NUMERICAL ANALYSIS OF FAST AND SLOW TRANSIENTS IN GAS TRANSMISSION NETWORKS

By

John David Taylor

Submitted for the Degree of
Doctor of Philosophy
at Heriot-Watt University
on Completion of Research in the
Department of Mathematics
May 1997.

I hereby declare that the work presented in this thesis was carried out by me at Heriot-Watt University, Edinburgh, except where due acknowledgement is made, and has not been submitted for any other degree.

John David Taylor (Candidate)

Dugald Duncan (Supervisor)

Date

Contents

Acknowledgements

This work was supported by EPSRC and by BG plc (formerly British Gas), who have been very generous with both their time and money.

The long list of people to whom I owe thanks begins with my supervisor Dr Dugald Duncan who has kept me on the straight and narrow for three years, if I had listened to his advice straightaway I would have been finished in half the time.

My mentors at BG, Steve Bates and Jim Mallinson, not only had many practical suggestions but corrected some of my misconceptions in the introduction.

I am grateful to Alan Clark and Derek Banks of Edinburgh Petroleum Services for their patience while I was distracted by this thesis. I am also indebted to Campbell Airlie of EPS, for the use of his figures.

It has been a pleasure to be a member of Heriot-Watt Maths Department, thanks to everyone there who has made it so enjoyable. I single out Rachel Dunwell for her friendship and her LaTeXing skills.

Finally, thanks to my family, who could never have imagined that I would avoid paying taxes for this long, and to Louise, for so many things.

Abstract

This thesis presents two mathematical models which are used to simulate large gas transmission networks. Numerical methods are discussed for each model, and their relative merits assessed. Techniques to combine the models on a single linear network are developed, and their extension to branched networks considered.

Chapter 1

Introduction

This PhD project originated as a collaboration between BG plc (formerly part of British Gas) and Heriot-Watt University under the EPSRC's CASE scheme.

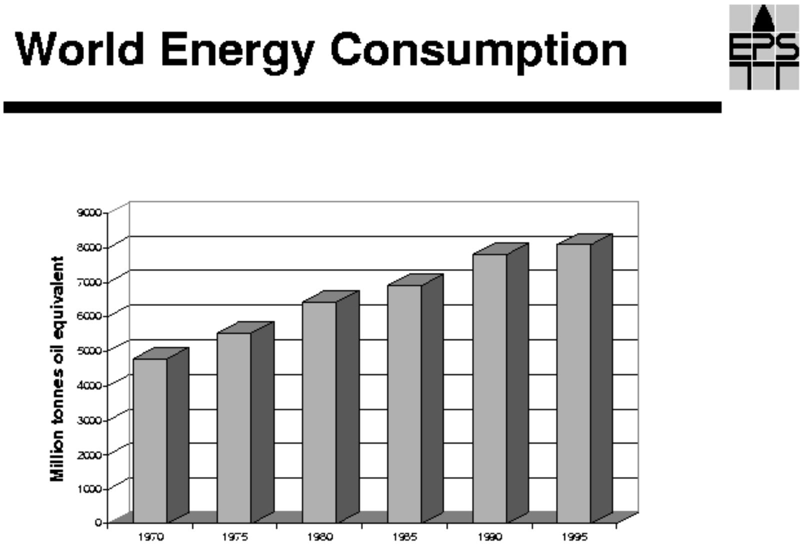
The aim was to investigate methods to combine different numerical methods for gas dynamics in large pipeline networks. This chapter begins with a general discussion of the gas industry, the rôle played by Transco, the pipeline transportation unit of BG, and their need for accurate numerical simulation of the gas transmission system. Chapter 2 surveys various mathematical models for gas flow in pipes, including the empirical corrections that are needed in practice. It focuses on two models which are applicable in very different circumstances. The first, known as the hyperbolic model, is essentially the Euler equations augmented by friction and heat transfer terms. It is suitable for the modelling of rapid changes such as those that occur at a pipe-break, but is too expensive to be used on the entire network. The second, known as the parabolic model, is a simplified set of equations which is cheaper to solve, but only accurate when changes are occurring over much longer timescales. Each model then has a chapter devoted to it, describing the properties and peculiarities of the equations and some of the numerical methods that may be employed in their solution. Chapter 5 examines the problem of combining the two models, and their numerical schemes, on a single network. The idea is to use the expensive hyperbolic method only where accuracy is required, using the cheaper parabolic method elsewhere. The problem is complicated by the need to use much smaller timesteps for the hyperbolic method. Up to this point only linear networks, that is straight pipelines, are considered. The additional difficulties in branching networks are examined in Chapter 6. Finally, Chapter 7 concludes the thesis with proposals to implement the methods in practice, further work, and other applications.

Chapters 1 - 4 introduce the problem, models and numerical schemes and review the literature. New material is also presented in these chapters, in particular the cell-edge boundary conditions in Section 3.4.2, the asymptotics in Section 4.1 and the implementation of the temperature variation in Section 4.3.2. Chapters 5 and 6 contain most of the original work with some analysis and testing of several interfacing methods and a new way of applying boundary conditions at junctions for the hyperbolic equations.

1.1 The Gas Industry

World energy consumption is rising at a rapid rate, as Figure 1.1.1 shows, almost doubling between 1970 and 1995. Almost all of this energy is provided by fossil fuels, and the fraction of this supplied as natural gas is becoming increasingly important.

The increase in energy consumption can be attributed to the massive growth in the number of cars, the growth in air travel, and the rising use of energy by the emerging market economies (EMEs). Currently the average US citizen uses the energy equivalent of 7.7 tonnes of oil per annum compared to a world average of 1.5 tonnes [Com93] so it is likely that this trend will continue in the future as the EMEs catch up with the industrialised nations. Although non-fossil fuel consumption has risen considerably since 1950, oil, gas and coal still provide most of our energy, as the chart in Figure 1.1.2 shows. Gas is starting to replace oil in the industrial and domestic sectors and is beginning to be seen as an environmentally preferable



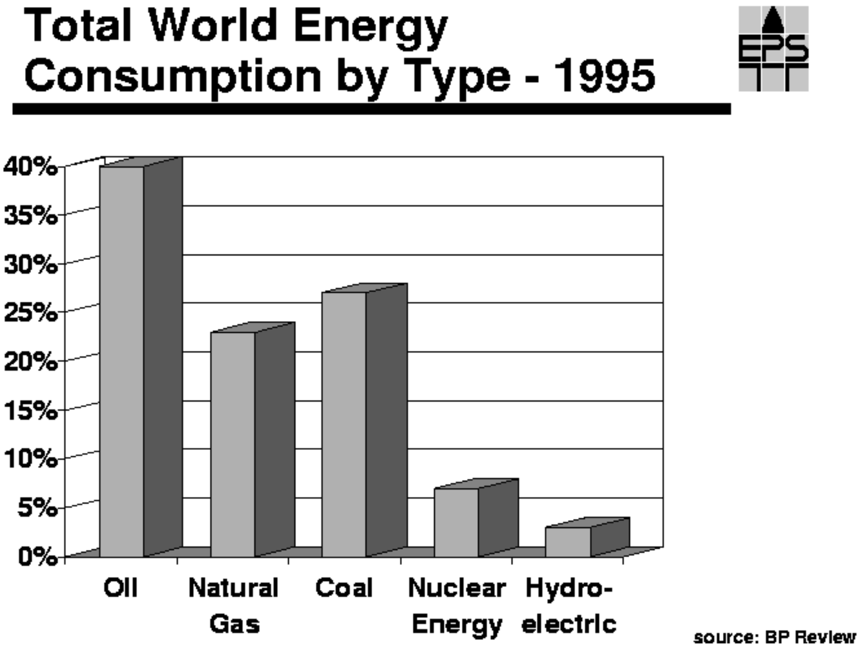
source: BP Review

Figure 1.1.1: Reproduced by permission of Edinburgh Petroleum Services Ltd.

alternative to oil, as oil was to coal in the 1950s. In Britain, a new generation of power stations is becoming more common, which use natural gas as the primary fuel. In addition, gas-fired CHP (Combined Heat and Power) units and bus and taxi fleets running on Natural Gas are increasing in popularity.

The other major influence on the increased use of gas is the change in the gas markets. While oil is a globally traded commodity, transported around the world in tankers, gas has traditionally been confined to regions where it is produced. The transportation of gas requires either the building of an expensive pipeline or its liquification into liquid natural gas which is also very costly. However, international gas transmission networks such as the US and Canada grid system, the Siberia to Western Europe line and the Euroconnector are now becoming available. This will give gas producers access to previously untapped markets and influence the way gas is managed and the price at which it is sold.

The increasing importance of natural gas is reflected in the chart of proven



source: BP Review

Figure 1.1.2: Reproduced by permission of Edinburgh Petroleum Services Ltd.

reserves in Figure 1.1.3. For example, estimates of UK Continental Shelf reserves (UKCS) have increased every year, doubling between 1980 and 1995. Figure 1.1.4 shows the locations of the known gas reserves. As can be seen, the former Soviet Union (FSU) holds the largest share, equivalent to about 70 years of production. By contrast, the USA while currently producing a similar amount to the FSU, has reserves only for a further 8.8 years. The UK produces about 3% of the world's gas, and imports slightly more.

Assuming a price of $2.50/Mscf then there remains about $12,400 billion to be extracted worldwide (compared with about $18,300 billion worth of oil.)

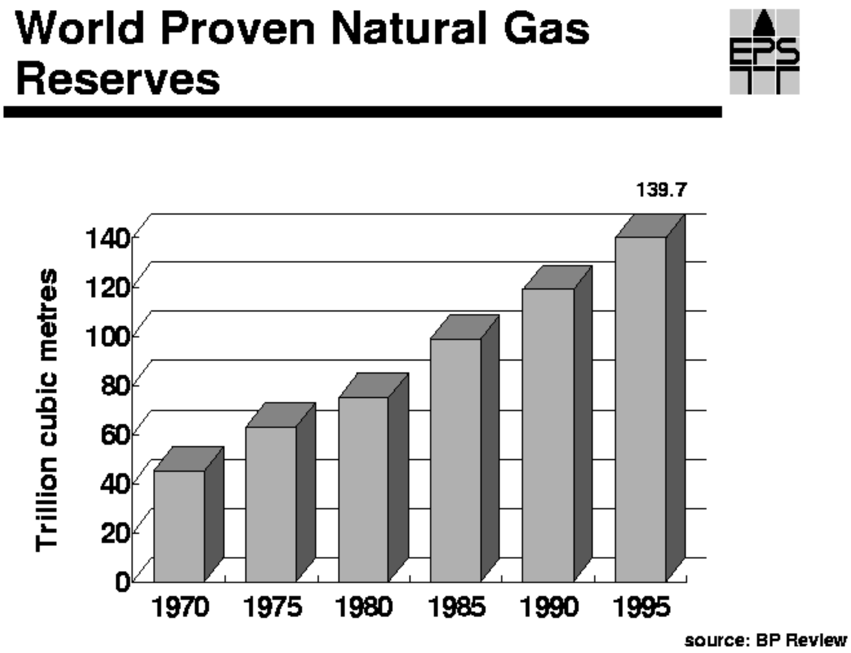


Figure 1.1.3: Reproduced by permission of Edinburgh Petroleum Services Ltd.

1.2 The Gas Industry in Britain

1.2.1 History

Britain's energy sources have changed dramatically over the last 40 years. In 1950 90% of the country's primary energy came from coal, which included the artificially manufactured 'town gas' piped in small local networks. By the mid-sixties oil had gained a 40% share of the market and it was at this time that there was the first hint of the vast hydrocarbon wealth under the North Sea. In the Netherlands the vast Groningen gas field had been discovered, and the similar geology of the North Sea led oil companies to prospect for oil off-shore. The first commercial discovery was the West Sole gas field which was found by the rig "Sea Gem" for BP in 1965. This was soon followed by the discovery of Leman, Indefatigable and Hewitt and within 10 years natural gas provided 25% of Britain's primary energy.

Unlike oil, all of the gas was controlled by a single body, the British Gas Council,

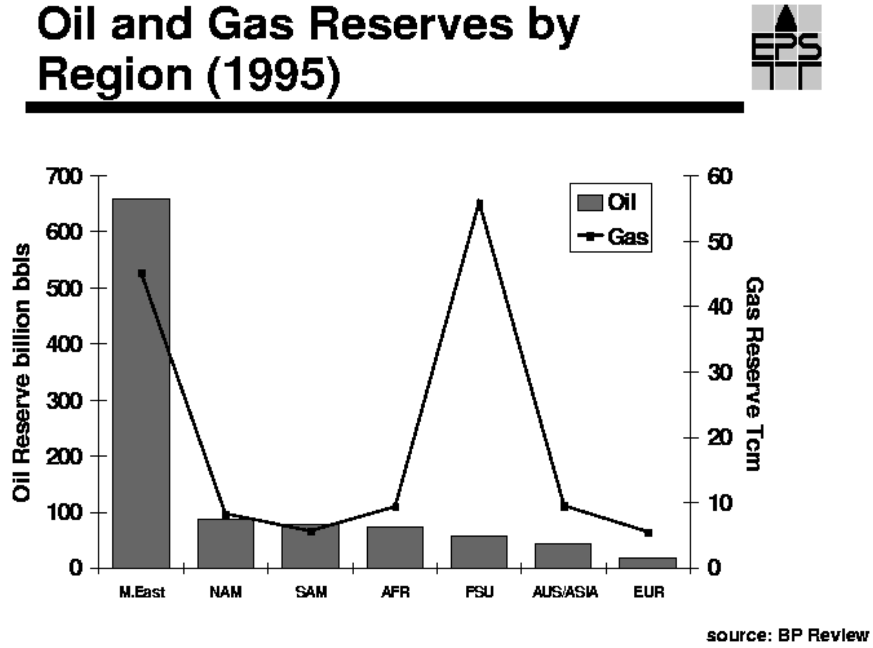


Figure 1.1.4: Reproduced by permission of Edinburgh Petroleum Services Ltd.

the forerunner of British Gas. This organisation was responsible for marketing the new fuel in the UK and constructing the pipeline network required to deliver it.

Although demand for both oil and gas is seasonal, the extra stages such as refining between the supplier and the user of oil iron out some of the fluctuations. Gas on the other hand is produced as it is required, with the off-shore platforms in effect acting as taps. It therefore requires a greater degree of control to cope with the huge variations in demand between summer and winter days.

The complication of the gas market in the UK has increased in recent years due to deregulation (see [Upt96]). This process began in 1986 with the privatisation of British Gas and has continued with the introduction of competition. Customers in the South West of England were allowed to choose their gas supplier from 1996 and this will be extended to the remainder of the country in the next few years. British Gas has now divided into two separate companies: Centrica, which competes with other gas suppliers, and BG plc which owns and operates the UK pipeline and storage system through its subsidiary, Transco. Another important factor will be the opening of the Interconnector from Bacton to Zeebrugge in 1998 which will link the UK gas market with that of continental Europe. All of these changes make the operating of the pipeline network a more complex business.

1.2.2 The National Transmission System

Transco owns and operates the pipeline system and the associated storage facilities in the UK. This includes the National Transmission System (NTS), shown in Figure 1.2.1, which is the high pressure network which transports gas for around 40 shippers. The NTS transports gas from the six shore terminals at a speed of around 25 mph and a pressure of between 40 and 75 bar. It consists of approximately 6,000 km of pipeline of between 500 mm and 1,200 mm in diameter. The pressure is maintained by about 20 compressor stations, which are industrial versions of aero engines such as the 33,000 hp Rolls Royce RB211, powered by natural gas taken from the network.

As well as the pipe network Transco operates gas storage facilities such as

- gas holders (>500, total 27.5 million cubic metres)
- salt cavities (Hornsea, East Yorkshire 189 million cubic metres)
- Liquid Natural Gas (5 sites in the UK, each 25 million cubic metres).

The largest storage site is the Rough gas field, 29 kms off the Yorkshire coastline. This is a partially depleted gas field which is now used to store up to 2.8 billion cubic metres at pressures up to 205 bar. Additionally, the pipelines themselves can be used for storage, known as linepack. At certain times of the day, when demand is high, the pressure can be allowed to drop as linepack is used, and replenished when demand is low.

The storage facilities help smooth the extreme fluctuations in demand for gas. On average, the daily demand is about 200 million cubic metres, but it reached a record 378 million cubic metres on one day in February 1996.

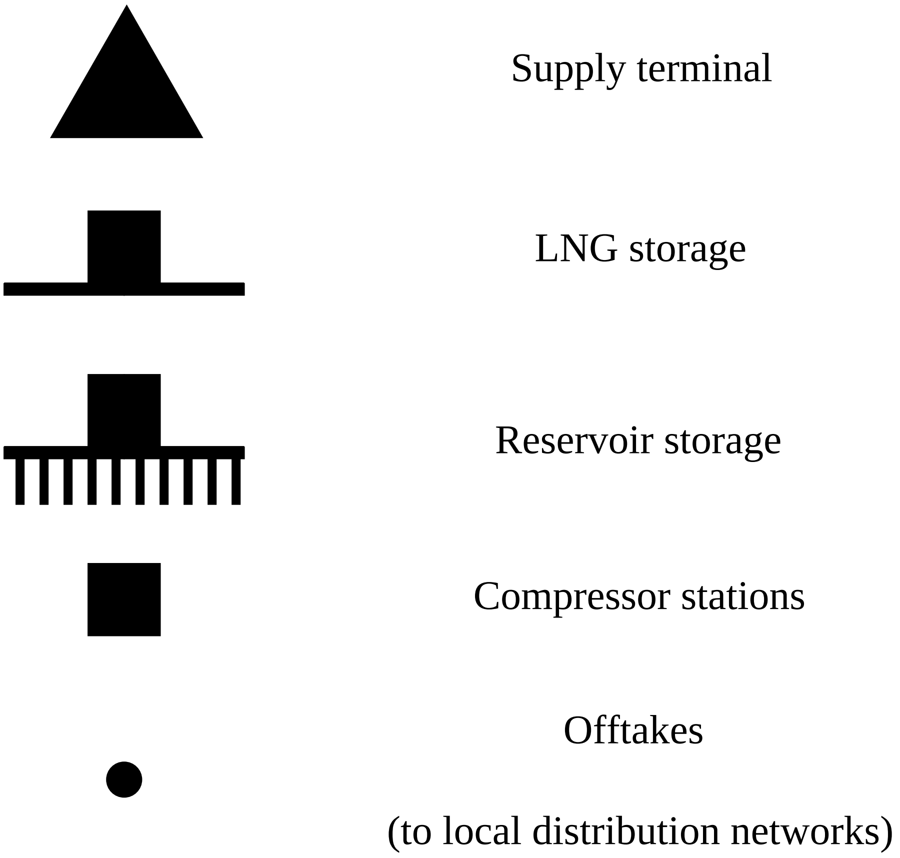


Figure 1.2.1: Key for the United Kingdom NTS, shown overleaf (reproduced by permission of BG plc)

The pipes are connected to sources, demands, machines and each other at *nodes*. Machines such as valves and compressors may be modelled by applying boundary conditions at the nodes. However, in this thesis we are only concerned with the pipe network itself.

1.2.3 The importance of simulation

The highly dynamic nature of gas supply and demand means that high quality simulation is essential. Simulation over long timescales is needed for forecasting potential shortages in the system and to make plans to deal with unexpected or expected disruption (e.g. maintenance) to the network. Accurate simulation over shorter times could yield better detection of leaks and other potential hazards such as compressor shutdowns. Pipe leakages and the unnecessary operation of compressor stations will clearly have a bearing on the efficiency of the network. Furthermore, in the new climate of competition, accurate modelling is needed to predict the operation of the network under greatly varying conditions caused by the unpredictable behaviour of the transportation requirements of shippers.

1.3 Summary

- Natural gas will be an important fuel over the next decades.
- A changing market increases the necessity for accurate mathematical models and numerical methods.
- This thesis discusses two different models with different characteristics.
- We investigate how to combine them on linear and branching networks.

Chapter 2

Mathematical models for gas flow in pipes

This chapter surveys some of the equations that have been used to model time-dependent flow in gas networks. Although numerical methods will not be described in detail until Chapter 3 a brief mention will also be made of some of the methods of solution. An excellent introduction to the subject is the classic Goldwater and Fincham paper [GF81] which will be referred to frequently throughout this chapter.

The usual starting point is the one-dimensional Euler equations,

$$\begin{aligned}
\frac{\partial(A\rho)}{\partial t} + \frac{\partial(A\rho u)}{\partial x} &= 0 \\
\frac{\partial(A\rho u)}{\partial t} + \frac{\partial(A\rho u^2)}{\partial x} + A\frac{\partial p}{\partial x} &= 0 \\
\frac{\partial(AE)}{\partial t} + \frac{\partial(A(E+p)u)}{\partial x} &= 0,
\end{aligned}$$

which describe the conservation of mass, momentum and energy. Their derivation is omitted since it may be found in any fluid dynamics textbook. Refer to Table 2.0.1 for definitions of the notation.

Goldwater and Fincham discuss the validity of a one-dimensional model and some modifications that may be used for turbulent flow. Since a typical pipe has a length of at least 1,000 times its diameter a one-dimensional model is usually justified,

p	pressure	A	pipe area	ρ	density
u	gas speed	E	energy density	t	time

Table 2.0.1: Notation

with quantities considered to be averages across the pipe cross section. The size of the network generally prohibits a multidimensional model and furthermore the limited accuracy with which measurements can be made does not justify the extra complexity.

In the form given above, the time derivatives are of the conserved quantities, while the spatial derivatives are of the fluxes*. Thus $A\rho$ is the mass per unit length, $A\rho u$ the momentum per unit length and AE the energy per unit length. The Euler equations are closed by an algebraic relation between E, p, ρ, u. This relation depends on the properties of the gas and is considered further in Section 2.2. The simplest is the ideal gas equation,

$$E = \frac{1}{2}\rho u^2 + \frac{p}{\gamma - 1},$$

which is used throughout this thesis.

In reality momentum and energy are not strictly conserved. In long pipelines there are significant frictional losses and often this is the dominating factor. Other important influences are the conduction of heat between the gas and its surroundings and the change in potential energy of gas in inclined pipes. This latter effect is easily dealt with by including source terms in the momentum and energy equations, as shown by Goldwater and Fincham.

*It is worth noting that the momentum flux is only an exact derivative if the pipe area A is constant. If the pressure term is written

$$A\frac{\partial p}{\partial x} = \frac{\partial (Ap)}{\partial x} - p\frac{\partial A}{\partial x}$$

then the variation in A appears as a source term in the momentum equation, in other words momentum is lost or gained from the pipe if the area changes. In extreme cases such as an abrupt change in pipe diameter, a one-dimensional model may no longer be appropriate.

The friction and heat conduction source terms are more difficult since they require empirical models of the physics. The frictional losses are modelled by the inclusion of a source term in the momentum equation,

$$\frac{\partial(A\rho u)}{\partial t}+\frac{\partial(A\rho u^2)}{\partial x}+A\frac{\partial p}{\partial x}+C\tau=0,$$

where C is the circumference of the pipe and τ is the average wall stress. The wall stress usually takes the form,

$$\tau=\frac{1}{2}f\rho u|u|,$$

where f is the Fanning friction factor discussed in detail in Section 2.4.2. Heat conduction is similarly modelled by a source term in the energy equation,

$$\frac{\partial(AE)}{\partial t}+\frac{\partial(A(E+p)u)}{\partial x}=\Omega,$$

where Ω is the rate of heat transfer per unit length of pipe. Since most pipes are circular and of constant area we may divide through by A and write $C/A=4/D$ to give,[†]

$$\frac{\partial \rho}{\partial t}+\frac{\partial q}{\partial x}=0 \tag{2.0.1}$$

$$\frac{\partial q}{\partial t}+\frac{\partial(q^2/\rho)}{\partial x}+\frac{\partial p}{\partial x}+\mu\frac{q|q|}{\rho}=0 \tag{2.0.2}$$

$$\frac{\partial E}{\partial t}+\frac{\partial((E+p)q/\rho)}{\partial x}=\Omega/A, \tag{2.0.3}$$

where $\mu\overset{\text{def}}{=}2f/D$ and $q=\rho u$ is defined by Landau and Lifschitz [LL89] as the mass flux density and has units of kg/s/m^2. An engineer would be more interested in the mass flow rate,

$$Q_{mass}=\rho uA=qA \qquad [\text{kg/s}],$$

the molar flow rate,

$$Q_{molar}=\frac{\rho uA}{MW}=\frac{Q_{mass}}{MW} \qquad [\text{kmol/s}],$$

[†] For noncircular pipes and annuli this is used to *define* the hydraulic diameter D.

or the volumetric flow rate,

$$Q = \frac{836}{3600 \times 24} Q_{molar} \qquad \text{[standard cubic feet per day (SCFD)]},$$

but we work with q since it eliminates the area from all but the source terms.

Equations (2.0.1) to (2.0.3) are a system of nonlinear hyperbolic conservations laws. Numerical methods for their solution are described in Chapter 3.

2.1 Model simplifications

The heat transfer term Ω is dependent on the gas temperature, the external temperature and the conductivity of the pipe and surrounding soil (if any). Two semi-empirical models for Ω are described in Section 2.4.3. However, often the problem of interest falls into one of two extreme cases. If we are simulating an event such as a pipe-break which takes place over a time-scale of seconds then the changes are too rapid for heat conduction to play a significant rôle and we may safely neglect it and set $\Omega = 0$. The opposite situation is where changes take place so slowly that heat conduction is sufficient to keep the gas at ambient temperature, that is, the gas is isothermal. This vastly simplifies the problem since the energy equation is now redundant (except for calculating Ω, the actual magnitude of the heat transfer). We may also eliminate ρ from the equations by substituting the ideal gas equation of state (see Section 2.2),

$$\rho = \frac{1}{RT} p,$$

into the mass and momentum equations to give,

$$\frac{1}{RT}\frac{\partial p}{\partial t} + \frac{\partial q}{\partial x} = 0 \tag{2.1.1}$$

$$\frac{\partial q}{\partial t} + RT\frac{\partial (q^2/p)}{\partial x} + \frac{\partial p}{\partial x} + \mu RT \frac{q|q|}{p} = 0, \tag{2.1.2}$$

since T is constant.

These equations are also hyperbolic and may be solved by the standard numerical methods described in the Chapter 3. Goldwater and Fincham give two methods, a box-like scheme and the method of characteristics.

Many authors neglect terms in the momentum equation to make the problem more tractable. To compare the relative importance of the terms we nondimensionalise the equations (2.1.1) and (2.1.2) to give,

$$\begin{gathered}\alpha\frac{\partial p}{\partial t}+\frac{\partial q}{\partial x}=0\\ \epsilon\frac{\partial q}{\partial t}+\frac{\epsilon}{\alpha}\frac{\partial(q^2/p)}{\partial x}+\frac{\partial p}{\partial x}+\nu\frac{q|q|}{p}=0,\end{gathered}$$

where

$$\alpha=\frac{P_0L}{q_0RTt_0},\quad \epsilon=\frac{q_0L}{P_0t_0},\quad \nu=\frac{2fq_0^2LRT}{P_0^2D},$$

and as usual we have substituted $p\to P_0p$, $x\to Lx$ etc. If we are simulating the daily changes of a large gas network then it is reasonable to consider a time-scale of hours and use the following typical values $P_0=70\times10^5$ Pa, $q_0=500$kg/m^2/s, $L=50,000$m, $R=481$J/kg/K, $T=280$K, $t_0=3600$s, $f=0.002$, $D=1$m, implying the following values $\alpha=1.44, \epsilon=9.9\times10^{-4}, \nu=1.37\times10^{-1}$.

Over this time-scale the first two "inertia" terms of the momentum equation are sufficiently small to be neglected.

Often only the nonlinear inertia term $(q^2/\rho)_x$ is dropped and the resulting (dimensional) equations,

$$\begin{gathered}\frac{1}{RT}\frac{\partial p}{\partial t}+\frac{\partial q}{\partial x}=0\\ \frac{\partial q}{\partial t}+\frac{\partial p}{\partial x}+\mu RT\frac{q|q|}{p}=0,\end{gathered}$$

remain hyperbolic. Furthermore, their characteristics are now straight lines and so the equations may be solved easily by the method of characteristics as shown by Stoner [Sto69]. Kiuchi [Kiu94] also solves these equations, but by the implicit Crank-Nicolson method. He also demonstrates a technique for solving branched networks as well as simple linear pipelines. Both authors include real gas effects in their models.

In this thesis we solve the equations obtained by neglecting both inertia terms,

$$\begin{aligned} \frac{1}{RT}\frac{\partial p}{\partial t} + \frac{\partial q}{\partial x} &= 0 \\ \frac{\partial p}{\partial x} + \mu RT\frac{q|q|}{p} &= 0, \end{aligned} \tag{2.1.3}$$

or on rewriting the momentum equation,

$$\frac{1}{2}\frac{\partial (p^2)}{\partial x} + \mu RTq|q| = 0. \tag{2.1.4}$$

This quasi-steady approximation is referred to as the parabolic equations in the remainder of the thesis.

Returning to the full equations (2.0.1) to (2.0.3), Lang [Lan91] investigates the application of a spectral method to ideal gas flow following a pipe rupture. Since this is a short time-scale phenomenon he uses the adiabatic $\Omega = 0$ assumption. Issa and Spalding [IS72] go two steps further by including real gas effects, and by modelling the heat flow in terms of the Stanton number (refer to Section 2.4.3). They solve the equations by the method of characteristics and investigate the effect that varying the Stanton number has on the solution. Heat conduction is also included in the equations solved by van Deen and Reintsema [vDR83] by an interesting approach which attempts to take into account the heat capacity of the pipe itself. Finally, Thorley and Tiley [TT87] review the numerical methods commonly used to simulate gas networks and give a very good discussion of issues such as the validity of the friction and heat transfer terms.

2.2 Equations of state

The Euler equations are closed by two algebraic relationships describing the variation of pressure,

$$p = p(\rho, T),$$

and the internal energy,

$$e = e(\rho, T),$$

with density and temperature. Once the internal energy is known the total energy density E may be found by adding the kinetic energy,

$$E = \rho e + \frac{1}{2}\rho u^2. \tag{2.2.1}$$

The first relation is known as an *equation of state (EOS)* and the simplest is a combination of Charles' and Boyle's Laws, the ideal gas law,

$$pV = nR^*T,$$

where V is the volume, n the number of moles of gas and R^* the gas constant whose value is given in appendix A.

We require the EOS to be expressed in terms of pressure, density and temperature. The mass M of n moles of gas is $n \times MW$ where MW is the mean molecular weight. Since density $\rho = M/V$ we have,

$$p = \frac{\rho n R^* T}{M} = \frac{\rho R^* T}{MW}.$$

Defining a reduced gas constant $R = R^*/MW$ we have the equation which is used throughout this thesis,

$$p = \rho R T. \tag{2.2.2}$$

The internal energy of the gas may be found from

$$e = \int_0^T c_v dT,$$

where c_v is the specific heat capacity at constant volume and is a function of temperature and pressure. The simplest model assumes that c_v is constant, which is a reasonable assumption over a limited temperature range, to give,

$$e = c_v T. \tag{2.2.3}$$

Equations (2.2.3), (2.2.2) and (2.2.1) may then be combined to give the energy density as a function of pressure, density and speed,

$$E = \frac{1}{2}\rho u^2 + \frac{c_v p}{R},$$

but from thermodynamics, for an ideal gas (see Zemansky [Zem68]) we have

$$R = c_p - c_v.$$

Therefore, defining the ratio of specific heats $\gamma \overset{\text{def}}{=} c_p/c_v$ we have,

$$E = \frac{1}{2}\rho u^2 + \frac{p}{\gamma - 1},$$

as used at the beginning of the chapter.

2.3 Alternative forms of the energy equation

Several alternative forms of the energy equation (2.0.3) are required in this thesis. Their derivations are not particularly interesting but for completeness they are included here.

Defining the specific enthalpy $h \overset{\text{def}}{=} e + p/\rho$, the energy equation (2.0.3) may be written,

$$\frac{\partial(\rho(h + u^2/2) - p)}{\partial t} + \frac{\partial(u\rho(h + u^2/2))}{\partial x} = \Omega/A,$$

or,

$$(h + u^2/2)\left(\frac{\partial \rho}{\partial t} + \frac{\partial(\rho u)}{\partial x}\right) + \rho\frac{\partial(h + u^2/2)}{\partial t} + \rho u\frac{\partial(h + u^2/2)}{\partial x} - \frac{\partial p}{\partial t} = \Omega/A.$$

Using the total derivative notation,

$$\frac{D}{Dt} = \frac{\partial}{\partial t} + u\frac{\partial}{\partial x}$$

and the continuity equation (2.0.1) to eliminate the first term this simplifies to,

$$\rho\frac{Dh}{Dt} + \rho u\frac{Du}{Dt} - \frac{\partial p}{\partial t} = \Omega/A.$$

This may be written in terms of temperature and pressure by using the thermodynamic relation,

$$\begin{aligned} dh &= \frac{\partial h}{\partial T} dT + \frac{\partial h}{\partial p} dp \\ &= c_p dT - c_p \hat{\mu} dp, \end{aligned}$$

where the specific heat at constant pressure $c_p \overset{\text{def}}{=} (\partial h/\partial T)_p$ and the Joule-Thompson coefficient $\hat{\mu} \overset{\text{def}}{=} (\partial T/\partial p)_h$. Substituting for h gives,

$$\rho c_p \frac{DT}{Dt} - (1 + \rho c_p \hat{\mu}) \frac{\partial p}{\partial t} - \rho c_p \hat{\mu} u \frac{\partial p}{\partial x} + \rho u \frac{Du}{Dt} = \Omega/A. \tag{2.3.1}$$

For slow transients the inertia term uDu/Dt is often neglected with the same justification as in the parabolic momentum equation (2.1.4). This form of the energy equation is used in Section 4.3.1 to include temperature variation in the parabolic model. For the ideal gases which we consider in this thesis the Joule-Thompson coefficient $\hat{\mu} = 0$ reducing equation (2.3.1) to,

$$\rho c_p \frac{DT}{Dt} - \frac{\partial p}{\partial t} + \rho u \frac{Du}{Dt} = \Omega/A. \tag{2.3.2}$$

We can also eliminate Du/Dt by making use of the momentum equation (2.0.2), which is reduced by the continuity equation (2.0.1) to,

$$\rho \frac{Du}{Dt} + \frac{\partial p}{\partial x} + \mu \rho u|u| = 0,$$

and substituted into equation (2.3.2) to leave,

$$\rho c_p \frac{DT}{Dt} - \frac{\partial p}{\partial t} + \rho u \left(-\frac{1}{\rho} \frac{\partial p}{\partial x} - \mu u|u| \right) = \Omega/A.$$

This simplifies to,

$$\rho c_p \frac{DT}{Dt} - \frac{Dp}{Dt} - \mu \rho u^2 |u| = \Omega/A.$$

On dividing through by ρT we get,

$$\frac{c_p}{T} \frac{DT}{Dt} - \frac{1}{\rho T} \frac{Dp}{Dt} = \frac{\Omega/A + \mu \rho u^2 |u|}{\rho T},$$

and, on using the ideal gas equation of state (2.2.2),

$$\frac{c_p}{T}\frac{DT}{Dt} - \frac{R}{p}\frac{Dp}{Dt} = \frac{\Omega/A + \mu\rho u^2|u|}{\rho T}.$$

Since this is a perfect derivative we have,

$$\frac{D}{Dt}(c_p \ln T - R\ln p) = \frac{Ds}{Dt} = \frac{\Omega/A + \mu\rho u^2|u|}{\rho T}, \tag{2.3.3}$$

which is the entropy form of the energy equation. The quantity Ds is the change in specific entropy of the gas. This form will be used in Section 4.3.2 to include temperature variation in the parabolic equations.

The entropy is often written,

$$\begin{aligned} s &= c_p \ln T - R\ln p + \text{const} \\ &= c_v(\gamma \ln T - (\gamma - 1)\ln p) + \text{const} \\ &= c_v \ln\left(\frac{T^\gamma}{p^{\gamma-1}}\right) + \text{const}, \end{aligned}$$

and again using the ideal equation of state (2.2.2) in the form $T = p/R\rho$, it can be written as

$$s = c_v \ln\left(\frac{p}{\rho^\gamma}\right) + \text{const}.$$

See Zemansky [Zem68] for a full discussion of entropy.

