1.52704          | 0.57190              | 0.56954          | 2.28363E-02   |
| 3  | 1.83669          | 2.16331          | 1.59994          | 0.57142              | 0.56338          | 9.59919E-03   |
| 4  | 1.81767          | 2.18233          | 1.61327          | 0.57138              | 0.56906          | 2.39280E-03   |
| 5  | 1.81502          | 2.18498          | 1.61377          | 0.57138              | 0.57120          | 1.73776E-04   |
| 6  | 1.81494          | 2.18506          | 1.61369          | 0.57138              | 0.57137          | 5.52005E-06   |
| 7  | 1.81493          | 2.18507          | 1.61369          | 0.57138              | 0.57138          | 3.72745E-08   |
| 8  | 1.81493          | 2.18507          | 1.61369          | 0.57138              | 0.57138          | 1.72267E-11   |
| 9  | 1.81493          | 2.18507          | 1.61369          | 0.57138              | 0.57138          | 7.84251E-13   |
| 10 | 1.81493          | 2.18507          | 1.61369          | 0.57138              | 0.57138          | 9.26627E-14   |
| 11 | 1.81493          | 2.18507          | 1.61369          | 0.57138              | 0.57138          | 1.37708E-14   |
| 12 | 1.81493          | 2.18507          | 1.61369          | 0.57138              | 0.57138          | 1.93892E-15   |
| 13 | 1.81493          | 2.18507          | 1.61369          | 0.57138              | 0.57138          | 1.86107E-15   |
| 14 | 1.81493          | 2.18507          | 1.61369          | 0.57138              | 0.57138          | 2.22045E-16   |
| 15 | 1.81493          | 2.18507          | 1.61369          | 0.57138              | 0.57138          | 2.22045E-16   |
| 16 | 1.81493          | 2.18507          | 1.61369          | 0.57138              | 0.57138          | 1.11022E-16   |
| 17 | 1.81493          | 2.18507          | 1.61369          | 0.57138              | 0.57138          | 0            |
| Methods | $n$ | $N_t$ | $|q_0 - \hat{q}|$ | $|q_n - \hat{q}|$ | $\eta_\delta$ |
|---------|----|------|-----------------|-----------------|-----------|
| NR      | 3  | 9    | 7.93130E-01     | 1.00000E-09     | 2.277     |
| BR      | 7  | 7    | 7.93130E-01     | 5.83095E-07     | 2.018     |
| DFP     | 8  | 8    | 7.93130E-01     | 2.89137E-06     | 1.565     |
| BFGS    | 7  | 7    | 7.93130E-01     | 1.00000E-07     | 2.269     |

Table 6.7. Overall performance of various solution methods.
6.4.2 Known Constant Outflow

To illustrate the principle of using virtual string to analyse the known constant outflow situation, a numerical example shown in Figures 6.6 is used, and the physical characteristics are shown in Table 6.8. As previously, $C_d = 0.75$ is assumed to be a constant.

This example assumes a constant outflow of 2.0 (m³/s); which enables side-weir flow Type-1 to be analysed. An initial distribution of flow is assumed as shown in Table 6.9a. This is obtained by splitting the constant outflow from the point of known constant outflow and then proceeding upstream towards the point of unknown inflow. For example,

$$Q_1 = 1.0, \quad Q_3 = 2.0 - Q_1 = 1.0, \quad Q_{w,2} = Q_4 = 1.0$$

and

$$Q_2 = 2.0, \quad Q_4 = Q_2 - Q_3 = 1.0, \quad Q_{w,1} = Q_2 = 1.0 \quad (m^3/s)$$

The $Q_{w,o} = 0.55169$ (m³/s) is the actual flow over the side-weir calculated by (6.8) from $Q_{w,2} = 1.0$ (m³/s) and $y_{w,2} = 0.94746$ (m) (see below). With the initial distribution of flow assumed as shown below, expressed by the vector $Q_0$

$$Q_0 = \begin{bmatrix} Q_1 \\ Q_2 \\ Q_3 \\ Q_{w,2} \\ Q_w \end{bmatrix} = \begin{bmatrix} 1.00000 \\ 2.00000 \\ 1.00000 \\ 1.00000 \\ 1.00000 \end{bmatrix}$$

the head-losses around the closed loops can be calculated. This is accomplished by calculating the water surface profile for each string starting from the point of unknown outflow and then proceeds upstream towards the point of known constant inflow. Once the water surface profiles are determined, it follows that the energy heads as well as the flow depths at nodal junctions are also obtained, and those are tabulated in Table 6.9a. Hence, the sum of head-losses around the closed loop are calculated as follows:
PROBLEMS ARISING FROM NET INFLOW AND OUTFLOW

\[ f_0 = \begin{bmatrix} H_1 - H_2 \\ Q_v - Q_{w,o} \end{bmatrix} = \begin{bmatrix} 1.03078 - 1.07047 \\ 1.00000 - 0.55169 \end{bmatrix} = \begin{bmatrix} -0.03969 \\ 0.44831 \end{bmatrix} \]

Since \( e \) in (6.25) is assumed unity, \( Q_{w,o}, Q_v \) and energy head are therefore linearly related. Hence, both \( Q_{w,o} \) and \( Q_v \) can be directly substituted in (6.25) in order to complete the calculations. This example assumes \( H_0 = I \), and substituting \( f_0 \) calculated above into

\[ q_0 = t_0 H_0 f_0 \]

yields the loop flow correction for \( n = 0 \)

\[ q_0 = t_0 \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} -0.03969 \\ 0.44831 \end{bmatrix} \]

Putting \( t_0 = 1 \), the value of \( q_0 \) is evaluated to give

\[ q_0 = \begin{bmatrix} -0.03969 \\ 0.44831 \end{bmatrix} \]

and hence, a new set of flow distributions

\[ Q_1 = \begin{bmatrix} 1.00000 + 0.03969 \\ 2.00000 - 0.03969 - 0.44831 \\ 1.00000 - 0.03969 \\ 1.00000 - 0.44831 \end{bmatrix} = \begin{bmatrix} 1.03969 \\ 1.51200 \\ 0.96031 \\ 0.55169 \end{bmatrix} \]

The procedure is continued for \( n = 1 \). With \( Q_1 \) calculated, the calculation procedure for \( f_1 \) is identical to that for \( f_0 \), and hence

\[ q_1 = t_1 \begin{bmatrix} 1.00418 & -0.02311 \\ -0.02311 & 0.98906 \end{bmatrix} \begin{bmatrix} -0.01058 \\ -0.00428 \end{bmatrix} \]
PROBLEMS ARISING FROM NET INFLOW AND OUTFLOW

Putting \( t_1 = 1 \),

\[
q_1 = \begin{bmatrix}
-0.10153 \\
-0.00399
\end{bmatrix}
\]

and thence,

\[
Q_2 = \begin{bmatrix}
1.05022 \\
1.50546 \\
0.94978 \\
0.94978 \\
0.55568
\end{bmatrix}
\]

The procedure is continued until the condition \( \| f \| < \xi \) is met. These results together with those from subsequent iterations are tabulated in Tables 6.10.

The solution algorithm converges reasonably uniformly except for the oscillatory behaviour when near the solution. This was caused by the loop flow corrections being over-estimated by the BFGS updating formula. Calculations were also performed using NR, BR and DFP methods; the trends in convergence were similar to those described in Section 4.6 when the performance of each method was analysed in details. In fact, the NR, BR, DFP and BFGS methods converged to \( \chi_{j+1} < 10^{-4} \) (m³/s) for \( j = 1, 2, \ldots, L \) in 3, 6, 8 and 7 iterations respectively. Note that this example converged to \( \chi_{j+1} \) in 7 iterations, the same as that obtained in Section 6.4.1. A summary of performance is given in Table 6.11.

Tables 6.9b shows the final flow distribution in the network along with energy heads and flow depths at the nodal junctions. It was found that the flow along the side-weir belongs to Type-1. This can be verified by calculating the specific energy of at the upstream end of the side-weir, that is,

\[
E_{u,1} = 0.907 < 2H_w \text{ (m)}
\]
### Table 6.8. Known Outflow Network - Physical characteristics, Type-1 flow.

<table>
<thead>
<tr>
<th>Strings</th>
<th>Height, $H_w$ (m)</th>
<th>Width, $B$ (m)</th>
<th>Length, $L$ (m)</th>
<th>Bed Slope, $S_0$</th>
<th>Bank Slope, $m$</th>
<th>Manning, $n$</th>
<th>Coeffic't, $C_d$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inflow</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Channel-1</td>
<td>-</td>
<td>2.00</td>
<td>250.0</td>
<td>0.00042</td>
<td>0.0</td>
<td>0.0130</td>
<td>-</td>
</tr>
<tr>
<td>Channel-2</td>
<td>-</td>
<td>2.00</td>
<td>100.0</td>
<td>0.00050</td>
<td>0.0</td>
<td>0.0130</td>
<td>-</td>
</tr>
<tr>
<td>Channel-3</td>
<td>-</td>
<td>2.00</td>
<td>100.0</td>
<td>0.00050</td>
<td>0.0</td>
<td>0.0130</td>
<td>-</td>
</tr>
<tr>
<td>Side-Weir</td>
<td>0.85</td>
<td>2.00</td>
<td>10.0</td>
<td>0.00050</td>
<td>0.0</td>
<td>0.0130</td>
<td>0.75</td>
</tr>
<tr>
<td>Outflow</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
PROBLEMS ARISING FROM NET INFLOW AND OUTFLOW

<table>
<thead>
<tr>
<th>Strings</th>
<th>Initial Energy Head, $H$, (m)</th>
<th>Initial Water Depth, $y$, (m)</th>
<th>Initial Discharge, $Q$, (m³/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Upstream</td>
<td>Downstream</td>
<td>Upstream</td>
</tr>
<tr>
<td>Inflow</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Channel-1</td>
<td>1.03078</td>
<td>1.00000</td>
<td>0.91040</td>
</tr>
<tr>
<td>Channel-2</td>
<td>1.07047</td>
<td>1.01383</td>
<td>0.90296</td>
</tr>
<tr>
<td>Channel-3</td>
<td>1.01165</td>
<td>1.00000</td>
<td>0.94746</td>
</tr>
<tr>
<td>Side-Weir</td>
<td>1.01383</td>
<td>1.01165</td>
<td>0.92281</td>
</tr>
<tr>
<td>Overflow</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Outflow</td>
<td>1.00000</td>
<td>1.00000</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 6.9a. Known Outflow Network - Initial flow distribution, Type-1 flow.

<table>
<thead>
<tr>
<th>Strings</th>
<th>Final Energy Head, $H$, (m)</th>
<th>Final Water Depth, $y$, (m)</th>
<th>Final Discharge, $Q$, (m³/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Upstream</td>
<td>Downstream</td>
<td>Upstream</td>
</tr>
<tr>
<td>Inflow</td>
<td>1.03889</td>
<td>1.03889</td>
<td>-</td>
</tr>
<tr>
<td>Channel-1</td>
<td>1.03889</td>
<td>1.00000</td>
<td>0.91462</td>
</tr>
<tr>
<td>Channel-2</td>
<td>1.03889</td>
<td>1.01068</td>
<td>0.90135</td>
</tr>
<tr>
<td>Channel-3</td>
<td>1.00888</td>
<td>1.00000</td>
<td>0.94802</td>
</tr>
<tr>
<td>Side-Weir</td>
<td>1.01068</td>
<td>1.00888</td>
<td>0.92477</td>
</tr>
<tr>
<td>Overflow</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Outflow</td>
<td>1.00000</td>
<td>1.00000</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 6.9b. Known Outflow Network - Final flow distribution, Type-1 flow.
### Problems Arising from Net Inflow and Outflow

#### Broyden-Fletcher-Goldfarb-Shanno

| $n$ | $Q_1$ (m³/s) | $Q_2$ (m³/s) | $Q_3$ (m³/s) | $Q_{x'}$ (m³/s) | $Q_4$ (m³/s) | $||r_0||$, (m) |
|-----|--------------|--------------|--------------|----------------|--------------|----------------|
| 0   | 1.00000      | 2.00000      | 1.00000      | 0.55169        | 1.00000      | 4.50061E-01    |
| 1   | 1.03969      | 1.51200      | 0.96031      | 0.55597        | 0.55169      | 1.14124E-02    |
| 2   | 1.05022      | 1.50546      | 0.94978      | 0.55710        | 0.55568      | 9.48577E-03    |
| 3   | 1.11173      | 1.45462      | 0.88827      | 0.56371        | 0.56634      | 3.16937E-03    |
| 4   | 1.12274      | 1.44280      | 0.87726      | 0.56490        | 0.56554      | 7.04514E-04    |
| 5   | 1.12467      | 1.44047      | 0.87533      | 0.56510        | 0.56514      | 3.84935E-05    |
| 6   | 1.12477      | 1.44034      | 0.87523      | 0.56511        | 0.56511      | 3.20932E-07    |
| 7   | 1.12477      | 1.44034      | 0.87523      | 0.56511        | 0.56511      | 3.09138E-10    |
| 8   | 1.12477      | 1.44034      | 0.87523      | 0.56511        | 0.56511      | 1.53048E-11    |
| 9   | 1.12477      | 1.44034      | 0.87523      | 0.56511        | 0.56511      | 1.38704E-12    |
| 10  | 1.12477      | 1.44034      | 0.87523      | 0.56511        | 0.56511      | 4.55474E-13    |
| 11  | 1.12477      | 1.44034      | 0.87523      | 0.56511        | 0.56511      | 3.11061E-15    |
| 12  | 1.12477      | 1.44034      | 0.87523      | 0.56511        | 0.56511      | 4.78299E-15    |
| 13  | 1.12477      | 1.44034      | 0.87523      | 0.56511        | 0.56511      | 4.55191E-15    |
| 14  | 1.12477      | 1.44034      | 0.87523      | 0.56511        | 0.56511      | 6.91843E-14    |
| 15  | 1.12477      | 1.44034      | 0.87523      | 0.56511        | 0.56511      | 9.76996E-15    |
| 16  | 1.12477      | 1.44034      | 0.87523      | 0.56511        | 0.56511      | 4.88624E-15    |
| 17  | 1.12477      | 1.44034      | 0.87523      | 0.56511        | 0.56511      | 4.88624E-15    |
| 18  | 1.12477      | 1.44034      | 0.87523      | 0.56511        | 0.56511      | 0            |

Table 6.10. Known Outflow Network - Convergence characteristics, Type-1 flow.
Table 6.11. Overall performance of various solution methods.
The flow in the virtual string obtained is identical to the actual flow over the side-weir which has a value of

\[ Q_v = Q_{w,o} = 0.56511 \text{ (m}^3\text{/s)} \]

Both of these examples have illustrated the importance of the virtual string concept as a mechanism for solving networks in which the outflow from certain modules cannot be specified a priori. The string provides a mean of making a connection in order to maintain flow continuity in the network. It enables the existing network solution algorithms to be used, rather than having to resort to external trial-and-error procedures which would be extremely cumbersome. The virtual string incurs an additional loop equation; in the examples used the solution was found without any difficulty.

### 6.4.3 Hydraulic Jump

To test whether the Type-2 flow can be analysed by the proposed procedures, the network shown in Figures 6.6 is again used. The physical characteristics of the network are shown in Table 6.12. This set of data was chosen to investigate side-weir Type-2 flow. If the flow belongs to Type-2, then a critical depth control will be established at the upstream end of the side-weir, and the flow becomes supercritical. Consequently, a hydraulic jump may form in the downstream channel, depending on the channel characteristics.

When the flow approaching the side-weir is subcritical, the water surface is drawn down to a water depth \(0.8y_e < y_{w,1} < y_e\) (Balmforth and Sarginson, 1978) at the upstream end of the side-weir. For the purposes of this investigation, the water depth at the control section is assumed to be critical.

The flow along the side-weir is assumed supercritical with a constant outflow of \(4.0 \text{ (m}^3\text{/s)}\). An initial distribution of flow is assumed and shown in Table 6.13a. This is obtained by splitting the constant outflow from the point of known constant outflow and then proceeding upstream towards the point of unknown inflow. The \(Q_{w,o} = 0.50769 \text{ (m}^3\text{/s)}\) is the actual flow over the side-weir calculated by (6.8) from \(Q_{w,1} = 2.75769 \text{ (m}^3\text{/s)}\) and the critical depth \(y_{w,1} = 0.57870 \text{ (m)}\). The values of \(Q_{w,1}\) and \(y_{w,1}\) are obtained to satisfy \(Q_{w,2} = 2.25 \text{ (m}^3\text{/s)}\).
With the initial flow distribution assumed, the computation can begin. The procedure used to calculate the head-losses around the closed loops is identical to that for the two previous examples except that the calculation starts from the upstream end of the side-weir, since this is a critical flow control section. The energy heads and the flow depths at nodal junctions are obtained, and are tabulated in Table 6.13a.

The procedure (BFGS) used to obtain the solution is identical to that outlined in Section 6.4.2. A complete run on the network was conducted, and the final flow distribution is shown in Table 6.13b. It was found that the flow along the side-weir belongs to Type-2. This can be verified by calculating the specific energy at the upstream end of the side-weir, that is

\[ E_{w,1} = 0.776 > 2H_w \text{ (m)} \]

It implies that the flow along the side-weir has a falling water surface in the downstream direction. The flow in the virtual string obtained was identical to the actual flow over the side-weir, namely

\[ Q_v = Q_{w,o} = 0.33742 \text{ (m}^3/\text{s}). \]

The highlighted areas in Table 6.13b denote a discrepancy in energy heads of 0.00472 (m). It was due to the head-loss caused by the hydraulic jump formed in Channel-3. The situation is analysed below.

