Transient Response Analysis for Fault Detection and Pipeline Wall Condition Assessment in Field Water Transmission and Distribution Pipelines and Networks

by

Mark Leslie Stephens

February 2008

A Thesis Submitted for the Degree of Doctor of Philosophy

School of Civil and Environmental Engineering
The University of Adelaide, SA 5005
South Australia

Appendix N

Wave speed estimation and other complexities (including restraints and entrained air) for transmission pipelines

N.1 Entrained air in transmission pipelines

Fox (1977) determined that the level of dissolved air in water is typically 2% at normal temperatures. This dissolved air can come out of solution during either a pressure drop or temperature increase. The other major mechanism whereby air can be entrained into a pipeline is where insufficient submergence at a pump allows the formation of vortices. In the absence of air valves on all summits of an undulating pipeline the presence of air pockets, even if migratory, is inevitable. That said, the percentage of entrained air that will accumulate in such an undulating pipeline can be reduced by maintaining minimum grades of 1:250 and 1:500 on downward and upward slopes, relative to the direction of flow, respectively (and periodic flushing at air valves located at high points).

N.2 Specific assessment of entrained air

The Hanson Transmission Pipeline (HTP) is a gravity main leading from five 9.1ML summit tanks. At the time of the testing, in May 2004, the temperature in the region varied diurnally between 10 to 20 degrees Celsius. A GPS unit was used to accurately survey the position of all air valves (fire plugs) along the HTP. This information was compared to that from available “as constructed” survey information recorded shortly after the construction of the HTP. Table N-1 compares the spacing information from the two sources and confirms that there are 23 air valves along the HTP at an average spacing of approximately 550m.

The MTP is a pumped rising main leading from the Morgan filtration/treatment plant to two 9.1ML staging tanks. It was presumed that there would be more entrained air in the MTP relative to the HTP. The tests on the Morgan Transmission Pipeline (MTP) were conducted on days with similar temperatures to those experienced for the

HTP. As for the HTP, GPS survey was undertaken to confirm the position of all the air valves (fire plugs) along the MTP. Table N-2 summarises the GPS information and confirms that there are 62 air valves along the MTP at an average spacing of approximately 400m.

Table N-1 – Air valve spacing determined from GPS and map information for HTP

AVFP nos.	GPS	Map	AVFP nos.	GPS	Map
1-2	61.0	380.7	13-14	213.4	259.0
2-3	1304.5	1313.7	14-15	365.8	360.0
3-4	914.4	907.2	15-16	969.3	961.7
4-5	585.2	546.4	16-17	701.0	666.6
5-6	243.8	261.8	17-18	121.9	111.2
6-7	975.4	984.0	18-19	146.3	188.3
7-8	329.2	334.8	19-20	518.2	486.6
8-9	731.5	744.7	20-21	701.0	703.5
9-10	487.7	456.1	21-22	512.1	509.9
10-11	121.9	108.8	22-23	963.2	888.3
11-12	390.1	403.5	23-BV	755.9	778.4
12-13	396.2	380.3	**Avg.**	**543.9**	**553.7**

Table N-2 – Air valve spacing determined from GPS information for MTP

AVFP no.	GPS	AVFP no.	GPS	AVFP no.	GPS	AVFP no.	GPS
1-2	100.0	17-18	329.0	33-34	436.5	49-50	400.9
2-3	42.2	18-19	574.7	34-35	447.5	50-51	705.1
3-4	89.1	19-20	409.4	35-36	511.7	51-52	473.9
4-5	613.7	20-21	383.7	36-37	623.4	52-53	528.2
5-6	356.1	21-22	305.0	37-38	96.9	53-54	316.5
6-7	107.2	22-23	8.2	38-39	201.2	54-55	632.7
7-8	563.1	23-24	410.4	39-40	303.1	55-56	634.1
8-9	652.7	24-25	493.9	40-41	339.5	56-57	369.7
9-10	164.6	25-26	381.7	41-42	344.8	57-58	636.6
10-11	243.7	26-27	635.8	42-43	244.2	58-59	318.9
11-12	657.1	27-28	702.2	43-44	918.9	59-60	93.6
12-13	470.4	28-29	519.3	44-45	570.5	60-61	477.8
13-14	340.9	29-30	572.3	45-46	79.2	61-62	319.8
14-15	127.5	30-31	132.9	46-47	177.9	62-tanks	400.9
15-16	877.3	31-32	236.7	47-48	441.3	**Avg.**	**401.3**
16-17	288.9	32-33	579.7	48-49	468.4		

Appendix N – Wave speed estimation and other complexities for transmission pipes

Apart from pipeline usage and the frequency of air valves, the most relevant topological characteristic, affecting the migration of entrained air, is pipeline slope. Equations determined by Kalinske and Bliss (1943) (refer to Appendix O), together with the information regarding pipe gradient along sections of the transmission pipelines, have been used to determine the critical velocity required to sweep entrained air bubbles along the pipelines to nearby high points. The lengths of separate sections of pipeline, with approximately uniform slopes, together with the minimum Kalinske and Bliss (1943) critical velocity for migration of entrained air, are listed in Tables N-3 for the HTP.

Table N-3 – HTP length, slope and critical bubble movement velocities

HTP section	Length (m)	ΔEL (m)	Slope (deg.)	Kalinske & Bliss (1943)	HTP section	Length (m)	ΔEL (m)	Slope (deg.)	Kalinske & Bliss (1943)
1	2363.2	9.1	0.22	0.23	16	168.8	-2.3	-0.78	0.44
2	1941.2	17.3	0.51	0.35	17	253.2	13.3	3.02	0.86
3	337.6	0.2	0.04	0.09	18	253.2	0.5	0.10	0.16
4	337.6	-6.2	-1.06	0.51	19	84.4	-17.8	-11.91	1.72
5	844.0	9.0	0.61	0.39	20	168.8	6.1	2.07	0.71
6	337.6	-6.6	-1.12	0.52	21	168.8	-6.3	-2.12	0.72
7	422.0	2.2	0.30	0.27	22	168.8	9.3	3.15	0.88
8	84.4	-1.6	-1.09	0.51	23	506.4	11.3	1.28	0.56
9	253.2	4.0	0.91	0.47	24	1350.4	5.7	0.24	0.24
10	168.8	-1.2	-0.40	0.31	25	422.0	5.4	0.73	0.42
11	422.0	2.4	0.32	0.28	26	253.2	-3.3	-0.75	0.43
12	168.8	-4.8	-1.63	0.63	27	422.0	3.3	0.45	0.33
13	422.0	11.6	1.57	0.62	28	337.6	-4.0	-0.68	0.41
14	253.2	17.1	3.87	0.97	29	422.0	3.5	0.48	0.34
15	168.8	1.6	0.54	0.36					

Table N-4 shows the variation in the actual flow velocity in the HTP during the 24 hours prior to the transient tests. The average flow velocity during this period was 0.51m/s. This average velocity typically exceeds the critical velocity required to sweep entrained air bubbles along the HTP to one of the local high points. The conclusion that there was little entrained air in the HTP, was reinforced on this basis.

Table N-4 – Average flow in HTP over 24 hours prior to the transient tests

Time	Flow (m^3/s)	Velocity (m/s)	Time	Flow (m^3/s)	Velocity (m/s)
12noon	0.17	0.54	12midnight	0.18	0.59
1pm	0.14	0.47	1am	0.17	0.56
2pm	0.14	0.47	2am	0.17	0.56
3pm	0.13	0.41	3am	0.16	0.53
4pm	0.11	0.37	4am	0.16	0.53
5pm	0.11	0.37	5am	0.16	0.51
6pm	0.12	0.38	6am	0.16	0.53
7pm	0.13	0.41	7am	0.17	0.56
8pm	0.14	0.47	8am	0.18	0.59
9pm	0.18	0.59	9am	0.20	0.64
10pm	0.16	0.53	10am	0.23	0.73
11pm	0.16	0.53	11am (20^{th} May 2004)	0.18	0.59

The lengths of separate sections of pipeline, with approximately uniform slopes, together with the critical velocity for migration of entrained air, are listed in Tables N-5 for the MTP.

Table N-5 – MTP length, slope and critical bubble movement velocities

HTP section	Length (m)	ΔEL (m)	Slope (deg.)	Kalinske & Bliss (1943)	HTP section	Length (m)	ΔEL (m)	Slope (deg.)	Kalinske & Bliss (1943)
1	2382.1	4.0	0.10	0.16	15	334.2	2.0	0.34	0.31
2	1065.5	-5.0	-0.27	0.28	16	364.2	0.2	0.03	0.09
3	2105.0	16.0	0.44	0.35	17	159.5	2.0	0.72	0.45
4	329.0	2.0	0.35	0.31	18	308.9	-1.0	-0.19	0.23
5	1367.8	-3.0	-0.13	0.19	19	2044.9	8.0	0.22	0.25
6	1599.0	7.0	0.25	0.27	20	379.6	4.0	0.60	0.41
7	1857.2	6.0	0.19	0.23	21	632.7	-2.0	-0.18	0.23
8	413.7	-4.0	-0.55	0.40	22	1640.4	8.0	0.28	0.28
9	2503.5	18.0	0.41	0.34	23	318.8	-2.0	-0.36	0.32
10	421.5	-2.0	-0.27	0.28	24	571.5	6.0	0.60	0.41
11	499.9	9.0	1.03	0.54	25	319.8	0.2	0.04	0.10
12	303.1	-2.0	-0.38	0.33	26	230.4	2.0	0.50	0.37
13	1847.3	10.0	0.31	0.30	27	1387.8	1.0	0.04	0.11
14	570.5	1.0	0.10	0.17	28	2382.1	4.0	0.10	0.16

In the case of the MTP, flow records were not available. However, it was ascertained from the South Australian Water Corporation operators that the typical pumped flow in the MTP was 550L/s and that pumping occurred for 1-2hours at least once a day. Using an average internal diameter of 724.3mm, the velocity in the MTP can be calculated as 1.21m/s. This velocity exceeds the critical velocities required to sweep entrained air bubbles along the MTP to one of the local high points.

N.3 Restraint of aboveground transmission pipelines

Restraints may be used to attempt to reduce the longitudinal and lateral movement of a pipeline. However, if mechanical motion and vibration, related to precursor, flexural and shear waves, are not completely restrained between supports then energy will be transferred, via radiation to, transmission into, or impact on, the supporting structures. Similar forms of pipeline restraint and support are used for both transmission pipelines investigated in this research. In fact, the use of aboveground MSCL pipelines with the types of restraints and supports described below is common to nearly all transmission pipeline systems throughout South Australia. The construction of underground gullets (i.e., locations where the pipeline is diverted underground for short lengths to facilitate vehicular crossing), underground valve units, lateral pipes, changes in direction and end restraints at tanks or pump stations are also common to most South Australian transmission pipelines.

The main method of supporting the pipelines involves the placement of concrete saddle supports at a spacing of approximately 10m. These supports are widened at the base to improve bearing capacity on the soil beneath. On their own, these saddles provide vertical support and limited longitudinal and lateral support. The saddles are similar to the brackets used by Budny et al. (1991) to investigate the effect of pipeline restraint in the laboratory. In addition to the saddle supports, concrete collar restraints are used. The spacing of these collar restraints is variable along the transmission pipelines but is typically 75m. Figure N-1 shows typical saddle supports along the HTP and a collar restraint adjacent to the location of the transient generator.

Figures N-1 - Typical collar ring restraints and saddle supports on the Hanson Transmission Pipeline

N.4 Wave speed estimation

Theoretical estimation of wave speed for composite pipelines

The theoretical wave speed for a composite steel and cement walled pipeline can be estimated using the procedure described by Wylie and Streeter (1993). The thickness of cement lining is converted to an equivalent thickness of steel using the ratio between the elastic modulus for the cement and steel as follows:

$$t_{eqS} = t_C \times \frac{E_C}{E_S} \tag{N-1}$$

The approximate theoretical wave speed for the composite pipeline can then be determined (assuming there is no water/air mixture) using the usual equation:

$$a = \sqrt{\frac{K/\rho}{1+(K/E_S)(D/e)c}} \tag{N-2}$$

where K is the bulk modulus of water, ρ is the density of water, E_S is the elastic modulus of steel, D is the internal diameter of the pipeline, e is the thickness of the equivalent steel wall and c is a pipe restraint factor which, for an axially restrained pipe, is calculated using:

$$c = 1 - \nu^2 \qquad \text{(N-3)}$$

where ν is Poisson's ratio for steel

Theoretical estimation of wave speed for Hanson Transmission Pipeline

The Hanson Transmission Pipeline (HTP) comprises 650mm nominal diameter Mild Steel Cement Mortar Lined (MSCL) pipe. It is located aboveground, supported by saddles (at approximately a 10m spacing) and restrained by collars at an approximate spacing of 75m. Pipe joints are fully welded. Plans of the HTP indicate that it has a wall thickness of 3/16 of an inch (4.76mm) along its entire length. Sections of pipe (previously cut out to be replaced) reveal that the thickness of the cement lining varies between 10 to 15mm (with an average thickness of approximately 12.5mm). Figure N-2 shows the abovementioned and other details of the material types, thickness and properties for the HTP.

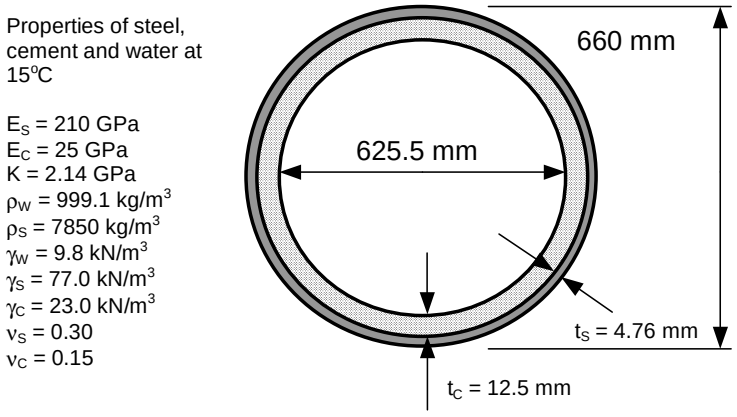


Figure N-2 - Cross section and details for the Hanson Transmission Pipeline

It is arguable whether the HTP is axially restrained. The below ground chambers for in-line valves and cross connections, the lateral offtake, the insertion flowmeter and multiple vehicular crossings provide axial restraint. Furthermore, the collars and, to a lesser extent, each saddle support provide additional axial restraint. Assuming the HTP is effectively restrained, and using the properties specified in Figure N-2, the approximate theoretical wave speed for the HTP is 1055m/s.

Theoretical estimation of wave speed for Morgan Transmission Pipeline

The Morgan Transmission Pipeline (MTP) comprises 750mm nominal diameter Mild Steel Cement Mortar Lined (MSCL) pipe. As for the HTP, it is located aboveground, supported by saddles and restrained by collars. The pipe joints are again fully welded. Plans of the MTP indicate that its wall thickness varies for known sections from 5/16, 1/4 to 3/16 of an inch (7.94, 6.35 to 4.76mm). Sections of pipe cut to enable CCTV camera access revealed that the thickness of the cement lining was approximately 12.5mm at multiple locations. Figure N-3 shows the abovementioned and other details of the material types, thickness and properties for the MTP.

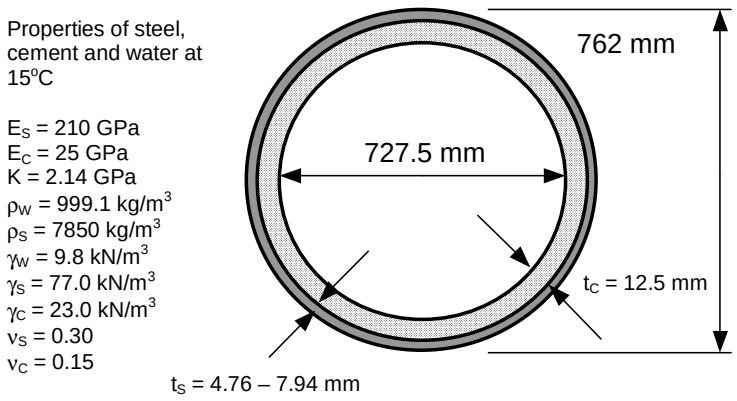


Figure N-3 - Cross section and details for the Morgan Transmission Pipeline

The determination of a theoretical wave speed for the MTP is significantly more complex than for the HTP because of the changes in wall thickness for different sections that vary between 3/16 and 5/16 of an inch (4.76 and 7.94mm). A section of

3/8 of an inch pipe between in-line gate valve “No.1” and the Morgan filtration/treatment plant is excluded by the use of this valve as a boundary condition. The changes in wall thickness along the MTP are summarised in Table N-6:

Table N-6 – Variation of wall thickness for Morgan Transmission Pipeline

Start chainage (m)	End chainage (m)	Internal diameter (mm)	Thickness of steel (mm)
0	108	717.9	9.53
108	5614	721.1	7.94
5614	5833	724.3	6.35
5833	5842	721.1	7.94
5842	9832	724.3	6.35
9832	9841	721.1	7.94
9841	11740	724.3	6.35
11740	15731	727.5	4.76
15731	15839	724.3	6.35
15839	26100	727.5	4.76

As for the HTP, it is arguable whether the MTP is axially restrained. The below ground chambers for in-line valves and cross connections, the lateral offtake, the insertion flowmeter and multiple vehicular crossings again provide axial restraint. Furthermore, the collars and, to a lesser extent, each saddle support provide additional axial restraint. Assuming composite action between the cement lining and the steel wall and that the MTP is effectively restrained, and using the properties specified in Figure N-3, the approximate theoretical wave speed for the MTP varies with wall thickness as shown in Table N-7:

Table N-7 – Theoretical variation of wave speed for Morgan Transmission Pipeline with intact cement lining and full restraint

Internal diameter (mm)	Thickness of steel (mm)	Thickness of cement lining (mm)	Wave speed (m/s)
721.1	7.94	12.5	1120
724.3	6.35	12.5	1074
727.5	4.76	12.5	1015

Direct wavefront timing estimation of wave speed

Figure N-4 shows the dispersion of the incident transient wavefronts at station 1, for tests 1 and 2, conducted on the 21st May 2004, on the Hanson Transmission Pipeline (HTP). Significant dispersion is evident after the wavefronts, initially approximately 10ms steep (estimate based on potentiometer measurements of the rotation of the steel axis and torsion spring mounted in the transient generator), have travelled to station 1. Determination of the arrival time of the wavefronts becomes ambiguous with increasing dispersion. Figure N-4 shows four points that could be used to assess the travel time of the incident wavefronts. However, the dispersion of the wavefronts means that the estimated wave speed will decrease as points 1 (least dispersion) through to 4 (greatest dispersion) are used.

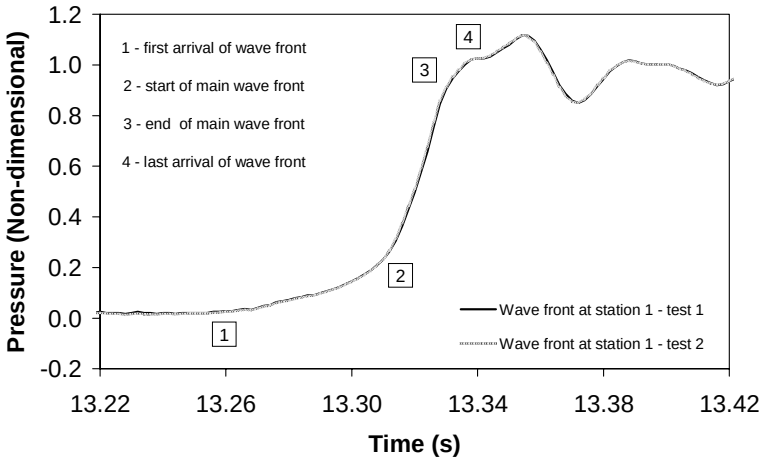


Figure N-4 – Dispersion of wavefront along the Hanson Transmission Pipeline

Table N-8 shows the wave speeds estimated using all points. The results for points 1 and 2 are considered the most relevant, because significant dispersion is not erroneously incorporated, as for points 3 and 4, and they define a range encompassing the estimated theoretical wave speed. The average of the values for points 1 and 2 is 1050m/s and relatively close to the theoretically predicted value. Given the variability in the wall thicknesses along the MTP, and more dispersion than observed for the HTP, a direct analysis of the wave speeds along the MTP is not presented.

Table N-8 – Wave speeds from wavefront along the Hanson Transmission Pipeline

Test No.	Point 1 (m/s)	Point 2 (m/s)	Point 3 (m/s)	Point 4 (m/s)
1	1084	1019	1001	981
2	1084	1019	1001	981
3	1077	1019	1000	985
4	1077	1019	1000	985
Average	**1080.5**	**1019.0**	**1000.5**	**983.0**

Periodic timing estimation of wave speed

Analysis of the period of a transient response has been used as a method for inferring the wave speed of a pipeline. Covas et al. (2004a) used this method to determine the wave speed for their field tests described in Chapter 3. However, the use of this method is only appropriate for the case where non-frequency dependent effects are insignificant. Unfortunately, unsteady friction, entrained air and, possibly to a greater extent, mechanical damping act to disperse transient responses from field pipelines such that inferring the wave speed from the period of the measured response will lead to underestimation of the "true" wave speed.

In the case of the Hanson Transmission Pipeline (HTP), the theoretical period of a transient response is equal to $4L/a$ seconds where L and a are the length and wave speed, respectively. If the half-period of the measured response is known, and is not dispersed, then an apparent wave speed can be calculated using $2L/T$ where T is the measured period of the HTP. The average half-period of the transient responses for the tests, at both stations 1 and 2, is 26s. Given the HTP is 13525m long, the apparent wave speed can be calculated as 1040m/s. This wave speed is slightly less than the theoretical value, and that determined directly using the leading edge of the measured wavefronts, because the dispersion caused by unsteady friction, entrained air and mechanical damping act to extend the period of the measured response. The periods of the responses for the Morgan Transmission Pipeline (MTP) have not been used to infer wave speeds.

N.5 Direct CCTV camera inspection information

CCTV camera footage for the Hanson Transmission Pipeline

Closed circuit television (CCTV) camera investigations conducted over a short length of the Hanson Transmission Pipeline (HTP) near Gum Creek indicates that the roughness of the cement mortar lining for that length is, on average, approximately 2mm. Table N-9 presents a summary of the logs from the CCTV camera investigation. The observed damaged can be classified in terms of the presence of cement pieces (P), corrosion (R), tuberculation (T) and delamination (D).

Table N-9 – Summary of log of CCTV camera investigation for the HTP

Chainage (m)	Damage classification	Exposed steel (m^2)	Roughness (mm)	Chainage	Damage classification	Exposed steel (m^2)	Roughness (mm)
8800	Nil	0	1	8960	Nil	0	1
8810	Nil	0	1	8970	Nil	0	2
8820	**P**	0	1	8980	Nil	0	1
8830	Nil	0	1	8990	Nil	0	1
8840	**P+D**	0.25	6	9000	**R+D**	0	4
8850	Nil	0	1	9010	Nil	0	1
8860	Nil	0	1	9020	Nil	0	1
8870	**P**	0	4	9030	Nil	0	1
8880	**D**	0	6	9040	Nil	0	1
8890	**R+T**	0	6	9050	Nil	0	1
8900	Nil	0	1	9060	Nil	0	1
8910	Nil	0	1	9070	**R+D**	0.25	6
8920	Nil	0	1	9080	Nil	0	1
8930	Nil	0	2	9090	**R+D**	0	4
8940	Nil	0	2	9100	Nil	0	1
8950	Nil	0	1	**Average**			**2.0mm**

This is the best available direct information regarding the wall condition of the HTP apart from general theoretical information relating to the likely roughness of the cement mortar lining of a MSCL pipe. A roughness of 2mm has been adopted in the initial analysis presented in Chapter 7. A methodology for performing roughness calibration using Inverse Transient Analysis (ITA) is developed in Chapter 8.

Appendix N – Wave speed estimation and other complexities for transmission pipes

CCTV camera footage for the Morgan Transmission Pipeline

Closed circuit television (CCTV) camera inspections conducted over two short lengths of the Morgan Transmission Pipeline (MTP), between chainages 15000 and 15400m, and then chainages 17200 and 17700m, indicate that the roughness of the cement mortar lining for that length is, on average, approximately 3mm. Tables N-10 and N-11 present summaries of the logs from the CCTV camera investigations. Significantly more exposed steel was observed than for the HTP with a total of 28.3m^2 and 8.5m^2 at the two locations, respectively. Corrosion, tuberculation, delamination and the build-up of cement pieces provided considerable additional evidence of internal pipe wall damage along the section of MTP from chainage 15000 to 15400m.

Table N-10 – Summary of log of CCTV camera investigation for MTP between chainages 15000 and 15400m

Chainage (m)	Damage classification	Exposed steel (m^2)	Roughness (mm)	Chainage	Damage classification	Exposed steel (m^2)	Roughness (mm)
15000	Nil	0	1	15210	Nil	0	1
15010	Nil	0	1	15220	**P+D**	0.1	8
15020	**P+D**	0.2	8	15230	Nil	0	1
15030	Nil	0	1	15240	Nil	0	1
15040	Nil	0	1	15250	Nil	0	1
15050	Nil	0	1	15260	**P+D**	0.6	8
15060	Nil	0	1	15270	**P+D+R**	4.7	8
15070	**P+D+R**	6.0	8	15280	**P+D+R**	12.2	8
15080	Nil	0	1	15290	**P+R**	0.7	8
15090	Nil	0	1	15300	Nil	0	1
15100	**P+R**	0.1	6	15310	Nil	0	1
15110	**P**	0	2	15320	Nil	0	1
15120	Nil	0	1	15330	**P+D**	0.5	8
15130	**P**	0	2	15340	**P+R**	0.2	8
15140	Nil	0	1	15350	Nil	0	1
15150	**P+R**	0.2	8	15360	**P+D**	0.1	6
15160	**P+D+R**	0.6	8	15370	Nil	0	1
15170	Nil	0	1	15380	Nil	0	1
15180	**P**	0	2	15390	**P+D**	0.6	8
15190	Nil	0	1	15400	**P+D+R**	1.5	8
15200	Nil	0	1	**Average**			**3.5mm**

This is the best available direct information regarding the wall condition of the MTP apart from general theoretical information relating to the likely roughness of the cement mortar lining of a MSCL pipe. A roughness of 3mm has been adopted in the initial analysis presented in this research.

Table N-11 – Summary of log of CCTV camera investigation for MTP between chainages 17200 and 17700m

Chainage (m)	Damage classification	Exposed steel (m^2)	Roughness (mm)	Chainage	Damage classification	Exposed steel (m^2)	Roughness (mm)
17200	Nil	0	1	17460	Nil	0	1
17210	Nil	0	1	17470	Nil	0	1
17220	Nil	0	1	17480	Nil	0	1
17230	Nil	0	1	17490	**D**	0.4	6
17240	Nil	0	1	17500	Nil	0	1
17250	Nil	0	1	17510	Nil	0	1
17260	Nil	0	1	17520	Nil	0	1
17270	**P+R**	0.5	6	17530	**D**	0.2	6
17280	Nil	0	1	17540	Nil	0	1
17290	Nil	0	1	17550	**D**	0.1	4
17300	**P**	0.1	6	17560	Nil	0	1
17310	Nil	0	1	17570	**P+D**	0.1	6
17320	Nil	0	1	17580	**P+D**	0.2	6
17330	**P+R**	1.1	8	17590	Nil	0	1
17340	**P**	0	2	17600	Nil	0	1
17350	Nil	0	1	17610	Nil	0	1
17360	**P+R**	0.1	6	17620	Nil	0	1
17370	Nil	0	1	17630	**D**	0.1	6
17380	**D+R**	2.7	8	17640	Nil	0	1
17390	Nil	0	1	17650	Nil	0	1
17400	**D+R**	1.5	6	17660	Nil	0	1
17410	Nil	0	1	17670	**D**	0.1	6
17420	Nil	0	1	17680	Nil	0	1
17430	Nil	0	1	17690	Nil	0	1
17440	**D+R**	0.6	6	17700	Nil	0	1
17450	**D+R**	0.7	6	**Average**			**2.5mm**

N.6 Wall condition by inference using wave speeds

Two very important assumptions are involved in the theoretical calculation of wave speed. Firstly, it is assumed that, in the case of composite pipelines, the cement mortar lining is acting in unison with the steel pipeline wall giving composite action. The

physical condition of the cement mortar lining (and hence internal condition of the pipeline wall) is directly related to the apparent wave speed of a pipeline. For example, if the cement mortar lining along the Hanson Transmission Pipeline (HTP) is ignored (i.e., it is assumed to be largely delaminated), then the theoretical wave speed reduces from 1055 to 983m/s. Unfortunately, it is not possible to know the variation in the condition of the cement lining a priori. That said, in the case of both transmission pipelines, the author was fortunate to have access to logs of CCTV camera investigations undertaken along sections as described above. Although only samples, the information provides a direct basis upon which to base assumptions regarding the likely average condition of the cement mortar lining along each pipeline.

Secondly, it is assumed that the pipelines are effectively restrained. The degree of restraint is directly related to the apparent wave speed of a pipeline. For example, if the assumption that the HTP is effectively restrained is wrong, and it is unrestrained, then the theoretical wave speed reduces from 1055 to 1030m/s. In this context, the problems of Fluid Structure Interaction (FSI) and mechanical dispersion and damping have been mentioned above and are considered in the thesis. While it is not as difficult to gauge the restraint of an aboveground pipeline as it is to know the condition of the internal cement mortar lining, it is, nevertheless, a highly variable parameter.

In the case of the HTP, the uniformity of wall thickness and generally good condition of the internal cement mortar lining suggest that the theoretically estimated wave speed of 1055m/s is likely, subject to an assessment of the effect of entrained air, to be satisfactory. However, in the case of the Morgan Transmission Pipeline (MTP), variable wave speeds need to be included in any model to account for the variation in wall thickness and the known deterioration in the cement mortar lining (particularly between chainages 15000 and 26100m). The results of the external survey performed by the author have been used to include the appropriate level of pipeline restraint along the different sections of the MTP.

Table N-12 lists the variation in wave speed obtained for sections of the MTP with different degrees of composite action between the cement mortar lining and steel wall

and for different degrees of restraint. Using the available logs from internal CCTV camera investigation and the results from the external survey of the MTP, different wave speeds have been included, for the different sections of pipeline identified in Table N-6 above, in the models of the MTP used in the thesis.

Table N-12 – Different wave speeds for varying combinations of deteriorated wall condition and pipeline restraint

Internal diameter (mm)	t_S (mm)	t_C (mm)	t_{EQ} (mm)	No delam. & full restraint (m/s)	No delam. & no restraint (m/s)	Delam. & full restraint (m/s)	Delam. & no restraint (m/s)
721.1	7.94	12.5	9.43	1120	1097	1078	1055
724.3	6.35	12.5	7.84	1074	1050	1020	995
727.5	4.76	12.5	6.25	1015	990	941	915

where delam. is an abbreviation of delamination

Appendix O

Transient modelling of air pockets and entrained air

O.1 Air release and movement along pipelines

Air is usually entrained in pipe systems at boundaries that are periodically open to the atmosphere. In the case of transmission pipelines, air entrainment is most likely to occur when vortices at pump suction points occasionally form. The other possibility is at leaky fittings. In the case of water distribution systems, pumps and leaky fittings are again potential sources together with the operation of valves, fire plug flushing and the operation of private plumbing within individual residences. The occurrence of low or negative pressure within a system is the other circumstance likely to result in the release of air that subsequently becomes entrained. The theoretical equations describing the release of air and its behaviour within pipeline systems are presented below.

Equations governing the release of air

Henry's Law governs the mass of dissolved air in a volume of water when the mixture is in an equilibrium state. This law states that the concentration of dissolved air is directly proportional to the partial pressure of the air at a constant temperature:

$$C = SP_a^* \tag{O-1}$$

where C is the concentration of dissolved air and S is the Henry's solubility coefficient and the amount of evolution that is possible at a given temperature is given by:

$$\Delta C = S\left(P_{Si}^* - P_{Sf}^{'}\right) \tag{O-2}$$

where P_{Si}^* and P_{Sf}^* are the initial and final absolute saturated pressures, respectively

When low or negative pressures are experience in a pipe system, the pressure may fall below the saturation pressure and dissolved air may come out of solution. The rate of evolution is dependent upon the turbulence in the flow mixture, the presence, size and distribution of gas nuclei, the solubility coefficient and, finally, the magnitude of the pressure drop. That said, it is difficult to quantify the entrained air content of a pipe system before a low pressure transient, and therefore the further quantification of potential air release under low pressure is unlikely to be accurate. Furthermore, experimental evidence indicates that less than a further 10% of dissolved air will come out of solution, to increase any existing percentage of entrained air, under negative pressures as low as –5m. For these reasons, and because low and negative pressures were not induced in the field tests described in this research, the release of dissolved air will not be further considered.

The formation of discrete air pockets in pipe systems usually results from the migration of entrained air to points of concentration. Kalinske and Bliss (1943) determined the following equation for calculating the critical velocity for sweeping bubbles along a pipe to local high points and air valves:

$$\frac{V_C}{\sqrt{gD}} = 1.509\sqrt{\tan\theta} \qquad \text{(O-3)}$$

where $\theta \geq 5$ degrees upward or downward

Kent (1952) developed an alternative equation for calculating the critical velocity for sweeping bubbles along a pipe:

$$\frac{V_C}{\sqrt{gD}} = \sqrt{C_0 \sin\theta} \qquad \text{(O-4)}$$

where C_0 is 1.53 and θ is 15 to 70 degrees upward or downward

Equations governing pressure in air/water mixtures

Dalton's Law states that the total pressure exerted by a mixture of gases is equal to the sum of the partial pressures of the various components. Dalton's Law, for a pipe system containing a mixture of air and water vapour, is expressed as:

$$P^* = P_g^* + P_v^* \tag{O-5}$$

where P^* is the total absolute pressure, P_g^* is the absolute partial air pressure and P_v^* is the absolute water vapour pressure

Following the procedure elaborated by Wylie (1984), the absolute partial air pressure can be expressed in terms of hydraulic grade using:

$$P_g^* = \rho g\left(H - z - H_v\right) \tag{O-6}$$

where H is the piezometric head and z is elevation

Furthermore, the absolute water vapour pressure can be expressed in terms of hydraulic grade using:

$$P_v^* = \rho g\left(H_v + H_{atm}\right) \tag{O-7}$$

where H_{atm} is the atmospheric pressure (in metres of head)

Using Dalton's Law, the total absolute pressure can be determined by summing the absolute partial air pressure and absolute water vapour pressure to obtain:

$$P^* = \rho g\left(H - z + H_{atm}\right) \tag{O-8}$$

which reduces to $P^* = \rho g\left(H + H_{atm}\right)$ when elevation is ignored

O.2 Transient models for air pockets and entrained air

Pressure dependent wave speed model

The presence of discrete air pockets and/or entrained air can theoretically be modelled using the traditional waterhammer equations with a modified pressure dependent water-air mixture wave speed as described by Wylie (1984):

$$a' = \frac{a}{\left(1 + C_2 m / P_g^{*2}\right)^{1/2}} \tag{O-9}$$

where $m = M_g / V$ is the mass of air per unit volume of mixture and a is the wave speed without entrained air given by:

$$a = \left(\frac{K/\rho}{1 + (KD/\mathrm{Ee})c_1} \right)^{1/2} \tag{O-10}$$

and K is the bulk modulus of water, ρ is the density of water, D is the pipe diameter, E is the pipe elastic modulus, e is the pipe wall thickness and c_1 is a restraint factor

Returning to the definitions for Equation O-9:

$$C_2 = \frac{R_g T K}{(1 + KD/Ee)} \tag{O-11}$$

and

$$P_g^* = \rho g (H - z - H_v) \tag{O-12}$$

where R_g is the gas constant, T is temperature, ρ is the density of air, H is the hydraulic grade line, z is pipe elevation (measured from the same datum as the hydraulic grade line) and H_v is the gauge vapour pressure

Bergant et al. (2003) substituted an expression for the void fraction (i.e., fraction of air):

$$\alpha = mR_gT \big/ P_g^* \tag{O-13}$$

into the expression for C_2, eliminated the m, R_g and T terms, and then substituted for P_g^*, to derive an equivalent expression for the pressure dependent wave speed:

$$a' = \frac{a}{\sqrt{1+\frac{\alpha a^2}{g\left(H-z-H_v\right)}}} \tag{O-14}$$

The problem with the direct implementation of the pressure dependent water-air mixture wave speed in a Method of Characteristics (MOC) solution scheme is that the system of equations becomes highly non-linear.

Discrete Gas Cavity Model (DGCM)

Streeter and Wylie (1983) used the ideal gas equation, in combination with the compatibility equations from a Method of Characteristics (MOC) solution scheme, to describe the behaviour of a pipe system with a discrete air pocket. The form of the modified ideal gas equation is:

$$\left(V_p\right)^n P^* = C_a \tag{O-15}$$

where V_p is the volume of the air pocket, P^* is the total absolute pressure in the air pocket and C_a is a constant derived from ideal gas relationships for air and water vapour

Recalling the expression for total absolute pressure, $P^* = \rho g\left(H + H_{atm}\right)$, taking H equal to H_p in the air pocket, and manipulating to reduce the exponent on the volume term to unity, we obtain:

$$V_p\left[H_p + H_{atm}\right]^{1/n} = C_b \tag{O-16}$$

where $C_b = \left(\frac{C_a}{\rho g}\right)^{1/n}$

Utilising the fact that C_b is constant, and its value does not need to be determined, an equation can be developed expressing the current volume of air as a function of a reference volume and head and the current head:

$$V_p = V_0\left[\frac{H_0 + H_{atm}}{H_p + H_{atm}}\right]^{1/n} \tag{O-17}$$

where V_0 and H_0 are the reference volume and gauge pressure, H_{atm} is the atmospheric gauge pressure and H_p is the current gauge pressure at the node with air

Wylie (1984) subsequently developed the Discrete Gas Cavity Model (DGCM) for the calculation of the effects of air pockets and/or entrained air. A volume of air, either representing a discrete air pocket or entrained air (as a continuum of small discrete air pockets), is included at the relevant nodal point(s) in a Method of Characteristics (MOC) model as shown in Figure O-1 and a modified continuity equation is used to describe the behaviour of each discrete air pocket:

$$\frac{dV}{dt} = Q_{out} - Q_{in} \quad \text{or, alternatively} \quad dV = V_p - V_p^{'} = dt\left(Q_{out} - Q_{in}\right) \tag{O-18}$$

where and Q_{in} and Q_{out} are the flows upstream and downstream of the air pocket, and V_p and $V_p^{'}$ are the volumes of air at the current and previous time steps, respectively

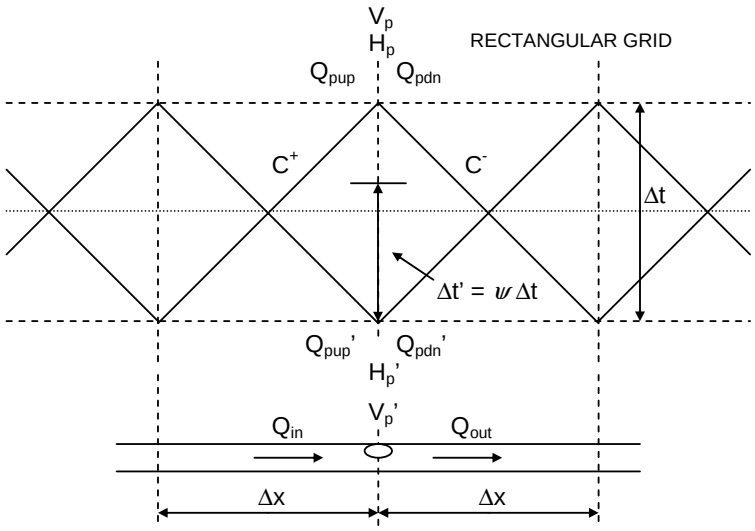


Figure O-1 – Characteristic MOC grid including a discrete air pocket

The continuity equation at the discrete air pocket must be integrated to enable Q_{in} and Q_{out}, and hence an updated volume of air, to be determined at each computational time step. This integration is performed using weighted finite differencing in the time direction and is analogous to the finite difference solution schemes that can be used to directly solve the governing unsteady flow equations (e.g., the "Priessman Scheme"):

$$\frac{V_p - V_p^{'}}{dt} = \psi\left(Q_{pdn} - Q_{pup}\right) + \left(1-\psi\right)\left(Q_{pdn}^{'} - Q_{pup}^{'}\right) \qquad \text{(O-19)}$$

where Q_{pdn}, Q_{pup}, $Q_{pdn}^{'}$ and $Q_{pup}^{'}$ are defined in Figure O-1 and ψ is the weighting factor used in the integration

Equation O-19 can be rearranged to give:

$$V_p = V_p^{'} + \left[\psi\left(Q_{pdn} - Q_{pup}\right) + \left(1-\psi\right)\left(Q_{pdn}^{'} - Q_{pup}^{'}\right)\right]\Delta t \qquad \text{(O-20)}$$

The value of ψ must be greater than 0.5 to maintain the stability of the scheme and avoid numerical oscillations. However, the numerical dispersion of the calculated

response increases as the value of ψ increases from 0.5 to 1.0. This problem is more significant when modelling the long term response of a system.

The response of the discrete air pocket to a transient can now be solved explicitly using the integrated expression representing the continuity of flow in the pipe upstream and downstream of the discrete air pocket, the ideal gas equation and the two compatibility equations from the adjacent pipe computational units:

$$V_0\left[\frac{H_0+H_{atm}}{H_p+H_{atm}}\right]^{1/n}-V_p^{'}-\left[\psi\left(Q_{pdn}-Q_{pup}\right)+(1-\psi)\left(Q_{pdn}^{'}-Q_{pup}^{'}\right)\right]\Delta t=0 \qquad \text{(O-21)}$$

$$V_0\left[\frac{H_0+H_{atm}}{H_p+H_{atm}}\right]^{1/n}-V_p^{'}-\left[\psi\left(\frac{2H_p-\left(C_p+C_m\right)}{B}\right)+(1-\psi)\left(Q_{pdn}^{'}-Q_{pup}^{'}\right)\right]\Delta t=0 \qquad \text{(O-22)}$$

The only variable in Equation O-22 that is unknown at the current time step is H_p since:

$$C_p=H_a^{'}+Q_a^{'}\left(B-R\left|Q_a^{'}\right|\right)=H_a^{'}+Q_{pup}^{'}\left(B-R\left|Q_{pup}^{'}\right|\right) \qquad \text{(O-23)}$$

$$C_m=H_b^{'}+Q_b^{'}\left(B-R\left|Q_b^{'}\right|\right)=H_b^{'}+Q_{pdn}^{'}\left(B-R\left|Q_{pdn}^{'}\right|\right) \qquad \text{(O-24)}$$

where $H_a^{'}$, $H_b^{'}$, $Q_a^{'}$ and $Q_b^{'}$ are defined in Figure O-1, *B* is the pipe impedance and *R* is the friction coefficient

Equation O-22 may be solved using, for example, the Newton-Raphson root-finding technique. Alternatively, Equations O-22, O-23 and O-24 can be combined simultaneously to derive an expression in quadratic form in which the only unknown is H_p. Wylie (1984) adopted this approach in deriving an explicit equation for the unknown head at the current time step, for n = 1.0 and $C=V_0\left[H_0+H_{atm}\right]$, as follows:

$$C = \left[V_p^{'} + \left[\psi \left(\frac{2H_p - \left(C_p + C_m\right)}{B} \right) + \left(1-\psi\right)\left(Q_{pdn}^{'} - Q_{pup}^{'}\right) \right] \Delta t \right] \left[H_p + H_{atm} \right] \qquad \text{(O-25)}$$

which upon expansion and multiplication by $\frac{B}{\psi \Delta t}$ becomes:

$$2H_p^2 + H_p\left[B_1 - 2H_{atm}\right] + \left[H_{atm}B_1 - C_1\right] = 0 \qquad \text{(O-26)}$$

where $B_1 = -\left(C_p + C_m\right) + \frac{B}{\psi}\left[\left(1-\psi\right)\left(Q_{pdn}^{'} - Q_{pup}^{'}\right) + \frac{V_p^{'}}{\Delta t} \right]$ and $C_1 = C\frac{B}{\psi \Delta t}$

Equation O-26 is in quadratic form and can be solved for H_p:

$$H_p = \frac{-\left[B_1 + 2H_{atm}\right] + \sqrt{\left[B_1 + 2H_{atm}\right]^2 - 8\left[H_{atm}B_1 - C_1\right]}}{4} \qquad \text{(O-27)}$$

A value for the volume of air at the beginning of each computational time step must be provided to enable solution. This value is initially determined under steady state conditions and then updated and stored after each subsequent computational time step.

The effect of entrained air can be modelled using a DGCM by lumping the volume of distributed air throughout the pipe network at relevant nodal points in the characteristic grid. Instead of one or two discrete air pockets, small volumes of air are included at many locations (if not all). This approximation is satisfactory providing the volume of air at each nodal point is an order of magnitude less than the volume of water in adjacent pipe computational units. The same equations are used to model the effect of a single discrete air pocket and numerous distributed air pockets (representing entrained air).

O.3 Numerical analysis of air pockets and entrained air

Effect of air pockets upon the transient response of a pipeline

Figure O-2 shows the configuration of an artificial distribution pipeline similar to the Kookaburra Court Pipeline (KCP) tested in this research. This approximate configuration has been used to numerically assess the response of typical distribution pipelines to a range of air pocket sizes. The upstream and downstream boundary conditions are formed by a reservoir with a pressure of 35m and a closed in-line valve, respectively.

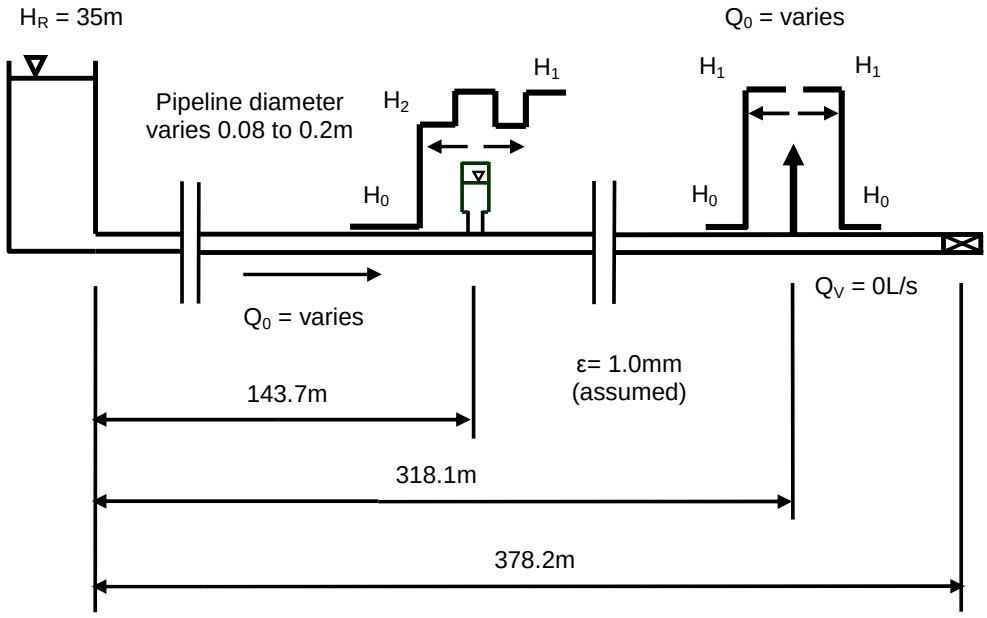


Figure O-2 – Artificial distribution pipeline for assessing air pocket detection

The transient model developed in Chapter 11 for the KCP has been adapted and applied to the pipeline configuration shown in Figure O-2 above. The sizes of the reflected and transmitted wavefronts from the air pocket are recorded at measurement points located 225.0m and 318.1m along the pipeline (these locations correspond to measurement stations 1 and 2 along the KCP, respectively). Figures O-3 and O-4 show the responses to 0.9, 1.6 and 2.3L air pockets, as measured at station 2, for 80 and 200mm diameter pipelines, respectively. The volumes of the air pockets are

specified for a reference pressure of 30m. The initial flow along the pipeline, and through the side discharge valve used to approximate the transient generator, is adjusted to maintain a 10m transient pressure rise for both diameters.

Figure O-3 shows that all three air pocket sizes have a severe effect upon the transient response of the 80mm pipeline. The second reflection corresponds to the interaction of the wavefront reflected from the dead end of the pipeline with the air pocket. Figure O-4 shows that the effect of each air pocket is less significant for the 200mm diameter pipeline. The initial negative reflection from the air pockets, while smaller than in the 80mm diameter pipeline, vary more significantly with air pocket size. Furthermore, the pressure recovers relatively quickly in the 200mm diameter pipe such that the secondary reflection from the dead end of the pipeline interacts with the air pocket under a pressure approximately equal to that of the initial plateau.

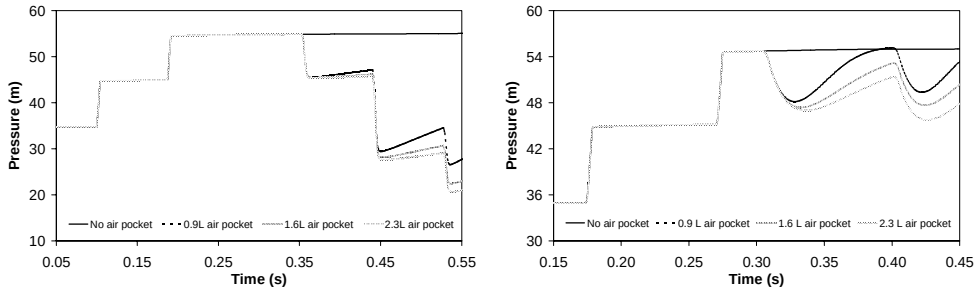


Figure O-3 and O-4 – Response to 0.9, 1.6 and 2.3L air pockets, as measured at station 2, for 80 and 200mm diameter pipelines, respectively

Effect of entrained air upon the transient response of a pipeline

Figure O-5 shows the response to 0, 0.001, 0.005 and 0.025% of entrained air, as measured at station 2, for a 200mm diameter pipeline. The volume of air in each discrete air pocket used to represent entrained air is specified for a reference pressure of 30m. The magnitude of the incident wavefront decreases as the percentage of entrained air increases. Furthermore, the arrival of the incident wavefront is progressively lagged as the percentage of entrained air increases. The effect of higher

percentages of entrained air is relatively distinct. However, very small quantities have less impact and are more difficult to either identify or discount.