If there is no friction and the system is adiabatic then $\mu = \Omega = 0$ and p/ρ^γ is constant along stream lines. This is a common assumption which considerably simplifies the problem and in some circumstances allows an analytical solution. See for example Ockendon and Tayler [OT83]. If we assume that the gas is homentropic, that is the entropy is constant everywhere, then after some manipulation and the substitution of the sound speed $a = \sqrt{\gamma p/\rho}$, the mass and momentum equations may be obtained in characteristic form,

$$\left(\frac{\partial}{\partial t} + (u \pm a)\frac{\partial}{\partial x}\right)\left(u \pm \frac{2a}{\gamma - 1}\right) = 0. \tag{2.3.4}$$

This equation states that the *Riemann invariants* $u \pm 2a/(\gamma - 1)$ are constant along the characteristic curves defined by $\dot{X} = u \pm a$. The assumption that the gas is isentropic is unrealistic for gas pipelines. However, it is possible to model the entropy change empirically by replacing the ratio of specific heats γ with another constant k. This "polytropic" model was investigated for British Gas by Emmerson [Emm90]. The isentropic characteristics defined by equations (2.3.4) and

$$\frac{\partial s}{\partial t} + u\frac{\partial s}{\partial x} = 0$$

will be used in Chapter 3 to generate numerical boundary conditions.

If the gas is not isentropic then the Euler equations do not possess Riemann invariants. However, the equations may still be put into characteristic form,

$$\frac{\partial p}{\partial t} \pm \rho a\frac{\partial u}{\partial t} + (u \pm a)\left(\frac{\partial p}{\partial x} \pm \rho a\frac{\partial u}{\partial x}\right) = \gamma\mu\rho u^2|u| - (u \pm a)(\mu\rho u|u|) - (\gamma - 1)\frac{\Omega}{A}, \tag{2.3.5}$$

which, together with equation (2.3.3), form a system of coupled ordinary differential equations. These also will be used to derive numerical boundary conditions in Chapter 3.

2.4 Including real gas effects

2.4.1 Real gas equations of state

In this thesis the ideal gas EOS has been used to avoid the introduction of extra complications. However, accurate simulation of gases at high pressure is not possible without the inclusion of real gas effects. This section briefly outlines how the methods of later chapters may be adapted to use a nonideal EOS.

The ideal gas law has been very successful at predicting the behaviour of gases at low pressures and temperatures but can give errors in pressure as large as 500% at the pressures encountered in the gas industry. In the last 100 years many alternative equations of state have been proposed to account for the nonideality of a real gas. They are usually written as a correction to the ideal gas EOS in the form,

$$pV = z(p,T)R^*T. \tag{2.4.1}$$

One of the first, and most successful attempts to improve on the ideal gas EOS was by van der Waals in the late nineteenth century who proposed,

$$p = \frac{R^*T}{V-b} - \frac{a}{V^2}.$$

The constant b models the finite volume of the molecules which becomes more significant at low temperatures and high pressures. The second term on the right hand side compensates for the mutual attraction between the molecules which tends to reduce the force with which molecules strike the walls of the container, and thus reduce the pressure. Substituting the van der Waals equation into equation (2.4.1) results in a cubic equation for the factor z which may be easily solved either directly or by Newton iteration. The most common EOS in use today, the Soave-Redlich-Kwong (SRK) and the Peng-Robinson equations, may also be expressed as the roots of a cubic and there are many standard packages available to calculate z-factors and other so called Pressure-Volume-Temperature (PVT) information for a given fluid.

Incorporating the z-factor into the parabolic equations (2.1.3), (2.1.4) is quite straightforward. A z-factor will occur wherever density has been replaced by pressure, for example the momentum equation becomes,

$$\frac{1}{2}\frac{\partial(p^2)}{\partial x} + \mu z(p,T)RTq|q| = 0.$$

The numerical method (4.2.1) and (4.2.2) described in Chapter 4 will solve this equation with little modification. The extra complication arises in the calculation of the Jacobian which requires $\partial z/\partial p$. However this is usually supplied by z-factor packages and even if not, it is small enough not to have a significant effect if omitted.

Including the z-factor in the hyperbolic scheme is more difficult since Roe's scheme (see Section 3.3) requires an approximate Riemann solver. This has not been found for a general EOS, so an alternative is to 'freeze' the z-factor at each Riemann problem. Given some suitably averaged $\bar{z}$ then equation (2.2.1) becomes,

$$E = \frac{c_v}{\bar{z}R}p + \frac{1}{2}\rho u^2.$$

If we *define* $\bar{\gamma}$ by,

$$\frac{1}{\bar{\gamma} - 1} = \frac{c_v}{\bar{z}R},$$

then the equations are formally the same as the ideal gas equations, when expressed in terms of ρ, q, E. Provided that the averaging of z (or alternatively of γ) is chosen well, then Roe's scheme should still give a reasonable solution, although the conservation condition (3.3.11) will no longer be exactly satisfied.

2.4.2 Friction

The derivation of the Euler equations assumes that the fluid is inviscid, that is, there are no frictional losses. For a long pipeline this is certainly not the case, and as Section 2.1 showed, frictional effects dominate in certain circumstances. The frictional losses are usually simulated by a semiempirical term of the form,

$$\tau = \frac{f\rho u|u|}{2},$$

where the dimensionless constant f is known as the Fanning friction factor. Various formulae are available for the friction factor, a good source is Perry's Chemical Engineering Handbook [Gre88]. Two of the most popular for turbulent flow are the Blasius smooth pipe law,

$$f = 0.079 Re^{-0.25},$$

where the Reynolds number is,

$$Re = \frac{\rho u D}{\mu},$$

and μ is the viscosity. For rough pipes, the Colebrook equation is often used,

$$\frac{1}{\sqrt{f}} = -4\log_{10}\left(\frac{\epsilon}{3.7D} + \frac{1.256}{Re\sqrt{f}}\right),$$

where ϵ is the pipe roughness*.

All of these formulae were developed for steady, isothermal flow and thus cannot be expected to be accurate for transients. However, in the absence of anything better, we use them throughout this thesis both for quasi-steady state simulations and for the rapid transients. According to Stewart [Ste] in commercial situations where pipes are subject to corrosion, distortion and scaling, the accuracy with which pressure drops can be predicted is often no better than 25% even for steady flow. In real simulation software parameters such as the friction factor are "tuned" to reproduce experimental data in the hope that this will provide a better model for predictive purposes.

It is possible to model the additional frictional losses incurred at bends and junctions by increasing the friction factor, or more commonly, replacing a bend with the equivalent amount of straight pipe.

2.4.3 The heat transfer term

The two extremes of adiabatic $\Omega = 0$ and isothermal flow $T = \text{const}$ are not always satisfactory. The most common intermediate model is to define the Stanton number St,

$$St = \frac{\Omega}{\pi \rho c_p u D (T_w - T_0)},$$

where T_w is the wall temperature and T_0 the *stagnation temperature* of the gas (that is, the temperature the gas would have if it was brought to rest isentropically). Rearranging gives the heat conduction as,

$$\Omega = \pi \rho u c_p D St (T_w - T_0).$$

The Stanton number may be found from boundary layer theory, or calculated by a relationship such as,

$$St(Re^{0.2})(Pr^{0.6}) = \text{constant}.$$

*measured to be about 0.1mm by Shell on the high pressure loop at Bacton

Issa and Spalding [IS72] used a heat transfer term of this form and concluded that variations in St made such a small difference that it was sensible to keep it constant.

BG use a less empirical model for heat conduction which is based on solving Laplace's equation in the soil surrounding the pipe. The resulting heat conduction term is given by,

$$\Omega = \frac{2\pi k}{\ln\left(h/R + \sqrt{h^2/R^2 - 1}\right)}(T_p - T_g). \tag{2.4.2}$$

The assumptions underlying this formula are that we have a straight pipe of length L and radius R aligned along the z-axis and buried at depth h from the surface given by $y = 0$. In the soil surrounding the pipe we assume the linear heat equation,

$$\rho c \frac{\partial T}{\partial t} - k\left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2}\right) = 0, \tag{2.4.3}$$

and that the gas in the pipe is well mixed so that we can assume that the pipe surface temperature $\tilde{T}$ is a function only of z and t. Conservation of energy requires that the heat transfer term is given by,

$$\Omega = \oint_{\text{pipe}} -k\frac{\partial T}{\partial r}\, ds. \tag{2.4.4}$$

If we non-dimensionalise with $x \to hx$, $y \to hy$, $z \to Lz$, $T \to T_0 T$ and $t \to \tau t$ the heat equation (2.4.3) becomes,

$$\frac{\rho c h^2}{k\tau}\frac{\partial T}{\partial t} - \left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2}\right) - \frac{h^2}{L^2}\frac{\partial^2 T}{\partial z^2} = 0.$$

We can approximate this by Laplace's equation if both $h^2 << L^2$ and $\rho c h^2/k\tau <<$ 1. The former requirement is clearly satisfied since typically L = 50000m and h = 1m. Soil typically has a density of ρ = 2050kg/m^3, a specific heat capacity of c = 1840J/kg/K and a conductivity of about k = 0.52J/s/m/K. Therefore the latter assumption only holds if we are considering a time-scale of $t > 7 \times 10^6$s $\simeq$ 84 days. This is valid for steady state calculations and in this case we need to solve,

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} = 0,$$

with the boundary conditions,

$$\begin{aligned} T &= T_g \text{ on } y = 0 \\ T &= T_p \text{ on the pipe surface, the circle } (R\cos\theta, -h + R\sin\theta). \end{aligned}$$

The method of images may be used to find the solution,

$$T(x,y) = T_g + \frac{T_g - T_p}{\ln\left(\frac{h-a}{h+a}\right)} \ln\left(\frac{x^2 + (y-a)^2}{x^2 + (y+a)^2}\right),$$

where $a = -\sqrt{h^2 - R^2}$. To find Ω we want to calculate the integral (2.4.4) but this is equivalent to integrating around any closed curve enclosing only the singularity at $(0,a)$. If we choose a circle of radius $r < |a|$ centred on $(0,a)$ we find (since the denominator in the logarithm makes no contribution to the integral),

$$\begin{aligned} \Omega &= \oint -k\frac{\partial T}{\partial r} \\ &= -k\frac{T_g - T_p}{\ln\left(\frac{h-a}{h+a}\right)} \int_0^{2\pi} \frac{2}{r} r \, d\theta \\ &= -4\pi k \frac{T_g - T_p}{\ln\left(\frac{h-a}{h+a}\right)} \\ &= \frac{4\pi k (T_p - T_g)}{\ln(h/R + \sqrt{h^2/R^2 - 1}) - \ln(h/R - \sqrt{h^2/R^2 - 1})}, \end{aligned}$$

and on simplifying the logarithm by multiplying by the conjugate of the denominator we have,

$$\Omega = \frac{2\pi k (T_p - T_g)}{\ln(h/R + \sqrt{h^2/R^2 - 1})}.$$

2.5 Summary of model assumptions

- One-dimensional, turbulent gas.
- Circular, horizontal pipes.
- Ideal, polytropic EOS.
- Friction modelled by steady state friction factors.

- Single phase.

Additional assumptions for the hyperbolic model:

- Adiabatic flow assumed in this thesis, $\Omega = 0$.

Additional assumptions for the parabolic model:

- Isothermal flow $T =$const or adiabatic flow $\Omega = 0$.

- Inertia negligible compared to friction.

Chapter 3

Numerical solution of the hyperbolic equations

3.1 Introduction

This chapter describes the numerical method chosen to solve the hyperbolic equations. It begins with a brief survey of numerical methods for hyperbolic conservation laws, concentrating on the high resolution finite difference schemes that were developed in the eighties. This is followed by a detailed description of Roe's scheme as applied to the hyperbolic equations. Finally, we describe several means of implementing boundary conditions, including a new method which follows naturally from Roe's scheme.

To recapitulate, by the "hyperbolic equations" we mean the one dimensional Euler equations augmented with a frictional source term,

$$\frac{\partial \rho}{\partial t} + \frac{\partial q}{\partial x} = 0 \tag{3.1.1}$$

$$\frac{\partial q}{\partial t} + \frac{\partial (q^2/\rho)}{\partial x} + \frac{\partial p}{\partial x} + \mu \frac{q|q|}{\rho} = 0 \tag{3.1.2}$$

$$\frac{\partial E}{\partial t} + \frac{\partial ((E+p)q/\rho)}{\partial x} = 0. \tag{3.1.3}$$

Note that in the remainder of this thesis the heat conduction term Ω has been neglected. These equations fall into the category of inhomogeneous hyperbolic conservation laws, which take the form

$$\frac{\partial \mathbf{u}}{\partial t} + \frac{\partial \mathbf{f}(\mathbf{u})}{\partial x} = \mathbf{s} \tag{3.1.4}$$

and arise in areas as diverse as modelling the flow of traffic to describing the flow of water after a dam break. The vector $\mathbf{u}$ represents some conserved quantity, in this case $[\rho, q, E]$, while $\mathbf{f}$ is the corresponding flux. Often the system is written as

$$\frac{\partial \mathbf{u}}{\partial t} + A(\mathbf{u})\frac{\partial \mathbf{u}}{\partial x} = \mathbf{s} \tag{3.1.5}$$

where A is the Jacobian matrix of $\mathbf{f}$ with respect to $\mathbf{u}$. For a linear system A is a constant matrix and $\mathbf{f} = A\mathbf{u}$.

Hyperbolic equations have some interesting properties. They describe the transmission of information along curves in $x - t$ space known as characteristics. This is associated with the propagation of waves at a finite speed. Sometimes the transmitted information can be described by a function of the conserved variables, called a Riemann invariant, which is constant along a characteristic curve. An example of such a system is the homentropic Euler equations (2.3.4) of Chapter 2. When this is not possible the equations can still be recast in characteristic form, that is, a system of coupled ODEs along the characteristics such as equations (2.3.5).

The most important feature of this type of partial differential equation is that discontinuities can appear in solutions even with smooth initial data. These discontinuities are known as shocks, since they were first studied by aerodynamicists, and often occur because of some failing of the underlying model. In gas dynamics, shocks represent narrow zones with steep velocity gradients where viscosity dominates. Since the Euler equations neglect viscosity the zones appear as discontinuities.

An associated aspect of hyperbolic conservation laws is nonuniqueness of solution, which arises in two ways. Firstly, the PDE itself is derived from an integral equation describing the conservation of $\int udx$. However, the same PDE may also be derived for the conservation of $\int u^2dx$, $\int u^3dx$ etc. These all have identical solutions when u is smooth but not when discontinuities are present. Care must be taken when constructing numerical methods that the correct *weak solution*, conserving $\int u dx$, is selected. Secondly, valid weak solutions may be constructed which cannot physically exist. These *rarefaction shocks* do not exist in nature because viscosity spreads them into rarefaction waves. In the idealised inviscid model described by the PDE an entropy condition is imposed to reject such unphysical solutions.

The properties of hyperbolic conservation laws and the rôle of entropy is discussed in detail in the classic paper by Lax [Lax73]. For more general information on characteristics and hyperbolic partial differential equations consult a textbook such as Kevorkian [Kev90].

The major challenge for numerical methods for conservation laws is to resolve discontinuities correctly and sharply while giving accurate results on the smooth parts of the flow. Of the numerical methods available we will pursue finite difference methods rather than finite volumes or finite elements. In one dimension finite volume methods and finite difference methods are indistinguishable. We do not consider finite element methods for several reasons. Firstly, they are more difficult to implement than finite differences. It is also more difficult to introduce 'upwinding' into finite elements, which is essential if there is strong advection.*

Furthermore, the aim is to interface the hyperbolic method with that used for the parabolic equations. The parabolic scheme described in the next chapter is a finite difference scheme and therefore it is easier conceptually also to use a finite difference scheme for the hyperbolic equations.

*Although developments such as Streamline Upwind Petrov/Galerkin [HT84] have been successful.

3.2 Finite difference schemes for hyperbolic conservation laws

In this section we describe some of the finite difference methods for equation (3.1.4), neglecting for the moment the source term $\mathbf{s}$ which will be discussed in Section 3.3.2.

Having narrowed the choice to finite difference schemes there is still a vast number available. The first decision is between explicit and implicit schemes. Explicit schemes are far easier to implement since they do not require the solution of a system of algebraic equations at each timestep. However, they suffer from a stability restriction on the timestep, usually that the Courant number λ satisfies the CFL condition (in one dimension),

$$\lambda := \frac{|a|\Delta t}{\Delta x} \leq 1 \tag{3.2.1}$$

where a is the wave speed of largest magnitude. Unlike the stability condition for parabolic equations, this is not unduly restrictive since we require a reasonably short timestep to resolve the solution with sufficient accuracy. The stability condition would be a handicap if we were interested in the steady-state solution or if we were simulating long timescale phenomena, and in this case an implicit method would be appropriate (see Kiuchi [Kiu94]). For our problem though we follow the advice of K.W.Morton [Mor71] - "Implicit and semi-explicit algorithms of the Crank-Nicholson type should be restricted to use in diffusion equations where an explicit time differencing would result in a prohibitively small timestep to ensure numerical stability."

There has been a huge amount of progress in the last two decades on explicit finite difference schemes for hyperbolic conservation laws. An excellent introduction to the subject is the book by Randall LeVeque [LeV90] which also describes the analytical properties of the equations. It is not possible to go into such detail here and so this section will be restricted to detailed description of the schemes actually used, along with a sketch of their development and references to the appropriate papers.

Simple schemes

Perhaps the simplest, naive, explicit finite difference scheme to solve the conservation law (3.1.4) is to use a centred difference for the spatial derivative and a forward difference for the temporal derivative,

$$\frac{\mathbf{U}_i^{n+1} - \mathbf{U}_i^n}{\Delta t} + \frac{\mathbf{f}(\mathbf{U}_{i+1}^n) - \mathbf{f}(\mathbf{U}_{i-1}^n)}{2\Delta x} = 0.$$

It is well known that this scheme is unconditionally unstable (that is, no matter how small the timestep is made, the true solution will be swamped by an ever growing error), and this can be easily proved by the use of Fourier analysis. See for example Morton and Mayers [MM94]. The scheme can be stabilised, subject to the CFL condition (3.2.1), simply by replacing $\mathbf{U}_i^n$ by $(\mathbf{U}_{i-1}^n + \mathbf{U}_{i+1}^n)/2$ to give the Lax-Friedrichs scheme,

$$\frac{\mathbf{U}_i^{n+1} - (\mathbf{U}_{i-1}^n + \mathbf{U}_{i+1}^n)/2}{\Delta t} + \frac{\mathbf{f}(\mathbf{U}_{i+1}) - \mathbf{f}(\mathbf{U}_{i-1})}{2\Delta x} = 0. \qquad (3.2.2)$$

The Lax-Friedrichs scheme is the simplest stable, centred difference scheme and it requires no knowledge of the underlying equations. However it is only first-order accurate and is notorious for being very diffusive, that is, it smears discontinuities such as shocks. A second-order accurate method, which uses the same three-point stencil (the pattern of points used to calculate a value at the new timestep), is the famous Lax-Wendroff scheme. Lax-Wendroff, and its two-step equivalents for nonlinear equations, such as the Richtmyer scheme (see Sod [Sod85]), give excellent results for smooth solutions but oscillate severely around discontinuities (see the examples in LeVeque's book [LeV90]). Such oscillatory behaviour is characteristic of second-order schemes such as Lax-Wendroff and Beam-Warming (see [RB76]) just as diffusion is characteristic of first-order schemes such as Lax-Friedrichs.

Conservation

The numerical schemes described so far have in common the property that they can be written in conservation form,

$$\frac{\mathbf{U}_i^{n+1} - \mathbf{U}_i^n}{\Delta t} + \frac{\mathbf{F}_{i+1/2}^n - \mathbf{F}_{i-1/2}^n}{\Delta x} = 0. \tag{3.2.3}$$

For example, in the case of centred, three-point schemes $\mathbf{F}_{i+1/2} = \mathbf{F}(\mathbf{U}_i, \mathbf{U}_{i+1})$, and in particular for Lax-Friedrichs (3.2.2),

$$\mathbf{F}_{i+1/2} = \frac{1}{2}(\mathbf{f}(\mathbf{U}_i) + \mathbf{f}(\mathbf{U}_{i+1})) - \frac{\Delta x}{\Delta t}(\mathbf{U}_{i+1} - \mathbf{U}_i).$$

Lax and Wendroff showed in [LW60] that this form is essential if the scheme is to converge to the correct weak solution and shocks are to move at the correct speed. In addition a numerical entropy condition is usually imposed to ensure that unphysical rarefaction shocks are disallowed.

Upwinding for scalar laws

Including some of the physics of the problem into the method often gives better results than centred schemes.

For a scalar linear conservation law,

$$\frac{\partial u}{\partial t} + a\frac{\partial u}{\partial x} = 0, \tag{3.2.4}$$

'upwinding', that is taking account of the direction of wave propagation, is straightforward since the direction of wave propagation is clear: to the left if $a < 0$ and to the right if $a > 0$. A first-order accurate upwind scheme for this equation is,

$$\frac{U_i^{n+1} - U_i^n}{\Delta t} = -\begin{cases} a\frac{U_i^n - U_{i-1}^n}{\Delta x} & a > 0 \\ a\frac{U_{i+1}^n - U_i^n}{\Delta x} & a < 0, \end{cases} \tag{3.2.5}$$

where the 'leg' of the stencil points in the upwind direction. Although it is less diffusive than Lax-Friedrichs the smearing of discontinuities is still unacceptable for many problems.

Upwinding for systems

When dealing with systems of equations rather than scalar conservation laws it is not straightforward to introduce upwinding to take account of the wave direction. If the system is linear, then it can be put into characteristic form (see Section 3.1) simply by multiplying by a matrix constructed from the eigenvectors of A. The system then decouples into independent scalar PDEs of the form (3.2.4) each of which can be solved by a scheme such as (3.2.5). When the system is nonlinear then some form of linearisation is usually done to determine the characteristic directions. Issa and Spalding [IS72] provide a good example where they cast the equations in characteristic form and assume that over a single timestep the characteristics are straight lines. They trace these lines back from a grid point at the advanced timestep to the intersection with the initial timestep. They then calculate the variables at these points by linear interpolation between the adjacent grid points and integrate forward along the characteristics to the new timestep. This characteristic based method can be shown to be equivalent to first-order upwinding in the case of a linear system.

Although the method of characteristics is more complicated to implement than the simple difference schemes of the last section, it includes some of the physics behind the model and it allows a very natural method of applying boundary conditions, as will be shown in Section 3.4. The disadvantage is that it cannot be put into the conservation form (3.2.3) and therefore should not be used on problems which might contain shocks. This is demonstrated by Issa and Spalding's figures 2 and 3 which clearly show that the shock is misplaced.

Godunov's method

The dilemma of wanting to have a conservative scheme which includes some form of upwinding was resolved in an ingenious manner by Godunov [God59]. His revolutionary idea was to abandon the use of a solution at a finite number of points on a mesh. Instead, he replaces the initial solution by a simplified approximation, defined everywhere, though not necessarily continuous. The conservation law is then solved exactly for a short timestep, which is a practical proposition with the simplified data. At the end of the timestep the solution is projected back to an approximation, and the process begins again.

Godunov chose the most simple approximation possible. The computational domain is divided into a sequence of cells, rather than points, and the approximation is constant within each one. At the cell edges the approximation is discontinuous and it is here that the computation is required. Waves are generated at the cell-edges which propagate into the adjacent cells. Provided the time step is not too long, waves from neighbouring cell-edges will not have time to interact and each cell-edge can be considered in isolation. The minimum time for the waves to interact is the time taken for a wave to cover half a cell,

$$\frac{|a|\Delta t}{\Delta x} \leq \frac{1}{2}.$$

The problem therefore reduces to solving a Riemann problem at each cell edge, that is, solve the conservation law (3.1.4) with the initial conditions,

$$\mathbf{u}(x,0) = \begin{cases} \mathbf{u}_l & x < 0 \\ \mathbf{u}_r & x > 0. \end{cases}$$

at each cell edge.

The conservation law provides a solution which, by definition, is conservative. This is followed by the projection step, replacing the calculated solution with a piecewise constant approximation chosen to ensure conservation. Hence the overall method is conservative, and provided that an entropy condition is satisfied we can be sure that the approximation will converge to the correct weak solution. In practice, these stages can be combined into a single step as will be shown in detail in Section 3.3 and the entire scheme written in conservative form (3.2.3).

There are two disadvantages to Godunov's original scheme. Firstly, it is still only first-order accurate. This problem is addressed in Section 3.2. Secondly, the Euler equations are so complicated that even solving a Riemann problem is a nontrivial task and requires iteration (see Holt [Hol77] for a detailed description).

It was pointed out by Roe [Roe81] that since the solution to the Riemann problem is re-averaged at each timestep information is inevitably lost, and it is questionable whether it is worth solving the conservation law exactly. He suggested that there would be little loss of accuracy if the conservation law was replaced by one with similar properties which is easier to solve. This led to a category of methods based on Godunov's but using *approximate Riemann solvers.* Roe's own idea was to solve the linear system obtained by replacing $A(\mathbf{u})$ in equation (3.1.5) with a constant matrix A. This idea is discussed in detail in Section 3.3. Harten et al [HLvL83] propose a class of simpler approximate Riemann solvers, the simplest of which approximates the solution by two constant states separated by a single intermediate state. They also specify a number of conditions to ensure that the solver is conservative and satisfies an entropy condition. Their solver was used by Einfeldt [Ein88] to solve the Euler equations, with a slight modification to improve the resolution of contact discontinuities.

Second-order accuracy

Traditionally, first-order schemes have been too diffusive for practical use while second-order schemes suffered from oscillations, and sometimes instability, at discontinuities.

Warming and Beam [RB76] noticed that the oscillations tend to follow the discontinuity for the Lax-Wendroff scheme but precede it for the Beam-Warming scheme. They then proposed a way of switching between the schemes so that Beam-Warming was used on one side of the shock and Lax-Wendroff on the other, significantly improving the solution. This idea of automatically switching scheme near a discontinuity has been used by many authors to develop so-called high resolution schemes.

The problem of achieving second-order accuracy while avoiding oscillations around shocks was first attacked with some success by Boris and Book [BB73]. Their 'flux-corrected transport' algorithm was a two stage process which calculated the diffusion caused by the basic first-order scheme and subtracted it from the solution. However, they recognised that too much 'correction' would induce oscillations and so the backwards diffusion was limited to ensure that monotonic profiles remained monotonic. This was the forerunner of flux-limited methods which decompose the flux term into its first and second-order components,

$$F_{i-1/2} = F^1_{i-1/2} + (F^2_{i-1/2} - F^1_{i-1/2}),$$

where F^1 is a first-order flux such as Lax-Friedrichs and F^2 is a second-order flux such as Lax-Wendroff. The correction term is then restricted by a *limiter function* Φ,

$$F_{i-1/2} = F^1_{i-1/2} + (F^2_{i-1/2} - F^1_{i-1/2})\Phi.$$

The limiter is a function of the slope of the solution, switching on (i.e. becoming zero) near steep gradients to switch to first order, and remaining close to unity on smooth parts of the solution to ensure that solver is second-order accurate where possible. Sweby [Swe85] gives some example solutions of Burger's equation comparing a wide range of limiter functions.

In a parallel development in the seventies van Leer investigated several ways of generating a second-order method in a series of papers searching for "the ultimate conservative difference scheme" (a quest which Leonard claims to have completed with his ULTIMATE scheme [Leo91]). In the second paper of the series [vL74] van Leer generalised Fromm's scheme, itself an arithmetic average of the Lax-Wendroff and Beam-Warming schemes. He used a different average based on the slope of the solution, mirroring flux limiters. By the fourth instalment [vL77] van Leer decided that the way forward lay with Godunov's approach. He showed that a second-order method could be generated by replacing the piecewise constant data used by Godunov by a piecewise linear or higher order approximation. The slopes were generated in a variety of ways and needed to be limited to ensure monotonicity of the solution. The final paper of the series [vL79] applied these ideas to the Euler equations in Lagrangian form and produced his famous 'MUSCL' scheme.

Many authors have extended this 'slope-limiter' approach such as Goodman and LeVeque [GL88] who solved a scalar conservation law by using a piecewise linear approximation to the flux. Another step forward came with Woodward and Collela's piecewise parabolic method [CW84]. LeVeque's 1990 book [LeV90] shows how flux and slope limiters are equivalent for linear systems.

The goal of oscillation-free solutions was formalised by Harten [Har83a] with the concept of Total Variation Diminishing (TVD) methods which ensure that the variation

$$\sum_{j=j_1}^{j_2} |u_j - u_{j+1}|$$

will not increase. He showed how to construct second-order TVD schemes by solving a first-order scheme's modified equation. In a subsequent paper [Har84] Harten showed that a numerical method in conservation form which is TVD and satisfies an entropy inequality will converge to the unique entropy solution of the conservation law.

Second-order TVD schemes are constructed to ensure that scalar conservation laws (or *linear* systems) do not generate spurious oscillations and the theory does not carry across to nonlinear systems. However, the methods often give excellent results when extended to systems such as the Euler equations. One popular technique, which was used by Harten, is to use Roe's linearisation to apply the scheme 'scalarly' to each characteristic field. See Roe and Baines [RB82] for more information on this subject.

In 1984 Osher [OC84] introduced a class of 'E-schemes' which are not only TVD but converge to the unique entropy condition. In the same year Sweby [Swe84] established the conditions on flux limiters to ensure that the scheme is TVD.

A disadvantage of TVD schemes is that they degenerate to first-order at extrema of the solution. This problem has been tackled by several authors such as Harten and Osher [HO87], and more recently Huynh [Huy95], who relaxed the criterion of Total Variation Diminishing to forbidding the creation of new extrema in the solution.

There are many other finite-difference methods which do not fall into the categories described above. An interesting example is Boltzmann- type schemes. These begin with a particle based description of the fluid, it can be shown that the Euler equations for an ideal monatomic gas follow if certain assumptions are made about the collisions between the particles. These schemes solve the Boltzmann equation governing the probability density of the molecular velocities, see for example [Rei81], [Pul80], [Per92] and for a description of how they relate to flux-split schemes [HLvL83].

Monte Carlo schemes such as Chorin's Random Choice Method [Cho76], [Cho77] are notable as analytical tools and also as methods which perfectly resolve discontinuities.

LeVeque's long time step scheme [Lev82] is unusual in that it is not restricted by the CFL condition (3.2.1). His idea is to explicitly track discontinuities and calculate their interactions as they collide. A similar and more recent method by Risebro and Tveito [RT90] breaks smooth parts of the solution into discontinuities, which are then tracked.

3.3 The Roe Scheme

Roe's scheme was chosen to solve the hyperbolic equations for the following reasons. It is a Godunov-type scheme and so has the advantages of upwinding while being conservative and correctly positioning shocks. Emmerson [Emm90] demonstrated its superiority over characteristic and directional difference schemes for several problems of gas dynamics. Although there are now many schemes available which have superior accuracy (e.g. Huynh [Huy95]) it remains one of the simplest Godunov-type methods to understand and implement and can be made second-order accurate by the use of flux or slope limiters as shown in Section 3.3.1.

Details of Godunov's scheme

We begin by setting out the details of the implementation of Godunov's scheme, which is the starting point of Roe's scheme.

Consider a single cell x_i with cell edges at $x_{i-1/2}$ and $x_{i+1/2} = x_{i-1/2} + \Delta x$ at time $t = 0$. Godunov first considers a piecewise constant approximation $\hat{\mathbf{u}}(x, 0)$ to the solution $\mathbf{u}(x, 0)$ which, in cell x_i takes the form,

$$\hat{\mathbf{u}}(x,0) = \mathbf{U}_i^n := \frac{1}{\Delta x}\int_{x_{i-1/2}}^{x_{i+1/2}} \mathbf{u}(x,0)dx.$$

This function is then used as initial data for the conservation law

$$\frac{\partial \hat{\mathbf{u}}}{\partial t} + \frac{\partial \mathbf{f}(\hat{\mathbf{u}})}{\partial x} = 0. \tag{3.3.1}$$

The numerical approximation at the new time level $t = \Delta t$ is found by averaging the solution of equation (3.3.1) over each cell, that is,

$$\mathbf{U}_i^{n+1} = \frac{1}{\Delta x}\int_{x_{i-1/2}}^{x_{i+1/2}} \hat{\mathbf{u}}(x,\Delta t)dx. \tag{3.3.2}$$

However, instead of explicitly calculating $\hat{\mathbf{u}}(x, t)$ we integrate the conservation law (3.3.1) from $x = x_{i-1/2}$ to $x_{i+1/2}$ and from $t = 0$ to Δt to give,

$$\int_{x_{i-1/2}}^{x_{i+1/2}} (\hat{\mathbf{u}}(x,\Delta t) - \hat{\mathbf{u}}(x,0))dx + \int_0^{\Delta t} \mathbf{f}(\hat{\mathbf{u}}(x_{i+1/2},t)) - \mathbf{f}(\hat{\mathbf{u}}(x_{i-1/2},t))dt = 0$$

and using the definitions of $\mathbf{U}_i^n$ and $\mathbf{U}_i^{n+1}$,

$$\mathbf{U}_i^{n+1} - \mathbf{U}_i^n + \frac{1}{\Delta x}\int_0^{\Delta t} \mathbf{f}(\hat{\mathbf{u}}(x_{i+1/2},t)) - \mathbf{f}(\hat{\mathbf{u}}(x_{i-1/2},t))dt = 0. \tag{3.3.3}$$

If we consider the interface $x_{i-1/2}$ in isolation, with the initial conditions

$$\hat{\mathbf{u}}_{i-1/2}(x,0) = \begin{cases} \mathbf{u}_{i-1} & x < x_{i-1/2} \\ \mathbf{u}_i & x > x_{i-1/2} \end{cases}$$

then the solution can be expressed as a similarity solution,

$$\hat{\mathbf{u}}_{i-1/2}(x,t) = \mathbf{w}_{i-1/2}\left(\frac{x - x_{i-1/2}}{t}\right),$$

due to the invariance of the problem when the x and t axes are scaled by the same factor. Consequently the integrand in equation (3.3.3) is constant, provided that Δt is sufficiently small that neighbouring Riemann problems do not interact. The Godunov scheme can therefore be written,

$$\mathbf{U}_i^{n+1} - \mathbf{U}_i^n + \frac{\Delta t}{\Delta x}(\mathbf{F}_{i+1/2}^n - \mathbf{F}_{i-1/2}^n) = 0,$$

where,

$$\mathbf{F}_{i-1/2}^n := \mathbf{f}(\mathbf{w}_{i-1/2}(0)), \tag{3.3.4}$$

and $\mathbf{w}_{i-1/2}(\xi)$ is the similarity solution to the Riemann problem at $x_{i-1/2}$.

Approximate Riemann solvers

For the Euler equations the calculation of $\mathbf{w}(x/t)$ remains a nontrivial task and Roe suggested that since so much information is lost in the projection (3.3.2) it would not be detrimental to the method to use an approximation to the similarity solution instead. We can use this approximate solution, $\hat{\mathbf{w}}(x/t)$, in two ways. The first is simply to replace $\mathbf{w}$ in the expression for the numerical flux (3.3.4) to give,

$$\mathbf{F}_{i-1/2}^n := \mathbf{f}(\hat{\mathbf{w}}_{i-1/2}(0)),$$

which would immediately give a scheme in conservation form. The more common approach which is followed here is to go back to the original idea behind the Godunov method and average the approximate solution at the new time level,

$$\mathbf{U}_i^{n+1} = \frac{1}{\Delta x}\int_{\xi=0}^{\Delta x/2} \hat{\mathbf{w}}_{i-1/2}(\xi/\Delta t)\mathrm{d}\xi + \frac{1}{\Delta x}\int_{\xi=-\Delta x/2}^{0} \hat{\mathbf{w}}_{i+1/2}(\xi/\Delta t)\mathrm{d}\xi. \tag{3.3.5}$$

A method of this form is defined by Harten et al [HLvL83] as a *Godunov-type scheme.* The two approaches do not in general give the same scheme and some care must be taken over the choice of approximate solution to ensure that this second method is conservative.

To calculate such a condition on $\hat{\mathbf{w}}$ consider the original conservation law,

$$\mathbf{u}_t + \mathbf{f}(\mathbf{u})_x = 0,$$

with initial conditions,

$$\mathbf{u}(x,0) = \begin{cases} \mathbf{u}_l & x < 0 \\ \mathbf{u}_r & x > 0. \end{cases} \tag{3.3.6}$$

If we integrate around the box $[-M, M] \times [0, T]$ we get, for M large enough,

$$\int_{x=-M}^{M} \mathbf{w}(x/T)\mathrm{dx} - M(\mathbf{u}_l + \mathbf{u}_r) + T\,(\mathbf{f}(\mathbf{u}_r)) - \mathbf{f}(\mathbf{u}_l)) = 0,$$

where $\mathbf{w}(x/t) = \mathbf{u}(x,t)$ is the exact solution to the Riemann problem.