The flow in Channel-2 was subcritical and its water surface was drawn down to the critical depth of

\[ y_{2,2} = y_{w,1} = 0.51753 \text{ (m)} \]

when approaching the upstream end of the side-weir. The flow along the side-weir was supercritical until the end of the side-weir. It was impossible for the Channel-3 to sustain the supercritical flow because the slope of Channel-3 is less than the critical slope of \( S_c = 0.00356 \) so that the flow in this channel must be subcritical. A further check on the Froude number can confirm this finding. The Froude numbers just upstream of the
downstream end of the side-weir and downstream of the upstream end of Channel-3 are calculated respectively to be

\[ Fr_{w,2}^2 = 1.69562 > 1 \]
\[ Fr_{3,1}^2 = 0.60725 < 1 \]

Therefore, the flow regimes in the side-weir and Channel-3 were supercritical and subcritical respectively. This change in flow regime is achieved through the agency of a hydraulic jump. Because \( 1.0 < Fr_{w,2}^2 < 1.7 \), the jump is classified as an undular jump (Chow, 1959).

To calculate the head-loss \( h_j \) caused by the jump, it is necessary to determine whether the jump is stable. Denoting the sequent depth by \( y_{sequent} \) (the water depth after the hydraulic jump) as shown in Figure 6.7, then the following observations may be made

- if \( y_{3,1} > y_{sequent} \), then the jump moves upstream
- if \( y_{3,1} < y_{sequent} \), then the jump moves downstream
- if \( y_{3,1} = y_{sequent} \), then a stable jump forms

The \( y_{sequent} \) is given by

\[ y_{sequent} = \frac{y_{w,2}}{2} \left( \sqrt{1 + 8 Fr_{w,2}^2} - 1 \right) \]

(Chow, 1959). In this case

\[ y_{sequent} = \frac{0.39107}{2} \left( \sqrt{1 + 8(1.69562)} - 1 \right) = 0.55070 \text{ (m)} \]

Thus, \( y_{sequent} \) is identical with \( y_{3,1} \) (see Table 6.13b), which means that a stable jump is formed. Hence, the \( h_j \) is given by

\[ h_j = \frac{(y_{3,1} - y_{w,2})^3}{4y_{w,2}y_{3,1}} \]
(Chow, 1959), in which $y_{3,i}$ is the water depth just upstream of Channel-3. Hence,

$$h_g = \frac{(0.55070 - 0.39107)^3}{4(0.55070)(0.39107)} = 0.00472 \text{ (m)}$$

The above calculated results agree with those found in Table 6.13b.

The convergence characteristics of the BFGS method may be judged by examining Tables 6.14. It converged to $x_{j+1} < 10^{-4}$ (m$^3$/s) for $j = 1, 2, \ldots, L$ in 8 iterations. Again, there is little to distinguish from previous examples (see Sections 6.4.1 and 6.4.2) except there was a very rapid rate of convergence towards the solution. In fact, the convergence obtained was second-order. This convergence behaviour can only be exhibited by the NR method. It means that the $H_a$ was approximated very close to the inversion of the Jacobian matrix. This behaviour was not observed in previous examples including those in Chapter 5. It may be due to relatively high head-loss incurred in the network so that the energy head was sensitive to the changes made in the flow. The head-loss was found approximately to be 0.267 (m) as compared with those 0.153 (m) and 0.039 (m) obtained in Sections 6.4.1 and 6.4.2 respectively.

The network solution algorithm containing a virtual string is capable of solving a situation which contains an hydraulic discontinuity in the form of an hydraulic jump. Whilst only one example has been analysed, it is encouraging that such a complex system containing a side-weir (for which the outflow cannot be determined a priori) and involving an hydraulic jump can be analysed and a solution obtained. The above analysis illustrates that the discontinuity in energy level caused by the hydraulic jump can be dealt with by the solution procedure without creating obvious difficulties either in convergence or obtaining a final solution.
## Problems Arising from Net Inflow and Outflow

### Table 6.12. Known Outflow Network - Physical characteristics, Type-2 flow.

<table>
<thead>
<tr>
<th>Strings</th>
<th>Height, ( H_w ) (m)</th>
<th>Width, ( B_w ) (m)</th>
<th>Length, ( L ) (m)</th>
<th>Bed Slope, ( S_0 )</th>
<th>Bank Slope, ( m )</th>
<th>Manning, ( n )</th>
<th>Coeffic'nt, ( C_d )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inflow</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Channel-1</td>
<td>-</td>
<td>2.00</td>
<td>250.0</td>
<td>0.00042</td>
<td>0.0</td>
<td>0.0130</td>
<td>-</td>
</tr>
<tr>
<td>Channel-2</td>
<td>-</td>
<td>2.00</td>
<td>100.0</td>
<td>0.00050</td>
<td>0.0</td>
<td>0.0130</td>
<td>-</td>
</tr>
<tr>
<td>Channel-3</td>
<td>-</td>
<td>2.00</td>
<td>20.0</td>
<td>0.00250</td>
<td>0.0</td>
<td>0.0130</td>
<td>-</td>
</tr>
<tr>
<td>Side-Weir</td>
<td>0.35</td>
<td>2.00</td>
<td>10.0</td>
<td>0.00050</td>
<td>0.0</td>
<td>0.0130</td>
<td>0.75</td>
</tr>
<tr>
<td>Outfall</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 6.12. Known Outflow Network - Physical characteristics, Type-2 flow.
### Problems Arising from Net Inflow and Outflow

<table>
<thead>
<tr>
<th>Strings</th>
<th>Initial Energy Head, $H$, (m)</th>
<th>Initial Water Depth, $y$, (m)</th>
<th>Initial Discharge, $Q$, (m$^3$/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Upstream</td>
<td>Downstream</td>
<td>Upstream</td>
</tr>
<tr>
<td>Inflow</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Channel-1</td>
<td>0.97014</td>
<td>0.80627</td>
<td>0.80491</td>
</tr>
<tr>
<td>Channel-2</td>
<td>1.20072</td>
<td>1.02351</td>
<td>0.94501</td>
</tr>
<tr>
<td>Channel-3</td>
<td>0.84246</td>
<td>0.80627</td>
<td>0.62987</td>
</tr>
<tr>
<td>Side-Weir</td>
<td>0.92305</td>
<td>0.85482</td>
<td>0.57870</td>
</tr>
<tr>
<td>Overflow</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Outflow</td>
<td>0.80627</td>
<td>0.80627</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 6.13a. Known Outflow Network - Initial flow distribution, Type-2 flow.

<table>
<thead>
<tr>
<th>Strings</th>
<th>Final Energy Head, $H$, (m)</th>
<th>Final Water Depth, $y$, (m)</th>
<th>Final Discharge, $Q$, (m$^3$/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Upstream</td>
<td>Downstream</td>
<td>Upstream</td>
</tr>
<tr>
<td>Inflow</td>
<td>0.99256</td>
<td>0.72571</td>
<td>-</td>
</tr>
<tr>
<td>Channel-1</td>
<td>0.99256</td>
<td>0.72571</td>
<td>0.80936</td>
</tr>
<tr>
<td>Channel-2</td>
<td>0.99256</td>
<td>0.83131</td>
<td>0.77095</td>
</tr>
<tr>
<td>Channel-3</td>
<td>0.76790</td>
<td>0.72571</td>
<td>0.55070</td>
</tr>
<tr>
<td>Side-Weir</td>
<td>0.83131</td>
<td>0.77262</td>
<td>0.51754</td>
</tr>
<tr>
<td>Overflow</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Outflow</td>
<td>0.72571</td>
<td>0.72571</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 6.13b. Known Outflow Network - Final flow distribution, Type-2 flow.
### Broyden-Fletcher-Goldfarb-Shanno

<table>
<thead>
<tr>
<th>n</th>
<th>$Q_1,$ (m³/s)</th>
<th>$Q_2,$ (m³/s)</th>
<th>$Q_3,$ (m³/s)</th>
<th>$Q_{e,0},$ (m³/s)</th>
<th>$Q_e,$ (m³/s)</th>
<th>$|f_e|,$ (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1.75000</td>
<td>3.25000</td>
<td>2.25000</td>
<td>0.50769</td>
<td>1.00000</td>
<td>5.43633E-01</td>
</tr>
<tr>
<td>1</td>
<td>1.98058</td>
<td>2.52711</td>
<td>2.01942</td>
<td>0.35209</td>
<td>0.50769</td>
<td>1.63445E-01</td>
</tr>
<tr>
<td>2</td>
<td>2.05256</td>
<td>2.23196</td>
<td>1.94744</td>
<td>0.31021</td>
<td>0.28452</td>
<td>4.09796E-02</td>
</tr>
<tr>
<td>3</td>
<td>2.02312</td>
<td>2.29239</td>
<td>1.97688</td>
<td>0.32693</td>
<td>0.31551</td>
<td>1.66442E-02</td>
</tr>
<tr>
<td>4</td>
<td>2.00422</td>
<td>2.33482</td>
<td>1.99578</td>
<td>0.33797</td>
<td>0.33903</td>
<td>1.31321E-03</td>
</tr>
<tr>
<td>5</td>
<td>2.00543</td>
<td>2.33169</td>
<td>1.99457</td>
<td>0.33725</td>
<td>0.33711</td>
<td>2.31387E-04</td>
</tr>
<tr>
<td>6</td>
<td>2.00520</td>
<td>2.33214</td>
<td>1.99480</td>
<td>0.33738</td>
<td>0.33734</td>
<td>5.75504E-05</td>
</tr>
<tr>
<td>7</td>
<td>2.00513</td>
<td>2.33229</td>
<td>1.99487</td>
<td>0.33742</td>
<td>0.33743</td>
<td>7.87274E-06</td>
</tr>
<tr>
<td>8</td>
<td>2.00514</td>
<td>2.33228</td>
<td>1.99486</td>
<td>0.33742</td>
<td>0.33742</td>
<td>1.36989E-06</td>
</tr>
<tr>
<td>9</td>
<td>2.00514</td>
<td>2.33227</td>
<td>1.99486</td>
<td>0.33742</td>
<td>0.33742</td>
<td>1.28308E-06</td>
</tr>
<tr>
<td>10</td>
<td>2.00514</td>
<td>2.33227</td>
<td>1.99486</td>
<td>0.33742</td>
<td>0.33742</td>
<td>2.18923E-07</td>
</tr>
<tr>
<td>11</td>
<td>2.00514</td>
<td>2.33227</td>
<td>1.99486</td>
<td>0.33742</td>
<td>0.33742</td>
<td>1.25163E-07</td>
</tr>
<tr>
<td>12</td>
<td>2.00514</td>
<td>2.33227</td>
<td>1.99486</td>
<td>0.33742</td>
<td>0.33742</td>
<td>2.20063E-08</td>
</tr>
<tr>
<td>13</td>
<td>2.00514</td>
<td>2.33227</td>
<td>1.99486</td>
<td>0.33742</td>
<td>0.33742</td>
<td>2.28755E-09</td>
</tr>
<tr>
<td>14</td>
<td>2.00514</td>
<td>2.33227</td>
<td>1.99486</td>
<td>0.33742</td>
<td>0.33742</td>
<td>3.56155E-11</td>
</tr>
<tr>
<td>15</td>
<td>2.00514</td>
<td>2.33227</td>
<td>1.99486</td>
<td>0.33742</td>
<td>0.33742</td>
<td>5.32907E-15</td>
</tr>
<tr>
<td>16</td>
<td>2.00514</td>
<td>2.33227</td>
<td>1.99486</td>
<td>0.33742</td>
<td>0.33742</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 6.14. Known Inflow Network - Convergence characteristics, Type-2 flow.
Figure 6.7. A hydraulic jump is formed at downstream because of Type-2 flow.
6.4.4 The Choice of Exponent $e$

Already shown in Section 6.3, the virtual string is a linear function of flows (6.8). However, the requirement for convergence is that the function must be convex; otherwise, the BFGS method may have difficulties converging. To compensate for the inadequacy in convexity of the function, the most direct modification is to raise the function to some power $e$. Although $e=1$ arises naturally in the derivation, it is advisable to investigate some other values to see if the convergence can be accelerated.

The network problems presented in Sections 6.4.1 and 6.4.2 were used for this investigation. A total of eighteen computer runs were conducted for the examples using various values of $e$ ranging between $0.2 \leq e \leq 1.8$. However, it is considered more appropriate to express the results in terms of performance efficiency. The calculations were performed and the corresponding results are tabulated in Tables 6.15 and plotted in Figure 6.8. It must be noted that the results were calculated based on the criteria $x_{j+1} < 10^{-4} \text{ (m}^3/\text{s)}$ for $j=1,2,\ldots,L$, as before.

The overall performance varied with the value of $e$ as follows.

i) Both examples showed similar trends for the mean rate of convergence as a function of $e$.

ii) The rate of convergence varied between 1.894 and 0.997; the corresponding number of iterations were found to be 5 and 10 respectively. These figures suggest that a 50% reduction in computational efforts can result if the value of $e$ is suitably chosen. The choice of $e$ is critical to overall performance implying that the BFGS method is sensitive to the choice of $e$.

iii) There is a trend of increasing rate of convergence with increasing $e$, up to an optimum value of $e$ between $0.8 < e < 1.2$; $e=1$ gave the best convergence rate for the two examples. From these results, and since $e=1$ arises naturally in the derivation, the choice of $e=1$ is recommended.
Table 6.1a. Known Inflow Network - Mean convergence rate, Type-1 flow.

<table>
<thead>
<tr>
<th>Exponent, $e$</th>
<th>Iteration, $n$</th>
<th>$N_f$</th>
<th>$|q_0 - \hat{q}|$</th>
<th>$|q_n - \hat{q}|$</th>
<th>$\eta_\delta$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.20</td>
<td>9</td>
<td>9</td>
<td>4.66870E-01</td>
<td>1.62890E-05</td>
<td>1.140</td>
</tr>
<tr>
<td>0.40</td>
<td>7</td>
<td>7</td>
<td>4.66870E-01</td>
<td>4.16850E-05</td>
<td>1.332</td>
</tr>
<tr>
<td>0.60</td>
<td>8</td>
<td>8</td>
<td>4.66870E-01</td>
<td>1.60590E-05</td>
<td>1.285</td>
</tr>
<tr>
<td>0.80</td>
<td>5</td>
<td>5</td>
<td>4.66870E-01</td>
<td>3.60139E-05</td>
<td>1.894</td>
</tr>
<tr>
<td>1.00</td>
<td>6</td>
<td>6</td>
<td>4.66870E-01</td>
<td>5.94643E-06</td>
<td>1.879</td>
</tr>
<tr>
<td>1.20</td>
<td>7</td>
<td>7</td>
<td>4.66870E-01</td>
<td>5.15980E-05</td>
<td>1.301</td>
</tr>
<tr>
<td>1.40</td>
<td>8</td>
<td>8</td>
<td>4.66870E-01</td>
<td>1.02000E-05</td>
<td>1.341</td>
</tr>
<tr>
<td>1.60</td>
<td>7</td>
<td>7</td>
<td>4.66870E-01</td>
<td>5.83398E-05</td>
<td>1.284</td>
</tr>
<tr>
<td>1.80</td>
<td>6</td>
<td>6</td>
<td>4.66870E-01</td>
<td>1.50283E-05</td>
<td>1.724</td>
</tr>
</tbody>
</table>

Table 6.1b. Known Outflow Network - Mean Convergence rate, Type-1 flow.

<table>
<thead>
<tr>
<th>Exponent, $e$</th>
<th>Iteration, $n$</th>
<th>$N_f$</th>
<th>$|q_0 - \hat{q}|$</th>
<th>$|q_n - \hat{q}|$</th>
<th>$\eta_\delta$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.20</td>
<td>10</td>
<td>10</td>
<td>4.52430E-01</td>
<td>2.12382E-05</td>
<td>0.997</td>
</tr>
<tr>
<td>0.40</td>
<td>10</td>
<td>10</td>
<td>4.52430E-01</td>
<td>9.22025E-06</td>
<td>1.080</td>
</tr>
<tr>
<td>0.60</td>
<td>9</td>
<td>9</td>
<td>4.52430E-01</td>
<td>2.59894E-06</td>
<td>1.341</td>
</tr>
<tr>
<td>0.80</td>
<td>8</td>
<td>8</td>
<td>4.52430E-01</td>
<td>2.52376E-06</td>
<td>1.512</td>
</tr>
<tr>
<td>1.00</td>
<td>7</td>
<td>7</td>
<td>4.52430E-01</td>
<td>9.21954E-07</td>
<td>1.872</td>
</tr>
<tr>
<td>1.20</td>
<td>9</td>
<td>9</td>
<td>4.52430E-01</td>
<td>1.95357E-06</td>
<td>1.373</td>
</tr>
<tr>
<td>1.40</td>
<td>9</td>
<td>9</td>
<td>4.52430E-01</td>
<td>2.12497E-06</td>
<td>1.363</td>
</tr>
<tr>
<td>1.60</td>
<td>9</td>
<td>9</td>
<td>4.52430E-01</td>
<td>3.55434E-06</td>
<td>1.306</td>
</tr>
<tr>
<td>1.80</td>
<td>11</td>
<td>11</td>
<td>4.52430E-01</td>
<td>2.58540E-06</td>
<td>1.098</td>
</tr>
</tbody>
</table>

269
Figure 6.8. Exponent $e$ for virtual string characteristics.
6.5 Concluding Remarks

Constant inflow and outflow can be dealt with satisfactorily by the methods investigated for network analysis. However, a particular problem is encountered with certain types of overflow structures, such as side discharging weirs. The difficulty is that the outflow (that is, the overflow or side-discharge over the weir) cannot be determined a priori. It is only evaluated as a result of detailed calculation during an iteration of the network analysis procedure. Thus, the problem associated with a side-weir structure is that it introduces an additional variable, namely the flow over the side-weir, in the system equations. Consequently, the set of equations is not complete and that cannot apparently be solved without recourse to some subsidiary trial-and-error procedures.