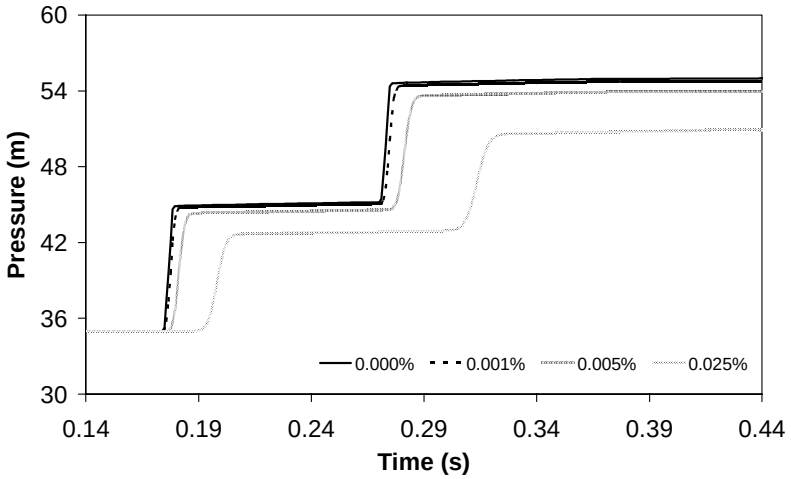


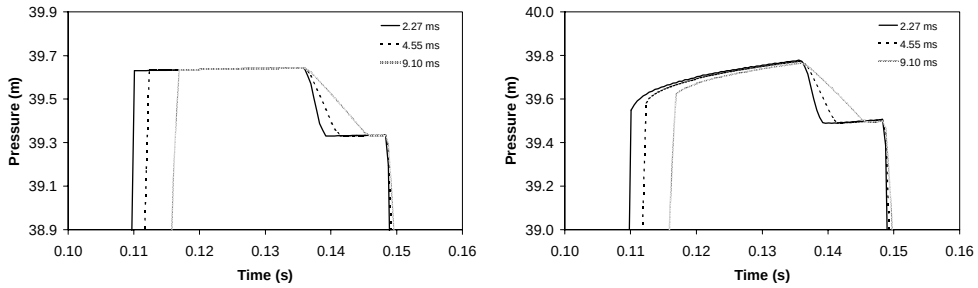
Figure O-5 – Effect of 0, 0.001, 0.005 and 0.025% of entrained air, as measured at station 2, in a 200mm diameter pipeline

Appendix P

Effect of wavefront sharpness on location of faults

The numerical effect of the speed of the valve operation on the sharpness of a transient wavefront and, for example, a leak reflection can be demonstrated, using both quasi-steady and unsteady friction models, for the laboratory apparatus used by Vitkovsky (2001). The apparatus comprised a 37.2m long copper pipe with an internal diameter of 22.1mm, wall thickness of 1.6mm, roughness height of 0.0015mm, typical pressure of 30m and a wave speed of 1319m/s. An in-line valve was located at the downstream end with a fastest physical closure speed of 9ms. Leaks of known sizes were installed at specific locations.

A controlled transient is generated using the in-line valve, with variable closure times (less than or equal to approximately 9ms), and a measurement point located 1/4 of the pipe length from the downstream valve. A 1.0mm diameter leak is located 3/4 of the pipe length from the downstream valve. Figures P-1 and P-2 show that the dispersion of the incident wavefront and leak reflection increases as the in-line valve closure speed decreases, when using quasi-steady and unsteady friction models, respectively. This makes the identification of the leak location, using only reflection information, less accurate as the speed of the valve closure decreases. The valve closure speeds, and corresponding broadening of the leak reflections, are listed in Table P-1.



Figures P-1 and P-2 – Transient wavefronts and leak reflections after in-line valve closure for quasi-steady and unsteady friction models, respectively

Appendix P – Effect of wavefront sharpness on location of faults

Table P-1 – Effect of reducing the sharpness of an incident transient wavefront on leak reflection

Valve closure speed	Leak reflection broadening (ms)	Leak reflection broadening (m)
2.3ms	3.2	4.2
4.6ms	5.5	7.2
9.1ms	10.0	13.2

Appendix Q

Skalak (1956) derivations and equations

Q.1 Fluid Structure Interaction (FSI) and mechanical damping

Precursor waves

High frequency transient events are typically accompanied by precursor waves propagating longitudinally in a pipeline wall. Circumferential strain related to the transient event is transformed into axial strain via Poisson coupling. The change in axial strain then propagates at the velocity of sound in the pipeline wall. The velocity of sound is generally higher in a pipeline wall than in the contained fluid and so the axial strain wave travels faster than the main transient wave in the fluid and is called a precursor wave. In the case of a sharp transient step event, the precursor waves theoretically manifest as oscillations about the top of a small step change in pressure. The oscillation or overshoot is related to the radial inertia of the system. Precursor waves are typically an order of magnitude smaller than the main transient waves and are normally neglected in most transient analysis.

Mechanical damping

Bracing and/or brackets, continuous restraints (e.g. concrete encasement) and/or end restraints (e.g. thrust blocks) may be used to reduce the formation of precursor waves (and oscillation of the main waterhammer wavefront as predicted by Skalak (1956) – see below). Restraints will also reduce longitudinal and/or lateral motion of a pipeline (and the formation of accompanying flexural and shear waves). In the absence of completely effective restraint transient energy will be dissipated via radiation to, transmission into or impact upon bracing/brackets or other supports as motion and vibration occur. Furthermore, the radial inertia of a pipeline may reduce the sharpness of a wavefront and disrupt its form (resulting in dispersion). This is particularly so for higher frequency transients (i.e., sharp wavefronts). Pipelines with relatively thick walls, comprising metal with cement mortar lining, which are supported above

ground, are susceptible to fluid structure interaction and mechanical damping because of their relatively high radial inertia and a lack of continuous and rigid restraint.

Q.2 Summary of findings by Skalak (1956)

Skalak (1956) first theoretically predicted the formation of precursor waves, and associated wavefront dispersion, by extending the theory of waterhammer as it was then known. In addition to the formation of precursor waves, Skalak (1956) theoretically predicted significant oscillations in the main waterhammer wave. Skalak (1956) developed a theoretical model for thin walled pipes that included radial and inertial effects, and the effect of longitudinal stress waves, in pipe walls. However, while bending stresses in pipe walls were taken into account, the effect of flexural modes of vibration were neglected (the significance of this is further discussed below in the context of the research conducted by Williams (1977). Tijsseling et al. (2006) recently presented a review of Skalak's work and a summary of the fundamental equations is presented below.

Skalak (1956) considered the propagation of a waterhammer wave in a long tube as illustrated in Figure Q-1 below.

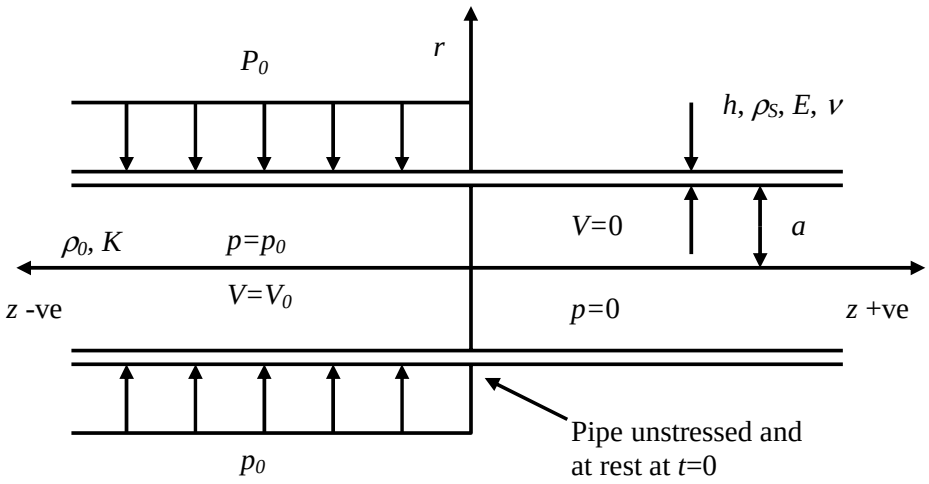


Figure Q-1 – Ideal thin walled long tube and initial conditions for wave propagation

The pressure and velocity after the propagating wave front are p_0 and V_0, r is the axis in the radial direction and a is the radius of the pipe. For the pipe wall, h, ρ_S and E are

the thickness, density, Young's modulus of elasticity and Poisson's ratio, respectively. For the fluid, ρ_0 and K are the density and bulk modulus and z is the spatial axis along the pipe relative to the wave front. The pipe is not in equilibrium at time t = 0s and the initial conditions correspond to a step pressure increase moving at the speed of sound in water. The configuration was designed to simulate a pipe leading from a reservoir, filled with fluid at rest, but below reservoir pressure, and a valve at the reservoir end being suddenly opened. This method of generating the pressure and velocity increase can be readily translated to other initial conditions.

Skalak (1956) derived equations relating the pressure and axial velocity in a fluid to the axial displacement and radial deflection of the containing pipe wall in a coupled fluid-pipe system, for different modes of vibration, using inverse Fourier and Laplace transforms. During the evaluation of these equations, Skalak realised that the root solutions were the circular frequencies of the modes of free vibration of the coupled fluid-pipe system and derived expressions for the phase velocities of the waves in the fluid and pipe wall (c_1 and c_2). The expanded form of the root solutions includes the phase velocities, $c_{1,2}$ and further parameters $d_{1,2}$. The velocities c_1 and c_2 represent the speed of propagation of the main waterhammer and precursor waves, respectively:

$$c_{1,2} = c\left[\frac{2AR + R + R^2\left(1-\nu^2\right) \mp \sqrt{\left[2AR + R + R^2\left(1-\nu^2\right)\right]^2 - 4R^2\left(1-\nu^2\right)\left(2A+R\right)}}{2\left(2A+R\right)}\right]^{0.5} \quad \text{(Q-1)}$$

and

$$d_{1,2} = ca^2\left[\frac{(A+4)\left[\left(\frac{c_{1,2}}{c}\right)^5 - \left(\frac{c_{1,2}}{c}\right)^3(1+R) + \left(\frac{c_{1,2}}{c}\right)R\right] - \left(\frac{c_{1,2}}{c}\right)\frac{h^2}{12a^2}\left(1+2R^2\nu\right)}{-16\left(\frac{c_{1,2}}{c}\right)^2(2A+R) + 8R(2A+1) + 8R^2\left(1-\nu^2\right)}\right] \quad \text{(Q-2)}$$

where

$$R = \frac{c_0^2}{c^2} \text{ and } A = \frac{\rho_0 a}{m} \tag{Q-3}$$

$$\text{and } c_0 = \sqrt{\frac{Eh}{m\left(1-\nu^2\right)}} = \text{the velocity of sound in the pipe wall} \tag{Q-4}$$

and c is the velocity of sound in water, ρ_0 is the initial density of the water, E is the modulus of elasticity of the pipe wall, h is the thickness of the pipe wall, m is the mass of the pipe per unit surface area and ν is Poisson's ratio for the pipe wall

Main waterhammer wave oscillations

In general, it is the response of the main waterhammer wave in a pipeline that is measured and so the equation describing this pressure variation, derived by Skalak (1956), is of particular interest:

$$p_n\left(\beta_n\right) = \frac{p_0}{2} - Cp_n Cw_n \int_0^{\pm\infty} \frac{\sin\left(\eta + \beta_n \eta^3\right)}{\eta} d\eta \tag{Q-5}$$

In the above equation, $\beta_n = d_n t / \left(z^*\right)^3$, $z^* = z - c_n t$ (defined as the dimensionless relative distance from the propagating wavefront) and n is equal to 1 (for the main waterhammer wave). The phase velocities $c_{1,\,2}$ and coefficients $d_{1,\,2}$ are functions of the velocity of sound in the fluid and pipe wall, the density of the fluid, the radius of the pipe, the mass of the pipe per unit surface area and the elastic modulus, thickness and Poisson's ratio for the pipe wall. The coefficients Cp_n and Cw_n are defined as:

$$Cp_n = \frac{2\rho_0 c_n^2}{a\left(\frac{c_n^2}{c^2} - 1\right)} \text{ and } Cw_n = \frac{p_0}{\pi\left(\frac{c_n}{c} - 1\right) m D_n} \tag{Q-6}$$

with

$$D_n = \frac{4\rho_0 c_n^4}{mc^2 a\left(\frac{c_n^2}{c^2}-1\right)^2} + \frac{2c_0^2}{a^2}\left[\frac{\nu^2}{\left(1-c_0^2/c_n^2\right)^2}+1-\nu^2\right] \quad \text{(Q-7)}$$

where all the terms have been previously defined above

The evaluation of the integral in Equation Q-5 is fundamental to the solution of the expressions for pressure, axial velocity, axial displacement and radial deflection which each vary with the dimensionless wave height given by:

$$I(\beta_n) = \frac{1}{2} - \frac{1}{\pi}\int_0^{\pm\infty} \frac{\sin\left(\eta+\beta_n\eta^3\right)}{\eta} d\eta \quad \text{(Q-8)}$$

where η is an integration variable

Equation Q-8 can be evaluated analytically when β_n approaches 0 from negative and positive directions, and when β_n approaches $\pm\infty$, to give values of 1, 0 and 1/3, respectively. However, Equation Q-8 needs to be numerically evaluated for other values of β_n using, for example, the trapezoidal rule. The integration needs to be performed over two divided ranges from near 0^+ to a large positive bound and 0^- to a large negative bound, respectively. Furthermore, the range of $|\beta_n|$ values needs to be limited and, in this case, 1000 values between 1.0e-06 and 1.0e+03 have been selected.

Figure Q-2 shows the results of the integration for cases where:

1) $|\eta|$ varies from 0.01 to 50000 with $\Delta\eta$ = 0.05
2) $|\eta|$ varies from 0.005 to 500000 with $\Delta\eta$ = 0.001

The solution is plotted against the dimensionless distance from the wavefront (z^*) divided by $\sqrt[3]{d_n t}$ (which equates to $1/\sqrt[3]{\beta_n}$). The Joukowsky pressure rise is plotted

for comparison. Although the integration bounds are significantly more computationally demanding for case 2) there is little variation in the accuracy with which the integral is evaluated at this scale. Figure Q-3 shows that the solution is relatively numerically stable for case 1) and 2) but deteriorates as the bounds of the integration are decreased, and the step is increased, for case 1). This deterioration is most apparent as β_n approaches large positive and negative values and $1/\sqrt[3]{\beta_n}$ approaches 0 from negative and positive directions.

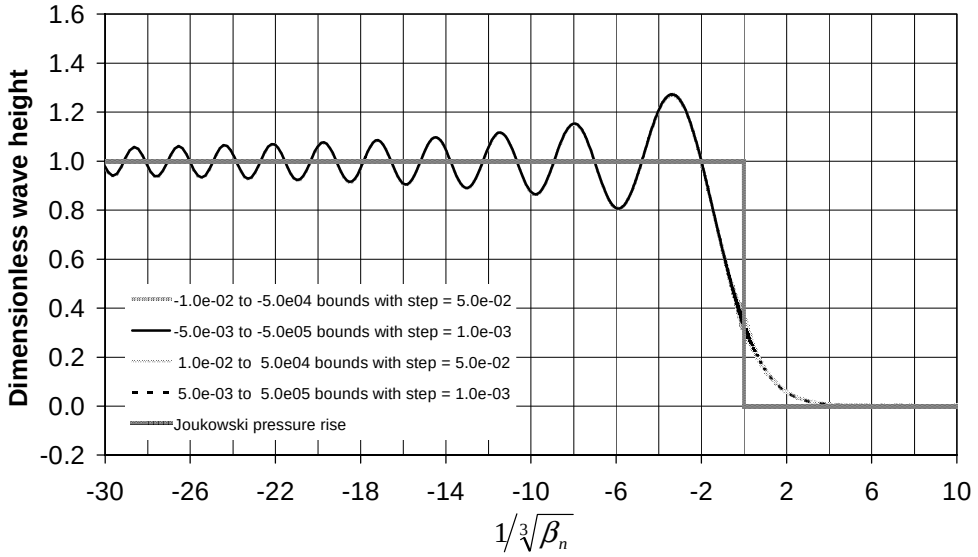


Figure Q-2 – Plot of integrated dimensionless wave height

The coefficients Cp_n and Cw_n from Equation Q-5 can be determined for the main waterhammer wave, for a particular pipeline, and used to predict the pressure response of the pipeline as a function of β_n. The solution for the integral shown in Equation Q-8 is generic and can be applied to any pipeline to determine the oscillatory form of both the main waterhammer and precursor waves. The relation $1/\sqrt[3]{\beta_n}$ is equivalent to $z^*/\sqrt[3]{d_n t}$ and this relation can, in turn, be expressed as a function of the distance from the wavefront using the relation $z = z^* + c_n t$. Hence, if the phase velocities $c_{1,\,2}$ and coefficients $d_{1,\,2}$ are known, for a particular pipe

configuration, then the predicted pressure can be expressed as a function of distance from the propagating wavefront.

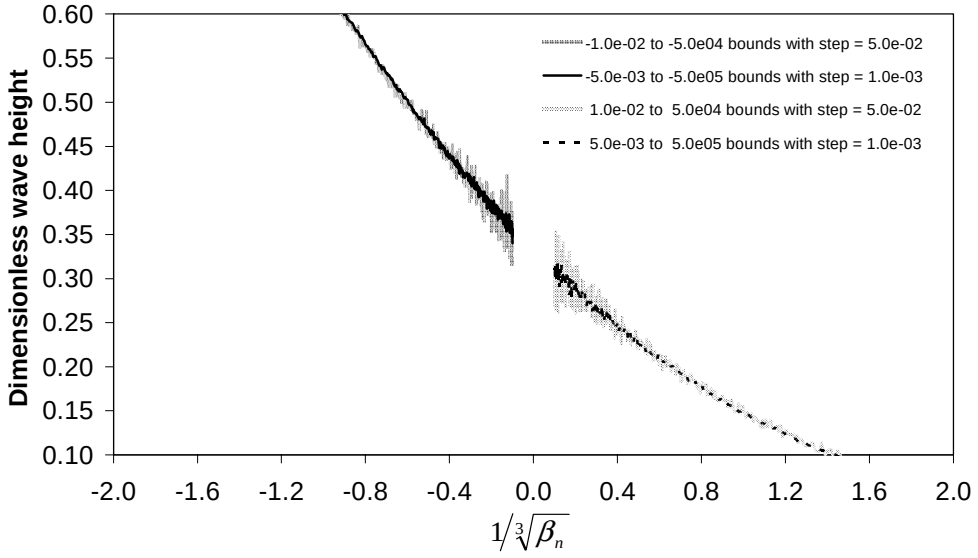


Figure Q-3 – Focus on region with β_n approaching large positive and negative values

Figure Q-2 shows decaying oscillations that increase in frequency following the passage of the wavefront. As mentioned above, the dimensionless wave height, based on the solution for the integral shown in Equation Q-8, is generic. Using the reciprocal of the first 10 periods shown in Figure Q-2 (i.e., a value of 0.36), the average expected frequency of the oscillations following the wavefront, for both main waterhammer and precursor waves, can be determined for a particular pipe configuration using:

$$f_n = 0.36\frac{c_n}{\sqrt[3]{d_n t}} \tag{Q-9}$$

where all terms have been previously defined

Appendix Q – Skalak (1956) derivations and equations

Main waterhammer wave dispersion and flexural waves

The dispersion of transient wavefronts, caused by the radial inertia of a pipeline system, was predicted by Skalak (1956) and then confirmed in the laboratory by Thorley (1969). Inertial forces associated with the radial motion of the fluid during a transient are traditionally ignored. Similarly, the effect of the mass of the pipeline, and longitudinal and bending stresses in its wall, are commonly neglected. The inclusion of radial fluid inertia, and the effect of the mass of the pipeline, may account for wavefront dispersion. The magnitude of the dispersion (L_n), for both main waterhammer and precursor waves, can be determined for a particular pipeline using the equation:

$$L_n = \frac{3\pi\sqrt[3]{d_n t}}{\Gamma\left(\frac{1}{3}\right)\sin\frac{\pi}{3}} \qquad \text{(Q-10)}$$

where the Gamma function $\Gamma(z) = \int_0^\infty t^{z-1}e^{-t}dt$ and $\Gamma\left(\frac{1}{3}\right) = 2.679$

Application of this equation to field transmission pipelines, with varying degrees of longitudinal and lateral restraint, may overly simplify the physical complexity of the system (and therefore not predict all of the observed dispersion). Williams (1977) identified the formation of flexural waves, following the interaction of main waterhammer and precursor wavefronts with changes in pipeline profiles, as a much more significant source of dispersion. Nevertheless, Equation Q-10 has been used in Chapter 7 to obtain an approximate estimate of the magnitude of the dispersion, related to the radial inertia of a pipeline, predicted in field measurements.

Appendix R

Direct reflection analysis for leak detection

R.1 Theoretical treatment of demands and/or leaks

Demands and leakage can be represented using the steady state orifice equation to describe the leak:

$$Q_L = C_d A_L \sqrt{2gH_L} \tag{R-1}$$

where Q_L is the leak flow, C_d is the coefficient of discharge for the orifice, A_L is the area of the leak orifice and H_L is the pressure at the leak

The use of Equation R-1 represents a quasi-steady approximation under unsteady conditions. Research by Funk et al. (1972) has indicated that an additional damping loss and inertial lag can occur in association with discharges to atmosphere through a side or end mounted orifice. However, the damping and timing effects associated with the acceleration and deceleration of flow through the orifice are generally insignificant relative to the average discharge that is determined using the quasi-steady approximation.

R.2 Numerical tests with an artificial pipeline

Details of the artificial pipeline

Figure R-1 shows the configuration of an artificial distribution pipeline similar to the Kookaburra Court Pipeline (KCP) tested in this research. This approximate configuration has been used to numerically assess the response of typical distribution pipelines to a 10mm diameter leak. The upstream and downstream boundary conditions are formed by a reservoir with a pressure of 35m and a closed in-line valve, respectively. The position at which the transient generator is located matches

that for the field pipeline. However, the 10mm leak has been moved closer to the reservoir boundary condition than the corresponding leak on the KCP.

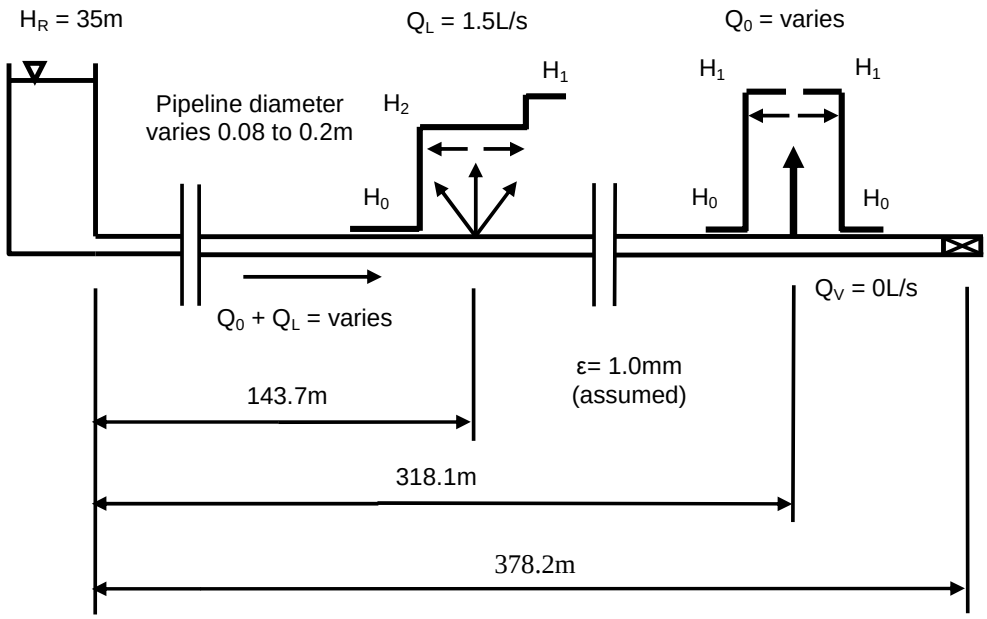


Figure R-1 – Artificial distribution pipeline for assessing leak detection

Assessing leak threshold using dimensionless leak parameter

Table R-1 summarises the changes in leak detection threshold for different diameter pipes for a 10mm leak introduced to the artificial pipeline (a C_dA_L equal to 0.0581m^2 is used as calibrated for the leak introduced to the KCP). This leak gives a discharge of approximately 1.5L/s for pipeline diameters between 80 and 200mm. The detection threshold for the leak decreases as pipeline diameter increases. The comparative leak detection threshold from previous laboratory research, presented in Chapter 3, is specified for a transient pressure rise of 10m.

It is apparent that dimensionless leak parameter for the approximate field configuration falls below the limiting value from laboratory experiments once the pipeline diameter exceeds 125mm. This indicates that, based on the results of the laboratory research, that leaks greater than 10mm or 1.5L/s are likely to be difficult to detect on pipelines greater than 125mm in diameter.

Table R-1 – Dimensionless leak parameters and detection threshold

Nom. Diam. (mm)	C_dA_L (x e03 m^2)	Area of pipe (m^2)	Wave speed (m/s)	Initial head (m)	F_{leak}	F_{leak} (laboratory)
80	0.05848	0.003848	1377.9	33.67	0.816	0.273
100	0.05765	0.007238	1349.8	34.61	0.413	0.273
150	0.05738	0.017908	1295.5	34.90	0.159	0.273
200	0.05732	0.033329	1247.2	34.95	0.082	0.273

Leak reflections predicted using direct reflection equation

Providing the pressure under steady conditions, the transmitted and reflected pressures and the transient overpressure at the location of a leak are known, and the friction in the pipe system is not significant, the size of the lumped leak coefficient C_dA_L can be determined using the relation:

$$C_d A_L = \frac{A}{a}\sqrt{\frac{g}{2}}\frac{(H_1 - H_2)}{\left(\sqrt{\frac{1}{2}(H_1 + H_2)} - \sqrt{H_0}\right)} \qquad \text{(R-2)}$$

where the terms H_0, H_1, H_2 and C_dA_L, the pressure under steady conditions, reflected pressure, transmitted pressure and the lumped leak coefficient, respectively, have been previously defined in Chapter 3

Table R-2 lists the anticipated leak reflections when the artificial pipeline is subject to a transient pressure rise of 10m propagating upstream along the pipeline (from the location of the transient generator. Equation R-2 is applied iteratively to determine the pressure after the leak has been encountered (i.e., H_T) and the size of the leak reflection (i.e., H_R-H_T). The size of the predicted leak reflection significantly decreases as pipe diameter increases (provided the pressure rise is maintained at a constant value).

Table R-2 – Decreasing size of leak reflection with increasing pipe diameter for constant leak size and 10m transient pressure rise

Nom. Diam. (mm)	C_dA_L (x e03 m^2)	H_0 (m)	H_1-H_0 (m)	H_1 (m)	H_2 (m)	Leak reflection (m)
80	0.05848	33.67	10.0	43.67	38.08	5.59
100	0.05765	34.61	10.0	44.61	41.35	3.27
150	0.05738	34.90	10.0	44.90	43.51	1.39
200	0.05732	34.95	10.0	44.95	44.21	0.74

Leak reflections predicted using a transient model

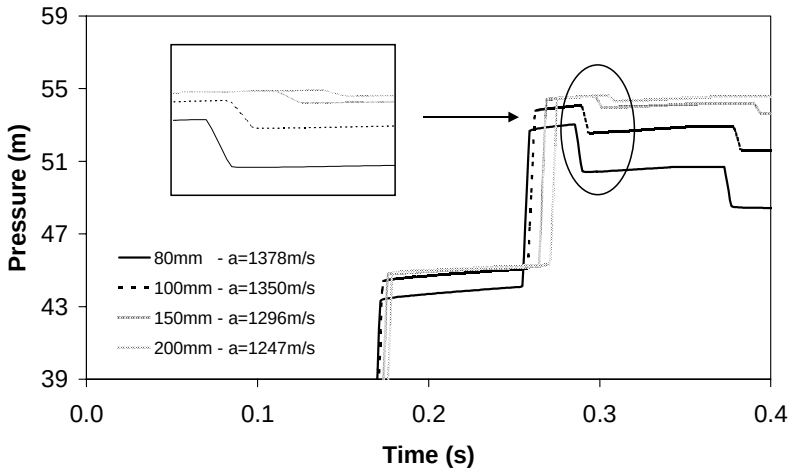
Unfortunately, Equation R-2 does not take friction into account and therefore overestimates the size of the reflection from a leak for pipelines with significant friction (or other) losses. The transient model developed in Chapter 11 for the KCP has been adapted and applied to the pipeline configuration shown in Figure R-1 above. The sizes of the reflected wavefronts from the leak are recorded at measurement points located 225.0m and 318.1m along the pipeline (these locations correspond to measurement stations 1 and 2 along the KCP, respectively).

Table R-3 lists the size of the leak reflections, predicted using the transient model, for 80, 100, 150 and 200mm diameter pipelines, with a 10mm leak in each case. The magnitude of the leak reflections predicted using Equation R-2 are presented for comparison.

Table R-3 – Leak reflections determined using model for different diameter pipes

Nom. Diam. (mm)	C_dA_L (x e03 m^2)	Q_{g+L} (L/s)	Q_L (L/s)	Q_g (L/s)	Leak refln stn 1 (m)	Leak refln stn 2 (m)	Leak refln equation x
80	0.05848	2.05	1.50	0.55	2.578	2.562	5.59
100	0.05765	2.55	1.50	1.05	1.527	1.547	3.27
150	0.05738	4.21	1.50	2.71	0.655	0.671	1.39
200	0.05732	6.74	1.50	5.24	0.325	0.341	0.74

Figure R-2 shows the leak reflections obtained using the transient model, at measurement location 1, for 80, 100, 150 and 200mm diameter pipelines, respectively. The results confirm that the size of the leak reflections significantly decrease as pipe diameter increases (provided the leak size/discharge and transient pressure rise remain constant).



Figures R-2 – Decreasing leak reflection size, at measurement station 1, as pipe diameter increases for a fixed 10mm leak

R.3 Application of direct reflection equation to field results

Interpretation of field results using direct reflection equation

The reflected pressures from the Kookaburra Court Pipeline (KCP) need to be carefully interpreted. Firstly, the incident wavefront is negative and so the interaction with the leak gives rise to an increase and not decrease in pressure. Secondly, the incident wavefront approximately doubles after reflection from the dead end of the KCP. Two reflections from the interaction of the incident and dead end reflected wavefronts with the leak are observed. However, the absolute size of the reflected wavefront from the dead end of the KCP has been reduced, by the superposition of the leak reflection from the incident transient wavefront, such that a reduced reflected wavefront interacts with the leak.

The direct reflection formulation should be carefully applied in pipelines with significant friction or other losses (e.g., mechanical damping) to avoid overestimation of the size of any leak. Figure R-3 shows distortions of the pressure plateaus in the measured response of the KCP for test 10, conducted with a 10mm leak on the 28th August 2003, at measurement station 2. These distortions relate to the interaction of the wavefronts with fire plugs, flexible joints, bends and water service connections and reduce the accuracy with which Equation R-2 can be applied.

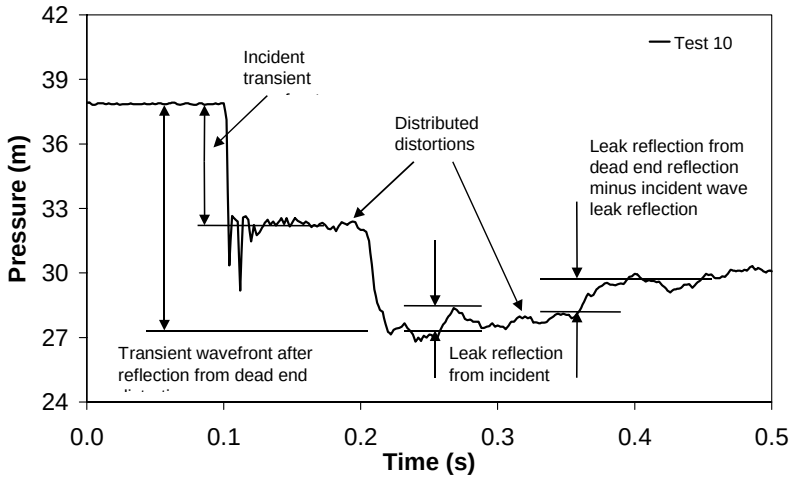


Figure R-3 – General reflection response for test 10, conducted with a 10mm leak on the 28th August 2003, at measurement station 2

Nevertheless, a lumped leak coefficient can be theoretically determined using the reflected wavefronts from the leak (neglecting friction and other losses) using Equation R-2. Table R-4 shows the parameters adopted for test 10. The measured incident and reflected wavefronts are used to determine the magnitude of the leak reflections and the value of the lumped leak coefficient and equivalent leak diameter, for a discharge coefficient of 0.74 (the laboratory calibrated leak orifice discharge coefficient). The calculated lumped leak coefficients are then compared with the known value.

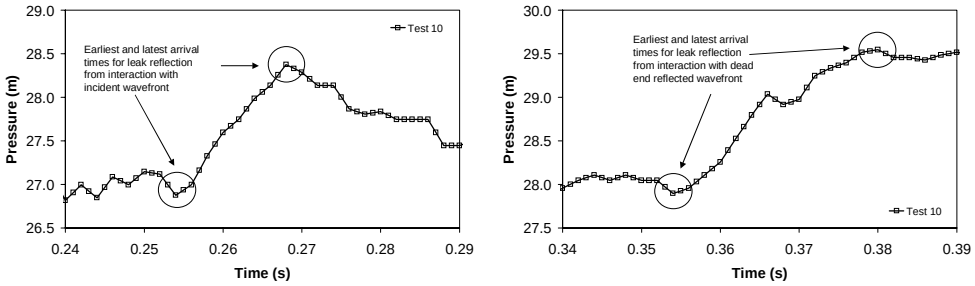
Table R-4 – Determination of lumped leak coefficients using direct reflection formulation and a measured transient response

Parameter	Test 10 1st wavefront	Test 10 2nd wavefront	Comment
H_0 (m)	37.87	37.87	HGL prior to 1st transient wavefront
$H_{R=1}$ (m)	27.16+5.01=32.17	27.16+0.69=27.85	HGL after 2nd wavefront is increased by previous passage of leak reflection from 1st wavefront
$H_{T=2}$ (m)	27.85+5.01=32.86	29.58	HGL after 1st wavefront includes +5.01m correction because only 1st wavefront contributes to size of 1st leak reflection
Δ_R (m)	0.69	1.73	1st and 2nd reflections proportional to 5.70m and 10.16m incident wavefronts
A (m/s)	1138.0	1138.0	Average wave speed as previously determined
A_p (m^2)	0.006969	0.006969	Area of pipe
$C_dA_{L\ pred}$ (m^2)	0.0000207	0.0000295	Cd = 0.74 (as previously calibrated)
$C_dA_{L\ true}$ (m^2)	0.0000581	0.0000581	Cd = 0.74 (as previously calibrated)
D_{leak} (m)	0.00597	0.00613	Predicted diameter of equivalent leak
$D_{leak\ true}$ (m)	0.01	0.01	True diameter of equivalent leak

The results presented in Table R-4 confirm that neglecting friction and/or mechanical damping, and other discrete and distributed distortions in the measured response of the KCP, leads to the lumped leak coefficient being underestimated by 40.3% and 38.7%, when calculated on the basis of the reflections from the incident and dead end reflected wavefronts, respectively.

Using reflection information to locate the leak

Figures R-4 and R-5 show that the incident and dead end reflected wavefronts have dispersed, after reflection from the leak, to approximately 15ms and 30ms, respectively. This reduces the accuracy with which the location of the leak can be determined. The distance to the leak can be estimated using the reflection from the incident wavefront and is underestimated and overestimated when using the earliest and latest reflection arrival times from the wavefront, respectively. The distance to the leak can also be estimated using the dead end reflected wavefront and is underestimated when using both the earliest and latest reflection arrival times.



Figures R-4 and R-5 – Detailed leak reflections from incident and dead end reflected wavefronts for test 10 conducted on 28th August 2003

Appendix S

Direct reflection analysis for discrete blockage detection and unsteady minor losses

S.1 Numerical tests with an artificial pipeline

Details of the artificial pipeline

Figure S-1 shows the configuration of an artificial distribution pipeline similar to the Saint Johns Terrace Pipeline (SJTP) tested in this research. This approximate configuration has been used to numerically assess the response of typical distribution pipelines for a variety of discrete blockages. The upstream and downstream boundary conditions are formed by a reservoir with a head of 25m and an in-line valve, opened to establish flow along the pipeline, respectively. The positions at which the transient generator and the in-line gate valve, used to simulate discrete blockage, are located match those for the field pipeline.

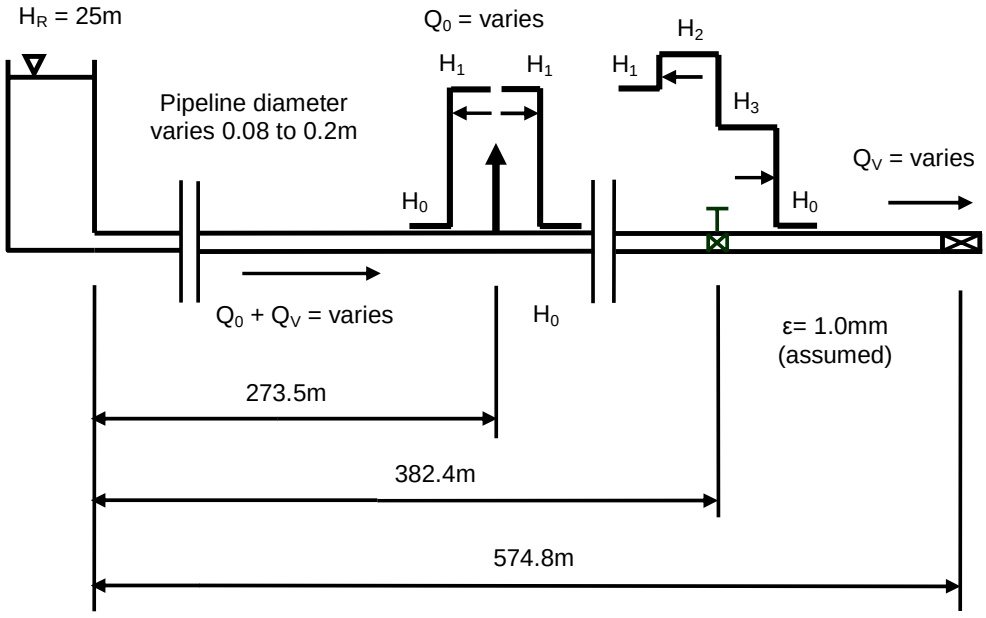


Figure S-1 – Artificial distribution pipeline for discrete blockage detection

Appendix S – Direct reflection analysis for block detection and unsteady minor losses

Establishing baseflow and hydraulic loss across discrete blockages

The magnitude of baseflow through a discrete blockage and along the pipeline is critical. This baseflow exacerbates the pressure loss across the discrete blockage and increases the size of the corresponding transient reflection (making the discrete blockage more discernable). That said, there are a number of practical constraints that limit the magnitude of the baseflow that can be established along a pipeline. Importantly, United Water operators specified that the steady state pressure along the pipeline was not to fall below 5m at any location. Furthermore, the pressure at all locations was not to become negative once the transient event was initiated.

For the configuration shown in Figure S-1, the baseflow is governed by the pressure at the tee intersection, the diameter of the pipeline, size of the discharge orifice at the end of the pipeline, friction and the extent of constriction at the discrete blockage (and subsequent pressure loss). The size of the end orifice, required to restrict the baseflow and maintain a minium head of 5m, will vary for each pipeline diameter assuming a constant head of 25m is maintained at the tee intersection.

Block reflections predicted using direct reflection equation

Wylie and Streeter (1993) presented a formulation which can be used to estimate the size of reflected and transmitted wavefronts from a discrete blockage (or in-line gate valve) providing the pressure loss under steady conditions, the baseflow along the pipeline and the blockage coefficient (*K*) are known:

$$\Delta H_T = \frac{-\left(1 + H_V / BQ_V\right) + \sqrt{\left(1 + H_V / BQ_V\right)^2 + Kg\Delta H_W / a^2}}{Kg / 2a^2} \tag{S-1}$$

$$\Delta H_R = 2\Delta H_W - \Delta H_T \tag{S-2}$$

where the terms ΔH_T, ΔH_R, Q_V, H_V and K, the transmitted pressure, reflected pressure, underlying baseflow, steady pressure loss and blockage coefficient, respectively, have been previously defined in Chapter 3

Figure S-2 shows the magnitude of the reflections from a range of discrete blockages (expressed as percentage constrictions of pipeline diameter), determined using Equations S-1 and S-2, and, concurrently, the baseflow versus pipeline diameter. The results are particular to the pipeline configuration shown in Figure S-2, a transient pressure rise of 10m and the requirement that the baseflow in the pipeline be such that a minimum head of 5m is maintained downstream of the blockage.

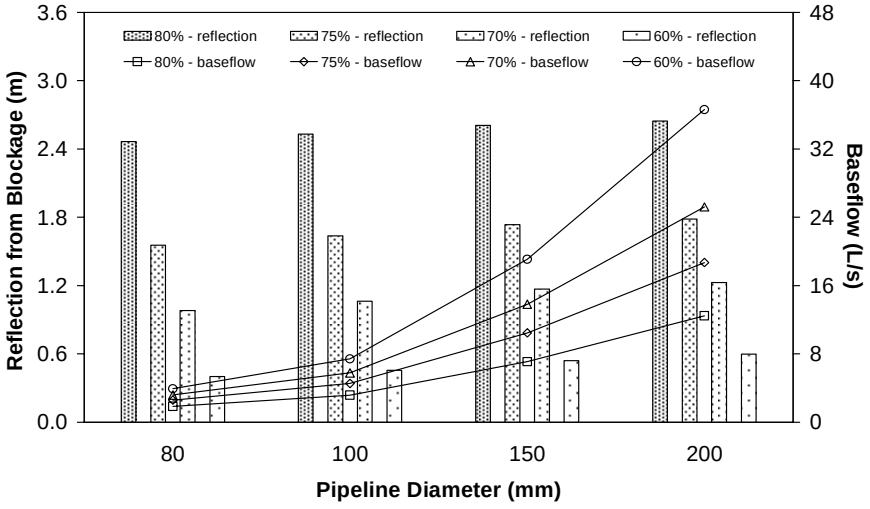


Figure S-2 – Blockage reflection and baseflow versus pipeline diameter for 80, 75, 70 and 60% constrictions

The largest reflection from the discrete blockage is obtained for the greatest percentage constriction despite a lower baseflow. Furthermore, the size of reflection is relatively uniform across a range of pipeline diameters despite an increase in the corresponding baseflow. The proportionality of the blockage reflection to pressure loss across the constriction, which is in turn proportional to the square of the baseflow, accounts for this behaviour.

The results presented above relate to the case where a pipeline has been configured, regardless of diameter and the extent of constriction, for a steady state pressure at the end of the pipeline of 5m. However, it is useful to explore the sensitivity of the blockage reflections to various baseflow and steady loss conditions. Figure S-3 shows the variation in the size of the reflections from the blockages, for 80, 75, 70 and 60% constrictions, when the diameter of the pipeline is fixed to 100mm. Different baseflows are established for a range of pressures downstream of the discrete blockage, and at the end of the pipeline, from 5 to approximately 25m. The pressures downstream of the discrete blockage are labelled in for the 80% constriction.

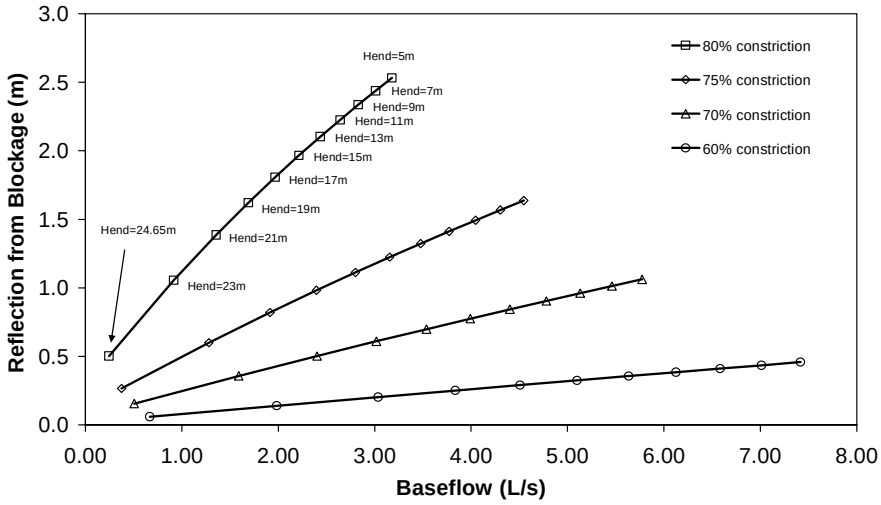


Figure S-3 – Reflection size versus initial flow for 80%, 75%, 70% and 60% constrictions in a 100mm diameter pipeline

Block reflections predicted using a transient model

The formulation presented by Wylie and Streeter (1993) does not take friction into account and therefore overestimates the size of the reflection from a discrete blockage for pipelines with significant friction (or other) losses. That said, the pressure loss associated with a potential blockage may dominate relative to friction loss. If this is the case, then the effect of friction is reduced and less significant. Nevertheless, a transient model should be used to include the effect of friction. The transient model developed in Chapter 11 for the SJTP has been adapted and applied to the pipeline

configuration shown in Figure S-1 above. The sizes of the reflected and transmitted wavefronts from the blockage are recorded at measurement points located 273.5m and 483.2m along the pipeline (these locations correspond to measurement stations 2 and 3 along the SJTP, respectively).

Table S-1 lists the size of the reflected and transmitted wavefronts, predicted using the transient model, for 80, 100, 150 and 200mm diameter pipelines, with a 75% constriction and a variable baseflow, such that 5m of head is maintained downstream of the blockage. The magnitude of the reflected and transmitted wavefronts, predicted using Equations S-1 and S-2, are presented for comparison.

Table S-1 – Blockage reflections determined using a transient model with varying diameters, a 75% constriction and 5m residual head

Diameter (mm)	Q_g (L/s)	Q_V (L/s)	Block reflection stn 2 (m)	Block reflection equation x	Δ refln. (m)	Block trans. stn 3 (m)	Block trans. equation x	Δ trans. (m)
80	0.73	2.60	1.51	1.56	0.05	8.16	8.45	0.29
100	1.20	4.55	1.60	1.64	0.04	8.15	8.36	0.21
150	2.62	10.47	1.70	1.73	0.03	8.14	8.29	0.15
200	4.54	18.70	1.76	1.78	0.02	8.13	8.24	0.11

Table S-2 lists, for comparison, the size of the reflected and transmitted wavefronts, obtained for a 100mm diameter pipeline with 80, 75, 70 and 60% constrictions and a variable baseflow such that 5m of head is maintained downstream of the discrete blockage.

Table S-2 – Blockage reflections determined using a transient model with varying constrictions, a 100mm diameter and 5m residual head

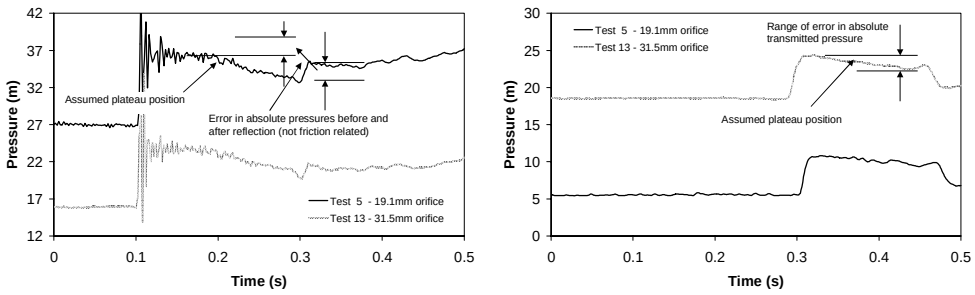
Block (%)	Q_g (L/s)	Q_V (L/s)	Block reflection stn 2 (m)	Block reflection equation x	Δ refln. (m)	Block trans. stn 3 (m)	Block trans. equation x	Δ trans. (m)
80	1.20	3.18	2.48	2.53	0.05	7.31	7.47	0.16
75	1.20	4.55	1.60	1.64	0.04	8.15	8.36	0.21
70	1.20	5.77	1.03	1.06	0.03	8.65	8.94	0.29
60	1.20	7.42	0.44	0.46	0.02	9.14	9.54	0.40

While the size of the reflected wavefronts are significant for all pipeline diameters and percentages of constriction, reflections from other sources along a pipeline, including flexible joints, fittings and water service connections, may also be significant along field pipelines. In this context, establishing a baseflow along a pipeline, and exacerbating pressure loss across any potential blockage, will enhance the size of constriction that may be located.

S.2 Application of direct reflection equation to field results

Interpretation of field results using direct reflection equation

The direct reflection formulation should be carefully applied in pipelines with significant friction or other losses (e.g., mechanical damping) to avoid overestimation of the size of any leak. Figures S-4 and S-5 show the distortion of the pressure plateaus in the measured responses of the SJTP for tests 5 and 13, conducted on the 15th August 2003, at measurement stations 2 and 3, respectively. These distortions relate to the interaction of the wavefronts with fire plugs, flexible joints, bends and water service connections and reduce the accuracy with which Equations S-1 and S-2 can be applied.



Figures S-4 and S-5 – General reflection response for tests 5 and 13, conducted on the 15th August 2003, at measurement stations 2 and 3, respectively

Nevertheless, known blockage coefficients can be theoretically determined using the reflected and transmitted wavefronts from the discrete blockages (neglecting friction

and other losses) using equations S-1 and S-2. Table S-3 shows the parameters adopted for tests 5 and 13. A known blockage coefficient, determined from laboratory calibration, is applied in each case to predict the transmitted pressures using the measured transient overpressures. Changes in elevation contribute significantly to the measured pressures at measurement stations 2 (reflection) and 3 (transmission) and need to be taken into account if the pressure loss across the blockage under steady conditions is to be properly calculated.

Table S-3 – Analysis of measured transient responses using direct reflection formulation

Parameter	Test 5	Test 13	Comment
H_{0UP} (m)	52.04	40.97	HGL includes 25.07m elevation head
H_{0DN} (m)	12.51	25.48	HGL includes 6.95m elevation head
H_V (m)	39.53	15.49	Steady stat head loss across block
H_1 (m)	61.30	48.67	HGL includes 25.07m elevation head
Δ_R (m)	1.822	0.485	Representative reflection excluding accumulator effects
H_2 (m)	63.12	49.16	HGL includes 25.07m elevation head
H_3 (m)	17.58	30.79	HGL includes 6.95m elevation head
a (m/s)	1118.3	1118.3	Average wave speed as previously determined
B (s/m^2)	16223.9	16223.9	Pipeline impedance
A_p (m^2)	0.007014	0.007014	Area of unblocked pipe
D_{orf} (m)	0.0191	0.0315	Diameter of equivalent orifice opening
A_{orf} (m^2)	0.000287	0.000779	Area of equivalent orifice opening
C_dA_{orf} (m^2)	0.000172	0.000467	C_d = 0.6 (as previously explained)
K	1662.9	225.6	Lumped block coefficient
Q_V (L/s)	4.79	8.14	Flow through blockage calculated using $Q_V = \sqrt{2gH_V A_p^2 / K}$
ΔH_W (m)	9.26	7.70	Measured incident transient wave
ΔH_T (m)	**5.07**	**5.31**	Measured transmitted transient wave
ΔH_R (m)	**11.08**	**8.19**	Measured reflected transient wave
ΔH_{Tpred} (m)	**6.06**	**6.87**	Predicted transmitted transient wave
ΔH_{Rpred} (m)	**12.46**	**8.53**	Predicted reflected transient wave
Q_{Vpred} (L/s)	4.43	6.61	Flow predicted using steady model with estimated friction data

The results presented in Table S-3 confirm that the observed reflected and transmitted wavefronts are attenuated by friction and/or mechanical damping, and other discrete and distributed distortions in the measured responses of the STJP.

S.4 Unsteady minor loss effects in the field

Figure S-6 shows that, for both tests 5 and 13, the incident transient wavefronts initially propagate downstream until they reach the in-line gate valve and partially reflect. The reflected wavefronts then propagate upstream, against the direction of the baseflow along the SJTP, until they reach station 2. Under such decelerating flow conditions, any unsteady inertia at the valve will result in the reflected wavefronts arriving earlier than predicted on the basis of quasi-steady analysis. Interestingly, the time of the first rise from the block reflection for test 5 precedes that for test 13 by approximately 4ms.