To have conservation we require that,

$$\int_{x=-M}^{M} \hat{\mathbf{w}}(x/T)\mathrm{dx} = \int_{x=-M}^{M} \mathbf{w}(x/T)\mathrm{dx}$$

thus the approximate solution must satisfy,

$$\int_{x=-M}^{M} \hat{\mathbf{w}}(x/T)\mathrm{dx} - M(\mathbf{u}_l + \mathbf{u}_r) + T\,(\mathbf{f}(\mathbf{u}_r)) - \mathbf{f}(\mathbf{u}_l)) = 0. \tag{3.3.7}$$

Roe's approximate Riemann solver

Roe [Roe81] suggested that an approximate solution to the Riemann problem can be produced by solving a linearised version of the conservation laws

$$\mathbf{u}_t + A\mathbf{u}_x = 0 \tag{3.3.8}$$

where the matrix $A = A(\mathbf{u}_l, \mathbf{u}_r)$ is a constant approximation to the Jacobian of $\mathbf{f}$ at the interface between left and right cells. The solution to this linear system with the initial conditions (3.3.6) is simply expressed in terms of the eigenvectors $\mathbf{r}_j$ and eigenvalues λ_j of A as

$$\hat{\mathbf{w}}(x/t) = \mathbf{u}(x,t) = \sum_{\lambda_j t - x \geq 0} \alpha_j \mathbf{r}_j + \sum_{\lambda_j t - x < 0} \beta_j \mathbf{r}_j$$

where the left and right states have been decomposed into $\mathbf{u}_L = \Sigma\alpha_j\mathbf{r}_j$ and $\mathbf{u}_R = \sum\beta_j\mathbf{r}_j$. If the decomposition of the jump is given by $\mathbf{u}_R - \mathbf{u}_L = \sum(\beta_j - \alpha_j)\mathbf{r}_j = \sum\delta_j\mathbf{r}_j$ then the solution can also be expressed as

$$\hat{\mathbf{w}}(x/t) = \mathbf{u}_L + \sum_{\lambda_j t - x < 0} \delta_j\mathbf{r}_j \tag{3.3.9}$$

$$= \mathbf{u}_R - \sum_{\lambda_j t - x \geq 0} \delta_j\mathbf{r}_j. \tag{3.3.10}$$

The particular form of the matrix A must be chosen to have linearly independent eigenvectors so that this decomposition is possible and it must satisfy the consistency condition that $A(\mathbf{u}_l, \mathbf{u}_r) = A(\mathbf{u})$. A more difficult requirement to satisfy is the conservation condition (3.3.7) which in general rules out the two most obvious candidates $A((\mathbf{u}_r + \mathbf{u}_l)/2)$ and $(A(\mathbf{u}_r) + A(\mathbf{u}_l))/2$. For this linear problem the integral of $\hat{\mathbf{w}}$ is easily calculated to be

$$\int_{x=-M}^{M} \hat{\mathbf{w}}(x/T)\mathrm{dx} = M(\mathbf{u}_l + \mathbf{u}_r) - T(A\mathbf{u}_r - A\mathbf{u}_l)$$

reducing the conservation condition to

$$A(\mathbf{u}_l, \mathbf{u}_r)(\mathbf{u}_r - \mathbf{u}_l) = \mathbf{f}(\mathbf{u}_r) - \mathbf{f}(\mathbf{u}_l). \tag{3.3.11}$$

Although Harten [Har83b] demonstrated that a matrix satisfying (3.3.11) exists for all conservation laws with an entropy condition, his construction is too complicated to use in practice. By the ingenious use of “parameter vectors”, Roe [Roe81] derived an averaged matrix for the Euler equations which has the following eigenvectors and eigenvalues,

$$\begin{array}{ccc} \lambda_1 = \bar{u} - \bar{a} & \lambda_2 = \bar{u} & \lambda_3 = \bar{u} + \bar{a} \\ \mathbf{r}_1 = \begin{bmatrix} 1 \\ \bar{u} - \bar{a} \\ \bar{h} - \bar{a}\bar{u} \end{bmatrix} & \mathbf{r}_2 = \begin{bmatrix} 1 \\ \bar{u} \\ \bar{u}^2/2 \end{bmatrix} & \mathbf{r}_3 = \begin{bmatrix} 1 \\ \bar{u} + \bar{a} \\ \bar{h} + \bar{a}\bar{u} \end{bmatrix} \end{array} \tag{3.3.12}$$

where

$$\bar{u} = \frac{\sqrt{\rho_l}u_l + \sqrt{\rho_r}u_r}{\sqrt{\rho_l} + \sqrt{\rho_r}},$$

and

$$\bar{h} = \frac{\sqrt{\rho_l}h_l + \sqrt{\rho_r}h_r}{\sqrt{\rho_l} + \sqrt{\rho_r}}$$

are the Roe-averaged velocity and enthalpy respectively, and

$$h = \frac{E+p}{\rho},$$

$$\bar{a}^2 = (\gamma - 1)(\bar{h} - \bar{u}^2/2).$$

These are substituted into expressions (3.3.9) and (3.3.10) to get the solution to the linear system which in turn are substituted in to equation (3.3.5) give Roe's scheme.

Taking each integral in turn, we have from equation (3.3.8) that

$$\begin{aligned} I_1 :=& \frac{1}{\Delta x}\int_{\xi=0}^{\Delta x/2} \hat{\mathbf{w}}_{i-1/2}(\xi/\Delta t)\mathrm{d}\xi \\ =& \frac{1}{2}\mathbf{U}_i^n - \frac{1}{\Delta x}\int_0^{\Delta t} A_{i-1/2}\left(\mathbf{U}_i^n - \hat{\mathbf{w}}_{i-1/2}(0)\right)\mathrm{d}\xi \end{aligned} \tag{3.3.13}$$

and on using the expression (3.3.10) for $\hat{\mathbf{w}}$ (with $R = i, L = i-1$)

$$I_1 = \frac{1}{2}\mathbf{U}_i^n - \frac{\Delta t}{\Delta x}A_{i-1/2}\sum_{\lambda_{j,i-1/2}\geq 0} \delta_{j,i-1/2}\mathbf{r}_{j,i-1/2}$$

but since $\mathbf{r}_j$ are eigenvectors of A this simplifies to

$$I_1 = \frac{1}{2}\mathbf{U}_i^n - \frac{\Delta t}{\Delta x}\sum_{\lambda_{j,i-1/2}\geq 0} \lambda_{j,i-1/2}\delta_{j,i-1/2}\mathbf{r}_{j,i-1/2}.$$

Similarly the second integral

$$I_2 := \frac{1}{\Delta x}\int_{\xi=-\Delta x/2}^{0} \hat{\mathbf{w}}_{i+1/2}(\xi/\Delta t)\mathrm{d}\xi$$

on using expression (3.3.9) for $\hat{\mathbf{w}}$ (with $L = i, R = i+1$) evaluates to

$$= \frac{1}{2}\mathbf{U}_i^n - \frac{\Delta t}{\Delta x}\sum_{\lambda_{j,i+1/2}\leq 0} \lambda_{j,i+1/2}\delta_{j,i+1/2}\mathbf{r}_{j,i+1/2}$$

giving an overall scheme

$$\begin{aligned} \mathbf{U}_i^{n+1} =& I_1 + I_2 \\ =& \mathbf{U}_i^n - \frac{\Delta t}{\Delta x}\left(\sum_{\lambda_{j,i+1/2}\leq 0} \lambda_{j,i+1/2}\delta_{j,i+1/2}\mathbf{r}_{j,i+1/2} + \sum_{\lambda_{j,i-1/2}\geq 0} \lambda_{j,i-1/2}\delta_{j,i-1/2}\mathbf{r}_{j,i-1/2}\right). \end{aligned} \tag{3.3.14}$$

Note that as written above this does not appear to be in conservation form (3.2.3). However, if the alternative expression (3.3.9) for $\hat{\mathbf{w}}_{i-1/2}$ is used then I_1 becomes

$$I_1 := \frac{1}{2}\mathbf{U}_i^n + A_{i-1/2}(\mathbf{U}_i^n - \mathbf{U}_{i-1}^n) + \frac{\Delta t}{\Delta x} \sum_{\lambda_{j,i-1/2} \leq 0} \lambda_{j,i-1/2}\delta_{j,i-1/2}\mathbf{r}_{j,i-1/2}$$

and using the conservation property (3.3.11) this can be written

$$I_1 = \frac{1}{2}\mathbf{U}_i^n + (\mathbf{f}(\mathbf{U}_i^n) - \mathbf{f}(\mathbf{U}_{i-1}^n)) + \frac{\Delta t}{\Delta x} \sum_{\lambda_{j,i-1/2} \leq 0} \lambda_{j,i-1/2}\delta_{j,i-1/2}\mathbf{r}_{j,i-1/2}$$

and the scheme now takes the form

$$\mathbf{U}_i^{n+1} = \mathbf{U}_i^n - \frac{\Delta t}{\Delta x}\left(\mathbf{F}_{i+1/2} - \mathbf{F}_{i-1/2}\right)$$

where

$$\mathbf{F}_{i-1/2} = \sum_{\lambda_{j,i-1/2} \leq 0} \lambda_{j,i-1/2}\delta_{j,i-1/2}\mathbf{r}_{j,i-1/2} + \frac{\Delta x}{\Delta t}\mathbf{f}(\mathbf{U}_{i-1}^n).$$

The scheme as formulated in equation (3.3.14) is preferred in practice since it does not require the explicit calculation of $\mathbf{f}$ and is therefore less expensive.

Entropy corrections

It is well known that Roe's scheme does not satisfy an entropy condition and can give unphysical solutions in certain circumstances. This is because the approximate Riemann solver replaces all of the waves in the Riemann solution with discontinuities, even rarefaction waves. Simple entropy corrections have been proposed by several authors such as Osher [Osh84], Roe [Roe85] and Yee et al [YWH85]. However, since the problem only occurs at sonic points, where a characteristic changes direction, there is no need for a correction for our purposes since the flow in gas transmission networks is never supersonic. The most extreme conditions occur at a pipebreak where the flow is sonic at the break. The frictional term ensures that the flow *within* the pipe remains subsonic.

3.3.1 The Second-order Correction

Analysis of the truncation error or the modified equation of Roe's scheme shows that the scheme is made second order accurate by a correction to the fluxes corresponding to $F^2 - F^1$ in Section 3.2. This correction takes the form,

$$F_{i+1/2} \rightarrow F_{i+1/2} + \frac{1}{2}\sum_j |\lambda_{j,i+1/2}| \left(1 - \frac{\Delta t}{\Delta x}|\lambda_{j,i+1/2}|\right) \mathbf{W}_{j,i+1/2}$$

where the waves $\mathbf{W}_j$ are defined by

$$\mathbf{W}_j := \delta_j \mathbf{r}_j.$$

However, this would introduce Lax-Wendroff style oscillations near discontinuities. Several remedies for this were described in Section 3.2 and the details of the implementation of flux-limiters are given here.

The correction to the flux is modified to be

$$F_{i+1/2} \rightarrow F_{i+1/2} + \frac{1}{2}\sum_j |\lambda_{j,i+1/2}| \left(1 - \frac{\Delta t}{\Delta x}|\lambda_{j,i+1/2}|\right) \bar{\mathbf{W}}_{j,i+1/2}.$$

where $\bar{\mathbf{W}}_{j,i+1/2}$ is a limited version of the jth wave $\mathbf{W}_{j,i+1/2}$. The wave is limited by comparing it to the upstream wave, $\mathbf{W}_{j,i-1/2}$ if $\lambda_{j,i+1/2} \geq 0$ or $\mathbf{W}_{j,i+3/2}$ otherwise,

$$\bar{\mathbf{W}}_{j,i+1/2} = \mathbf{W}_{j,i+1/2}\Phi\left(|\mathbf{W}|_{j,i+1/2}, \frac{\mathbf{W}_{j,i+1/2}.\mathbf{W}_{j,i+1/2\pm1}}{|\mathbf{W}|_{j,i+1/2}}\right)$$

where Φ is a limiter, for example Roe's superbee [Roe85]

$$\Phi(a,b) = \max(0, \max(\min(1, 2b/a), \min(2, b/a))).$$

In the results in this thesis the superbee limiter was used exclusively, but it is possible to use different limiters on each characteristic field. A popular combination (see LeVeque [LeV95]) is to use the minmod limiter on the nonlinear fields and the more compressive superbee limiter on the linear field to steepen contact discontinuities. See LeVeque [LeV90] for other types of limiter.

3.3.2 The implementation of the source terms by Strang splitting

So far we have only covered solvers for homogeneous systems of conservation laws. The hyperbolic equations (3.1.1) - (3.1.3) include source terms modelling friction and heat conduction and this section describes the corresponding modification of Roe's scheme.

Emmerson [Emm90] gave a review of some of the state of the art techniques for solving hyperbolic equations with source terms. The source terms for the hyperbolic equations are not particularly stiff and can be handled adequately by a splitting method, which is easy to implement.

Given a system of conservation laws with a source term

$$\mathbf{u}_t + \mathbf{f}_x = \mathbf{s}$$

a splitting method generates an approximate solution by alternately solving the homogeneous system

$$\mathbf{u}_t + \mathbf{f}_x = 0$$

and the system of ODEs

$$\mathbf{u}_t = \mathbf{s}. \tag{3.3.15}$$

If $S_I(t)$ is the solution operator for the inhomogeneous system (that is $u(x,t) = S_I(t)u(x,0)$) and $S_H(t)$ and $S_O(t)$ the solution operators for the homogeneous and ODE system respectively then the idea is that we can approximate

$$S_I(\Delta t) \simeq S_H(\Delta t)S_O(\Delta t).$$

In fact, more accurate results are obtained by using Strang's splitting

$$S_I(\Delta t) \simeq S_O(\Delta t/2)S_H(\Delta t)S_O(\Delta t/2)$$

which is second order accurate in time. Although at first sight this appears to require 50% more work a simplification is possible which means that Strang's splitting requires little more work than the first splitting proposed:

$$\begin{aligned} S_I(t) &\simeq (S_O(\Delta t/2) S_H(\Delta t) S_O(\Delta t/2))^{t/\Delta t} \\ &= S_O(\Delta t/2)(S_H(\Delta t) S_O(\Delta t))^{(t/\Delta t)-1} S_H(\Delta t) S_O(\Delta t/2). \end{aligned}$$

In practice the exact solution operators S are usually replaced by numerical approximations (such as the Roe scheme in place of S_H) but Tang and Zheng [TZ95] have shown that, under certain assumptions on the numerical schemes, the accuracy is retained.

In all of the results in this thesis the frictional term takes the form $\mu q|q|/\rho$ while heat conduction has been neglected, $\Omega = 0$. In this particular case we are fortunate to be able to obtain an exact solution to the ODE

$$\begin{bmatrix} \rho \\ q \\ E \end{bmatrix}_t = - \begin{bmatrix} 0 \\ \mu q|q|/\rho \\ 0 \end{bmatrix}$$

which is

$$\begin{bmatrix} \rho(t) \\ q(t) \\ E(t) \end{bmatrix} = \begin{bmatrix} \rho(0) \\ \frac{q(0)}{1+|q(0)|\mu t/\rho(0)} \\ E(0) \end{bmatrix}.$$

If the source term includes a heat transfer term or a more sophisticated friction term then quadrature would probably have to be used to solve the equations.

3.4 Boundary Conditions

Most of the literature on the numerical solution of hyperbolic conservation laws assumes an infinite or periodic spatial domain and so does not consider boundary conditions. In industrial problems where we may wish to constrain the flow or pressure at a node to a certain value, the accurate formulation of boundary conditions is crucial.

Typical boundary conditions of interest are

- a fixed pressure and temperature downstream of a compressor
- a fixed mass flow at a demand or source
- sonic flow at a pipe break (where the flow is choked).

The number of boundary conditions that can be applied to a hyperbolic system to have a well-posed problem depends on the number of incoming and outgoing characteristics at the boundary. A full discussion of this problem can be found in a PDEs textbook such as Kevorkian [Kev90]. The characteristics of the Euler equations travel at speeds $u-a$, u and $u+a$. In gas transmission networks the flow is never supersonic and so $|u| \leq a$ and the first and last characteristics are always left-going and right-going respectively. The possibilities are now reduced to two. If the gas flow is away from the boundary then there are two incoming characteristics and two boundary conditions must be applied, a so-called inflow boundary. Conversely, if the gas flow is towards the boundary then there is one incoming characteristic and one boundary condition must be supplied and we have an outflow boundary.

However, the numerical method often requires more boundary conditions than can be applied to the underlying equations.

As an example, consider the Lax-Friedrichs scheme (3.2.2) which has a three point stencil. At the left edge of the domain the calculation of $\mathbf{U}_1^{n+1}$ requires values for all of the components of $\mathbf{U}_0^n$, three in the case of the Euler equations, irrespective of the number of outgoing characteristics. These additional *numerical* boundary conditions must be chosen carefully to ensure stability and accuracy. For some schemes this problem does not arise, the scalar upwind scheme (3.2.5) and its multivariate equivalents (such as that used by Issa and Spalding [IS72]) naturally adjust the number of boundary conditions required according to the direction of the characteristics. It will be shown in Section 3.4.2 that Roe's scheme also has this advantage.

3.4.1 Generating numerical boundary conditions

There are numerous ways of producing the additional boundary conditions required ranging from simple extrapolation to discretisation of the characteristic equations. Chu and Sereny [CS74] produced one of the few papers on the subject, describing the merits of a wide range of techniques for the gas dynamics equations. A more theoretical approach has been taken by Kreiss et al (see for example [Kre68]) who examined the stability of linear systems for general linear boundary conditions. However this work is not yet at a stage to be applied to nonlinear systems such as the Euler equations.

In this section we describe and compare several methods for generating numerical boundary conditions, one of which is specific to Roe's method. For definiteness we consider the left boundary condition

$$p_0 = \text{const}$$

and assume that the flow is towards the left so that only this boundary condition is required. The simplest strategy is to extrapolate from the interior of the domain to calculate the numerical boundary conditions. The extrapolation could be constant, linear or quadratic and there is a choice of quantities to extrapolate from the conserved variables

$$\rho_0 = \rho_1$$

$$q_0 = q_1$$

to the Riemann invariants (if they exist).

Another strategy is to use a one-sided numerical method such as (3.2.5) to provide, say, ρ_0 and q_0. The most consistent approach is to use the information from the outgoing characteristics. For systems possessing Riemann invariants this is trivial. For the Euler equations we can discretise the characteristic equations (2.3.5), (2.3.3), which is akin to using a one-sided numerical method, or use the Riemann invariants of the isentropic equations (2.3.4) in the hope that they will be sufficiently close.

The following section describes an approach specific to Roe's scheme which is based on characteristic principles and has a natural extension to the branched pipe networks of Chapter 6.

3.4.2 Cell-edge boundary conditions

In this section a method for imposing boundary conditions is described which is a natural consequence of Roe's method and automatically ensures that the correct characteristic information from the interior of the domain is used.

As in Roe's method we solve a linear system, but instead of finding the solution between two cells we solve in the adjacent cell to the boundary as in Figure 3.4.1, in this case a left hand boundary.

Assume that we can linearise the system and wish to solve equation (3.3.8) where the matrix A may simply be $A(\mathbf{U}_0)$ or perhaps the Roe averaged $A(\mathbf{U}_{-1}, \mathbf{U}_0)$ where $\mathbf{U}_{-1}$ is obtained by extrapolation (or reflection for a symmetric boundary condition such as zero flow). However A is obtained, it must have linearly independent

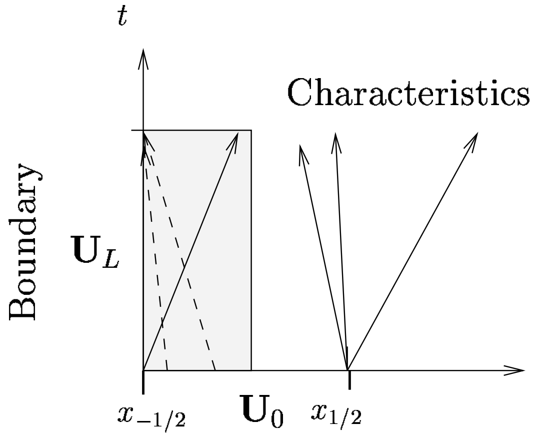


Figure 3.4.1: The cell adjacent to the boundary

eigenvectors $\mathbf{r}_j$ so that the solution in cell 0 can be decomposed as

$$\mathbf{U}_0 = \sum_j \beta_j \mathbf{r}_j.$$

Similarly we decompose the unknown solution at the boundary, that is the left hand edge of cell 0 as

$$\mathbf{U}_L = \sum_j \alpha_j \mathbf{r}_j$$

where the α_j remain to be determined. For definiteness assume that we have an outflow boundary with the single boundary condition

$$g(\mathbf{U}_L) = g(\alpha_1, \alpha_2, \alpha_3) = 0$$

imposed on the cell edge, and of course $\lambda_2 < 0$. The two other equations required to determine α_j come from the outgoing characteristics

$$\begin{aligned} \alpha_1 &= \beta_1 \\ \alpha_2 &= \beta_2. \end{aligned}$$

In general this system will be nonlinear and require Newton iteration for solution. The Jacobian of the system is easy to calculate as

$$J(\alpha_1, \alpha_2, \alpha_3) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ a_1 & a_2 & a_3 \end{bmatrix}$$

where

$$\begin{aligned} a_j := \frac{\partial g}{\partial \alpha_j} &= \sum_k \frac{\partial g}{\partial \mathbf{U}^{(k)}} \frac{\partial \mathbf{U}_L^{(k)}}{\partial \alpha_j} \\ &= \sum_k \frac{\partial g}{\partial \mathbf{U}^{(k)}} \mathbf{r}_j^{(k)} \\ &= \nabla_{\mathbf{U}} g . \mathbf{r}_j. \end{aligned} \tag{3.4.1}$$

Although the second and third equations are trivial and this system could be reduced to a single equation it is easier to code in this form and incurs little overhead. For an inflow boundary there are two boundary conditions and only the third equation $\alpha_3 = \beta_3$ is trivial.

Once the α_j have been calculated the solution at the cell edge can be constructed

$$\mathbf{U}_L = \mathbf{U}_0 - \sum_{\lambda_j < 0} \delta_j \mathbf{r}_j$$

with $\delta_j := \beta_j - \alpha_j$. This is then be substituted into the formula for I_1 (3.3.13) to give an expression for $\mathbf{U}_0^1$ which is formally identical to that for interior cells. The only difference is in the calculation of α_j and the choice of A. It will be shown in Chapter 6 that this greatly simplifies the formulation of boundary conditions for branched networks.

This cell-edge method calculates waves right up to the outside of the boundary cells which can be used, as shown in Section 3.3.1, to calculate second order corrections. Since the corrections are limited by an upstream wave it is not clear how to apply a second order correction at the boundary cell outside edge, or even if we should since the cell-edge would no longer satisfy the boundary condition. If the correction is omitted, in effect suddenly switching on the limiter at the boundary, then the solution has a small kink at the boundary cell. This can be removed by extrapolating the waves from the boundary, purely for use in the limiter.

Emmerson [Emm90] describes an alternative method for imposing boundary conditions on Roe's method. Instead of imposing the boundary conditions on the cell edge he enforces them on the cell itself. This gives a method which is formally very similar.

3.4.3 A comparison of numerical boundary conditions

Some of the numerical boundary condition methods are compared in this section. The tests are not rigorous comparisons of error, merely a look at the quality of the solutions near the boundary. Initially the gas is at rest at a constant temperature and pressure. For simplicity the test boundary condition is a sudden drop in pressure to a new constant value. Since the pressure drops, the boundary is an outflow boundary and no other boundary condition is required.

The boundary conditions tested are:

1. constant extrapolation of ρ and q
2. linear extrapolation of ρ and q
3. cell-edge boundary condition
4. isentropic characteristics (2.3.4)
5. discretised characteristics (2.3.5)

The solutions near the boundary for all five numerical boundary conditions are shown in Figure 3.4.2. In this figure the friction factor has been set to zero and since the initial solution is uniform, the gas remains homentropic. The solutions are shown after three timesteps at $t = 0.15s$, compared against the 'exact solution' (calculated on a finer grid using the same boundary condition method for fairness). It is immediately apparent that the best results come from methods 3, 4 and 5 which make use of the characteristics, confirming the findings of Chu and Sereny [CS74]. Method 5 is a first-order accurate one-sided discretisation of the equations in characteristic form similar to that used by Engl [Eng96] (she used a second-order accurate discretisation). Unsurprisingly, since the problem is homentropic, method 4 gives an almost identical result. Note that the grid is displaced by half a cell for method 3 since the boundary conditions are applied at the cell edge.

When friction is introduced the problem is no longer homentropic. In Figure 3.4.3 the results of a second test with $\mu = 0.1$ are presented. This is an unrealistically large friction coefficient, but necessary to show an effect in such a short space of time. Methods 3 and 5 still represent the flow field accurately, but method 4 overestimates the flow since it does not include friction.

These results are not conclusive, after two timesteps the picture was different, with method 5 being virtually perfect, and method 3 apparently giving poorer accuracy in the interior of the domain (possibly due to the shifted mesh). Furthermore, after more time steps numerical diffusion begins to dominate and there is little difference between any of the schemes. However, we can conclude that characteristic based methods provide smooth solutions near the boundaries, with method 3 performing as well as the others.

3.5 Summary

- Numerical schemes for hyperbolic equations should be explicit, upwinded and, if shocks are present, conservative
- First-order schemes are too inaccurate, second-order schemes oscillate near discontinuities so compromise with a flux-limited method
- Roe's scheme satisfies these criteria, and is easy to implement
- Strang splitting offers a simple way to include the source terms

Summary of Roe's scheme:

- At the old time level we have the cell-averaged quantities $\mathbf{U}_i$, $i = 0..J$,
- solve the inhomogeneous problem (3.3.15) over $\Delta t/2$,
- calculate the Roe-averaged eigenvalues $\lambda_{j,i+1/2}$ and vectors $\mathbf{r}_{j,i+1/2}$, $i = 0..J-1$, $j = 1..3$ from (3.3.12),
- calculate the decomposition of the jumps $\Delta\mathbf{U}_{i+1/2} = \sum_j \delta_{j,i+1/2}\mathbf{r}_{j,i+1/2}$ in the interior $i = 0..J-1$,
- calculate the decomposition of the jumps at the boundary as in Section 3.4.2,
- calculate the waves $\mathbf{W}_{j,i+1/2} = \delta_{j,i+1/2}\mathbf{r}_{j,i+1/2}$, $i = -1..J$, $j = 1..3$,
- set up the flux differences $A^+_{i+1/2}\Delta\mathbf{U} = \sum_{\lambda_{j,i+1/2}\geq 0} \lambda_{j,i+1/2}\mathbf{W}_{j,i+1/2}$ and $A^-_{i+1/2}\Delta\mathbf{U} = \sum_{\lambda_{j,i+1/2}<0} \lambda_{j,i+1/2}\mathbf{W}_{j,i+1/2}$,

- update each cell $\mathbf{U}_i^1 = \mathbf{U}_i - \frac{\Delta t}{\Delta x}(A_{i-1/2}^+ \Delta \mathbf{U} + A_{i+1/2}^- \Delta \mathbf{U})$,
- calculate the second-order corrections from Section 3.3.1 and update the cells,
- solve the inhomogeneous problem (3.3.15) over $\Delta t/2$.

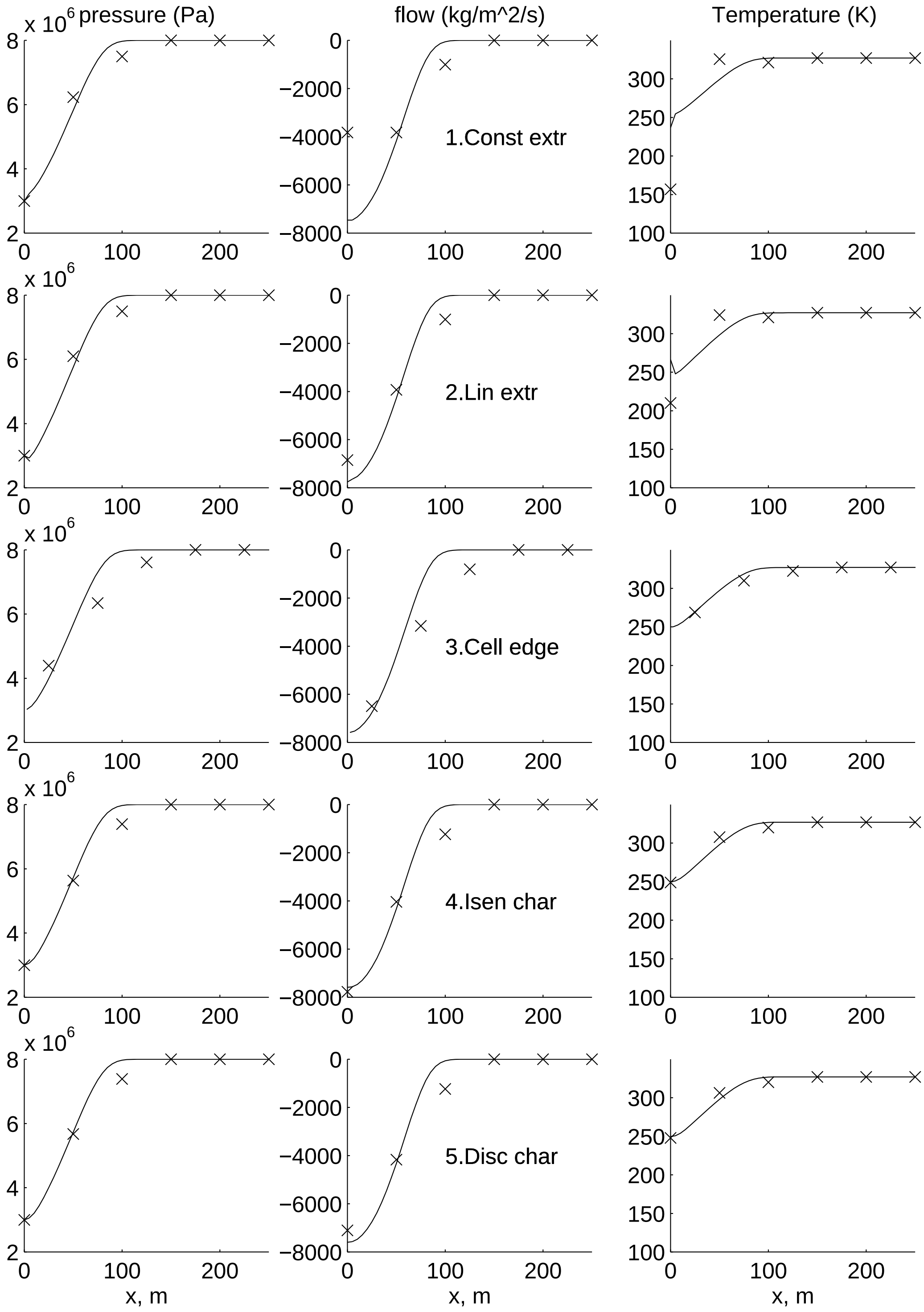


Figure 3.4.2: The solution near the boundary after three timesteps, each row represents a method in the order given in the text. No friction.

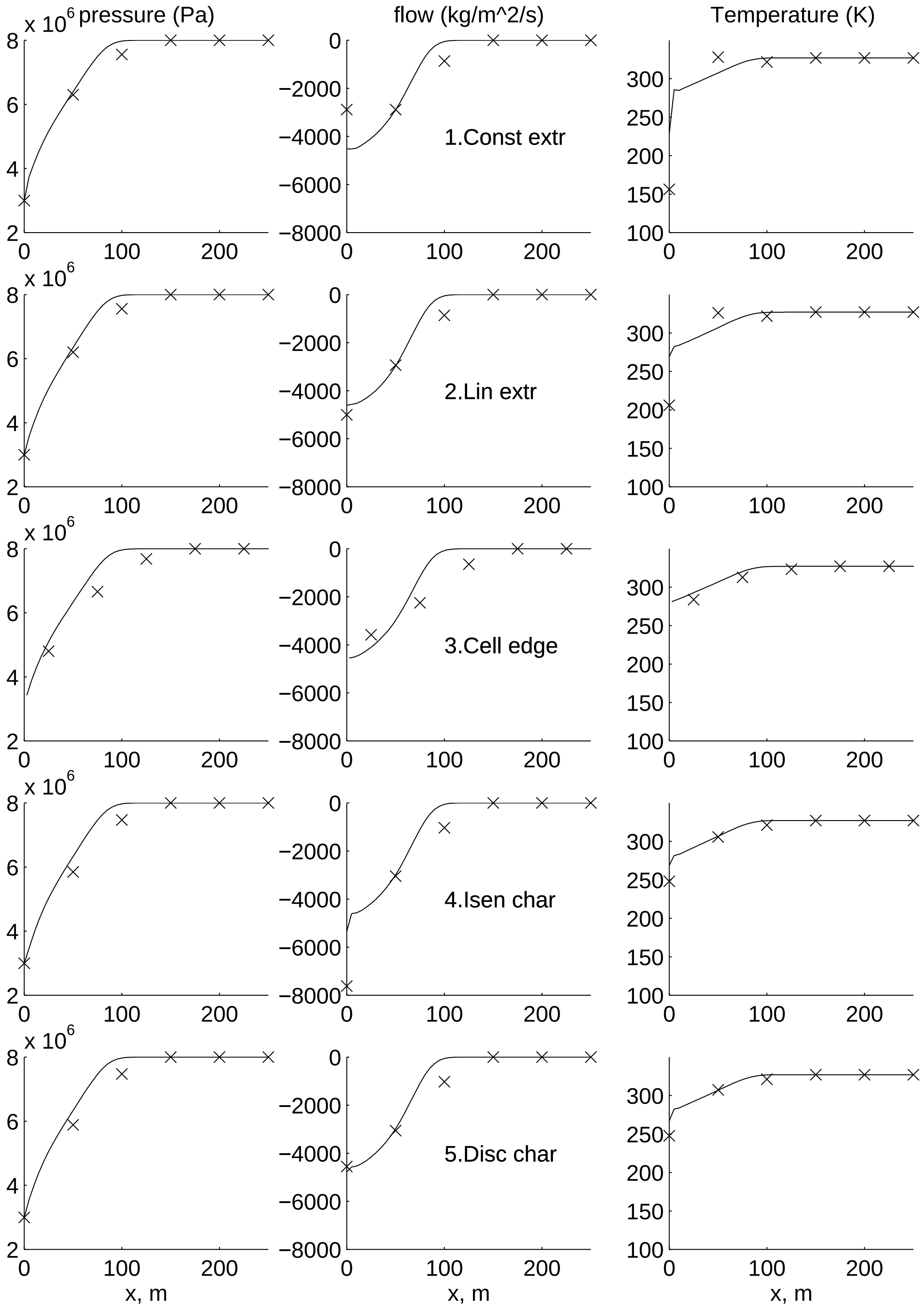


Figure 3.4.3: The solution near the boundary after three timesteps, each row represents a method in the order given in the text. $\mu = 0.1$.

Chapter 4

Numerical solution of the parabolic equations

Several models for one dimensional gas flow were discussed in Chapter 2 each of which has its merits and shortcomings. British Gas has adopted equations (2.1.3) and (2.1.4), obtained by neglecting the inertia terms in the momentum equation, to simulate large networks of pipes undergoing relatively slow changes. The parabolic model in its simplest form is

$$\frac{1}{RT}\frac{\partial p}{\partial t} + \frac{\partial q}{\partial x} = 0 \tag{4.0.1}$$

$$\frac{1}{2}\frac{\partial (p^2)}{\partial x} + \mu RTq|q| = 0, \tag{4.0.2}$$

where the gas is treated as isothermal and so the temperature T is a given constant. It will be demonstrated in Section 4.3 that the accuracy of the parabolic model is improved if temperature variation is included. This more sophisticated model

$$\begin{gathered} \frac{1}{R}\frac{\partial (p/T)}{\partial t} + \frac{\partial q}{\partial x} = 0 \\ \frac{1}{2}\frac{\partial (p^2)}{\partial x} + \mu RTq|q| = 0 \\ + \text{ some energy equation such as (2.3.1)} \end{gathered} \tag{4.0.3}$$

is naturally more difficult to solve.

The parabolic equations justify their name by having more in common with the heat equation than the wave equation. The equations transmit information instantaneously rather than at a finite speed. Furthermore, unlike the hyperbolic equations where we must specify two initial conditions, the parabolic equations only allow initial conditions to be imposed on a single variable, p, q or some combination of the two. Whichever is specified, the other quantity must be chosen to be consistent with the momentum equation (4.0.2). The question of which variable we should choose is answered in the following section by the use of asymptotic expansions.

4.1 Asymptotic analysis of the isothermal parabolic equations

The parabolic equations (4.0.1) and (4.0.2) are derived from the hyperbolic isothermal equations (2.1.1) and (2.1.2) by assuming that the inertia of the gas is small. We return to the nondimensional isothermal equations

$$\begin{aligned} \alpha\frac{\partial p}{\partial t} + \frac{\partial q}{\partial x} &= 0 \\ \epsilon\frac{\partial q}{\partial t} + \frac{\epsilon}{\alpha}\frac{\partial (q^2/p)}{\partial x} + \frac{\partial p}{\partial x} + \nu\frac{q|q|}{p} &= 0 \end{aligned} \tag{4.1.1}$$

of Section 2.1 and examine the consequences of this assumption in more detail. The small inertia assumption means that ϵ is a small parameter, and we seek an asymptotic solution of the equations in the form

$$\begin{aligned} p &= p_0 + \epsilon p_1 + \dots \\ q &= q_0 + \epsilon q_1 + \dots \quad . \end{aligned}$$

This expansion is singular, since as $\epsilon \to 0$ the nature of equations (4.1.1) changes from hyperbolic to parabolic.