An alternative approach which defines a 'virtual string' is investigated. The virtual string is no different to any other string except that its flow characteristics are solely determined to maintain flow continuity throughout the system. Two distinct situations arise, which have been defined as either 'constant inflow' or 'constant outflow', and relate to whether the total inflow or total outflow to the network is predetermined. Three network problems which included Type-1 and Type-2 side-weir flows and a situation in which a hydraulic jump occurred were designed to investigate the effectiveness of this concept. The main factors concerned are solvability and convergence.

The technique was applied successfully to the network problems studied. The rate of convergence obtained for the examples were similar. The criteria \( \chi_{j+m} < 10^{-4} \text{ (m}^3/\text{s}) \) for \( j = 1, 2, ..., L \) was achieved in at most 8 iterations. The only differences noted between these examples was the convergence behaviour near the solution. With the network of low head-loss 0.039 (m) as described in Sections 6.4.2, oscillatory behaviour was observed. With the network of high head-loss 0.267 (m) as described in Sections 6.4.3, the convergence was very rapid towards the solution; the rate of convergence was, in fact, second-order. It implies that the energy head is sensitive to the changes made in the flow, and hence a more desirable convergence behaviour was exhibited through the results.

It was envisaged that the characteristics of the virtual string may have certain impact on convergence. The characteristic of the string can be raised to some power \( e \) in order to compensate the convexity of the function. Investigations showed that \( e = 1 \) gave the best convergence in the examples investigated.
One important aspect observed in the examples is that the results obtained were equally successful for both the constant inflow and constant outflow situations. It is also encouraging that such complex situations incorporating hydraulic jump can be modelling using the procedures described.
CHAPTER 7

SENSITIVITY OF NETWORK FLOW DISTRIBUTION TO WATER LEVEL
7 SENSITIVITY OF NETWORK FLOW DISTRIBUTION TO WATER LEVEL

7.1 Introduction

The analysis of an open channel network involves investigating the effect of changing network variables on the behaviour of the network. One of the difficulties is that there are many different variables, and it is difficult to generalise the effects of changing these variables on a particular solution algorithm. In many situations, the sensitivity of flow distribution in the network to changes made in a single network variable are of particular importance. For example, the situation arises when evaluating a design for the proposed network that involves some degrees of uncertainty, in a particular network variable. If the network is sensitive to that variable, then a small error will cause the flow distribution throughout the network to be incorrectly calculated.

Whilst any adjustment made to a network variable requires a complete new solution. A sensitivity analysis may be used to give some indications of the behaviour of the network. A sensitivity analysis yields the rate of change of the selected variables with respect to the changes in a single variable of interest. For instance, what effect will be imposed on the final flow distribution when the downstream water level is adjusted.

Apart from the above design purposes, sensitivity analysis lends itself well to the field of real-time monitoring by computer operations. The real-time information could be fed into the analysis to foresee any abnormal network flow condition, and thereby adjust the network variables (such as sluice gate or valve openings) to keep the network operating under the control conditions.
7.2 Sensitivity Analysis of an Open Channel

For a prolonged period of operation, solid matter carried by the flow may deposit on the channel bed causing its slope and surface roughness gradually to change. If the flow in the channel is sensitive enough to be affected by the depositions, then the head-loss along the channel length will change accordingly. This, in turn, causes a change in water level along the channel length.

A change head-loss or water level in one particular part of network can affect the final flow distribution. It is therefore relevant to study the significance of variations in water level and to derive a relationship to describe the sensitivity of the network. To accomplish this, the sensitivity of flow in a channel must first be investigated. The work presented herein involves investigating the fluctuations in head-loss as well as water level with respect to changes made in downstream water level.

The differential equation that governs the gradually varied flow in an open channel is

\[
\frac{dy}{dx} = \frac{S_o - S_f}{1 - Fr^2}
\]  

(7.1)

This can be expressed in non-dimensional form; the advantage of having (7.1) in non-dimensional form is that all channel depths can be cross referenced to a single depth of interest. The reference depth may either be the normal depth, \( y_n \) or the critical depth, \( y_c \). This idea has been put forward by many researchers, such as Bresse (1860), Chow (1959) and Minton and Sobey (1973).

The non-dimensional form of (7.1) presented by Chow in terms of conveyance, \( K \), and section factor, \( Z \), is shown below:

\[
\frac{dy}{dx} = S_o \frac{1 - \left( \frac{K_n}{K} \right)^2}{1 - \left( \frac{Z_c}{Z} \right)^2}
\]

(7.2)

in which subscripts \( n \) and \( c \) denote the values of \( K \) and \( Z \) are evaluated at \( y_n \) and \( y_c \) respectively. The value of \( Z \) is evaluated as
SENSITIVITY OF NETWORK FLOW DISTRIBUTION TO WATER LEVEL

\[ Z = \sqrt{\frac{A^2}{T}} \]  

(7.3)

and when the Manning friction formula is used, \( K \) is evaluated as

\[ K = \frac{AR^\kappa}{n} \]  

(7.4)

With a rectangular channel, (7.2) can be simplified further by substituting (7.3) and (7.4) into (7.2), and re-arranging the expression in terms of \( y/y_e \)

\[
f(x, y) = \frac{d}{d\left(\frac{xs}{y_e}\right)} \left( \frac{y}{y_e} \right)^n - \left( \frac{y}{y_e} \right)^n \left( \frac{B/2y_e + y/y_e}{B/2y_e + y/y_e} \right)^\kappa \]  

(7.5)

In the above equation, the \( y_e \) is chosen as the reference depth because it is not affected by the friction coefficient or the bed slope. In gradually varied flow, \( y_e \) is dependent upon frictional resistance; it seems more appropriate to let \( y_e \) be the independent variable, and choose \( y_e \) as the reference depth.

For the given rectangular section and discharge, the terms \( B/2y_e \) and \( y_e/y_e \) can be calculated. The only unknown in the (7.5) appears to be \( y/y_e \), which can be solved numerically using the fourth-order Runge-Kutta method described in Chapter 2. When (7.5) is integrated, the solution is represented by

\[
\frac{y}{y_e} = f\left( \frac{B}{2y_e}, \frac{y_e}{y_e}, \frac{y}{y_e}, \frac{xs}{y_e} \right) \]  

(7.6)

in which \( y_e \) denotes the initial depth of the integration. It can be noted that none of the terms in (7.6) is dimensional; thus, a series of graphs can be plotted to unify the analysis of open channel flow.
A typical graph obtained from (7.6) is shown in Figure 7.1. For clarity only, a limited number of non-dimensional subcritical water surface profiles are plotted. The graph is continuous throughout and can be extended to include various profile categories if necessary.

Any co-ordinate point, \((y_n/y_e, y/y_e, xS_o/y_e)\), on the graph represents the state of flow along the channel length. To establish a water surface profile, both \(y_n\) and \(y_e\) must be known a priori. For a given channel and discharge, both \(y_n\) and \(y_e\) can be calculated to give \(y_n/y_e\). A flow with known \(y_n/y_e\) is represented by a curve on the graph, which categorises the water surface profile for the flow as shown.

To evaluate the water depth at Section -1 some distance, \(L\), away from the control Section -2, the non-dimensional distance, \(xS_o/y_e = LS_o/y_e\), between Section -1 and Section -2 is required. Consider an example where the \(y_n/y_e = 3.5\). The non-dimensional depth of control Section -2, \(y_2/y_e\), is calculated and transferred to the graph (shown as a). The intersection point (shown as b) of \(y_2/y_e\) and \(y_n/y_e\) marks the non-dimensional distance of Section -2 on the \(xS_o/y_e\)-axis as \(x_2S_o/y_e\). Since the horizontal axis is the non-dimensional distance, it follows that the non-dimensional distance of Section -1 is given by

\[
\frac{x_1S_o}{y_e} = \frac{x_2S_o}{y_e} \pm \frac{LS_o}{y_e}
\]

in which the sign, \(\pm\), denotes whether the position of Section -1 is upstream or downstream of Section -2. Since the \(y_n/y_e\) is constant and the \(x_1S_o/y_e\) is obtained, the non-dimensional depth, \(y_1/y_e\), can be found directly from the graph (shown as c); thence the actual water depth \(y_1\) is determined.

With the solution presented in non-dimensional form, the positions of Section -1 and Section -2 are merely 'mapped' onto the \(xS_o/y_e\)-axis, denoted by \(d\) and \(e\) respectively. A slight change in the \(y_2/y_e\) causes \(d\) and \(e\) to travel simultaneously along the \(xS_o/y_e\)-axis by the same distance. The distance travelled is dependent on the non-dimensional water surface slope at the point of interest.
Figure 7.1. Non-dimensional plot of water surface profile, $B/2y_c = 1.0$. 
Suppose the \( y_2/y_e \) is increased by a very small \( \Delta y_2/y_e \); the corresponding change in non-dimensional distance, \( \Delta x_2 S_o/y_e \), in terms of the non-dimensional water surface slope can be found by direct integration of (7.5) to give

\[
\int_{y_2/y_e}^{y_2+y_2/\Delta y_2/y_e} \, d\left( \frac{y_2}{y_e} \right) = \int_{x_2 S_o/y_e}^{x_2 S_o+y_2 S_o/y_e} \, f(x, y) \, d\left( \frac{x S_o}{y_e} \right)
\]  

(7.8a)

this approximates numerically to

\[
\frac{\Delta y_2}{y_e} = \frac{d\left( \frac{y_2}{y_e} \right)}{d\left( \frac{x_2 S_o}{y_e} \right)} \frac{\Delta x_2 S_o}{y_e} + O\left( \left( \frac{\Delta x_2 S_o}{y_e} \right)^2 \right)
\]  

(7.8b)

By neglecting the higher order derivative terms and assuming \( \Delta x_2 \) is so small that the numerical errors are negligible, (7.8b) is simplified into

\[
\frac{\Delta y_2}{y_e} = \frac{d\left( \frac{y_2}{y_e} \right)}{d\left( \frac{x_2 S_o}{y_e} \right)} \frac{\Delta x_2 S_o}{y_e}
\]  

(7.9a)

in which the subscript 2 denotes the derivative function is evaluated at Section-2. Similarly, the relationship between \( \Delta y_1/y_e \) and \( \Delta x_1 S_o/y_e \) is written as

\[
\frac{\Delta y_1}{y_e} = \frac{d\left( \frac{y_1}{y_e} \right)}{d\left( \frac{x_1 S_o}{y_e} \right)} \frac{\Delta x_1 S_o}{y_e}
\]  

(7.9b)

in which the subscript 1 denotes the derivative function is evaluated at Section-1. Because the non-dimensional distance in (7.5) is linearised along the horizontal axis shown in Figure 7.1, it follows that the \( \Delta x_1 \) will travel linearly through the same distance as \( \Delta x_2 \). Hence, equating \( \Delta x_1 = \Delta x_2 \) in (7.9a) and (7.9b) gives
SENSITIVITY OF NETWORK FLOW DISTRIBUTION TO WATER LEVEL

\[ \hat{\theta}(y_1, y_2) = \frac{\left( \frac{\Delta y_1}{y_c} \right)}{\left( \frac{\Delta y_2}{y_c} \right)} = \frac{d\left( \frac{y_1}{y_c} \right)}{d\left( \frac{x_1S_2}{y_c} \right)} \frac{d\left( \frac{x_2S_2}{y_c} \right)}{d\left( \frac{y_2}{y_c} \right)} \quad (7.10) \]

and taking the limit as \( \Delta x \to 0 \), (7.10) becomes

\[ \hat{\theta}(y_1, y_2) = \frac{d\left( \frac{y_1}{y_c} \right)}{d\left( \frac{y_2}{y_c} \right)} = \frac{f(x_1, y_1)}{f(x_2, y_2)} \quad (7.11) \]

(Also see (7.5)). The above equation defines the sensitivity of flow in a channel with respect to water depth. It is apparent that the \( \hat{\theta}(y_1, y_2) \) is independent of the length of a channel but depends on the water depths under consideration. Calculation of \( \hat{\theta}(y_1, y_2) \) can be interpreted as the approximate variation of \( y_1 \) caused by a unit change in \( y_2 \). For the case where \( y_1 \) and \( y_2 \) are both known, the sensitivity of a channel flow can then be drawn; common situations are shown in Table 7.1. It can be noted that the sensitivity of a channel as defined above lies between zero and unity.

The study of sensitivity may also includes investigating the effects of changes in energy heads with variations of water depths, since a change in water depth will also cause a change in velocity head. The term of interest is \( dH_i/dy_i \), in which \( H_i \) and \( y_i \) are total energy head and water depth respectively; the subscript \( i \) denotes the section. With a section of known \( Q \) and \( y_i \), the total energy head can be found by

\[ H_i = z_i + y_i + \frac{Q_i^2}{2gA_i^2} \quad (7.12) \]

It follows that the first-order derivative of (7.12) with respect to \( y_i \) is

\[ \frac{dH_i}{dy_i} = 1 - F_{i_1}^2 \quad (7.13) \]
<table>
<thead>
<tr>
<th>Case</th>
<th>Upstream</th>
<th>Downstream</th>
<th>Sensitivity</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$y_c \equiv y_1$</td>
<td>$y_c &lt; y_2 &lt; y_1$</td>
<td>$\delta(y_1, y_2) &lt; 1$</td>
</tr>
<tr>
<td>2</td>
<td>$y_a \equiv y_1$</td>
<td>$y_c &lt; y_2 &lt; y_a$</td>
<td>$\delta(y_1, y_2) \geq 0$</td>
</tr>
<tr>
<td>3</td>
<td>$y_a \equiv y_1$</td>
<td>$y_a &lt; y_2$</td>
<td>$\delta(y_1, y_2) \geq 1$</td>
</tr>
<tr>
<td>4</td>
<td>$y_a &lt; y_1$</td>
<td>$y_2 &gt; y_a$</td>
<td>$\delta(y_1, y_2) &lt; 1$</td>
</tr>
<tr>
<td>5</td>
<td>$y_a &lt;&lt; y_1$</td>
<td>$y_a &lt;&lt; y_2$</td>
<td>$\delta(y_1, y_2) \leq 1$</td>
</tr>
</tbody>
</table>

Table 7.1. Sensitivity of a channel flow with depths.
The above equation can be interpreted as the variation of $H_i$ caused by a unit change of $y_i$. The variation in head at section $i$ caused by fluctuation in water depth at a remote section $i + 1$ is expressed as

\[
\frac{dH_i}{dy_{i+1}} = \frac{dH_i}{dy_i} \frac{dy_i}{dy_{i+1}}
\]  

(7.14)

It is known from the (7.11) that

\[
\frac{dy_i}{dy_{i+1}} = \theta(y_i, y_{i+1})
\]  

(7.15)

Substituting (7.15) into (7.14) and using (7.13) gives

\[
\frac{dH_i}{dy_{i+1}} = \theta(y_i, y_{i+1})(1 - F_{i}^2)
\]  

(7.16)

The above equation can be solved directly for $M_1$ and $M_2$ flow profiles only. With this restriction, neither the $|\theta(y_i, y_{i+1})|$ nor the $|1 - F_{i}^2|$ is greater than unity, and hence

\[
\left| \frac{dH_i}{dy_{i+1}} \right| < 1
\]  

(7.17)

### 7.3 Sensitivity Analysis for Open Channel Networks

Consider the network problem, shown in Figure 7.2, in which the total energy head at the point of outfall is assumed to increase. As a practical example, this situation could occur as the water level in the receiving water course rises during flood conditions. The flows in the channels as well as water levels will change gradually in order to balance the change in the outfall condition. The flow re-distribution can be described by

\[
\frac{dQ_i}{dH_{\text{outfall}}}, \quad i = 1, \ldots, S_N
\]  

(7.18)
in which $Q_i$ and $H_{\text{outfall}}$ denote the flow in a string and total energy head at the point of outfall respectively; $S_N$ denotes number of strings in the network. The (7.16) describes the variation in $Q_i$ caused by an unit change of $H_{\text{outfall}}$. At equilibrium, a system of loop equations must satisfy:

$$f_i = 0, \quad i = 1, \ldots, L_N$$  \hspace{1cm} (7.19)

in which $L$ denotes the total number of equations, which corresponds with the number of loops in the system. At a solution, the head-losses around a closed loop must equal zero. That is,

$$f_i = h_i - h_{i+1} = 0, \quad i = 1, \ldots, L_N$$  \hspace{1cm} (7.20)

in which subscript $i$ denotes equation number or string number. Referring to Figure 7.2, using the energy equation to expand $h_i$ in terms of loop flow correction factors $q_j$ for $j = 1, 2, \ldots, L_N$ yields

$$h_{f1} = (z_{i1} - z_{i2}) + (y_{i1} - y_{i2}) + \left( \frac{\alpha_{i1}(Q_i - q_i)^2}{2gA_{i1}^2} - \frac{\alpha_{i2}(Q_i - q_i)^2}{2gA_{i2}^2} \right)$$

$$h_{f2} = (z_{21} - z_{22}) + (y_{21} - y_{22}) + \left( \frac{\alpha_{21}(Q_2 + q_1 - q_2)^2}{2gA_{21}^2} - \frac{\alpha_{22}(Q_2 + q_1 - q_2)^2}{2gA_{22}^2} \right)$$

$$h_{f3} = (z_{31} - z_{32}) + (y_{31} - y_{32}) + \left( \frac{\alpha_{31}(Q_3 + q_2 - q_3)^2}{2gA_{31}^2} - \frac{\alpha_{32}(Q_3 + q_2 - q_3)^2}{2gA_{32}^2} \right)$$

$$h_{f4} = (z_{41} - z_{42}) + (y_{41} - y_{42}) + \left( \frac{\alpha_{41}(Q_4 + q_3)^2}{2gA_{41}^2} - \frac{\alpha_{42}(Q_4 + q_3)^2}{2gA_{42}^2} \right)$$  \hspace{1cm} (7.21)

where

$i,1 =$ subscript denotes upstream end of string $i$

$i,2 =$ subscript denotes downstream end of string $i$
Figure 7.2. A parallel open channel network.