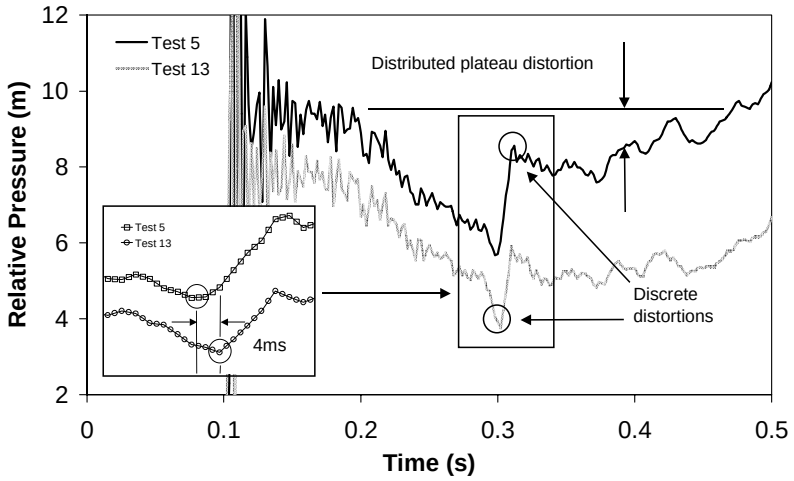


Figure S-6 – Details of reflected wavefronts for tests 5 and 13

That said, the geometry of the constriction for test 5 gives a ratio between the area of the equivalent orifice opening and pipe of approximately 1:25. Prenner (2000) indicated that unsteady inertial effects become significant when the ratio between the area of the equivalent orifice opening and pipe exceeds 1:50. Hence, based on

Appendix S – Direct reflection analysis for block detection and unsteady minor losses

Prenner's work, unsteady inertial effects are not anticipated for test 5. Nevertheless, the reflected wavefront for test 5 precedes that for test 13 by approximately 4ms. Inconsistencies between the threshold identified by Prenner (2000) and that for the SJTP are plausible given geometric differences between the concentric metering orifices used by Prenner (2000) and the aperture formed between the in-line gate valve wedge and seat for the SJTP.

Figure S-7 shows that, for both tests 5 and 13, the incident transient wavefronts initially propagate downstream until they reach the in-line gate valve and partially transmit. The transmitted wavefronts then continue to propagate downstream, in the direction of the baseflow along the pipe, until they reach station 3. Under such accelerating flow conditions, any unsteady inertia will result in the transmitted wavefronts arriving later than predicted on the basis of quasi-steady analysis. Interestingly, a lag of 9ms is observed in the arrival of the transmitted wavefront for test 5 relative to test 13. The magnitude of the lag is significant and consistent with the possibility of unsteady inertial effects.

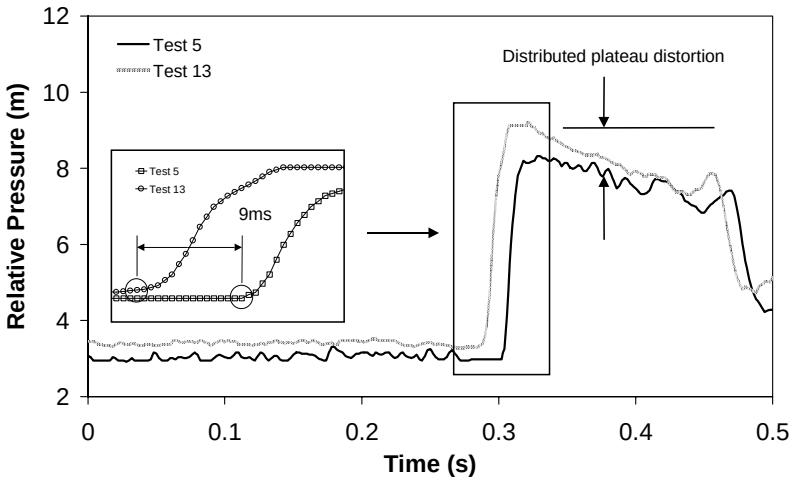


Figure S-7 – Details of transmitted wavefronts for tests 5 and 13

As mentioned above, the geometry of test 5 gives a ratio between the area of the equivalent orifice opening and pipe that is less than the threshold identified by Prenner (2000). Nevertheless, a distinct lag in the arrival of the transmitted

wavefronts is observed for test 5 relative to test 13. Given that unsteady inertial effects are geometry dependent, and the significant differences between the characteristics of the orifices experimentally tested by Prenner (2000) and the simulated discrete blockage formed in the SJTP, the observation of unsteady inertial effects for test 5 seems likely.

Appendix T

Miscellaneous artificial faults for transmission and distribution pipelines

T.1 Leak at air valve/fire plug for Hanson Transmission Pipeline

Figure T-1 shows the convoluted path through the valve seat in the fire plug/air valve used to establish a 9L/s leak on the Hanson Transmission Pipeline (HTP). A relatively low value for C_d of approximately 0.6 is estimated. Knowing that an equivalent aperture opening of approximately 25mm existed across the seat of the valve, when open 6 fo 10 turns to establish the leak discharge, the C_dA_L for the leak was estimated to be approximately 0.0003m^2. Using this C_dA_L, the approximate pressure at the location of the leak (determined using pressure transducer measurements and a correction for change in elevation) and the orifice equation the leak discharge was estimated to be approximately 9L/s.

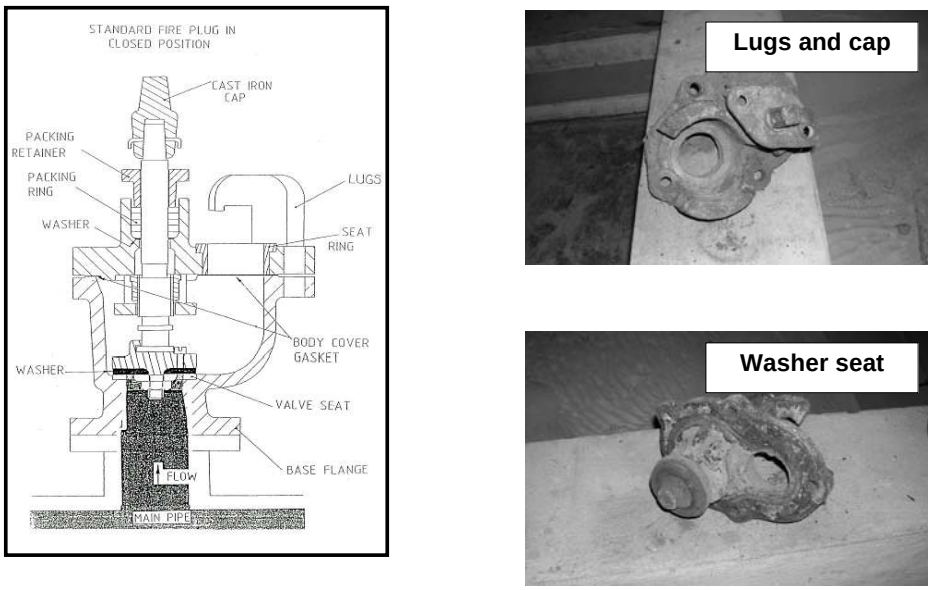


Figure T-1 – Details of the internal configuration of a fire plug/air valve as installed on the Hanson Transmission Pipeline

T.2 Leak nozzle and calibration for the Kookaburra Court Pipeline

The 10mm nozzle was calibrated in the laboratory in the same fashion as the nozzles for the transient generator. The standpipe section with the 10mm nozzle was mounted under the 100mm diameter offtake from the 300mm roof tank discharge pipe. The pressure head from the roof tank water level to the centerline of the 10mm nozzle was approximately 11.7m. The discharge from the 10mm nozzle was directed to the volumetric tank where the time for the depth to increase 0.5m was recorded. Table T-1 shows the calibrated discharge coefficient calculated on the basis of the average of three recorded times.

Table T-1 – Calibration of 10mm nozzle used to simulate leakage for the Kookaburra Court Pipeline

Leak Size (mm)	Area Leak Orifice (m^2)	Volumetric Tank Time (s)				Volume (m^3)	Discharge (m^3 / s)	C_dA_L (x e 03)	C_d
		1	2	3	Avg.				
10	0.0000785	1242.1	1241.4	1242.8	1242.1	1.0925	0.00088	0.0581	0.74

T.3 Air chamber for the Morgan Transmission Pipeline

An artificial air pocket was introduced to the Morgan Transmission Pipeline (MTP), for the tests conducted in May 2004, by attaching a 1.6m high by 250mm square welded mild steel (12mm plate) box section, sealed at the top with a welded plate, to an existing scour valve as shown in Figure T-2. A Perspex window was built into one of the box section walls and a manual pressure gauge, safety valve and compressed air injection port were fitted. The box section had a 150mm diameter pipe section and flange at its base that was used to make a bolted connection to the existing scour valve. The scour valve was then opened to allow the pressure in the MTP to compress the volume of air in the box section.

The volume of unpressurised air contained within the box section was approximately 100L. The pressure at the top of the box section was 53m when the scour valve was opened. Under this pressure, the air column within the box section compressed from a

length of 1.6m to approximately 0.3m. The volume of compressed air contained in the box section was approximately 18.8L.

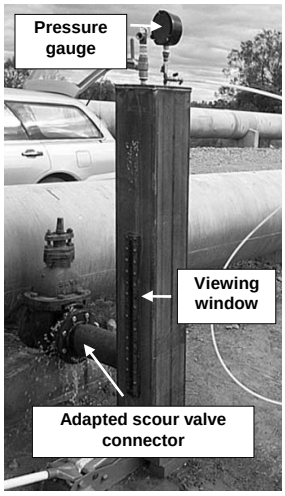


Figure T-2 – Mild steel welded box section used to create an artificial air pocket on the Morgan Transmission Pipeline

The artificial air pocket was trapped at the top of a vertical pipe (square box section) attached to the MTP via a scour valve. This arrangement meant that instead of a simple air pocket, there was a column of water, with a constriction at the scour valve, with a trapped air pocket above, interacting with the transient within the MTP. As a consequence, a pressure fall and then rise are observed with the reflected response from the artificially introduced air pocket.

Appendix U

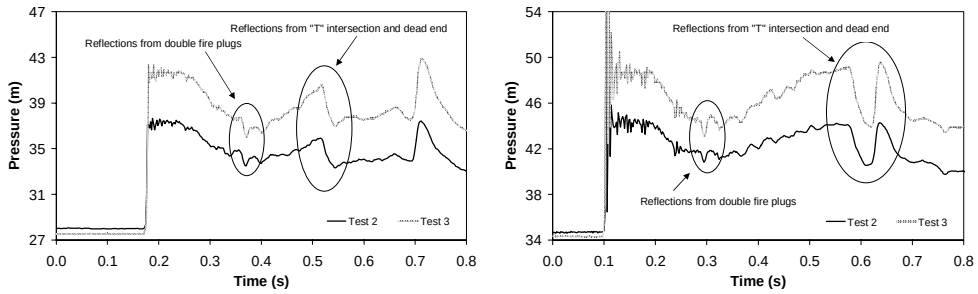
General transient test results for distribution pipelines

U.1 Selection of results for the SJTP in its in-situ condition

Selected tests without baseflow

The tests on the 23rd July 2003 were conducted without any baseflow along the Saint Johns Terrace Pipeline (SJTP). Figure U-1 shows that, for tests 2 and 3, the initial pressure rise at station 1 increases with the size of nozzle. However, the pressure rise is not sustained and a loss and then recovery is observed. Various features are apparent in the measured responses including reflections from the double fire plug risers either side of the in-line gate valve and the "T" intersection with the Willunga Network at approximately 0.37s and 0.52s, respectively.

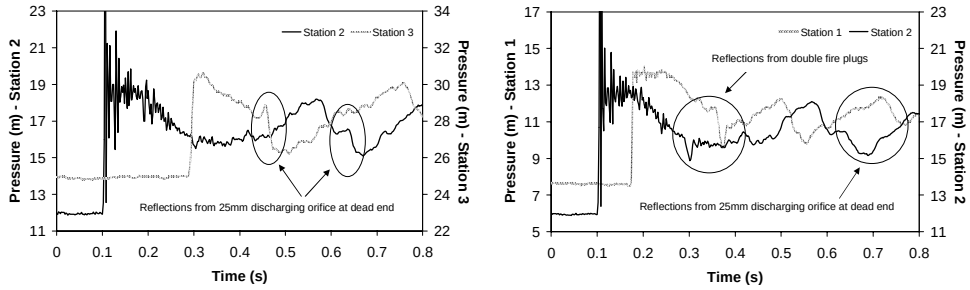
Figure U-2 shows relatively high frequency oscillations at station 2. These oscillations are related to local pressure effects ("ringing") caused by the short vertical branch of pipe comprising the transient generator. Reflections from the fire plug riser at station 1, the double fire plug risers either side of the in-line gate valve and the "T" intersection with the Willunga Network are apparent at approximately 0.24s, 0.29s and 0.58s, respectively. A positive reflection from the dead end of the SJTP is observed at approximately 0.70s and 0.63s, for stations 1 and 2, respectively.



Figures U-1 and U-2 – Measured responses for tests 2 and 3, conducted on 23rd July 2003, over 0.8 seconds, at stations 1 and 2, respectively

Selected tests with baseflow

Baseflow was established along the Saint Johns Terrace Pipeline (SJTP) for the majority of tests conducted on the 15th and 26th August 2003. Figure U-3 shows the response of the SJTP, measured at stations 2 and 3, to a controlled transient induced using an 8mm nozzle for test 3 on the 15th August 2003. As for the case without baseflow, the initial pressure rise is not sustained at station 2 and a loss and then recovery is observed. Furthermore, relatively high frequency oscillations are again apparent following the induction of the transient. The initial pressure rise is not sustained at station 3 and declines at a relatively uniform rate. Figure U-4 shows the response of the SJTP, measured at stations 1 and 2, for test 1 on the 26th August 2003. Relatively high frequency oscillations are again apparent in the recorded response for station 2 but not for station 1.



Figures U-3 and U-4 – Measured responses for test 3, on 15th August 2003, at stations 2 and 3, and for test 1, on 26th August 2003, at stations 1 and 2, over 0.8s

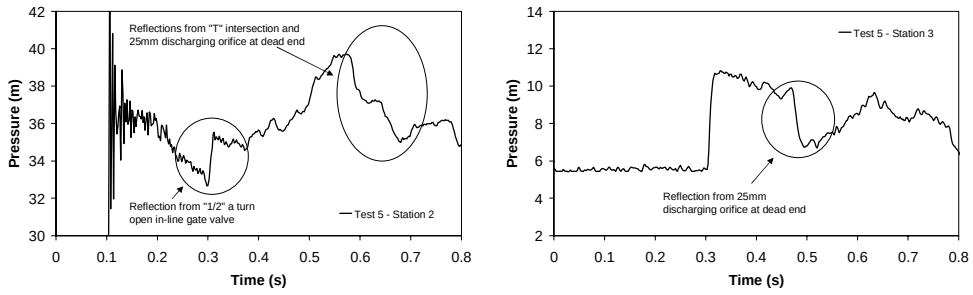
Various features are apparent in the measured responses at station 2 including reflections from the double fire plug risers either side of the in-line gate valve and the "T" intersection with the Willunga Network at approximately 0.30s and 0.58s, respectively. A relatively large negative leak reflection, related to the 25mm diameter orifice discharging at the end of the SJTP, is observed at station 3 at approximately 0.46s.

U.2 Selection of results for the SJTP with blockage and baseflow

Selected tests with a 19.1mm equivalent diameter constriction

Transient tests were conducted on the 15th August 2003 with the in-line gate valve opened only "1/2" a turn from its closed position ("10" turns are required to fully open the in-line gate valve). This formed a severe constriction with an equivalent orifice diameter of approximately 19.1mm (as determined in Appendix I). Similar transient tests were conducted on the 26th August 2003. The same constriction was established but the response was measured at stations 1 and 2 rather than 2 and 3.

Figure U-5 shows the reflected wavefront for test 5, conducted on the 15th August 2003, measured at station 2. A positive reflection from the constriction formed by the "1/2" a turn open in-line gate valve is apparent at 0.30s. This positive reflection is approximately 2m in magnitude and significantly larger than the reflections from other in-situ features along the Saint Johns Terrace Pipeline (SJTP). Figure U-6 shows the transmitted wavefront recorded at station 3. Interestingly, the transmitted wavefront is not significantly attenuated by the constriction.

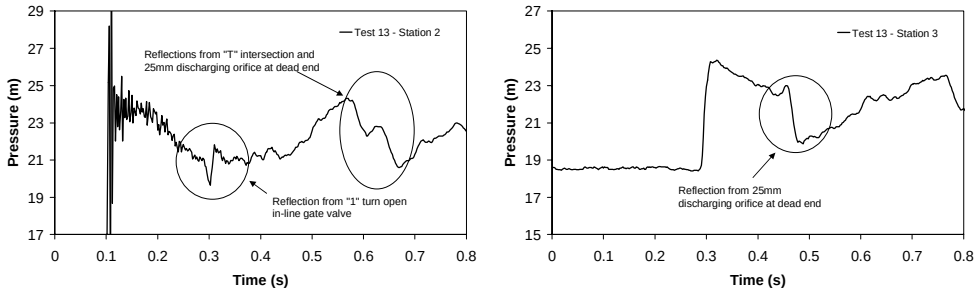


Figures U-5 and U-6 – Measured responses for test 5, conducted on 15th August 2003, with the valve "1/2" a turn open, over 0.8 seconds, at stations 2 and 3, respectively

Selected tests with a 31.5mm equivalent diameter constriction

Transient tests were conducted on the 15th August 2003 with the in-line gate valve opened "1" turn from its closed position. This formed a relatively severe constriction

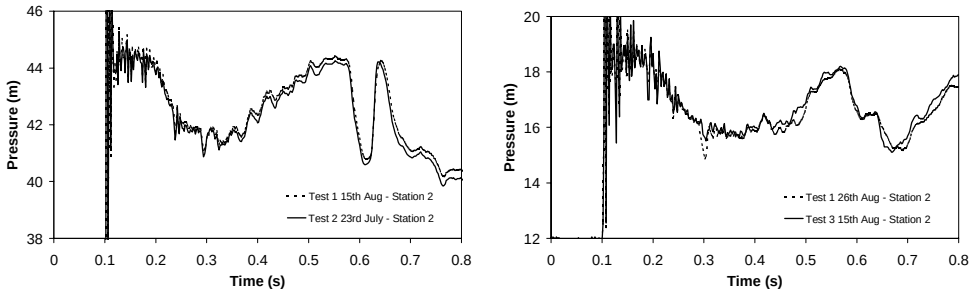
with an equivalent orifice diameter of approximately 31.5mm (as determined in Appendix I). Figure U-7 shows the reflected wavefront for test 13, conducted on the 15th August 2003, measured at station 2. A positive reflection from the constriction formed by the "1" turn open in-line gate valve is apparent at 0.30s. This positive reflection is approximately 0.75m in magnitude. The drop in the magnitude of the positive reflection relative to the case with the in-line gate valve "1/2" a turn open is significant. Figure U-8 shows the transmitted wave recorded at station 3. As for the case with the in-line gate valve "1/2" a turn open, the transmitted wave is not significantly attenuated by the constriction.



Figures U-7 and U-8 – Measured responses for test 13, conducted on 15th August 2003, with the valve "1" turn open, over 0.8 seconds, at stations 2 and 3, respectively

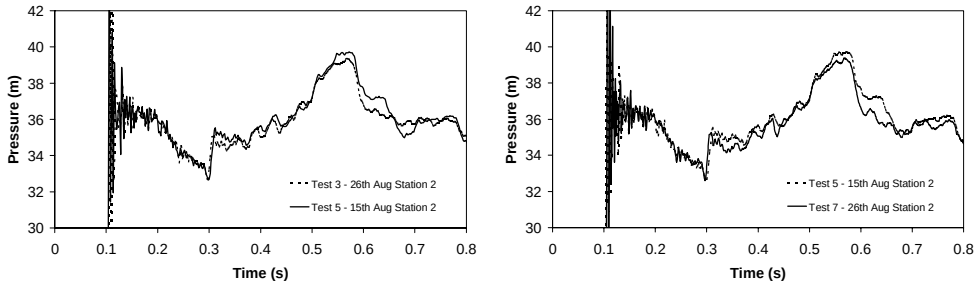
U.3 Consistency in measured responses from the SJTP

Figure U-9 shows the measured responses at station 2 for tests 1 and 2, conducted without baseflow, on the 15th August and 23rd July 2003, respectively. The structured reflections evident in the measured responses are consistent. This suggests that they are not random and are related to invariant physical features along the Saint Johns Terrace Pipeline (SJTP). Figure U-10 shows the measured responses at station 2 for tests 1and 3, conducted with baseflow, on the 26th August and 15th August 2003, respectively. As for the tests conducted without baseflow, the structured reflections evident in the measured responses are consistent. In some cases, reflections can be explained by known elements along the SJTP (e.g., the double fire plug risers at approximately 0.30s).



Figures U-9 and U-10 – Measured responses from the 15th August and 23rd July 2003, and from the 26th and 15th August 2003, respectively, over 0.8s

Figure U-11 shows the measured responses at station 2 for tests 3 and 5, with baseflow and the in-line valve "1/2" a turn open, conducted on the 26th August and 15th August 2003, respectively. As for the tests conducted without simulated blockage, the structured reflections evident in the measured responses, including the reflections from the partially closed valve, are consistent. Figure U-12 shows, in addition to the measured responses for test 5, the responses for test 7, conducted on the 26th August 2003, measured at station 2. The only difference for test 7 is that the recording rate has been increased from 500Hz to 2000Hz.



Figures U-11 and U-12 – Measured responses for tests 3 and 5 from the 26th and 15th August 2003, and for test 7 from the 26th August 2003, respectively, over 0.6s

U.4 Selection of results for the KCP with artificial faults

Selected tests with a 1.635L air pocket

Transient tests were conducted on the Kookaburra Court Pipeline (KCP) on the 28th August 2003 with an artificial 1.635L air pocket. As described in Chapter 6, a column of air contained in a standpipe, was compressed to a volume of 1.635L, under a pressure of 45m, when the fire plug valve at the base of the standpipe was opened. The size of this air pocket was approximately equal to that of another in-situ air pocket that had been inadvertently discovered during general flushing of the Willunga Network, performed by the author, during July 2002.

Figure U-13 shows the measured responses of the KCP at station 2, without and with the air pocket, for tests 3 and 8, respectively. The measured responses are similar until the incident and dead end reflected wavefronts reach the artificial air pocket for test 8. After the wavefronts reach the air pocket the measured responses of the KCP diverge significantly. The pressure drops associated with the reflections of the incident and dead end reflected wavefronts with the air pocket are approximately 4.5m and 7.3m, respectively.

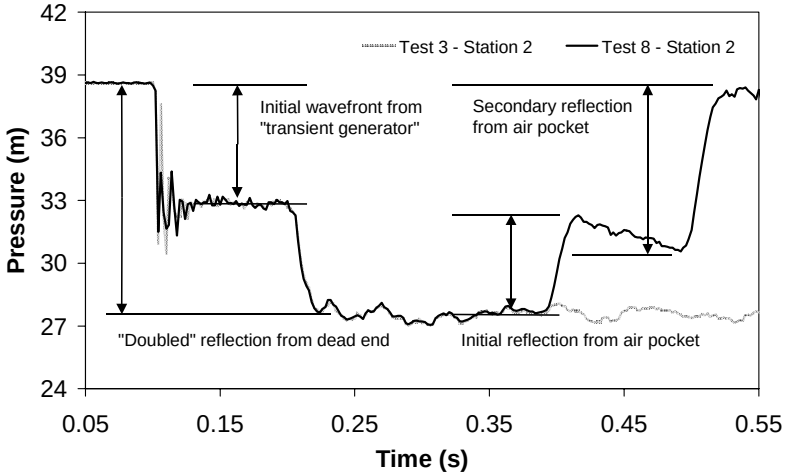
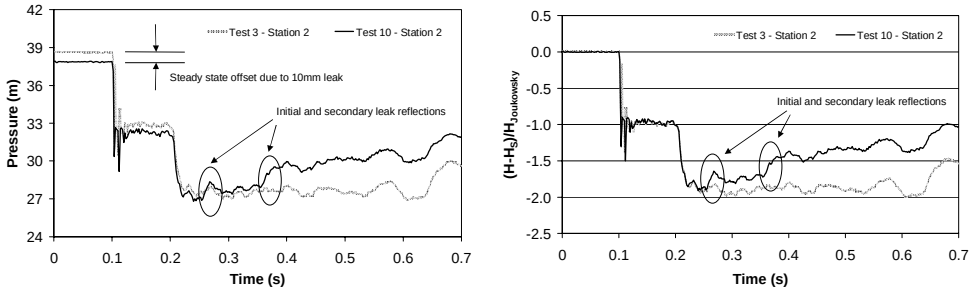


Figure U-13 – Measured response at station 2, without and with a 1.635L air pocket, for tests 3 and 8, conducted on the 28th August 2003, respectively

Selected tests with a 10mm diameter leak

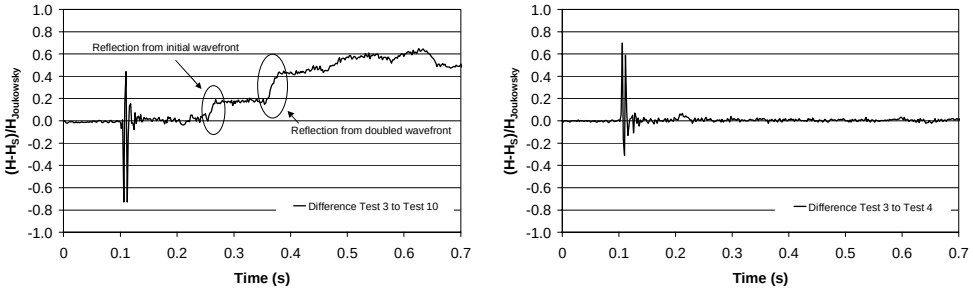
Transient tests were conducted on the 28th August 2003 with an artificial 10mm leak. The artificial leak was installed at the location previously occupied by measurement station 1 (refer to Appendix T for details of the calibration of the 10mm diameter leak). Figure U-14 shows the measured responses of the Kookaburra Court Pipeline (KCP) at station 2, without and with the artificial leak, for tests 3 and 10, respectively. The steady state pressure offset between the no-leak and leak cases complicates the comparison. Figure U-15 shows the dimensionless responses for tests 3 and 10 in which the steady state pressure offset has been eliminated. Leak reflections, from the incident and dead end reflected waves, are apparent at approximately 0.26s and 0.37s, respectively.



Figures U-14 and U-15 – Measured and dimensionless no-leak and leak responses, for tests 3 and 10, conducted on the 28th August 2003, over 0.7s, at station 2

Figure U-16 shows the result of differencing the dimensionless responses of the KCP for tests 3 and 10. Variability associated with the "ringing" effect in the transient generator is apparent at approximately 0.10s. Thereafter, no significant difference between the no-leak and leak cases is observed until a distinct reflection from the interaction of the incident wavefront and artificial leak is recorded at approximately 0.26s. No further significant difference between the no-leak and leak cases is then observed until a secondary reflection from the interaction of the dead end reflected wavefront and artificial leak is recorded at approximately 0.37s.

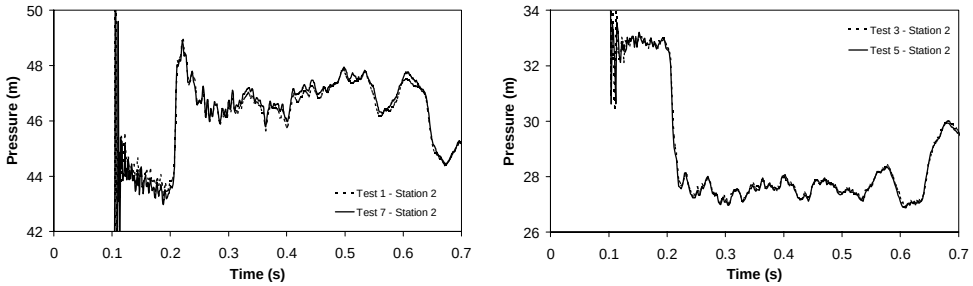
Figure U-17 shows the result of differencing the dimensionless responses of the KCP for tests 3 and 4 (both no-leak cases). Variability associated with the "ringing" effect is again apparent. However, the remainder of the differenced response shows no distinct divergence. This result is significant because it confirms that the non-leak related reflections along the KCP are consistent and can be effectively negated when a historical or previously recorded response is available for comparison.



Figures U-16 and U-17 – Differencing dimensionless responses for no-leak test 3 and leak test 10, and no-leak tests 3 and 4, conducted on the 28th August 2003, over 0.7s

U.5 Consistency in measured responses from the KCP

Figure U-18 shows the measured positive transient responses at station 2, for tests 1 and 7, conducted on the 28th August 2003, using recording rates of 500Hz and 2000Hz, respectively. The structured reflections evident in the measured responses are consistent. This suggests they are not random. Both recording rates capture similar structured reflections in the measured responses. Figure U-19 shows the measured negative transient responses at station 2, for tests 3 and 5, conducted on the 28th August 2003, using recording rates of 500Hz and 2000Hz, respectively. Again, consistent structured reflections are observed. As for the Saint Johns Terrace Pipeline (SJTP), it is thought that the reflections relate to invariant physical features along the KCP that are interacting with incident transient wavefronts in a consistent manner. The magnitude of the reflections is greater for the KCP and there are significant differences between the results for the positive and negative transients.



Figures U-18 and U-19 – Test 1 versus test 7 and test 3 versus test 5, conducted on the 28th August2003, over 0.7s, at station 2, respectively

U.6 Selection of results for the FSP in its in-situ condition

Figures U-20 and U-21 show the measured responses for four tests conducted in the Foster Street Pipeline (FSP) in its in-situ condition. As for the tests on the other distribution pipelines, a "ringing" effect associated with the short vertical branch of pipe comprising the transient generator is apparent. Tests 1 and 2 confirm that a consistent response is extracted from the FSP when configured with the transient generator at station 1. Similarly consistent responses are obtained for tests 3 and 4 when the FSP is configured with the transient generator at station 2.

Figure U-20 shows that, for tests 1 and 2, the "ringing" effect damps out relatively rapidly. However, a longer period oscillation becomes established. Furthermore, no sustained pressure rise is observed and the transmitted wavefront to station 2 is severely attenuated. Figure U-21 shows that, for tests 3 and 4, a multitude of reflections from a combination of the "ringing" effect, the nearby "T" junction and dead end branch, water service connections and, potentially, extended blockage(s) are apparent. Interpreting this signal is a significant challenge. As for tests 1 and 2, the transmitted wavefront to station 1 (i.e., in the reverse direction of travel to that for tests 1 and 2) is severely attenuated.

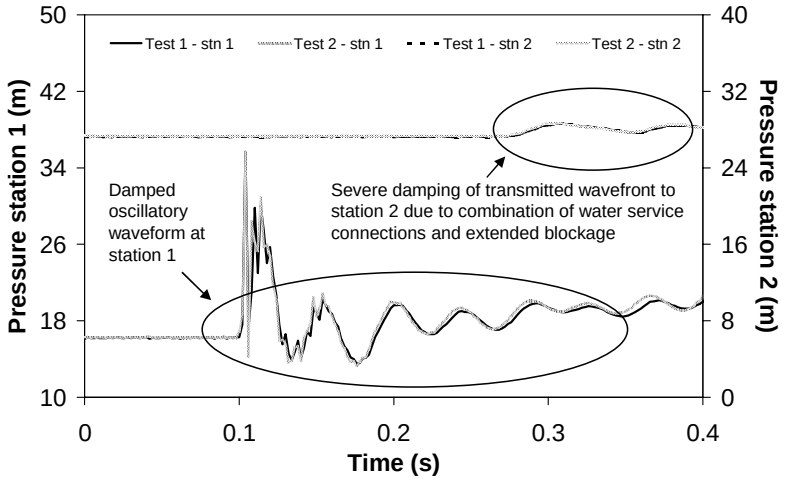


Figure U-20 – Measured responses from the FSP for tests 1 and 2 conducted on the 16th July 2003

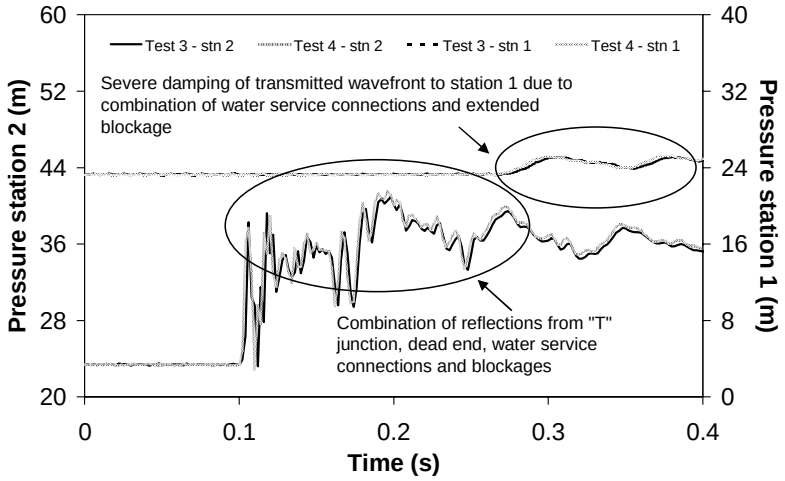


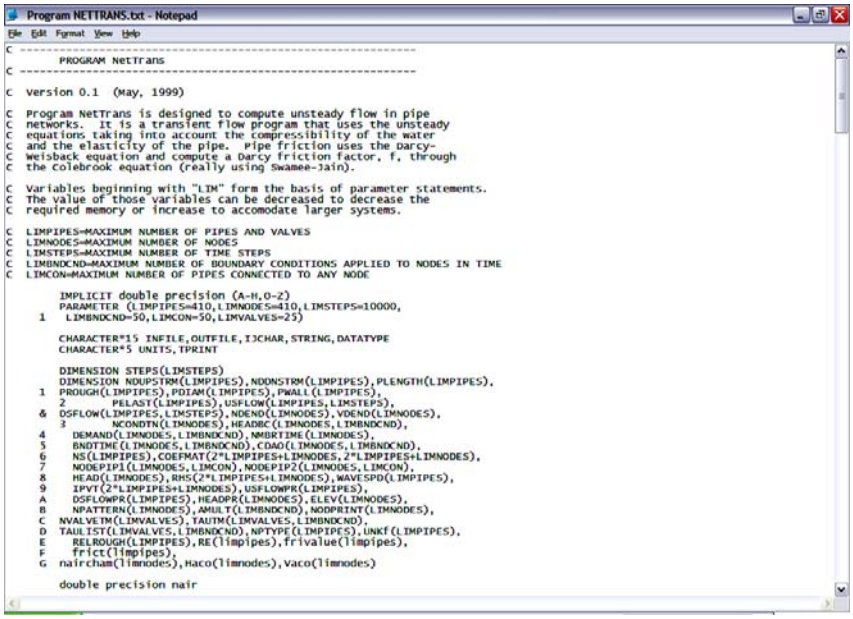
Figure U-21 – Measured responses from the FSP for tests 3 and 4 conducted on the 7th August 2003

Appendix V

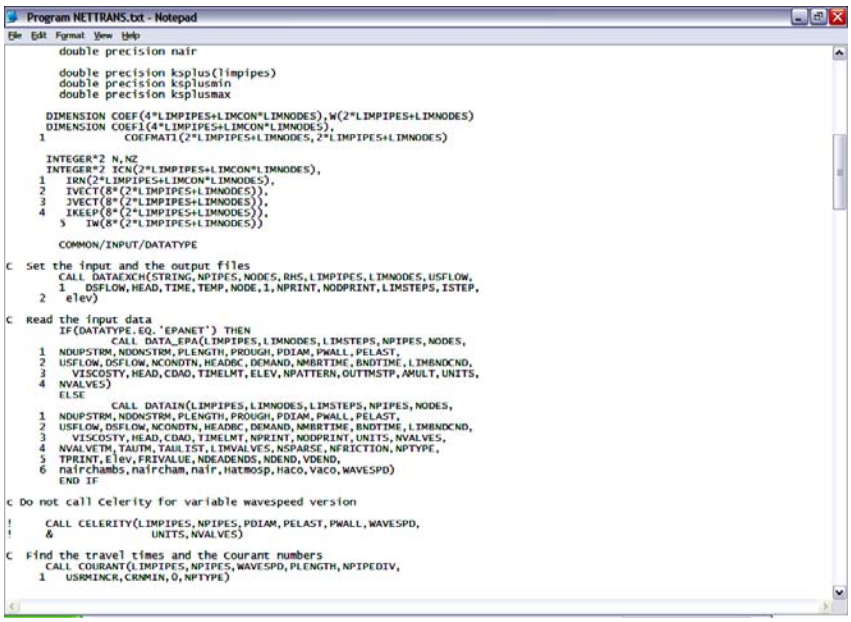
Fortran code for NETTRANS and NLFIT subroutines

V.1 Forward transient subroutines for NLFIT

Program NETTRANS

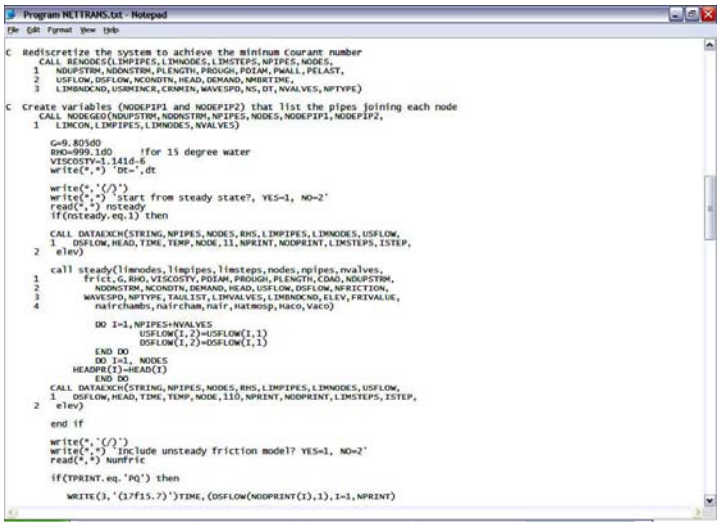



```
C -----------------------------------------------------------------
      PROGRAM NETTrans
C -----------------------------------------------------------------

C  Version 0.1  (May, 1999)

C  Program NetTrans is designed to compute unsteady flow in pipe
C  networks.  It is a transient flow program that uses the unsteady
C  equations taking into account the compressibility of the water
C  and the elasticity of the pipe.  Pipe friction uses the Darcy-
C  Weisback equation and compute a Darcy friction factor, f, through
C  the Colebrook equation (really using Swamee-Jain).

C  Variables beginning with "LIM" form the basis of parameter statements.
C  The value of those variables can be decreased to decrease the
C  required memory or increase to accomodate larger systems.

C  LIMPIPES=MAXIMUM NUMBER OF PIPES AND VALVES
C  LIMNODES=MAXIMUM NUMBER OF NODES
C  LIMSTEPS=MAXIMUM NUMBER OF TIME STEPS
C  LIMBNDCND=MAXIMUM NUMBER OF BOUNDARY CONDITIONS APPLIED TO NODES IN TIME
C  LIMCON=MAXIMUM NUMBER OF PIPES CONNECTED TO ANY NODE

      IMPLICIT double precision (A-H,O-Z)
      PARAMETER (LIMPIPES=410,LIMNODES=410,LIMSTEPS=10000,
     1   LIMBNDCND=50,LIMCON=50,LIMVALVES=25)

      CHARACTER*15 INFILE,OUTFILE,IJCHAR,STRING,DATATYPE
      CHARACTER*5 UNITS,TPRINT

      DIMENSION STEPS(LIMSTEPS)
      DIMENSION NDUPSTRM(LIMPIPES),NDONSTRM(LIMPIPES),PLENGTH(LIMPIPES),
     1  PROUGH(LIMPIPES),PDIAM(LIMPIPES),PWALL(LIMPIPES),
     2          PELAST(LIMPIPES),USFLOW(LIMPIPES,LIMSTEPS),
     &  DSFLOW(LIMPIPES,LIMSTEPS),NDEND(LIMNODES),VDEND(LIMNODES),
     3          NCONDTN(LIMNODES),HEADBC(LIMNODES,LIMBNDCND),
     4    DEMAND(LIMNODES,LIMBNDCND),NMBRTIME(LIMNODES),
     5    BNDTIME(LIMNODES,LIMBNDCND),CDAO(LIMNODES,LIMBNDCND),
     6    NS(LIMPIPES),COEFMAT(2*LIMPIPES+LIMNODES,2*LIMPIPES+LIMNODES),
     7    NODEPIP1(LIMNODES,LIMCON),NODEPIP2(LIMNODES,LIMCON),
     8    HEAD(LIMNODES),RHS(2*LIMPIPES+LIMNODES),WAVESPD(LIMPIPES),
     9    IPVT(2*LIMPIPES+LIMNODES),USFLOWPR(LIMPIPES),
     A    DSFLOWPR(LIMPIPES),HEADPR(LIMNODES),ELEV(LIMNODES),
     B    NPATTERN(LIMNODES),AMULT(LIMBNDCND),NODPRINT(LIMNODES),
     C  NVALVETM(LIMVALVES),TAUTM(LIMVALVES,LIMBNDCND),
     D  TAULIST(LIMVALVES,LIMBNDCND),NPTYPE(LIMPIPES),UNKF(LIMPIPES),
     E    RELROUGH(LIMPIPES),RE(limpipes),frivalue(limpipes),
     F    frict(limpipes),
     G  naircham(limnodes),Haco(limnodes),Vaco(limnodes)

      double precision nair
```

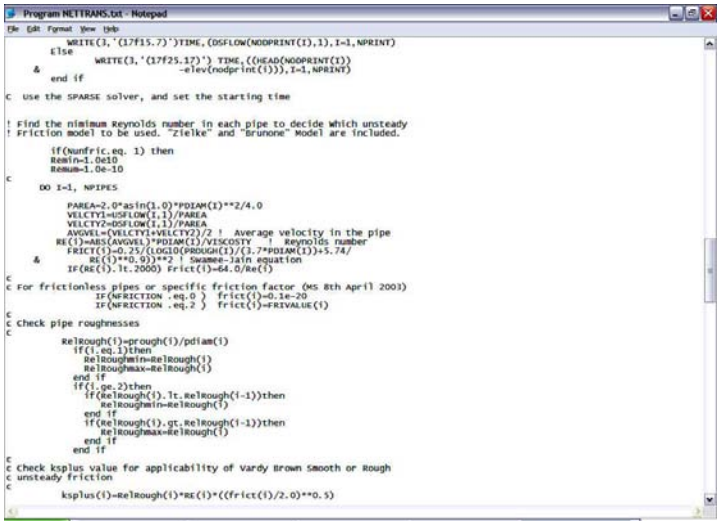



```
      double precision nair

      double precision ksplus(limpipes)
      double precision ksplusmin
      double precision ksplusmax

     DIMENSION COEF(4*LIMPIPES+LIMCON*LIMNODES),W(2*LIMPIPES+LIMNODES)
     DIMENSION COEF1(4*LIMPIPES+LIMCON*LIMNODES),
    1           COEFMAT1(2*LIMPIPES+LIMNODES,2*LIMPIPES+LIMNODES)

     INTEGER*2 N,NZ
     INTEGER*2 ICN(2*LIMPIPES+LIMCON*LIMNODES),
    1  IRN(2*LIMPIPES+LIMCON*LIMNODES),
    2  IVECT(8*(2*LIMPIPES+LIMNODES)),
    3  JVECT(8*(2*LIMPIPES+LIMNODES)),
    4  IKEEP(8*(2*LIMPIPES+LIMNODES)),
    5   IW(8*(2*LIMPIPES+LIMNODES))

      COMMON/INPUT/DATATYPE

C  Set the input and the output files
      CALL DATAEXCH(STRING,NPIPES,NODES,RHS,LIMPIPES,LIMNODES,USFLOW,
     1   DSFLOW,HEAD,TIME,TEMP,NODE,1,NPRINT,NODPRINT,LIMSTEPS,ISTEP,
     2   elev)

C  Read the input data
      IF(DATATYPE.EQ.'EPANET') THEN
            CALL DATA_EPA(LIMPIPES,LIMNODES,LIMSTEPS,NPIPES,NODES,
     1  NDUPSTRM,NDONSTRM,PLENGTH,PROUGH,PDIAM,PWALL,PELAST,
     2  USFLOW,DSFLOW,NCONDTN,HEADBC,DEMAND,NMBRTIME,BNDTIME,LIMBNDCND,
     3    VISCOSTY,HEAD,CDAO,TIMELMT,ELEV,NPATTERN,OUTTMSTP,AMULT,UNITS,
     4  NVALVES)
      ELSE
            CALL DATAIN(LIMPIPES,LIMNODES,LIMSTEPS,NPIPES,NODES,
     1  NDUPSTRM,NDONSTRM,PLENGTH,PROUGH,PDIAM,PWALL,PELAST,
     2  USFLOW,DSFLOW,NCONDTN,HEADBC,DEMAND,NMBRTIME,BNDTIME,LIMBNDCND,
     3    VISCOSTY,HEAD,CDAO,TIMELMT,NPRINT,NODPRINT,UNITS,NVALVES,
     4  NVALVETM,TAUTM,TAULIST,LIMVALVES,NSPARSE,NFRICTION,NPTYPE,
     5  TPRINT,Elev,FRIVALUE,NDEADENDS,NDEND,VDEND,
     6  nairchambs,naircham,nair,Hatmosp,Haco,Vaco,WAVESPD)
      END IF

c Do not call Celerity for variable wavespeed version

!     CALL CELERITY(LIMPIPES,NPIPES,PDIAM,PELAST,PWALL,WAVESPD,
!     &                UNITS,NVALVES)

C  Find the travel times and the Courant numbers
      CALL COURANT(LIMPIPES,NPIPES,WAVESPD,PLENGTH,NPIPEDIV,
     1   USRMINCR,CRNMIN,0,NPTYPE)
```

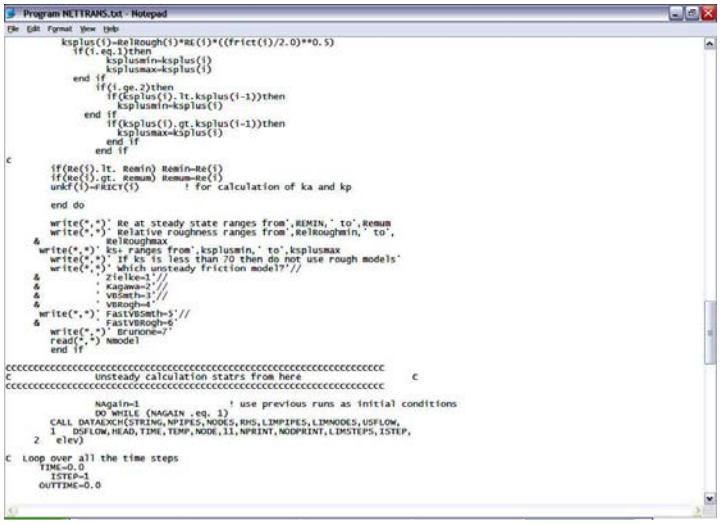
Appendix V – Fortran code for NETTRANS and NLFIT subroutines



```
Program NETTRANS.txt - Notepad
File Edit Format View Help

C  Rediscretize the system to achieve the minimum Courant number
     CALL RENODES(LIMPIPES,LIMNODES,LIMSTEPS,NPIPES,NODES,
     1   NDUPSTRM,NDONSTRM,PLENGTH,PROUGH,PDIAM,PWALL,PELAST,
     2   USFLOW,DSFLOW,NCONDTN,HEAD,DEMAND,NMBRTIME,
     3   LIMBNDCND,USRMINCR,CRMIN,WAVESPD,NS,DT,NVALVES,NPTYPE)

C  Create variables (NODEPIP1 and NODEPIP2) that list the pipes joining each node
     CALL NODEGEO(NDUPSTRM,NDONSTRM,NPIPES,NODES,NODEPIP1,NODEPIP2,
     1   LIMCON,LIMPIPES,LIMNODES,NVALVES)

       G=9.805d0
       RHO=999.1d0     !for 15 degree water
       VISCOSTY=1.141d-6
       write(*,*) 'Dt=',dt

       write(*,'(/)')
       write(*,*) 'start from steady state?, YES=1, NO=2'
       read(*,*) nsteady
       if(nsteady.eq.1) then

       CALL DATAEXCH(STRING,NPIPES,NODES,RHS,LIMPIPES,LIMNODES,USFLOW,
       1   DSFLOW,HEAD,TIME,TEMP,NODE,11,NPRINT,NODPRINT,LIMSTEPS,ISTEP,
     2   elev)

       call steady(limnodes,limpipes,limsteps,nodes,npipes,nvalves,
     1         frict,G,RHO,VISCOSTY,PDIAM,PROUGH,PLENGTH,CDAO,NDUPSTRM,
     2         NDONSTRM,NCONDTN,DEMAND,HEAD,USFLOW,DSFLOW,NFRICTION,
     3         WAVESPD,NPTYPE,TAULIST,LIMVALVES,LIMBNDCND,ELEV,FRIVALUE,
     4           nairchambs,naircham,nair,Hatmosp,Haco,Vaco)

                 DO I=1,NPIPES+NVALVES
                         USFLOW(I,2)=USFLOW(I,1)
                         DSFLOW(I,2)=DSFLOW(I,1)
                 END DO
                 DO I=1, NODES
             HEADPR(I)=HEAD(I)
                 END DO
       CALL DATAEXCH(STRING,NPIPES,NODES,RHS,LIMPIPES,LIMNODES,USFLOW,
       1   DSFLOW,HEAD,TIME,TEMP,NODE,110,NPRINT,NODPRINT,LIMSTEPS,ISTEP,
     2   elev)

       end if

       write(*,'(/)')
       write(*,*) 'include unsteady friction model? YES=1, NO=2'
       read(*,*) Numfric

       if(TPRINT.eq.'PQ') then

          WRITE(3,'(17f15.7)')TIME,(DSFLOW(NODPRINT(I),1),I=1,NPRINT)
```

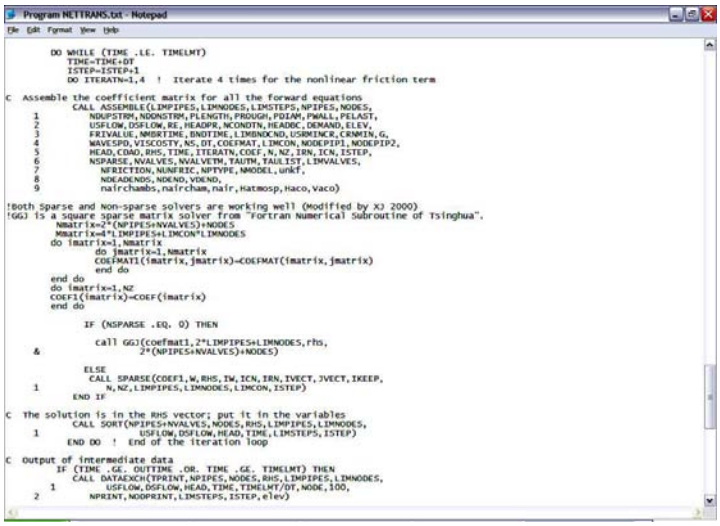


```
Program NETTRANS.txt - Notepad
File Edit Format View Help
          WRITE(3,'(17f15.7)')TIME,(DSFLOW(NODPRINT(I),1),I=1,NPRINT)
       Else
             WRITE(3,'(17f25.17)') TIME,((HEAD(NODPRINT(I))
     &                    -elev(nodprint(i))),I=1,NPRINT)
       end if

C  Use the SPARSE solver, and set the starting time


! Find the minimum Reynolds number in each pipe to decide which unsteady
! Friction model to be used. "Zielke" and "Brunone" Model are included.

       if(Numfric.eq. 1) then
       Remin=1.0e10
       Remum=1.0e-10
C
      DO I=1, NPIPES

          PAREA=2.0*asin(1.0)*PDIAM(I)**2/4.0
          VELCTY1=USFLOW(I,1)/PAREA
          VELCTY2=DSFLOW(I,1)/PAREA
          AVGVEL=(VELCTY1+VELCTY2)/2 !  Average velocity in the pipe
        RE(I)=ABS(AVGVEL)*PDIAM(I)/VISCOSTY    !  Reynolds number
          FRICT(I)=0.25/(LOG10(PROUGH(I)/(3.7*PDIAM(I))+5.74/
     &         RE(I)**0.9))**2 ! Swamee-Jain equation
          IF(RE(I).lt.2000) Frict(I)=64.0/Re(I)
C
C For frictionless pipes or specific friction factor (MS 8th April 2003)
                IF(NFRICTION .eq.0 )  frict(I)=0.1e-20
                IF(NFRICTION .eq.2 )  frict(I)=FRIVALUE(I)
C
C Check pipe roughnesses
C
          RelRough(I)=prough(I)/pdiam(I)
            if(I.eq.1)then
              RelRoughmin=RelRough(I)
              RelRoughmax=RelRough(I)
            end if
            if(I.ge.2)then
              if(RelRough(I).lt.RelRough(I-1))then
                RelRoughmin=RelRough(I)
              end if
              if(RelRough(I).gt.RelRough(I-1))then
                RelRoughmax=RelRough(I)
              end if
            end if
C
C Check ksplus value for applicability of Vardy Brown Smooth or Rough
C unsteady friction
C
          ksplus(I)=RelRough(I)*RE(I)*(frict(I)/2.0)**0.5)
```

