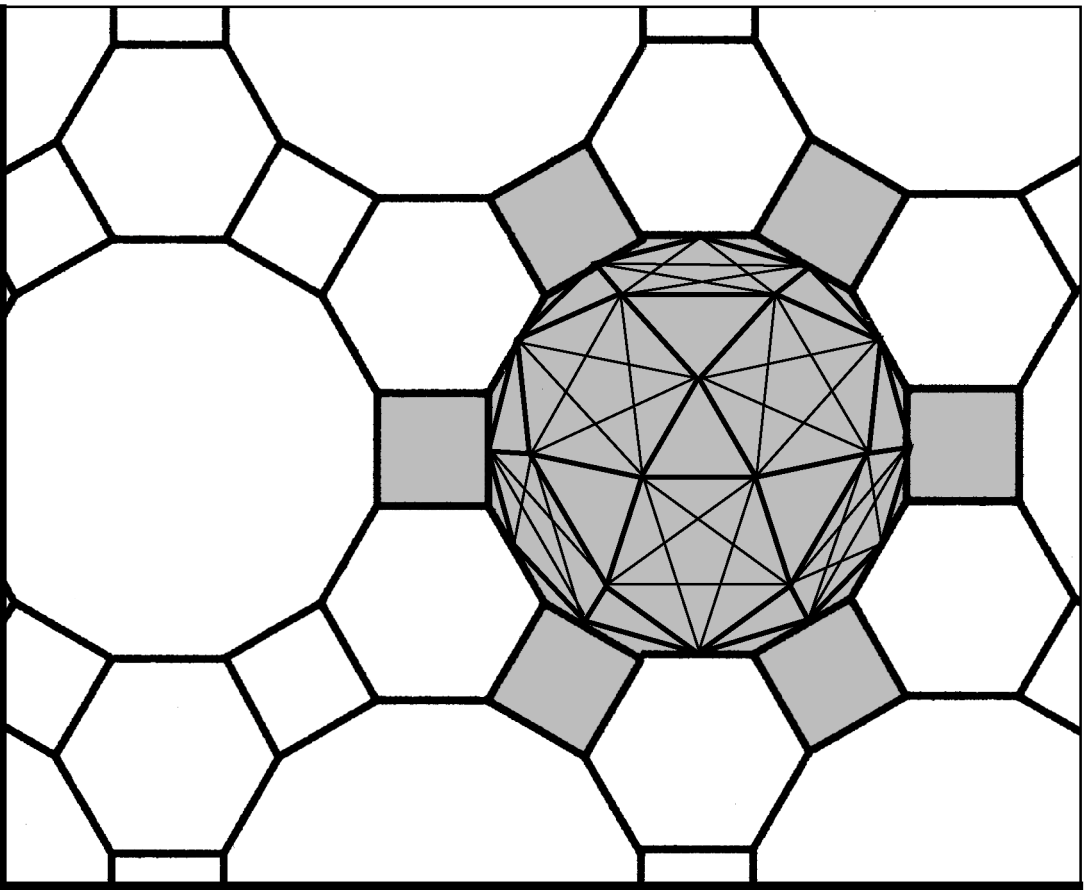
Workbook 32



Numerical Initial Value Problems

HELM®

HELM: Helping Engineers Learn Mathematics

http://helm.lboro.ac.uk

About the HELM Project

HELM (Helping Engineers Learn Mathematics) materials were the outcome of a three-year curriculum development project undertaken by a consortium of five English universities led by Loughborough University, funded by the Higher Education Funding Council for England under the Fund for the Development of Teaching and Learning for the period October 2002 – September 2005, with additional transferability funding October 2005 – September 2006.

HELM aims to enhance the mathematical education of engineering undergraduates through flexible learning resources, mainly these Workbooks.

HELM learning resources were produced primarily by teams of writers at six universities: Hull, Loughborough, Manchester, Newcastle, Reading, Sunderland.

HELM gratefully acknowledges the valuable support of colleagues at the following universities and colleges involved in the critical reading, trialling, enhancement and revision of the learning materials:
Aston, Bournemouth & Poole College, Cambridge, City, Glamorgan, Glasgow, Glasgow Caledonian, Glenrothes Institute of Applied Technology, Harper Adams, Hertfordshire, Leicester, Liverpool, London Metropolitan, Moray College, Northumbria, Nottingham, Nottingham Trent, Oxford Brookes, Plymouth, Portsmouth, Queens Belfast, Robert Gordon, Royal Forest of Dean College, Salford, Sligo Institute of Technology, Southampton, Southampton Institute, Surrey, Teesside, Ulster, University of Wales Institute Cardiff, West Kingsway College (London), West Notts College.

HELM Contacts:
Post: HELM, Mathematics Education Centre, Loughborough University, Loughborough, LE11 3TU.
Email: helm@lboro.ac.uk *Web:* http://helm.lboro.ac.uk

HELM Workbooks List

1	Basic Algebra	26	Functions of a Complex Variable
2	Basic Functions	27	Multiple Integration
3	Equations, Inequalities & Partial Fractions	28	Differential Vector Calculus
4	Trigonometry	29	Integral Vector Calculus
5	Functions and Modelling	30	Introduction to Numerical Methods
6	Exponential and Logarithmic Functions	31	Numerical Methods of Approximation
7	Matrices	32	Numerical Initial Value Problems
8	Matrix Solution of Equations	33	Numerical Boundary Value Problems
9	Vectors	34	Modelling Motion
10	Complex Numbers	35	Sets and Probability
11	Differentiation	36	Descriptive Statistics
12	Applications of Differentiation	37	Discrete Probability Distributions
13	Integration	38	Continuous Probability Distributions
14	Applications of Integration 1	39	The Normal Distribution
15	Applications of Integration 2	40	Sampling Distributions and Estimation
16	Sequences and Series	41	Hypothesis Testing
17	Conics and Polar Coordinates	42	Goodness of Fit and Contingency Tables
18	Functions of Several Variables	43	Regression and Correlation
19	Differential Equations	44	Analysis of Variance
20	Laplace Transforms	45	Non-parametric Statistics
21	z-Transforms	46	Reliability and Quality Control
22	Eigenvalues and Eigenvectors	47	Mathematics and Physics Miscellany
23	Fourier Series	48	Engineering Case Study
24	Fourier Transforms	49	Student's Guide
25	Partial Differential Equations	50	Tutor's Guide

© Copyright Loughborough University, 2015

Production of this 2015 edition, containing corrections and minor revisions of the 2008 edition, was funded by the **sigma** Network.

sigma Σ
network for excellence in mathematics and statistics support

Contents 32

Numerical Initial Value Problems

Learning outcomes

In this Workbook you will learn about numerical methods for approximating solutions relating to a certain type of application area. Specifically you will see methods that approximate solutions to differential equations.

Initial Value Problems 32.1

Introduction

Many engineering applications describe the evolution of some process with time. In order to define such an application we require two distinct pieces of information: we need to know what the process is and also when or where the application started.

In this Section we begin with a discussion of some of these so-called **initial value problems**. Then we look at two numerical methods that can be used to approximate solutions of certain initial value problems. These two methods will serve as useful instances of a fairly general class of methods which we will describe in Section 32.2.

Prerequisites

Before starting this Section you should ...

- revise the trapezium method for approximating integrals in HELM 31.2
- review the material concerning approximations to derivatives in HELM 31.3

Learning Outcomes

On completion you should be able to ...

- recognise an initial value problem
- implement the Euler and trapezium method to approximate the solutions of certain initial value problems

1. Initial value problems

In HELM 19.4 we saw the following initial value problem which arises from Newton's law of cooling

$$\frac{d\theta}{dt} = -k(\theta - \theta_s), \qquad \theta(0) = \theta_0.$$

Here $\theta = \theta(t)$ is the temperature of some liquid at time t, θ_0 is the initial temperature at $t = 0$ and θ_s is the surrounding temperature. The constant of proportion k has units s^{-1} and depends on the properties of the liquid.

This initial value problem has two parts: the differential equation $\frac{d\theta}{dt} = -k(\theta - \theta_s)$, which models the physical process, and the initial condition $\theta(0) = \theta_0$.

Key Point 1

An initial value problem may be made up of two components

1. A mathematical model of the *process*, stated in the form of a differential equation.
2. An initial value, given at some value of the independent variable.

It should be noted that there are applications in which initial value problems do not model processes that are time dependent, but we will not dwell on this fact here.