<table>
<thead>
<tr>
<th>Strings</th>
<th>Width, ( B ), (m)</th>
<th>Length, ( L ), (m)</th>
<th>Bed Slope, ( S_b )</th>
<th>Bank Slope, ( S_m )</th>
<th>Manning, ( n )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inflow</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Channel - 1</td>
<td>1.00</td>
<td>50.0</td>
<td>0.0005</td>
<td>0.0</td>
<td>0.013</td>
</tr>
<tr>
<td>Channel - 2</td>
<td>1.50</td>
<td>50.0</td>
<td>0.0005</td>
<td>0.0</td>
<td>0.013</td>
</tr>
<tr>
<td>Channel - 3</td>
<td>2.00</td>
<td>50.0</td>
<td>0.0005</td>
<td>0.0</td>
<td>0.013</td>
</tr>
<tr>
<td>Channel - 4</td>
<td>2.50</td>
<td>50.0</td>
<td>0.0005</td>
<td>0.0</td>
<td>0.013</td>
</tr>
<tr>
<td>Outfall</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 7.2. Physical characteristics for the parallel open channel network.
In the above expansion, the direction of loop flow correction factors follows the anti-clockwise sense and that is illustrated in Figure 7.2. Note that from the above equations

\[ h_R = f(y_{i1}, y_{i2}, q_j), \quad \begin{cases} i = 1, 2, \ldots, S_N \\ j = 1, 2, \ldots, L_N \end{cases} \]  

(7.22)

This means that any separate change in variables \( y_{i1}, y_{i2} \) and \( q_j \) causes \( h_R \) to change accordingly. If \( y_{i2} \) is chosen for particular interest, then a change in \( y_{i2} \) causes a change in \( h_R \) with a response rate of

\[ \frac{dh_R}{dy_{i2}} = \frac{df(y_{i1}, y_{i2}, q_j)}{dy_{i2}}, \quad \begin{cases} i = 1, 2, \ldots, S_N \\ j = 1, 2, \ldots, L_N \end{cases} \]  

(7.23)

It must be noted that the \( y_{i1} \) and \( q_j \) are not independent of \( y_{i2} \), but are connected by equations of the general form \( y_{i1} = f(y_{i2}) \) and \( q_j = f(y_{i2}) \) respectively. Using the chain rule (Bajpai, 1974) to expand the right-hand side of (7.23) yields

\[ \frac{dh_R}{dy_{i2}} = \frac{\partial h_R}{\partial y_{i2}} + \frac{\partial h_R}{\partial y_{i1}} \frac{dy_{i1}}{dy_{i2}} + \sum_{j=1}^{L} \frac{\partial h_R}{\partial q_j} \frac{dq_j}{dy_{i2}}, \quad i = 1, 2, \ldots, S_N \]  

(7.24)

This example is interested in the term \( \frac{dQ_i}{dH_{\text{outfall}}} \). With minor losses neglected, the downstream total head of individual channels can be equated directly to the total head at the outfall, \( H_{\text{outfall}} \). That is,

\[ H_{i2} = H_{\text{outfall}}, \quad i = 1, 2, \ldots, S_N \]  

(7.25)

It follows that the \( y_{i2} \) becomes

\[ y_{i2} = f(H_{\text{outfall}}), \quad i = 1, 2, \ldots, S_N \]  

(7.26)
Referring to (7.13), the derivative of (7.26) is

\[ \frac{dy_{i2}}{dH_{\text{outfall}}} = \frac{1}{1 - F_{r2}^2} \]  

(7.27)

Multiplying (7.24) by (7.27) yields

\[ \frac{dh_f}{dH_{\text{outfall}}} = \frac{\partial h_f}{\partial y_{i1}} \frac{dy_{i1}}{dH_{\text{outfall}}} + \frac{\partial h_f}{\partial y_{i2}} \frac{dy_{i2}}{dH_{\text{outfall}}} + \sum_{j=1}^{L} \frac{\partial h_f}{\partial q_j} \frac{dq_j}{dH_{\text{outfall}}}, \quad i = 1, 2, \ldots, S_N \]  

(7.28)

Two unknowns \( \frac{dh_f}{dH_{\text{outfall}}} \) and \( \frac{dq_j}{dH_{\text{outfall}}} \) appear in (7.28). However, the former is readily determined since an equal but opposite response rate of \( \frac{dh_f}{dH_{\text{outfall}}} \), must apply to (7.28) to counteract the changes made in \( h_f \) in order to maintain the solution condition that \( f_i = 0 \) for \( i = 1, 2, \ldots, L_N \). That is to say,

\[ \frac{dh_f}{dH_{\text{outfall}}} = - \frac{dh_{f\text{outfall}}}{dH_{\text{outfall}}}, \quad i = 1, 2, \ldots, L_N \]  

(7.29)

Using condition (7.29) to deduct (7.28), the resulting equation becomes

\[ 0 = f_1 + f_2 + J\delta \]  

(7.30)

in which \( f_1, f_2 \) and \( \delta \) denote column vectors and \( J \) denotes a Jacobian matrix. In which terms

\[ [f_{1i}] = \left[ \begin{array}{c} \frac{\partial h_f}{\partial y_{i1}} \frac{dy_{i1}}{dH_{\text{outfall}}} - \frac{\partial h_{f+1}}{\partial y_{i1}} \frac{dy_{i+11}}{dH_{\text{outfall}}} \\ \frac{\partial h_f}{\partial y_{i2}} \frac{dy_{i2}}{dH_{\text{outfall}}} - \frac{\partial h_{f+1}}{\partial y_{i2}} \frac{dy_{i+12}}{dH_{\text{outfall}}} \end{array} \right] \]

\[ [f_{2i}] = \left[ \begin{array}{c} \frac{\partial h_f}{\partial y_{i2}} \frac{dy_{i2}}{dH_{\text{outfall}}} - \frac{\partial h_{f+1}}{\partial y_{i2}} \frac{dy_{i+12}}{dH_{\text{outfall}}} \\ \frac{\partial h_f}{\partial y_{i1}} \frac{dy_{i1}}{dH_{\text{outfall}}} - \frac{\partial h_{f+1}}{\partial y_{i1}} \frac{dy_{i+11}}{dH_{\text{outfall}}} \end{array} \right] \]

\[ [J_{ij}] = \left[ \begin{array}{c} \frac{\partial h_f}{\partial q_j} \\ \frac{\partial h_{f+1}}{\partial q_j} \end{array} \right] \]

(7.31)

\[ [\delta_i] = \left[ \begin{array}{c} \frac{dq_j}{dH_{\text{outfall}}} \end{array} \right] \]
and the total derivatives are evaluated as

\[
\frac{\partial y_{i,2}}{\partial H_{\text{outfall}}} = \frac{1}{1 - Fr^2_{i,2}}
\]

\[
\frac{\partial y_{i,1}}{\partial H_{\text{outfall}}} = \frac{\partial y_{i,2}}{\partial H_{\text{outfall}}} \frac{\partial y_{i,1}}{\partial y_{i,2}} = \frac{\vartheta(y_{i,1}, y_{i,2})}{1 - Fr^2_{i,2}}
\]

also see (7.15); and the partial derivatives are obtained by directly differentiating the

\[ h_i = H_{i,1} - H_{i,2} \]

for \( i = 1, 2, \ldots, S_N \) to give

\[
\frac{\partial h_i}{\partial y_{i,1}} = 1 - Fr^2_{i,1}
\]

\[
\frac{\partial h_i}{\partial y_{i,2}} = 1 - Fr^2_{i,2}
\]

\[
\frac{\partial h_i}{\partial q_j} = \pm \frac{(Q_j \pm q_j)}{gA_{i,1}^2} \pm \frac{(Q_j \pm q_j)}{gA_{i,2}^2}
\]

Substituting (7.32a) and (7.32b) into (7.31) gives

\[
[f_{i,1}] = \begin{bmatrix} \vartheta(y_{i,1}, y_{i,2}) & 1 - Fr^2_{i,1} & \vartheta(y_{i+1,1}, y_{i+1,2}) & 1 - Fr^2_{i+1,1} & \end{bmatrix}
\]

\[
[f_{i,2}] = [0]
\]

respectively. Because \( f_2 \) shown above is a null matrix, eliminating \( f_2 \) from (7.30) and equating the result in terms of \( J \) and \( f_1 \) yields

\[
\delta = -J^{-1}f_1
\]

The above expression can be seen to resemble the Newton-Raphson iterative formula. If a network solution is available, then the \( J \) or even the \( J^{-1} \) is already evaluated, depending upon the solution method adopted. The method used to evaluate \( J \) is similar to that
presented in Chapter 4. The only difference between the procedures is the method to evaluate \( f_1 \).

This particular example is interested in \( \frac{dQ_i}{dH_{\text{outfall}}} \); the solution takes the form of

\[
\frac{dQ}{dH_{\text{outfall}}} = \begin{bmatrix}
\frac{dQ_1}{dH_{\text{outfall}}} \\
\frac{dQ_2}{dH_{\text{outfall}}} \\
\frac{dQ_3}{dH_{\text{outfall}}} \\
\frac{dQ_4}{dH_{\text{outfall}}}
\end{bmatrix} = \begin{bmatrix}
-\frac{dq_1}{dH_{\text{outfall}}} \\
\frac{dq_1}{dH_{\text{outfall}}} & \frac{dq_2}{dH_{\text{outfall}}} & \frac{dq_3}{dH_{\text{outfall}}} & \frac{dq_4}{dH_{\text{outfall}}}
\end{bmatrix}
\]

(7.34)

A positive value of \( \frac{dQ_i}{dH_{\text{outfall}}} \) means that the flow, \( Q_i \), in string \( i \) will be increasing with \( H_{\text{outfall}} \). The larger the value of \( \frac{dQ_i}{dH_{\text{outfall}}} \), the greater the sensitivity of flow, \( Q_i \), in string \( i \) to the increasing \( H_{\text{outfall}} \).

In order to show the mechanism of the above analysis, the network shown in Table 7.2 is analysed. Firstly, the network is solved iteratively using the method outlined in Chapter 4. The BFGS solution algorithm is chosen for the network solution because it does not require partial derivatives and is comparatively more stable than other quasi-Newton methods. The solution is shown in Table 7.3a; the critical depth and the normal depth were also calculated for each channel and are tabulated in Table 7.3b, in which the figures under the heading \( '\hat{y}(y_1, y_2)' \) are calculated using (7.10). When the solution is obtained, for example using the BFGS method, the Jacobian matrix \( J \) is also evaluated. Hence, the numerical values in \( J \) can be abstracted from the final iteration of the BFGS solution algorithm. The results are as follows:

\[
H_n = -J^{-1} = \begin{bmatrix}
10.51166 & 6.09058 & 2.46647 \\
6.09058 & 17.22700 & 9.28658 \\
2.46647 & 9.28658 & 18.70512
\end{bmatrix}
\]
Secondly, the \( f_1 \) in (7.33) is evaluated from the definitions in (7.32c), and numerical values are obtained using the final flow distribution shown in Tables 7.3a and 7.3b; the evaluation is given below:

\[
\begin{align*}
\mathbf{f}_1 &= \left[ (0.98626)(0.99878) - (0.98580)(0.99835) \right] \left[ 0.00009 \right] \\
&\quad \left[ (0.98580)(0.99835) - (0.98548)(0.99801) \right] \left[ 0.00065 \right] \\
&\quad \left[ (0.98548)(0.99801) - (0.98525)(0.99774) \right] \left[ 0.00049 \right]
\end{align*}
\]

Having found both \( \mathbf{J} \) and \( \mathbf{f}_1 \), the sensitivity analysis can begin. The quantity of flow to be re-distributed due to a unit change in \( H_{\text{Outfall}} \) is given by

\[
\delta = -\mathbf{J}^{-1}\mathbf{f}_1 = \begin{bmatrix} 10.51166 & 6.09058 & 2.46647 & 0.00090 \end{bmatrix} \begin{bmatrix} 0.01463 \end{bmatrix} = \begin{bmatrix} 0.02128 \end{bmatrix}
\]

Finally, the sensitivity \( \frac{dQ}{dH_{\text{Outfall}}} \) are evaluated as follows from (7.34)

\[
\frac{dQ}{dH_{\text{Outfall}}} = \begin{bmatrix} -0.01463 \\ 0.01463 - 0.02128 \\ 0.02128 - 0.01750 \\ 0.01750 \end{bmatrix} = \begin{bmatrix} -0.01463 \\ -0.00665 \\ 0.00378 \\ 0.01750 \end{bmatrix}
\]

in which the negative sign denotes that the flow decreases as \( H_{\text{Outfall}} \) increases.

To verify the results obtained above, the network was re-run using \( H_{\text{Outfall}} \in \{0.751, 0.850\} \) (m), that is to increase the \( H_{\text{Outfall}} \) by one millimetre and a hundred millimetres. The intention is to derive the actual sensitivity of the flow re-distribution with respect to \( H_{\text{Outfall}} \) using the technique of finite differencing (forward difference), and then make a comparison to those sensitivities obtained by theoretical means as shown above. The solution of the re-run is tabulated in Table 7.4a and Table 7.4b respectively, and the sensitivity to flow for each channel is calculated in Table 7.5a and Table 7.5b respectively. The heading 'Actual' denotes the actual \( \frac{dQ}{dH_{\text{Outfall}}} \) obtained by the technique of finite differencing and using those data shown in Table 7.3a and Table 7.4a; whilst the heading 'Theoretical' denotes the theoretical
SENSITIVITY OF NETWORK FLOW DISTRIBUTION TO WATER LEVEL

dQ/dH_{\text{Outfall}} calculated by (7.33). Because the unit of Q is cubic metre per second, whilst the H_{\text{Outfall}} is in metre, it appears that the unit for dQ/dH_{\text{Outfall}} is cubic metre per second per metre (m$^3$/s/m). It is therefore interpreted as the rate of change of Q due to a unit change of H_{\text{Outfall}}.

On comparing the Actual and the Theoretical results, the latter was considerably underestimated (see Tables 7.5a and 7.5b). A possible explanation is that the system of equations is non-linear, but the model (7.33) pursued is linearised. This is seen by comparing the Actual sensitivity between two sets of results Tables 7.5a and 7.5b. If the network being solved is linear, then the sensitivity should be exactly the same. However, it is seen that both sets of results are different, because the actual system of equations is non-linear. Because both sets of results varied little from each other, it is regarded that the network is insensitive to changes made in H_{\text{Outfall}}, even though a sizeable (0.100 m) increase in H_{\text{Outfall}} is used in the analysis.

The theoretical analysis consistently under-estimates flow re-distribution. This may be caused by two reasons. Firstly, (7.33) may not scale the predictions accurately, and secondly, (7.33) may not provide a direct path (direction) to pinpoint the exact position of the solution.

The first point is seen by comparing the norms (Euclidean length or distance) of two vectors obtained by Actual and Theoretical means. The norm referred herein describes the flow which needs to be re-distributed. The difference of the two norms determines the appropriateness of the prediction given by (7.33). From Table 7.5a, the two vectors are

Actual: \( \mathbf{n}_A = [-0.03830, -0.02440, 0.00830, 0.05440]^T \) (m$^3$/s/m)

Theoretical: \( \mathbf{n}_T = [-0.01463, -0.00665, 0.00378, 0.01750]^T \) (m$^3$/s/m)

Hence, the norms for \( \mathbf{n}_A \) and \( \mathbf{n}_T \) are given below:

Actual: \( \| \mathbf{n}_A \| = 0.07134 \) (m$^3$/s/m)

Theoretical: \( \| \mathbf{n}_T \| = 0.02406 \) (m$^3$/s/m)
or the flow being re-distributed caused by the 0.001 (m) increase in \( H_{\text{osfall}} \) is

Actual: \[ \| n_A \| = 0.07134 \times 10^{-3} \text{ (m}^3/\text{s)} \]

Theoretical: \[ \| n_T \| = 0.02406 \times 10^{-3} \text{ (m}^3/\text{s)} \]

Evidently, the \( \| n_T \| \) is under-estimated, and by as much as 66%. As described in Chapter 4, (7.33) solely provides the direction in which the solution lies, but it does not consider the distance of the solution from the current approximation, since the model is linearised. This is one of the principle limitations of the linearised model (7.33).