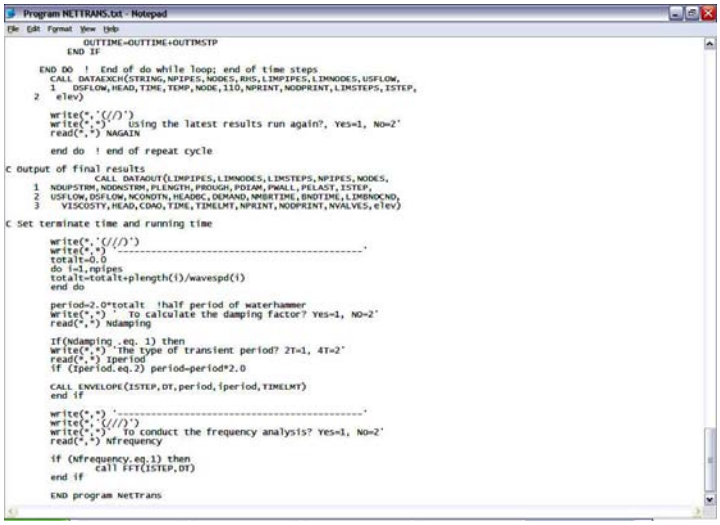


```
Program NETTRANS.txt - Notepad
File Edit Format View Help
          ksplus(I)=RelRough(I)*RE(I)*(frict(I)/2.0)**0.5)
            if(I.eq.1)then
                ksplusmin=ksplus(I)
                ksplusmax=ksplus(I)
            end if
                if(I.ge.2)then
                  if(ksplus(I).lt.ksplus(I-1))then
                    ksplusmin=ksplus(I)
                  end if
                  if(ksplus(I).gt.ksplus(I-1))then
                    ksplusmax=ksplus(I)
                  end if
                end if
C
       if(Re(I).lt. Remin) Remin=Re(I)
       if(Re(I).gt. Remum) Remum=Re(I)
       unkf(I)=FRICT(I)        ! for calculation of ka and kp

       end do

       write(*,*)' Re at steady state ranges from',REMIN,' to',Remum
       write(*,*)' Relative roughness ranges from',RelRoughmin,' to',
     &           RelRoughmax
       write(*,*)' Ks= ranges from',ksplusmin,' to',ksplusmax
       write(*,*)' If ks is less than 70 then do not use rough models'
       write(*,*)' Which unsteady friction model?'//
     &         ' Zielke=1'//
     &         ' Kagawa=2'//
     &         ' VBSmth=3'//
     &         ' VBRogh=4'
      write(*,*)' FastVBSmth=5'//
     &         ' FastVBRogh=6'
       write(*,*)' Brunone=7'
       read(*,*) Nmodel
       end if

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C              Unsteady calculation statrs from here              C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

              NAgain=1                ! use previous runs as initial conditions
              DO WHILE (NAGAIN .eq. 1)
       CALL DATAEXCH(STRING,NPIPES,NODES,RHS,LIMPIPES,LIMNODES,USFLOW,
       1   DSFLOW,HEAD,TIME,TEMP,NODE,11,NPRINT,NODPRINT,LIMSTEPS,ISTEP,
     2   elev)

C  Loop over all the time steps
      TIME=0.0
        ISTEP=1
      OUTTIME=0.0
```

Appendix V – Fortran code for NETTRANS and NLFIT subroutines

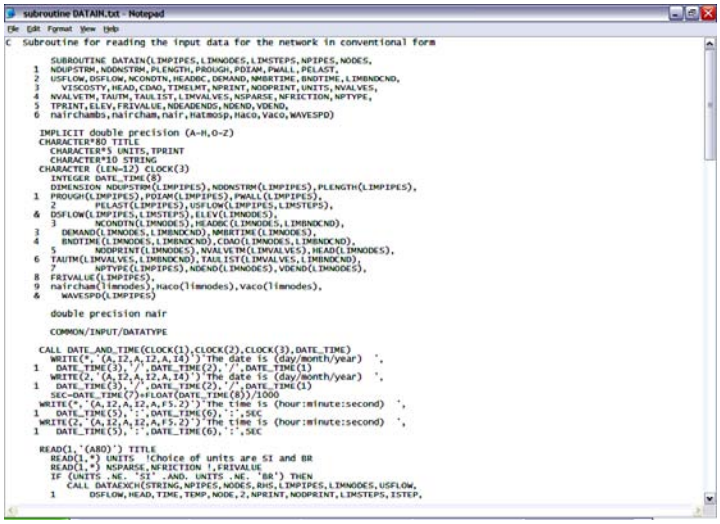



```
      DO WHILE (TIME .LE. TIMELMT)
         TIME=TIME+DT
         ISTEP=ISTEP+1
         DO ITERATN=1,4  !  Iterate 4 times for the nonlinear friction term

C  Assemble the coefficient matrix for all the forward equations
           CALL ASSEMBLE(LIMPIPES,LIMNODES,LIMSTEPS,NPIPES,NODES,
     1          NDUPSTRM,NDDNSTRM,PLENGTH,PROUGH,PDIAM,PWALL,PELAST,
     2          USFLOW,DSFLOW,RE,HEADPR,NCONDTN,HEADBC,DEMAND,ELEV,
     3          FRIVALUE,NMBRTIME,BNDTIME,LIMBNDCND,USRMINCR,CRNMIN,G,
     4          WAVESPD,VISCOSTY,NS,DT,COEFMAT,LIMCON,NODEPIP1,NODEPIP2,
     5          HEAD,CDAO,RHS,TIME,ITERATN,COEF,N,NZ,IRN,ICN,ISTEP,
     6          NSPARSE,NVALVES,NVALVETM,TAUTM,TAULIST,LIMVALVES,
     7            NFRICTION,NUNFRIC,NPTYPE,NMODEL,unkf,
     8            NDEADENDS,NDEND,VDEND,
     9            nairchambs,nairchan,nair,Hatmosp,Haco,Vaco)

!Both Sparse and Non-sparse solvers are working well (Modified by XJ 2000)
!GGJ is a square sparse matrix solver from "Fortran Numerical Subroutine of Tsinghua".
          NMatrix=2*(NPIPES+NVALVES)+NODES
          NMatrix=4*LIMPIPES+LIMCON*LIMNODES
         do imatrix=1,Nmatrix
                do jmatrix=1,NMatrix
                COEFMAT1(imatrix,jmatrix)=COEFMAT(imatrix,jmatrix)
                end do
         end do
         do imatrix=1,NZ
         COEF1(imatrix)=COEF(imatrix)
         end do

               IF (NSPARSE .EQ. 0) THEN

                 call GGJ(coefmat1,2*LIMPIPES+LIMNODES,rhs,
     &                    2*(NPIPES+NVALVES)+NODES)

               ELSE
                 CALL SPARSE(COEF1,W,RHS,IW,ICN,IRN,IVECT,JVECT,IKEEP,
     1              N,NZ,LIMPIPES,LIMNODES,LIMCON,ISTEP)
               END IF

C  The solution is in the RHS vector; put it in the variables
           CALL SORT(NPIPES+NVALVES,NODES,RHS,LIMPIPES,LIMNODES,
     1                USFLOW,DSFLOW,HEAD,TIME,LIMSTEPS,ISTEP)
         END DO  !  End of the iteration loop

C  Output of intermediate data
         IF (TIME .GE. OUTTIME .OR. TIME .GE. TIMELMT) THEN
           CALL DATAEXCH(TPRINT,NPIPES,NODES,RHS,LIMPIPES,LIMNODES,
     1          USFLOW,DSFLOW,HEAD,TIME,TIMELMT/DT,NODE,100,
     2          NPRINT,NODPRINT,LIMSTEPS,ISTEP,elev)
```



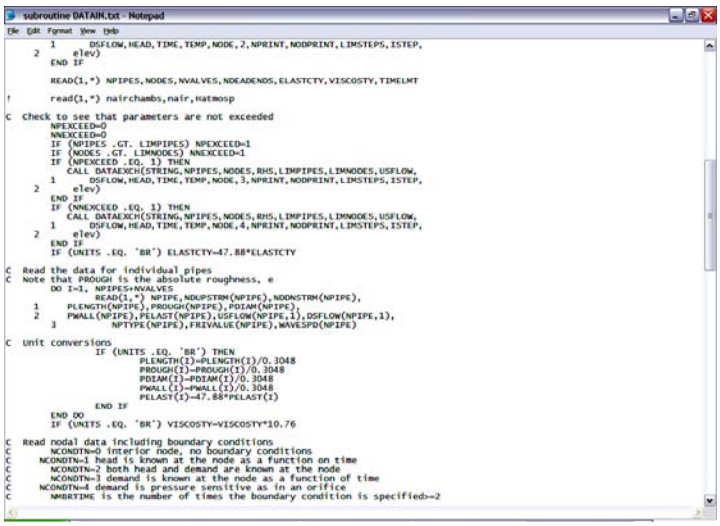

```
           OUTTIME=OUTTIME+OUTTMSTP
         END IF

       END DO  !  End of do while loop; end of time steps
         CALL DATAEXCH(STRING,NPIPES,NODES,RHS,LIMPIPES,LIMNODES,USFLOW,
     1    DSFLOW,HEAD,TIME,TEMP,NODE,110,NPRINT,NODPRINT,LIMSTEPS,ISTEP,
     2    elev)

         Write(*,'(//)')
         Write(*,*)  ' Using the latest results run again?, Yes=1, NO=2'
         read(*,*) NAGAIN

         end do  ! end of repeat cycle

C Output of final results
              CALL DATAOUT(LIMPIPES,LIMNODES,LIMSTEPS,NPIPES,NODES,
     1  NDUPSTRM,NDDNSTRM,PLENGTH,PROUGH,PDIAM,PWALL,PELAST,ISTEP,
     2  USFLOW,DSFLOW,NCONDTN,HEADBC,DEMAND,NMBRTIME,BNDTIME,LIMBNDCND,
     3    VISCOSTY,HEAD,CDAO,TIME,TIMELMT,NPRINT,NODPRINT,NVALVES,elev)

C Set terminate time and running time

         Write(*,'(//)')
         Write(*,*)  '-----------------------------------------------'
         totalt=0.0
         do i=1,npipes
         totalt=totalt+plength(i)/wavespd(i)
         end do

         period=2.0*totalt  !half period of waterhammer
         Write(*,*)  ' To calculate the damping factor? Yes=1, NO=2'
         read(*,*) ndamping

         If(ndamping .eq. 1) then
         Write(*,*) 'The type of transient period? 2T=1, 4T=2'
         read(*,*) iperiod
         if (iperiod.eq.2) period=period*2.0

         CALL ENVELOPE(ISTEP,DT,period,iperiod,TIMELMT)
         end if

         Write(*,*) '-----------------------------------------------'
         Write(*,'(///)')
         Write(*,*)  '  To conduct the frequency analysis? Yes=1, NO=2'
         read(*,*) Nfrequency

         if (Nfrequency.eq.1) then
                call FFT(ISTEP,DT)
         end if

         END program NetTrans
```

Subroutine DATAIN

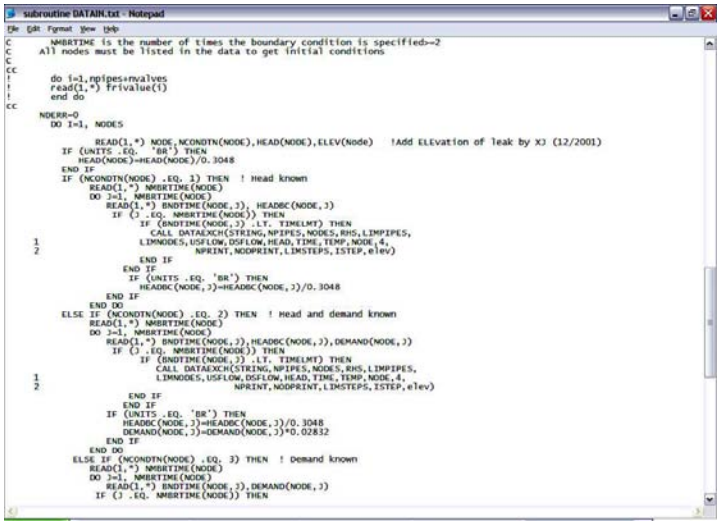



```
C  Subroutine for reading the input data for the network in conventional form

       SUBROUTINE DATAIN(LIMPIPES,LIMNODES,LIMSTEPS,NPIPES,NODES,
     1  NDUPSTRM,NDDNSTRM,PLENGTH,PROUGH,PDIAM,PWALL,PELAST,
     2  USFLOW,DSFLOW,NCONDTN,HEADBC,DEMAND,NMBRTIME,BNDTIME,LIMBNDCND,
     3    VISCOSTY,HEAD,CDAO,TIMELMT,NPRINT,NODPRINT,UNITS,NVALVES,
     4  NVALVETM,TAUTM,TAULIST,LIMVALVES,NSPARSE,NFRICTION,NPTYPE,
     5  TPRINT,ELEV,FRIVALUE,NDEADENDS,NDEND,VDEND,
     6  nairchambs,nairchan,nair,Hatmosp,Haco,Vaco,WAVESPD)

        IMPLICIT double precision (A-H,O-Z)
        CHARACTER*80 TITLE
          CHARACTER*5 UNITS,TPRINT
          CHARACTER*10 STRING
        CHARACTER (LEN=12) CLOCK(3)
          INTEGER DATE_TIME(8)
          DIMENSION NDUPSTRM(LIMPIPES),NDDNSTRM(LIMPIPES),PLENGTH(LIMPIPES),
     1  PROUGH(LIMPIPES),PDIAM(LIMPIPES),PWALL(LIMPIPES),
     2        PELAST(LIMPIPES),USFLOW(LIMPIPES,LIMSTEPS),
     &  DSFLOW(LIMPIPES,LIMSTEPS),ELEV(LIMNODES),
     3        NCONDTN(LIMNODES),HEADBC(LIMNODES,LIMBNDCND),
     3    DEMAND(LIMNODES,LIMBNDCND),NMBRTIME(LIMNODES),
     4    BNDTIME(LIMNODES,LIMBNDCND),CDAO(LIMNODES,LIMBNDCND),
     5        NDOPRINT(LIMNODES),NVALVETM(LIMVALVES),HEAD(LIMNODES),
     6  TAUTM(LIMVALVES,LIMBNDCND),TAULIST(LIMVALVES,LIMBNDCND),
     7        NPTYPE(LIMPIPES),NDEND(LIMNODES),VDEND(LIMNODES),
     8  FRIVALUE(LIMPIPES),
     9  nairchan(limnodes),Haco(limnodes),Vaco(limnodes),
     &    WAVESPD(LIMPIPES)

          double precision nair

          COMMON/INPUT/DATATYPE

         CALL DATE_AND_TIME(CLOCK(1),CLOCK(2),CLOCK(3),DATE_TIME)
          WRITE(*,'(A,I2,A,I2,A,I4)')'The date is (day/month/year)  ',
     1    DATE_TIME(3),'/',DATE_TIME(2),'/',DATE_TIME(1)
          WRITE(2,'(A,I2,A,I2,A,I4)')'The date is (day/month/year)  ',
     1    DATE_TIME(3),'/',DATE_TIME(2),'/',DATE_TIME(1)
          SEC=DATE_TIME(7)+FLOAT(DATE_TIME(8))/1000
         WRITE(*,'(A,I2,A,I2,A,F5.2)')'The time is (hour:minute:second)  ',
     1    DATE_TIME(5),':',DATE_TIME(6),':',SEC
         WRITE(2,'(A,I2,A,I2,A,F5.2)')'The time is (hour:minute:second)  ',
     1    DATE_TIME(5),':',DATE_TIME(6),':',SEC

         READ(1,'(A80)') TITLE
          READ(1,*) UNITS  !Choice of units are SI and BR
          READ(1,*) NSPARSE,NFRICTION ,FRIVALUE
          IF (UNITS .NE. 'SI' .AND. UNITS .NE. 'BR') THEN
            CALL DATAEXCH(STRING,NPIPES,NODES,RHS,LIMPIPES,LIMNODES,USFLOW,
     1        DSFLOW,HEAD,TIME,TEMP,NODE,2,NPRINT,NODPRINT,LIMSTEPS,ISTEP,
```

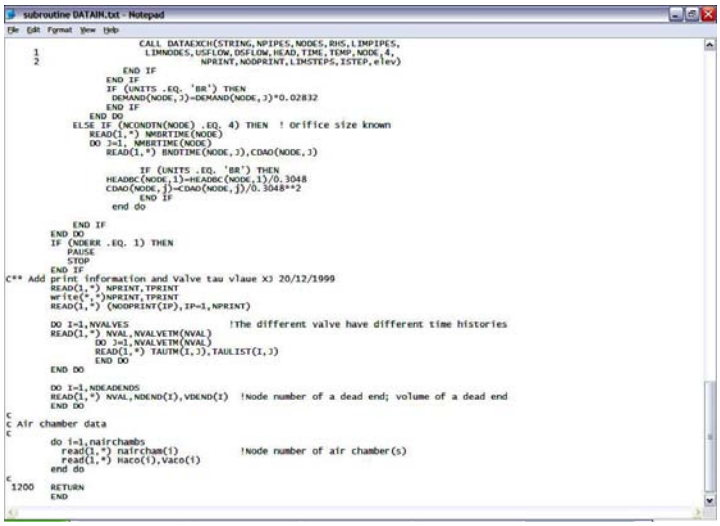
Appendix V – Fortran code for NETTRANS and NLFIT subroutines



```
subroutine DATAIN.txt - Notepad
File  Edit  Format  View  Help
      1      DSFLOW,HEAD,TIME,TEMP,NODE,2,NPRINT,NODPRINT,LIMSTEPS,ISTEP,
     2     eTev)
        END IF

        READ(1,*) NPIPES,NODES,NVALVES,NDEADENDS,ELASTCTY,VISCOSTY,TIMELMT

!       read(1,*) nairchambs,nair,Hatmosp

C  Check to see that parameters are not exceeded
        NPEXCEED=0
        NNEXCEED=0
        IF (NPIPES .GT. LIMPIPES) NPEXCEED=1
        IF (NODES .GT. LIMNODES) NNEXCEED=1
        IF (NPEXCEED .EQ. 1) THEN
           CALL DATAEXCH(STRING,NPIPES,NODES,RHS,LIMPIPES,LIMNODES,USFLOW,
     1      DSFLOW,HEAD,TIME,TEMP,NODE,3,NPRINT,NODPRINT,LIMSTEPS,ISTEP,
     2     eTev)
        END IF
        IF (NNEXCEED .EQ. 1) THEN
           CALL DATAEXCH(STRING,NPIPES,NODES,RHS,LIMPIPES,LIMNODES,USFLOW,
     1      DSFLOW,HEAD,TIME,TEMP,NODE,4,NPRINT,NODPRINT,LIMSTEPS,ISTEP,
     2     eTev)
        END IF
        IF (UNITS .EQ. 'BR') ELASTCTY=47.88*ELASTCTY

C  Read the data for individual pipes
C  Note that PROUGH is the absolute roughness, e
        DO I=1, NPIPES+NVALVES
              READ(1,*) NPIPE,NDUPSTRM(NPIPE),NDONSTRM(NPIPE),
     1     PLENGTH(NPIPE),PROUGH(NPIPE),PDIAM(NPIPE),
     2     PWALL(NPIPE),PELAST(NPIPE),USFLOW(NPIPE,1),DSFLOW(NPIPE,1),
     3           NPTYPE(NPIPE),FRIVALUE(NPIPE),WAVESPD(NPIPE)

C  Unit conversions
                IF (UNITS .EQ. 'BR') THEN
                     PLENGTH(I)=PLENGTH(I)/0.3048
                     PROUGH(I)=PROUGH(I)/0.3048
                     PDIAM(I)=PDIAM(I)/0.3048
                     PWALL(I)=PWALL(I)/0.3048
                     PELAST(I)=47.88*PELAST(I)
                END IF
        END DO
        IF (UNITS .EQ. 'BR') VISCOSTY=VISCOSTY*10.76

C  Read nodal data including boundary conditions
C     NCONDTN=0 interior node, no boundary conditions
C     NCONDTN=1 head is known at the node as a function on time
C     NCONDTN=2 both head and demand are known at the node
C     NCONDTN=3 demand is known at the node as a function of time
C     NCONDTN=4 demand is pressure sensitive as in an orifice
C     NMBRTIME is the number of times the boundary condition is specified>=2
```

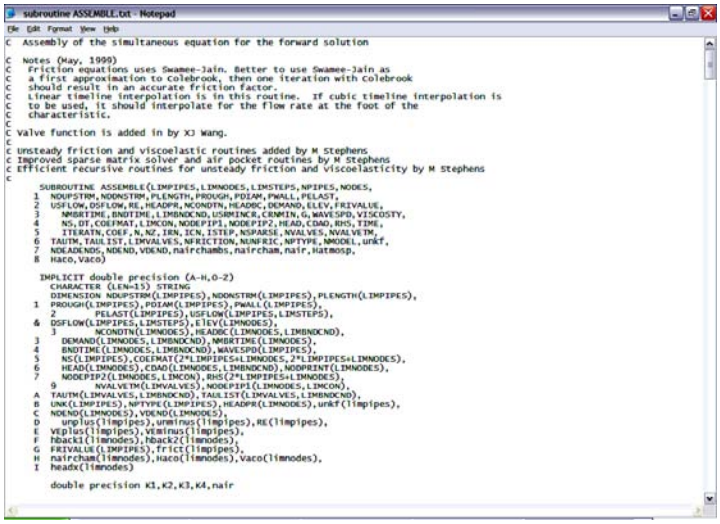


```
subroutine DATAIN.txt - Notepad
File  Edit  Format  View  Help
C     NMBRTIME is the number of times the boundary condition is specified>=2
C     All nodes must be listed in the data to get initial conditions
C
Cc
!       do i=1,npipes+nvalves
!       read(1,*) frivalue(i)
!       end do
cc
        NDERR=0
        DO I=1, NODES

              READ(1,*) NODE,NCONDTN(NODE),HEAD(NODE),ELEV(Node)   !Add ELEvation of leak by XJ (12/2001)
           IF (UNITS .EQ.  'BR') THEN
             HEAD(NODE)=HEAD(NODE)/0.3048
           END IF
           IF (NCONDTN(NODE) .EQ. 1) THEN  ! Head known
                READ(1,*) NMBRTIME(NODE)
                DO J=1, NMBRTIME(NODE)
                READ(1,*) BNDTIME(NODE,J), HEADBC(NODE,J)
                 IF (J .EQ. NMBRTIME(NODE)) THEN
                       IF (BNDTIME(NODE,J) .LT. TIMELMT) THEN
                         CALL DATAEXCH(STRING,NPIPES,NODES,RHS,LIMPIPES,
     1                  LIMNODES,USFLOW,DSFLOW,HEAD,TIME,TEMP,NODE,4,
     2                                NPRINT,NODPRINT,LIMSTEPS,ISTEP,elev)
                       END IF
                     END IF
                     IF (UNITS .EQ. 'BR') THEN
                       HEADBC(NODE,J)=HEADBC(NODE,J)/0.3048
                     END IF
                END DO
           ELSE IF (NCONDTN(NODE) .EQ. 2) THEN  ! Head and demand known
                READ(1,*) NMBRTIME(NODE)
                DO J=1, NMBRTIME(NODE)
                   READ(1,*) BNDTIME(NODE,J),HEADBC(NODE,J),DEMAND(NODE,J)
                    IF (J .EQ. NMBRTIME(NODE)) THEN
                          IF (BNDTIME(NODE,J) .LT. TIMELMT) THEN
                            CALL DATAEXCH(STRING,NPIPES,NODES,RHS,LIMPIPES,
     1                       LIMNODES,USFLOW,DSFLOW,HEAD,TIME,TEMP,NODE,4,
     2                                   NPRINT,NODPRINT,LIMSTEPS,ISTEP,elev)
                          END IF
                    END IF
                  IF (UNITS .EQ. 'BR') THEN
                    HEADBC(NODE,J)=HEADBC(NODE,J)/0.3048
                    DEMAND(NODE,J)=DEMAND(NODE,J)*0.02832
                  END IF
                END DO
             ELSE IF (NCONDTN(NODE) .EQ. 3) THEN  ! Demand known
                READ(1,*) NMBRTIME(NODE)
                DO J=1, NMBRTIME(NODE)
                   READ(1,*) BNDTIME(NODE,J),DEMAND(NODE,J)
                 IF (J .EQ. NMBRTIME(NODE)) THEN
```

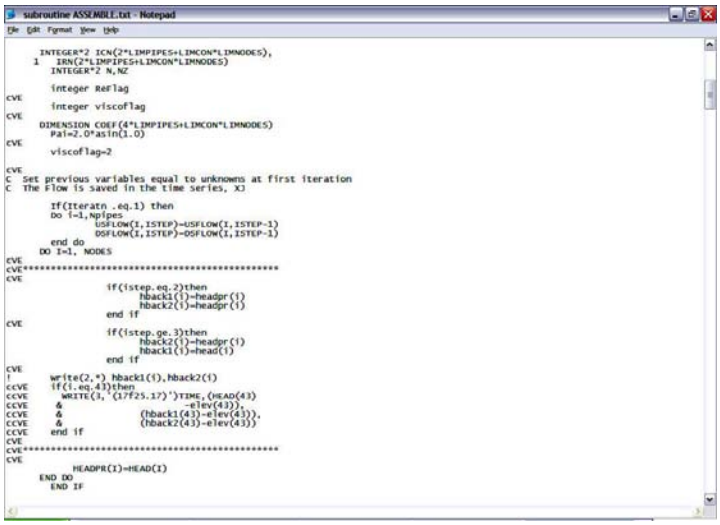


```
subroutine DATAIN.txt - Notepad
File  Edit  Format  View  Help
                          CALL DATAEXCH(STRING,NPIPES,NODES,RHS,LIMPIPES,
     1                       LIMNODES,USFLOW,DSFLOW,HEAD,TIME,TEMP,NODE,4,
     2                                   NPRINT,NODPRINT,LIMSTEPS,ISTEP,elev)
                          END IF
                       END IF
                       IF (UNITS .EQ. 'BR') THEN
                        DEMAND(NODE,J)=DEMAND(NODE,J)*0.02832
                       END IF
                    END DO
             ELSE IF (NCONDTN(NODE) .EQ. 4) THEN  ! Orifice size known
                READ(1,*) NMBRTIME(NODE)
                DO J=1, NMBRTIME(NODE)
                   READ(1,*) BNDTIME(NODE,J),CDAO(NODE,J)

                             IF (UNITS .EQ. 'BR') THEN
                          HEADBC(NODE,1)=HEADBC(NODE,1)/0.3048
                          CDAO(NODE,J)=CDAO(NODE,J)/0.3048**2
                             END IF
                           end do

                END IF
             END DO
             IF (NDERR .EQ. 1) THEN
                PAUSE
                STOP
             END IF
C** Add print information and Valve tau vlaue XJ 20/12/1999
             READ(1,*) NPRINT,TPRINT
             write(*,*)NPRINT,TPRINT
             READ(1,*) (NODPRINT(IP),IP=1,NPRINT)

             DO I=1,NVALVES                  !The different valve have different time histories
             READ(1,*) NVAL,NVALVETM(NVAL)
                    DO J=1,NVALVETM(NVAL)
                    READ(1,*) TAUTM(I,J),TAULIST(I,J)
                    END DO
             END DO

             DO I=1,NDEADENDS
             READ(1,*) NVAL,NDEND(I),VDEND(I)  !Node number of a dead end; volume of a dead end
             END DO
C
C Air chamber data
C
             do i=1,nairchambs
               read(1,*) naircham(i)           !Node number of air chamber(s)
               read(1,*) Haco(i),Vaco(i)
             end do
C
 1200    RETURN
         END
```

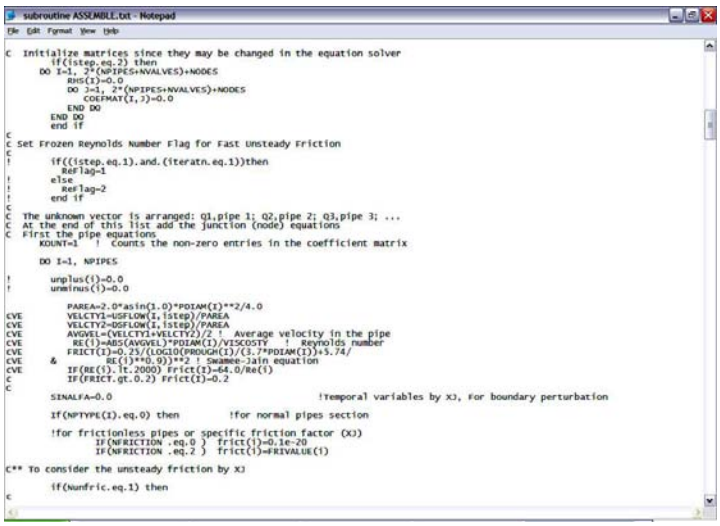
Subroutine ASSEMBLE



```
subroutine ASSEMBLE.txt - Notepad
File  Edit  Format  View  Help
C  Assembly of the simultaneous equation for the forward solution

C  Notes (May, 1999)
C   Friction equations uses Swamee-Jain. Better to use Swamee-Jain as
C   a first approximation to Colebrook, then one iteration with Colebrook
C   should result in an accurate friction factor.
C   Linear timeline interpolation is in this routine.  If cubic timeline interpolation is
C   to be used, it should interpolate for the flow rate at the foot of the
C   characteristic.
C
C Valve function is added in by XJ Wang.
C
C Unsteady friction and viscoelastic routines added by M Stephens
C Improved sparse matrix solver and air pocket routines by M Stephens
C Efficient recursive routines for unsteady friction and viscoelasticity by M Stephens
C
      SUBROUTINE ASSEMBLE(LIMPIPES,LIMNODES,LIMSTEPS,NPIPES,NODES,
     1  NDUPSTRM,NDONSTRM,PLENGTH,PROUGH,PDIAM,PWALL,PELAST,
     2  USFLOW,DSFLOW,RE,HEADPR,NCONDTN,HEADDC,DEMAND,ELEV,FRIVALUE,
     3    NMBRTIME,BNDTIME,LIMBNDCND,USRMINCR,CRMMIN,G,WAVESPD,VISCOSTY,
     4    NS,DT,COEFMAT,LIMCON,NODEPIP1,NODEPIP2,HEAD,CDAD,RHS,TIME,
     5    ITERATN,COEF,N,NZ,IRN,ICN,ISTEP,NSPARSE,NVALVES,NVALVETM,
     6  TAUTM,TAULIST,LIMVALVES,NFRICTION,NUNFRIC,NPTYPE,NMODEL,unkf,
     7  NDEADENDS,NDEND,VDEND,nairchambs,nairchan,nair,Hatmosp,
     8  Haco,Vaco)

      IMPLICIT double precision (A-H,O-Z)
        CHARACTER (LEN=15) STRING
        DIMENSION NDUPSTRM(LIMPIPES),NDONSTRM(LIMPIPES),PLENGTH(LIMPIPES),
     1  PROUGH(LIMPIPES),PDIAM(LIMPIPES),PWALL(LIMPIPES),
     2        PELAST(LIMPIPES),USFLOW(LIMPIPES,LIMSTEPS),
     &  DSFLOW(LIMPIPES,LIMSTEPS),ELEV(LIMNODES),
     3        NCONDTN(LIMNODES),HEADDC(LIMNODES,LIMBNDCND),
     3    DEMAND(LIMNODES,LIMBNDCND),NMBRTIME(LIMNODES),
     4    BNDTIME(LIMNODES,LIMBNDCND),WAVESPD(LIMPIPES),
     5    NS(LIMPIPES),COEFMAT(2*LIMPIPES+LIMNODES,2*LIMPIPES+LIMNODES),
     6    HEAD(LIMNODES),CDAD(LIMNODES,LIMBNDCND),NODPRINT(LIMNODES),
     7    NODEPIP2(LIMNODES,LIMCON),RHS(2*LIMPIPES+LIMNODES),
     9        NVALVETM(LIMVALVES),NODEPIP1(LIMNODES,LIMCON),
     A  TAUTM(LIMVALVES,LIMBNDCND),TAULIST(LIMVALVES,LIMBNDCND),
     B  UNK(LIMPIPES),NPTYPE(LIMPIPES),HEADPR(LIMNODES),unkf(limpipes),
     C  NDEND(LIMNODES),VDEND(LIMNODES),
     D    unplus(limpipes),unminus(limpipes),RE(limpipes),
     E  Veplus(limpipes),Veminus(limpipes),
     F  hback1(limnodes),hback2(limnodes),
     G  FRIVALUE(LIMPIPES),frict(limpipes),
     H  nairchan(limnodes),Haco(limnodes),Vaco(limnodes),
     I  headx(limnodes)

        double precision K1,K2,K3,K4,nair
```

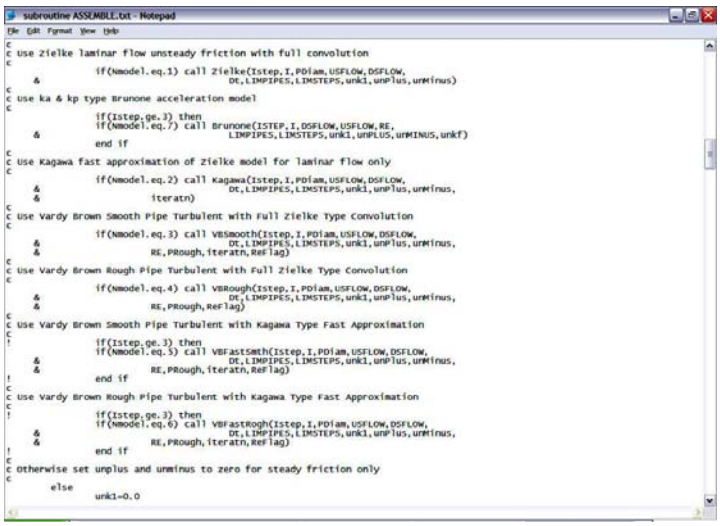


```
subroutine ASSEMBLE.txt - Notepad
File  Edit  Format  View  Help
      INTEGER*2 ICN(2*LIMPIPES+LIMCON*LIMNODES),
     1   IRN(2*LIMPIPES+LIMCON*LIMNODES)
        INTEGER*2 N,NZ

        integer ReFlag
CVE
        integer viscoflag
CVE
      DIMENSION COEF(4*LIMPIPES+LIMCON*LIMNODES)
        Pai=2.0*asin(1.0)
CVE
        viscoflag=2

CVE
C  Set previous variables equal to unknowns at first iteration
C  The Flow is saved in the time series, XJ
        If(Iteratn .eq.1) then
        Do I=1,Npipes
                USFLOW(I,ISTEP)=USFLOW(I,ISTEP-1)
                DSFLOW(I,ISTEP)=DSFLOW(I,ISTEP-1)
        end do
      DO I=1, NODES
CVE
CVE**************************************************
CVE
                  if(istep.eq.2)then
                        hback1(I)=headpr(I)
                        hback2(I)=headpr(I)
                  end if
CVE
                  if(istep.ge.3)then
                        hback2(I)=headpr(I)
                        hback1(I)=head(I)
                  end if
CVE
!     write(2,*) hback1(I),hback2(I)
!CVE   if(I.eq.43)then
!CVE     WRITE(3,'(1PF25.17)')TIME,(HEAD(43)
!CVE    &                          -elev(43)),
!CVE    &                  (hback1(43)-elev(43)),
!CVE    &                  (hback2(43)-elev(43))
!CVE    end if
CVE
CVE**************************************************
CVE
            HEADPR(I)=HEAD(I)
      END DO
        END IF
```

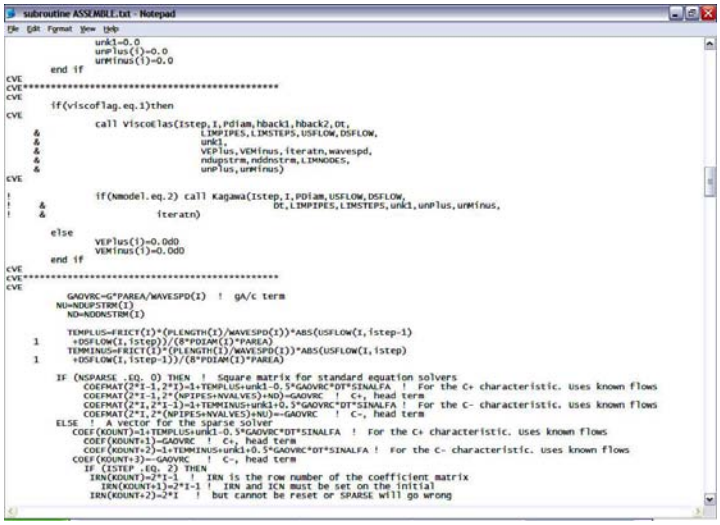


```
subroutine ASSEMBLE.txt - Notepad
File  Edit  Format  View  Help
C  Initialize matrices since they may be changed in the equation solver
        if(istep.eq.2) then
      DO I=1, 2*(NPIPES+NVALVES)+NODES
            RHS(I)=0.0
            DO J=1, 2*(NPIPES+NVALVES)+NODES
                COEFMAT(I,J)=0.0
            END DO
        END DO
        end if
C
C Set Frozen Reynolds Number Flag for Fast Unsteady Friction
C
!       if((istep.eq.1).and.(iteratn.eq.1))then
!         ReFlag=1
!       else
!         ReFlag=2
!       end if
C
C  The unknown vector is arranged: Q1,pipe 1; Q2,pipe 2; Q3,pipe 3; ...
C  At the end of this list add the junction (node) equations
C  First the pipe equations
        KOUNT=1   !  Counts the non-zero entries in the coefficient matrix

      DO I=1, NPIPES

!       unplus(I)=0.0
!       unminus(I)=0.0

          PAREA=2.0*asin(1.0)*PDIAM(I)**2/4.0
CVE       VELCTY1=USFLOW(I,istep)/PAREA
CVE       VELCTY2=DSFLOW(I,istep)/PAREA
CVE       AVGVEL=(VELCTY1+VELCTY2)/2 !  Average velocity in the pipe
CVE         RE(I)=ABS(AVGVEL)*PDIAM(I)/VISCOSTY   !  Reynolds number
CVE       FRICT(I)=0.25/(LOG10(PROUGH(I)/(3.7*PDIAM(I))+5.74/
CVE     &         RE(I)**0.9))**2 ! Swamee-Jain equation
CVE       IF(RE(I).lt.2000) frict(I)=64.0/Re(I)
C         IF(FRICT.gt.0.2) Frict(I)=0.2
C
        SINALFA=0.0                                     !Temporal variables by XJ, For boundary perturbation

        If(NPTYPE(I).eq.0) then         !for normal pipes section

        !for frictionless pipes or specific friction factor (XJ)
              IF(NFRICTION .eq.0 )  frict(I)=0.1e-20
              IF(NFRICTION .eq.2 )  frict(I)=FRIVALUE(I)

C** To consider the unsteady friction by XJ
        If(Nunfric.eq.1) then
C
```

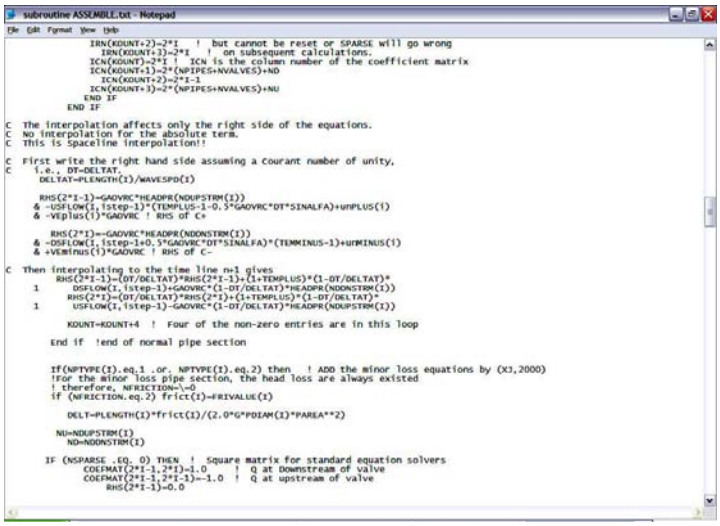
Appendix V – Fortran code for NETTRANS and NLFIT subroutines



```
subroutine ASSEMBLE.txt - Notepad
File  Edit  Format  View  Help
C
C Use Zielke laminar flow unsteady friction with full convolution
C
                if(Nmodel.eq.1) call Zielke(Istep,I,PDiam,USFLOW,DSFLOW,
     &                                      Dt,LIMPIPES,LIMSTEPS,unk1,unPlus,unMinus)
C
C Use ka & kp type Brunone acceleration model
C
                if(Istep.ge.3) then
                if(Nmodel.eq.7) call Brunone(ISTEP,I,DSFLOW,USFLOW,RE,
     &                                      LIMPIPES,LIMSTEPS,unk1,unPLUS,unMINUS,unkf)
                end if
C
C Use Kagawa fast approximation of Zielke model for laminar flow only
C
                if(Nmodel.eq.2) call Kagawa(Istep,I,PDiam,USFLOW,DSFLOW,
     &                                      Dt,LIMPIPES,LIMSTEPS,unk1,unPlus,unMinus,
     &                    iteratn)
C
C Use Vardy Brown Smooth Pipe Turbulent with Full Zielke Type Convolution
C
                if(Nmodel.eq.3) call VBSmooth(Istep,I,PDiam,USFLOW,DSFLOW,
     &                                      Dt,LIMPIPES,LIMSTEPS,unk1,unPlus,unMinus,
     &                    RE,PRough,iteratn,ReFlag)
C
C Use Vardy Brown Rough Pipe Turbulent with Full Zielke Type Convolution
C
                if(Nmodel.eq.4) call VBRough(Istep,I,PDiam,USFLOW,DSFLOW,
     &                                      Dt,LIMPIPES,LIMSTEPS,unk1,unPlus,unMinus,
     &                    RE,PRough,ReFlag)
C
C Use Vardy Brown Smooth Pipe Turbulent with Kagawa Type Fast Approximation
C
!               if(Istep.ge.3) then
!               if(Nmodel.eq.5) call VBFastSmth(Istep,I,PDiam,USFLOW,DSFLOW,
!    &                                      Dt,LIMPIPES,LIMSTEPS,unk1,unPlus,unMinus,
!    &                    RE,PRough,iteratn,ReFlag)
!               end if
C
C Use Vardy Brown Rough Pipe Turbulent with Kagawa Type Fast Approximation
C
!               if(Istep.ge.3) then
!               if(Nmodel.eq.6) call VBFastRogh(Istep,I,PDiam,USFLOW,DSFLOW,
!    &                                      Dt,LIMPIPES,LIMSTEPS,unk1,unPlus,unMinus,
!    &                    RE,PRough,iteratn,ReFlag)
!               end if
C
C Otherwise set unplus and unminus to zero for steady friction only
C
        else
                unk1=0.0
```

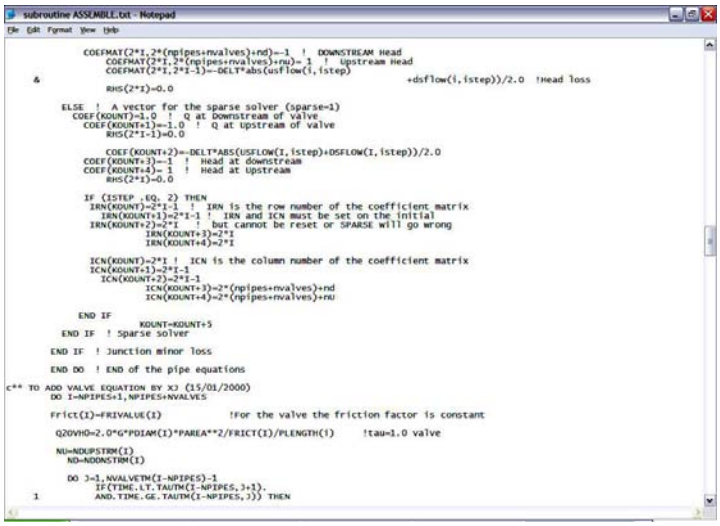


```
subroutine ASSEMBLE.txt - Notepad
File  Edit  Format  View  Help
                unk1=0.0
                unPlus(I)=0.0
                unMinus(I)=0.0
        end if
CVE
CVE***********************************************
CVE
        if(viscoflag.eq.1)then
CVE
                call ViscoClas(Istep,I,PDiam,hback1,hback2,Dt,
     &                          LIMPIPES,LIMSTEPS,USFLOW,DSFLOW,
     &                          unk1,
     &                          VEPlus,VEMinus,iteratn,wavespd,
     &                          ndupstrm,nddnstrm,LIMNODES,
     &                          unPlus,unMinus)
CVE
!               if(Nmodel.eq.2) call Kagawa(Istep,I,PDiam,USFLOW,DSFLOW,
!    &                                      Dt,LIMPIPES,LIMSTEPS,unk1,unPlus,unMinus,
!    &                    iteratn)
        else
                VEPlus(I)=0.0d0
                VEMinus(I)=0.0d0
        end if
CVE
CVE***********************************************
CVE
          GAOVRC=G*PAREA/WAVESPD(I)  !  gA/c term
         NU=NDUPSTRM(I)
           ND=NDDNSTRM(I)

          TEMPLUS=FRICT(I)*(PLENGTH(I)/WAVESPD(I))*ABS(USFLOW(I,istep-1)
     1     +DSFLOW(I,istep))/(8*PDIAM(I)*PAREA)
          TEMMINUS=FRICT(I)*(PLENGTH(I)/WAVESPD(I))*ABS(USFLOW(I,istep)
     1     +DSFLOW(I,istep-1))/(8*PDIAM(I)*PAREA)

        IF (NSPARSE .EQ. 0) THEN  !  Square matrix for standard equation solvers
           COEFMAT(2*I-1,2*I)=1+TEMPLUS+unk1-0.5*GAOVRC*DT*SINALFA  !  For the C+ characteristic. Uses known flows
           COEFMAT(2*I-1,2*(NPIPES+NVALVES)+ND)=GAOVRC  !  C+, head term
           COEFMAT(2*I,2*I-1)=1+TEMMINUS+unk1+0.5*GAOVRC*DT*SINALFA !  For the C- characteristic. Uses known flows
           COEFMAT(2*I,2*(NPIPES+NVALVES)+NU)=-GAOVRC   !  C-, head term
        ELSE  !  A vector for the sparse solver
           COEF(KOUNT)=1+TEMPLUS+unk1-0.5*GAOVRC*DT*SINALFA  !  For the C+ characteristic. Uses known flows
           COEF(KOUNT+1)=-GAOVRC  !  C+, head term
           COEF(KOUNT+2)=1+TEMMINUS+unk1+0.5*GAOVRC*DT*SINALFA !  For the C- characteristic. uses known flows
          COEF(KOUNT+3)=-GAOVRC   !  C-, head term
           IF (ISTEP .EQ. 2) THEN
            IRN(KOUNT)=2*I-1  !  IRN is the row number of the coefficient matrix
             IRN(KOUNT+1)=2*I-1 !  IRN and ICN must be set on the initial
            IRN(KOUNT+2)=2*I   !  but cannot be reset or SPARSE will go wrong
```

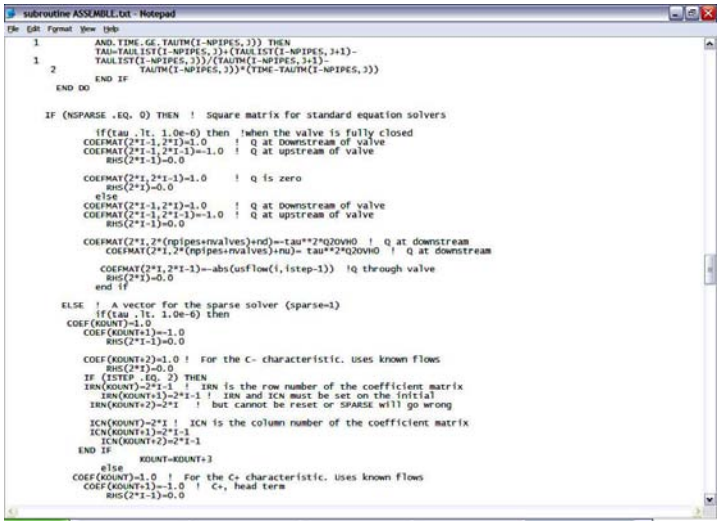


```
subroutine ASSEMBLE.txt - Notepad
File  Edit  Format  View  Help
            IRN(KOUNT+2)=2*I   !  but cannot be reset or SPARSE will go wrong
             IRN(KOUNT+3)=2*I   !  on subsequent calculations.
            ICN(KOUNT)=2*I !  ICN is the column number of the coefficient matrix
            ICN(KOUNT+1)=2*(NPIPES+NVALVES)+ND
             ICN(KOUNT+2)=2*I-1
            ICN(KOUNT+3)=2*(NPIPES+NVALVES)+NU
           END IF
         END IF

C  The interpolation affects only the right side of the equations.
C  No interpolation for the absolute term.
C  This is Spaceline interpolation!!

C  First write the right hand side assuming a Courant number of unity,
C    i.e., DT=DELTAT.
      DELTAT=PLENGTH(I)/WAVESPD(I)

      RHS(2*I-1)=GAOVRC*HEADPR(NDUPSTRM(I))
     & -USFLOW(I,istep-1)*(TEMPLUS-1-0.5*GAOVRC*DT*SINALFA)+unPLUS(I)
     & -VEplus(I)*GAOVRC ! RHS of C+

        RHS(2*I)=-GAOVRC*HEADPR(NDDNSTRM(I))
     & -DSFLOW(I,istep-1+0.5*GAOVRC*DT*SINALFA)*(TEMMINUS-1)+unMINUS(I)
     & +VEminus(I)*GAOVRC ! RHS of C-

C  Then interpolating to the time line n+1 gives
        RHS(2*I-1)=(DT/DELTAT)*RHS(2*I-1)+(1+TEMPLUS)*(1-DT/DELTAT)*
     1     DSFLOW(I,istep-1)+GAOVRC*(1-DT/DELTAT)*HEADPR(NDDNSTRM(I))
        RHS(2*I)=(DT/DELTAT)*RHS(2*I)+(1+TEMPLUS)*(1-DT/DELTAT)*
     1     USFLOW(I,istep-1)-GAOVRC*(1-DT/DELTAT)*HEADPR(NDUPSTRM(I))

        KOUNT=KOUNT+4  !  Four of the non-zero entries are in this loop

      End if  !end of normal pipe section


      If(NPTYPE(I).eq.1 .or. NPTYPE(I).eq.2) then   ! ADD the minor loss equations by (XJ,2000)
      !For the minor loss pipe section, the head loss are always existed
      ! therefore, NFRICTION=\=0
      if (NFRICTION.eq.2) frict(I)=FRIVALUE(I)

         DELT=PLENGTH(I)*frict(I)/(2.0*G*PDIAM(I)*PAREA**2)

       NU=NDUPSTRM(I)
         ND=NDDNSTRM(I)

      IF (NSPARSE .EQ. 0) THEN  !  Square matrix for standard equation solvers
            COEFMAT(2*I-1,2*I)=1.0     !  Q at Downstream of valve
            COEFMAT(2*I-1,2*I-1)=-1.0  !  Q at upstream of valve
               RHS(2*I-1)=0.0
```