On substitution the lowest order terms satisfy

$$\alpha\frac{\partial p_0}{\partial t}+\frac{\partial q_0}{\partial x}=0$$
$$\frac{\partial p_0}{\partial x}+\nu\frac{q_0|q_0|}{p_0}=0$$

which are precisely the parabolic equations in nondimensional form. To understand why these equations require only one initial condition we must examine the solution of the full equations for small time. For small ϵ there is an initial layer in which the inertia terms play an important rôle. To investigate this initial layer we rescale the time variable

$$t=\epsilon\tau,\quad \tau\sim O(1)$$

and reformulate equations (4.1.1) as

$$\alpha\frac{\partial p}{\partial \tau}+\epsilon\frac{\partial q}{\partial x}=0$$
$$\frac{\partial q}{\partial \tau}+\frac{\epsilon}{\alpha}\frac{\partial(q^2/p)}{\partial x}+\frac{\partial p}{\partial x}+\nu\frac{q|q|}{p}=0.$$

Now we seek an *inner* expansion of the form

$$p=\tilde{p}_0+\epsilon\tilde{p}_1+\dots$$
$$q=\tilde{q}_0+\epsilon\tilde{q}_1+\dots.$$

Once again, we collect lowest order terms to derive the equations

$$\frac{\partial \tilde{p}_0}{\partial \tau}=0 \qquad (4.1.2)$$
$$\frac{\partial \tilde{q}_0}{\partial \tau}+\frac{\partial \tilde{p}_0}{\partial x}+\nu\frac{\tilde{q}_0|\tilde{q}_0|}{\tilde{p}_0}=0. \qquad (4.1.3)$$

Notice that we can impose two initial conditions on these equations and thus the key to knowing what initial conditions to impose on the parabolic equations (4.0.1) and (4.0.2) lies in the matching of the inner and outer asymptotic expansions.

It follows immediately from equation (4.1.2) that for small time, and away from boundaries, $\tilde{p}_0$ can be considered constant in time and equal to its initial value.

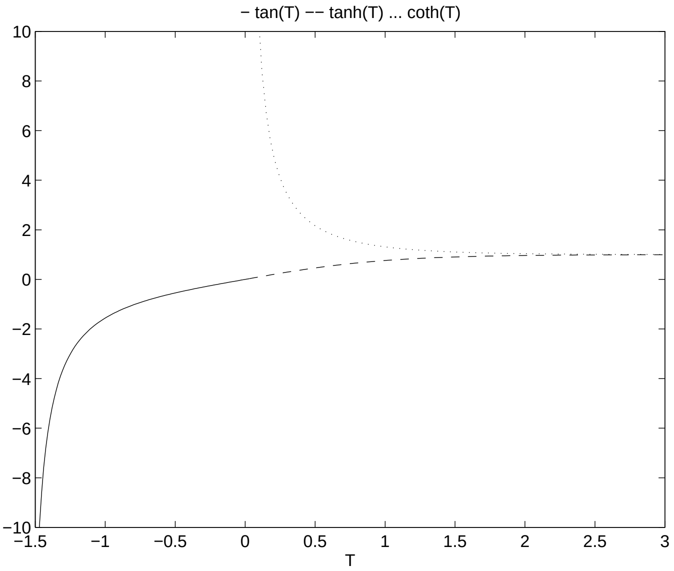


Figure 4.1.1: The inner asymptotic expansion

Since $\tilde{p}_0$ depends only on x, equation (4.1.3) is an ODE for $\tilde{q}_0$ as a function of τ. Fixing x and, without loss of generality, assuming that

$$\frac{\partial \tilde{p}_0}{\partial x} < 0$$

(if not then simply reverse the direction of x increasing), we define

$$B^2(x) := -\tilde{p}_0 \frac{\partial \tilde{p}_0}{\partial x}.$$

We then integrate equation (4.1.3) to give the following expression for $\tilde{q}_0$

$$\tilde{q}_0(x,\tau) = \begin{cases} \frac{B}{\sqrt{\nu}} \tan\left(\frac{B\sqrt{\nu}}{\tilde{p}_0}(\tau - C(x))\right) & \tilde{q}_0(x,0) < 0 \text{ and } \tau < C(x) \\ \frac{B}{\sqrt{\nu}} \tanh\left(\frac{B\sqrt{\nu}}{\tilde{p}_0}(\tau - C(x))\right) & \tilde{q}_0(x,0) < 0 \text{ and } \tau > C(x) \text{ or } 0 < \tilde{q}_0(x,0) < \frac{B}{\sqrt{\nu}} \\ \frac{B}{\sqrt{\nu}} \coth\left(\frac{B\sqrt{\nu}}{\tilde{p}_0}(\tau - C(x))\right) & \tilde{q}_0(x,0) > \frac{B}{\sqrt{\nu}} \end{cases}$$

with $C(x)$ chosen to satisfy the initial conditions. These functions are plotted in Figure 4.1.1. For all initial flows $\tilde{q}_0(x,0)$

$$\lim_{\tau\to\infty} \tilde{q}_0(x,\tau) = \frac{B(x)}{\sqrt{\nu}}.$$

Matching this solution to the lowest order outer solution, (see Hinch [Hin91]), shows that the correct initial conditions for the parabolic equations are to use the same

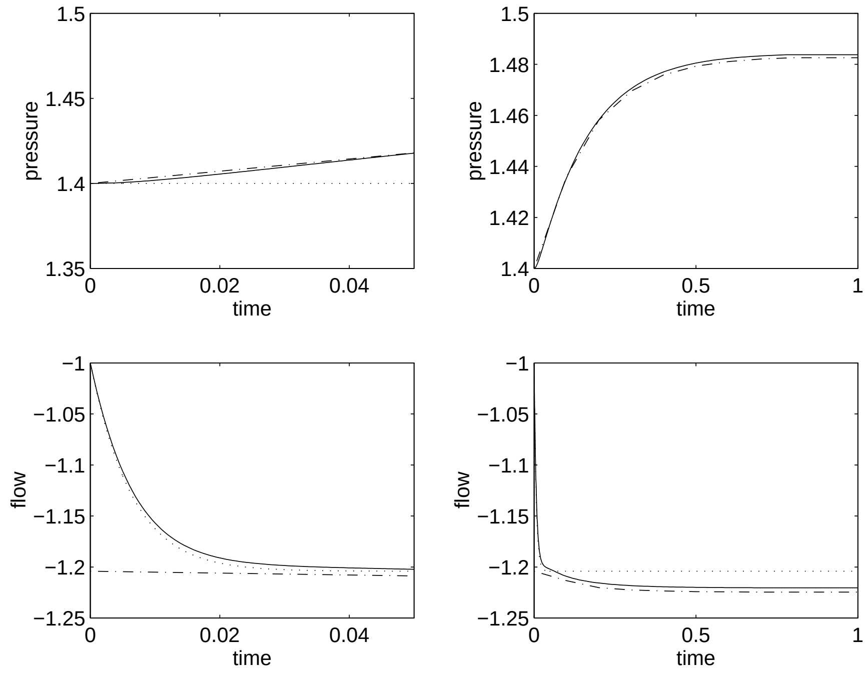


Figure 4.1.2: A comparison of the exact (-) ($\epsilon = 0.01$), lowest order inner (.) and lowest order outer (-.) solutions at fixed x

initial pressure as for the inner solution but choose the initial flow to be

$$q_0(x,0) = \frac{B(x)}{\sqrt{\nu}} = \sqrt{-\frac{p_0}{\nu}\frac{\partial p_0}{\partial x}}.$$

In other words, the initial condition is imposed on the pressure, and the flow is chosen to be consistent according to the momentum equation (4.0.2). The validity of the asymptotics is confirmed by Figure 4.1.2 which shows the expansions compared with the solution to the full equations with $\epsilon = 0.01$. The "exact" solution and the outer solution were calculated numerically on a periodic domain using a method similar to that which will be described in Section 4.2. The left hand figures show the solutions for $O(\epsilon)$ time, the right hand figures for $O(1)$ time.

The asymptotic expansions rely on the inertia terms being of much smaller magnitude than the pressure and friction terms in the momentum equation. Examination of the definitions of ϵ, ν and α shows that this does not hold as $q \to 0$. Consequently, the parabolic model is not valid for small flows. This is supported by numerical experiments which show that the parabolic solution to problems which have stagnation points, and therefore small flows, are less accurate than those which do not. Another demonstration of this is given in Section 6.3 where a stagnation point is automatically tracked by the hyperbolic solver because it correctly deduces that switching to the parabolic solver would introduce a large error.

4.2 Numerical schemes for the parabolic equations

The literature survey of Chapter 2 mentioned a number of numerical methods applicable to the equations of gas dynamics. In this section we describe in detail a particular scheme which is used successfully by BG to solve the parabolic equations. The scheme is used in the program FALCON (Fast Analysis of Large COnstrained Networks) to simulate large networks over long timescales (hours and days).

The parabolic nature of the equations and the long timescales rule out explicit schemes such as those of Chapter 3. Instead, we use an implicit method with time discretisation based on the θ method,

$$\frac{1}{RT}\frac{P_i^{n+1}-P_i^n}{\Delta t}+\frac{Q_{i+1/2}^{n+\theta}-Q_{i-1/2}^{n+\theta}}{\Delta x}=0 \qquad (4.2.1)$$

$$\frac{1}{2}\frac{(P^2)_{i+1}^{n+1}-(P^2)_i^{n+1}}{\Delta x}+\mu RTQ_{i+1/2}^{n+1}|Q_{i+1/2}^{n+1}|=0. \qquad (4.2.2)$$

The nonlinear algebraic equations at each timestep are solved by Newton's method in a few iterations with the initial guesses provided by the solution at the previous timestep. Note that the spatial discretisation is made on a staggered mesh. This has a number of benefits, especially for generalising the scheme to branched networks, as will be seen in Chapter 6. A useful way of looking at the staggered mesh is to consider the pressure as being evaluated at nodes, and the flow at the mid-point of pipes, in other words a pipe is a mesh element. If such a mesh is too coarse then a "real" pipe is split into a number of "artificial" pipes to improve resolution. A typical section of mesh is shown in Chapter 6, Figure 6.1.2. Like most schemes, equations (4.2.1) and (4.2.2) may be motivated from several standpoints. A particularly useful point of view for the mass equation (4.2.1) is to think of it as a discretisation of the integral form of the conservation law,

$$A\Delta x\left(\frac{P_i^{n+1}}{RT}-\frac{P_i^n}{RT}\right)+\Delta t\left(AQ_{i+1/2}^{n+\theta}-AQ_{i-1/2}^{n+\theta}\right)=0, \tag{4.2.3}$$

that is, the mass associated with node i is $A\Delta xP_i/RT$ and the mass flow out of the node in one timestep is denoted $\Delta tAQ_{i+1/2}^{\theta}$. This will be useful when interfacing the scheme with hyperbolic schemes in Chapter 5 and also in its generalisation to branched networks in Chapter 6.

On a uniform mesh with $\theta = 1/2$ this scheme is second order accurate in time and space. On a nonuniform mesh, that is where the pipes are of different lengths, the momentum discretisation (4.2.2) remains unchanged while in the mass discretisation (4.2.1) Δx is replaced by $(\Delta x_{i-1/2}+\Delta x_{i+1/2})/2$. A further advantage of the FALCON scheme arises from the formulation of the Newton iteration. Rewriting the mass and momentum discretisations (4.2.1) and (4.2.2) in the form

$$\begin{aligned} f_i &:= P_i+\frac{\theta\Delta tRT}{\Delta x}\left(Q_{i+1/2}-Q_{i-1/2}\right)+a_i \\ g_{i+1/2} &:= \frac{1}{2}\left(P_{i+1}^2-P_i^2\right)+\Delta x\mu RTQ_{i+1/2}|Q_{i+1/2}|, \end{aligned}$$

where the superscripts on the variables at the new timestep have been dropped for clarity and all the quantities at the old time step have been absorbed into a_i, the system we wish to solve is

$$\begin{aligned} f_i &= 0 \\ g_{i+1/2} &= 0. \end{aligned}$$

The Newton iteration takes the form

$$\begin{aligned} P_i|_{n+1} &= P_i|_n+\Delta P_i|_n \\ Q_{i+1/2}|_{n+1} &= Q_{i+1/2}|_n+\Delta Q_{i+1/2}|_n \end{aligned}$$

where the updates ΔP_i and $\Delta Q_{i+1/2}$ satisfy

$$\begin{bmatrix} 1 & 0 & 0 & \dots & -\frac{\theta\Delta tRT}{\Delta x} & \frac{\theta\Delta tRT}{\Delta x} & \dots \\ 0 & 1 & 0 & \dots & 0 & -\frac{\theta\Delta tRT}{\Delta x} & \dots \\ 0 & 0 & 1 & \dots & 0 & 0 & \dots \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots \\ -P_1 & P_2 & 0 & \dots & 2\Delta x\mu RT|Q_{3/2}| & 0 & \dots \\ 0 & -P_2 & P_3 & \dots & 0 & 2\Delta x\mu RT|Q_{5/2}| & \dots \\ 0 & 0 & -P_3 & \dots & 0 & 0 & \dots \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots \end{bmatrix} \begin{bmatrix} \Delta P_1 \\ \Delta P_2 \\ \Delta P_3 \\ \vdots \\ \Delta Q_{3/2} \\ \Delta Q_{5/2} \\ \vdots \end{bmatrix} = - \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ \vdots \\ g_{3/2} \\ g_{5/2} \\ \vdots \end{bmatrix}$$

or

$$\mathcal{I}\Delta\mathbf{P} + S_1\Delta\mathbf{Q} = -\mathbf{f}$$

$$S_2\Delta\mathbf{P} + \Lambda\Delta\mathbf{Q} = -\mathbf{g}$$

where the Jacobian matrix is

$$\begin{bmatrix} I & S_1 \\ S_2 & \Lambda \end{bmatrix}.$$

For this discretisation the matrix multiplying the pressure correction in the mass equation is the identity and we can eliminate $\Delta\mathbf{P}$ from the equations to leave

$$(\Lambda - S_2S_1)\Delta\mathbf{Q} = S_2\mathbf{f} - \mathbf{g}.$$

This reduces the size of the problem by a factor of about two, resulting in a considerable saving in the linear algebra.

4.2.1 Boundary conditions

Like the heat equation, the parabolic equations (4.0.1) and (4.0.2) require two boundary conditions. Consider a pipeline divided into J pipes and $J+1$ nodes. The mass discretisation (4.2.1) is applied to the $J-1$ internal nodes $i = 1 \dots J-1$ and the momentum equation to the J pipes $i = 0 \dots J-1$, giving $2J-1$ equations for the $2J+1$ unknown P_i^{n+1} and $Q_{i+1/2}^{n+1}$. The remaining two equations are supplied by the physical boundary conditions, so there is no requirement for troublesome numerical boundary conditions as there was for the hyperbolic equations of Chapter 3.

Implementing a pressure boundary condition is trivial but a flow boundary condition is usually enforced at a node where q is unknown, rather than at a pipe midpoint. Therefore a slightly more complicated boundary condition is required and the mass balance of equation (4.2.3) is modified to give, for example,

$$A\Delta x\left(\frac{P_0^{n+1}}{RT}-\frac{P_0^n}{RT}\right)+\Delta t\left(AQ_{+1/2}^{n+\theta}-F_L(t^{n+\theta})\right)=0$$

where $F_L(t)$ is the required mass flow. Another commonly required boundary condition is to prescribe the Mach number. This arises when we wish to simulate a pipe break as described in Chapter 2. The gas speed at the left boundary is approximated by using the mass flux in the nearest pipe,

$$U_0=\frac{Q_{1/2}}{\varrho_0}=\frac{RT_0Q_{1/2}}{P_0}.$$

The sound speed is given by $a^2=RT$ so the boundary condition to enforce a Mach number M is

$$\frac{RT_0Q_{1/2}}{\sqrt{RT_0}P_0}=M$$

or

$$MP_0-Q_{1/2}\sqrt{RT_0}=0.$$

Normally the parabolic model should not be used to simulate pipe-breaks since the changes in momentum are great enough to require the inclusion of the inertia terms in the momentum equation.

4.3 Including temperature variation in the parabolic model

There are often areas of a gas transmission network which are not isothermal but are steady enough to be modelled by the parabolic equations. At a compressor, for example, gas is strongly heated, raising the temperature by as much as 60K, and this effect persists downstream long after friction has damped transients. If temperature variation can be included in the model without serious disruption to the numerical scheme then we gain modelling accuracy while retaining the simplicity of the scheme of the previous section. Two methods which achieve this are described below, the second was implemented successfully in the software used for this thesis.

Since temperature is allowed to vary, the discretisation (4.2.1) must be slightly modified to approximate equations (4.0.3) and (4.0.4), taking into account that $\varrho = P/RT$ is the conserved variable. A suitable discretisation of the mass equation is

$$\frac{1}{R}\frac{\frac{P_i^{n+1}}{T_i^{n+1}} - \frac{P_i^n}{T_i^n}}{\Delta t} + \frac{Q_{i+1/2}^{n+\theta} - Q_{i-1/2}^{n+\theta}}{\Delta x} = 0 \tag{4.3.1}$$

while the momentum discretisation becomes

$$\frac{P_{i+1}^{n+1} - P_i^{n+1}}{\Delta x} + \mu R(T_i^{n+1} + T_{i+1}^{n+1})\frac{Q_{i+1/2}^{n+1}|Q_{i+1/2}^{n+1}|}{P_i^{n+1} + P_{i+1}^{n+1}} = 0$$

or equivalently

$$\frac{(P^2)_{i+1}^{n+1} - (P^2)_i^{n+1}}{\Delta x} + \mu R(T_i^{n+1} + T_{i+1}^{n+1})Q_{i+1/2}^{n+1}|Q_{i+1/2}^{n+1}| = 0. \tag{4.3.2}$$

4.3.1 Method 1: Explicitly solving for temperatures

Hill [Hil94] devised a method for BG to include temperature variation into FALCON. He took the energy equation in the form (see Section 2.3)

$$\rho A c_p \frac{DT}{Dt} - (1 + \rho\hat{\mu}c_p)A\frac{\partial p}{\partial t} - \rho A u \hat{\mu} c_p \frac{\partial p}{\partial x} + \rho A u \frac{Du}{Dt} = \Omega$$

and used the small inertia assumption to neglect the Du/Dt term. Since he was concerned with writing a method for industrial use he retained the Joule-Thompson coefficient $\hat{\mu}$. He used the non empirical equation (2.4.2) to model the heat conduction term, both for transient and steady state flow, which is of the form

$$\Omega = B(T - T_g)$$

where T_g is the ground temperature and B a constant.

Hill's first concern was with the steady state, for which the equation reduces to

$$Aqc_p \frac{\partial T}{\partial x} - Aq\hat{\mu}c_p \frac{\partial p}{\partial x} = B(T - T_g).$$

Assuming that all quantities except T are constant and that p is linear in x, this expression can be integrated from the inlet to the outlet to give an expression for the outlet temperature. The algorithm for including this in the isothermal solver is straightforward:

Algorithm 1

1. *Calculate the pressures and flows in the pipes for a given average pipe temperature*
2. *For this set of pressures and flows track and calculate the temperatures in the pipes (starting at inlets and tracking to the outlet)*
3. *Go back to step 1 until convergence.*

The tracking is slightly more involved for branching networks, this is discussed in Section 6.1.1.

The exact integrals must be dispensed with for the unsteady case. Instead the spatial derivatives are approximated by backward differences (backward with respect to the flow direction). The time integration is then implemented by the theta-method, resulting in a method which is explicit in T. Algorithm 1 may again be used at each timestep.

The unsteady flow causes a slight complication to the temperature tracking: it is possible to have a node connected to two pipes flowing in the opposite directions, that is, a stagnation point. Hill resolved this by setting $u = 0$ in the equations for this point and solving the resulting ODE

$$\rho A c_p \frac{DT}{Dt} - (1 + \rho \hat{\mu} c_p) A \frac{\partial p}{\partial t} = B(T - T_g).$$

It has been reported by Mallinson [Mal] that the time-dependent method suffers from some instability. An explanation for this is suggested at the end of the next section.

4.3.2 Method 2: An explicit solver for entropy

Explicit numerical schemes for the parabolic equations (4.0.1) and (4.0.2) are effectively ruled out by the stability restriction on the timestep. However, isolating the entropy form (2.3.3) of the energy equation

$$\frac{\partial s}{\partial t} + u \frac{\partial s}{\partial x} = \frac{\Omega / A + \mu \rho u^2 |u|}{\rho T}, \tag{4.3.3}$$

we can consider it as an advection equation for s.

In contrast to the parabolic equations (4.0.2) and (4.0.3), this equation can be solved by a number of explicit schemes such as the Lax-Friedrichs (3.2.2), subject to the stability restriction that

$$\frac{|u| \Delta t}{\Delta x} < 1.$$

Typical gas velocities are around 10m/s and so if the minimum pipe length is about 1,000m this imposes a maximum timestep of 100s, which is not unreasonable. We use this in the following algorithm

Algorithm 2

1. *Calculate the entropy at the new time level from an explicit discretisation of equation (4.3.3)*

2. Using estimates for the pressures and flows estimate the temperatures at the new timestep from the calculated entropy

3. Solve for pressures and flows at the new timesteps, using equations (4.3.1), (4.3.2) and the estimated temperatures by Newton iteration.

4. Recalculate the temperatures using the new pressures and flows

5. Repeat from step 3 until convergence.

A variation on this algorithm is to update the temperature inside the Newton iteration for the pressures and flows. This is slightly faster than the nested iterations of algorithm 2. Unfortunately, even if the temperature variation in the solution is small this has the effect of increasing the number of Newton iterations from around 3 to 10 or more. However, it would not be difficult to rewrite the mass and momentum discretisation in terms of pressure, flow and entropy rather than pressure, flow and temperature, removing the need for step 3 and restoring a quadratic convergence rate.

Although it would be simpler to use an energy equation explicit in T such as that used in Section 4.3.1, this imposes a much more severe restriction on the timestep since temperature waves travel at the speed of sound, unlike variations in entropy. This might explain the instability reported by Mallinson [Mal] for the previous method.

The entropy equation (4.3.3) is a linear advection equation with source terms and a variable wave speed $u(x,t)$. Many solution techniques are available for this equation and are fully described in the standard text books such as LeVeque [LeV90]. The first-order upwind method (3.2.5)

$$\frac{s_i^{n+1} - s_i^n}{\Delta t} = \begin{cases} -u_{i+1/2}^n \frac{s_{i+1}^n - s_i^n}{\Delta x} + B_i^n & u_{i+1/2}^n < 0 \\ -u_{i+1/2}^n \frac{s_i^n - s_{i-1}^n}{\Delta x} + B_i^n & u_{i+1/2}^n > 0 \end{cases}$$

was chosen for its simplicity, where B_i^n is some suitable discretisation of the source term. The gas speeds are calculated from

$$u_{i+1/2} = \frac{2Q_{i+1/2}}{\varrho_i + \varrho_{i+1}} \quad \text{where} \quad \varrho_i = \frac{P_i}{RT_i}.$$

If $u_{1/2} > 0$ then a temperature boundary condition must be supplied at the left hand side. Similarly the right hand side needs an extra boundary condition if the gas speed in the rightmost pipe is negative. Note that this scheme will break down if the flow reverses direction at some point in the pipeline. This problem will be addressed in more detail in Section 6.1.1.

The improvement in accuracy when using this scheme is demonstrated by Figures 4.3.1 and 4.3.2. They show solutions calculated by the parabolic solver with and without temperature variation compared with the exact solution of the Euler equations. Here, as in the rest of this thesis, the adiabatic model is used, so $\Omega = 0$. The test problem consisted of a pipeline of 200,000m initially at a uniform state of $P = 70 \times 10^5$Pa, $Q = 0$, $T = 280K$. The pressure at the left boundary was lowered to $P = 65 \times 10^5$Pa over a period of 900s while no flow was imposed at the right boundary. The temperature-varying solution is noticeably better, particularly for the flow field, even though the temperature variation is small.

4.4 Summary

- The parabolic equations are a suitable model for slow dynamics where the inertia of the gas is negligible.
- The correct initial conditions are to supply the pressure, and choose the flow to be consistent with it.
- The parabolic equations may be less accurate when the flow is small.
- The staggered mesh offers an effective numerical scheme which is easy to solve and straightforward to extend to networks.

- Including variations in temperature improves the accuracy of the solution.

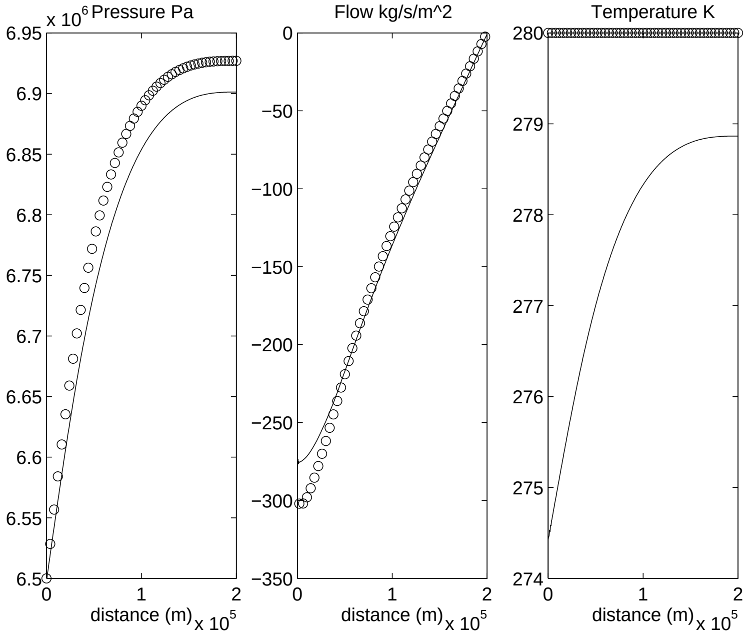


Figure 4.3.1: A parabolic solution with no temperature variation (o) compared with the hyperbolic solution (-).

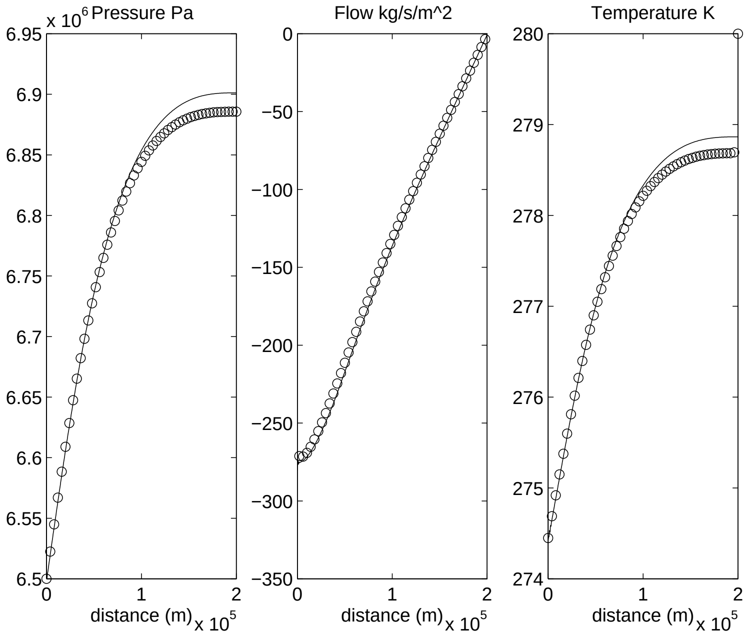


Figure 4.3.2: A parabolic solution with temperature variation (o) compared with the hyperbolic solution (-). Note the single parabolic point at the right-hand side at 280 K. Since the flow is zero at the boundary the temperature discontinuity does not propagate into the pipe.

Chapter 5

Interfaced methods

The hyperbolic and parabolic models of Chapters 3 and 4 each have their applications. The parabolic model is currently used by BG to simulate the NTS under general operating conditions, but there are situations for which it is not sufficiently accurate. On the other hand using the more accurate hyperbolic model is simply not practical on the entire network.

Our solution is to combine the two approaches, using the hyperbolic model only where necessary to capture rapid transients. An "interfaced" method has some of the characteristics of a domain decomposition method since the network is divided into regions which are treated separately. It is also an adaptive method because the hyperbolic regions must move to track the rapid transients. Figure 5.0.1 is a sketch of a typical situation, the tracking of a shock as it moves through a network.

We begin by considering the domain decomposition problem and discuss the tracking of rapid transients in Section 5.6.

5.1 Interfacing two models

Domain decomposition methods have recently received much attention as techniques to enable the solution of 2 and 3D problems which would otherwise be too large for a single computer. Typically, parallel computers are used to solve each subdomain

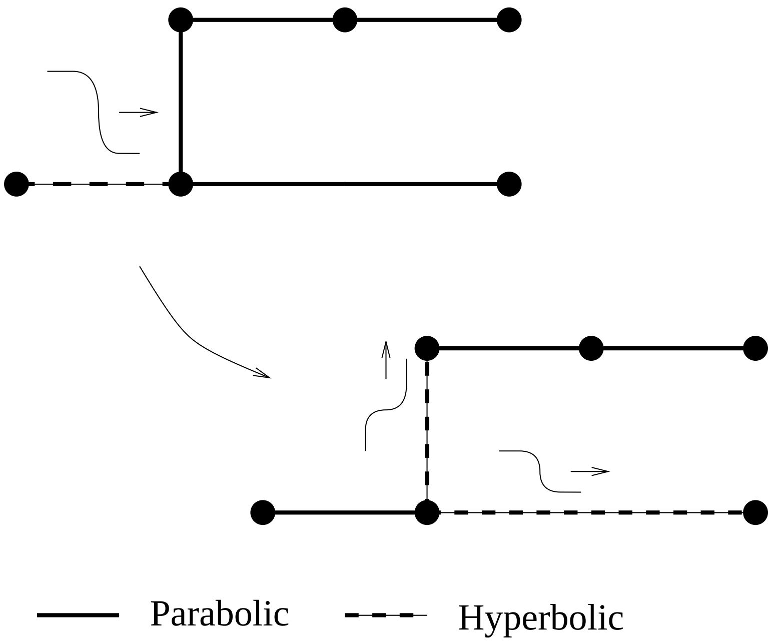


Figure 5.0.1: The tracking of a wave in a typical network

on a separate processor. For elliptic and parabolic problems the difficulty is that the domains cannot, by the nature of the equations, be completely decoupled. The numerical scheme is almost always implicit, and requires the simultaneous solution at every grid point (although, for certain domains, there are ingenious frontal methods [Duf96] which 'nibble' away at protruding areas and reduce the size of the problem to be solved) or an iterative method. The best known iterative scheme is due to Schwartz [Sch90] who proposed the algorithm which bears his name. His idea was to decompose the initial domain Ω into two (or more) *overlapping* subdomains Ω_1 and Ω_2. The solution is first calculated on Ω_1. Boundary data is available on the boundary that Ω_1 shares with Ω, but must be guessed on $\partial\Omega_1/\partial\Omega$. The estimated solution on Ω_1 is then used to provide boundary data on $\partial\Omega_2/\partial\Omega$ for the solution in Ω_2. This process continues, alternating from solving Ω_1 to Ω_2 until convergence. Due to the sequential nature of the algorithm it is not suited to parallelisation. An alternative algorithm which may be parallelised is to have domains which do not overlap, but meet at a boundary Γ. An estimate of the solution on Γ is used as boundary data in Ω_1 and Ω_2 which are then solved, possibly in parallel. Some interface condition is then used to update the estimate of the solution on Γ, usually that the solution and its normal derivative are continuous. The process is iterated to convergence. A version of this algorithm is used in Section 5.2 while a Schwartz-type method is proposed in Section 5.3.1 and shown to be equivalent, given the right interface condition.

Domain decomposition for hyperbolic problems again allows advantage to be taken of parallel machines, but is also a tool to use adaptive or fitted meshes. Starius [Sta80] used a composite mesh with an outer mesh fitted to the boundary of the domain and a simple rectilinear mesh in the interior to solve the two dimensional shallow water equations. The challenges of domain decomposition for hyperbolic equations are rather different to those for parabolic and elliptic equations. Usually the numerical method is explicit and so the domains may be decoupled without requiring iteration. However, when shocks are present it is essential that conservation is respected between the domains. Authors such as Chesshire and Henshaw [CW94] and Berger [Ber87] have investigated this aspect. Since the domains are decoupled it is possible to use different timesteps on the different domains. The stability of overlapping meshes has been examined by Berger [Ber85] using G-K-S theory for the Lax-Wendroff method and by Duncan [Dun95] for the one dimensional acoustic wave equation.

Similar issues must be resolved to link the parabolic and hyperbolic models. The parabolic solver is implicit and so we cannot completely decouple the domains, the timesteps are different, and we may be concerned about conservation between the domains. Additionally the models are based on different assumptions, leading to questions such as how to assess error, that is, with which model do we compare the calculated solution?

On the positive side, the problem is only one-dimensional and so the interfaces are just single points. This results in faster convergence of iterative methods and makes it practical to use alternative iterative strategies as described in Section 5.2.5.

The question of error measurement is dealt with in Section 5.7, while the problem of the inconsistency of the models is deferred to Section 5.2.4 by initially considering decomposition using the same model in both domains.

To simplify the problem in the following sections we always place interfaces between the hyperbolic and parabolic domains at nodes and only move them at the end of the parabolic timestep. Thus, a pipe (i.e. a parabolic mesh element) is either completely hyperbolic or parabolic for the duration of a parabolic timestep. The hyperbolic equations are solved on a finer mesh, both in space and in time, since we wish to capture fine detail. Additionally, we constrain the ratio of hyperbolic to parabolic timesteps to be an integer to ensure that the hyperbolic solution is available at the end of a parabolic step without additional interpolation.

In this chapter we denote the parabolic timestep by Δt and the hyperbolic timestep by δt. If δt_{max} is the hyperbolic timestep required for stability then we choose

$$\delta t = \frac{\Delta t}{\mathrm{ceil}(\Delta t/\delta t_{\mathrm{max}})}$$

where $\mathrm{ceil}(x) \overset{\mathrm{def}}{=} x$ rounded up to the nearest integer. In the following sections we will consider strategies to link a single parabolic domain to a single hyperbolic domain where the position of the interface is given. A typical interface is depicted in Figure 5.1.1. In most cases the methods are directly applicable when there are many interfaces; details are given where necessary. The ideas are extended to networks in Chapter 6.

5.2 Iterative Methods

We begin by considering the simplified problem of linking two domains of a "linear" network where both domains are modelled by the hyperbolic isothermal equations (2.1.1)- (2.1.2). The equations may be solved using the FALCON-like scheme

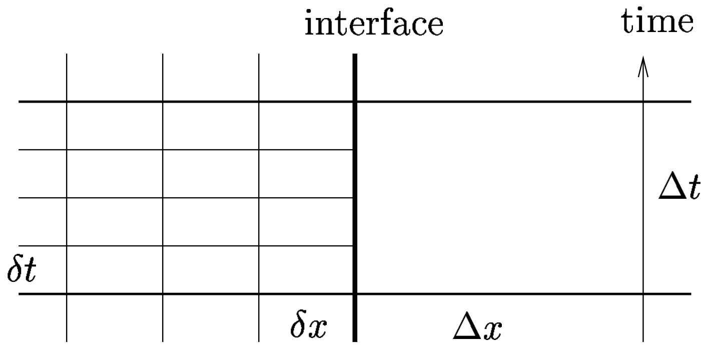


Figure 5.1.1: A typical interface.

(see Chapter 4)

$$\begin{gathered}
\frac{1}{RT}\frac{P_i^{n+1}-P_i^n}{\Delta t}+\frac{Q_{i+1/2}^{n+\theta}-Q_{i-1/2}^{n+\theta}}{\Delta x}=0 \\
\frac{Q_{i+1/2}^{n+1}-Q_{i+1/2}^{n}}{\Delta t}+\frac{P_{i+1}^{n+\theta}-P_i^{n+\theta}}{\Delta x}+\mu RT\frac{Q_{i+1/2}^{n+\theta}|Q_{i+1/2}^{n+\theta}|}{(P_{i+1}^{n+\theta}+P_i^{n+\theta})/2}=0
\end{gathered} \tag{5.2.1}$$

which uses a staggered mesh such as that in Figure 5.2.1 with different mesh sizes and numbers of timesteps in the left and right domains. Calculating all of the variables simultaneously at the end of a large timestep is impractical so the problem is decoupled into left and right domains and the interface treated as a boundary.