The initial value problem above is such that we can write down an *exact* or **analytic** solution (it is $\theta(t) = \theta_s + (\theta_0 - \theta_s)e^{-kt}$) but there are many applications where it is impossible or undesirable to seek such a solution. The aim of this Section is to begin to describe numerical methods that can be used to find **approximate** solutions of initial value problems.

Rather than using the application-specific notation given above involving θ we will consider the following initial value problem in this Section. We seek $y = y(t)$ (or an approximation to it) that satisfies the differential equation

$$\frac{dy}{dt} = f(t, y), \qquad (t > 0)$$

and which is subject to the **initial condition**

$$y(0) = y_0,$$

a known quantity.

Some of the examples we will consider will be such that an analytic solution is readily available, and this fact can be used as a check on the accuracy of the numerical methods that follow.

2. Numerical solutions

We suppose that the initial value problem

$$\frac{dy}{dt} = f(t, y) \qquad\qquad y(0) = y_0$$

is such that we are unable (or unwilling) to seek a solution analytically (that is, by hand) and that we prefer to use a computer to approximate y instead. We begin by asking what we expect a numerical solution to look like.

Numerical solutions to initial value problems discussed in this Workbook will be in the form of a sequence of numbers approximating $y(t)$ at a sequence of values of t. The simplest methods choose the t-values to be equally spaced, and we will stick to these methods. We denote the common distance between consecutive t-values as h.

Key Point 2

A numerical approximation to the initial value problem

$$\frac{dy}{dt} = f(t, y), \qquad\qquad y(0) = y_0$$

is a sequence of numbers y_0, y_1, y_2, y_3, ...
The value y_0 will be exact, because it is defined by the initial condition.
For $n \geq 1$, y_n is the approximation to the exact value $y(t)$ at $t = nh$.

In Figure 1 the exact solution $y(t)$ is shown as a thick curve and approximations to $y(nh)$ are shown as crosses.

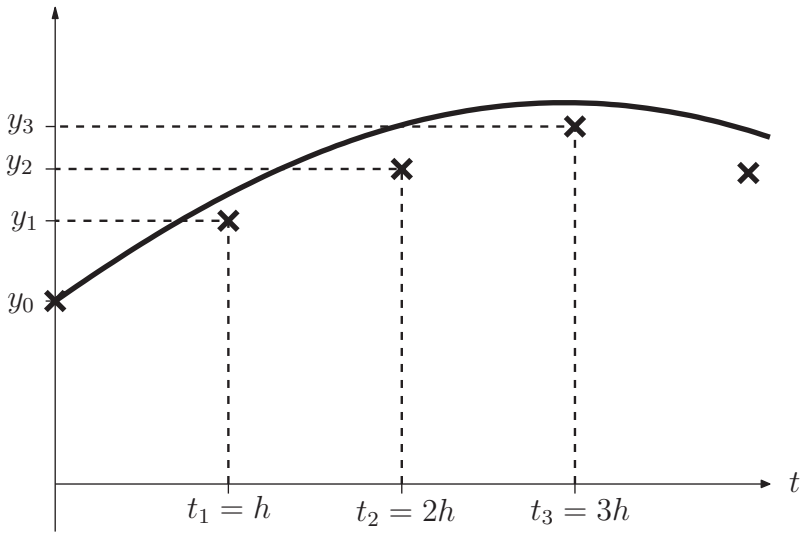


Figure 1

The general idea is to take the given initial condition y_0 and then use it together with what we know about the physical process (from the differential equation) to obtain an approximation y_1 to $y(h)$. We will have then carried out the first **time step**.

Then we use the differential equation to obtain y_2, an approximation to $y(2h)$. Thus the second time step is completed.

And so on, at the n^{th} time step we find y_n, an approximation to $y(nh)$.

Key Point 3

A **time step** is the procedure carried out to move a numerical approximation one increment forward in time.

The way in which we choose to "use the differential equation" will define a particular numerical method, and some ways are better than others. We begin by looking at the simplest method.

3. An explicit method

Guided by the fact that we only seek approximations to $y(t)$ at t-values that are a distance h apart we could use a **forward difference formula** to approximate the derivative in the differential equation. This leads to

$$\frac{y(t+h)-y(t)}{h} \approx f(t,y)$$

and we use this as the inspiration for the numerical method

$$y_{n+1} - y_n = hf(nh, y_n)$$

For clarity we denote $f(nh, y_n)$ as f_n. The procedure for implementing the method (called Euler's method - pronounced "Oil-er's method" - is summarised in the following Key Point.

Key Point 4

Euler's method for approximating the solution of

$$\frac{dy}{dt} = f(t, y), \qquad y(0) = y_0$$

is as follows. We choose a time step h, then

$$\begin{aligned} y(h) &\approx y_1 = y_0 + hf(0, y_0) \\ y(2h) &\approx y_2 = y_1 + hf(h, y_1) \\ y(3h) &\approx y_3 = y_2 + hf(2h, y_2) \\ y(4h) &\approx y_4 = y_3 + hf(3h, y_3) \\ &\vdots \end{aligned}$$

In general, $y(nh)$ is approximated by $\boxed{y_n = y_{n-1} + hf_{n-1}}$.

This is called an **explicit** method, but the reason why will be clearer in a page or two when we encounter an **implicit** method. First we look at an Example.

Example 1

Suppose that $y = y(t)$ is the solution to the initial value problem

$$\frac{dy}{dt} = -1/(t + y)^2, \qquad y(0) = 0.9$$

Carry out two time steps of Euler's method with a step size of $h = 0.125$ so as to obtain approximations to $y(0.125)$ and $y(0.25)$.

Solution

In general, Euler's method may be written $y_{n+1} = y_n + hf_n$ and here $f(t, y) = -1/(t + y)^2$.
For the first time step we require $f_0 = f(0, y_0) = f(0, 0.9) = -1.23457$ and therefore

$$y_1 = y_0 + hf_0 = 0.9 + 0.125 \times (-1.23457) = 0.745679$$

For the second time step we require $f_1 = f(h, y_1) = f(0.125, 0.745679) = -1.31912$ and therefore

$$y_2 = y_1 + hf_1 = 0.745679 + 0.125 \times (-1.31912) = 0.580789$$

We conclude that

$$y(0.125) \approx 0.745679 \qquad y(0.25) \approx 0.580789$$

where these approximations are given to 6 decimal places.

The simple, repetitive nature of this process makes it ideal for computational implementation, but this next exercise can be carried out by hand.

Suppose that $y = y(t)$ is the solution to the initial value problem

$$\frac{dy}{dt} = -y^2, \qquad y(0) = 0.5$$

Carry out two time steps of Euler's method with a step size of $h = 0.01$ so as to obtain approximations to $y(0.01)$ and $y(0.02)$.

Your solution

Answer

For the first time step we require $f_0 = f(0, y_0) = f(0, 0.5) = -(0.5)^2 = -0.25$ and therefore

$$y_1 = y_0 + hf_0 = 0.5 + 0.01 \times (-0.25) = 0.4975$$

For the second time step we require $f_1 = f(h, y_1) = f(0.01, 0.4975) = -(0.4975)^2 = -0.24751$ and therefore

$$y_2 = y_1 + hf_1 = 0.4975 + 0.01 \times (-0.24751) = 0.495025$$

We conclude that

$$y(0.01) \approx 0.497500 \qquad y(0.02) \approx 0.495025$$ to six decimal places.

The following Task involves the so-called **logistic approximation** that may be used in modelling population dynamics.

Given the logistic population dynamic model

$$\frac{dy}{dt} = 2y(1-y), \qquad y(0) = 1.2$$

carry out two time steps of Euler's method with a step size of $h = 0.125$ to obtain approximations to $y(0.125)$ and $y(0.25)$.

Your solution

Answer

For the first time step we require $f_0 = f(0, y_0) = f(0, 1.2) = 2 \times 1.2(1-1.2) = -0.48$ and therefore

$$\begin{aligned} y_1 &= y_0 + hf_0 \\ &= 1.2 + 0.125 \times (-0.48) \\ &= 1.14 \end{aligned}$$

For the second time step we require $f_1 = f(h, y_1) = f(0.125, 1.14) = -0.3192$ and therefore

$$\begin{aligned} y_2 &= y_1 + hf_1 \\ &= 1.14 + 0.125 \times (-0.3192) \\ &= 1.1001 \end{aligned}$$

We conclude that

$$\begin{aligned} y(0.125) &\approx 1.14 \\ y(0.25) &\approx 1.1001 \end{aligned}$$

The following initial value problem models the population of the United Kingdom, suppose that

$$\frac{dP}{dt} = 2.5 \times 10^{-3} P, \qquad P(0) = 58.043$$

where P is the population in millions, t is measured in years and $t = 0$ corresponds to the year 1996.