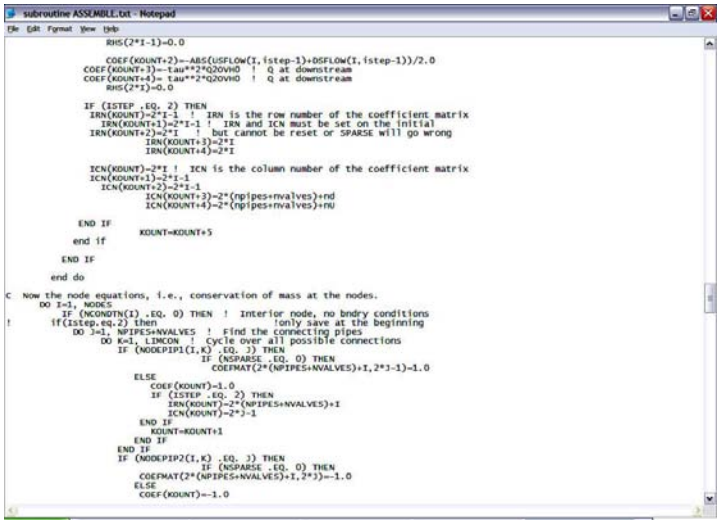
Appendix V – Fortran code for NETTRANS and NLFIT subroutines



```
subroutine ASSEMBLE.txt - Notepad
File  Edit  Format  View  Help

              COEFMAT(2*I,2*(npipes+nvalves)+nd)=-1  !  DOWNSTREAM Head
                  COEFMAT(2*I,2*(npipes+nvalves)+nu)= 1  !  Upstream Head
                  COEFMAT(2*I,2*I-1)=-DELT*abs(usflow(i,istep)
     &                                                    +dsflow(i,istep))/2.0  !Head loss
                  RHS(2*I)=0.0

            ELSE  !  A vector for the sparse solver (sparse=1)
              COEF(KOUNT)=1.0  !  Q at Downstream of valve
                COEF(KOUNT+1)=-1.0  !  Q at Upstream of valve
                  RHS(2*I-1)=0.0

                  COEF(KOUNT+2)=-DELT*ABS(USFLOW(I,istep)+DSFLOW(I,istep))/2.0
                COEF(KOUNT+3)=-1  !  Head at downstream
                COEF(KOUNT+4)= 1  !  Head at Upstream
                  RHS(2*I)=0.0

                IF (ISTEP .EQ. 2) THEN
                 IRN(KOUNT)=2*I-1  !  IRN is the row number of the coefficient matrix
                  IRN(KOUNT+1)=2*I-1 !  IRN and ICN must be set on the initial
                 IRN(KOUNT+2)=2*I   !  but cannot be reset or SPARSE will go wrong
                        IRN(KOUNT+3)=2*I
                        IRN(KOUNT+4)=2*I

                 ICN(KOUNT)=2*I !  ICN is the column number of the coefficient matrix
                 ICN(KOUNT+1)=2*I-1
                   ICN(KOUNT+2)=2*I-1
                        ICN(KOUNT+3)=2*(npipes+nvalves)+nd
                        ICN(KOUNT+4)=2*(npipes+nvalves)+nu

                END IF
                        KOUNT=KOUNT+5
              END IF  ! Sparse solver

            END IF  ! Junction minor loss

            END DO  ! END of the pipe equations

c** TO ADD VALVE EQUATION BY XJ (15/01/2000)
             DO I=NPIPES+1,NPIPES+NVALVES

             Frict(I)=FRIVALUE(I)            !For the valve the friction factor is constant

              Q2OVHO=2.0*G*PDIAM(I)*PAREA**2/FRICT(I)/PLENGTH(I)     !tau=1.0 valve

             NU=NDUPSTRM(I)
               ND=NDDNSTRM(I)

               DO J=1,NVALVETM(I-NPIPES)-1
                 IF(TIME.LT.TAUTM(I-NPIPES,J+1).
     1           AND.TIME.GE.TAUTM(I-NPIPES,J)) THEN
```

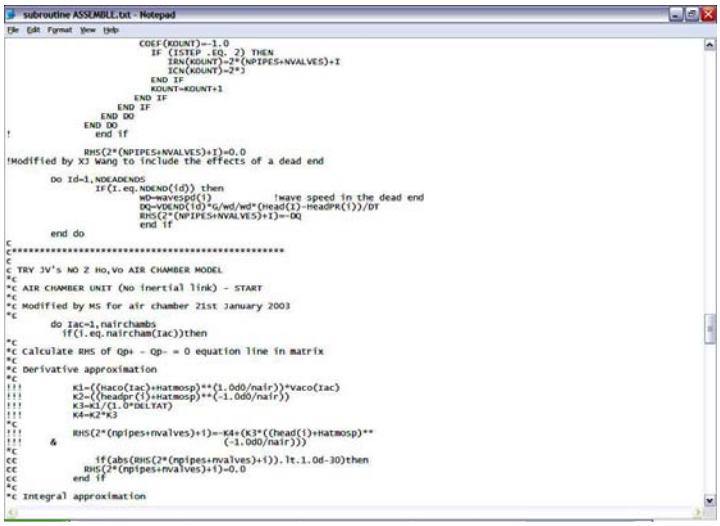


```
subroutine ASSEMBLE.txt - Notepad
File  Edit  Format  View  Help
     1           AND.TIME.GE.TAUTM(I-NPIPES,J)) THEN
                 TAU=TAULIST(I-NPIPES,J)+(TAULIST(I-NPIPES,J+1)-
     1           TAULIST(I-NPIPES,J))/(TAUTM(I-NPIPES,J+1)-
     2                 TAUTM(I-NPIPES,J))*(TIME-TAUTM(I-NPIPES,J))
                 END IF
              END DO


            IF (NSPARSE .EQ. 0) THEN  !  Square matrix for standard equation solvers

                 if(tau .lt. 1.0e-6) then  !when the valve is fully closed
               COEFMAT(2*I-1,2*I)=1.0     !  Q at Downstream of valve
               COEFMAT(2*I-1,2*I-1)=-1.0  !  Q at upstream of valve
                  RHS(2*I-1)=0.0

               COEFMAT(2*I,2*I-1)=1.0     !  Q is zero
                  RHS(2*I)=0.0
                 else
               COEFMAT(2*I-1,2*I)=1.0     !  Q at Downstream of valve
               COEFMAT(2*I-1,2*I-1)=-1.0  !  Q at upstream of valve
                  RHS(2*I-1)=0.0

               COEFMAT(2*I,2*(npipes+nvalves)+nd)=-tau**2*Q2OVHO  !  Q at downstream
                  COEFMAT(2*I,2*(npipes+nvalves)+nu)= tau**2*Q2OVHO  !  Q at downstream

                  COEFMAT(2*I,2*I-1)=-abs(usflow(i,istep-1))  !Q through valve
                  RHS(2*I)=0.0
                 end if

            ELSE  !  A vector for the sparse solver (sparse=1)
                 if(tau .lt. 1.0e-6) then
              COEF(KOUNT)=1.0
                 COEF(KOUNT+1)=-1.0
                   RHS(2*I-1)=0.0

                 COEF(KOUNT+2)=1.0 !  For the C- characteristic. Uses known flows
                   RHS(2*I)=0.0
                 IF (ISTEP .EQ. 2) THEN
                 IRN(KOUNT)=2*I-1  !  IRN is the row number of the coefficient matrix
                   IRN(KOUNT+1)=2*I-1 !  IRN and ICN must be set on the initial
                  IRN(KOUNT+2)=2*I   !  but cannot be reset or SPARSE will go wrong

                  ICN(KOUNT)=2*I !  ICN is the column number of the coefficient matrix
                  ICN(KOUNT+1)=2*I-1
                    ICN(KOUNT+2)=2*I-1
               END IF
                        KOUNT=KOUNT+3
                   else
              COEF(KOUNT)=1.0  !  For the C+ characteristic. Uses known flows
                COEF(KOUNT+1)=-1.0  !  C+, head term
                  RHS(2*I-1)=0.0
```

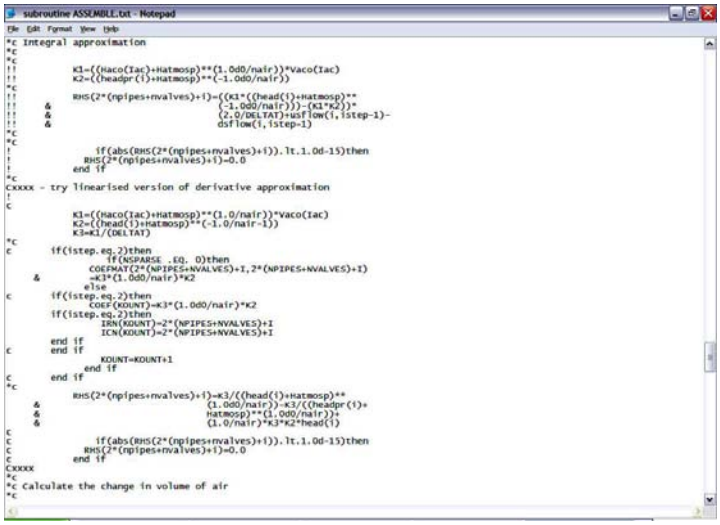


```
subroutine ASSEMBLE.txt - Notepad
File  Edit  Format  View  Help
                  RHS(2*I-1)=0.0

                  COEF(KOUNT+2)=-ABS(USFLOW(I,istep-1)+DSFLOW(I,istep-1))/2.0
                COEF(KOUNT+3)=-tau**2*Q2OVHO  !  Q at downstream
                COEF(KOUNT+4)= tau**2*Q2OVHO  !  Q at downstream
                  RHS(2*I)=0.0

                IF (ISTEP .EQ. 2) THEN
                 IRN(KOUNT)=2*I-1  !  IRN is the row number of the coefficient matrix
                  IRN(KOUNT+1)=2*I-1 !  IRN and ICN must be set on the initial
                 IRN(KOUNT+2)=2*I   !  but cannot be reset or SPARSE will go wrong
                        IRN(KOUNT+3)=2*I
                        IRN(KOUNT+4)=2*I

                 ICN(KOUNT)=2*I !  ICN is the column number of the coefficient matrix
                 ICN(KOUNT+1)=2*I-1
                   ICN(KOUNT+2)=2*I-1
                        ICN(KOUNT+3)=2*(npipes+nvalves)+nd
                        ICN(KOUNT+4)=2*(npipes+nvalves)+nu

                END IF
                        KOUNT=KOUNT+5
               end if

              END IF

             end do

C  Now the node equations, i.e., conservation of mass at the nodes.
       DO I=1, NODES
          IF (NCONDTN(I) .EQ. 0) THEN  !  Interior node, no bndry conditions
!         if(istep.eq.2) then              !only save at the beginning
            DO J=1, NPIPES+NVALVES  !  Find the connecting pipes
              DO K=1, LIMCON  !  Cycle over all possible connections
                IF (NODEPIP1(I,K) .EQ. J) THEN
                                  IF (NSPARSE .EQ. 0) THEN
                                    COEFMAT(2*(NPIPES+NVALVES)+I,2*J-1)=1.0
                  ELSE
                    COEF(KOUNT)=1.0
                    IF (ISTEP .EQ. 2) THEN
                      IRN(KOUNT)=2*(NPIPES+NVALVES)+I
                      ICN(KOUNT)=2*J-1
                    END IF
                    KOUNT=KOUNT+1
                  END IF
                END IF
                IF (NODEPIP2(I,K) .EQ. J) THEN
                                  IF (NSPARSE .EQ. 0) THEN
                    COEFMAT(2*(NPIPES+NVALVES)+I,2*J)=-1.0
                  ELSE
                    COEF(KOUNT)=-1.0
```

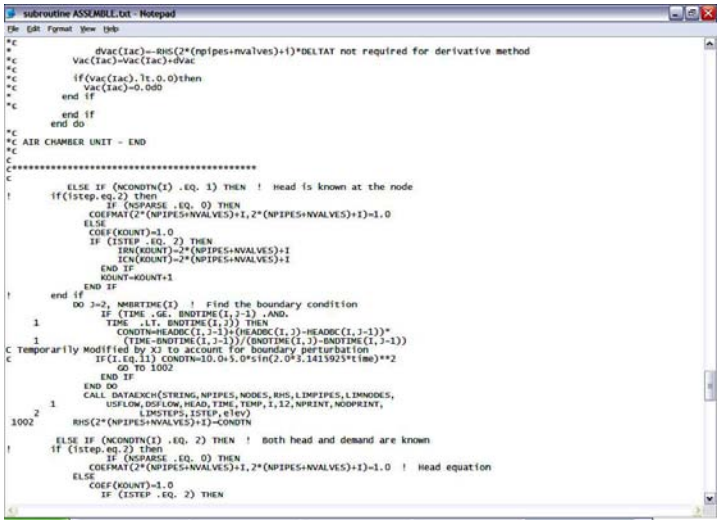
Appendix V – Fortran code for NETTRANS and NLFIT subroutines



```
subroutine ASSEMBLE.txt - Notepad
File  Edit  Format  View  Help
                        COEF(KOUNT)=-1.0
                          IF (ISTEP .EQ. 2) THEN
                            IRN(KOUNT)=2*(NPIPES+NVALVES)+I
                            ICN(KOUNT)=2*3
                          END IF
                          KOUNT=KOUNT+1
                        END IF
                      END IF
                    END DO
                  END DO
!                   end if

                  RHS(2*(NPIPES+NVALVES)+I)=0.0
!Modified by XJ Wang to include the effects of a dead end

        Do Id=1,NDEADENDS
                IF(I.eq.NDEND(Id)) then
                        WD=WAVESPD(I)           !wave speed in the dead end
                        DQ=VDEND(Id)*G/WD/WD*(Head(I)-HeadPR(I))/DT
                        RHS(2*(NPIPES+NVALVES)+I)=-DQ
                        end if
        end do
C
C*************************************************
C TRY JV's NO Z Ho,Vo AIR CHAMBER MODEL
*C
*C AIR CHAMBER UNIT (No inertial link) - START
*C
*C Modified by MS for air chamber 21st January 2003
*C
        do Iac=1,nairchambs
          if(I.eq.nairchan(Iac))then
*C
*C Calculate RHS of Qp+ - Qp- = 0 equation line in matrix
*C
*C Derivative approximation
*C
!!!           K1=((Haco(Iac)+Hatmosp)**(1.0d0/nair))*Vaco(Iac)
!!!           K2=((headpr(I)+Hatmosp)**(-1.0d0/nair))
!!!           K3=K1/(1.0*DELTAT)
!!!           K4=K2*K3
*C
!!!           RHS(2*(npipes+nvalves)+I)=-K4+(K3*((head(I)+Hatmosp)**
!!!     &                                    (-1.0d0/nair)))
*C
CC              if(abs(RHS(2*(npipes+nvalves)+I)).lt.1.0d-30)then
CC            RHS(2*(npipes+nvalves)+I)=0.0
CC            end if
*C
*C Integral approximation
```

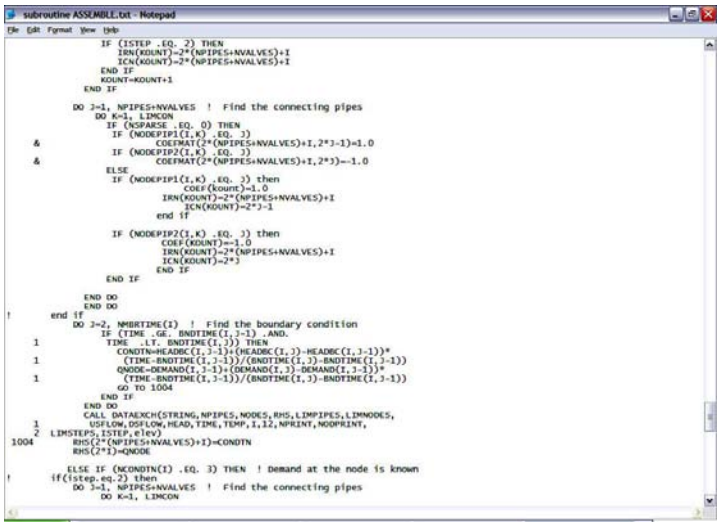


```
subroutine ASSEMBLE.txt - Notepad
File  Edit  Format  View  Help
*C Integral approximation
*C
*C
!!            K1=((Haco(Iac)+Hatmosp)**(1.0d0/nair))*Vaco(Iac)
!!            K2=((headpr(I)+Hatmosp)**(-1.0d0/nair))
*C
!!            RHS(2*(npipes+nvalves)+I)=((K1*((head(I)+Hatmosp)**
!!     &                                 (-1.0d0/nair)))-(K3*K2))*
!!     &                                 (2.0/DELTAT)+usflow(I,istep-1)-
!!     &                                 dsflow(I,istep-1)
*C
*C
!               if(abs(RHS(2*(npipes+nvalves)+I)).lt.1.0d-15)then
!             RHS(2*(npipes+nvalves)+I)=0.0
!             end if
*C
Cxxxx - try linearised version of derivative approximation
!
C
              K1=((Haco(Iac)+Hatmosp)**(1.0/nair))*Vaco(Iac)
              K2=((head(I)+Hatmosp)**(-1.0/nair-1))
              K3=K1/(DELTAT)
*C
C       if(istep.eq.2)then
                  if(NSPARSE .EQ. 0)then
                COEFMAT(2*(NPIPES+NVALVES)+I,2*(NPIPES+NVALVES)+I)
     &          =K3*(1.0d0/nair)*K2
                else
C       if(istep.eq.2)then
                  COEF(KOUNT)=K3*(1.0d0/nair)*K2
        if(istep.eq.2)then
                  IRN(KOUNT)=2*(NPIPES+NVALVES)+I
                  ICN(KOUNT)=2*(NPIPES+NVALVES)+I
        end if
C       end if
                  KOUNT=KOUNT+1
                end if
C       end if
*C
              RHS(2*(npipes+nvalves)+I)=K3/((head(I)+Hatmosp)**
     &                                 (1.0d0/nair))-K3/((headpr(I)+
     &                                 Hatmosp)**(1.0d0/nair))+
     &                                 (1.0/nair)*K3*K2*head(I)
C
                if(abs(RHS(2*(npipes+nvalves)+I)).lt.1.0d-15)then
C             RHS(2*(npipes+nvalves)+I)=0.0
C             end if
CXXXX
*C
*C Calculate the change in volume of air
*C
```

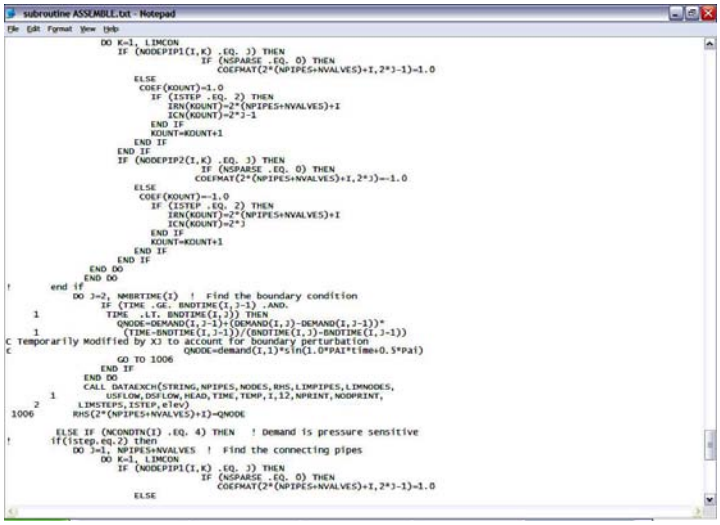


```
subroutine ASSEMBLE.txt - Notepad
File  Edit  Format  View  Help
*C
*               dVac(Iac)=-RHS(2*(npipes+nvalves)+I)*DELTAT not required for derivative method
*           Vac(Iac)=Vac(Iac)+dVac
*C
*C          if(Vac(Iac).lt.0.0)then
*C            Vac(Iac)=0.0d0
*         end if
*C
          end if
        end do
*C
*C AIR CHAMBER UNIT - END
*C
C
C*************************************************
C
            ELSE IF (NCONDTN(I) .EQ. 1) THEN  !  Head is known at the node
!       if(istep.eq.2) then
                IF (NSPARSE .EQ. 0) THEN
              COEFMAT(2*(NPIPES+NVALVES)+I,2*(NPIPES+NVALVES)+I)=1.0
              ELSE
               COEF(KOUNT)=1.0
               IF (ISTEP .EQ. 2) THEN
                  IRN(KOUNT)=2*(NPIPES+NVALVES)+I
                  ICN(KOUNT)=2*(NPIPES+NVALVES)+I
                END IF
                KOUNT=KOUNT+1
              END IF
!       end if
        DO J=2, NMBRTIME(I)  !  Find the boundary condition
             IF (TIME .GE. BNDTIME(I,J-1) .AND.
     1         TIME  .LT. BNDTIME(I,J)) THEN
               CONDTN=HEADBC(I,J-1)+(HEADBC(I,J)-HEADBC(I,J-1))*
     1          (TIME-BNDTIME(I,J-1))/(BNDTIME(I,J)-BNDTIME(I,J-1))
C Temporarily Modified by XJ to account for boundary perturbation
C             IF(I.EQ.11) CONDTN=10.0+5.0*sin(2.0*3.1415925*time)**2
                GO TO 1002
              END IF
            END DO
            CALL DATAEXCH(STRING,NPIPES,NODES,RHS,LIMPIPES,LIMNODES,
     1         USFLOW,DSFLOW,HEAD,TIME,TEMP,I,12,NPRINT,NODPRINT,
     2              LIMSTEPS,ISTEP,elev)
 1002       RHS(2*(NPIPES+NVALVES)+I)=CONDTN

         ELSE IF (NCONDTN(I) .EQ. 2) THEN  !  Both head and demand are known
!       if (istep.eq.2) then
                IF (NSPARSE .EQ. 0) THEN
              COEFMAT(2*(NPIPES+NVALVES)+I,2*(NPIPES+NVALVES)+I)=1.0  !  Head equation
            ELSE
               COEF(KOUNT)=1.0
                IF (ISTEP .EQ. 2) THEN
```

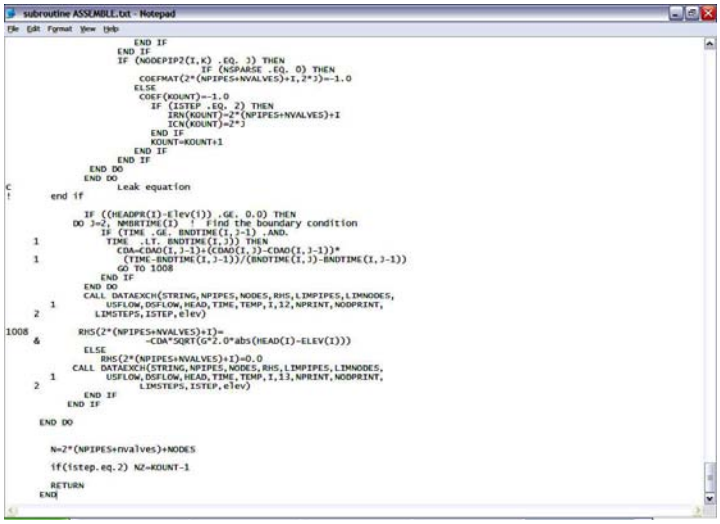
Appendix V – Fortran code for NETTRANS and NLFIT subroutines




```
                IF (ISTEP .EQ. 2) THEN
                   IRN(KOUNT)=2*(NPIPES+NVALVES)+I
                   ICN(KOUNT)=2*(NPIPES+NVALVES)+I
                END IF
                KOUNT=KOUNT+1
             END IF

          DO J=1, NPIPES+NVALVES  !  Find the connecting pipes
             DO K=1, LIMCON
               IF (NSPARSE .EQ. 0) THEN
                 IF (NODEPIP1(I,K) .EQ. J)
     &              COEFMAT(2*(NPIPES+NVALVES)+I,2*J-1)=1.0
                 IF (NODEPIP2(I,K) .EQ. J)
     &              COEFMAT(2*(NPIPES+NVALVES)+I,2*J)=-1.0
               ELSE
                IF (NODEPIP1(I,K) .EQ. J) then
                        COEF(kount)=1.0
                     IRN(KOUNT)=2*(NPIPES+NVALVES)+I
                        ICN(KOUNT)=2*J-1
                    end if

                IF (NODEPIP2(I,K) .EQ. J) then
                     COEF(KOUNT)=-1.0
                     IRN(KOUNT)=2*(NPIPES+NVALVES)+I
                     ICN(KOUNT)=2*J
                    END IF
                END IF

             END DO
             END DO
!       end if
          DO J=2, NMBRTIME(I)  !  Find the boundary condition
             IF (TIME .GE. BNDTIME(I,J-1) .AND.
     1          TIME  .LT. BNDTIME(I,J)) THEN
               CONDTN=HEADBC(I,J-1)+(HEADBC(I,J)-HEADBC(I,J-1))*
     1           (TIME-BNDTIME(I,J-1))/(BNDTIME(I,J)-BNDTIME(I,J-1))
               QNODE=DEMAND(I,J-1)+(DEMAND(I,J)-DEMAND(I,J-1))*
     1           (TIME-BNDTIME(I,J-1))/(BNDTIME(I,J)-BNDTIME(I,J-1))
               GO TO 1004
             END IF
          END DO
          CALL DATAEXCH(STRING,NPIPES,NODES,RHS,LIMPIPES,LIMNODES,
     1       USFLOW,DSFLOW,HEAD,TIME,TEMP,I,12,NPRINT,NODPRINT,
     2  LIMSTEPS,ISTEP,elev)
 1004     RHS(2*(NPIPES+NVALVES)+I)=CONDTN
          RHS(2*I)=QNODE

        ELSE IF (NCONDTN(I) .EQ. 3) THEN  ! Demand at the node is known
!       if(istep.eq.2) then
          DO J=1, NPIPES+NVALVES  !  Find the connecting pipes
             DO K=1, LIMCON
```

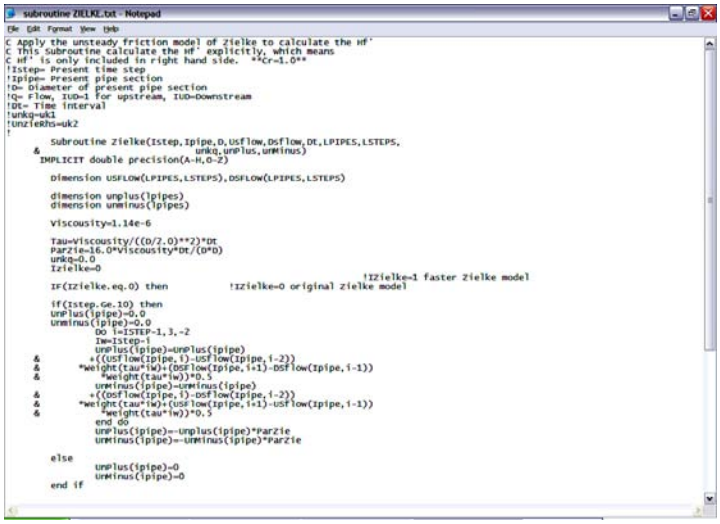



```
             DO K=1, LIMCON
                IF (NODEPIP1(I,K) .EQ. J) THEN
                                IF (NSPARSE .EQ. 0) THEN
                                   COEFMAT(2*(NPIPES+NVALVES)+I,2*J-1)=1.0
                   ELSE
                     COEF(KOUNT)=1.0
                       IF (ISTEP .EQ. 2) THEN
                         IRN(KOUNT)=2*(NPIPES+NVALVES)+I
                         ICN(KOUNT)=2*J-1
                       END IF
                       KOUNT=KOUNT+1
                   END IF
                END IF
                IF (NODEPIP2(I,K) .EQ. J) THEN
                                IF (NSPARSE .EQ. 0) THEN
                                COEFMAT(2*(NPIPES+NVALVES)+I,2*J)=-1.0
                   ELSE
                    COEF(KOUNT)=-1.0
                      IF (ISTEP .EQ. 2) THEN
                        IRN(KOUNT)=2*(NPIPES+NVALVES)+I
                        ICN(KOUNT)=2*J
                      END IF
                      KOUNT=KOUNT+1
                   END IF
                END IF
             END DO
             END DO
!       end if
          DO J=2, NMBRTIME(I)  !  Find the boundary condition
             IF (TIME .GE. BNDTIME(I,J-1) .AND.
     1          TIME  .LT. BNDTIME(I,J)) THEN
               QNODE=DEMAND(I,J-1)+(DEMAND(I,J)-DEMAND(I,J-1))*
     1           (TIME-BNDTIME(I,J-1))/(BNDTIME(I,J)-BNDTIME(I,J-1))
C Temporarily Modified by XJ to account for boundary perturbation
C                              QNODE=demand(I,1)*sin(1.0*PAI*time+0.5*Pai)
               GO TO 1006
             END IF
          END DO
          CALL DATAEXCH(STRING,NPIPES,NODES,RHS,LIMPIPES,LIMNODES,
     1       USFLOW,DSFLOW,HEAD,TIME,TEMP,I,12,NPRINT,NODPRINT,
     2       LIMSTEPS,ISTEP,elev)
 1006     RHS(2*(NPIPES+NVALVES)+I)=QNODE

        ELSE IF (NCONDTN(I) .EQ. 4) THEN   ! Demand is pressure sensitive
!       if(istep.eq.2) then
          DO J=1, NPIPES+NVALVES  !  Find the connecting pipes
             DO K=1, LIMCON
                IF (NODEPIP1(I,K) .EQ. J) THEN
                                IF (NSPARSE .EQ. 0) THEN
                                   COEFMAT(2*(NPIPES+NVALVES)+I,2*J-1)=1.0
                   ELSE
```

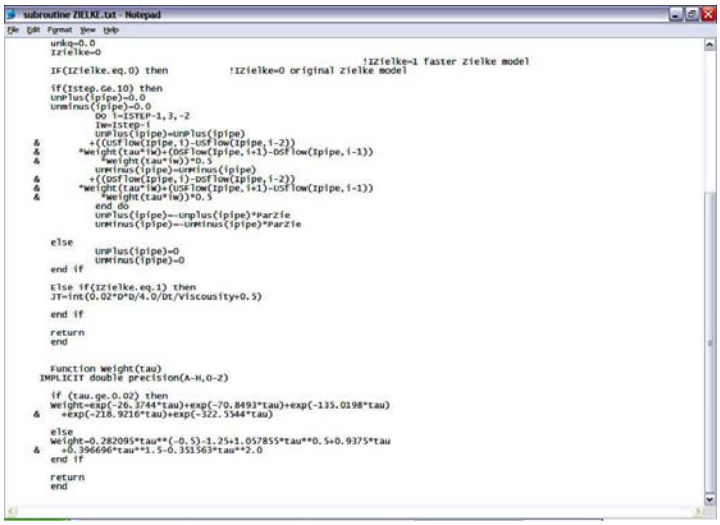



```
                END IF
             END IF
             IF (NODEPIP2(I,K) .EQ. J) THEN
                             IF (NSPARSE .EQ. 0) THEN
                COEFMAT(2*(NPIPES+NVALVES)+I,2*J)=-1.0
               ELSE
                COEF(KOUNT)=-1.0
                  IF (ISTEP .EQ. 2) THEN
                    IRN(KOUNT)=2*(NPIPES+NVALVES)+I
                    ICN(KOUNT)=2*J
                  END IF
                  KOUNT=KOUNT+1
               END IF
             END IF
          END DO
          END DO
C               Leak equation
!       end if

          IF ((HEADPR(I)-Elev(I)) .GE. 0.0) THEN
          DO J=2, NMBRTIME(I)  !  Find the boundary condition
             IF (TIME .GE. BNDTIME(I,J-1) .AND.
     1          TIME  .LT. BNDTIME(I,J)) THEN
               CDA=CDAO(I,J-1)+(CDAO(I,J)-CDAO(I,J-1))*
     1           (TIME-BNDTIME(I,J-1))/(BNDTIME(I,J)-BNDTIME(I,J-1))
               GO TO 1008
             END IF
          END DO
          CALL DATAEXCH(STRING,NPIPES,NODES,RHS,LIMPIPES,LIMNODES,
     1       USFLOW,DSFLOW,HEAD,TIME,TEMP,I,12,NPRINT,NODPRINT,
     2       LIMSTEPS,ISTEP,elev)

1008      RHS(2*(NPIPES+NVALVES)+I)=
     &              -CDA*SQRT(G*2.0*abs(HEAD(I)-ELEV(I)))
          ELSE
             RHS(2*(NPIPES+NVALVES)+I)=0.0
          CALL DATAEXCH(STRING,NPIPES,NODES,RHS,LIMPIPES,LIMNODES,
     1       USFLOW,DSFLOW,HEAD,TIME,TEMP,I,13,NPRINT,NODPRINT,
     2              LIMSTEPS,ISTEP,elev)
          END IF
        END IF

      END DO


        N=2*(NPIPES+nvalves)+NODES

        if(istep.eq.2) NZ=KOUNT-1

       RETURN
      END
```

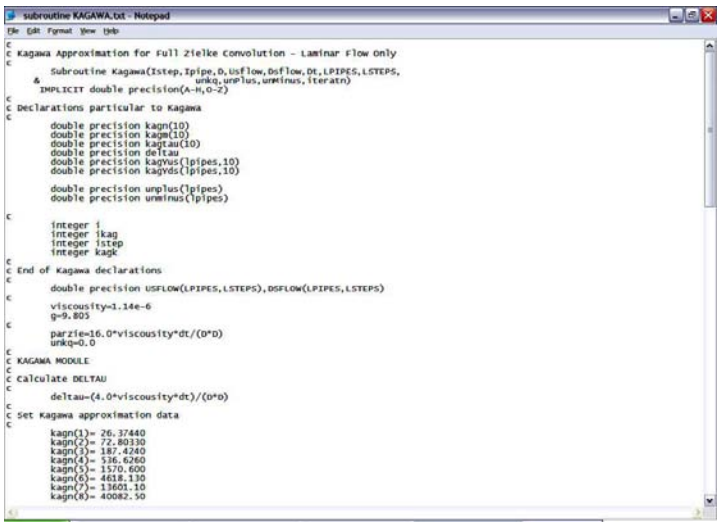
Subroutines for unsteady friction




```
C Apply the unsteady friction model of Zielke to calculate the Hf'
C This Subroutine calculate the Hf' explicitly, which means
C Hf' is only included in right hand side.  **Cr=1.0**
!Istep= Present time step
!Ipipe= Present pipe section
!D= Diameter of present pipe section
!Q= Flow, IUD=1 for upstream, IUD=Downstream
!Dt= Time interval
!unkq=uk1
!UnzielRhs=uk2
!
      Subroutine Zielke(Istep,Ipipe,D,Usflow,Dsflow,Dt,LPIPES,LSTEPS,
     &                  unkq,unPlus,unMinus)
      IMPLICIT double precision(A-H,O-Z)

        Dimension USFLOW(LPIPES,LSTEPS),DSFLOW(LPIPES,LSTEPS)

        dimension unplus(lpipes)
        dimension unminus(lpipes)

        Viscousity=1.14e-6

        Tau=Viscousity/((D/2.0)**2)*Dt
        Parzie=16.0*Viscousity*Dt/(D*D)
        unkq=0.0
        Izielke=0
                                                    !Izielke=1 faster Zielke model
        IF(Izielke.eq.0) then           !Izielke=0 original zielke model

        if(Istep.Ge.10) then
        unPlus(ipipe)=0.0
        unminus(ipipe)=0.0
                Do I=ISTEP-1,3,-2
                Iw=Istep-i
                unPlus(ipipe)=unPlus(ipipe)
     &         +((Usflow(Ipipe,i)-Usflow(Ipipe,i-2))
     &       *Weight(tau*iw)+(Dsflow(Ipipe,i+1)-Dsflow(Ipipe,i-1))
     &          *Weight(tau*iw))*0.5
                unminus(ipipe)=unminus(ipipe)
     &         +((Dsflow(Ipipe,i)-Dsflow(Ipipe,i-2))
     &       *Weight(tau*iw)+(Usflow(Ipipe,i+1)-Usflow(Ipipe,i-1))
     &          *Weight(tau*iw))*0.5
                end do
                unPlus(ipipe)=-unplus(ipipe)*Parzie
                unminus(ipipe)=-unMinus(ipipe)*Parzie

        else
                unPlus(ipipe)=0
                unMinus(ipipe)=0
        end if
```

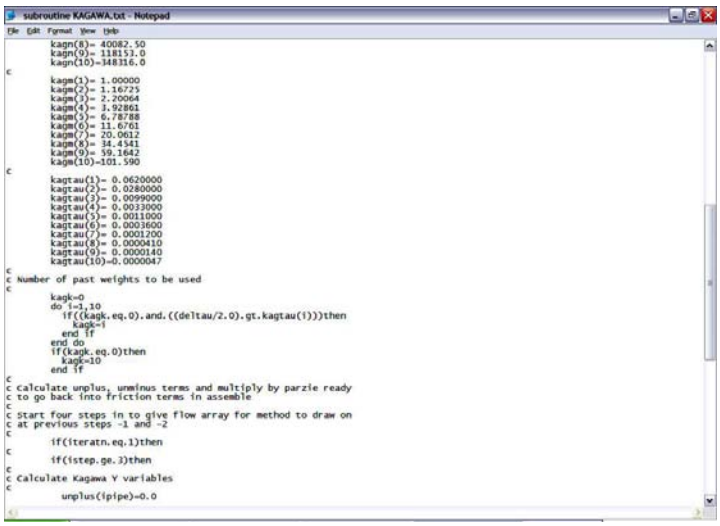



```
        unkq=0.0
        Izielke=0
                                                    !Izielke=1 faster Zielke model
        IF(Izielke.eq.0) then           !Izielke=0 original zielke model

        if(Istep.Ge.10) then
        unPlus(ipipe)=0.0
        unminus(ipipe)=0.0
                Do I=ISTEP-1,3,-2
                Iw=Istep-i
                unPlus(ipipe)=unPlus(ipipe)
     &         +((Usflow(Ipipe,i)-Usflow(Ipipe,i-2))
     &       *Weight(tau*iw)+(Dsflow(Ipipe,i+1)-Dsflow(Ipipe,i-1))
     &          *Weight(tau*iw))*0.5
                unminus(ipipe)=unminus(ipipe)
     &         +((Dsflow(Ipipe,i)-Dsflow(Ipipe,i-2))
     &       *Weight(tau*iw)+(Usflow(Ipipe,i+1)-Usflow(Ipipe,i-1))
     &          *Weight(tau*iw))*0.5
                end do
                unPlus(ipipe)=-unplus(ipipe)*Parzie
                unminus(ipipe)=-unMinus(ipipe)*Parzie

        else
                unPlus(ipipe)=0
                unMinus(ipipe)=0
        end if

        Else if(Izielke.eq.1) then
        JT=int(0.02*D*D/4.0/Dt/Viscousity+0.5)

        end if

        return
        end


        Function Weight(tau)
      IMPLICIT double precision(A-H,O-Z)

        if (tau.ge.0.02) then
        Weight=exp(-26.3744*tau)+exp(-70.8493*tau)+exp(-135.0198*tau)
     &     +exp(-218.9216*tau)+exp(-322.5544*tau)

        else
        Weight=0.282095*tau**(-0.5)-1.25+1.057855*tau**0.5+0.9375*tau
     &     +0.396696*tau**1.5-0.351563*tau**2.0
        end if

        return
        end
```

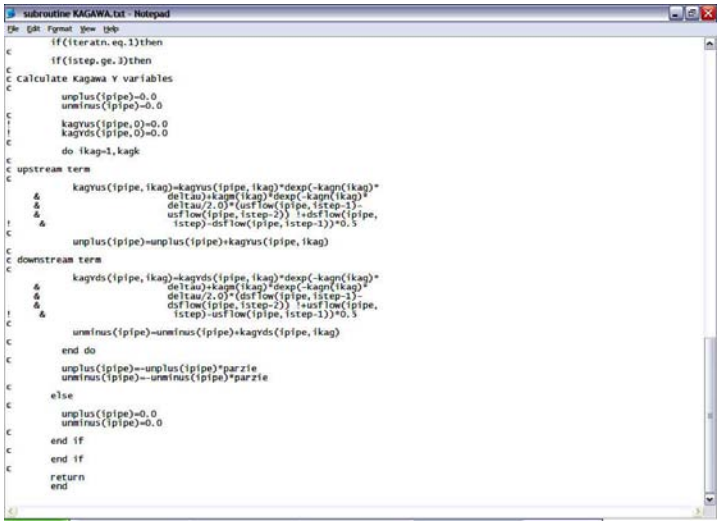



```
C
C Kagawa Approximation for Full Zielke Convolution - Laminar Flow Only
C
      Subroutine Kagawa(Istep,Ipipe,D,Usflow,Dsflow,Dt,LPIPES,LSTEPS,
     &                  unkq,unPlus,unMinus,iteratn)
      IMPLICIT double precision(A-H,O-Z)
C
C Declarations particular to Kagawa
C
      double precision kagn(10)
      double precision kagm(10)
      double precision kagtau(10)
      double precision deltau
      double precision kagvus(lpipes,10)
      double precision kagvds(lpipes,10)

      double precision unplus(lpipes)
      double precision unminus(lpipes)

C
      integer i
      integer ikag
      integer istep
      integer kagk
C
C End of Kagawa declarations
C
      double precision USFLOW(LPIPES,LSTEPS),DSFLOW(LPIPES,LSTEPS)
C
      viscousity=1.14e-6
      g=9.805
C
      parzie=16.0*viscousity*dt/(D*D)
      unkq=0.0
C
C KAGAWA MODULE
C
C Calculate DELTAU
C
      deltau=(4.0*viscousity*dt)/(D*D)
C
C Set Kagawa approximation data
C
      kagn(1)= 26.37440
      kagn(2)= 72.80330
      kagn(3)= 187.4240
      kagn(4)= 536.6260
      kagn(5)= 1570.600
      kagn(6)= 4618.130
      kagn(7)= 13601.10
      kagn(8)= 40082.50
```

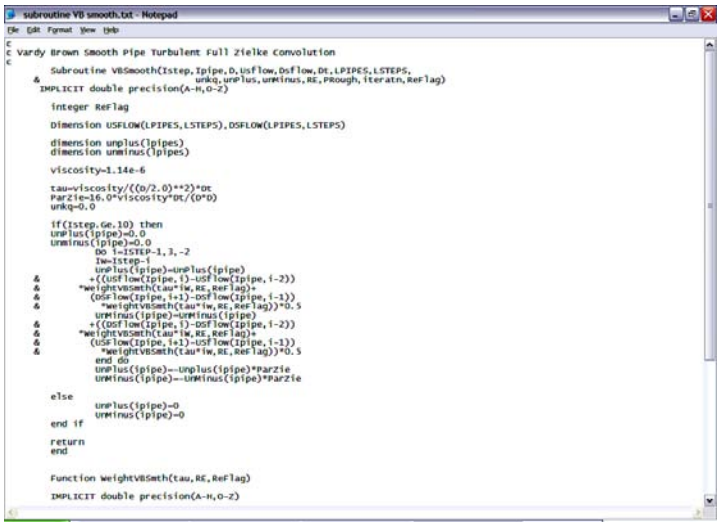
Appendix V – Fortran code for NETTRANS and NLFIT subroutines



```
subroutine KAGAWA.txt - Notepad
File  Edit  Format  View  Help
      kagn(8)= 40082.50
      kagn(9)= 118153.0
      kagn(10)=348316.0
C
      kagm(1)= 1.00000
      kagm(2)= 1.16725
      kagm(3)= 2.20064
      kagm(4)= 3.92861
      kagm(5)= 6.78788
      kagm(6)= 11.6761
      kagm(7)= 20.0612
      kagm(8)= 34.4541
      kagm(9)= 59.1642
      kagm(10)=101.590
C
      kagtau(1)= 0.0620000
      kagtau(2)= 0.0280000
      kagtau(3)= 0.0099000
      kagtau(4)= 0.0033000
      kagtau(5)= 0.0011000
      kagtau(6)= 0.0003600
      kagtau(7)= 0.0001200
      kagtau(8)= 0.0000410
      kagtau(9)= 0.0000140
      kagtau(10)=0.0000047
C
C Number of past weights to be used
C
      kagk=0
      do i=1,10
        if((kagk.eq.0).and.((deltau/2.0).gt.kagtau(i)))then
          kagk=i
        end if
      end do
      if(kagk.eq.0)then
        kagk=10
      end if
C
C Calculate unplus, unminus terms and multiply by parzie ready
C to go back into friction terms in assemble
C
C Start four steps in to give flow array for method to draw on
C at previous steps -1 and -2
C
      if(iteratn.eq.1)then
C
      if(istep.ge.3)then
C
C Calculate Kagawa Y variables
C
        unplus(ipipe)=0.0
```



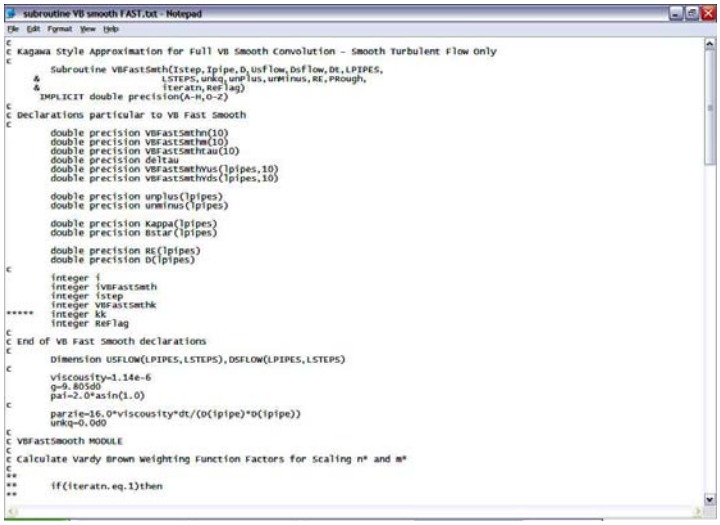
```
subroutine KAGAWA.txt - Notepad
File  Edit  Format  View  Help
      if(iteratn.eq.1)then
C
      if(istep.ge.3)then
C
C Calculate Kagawa Y variables
C
        unplus(ipipe)=0.0
        unminus(ipipe)=0.0
C
!       kagyus(ipipe,0)=0.0
!       kagyds(ipipe,0)=0.0
C
        do ikag=1,kagk
C
C upstream term
C
          kagyus(ipipe,ikag)=kagyus(ipipe,ikag)*dexp(-kagn(ikag)*
     &                    deltau)+kagm(ikag)*dexp(-kagn(ikag)*
     &                    deltau/2.0)*(usflow(ipipe,istep-1)-
     &                    usflow(ipipe,istep-2)) !+dsflow(ipipe,
!    &                     istep)-dsflow(ipipe,istep-1))*0.5
C
          unplus(ipipe)=unplus(ipipe)+kagyus(ipipe,ikag)
C
C downstream term
C
          kagyds(ipipe,ikag)=kagyds(ipipe,ikag)*dexp(-kagn(ikag)*
     &                    deltau)+kagm(ikag)*dexp(-kagn(ikag)*
     &                    deltau/2.0)*(dsflow(ipipe,istep-1)-
     &                    dsflow(ipipe,istep-2)) !+usflow(ipipe,
!    &                     istep)-usflow(ipipe,istep-1))*0.5
C
          unminus(ipipe)=unminus(ipipe)+kagyds(ipipe,ikag)
C
        end do
C
        unplus(ipipe)=-unplus(ipipe)*parzie
        unminus(ipipe)=-unminus(ipipe)*parzie
C
      else
C
        unplus(ipipe)=0.0
        unminus(ipipe)=0.0
C
      end if
C
      end if
C
      return
      end
```



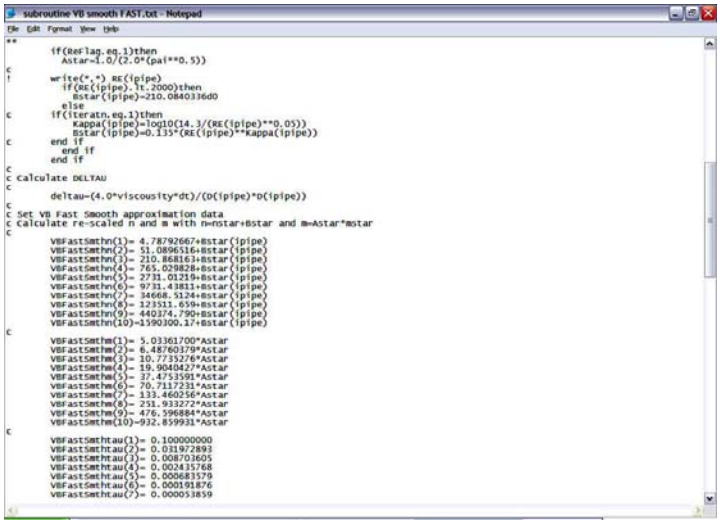
```
subroutine VB smooth.txt - Notepad
File  Edit  Format  View  Help
C
C Vardy Brown Smooth Pipe Turbulent Full Zielke Convolution
C
      Subroutine VBSmooth(Istep,Ipipe,D,Usflow,Dsflow,Dt,LPIPES,LSTEPS,
     &                    unkq,unPlus,unMinus,RE,PRough,iteratn,ReFlag)
      IMPLICIT double precision(A-H,O-Z)

      integer ReFlag

      Dimension USFLOW(LPIPES,LSTEPS),DSFLOW(LPIPES,LSTEPS)

      dimension unplus(lpipes)
      dimension unminus(lpipes)

      viscosity=1.14e-6

      tau=viscosity/((D/2.0)**2)*Dt
      Parzie=16.0*viscosity*Dt/(D*D)
      unkq=0.0

      if(Istep.Ge.10) then
      unPlus(ipipe)=0.0
      unminus(ipipe)=0.0
              DO i=ISTEP-1,3,-2
              IW=Istep-i
              unPlus(ipipe)=unPlus(ipipe)
     &       +((Usflow(Ipipe,i)-Usflow(Ipipe,i-2))
     &      *WeightVBSmth(tau*IW,RE,ReFlag)+
     &        (Dsflow(Ipipe,i+1)-Dsflow(Ipipe,i-1))
     &         *WeightVBSmth(tau*iw,RE,ReFlag))*0.5
              unMinus(ipipe)=unMinus(ipipe)
     &       +((Dsflow(Ipipe,i)-Dsflow(Ipipe,i-2))
     &      *WeightVBSmth(tau*IW,RE,ReFlag)+
     &        (Usflow(Ipipe,i+1)-Usflow(Ipipe,i-1))
     &         *WeightVBSmth(tau*iw,RE,ReFlag))*0.5
              end do
              unPlus(ipipe)=-Unplus(ipipe)*Parzie
              unMinus(ipipe)=-UnMinus(ipipe)*Parzie

      else
              unPlus(ipipe)=0
              unMinus(ipipe)=0
      end if

      return
      end


      Function WeightVBSmth(tau,RE,ReFlag)

      IMPLICIT double precision(A-H,O-Z)
```