Suppose we are given the pressure at the interface at the end of the large timestep. We can use this as a boundary condition for the right domain. The boundary conditions for the left domain must also be provided at intermediate times along the interface by interpolation. The solution in the left domain may then be found

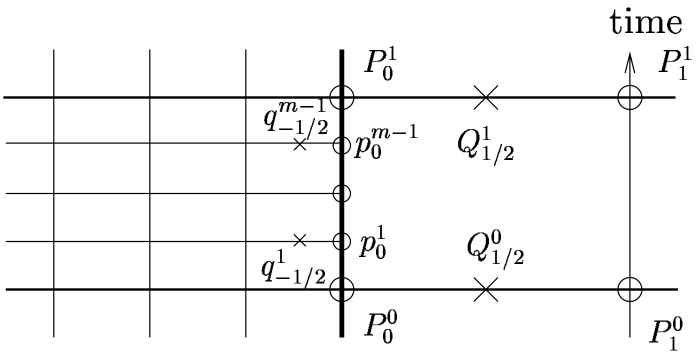


Figure 5.2.1: The staggered mesh at an interface.

at each small timestep until the solution at the end of the large timestep is known. The remaining question is how to calculate the pressure at the interface at the end of the large timestep. This is generated by an *interface condition* such as enforcing conservation of mass at the interface and from this an iterative algorithm arises:

Algorithm 3

1. *Choose a starting value P.*
2. *Solve for a large timestep in the right domain using P as a left boundary condition.*
3. *Solve the left domain until a large timestep has elapsed using interpolation to provide right boundary conditions from P.*
4. *Use an interface condition to generate a new value of P.*
5. *Return to step 2 and iterate until convergence.*

We can choose the initial guess for P in several ways, for example by extrapolating from the previous timestep or by first solving over the entire network using the large timestep. Both the interface condition and the type of interpolation at the interface may be derived by consideration of conservation of mass and momentum. It was noted in Chapter 4 that the numerical scheme (5.2.1) can be considered as a finite volume scheme where mass in node i is

$$\frac{A\Delta x P_i}{RT}$$

and momentum in a pipe $i + 1/2$ is

$$A\Delta x Q_{i+1/2}.$$

Conservation of mass at the interface then leads to the interface condition

$$\frac{\Delta x + \delta x}{2RT}(P_0^1 - P_0^0) + \Delta t Q_{1/2}^{\theta} - \sum_{k=0}^{m-1} \delta t q_{-1/2}^{k+\theta} = 0. \tag{5.2.2}$$

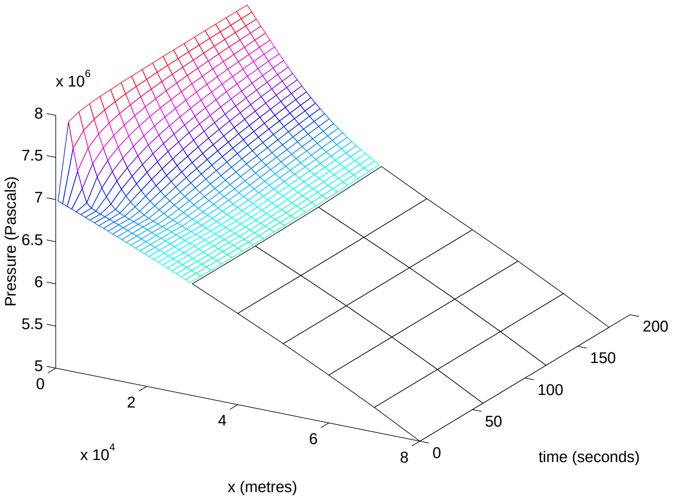


Figure 5.2.2: A solution calculated by Algorithm 3

In the specific case of $\theta = 0.5$ linear interpolation of P guarantees momentum conservation since

$$\begin{aligned}
\text{Momentum flux from left} &= \sum_{k=0}^{m-1} \delta t p_0^{k+1/2} \\
&= \sum_{k=0}^{m-1} \delta t (p_0^k + p_0^{k+1})/2 \\
&= \sum_{k=0}^{m-1} \delta t \left(\frac{k}{m}(P_0^1 - P_0^0) + \frac{k+1}{m}(P_0^1 - P_0^0) \right)/2 + \delta t P_0^0 \\
&= (P_0^1 + P_0^0)\Delta t/2 = \Delta t P_0^{1/2} \\
&= \text{Momentum flux to right.}
\end{aligned}$$

A typical result from this algorithm is shown in Figure 5.2.2.

5.2.1 Is conservation necessary?

The introduction to this chapter mentioned the efforts made by several authors to generate conservative interface conditions when using domain decomposition for hyperbolic conservation laws. A paper by Pärt-Enander and Sjögreen [PES94] suggests that conservative interpolation between meshes is less accurate than a nonconservative method, though their results confirm that conservation is vital where there are shocks. However, the aim of this work is to link hyperbolic domains with parabolic domains. If a shock crosses an interface and enters a parabolic domain then it will no longer be resolved correctly which means that the interface is incorrectly placed. Thus, strict conservation is not necessary for the situations we consider, though it is useful to suggest possible interface conditions. For example, linear interpolation may be used at the interface for *any* value of θ in (5.2.2) and provide good results even though momentum is not conserved. We also have the freedom to see if quadratic or higher order interpolation will improve accuracy.

As for the interface condition, many variations of equation (5.2.2) are possible, such as this discretisation of the continuity equation

$$\frac{1}{RT}\frac{P_0^1 - P_0^0}{\Delta t} + \frac{Q_{1/2}^{1/2} - \tilde{Q}_{-1/2}}{\tilde{\Delta x}} = 0 \tag{5.2.3}$$

where $\tilde{Q}_{-1/2}$ is some suitable measure of the flow in the left hand domain, for example an average of q_i and $\tilde{\Delta x}$ is an appropriate distance.

Different interface conditions are evaluated at the end of this chapter in Section 5.7

5.2.2 Convergence behaviour of Algorithm 3

Algorithm 3 is simple to implement but does not converge quickly or reliably, with performance deteriorating as the ratio of Δt to Δx grows. The rate of convergence may be improved by use of a relaxation parameter ω, modifying step 4 of Algorithm 3 to

4. *Use an interface condition to generate a new value of P: P^*. Use $(1-\omega)P_{old} + \omega P^*$ in the next iteration.*

Although ω can usually be chosen to ensure convergence, this method will at best converge linearly and the rate is still poor. Typically we might have $\Delta t = 600$ seconds, $\Delta x = 5000$ metres and choose $\omega = 0.5$. The relative error at the interface converges from 0.02 to 10^{-6} in about 14 iterations, which is too slow.

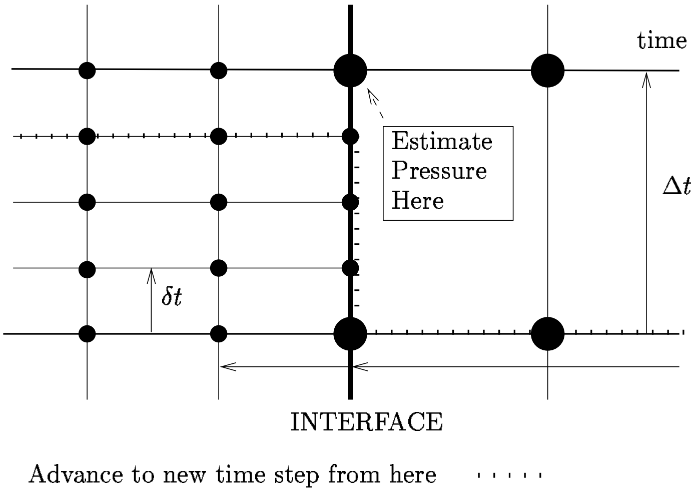


Figure 5.2.3: Advancing to the new timestep simultaneously from both domains

In an effort to improve reliability we try a slightly different strategy which advances to the new large timestep simultaneously in both domains. As in Algorithm 3 the pressure at the interface at the end of a large timestep is estimated and interpolated to provide boundary conditions for the left hand domain. The solution in the left hand domain is then calculated up to the penultimate small timestep. A generalised numerical method which takes account of the different timesteps is then used to advance the final timestep in left and right domains simultaneously as shown in Figure 5.2.3. This new strategy performs little better than Algorithm 3 in practice and would be quite difficult to implement when the different models are used. It is not pursued further here since a superior method is presented in Section 5.2.5.

5.2.3 A model for the behaviour of Algorithm 3

The analysis of Algorithm 3 is complicated and is even more so when the full parabolic and hyperbolic models are used. Some insight may be gained by considering a model problem, the same algorithm applied to the linear wave equation

$$\begin{aligned} u_t + v_x &= 0 \qquad (5.2.4) \\ v_t + u_x &= 0 \end{aligned}$$

where the exact solution is used in the algorithm in place of a numerical one. In a similar spirit to the use of modified equations to simulate numerical schemes we are using an analytical solution to explain the behaviour of a numerical algorithm.

The wave equations are solved for $t > 0, x \in (-\infty, \infty)$ with initial conditions

$$u(x,0) = 0$$
$$v(x,0) = 0,$$

which gives the trivial solution $u = v = 0$. Since the problem we are considering is linear, we may choose these initial conditions to simplify the calculation without loss of generality. Suppose now we place an interface at $x = 0$ on which we impose the boundary condition $u(0,t) = P(t)$. We examine different interface conditions by using them to deduce P and thus the exact solution.

For general $P(t)$ the solution is

$$u(x,t) = \begin{cases} 0 & x < -t \\ P(x+t) & -t \leq x < 0 \\ P(t-x) & 0 \leq x < t \\ 0 & x \geq t \end{cases} \tag{5.2.5}$$

$$v(x,t) = \begin{cases} 0 & x < -t \\ -P(x+t) & -t \leq x < 0 \\ P(t-x) & 0 \leq x < t \\ 0 & x \geq t \end{cases} \tag{5.2.6}$$

and so $|P(\xi)|$ could be considered as a measure of the error introduced by the interface.

We now apply the interface condition that we wish to analyse. For example, suppose we have a mesh of uniform spacing Δx and Δt with u_i^n defined as usual. Choose the function $P(t)$ to be piecewise linear and consistent with the initial conditions

$$P(t) = \frac{t}{\Delta t} P^* \qquad 0 \leq t \leq \Delta t$$

where P^* is to be determined by discretising the first of the wave equations (5.2.4),

$$\frac{P^*|_{n+1}-0}{\Delta t}+\frac{v_1^1|_n-v_{-1}^1|_n}{4\Delta x}+\frac{v_1^0-v_{-1}^0}{4\Delta x}=0. \qquad (5.2.7)$$

The bars beside some terms such as $P^*|_{n+1}$ denote iterate numbers. Recall that the algorithm is to guess $P^*|_0$ and calculate the solution by substituting $P(t)|_0$ into (5.2.5) and (5.2.6). We then use interface condition (5.2.7) to generate $P^*|_1$ and iterate. The algorithm can be written compactly as

$$P^*|_{n+1}=F(P^*|_n)$$

and to investigate whether it converges we need to determine if $|dF(P)/dP|<1$.

Differentiating the interface condition (5.2.7) gives

$$\frac{1}{\Delta t}\frac{d(P^*|_{n+1})}{d(P^*|_n)}+\frac{1}{4\Delta x}\left(\frac{d(v_1^1|_n)}{d(P^*|_n)}-\frac{d(v_{-1}^1|_n)}{d(P^*|_n)}\right)=0.$$

To calculate the derivatives of $v_{\pm1}^1|_n$ we need to know whether they lie in the domain of dependence of the interface and this depends on the mesh ratio $\lambda=\Delta t/\Delta x$. If $\lambda\le1$ then they do not, the derivatives are zero, and the iteration converges immediately. If on the other hand $\lambda>1$ we have, using the exact solution (5.2.6)

$$\begin{aligned} v_{-1}^1|_n=&-P(-\Delta x+\Delta t)|_n\\ v_1^1|_n=&P(-\Delta x+\Delta t)|_n \end{aligned}$$

and on differentiating

$$\begin{aligned} \frac{d(v_{-1}^1)|_n}{d(P^*|_n)}=&-\frac{d(P(-\Delta x+\Delta t)|_n)}{d(P^*|_n)}\\ \frac{d(v_1^1|_n)}{d(P^*|_n)}=&\frac{d(P(-\Delta x+\Delta t)|_n)}{d(P^*|_n)}. \end{aligned}$$

We now use the linearity of $P(t)$ to deduce that

$$\frac{d(P(t)|_n)}{d(P^*|_n)}=\frac{t}{\Delta t}.$$

Finally, combining all of these results we find that

$$\frac{dF}{dp}=\frac{1}{2}(1-\lambda)$$

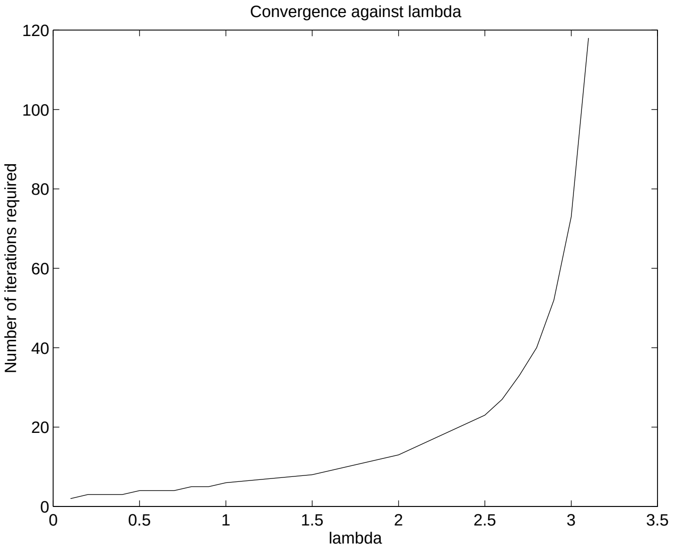


Figure 5.2.4: The number of iterations required for convergence against lambda

and the iteration will not converge for $\lambda \geq 3$. If a relaxation parameter ω is introduced as in Section 5.2 then the equivalent criterion is

$$|(1-\omega)+\omega\frac{1}{2}(1-\lambda)| < 1.$$

If λ is large then it can be seen that a small positive ω will ensure convergence, agreeing with the results seen in practice.

If we solve the wave equations (5.2.4) with a numerical scheme and apply Algorithm 3 we find very good agreement with this theory, even though we are no longer using an exact solution. Figure 5.2.4 shows the number of iterations required to satisfactorily converge as the mesh ratio λ is increased. As predicted, the algorithm diverges as λ passes 3. Note that for $\lambda < 1$ the expected immediate convergence does not occur in practice since the implicit numerical scheme has an infinite domain of dependence. Consequently a change at the interface has a small influence on the interface condition even though its stencil takes values from outside the analytical domain of influence of the interface.

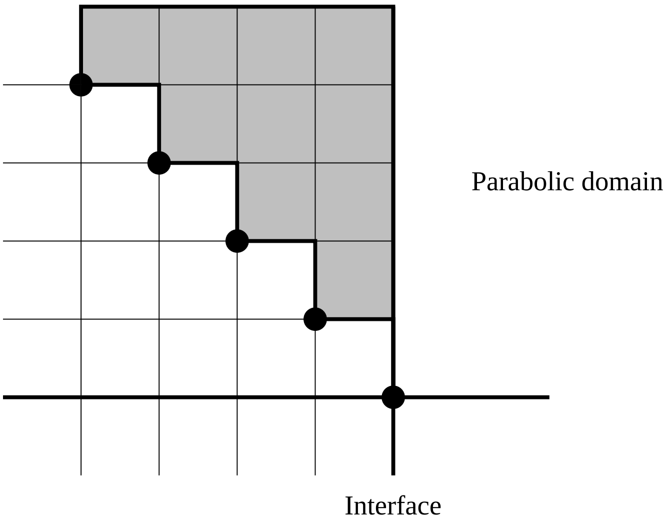


Figure 5.2.5: The domain of dependence of the interface for a numerical method with a three point stencil.

5.2.4 Interfacing the hyperbolic and parabolic models

Algorithm 3 requires little modification to interface a parabolic and a hyperbolic domain. We now refer to the domain with the finer timestep as the hyperbolic domain, modelled by the Euler equations (2.0.1) - (2.0.1) and solved by a method such as Roe's scheme, described in Chapter 3. The domain with the coarser timestep is the parabolic domain, modelled by the parabolic isothermal equations (2.1.3) - (2.1.4) and solved by the staggered mesh scheme of Chapter 4.

If an explicit numerical method such as Roe's scheme is used for the hyperbolic domain, it is only necessary to compute over the entire domain on the first iteration. On subsequent iterations only the numerical domains of dependence of the interfaces, shown in Figure 5.2.5, need to be recalculated allowing large efficiency savings.

The new feature of the hyperbolic model is the extra variable, temperature. Since temperature may now vary, interface conditions such as (5.2.3) must be reformulated, for example as

$$\frac{\varrho_0^1 - \varrho_0^0}{\Delta t} + \frac{Q_{1/2}^{1/2} - \tilde{Q}_{-1/2}}{\tilde{\Delta x}} = 0 \tag{5.2.8}$$

or

$$\frac{P_0^1/T_0^1 - P_0^0/T_0^0}{R\Delta t} + \frac{Q_{1/2}^{1/2} - \tilde{Q}_{1/2}}{\tilde{\Delta x}} = 0. \tag{5.2.9}$$

allowing the choice of stipulating ϱ or P on the boundary. When the gas is flowing from the parabolic domain to the hyperbolic domain the hyperbolic domain requires an extra boundary condition which should come from the energy equation (2.0.3). This equation could be discretised in a similar manner to the mass equation to produce an extra interface condition. However, as was remarked in Section 5.2.1 strict conservation of energy is not required. It is simpler to set the temperature of the incoming gas in the hyperbolic domain to that of the gas coming out of the parabolic domain.

When the gas flow is in the opposite direction, from the hyperbolic to the parabolic domain, we cannot impose an extra interface condition. Since the parabolic domain is isothermal we must tolerate a discontinuity in the temperature at the interface as a consequence of the use of inconsistent models. One way round this is to use the temperature-dependent parabolic model (4.0.3) and this model is used throughout the remainder of this thesis.

In summary, the only change required of Algorithm 3 is that a step is included transferring the temperature from one domain to the other, interpolating in time along the interface as necessary. The convergence rate of the iteration appears to be better if ϱ is specified at the interface rather than P.

In Section 5.2.3 a model was presented which predicted the convergence behaviour of Algorithm 3 in an idealised case of solving the wave equation. The isothermal Euler equations also transmit waves, with speed $a = \sqrt{RT}$. Even though a friction term is now present and the parabolic model is quite different from the wave equation, the linear model provides a reasonable estimate of the properties of Algorithm 3 as applied to the parabolic-hyperbolic problem. Figure 5.2.6 shows the rate of convergence against $\lambda = a\Delta t/\Delta x$ for a large range of mesh ratios (note the log scale). Once again, the Algorithm begins to diverge at around $\lambda = 3$.

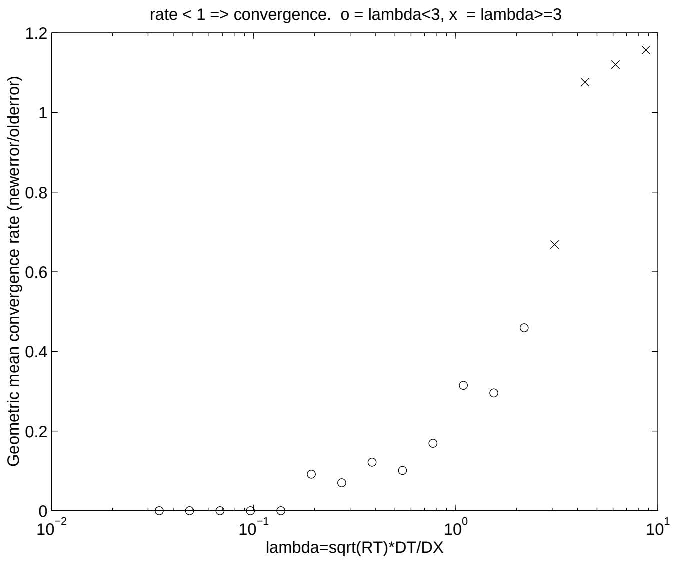


Figure 5.2.6: Geometric mean convergence rate over 6 iterations for different mesh ratios. Circles for $\lambda < 3$, crosses for $\lambda > 3$.

When several interfaces are present Algorithm 3 may be applied unchanged, with each interface having its own interface condition. If two interfaces are separated by a hyperbolic domain then the hyperbolic domain will be larger than the domains of influence of the interfaces and they will not interfere with each other. Due to the infinite wave speed in the parabolic domains, two interfaces separated by a parabolic domain will influence each other but the dominating friction term will keep this effect small. If the interfaces are placed correctly therefore, we may consider each interface independently.

5.2.5 A faster converging algorithm

Algorithm 3 is an effective and simple way to link hyperbolic and parabolic domains but converges unacceptably slowly. Even with a good choice of relaxation parameter ω convergence will always be at a linear rate. The algorithm may be improved by the use of the secant method. If we treat the entire process as a "black box" function then the algorithm may be expressed, as in Section 5.2.3, as

$$P|_{n+1} = F(P|_n) \tag{5.2.10}$$

where F is equivalent to the entire process of solving the hyperbolic and parabolic domains and generating a new estimate of P from equation (5.2.3) (for example). We then define the function $G(P) = F(P) - P$ and solve $G(P) = 0$ by the secant method*:

$$P|_n = P|_{n-1} - \frac{P|_{n-1} - P|_{n-2}}{G(P|_{n-1}) - G(P|_{n-2})} G(P|_{n-1}).$$

Table 5.2.1 shows the dramatic improvement in the number of iterations that this makes.

Unfortunately it is not possible to investigate the convergence of the secant method with the same model as for Algorithm 3 since a linear theory predicts immediate convergence. However, practical experience has shown that provided the first and second guesses are chosen reasonably well this method will converge. The second guess may be made by using Algorithm 3 with a suitable choice of relaxation parameter ω.

5.3 Extending the method to several interfaces

The iterative Algorithm treats a problem with several interfaces is a similar fashion. We generate the new interface pressures with the vector version of equation (5.2.10)

$$\mathbf{P}|_{n+1} = \mathbf{F}(\mathbf{P}|_n). \tag{5.3.1}$$

For Δt larger than a certain size equation (5.3.1) suffers from the same convergence problems as equation (5.2.10) and is cured in a similar way. We define the vector function $\mathbf{G}(\mathbf{P}) = \mathbf{F}(\mathbf{P}) - \mathbf{P}$ and use an $(n+1)$-point sequential secant method (see

*It is not practical to use Newton's method since in all nontrivial cases F cannot be written down explicitly.

	Algorithm 3		Secant method	
Iteration	error	ln(error)	error	ln(error)
1	9.3×10^{-2}	-2.38	9.3×10^{-2}	-2.38
2	6.8×10^{-2}	-2.69	6.8×10^{-2}	-2.69
3	2.8×10^{-2}	-3.58	3.5×10^{-3}	-5.66
4	9.6×10^{-3}	-4.64	1.0×10^{-4}	-9.21
5	3.7×10^{-3}	-5.60	1.7×10^{-7}	-15.59
6	1.4×10^{-3}	-6.57	8.5×10^{-12}	-25.49
7	5.3×10^{-4}	-7.54	4.0×10^{-16}	-35.46

Table 5.2.1: A comparison of the fixed point and secant methods for Algorithm 3.

[OR70])

$$\mathbf{P}|_{\mathbf{n}} = \mathbf{P}|_{\mathbf{n-1}} - J^{-1}|_{n-1}\mathbf{G}(\mathbf{P}|_{n-1}) \tag{5.3.2}$$

where $J|_{n-1}$ is a numerical approximation to the Jacobian of $\mathbf{G}$ constructed from the last k iterates, where k is the number of interfaces and the length of the vectors. It is calculated from

$$J|_{n-1}\begin{bmatrix} \vdots & \vdots & \cdots \\ \mathbf{y}|_{n-1} & \mathbf{y}|_{n-2} & \cdots \\ \vdots & \vdots & \cdots \end{bmatrix} = \begin{bmatrix} \vdots & \vdots & \cdots \\ \mathbf{H}|_{n-1} & \mathbf{H}|_{n-2} & \cdots \\ \vdots & \vdots & \cdots \end{bmatrix}$$

where

$$\mathbf{y}|_k \stackrel{\text{def}}{=} \mathbf{P}|_k - \mathbf{P}|_{k-1} \tag{5.3.3}$$

$$\mathbf{H}|_k \stackrel{\text{def}}{=} \mathbf{G}(\mathbf{P}|_k) - \mathbf{G}(\mathbf{P}|_{k-1}) \tag{5.3.4}$$

and requires the first k values of $\mathbf{P}$ to be given. These can be calculated from a relaxed version of equation (5.3.1) since all we require are k values of $\mathbf{P}$ and the corresponding $\mathbf{F}(\mathbf{P})$, it is immaterial whether they converge, though (5.3.2) will converge faster if they are close to the solution. We assume that the successive $\mathbf{y}|_k$

are linearly independent - a code for practical use would need to verify this and take remedial action if this is not the case.

Due to the minimal interaction between the interfaces the off-diagonal elements of the Jacobian will be relatively small. We can exploit this by using the secant method independently on each interface for the 3rd to kth iterations.

When we include the temperature interface condition in the process the rate of convergence improves still further.

5.3.1 A Schwartz-type method

This method is based on allowing an overlap between the meshes of the hyperbolic and parabolic domains as shown in Figure 5.3.1.

Assume as before that estimates of the pressure (or density) and temperature are available on the interface. These are used as boundary conditions for the hyperbolic domain. Once the hyperbolic solution has been calculated at the midpoint of the adjacent pipe, it is used as a boundary condition for the parabolic domain, which updates the interface estimates, and so on. In fact this is the same as the nonoverlapping method (Algorithm 3), but with the discretisation of the parabolic mass equation,

$$\frac{\varrho_0^1 - \varrho_0^0}{\Delta t} + \frac{Q_{1/2}^{n+\theta} - \mathcal{P}q}{\tilde{\Delta x}} = 0.$$

as the interface condition.

We use the notation $Q = \mathcal{P}q$ to represent the transfer of information from the hyperbolic domain to the boundary of the parabolic domain. This will be discussed in Section 5.5.

In practice this algorithm converges less quickly than the previous method.

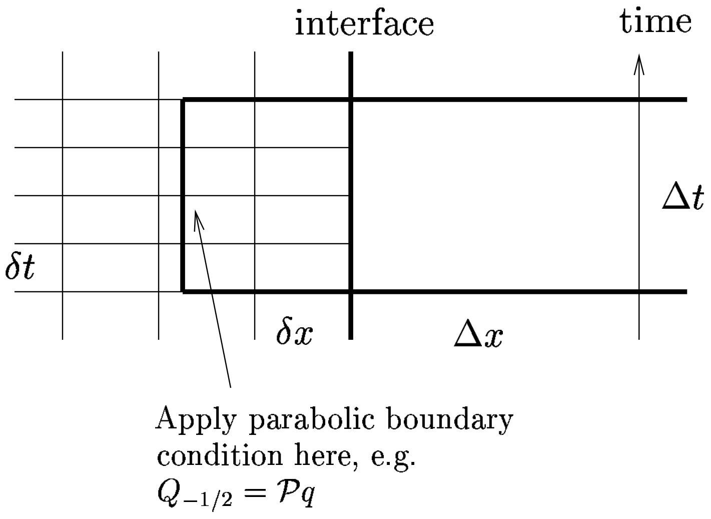

interface
time
Δt
δt
δx
Δx
Apply parabolic boundary condition here, e.g.
$Q_{-1/2} = \mathcal{P}q$

Figure 5.3.1: Overlapping meshes

5.4 Noniterative methods

The iterative method of Section 5.2 has the flexibility to allow a large choice of conditions at the interface between the parabolic and hyperbolic domains. However, iteration is computationally expensive and we must investigate whether it is justified.

This section outlines 2 noniterative methods. We could also include using equation (5.2.2) with $\theta = 0$ as a noniterative method since it gives the interface condition explicitly, although it imposes a severe restriction on the size of Δt to ensure stability. Comparisons between the accuracy and expense of the following methods and the iterative method of Section 5.2 are made in Section 5.7.

5.4.1 Method 1: Superimposed grids

The basis of this method is to take advantage of the fact that the parabolic model should produce a reasonably good solution in the domain of the interface (otherwise the interface is too close to the transient.) The first step is to use the parabolic method on the entire network, including hyperbolic domains. Although the parabolic model will not accurately resolve the fast transients it will give an estimate of the effect that the fast transients have on the rest of the network. We then superimpose the hyperbolic mesh, using the parabolic solution to provide the boundary conditions at the interfaces and interpolating as usual.

To calculate the parabolic solution on the entire network we need to map the hyperbolic solution onto the parabolic mesh at each parabolic step as described in Section 5.5. Consequently there will be discontinuities in the data for the parabolic solver at interfaces. However, the damping inherent in the parabolic model seems to be enough to smooth out the discontinuities at interfaces. A disadvantage is that transients will affect the entire network instantaneously due to the infinite transmission speed of the parabolic model. Nevertheless, this method is particularly attractive because of its simplicity.

In summary the algorithm is,

Algorithm 4

Assuming we have arrived at the beginning of a parabolic step with hyperbolic and parabolic domains,

- *Map the hyperbolic solution to the parabolic domain.*
- *Calculate one parabolic step on the entire mesh.*
- *Calculate hyperbolic steps, using the parabolic solution to provide boundary conditions, until one parabolic timestep has elapsed.*

5.4.2 Method 2: Domain of dependence

This method exploits the finite domain of dependence of the hyperbolic equations to explicitly generate the variables at the interfaces at the end of a parabolic timestep. The idea is to calculate the hyperbolic solution in the parts of the parabolic domain belonging to the finite domains of dependence of the interfaces as illustrated in Figure 5.4.1.

At the initial time-level we transfer the data from the parabolic mesh to the now enlarged hyperbolic mesh by one of the methods that will be discussed in Section 5.5.

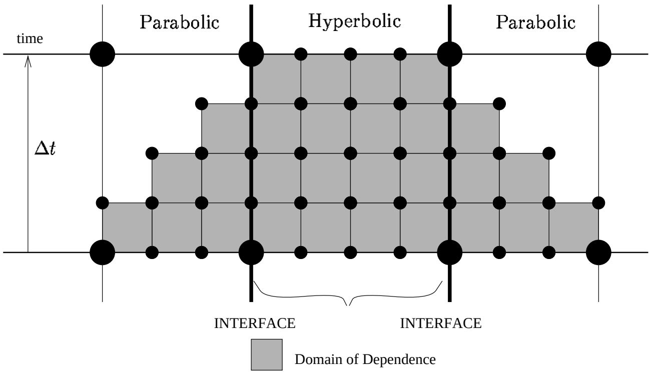


Figure 5.4.1: The domain of dependence of a hyperbolic domain

The states at the interface are then used as boundary conditions for the parabolic domain.

A drawback of this method is the rate at which the domains of dependence grow with the timestep. Furthermore, the task of calculating the domains of dependence of the hyperbolic regions is complicated, and would be even more so with a branching network. We could have domains of dependence overlapping each other, or themselves in a looped network, which would be inefficient but difficult to avoid. This method was therefore not pursued further.

5.5 Transferring data between the meshes

All of the interface methods require the transfer of data between the hyperbolic and parabolic domains. The transfer is required for two reasons, firstly when a pipe switches from one mode to another its data must be transferred to a coarser or finer mesh. Secondly, the linking condition at an interface often requires information from one type of domain on the mesh of the other.

If we denote the projection operator from the hyperbolic mesh to the parabolic mesh by $\mathcal{P}$ and that from the parabolic mesh to the hyperbolic by $\mathcal{H}$ we should require that $\mathcal{P} \circ \mathcal{H} = \mathcal{I}$, the identity. This ensures that if a pipe is switched repeatedly from hyperbolic to parabolic and back again the operation of transferring data will

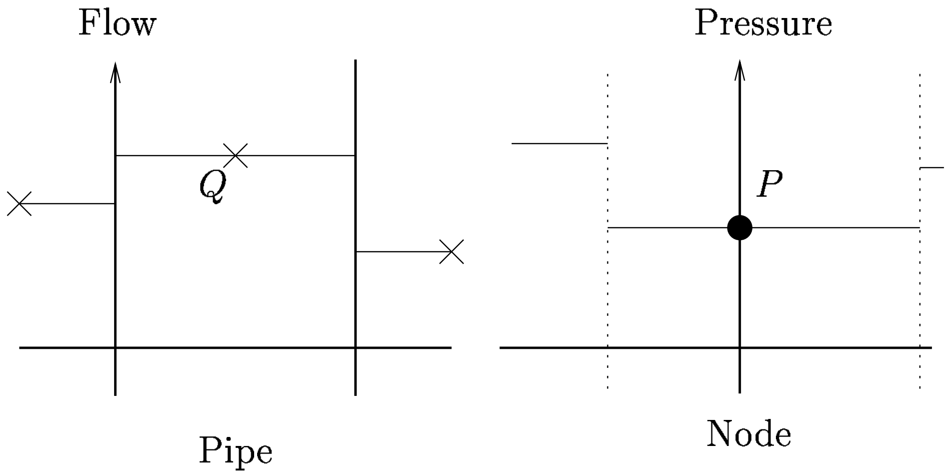


Figure 5.5.1: Transfer of flow and pressure data between hyperbolic and parabolic meshes

not corrupt the solution. The hyperbolic mesh is generally finer than the parabolic mesh and so some information is inevitably discarded by $\mathcal{P}$ and we cannot require that $\mathcal{H} \circ \mathcal{P} = \mathcal{I}$.

5.5.1 Some illustrative examples

Due to the staggered parabolic mesh there are some small technical differences between the transfer of pressure/temperature data and flow data. Consider first the transfer of the flow data from parabolic to hyperbolic mesh as depicted in Figure 5.5.1. Perhaps the simplest strategy is to use piecewise constant interpolation and set $q_i = Q$, allowing discontinuities in the hyperbolic data at nodes. For the pressure and temperature data the method is similar, except that the discontinuities occur at the pipe mid-points as shown in Figure 5.5.1.

There are many consistent possibilities for the reverse process of mapping from the hyperbolic mesh to the parabolic mesh. The two most obvious are pointwise

$$Q = q_i \quad x_i \text{ is the point nearest the pipe centre} \tag{5.5.1}$$

where the hyperbolic point closest to the midpoint of the pipe is used, or averaged

$$Q = \bar{q_i} \tag{5.5.2}$$

which is more computationally expensive but less affected by discontinuities in the hyperbolic solution. Figure 5.5.2 shows an initial wave transferred to the coarse parabolic mesh using equation (5.5.2) and back to the hyperbolic mesh by piecewise constant interpolation.

Although simple to implement, piecewise constant interpolation is not accurate enough. This causes difficulties when a pipe becomes hyperbolic; the discontinuities caused by the interpolation propagate down the pipe causing the adjacent pipe to become hyperbolic, and so on. However, using linear interpolation instead is not completely trivial. The simplest approach is to simply linearly interpolate between nodes and pipes as shown in Figure 5.5.3:

$$p_i = \mathcal{H}_p\mathbf{P} = \frac{x_i - X_j}{\Delta x}P_{j+1} + \frac{X_{j+1} - x_i}{\Delta x}P_j \qquad X_j < x_i < X_{j+1} \tag{5.5.3}$$

$$q_i = \mathcal{H}_q\mathbf{Q} = \frac{x_i - X_{j-1/2}}{\Delta x}Q_{j+1/2} + \frac{X_{j+1/2} - x_i}{\Delta x}Q_{j-1/2} \qquad X_{j-1/2} < x_i < X_{j+1/2} \tag{5.5.4}$$

where x_i is position of hyperbolic cell i, X_j the position of node j and $X_{j+1/2}$ the pipe midpoint.

This is consistent with choosing $\mathcal{P}$ to be (5.5.1) but not the averaging (5.5.2). This latter projection does not satisfy $\mathcal{P} \circ \mathcal{H} = \mathcal{I}$ and tends to smooth peaks and troughs if used repeatedly, as demonstrated in Figure 5.5.4.