The second point can be verified by calculating the angle between \( n_A \) and \( n_T \). This is accomplished by using the formula shown below:

\[
\cos \theta = \frac{n_A \cdot n_T}{\| n_A \| \| n_T \|}
\]

(Marsen and Tromba, 1988). If \( \theta \) is calculated to be zero (ideal case), then the vector \( n_T \) is said to be in-line with the vector \( n_A \), meaning that \( n_T \) provides the direct path pointing to the solution. A negative \( \theta \) denotes \( n_T \) is travelling in an opposite direction away from \( n_A \). This also happens when \( \theta > 90.0^\circ \). In this case, the predictions given by (7.33) is said to be inappropriate because it provides the incorrect path to locate where the actual solution lies. A small positive value indicates that the prediction is correct and the goodness of the prediction is depending upon the magnitude of \( \theta \). The closer to zero, the greater the degree of accuracy of the prediction. In this example, the angle between the two vectors \( n_A \) and \( n_T \) is calculated to be

\[ \theta = 6.35^\circ \]

This indicates that \( n_T \) is not providing the direct path pointing to where the solution lies, but deviates slightly away from \( n_A \) by 6.35°. Because \( 0 \leq |\theta| \leq 180.0^\circ \), it is therefore possible to express the above result in term of percentage, and hence

\[ \frac{6.35^\circ}{180.0^\circ} \times 100\% = 3.5\% \]
This deviation (or error) also illustrates the under-estimations of the theoretical flow redistribution. However, in comparing this with the 66% error in the calculation of norms above, the deviation is considered small. In general, therefore, the prediction of the behaviour of the network is reasonable. The error for the worst case (Channel-2) was found to be 73%. This suggests that the error involved in the analysis is caused and dominated by the scaling calculated by (7.33).

Collectively, (7.33) predicts the general behaviour of the network correctly, but it does not predict the flow re-distribution particularly accurately. However, the importance of this analysis is that it has validated the use of (7.33) in qualitatively predicting the behaviour of the open channel networks. This can be seen by examining Table 7.8. The numerical signs (Theoretical) agree with the results (Actual) obtained, thereby supporting the argument that (7.33) is useful for predicting the general behaviour of networks in a qualitative fashion, but not with sufficient accuracy to predict with confidence with the actual flow being re-distributed.

In order to further explore the reliability of (7.33), four further computer runs using $H_{\text{outfall}} \in \{0.55, 0.65, 0.85, 0.95\} \text{(m)}$ were conducted under identical conditions. The sensitivity analysis using (7.33) was performed on each test case and the results are tabulated in Table 7.6a (Theoretical). For comparison only, the $H_{\text{outfall}}$ was again increased by one millimetre for each test case and the corresponding sensitivity analysis is tabulated in Table 7.6b (Actual). Four points observed consistently over the results are:

i) All the predicted sensitivities were considerably under-estimated but the order of magnitude, in general, agreed well with the actual results obtained. This phenomenon has been explained above.

ii) The signs of the theoretical prediction agreed with the actual results obtained. This implies that the procedure used to predict the behaviour of the network were qualitative rather than quantitative.

iii) A common trend of decreasing sensitivity with increasing $H_{\text{outfall}}$ was observed in both set of results. This was due to the $\theta(y_{i1}, y_{i2}) \rightarrow 1$, and consequently $f_i \rightarrow 0$, as $H_{\text{outfall}}$ increased.

iv) The flow distribution was relatively insensitive to the variation of $H_{\text{outfall}}$ for all the test cases being investigated.
The point iv) can be justified by multiplying the $\frac{dQ}{dH_{outfall}}$ by a small change in $H_{outfall}$ to determine the amount of flow which needs to be re-distributed. For example, if ten millimetres is used to measure the fluctuation of energy head at the point of outfall because of surface waves, or solid matter deposited on the invert, then the flow redistribution is approximated as shown below:

$$\Delta Q = \frac{dQ}{dH_{outfall}} \times 10^{-2} = \begin{bmatrix} -0.000383 \\ -0.000244 \\ 0.000083 \\ 0.000544 \end{bmatrix} \text{ (m}^3/\text{s})$$

or in term of percentages

$$\frac{\Delta Q}{Q} \times 100\% = \begin{bmatrix} -0.164 \\ -0.060 \\ 0.021 \\ 0.070 \end{bmatrix} \text{ (\%)}$$

which is far less than the variation of $H_{outfall}$

$$\frac{\Delta H_{outfall}}{H_{outfall}} \times 100\% = \frac{0.01}{0.750} \times 100\% = 1.33\%$$

In this example, the re-distribution of flow is seen to be sufficiently small to be considered negligible as far as applications to water treatment works is concerned.

Collectively, the predictions given by (7.33) might be too crude for specific usage but they do provide some hints on the magnitude of the flow being re-distributed, and most important of all is that the general behaviour of the network is correctly predicted for the example used.
<table>
<thead>
<tr>
<th>Strings</th>
<th>Final Energy Head, $H_f$ (m)</th>
<th>Final Water Depth, $y_i$ (m)</th>
<th>Final Discharge, $Q_f$ (m$^3$/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Upstream</td>
<td>Downstream</td>
<td>Upstream</td>
</tr>
<tr>
<td>Inflow</td>
<td>0.75429</td>
<td>0.75429</td>
<td>-</td>
</tr>
<tr>
<td>Channel -1</td>
<td>0.75429</td>
<td>0.75000</td>
<td>0.72400</td>
</tr>
<tr>
<td>Channel -2</td>
<td>0.75429</td>
<td>0.75000</td>
<td>0.72220</td>
</tr>
<tr>
<td>Channel -3</td>
<td>0.75429</td>
<td>0.75000</td>
<td>0.72085</td>
</tr>
<tr>
<td>Channel -4</td>
<td>0.75429</td>
<td>0.75000</td>
<td>0.71981</td>
</tr>
<tr>
<td>Outfall</td>
<td>0.75000</td>
<td>0.75000</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 7.3a. Final flow distribution, $H_{Outfall} = 0.750$ (m).

<table>
<thead>
<tr>
<th>Strings</th>
<th>$y_c$, (m)</th>
<th>$y_n$, (m)</th>
<th>$y_n/y_c$</th>
<th>$\delta(y_1, y_2)$</th>
<th>$Fr_1$</th>
<th>$Fr_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inflow</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Channel -1</td>
<td>0.17696</td>
<td>0.37756</td>
<td>2.13359</td>
<td>0.98626</td>
<td>0.12084</td>
<td>0.11576</td>
</tr>
<tr>
<td>Channel -2</td>
<td>0.19481</td>
<td>0.38844</td>
<td>1.99394</td>
<td>0.98580</td>
<td>0.11439</td>
<td>0.10955</td>
</tr>
<tr>
<td>Channel -3</td>
<td>0.20623</td>
<td>0.39533</td>
<td>1.91694</td>
<td>0.98548</td>
<td>0.10820</td>
<td>0.10360</td>
</tr>
<tr>
<td>Channel -4</td>
<td>0.21419</td>
<td>0.40007</td>
<td>1.86783</td>
<td>0.98525</td>
<td>0.10266</td>
<td>0.09828</td>
</tr>
<tr>
<td>Outfall</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 7.3b. Critical and normal depths for $H_{Outfall} = 0.750$ (m).
Table 7.4a. Final flow distribution with 0.001 (m) increased in $H_{\text{outfall}} = 0.750$ (m).

<table>
<thead>
<tr>
<th>Strings</th>
<th>Final Energy Head, $H$, (m)</th>
<th>Final Water Depth, $y$, (m)</th>
<th>Final Discharge, $Q$, (m$^3$/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Upstream</td>
<td>Downstream</td>
<td>Upstream</td>
</tr>
<tr>
<td>Inflow</td>
<td>0.75527</td>
<td>0.75527</td>
<td>-</td>
</tr>
<tr>
<td>Channel -1</td>
<td>0.75527</td>
<td>0.75100</td>
<td>0.72500</td>
</tr>
<tr>
<td>Channel -2</td>
<td>0.75527</td>
<td>0.75100</td>
<td>0.72320</td>
</tr>
<tr>
<td>Channel -3</td>
<td>0.75527</td>
<td>0.75100</td>
<td>0.72185</td>
</tr>
<tr>
<td>Channel -4</td>
<td>0.75527</td>
<td>0.75100</td>
<td>0.72081</td>
</tr>
<tr>
<td>Outfall</td>
<td>0.75100</td>
<td>0.75100</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 7.4b. Final flow distribution with 0.100 (m) increased in $H_{\text{outfall}} = 0.750$ (m).

<table>
<thead>
<tr>
<th>Strings</th>
<th>Final Energy Head, $H$, (m)</th>
<th>Final Water Depth, $y$, (m)</th>
<th>Final Discharge, $Q$, (m$^3$/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Upstream</td>
<td>Downstream</td>
<td>Upstream</td>
</tr>
<tr>
<td>Inflow</td>
<td>0.85301</td>
<td>0.85301</td>
<td>-</td>
</tr>
<tr>
<td>Channel -1</td>
<td>0.85301</td>
<td>0.85000</td>
<td>0.82405</td>
</tr>
<tr>
<td>Channel -2</td>
<td>0.85301</td>
<td>0.85000</td>
<td>0.82261</td>
</tr>
<tr>
<td>Channel -3</td>
<td>0.85301</td>
<td>0.85000</td>
<td>0.82150</td>
</tr>
<tr>
<td>Channel -4</td>
<td>0.85301</td>
<td>0.85000</td>
<td>0.82062</td>
</tr>
<tr>
<td>Outfall</td>
<td>0.85000</td>
<td>0.85000</td>
<td>-</td>
</tr>
<tr>
<td>Strings</td>
<td>$Q_{0.750}$ (m$^3$)</td>
<td>$Q_{0.751}$ (m$^3$)</td>
<td>$Q_{0.751} - Q_{0.750}$ (m$^3$/s)</td>
</tr>
<tr>
<td>---------</td>
<td>----------------------</td>
<td>----------------------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td></td>
<td>Actual</td>
<td>Theoretical</td>
<td></td>
</tr>
<tr>
<td>Inflow</td>
<td>2.00000</td>
<td>2.00000</td>
<td>0.0</td>
</tr>
<tr>
<td>Channel -1</td>
<td>0.23316</td>
<td>0.23312</td>
<td>-3.830E-05</td>
</tr>
<tr>
<td>Channel -2</td>
<td>0.40397</td>
<td>0.40394</td>
<td>-2.440E-05</td>
</tr>
<tr>
<td>Channel -3</td>
<td>0.58665</td>
<td>0.58666</td>
<td>8.30E-05</td>
</tr>
<tr>
<td>Channel -4</td>
<td>0.77622</td>
<td>0.77628</td>
<td>5.440E-05</td>
</tr>
<tr>
<td>Outfall</td>
<td>2.00000</td>
<td>2.00000</td>
<td>0.0</td>
</tr>
</tbody>
</table>

Table 7.5a. Sensitivity of flow re-distribution with 0.001 (m) increased in $H_{Outfall} = 0.750$ (m).

<table>
<thead>
<tr>
<th>Strings</th>
<th>$Q_{0.750}$ (m$^3$)</th>
<th>$Q_{0.850}$ (m$^3$)</th>
<th>$Q_{0.850} - Q_{0.750}$ (m$^3$/s)</th>
<th>$\frac{dQ}{dH_{Outfall}}$ (m$^3$/s/m)</th>
<th>Difference (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Actual</td>
<td>Theoretical</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Inflow</td>
<td>2.00000</td>
<td>2.00000</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Channel -1</td>
<td>0.23316</td>
<td>0.22966</td>
<td>-3.500E-03</td>
<td>-0.0350</td>
<td>-58.2</td>
</tr>
<tr>
<td>Channel -2</td>
<td>0.40397</td>
<td>0.40170</td>
<td>-2.270E-03</td>
<td>-0.0227</td>
<td>-70.7</td>
</tr>
<tr>
<td>Channel -3</td>
<td>0.58665</td>
<td>0.58739</td>
<td>7.400E-04</td>
<td>0.0074</td>
<td>-48.9</td>
</tr>
<tr>
<td>Channel -4</td>
<td>0.77622</td>
<td>0.78125</td>
<td>5.030E-03</td>
<td>0.0503</td>
<td>-65.2</td>
</tr>
<tr>
<td>Outfall</td>
<td>2.00000</td>
<td>2.00000</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>

Table 7.5b. Sensitivity of flow re-distribution with 0.100 (m) increased in $H_{Outfall} = 0.750$ (m).
### Table 7.6a. Theoretical sensitivity of flow re-distribution with various $H_{\text{Outfall}}$.

<table>
<thead>
<tr>
<th>$H_{\text{Outfall}}$ (m)</th>
<th>$\frac{dQ}{dH_{\text{Outfall}}}$, (m$^3$/s/m)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Channel-1</td>
</tr>
<tr>
<td>0.55</td>
<td>-0.0315</td>
</tr>
<tr>
<td>0.65</td>
<td>-0.0233</td>
</tr>
<tr>
<td>0.75</td>
<td>-0.0146</td>
</tr>
<tr>
<td>0.85</td>
<td>-0.0134</td>
</tr>
<tr>
<td>0.95</td>
<td>-0.0100</td>
</tr>
</tbody>
</table>

### Table 7.6b. Actual sensitivity of flow re-distribution with various $H_{\text{Outfall}}$.

<table>
<thead>
<tr>
<th>$H_{\text{Outfall}}$ (m)</th>
<th>$\frac{dQ}{dH_{\text{Outfall}}}$, (m$^3$/s/m)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Channel-1</td>
</tr>
<tr>
<td>0.55</td>
<td>-0.0621</td>
</tr>
<tr>
<td>0.65</td>
<td>-0.0475</td>
</tr>
<tr>
<td>0.75</td>
<td>-0.0383</td>
</tr>
<tr>
<td>0.85</td>
<td>-0.0319</td>
</tr>
<tr>
<td>0.95</td>
<td>-0.0273</td>
</tr>
</tbody>
</table>
7.4 Sensitivity Analysis for an Erratic Water Surface Profile

Since an analytical solution of the dynamic equation of gradually varied flow does not exist, it must be solved numerically using the methods described in Chapter 2. As a consequence, numerical error will be introduced during the computation of a water surface profile even though the step size is carefully chosen. Apparently, the numerical error can affect the final flow distribution throughout the network. It is, therefore, relevant to study the effect of numerical error on the final flow distribution.

Suppose the water surface profile along each string shown in Figure 7.2 is incorrectly calculated and subjected to some error $\varepsilon_i$. This error $\varepsilon_i$ is regarded as the errors being introduced in the solution during the numerical integration of the gradually varied flow equation. It is understood from Chapter 2 that the size of $\varepsilon_i$ is influenced by machine precision, numerical method used and the step size chosen for integration. It follows that $\varepsilon_i$ is a term which simply adds on to the true water depth, and its size is unaffected and independent of the water depth of interest. Therefore, $y_{i,2}$ the downstream depth for $i=1,2,...,S_n$ will be exact and does not involve $\varepsilon_i$ since, for M1 and M2 profiles, numerical integrations are started from downstream and then proceed in the upstream direction. On the other hand, $\varepsilon_i$ will affect $y_{i,1}$ the upstream depth. Therefore, the following relationship is derived

$$y_{i,1} = \hat{y}_{i,1} + \varepsilon_i$$

in which $\hat{y}_{i,1}$ denotes the true water depth. It implies that $\hat{y}_{i,1}$ varies linearly with $\varepsilon_i$. Apparently, when $\varepsilon_i \to 0$, then $y_{i,1}$ will become exact. Intuitively, by direct differentiation of $y_{i,1}$ with respect to $\varepsilon_i$, the following results are obtained:

if $\varepsilon_i \in i$, then $\frac{dy_{i,1}}{d\varepsilon_i} = 1$ \hspace{1cm} (7.36a)

if $\varepsilon_i \notin i$, then $\frac{dy_{i,1}}{d\varepsilon_i} = 0$ \hspace{1cm} (7.36b)
in which \( e_i \) denotes \( e_i \) is associated with the \( i \) th string, for example, if \( e_i \) is associated with \( y_{i1} \) only, then

\[
\frac{dy_{i1}}{de_i} = 1
\]

otherwise,

\[
\frac{dy_{i1}}{de_i} = 0, \quad i = 2, \ldots, S_N
\]

The term \( dy_{i1}/de_i \) in (7.36) is interpreted as the variation in \( y_{i1} \) due to a unit change of \( e_i \). The use of (7.36) will be described later in the sensitivity analysis.