Appendix V – Fortran code for NETTRANS and NLFIT subroutines

```
subroutine VB smooth.txt - Notepad
File  Edit  Format  View  Help
        unminus(ipipe)=0.0
                DO I=ISTEP-1,3,-2
                IW=Istep-I
                unPlus(ipipe)=unPlus(ipipe)
     &          +((usflow(ipipe,i)-usflow(ipipe,i-2))
     &       *weightVBSmth(tau*Iw,RE,ReFlag)+
     &         (dsFlow(ipipe,i+1)-dsflow(ipipe,i-1))
     &           *WeightVBSmth(tau*Iw,RE,ReFlag))*0.5
                unminus(ipipe)=unminus(ipipe)
     &          +((dsflow(ipipe,i)-dsflow(ipipe,i-2))
     &       *weightVBSmth(tau*Iw,RE,ReFlag)+
     &         (usFlow(ipipe,i+1)-usflow(ipipe,i-1))
     &           *WeightVBSmth(tau*Iw,RE,ReFlag))*0.5
                end do
                unPlus(ipipe)=-unplus(ipipe)*Parzie
                unminus(ipipe)=-unMinus(ipipe)*Parzie

        else
                unPlus(ipipe)=0
                unMinus(ipipe)=0
        end if

        return
        end


        Function WeightVBSmth(tau,RE,ReFlag)

        IMPLICIT double precision(A-H,O-Z)
        double precision Kappa
        integer ReFlag
C
        pai=2.0*asin(1.0)

        if(ReFlag.eq.1)then
          if(RE.lt.2000)then
            Cstar=0.00476
          else
!       if(iteratn.eq.1)then
            kappa=log10(14.3/(RE**0.05))
            Cstar=7.41/(RE**Kappa)
!       end if
          end if
        end if

        WeightVBSmth=(1.0/(2.0*((pai*tau)**0.5)))*
     &              exp(-tau/Cstar)

        return
        end
```

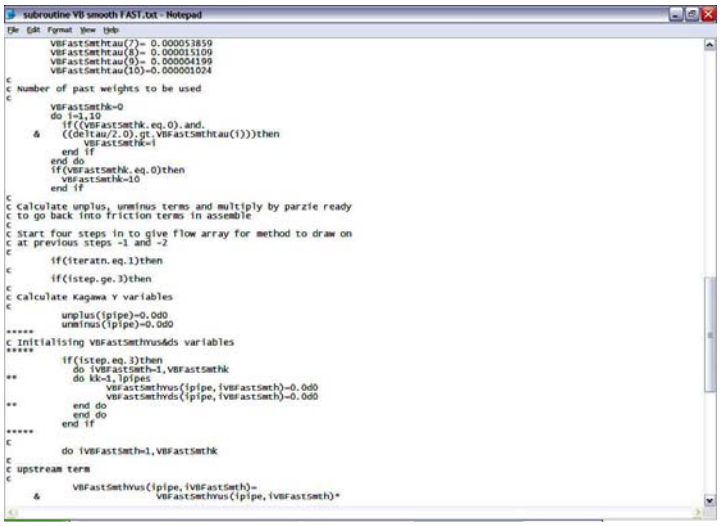


```
subroutine VB smooth FAST.txt - Notepad
File  Edit  Format  View  Help
C
C Kagawa Style Approximation for Full VB Smooth Convolution - Smooth Turbulent Flow Only
C
        Subroutine VBFastSmth(Istep,Ipipe,D,Usflow,Dsflow,Dt,LPIPES,
     &                        LSTEPS,unkq,unPlus,unMinus,RE,PRough,
     &                        Iteratn,ReFlag)
      IMPLICIT double precision(A-H,O-Z)
C
C Declarations particular to VB Fast Smooth
C
        double precision VBFastSmthn(10)
        double precision VBFastSmthm(10)
        double precision VBFastSmthtau(10)
        double precision deltau
        double precision VBFastSmthVus(lpipes,10)
        double precision VBFastSmthVds(lpipes,10)

        double precision unplus(lpipes)
        double precision unminus(lpipes)

        double precision Kappa(lpipes)
        double precision Bstar(lpipes)

        double precision RE(lpipes)
        double precision D(lpipes)
C
        integer I
        integer ivBFastSmth
        integer istep
        integer VBFastSmthk
*****   integer kk
        integer ReFlag
C
C End of VB Fast Smooth declarations
C
        Dimension USFLOW(LPIPES,LSTEPS),DSFLOW(LPIPES,LSTEPS)
C
        viscousity=1.14e-6
        g=9.805d0
        pai=2.0*asin(1.0)
C
        parzie=16.0*viscousity*dt/(D(ipipe)*D(ipipe))
        unkq=0.0d0
C
C VBFastSmooth MODULE
C
C Calculate Vardy Brown Weighting Function Factors for Scaling n* and m*
C
**
**      if(iteratn.eq.1)then
**
```

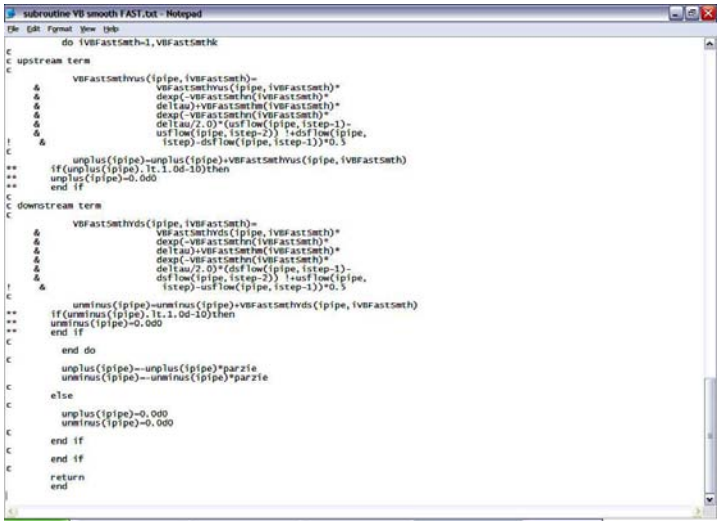


```
subroutine VB smooth FAST.txt - Notepad
File  Edit  Format  View  Help
**
        if(ReFlag.eq.1)then
          Astar=1.0/(2.0*(pai**0.5))
C
!       write(*,*) RE(ipipe)
          if(RE(ipipe).lt.2000)then
            Bstar(ipipe)=210.0840336d0
          else
C       if(iteratn.eq.1)then
            Kappa(ipipe)=log10(14.3/(RE(ipipe)**0.05))
            Bstar(ipipe)=0.135*(RE(ipipe)**Kappa(ipipe))
C       end if
          end if
        end if
C
C Calculate DELTAU
C
        deltau=(4.0*viscousity*dt)/(D(ipipe)*D(ipipe))
C
C Set VB Fast Smooth approximation data
C Calculate re-scaled n and m with n=nstar+Bstar and m=Astar*mstar
C
        VBFastSmthn(1)= 4.78792667+Bstar(ipipe)
        VBFastSmthn(2)= 51.0896516+Bstar(ipipe)
        VBFastSmthn(3)= 210.868163+Bstar(ipipe)
        VBFastSmthn(4)= 765.029828+Bstar(ipipe)
        VBFastSmthn(5)= 2731.01219+Bstar(ipipe)
        VBFastSmthn(6)= 9731.43811+Bstar(ipipe)
        VBFastSmthn(7)= 34668.5124+Bstar(ipipe)
        VBFastSmthn(8)= 123511.659+Bstar(ipipe)
        VBFastSmthn(9)= 440374.790+Bstar(ipipe)
        VBFastSmthn(10)=1590300.17+Bstar(ipipe)
C
        VBFastSmthm(1)= 5.03361700*Astar
        VBFastSmthm(2)= 8.48760379*Astar
        VBFastSmthm(3)= 10.7735276*Astar
        VBFastSmthm(4)= 19.9040427*Astar
        VBFastSmthm(5)= 37.4753591*Astar
        VBFastSmthm(6)= 70.7117231*Astar
        VBFastSmthm(7)= 133.460256*Astar
        VBFastSmthm(8)= 251.933272*Astar
        VBFastSmthm(9)= 476.596884*Astar
        VBFastSmthm(10)=932.859931*Astar
C
        VBFastSmthtau(1)= 0.100000000
        VBFastSmthtau(2)= 0.031972893
        VBFastSmthtau(3)= 0.008703605
        VBFastSmthtau(4)= 0.002435768
        VBFastSmthtau(5)= 0.000683579
        VBFastSmthtau(6)= 0.000191876
        VBFastSmthtau(7)= 0.000053859
```

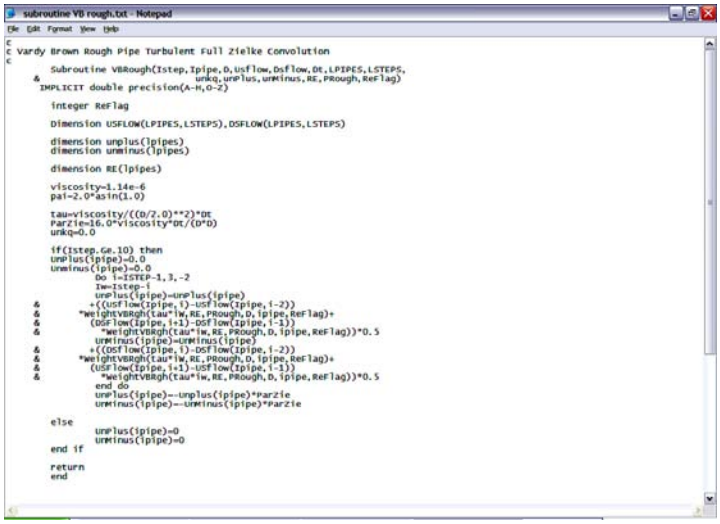
Appendix V – Fortran code for NETTRANS and NLFIT subroutines



```
subroutine VB smooth FAST.txt - Notepad
File Edit Format View Help
      VBFastSmthtau(7)= 0.000053859
      VBFastSmthtau(8)= 0.000015109
      VBFastSmthtau(9)= 0.000004199
      VBFastSmthtau(10)=0.000001024
C
C Number of past weights to be used
C
      VBFastSmthk=0
      do i=1,10
        if((VBFastSmthk.eq.0).and.
     &    ((deltau/2.0).gt.VBFastSmthtau(i)))then
            VBFastSmthk=i
        end if
      end do
      if(VBFastSmthk.eq.0)then
        VBFastSmthk=10
      end if
C
C calculate unplus, unminus terms and multiply by parzie ready
C to go back into friction terms in assemble
C
C Start four steps in to give flow array for method to draw on
C at previous steps -1 and -2
C
      if(iteratn.eq.1)then
C
      if(istep.ge.3)then
C
C calculate Kagawa Y variables
C
        unplus(ipipe)=0.0d0
        unminus(ipipe)=0.0d0
*****
C Initialising VBFastSmthvus&ds variables
*****
        if(istep.eq.3)then
          do iVBFastSmth=1,VBFastSmthk
**        do kk=1,lpipes
            VBFastSmthvus(ipipe,iVBFastSmth)=0.0d0
            VBFastSmthvds(ipipe,iVBFastSmth)=0.0d0
**        end do
          end do
        end if
*****
C
        do iVBFastSmth=1,VBFastSmthk
C
C upstream term
C
          VBFastSmthvus(ipipe,iVBFastSmth)=
     &                  VBFastSmthvus(ipipe,iVBFastSmth)*
```

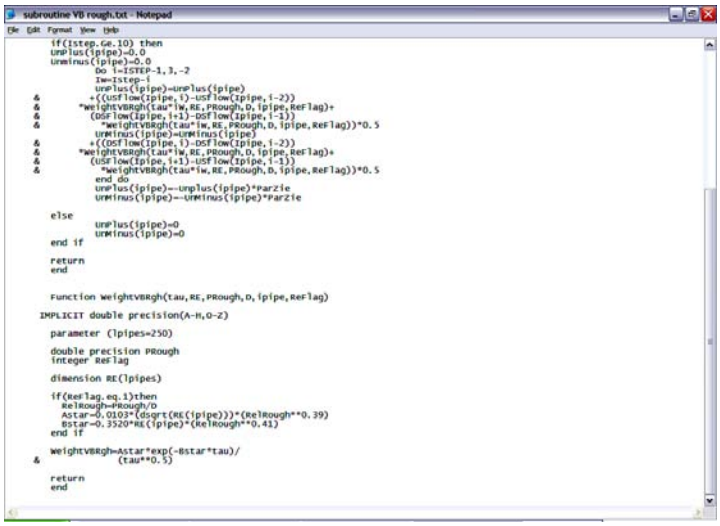


```
subroutine VB smooth FAST.txt - Notepad
File Edit Format View Help
        do iVBFastSmth=1,VBFastSmthk
C
C upstream term
C
          VBFastSmthvus(ipipe,iVBFastSmth)=
     &                  VBFastSmthvus(ipipe,iVBFastSmth)*
     &                  dexp(-VBFastSmthn(iVBFastSmth)*
     &                  deltau)+VBFastSmthm(iVBFastSmth)*
     &                  dexp(-VBFastSmthn(iVBFastSmth)*
     &                  deltau/2.0)*(usflow(ipipe,istep-1)-
     &                  usflow(ipipe,istep-2)) !+dsflow(ipipe,
!    &                   istep)-dsflow(ipipe,istep-1))*0.5
C
          unplus(ipipe)=unplus(ipipe)+VBFastSmthvus(ipipe,iVBFastSmth)
**        if(unplus(ipipe).lt.1.0d-10)then
**        unplus(ipipe)=0.0d0
**        end if
C
C downstream term
C
          VBFastSmthvds(ipipe,iVBFastSmth)=
     &                  VBFastSmthvds(ipipe,iVBFastSmth)*
     &                  dexp(-VBFastSmthn(iVBFastSmth)*
     &                  deltau)+VBFastSmthm(iVBFastSmth)*
     &                  dexp(-VBFastSmthn(iVBFastSmth)*
     &                  deltau/2.0)*(dsflow(ipipe,istep-1)-
     &                  dsflow(ipipe,istep-2)) !+usflow(ipipe,
!    &                   istep)-usflow(ipipe,istep-1))*0.5
C
          unminus(ipipe)=unminus(ipipe)+VBFastSmthvds(ipipe,iVBFastSmth)
**        if(unminus(ipipe).lt.1.0d-10)then
**        unminus(ipipe)=0.0d0
**        end if
C
          end do
C
          unplus(ipipe)=-unplus(ipipe)*parzie
          unminus(ipipe)=-unminus(ipipe)*parzie
C
        else
C
          unplus(ipipe)=0.0d0
          unminus(ipipe)=0.0d0
C
        end if
C
        end if
C
        return
        end
```

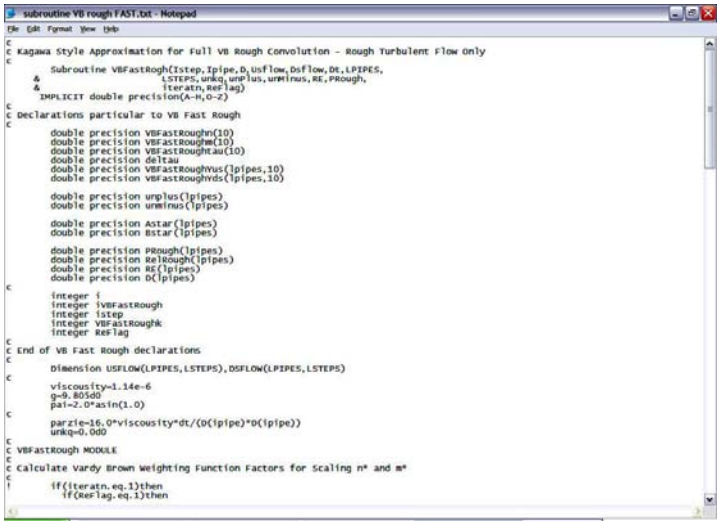


```
subroutine VB rough.txt - Notepad
File Edit Format View Help
C
C Vardy Brown Rough Pipe Turbulent Full Zielke Convolution
C
      Subroutine VBRough(Istep,Ipipe,D,Usflow,Dsflow,Dt,LPIPES,LSTEPS,
     &                   unkq,unPlus,unMinus,RE,PRough,ReFlag)
 IMPLICIT double precision(A-H,O-Z)

      integer ReFlag

      Dimension USFLOW(LPIPES,LSTEPS),DSFLOW(LPIPES,LSTEPS)

      dimension unplus(lpipes)
      dimension unminus(lpipes)

      dimension RE(lpipes)

      viscosity=1.14e-6
      pai=2.0*asin(1.0)

      tau=viscosity/((D/2.0)**2)*Dt
      Parzie=16.0*viscosity*Dt/(D*D)
      unkq=0.0

      if(Istep.Ge.10) then
      unPlus(ipipe)=0.0
      unminus(ipipe)=0.0
                Do i=ISTEP-1,3,-2
                iw=Istep-i
                unPlus(ipipe)=unPlus(ipipe)
     &          +((Usflow(ipipe,i)-Usflow(ipipe,i-2))
     &         *WeightVBRgh(tau*iw,RE,PRough,D,ipipe,ReFlag)+
     &           (Dsflow(ipipe,i+1)-Dsflow(ipipe,i-1))
     &            *WeightVBRgh(tau*iw,RE,PRough,D,ipipe,ReFlag))*0.5
                unMinus(ipipe)=unMinus(ipipe)
     &          +((Dsflow(ipipe,i)-Dsflow(ipipe,i-2))
     &         *WeightVBRgh(tau*iw,RE,PRough,D,ipipe,ReFlag)+
     &           (Usflow(ipipe,i+1)-Usflow(ipipe,i-1))
     &            *WeightVBRgh(tau*iw,RE,PRough,D,ipipe,ReFlag))*0.5
                end do
                unPlus(ipipe)=-unplus(ipipe)*Parzie
                unMinus(ipipe)=-unMinus(ipipe)*Parzie

      else
                unPlus(ipipe)=0
                unMinus(ipipe)=0
      end if

      return
      end
```

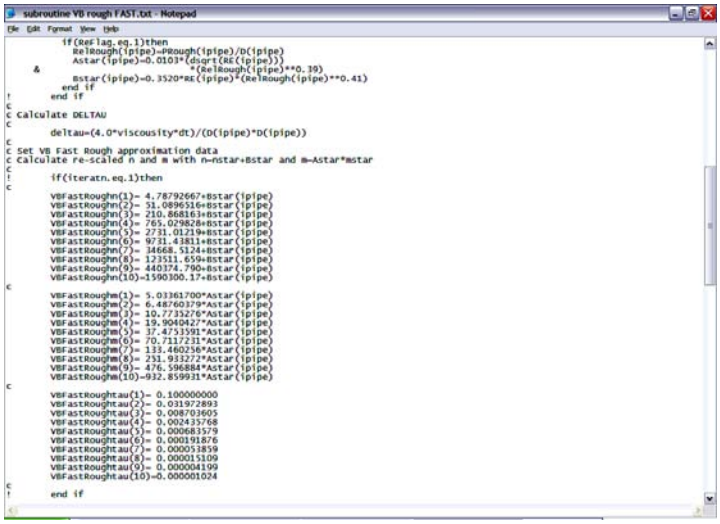
Appendix V – Fortran code for NETTRANS and NLFIT subroutines



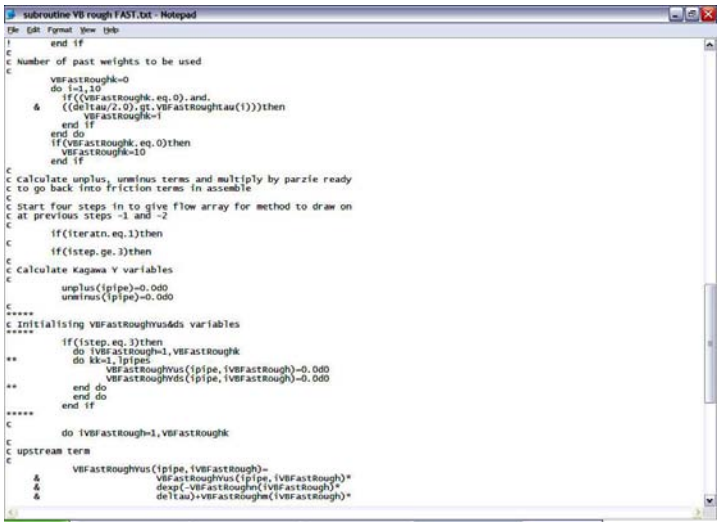
```
subroutine VB rough.txt - Notepad
File Edit Format View Help
        if(Istep.Ge.10) then
        UnPlus(Ipipe)=0.0
        UnMinus(Ipipe)=0.0
                Do I=ISTEP-1,3,-2
                Iw=Isteg-I
                UnPlus(Ipipe)=UnPlus(Ipipe)
     &        +((Usflow(Ipipe,I)-USFlow(Ipipe,I-2))
     &       *WeightVBRgh(tau*Iw,RE,PRough,D,Ipipe,ReFlag)+
     &         (DSFlow(Ipipe,I+1)-DSFlow(Ipipe,I-1))
     &           *WeightVBRgh(tau*Iw,RE,PRough,D,Ipipe,ReFlag))*0.5
                UnMinus(Ipipe)=UnMinus(Ipipe)
     &        +((DSFlow(Ipipe,I)-DSFlow(Ipipe,I-2))
     &       *WeightVBRgh(tau*Iw,RE,PRough,D,Ipipe,ReFlag)+
     &         (USFlow(Ipipe,I+1)-USFlow(Ipipe,I-1))
     &           *WeightVBRgh(tau*Iw,RE,PRough,D,Ipipe,ReFlag))*0.5
                end do
                UnPlus(Ipipe)=-UnPlus(Ipipe)*Parzie
                UnMinus(Ipipe)=-UnMinus(Ipipe)*Parzie

        else
                UnPlus(Ipipe)=0
                UnMinus(Ipipe)=0
        end if

        return
        end


        Function WeightVBRgh(tau,RE,PRough,D,Ipipe,ReFlag)

      IMPLICIT double precision(A-H,O-Z)

        parameter (Ipipes=250)

        double precision PRough
        integer ReFlag

        dimension RE(Ipipes)

        if(ReFlag.eq.1)then
          RelRough=PRough/D
          Astar=0.0103*(dsqrt(RE(Ipipe)))*(RelRough**0.39)
          Bstar=0.3520*RE(Ipipe)*(RelRough**0.41)
        end if

        WeightVBRgh=Astar*exp(-Bstar*tau)/
     &                (tau**0.5)

        return
        end
```



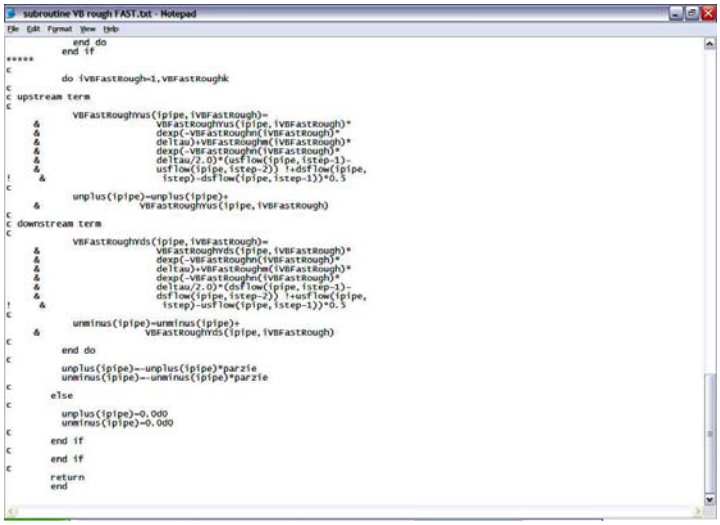
```
subroutine VB rough FAST.txt - Notepad
File Edit Format View Help
C
C Kagawa Style Approximation for Full VB Rough Convolution - Rough Turbulent Flow Only
C
        Subroutine VBFastRogh(Istep,Ipipe,D,Usflow,Dsflow,Dt,LPIPES,
     &                        LSTEPS,unkq,unPlus,unMinus,RE,PRough,
     &                        Iteratn,ReFlag)
      IMPLICIT double precision(A-H,O-Z)
C
C Declarations particular to VB Fast Rough
C
        double precision VBFastRoughn(10)
        double precision VBFastRoughm(10)
        double precision VBFastRoughtau(10)
        double precision deltau
        double precision VBFastRoughYus(lpipes,10)
        double precision VBFastRoughYds(lpipes,10)

        double precision unplus(lpipes)
        double precision unminus(lpipes)

        double precision Astar(lpipes)
        double precision Bstar(lpipes)

        double precision PRough(lpipes)
        double precision RelRough(lpipes)
        double precision RE(lpipes)
        double precision D(lpipes)
C
        integer I
        integer IVBFastRough
        integer Istep
        integer VBFastRoughk
        integer ReFlag
C
C End of VB Fast Rough declarations
C
        Dimension USFLOW(LPIPES,LSTEPS),DSFLOW(LPIPES,LSTEPS)
C
        viscousity=1.14e-6
        g=9.805d0
        pai=2.0*asin(1.0)
C
        parzie=16.0*viscousity*dt/(D(ipipe)*D(ipipe))
        unkq=0.0d0
C
C VBFastRough MODULE
C
C Calculate Vardy Brown Weighting Function Factors for Scaling n* and m*
C
!       if(iteratn.eq.1)then
          if(ReFlag.eq.1)then
```



```
subroutine VB rough FAST.txt - Notepad
File Edit Format View Help
            if(ReFlag.eq.1)then
              RelRough(ipipe)=PRough(ipipe)/D(ipipe)
              Astar(ipipe)=0.0103*(dsqrt(RE(ipipe)))
     &                      *(RelRough(ipipe)**0.39)
              Bstar(ipipe)=0.3520*RE(ipipe)*(RelRough(ipipe)**0.41)
            end if
!       end if
C
C Calculate DELTAU
C
        deltau=(4.0*viscousity*dt)/(D(ipipe)*D(ipipe))
C
C Set VB Fast Rough approximation data
C Calculate re-scaled n and m with n=nstar+Bstar and m=Astar*mstar
C
!       if(iteratn.eq.1)then
C
        VBFastRoughn(1)= 4.78792667+Bstar(ipipe)
        VBFastRoughn(2)= 51.0896516+Bstar(ipipe)
        VBFastRoughn(3)= 210.868163+Bstar(ipipe)
        VBFastRoughn(4)= 765.029828+Bstar(ipipe)
        VBFastRoughn(5)= 2731.01219+Bstar(ipipe)
        VBFastRoughn(6)= 9731.43811+Bstar(ipipe)
        VBFastRoughn(7)= 34668.5124+Bstar(ipipe)
        VBFastRoughn(8)= 123511.659+Bstar(ipipe)
        VBFastRoughn(9)= 440374.790+Bstar(ipipe)
        VBFastRoughn(10)=1590300.17+Bstar(ipipe)
C
        VBFastRoughm(1)= 5.03361700*Astar(ipipe)
        VBFastRoughm(2)= 6.48760379*Astar(ipipe)
        VBFastRoughm(3)= 10.7735276*Astar(ipipe)
        VBFastRoughm(4)= 19.9040427*Astar(ipipe)
        VBFastRoughm(5)= 37.4753591*Astar(ipipe)
        VBFastRoughm(6)= 70.7117231*Astar(ipipe)
        VBFastRoughm(7)= 133.460256*Astar(ipipe)
        VBFastRoughm(8)= 251.933272*Astar(ipipe)
        VBFastRoughm(9)= 476.596884*Astar(ipipe)
        VBFastRoughm(10)=932.859931*Astar(ipipe)
C
        VBFastRoughtau(1)= 0.100000000
        VBFastRoughtau(2)= 0.031972893
        VBFastRoughtau(3)= 0.008703605
        VBFastRoughtau(4)= 0.002435768
        VBFastRoughtau(5)= 0.000683579
        VBFastRoughtau(6)= 0.000191876
        VBFastRoughtau(7)= 0.000053859
        VBFastRoughtau(8)= 0.000015109
        VBFastRoughtau(9)= 0.000004199
        VBFastRoughtau(10)=0.000001024
C
!       end if
```

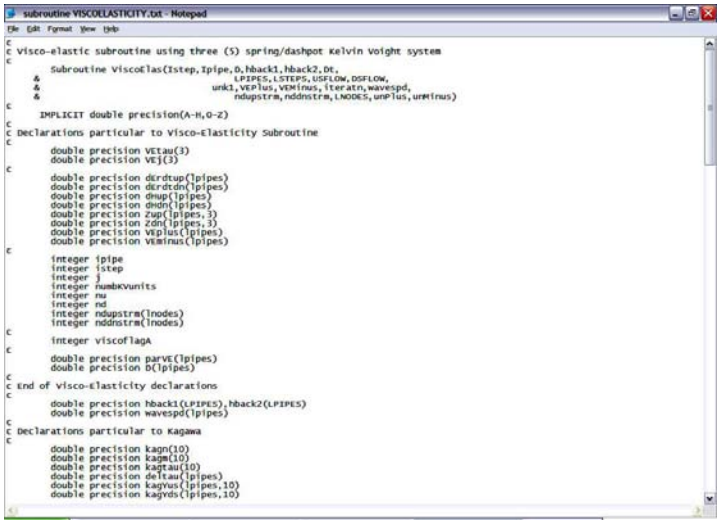



```
!       end if
C
C Number of past weights to be used
C
        VBFastRoughk=0
        do i=1,10
          if((VBFastRoughk.eq.0).and.
     &    ((deltau/2.0).gt.VBFastRoughtau(i)))then
            VBFastRoughk=i
          end if
        end do
        if(VBFastRoughk.eq.0)then
          VBFastRoughk=10
        end if
C
C Calculate unplus, unminus terms and multiply by parzie ready
C to go back into friction terms in assemble
C
C Start four steps in to give flow array for method to draw on
C at previous steps -1 and -2
C
        if(iteratn.eq.1)then
C
        if(istep.ge.3)then
C
C Calculate Kagawa Y variables
C
          unplus(ipipe)=0.0d0
          unminus(ipipe)=0.0d0
C
*****
C Initialising VBFastRoughvus&ds variables
*****
          if(istep.eq.3)then
            do iVBFastRough=1,VBFastRoughk
**          do kk=1,lpipes
                VBFastRoughvus(ipipe,iVBFastRough)=0.0d0
                VBFastRoughvds(ipipe,iVBFastRough)=0.0d0
**          end do
            end do
          end if
*****
C
          do iVBFastRough=1,VBFastRoughk
C
C upstream term
C
            VBFastRoughvus(ipipe,iVBFastRough)=
     &                    VBFastRoughvus(ipipe,iVBFastRough)*
     &                    dexp(-VBFastRoughn(iVBFastRough)*
     &                    deltau)+VBFastRoughm(iVBFastRough)*
```



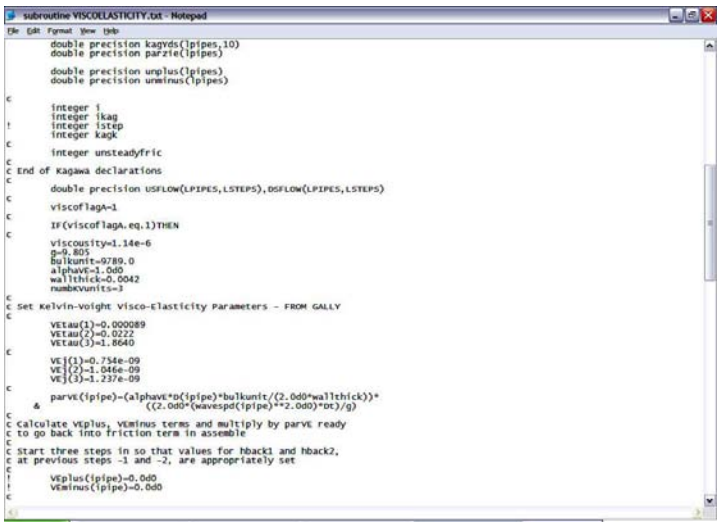

```
            end do
          end if
*****
C
          do iVBFastRough=1,VBFastRoughk
C
C upstream term
C
            VBFastRoughvus(ipipe,iVBFastRough)=
     &                    VBFastRoughvus(ipipe,iVBFastRough)*
     &                    dexp(-VBFastRoughn(iVBFastRough)*
     &                    deltau)+VBFastRoughm(iVBFastRough)*
     &                    dexp(-VBFastRoughn(iVBFastRough)*
     &                    deltau/2.0)*(usflow(ipipe,istep-1)-
     &                    usflow(ipipe,istep-2)) !+dsflow(ipipe,
!     &                     istep)-dsflow(ipipe,istep-1))*0.5
C
            unplus(ipipe)=unplus(ipipe)+
     &                    VBFastRoughvus(ipipe,iVBFastRough)
C
C downstream term
C
            VBFastRoughvds(ipipe,iVBFastRough)=
     &                    VBFastRoughvds(ipipe,iVBFastRough)*
     &                    dexp(-VBFastRoughn(iVBFastRough)*
     &                    deltau)+VBFastRoughm(iVBFastRough)*
     &                    dexp(-VBFastRoughn(iVBFastRough)*
     &                    deltau/2.0)*(dsflow(ipipe,istep-1)-
     &                    dsflow(ipipe,istep-2)) !+usflow(ipipe,
!     &                     istep)-usflow(ipipe,istep-1))*0.5
C
            unminus(ipipe)=unminus(ipipe)+
     &                    VBFastRoughvds(ipipe,iVBFastRough)
C
          end do
C
          unplus(ipipe)=-unplus(ipipe)*parzie
          unminus(ipipe)=-unminus(ipipe)*parzie
C
        else
C
          unplus(ipipe)=0.0d0
          unminus(ipipe)=0.0d0
C
        end if
C
        end if
C
        return
        end
```

Subroutine for viscoelasticity

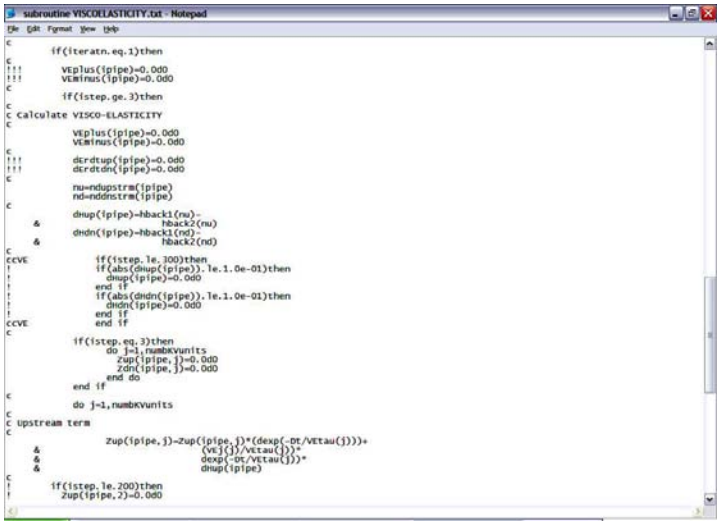



```
C
C Visco-elastic subroutine using three (5) spring/dashpot Kelvin Voight system
C
        Subroutine ViscoElas(Istep,Ipipe,0,hback1,hback2,Dt,
     &                       LPIPES,LSTEPS,USFLOW,DSFLOW,
     &                  unk1,VEPlus,VEMinus,iteratn,wavespd,
     &                       ndupstrm,nddnstrm,LNODES,unPlus,unMinus)
C
      IMPLICIT double precision(A-H,O-Z)
C
C Declarations particular to Visco-Elasticity Subroutine
C
        double precision VEtau(3)
        double precision VEj(3)
C
        double precision dirdtup(lpipes)
        double precision dirdtdn(lpipes)
        double precision dnup(lpipes)
        double precision dndn(lpipes)
        double precision zup(lpipes,3)
        double precision zdn(lpipes,3)
        double precision VEplus(lpipes)
        double precision VEminus(lpipes)
C
        integer ipipe
        integer istep
        integer j
        integer numbKVunits
        integer nu
        integer nd
        integer ndupstrm(lnodes)
        integer nddnstrm(lnodes)
C
        integer viscoflagA
C
        double precision parVE(lpipes)
        double precision D(lpipes)
C
C End of Visco-Elasticity declarations
C
        double precision hback1(LPIPES),hback2(LPIPES)
        double precision wavespd(lpipes)
C
C Declarations particular to Kagawa
C
        double precision kagn(10)
        double precision kagm(10)
        double precision kagtau(10)
        double precision deltau(lpipes)
        double precision kagYus(lpipes,10)
        double precision kagYds(lpipes,10)
```

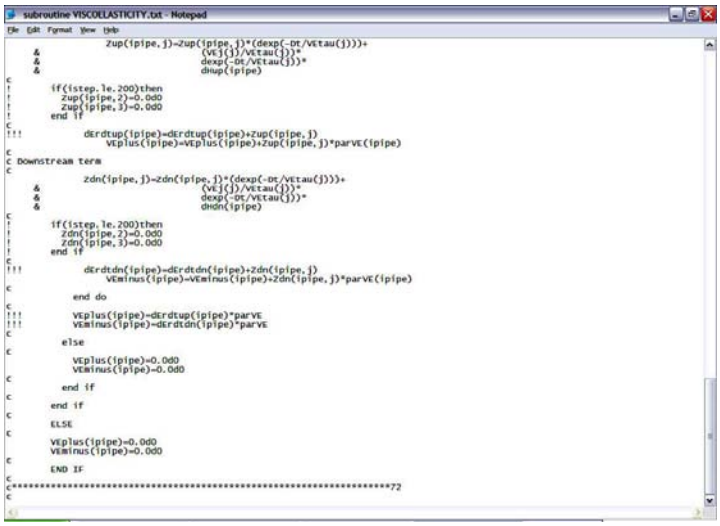
Appendix V – Fortran code for NETTRANS and NLFIT subroutines



```
subroutine VISCOELASTICITY.txt - Notepad
File  Edit  Format  View  Help
      double precision kagVds(lpipes,10)
      double precision parzie(lpipes)

      double precision unplus(lpipes)
      double precision unminus(lpipes)

C
      integer i
      integer ikag
!     integer istep
      integer kagk
C
      integer unsteadyfric
C
C End of Kagawa declarations
C
      double precision USFLOW(LPIPES,LSTEPS),DSFLOW(LPIPES,LSTEPS)
C
      viscoflagA=1
C
      IF(viscoflagA.eq.1)THEN
C
      viscousity=1.14e-6
      g=9.805
      bulkunit=9789.0
      alphaVE=1.0d0
      wallthick=0.0042
      numbKVunits=3
C
C Set Kelvin-Voight Visco-Elasticity Parameters - FROM GALLY
C
      VEtau(1)=0.000089
      VEtau(2)=0.0222
      VEtau(3)=1.8640
C
      VEj(1)=0.754e-09
      VEj(2)=1.046e-09
      VEj(3)=1.237e-09
C
      parVE(ipipe)=(alphaVE*D(ipipe)*bulkunit/(2.0d0*wallthick))*
     &                    ((2.0d0*(wavespd(ipipe)**2.0d0)*Dt)/g)
C
C Calculate VEplus, VEminus terms and multiply by parVE ready
C to go back into friction term in assemble
C
C Start three steps in so that values for hback1 and hback2,
C at previous steps -1 and -2, are appropriately set
C
!     VEplus(ipipe)=0.0d0
!     VEminus(ipipe)=0.0d0
C
```



```
subroutine VISCOELASTICITY.txt - Notepad
File  Edit  Format  View  Help
C
      if(iteratn.eq.1)then
C
!!!     VEplus(ipipe)=0.0d0
!!!     VEminus(ipipe)=0.0d0
C
        if(istep.ge.3)then
C
C Calculate VISCO-ELASTICITY
C
          VEplus(ipipe)=0.0d0
          VEminus(ipipe)=0.0d0
C
!!!       dErdtup(ipipe)=0.0d0
!!!       dErdtdn(ipipe)=0.0d0
C
          nu=ndupstrm(ipipe)
          nd=nddnstrm(ipipe)
C
          dhup(ipipe)=hback1(nu)-
     &                hback2(nu)
          dhdn(ipipe)=hback1(nd)-
     &                hback2(nd)
C
CCVE          if(istep.le.300)then
!             if(abs(dhup(ipipe)).le.1.0e-01)then
!               dhup(ipipe)=0.0d0
!             end if
!             if(abs(dhdn(ipipe)).le.1.0e-01)then
!               dhdn(ipipe)=0.0d0
!             end if
CCVE          end if
C
          if(istep.eq.3)then
              do j=1,numbKVunits
                zup(ipipe,j)=0.0d0
                zdn(ipipe,j)=0.0d0
              end do
          end if
C
          do j=1,numbKVunits
C
C Upstream term
C
              zup(ipipe,j)=zup(ipipe,j)*(dexp(-Dt/VEtau(j)))+
     &                     (VEj(j)/VEtau(j))*
     &                     dexp(-Dt/VEtau(j))*
     &                     dhup(ipipe)
C
!       if(istep.le.200)then
!         zup(ipipe,2)=0.0d0
```

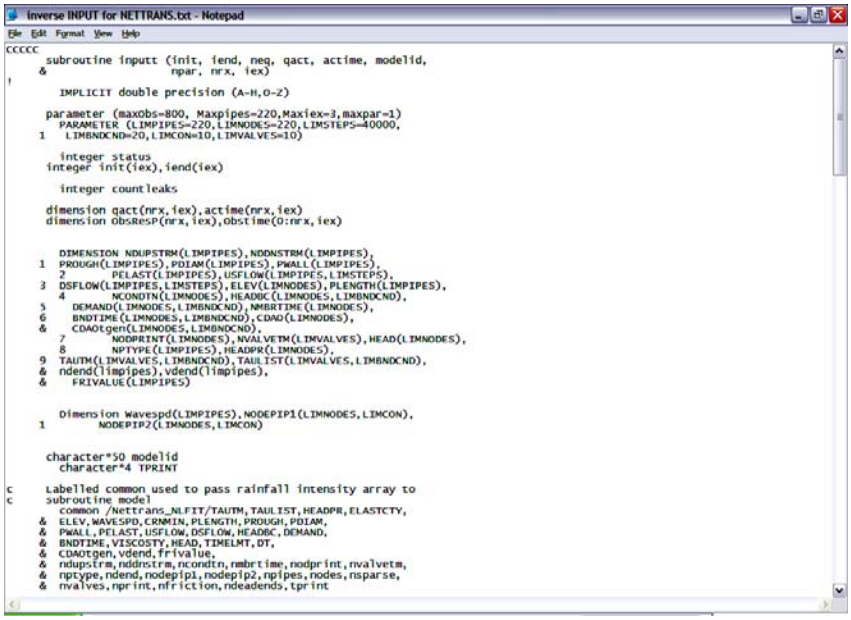


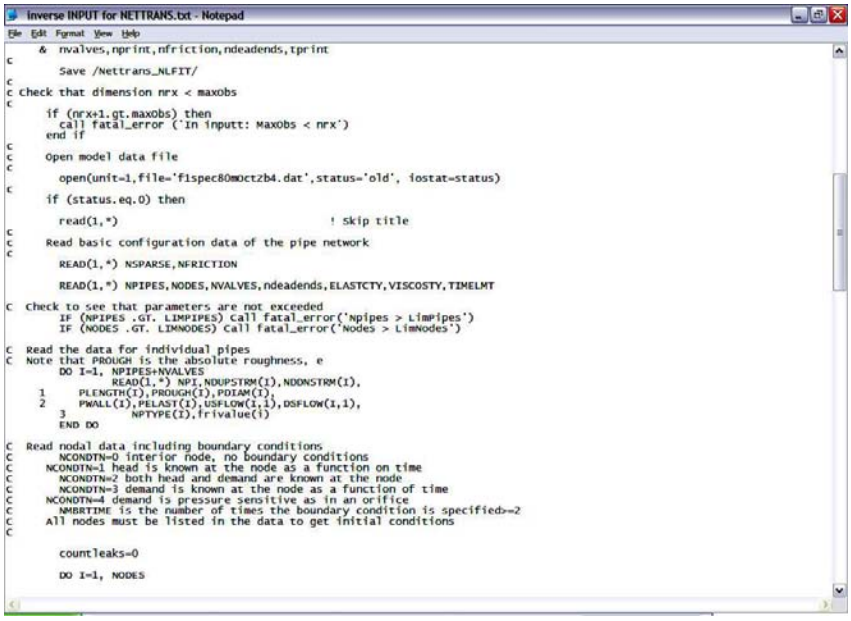
```
subroutine VISCOELASTICITY.txt - Notepad
File  Edit  Format  View  Help
              zup(ipipe,j)=zup(ipipe,j)*(dexp(-Dt/VEtau(j)))+
     &                     (VEj(j)/VEtau(j))*
     &                     dexp(-Dt/VEtau(j))*
     &                     dhup(ipipe)
C
!       if(istep.le.200)then
!         zup(ipipe,2)=0.0d0
!         zup(ipipe,3)=0.0d0
!       end if
C
!!!           dErdtup(ipipe)=dErdtup(ipipe)+zup(ipipe,j)
                VEplus(ipipe)=VEplus(ipipe)+zup(ipipe,j)*parVE(ipipe)
C
C Downstream term
C
              zdn(ipipe,j)=zdn(ipipe,j)*(dexp(-Dt/VEtau(j)))+
     &                     (VEj(j)/VEtau(j))*
     &                     dexp(-Dt/VEtau(j))*
     &                     dhdn(ipipe)
C
!       if(istep.le.200)then
!         zdn(ipipe,2)=0.0d0
!         zdn(ipipe,3)=0.0d0
!       end if
C
!!!           dErdtdn(ipipe)=dErdtdn(ipipe)+zdn(ipipe,j)
                VEminus(ipipe)=VEminus(ipipe)+zdn(ipipe,j)*parVE(ipipe)
C
            end do
C
!!!         VEplus(ipipe)=dErdtup(ipipe)*parVE
!!!         VEminus(ipipe)=dErdtdn(ipipe)*parVE
C
          else
C
            VEplus(ipipe)=0.0d0
            VEminus(ipipe)=0.0d0
C
          end if
C
        end if
C
        ELSE
C
        VEplus(ipipe)=0.0d0
        VEminus(ipipe)=0.0d0
C
        END IF
C
C*******************************************************************72
C
```

V.2 Inverse transient subroutines for NLFIT

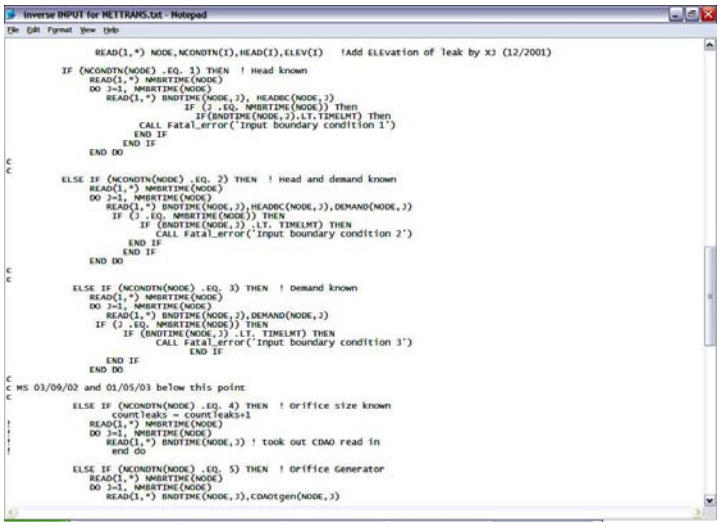
The INPUT and MODEL subroutines required by NLFIT are reproduced below. The subroutine form of DATAIN is absorbed in INPUT. The modified subroutines for ASSEMBLE and the modelling of unsteady friction and viscoelasticity, which are called by MODEL, are not reproduced.