(a) Show that Euler's method applied to this initial value problem leads to

$$P_n = (1 + 2.5 \times 10^{-3} h)^n \times 58.043$$

where P_n is the approximation to $P(nh)$.

(b) Use a time step of h equal to 6 months to approximate the predicted population for the year 2050.

Your solution

(a)

Answer

In general $P_{n+1} = P_n + hf_n$ where, in this case, $f(h, P_n) = 2.5 \times 10^{-3} P_n$ hence

$$P_{n+1} = P_n + 2.5 \times 10^{-3} h P_n \qquad \text{and so} \qquad P_{n+1} = (1 + 2.5 \times 10^{-3} h) P_n$$

But P_n will have come from the previous time step ($P_n = (1 + 2.5 \times 10^{-3} h) P_{n-1}$) and P_{n-1} will have come from the time step before that ($P_{n-1} = (1 + 2.5 \times 10^{-3} h) P_{n-2}$). Repeatedly applying this observation leads to

$$P_n = (1 + 2.5 \times 10^{-3} h)^n \times 58.043$$

since $P_0 = P(0) = 58.043$.

Your solution

(b)

Answer
For a time step of 6 months we take $h = \frac{1}{2}$ (in years) and we require 108 time steps to cover the 54 years from 1996 to 2050. Hence

UK population (in millions) in 2050 $\approx P(54) \approx P_{108} = (1 + 2.5 \times 10^{-3} \times \frac{1}{2})^{108} \times 58.043 = 66.427$

where this approximation is given to 3 decimal places.

Accuracy of Euler's method

There are two issues to consider when concerning ourselves with the accuracy of our results.

1. How accurately does the differential equation model the physical process?
2. How accurately does the numerical method approximate the solution of the differential equation?

Our aim here is to address only the second of these two questions.
Let us now consider an example with a known solution and consider just how accurate Euler's method is. Suppose that

$$\frac{dy}{dt} = y \qquad y(0) = 1.$$

We know that the solution to this problem is $y(t) = e^t$, and we now compare exact values with the values given by Euler's method. For the sake of argument, let us consider approximations to $y(t)$ at $t = 1$. The exact value is $y(1) = 2.718282$ to 6 decimal places. The following table shows results to 6 decimal places obtained on a spreadsheet program for a selection of choices of h.

h	Euler approximation to $y(1) = 2.718282$	Difference between exact value and Euler approximation
0.2	$y_5 = 2.488320$	0.229962
0.1	$y_{10} = 2.593742$	0.124539
0.05	$y_{20} = 2.653298$	0.064984
0.025	$y_{40} = 2.685064$	0.033218
0.0125	$y_{80} = 2.701485$	0.016797

Notice that the smaller h is, the more time steps we have to take to get to $t = 1$. In the table above each successive implementation of Euler's method halves h. Interestingly, the error halves (approximately) as h halves. This observation verifies something we will see in Section 32.2, that is that the error in Euler's method is (approximately) proportional to the step size h. This sort of behaviour is called **first-order**, and the reason for this name will become clear later.

Key Point 5

Euler's method is first order. In other words, the error it incurs is approximately proportional to h.

4. An implicit method

Another approach that can be used to address the initial value problem

$$\frac{dy}{dt} = f(t, y), \qquad y(0) = y_0$$

is to consider integrating the differential equation

$$\frac{dy}{dt} = f(t, y)$$

from $t = nh$ to $t = nh + h$. This leads to

$$\Big[y(t)\Big]_{t=nh}^{t=nh+h} = \int_{nh}^{(n+1)h} f(t, y)\, dt$$

that is,

$$y(nh + h) - y(nh) = \int_{nh}^{(n+1)h} f(t, y)\, dt$$

and the problem now becomes one of approximating the integral on the right-hand side.

If we approximate the integral using the simple trapezium rule and replace the terms by their approximations we obtain the numerical method

$$y_{n+1} - y_n = \tfrac{1}{2}h\,(f_n + f_{n+1})$$

The procedure for time stepping with this method is much the same as that used for Euler's method, but with one difference. Let us imagine applying the method, we are given y_0 as the initial condition and now aim to find y_1 from

$$\begin{aligned} y_1 &= y_0 + \tfrac{h}{2}\,(f_0 + f_1) \\ &= y_0 + \tfrac{h}{2}\,\{f(0, y_0) + f(h, y_1)\} \end{aligned}$$

And here is the problem: the unknown y_1 appears on both sides of the equation. We cannot, in general, find an explicit expression for y_1 and for this reason the numerical method is called an **implicit** method.

In practice the particular form of f may allow us to find y_1 fairly simply, but in general we have to approximate y_1 for example by using the bisection method, or Newton-Raphson. (Another approach that can be used involves what is called a **predictor-corrector** method, in other words, a "guess and improve" method, and we will discuss this again later in this Workbook.)

And then, of course, we encounter the problem again in the second time step, when calculating y_2. And again for y_3 and so on. There is, in general, a genuine cost in implementing implicit methods, but they are popular because they have desirable properties, as we will see later in this Workbook.

Key Point 6

The trapezium method for approximating the solution of

$$\frac{dy}{dt} = f(t, y) \qquad\qquad y(0) = y_0$$

is as follows. We choose a time step h, then

$$\begin{aligned} y(h) &\approx y_1 = y_0 + \tfrac{1}{2}\, h\left(f(0, y_0) + f(h, y_1)\right) \\ y(2h) &\approx y_2 = y_1 + \tfrac{1}{2}\, h\left(f(h, y_1) + f(2h, y_2)\right) \\ y(3h) &\approx y_3 = y_2 + \tfrac{1}{2}\, h\left(f(2h, y_2) + f(3h, y_3)\right) \\ y(4h) &\approx y_4 = y_3 + \tfrac{1}{2}\, h\left(f(3h, y_3) + f(4h, y_4)\right) \\ &\vdots \end{aligned}$$

In general, $y(nh)$ is approximated by $\boxed{y_n = y_{n-1} + \tfrac{1}{2}\, h\left(f_{n-1} + f_n\right)}$

In Example 2 the implicit nature of the method is not a problem because y does not appear on the right-hand side of the differential equation. In other words, $f = f(t)$.

Example 2

Suppose that $y = y(t)$ is the solution to the initial value problem

$$\frac{dy}{dt} = 1/(t+1), \qquad y(0) = 1$$

Carry out two time steps of the trapezium method with a step size of $h = 0.2$ so as to obtain approximations to $y(0.2)$ and $y(0.4)$.

Solution

For the first time step we require $f_0 = f(0) = 1$ and $f_1 = f(0.2) = 0.833333$ and therefore

$$y_1 = y_0 + \tfrac{1}{2}h(f_0 + f_1) = 1 + 0.1 \times 1.833333 = 1.183333$$

For the second time step we also require $f_2 = f(2h) = f(0.4) = 0.714286$ and therefore

$$y_2 = y_1 + \tfrac{1}{2}h(f_1 + f_2) = 1.183333 + 0.1 \times 1.547619 = 1.338095$$

We conclude that

$$y(0.1) \approx 1.183333 \qquad y(0.2) \approx 1.338095$$

Example 3 has f dependent on y, so the implicit nature of the trapezium method could be a problem. However in this case the way in which f depends on y is simple enough for us to be able to rearrange for an explicit expression for y_{n+1}.

Example 3

Suppose that $y = y(t)$ is the solution to the initial value problem

$$\frac{dy}{dt} = 1/(t^2+1) - 2y, \qquad y(0) = 2$$

Carry out two time steps of the trapezium method with a step size of $h = 0.1$ so as to obtain approximations to $y(0.1)$ and $y(0.2)$.