Having identified the cause of error \( e_i \) in the distribution of flow, the sensitivity analysis can begin. The re-distribution of flow due to \( e_i \) can be described by

\[
\frac{dQ_{/i}}{de_i}, \quad i = 1, 2, \ldots, S_N
\]  

(7.37)

It describes the variation in \( Q_i \) due to a unit change of \( e_i \). It must be noted that \( e_i \) is associated with \( y_{i1} \) only and is independent of \( y_{i2} \). By analogy with the previous example, the change in head-loss along a string with respect to \( e_i \) can be found by the chain rule, and the expansion is given below:

\[
\frac{dh_i}{de_i} = \frac{\partial h_i}{\partial y_{i1}} \frac{dy_{i1}}{de_i} + \sum_{j=1}^{l} \frac{\partial h_i}{\partial q_j} \frac{dq_j}{de_i}; \quad i = 1, 2, \ldots, S_N
\]  

(7.38)

It is seen that terms in \( y_{i2} \) are not involved in (7.38), as compared with (7.28), because \( y_{i2} \) is the control section at which the computation of a water surface profile begins. Therefore, two unknowns \( dh_i/de_i \) and \( dq_j/de_i \) appear in (7.38). As in (7.29), the former is eliminated by equating
Sensitivity of Network Flow Distribution to Water Level

\[ \frac{\partial h_i}{\partial e_s} - \frac{\partial h_f}{\partial e_s} = 0; \quad i = 1, 2, \ldots, L_N \] (7.39)

in order to maintain the solution condition that \( f_i = 0 \) for \( i = 1, 2, \ldots, L_N \). Using condition (7.39) to deduce (7.38) the resulting equation becomes

\[ \delta = -J^{-1}f \] (7.40)

in which \( f \) and \( \delta \) denote column vectors and \( J \) denotes a Jacobian matrix.

\[ [f] = \left[ \frac{\partial h_i}{\partial y_{i,1}} \frac{dy_{i,1}}{de_s} \frac{\partial h_{f,i+1}}{\partial y_{i,1}} \frac{dy_{i+1,1}}{de_s} \right] \]

\[ [J] = \left[ \frac{\partial h_i}{\partial q_j} \frac{\partial h_{f,i+1}}{\partial q_j} \right] \]

\[ [\delta] = \left[ \frac{dq_j}{de_s} \right] \]

in which the total derivatives \( dy_{i,1}/de_s \) are evaluated using (7.36). As in (7.32), the partial derivatives are evaluated as

\[ \frac{\partial h_i}{\partial y_{i,1}} = 1 - F_{r_{i,1}}^2 \]

\[ \frac{\partial h_i}{\partial y_{i,2}} = 1 - F_{r_{i,2}}^2 \]

\[ \frac{\partial h_i}{\partial q_j} = \pm \frac{(Q_i \pm q_j)}{gA_{i,1}^2} \pm \frac{(Q_j \pm q_i)}{gA_{i,2}^2} \]

Again, (7.40) resembles the Newton-Raphson iterative formula. If a network solution is available, then the \(-J^{-1}\) is already evaluated, depending upon the solution method adopted. The only unknown in (7.40) is the \( dq_j/de_s \), which is interpreted as the flow redistribution due to a unit change of \( e_s \). This is the prime objective of this analysis.
As the given problem is interested in $dQ_i/d\varepsilon$, the solution takes the form of

$$\frac{dQ}{d\varepsilon} = \begin{bmatrix} \frac{dQ_1}{d\varepsilon} \\ \frac{dQ_2}{d\varepsilon} \\ \frac{dQ_3}{d\varepsilon} \end{bmatrix} = \begin{bmatrix} \frac{dq_1}{d\varepsilon} \\ \frac{dq_2}{d\varepsilon} \\ \frac{dq_3}{d\varepsilon} \end{bmatrix}$$

(7.41)

A positive value of $dQ_i/d\varepsilon$ means that the flow, $Q_i$, in the $i$th string will be increasing with $\varepsilon$. A large value of $dQ_i/d\varepsilon$ means that the flow, $Q_i$, in the $i$th string is highly sensitive to the increasing $\varepsilon$.

In order to illustrate the mechanism of this analysis, the network shown in Figure 7.2 is used. Note the only difference between this and the previous illustration is the way to evaluate the vector functions in (7.33) and (7.40). As previously, the network is first solved iteratively using the BFGS method. Once the solution is obtained, the Jacobian matrix $-J^{-1}$ which appears in (7.40) can be depicted directly from the solution, that is,

$$H_n = -J^{-1} = \begin{bmatrix} 10.51166 & 6.09058 & 2.46647 \\ 6.09058 & 17.22700 & 9.28658 \\ 2.46647 & 9.28658 & 18.70512 \end{bmatrix}$$

The $f$ in (7.40) is evaluated using the final flow distribution obtained and the evaluation is given below:

$$f = \begin{bmatrix} (0.98540)(1.0)-(0.98037)(1.0) \\ (0.98037)(1.0)-(0.97658)(1.0) \\ (0.97658)(1.0)-(0.97365)(1.0) \end{bmatrix} = \begin{bmatrix} 0.00503 \\ 0.00379 \\ 0.00293 \end{bmatrix}$$

The calculation assumes that the upstream water depth $\gamma_{ui}$ for each channel is subject to error $\varepsilon$, introduced in the solution during the process of numerical integration. The term $dy_{ui}/d\varepsilon$ in (7.40) is evaluated using (7.36a). It must be stressed that the actual value $\varepsilon$ is not yet incorporated in the analysis because it is trying to establish the effects of the
flow distribution due to a unit change of $\varepsilon_z$. Substituting $\mathbf{J}$ and $\mathbf{f}$ into (7.40), and solving for $\delta$ yields

$$
\delta = -\mathbf{J}^{-1}\mathbf{f} = \begin{bmatrix} 10.51166 & 6.09058 & 2.46647 & 0.00503 \\ 6.09058 & 17.22700 & 9.28658 & 0.00379 \\ 2.46647 & 9.28658 & 18.70512 & 0.00293 \end{bmatrix} \begin{bmatrix} 0.08318 \\ 0.12314 \\ 0.10241 \end{bmatrix}
$$

The $\delta$ obtained above describes the flow re-distribution due to a unit change of $\varepsilon_z$. Hence, the $dQ_i/d\varepsilon_z$ for each string is evaluated below using (7.41)

$$
\frac{dQ_i}{d\varepsilon_z} = \begin{bmatrix} -0.08318 \\ 0.08318 - 0.12314 \\ 0.12314 - 0.10241 \\ 0.10241 \end{bmatrix} = \begin{bmatrix} -0.03996 \\ -0.08318 \\ 0.02073 \\ 0.10241 \end{bmatrix}
$$

in which the negative sign denotes $Q_i$ decreases as $\varepsilon_z$ increases.

To verify the sensitivities obtained above, the network was re-run by deducting $\varepsilon_z$ from $\gamma_o$. $\varepsilon_z = -10^{-6}$ (m) is chosen for this analysis because this is a reasonably value to be introduced in the solution during the integration of the gradually varied flow equation, providing the step size is carefully chosen. The intention was to simulate the situation where numerical errors can be eliminated so that the final flow distribution can be correctly determined. The solution of the re-run was obtained, and the sensitivity of flow for each channel is tabulated in Table 7.7. The heading 'Actual' denotes the actual $dQ_i/d\varepsilon_z$ obtained by the technique of finite differencing and using those network solutions obtained from the two independent computer runs. The heading 'Theoretical' denotes the theoretical $dQ_i/d\varepsilon_z$ calculated by (7.40). Because the unit of $Q_i$ is cubic metre per second, whilst the $\varepsilon_z$ is in metre, it appears that the unit for $dQ_i/d\varepsilon_z$ is cubic metre per second per metre ($m^3/s/m$). It is, therefore, interpreted as the rate of change of $Q_i$ due to a unit change of $\varepsilon_z$.

On comparing the Actual and Theoretical results shown in Table 7.7, the latter was considerably under-estimated, and by as much as 72% for the worst case. This has been explained in previous example; the equations involved are non-linear, and the model
SENSITIVITY OF NETWORK FLOW DISTRIBUTION TO WATER LEVEL

(7.40) is linearised. However, the qualitative behaviour of the network was again determined correctly. This can be examined through the numerical signs of the figures shown in Table 7.7; or it can be justified by comparing the direction \( \theta \) (see Section 7.3) to which the Theoretical results deviated away from the Actual results. The angle is found to be

\[ \theta = 6.00^\circ \]

The above result not only validates the use of (7.40), but also it has proven that the qualitative behaviour of the network was well predicted.

On comparing the results obtained from the previous example (see Section 7.3) and this example, the extent of flow re-distribution due to the two different parameters is noticeably different. The Actual results shown in Table 7.5 suggest that the network is comparatively insensitive to the changes made in \( H_{\text{Outlet}} \). The Actual results shown in Table 7.7 suggest that the network is comparatively more sensitive to the changes made in \( e_r \). For the examples analysed, it appears that effects due even to small errors in the computation of the water surface profile have a significantly greater effect on the flow re-distribution than do changes to downstream water level. Comparison of Tables 7.5 and 7.7 indicates that the sensitivity is between 5 and 6 times greater for errors due to flow profile computation for this example. It is understood that this analysis is limited, but it perhaps serves to indicate that careful attention needs to be paid to errors in flow profile computation when factors affecting flow re-distribution are being considered.

Because the change in upstream parameter is of particular importance, a further example is included to illustrate the sensitivity of the network. Suppose the integration for water surface profile for all channels are exact, except for Channel-1. This means that the \( e_r \) will be introduced in \( y_{11} \) but not other channels. The intention is to simulate the situation where a relatively high numerical error exists in one part of the network. As in previous example, the \( f \) is evaluated using (7.40) to give

\[
\begin{bmatrix}
(0.98540)(1.0) & -(0.98037)(0.0) \\
(0.98037)(0.0) & -(0.97658)(0.0) \\
(0.97658)(0.0) & -(0.97365)(0.0)
\end{bmatrix}
\begin{bmatrix}
0.98540 \\
0.00000 \\
0.00000
\end{bmatrix}
\]
SENSITIVITY OF NETWORK FLOW DISTRIBUTION TO WATER LEVEL

Note the zero entries in the above evaluation denotes the water surface profile is computed exactly. The \( y_{i,i} \) for \( i=2,3,...,S \) are independent of \( \varepsilon_r \), and therefore \( dy_{i,i}/d\varepsilon_r \) in (7.40) is evaluated using (7.36b) to give zero entries. It must be stressed that the actual value \( \varepsilon_r \) is not yet incorporated in the analysis because it is trying to establish the effects of the flow distribution due to a unit change of \( \varepsilon_r \). Substituting \( J \) and \( f \) into (7.40), and solving for \( \delta \) yields

\[
\delta = -J^{-1}f = \begin{bmatrix} 10.51166 & 6.09058 & 2.46647 & 0.98540 \\ 6.09058 & 17.22700 & 9.28658 & 0.00000 \\ 2.46647 & 9.28658 & 18.70512 & 0.00000 \end{bmatrix} = \begin{bmatrix} 10.35819 \\ 6.00166 \\ 2.43046 \end{bmatrix}
\]

The \( \delta \) obtained above describes the quantity of flow re-distribution due to a unit change of \( \varepsilon_r \). Hence, the \( dQ_i/d\varepsilon_r \) for each string is evaluated below using (7.41):

\[
\frac{dQ}{d\varepsilon_r} = \begin{bmatrix} -10.35819 \\ 10.35819-6.00166 \\ 6.00166-2.43046 \\ 2.43046 \end{bmatrix} = \begin{bmatrix} -10.35819 \\ 4.35653 \\ 3.57120 \\ 2.43046 \end{bmatrix}
\]

It must be noted that the above analysis can be carried out for other channels with the \( \varepsilon_r \) be included in the selected \( y_{i,i} \).

To verify the above results, the actual sensitivity for the above example was calculated and is shown in Table 7.8. The points observed in the previous examples generally apply here. However, the sensitivity obtained was found to be much higher than these sensitivity obtained from previous examples. This may be due to the relative high error incurred in the \( y_{i,i} \) in one part of the network. For example, using the Actual results shown in Table 7.8, a \( \varepsilon_r = 10^{-3} \) (m) error can incur

\[
\Delta Q = \begin{bmatrix} -0.02338 \\ 0.005538 \\ 0.00777 \\ 0.01023 \end{bmatrix} (m^3/s)
\]
Sensitivity of network flow distribution to water level

or in term of percentages

\[
\frac{\Delta Q}{Q} = \begin{bmatrix} -10.026 \\ 1.332 \\ 1.324 \\ 1.318 \end{bmatrix} (\%) 
\]

of flow incorrectly distributed. Although this \( \varepsilon_r = 10^{-3} \) (m) variation in water depth may be regarded as insignificant in practice, this has shown the significance of numerical errors involved in the computation of the water surface profile on the final distribution of flow throughout the network. This suggests that higher computational criteria, such as \( \varepsilon_r < 10^{-6} \) (m), must be enforced in order to keep the computation as accurate as possible. It has also shown that the final flow distribution is very sensitive to changes in upstream parameters such as \( \varepsilon_r \) or \( y_i \).

From this illustration alone, it is seen that the accuracy of the computation, whether it is for a water surface profile or general purposes of computation, has significant impact on the final flow distribution. A small error in upstream parameters can affect the final flow distribution significantly. It is therefore vitally important to minimise such errors during numerical computation.
## Sensitivity of Network Flow Distribution to Water Level

### Table 7.7. Sensitivity of flow re-distribution, $\varepsilon_s = 10^{-6}$ (m) (all channels).

<table>
<thead>
<tr>
<th>Strings</th>
<th>$Q_s$ (m$^3$)</th>
<th>$Q_j$ (m$^3$)</th>
<th>$Q_j - Q_s$ (m$^3$)</th>
<th>$\frac{dQ}{de_s}$, (m$^3$/s/m)</th>
<th>Difference (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inflow</td>
<td>2.00000</td>
<td>2.00000</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Channel -1</td>
<td>0.23316</td>
<td>0.23316</td>
<td>2.19E-07</td>
<td>-0.021900</td>
<td>-62.0</td>
</tr>
<tr>
<td>Channel -2</td>
<td>0.40397</td>
<td>0.40397</td>
<td>1.44E-07</td>
<td>-0.014400</td>
<td>-72.3</td>
</tr>
<tr>
<td>Channel -3</td>
<td>0.58665</td>
<td>0.58665</td>
<td>-4.51E-08</td>
<td>0.04500</td>
<td>-53.9</td>
</tr>
<tr>
<td>Channel -4</td>
<td>0.77622</td>
<td>0.77622</td>
<td>-3.18E-07</td>
<td>0.31800</td>
<td>-67.8</td>
</tr>
<tr>
<td>Outfall</td>
<td>2.00000</td>
<td>2.00000</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>

### Table 7.8. Sensitivity of flow re-distribution, $\varepsilon_s = 10^{-6}$ (m) (Channel-1 only).

<table>
<thead>
<tr>
<th>Strings</th>
<th>$Q_s$ (m$^3$)</th>
<th>$Q_j$ (m$^3$)</th>
<th>$Q_j - Q_s$ (m$^3$)</th>
<th>$\frac{dQ}{de_s}$, (m$^3$/s/m)</th>
<th>Difference (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inflow</td>
<td>2.00000</td>
<td>2.00000</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Channel -1</td>
<td>0.23316</td>
<td>0.23318</td>
<td>2.338E-05</td>
<td>-23.387800</td>
<td>-55.7</td>
</tr>
<tr>
<td>Channel -2</td>
<td>0.40397</td>
<td>0.40396</td>
<td>-5.380E-06</td>
<td>5.38000</td>
<td>-19.0</td>
</tr>
<tr>
<td>Channel -3</td>
<td>0.58665</td>
<td>0.58664</td>
<td>-7.768E-06</td>
<td>7.76800</td>
<td>54.0</td>
</tr>
<tr>
<td>Channel -4</td>
<td>0.77622</td>
<td>0.77621</td>
<td>-1.023E-05</td>
<td>10.23100</td>
<td>76.2</td>
</tr>
<tr>
<td>Outfall</td>
<td>2.00000</td>
<td>2.00000</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>
7.5 Concluding Remarks

The objective of this Chapter is to study the sensitivity of the behaviour of the networks and the flow re-distribution within networks in response to changes in network variables. Two models were derived to determine the effects on the final flow distribution. The first investigated the effects of varying the head at the downstream outfall point; the second investigated the effects of error propagation from the numerical integration of open channel profiles on network flow distribution. The following points emerged from the numerical experiments.

i) The theoretical models predicted the qualitative behaviour of the network sensitivity correctly.

ii) The theoretical models largely under-estimated the re-distribution of flow. A 73% under-estimation was obtained for the worst test case used. This was believed to be because the actual equations are non-linear and the theoretical models were linearised.

iii) The flow distribution is more sensitive to the accuracy of numerical computations for the water surface profile than to changes in the downstream head.

iv) Whilst the numerical predictions from the models may not be sufficiently reliable for detailed predictions, they are nevertheless useful in assessing order-of magnitude changes to network flow distribution.
CHAPTER 8

CONCLUSIONS AND RECOMMENDATIONS
8 CONCLUSIONS AND RECOMMENDATIONS

8.1 Conclusions

A model for analysing flow distributions throughout open channel hydraulic networks was developed using the steady state principles of head-loss and continuity of flow in conjunction with the loop formulation. The model is generalised, and can be used to analyse many hydraulic systems, including water and wastewater treatment works; irrigation and drainage systems.

The basic 'building blocks' of the model are strings and modules. A module is any hydraulic structure or feature with a defined depth to discharge relationship; a string is a series of linked modules and is the basic component required to formulate the system of equations which models the hydraulic network. The method of 'loop formulation' is used to relate discharge and head-loss along the strings. The use of strings is the key to the successful generalisation of the model because strings are treated as discrete elements. Any changes which are to the network made involve altering parameters only within strings. This, therefore, does not affect the overall formulation of the system network equations. When strings are used in conjunction with loop formulation, it is possible to handle all types of hydraulic structures, and to analyse most flow conditions.