Subroutine INPUT

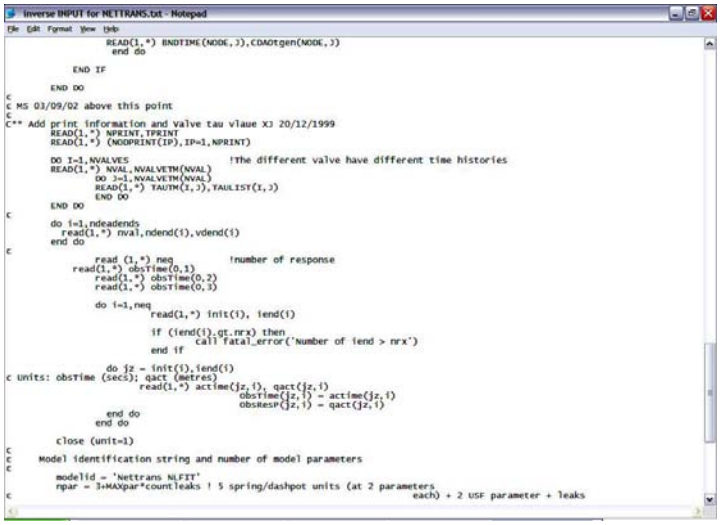




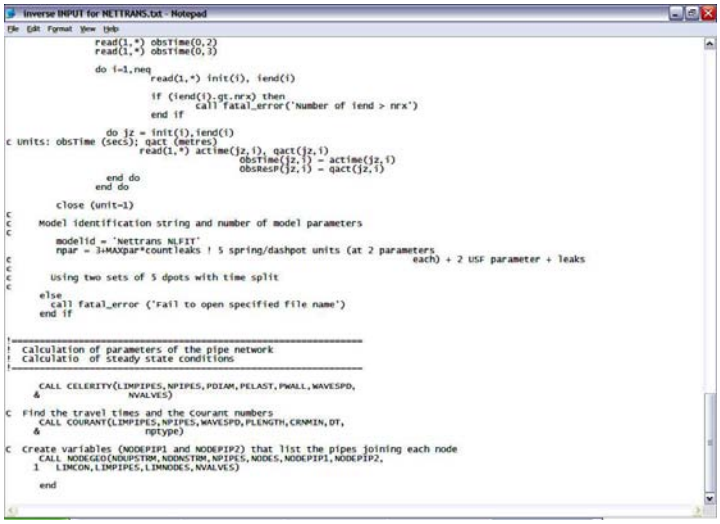
Appendix V – Fortran code for NETTRANS and NLFIT subroutines




```
                READ(1,*) NODE,NCONDTN(I),HEAD(I),ELEV(I)   !Add ELEvation of leak by XJ (12/2001)

          IF (NCONDTN(NODE) .EQ. 1) THEN  ! Head known
              READ(1,*) NMBRTIME(NODE)
              DO J=1, NMBRTIME(NODE)
                 READ(1,*) BNDTIME(NODE,J),  HEADBC(NODE,J)
                           IF (J .EQ. NMBRTIME(NODE)) Then
                                IF(BNDTIME(NODE,J).LT.TIMELMT) Then
                    CALL Fatal_error('Input boundary condition 1')
                  END IF
                END IF
              END DO
C
C
          ELSE IF (NCONDTN(NODE) .EQ. 2) THEN  ! Head and demand known
              READ(1,*) NMBRTIME(NODE)
              DO J=1, NMBRTIME(NODE)
                 READ(1,*) BNDTIME(NODE,J),HEADBC(NODE,J),DEMAND(NODE,J)
                  IF (J .EQ. NMBRTIME(NODE)) THEN
                      IF (BNDTIME(NODE,J) .LT. TIMELMT) THEN
                          CALL Fatal_error('Input boundary condition 2')
                    END IF
                  END IF
              END DO
C
C
          ELSE IF (NCONDTN(NODE) .EQ. 3) THEN  ! Demand known
              READ(1,*) NMBRTIME(NODE)
              DO J=1, NMBRTIME(NODE)
                 READ(1,*) BNDTIME(NODE,J),DEMAND(NODE,J)
               IF (J .EQ. NMBRTIME(NODE)) THEN
                  IF (BNDTIME(NODE,J) .LT. TIMELMT) THEN
                        CALL Fatal_error('Input boundary condition 3')
                              END IF
                END IF
              END DO
C
C MS 03/09/02 and 01/05/03 below this point
C
          ELSE IF (NCONDTN(NODE) .EQ. 4) THEN  ! Orifice size known
                countleaks = countleaks+1
!             READ(1,*) NMBRTIME(NODE)
!             DO J=1, NMBRTIME(NODE)
!                READ(1,*) BNDTIME(NODE,J) ! took out CDAO read in
!              end do

          ELSE IF (NCONDTN(NODE) .EQ. 5) THEN  ! Orifice Generator
              READ(1,*) NMBRTIME(NODE)
              DO J=1, NMBRTIME(NODE)
                 READ(1,*) BNDTIME(NODE,J),CDAOtgen(NODE,J)
```

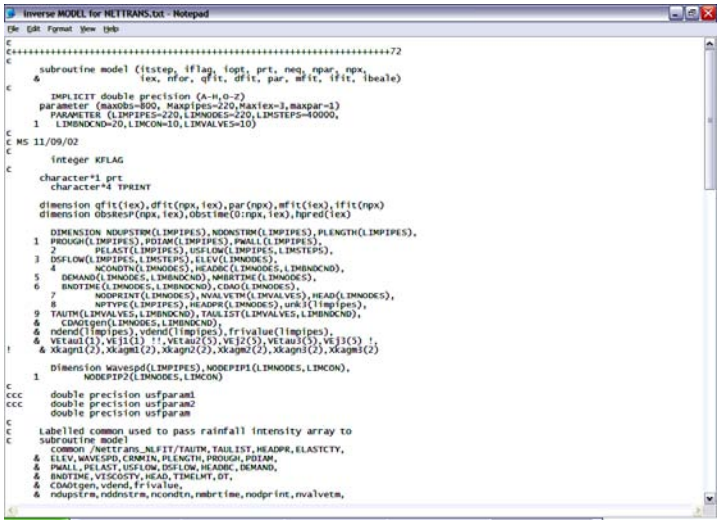



```
                READ(1,*) BNDTIME(NODE,J),CDAOtgen(NODE,J)
                 end do

            END IF

        END DO
C
C MS 03/09/02 above this point
C
C** Add print information and Valve tau vlaue XJ 20/12/1999
        READ(1,*) NPRINT,TPRINT
        READ(1,*) (NODPRINT(IP),IP=1,NPRINT)

        DO I=1,NVALVES                  !The different valve have different time histories
        READ(1,*) NVAL,NVALVETM(NVAL)
                DO J=1,NVALVETM(NVAL)
                READ(1,*) TAUTM(I,J),TAULIST(I,J)
                END DO
        END DO
C
        do i=1,ndeadends
          read(1,*) nval,ndend(i),vdend(i)
        end do
C
                read (1,*) neq          !number of response
            read(1,*) obsTime(0,1)
                read(1,*) obsTime(0,2)
                read(1,*) obsTime(0,3)

                do i=1,neq
                            read(1,*) init(i), iend(i)

                            if (iend(i).gt.nrx) then
                                call fatal_error('Number of iend > nrx')
                            end if

                  do jz = init(i),iend(i)
C Units: obsTime (secs); qact (metres)
                            read(1,*) actime(jz,i), qact(jz,i)
                                                obsTime(jz,i) = actime(jz,i)
                                                obsResp(jz,i) = qact(jz,i)
                  end do
                end do

        close (unit=1)
C
C     Model identification string and number of model parameters
C
        modelid = 'Nettrans NLFIT'
        npar = 3+MAXpar*countleaks ! 5 spring/dashpot units (at 2 parameters
C                                                                  each) + 2 USF parameter + leaks
```

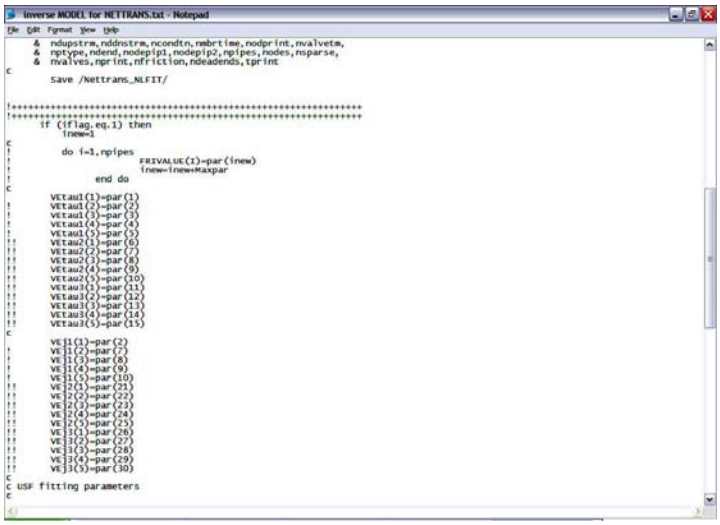



```
                read(1,*) obsTime(0,2)
                read(1,*) obsTime(0,3)

                do i=1,neq
                            read(1,*) init(i), iend(i)

                            if (iend(i).gt.nrx) then
                                call fatal_error('Number of iend > nrx')
                            end if

                  do jz = init(i),iend(i)
C Units: obsTime (secs); qact (metres)
                            read(1,*) actime(jz,i), qact(jz,i)
                                                obsTime(jz,i) = actime(jz,i)
                                                obsResp(jz,i) = qact(jz,i)
                  end do
                end do

        close (unit=1)
C
C     Model identification string and number of model parameters
C
        modelid = 'Nettrans NLFIT'
        npar = 3+MAXpar*countleaks ! 5 spring/dashpot units (at 2 parameters
C                                                                  each) + 2 USF parameter + leaks
C
C       Using two sets of 5 dpots with time split
C
      else
        call fatal_error ('Fail to open specified file name')
      end if


!-------------------------------------------------------------
!  Calculation of parameters of the pipe network
!  Calculatio  of steady state conditions
!-------------------------------------------------------------

       CALL CELERITY(LIMPIPES,NPIPES,PDIAM,PELAST,PWALL,WAVESPD,
     &                NVALVES)

C  Find the travel times and the Courant numbers
       CALL COURANT(LIMPIPES,NPIPES,WAVESPD,PLENGTH,CRNMIN,DT,
     &                nptype)

C  Create variables (NODEPIP1 and NODEPIP2) that list the pipes joining each node
       CALL NODEGEO(NDUPSTRM,NDDNSTRM,NPIPES,NODES,NODEPIP1,NODEPIP2,
     1   LIMCON,LIMPIPES,LIMNODES,NVALVES)

       end
```

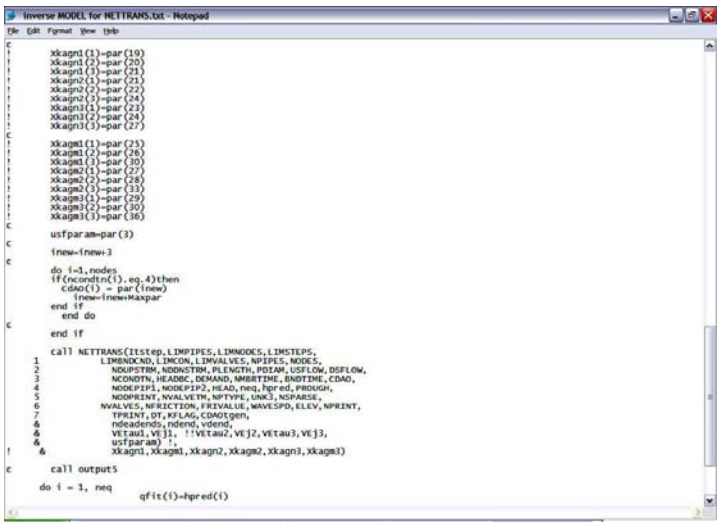
Subroutine MODEL



```
inverse MODEL for NETTRANS.txt - Notepad
File Edit Format View Help
c
c++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++72
c
      subroutine model (itstep, iflag, iopt, prt, neq, npar, npx,
     &                  iex, nfor, qfit, dfit, par, mfit, ifit, ibeale)
c
        IMPLICIT double precision (A-H,O-Z)
       parameter (maxObs=800, Maxpipes=220,Maxiex=3,maxpar=1)
        PARAMETER (LIMPIPES=220,LIMNODES=220,LIMSTEPS=40000,
     1    LIMBNDCND=20,LIMCON=10,LIMVALVES=10)
c
c MS 11/09/02
c
        integer KFLAG
c
       character*1 prt
         character*4 TPRINT

       dimension qfit(iex),dfit(npx,iex),par(npx),mfit(iex),ifit(npx)
       dimension obsResP(npx,iex),obstime(0:npx,iex),hpred(iex)

        DIMENSION NDUPSTRM(LIMPIPES),NDONSTRM(LIMPIPES),PLENGTH(LIMPIPES),
     1  PROUGH(LIMPIPES),PDIAM(LIMPIPES),PWALL(LIMPIPES),
     2        PELAST(LIMPIPES),USFLOW(LIMPIPES,LIMSTEPS),
     3  DSFLOW(LIMPIPES,LIMSTEPS),ELEV(LIMNODES),
     4        NCONDTN(LIMNODES),HEADBC(LIMNODES,LIMBNDCND),
     5    DEMAND(LIMNODES,LIMBNDCND),NMBRTIME(LIMNODES),
     6    BNDTIME(LIMNODES,LIMBNDCND),CDAO(LIMNODES),
     7        NODPRINT(LIMNODES),NVALVETM(LIMVALVES),HEAD(LIMNODES),
     8        NPTYPE(LIMPIPES),HEADPR(LIMNODES),unk3(limpipes),
     9  TAUTM(LIMVALVES,LIMBNDCND),TAULIST(LIMVALVES,LIMBNDCND),
     &    CDAOtgen(LIMNODES,LIMBNDCND),
     &  ndend(limpipes),vdend(limpipes),frivalue(limpipes),
     &  VEtau1(1),VEj1(1) !!,VEtau2(5),VEj2(5),VEtau3(5),VEj3(5) !
!     & Xkagn1(2),Xkagm1(2),Xkagn2(2),Xkagm2(2),Xkagn3(2),Xkagm3(2)

       Dimension Wavespd(LIMPIPES),NODEPIP1(LIMNODES,LIMCON),
     1        NODEPIP2(LIMNODES,LIMCON)
c
ccc     double precision usfparam1
ccc     double precision usfparam2
        double precision usfparam
c
c     Labelled common used to pass rainfall intensity array to
c     subroutine model
      common /Nettrans_NLFIT/TAUTM,TAULIST,HEADPR,ELASTCTY,
     &  ELEV,WAVESPD,CRNMIN,PLENGTH,PROUGH,PDIAM,
     &  PWALL,PELAST,USFLOW,DSFLOW,HEADBC,DEMAND,
     &  BNDTIME,VISCOSTY,HEAD,TIMELMT,DT,
     &  CDAOtgen,vdend,frivalue,
     &  ndupstrm,nddnstrm,ncondtn,nmbrtime,nodprint,nvalvetm,
```

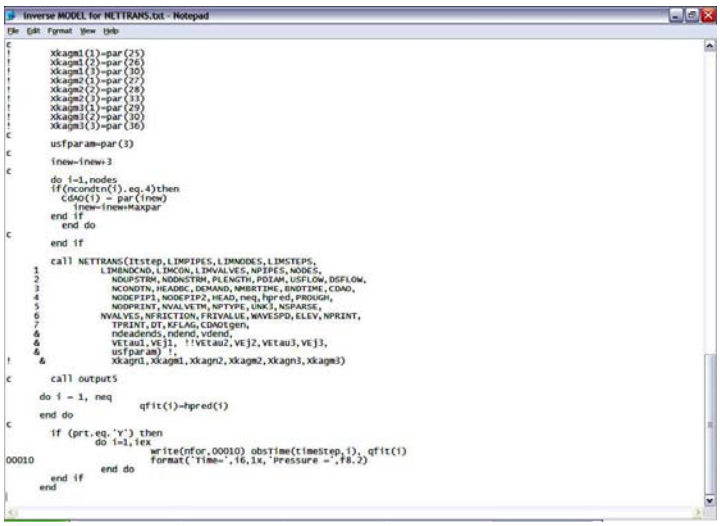


```
inverse MODEL for NETTRANS.txt - Notepad
File Edit Format View Help
     &  ndupstrm,nddnstrm,ncondtn,nmbrtime,nodprint,nvalvetm,
     &  nptype,ndend,nodepip1,nodepip2,npipes,nodes,nsparse,
     &  nvalves,nprint,nfriction,ndeadends,tprint
c
       Save /Nettrans_NLFIT/

!*****************************************************************
!*****************************************************************
      if (iflag.eq.1) then
          inew=1
c
!       do i=1,npipes
!                       FRIVALUE(I)=par(inew)
!                       inew=inew+Maxpar
!             end do
c
       VEtau1(1)=par(1)
!      VEtau1(2)=par(2)
!      VEtau1(3)=par(3)
!      VEtau1(4)=par(4)
!      VEtau1(5)=par(5)
!!     VEtau2(1)=par(6)
!!     VEtau2(2)=par(7)
!!     VEtau2(3)=par(8)
!!     VEtau2(4)=par(9)
!!     VEtau2(5)=par(10)
!!     VEtau3(1)=par(11)
!!     VEtau3(2)=par(12)
!!     VEtau3(3)=par(13)
!!     VEtau3(4)=par(14)
!!     VEtau3(5)=par(15)
c
       VEj1(1)=par(2)
!      VEj1(2)=par(7)
!      VEj1(3)=par(8)
!      VEj1(4)=par(9)
!      VEj1(5)=par(10)
!!     VEj2(1)=par(21)
!!     VEj2(2)=par(22)
!!     VEj2(3)=par(23)
!!     VEj2(4)=par(24)
!!     VEj2(5)=par(25)
!!     VEj3(1)=par(26)
!!     VEj3(2)=par(27)
!!     VEj3(3)=par(28)
!!     VEj3(4)=par(29)
!!     VEj3(5)=par(30)
c
c USF fitting parameters
c
```

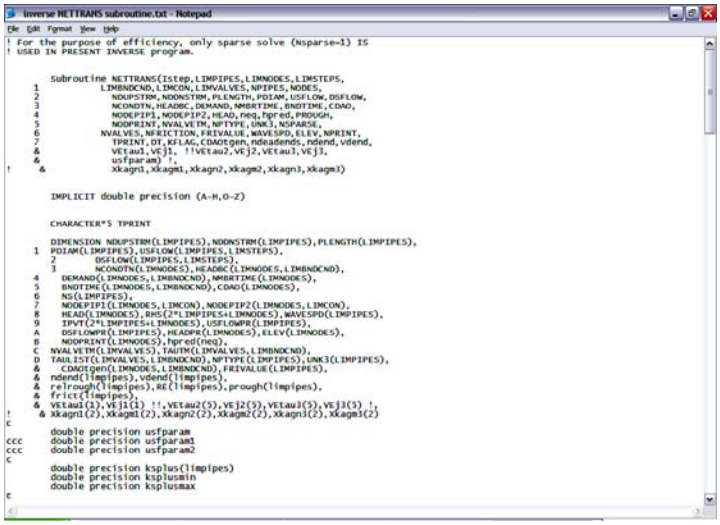


```
inverse MODEL for NETTRANS.txt - Notepad
File Edit Format View Help
c
!       Xkagn1(1)=par(19)
!       Xkagn1(2)=par(20)
!       Xkagn1(3)=par(21)
!       Xkagn2(1)=par(21)
!       Xkagn2(2)=par(22)
!       Xkagn2(3)=par(24)
!       Xkagn3(1)=par(23)
!       Xkagn3(2)=par(24)
!       Xkagn3(3)=par(27)

!       Xkagm1(1)=par(25)
!       Xkagm1(2)=par(26)
!       Xkagm1(3)=par(30)
!       Xkagm2(1)=par(27)
!       Xkagm2(2)=par(28)
!       Xkagm2(3)=par(33)
!       Xkagm3(1)=par(29)
!       Xkagm3(2)=par(30)
!       Xkagm3(3)=par(36)
c
        usfparam=par(3)
c
        inew=inew+3
c
        do i=1,nodes
        if(ncondtn(i).eq.4)then
          CdAO(i) = par(inew)
            inew=inew+Maxpar
        end if
          end do
c
        end if

        call NETTRANS(itstep,LIMPIPES,LIMNODES,LIMSTEPS,
     1            LIMBNDCND,LIMCON,LIMVALVES,NPIPES,NODES,
     2              NDUPSTRM,NDONSTRM,PLENGTH,PDIAM,USFLOW,DSFLOW,
     3              NCONDTN,HEADBC,DEMAND,NMBRTIME,BNDTIME,CDAO,
     4              NODEPIP1,NODEPIP2,HEAD,neq,hpred,PROUGH,
     5              NODPRINT,NVALVETM,NPTYPE,UNK3,NSPARSE,
     6            NVALVES,NFRICTION,FRIVALUE,WAVESPD,ELEV,NPRINT,
     7              TPRINT,DT,KFLAG,CDAOtgen,
     &              ndeadends,ndend,vdend,
     &              VEtau1,VEj1, !!VEtau2,VEj2,VEtau3,VEj3,
     &              usfparam) !,
!    &              Xkagn1,Xkagm1,Xkagn2,Xkagm2,Xkagn3,Xkagm3)

c       call outputs

      do i = 1, neq
                      qfit(i)=hpred(i)
```

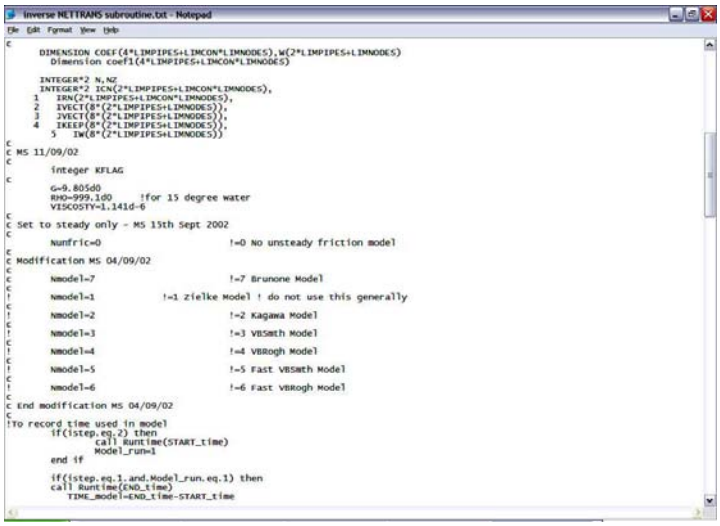
Appendix V – Fortran code for NETTRANS and NLFIT subroutines



```
inverse MODEL for NETTRANS.txt - Notepad
File  Edit  Format  View  Help
C
!       Xkagm1(1)=par(25)
!       Xkagm1(2)=par(26)
!       Xkagm1(3)=par(30)
!       Xkagm2(1)=par(27)
!       Xkagm2(2)=par(28)
!       Xkagm2(3)=par(31)
!       Xkagm3(1)=par(29)
!       Xkagm3(2)=par(32)
!       Xkagm3(3)=par(36)
C
        usfparam=par(3)
C
        inew=inew+3
C
        do i=1,nodes
        if(ncondtn(i).eq.4)then
          cdAO(i) = par(inew)
            inew=inew+Maxpar
        end if
          end do
C
        end if

        call NETTRANS(itstep,LIMPIPES,LIMNODES,LIMSTEPS,
     1            LIMBNDCND,LIMCON,LIMVALVES,NPIPES,NODES,
     2              NDUPSTRM,NDDNSTRM,PLENGTH,PDIAM,USFLOW,DSFLOW,
     3              NCONDTN,HEADBC,DEMAND,NMBRTIME,BNDTIME,CDAO,
     4              NODEPIP1,NODEPIP2,HEAD,neq,hpred,PROUGH,
     5              NODPRINT,NVALVETM,NPTYPE,UNK3,NSPARSE,
     6            NVALVES,NFRICTION,FRIVALUE,WAVESPD,ELEV,NPRINT,
     7              TPRINT,DT,KFLAG,CDAOtgen,
     &              ndeadends,ndend,vdend,
     &              VEtau1,VEj1,  !!VEtau2,VEj2,VEtau3,VEj3,
     &              usfparam) !,
!     &              Xkagn1,xkagm1,xkagn2,Xkagm2,Xkagn3,Xkagm3)

C       call output5

        do i = 1, neq
                        qfit(i)=hpred(i)
        end do
C
        if (prt.eq.'Y') then
              do i=1,tex
                        write(nfor,00010) obsTime(timeStep,i), qfit(i)
00010                   format('Time=',f6.1x,'Pressure =',f8.2)
              end do
        end if
      end
```

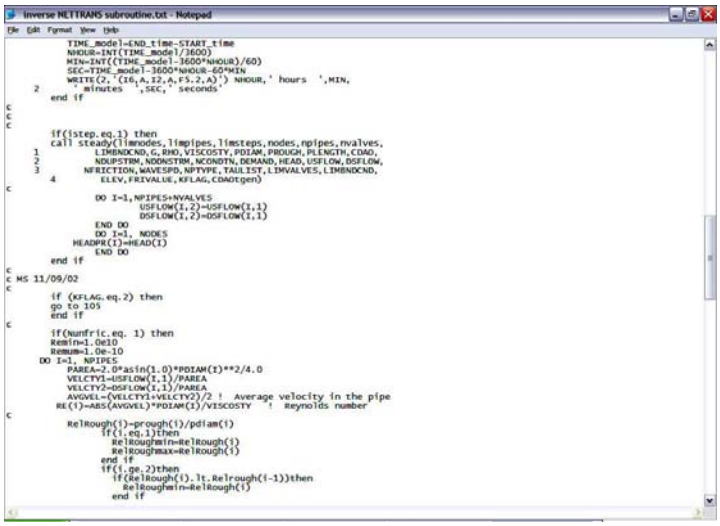


```
inverse NETTRANS subroutine.txt - Notepad
File  Edit  Format  View  Help
! For the purpose of efficiency, only sparse solve (Nsparse=1) IS
! USED IN PRESENT INVERSE program.

        Subroutine NETTRANS(Istep,LIMPIPES,LIMNODES,LIMSTEPS,
     1            LIMBNDCND,LIMCON,LIMVALVES,NPIPES,NODES,
     2              NDUPSTRM,NDDNSTRM,PLENGTH,PDIAM,USFLOW,DSFLOW,
     3              NCONDTN,HEADBC,DEMAND,NMBRTIME,BNDTIME,CDAO,
     4              NODEPIP1,NODEPIP2,HEAD,neq,hpred,PROUGH,
     5              NODPRINT,NVALVETM,NPTYPE,UNK3,NSPARSE,
     6            NVALVES,NFRICTION,FRIVALUE,WAVESPD,ELEV,NPRINT,
     7              TPRINT,DT,KFLAG,CDAOtgen,ndeadends,ndend,vdend,
     &              VEtau1,VEj1,  !!VEtau2,VEj2,VEtau3,VEj3,
     &              usfparam) !,
!     &              Xkagn1,xkagm1,xkagn2,Xkagm2,Xkagn3,Xkagm3)

        IMPLICIT double precision (A-H,O-Z)


        CHARACTER*5 TPRINT

        DIMENSION NDUPSTRM(LIMPIPES),NDDNSTRM(LIMPIPES),PLENGTH(LIMPIPES),
     1  PDIAM(LIMPIPES),USFLOW(LIMPIPES,LIMSTEPS),
     2        DSFLOW(LIMPIPES,LIMSTEPS),
     3        NCONDTN(LIMNODES),HEADBC(LIMNODES,LIMBNDCND),
     4    DEMAND(LIMNODES,LIMBNDCND),NMBRTIME(LIMNODES),
     5    BNDTIME(LIMNODES,LIMBNDCND),CDAO(LIMNODES),
     6    NS(LIMPIPES),
     7    NODEPIP1(LIMNODES,LIMCON),NODEPIP2(LIMNODES,LIMCON),
     8    HEAD(LIMNODES),RHS(2*LIMPIPES+LIMNODES),WAVESPD(LIMPIPES),
     9    IPVT(2*LIMPIPES+LIMNODES),USFLOWPR(LIMPIPES),
     A    DSFLOWPR(LIMPIPES),HEADPR(LIMNODES),ELEV(LIMNODES),
     B    NODPRINT(LIMNODES),hpred(neq),
     C  NVALVETM(LIMVALVES),TAUTM(LIMVALVES,LIMBNDCND),
     D  TAULIST(LIMVALVES,LIMBNDCND),NPTYPE(LIMPIPES),UNK3(LIMPIPES),
     &    CDAOtgen(LIMNODES,LIMBNDCND),FRIVALUE(LIMPIPES),
     &  ndend(limpipes),vdend(limpipes),
     &  relrough(limpipes),RE(limpipes),prough(limpipes),
     &  frict(limpipes),
     &  VEtau1(3),VEj1(3) !!,VEtau2(5),VEj2(5),VEtau3(5),VEj3(5) !,
!     & Xkagn1(2),xkagm1(2),xkagn2(2),Xkagm2(2),xkagn3(2),Xkagm3(2)
C
        double precision usfparam
ccc     double precision usfparam1
ccc     double precision usfparam2
C
        double precision ksplus(limpipes)
        double precision ksplusmin
        double precision ksplusmax
C
```

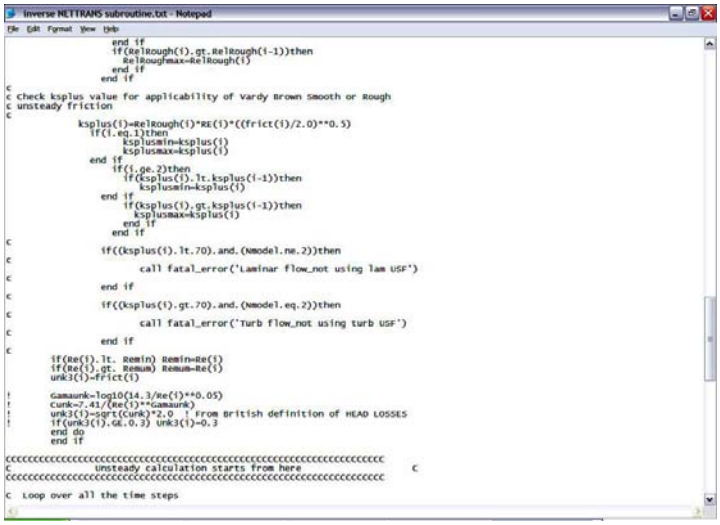


```
inverse NETTRANS subroutine.txt - Notepad
File  Edit  Format  View  Help
C
      DIMENSION COEF(4*LIMPIPES+LIMCON*LIMNODES),W(2*LIMPIPES+LIMNODES)
        Dimension coef1(4*LIMPIPES+LIMCON*LIMNODES)

      INTEGER*2 N,NZ
      INTEGER*2 ICN(2*LIMPIPES+LIMCON*LIMNODES),
     1    IRN(2*LIMPIPES+LIMCON*LIMNODES),
     2    IVECT(8*(2*LIMPIPES+LIMNODES)),
     3    JVECT(8*(2*LIMPIPES+LIMNODES)),
     4    IKEEP(8*(2*LIMPIPES+LIMNODES)),
     5    IW(8*(2*LIMPIPES+LIMNODES))
C
C MS 11/09/02
C
        integer KFLAG
C
        G=9.805d0
        RHO=999.1d0     !for 15 degree water
        VISCOSTY=1.141d-6
C
C Set to steady only - MS 15th Sept 2002
C
        Nunfric=0                       !=0 No unsteady friction model
C
C Modification MS 04/09/02
C
C       Nmodel=7                        !=7 Brunone Model
C
!       Nmodel=1            !=1 Zielke Model ! do not use this generally
C
!       Nmodel=2                        !=2 Kagawa Model
C
!       Nmodel=3                        !=3 VBSmth Model
C
!       Nmodel=4                        !=4 VBRogh Model
C
!       Nmodel=5                        !=5 Fast VBSmth Model
C
!       Nmodel=6                        !=6 Fast VBRogh Model
C
C End modification MS 04/09/02
C
!To record time used in model
        if(istep.eq.2) then
              call Runtime(START_time)
              Model_run=1
        end if

        if(istep.eq.1.and.Model_run.eq.1) then
        call Runtime(END_time)
          TIME_model=END_time-START_time
```

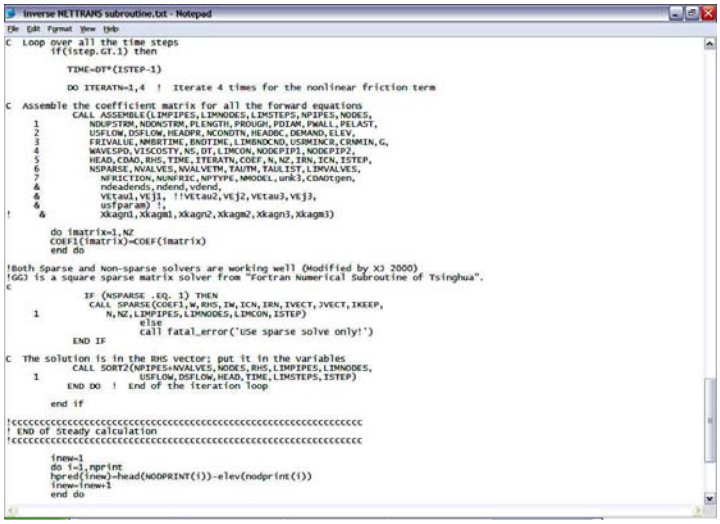
Appendix V – Fortran code for NETTRANS and NLFIT subroutines




```
          TIME_model=END_time-START_time
          NHOUR=INT(TIME_model/3600)
          MIN=INT((TIME_model-3600*NHOUR)/60)
          SEC=TIME_model-3600*NHOUR-60*MIN
          WRITE(2,'(I6,A,I2,A,F5.2,A)') NHOUR,' hours  ',MIN,
     2      ' minutes  ',SEC,' seconds'
        end if
C
C
C
        if(istep.eq.1) then
        call steady(limnodes,limpipes,limsteps,nodes,npipes,nvalves,
     1          LIMBNDCND,G,RHO,VISCOSTY,PDIAM,PROUGH,PLENGTH,CDAO,
     2          NDUPSTRM,NODNSTRM,NCONDTN,DEMAND,HEAD,USFLOW,DSFLOW,
     3        NFRICTION,WAVESPD,NPTYPE,TAULIST,LIMVALVES,LIMBNDCND,
     4          ELEV,FRIVALUE,KFLAG,CDAOtgen)
C
                DO I=1,NPIPES+NVALVES
                        USFLOW(I,2)=USFLOW(I,1)
                        DSFLOW(I,2)=DSFLOW(I,1)
                END DO
                DO I=1, NODES
            HEADPR(I)=HEAD(I)
                END DO
        end if
C
C MS 11/09/02
C
        if (KFLAG.eq.2) then
        go to 105
        end if
C
        if(Numfric.eq. 1) then
        Remin=1.0e10
        Remum=1.0e-10
       DO I=1, NPIPES
          PAREA=2.0*asin(1.0)*PDIAM(I)**2/4.0
          VELCTY1=USFLOW(I,1)/PAREA
          VELCTY2=DSFLOW(I,1)/PAREA
          AVGVEL=(VELCTY1+VELCTY2)/2 !  Average velocity in the pipe
         RE(I)=ABS(AVGVEL)*PDIAM(I)/VISCOSTY   !  Reynolds number
C
          RelRough(I)=prough(I)/pdiam(I)
                if(I.eq.1)then
                  RelRoughmin=RelRough(I)
                  RelRoughmax=RelRough(I)
                end if
                if(I.ge.2)then
                  if(RelRough(I).lt.Relrough(I-1))then
                    RelRoughmin=RelRough(I)
                  end if
```

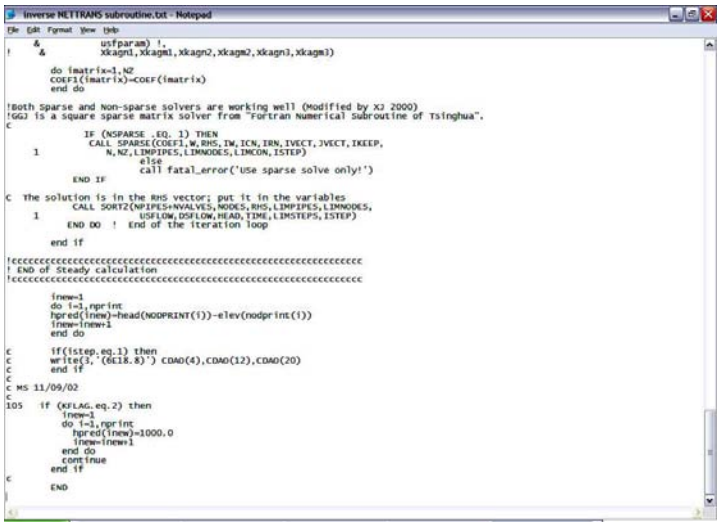



```
                end if
                if(RelRough(I).gt.RelRough(I-1))then
                  RelRoughmax=RelRough(I)
                end if
              end if
C
C Check ksplus value for applicability of Vardy Brown Smooth or Rough
C unsteady friction
C
            ksplus(I)=RelRough(I)*RE(I)*((frict(I)/2.0)**0.5)
              if(I.eq.1)then
                  ksplusmin=ksplus(I)
                  ksplusmax=ksplus(I)
              end if
              if(I.ge.2)then
                if(ksplus(I).lt.ksplus(I-1))then
                  ksplusmin=ksplus(I)
                end if
                if(ksplus(I).gt.ksplus(I-1))then
                  ksplusmax=ksplus(I)
                end if
              end if
C
              if((ksplus(I).lt.70).and.(Nmodel.ne.2))then
C
                    call fatal_error('Laminar flow_not using lam USF')
C
              end if
C
              if((ksplus(I).gt.70).and.(Nmodel.eq.2))then
C
                    call fatal_error('Turb flow_not using turb USF')
C
              end if
C
      if(Re(I).lt. Remin) Remin=Re(I)
      if(Re(I).gt. Remum) Remum=Re(I)
      unk3(I)=frict(I)

!     Gamaunk=log10(14.3/Re(I)**0.05)
!     Cunk=7.41/(Re(I)**Gamaunk)
!     unk3(I)=sqrt(Cunk)*2.0  ! From British definition of HEAD LOSSES
!     if(unk3(I).GE.0.3) Unk3(I)=0.3
      end do
      end if

ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
C          Unsteady calculation starts from here                       C
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc

C  Loop over all the time steps
```




```
C  Loop over all the time steps
      if(istep.GT.1) then

        TIME=DT*(ISTEP-1)

        DO ITERATN=1,4  !  Iterate 4 times for the nonlinear friction term

C  Assemble the coefficient matrix for all the forward equations
        CALL ASSEMBLE(LIMPIPES,LIMNODES,LIMSTEPS,NPIPES,NODES,
     1      NDUPSTRM,NDDNSTRM,PLENGTH,PROUGH,PDIAM,PWALL,PELAST,
     2      USFLOW,DSFLOW,HEADPR,NCONDTN,HEADDC,DEMAND,ELEV,
     3      FRIVALUE,NMBRTIME,ENDTIME,LIMBNDCND,USDMINCR,CRNMIN,G,
     4      WAVESPD,VISCOSTY,NS,DT,LIMCON,NODEPIP1,NODEPIP2,
     5      HEAD,CDAO,RHS,TIME,ITERATN,COEF,N,NZ,IRN,ICN,ISTEP,
     6      NSPARSE,NVALVES,NVALVETM,TAUTM,TAULIST,LIMVALVES,
     7        NFRICTION,NUNFRIC,NPTYPE,NMODEL,unk3,CDAOtgen,
     &        ndeadends,ndend,vdend,
     &        VEtau1,VEj1, !VEtau2,VEj2,VEtau3,VEj3,
     &        usfparam) !,
!    &        Xkagn1,Xkagm1,Xkagn2,Xkagm2,Xkagn3,Xkagm3)

      do imatrix=1,NZ
      COEF1(imatrix)=COEF(imatrix)
      end do

!Both Sparse and Non-sparse solvers are working well (Modified by XJ 2000)
!GGJ is a square sparse matrix solver from "Fortran Numerical Subroutine of Tsinghua".
C
            IF (NSPARSE .EQ. 1) THEN
             CALL SPARSE(COEF1,W,RHS,IW,ICN,IRN,IVECT,JVECT,IKEEP,
     1          N,NZ,LIMPIPES,LIMNODES,LIMCON,ISTEP)
                        else
                        call fatal_error('USe sparse solve only!')
            END IF

C  The solution is in the RHS vector; put it in the variables
            CALL SORT2(NPIPES+NVALVES,NODES,RHS,LIMPIPES,LIMNODES,
     1                  USFLOW,DSFLOW,HEAD,TIME,LIMSTEPS,ISTEP)
          END DO  !  End of the iteration loop

        end if

!ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
! END of Steady calculation
!ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc

        inew=1
        do i=1,nprint
        hpred(inew)=head(NODPRINT(i))-elev(nodprint(i))
        inew=inew+1
        end do
```

Appendix V – Fortran code for NETTRANS and NLFIT subroutines




```
     &           usfparam) !,
!    &           Xkagn1,Xkagm1,Xkagn2,Xkagm2,Xkagn3,Xkagm3)

       do imatrix=1,NZ
       COEF1(imatrix)=COEF(imatrix)
       end do

!Both Sparse and Non-sparse solvers are working well (Modified by XJ 2000)
!GGJ is a square sparse matrix solver from "Fortran Numerical Subroutine of Tsinghua".
C
            IF (NSPARSE .EQ. 1) THEN
             CALL SPARSE(COEF1,W,RHS,IW,ICN,IRN,IVECT,JVECT,IKEEP,
     1          N,NZ,LIMPIPES,LIMNODES,LIMCON,ISTEP)
                        else
                        call fatal_error('USe sparse solve only!')
            END IF

C  The solution is in the RHS vector; put it in the variables
            CALL SORTZ(NPIPES+NVALVES,NODES,RHS,LIMPIPES,LIMNODES,
     1                 USFLOW,DSFLOW,HEAD,TIME,LIMSTEPS,ISTEP)
          END DO  !  End of the iteration loop

       end if

!cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
! END of Steady calculation
!cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc

       inew=1
       do i=1,nprint
       hpred(inew)=head(NODPRINT(i))-elev(nodprint(i))
       inew=inew+1
       end do

C      if(istep.eq.1) then
C      write(3,'(6E18.8)') CDAO(4),CDAO(12),CDAO(20)
C      end if
C
C MS 11/09/02
C
105   if (KFLAG.eq.2) then
         inew=1
         do i=1,nprint
           hpred(inew)=1000.0
           inew=inew+1
         end do
         continue
       end if
C
       END
```

Appendix W

Wave speed estimation and other complexities for distribution pipelines

W.1 Concept of spatial decomposition of distribution network

Distribution pipelines are more topologically and otherwise complex than transmission pipelines. However, by reducing the testing and analysis to the scale of individual pipes within a network, various physical complexities can be potentially managed. In particular, the demands, local pipe roughness and effect of accumulated minor losses can be either eliminated or included in a forward transient model providing the time and space over which a detailed understanding is sought is limited.

Figure W-1 shows a typical section of the network supplying the City of Adelaide, South Australia and the inset shows the section of the system encompassing the Foster Street Pipeline (FSP). The universal presence of isolation valves (in-line gate valves) at ends of sections of pipe in the network allows decomposition into branch pipelines by closing an isolation valve at one end while leaving the other end open.

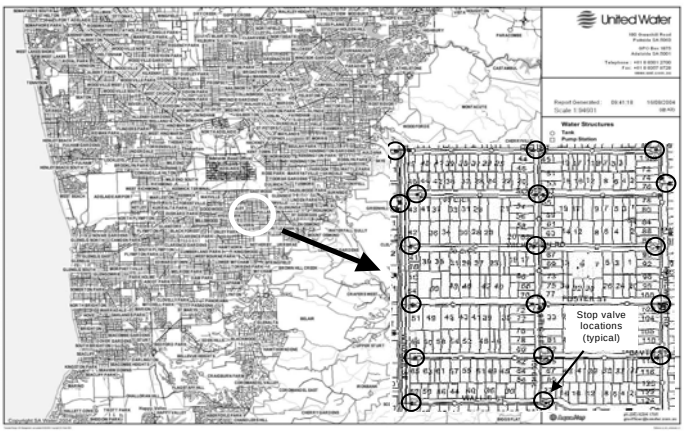


Figure W-1 – Typical sub-network zone within the Adelaide network showing isolation valves at the ends of pipes servicing individual streets

Any transient induced in a pipeline near a closed isolation valve will not be influenced by the network beyond the opposite open end until it has travelled along the pipeline, reached the open end, and information from the network begins influencing the reflected wave propagating back along the pipeline. Information contained within the transient response for the period prior to reflection from the open end will only relate to the individual section of pipeline (including any potential fault along that section).

W.2 Assessment of demands

Neither the Saint Johns Terrace Pipeline (SJTP) nor Kookaburra Court Pipeline (KCP) were isolated from the Willunga Network during the period of the transient tests. The main reason for this was not to disrupt supply to the residences with water services along either street. However, another important reason was the need to maintain pressure and supply to the transient generator. As a consequence, it was necessary to observe background transient fluctuations in the Willunga Network and initiate tests during time windows when less significant pressure fluctuations were occurring.

While it was practical to isolate water services by closing isolation valves at individual water meters it was decided not to do so during the transient tests. This would have eliminated the physical complexity associated with the transient response of each water service network (i.e., the private plumbing associated with each residence), beyond the point of the water meters. That said, notices were issued, through United Water, to each resident, two days prior to each test date, informing each household that while there would be no formal interruption to water supply on the tests dates, it was recommended that water not be used during a specified period.

W.3 Assessment of entrained air

Saint Johns Terrace Pipeline (SJTP)

The minimum and maximum grades along the Saint Johns Terrace Pipeline (SJTP), of approximately 1:25 and 1:8, respectively, are greater than those required for entrained

air to migrate to the high point(s). Furthermore, fire plugs are located at the high, and other, points along the SJTP. It was therefore anticipated that any entrained air along the SJTP would rapidly migrate to one of the 8 fire plug risers and from there could be flushed. The author personally flushed all fire plugs on the 23rd July, 15th August and 26th August 2003 and observed, at most, the release of a maximum of 10 bubbles, approximately 1 to 2mm in diameter, at any one fire plug. It was considered unlikely that any significant quantity of entrained air remained along the SJTP after flushing.

Kookaburra Court Pipeline (KCP)

The minimum and maximum grades along the Kookaburra Court Pipeline (KCP), of approximately 1:27 and 1:10, respectively, are greater than those required for entrained air to migrate to the high point(s). Furthermore, fire plugs are located at the high, and other, points along the KCP. It was therefore anticipated that any entrained air along the KCP would rapidly migrate to one of the 5 fire plug risers and from there could be flushed. The author personally flushed all fire plugs on the 28th August 2003 and observed, as for the SJTP, the release of a maximum of 10 bubbles, approximately 1 to 2mm in diameter, at any one fire plug.

W.4 Pipeline roughness and steady state friction factors

As for transmission pipelines, estimating the roughness along individual distribution pipelines is not straightforward. Theoretical estimates of the likely roughness of the pipe wall can be made taking into account the time since installation and the nature of the potable water transported within the pipes. However these estimates only limit the range of roughness values to between 0.5 and 2mm. Furthermore, such estimates do not account for the possibility of spalling of cement mortar lining, corrosion of exposed metal surfaces, formation of bio-films or the development of turberculation.

Steady flow and pressures information was used to calibrate the roughness of the distribution pipelines. Steady pressure data was primarily obtained from the pressure records for each transient test prior to the initiation of the transient (i.e., the steady period on each test record prior to the transient event). However, additional steady

flow and pressure tests were separately conducted in the month of September 2003 (after the last transient tests conducted during August 2003).

Assessment of pipeline roughness for the SJTP

A steady state model of the Saint Johns Terrace Pipeline (SJTP) was developed and analysis was performed with different distributions of roughness along the SJTP (the steady solver developed within the forward transient model presented in Appendix N was used for this purpose). A uniform roughness of 1mm along the entire SJTP gave a satisfactory match between the pre-transient record, manually gauged and predicted steady state pressures for a range of nozzle sizes. Furthermore, a satisfactory match between the pre-transient record and predicted steady state pressures for the case with baseflow along the SJTP, through a 25mm standpipe mounted orifice at the dead end, and simultaneous discharge through a 8mm nozzle, was achieved.

Table W-1 summarises the relevant lumped discharge coefficients, flow rates, velocities, Reynolds numbers and friction factors for the 6, 8 and 10mm nozzles, without baseflow, and for the 8mm nozzle with baseflow. In all cases, the Reynolds number for the flow is above the laminar to turbulent threshold of 4000. However, the flow remains in the transition zone on the Moody diagram and does not reach the turbulent region. Nevertheless, the Swamee and Jain equation can be applied to determine the friction factor along the SJTP for each of the flow conditions listed in Table W-1:

$$f = \frac{1.325}{\left[\ln\left(\varepsilon/3.7D + 5.74/R^{0.8}\right)\right]^2} \qquad \text{(W-1)}$$

where ε is the pipe roughness (mm), D is the pipe diameter (mm) and R is the Reynolds number

and $10^{-6} \le \frac{\varepsilon}{D} \le 10^{-2}$ with $5000 \le R \le 10^{8}$

Appendix W – Wave speed estimation and other complexities for distribution pipes

Table W-1 – Steady flow rates, Reynolds numbers and friction factors for the SJTP

Case	C_dA_{nozzle} (m^2)	$C_dA_{endorifice}$ (m^2)	Q_{nozzle} (L/s)	Q_{total} (L/s)	$V_{main\ pipe}$ (m/s)	Reynolds number	Friction factor
6mm nozzle	0.0000254	NA	0.66	0.69	0.098	8117	0.046
8mm nozzle	0.0000452	NA	1.16	1.19	0.170	14080	0.043
10mm nozzle	0.0000707	NA	1.80	1.83	0.261	21617	0.042
Baseflow + 8mm nozzle	0.0000452	0.0003191	0.66	8.12	1.158	95908	0.0395

Table W-2 lists the predicted pressures, for discharge through an 8mm nozzle with baseflow, for wall roughness values of 0.5, 1.0, 2.0, 3.0 and 4.0mm. The results confirm that the pre-transient record and manually gauged steady pressures, for each respective flow condition, are sensitive to the roughness of the SJTP and subsequently determined friction factors. A roughness value of 1mm has been adopted.

Table W-2 – Comparison of predicted and measured pressures at three locations along the SJTP for a range wall roughness values

Roughness (mm)	Location 1		Location 2		Location 3	
	Predicted	Measured	Predicted	Measured	Predicted	Measured
0.5	8.36	7.6	12.98	12.0	26.53	25.0
1.0	**7.63**	**7.6**	**11.97**	**12.0**	**24.87**	**25.0**
2.0	6.74	7.6	10.71	12.0	22.80	25.0
3.0	6.13	7.6	9.85	12.0	21.38	25.0
4.0	5.65	7.6	9.19	12.0	20.28	25.0

where the locations correspond to points at which the steady state pressure was manually gauged

Assessment of pipeline roughness for the KCP

A steady state model of the Kookaburra Court Pipeline (KCP) was developed and analysis was performed with different distributions of roughness along the KCP. Roughness values of 1mm, 4mm and 2mm along the first 225.0m, next 93.1m and final 60.1m of the KCP, respectively, gave a satisfactory match between the pre-transient record, manually gauged and predicted steady state pressures for a range of nozzle sizes.

Anecdotal evidence, obtained during discussions with United Water operators, suggested that there may be a problem with the cement mortar lining of the KCP along the 93.1m section between stations 1 and 2. Residents along this section had lodged water quality complaints and discoloured discharge through the fourth fire plug was noted during subsequent flushing (i.e., through the fire plug upon which the transient generator was mounted. The absence of discoloured discharge through the upstream fire plug suggested that the 93.1m long section of the KCP was in a deteriorated condition.

Table W-3 summarises the relevant lumped discharge coefficients, flow rates, velocities, Reynolds numbers and friction factors for the 6, 8 and 10mm nozzles. In all cases, the Reynolds number for the flow is above the laminar to turbulent threshold of 4000. However, the flow remains in the transition zone on the Moody diagram and does not reach the fully turbulent region. Nevertheless, the Moody diagram can be used to determine the friction factors listed in Table W-3:

Table W-3 – Steady flow rates, Reynolds numbers and friction factors for KCP

Case	C_dA_{nozzle} (m^2)	Q_{nozzle} (L/s)	Q_{total} (L/s)	$V_{\varepsilon=1mm}$ (m/s)	$R_{\varepsilon=1mm}$	$f_{\varepsilon=1mm}$	$V_{\varepsilon=4mm}$ (m/s)	$R_{\varepsilon=4mm}$	$f_{\varepsilon=4mm}$
6mm nozzle	0.000025447	0.69	0.74	0.102	8582	0.045	0.116	9150	0.072
8mm nozzle	0.000045239	1.21	1.26	0.174	14640	0.043	0.198	15618	0.070
10mm nozzle	0.000070685	1.87	1.92	0.265	22296	0.042	0.302	23821	0.0695

W.5 Assessment of minor losses

Water meter assemblies

The mechanism of the typical water meter installed on every water service connection in South Australia represents a physical complexity that is likely to contribute to observed dispersion and damping of transient wavefronts along distribution pipelines. Figure W-2 shows a typical water meter commonly installed by United Water. Flow through the meter eccentrically rotates a circular piston within the measuring chamber such that each revolution corresponds to the transfer of a known volume of water. The

rotation is gauged using a gear and dial unit. United Water operators indicated that most water meters are fitted with check valves at their inlets to prevent backflow and contamination from private plumbing to the street main. Both the circular piston and check valve act to create hydraulic loss.

NOTE: This figure is included on page 688 of the print copy of the thesis held in the University of Adelaide Library.

Figure W-2 – Typical water meter internal mechanism (courtesy of United Water)

Minor losses for the Saint Johns Terrace Pipeline (SJTP)

There are a number of potential minor loss elements along the Saint Johns Terrace Pipeline (SJTP), and associated water service networks, apart from the in-situ in-line gate valve used to simulate discrete blockages. These include 8 by 80mm diameter risers to fire plug valves, 6 by 25mm diameter water service offtakes to 6 by 20mm water meters and the private plumbing associated with each residence beyond. Vertical branch sections have been included in the transient model at the location of pipe risers to each fire plug valve. An orifice element has been included in these vertical branch sections in order to represent a constriction at the seating point in the fire plug valves. This orifice element is only important if the location of the transient generator coincides with a particular fire plug. Approximately horizontal branches have been included to represent the 25mm diameter water services. Water meters at property boundaries are represented using orifice elements that constrict the water service pipe diameter to 20mm. Details of pipework more than 5m beyond the water meters are not included.

Minor losses for the Kookaburra Court Pipeline (KCP)

There are a number of potential minor loss elements along the Kookaburra Court Pipeline (KCP) and associated water service networks. These include significant changes in profile, 5 by 80mm diameter risers to fire plug valves, 15 by 25mm diameter water service offtakes to 15 by 20mm water meters and private plumbing associated with each residence beyond. As for the Saint Johns Terrace Pipeline (SJTP), vertical branch sections have been included in the transient model at the location of pipe risers to each fire plug valve. Similarly, approximately horizontal branches have been included to represent the 25mm diameter water services. Water meters at property boundaries are represented using orifice elements that constrict the water service pipe diameter to 20mm. Details of pipework more than 5m beyond the water meters are not included.

While the elevation profile of the KCP does not change by more than approximately 15 degrees, the plan profile of the KCP changes significantly at two locations. The first change occurs at a sweeping 135 degree bend. This bend is not abrupt, and presents no specific location at which to include a minor loss. However, it is known that such bends are constructed by utilising “off-axis” tolerance at each flexible rubber ring joint to gradually adjust the alignment of a pipeline. The second change is an abrupt 90 degree bend near the dead end of the KCP. An equivalent in-line orifice is included at the location of the 90 degree bend and the associated pressure loss is calculated using a quasi-steady loss coefficient of 0.9.