Solution

The trapezium method is $y_{n+1} = y_n + \frac{h}{2}(f_n + f_{n+1})$ and in this case y_{n+1} will appear on both sides because f depends on y. We have

$$\begin{aligned} y_{n+1} &= y_n + \frac{h}{2}\left\{\left(\frac{1}{t_n^2+1} - 2y_n\right) + \left(\frac{1}{t_{n+1}^2+1} - 2y_{n+1}\right)\right\} \\ &= y_n + \frac{h}{2}\left\{g(t_n) - 2y_n + g(t_{n+1}) - 2y_{n+1}\right\} \end{aligned}$$

where $g(t) \equiv \dfrac{1}{(t^2+1)}$ which is the part of f that depends on t. On rearranging to get all y_{n+1} terms on the left, we get

$$(1+h)y_{n+1} = y_n + \tfrac{1}{2}h\left\{g(t_n) - 2y_n + g(t_{n+1})\right\}$$

In this case $h = 0.1$.

For the first time step we require $g(0) = 1$ and $g(0.1) = 0.990099$ and therefore

$$1.1y_1 = 2 + 0.05\,(1 - 2 \times 2 + 0.990099)$$

Hence $y_1 = 1.726823$, to six decimal places.

For the second time step we also require $g(2h) = g(0.2) = 0.961538$ and therefore

$$1.1y_2 = 1.726823 + 0.05\,(0.990099 - 2 \times 1.726823 + 0.961538)$$

Hence $y_2 = 1.501566$. We conclude that $y(0.1) \approx 1.726823$ and $y(0.2) \approx 1.501566$ to 6 d.p.

Suppose that $y = y(t)$ is the solution to the initial value problem

$$\frac{dy}{dt} = t - y, \qquad y(0) = 2$$

Carry out two time steps of the trapezium method with a step size of $h = 0.125$ so as to obtain approximations to $y(0.125)$ and $y(0.25)$.

Your solution

Answer

The trapezium method is $y_{n+1} = y_n + \frac{h}{2}(f_n + f_{n+1})$ and in this case y_{n+1} will appear on both sides because f depends on y. However, we can rearrange for y_{n+1} to give

$$1.0625y_{n+1} = y_n + \tfrac{1}{2}h\left(g(t_n) - y_n + g(t_{n+1})\right)$$

where $g(t) = t$ is the part of f that depends on t.
For the first time step we require $g(0) = 0$ and $g(0.125) = 0.125$ and therefore

$$1.0625y_1 = 2 + 0.0625\left(0 - 2 + 0.125\right)$$

Hence $y_1 = 1.772059$ to 6 d.p.
For the second time step we also require $g(2h) = g(0.25) = 0.25$ and therefore

$$1.0625y_2 = 1.772059 + 0.0625\left(0.125 - 1.772059 + 0.25\right)$$

Hence $y_2 = 1.58564$, to 6 d.p.

Example 4

The current i in a simple circuit involving a resistor of resistance R and an inductance loop of inductance L with applied voltage E satisfies the differential equation

$$L\frac{di}{dt} + Ri = E$$

Consider the case where $L = 1$, $R = 100$ and $E = 1000$. Given that $i(0) = 0$ use a value of $h = 0.001$ in implementation of the trapezium method to approximate the current i at times $t = 0.001$ and $t = 0.002$.

Solution

The current i satisfies

$$\frac{di}{dt} = 1000 - 100i$$

and the trapezium approximation to this is

$$i_{n+1} - i_n = \frac{h}{2}(2000 - 100i_{n+1} - 100i_n)$$

Rearranging this for i_{n+1} gives

$$i_{n+1} = 0.904762i_n + 0.952381$$

It follows that

$$\begin{aligned} i(0.001) &\approx 0.904762 \times 0 + 0.952381 = 0.952381 \\ i(0.002) &\approx 0.904762 \times 0.952381 + 0.952381 = 1.814059 \end{aligned}$$

where these approximations are given to 6 decimal places.

Accuracy of the trapezium method

Let us now consider an example with a known solution and consider just how accurate the trapezium method is. Suppose that we look at the same test problem we considered when looking at Euler's method

$$\frac{dy}{dt} = y, \qquad y(0) = 1.$$

We know that the solution to this problem is $y(t) = e^t$, and we now compare exact values with the values given by the trapezium method. For the sake of argument, let us consider approximations to $y(t)$ at $t = 1$. The exact value is $y(1) = 2.718282$ to 6 decimal places. The following table shows results to 6 decimal places obtained on a spreadsheet program for a selection of choices of h.

h	Trapezium approximation to $y(1) = 2.718282$	Difference between exact value and trapezium approximation
0.2	$y_5 = 2.727413$	0.009131
0.1	$y_{10} = 2.720551$	0.002270
0.05	$y_{20} = 2.718848$	0.000567
0.025	$y_{40} = 2.718423$	0.000142
0.0125	$y_{80} = 2.718317$	0.000035

Notice that each time h is reduced by a factor of $\frac{1}{2}$, the error reduces by a factor of (approximately) $\frac{1}{4}$. This observation verifies something we will see in Section 32.2, that is that the error in the trapezium approximation is (approximately) proportional to h^2. This sort of behaviour is called **second-order**.

Key Point 7

The trapezium approximation is second order. In other words, the error it incurs is approximately proportional to h^2.

Exercises

1. Suppose that $y = y(t)$ is the solution to the initial value problem

$$\frac{dy}{dt} = t + y \qquad y(0) = 3$$

Carry out two time steps of Euler's method with a step size of $h = 0.05$ so as to obtain approximations to $y(0.05)$ and $y(0.1)$.

2. Suppose that $y = y(t)$ is the solution to the initial value problem

$$\frac{dy}{dt} = 1/(t^2 + 1) \qquad y(0) = 2$$

Carry out two time steps of the trapezium method with a step size of $h = 0.1$ so as to obtain approximations to $y(0.1)$ and $y(0.2)$.

3. Suppose that $y = y(t)$ is the solution to the initial value problem

$$\frac{dy}{dt} = t^2 - y \qquad y(0) = 1.5$$

Carry out two time steps of the trapezium method with a step size of $h = 0.125$ so as to obtain approximations to $y(0.125)$ and $y(0.25)$.

4. The current i in a simple circuit involving a resistor of resistance R, an inductance loop of inductance L with applied voltage E satisfies the differential equation

$$L\frac{di}{dt} + Ri = E$$

Consider the case where $L = 1.5$, $R = 120$ and $E = 600$. Given that $i(0) = 0$ use a value of $h = 0.0025$ in implementation of the trapezium method to approximate the current i at times $t = 0.0025$ and $t = 0.005$.

Answers

1. For the first time step we require $f_0 = f(0, y_0) = f(0, 3) = 3$ and therefore

$$\begin{aligned} y_1 &= y_0 + hf_0 \\ &= 3 + 0.05 \times 3 \\ &= 3.15 \end{aligned}$$

For the second time step we require $f_1 = f(h, y_1) = f(0.05, 3.15) = 3.2$ and therefore

$$\begin{aligned} y_2 &= y_1 + hf_1 \\ &= 3.15 + 0.05 \times 3.2 \\ &= 3.31 \end{aligned}$$

We conclude that

$$\begin{aligned} y(0.05) &\approx 3.15 \\ y(0.1) &\approx 3.31 \end{aligned}$$

2. For the first time step we require $f_0 = f(0) = 1$ and $f_1 = f(0.1) = 0.990099$ and therefore

$$\begin{aligned} y_1 &= y_0 + \tfrac{1}{2}h(f_0 + f_1) \\ &= 2 + 0.05 \times 1.990099 \\ &= 2.099505 \end{aligned}$$

For the second time step we also require $f_2 = f(2h) = f(0.2) = 0.961538$ and therefore

$$\begin{aligned} y_2 &= y_1 + \tfrac{1}{2}h(f_1 + f_2) \\ &= 2.099505 + 0.05 \times 1.951637 \\ &= 2.197087 \end{aligned}$$

We conclude that

$$\begin{aligned} y(0.05) &\approx 2.099505 \\ y(0.1) &\approx 2.197087 \end{aligned}$$

to six decimal places.

Answers

3. The trapezium method is $y_{n+1} = y_n + \frac{h}{2}(f_n + f_{n+1})$ and in this case y_{n+1} will appear on both sides because f depends on y. However, we can rearrange for y_{n+1} to give

$$1.0625y_{n+1} = y_n + \tfrac{1}{2}h\left\{g(t_n) - y_n + g(t_{n+1})\right\}$$

where $g(t) = t^2$ is the part of f that depends on t.
For the first time step we require $g(0) = 0$ and $g(0.125) = 0.015625$ and therefore

$$1.0625y_1 = 1.5 + 0.0625\left(0 - 1.5 + 0.015625\right)$$

Hence $y_1 = 1.324449$.
For the second time step we also require $g(2h) = g(0.25) = 0.0625$ and therefore

$$1.0625y_2 = 1.324449 + 0.0625\left(0.015625 - 1.324449 + 0.0625\right)$$

Hence $y_2 = 1.173227$.