An important factor which affects the final flow distribution in a network is the accuracy of the calculation of the water surface profiles along the associated strings which incorporate open channels. The first part of the research therefore investigated the problem of water surface profile computation along open channels. The gradually varied flow equation which describes the flow profile cannot be explicitly integrated, other than under restrictive conditions, and therefore numerical solutions are required. The main disadvantage of numerical solutions is that numerical errors are introduced into the final
CONCLUSIONS AND RECOMMENDATIONS

solution so that an incorrect water surface profile may be obtained. This, in turn, causes an incorrect final distribution of flow throughout the network.

To investigate the accuracy of numerically calculated water surface profiles, an error analysis was used to study the behaviour of error growth as well as the stability of the numerical methods when the gradually varied flow equation is numerically integrated. Five categories of numerical methods for water profile computations were included in the investigations, namely, the Standard Step method, the Trapezoidal method the fourth-order Runge-Kutta method, the fourth-order Merson method and the fifth-order Butcher method. Of these methods, the Runge-Kutta method proved to be the most efficient in terms of numerical accuracy and number of function evaluations. The numerical errors did not grow as the integration steps proceeded; the errors were found to decay providing that the direction of computation, that is upstream for subcritical flow and downstream for supercritical flow, is being correctly exercised. The application of the fourth-order Runge-Kutta method is stable and no limitations on its application to solving the gradually varied flow equation were discovered. In conjunction with the Runge-Kutta method, adaptive step size control was investigated. Based on theoretical estimations of the local error, the step size for integration was adjusted accordingly. The global error obtained in the examples did not grow as the integration proceeded, and it was confined within the theoretical error bounds predicted.

The application of theoretical error bounds to the numerical calculation procedures was also investigated. It was found that the unstable bound was too conservative for any useful applications in adaptive step size control. On the other hand, although the stable bound was found to be an over-estimate, it gave an order-of-magnitude estimation and is therefore provides a useful guide to predetermine the step size prior to integration of the gradually varied flow profile.

The determination of the flow distribution within hydraulic networks requires the formulation and solution of a set of simultaneous equations which describe head-loss throughout the network. Several methods were investigated. In order to test the appropriateness of these methods, two networks were set up, one involving parallel strings and the other involving nested strings. Numerical experiments were carried out on both. A method commonly applied to solve the equations is the Newton-Raphson method. The concept of the method is simple and in practice it has a desirable rate of convergence; however, it suffers from three main drawbacks. Firstly, it requires the first-
order partial derivatives of the vector function. Obtaining these derivatives for strings containing open channels is extremely complex to achieve by analytical means. Further investigations were carried out to investigate numerical means of estimating these partial derivatives, but it was found that the resulting convergence of the method was highly dependent upon the step size chosen for the estimation. Secondly, the Newton-Raphson method requires the inversion of the Jacobian matrix which is a complex operation. Finally, the Newton-Raphson method may not converge if the initial approximation is far from the final solution.

To overcome these problems, a class of methods known as the quasi-Newton methods including the Broyden method, the DFP (Davidon, Fletcher and Powell) method and the BFGS (Broyden, Fletcher, Goldfarb and Shanno) method were investigated. The performance of the methods was found to depend on the sensitivity of the network. A 'sensitive' network is one in which a small change in head-loss causes a large change in flow distribution. If the network was not sensitive, then the Broyden method and especially the DFP method were very inefficient. If the network was sensitive, the performance of both Broyden and DFP methods was comparable to the BFGS method. The BFGS method was found to be stable and to possesses superior overall performance compared with the other two methods in the examples studied.

When the BFGS was compared to the Newton-Raphson method, the former converged more slowly than the latter method over a given number of iterations. This was because the approximated Jacobian matrix used in the BFGS method was not estimated sufficiently accurately. Despite the greater the number of iterations required by the BFGS method, it was in fact more efficient than the Newton-Raphson method when measured in terms of overall computational effort and accuracy. This was because the Newton-Raphson method required additional function evaluations to approximate the partial derivatives in the Jacobian matrix. The BFGS method improved the overall performance over the Newton-Raphson method by as much as 50%. The performance of the Newton-Raphson method deteriorated as the initial approximation moved further away from the final solution, whereas the BFGS method seemed to be unaffected by the accuracy of the initial approximation. The BFGS method was the best method adopted for hydraulic network analysis involving open channel and other hydraulic modules in this research under the conditions investigated.
'Switching of hydraulic controls' is a problem which arises from the fact that the flow regime along a string may change from subcritical to supercritical or vice-versa as iterative computation proceeds. This change in flow regime causes a switch in the equations which govern the flow conditions. If this switching happens frequently, serious stability problems with network analysis might be expected. By means of examples, it was shown that switching of control did cause serious problem with convergence. In order to overcome this problem, the technique of line searching was investigated. This technique is a sub-problem of optimisation (minimisation) which is designed to be imbedded in any iterative procedure. The main advantage of a line search is to prevent divergence of the iterative procedures, and hence to stabilise the iterative method in which it is imbedded. When the line search was used, the solution process successfully converged in circumstances where it had previously failed, as described above. However, this success was offset by the considerably greater computational effort required. If the process converges, then there is no need to apply the line search and computational effort can then be minimised. Therefore, it is suggested to use the line search only when divergence is encountered, or if there exists a very slow rate of convergence.

The ability to simulate variable net outflow from a network, for example through a side discharge structure, is important for many applications, such as wastewater treatment plants. The particular problem associated with side-discharge weirs is that it introduces an additional variable (namely, an outflow which cannot be determined a priori) into the system equations. This gives rise to insufficient number of equations to achieve a network solution. To tackle this problem, the concept of a virtual string was investigated. The virtual string has predefined flow characteristics and introduces a dummy loop, thereby providing an additional equation so that the system can be solved. When a virtual string is inserted, the network equations can be solved directly; otherwise, recourse has to be made to some other external procedure whereby another complete layer of trial-and-error iteration has to be imposed. Examples showed that the technique can be applied successfully and has no adverse effect on the rate of convergence. Any differences in rate of convergence obtained were due to the sensitivity of flow distribution to head-losses in the network. The concept was tested using both subcritical and supercritical flow conditions along the side-weir, and also for the situation in which a hydraulic jump occurred. The virtual string successfully modelled two general situations when either total inflow or total outflow to the network were known. This finding is vital as far as solvability and rate of convergence is concerned. It is concluded that the virtual string is
suitable for modelling lateral discharge structures along with various network flow situations.

In many situations, the sensitivity of the flow distribution within the network to changes made in network parameters is of particular importance. A sensitivity analysis may be used to give some general indications of how the network behaves. Two theoretical models were developed to study the network behaviour. The first investigated the effects on the flow distribution of varying the head at the downstream point of outfall; the second investigated the effects caused by errors introduced during the process of numerical integration of the gradually varied flow equation for open channel flow. It was found that both models predicted the qualitative behaviour of the example networks correctly, although the numerical predictions were under-estimated by as much as 72% for the worst test case. This is believed to be because the network equations are non-linear and the models developed are linearised. It is concluded that the models predict network behaviour qualitatively rather than quantitatively. The analysis also showed that the flow distribution is relatively insensitive to the changes made in downstream head. For example, a one-millimetre change (corresponding to 0.13%) in downstream head only caused about 0.02% re-distribution of flow. On the other hand, the flow distribution is much more sensitive to changes made in upstream parameters as a result of error in flow profile computation. For example, an one-millimetre error (corresponding to 0.14%) numerical error introduced in upstream water depth can cause a 10% re-distribution of flow. This suggests that careful attention needs to be paid to understanding and controlling the errors introduced by numerical calculation procedures.

8.2 Recommendations for Future Work

An algorithm has been developed to solve complex hydraulic networks incorporating open channels. However, it is clear that improvements in efficiency and performance might be obtained.

i) Although the accuracy of the initial approximation required by the BFGS method is less stringent than that of the Newton-Raphson method, a good approximation to the final solution is still desirable. The better the initial approximation, the more rapidly the method converges to the final solution. It would be useful to investigate
CONCLUSIONS AND RECOMMENDATIONS

subsidiary methods to determine the initial approximation before the commencing the main solution algorithm process.

ii) The procedures used to perform the line searches are considered to be inefficient and time consuming to perform. It is therefore useful to enhance the performance of the line searches. Other methods, such as the Fibonacci section search exist, and their application to open channel hydraulic network analysis requires detailed investigation.

iii) The existence of discontinuities in the loop equations caused by different flow conditions may be classified as a problem of non-smooth optimisation. Methods are available to solve this class of problem, and it would be useful to compare these methods with the BFGS method (with line searches imbedded) in order to determine which approach is best for tackling the switching of hydraulic controls.

iv) As a long term goal it would be useful to study the way in which the algorithms developed herein for steady state flow could be incorporated into a time-varying model, given the current importance of real-time modelling of certain types of system such as catchment and drainage systems.
REFERENCES
REFERENCES


BAIPAI A. C., MUSTOE L. R. AND WALKER D. (1977); Advanced Engineering Mathematics, 1st edition; John Wiley and Sons Ltd.

BALMFORTH D. J. AND SARGINSON E. J. (1977); Discussion of 'Experimental Investigation of Flow over Side-Weirs' by El-Khashab and Smith; Journal of the Hydraulics Division, ASCE, 941-943.


BROYDEN C. G. (1965); A Class of Methods for Solving Non-linear Simultaneous Equations; Mathematics of Computation, 1819-1826.


CHADWICK A. J. AND MORFETT J. C. (1986); Hydraulics in Civil Engineering, 1st edition; George Allen & Unwin Ltd.
REFERENCES


COLLATZ L. (1960); The Numerical Treatment of Differential Equations, 3rd edition; Springer-Verlag.


REFERENCES

FLETCHER R. (1987); Practical Methods of Optimisation, 2nd edition; John Wiley and Sons Ltd.


GILL M. A. (1976); Exact Solution of Gradually Varied Flow; Journal of Hydraulics Division, ASCE, Vol. 102, 1353-1364.


KESAVAN H. K. AND CHANDRASHEKAR M. (1972); Graph-Theoretic Models for Pipe Network Analysis; Journal of the Hydraulics Division, ASCE, Vol. 98, 345-364.


LAKSHMANA N. S. AND SRIDHARAN K. (1971); Effect of Channel Shape on Gradually Varied Flow Profiles; Journal of the Hydraulics Division, ASCE, Vol. 97, 55-64.
REFERENCES


LOTKIN M. (1951); On the Accuracy of Runge-Kutta's Method; MTAC, Vol. 5, 128.


MCBEAN E. A. AND PERKINS F. E. (1975a); Convergence Scheme in Water Profile Computation; Journal of the Hydraulics Division, ASCE, Vol. 101, 1380-1384.


RALSTON A. AND WILF H. S. (1960); Mathematical Methods for Digital Computer, 1st edition; John Wiley and Sons Ltd.


STROUSTRUP B. (1987); The C++ Programming Language, 1st edition; Addison-Wesley.


REFERENCES


APPENDIX A

DETAILED CALCULATION PROCEDURES
FOR SOLUTION ALGORITHMS
A DETAILED CALCULATION PROCEDURES FOR SOLUTION ALGORITHMS

A.1 System of Equations

The first step in analysing an open channel network is to decompose the network into strings and loops. With the network shown in Figure 5.3, four strings are identified and three loops are traced, and the physical characteristics of each string are shown in Table 5.4. With this configuration, the system equations are written as follows:

\[
\begin{align*}
  f_i &= h_p - h_{p+1} = 0, \\
  s &= 1, 2, 3
\end{align*}
\]

in which \( h_p \) denotes the head-loss associated with the \( s \) th string; and the \( f_i \) denotes the sum of head-losses around the closed loop \( l \). \( h_f \) can be found by using the energy equation to give

\[
h_f = H_{s1} - H_{s2}, \quad s = 1, 2, 3, 4
\]

in which \( H_{s1} \) and \( H_{s2} \) denote energy heads upstream and downstream of the \( s \) th string respectively. To obtain the solution of this network, an initial distribution of flow is required to start the computations; the total head at outfall point, \( H_{outfall} \), must be provided as a boundary condition. It follows that, by the law of conservation of energy, the total head at the downstream end of strings must be equal to

\[
H_{s2} = H_{outfall}, \quad s = 1, 2, 3, 4
\]
One assumption made in the above expression is that the minor losses are neglected. Since the $H_{s,2}$ is known, it follows that the water depth $y_{s,2}$ can also be calculated. With all the information available, the water surface profile along each string can be determined, and hence, the $H_{s,1}$. The procedure to determine a water surface profile is described in Chapter 2.

With the boundary condition given, the system of equations can be solved iteratively using the procedures described in Chapter 4. They are the NR, BR, DFP and BFGS methods, and the mode of operation of the methods is described separately in the following sections.

A.2 Newton-Raphson Method

The following example illustrates the use of the NR method for solving the parallel network (see Figure 5.3 and Table 5.4).

The procedure starts with the initial flow distribution $Q_n$ for $n = 0$,

$$Q_0 = \begin{bmatrix} 0.50000 \\ 0.50000 \\ 0.50000 \\ 0.50000 \end{bmatrix}$$

The head-loss along each string are calculated from the flow profile calculations. Figure A.1 shows the $h_f$ values along each string. Thus, the vector $f_0$ is assembled for $Q_0$ as shown.

$$f_0 = \begin{bmatrix} 0.00963-0.00304 \\ 0.00304-0.00139 \\ 0.00139-0.00077 \\ 0.00077-0.00062 \end{bmatrix} = \begin{bmatrix} 0.00659 \\ 0.00165 \\ 0.00062 \end{bmatrix}$$
Figure A.1. Head-losses evaluated with the initial approximation \( Q_0 \).
Performing the appropriate partial differentiation (refer to the finite differencing described in Section A.2.1), the $J_0$ is obtained to give

$$J_0 = \begin{bmatrix} -0.05162 & 0.01228 & 0.00000 \\ 0.01228 & -0.01787 & 0.00558 \\ 0.00000 & 0.00558 & -0.00868 \end{bmatrix}$$

and its inverse is

$$H_0 = -J_0^{-1} = \begin{bmatrix} 24.35712 & 20.95973 & 13.48638 \\ 20.96008 & 88.08723 & 56.67905 \\ 13.48708 & 56.68108 & 151.69622 \end{bmatrix}$$

Substituting $f_0$ and $H_0$ into (4.8) yields

$$q_0 = f_0 \begin{bmatrix} 24.35712 & 20.95973 & 13.48638 \\ 20.96008 & 88.08723 & 56.67905 \\ 13.48708 & 56.68108 & 151.69622 \end{bmatrix} \begin{bmatrix} 0.00659 \\ 0.00165 \\ 0.00062 \end{bmatrix}$$

Putting $t_0 = 1$, a full step NR method, $q_0$ is computed to give

$$q_0 = \begin{bmatrix} 0.20347 \\ 0.31846 \\ 0.27594 \end{bmatrix}$$

and thence a new set of flow distributions

$$Q_1 = \begin{bmatrix} 0.50000 & 0.20347 \\ 0.50000 & -0.20347 + 0.31846 \\ 0.50000 & -0.31846 + 0.27594 \end{bmatrix} = \begin{bmatrix} 0.29653 \\ 0.38501 \\ 0.54252 \end{bmatrix}$$
The procedure is continued for \( n = 1 \)

\[
q_1 = t_1 h_1 f_i
\]

in which \( f_i \) is recalculated from \( Q_i \) as follows. The flow profiles along the channels comprising each string are calculated using the revised flow estimate \( Q_1 \). Thus, \( f_i \) is given by

\[
f_i = \begin{bmatrix}
0.00155 \\
0.00016 \\
-0.00023
\end{bmatrix}
\]

and \( h_1 = J^{-1}_i \) is obtained by the technique of finite differencing. Hence, \( q_1 \) is given by

\[
q_1 = t_1 \begin{bmatrix}
40.33387 & 31.30612 & 17.37875 \\
31.30674 & 107.12340 & 59.46667 \\
17.37966 & 59.46864 & 124.40057
\end{bmatrix} \begin{bmatrix}
0.00155 \\
0.00016 \\
-0.00023
\end{bmatrix}
\]

Again, putting \( t_1 = 1 \),

\[
q_1 = \begin{bmatrix}
0.06335 \\
0.05154 \\
0.00735
\end{bmatrix}
\]

and thence,

\[
Q_2 = \begin{bmatrix}
0.23318 \\
0.39682 \\
0.58672 \\
0.78328
\end{bmatrix}
\]

The procedure is continued until the condition \( \|f_i\| < \xi \) is met. These results together with those from subsequent application of the procedure are tabulated in Table 5.6a.
A.2.1 Partial Derivatives for Newton-Raphson Method

The method of NR requires the partial derivatives of \( f_n \) with respect to \( q_n \) to be assembled in the Jacobian matrix \( J_n \),

\[
\begin{bmatrix}
\frac{\partial f_i}{\partial q_j}
\end{bmatrix}, \quad \{i = 1, 2, \ldots, L\}, \quad \{j = 1, 2, \ldots, L\}
\]

Each term in the matrix is an estimate of the change in head-loss around a closed loop caused by a variation in loop flow correction factor, \( q_r \). As described in Chapter 4, the evaluation of \( J_n \) analytically can be very laborious. In view of this, the most straightforward idea is just to estimate the partial derivatives by the method of finite difference. The forward difference formula (4.35) is chosen for evaluating the Jacobian matrix because of its simplicity. Unfortunately, the method is expensive to perform because it requires \( f \) to be evaluated twice with \( x \) and \( x+h \). Other considerations include what size of \( h \) should be chosen to minimise numerical errors. Assuming that reasonable decisions are made, then the procedure shown below can be used to approximate the Jacobian matrix.