The final interpolation discussed here is piecewise linear and consistent with both equations (5.5.1) and (5.5.2). We follow the idea introduced by van Leer [vL79] to provide a piecewise linear *conservative* interpolation. Within a "cell" the interpolation takes the form

$$q_i = \mathcal{H}_q\mathbf{Q} = Q_{j+1/2} + (x_i - X_{j+1/2})\sigma_{j+1/2} \qquad X_j < x_i < X_{j+1} \tag{5.5.5}$$

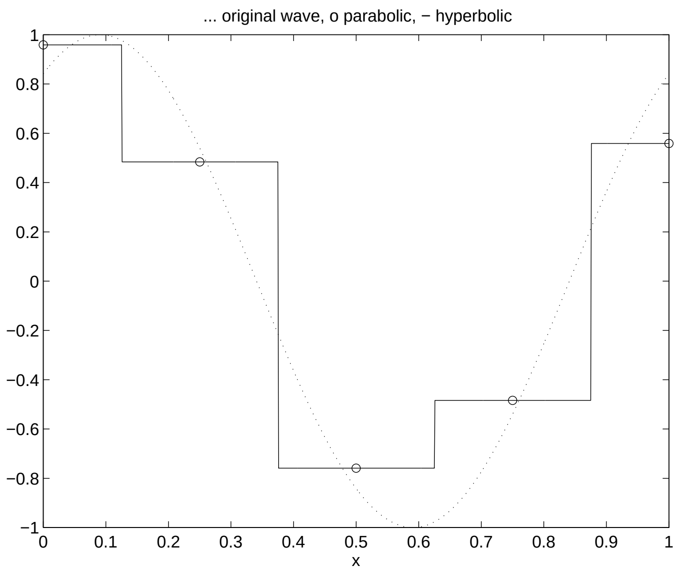


Figure 5.5.2: Averaging to the parabolic mesh followed by piecewise constant interpolation back to the hyperbolic mesh.

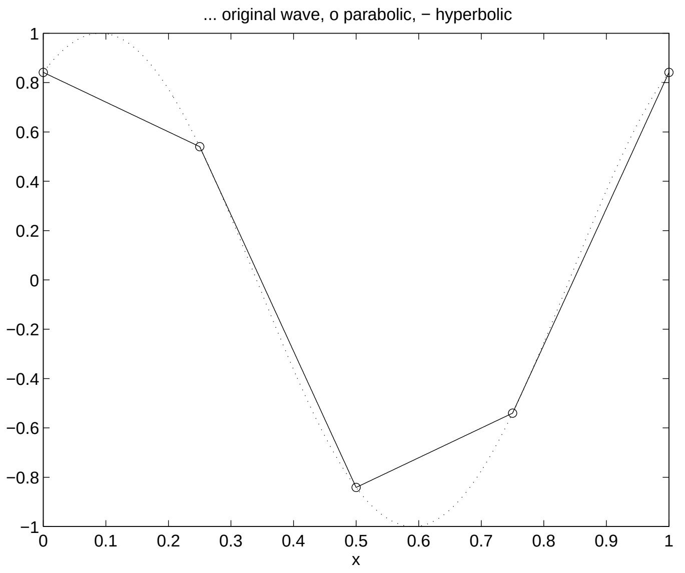


Figure 5.5.3: Linear interpolation between parabolic data points

where the slopes $\sigma_{j+1/2}$ are provided, for example, by the minmod limiter

$$\sigma_{j+1/2} = \begin{cases} \sigma_j & |\sigma_j| < |\sigma_{j+1}|, \sigma_j\sigma_{j+1} > 0 \\ \sigma_{j+1} & |\sigma_{j+1}| < |\sigma_j|, \sigma_j\sigma_{j+1} > 0 \\ 0 & \text{otherwise} \end{cases}$$

where

$$\begin{aligned} \sigma_j &= \frac{Q_{j+1/2} - Q_{j-1/2}}{\Delta x} \\ \sigma_{j+1} &= \frac{Q_{j+3/2} - Q_{j+1/2}}{\Delta x}. \end{aligned}$$

An example using this limiter is shown in Figure 5.5.5. Note that continuity has been sacrificed for conservation.

Some options for $\mathcal{P}$ and $\mathcal{H}$ are summarised in Table 5.5.1 and compared for accuracy in Figure 5.5.6. The additional complications of data transfer on a branching network are discussed in Chapter 6.

	$\mathcal{P}$	$\mathcal{H}$
1	Pointwise	Constant
2	Averaged	Constant
3	Pointwise	Linear, continuous
4	Pointwise	Linear, discontinuous
5	Averaged	Linear, discontinuous

Table 5.5.1: Methods for data transfer

5.6 Tracking transients

The hyperbolic domains are typically needed where there are travelling waves or shocks and so they have to move or expand to track these transients. Furthermore, the frictional terms in the hyperbolic model will damp the transients so that after

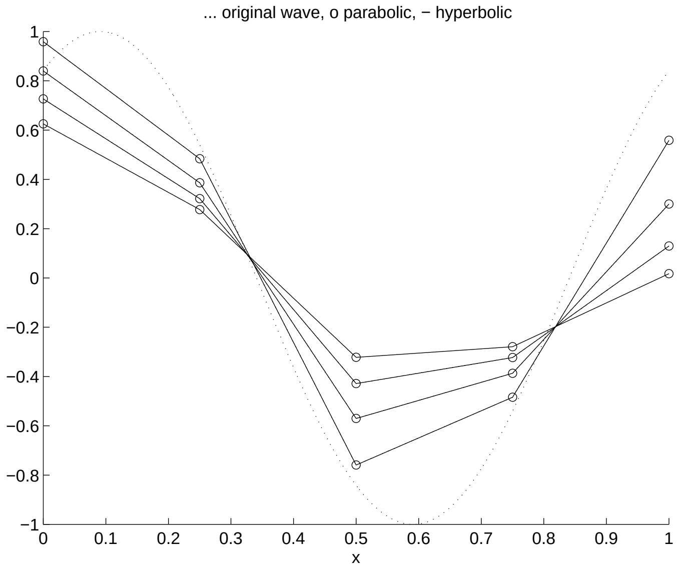


Figure 5.5.4: The damping caused by repeated use of inconsistent transfer operators

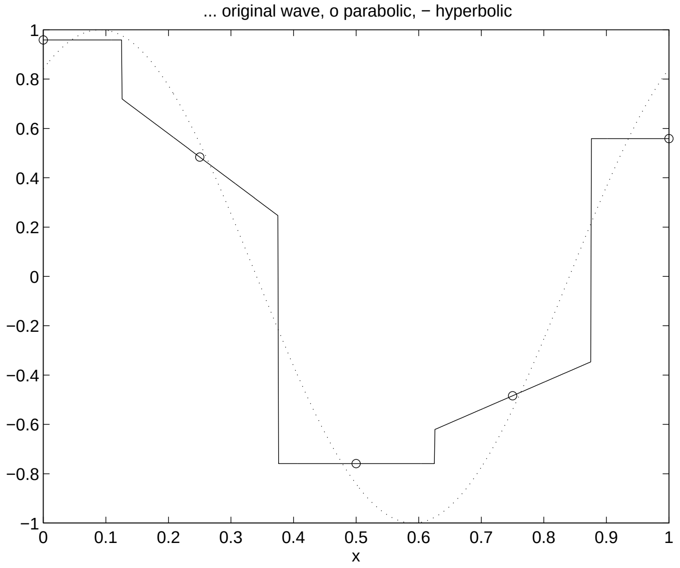


Figure 5.5.5: Piecewise linear interpolation around parabolic data points

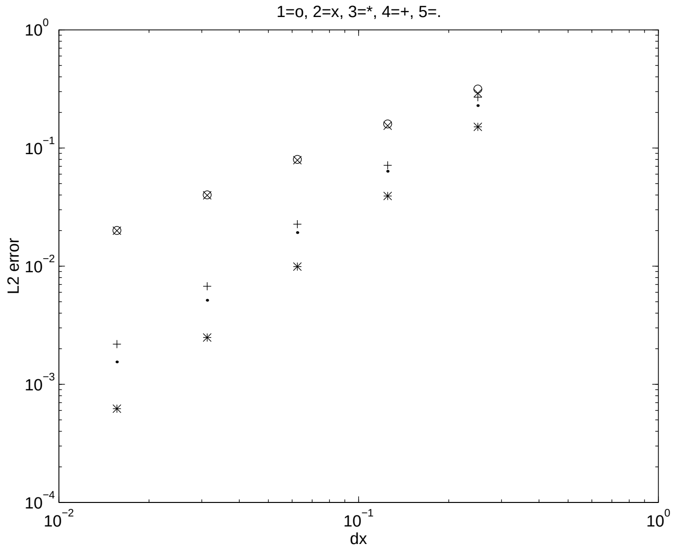


Figure 5.5.6: A comparison of the different forms of interpolation - see Table 5.5.1 for the key.

some time the hyperbolic domains may decrease in size or disappear altogether. It is possible that complicated geometries will arise as parabolic "holes" appear in hyperbolic domains or hyperbolic domains merge. Rather than track the interfaces explicitly it is simpler to examine each pipe at the end of a parabolic timestep and decide whether it should switch status. The interfaces are then recalculated from this data, a relatively easy process since the domain is one-dimensional. This will automatically account for the creation or destruction of interfaces.

To carry out this process we need tests to determine when to switch a pipe's status. We consider first the switching of a pipe from parabolic to hyperbolic. The parabolic domains do not contain enough information to determine whether they should switch to hyperbolic. Instead, this must be deduced from adjacent hyperbolic domains, which can be done in two ways.

We could try to detect transients as they strike an interface from a hyperbolic domain. If one is detected then the parabolic step recommences from the beginning with an enlarged hyperbolic domain (recall that a pipe is hyperbolic or parabolic for an entire parabolic step and cannot change halfway through.) Suppose that at the interface we are enforcing a linear density boundary condition on the hyperbolic domain. If a wave strikes the interface then a characteristic signature is made by the flow as shown in Figure 5.6.2. This was produced by the left-moving pressure wave in Figure 5.6.1 striking the interface at about $t = 5$ seconds.

A drawback of this approach is that it is difficult to identify such events. We might attempt to look for the rate of change of q at the interface exceeding a certain tolerance but this ad hoc indicator is difficult to relate to the difference between hyperbolic and parabolic solutions.

A more satisfactory method which is easier to implement is to include a "safety net" around a transient into which it can travel during one parabolic timestep. The size of this safety net is determined by the maximum speed of a transient. The method reduces to the following algorithm which is depicted in Figure 5.6.3:

Algorithm 5

1. *Extend each hyperbolic domain by a distance of* Δt *times the maximum wave or shock speed*
2. *Solve the hyperbolic and parabolic domains over one parabolic step*
3. *Examine each pipe to see if it can be returned to parabolic status*

In practice we overestimate the transient speed to ensure that we capture fast moving shocks.

This algorithm has the advantage of being dependent on the method for switching from hyperbolic to parabolic and thus not requiring its own tolerance to be given. A disadvantage is that the hyperbolic domains are extended whether or not it is necessary, requiring extra computation. However, this will not be too great if the parabolic timestep is quite small. For example, given pipes of length 10,000 metres and a timestep of 20 seconds, the safety net is only one pipe in length. For larger

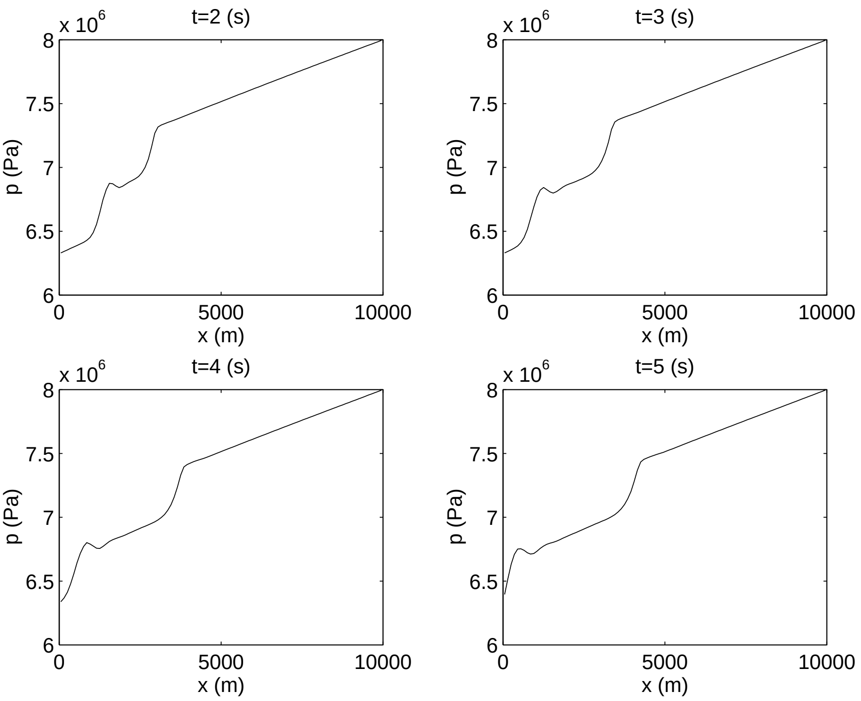


Figure 5.6.1: A pressure wave striking a left hand interface

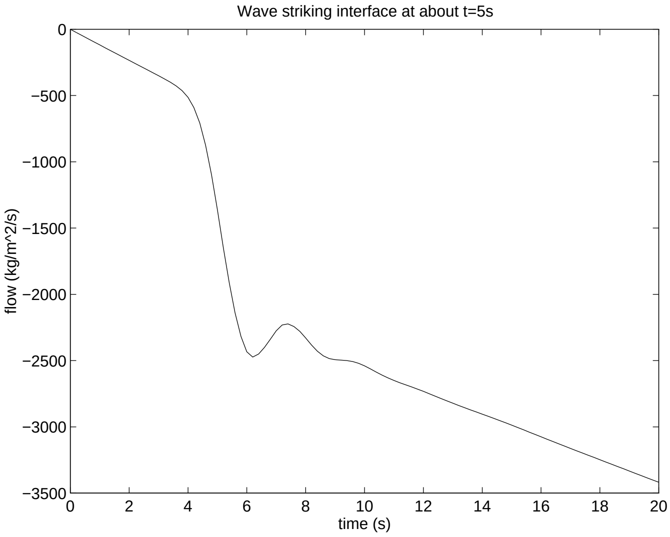


Figure 5.6.2: The signature made by a wave striking an interface

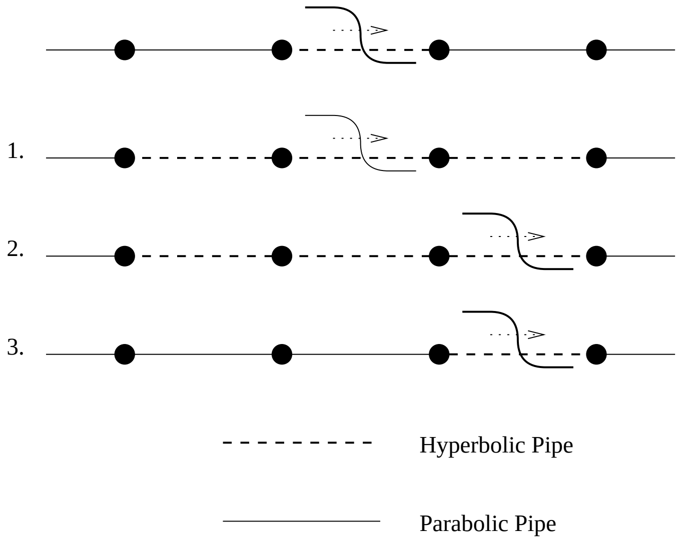


Figure 5.6.3: Algorithm 5

timesteps use can made of the frictional dissipation which will put a ceiling on the distance that a disturbance can travel.

There are many more options when we wish to determine when to switch a pipe from hyperbolic to parabolic status. Unlike switching from parabolic to hyperbolic all of the information we need is contained within the hyperbolic domain. Many tests to detect a transient in a pipe could be conceived such as examining the variance of the flow, testing the derivatives of the data or using the asymptotic expansions of Chapter 4. The best test is to compare the solutions calculated with the pipe hyperbolic and with it parabolic. Clearly it is impractical to solve a network with every possible size of hyperbolic domain. A simpler test is to solve each hyperbolic pipe with the parabolic solver individually and compare with the hyperbolic solution. This reduces to solving the parabolic momentum equation

$$\frac{1}{2}\frac{P_R^2 - P_L^2}{\Delta x} + \mu RTQ|Q| = 0$$

for Q on each pipe, using the pressures at the nodes as boundary conditions. The error introduced by not switching the pipe to hyperbolic may then be estimated by

$$|Q - \mathcal{P}_q q_i|$$

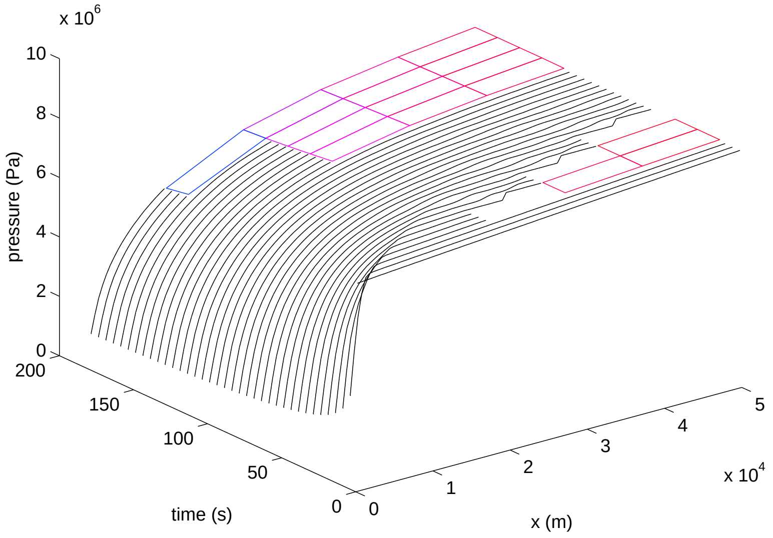


Figure 5.6.4: The progress of a wave through a linear network. Note the initial discontinuities in the (left hand) hyperbolic region caused by the mapping of the data from parabolic to hyperbolic meshes.

where $\mathcal{P}_q$ is the hyperbolic to parabolic projection operator of Section 5.5. If this quantity falls below a prescribed tolerance the pipe is switched to parabolic status. Figure 5.6.4 shows the expansion of a hyperbolic domain using the safety-net method and its subsequent contraction after applying the above test.

5.7 Experimental comparisons of the methods

In this section we evaluate the interfaced methods of this chapter by applying them to two representative test problems.

The test problems are on a “linear” network which for practical reasons is much smaller than the NTS. Since at this stage we are interested in the relative merits of the different methods and not the effectiveness of interfaced schemes as a whole the position of the interface was kept fixed. In each case the network was initially in a steady state and some disturbance at the left hand boundary caused a transient to move into the network at $t = 0$. Two runs were made of each test, one with the interface in a position where it was struck by the transient during the simulation time, the other where it was out of reach. This latter situation will occur in practice but it is interesting to examine the methods under more demanding conditions. The first test was a simulation of a pipe-break. Initially the gas was flowing steadily from right to left through the network. A pipe-break occurred and a rarefaction wave travelled into the network, eventually appearing to reach a steady state as friction forces began to dominate. The second test was a little more artificial in nature but was designed to have some of the characteristics of a compressor starting or of a valve closing. Again, the initial steady flow was from right to left but this time the pressure was raised suddenly at the left hand boundary causing a shock to travel into the network. The rise in pressure was of sufficient magnitude to reverse the direction of the flow and induce a stagnation point. The precise parameters of each test are given in Tables 5.7.1 and 5.7.2.

Figures 5.7.1 and 5.7.2 show sample results from the iterative method of Section 5.2. Each figure shows a 3D plot of the pressure in the parabolic and hyperbolic regions against time and space and a 2D plot of the final pressure in the hyperbolic region compared with the exact solution. Solutions with the interface at 10,000m and 20,000m are shown. It is evident that if a shock is allowed to strike an interface as in the first graphs from Figure 5.7.2 then the interface method performs very badly, while if the interface is struck by a rarefaction wave such as in Figure 5.7.1 then the solution is not as affected. The importance of keeping shocks inside the hyperbolic domains is clear.

5.7.1 Measuring accuracy

A question which was raised in Section 5.1 is: given an interfaced method which links two different models what is the correct "exact" solution to compare the numerical solution against? The best answer to this is probably to compare the numerical

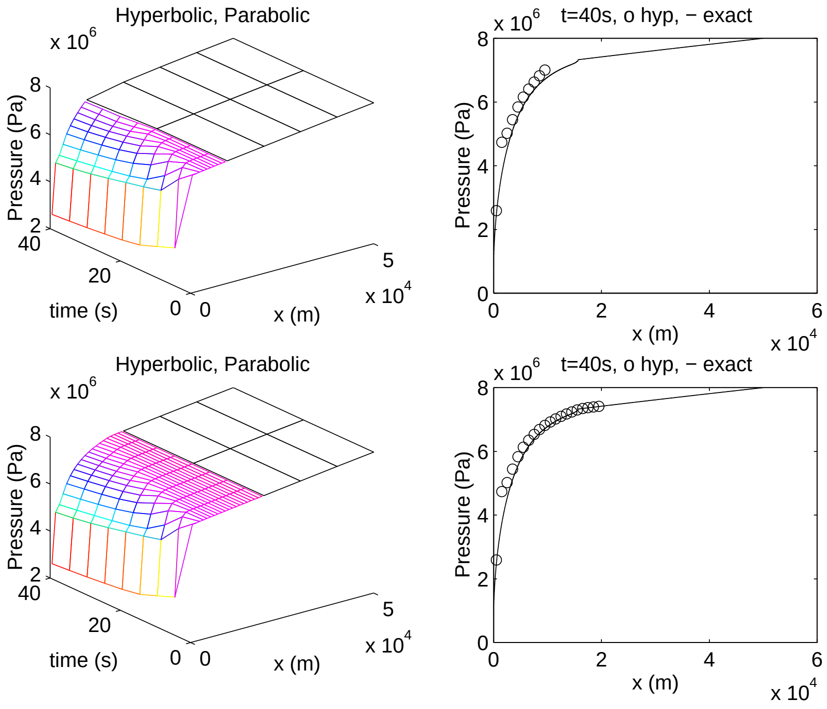


Figure 5.7.1: Results from the iterative method, test 1. Only the hyperbolic solution is shown in the right hand figures.

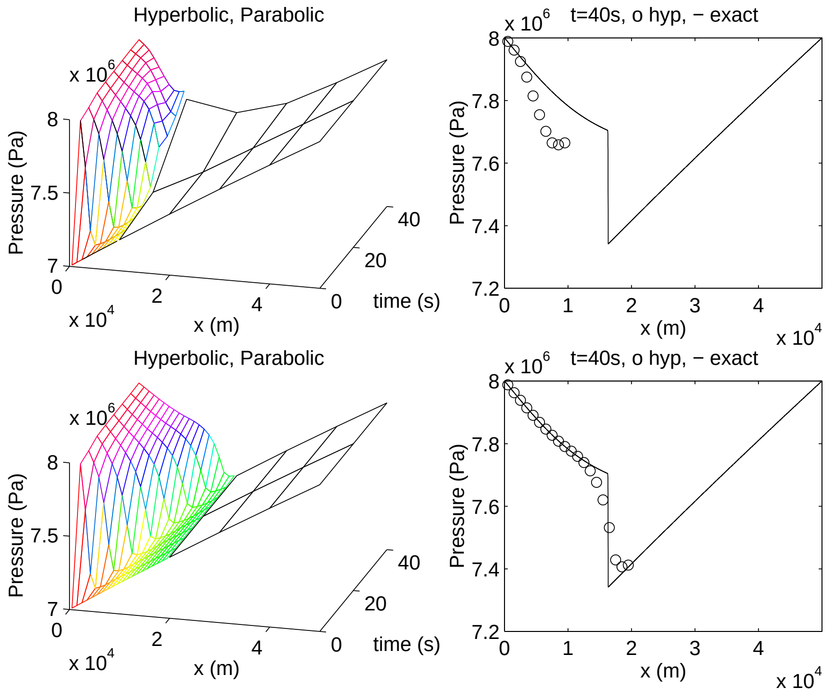


Figure 5.7.2: Results from the iterative method, test 2. Only the hyperbolic solution is shown in the right hand figures.

domain size		50,000 m
pipe diameter	d	.914 m
simulation time		40 s
initial pressure	p	$\sqrt{(1-\theta)70 \times 10^5 + \theta 80 \times 10^5}$ Pa
		where $\theta = x/50000$
initial flow	q	-543.52 kg/m^2/s
initial temperature	Θ	280 K
left boundary condition	mach #	-1
right boundary condition	pressure	80×10^5Pa
	temperature	280K
friction factor	f	0.002
ratio of specific heats	γ	7/5
reduced gas constant	R	414.37 J/K/kg

Table 5.7.1: Parameters for test 1

solution against a solution of the Euler equations since this is considered to be the more accurate model.

A consequence of this is that a solution generated by an interfaced method can never converge to the “exact” solution no matter how much the meshes are refined. The errors will instead plateau as the error from the numerical method is overtaken by the error from using an inconsistent model. The important questions are how quickly does the method reach the plateau and what is this final error? For these tests the “exact” solution was calculated by Roe’s method (see Chapter 3) on a mesh of $10,000$ points.

For each test the L_2 and interface errors of the pressure, flow and temperature fields were calculated as the mesh was refined. These variables were chosen as the most easily measurable in practice and therefore the most important. The mesh was refined in three different ways,

domain size		50,000 m
pipe diameter	d	.914 m
simulation time		40 s
initial pressure	p	$\sqrt{(1-\theta)70\times10^5+\theta 80\times10^5}$ Pa
		where $\theta = x/50000$
initial flow	q	-543.52 kg/m^{2}s
initial temperature	Θ	280 K
left boundary condition	pressure	80×10^5Pa
	temperature	291K
right boundary condition	pressure	80×10^5Pa
	temperature	280K
friction factor	f	0.002
ratio of specific heats	γ	7/5
reduced gas constant	R	414.37 J/K/Kg

Table 5.7.2: Parameters for test 2

- holding the parabolic mesh fixed and refining the hyperbolic mesh in space and time,
- refining the hyperbolic mesh in space and time and the parabolic in time only (the spatial parabolic mesh is determined by the NTS in BG's codes and is not as easy to refine)
- refining both meshes simultaneously in space and time.

The first set of tests was to compare the interface conditions for the iterative method. The second examines the data transfer method (see Section 5.5) which affects the noniterative method. Finally the best configurations of the iterative and noniterative methods are compared for accuracy and efficiency.

5.7.2 The iterative method of Section 5.2

In these tests we restricted ourselves to putting conditions on ϱ and T at the interface rather than P and T since this gave the best convergence behaviour. We also used the secant method of Section 5.2.5 to generate the interface values.

We compared the interface conditions

- Asymmetric

$$\frac{\varrho_0^1 - \varrho_0^0}{\Delta t} + \frac{Q_{1/2}^{1/2} - \sum_{k=0}^{m} q_{-1}^{k}}{\Delta x/2 + \delta x} = 0 \tag{5.7.1}$$

- pointwise

$$\frac{\varrho_0^1 - \varrho_0^0}{\Delta t} + \frac{Q_{1/2}^{1/2} - (q_i^0 + q_i^m)/2}{\Delta x} = 0 \tag{5.7.2}$$

  where point i is the nearest to the centre of the pipe and

- averaged

$$\frac{\varrho_0^1 - \varrho_0^0}{\Delta t} + \frac{Q_{1/2}^{1/2} - (\bar{q}_i^0 + \bar{q}_i^m)/2}{\Delta x} = 0 \tag{5.7.3}$$

  where the averages are taken over the pipe.

5.7.3 Results

Figures 5.7.3 to 5.7.5 compare the relative error from the interface conditions (5.7.1) (asymmetric), (5.7.2) (pointwise) and (5.7.3) (averaged) as δx is refined while $\Delta x =$ 10000m and $\Delta t = 20$s. The results from test problem 2 with the interface at 10,000 m were rendered meaningless by the presence of a shock near the interface so they are not presented.

As expected, the error reaches a plateau as the effect of the parabolic domain starts to dominate. The errors from the 'centred' interface conditions (5.7.2) (pointwise) and (5.7.3) (averaged) are too close to determine a winner. However, the asymmetric condition (5.7.1) is clearly superior in Figures 5.7.4 and 5.7.5. This is

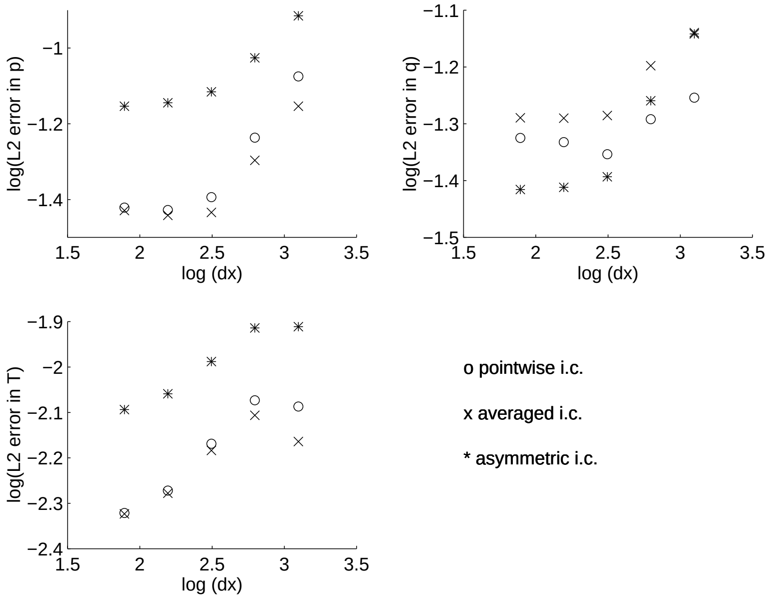


Figure 5.7.3: Interface conditions compared as δx is refined with the interface at 10,000m, test 1.

surprising at first since the other conditions should be an order of accuracy better. A likely explanation is that the centred conditions have stencils which lie in the transient, even when the interface is at 20,000 m, while the asymmetric condition takes its data from a smooth part of the solution. It is likely that the superiority of the centred schemes when the interface is at 10,000 m will be repeated when the interface is well away from the transient.

5.7.4 The noniterative method of Section 5.4.1

The noniterative method will be influenced by the method of data transfer used to map the solution from the parabolic mesh to the hyperbolic mesh. Figure 5.7.6 shows a comparison between

1. Mapping P, T, Q from hyperbolic to parabolic pointwise (equation (5.5.1)),

2. Mapping P, T as averages in each node and Q as an average in each pipe (equation (5.5.3)),

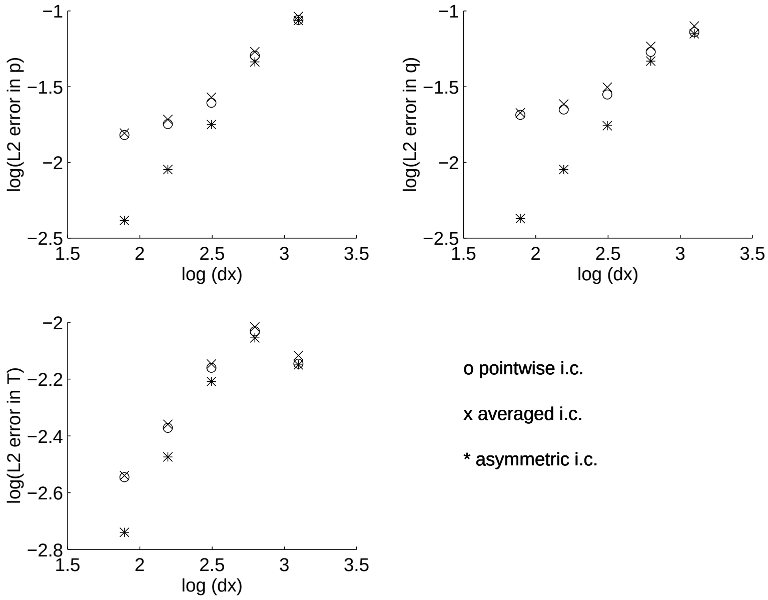


Figure 5.7.4: Interface conditions compared as δx is refined with the interface at 20,000m, test 1.

3. Mapping P, T pointwise and Q as an average,

4. Mapping P, T as averages and Q pointwise.

Only the results from test 1 with the interface at 20,000 m are presented since the other tests gave the same conclusion: mapping the nodal data (P and T) pointwise is superior to averaging. When the same experiment is tried but without the interface node being updated, that is it retains the values calculated by the parabolic solver, all four variants are identical. This implies that the inferior accuracy of the averaged technique is due to only averaging data in half a pipe at the interface node.

5.7.5 Comparison of the methods

Finally, we compare the best configurations of the iterative and noniterative methods for accuracy and computational expense. For the iterative method, we choose the asymmetric interface condition and for the noniterative method choose option 3, as offering the best performances.

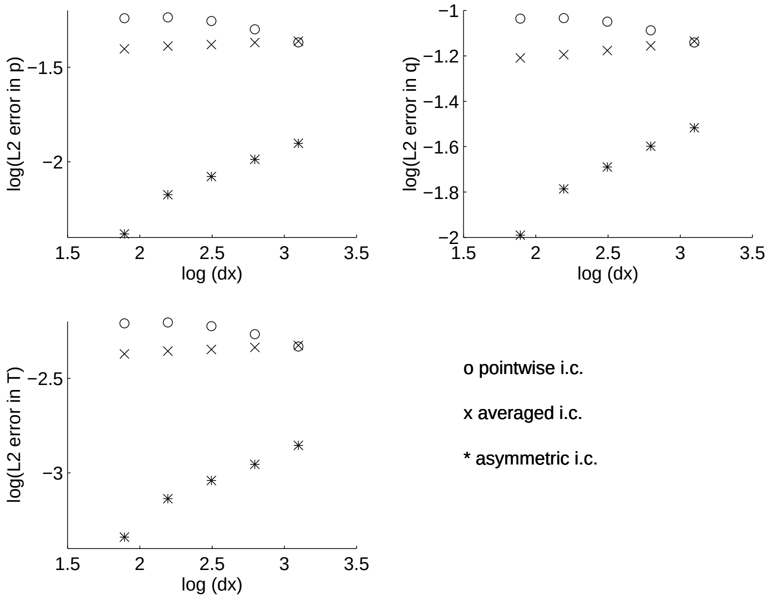


Figure 5.7.5: Interface conditions compared as δx is refined with the interface at 20,000m, test 2.

Figure 5.7.7 is an error plot for both methods with the three mesh refinement strategies described in Section 5.7.1. The left hand plot is of error against hyperbolic mesh size and it shows that the iterative scheme is superior for all three refinement strategies. It is clear from the improvement as the parabolic mesh is refined that the error in the parabolic solution has a much greater effect on the noniterative scheme, as would be expected. However, the important question is whether the superior accuracy of the iterative scheme outweighs its additional computational cost. The right-hand plot showing error against floating-point operations confirms that it does. The noniterative method can compete if the parabolic mesh is refined in space and time but this is impractical with current BG software since the spatial mesh is determined by the geometry of the network.

For a rough idea of the execution times see the plot of time against flops in Figure 5.7.8.

These results are supported by the other test problem. As well as being less accurate a disadvantage of the noniterative method is that since the parabolic method

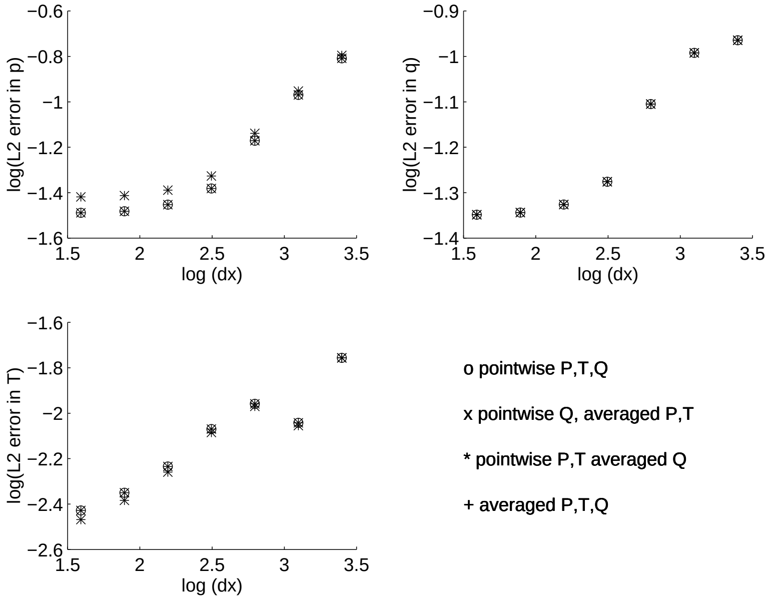


Figure 5.7.6: A comparison of methods to transfer data from the hyperbolic mesh to the parabolic mesh for the noniterative method as δx is refined with the interface at 20,000m, test 1.

is used on the entire network, disturbances are instantly transmitted everywhere which is unphysical.

5.8 Summary

- Interfaced methods offer a 'best of both worlds' approach.
- An iterative and a noniterative method have been described.
- The iterative method, although more complicated to implement, offers the most efficient and accurate solution.