4. Dividing through by $L = 1.5$ we find that the current i satisfies

$$\frac{di}{dt} = 400 - 80i$$

and the trapezium approximation to this is

$$i_{n+1} - i_n = \frac{h}{2}(800 - 80i_{n+1} - 80i_n)$$

Rearranging this for i_{n+1} gives

$$i_{n+1} = 0.818182i_n + 0.909091$$

It follows that

$$\begin{aligned} i(0.0025) &\approx 0.818182 \times 0 + 0.909091 = 0.909091 \\ i(0.005) &\approx 0.818182 \times 0.909091 + 0.909091 = 1.652893 \end{aligned}$$

Linear Multistep Methods

32.2

Introduction

In the previous Section we saw two methods (Euler and trapezium) for approximating the solutions of certain initial value problems. In this Section we will see that those two methods are special cases of a more general collection of techniques called linear multistep methods. Techniques for determining the properties of these methods will be presented.

Another class of approximations, called Runge-Kutta methods, will also be discussed briefly. These are not linear multistep methods, but the two are sometimes used in conjunction.

Prerequisites

Before starting this Section you should ...

- review Section 32.1

Learning Outcomes

On completion you should be able to ...

- implement linear multistep methods to carry out time steps of numerical methods
- evaluate the zero stability of linear multistep methods
- establish the order of linear multistep methods
- implement a Runge-Kutta method

1. General linear multistep methods

Euler's method and the trapezium method are both special cases of a wider class of so-called **linear multistep** methods. The following Key Point gives the most general situation that we will look at.

Key Point 8

The general k-step linear multistep method is given by

$$\alpha_k y_{n+k} + \cdots + \alpha_1 y_{n+1} + \alpha_0 y_n = h\left(\beta_k f_{n+k} + \cdots + \beta_1 f_{n+1} + \beta_0 f_n\right)$$

or equivalently

$$\sum_{j=0}^{k} \alpha_j \; y_{n+j} = h \sum_{j=0}^{k} \beta_j \; f_{n+j}.$$

It is always the case that $\alpha_k \neq 0$. Also, at least one of α_0 and β_0 will be non-zero.

A linear multistep method is defined by the choice of the quantities

$$k, \quad \alpha_0, \quad \alpha_1, \quad \ldots, \quad \alpha_k, \quad \beta_0, \quad \beta_1, \quad \ldots, \quad \beta_k$$

- If $\beta_k = 0$ the method is called **explicit**. (Because at each step, when we are trying to find the newest y_{n+k}, there is no appearance of this unknown on the right-hand side.)
- If $\beta_k \neq 0$ the method is called **implicit**. (Because y_{n+k} now appears on both sides of the equation (on the right-hand side it appears through $f_{n+k} = f((n+k)h, y_{n+k})$, and we cannot, in general, rearrange to get an explicit formula for y_{n+k}.)

The next Example shows one such choice.

Example 5

Write down the linear multistep scheme defined by the choices $k = 1$, $\alpha_0 = -1$, $\alpha_1 = 1$, $\beta_0 = \beta_1 = \frac{1}{2}$.

Solution

Here $k = 1$ so that

$$\alpha_1 y_{n+1} + \alpha_0 y_n = h(\beta_1 f_{n+1} + \beta_0 f_n)$$

and substituting the values in for the four coefficients gives

$$y_{n+1} - y_n = h\left(\tfrac{1}{2} f_{n+1} + \tfrac{1}{2} f_n\right)$$

which, as we know, is the trapezium method.

Write down the linear multistep scheme defined by the choices $k = 1$, $\alpha_0 = -1$, $\alpha_1 = 1$, $\beta_0 = 1$ and $\beta_1 = 0$.

Your solution

Answer

Here $k = 1$ and we have

$$\alpha_1 y_{n+1} + \alpha_0 y_n = h(\beta_1 f_{n+1} + \beta_0 f_n)$$

and substituting the values in for the four coefficients gives

$$y_{n+1} - y_n = h f_n$$

which, as we know, is Euler's method.

Write down the linear multistep scheme defined by the choices $k = 2$, $\alpha_0 = 0$, $\alpha_1 = -1$, $\alpha_2 = 1$, $\beta_2 = 0$, $\beta_1 = \frac{3}{2}$ and $\beta_0 = -\frac{1}{2}$.

Your solution

Answer

Here $k = 2$ (so we are looking at a 2-step scheme) and we have

$$\alpha_2 y_{n+1} + \alpha_1 y_{n+1} + \alpha_0 y_n = h(\beta_2 f_{n+2} + \beta_1 f_{n+1} + \beta_0 f_n)$$

Substituting the values in for the six coefficients gives

$$y_{n+2} - y_{n+1} = \frac{h}{2}(3f_{n+1} - f_n)$$

which is an example of a scheme that is explicit (because $\beta_k = \beta_2$ is zero).

In the preceding Section we saw several examples implementing the Euler and trapezium methods.

The next Example deals with the explicit 2-step that was the subject of the Task above.

Example 6

A numerical scheme has been used to approximate the solution of

$$\frac{dy}{dt} = t + y, \qquad y(0) = 3$$

and has produced the following estimates, to 6 decimal places,

$$y(0.4) \approx 4.509822, \quad y(0.45) \approx 4.755313$$

Now use the 2-step, explicit linear multistep scheme

$$y_{n+2} - y_{n+1} = h\,(1.5f_{n+1} - 0.5f_n)$$

to approximate $y(0.5)$.

Solution

Evidently the value $h = 0.05$ will serve our purposes and we seek $y_{10} \approx y(0.5)$. The values we will need to use in our implementation of the 2-step scheme are $y_9 = 4.755313$ and

$$f_9 = f(0.45, y_9) = 5.205313 \qquad f_8 = f(0.4, y_8) = 4.909822$$

to 6 decimal places since $f(t, y) = t + y$. It follows that

$$\begin{aligned} y_{10} &= y_9 + 0.05 \times (1.5f_9 - 0.5f_8) \\ &= 5.022966 \end{aligned}$$

And we conclude that $y(0.5) \approx 5.022966$, where this approximation has been given to 6 decimal places.

Notice that in this implementation of a 2-step method we needed to use the values of the *two* y values preceding the one currently being sought. Both y_8 and y_9 were used in finding y_{10}.

Similarly, a k-step method will use, in general, k previous y values at each time step.

This means that there is an issue to be resolved in implementing methods that are 2- or higher-step, because when we start we are only given *one* starting value y_0. This issue will be dealt with towards the end of this Section. The following exercise involves a 2-step method, but (like the example above) it does not encounter the difficulty relating to starting values as it assumes that the numerical procedure is already underway.

A numerical scheme has been used to approximate the solution of

$$\frac{dy}{dt} = t/y \qquad y(0) = -2$$

and has produced the following estimates, to 6 decimal places,

$$y(0.24) \approx -2.013162, \quad y(0.26) \approx -2.015546$$

Now use the 2-step, explicit linear multistep scheme

$$y_{n+2} - \tfrac{1}{2}y_{n+1} - \tfrac{1}{2}y_n = \tfrac{3}{2}hf_{n+1}$$

to approximate $y(0.28)$.

Your solution

Answer

Evidently the value $h = 0.02$ will serve our purposes and we seek $y_{14} \approx y(0.28)$. The values we will need to use in our implementation of the 2-step scheme are $y_{13} = -2.015546$, $y_{12} = -2.013162$ and

$$f_{13} = f(0.26, y_{13}) = -0.128997$$

to 6 decimal places since $f(t, y) = t/y$. It follows that

$$\begin{aligned} y_{14} &= \tfrac{1}{2}y_{13} + \tfrac{1}{2}y_{12} + 0.02 \times \tfrac{3}{2}f_{13} \\ &= -2.018224 \end{aligned}$$

And we conclude that $y(0.28) \approx -2.018224$, to 6 decimal places.

Zero stability

We now begin to classify linear multistep methods. Some choices of the coefficients give rise to schemes that work well, and some do not. One property that is required if we are to obtain reliable approximations is that the scheme be **zero stable**. A scheme that is zero stable will not produce approximations which grow unrealistically with t.