Using the example (see Figure 5.3 and Table 5.4) and assuming that the initial network flow, \( Q_0 \), is evenly distributed, the head-losses around the closed loops are given by

\[
\begin{bmatrix}
\hat{f}_1 \\
\hat{f}_2 \\
\hat{f}_3
\end{bmatrix} = \begin{bmatrix}
h_{i1} - h_{f1} \\
h_{i2} - h_{f2} \\
h_{i3} - h_{f3}
\end{bmatrix} = \begin{bmatrix}
0.00659255 \\
0.00165001 \\
0.00061635
\end{bmatrix}
\]

in which the superscript \( (0) \) denotes that the evaluations involves all the \( q_j \). This step corresponds to the \( f(x) \) in the forward difference formula (4.34). However, the corresponding result of \( f(x+h) \) requires the function to be re-evaluated with \( x+h \).
Consider the sum of head-losses around the first loop when $q_1$ is increased by $h = 10^{-3}$. The $f_n(1)$ becomes

$$f_n(1) = \begin{bmatrix} 0.00659204 \\ 0.00165013 \\ 0.00061635 \end{bmatrix}$$

in which the superscript (1) denotes the evaluation one $q_1 + h$ at a time while holding the $q_2$ and $q_3$ constant. This step corresponds to the $f(x+h)$ in the forward difference formula. When $q_2$ is increased by $h$, the sum of head-losses around the second loop is given by

$$f_n(2) = \begin{bmatrix} 0.00659268 \\ 0.00164983 \\ 0.00061640 \end{bmatrix}$$

in which the superscript (2) denotes the evaluation one $q_2 + h$ at a time while holding the $q_1$ and $q_3$ constant. Likewise, the sum of head-losses around the third loop is given by

$$f_n(3) = \begin{bmatrix} 0.00659255 \\ 0.00165006 \\ 0.00061626 \end{bmatrix}$$

in which the superscript (3) denotes the evaluation one $q_3 + h$ at a time while holding the $q_1$ and $q_2$ constant. Substituting $f_n^{(1)}$, $f_n^{(2)}$ and $f_n^{(3)}$ together with $f_n^{(0)}$ into the forward difference formula, the Jacobian matrix for the $n$th iterative step is approximated to be

$$\frac{\partial f}{\partial q_1} = \frac{f_n^{(1)} - f_n^{(0)}}{h} = \frac{1}{10^{-3}} \begin{bmatrix} 0.00659204 \\ 0.00165013 \\ 0.00061635 \end{bmatrix} - \begin{bmatrix} 0.00659225 \\ 0.00165006 \\ 0.00061626 \end{bmatrix} = \begin{bmatrix} -0.05162 \\ 0.01228 \\ 0.00000 \end{bmatrix}$$
\[
\begin{bmatrix}
\frac{\partial f_1}{\partial q_1} \\
\frac{\partial f_2}{\partial q_2} \\
\frac{\partial f_3}{\partial q_3}
\end{bmatrix}
= \frac{f^{(2)}_n - f^{(0)}_n}{h} = \frac{1}{10^{-3}} \begin{bmatrix}
0.00659268 \\
0.00164983 \\
0.00061640
\end{bmatrix} - \begin{bmatrix}
0.00659255 \\
0.00165001 \\
0.00061635
\end{bmatrix} = \begin{bmatrix}
0.01228 \\
-0.01787 \\
0.00558
\end{bmatrix}
\]

\[
\begin{bmatrix}
\frac{\partial f_1}{\partial q_1} \\
\frac{\partial f_2}{\partial q_2} \\
\frac{\partial f_3}{\partial q_3}
\end{bmatrix}
= \frac{f^{(3)}_n - f^{(0)}_n}{h} = \frac{1}{10^{-4}} \begin{bmatrix}
0.00659255 \\
0.00165006 \\
0.00061626
\end{bmatrix} - \begin{bmatrix}
0.00659255 \\
0.00165001 \\
0.00061635
\end{bmatrix} = \begin{bmatrix}
0.00000 \\
0.00558 \\
-0.00868
\end{bmatrix}
\]

Combining the above column vectors to form
\[
J = \begin{bmatrix}
\frac{\partial f_1}{\partial q_1} & \frac{\partial f_1}{\partial q_2} & \frac{\partial f_1}{\partial q_3} \\
\frac{\partial f_2}{\partial q_1} & \frac{\partial f_2}{\partial q_2} & \frac{\partial f_2}{\partial q_3} \\
\frac{\partial f_3}{\partial q_1} & \frac{\partial f_3}{\partial q_2} & \frac{\partial f_3}{\partial q_3}
\end{bmatrix} = \begin{bmatrix}
-0.05162 & 0.01228 & 0.00000 \\
0.01228 & -0.01787 & 0.00558 \\
0.00000 & 0.00558 & -0.00868
\end{bmatrix}
\]

A.3 Broyden Method

The following example illustrates the use of the BR procedure for solving the parallel network (see Figure 5.3 and Table 5.4).

The steps to evaluate \( f_0 \) are identical to those described in NR procedure. From the previous illustration, the \( f_0 \) is found to be
\[
f_0 = \begin{bmatrix}
0.00659 \\
0.00165 \\
0.00062
\end{bmatrix}
\]
This method assumes that in the equation

\[ q_0 = t_0 H_0 f_0 \]

\[ H_0 = I, \text{ in which } I \text{ is an identity matrix equal to} \]

\[ I = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \]

This contrasts with the NR method where \( H_0 = J_0^{-1} \) as described previously. Substituting \( f_0 \) and \( H_0 \) into (4.8) yields

\[ q_0 = t_0 \begin{bmatrix} 1 & 0 & 0 & 0.00659 \\ 0 & 1 & 0 & 0.00165 \\ 0 & 0 & 1 & 0.00062 \end{bmatrix} \]

As with NR method, putting \( t_0 = 1 \), the value of \( q_0 \) is computed to give

\[ q_0 = \begin{bmatrix} 0.00659 \\ 0.00165 \\ 0.00062 \end{bmatrix} \]

and thence a new set of flow distributions

\[ Q_1 = \begin{bmatrix} 0.50000 & 0.00659 \\ 0.50000 & -0.00659 + 0.00165 \\ 0.50000 & -0.00165 + 0.00062 \end{bmatrix} = \begin{bmatrix} 0.49341 \\ 0.50494 \\ 0.50103 \end{bmatrix} \]

The procedure is continued for \( n = 1 \)

\[ q_1 = t_1 H_1 f_1 \]
in which

\[
\begin{bmatrix}
0.00627 \\
0.00171 \\
0.00062
\end{bmatrix}
\]

and

\[
H_1 = H_0 + A_0
\]

The \(A_0\) is given by the Broyden updating formula (4.50)

\[
A_0 = -\left( t_0 \delta_0 + H_1 \gamma_0 \right) \delta_0^T H_0 \delta_0^T H_0
\]

in which \(\delta_0 = \gamma_0\); and \(\gamma_0\) is evaluated as follows

\[
\gamma_0 = f_1 - f_0 = \begin{bmatrix} 0.00627 \\ 0.00171 \\ 0.00062 \end{bmatrix} - \begin{bmatrix} 0.00062 \\ 0.00062 \\ 0.00062 \end{bmatrix} = \begin{bmatrix} -0.00032 \\ 0.00006 \\ 0.00000 \end{bmatrix}
\]

With the \(t_0, \delta_0\) and \(\gamma_0\) are known, the \(A_0\) is computed to give

\[
A_0 = \begin{bmatrix}
20.61437 & 5.15943 & 1.92727 \\
5.60292 & 1.40232 & 0.52383 \\
2.03784 & 0.51004 & 0.19052
\end{bmatrix}
\]

and thence,

\[
H_1 = \begin{bmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 1
\end{bmatrix} + \begin{bmatrix}
20.61437 & 5.15943 & 1.92727 \\
5.60292 & 1.40232 & 0.52383 \\
2.03784 & 0.51004 & 0.19052
\end{bmatrix}
\]

\[
= \begin{bmatrix}
21.61437 & 5.15943 & 1.92727 \\
5.60292 & 2.40232 & 0.52383 \\
2.03784 & 0.51004 & 1.19052
\end{bmatrix}
\]
Thus, \( q_1 \) becomes

\[
q_1 = t_1 \begin{bmatrix}
21.61437 & 5.15943 & 1.92727 \\
5.60292 & 2.40232 & 0.52383 \\
2.03784 & 0.51004 & 1.19052
\end{bmatrix} \begin{bmatrix}
0.006527 \\
0.00171 \\
0.00062
\end{bmatrix}
\]

Putting \( t_1 = 1 \),

\[
q_1 = \begin{bmatrix}
0.14560 \\
0.03957 \\
0.01439
\end{bmatrix}
\]

and thence,

\[
Q_1 = \begin{bmatrix}
0.34781 \\
0.61097 \\
0.52621 \\
0.51501
\end{bmatrix}
\]

The procedure is continued until the condition \( \| f_n \| < \xi \) is met. These results together with those from subsequent application of the procedure are tabulated in Table 5.6b.

### A.4 Davidon-Fletcher-Powell Method

The following example illustrates the use of the DFP procedure for solving the parallel network (see Figure 5.3 and Table 5.4).

The steps to evaluate \( f_0 \) are identical to those described in NR procedure. From the previous illustration, the \( f_0 \) is found to be

\[
f_0 = \begin{bmatrix}
0.00659 \\
0.00165 \\
0.00062
\end{bmatrix}
\]
Initially, the $H_0 = I$; substituting $f_0$ and $H_0$ into (4.8) yields

$$q_0 = t_0 \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0.00659 \\ 0.00165 \\ 0.00062 \end{bmatrix}$$

As with NR method, by putting $t_0 = 1$, the value of $q_0$ is computed to give

$$q_0 = \begin{bmatrix} 0.00659 \\ 0.00165 \\ 0.00062 \end{bmatrix}$$

and hence a new set of flow distributions

$$Q_1 = \begin{bmatrix} 0.49341 \\ 0.50494 \\ 0.50103 \\ 0.50062 \end{bmatrix}$$

The procedure is continued for $n = 1$

$$q_1 = t_1 H_1 f_1$$

in which

$$f_1 = \begin{bmatrix} 0.00627 \\ 0.00171 \\ 0.00062 \end{bmatrix}$$

and

$$H_1 = H_0 + A_0 + B_0$$
The $A_0$ and $B_0$ are given by the DFP updating formula (4.51)

$$A_0 = -\frac{H_0\gamma_0\gamma_0^T H_0}{\gamma_0^T H_0 \gamma_0}$$

$$B_0 = -\frac{t_0 \delta_0 \delta_0^T}{\delta_0^T \gamma_0}$$

in which $\delta_0 = q_0$; and the $\gamma_0$ is evaluated as follows

$$\gamma_0 = f - f_0 = \begin{bmatrix} 0.00627 \\ 0.00171 \\ 0.00062 \end{bmatrix} - \begin{bmatrix} 0.00659 \\ 0.00165 \\ 0.00062 \end{bmatrix} = \begin{bmatrix} -0.00032 \\ 0.00006 \\ 0.00000 \end{bmatrix}$$

With the $t_0$, $\delta_0$ and $\gamma_0$ are known, the $A_0$ and $B_0$ are evaluated to give

$$A_0 = \begin{bmatrix} -0.97065 & 0.16836 & 0.01179 \\ 0.16836 & -0.02920 & -0.00205 \\ 0.01179 & -0.00205 & -0.00014 \end{bmatrix}$$

$$B_0 = \begin{bmatrix} 21.66100 & 5.42139 & 2.02512 \\ 5.42139 & 1.35688 & 0.50685 \\ 2.02512 & 0.50685 & 0.18933 \end{bmatrix}$$

Thus,

$$H_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -0.97065 & 0.16836 & 0.01179 \\ 0.16836 & -0.02920 & -0.00205 \\ 0.01179 & -0.00205 & -0.00014 \end{bmatrix} + \begin{bmatrix} 21.66100 & 5.42139 & 2.02512 \\ 5.42139 & 1.35688 & 0.50685 \\ 2.02512 & 0.50685 & 0.18933 \end{bmatrix} = \begin{bmatrix} 21.69034 & 5.58975 & 2.03691 \\ 5.58975 & 2.32768 & 0.50481 \\ 2.03691 & 0.50481 & 1.18919 \end{bmatrix}$$
The $q_l$ becomes

\[
q_l = \begin{bmatrix}
21.69034 & 5.58975 & 2.03691 & 0.00627 \\
5.58975 & 2.32768 & 0.50481 & 0.00171 \\
2.03691 & 0.50481 & 1.18919 & 0.00062 \\
0.00627 & 0.00171 & 0.00062 & 1
\end{bmatrix}
\]

Putting $t_l = 1$,

\[
q_l = \begin{bmatrix}
0.14688 \\
0.03935 \\
0.01438
\end{bmatrix}
\]

and thence,

\[
Q_l = \begin{bmatrix}
0.34653 \\
0.61247 \\
0.52601 \\
0.51499
\end{bmatrix}
\]

The procedure is continued until the condition $\|f_o\| < \xi$ is met. These results together with those from subsequent application of the procedure are tabulated in Table 5.6c.

**A.5 Broyden-Fletcher-Goldfarb-Shanno Method**

The following example illustrates the use of the BFGS procedure for solving the parallel network (see Figure 5.3 and Table 5.4).

The steps to evaluate $f_o$ are identical to those described in NR procedure. From the previous illustration, the $f_o$ is found to be

\[
f_o = \begin{bmatrix}
0.00659 \\
0.00165 \\
0.00062
\end{bmatrix}
\]
Initially, the \( H_0 = I \); substituting \( f_0 \) and \( H_0 \) into (4.8) yields

\[
q_0 = t_0 \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0.00659 \\ 0.00165 \\ 0.00062 \end{bmatrix}
\]

As with NR method, by putting \( t_0 = 1 \), the value of \( q_0 \) is computed to give

\[
q_0 = \begin{bmatrix} 0.00659 \\ 0.00165 \\ 0.00062 \end{bmatrix}
\]

and hence a new set of flow distributions

\[
Q_1 = \begin{bmatrix} 0.49341 \\ 0.50494 \\ 0.50103 \\ 0.50062 \end{bmatrix}
\]

The procedure is continued for \( n = 1 \)

\[q_1 = t_1 H_1 f_1\]

in which

\[
f_1 = \begin{bmatrix} 0.00627 \\ 0.00171 \\ 0.00062 \end{bmatrix}
\]

and

\[H_1 = H_0 + A_0 + B_0\]
The $A_o$ and $B_o$ are given by the BFGS updating formula (4.52)

$$A_o = -\frac{\delta_o \gamma_o^T H_0 + H_0 \gamma_o \delta_o^T}{\delta_o^T \gamma_o}$$

$$B_o = -\left(t_o - \frac{\gamma_o^T H_0 \gamma_o}{\delta_o^T \gamma_o}\right) \frac{\delta_o \delta_o^T}{\delta_o^T \gamma_o}$$

respectively; in which $\delta_o = q_o$; and the $\gamma_o$ is evaluated as follows

$$\gamma_o = f_i - f_o =\begin{bmatrix} 0.00627 \\ 0.00171 \\ 0.00062 \end{bmatrix} - \begin{bmatrix} 0.00659 \\ 0.00165 \\ 0.00062 \end{bmatrix} = \begin{bmatrix} -0.00032 \\ -0.00006 \\ 0.00000 \end{bmatrix}$$

With the $t_o$, $\delta_o$ and $\gamma_o$ are known, the $A_o$ and $B_o$ are evaluated to give

$$A_o = \begin{bmatrix} -2.09325 & -0.08041 & -0.08514 \\ -0.08041 & 0.09087 & 0.02015 \\ 0.08514 & 0.02015 & 0.00238 \end{bmatrix}$$

$$B_o = \begin{bmatrix} 22.78954 & 5.70384 & 2.13063 \\ 5.70384 & 1.42758 & 0.53326 \\ 2.13063 & 0.53326 & 0.19920 \end{bmatrix}$$

Thus,

$$H_i = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} + \begin{bmatrix} -2.09325 & -0.08041 & -0.08514 \\ -0.08041 & 0.09087 & 0.02015 \\ 0.08514 & 0.02015 & 0.00238 \end{bmatrix} = \begin{bmatrix} 21.69629 & 5.62343 & 2.04550 \\ 5.62343 & 2.51845 & 0.55342 \\ 2.04550 & 0.55342 & 1.20157 \end{bmatrix}$$
The $q_i$ becomes

$$
\begin{pmatrix}
21.69629 & 5.62343 & 2.04550 & 0.00627 \\
5.62343 & 2.51845 & 0.55342 & 0.00171 \\
2.04550 & 0.55342 & 1.20157 & 0.00062
\end{pmatrix}
$$

Putting $t_i = 1$,

$$
q_i = \begin{pmatrix}
0.14698 \\
0.03992 \\
0.01452
\end{pmatrix}
$$

and thence,

$$
Q_i = \begin{pmatrix}
0.34643 \\
0.61200 \\
0.52643 \\
0.51514
\end{pmatrix}
$$

The procedure is continued until the condition $\|f_i\| < \xi$ is met. These results together with those from subsequent application of the procedure are tabulated in Table 5.6d.