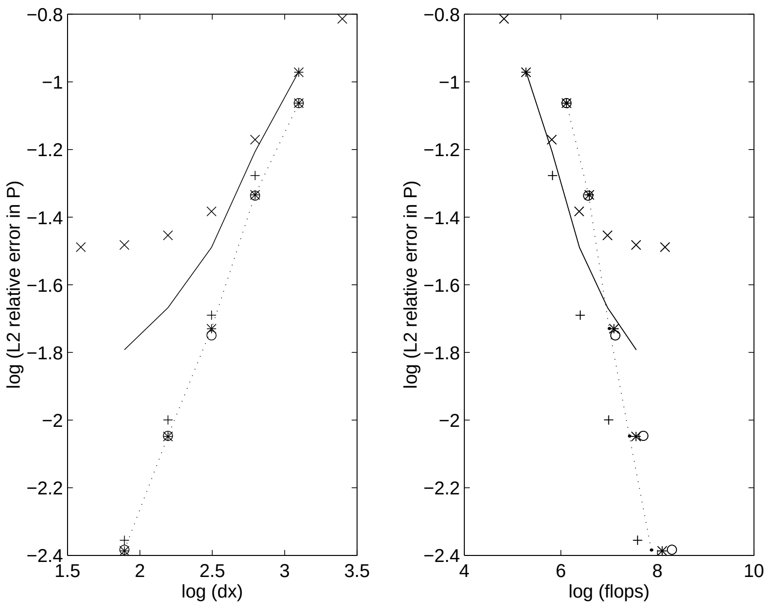


Figure 5.7.7: A comparison of the iterative and noniterative methods, the interface at 20,000m, test 1. Refining h-domain in space and time, o iterative, x noniterative; refining h-domain in space and time, p-domain in time only, ... iterative, - noniterative; refining h-domain and p-domain in space and time, * iterative, + noniterative.

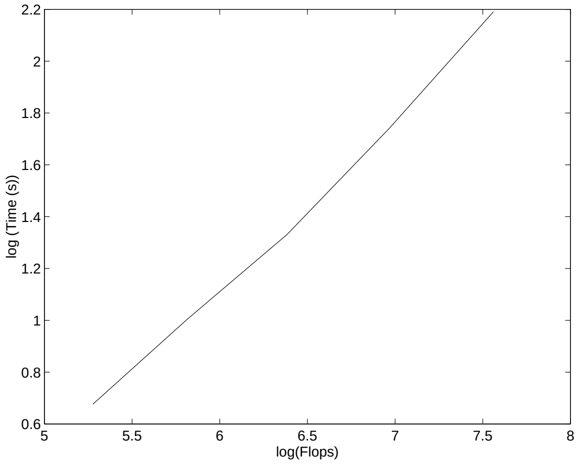


Figure 5.7.8: CPU Time against flops for one of the test problems as computed on a SparcStation 4/110.

Chapter 6

Branched networks

So far we have only considered the simulation of "linear" networks, that is, a single pipeline. In practice, we wish to model a complicated network of interconnecting pipes together with machines such as valves and compressors which impose constraints on the flow. In this chapter we examine the extension of the previous results to branched networks.

We begin by deriving natural extensions of the parabolic and hyperbolic numerical schemes and then discuss the additional problems for interfaced schemes in Section 6.3.

6.1 The parabolic domain

In this section we first state the the appropriate boundary conditions applicable at junctions of several pipes and then show how they may be implemented numerically. The Euler equations, from which the parabolic equations are derived, are based on the principle of conservation of mass, momentum and energy and we calculate the correct junction conditions from the same considerations. A typical junction is shown in Figure 6.1.1, the arrows represent the direction in which we measure positive flows, and not necessarily the direction of the flow. Conservation of mass at node 2 requires that mass fluxes at the ends of the associated pipes balance, that

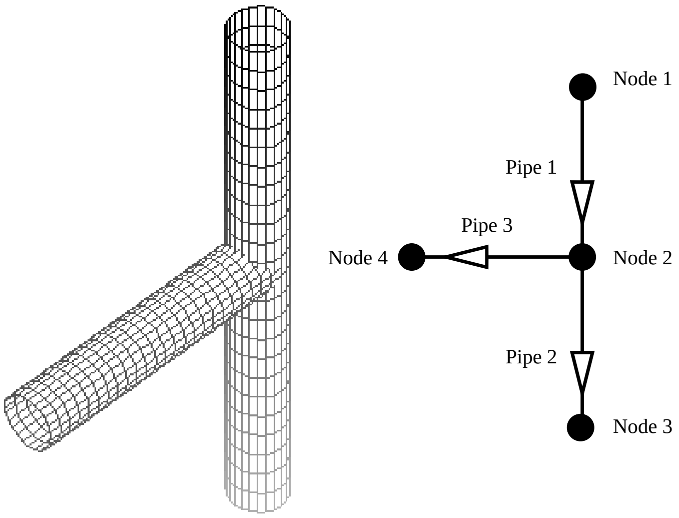


Figure 6.1.1: A typical junction

is,

$$A|_1 q|_1 = A|_2 q|_2 + A|_3 q|_3, \tag{6.1.1}$$

where $q|_m$ is the flow per unit area in pipe m at the end of the pipe connected to node 2 and $A|_m$ are the pipe cross section areas. We adopt the unusual notation $X|_m$ for quantity X in pipe m since the equations become more complicated in later sections.

Conservation of energy follows in a similar manner, with q replaced by the energy flux. Initially we consider the isothermal parabolic equations for which energy is not conserved and so this equation is not needed until Section 6.2.

In general, momentum will not be conserved due to the force exerted on the gas by the junction. This effect will be dependent on the shape of the junction, for example, we might expect a T-junction to impede the flow more than a fork. One way to model the momentum change is to use an empirical law relating the pressure drops across different types of junction. The simpler approach followed here is to model the momentum loss by adjusting the friction term.

At the junction itself we may simply equate pressures or the dynamic head $p + q^2/(2\rho)$. Goldwater and Fincham [GF81] note that in the gas transmission

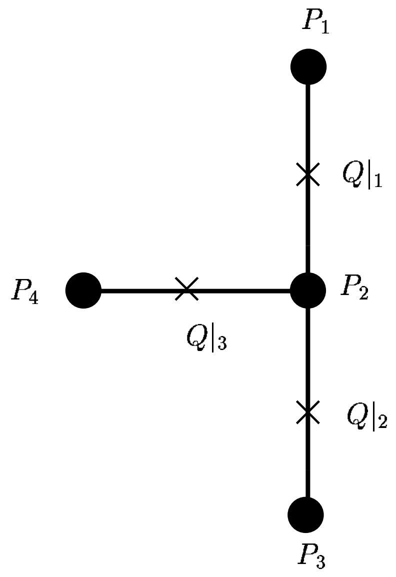


Figure 6.1.2: The mesh for the junction in Figure 6.1.1

industry the $q^2/(2\rho)$ term is negligible compared to the very high pressures, typically $p \simeq 70 \times 10^5$ Pa, $q \simeq 500$kg/s/m^2, $\rho \simeq 50$kg/m^3, and the conditions are nearly identical. Both conditions are only approximate since the momentum change is highly dependent on the junction geometry, so we choose to equate pressures since this is simpler to implement. In place of conservation of momentum we have

$$p|_1 = p|_2 = p|_3. \tag{6.1.2}$$

Incorporating boundary conditions (6.1.1) and (6.1.2) into the parabolic numerical scheme (4.2.1) is facilitated by the staggered mesh which is shown in Figure 6.1.2. The pressure at node 2, P_2, is common to all 3 pipes and so condition (6.1.2) is implemented automatically. It was pointed out in Section 5.2 that the staggered scheme (4.2.1) for the parabolic equations can be seen as a finite volume scheme where the mass associated with a node is given by

$$A\Delta x \frac{P}{RT}.$$

Generalising this to a node connected to several pipes gives the mass associated with node 2 as

$$\sum_{m=1}^{3} \frac{A|_m \Delta x|_m}{2} \frac{P_2}{RT}$$

and the numerical equivalent of condition (6.1.1) is

$$\sum_{m=1}^{3} \frac{A|_m \Delta x|_m}{2} \frac{(P_2^{n+1} - P_2^n)}{RT} + \Delta t \left(A|_2 Q^{n+\theta}|_2 + A|_3 Q^{n+\theta}|_3 - A|_1 Q^{n+\theta}|_1 \right) = 0. \tag{6.1.3}$$

General networks may be described by a *connectivity* matrix C which defines how pipes and nodes are connected. The entries of C are

$$c_{ml} = \begin{cases} -1 & \text{if the left-hand end of pipe } m \text{ connects to node } l \\ +1 & \text{if the right-hand end of pipe } m \text{ connects to node } l \\ 0 & \text{otherwise} \end{cases}$$

and we measure x-increasing from left to right. For example, the connectivity matrix for the simple network in Figure 6.1.1 is

$$C = \begin{pmatrix} -1 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & -1 & 0 & 1 \end{pmatrix}.$$

Extending equation (6.1.3) to a general network is straightforward once account of the pipe directions is taken. For node l we have

$$\sum_{c_{ml} \neq 0} \frac{A|_m \Delta x|_m}{2} \frac{(P_l^{n+1} - P_l^n)}{RT} - \Delta t \sum_m c_{ml} A|_m Q^{n+\theta}|_m = 0. \tag{6.1.4}$$

Note that for a linear network of equal area pipes with connectivity matrix

$$C = \begin{pmatrix} -1 & 1 & 0 & 0 & \dots \\ 0 & -1 & 1 & 0 & \dots \\ 0 & 0 & -1 & 1 & \dots \\ \vdots & \vdots & \vdots & \ddots & \ddots \end{pmatrix},$$

equation (6.1.4) collapses to the discretisation for the mass equation (4.2.1). It will be shown in Section 6.2 that this consistency between boundary conditions at junctions and the numerical method is harder to achieve for the hyperbolic equations. The discretisation of the momentum equation (4.0.2) remains unchanged as

$$\frac{1}{2} \frac{(P^2)_R^{n+1} - (P^2)_L^{n+1}}{\Delta t} + \mu RT Q^{n+1}|_m |Q^{n+1}|_m| = 0$$

where R and L are chosen such that $c_{mR} = 1$ and $c_{mL} = -1$.

In Figures 6.1.3 to 6.1.5 we show an example solution from this scheme. The network is a pipeline of six 10,000m pipes, with an off-take half way along. The off-take consists of three 1,000m pipes. All of the pipes are circular of area 0.5m^2 and the friction factor is $f = 0.002$. Initially, the gas is flowing steadily from a pressure of 80×10^5 Pa at the left to 75×10^5 Pa at the right. At time $t = 0$ the flow rate at the off-take is raised linearly from 0 to 300 kg/s/m^2 over a period of 5 minutes. Figures 6.1.3 and 6.1.4 show the initial and final conditions and Figure 6.1.5 plots pressures and flows against time at selected points in the network.

6.1.1 Including temperature variation in the parabolic domain

Temperature variation in the parabolic domain can be accounted for in the same way as in Section 4.3. The discretised mass equation (6.1.4) becomes

$$\sum_{c_{ml}\neq 0} \frac{A|_m \Delta x|_m}{2R}\left(\frac{P_l^{n+1}}{T_l^{n+1}} - \frac{P_l^n}{T_l^n}\right) - \Delta t \sum_m c_{ml} A|_m Q^{n+\theta}|_m = 0$$

and the momentum equation changes as before from (4.2.2) to (4.3.2). The temperature at the new time level is unknown and must be calculated by using an energy equation such as the entropy equation (2.3.3) and extending the method of Section 4.3.2. Within each pipe the entropy equation (4.3.3) is discretised as

$$\frac{s_Y^{n+1} - s_Y^n}{\Delta t} = -c_{mY} u^n|_m \frac{s_Y^n - s_X^n}{\Delta x|_m} + B^n|_m \qquad (6.1.5)$$

where nodes X and Y are chosen so that the flow $Q^n|_m$ is from X to Y and

$$u^n|_m = \frac{2Q^n|_m}{\varrho_X^n + \varrho_Y^n}.$$

Since each node may be associated with several pipes the new value of the entropy at a node as calculated by formula (6.1.5) might be undefined or multi-valued. The former occurs when the node is a stagnation point and all flows are away from it.

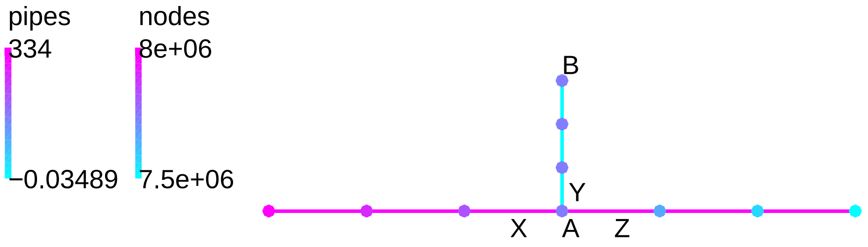


Figure 6.1.3: The initial conditions on the network, pressures in Pa, flows in kg/s/m^2

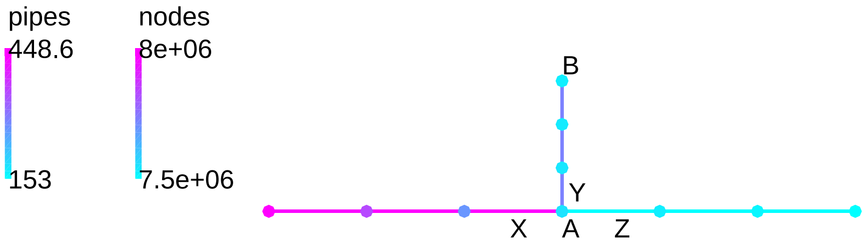


Figure 6.1.4: The final conditions on the network, pressures in Pa, flows in kg/s/m^2

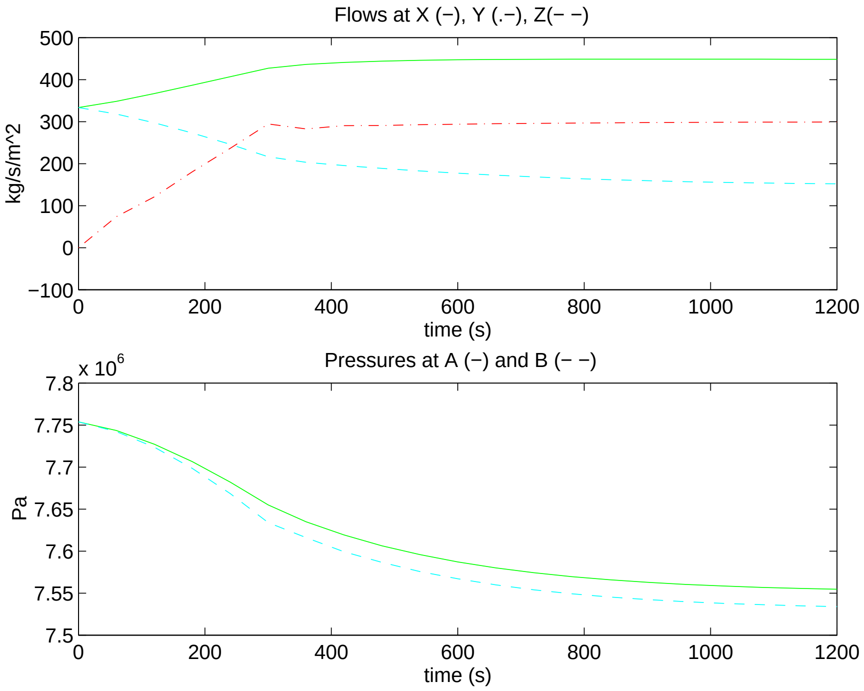


Figure 6.1.5: Pressures and flows against time

In this case we calculate the entropy by solving the ODE

$$\frac{s_Y^{n+1} - s_Y^n}{\Delta t} = B^n|_m.$$

In the latter case when several pipes are flowing into a node an average of the contributions from each pipe, weighted according to the magnitude of its flow, is assigned to the node. The need for these approximations is a drawback of the scheme, but the only correct way to simulate junctions and stagnation points is to discretise the energy equation in conservation form which would prohibit the use of an explicit scheme except with impractically small timesteps. Since in the parabolic domains the gas is almost isothermal the error incurred should be acceptable.

6.2 The hyperbolic domain

In this section we consider a junction between pipes all of which are hyperbolic. We begin by using characteristic directions to calculate the number of boundary conditions necessary at a junction for a well-posed problem. In Section 3.4 it was stated that each boundary can be designated as an inflow or an outflow depending on whether the flow is away from or towards it. In the networks in which we are interested all flows are subsonic and so an inflow boundary always has one outgoing characteristic, two incoming characteristics and requires two boundary conditions, for example a specified pressure and temperature. Similarly, an outflow boundary has two outgoing characteristics, one incoming characteristic and requires only a single boundary condition.

For a general junction of m_1 inflow pipes and m_2 outflow pipes we need therefore a total of $2m_1 + m_2$ boundary conditions.

In the previous section 6.1 we stated that the boundary conditions at a junction may be derived from conservation of mass and energy and a model for momentum loss. Conservation of mass at the lth node imposes the condition

$$\sum_{m=1}^{m_1+m_2} A|_m c_{ml} q|_m = 0. \tag{6.2.1}$$

In a similar way conservation of energy gives the second boundary condition

$$\sum_{m=1}^{m_1+m_2} A|_m c_{ml} \left(\frac{q|_m}{\rho|_m} \left(E|_m + p|_m\right) \right) = 0. \tag{6.2.2}$$

Momentum is not conserved at the junction and this effect must be modelled by adjusting the friction factors (see Section 6.1). Instead we have the $m_1 + m_2 - 1$ equations

$$p|_m = p|_{m+1} \qquad m \leq m_1 + m_2 - 1. \tag{6.2.3}$$

To provide the remaining $2m_1 + m_2 - (1 + 1 + m_1 + m_2 - 1) = m_1 - 1$ equations note that we have no information about the junction to distinguish between the m_1 inflow pipes (recall that an inflow pipe has gas flowing *out* of the junction), and so we assume a perfect mixing of the gas in the junction and equate the temperatures at the ends of the inflow pipes. Due to the equality of pressures (6.2.3) this is equivalent to equating entropies or densities. This latter condition

$$\rho|_m = \rho|_{m+1} \qquad m \leq m_1 - 1 \tag{6.2.4}$$

(after ordering the pipes appropriately), is the easiest to apply in practice. A number of alternative conditions taking into account the areas of the pipes are possible but it should be stressed that none of these conditions is "correct" since we are using a one-dimensional model to simulate a three dimensional phenomenon. Condition (6.2.4) simply states that we do not know enough about the junction to distinguish between the inflow pipes.

Note that $c_{ml}q|_m > 0$ implies that the pipe is outflow and $c_{ml}q|_m < 0$ implies that it is inflow. We must have at least one inflow pipe for the number of boundary conditions to be correct and to make physical sense. Fortunately, equation (6.2.1) ensures that there is always at least one inflow and one outflow pipe (except in the trivial case of $q|_m = 0 \quad \forall m$.)

6.2.1 Implementing the boundary conditions

The previous section shows that we have the correct number of boundary conditions to solve the junction problem. However, as was noted in Chapter 3, numerical methods often require more boundary conditions than the underlying equations. These additional numerical boundary conditions may be supplied by several means, such as extrapolation, one-sided differences or characteristic tracing.

A desirable property of a numerical method for a junction is that when it is applied to a junction of only two pipes of equal areas, the numerical boundary conditions collapse to the method used in the interior of each pipe. This ensures that where pipes are connected into pipelines numerical conservation will be respected between the pipes and shocks will be correctly positioned. In contrast to the parabolic numerical scheme it is difficult to derive a treatment for junctions in the hyperbolic scheme which is consistent with the underlying scheme in this way.

Figure 6.2.1 illustrates the potential danger of using an incompatible scheme at the junction. The figure shows two pipes connected at a junction denoted by the dotted line. Isentropic characteristics (see Chapter 3) are used to implement the boundary conditions of the previous section while the second-order Roe scheme (3.3.14) is used to calculate the flow in each pipe. In effect Roe's scheme is used everywhere except at a single point which is solved by the method of characteristics. Initially the gas is at rest with a sharp discontinuity in the pressure in the left-hand pipe. This resolves itself into a leftward moving rarefaction wave and a rightward moving shock and contact discontinuity. The shock passes through the junction almost unaffected. It is only when the contact discontinuity strikes that the method fails. The oscillations become so great that the scheme fails a few timesteps on from the figure, due to the CFL number exceeding unity.

If the method of characteristics (MOC) is used as the numerical scheme in the interior of the pipes then the junction boundary conditions can easily be implemented in a consistent way. Fox [Fox89] shows how to use characteristics to implement

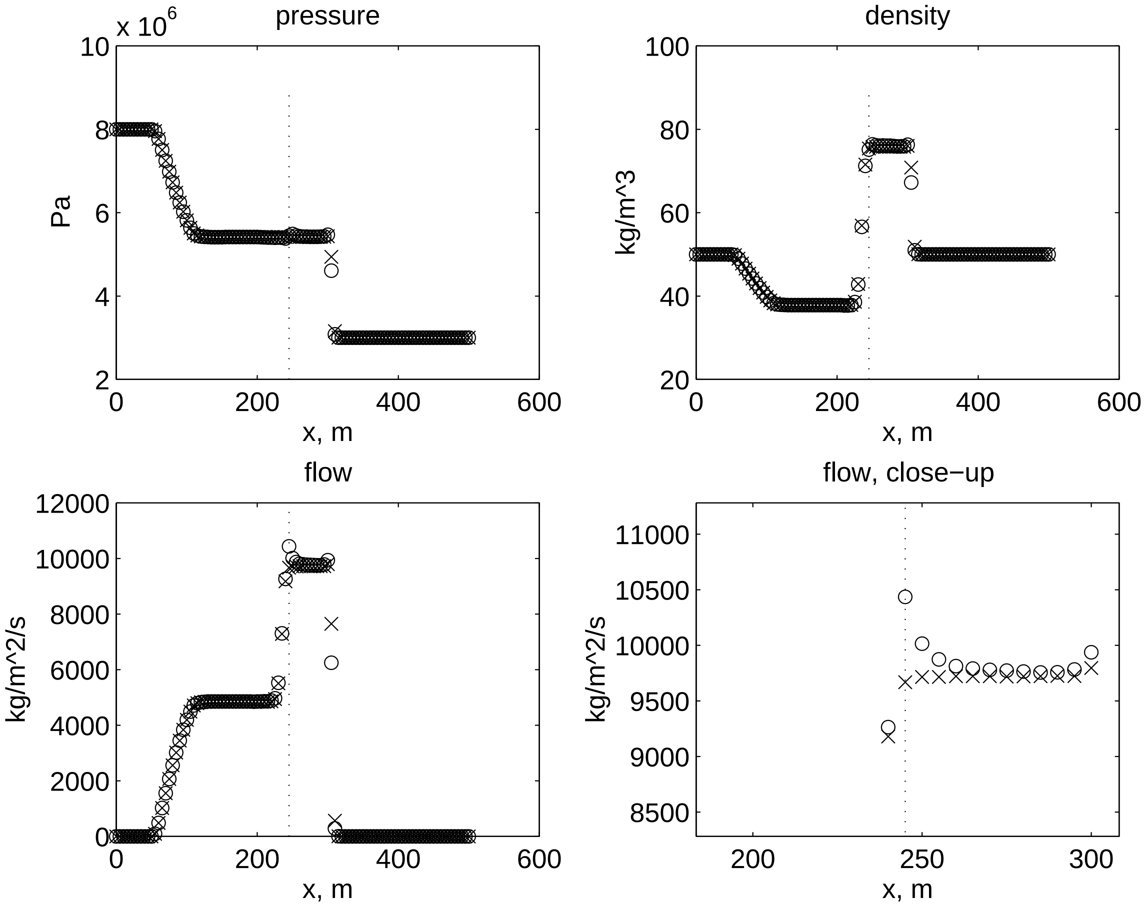


Figure 6.2.1: Instability caused by the use of an inconsistent scheme at a junction. The crosses represent the solution calculated by Roe's scheme everywhere while the circles represent the solution obtained by using MOC at the junction.

boundary conditions for junctions in river networks, which is a very similar problem.

Unfortunately, as was stated in Chapter 3, we should not use MOC if we wish to capture shocks since it is not conservative for the Euler equations. Godunov-type schemes were developed as ways of using characteristic information in a conservative scheme and so it is natural to turn to these as a way of using the characteristics to implement the boundary conditions. In the next section we show how Roe's method, and the cell-edge boundary conditions of Section 3.4.2 can be generalised to junctions.

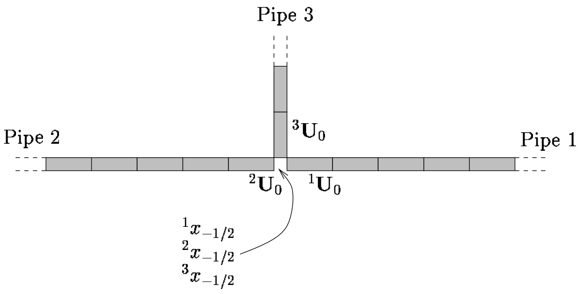


Figure 6.2.2: Arrangement of the computational cells at the junction

6.2.2 Numerical boundary conditions for junctions

Section 3.4.2 described a natural method of applying boundary conditions to Roe's method by enforcing the boundary conditions on the cell-edges. In this section we extend this method to the boundary conditions at junctions. Consider a junction of M pipes. The computational cells are arranged as in Figure 6.2.2 so that the junction lies at the outside edges of the extreme cells in the incident pipes and in general the flow variables will be discontinuous at this point. We must therefore solve for $3M$ cell-edge quantities, 3 in each pipe. Repeating the method of Section 3.4.2, we linearise the Euler equations in the cells next to the junction. The details of the linearisation, which will in general be different in each pipe are deferred until later in this section.

The basic algorithm is the same as before but the notation is more complicated.

Algorithm 6

- *Calculate the eigenvalues and vectors for the M matrices $A|_m$,*
- *Define the cell-edge states $\mathbf{U}_E|_m = \alpha_1|_m\mathbf{r}_1|_m + \alpha_2|_m\mathbf{r}_2|_m + \alpha_3|_m\mathbf{r}_3|_m$,*
- *Solve for the $3M$ coefficients $\alpha_j|_m$ by solving the junction boundary conditions (6.2.1) to (6.2.4),*
- *Use the $\alpha_j|_m$ to calculate the jumps $\delta_j|_m$ and update the boundary cells using equation (3.3.14).*

Calculating the $\alpha_j|_m$ requires Newton iteration since the boundary conditions are nonlinear. Initial guesses for $\alpha_j|_m$ are provided by $\beta_j|_m$. At each iteration the new approximations to the edge states $\mathbf{U}_E|_m = \sum_{j=1}^{3} \alpha_j|_m \mathbf{r}_j|_m$ are calculated, followed by the pressures $p_E|_m$ and energy fluxes $F|_m := ((E+p)q/\rho)_E|_m$.

Denoting the residual of the equations for $\alpha_j|_m$ by the $3M$-vector $\mathbf{g}$, the associated Jacobian J and the unknowns $\mathbf{x} = (\alpha_1|_1, \alpha_2|_1, \alpha_3|_1, \alpha_1|_2, \alpha_2|_2, \alpha_3|_2, \ldots)$, an algorithm to set up the Jacobian and residual is:

Algorithm 7

```
•Set inflowFlag=0

•for m=1:M

    –if $c_{ml}\lambda_1|_m \le 0$:   $g_{3(m-1)+1} = \alpha_1|_m - \beta_1|_m$ %outgoing characteristic

    –if $c_{ml}\lambda_3|_m \le 0$:   $g_{3(m-1)+1} = \alpha_3|_m - \beta_3|_m$ %outgoing characteristic

    –if $c_{ml}\lambda_2|_m \le 0$:   $g_{3(m-1)+2} = \alpha_2|_m - \beta_2|_m$ %outgoing characteristic

    –if $c_{ml}\lambda_2|_m > 0$ then

        *if inflowFlag$= 0$:  %conservation of energy
        $g_{3(m-1)+2} = \sum_{m'} F|_{m'}$

        *if inflowFlag$\neq 0$:  %equal densities
        $g_{3(m-1)+2} = \rho_E|_m - \rho_E|_{inflowFlag}$

    –set inflowFlag= $m$

    –endif

    –if $m = 1$:   $g_{3(m-1)+3} = g_3 = \sum_{m'} q_E|_{m'}$ %conservation of mass

    –if $m \neq 1$:   $g_{3(m-1)+3} = p_E|_m - p_E|_{m-1}$ %equal pressures

•endfor
```

The first two `if` *statements in the* `for`*-loop are mutually exclusive since the flow is subsonic.*

The entries of J are calculated from equation (3.4.1) and the Newton iteration $\mathbf{x}_{n+1} = \mathbf{x}_n - J^{-1}\mathbf{g}(\mathbf{x}_n)$ performed in the usual way.

Including the trivial equations $\alpha_j|_m = \beta_j|_m$ in the system to be solved considerably simplifies the code since we always have $3M$ equations to solve irrespective of the number of inflow and outflow pipes and their orientations while adding little to the computational work, since M is almost always three or less. The Jacobian J is a function of the flow speeds $\lambda_2|_m$ at the junction. In some situations $|J| \to 0$ as $\lambda_2|_m \to 0$ and the Newton iteration will fail. This can be remedied by imposing a lower limit on the absolute value of $\lambda_2|_m$ passed to J at a cost of slightly reducing the convergence rate.

The linearisation at the junction

We now show that the correct choice of linearisation at a junction of two equal area pipes leads to an implementation of the boundary conditions consistent with the underlying scheme, Roe's method. Consider a junction of two pipes and let pipe 1 be an outflow pipe with rightmost cell $x_J|_1$ adjacent to the leftmost cell $x_0|_2$ of pipe 2, an inflow pipe. Adjacent to the interface in cells $x_J|_1, x_0|_2$ we use the same linearisation $A(\mathbf{U}_J|_1, \mathbf{U}_0|_2)$ in both pipes so that the eigenvectors $\mathbf{r}_i$ and eigenvalues λ_i are identical. Constructing the edge states using Algorithm 6 gives

$$\mathbf{U}_E|_1 = \xi_1\mathbf{r}_1 + \beta_2|_1\mathbf{r}_2 + \beta_3|_1\mathbf{r}_3$$

$$\mathbf{U}_E|_2 = \beta_1|_2\mathbf{r}_1 + \xi_2\mathbf{r}_2 + \xi_3\mathbf{r}_3$$

where the 3 unknown ξ_j are chosen such that

$$q_E|_1 = q_E|_2 \qquad \text{conservation of mass}$$

$$p_E|_1 = p_E|_2 \qquad \text{continuity of pressure}$$

$$\frac{q|_1}{\rho|_1}(E|_1 + p|_1)|_E = \frac{q|_2}{\rho|_2}(E|_2 + p|_2)|_E \qquad \text{conservation of energy.}$$

Using the equation of state $E = q^2/2\rho + p/(\gamma - 1)$ and simplifying gives an equation relating $\rho_E|_1$ to $\rho_E|_2$ for given p_E and q_E

$$(\rho_E|_1 - \rho_E|_2)\left(\frac{\gamma p_E}{\gamma - 1}\rho_E|_1\rho_E|_2 + q_E^2(\rho_E|_1 + \rho_E|_2)\right) = 0.$$

This equation is has two solutions, $\rho_E|_1 = \rho_E|_2$, and a nonphysical one where the densities have opposite signs. We assume that the Newton iteration converges to the former solution since the starting values of $\xi_j|_m$ are given by the states in the end cells.

Thus, solving the junction boundary conditions will generate identical edge conditions at the boundary of both pipes:

$$\mathbf{U}_E|_1 = \mathbf{U}_E|_2 = \sum_{\lambda_j<0} \beta_j|_2\mathbf{r}_j + \sum_{\lambda_j\geq 0} \beta_j|_1\mathbf{r}_j.$$

The update to cell 0 in pipe 2 is then found from equation (3.3.14),

$$\mathbf{U}_0^{n+1}|_2 = \mathbf{U}_0^n|_2 - \frac{\Delta t}{\Delta x}\left(\sum_{\lambda_{-1/2,j}>0} \lambda_{-1/2,j}\delta_{-1/2,j}\mathbf{r}_{-1/2,j} + \sum_{\lambda_{+1/2,j}\leq 0} \lambda_{+1/2,j}\delta_{+1/2,j}\mathbf{r}_{+1/2,j}\right), \tag{6.2.5}$$

where

$$\delta_{i-1/2,j} = \begin{cases} \beta_j|_2 - \beta_j|_1 & \lambda_{i-1/2,j} \geq 0 \\ 0 & \lambda_{i-1/2,j} < 0. \end{cases}$$

Comparing this with $\mathbf{U}_0^{n+1}|_2$ as calculated when $x_J|_1$ and $x_0|_2$ are treated as adjacent internal cells in Roe's method, we see that the formulae are formally identical. The only difference is that for Roe's method,

$$\delta_{-1/2,j} = \beta_j|_2 - \beta_j|_1 \quad \forall j.$$

Since the $\delta_{-1/2,j}$ are not used in equation (6.2.5) when $\lambda_{-1/2,j} < 0$ this gives an identical value. Clearly this is only true if the eigenvalues λ and eigenvectors $\mathbf{r}$ are calculated by Roe's linearisation (3.3.12) which is based on the averages

$$\bar{u} = \frac{\sqrt{\rho_J|_1} u_J|_1 + \sqrt{\rho_1|_2} u_1|_2}{\sqrt{\rho_J|_1} + \sqrt{\rho_1|_2}} \tag{6.2.6}$$

$$\bar{h} = \frac{\sqrt{\rho_J|_1} h_J|_1 + \sqrt{\rho_1|_2} h_1|_2}{\sqrt{\rho_J|_1} + \sqrt{\rho_1|_2}} \tag{6.2.7}$$

$$\bar{a}^2 = (\gamma - 1)(\bar{h} - \bar{u}^2/2). \tag{6.2.8}$$

To summarise, if we use Roe's linearisation then this method of implementing the boundary conditions is consistent with Roe's method when $M = 2$. The only change required for a junction of more than two pipes is to the linearisation.

Generalising the linearisation to $M > 2$ may be done in many ways by choosing different averages $\bar{u}$, $\bar{h}$. An obvious choice is

$$\bar{u}|_m = \frac{\sqrt{\rho|_m} u|_m - c_{ml} \sum_{k \neq m}^{M} c_{kl} \sqrt{\rho|_k} u|_k}{\sum_{k=1}^{M} \sqrt{\rho|_k}} \tag{6.2.9}$$

$$\bar{h}|_m = \frac{\sum_{k=1}^{M} \sqrt{\rho|_k} H|_k}{\sum_{k=1}^{M} \sqrt{\rho|_k}}. \tag{6.2.10}$$

The sign of $c_{ml}\bar{u}|_m$ determines whether pipe m is an inflow or an outflow at node l and therefore the number and type of boundary conditions. It is a numerical approximation and so it does not necessarily reflect the true solution. For example, if we have a junction of $M > 2$ pipes of identical speeds and densities (that is, $c_{ml}\sqrt{\rho|_m} u|_m$ are the same) then, using equation (6.2.9),

$$\begin{aligned} c_{ml}\bar{u}|_m &= \frac{c_{ml}\sqrt{\rho|_m} u|_m - \sum_{k \neq m}^{M} c_{kl} \sqrt{\rho|_k} u|_k}{\sum_{k=1}^{M} \sqrt{\rho|_k}} \\ &= \frac{(2 - M) c_{ml} \sqrt{\rho|_m} u|_m}{M\sqrt{\rho|_m}}, \end{aligned}$$

that is, all the $c_{ml}\bar{u}|_m$ are the same and all of the pipes are inflows or all are outflows. As was noted in Section 6.2, this results in an ill-posed problem. This can be remedied by using a different generalised Roe-average such as

$$\begin{aligned}
\bar{u}|_m &= \frac{\sqrt{\rho|_m}u|_m - \frac{c_{ml}}{M-1}\sum_{k\neq m}^{M} c_{kl}\sqrt{\rho|_k}u|_k}{\sqrt{\rho|_m} + \frac{1}{M-1}\sum_{k\neq m}^{M}\sqrt{\rho|_k}} \\
\bar{h}|_m &= \frac{\sqrt{\rho|_m}u|_m + \frac{1}{M-1}\sum_{k\neq m}^{M}\sqrt{\rho|_k}H|_k}{\sqrt{\rho|_m} + \frac{1}{M-1}\sum_{k\neq m}^{M}\sqrt{\rho|_k}}
\end{aligned}$$

which gives $\bar{u} = 0$, a stagnation point, in the same situation.