We define the **first characteristic polynomial**

$$\rho(z) = \alpha_0 + \alpha_1 z + \alpha_2 z^2 + \ldots \alpha_k z^k$$

where the α_i are the coefficients of the linear multistep method as defined in Key Point 8 (page 21). This polynomial appears in the definition of zero stability given in the following Key Point.

Key Point 10

The linear multistep scheme

$$\sum_{j=0}^{k} \alpha_j \; y_{n+j} = h \sum_{j=0}^{k} \beta_j \; f_{n+j}.$$

is said to be **zero stable** if the zeros of the first characteristic polynomial are such that

1. none is larger than 1 in magnitude
2. any zero equal to 1 in magnitude is simple (that is, not repeated)

The **second characteristic polynomial** is defined in terms of the coefficients on the right-hand side (the β_j), but its use is beyond the scope of this Workbook.

Example 7

Find the roots of the first characteristic polynomial for each of the examples below and determine whether or not the method is zero stable.

(a) $y_{n+1} - y_n = hf_n$

(b) $y_{n+1} - 2y_n = hf_n$

(c) $y_{n+2} + 3y_{n+1} - 4y_n = h\left(2f_{n+2} + f_{n+1} + 2f_n\right)$

(d) $y_{n+2} - y_{n+1} = \frac{3}{2}hf_{n+1}$

(e) $y_{n+2} - 2y_{n+1} + y_n = h(f_{n+2} - f_n)$

(f) $y_{n+2} + 2y_{n+1} + 5y_n = h\left(f_{n+2} - f_{n+1} + 2f_n\right)$

Solution

(a) In this case $\rho(z) = z - 1$ and the single zero of ρ is $z = 1$. This is a simple (that is, not repeated) root with magnitude equal to 1, so the method is zero stable.

(b) $\rho(z) = z - 2$ which has one zero, $z = 2$. This has magnitude $2 > 1$ and therefore the method is not zero stable.

(c) $\rho(z) = z^2 + 3z - 4 = (z - 1)(z + 4)$. One root is $z = -4$ which has magnitude greater than 1 and the method is therefore not zero stable.

Solution (contd.)

(d) Here $\alpha_2 = 1$, $\alpha_1 = -1$ and $\alpha_0 = 0$, therefore

$$\rho(z) = z^2 - z = z(z-1)$$

which has two zeros, $z = 0$ and $z = 1$. These both have magnitude less than or equal to 1 and there is no repeated zero with magnitude equal to 1, so the method is zero stable.

(e) $\rho(z) = z^2 - 2z + 1 = (z-1)^2$. Here $z = 1$ is not a simple root, it is repeated and, since it has magnitude equal to 1, the method is not zero stable.

(f) $\rho(z) = z^2 + 2z + 5$ and the roots of $\rho(z) = 0$ can be found from the quadratic formula. In this case the roots are complex and are equal to Zero-stability requires that the absolute values have magnitude less than or equal to 1. Consequently we conclude that the method is not zero stable.

Find the roots of the first characteristic polynomial for the linear multistep scheme

$$y_{n+2} - 2y_{n+1} + y_n = h\left(f_{n+2} + 2f_{n+1} + f_n\right)$$

and hence determine whether or not the scheme is zero stable.

Your solution

Answer

The first characteristic polynomial is

$$\rho(z) = \alpha_2 z^2 + \alpha_1 z + \alpha_0 = z^2 - 2z + 1$$

and the roots of $\rho(z) = 0$ are both equal to 1. In the case of roots that are equal, zero-stability requires that the absolute value has magnitude less than 1. Consequently we conclude that the method is not zero stable.

At this stage, the notion of zero stability is rather abstract, so let us try using a **zero unstable** method and see what happens. We consider the simple test problem

$$\frac{dy}{dt} = -y, \qquad y(0) = 1$$

which we know to have analytic solution $y(t) = e^{-t}$, a quantity which decays with increasing t. Implementing the zero unstable scheme

$$y_{n+1} - 2y_n = hf_n$$

on a spreadsheet package with $h = 0.05$ gives the following results

n	$t = nh$	$y_n \approx y(nh)$
0	0.00	1.00000
1	0.05	1.95000
2	0.10	3.80250
3	0.15	7.41488
4	0.20	14.45901
5	0.25	28.19506
6	0.30	54.98037
7	0.35	107.21172
8	0.40	209.06286
9	0.45	407.67258
10	0.50	794.96153
11	0.55	1550.17499
12	0.60	3022.84122
13	0.65	5894.54039
14	0.70	11494.35376
15	0.75	22413.98982

where 5 decimal places have been given for y_n. The dramatic growth in the values of y_n is due to the zero instability of the method. (There are in fact other things than zero instability wrong with the scheme $y_{n+1} - 2y_n = hf_n$, but it is the zero instability that is causing the large numbers.)

Consistency and order

A scheme that is zero stable will produce approximations that do not grow in size in a way that is not present in the exact, analytic solution. Zero stability is a required property, but it is not enough on its own. There remains the issue of whether the approximations are close to the exact values.
The **truncation error** of the general linear multistep method is a measure of how well the differential equation and the numerical method agree with each other. It is defined by

$$\tau_j = \frac{1}{\beta}\left(\frac{c_0}{h}y(jh) + c_1 y'(jh) + c_2 h y''(jh) + c_3 h^2 y'''(jh) + \dots\right) = \frac{1}{\beta h}\sum_{p=0}^{\infty} c_p h^p y^{(p)}(jh)$$

where $\beta = \sum \beta_j$ is a normalising factor.
It is the first few terms in this expression that will matter most in what follows, and it helps us that there are formulae for the coefficients which appear

$$c_0 = \sum \alpha_j, \quad c_1 = \sum (j\alpha_j - \beta_j), \quad c_2 = \sum\left(\frac{j^2}{2}\alpha_j - j\beta_j\right), \quad c_3 = \sum\left(\frac{j^3}{3!}\alpha_j - \frac{j^2}{2}\beta_j\right)$$

and so on, the general formula for $p \geq 2$ is $c_p = \sum\left(\frac{j^p}{(p)!}\alpha_j - \frac{j^{p-1}}{(p-1)!}\beta_j\right)$.

Recall that the truncation error is intended to be a measure of how well the differential equation and its approximation agree with each other. We say that the numerical method is **consistent** with the differential equation if τ_j tends to zero as $h \to 0$. The following Key Point says this in other words.

Key Point 11

The linear multistep scheme is said to be **consistent** if $c_0 = 0$ and $c_1 = 0$.

Example 8

Show that Euler's method $(y_{n+1} = y_n + hf_n)$ is consistent.

Solution

In this case $\alpha_1 = 1$, $\alpha_0 = -1$, $\beta_1 = 0$ and $\beta_0 = 1$. It follows that

$$c_0 = \sum \alpha_j = 1 - 1 = 0 \qquad \text{and} \qquad c_1 = \sum j\alpha_j - \beta_j = 1\alpha_1 - (\beta_0 + \beta_1) = 1 - (1 + 0) = 0$$

and therefore Euler's method is consistent.

Show that the trapezium method $(y_{n+1} = y_n + \frac{h}{2}(f_{n+1} + f_n))$ is consistent.

Your solution

Answer

In this case $\alpha_1 = 1$, $\alpha_0 = -1$, $\beta_1 = \frac{1}{2}$ and $\beta_0 = \frac{1}{2}$. It follows that

$$c_0 = \sum \alpha_j = 1 - 1 = 0 \qquad \text{and} \qquad c_1 = \sum j\alpha_j - \beta_j = 1\alpha_1 - (\beta_0 + \beta_1) = 1 - (\tfrac{1}{2} + \tfrac{1}{2}) = 0$$

and therefore the trapezium method is consistent.

Determine the consistency (or otherwise) of the following 2-step linear multistep schemes

(a) $y_{n+2} - 2y_{n+1} + y_n = h(f_{n+2} - f_n)$

(b) $y_{n+2} - y_{n+1} = h(f_{n+1} - 2f_n)$

(c) $y_{n+2} - y_{n+1} = h(2f_{n+2} - f_{n+1})$

Your solution

Answer

(a) $c_0 = \alpha_2 + \alpha_1 + \alpha_0 = 1 - 2 + 1 = 0$, $c_1 = 2\alpha_2 + 1 \times \alpha_1 + 0 \times \alpha_0 - (\beta_2 + \beta_1 + \beta_0) = 2(1) + 1(-2) + 0 - (1 - 1) = 0$. Therefore the method is consistent.