W.6 Wave speed and wall condition for the SJTP

The Saint Johns Terrace Pipeline (SJTP) was constructed in 1976 from 100mm diameter Asbestos Cement (AC) pipe with flexible rubber ring joints at a spacing of 3m. Figure W-3 summarises the geometric and material properties of 100mm nominal diameter “Class D” AC pipe (manufactured in accordance with obsolete Australian Standard AS 1711 – 1975). The thickness of the asbestos cement comprising the pipe wall is 12.7mm. The elastic modulus for the AC pipe was initially estimated based on typical values for cement pipes. However, subsequent stress/strain laboratory tests

were performed on a test section, at the University of Adelaide, as described in Appendix D, and an elastic modulus of 32GPa was determined. This value is consistent with those obtained in other tests performed by South Australian Water Corporation personnel.

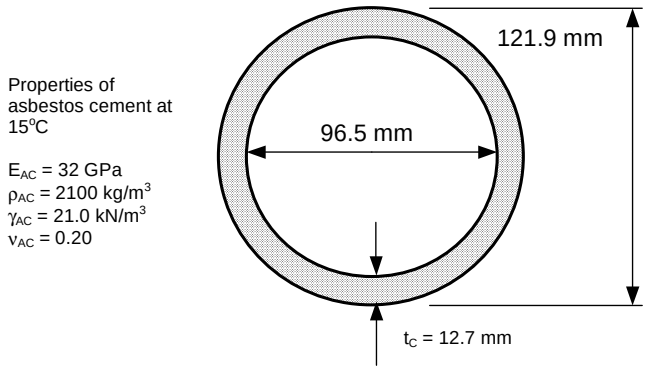


Figure W-3 – Cross section and details of the Saint Johns Terrace Pipeline

Theoretical estimation of wave speed for the SJTP

The theoretical wave speed for the Saint Johns Terrace Pipeline (SJTP) can be estimated using the following equation presented by Wylie and Streeter (1993):

$$a = \sqrt{\frac{K/\rho}{1+(K/E_{AC})(D/e)c}} \qquad \text{(W-2)}$$

where K is the bulk modulus of water, ρ is the density of water, E_{AC} is the elastic modulus of asbestos cement, D is the internal diameter of the pipe, e is the thickness of the pipe wall and c is a pipe restraint factor which, for a pipe with expansion joints, is 1

It is arguable whether the SJTP is axially restrained. The pipeline comprises a multitude of 3m long segments with flexible rubber ring collar joints between each segment. These joints are capable of acting as expansion joints. That said, the SJTP is buried in trenches with compacted backfill and has fixed points of restraint at the “T”

intersection, 2 in-line gate valves, 8 fire plugs, 6 water service connections and a dead end. It is assumed that these elements collectively comprise a considerable degree of axial restraint despite the presence of flexible joints. The restraint factor for the SJTP has consequently been determined using $c = 1 - \nu^2$. The approximate theoretical wave speed for the SJTP is 1199.9m/s.

Direct estimation of wave speed for the SJTP

The arrival times of the incident wavefronts, and reflected wavefronts, can be determined from the measured responses of the Saint Johns Terrace Pipeline (SJTP). The effect of dispersion has been dealt with by using consistent points of reference on each wavefront. A minimum travel time is determined by taking the difference between the time at the base (first rise) of the incident wavefront, recorded at the location of the transient generator, and at the base (first rise) of the incident or reflected wavefront at the next relevant station. A maximum travel time is determined by taking the difference between the top (last rise) of the incident wavefront and the top (last rise) of the incident or reflected wavefront at the next relevant station. The average travel time is determined by taking the difference between midway points on the incident and reflected wavefronts.

Table W-4 shows the wavefront travel times between stations 1 and 2, stations 2 and 3, and between station 2 and the "T" intersection, for the specified tests. In the case of tests 4 and 5, conducted on the 23rd July 2003, the closure of the in-line gate valve prevented the determination of the travel time between stations 2 and 3. Station 1 was not included for tests 3 and 4, conducted on the 15th August 2003, and so the travel time between stations 1 and 2 could not be determined. Station 3 was not included for tests 1 and 2, conducted on the 26th August 2003, and so the travel time between stations 2 and 3 could not be determined. Table W-5 shows the wavefront travel times between station 1 and the "T" intersection, station 3 and the dead end, and between station 3 and the "T" intersection, for the specified tests.

Table W-4 – Wavefront travel times for the SJTP from stations 1 to 2, stations 2 to 3 and station 2 to "T" intersection

Test Date	Station 1 – Station 2			Station 2 – Station 3			Station 2 – "T" end		
	Min	Max	Avg	Min	Max	Avg	Min	Max	Avg
T1 23/7/03	0.072	0.076	0.074	0.188	0.196	0.192	0.476	0.504	0.490
T2 23/7/03	0.072	0.076	0.074	0.188	0.196	0.192	0.476	0.504	0.490
T3 23/7/03	0.072	0.076	0.074	0.188	0.196	0.192	0.476	0.504	0.490
T4 23/7/03	0.074	0.078	0.076	NA	NA	NA	0.476	0.488	0.482
T5 23/7/03	0.074	0.078	0.076	NA	NA	NA	0.476	0.488	0.482
T3 15/8/03	NA	NA	NA	0.188	0.196	0.192	0.476	0.486	0.481
T4 15/8/03	NA	NA	NA	0.188	0.196	0.192	0.476	0.486	0.481
T1 26/8/03	0.072	0.074	0.073	NA	NA	NA	0.476	0.502	0.489
T2 26/8/03	0.072	0.074	0.073	NA	NA	NA	0.476	0.502	0.489
Average	0.073	0.076	0.074	0.188	0.196	0.192	0.476	0.496	0.486

Table W-5 – Wavefront travel times for the SJTP from station 1 to "T" intersection, station 3 to dead end and station 3 to "T" intersection

Test Date	Station 1 – "T" end			Station 3 – Dead end			Station 3 – "T" end		
	Min	Max	Avg	Min	Max	Avg	Min	Max	Avg
T1 23/7/03	0.344	0.364	0.354	0.158	0.160	0.159	0.472	0.494	0.483
T2 23/7/03	0.344	0.364	0.354	0.158	0.160	0.159	0.472	0.494	0.483
T3 23/7/03	0.344	0.364	0.354	0.158	0.160	0.159	0.472	0.494	0.483
T4 23/7/03	0.342	0.350	0.346	NA	NA	NA	NA	NA	NA
T5 23/7/03	0.342	0.350	0.346	NA	NA	NA	NA	NA	NA
T3 15/8/03	NA	NA	NA	0.166	0.172	0.169	NA	NA	NA
T4 15/8/03	NA	NA	NA	0.166	0.172	0.169	NA	NA	NA
T1 26/8/03	0.344	0.364	0.354	NA	NA	NA	NA	NA	NA
T2 26/8/03	0.344	0.364	0.354	NA	NA	NA	NA	NA	NA
Average	0.343	0.360	0.352	0.161	0.165	0.163	0.472	0.494	0.483

Table W-6 shows the wavefront travel times between station 1 and the closed in-line gate valve, and between station 2 and the closed in-line gate valve, for tests 4 and 5, conducted on the 23rd July 2003. The average travel time between midpoints on the respective wavefronts has been used to determine a representative wave speed for different sections of the SJTP. These average times of travel, for the relevant incident

and reflected wavefronts, between the measurement stations, and to the dead and "T" intersection ends of the SJTP, are summarised in Table W-7 below.

Table W-6 – Wavefront travel times for the SJTP for stations 1 and 2 to closed in-line gate valve

Test Date	Station 1 – GV			Station 2 – GV		
	Min	Max	Avg	Min	Max	Avg
T4 23/7/03	0.190	0.196	0.193	0.190	0.198	0.194
T5 23/7/03	0.190	0.196	0.193	0.190	0.198	0.194
Average	0.190	0.196	0.193	0.190	0.198	0.194

Table W-7 – Measured wave speed variation along the SJTP

Section of pipeline	Distance wave travels (m)	Average time wave travels (s)	Wave speed (m/s)
Station 2 – Station 1	78.8	0.074	1061
Station 2 – Station 3	209.7	0.192	1092
Station 2 – tee end	560.0	0.486	1152
Station 1 – tee end	389.4	0.352	1106
Station 3 – dead end	183.2	0.163	1124
Station 3 – tee end	560.0	0.483	1159
Station 2 – GV	217.8	0.194	1123
Station 1 – GV	217.8	0.193	1129

Inferred wall condition for the SJTP

The speed of propagation of a wavefront along the Saint Johns Terrace Pipeline (SJTP) is governed primarily by the thickness of pipe wall material, its elastic modulus and whether any entrained air is present along the pipe. If the presence of entrained air can been discounted because of a lack of significant wavefront dispersion in measured responses, then any reduction in observed wave speed can be attributed to deterioration in the condition of the pipe wall. The deterioration may comprise a change in the thickness and/or elasticity of the material in the pipe wall.

Figure W-4 shows that, in the case of an asbestos cement (AC) pipe, it is a reduction in the elasticity of the material comprising the pipe wall, rather than the thickness, that is more likely to occur. This reduction in elasticity typically occurs when the cement in the asbestos cement matrix is leached by acidic groundwater surrounding the pipe or "soft" potable water (i.e., low pH) flowing inside the pipe.

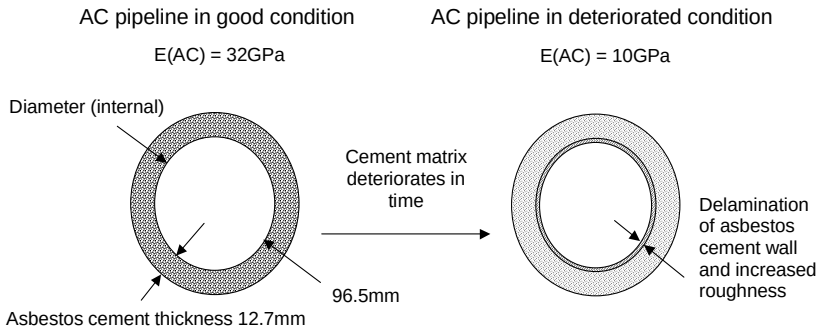


Figure W-4 – Changes in AC pipe properties affecting wave speed and roughness

As previously determined, the theoretical wave speed for the SJTP is 1199.9m/s. The average measured wave speed, determined using timing information for both incident and reflected waves, is 1118.3m/s. This represents a 6.8% reduction from the maximum theoretical wave speed. Given uncertainties regarding the restraint of the SJTP, and errors in the estimation of the wavefront travel times, it is difficult to attribute this reduction in the apparent wave speed of the SJTP to specific deterioration in the condition of the pipe wall. Furthermore, while the presence of a significant quantity of entrained air is considered unlikely, a small quantity of entrained air may also explain the reduction in apparent wave speed. Overall, it is reasonable to conclude that the pipeline wall is in good condition.

W.7 Wave speed and wall condition for the KCP

The Kookaburra Court Pipeline (KCP) was constructed in 1988 from 100mm diameter Ductile Iron Cement Mortar Lined (DICL) pipe with flexible spigot and socket joints at a spacing of 3m. Figure W-5 summarises the geometric and material properties of 100mm nominal diameter "Class K9" DICL pipe (manufactured in

accordance with Australian Standard AS 2280 – 1999). The thickness of the ductile iron comprising the pipe wall is 6.0mm. The thickness of the cement mortar lining is specified as 6.0mm in Australian Standard AS 2280 – 1999. Values for the elastic modulus of both the ductile iron and cement mortar lining were determined taking into account the results of material tests performed by South Australian Water Corporation personnel and other published data and information contained in the relevant Australian Standards.

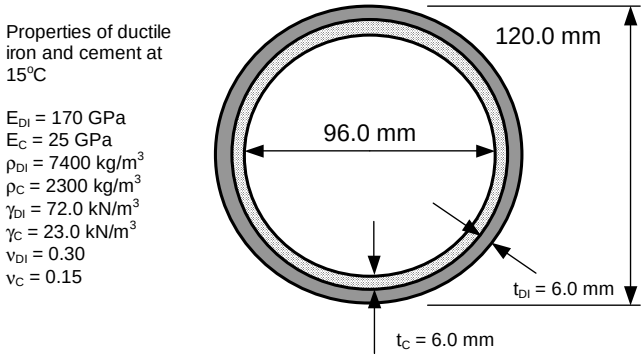


Figure W-5 – Cross section and details of the Kookaburra Court Pipeline

Theoretical estimation of wave speed for KCP

The theoretical wave speed for the ductile iron cement mortar lined Kookaburra Court Pipeline (KCP) can be estimated using the procedure, presented by Wylie and Streeter (1993), for pipe walls comprising two or more different materials. The thickness of cement mortar lining is converted to an equivalent thickness of ductile iron using the ratio between the elastic modulus for the cement mortar and ductile iron as follows:

$$t_{eqDI} = t_C \times \frac{E_C}{E_{DI}} \tag{W-3}$$

The approximate theoretical wave speed for the composite KCP can then be determined using the equation:

$$a = \sqrt{\frac{K/\rho}{1+(K/E_{DI})(D/e_{eq})c}} \tag{W-4}$$

where K is the bulk modulus of water, ρ is the density of water, E_{DI} is the elastic modulus of ductile iron, D is the internal diameter of the pipe, e_{eq} is the thickness of the equivalent ductile iron wall and c is a pipe restraint factor which, for a pipe with expansion joints, is 1

It is arguable whether the KCP is axially restrained. The pipeline comprises a multitude of 3m long segments with flexible spigot and socket joints between each segment. These joints are capable of acting as expansion joints. That said, the KCP is buried in trenches with compacted backfill and has fixed points of restraint at the "T" intersection, 2 in-line gate valves, 5 fire plugs, 15 water service connections and a dead end. It is assumed that these elements collectively comprise a considerable degree of axial restraint despite the presence of flexible joints. The restraint factor for the KCP has consequently been determined using $c = 1-\nu^2$. The approximate theoretical wave speed for the KCP is 1359.0 m/s.

Direct estimation of wave speed for the KCP

The arrival times of the incident wavefronts, and reflected wavefronts, can be determined from the measured responses of the Kookaburra Court Pipeline (KCP). While the incident wavefronts, and to a lesser extent reflected wavefronts, are relatively sharp, there is an inevitable lag between the arrival of the first and last rise in pressure associated with a wavefront (i.e., the wavefronts are not infinitely sharp and have a finite bandwidth). As for the Saint Johns Terrace Pipeline (SJTP), this variability has been dealt with by using consistent points of reference on each wavefront to determine minimum, maximum and average travel times.

Table W-8 shows the wavefront travel times between stations 1 and 2, between station 2 and the dead end, and between station 1 and the dead end, for the specified tests.

Table W-8 – Wavefront travel times for the KCP for stations 1 to 2 and stations 1 and 2 to dead end, respectively

Test Date	Station 1 – Station 2			Station 1 – Dead end			Station 2 – Dead end		
	Min	Max	Avg	Min	Max	Avg	Min	Max	Avg
T1 28/8/03	0.082	0.084	0.083	0.102	0.106	0.104	0.102	0.106	0.104
T2 28/8/03	0.082	0.084	0.083	0.102	0.106	0.104	0.102	0.106	0.104
T3 28/8/03	0.084	0.090	0.087	0.100	0.114	0.107	0.102	0.118	0.110
T4 28/8/03	0.084	0.090	0.087	0.100	0.114	0.107	0.102	0.118	0.110
Average	0.083	0.087	0.085	0.101	0.110	0.106	0.102	0.112	0.107

Table W-9 shows the wavefront travel times between stations 1 and 2 and the "T" intersection, for the specified tests. Unfortunately, the accuracy of the timing information from the arrival of reflections from the "T" intersection is reduced by cumulative dispersion along the KCP. Nevertheless, a relatively distinct change in pressure, corresponding to the reflection from the "T" intersection, occurs for both positive and negative transient tests, at stations 1 and 2, respectively.

Table W-9 – Wavefront travel times for the KCP for stations 1 and 2 to "T" intersection, respectively

Test Date	Station 1 – "T" end			Station 2 – "T" end		
	Min	Max	Avg	Min	Max	Avg
T1 28/8/03	0.382	0.396	0.389	0.532	0.548	0.540
T2 28/8/03	0.382	0.396	0.389	0.532	0.548	0.540
T3 28/8/03	0.380	0.392	0.386	0.534	0.554	0.544
T4 28/8/03	0.380	0.392	0.386	0.534	0.554	0.544
Average	0.381	0.394	0.388	0.533	0.551	0.542

The average travel time between midpoints on the respective wavefronts has been used to determine a representative wave speed for different sections of the KCP. These average times of travel, for the relevant incident and reflected wavefronts, between the measurement stations, and to the dead and "T" intersection ends of the SJTP, are summarised in Table W-10 below.

Table W-10 – Measured wave speed variation along the KCP

Section of pipeline	Distance wave travels (m)	Average time wave travels (s)	Wave speed (m/s)
Station 2 – Station 1	93.1	0.085	1095
Station 2 – dead end	120.2	0.107	1123
Station 1 – dead end	120.2	0.106	1134
Station 2 – tee end	638.5	0.542	1178
Station 1 – tee end	450.0	0.388	1160

Inferred wall condition for the KCP

Figure W-6 shows that, in the case of a ductile iron cement mortar lined (DICL) pipe, loss of cement mortar lining and consequential corrosion and/or tuberculation, rather than any specific change to elasticity, are more likely to constitute deterioration. The rate of spalling of cement mortar lining is dependent upon the quality of the cement mortar and manufacturing process. It is also dependent upon the degree of damage during installation. Once the cement mortar lining is lost, and the ductile iron is exposed, corrosion begins. In systems with "hard" potable water (i.e., high pH), tuberculation often accompanies the corrosion process.

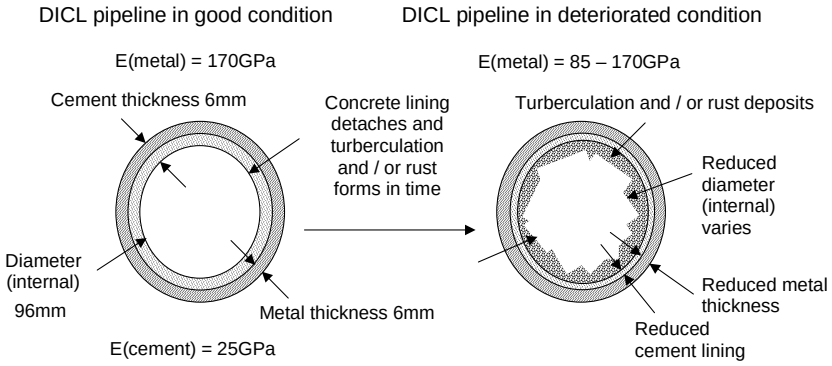


Figure W-6 – Changes in DICL pipe properties affecting wave speed and roughness

As previously determined, the theoretical wave speed for the Kookaburra Court Pipeline (KCP) is 1359.0m/s. The average measured wave speed, determined using

timing information for both incident and reflected waves, is 1138.0m/s. This represents a 16.3% reduction from the maximum theoretical wave speed. Given uncertainties regarding the restraint of the KCP, and errors in the estimation of the wavefront travel times, the certainty with which this reduction in the apparent wave speed of the KCP can be attributed to deterioration in the condition of the pipe wall is reduced. That said, given a history of complaints regarding water quality, made by residents along the section between measurement stations 1 and 2, there is a circumstantial case suggesting that the condition of the pipe wall may have deteriorated along this section of the KCP.

W.8 Wave speed and wall condition for the FSP

The Foster Street Pipeline (FSP) was constructed in 1932 from 80mm diameter Cast Iron Cement Mortar Lined (CICL) pipe with spigot and socket lead joints at a spacing of 3m. These joints were sealed by pouring molten lead into the gap between the spigot and socket (the interference fit between the spigot and socket reduced the intrusion of the lead into the pipe). That said, lead intrusion is commonly observed where these joints have been constructed. Figure W-7 summarises the geometric and material properties of 80mm nominal diameter CICL pipe. No Australian Standard relevant to the applicable construction period could be identified, and so a samples of the FSP were taken. These enabled the direct measurement of the thickness of the cast iron as 10.0mm. The thickness of the cement mortar lining was measured as 4.0mm.

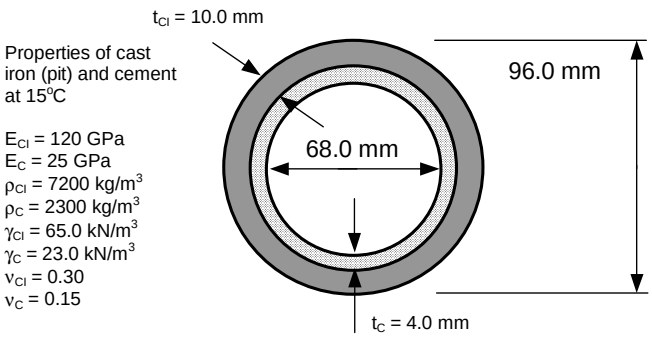


Figure W-7 – Cross section and material details for the FSP in its original condition

Theoretical estimation of wave speed for the FSP

The theoretical wave speed for the cast iron cement mortar lined Foster Street Pipeline (FSP) can be estimated, as for the Kookaburra Court Pipeline (KCP), using the procedure, presented by Wylie and Streeter (1993), for pipe walls comprising two or more different materials. The approximate theoretical wave speed for the composite FSP can be determined, once the cement mortar lining has been converted to an equivalent thickness of cast iron, using:

$$a = \sqrt{\frac{K/\rho}{1+(K/E_{CI})(D/e_{eq})c}} \qquad \text{(W-5)}$$

where K is the bulk modulus of water, ρ is the density of water, E_{CI} is the elastic modulus of cast iron, D is the internal diameter of the pipe, e_{eq} is the thickness of the equivalent cast iron wall and c is a pipe restraint factor which, for a pipe with expansion joints, is 1

In contrast to the joints for the other distribution pipes, the lead joints for the FSP are rigid and the restraint factor for the FSP can be unequivocally determined using $c = 1-\nu^2$. The theoretical wave speed for the FSP is 1394.2 m/s.

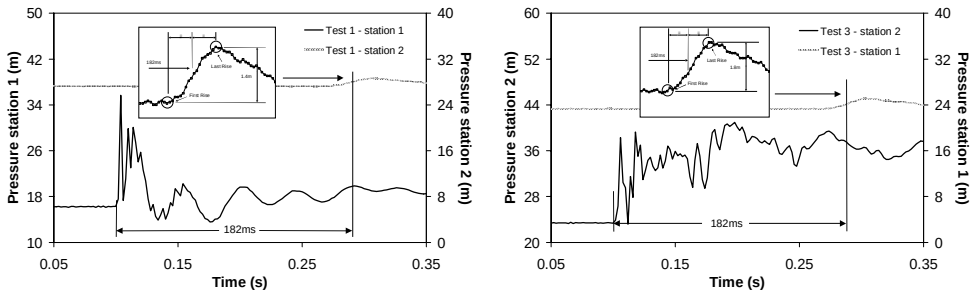
Direct estimation of wave speed for FSP

The arrival times of the incident wavefronts can be determined from the measured responses of the Foster Street Pipeline (FSP). Furthermore, the dispersion of the transmitted wavefront, as measured at the stations remote from the transient generator, can be taken into account by using consistent points of reference on each wavefront. As for the other distribution pipelines, a minimum travel time is determined by taking the difference between the time at the base (first rise) of the incident and transmitted wavefronts. Similarly, a maximum travel time is determined by taking the difference between the time at the top (last rise) of the incident and transmitted wavefronts. Average travel times can then be determined as listed in Table W-11.

Table W-11 – Wavefront travel times for the FSP for stations 1 to 2 and vice versa for tests 1, 2, 3 and 4

Test Date	Station 1 – Station 2		
	Min	Max	Avg
T1 16/7/03	0.170	0.194	0.182
T2 16/7/03	0.170	0.194	0.182
T3 7/8/03	0.168	0.196	0.182
T4 7/8/03	0.168	0.196	0.182
Average	0.169	0.195	0.182

Figures W-8 and W-9 show that approximately the same travel time is obtained regardless of whether the transient generator is located at stations 1 or 2, and the time is measured for the transmitted wavefront to reach stations 2 or 1, respectively. The insets show dispersion of the transmitted wavefronts. The directly measured wave speed between stations 1 and 2 is 960 m/s regardless of the direction of travel of the transmitted wavefront. This value is 31.1% lower than the theoretically estimated value of 1394.2 m/s that was based on the measured geometry of samples of the FSP in good condition.



Figures W-8 and W-9 – Direct measurement of wave speed for tests 1 and 3 on the Foster Street Pipeline

Inferred wall condition for the FSP

Figure W-10 shows that, using the transient model developed in Chapter 13, approximately 0.04% of entrained air is required to reduce the apparent wave speed to

the observed value. This percentage is inconsistent with the observed quantities of entrained air. All the fire plugs along the Foster Street Pipeline (FSP) were flushed on each test date and no significant quantity of entrained air was released.

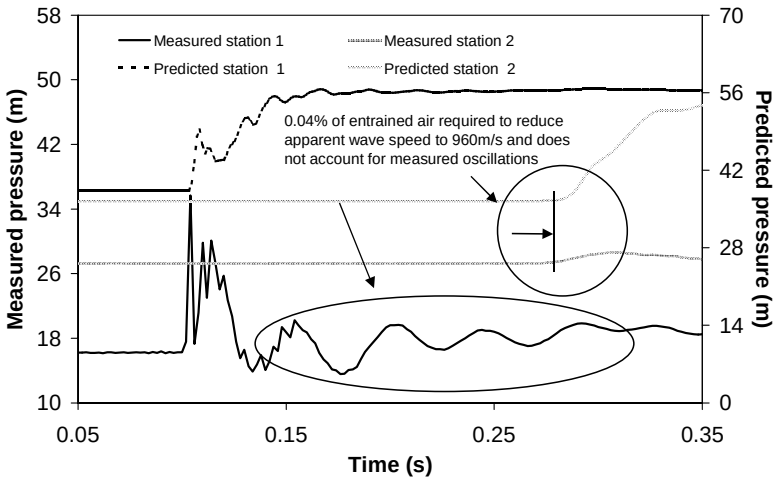


Figure W-10 – Measured and predicted responses, determined with 0.04% of entrained air, for test 1, at stations 1 and 2

It is speculated that the observed reduction in wave speed is related to deterioration of the FSP as cast iron is converted to corrosion product via graphitisation and/or the formation of tuberculation. As this conversion proceeds, the thickness of the wall of the FSP reduces as does the wave speed for the affected section. Another explanation for the apparent reduction in wave speed is unsteady inertia. It is known that there are multiple locations, within the extent of the extended blockages documented in Chapter 13, where the severity of constriction is sufficient for unsteady inertial effects. Unfortunately, the tests conducted on the FSP cannot be used to further resolve whether direct pipe wall deterioration, unsteady inertia or a combination of both give rise to the reduction, or apparent reduction, in wave speed and further tests need to be undertaken (either in the laboratory or in the field).

Appendix X

Method of Characteristics (MOC) solution of governing equations and implicit implementation of miscellaneous algorithms

X.1 Method of Characteristics (MOC)

The wave nature of the hyperbolic partial differential equations used to perform transient calculations promotes solution along specific lines called characteristics. The Method of Characteristics (MOC) transforms the partial differential equations into ordinary differential equations that apply along the characteristic lines. The hyperbolic partial differential equations are linearly combined and reduced to extract the directional (ordinary) derivates for flow and pressure. This reduction is valid provided the derivative of displacement with respect to time is given by:

$$\frac{dx}{dt} = V \pm a \tag{X-1}$$

where V is velocity and a is wave speed

In pipelines conveying water the wave speed is typically three orders of magnitude larger than velocity allowing the further approximation:

$$\frac{dx}{dt} = \pm a \tag{X-2}$$

This equation defines the characteristic lines (positive and negative) along which flow and pressure can be differentiated.

Two compatibility equations emerge from the process of linearly combining the governing equations and reducing them to ordinary differential equations valid along characteristic lines:

$$C^+ : \frac{a}{gA}\frac{dQ}{dt} + \frac{dH}{dt} + \frac{fQ|Q|a}{2gDA^2} = 0 \quad \text{along } \frac{dx}{dt} = +a \tag{X-3}$$

$$C^- : \frac{a}{gA}\frac{dQ}{dt} - \frac{dH}{dt} + \frac{fQ|Q|a}{2gDA^2} = 0 \quad \text{along } \frac{dx}{dt} = -a \tag{X-4}$$

These equations can be used to solve for flow and pressure in a displacement versus time plane as shown in Figure X-1:

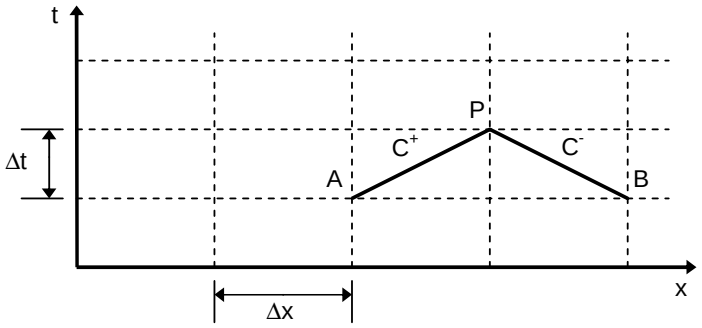


Figure x-1 – Solution of compatibility equations along characteristic lines in a Method of Characteristics (MOC) grid

The compatibility equations can be solved along their respective characteristic lines by integration. The quasi-steady friction term is obtained using a finite difference approximation for a defined grid spacing $dx = adt$:

$$C^+ : \frac{a}{gA}(Q_P - Q_A) + (H_P - H_A) + \frac{fQ_{AP}|Q_{AP}|\Delta x}{2gDA^2} = 0 \tag{X-5}$$

$$C^- : \frac{a}{gA}(Q_P - Q_B) - (H_P - H_B) + \frac{fQ_{BP}|Q_{BP}|\Delta x}{2gDA^2} = 0 \tag{X-6}$$

Integration of the friction term between two points in the characteristic grid requires an approximation of the flow between those points. Traditionally, this quasi-steady friction term has been approximated using the flow from the previous time step:

$$C^+ : \frac{fQ_A|Q_A|\Delta x}{2gDA^2} \quad \text{and} \quad C^- : \frac{fQ_B|Q_B|\Delta x}{2gDA^2} \tag{X-7}$$

However, for high velocity cases, a non-linear approximation, effectively averaging the flows at either end of the characteristic lines, may be required and solved by iteration:

$$C^+ : \frac{f}{2gDA^2}\frac{(Q_A+Q_P)}{2}\frac{|Q_A+Q_P|\Delta x}{2} \quad \text{and} \quad C^- : \frac{f}{2gDA^2}\frac{(Q_B+Q_P)}{2}\frac{|Q_B+Q_P|\Delta x}{2} \tag{X-8}$$

X.2 Interpolation schemes

The Method of Characteristics (MOC) (and other numerical schemes) requires a common time step. However, in real pipe networks, wave travel times vary for different computational units that have different wave speeds or variable lengths. In this situation, the time step dt is chosen as the shortest wave travel time for all of the computational units.

The Courant number is defined as $C_r = adt/dx$. For the shortest computational unit the Courant number is 1. Interpolation is required when the Courant number for other computational units, which either have slower wave speeds or greater lengths, is less than 1. The minimum Courant number for a pipe network can be increased by careful discretisation of pipes and selection of a small common time step. However, longer computational times are required for finer discretisations.

The three main categories of interpolation scheme are wave speed adjustment, spaceline and timeline interpolation. Wave speed adjustment schemes have the advantage that they preserve the total energy of a system. However, wave speed adjustment introduces significant wavefront dispersion (this effect becomes more severe as wavefronts become steeper). Spaceline interpolation extends the characteristic line back from a point at which flow and pressure are unknown to a point on the spaceline for the previous time step (at which flow and pressure are

known). Interpolation is then performed to obtain flow and pressure at the point at which the characteristic line intersects the spaceline. In contrast, timeline interpolation extends the characteristic line back from a point at which flow and pressure are unknown to a point on the timeline for the previous time step. Interpolation is then performed to obtain flow and pressure at the point at which the characteristic line intersects the timeline.

Spaceline and timeline interpolation can be performed linearly or using higher order interpolation polynomials and compact schemes. Higher order interpolation polynomials improve the approximation of the movement of wavefronts. Compact interpolations use spatial derivatives of flow and pressure to construct higher order interpolation schemes without the need to use extra points along a spaceline outside the computational unit. Compact schemes are not required for timeline interpolation because there is no need to use points along a timeline outside the computational unit.

X.3 Implicit MOC Solution Method

An implicit Method of Characteristics (MOC) solution solves for unknown flows and pressures, typically along the length of a pipe or pipes comprising a network, using simultaneous equations (as opposed to solving for the unknowns one point at a time in an explicit MOC solution). The implicit solution scheme is formed from the compatibility equations and boundary conditions relating flow and pressure at a known timeline to flow and head at an unknown timeline. The set of equations forms a simultaneous system that can be expressed in matrix form as:

$$[M]\{v^*\} = \{R\} \tag{X-9}$$

where matrix M contains coefficients that are multiplied by the unknowns, vector v^* contains the unknown conditions (flow and pressure) at a particular time step and vector R contains all constants including known boundary conditions

Non-linear equations must first be linearised, solved and the non-linear terms updated in an iterative procedure until the solutions converge. Although less efficient than an

explicit MOC solution, an implicit MOC solution reduces the complexity of the equations to be solved in network situations.

X.4 Implicit implementation of miscellaneous equations

Demands and leakage

When including demands or leakage in an implicit solution scheme, quasi-steady equations approximating the behaviour of each orifice are added to the system of simultaneous equations formed using the compatibility equations and system boundary conditions:

$$Q_{UP} - Q_{DN} - \left(C_d A_L \sqrt{\frac{2g}{H'}} \right) H = 0 \qquad \text{(X-10)}$$

where Q_{UP} and Q_{DN} are the flows upstream and downstream of the leak, C_d is the coefficient of discharge for the leak orifice and A_L is the area of the leak orifice

Continuity of flow, at the node at which the leak is located, is maintained. The non-linear orifice equation is linearised by introducing the variable H'. H' is the head at the orifice from the previous iteration in the implicit solution process and is updated when a new value of H is determined. This iterative process is continued until the values of H' and H converge.

Discrete air pockets and entrained air

In an explicit solution scheme, the two compatibility equations from adjacent sub-pipe units, the ideal gas equation and the integrated expression representing the continuity of flow in the pipe upstream and downstream of a discrete air pocket are combined to form a single non-linear equation that can then be solved using, for example, the Newton-Raphson root-finding technique (as explained in Appendix O). When introducing a discrete air pocket or entrained air to an implicit solution scheme, the ideal gas and continuity equations applicable to each air pocket(s) are added to the

system of simultaneous equations formed using the compatibility equations and system boundary conditions. The integral form of the equation describing the continuity of flow upstream and downstream of each air pocket is combined with the ideal gas equation, to express the volume of the air pocket in terms of the pressure at the current time step, and form a non-linear equation describing the behaviour of the air pocket:

$$V_0\left[\frac{H_0+H_{atm}}{H_p+H_{atm}}\right]^{1/n} = V_p^{'} + \left[\psi\left(Q_{pdn}-Q_{pup}\right)+(1-\psi)\left(Q_{pdn}^{'}-Q_{pup}^{'}\right)\right]\Delta t \qquad \text{(X-11)}$$

upon the substitution of the ideal gas equation into the integral form of the continuity equation.

This non-linear equation is linearised by performing a Taylor's formula expansion for the left hand side term. This term includes the pressure at the current time step in a non-linear exponential relationship. The expansion results in an expression for the volume of the air pocket that is linear in terms of the pressure at the current time step:

$$V_0\left[H_0+H_{atm}\right]^{1/n}\left[H_p^{'}+H_{atm}\right]^{-(1/n)} - \frac{V_0}{n}\left[H_0+H_{atm}\right]^{1/n}\left[H_p^{'}+H_{atm}\right]^{-(1/n)-1}\left[H_p-H_p^{'}\right]-V_p^{'}$$

$$-\left[\psi\left(Q_{pdn}-Q_{pup}\right)+(1-\psi)\left(Q_{pdn}^{'}-Q_{pup}^{'}\right)\right]\Delta t = 0 \qquad \text{(X-12)}$$

$Q_{up}^{'}$ and $Q_{dn}^{'}$ are the flows upstream and downstream of the air pocket from the previous time step. Q_{up} and Q_{dn} are determined in a successive substitution process used to solve the system of simultaneous equations describing the entire pipe network. The variable H'_p is introduced in the linearised calculation of the volume of the air pocket(s) at the current time step. H'_p is the pressure at the air pocket(s) from the previous iteration in the implicit solution process and is updated when a new value of H_p is determined. This iterative process is continued until the values of H'_p and H_p converge.

Unsteady friction and viscoelasticity

Unsteady friction and viscoelastic effects can be incorporated into the ordinary differential form of the compatibility equations by including an unsteady friction component in the friction term and adding a term for viscoelastic pipe wall deformation:

$$\frac{dH}{dt} \pm \frac{a}{gA}\frac{dQ}{dt} \pm ah_f + \frac{2a^2}{g}\left(\frac{\partial \varepsilon_r}{\partial t}\right) = 0 \qquad \text{(X-13)}$$

Equation (X-13) can be integrated along positive and negative characteristic lines to obtain:

$$\begin{aligned}&\left[H(x,t) - H(x \mp \Delta x, t - \Delta t)\right] \pm \frac{a}{gA}\left[Q(x,t) - Q(x \mp \Delta x, t - \Delta t)\right] \pm \\ &a\Delta h_f + \frac{2a^2\Delta t}{g}\left(\frac{\partial \varepsilon_r}{\partial t}\right) = 0\end{aligned} \qquad \text{(X-14)}$$

The quasi-steady component of the friction term h_f is determined using a non-linear approximation that averages flow at either end of the characteristic lines (solvable only by iteration). The unsteady component is calculated using an efficient recursive approximation, and flow information one and two time steps back from the current time, to estimate changes (refer to Appendix E). This approach introduces an additional approximation in the calculation of the unsteady friction component but allows the unsteady friction component to be treated as known for the current time step. The error introduced by this approximation is reduced if fine discretisations and small time steps are used. Similarly, the viscoelastic component is calculated using an efficient recursive approximation and pressure information one and two time steps back from the current time.

Unsteady minor losses

When introducing minor loss elements to an implicit solution scheme, equations for each minor loss element are added to the system of simultaneous equations formed

using the compatibility equations and system boundary conditions. For example, continuity and quasi-steady equations approximating the behaviour of an in-line orifice or valve can be readily included for solution. Continuity of flow through the in-line orifice or valve is established by including an additional equation:

$$Q_{UP} - Q_{DN} = 0 \tag{X-15}$$

where Q_{UP} and Q_{DN} are the flows upstream and downstream of the in-line orifice or valve, respectively

The non-linear orifice or valve equation is linearised by introducing the variable Q'. Q' is the flow through the orifice or valve from the previous iteration in the implicit solution process and is updated when a new value of Q is determined. This iterative process is continued until the values of Q' and Q converge:

$$Q\left|Q'\right| - (\tau C_v)^2 [H_{UP} - H_{DN}] = 0 \tag{X-16}$$

where Q is Q_{UP} or Q_{DN}, Q' is Q'_{UP} or Q'_{DN}, from the previous iteration, τ and C_v are dimensionless opening and reference condition coefficients respectively, and H_{UP} and H_{DN} are the pressures upstream and downstream of the in-line orifice or valve, respectively

Appendix Y

Limits to steady state C-Factor blockage detection

Y.1 Limitations to the characterisation of discrete blockages

Results of steady state pressure tests

In the case of the Saint Johns Terrace Pipeline (SJTP), the nearest available access points to the in-line gate valve, used to simulate discrete blockage, are the fire plugs at stations 2 and 3. If the HGL of the SJTP is considered as a whole, then a kink can be identified, when discrete blockage is introduced to the pipeline, and this kink occurs between stations 2 and 3. Figure Y-1 illustrates the effect of the discrete blockages for tests 5 and 13, conducted on the 15th August 2003, upon the HGL of the SJTP between stations 2 and 3. The discrete blockages for tests 5 and 13 are equivalent to constrictions in the SJTP with openings of 19.1mm and 31.5mm diameter, respectively.

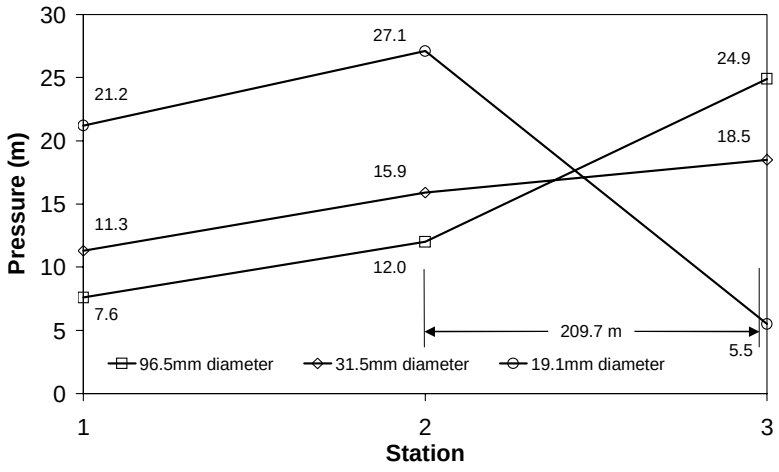


Figure Y-1 – Effect of discrete blockages upon the steady state pressures along the SJTP, for tests 5 and 13, conducted on 15th August 2003, at stations 1, 2 and 3

Figures Y-2 and Y-3 show the variation of the predicted versus measured steady state pressure as the location of the discrete blockage, formed with the in-line gate valve, is moved from 60 nodes upstream to 60 nodes downstream of the "true" blockage location, for tests 5 and 13, conducted on the 15th August 2003, respectively. Figure Y-2 shows that the predicted steady state pressures for test 5, measured at stations 2 and 3, are insensitive to the location of the blockage. Figure Y-3 shows the predicted steady state pressures for test 13, measured at stations 2 and 3, are similarly insensitive to the location of the blockage. That is, the steady state information confirming that there is a blockage between stations 2 and 3, cannot be used to locate or characterise the blockage between stations 2 and 3 (as either discrete or extended).

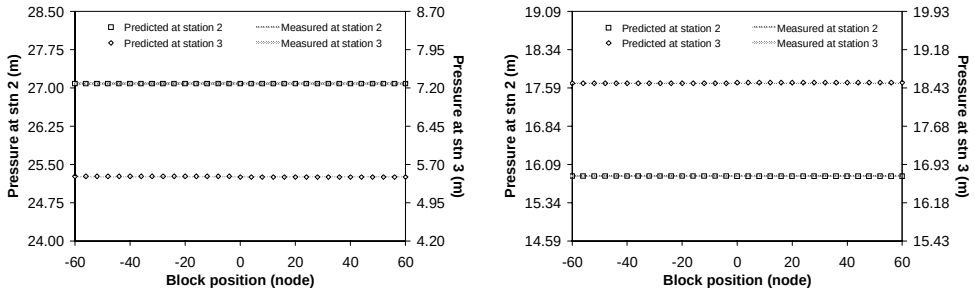


Figure Y-2 and Y-3 – Predicted steady versus average measured pressure at stations 2 and 3, for tests 5 and 13, conducted on the 15th August 2003, respectively

Results of steady state inverse analysis

Instead of using the entire transient response, inverse analysis can be limited to the time before the induction of the transient such that only steady state information is analysed. This provides a means by which the sensitivity of the fit between the measured and predicted steady state pressures can be numerically gauged as the position of the discrete blockage is progressively moved from 60 nodes upstream to 60 nodes downstream of the "true" blockage (i.e., the location of the in-line gate valve). Figure Y-4 shows the variation of the objective functions obtained for tests 5 and 13 conducted on the 15th August 2003. The values of the objective functions are insensitive to the location of the blockage as it is moved upstream and downstream of the position of the "true" blockage. The results confirm that although a section of

pipeline with pressure loss can be identified between measurement stations 2 and 3, neither the precise location of the blockage, nor information regarding its nature, can be determined using steady state inverse analysis.

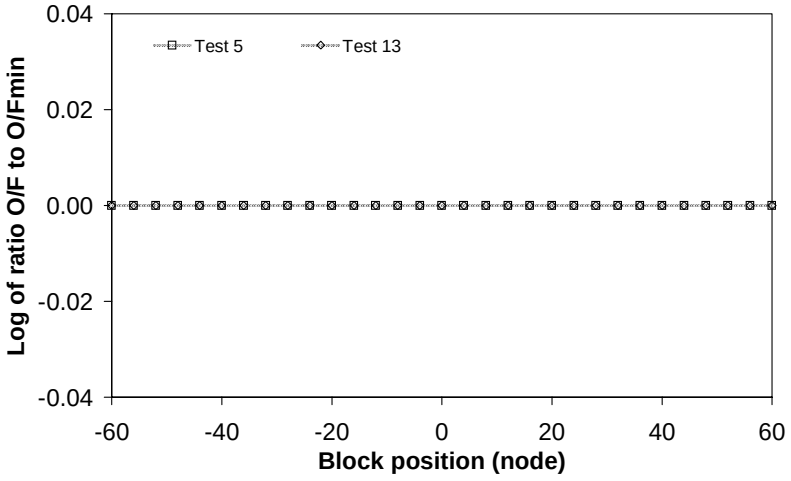


Figure Y-4 – Objective function versus block position when performing steady state inverse analysis for tests 5 and 13

Y.2 Limitations to the characterisation of extended blockages

The steady state pressure along the Foster Street Pipeline (FSP) was measured at the four locations prior to conducting transient tests 1 and 2 on the 16th July 2003. These steady state pressures, together with comparative pressures obtained using a steady state model, based on the transient model developed in Chapter 13, with no blockage and uniform roughness values of 1mm and 5mm, are shown in Figure Y-5.

If the HGL of the FSP is considered as a whole, then kinks can be discerned at each of the available measurement access points indicating that significant pressure loss occurs along each section of the FSP. The highest head loss occurs along the section of pipeline between measurement points 3 and 4 and it is this section to which the available physical information summarised in Chapter 13 applies (and it was this section that was replaced in April 2005). That said, there is significant pressure loss along the section of FSP that has not been replaced and this is indicative of

tuberculation and blockage forming along that section of pipeline. While the presence of potential blockage has been confirmed using steady state pressure and flow tests, it is not possible to ascertain the location of the blockage(s), or further information regarding their nature (i.e., discrete or extended), using a steady state approach, beyond the resolution of the nearest adjacent fireplugs.

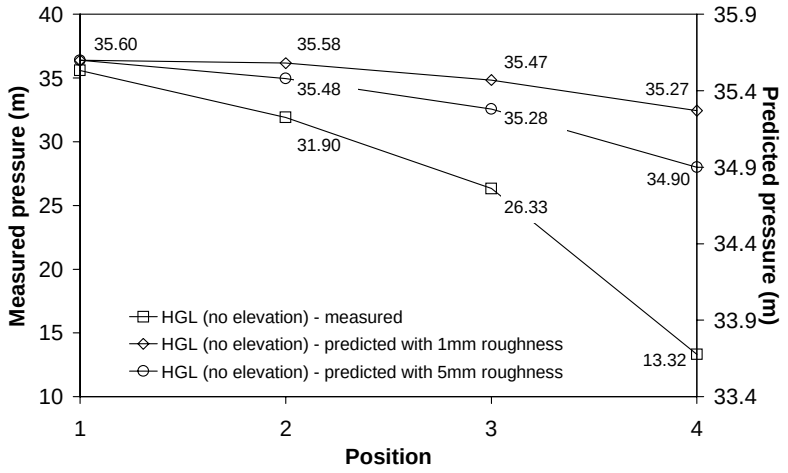


Figure Y-5 – Measured and predicted (assumed roughness with no extended blockage) steady state HGLs for test 1 conducted on the 16th July 2003

Appendix Z

Limits to steady state leak detection

Z.1 Results of steady state pressure tests

Figure Z-1 shows the variation of predicted versus measured steady state pressure as the location of the leak, comprising the 10mm orifice at the end of the standpipe installed on a fire plug, is moved from 60 nodes upstream to 60 nodes downstream of the "true" leak location, for test 10, conducted on the 28th August 2003. The predicted steady state pressure, at station 2, is insensitive to the location of the leak. That is, the steady state information confirming that there is a leak along the KCP cannot be used to locate the leak.

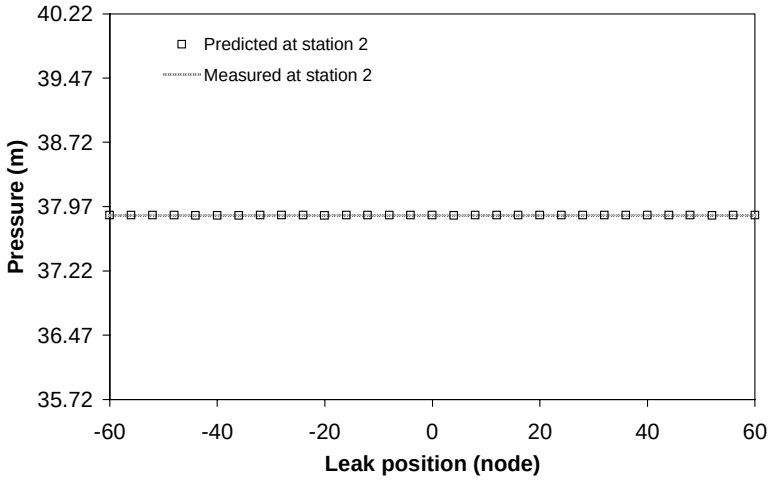


Figure Z-1 – Predicted steady versus average measured pressure at station 2 for test 10 conducted on 28th August 2003

It is theoretically possible to seek to identify distinct kinks or changes in slope along the steady HGL for a pipeline, and use this information to identify a discrete location at which a leak may be located. A large leak, along a rough pipeline, will give a different predicted pressure at a particular location (say station 2) depending on how far along the pipeline the leak is located. The further upstream the leak is (i.e., the

closer it is to the "T" intersection) the smaller the pressure loss recorded at station 2 and vice versa. The measured steady state pressure at station 2 should match that predicted when the leak is correctly located. Unfortunately, the size of leak required, to increase the sensitivity of the steady response of the KCP, is far greater than the threshold of interest to United Water, and other, operators.

Z.2 Results of steady state inverse analysis

Instead of using the entire transient response, inverse analysis can be limited to the time before the induction of the transient such that only steady state information is analysed. Figure Z-2 shows the variation of the objective functions obtained for test 10 conducted on the 28th August 2003. The values of the objective functions are relatively insensitive to the location of the leak as it is moved upstream and downstream of the position of the "true" leak (standpipe with 10mm orifice).

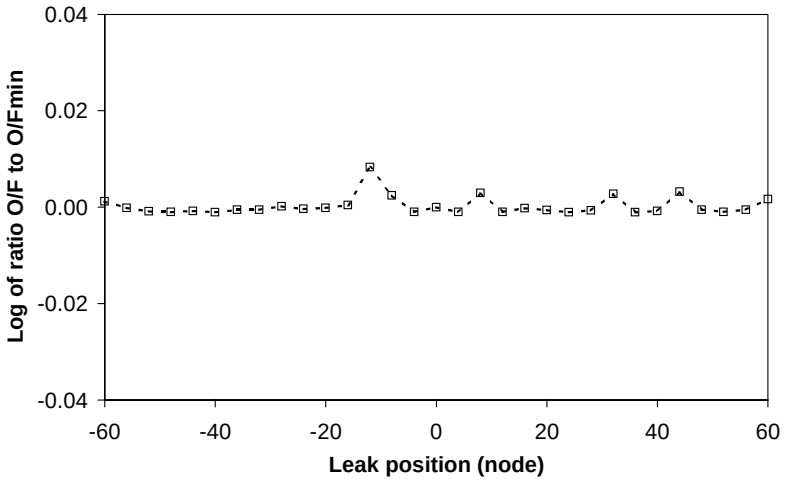


Figure Z-2 – Objective function versus leak position when performing steady state inverse analysis for test 10