A further possibility is to choose

$$\begin{aligned}
\bar{u}|_m &= \frac{\sqrt{\rho|_m}u|_m - \frac{c_{ml}}{M-1}\sum_{k\neq m}^{M} c_{kl}\sqrt{\rho|_k}u|_k}{\frac{2}{M}\sum_{k=1}^{M}\sqrt{\rho|_k}} \\
\bar{h}|_m &= \frac{\sqrt{\rho|_m}u|_m + \frac{1}{M-1}\sum_{k\neq m}^{M}\sqrt{\rho|_k}H|_k}{\frac{2}{M}\sum_{k=1}^{M}\sqrt{\rho|_k}}
\end{aligned} \tag{6.2.11}$$

which not only has the correct form when $M = 2$ but also has the property

$$\begin{aligned}
\sum_{m=1}^{M} c_{ml}\bar{u}|_m &= \left(\sum_{m=1}^{M} c_{ml}\sqrt{\rho|_m}u|_m - \frac{1}{M-1}\sum_{m=1}^{M}\sum_{k\neq m}^{M} c_{kl}\sqrt{\rho|_k}u|_k\right) / \left(\frac{2}{M}\sum_{k=1}^{M}\sqrt{\rho|_k}\right) \\
&= \left(\sum_{m=1}^{M} c_{ml}\sqrt{\rho|_m}u|_m - \frac{1}{M-1}\sum_{k=1}^{M-1}\sum_{m=1}^{M} c_{ml}\sqrt{\rho|_m}u|_m\right) / \left(\frac{2}{M}\sum_{k=1}^{M}\sqrt{\rho|_k}\right) \\
&= 0
\end{aligned}$$

guaranteeing that not all $c_{ml}\bar{u}|_m$ are of the same sign, and there is always one inflow and one outflow pipe.

The choice of linearisation makes little difference to the solution itself. Figures 6.2.3 and 6.2.4 shows two example solutions for a three-pipe junction calculated with linearisation (6.2.11).

It should be emphasised that these linearisations only give conservation in the $M = 2$ case. When $M > 2$ one-dimensional conservation of momentum is no longer appropriate and there may be a significant momentum transfer from the gas to the pipes. The linearisation will not exactly conserve mass and energy either, though the boundary conditions ensure that they will be conserved in the limit as the mesh is refined.

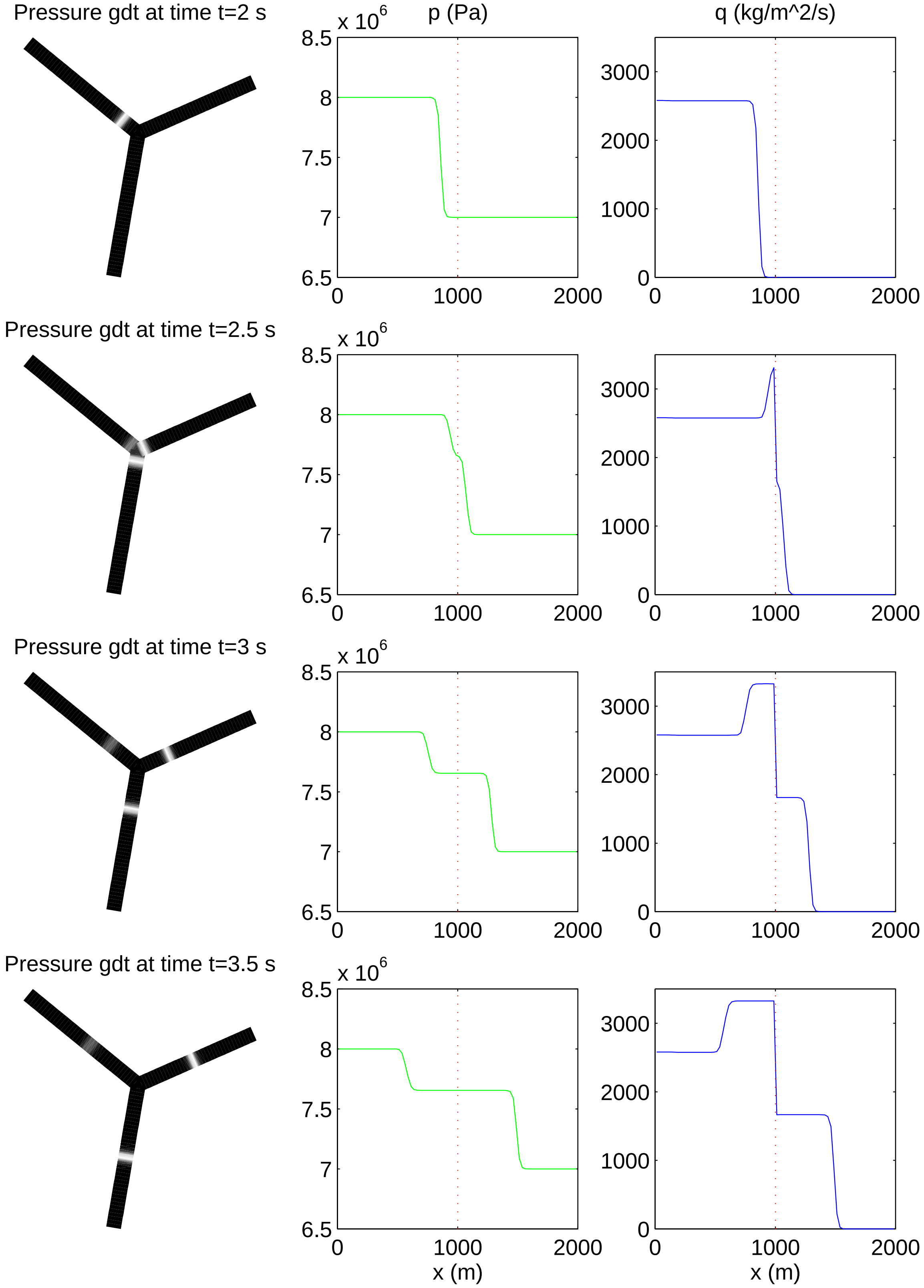


Figure 6.2.3: A shock striking a three pipe junction. Each pipe has 40 mesh points, and the right hand figures show the solution in the input pipe juxtaposed with that in one of the output pipes.

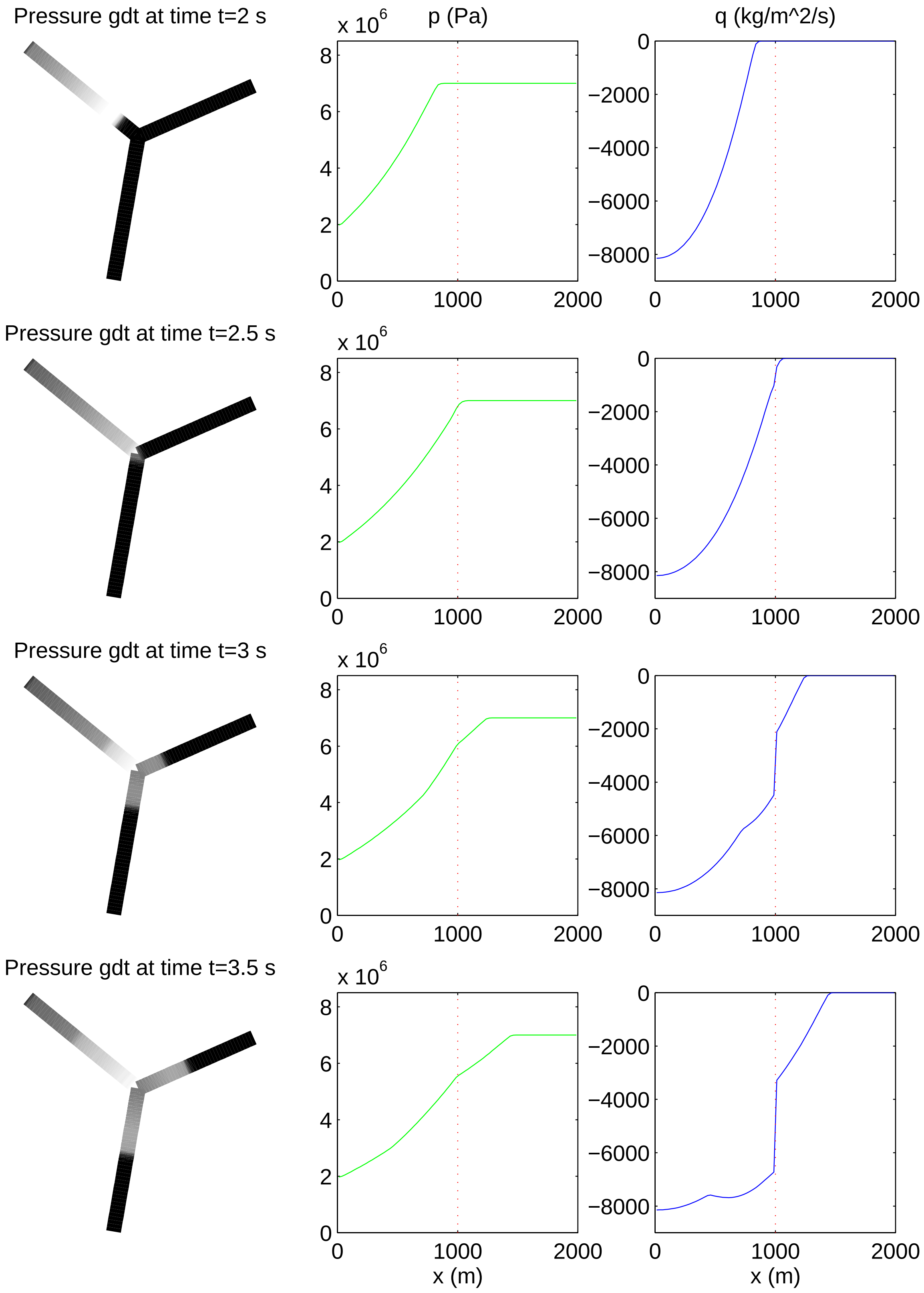


Figure 6.2.4: A rarefaction wave caused by a pipe-break striking a three pipe junction. Each pipe has 40 mesh points, and the right hand figures show the solution in the input pipe juxtaposed with that in one of the output pipes.

Second-order corrections

The previous section demonstrated a method of implementing the junction boundary conditions which collapses to Roe's scheme when the junction is of two equal pipes. However, Roe's scheme is only first-order accurate. Section 3.3.1 showed how to improve the order of accuracy by using flux limiters. To extend the second-order method to the the boundary requires extrapolation to provide the "missing" waves (see Section 3.4.2). However, if this is done at a junction then we will lose the consistency with the second-order Roe scheme. To maintain consistency requires that the interaction between the pipes is used to generate the "missing" waves at the boundary. It is not difficult to formulate an averaging procedure which does this, taking due account of the orientations of the pipes and directions of flows, but it does make the coding appreciably more complicated. In practice as Figure 6.2.3 shows, using extrapolation to provide the missing waves works well and the gain in simplicity outweighs the loss of consistency.

6.3 The interfaced methods

The previous two sections describe the additional considerations when using the parabolic and hyperbolic models on branched networks. The interfaced methods of Chapter 5 carry across to branched networks without requiring much modification. The areas which do require attention are described here.

6.3.1 Interfaces at junctions

An interface in a linear network always separates a hyperbolic domain from a parabolic one. In a branched network the possibility exists of an interface between several hyperbolic and parabolic domains.

For the noniterative method of Section 5.4.1 this does not present a problem. The network is first solved by the parabolic solver, using the numerical scheme

(6.1.4). The hyperbolic domains are then solved pipe by pipe using the method of Section 6.2. If a hyperbolic pipe joins to a node which is entirely hyperbolic then the boundary conditions are implemented as in Section 6.2. If one or more connecting pipes is parabolic then the boundary conditions are supplied by interpolation from the parabolic domain as described in Section 5.4.1.

An outline for an algorithm to apply the noniterative method to a branched network is

Algorithm 8

```
•for n=1 to numberParabolicSteps
    –[$P^{n+1}, Q^{n+1}, T^{n+1}$]=parabolic($P^n, Q^n, T^n$,time$^n$)
    –for k=1:numberHyperbolicSteps
        ∗for node=1 to numberNodes
            ·if allPipesHyper(node)=TRUE then
            ·calcNodalWaves1 % according to Section 6.2
            ·elseif allPipesPara(node)=FALSE then
            ·calcNodalWaves2 % according to Section 5.4.1,
             using $\varrho^{n+1}_{node}$ etc as a boundary condition
            ·else %do nothing
            ·Endif
        ∗Endfor
        ∗hPipes:=getIndices(isHyper(n,:)=TRUE)
        ∗for pipe∈hPipes
            ·[$\rho^{k+1}|_{pipe}, \ldots$]=doOneHyperStep($\rho^k|_{pipe}, \ldots$, nodalWaves)
        ∗Endfor
    –Endfor
```

```
•Endfor
```

On the other hand the iterative method of Section 5.2 does require special treatment at such junctions. Rather than devise generalised interface conditions for mixed nodes (which is not difficult - the interface condition can be generalised in the same way as equation (6.1.4)) it is perhaps simpler to disallow them. If an interface is moved to such a node then the hyperbolic domains are enlarged slightly to move the interface to a two-pipe node.

6.3.2 Tracking transients

Section 5.6 describes two ways of enlarging the hyperbolic domains. The first attempts to detect transients as they strike interfaces. The second method, which was preferred, is to extend the hyperbolic domain at every parabolic step and then retract it where it is unnecessary. The former method may be applied to branched networks almost unchanged, though the code is more complicated. The second method may be easily implemented by a recursive code such as:

Algorithm 9

Given an array `isHyperOld(i)=True` *if pipe i is hyperbolic, for each hyperbolic pipe m at the old parabolic time level call:*

```
%Extend to the left
isHyper=extendHyper(m, $\Delta t\times$ sound speed, -1, isHyperOld, isHyperOld)
%Extend to the right
isHyper=extendHyper(m, $\Delta t\times$ sound speed, +1, isHyperOld, isHyperOld)
```

where the function `extendHyper()` *is*

```
function isHyper=extendHyper(pipe, distance, direction, isHyper,
isHyperOld)
 •if distance $\leq$ 0 then return % Nothing to do
```

```
•if direction=-1 then node=leftNode(pipe)
•if direction=+1 then node=rightNode(pipe)
•pipes:= pipesConnectedTo(node)
•for i∈pipes
    –if (i≠pipe & isHyperOld(i)=FALSE) then % extend domain
        ∗isHyper(i):=TRUE
        ∗isHyper=extendHyper(i, distance-pipeLength(i),
         -connectivityMx(i,node), isHyper, isHyperOld)
    –Endif
•Endfor
•return
•end
```

This function requires a starting pipe which is hyperbolic at the old time level (that is `isHyperbolicOld(pipe)=TRUE`), and a direction and distance to spread. It expands through the network, branching as necessary, and calculates the required hyperbolic pipes at the new time level.

6.3.3 Data transfer between meshes

Transferring data between the meshes in branched networks is in most ways the same as for linear networks. There is a minor difference in the transfer from parabolic to hyperbolic meshes as it is not physically realistic to continuously interpolate the flow between adjacent pipes at a junction. At a junction the flow is not continuous, but rather sums to zero. Instead, the van Leer style interpolation (5.5.5) can be used for the flow, with the slope limited with respect to all adjacent pipes. It is important to take correct account of the pipe orientation since the flow has an associated direction, unlike pressure and temperature. From the results of chapter 5 the recommended strategy is,

- Linearly interpolate between nodes to get the hyperbolic pressure and temperature,
- Use van Leer style interpolation within pipes to get the hyperbolic flow,
- Average the flow in a pipe to get the parabolic flow,
- Take the parabolic temperature and pressure pointwise from the hyperbolic solution at the nodes.

6.3.4 Results

To conclude this chapter we present some illustrative results calculated by the iterative method of Section 5.4.1.

The test problem is similar to that of Section 6.1, a pipeline consisting of six 10,000m pipes and an offtake, this time two 5,000m pipes. Initially the network is at steady state, with a pressure of 80×10^5Pa at the left node to 75×10^5Pa at the right node and zero flow at the off-take. At $t = 0$ a demand of 1200kg/m^2/s is suddenly imposed at the off-take and Figure 6.3.1 shows how the rarefaction wave travels along the network.

Each row is a snapshot in time at $t =$ 1s, 11s ... showing the pressure (Pa) on the left and the flow (kg/m^2/s) on the right. The parabolic pressure is shown at the nodes, while the parabolic flow is shown in the pipes. The hyperbolic solution is represented by broader lines both for pressure and flow.

The hyperbolic domain propagates down the off-take and spreads left and right at the T-junction. Eventually the disturbance at the off-take decays to such an extent that the pipe nearest the off-take reverts to parabolic.

Notice that the hyperbolic region does not propagate symmetrically away from the T-junction, but instead follows the stagnation point, where the gas velocity is low. This confirms the findings of Chapter 4 which suggested that the parabolic model is not suitable for small gas velocities.

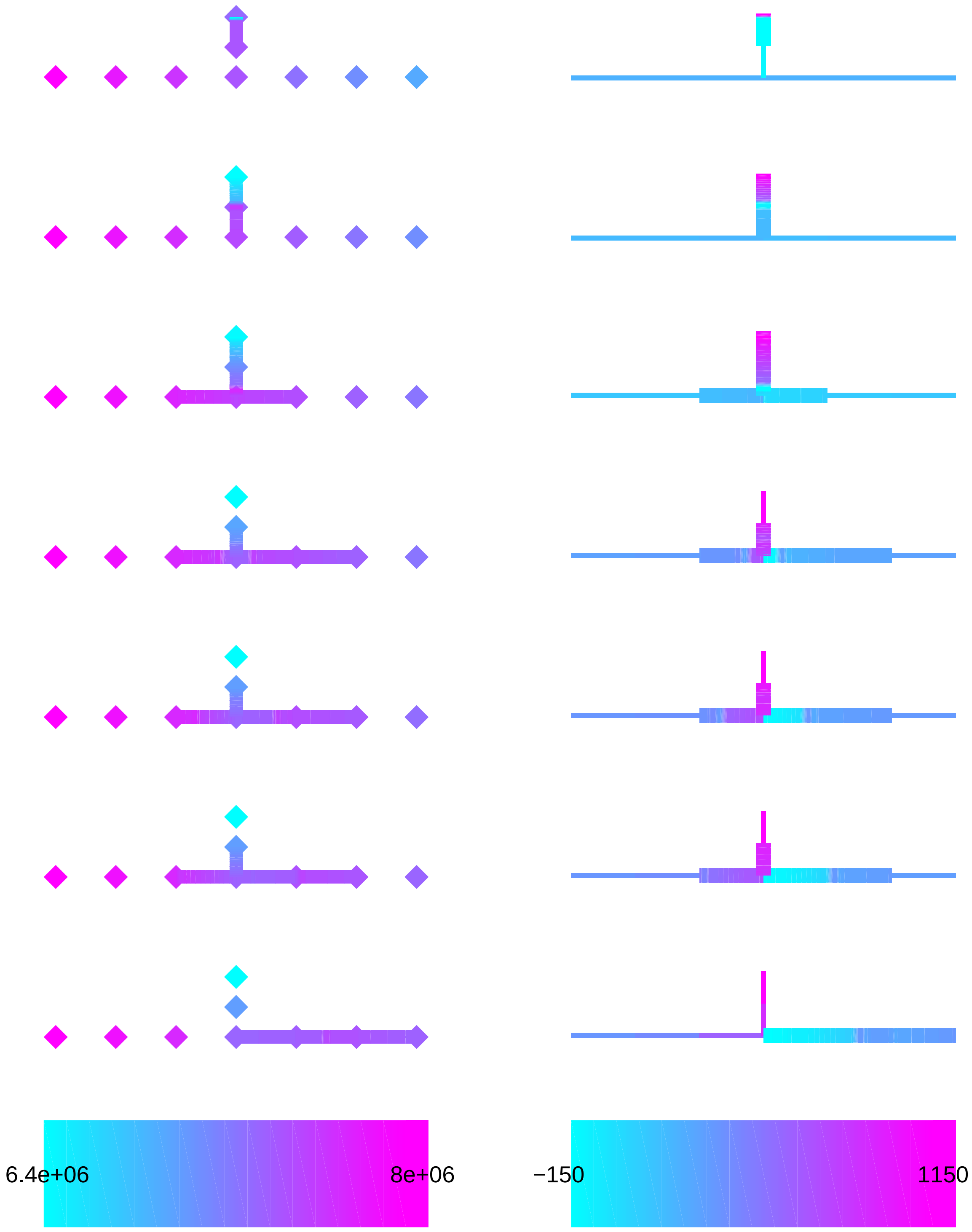


Figure 6.3.1: A solution calculated by the noniterative method of Section 5.4.1. Each row is a snapshot at 1, 11, 21 etc seconds, the left hand figure shows pressure (Pa), the right hand figure shows flow (kg/m^2/s).

Chapter 7

Conclusions and practical considerations

The aim of this thesis was to investigate the feasibility of combining two different numerical methods on a single gas network. Two practical algorithms have been proposed which fulfill this objective.

The hyperbolic solver for branched networks is simple to apply on a pipe-by-pipe basis, there is no need for an algorithm to identify linear sections of the network which should be solved as single pipelines. The particular treatment of the boundaries ensures that whenever there is a junction of only two pipes, the junction is treated in an identical fashion to an internal point in a pipe.

It is important to BG that any new algorithm is compatible with their existing software. Both algorithms can make use of FALCON (the BG code) as the parabolic solver with little modification. Although the network is theoretically split into hyperbolic and parabolic domains, in practice it is simpler to calculate the parabolic solution on the entire network to avoid the complication of restructuring the connectivity matrix. The noniterative method does this already; the iterative method will simply discard the parabolic solution in pipes which it has labelled as hyperbolic.

The results of Chapter 5 suggested that the iterative solver is more accurate and efficient than the noniterative solver. Furthermore, the noniterative solver suffers from instantaneous transmission of transients to all parts of the network, which not only is unphysical, but also disturbs the parabolic domains which the transient has not yet reached and results in the hyperbolic domain expanding faster than is necessary.

Some of the remaining questions include, how to include real gas effects (see Section 2.4.1) and what is the optimum ratio of the hyperbolic and parabolic mesh sizes.

The work of this thesis has other applications. One of the most suitable is the solution of river networks, which are modelled by a similar system of equations and have a similar geometry.

Appendix A

Notation

Appendix Notation

Symbol	Variable	Typical Value	Units
γ	ratio of specific heats	7/5	
δ_j	jump in jth wave between cells		
Δt	time-step, parabolic region		s
Δx	spatial mesh size, parabolic region		m
δt	time-step, hyperbolic region		s
δx	spatial mesh size, hyperbolic region		m
ϵ	pipe roughness		m
θ	parameter for implicit numerical method		
λ	mesh ratio	.1-10	
λ_j	jth eigenvalue		
μ	frictional constant	$2f/D \simeq 0.004$	m^{-1}
μ	viscosity		Pa s
$\hat{\mu}$	Joule-Thompson coefficient	0 (ideal)	K/Pa
ρ	density, hyperbolic region	50	$\mathrm{kg/m^3}$
ϱ	density, parabolic region	50	$\mathrm{kg/m^3}$
τ	frictional wall stress		$\mathrm{N/m^2}$
Ω	heat transfer term		W/m
ω	relaxation coefficient in iteration		
$\mathcal{H}$	projection operator from the p. to h. mesh		
$\mathcal{P}$	projection operator from the h. to p. mesh		

Appendix Notation

Symbol	Variable	Typical Value	Units
A	linearised matrix from Euler equations		
A	pipe area	1.1	m^2
C	pipe circumference	3.6	m
C	connectivity matrix		
D	pipe diameter	1.2	m
E	energy density	1.75×10^7	$\mathrm{J/m}^3$
F	energy flux	$(E+p)u$	
J	Jacobian matrix		
M	Mach number	0.05	
M	mass		kg
M	number of pipes at a junction		
MW	molecular weight	17	kg/kmol
P	pressure, parabolic region	70×10^5	Pa
Pr	Prandtl number		
Q	flow/unit area, parabolic region	500	$\mathrm{kg/s/m}^2$
R	pipe radius	0.6	m
R	reduced gas constant	490	J/kg/K
R^*	gas constant	8314	J/kmol/K
Re	Reynolds number	10^7	
St	Stanton number	0.00125	
T	temperature	280	K
V	volume		m^3
X_E	cell-edge value of X		

Appendix Notation

Symbol	Variable	Typical Value	Units
a	speed of sound	340	m/s
$\bar{a}$	roe-averaged sound speed		
a, b	EOS coefficients		
c	specific heat capacity of soil	1840	J/kg/K
c_v	specific heat capacity at constant V	1220	J/kg/K
c_{ij}	entries in C	$\pm 1, 0$	
e	specific internal energy		J/kg
f	friction factor	0.002	
h	pipe depth	1	m
h	specific enthalpy	$(E+p)/\rho$	
$\bar{h}$	roe-averaged enthalpy		
k	soil conductivity	0.52	J/s/m/K
m	number of h. time-steps in each p. step	600	
m	pipe number		
n	number of moles		
p	pressure, hyperbolic region	70×10^5	Pa
q	flow/unit area, hyperbolic region	500	kg/s/m^2
$\mathbf{r}_j$	jth eigenvector		
s	specific entropy		J/kg/K
t	time		s
u	gas speed		m/s
$\mathbf{u}$	conserved quantities	$[\rho, q, E]$	
$\bar{u}$	roe-averaged velocity		
$\mathbf{w}$	Riemann similarity solution		
$\hat{\mathbf{w}}$	Approximate Riemann similarity solution		
z	compressibility constant		

Bibliography

[BB73] J.P. Boris and D.L. Book. Flux-corrected transport. I. SHASTA, a fluid transport algorithm that works. *J. Comp. Phys.*, 11:38–69, 1973.

[Ber85] M.J. Berger. Adaptive mesh refinement for hyperbolic equations. In *Lectures in applied mathematics, volume 22*, pages 31–40. American Mathematical Society, 1985.

[Ber87] M.J. Berger. On conservation at grid interfaces. *SIAM J. Numer. Anal.*, 24(5):967–984, October 1987.

[Cho76] A.J. Chorin. Random choice solution of hyperbolic systems. *J.Comp.Phys.*, 22:517, 1976.

[Cho77] A.J. Chorin. Random choice methods, with applications to reaching gas flow. *J.Comp.Phys.*, 25:253, 1977.

[Com93] European Commision. Annual energy review DG xviii, 1993.

[CS74] C.K. Chu and A. Sereny. Boundary conditions in finite difference codes. *J. Comp. Phys.*, 15:476–491, 1974.

[CW84] P. Colella and C. Woodward. The piecewise parabolic method (PPM) for gas-dynamical simulations. *J. Comp. Phys.*, 54(1):174–201, 1984.

[CW94] G. Chesshire and Henshaw W.D. A scheme for conservative interpolation on overlapping grids. *SIAM J. Sci. Comput.*, 15(4):819–845, July 1994.

Bibliography

[Duf96] I. Duff. A review of frontal methods for solving linear-systems. *Computer physics communications*, 97(1-2):45–52, 1996.

[Dun95] D.B. Duncan. Difference approximations of acoustic and elastic wave equations. In *Proceedings of 1995 Workshop on numerical methods for wave propagation phenomena*, Manchester, U.K., 1995.

[Ein88] B. Einfeldt. On Godunov-type methods for gas dynamics. *SIAM J. Numer. Anal.*, 25(2):294–318, April 1988.

[Emm90] S. Emmerson. *Modelling of transient dynamics of gas flow in pipes.* PhD thesis, Reading University, 1990.

[Eng96] G. Engl. The modelling and numerical simulation of gas flow networks. *Numer. Math.*, 72:349–366, 1996.

[Fox89] J.A. Fox. *Transient flows in ducts and sewers.* Ellis Horwood, 1989.

[GF81] M.H. Goldwater and A.E. Fincham. Modelling of gas supply systems. In H. Nicholson, editor, *Modelling dynamical systems.* Peregrinus, 1981.

[GL88] J.B. Goodman and R.J. LeVeque. A geometric approach to high resolution TVD schemes. *SIAM J. Numer. Anal.*, 25(2):268–284, April 1988.

[God59] S.K. Godunov. *Mat. Sb.*, page 271, 47 1959.

[Gre88] D.W. Green, editor. *Perry's chemical engineering handbook.* McGraw-Hill, 1988.

[Har83a] A. Harten. High resolution schemes for hyperbolic conservation laws. *J. Comp. Phys.*, 49:357–393, 1983.

[Har83b] A. Harten. On the symmetric form of a systems of conservation-laws with entropy. *J. Comp. Phys.*, 49(1):151–164, 1983.

Bibliography

[Har84] A. Harten. On a class of high resolution total-variation-stable finite-difference schemes. *SIAM J. Numer. Anal.*, 21(1):1–23, February 1984.

[Hil94] D.J.M. Hill. Modelling temperature variations in transient simulations of high pressure gas networks. Internal report ERS, R.5098, British Gas, Gas Reseach Centre, Loughborough, Leics., July 1994.

[Hin91] E.J. Hinch. *Perturbation methodss.* CUP, 1991.

[HLvL83] A. Harten, P.D. Lax, and B. van Leer. On upstream differencing and Godunov-type schemes for hyperbolic conservation laws. *SIAM Review*, 25(1):35–61, January 1983.

[HO87] A. Harten and S. Osher. Uniformly high-order accuracte nonoscillatory schemes. i. *SIAM J. Numer. Anal.*, 24(2):279–309, 1987.

[Hol77] M. Holt. *Numerical methods in fluid dynamics.* Springer, 1977.

[HT84] T.J.R. Hughes and T.E. Tezduyar. Finite-element methods for 1st-order hyperbolic systems with particular emphasis on the compressible Euler equations. *Computer methods in applied mechanics and engineering*, 45(1-3):217–284, 1984.

[Huy95] H.T. Huynh. Accurate upwind methods for the Euler equations. *SIAM J. Numer. Anal.*, 32(5):1565–1619, October 1995.

[IS72] R.I. Issa and D.B. Spalding. Unsteady one-dimensional compressible frictional flow with heat transfer. *Journal Mechanical Engineering Science*, 14(6):365–369, 1972.

[Kev90] J. Kevorkian. *Partial differential equations: analytical solution techniques.* Wadsworth and Brooks, 1990.

[Kiu94] T. Kiuchi. An implicit method for transient gas flows in pipe networks. *Int. J. Heat and Fluid Flow*, 15(5):378 – 383, October 1994.

Bibliography

[Kre68] H.-O. Kreiss. Stability theory for difference approximations of mixed initial boundary value problems. i. *Math. Comp.*, 22:703–714, 1968.

[Lan91] E. Lang. Gas flow in pipelines following a rupture computed by a spectral method. *Journal of applied mathematics and physics (ZAMP)*, 42, March 1991.

[Lax73] P.D. Lax. *Hyperbolic sustems of conservation laws and the mathematical theory of shock waves.* SIAM, 1973.

[Leo91] B.P. Leonard. The ultimate conservative difference scheme applied to unsteady one-dimensional advection. *Computer methods in applied mechanics and engineering*, 88:17–74, 1991.

[Lev82] R.J. Leveque. Large time step shock-capturing techniques for scalar conservation laws. *SIAM J. Numer. Anal.*, 19(6), December 1982.

[LeV90] R.J. LeVeque. *Numerical methods for conservation laws.* Birkhauser-Verlag, 1990.

[LeV95] R.J. LeVeque. *CLAWPACK - Conservation LAW PACKage, a set of Fortran routines for solving conservation laws.* Department of Applied Mathematics, Washington University, http://www.hensa.ac.uk/netlib/pdes/claw/index.html, 1995.

[LL89] L.D. Landau and E.M. Lifshitz. *Fluid Mechanics*, volume 6 of *Course of Theoretical Physics.* Pergamon Press, 2nd edition, 1989.

[LW60] P.D. Lax and B. Wendroff. Systems of conservation laws. *Comm. Pure and App. Maths*, 13, 1960.

[Mal] J. Mallinson. British gas. Personal communication.

[MM94] K.W. Morton and D.F. Mayers. *Numerical solution of partial differential equations - an introduction.* CUP, 1994.

[Mor71] K.W. Morton. Stability and convergence in fluid flow problems. *Proceedings of the Royal Society of London A*, 323:237–253, 1971.

[OC84] S. Osher and S. Chakravarthy. High resolution schemes and the entropy condition. *SIAM J. Numer. Anal.*, 21(5):955–984, October 1984.

[OR70] J.M. Ortega and W.C. Rheinboldt. *Iterative solution of nonlinear equations in several variables.* Academic Press, 1970.

[Osh84] S. Osher. Riemann solvers, the entropy condition, and difference approximations. *SIAM J. Numer. Anal.*, 21(2):217–235, April 1984.

[OT83] H. Ockendon and A.B. Tayler. *Inviscid fluid flows.* Springer, 1983.

[Per92] B. Perthame. Second-order Bolzmann schemes for compressible Euler equations in one and two space dimensions. *SIAM J. Numer. Anal.*, 29(1):1–19, February 1992.

[PES94] E. Pärt-Enander and B. Sjögreen. Conservative and nonconservative interpolation between overlapping grids for finite volume solutions of hyperbolic problems. *Computers and Fluids*, 23(3):551–574, 1994.

[Pul80] D.I. Pullin. Direct simulation methods for compressible inviscid ideal-gas flow. *Journal of computational physics*, 34:231–244, 1980.

[RB76] R.F.Warming and R.M. Beam. Upwind second-order difference schemes and applications in aerodynamic flows. *AIAA Journal*, 14(9):1241–1249, September 1976.

[RB82] P.L. Roe and M.J. Baines. Algorithms for advection and shock problems. In H. Viviand, editor, *Proc. 4th GAMM Conf. on Numerical methods in fluid mechanics.* Vieweg, 1982.

[Rei81] R.D. Reitz. One-dimensional compressible gas dynamics calculations using the Boltzmann equation. *J. Comp. Phys.*, 42:108–123, 1981.

Bibliography

[Roe81] P.L. Roe. Approximate Riemann solvers, parameter vectors and difference schemes. *J. Comp. Phys.*, 43:357–372, 1981.

[Roe85] P.L. Roe. Some contributions to the modelling of discontinuous flows. In *Lectures in applied mathematics, volume 22*, pages 163–193. American Mathematical Society, 1985.

[RT90] R.B. Risebro and A. Tveiro. A front tracking method for conservation laws in one dimension. Technical Report 8, University of Oslo, June 1990. ISBN 82-553-0708-7.

[Sch90] H.A. Schwartz. *Gesammelete Mathematische Abhandlungen*, volume 2. Springer Verlag, 1890.

[Sod85] G.A. Sod. *Numerical methods in fluid dynamics.* CUP, 1985.

[Sta80] G. Starius. On composite mesh difference methods for hyperbolic differential equations. *Numer. Math.*, 35(241-255), 1980.

[Ste] G. Stewart. Edinburgh Petroleum Services Ltd. Personal communication.

[Sto69] M.A. Stoner. Analysis and control of unsteady flows in natural gas piping systems. *Journal of Basic Engineering*, pages 331 – 340, September 1969.

[Swe84] P.K. Sweby. High resolution schemes using flux limiters for hyperbolic conservation laws. *SIAM J. Numer. Anal.*, 21(5):995–1010, October 1984.

[Swe85] P.K. Sweby. High resolution TVD schemes using flux limiters. In *Lectures in applied mathematics, volume 22*, pages 289–309. American Mathematical Society, 1985.

[TT87] A.R.D. Thorley and C.H. Tiley. Unsteady and transient flow of compressible fluids in pipelines - a review of theoretical and some experimental studies. *Heat and fluid flow*, 8(1):3–14, March 1987.

[TZ95] T. Tang and H.T. Zheng. Error-bounds for fractional step methods for conservation-laws with source terms. *SIAM J. Numer. Anal.*, 32(1):110–127, February 1995.

[Upt96] D. Upton. *Waves of fortune.* Wiley, 1996.

[vDR83] J.K. van Deen and S.R. Reintsema. Modelling of high-pressure gas transmission lines. *Appl. Math. Modelling*, 7:268–273, August 1983.

[vL74] B. van Leer. Towards the ultimate conservative difference scheme. ii. monotonicity and conservation combined in a second-order scheme. *J. Comp. Phys.*, 14:361–370, 1974.

[vL77] B. van Leer. Towards the ultimate conservative difference scheme. iv. a new approach to numerical convection. *J. Comp. Phys.*, 23:276–299, 1977.

[vL79] B. van Leer. Towards the ultimate conservative difference scheme. v. a second-order sequel to godunov's method. *J. Comp. Phys.*, 32:101–136, 1979.

[YWH85] H.C. Yee, R.F. Warming, and A. Harten. Application of TVD schemes for the Euler equations of gas dynamics. In *Lectures in applied mathematics, volume 22*, pages 357–377. American Mathematical Society, 1985.

[Zem68] M.W. Zemansky. *Heat and thermodynamics.* McGraw-Hill, 1968.