(b) $c_0 = 1 - 1 + 0 = 0$, $c_1 = 2 - 1 - (1 - 2) = 2$ so the method is inconsistent.

(c) This method is consistent, because $c_0 = 1 - 1 = 0$ and $c_1 = 2 - 1 - (2 - 1) = 0$.

(Notice also that the first characteristic polynomial $\rho(z)$, defined on page 6 of this Section, evaluated at $z = 1$ is equal to $\alpha_0 + \alpha_1 + \cdots + \alpha_k = c_0$. It follows that a consistent scheme must always have $z = 1$ as one of the roots of its $\rho(z)$.)

Assuming that the method is consistent, the **order** of the scheme tells us how quickly the truncation error tends to zero as $h \to 0$. For example, if $c_0 = 0$, $c_1 = 0$, $c_2 = 0$ and $c_3 \neq 0$ then the first non-zero term in τ_j will be the one involving h^2 and the linear multistep method is called **second-order**. This means that if h is small then τ_j is dominated by the h^2 term (because the h^3 and subsequent terms will be tiny in comparison) and halving h will cause τ_j to decrease by a factor of approximately $\frac{1}{4}$. The decrease is only approximately known because the h^3 and other terms will have a small effect. We summarise the general situation in the following Key Point.

Key Point 12

A linear multistep method is said to be of **order** p if

$$c_0 = c_1 = c_2 = \cdots = c_p = 0 \qquad \text{and} \qquad c_{p+1} \neq 0$$

Combining the last two Key Points gives us another way of describing consistency: "A linear multistep method is consistent if it is at least first order".

Example 9

Find the order of

(a) Euler's method

(b) The trapezium method.

Solution

(a) We have already found that $c_0 = c_1 = 0$ so the first quantity to calculate is

$$c_2 = \sum \left(\frac{j^2}{2}\alpha_j - j\beta_j \right) = \tfrac{1}{2}\alpha_1 - \beta_1 = \tfrac{1}{2}$$

which is not zero and therefore Euler's method is of order 1. (Or, in other words, Euler's method is first order.)

(b) We have already found that $c_0 = c_1 = 0$ so the first quantity to calculate is

$$c_2 = \sum \left(\frac{j^2}{2}\alpha_j - j\beta_j \right) = \tfrac{1}{2}\alpha_1 - \beta_1 = \tfrac{1}{2} - \tfrac{1}{2} = 0$$

this is equal to zero, so we must calculate the next coefficient

$$c_3 = \sum \left(\frac{j^3}{3!}\alpha_j - \frac{j^2}{2}\beta_j \right) = \tfrac{1}{6}\alpha_1 - \tfrac{1}{2}\beta_1 = \tfrac{1}{6} - \tfrac{1}{4} = -\tfrac{1}{12}$$

which is not zero. Hence the trapezium method is of order 2 (that is, it is second order).

This finally explains some of the results we saw in the first Section of this Workbook. We saw that the errors incurred by the Euler and trapezium methods, for a particular test problem, were roughly proportional to h and h^2 respectively. This behaviour is dictated by the first non-zero term in the truncation error which is the one involving $c_2 h$ for Euler and the one involving $c_3 h^2$ for trapezium.

We now apply the method to another linear multistep scheme.

Example 10

Find the order of the 4-step, explicit linear multistep scheme

$$y_{n+4} - y_{n+3} = \frac{h}{24}\left(55f_{n+3} - 59f_{n+2} + 37f_{n+1} - 9f_n\right)$$

Solution

In the established notation we have $\alpha_4 = 1$, $\alpha_3 = -1$, $\alpha_2 = 0$, $\alpha_1 = 0$ and $\alpha_0 = 0$. The β terms similarly come from the coefficients on the right hand side (remembering the denominator of 24). Now

$$c_0 = \sum \alpha_j = 0 \qquad \text{and} \qquad c_1 = \sum j\alpha_j - \beta_j = 0$$

from which we conclude that the method is consistent.
We also find that

$$c_2 = \sum \tfrac{1}{2}j^2\alpha_j - j\beta_j = 0, \qquad c_3 = \sum \tfrac{1}{6}j^3\alpha_j - \tfrac{1}{2}j^2\beta_j = 0,$$

$$c_4 = \sum \tfrac{1}{24}j^4\alpha_j - \tfrac{1}{6}j^3\beta_j = 0 \qquad c_5 = \sum \tfrac{1}{120}j^5\alpha_j - \tfrac{1}{24}j^4\beta_j = 0.348611 \quad \text{to 6 d.p.}$$

(The exact value of c_5 is $\frac{251}{720}$.)

Because c_5 is the first non-zero coefficient we conclude that the method is of order 4.

So the scheme in Example 10 has the property that the truncation error will tend to zero proportional to h^4 (approximately) as $h \to 0$. This is a good thing, as it says that the error will decay to zero very quickly, when h is decreased.

Find the order of the 2-step linear multistep scheme

$$y_{n+2} - y_{n+1} = \frac{h}{12}\left(f_{n+2} + 8f_{n+1} - f_n\right)$$

Your solution

Answer

In the established notation we have $\alpha_2 = 1$, $\alpha_1 = -1$ and $\alpha_0 = 0$. Also $\beta_2 = \frac{5}{12}$, $\beta_1 = \frac{2}{3}$ and $\beta_0 = -\frac{1}{12}$. Now

$$c_0 = \sum \alpha_j = 1 - 1 + 0 = 0 \qquad \text{and} \qquad c_1 = \sum j\alpha_j - \beta_j = 2\alpha_2 + \alpha_1 - (\beta_2 + \beta_1 + \beta_0) = 0$$

from which we conclude that the method is consistent.
We also find that

$$\begin{aligned} c_2 &= \sum \tfrac{1}{2}j^2\alpha_j - j\beta_j = \tfrac{1}{2}(4\alpha_2 + \alpha_1) - (2\beta_2 + \beta_1) = 0 \\ c_3 &= \sum \tfrac{1}{6}j^3\alpha_j - \tfrac{1}{2}j^2\beta_j = \tfrac{1}{6}(8\alpha_2 + \alpha_1) - \tfrac{1}{2}(4\beta_2 + \beta_1) = 0 \\ c_4 &= \sum \tfrac{1}{24}j^4\alpha_j - \tfrac{1}{6}j^3\beta_j = \tfrac{1}{24}(16\alpha_2 + \alpha_1) - \tfrac{1}{6}(8\beta_2 + \beta_1) = -\tfrac{1}{24} \end{aligned}$$

so that the method is of order 3.

Convergence

The key result concerning linear multistep methods is given in the following Key Point.

Key Point 13

The numerical approximation to the initial value problem converges to the actual solution as $h \to 0$ if

1. the scheme is zero stable
2. the scheme is consistent

The proof of this result lies beyond the scope of this Workbook. It is worth pointing out that this is not the whole story. The convergence result is useful, but only deals with h as it tends to zero. In practice we use a finite, non-zero value of h and there are ways of determining how big an h it is possible to "get away with" for a particular linear multistep scheme applied to a particular initial value problem.

If, when implementing the methods described above, it is found that the numerical approximations behave in an unexpected way (for example, if the numbers are very large when they should not be, or if decreasing h does not seem to lead to results that converge) then one topic to look for in further reading is that of "absolute stability".

2. An example of a Runge-Kutta method

A full discussion of the so-called Runge-Kutta methods is not required here, but we do need to touch on them to resolve a remaining issue in the implementation of linear multistep schemes.

The problem with linear multistep methods is that a zero-stable, 1-step method can never be better than second order (you need not worry about why this is true, it was proved in the latter half of the last century by a man called Dahlquist). We have seen methods of higher order than 2, but they were all at least 2-step methods. And the problem with 2-step methods is that we need 2 starting values to implement them and we are only ever given 1 starting value: the initial condition $y(0)$.

One way out of this "Catch 22" is to use a **Runge-Kutta method** to generate the extra starting value(s) we need. Runge-Kutta methods are not linear multistep methods and do not suffer from the problem mentioned above. There is no such thing as a free lunch, of course, and Runge-Kutta methods are generally more expensive in effort to implement than linear multistep methods because of the number of evaluations of f required at each time step.

The following Key Point gives a statement of what is, perhaps, the most popular Runge-Kutta method (sometimes called "RK4").

Key Point 14

Runge Kutta method (RK4)

Consider the usual initial value problem

$$\frac{dy}{dt} = f(t,y), \qquad y(0) = y_0.$$

$$\begin{aligned}
\text{Calculate} \quad & K_1 = f(nh, y_n) \\
\text{then} \quad & K_2 = f((n+\tfrac{1}{2})h, y_n + \tfrac{1}{2}hK_1) \\
\text{then} \quad & K_3 = f((n+\tfrac{1}{2})h, y_n + \tfrac{1}{2}hK_2) \\
\text{then} \quad & K_4 = f((n+1)h, y_n + hK_3) \\
\text{finally} \quad & y_{n+1} = y_n + \frac{h}{6}(K_1 + 2K_2 + 2K_3 + K_4)
\end{aligned}$$

Notice that each calculation is explicit, all of the right-hand sides in the formulae in the Key Point above involve known quantities.

Example 11

Suppose that $y = y(t)$ is the solution to the initial value problem

$$\frac{dy}{dt} = \cos(y) \qquad y(0) = 3$$

Carry out one time step of the Runge-Kutta method RK4 with a step size of $h = 0.1$ so as to obtain an approximation to $y(0.1)$.

Solution

The iteration must be carried out in four stages. We start by calculating

$$K_1 = f(0, y_0) = f(0, 3) = -0.989992$$

a value we now use in finding

$$K_2 = f(\tfrac{1}{2}h, y_0 + \tfrac{1}{2}hK_1) = f(0.05, 2.950500) = -0.981797$$

This value K_2 is now used in our evaluation of

$$K_3 = f(\tfrac{1}{2}h, y_0 + \tfrac{1}{2}hK_2) = f(0.05, 2.950910) = -0.981875$$

which, in turn, is used in

$$K_4 = f(h, y_0 + hK_3) = f(0.1, 2.901812) = -0.971390$$

All four of these values are then used to complete the iteration

$$\begin{aligned} y_1 &= y_0 + \frac{h}{6}(K_1 + 2K_2 + 2K_3 + K_4) \\ &= 3 + \frac{0.1}{6}(-0.989992 + 2 \times -0.981797 + 2 \times -0.981875 - 0.971390) \\ &= 2.901855 \qquad \text{to 6 decimal places.} \end{aligned}$$

Suppose that $y = y(t)$ is the solution to the initial value problem

$$\frac{dy}{dt} = y(1-y) \qquad y(0) = 0.7$$

Carry out one time step of the Runge-Kutta method RK4 with a step size of $h = 0.1$ so as to obtain an approximation to $y(0.1)$.

Your solution

Answer

The time step must be carried out in four stages. We start by calculating

$$K_1 = f(0, y_0) = f(0, 0.7) = 0.210000$$

a value we now use in finding

$$K_2 = f(\tfrac{1}{2}h, y_0 + \tfrac{1}{2}hK_1) = f(0.05, 0.710500) = 0.205690$$

This value K_2 is now used in our evaluation of

$$K_3 = f(\tfrac{1}{2}h, y_0 + \tfrac{1}{2}hK_2) = f(0.05, 0.710284) = 0.205780$$

which, in turn, is used in

$$K_4 = f(h, y_0 + hK_3) = f(0.1, 0.720578) = 0.201345$$

All four of these values are then used to complete the time step

$$\begin{aligned} y_1 &= y_0 + \frac{h}{6}(K_1 + 2K_2 + 2K_3 + K_4) \\ &= 0.7 + \frac{0.1}{6}(0.210000 + 2 \times 0.205690 + 2 \times 0.205780 + 0.201345) \\ &= 0.720571 \quad \text{to 6 d.p.} \end{aligned}$$

Exercises

1. Assuming the notation established earlier, write down the linear multistep scheme corresponding to the choices $k = 2$, $\alpha_0 = 0$, $\alpha_1 = -1$, $\alpha_2 = 1$, $\beta_0 = \frac{-1}{12}$, $\beta_1 = \frac{2}{3}$, $\beta_2 = \frac{5}{12}$.

2. A numerical scheme has been used to approximate the solution of

   $$\frac{dy}{dt} = t^2 - y^2 \qquad y(0) = 2$$

   and has given the following estimates, to 6 decimal places,

   $$y(0.3) \approx 1.471433, \quad y(0.32) \approx 1.447892$$

   Now use the 2-step, explicit linear multistep scheme

   $$y_{n+2} - 1.6y_{n+1} + 0.6y_n = h\left(5f_{n+1} - 4.6f_n\right)$$

   to approximate $y(0.34)$.

3. Find the roots of the first characteristic polynomial for the linear multistep scheme

   $$5y_{n+2} + 3y_{n+1} - 2y_n = h\left(f_{n+2} + 2f_{n+1} + f_n\right)$$

   and hence determine whether or not the scheme is zero stable.

4. Find the order of the 2-step linear multistep scheme

   $$y_{n+2} + 2y_{n+1} - 3y_n = \frac{h}{10}\left(f_{n+2} + 16f_{n+1} + 17f_n\right)$$

   (Would you recommend using this method?)

5. Suppose that $y = y(t)$ is the solution to the initial value problem

   $$\frac{dy}{dt} = 1/y^2 \qquad y(0) = 2$$

   Carry out one time step of the Runge-Kutta method RK4 with a step size of $h = 0.4$ so as to obtain an approximation to $y(0.4)$.

Answers

1. $y_{n+2} - y_{n+1} = \dfrac{h}{12}\left(5f_{n+2} + 8f_{n+1} - f_n\right)$

2. Evidently the value $h = 0.02$ will serve our purposes and we seek $y_{17} \approx y(0.34)$. The values we will need to use in our implementation of the 2-step scheme are $y_{16} = 1.447892$, $y_{15} = 1.471433$ and $f_{16} = f(0.32, y_{16}) = -1.993991$ $\quad$ $f_{15} = f(0.3, y_{15}) = -2.075116$ since $f(t, y) = t^2 - y^2$. It follows that

$$y_{17} = 1.6y_{16} - 0.6y_{15} + 0.02 \times (5f_{16} - 4.6f_{15}) = 1.425279$$

And we conclude that $y(0.34) \approx 1.425279$, to 6 decimal places.

3. The first characteristic polynomial is $\rho(z) = \alpha_2 z^2 + \alpha_1 z + \alpha_0 = 5z^2 + 3z - 2$ and the roots of $\rho(z) = 0$ can be found from the quadratic formula. In this case the roots are real and distinct and are equal to 0.4 and -1. In the case of roots that are distinct zero-stability requires that the absolute values have magnitude less than or equal to 1 . Consequently we conclude that the method is zero stable.

4. In the established notation we have $\alpha_2 = 1$, $\alpha_1 = 2$ and $\alpha_0 = -3$. The $beta$ terms similarly come from the coefficients on the right hand side (remembering the denominator of 10).

   Now $\quad c_0 = \sum \alpha_j = 0$ $\quad$ and $\quad$ $c_1 = \sum j\alpha_j - \beta_j = 0$

   from which we conclude that the method is consistent.

   We also find that $\quad c_2 = \sum \frac{1}{2}j^2\alpha_j - j\beta_j = 0$ $\quad$ $c_3 = \sum \frac{1}{6}j^3\alpha_j - \frac{1}{2}j^2\beta_j = -0.533333$

   so that the method is of order 2 . This method is not to be recommended however (check the zero stability).

5. Each time step must be carried out in four stages. We start by calculating

$$K_1 = f(0, y_0) = f(0, 2) = 0.250000$$

   a value we now use in finding $K_2 = f(\frac{1}{2}h, y_0 + \frac{1}{2}hK_1) = f(0.2, 2.050000) = 0.237954$

   This value K_2 is now used in our evaluation of

$$K_3 = f(\tfrac{1}{2}h, y_0 + \tfrac{1}{2}hK_2) = f(0.2, 2.047591) = 0.238514$$

   which, in turn, is used in $K_4 = f(h, y_0 + hK_3) = f(0.4, 2.095406) = 0.227753$

   All four of these values are then used to complete the time step

$$\begin{aligned} y_1 &= y_0 + \frac{h}{6}(K_1 + 2K_2 + 2K_3 + K_4) \\ &= 2 + \frac{0.4}{6}(0.250000 + 2 \times 0.237954 + 2 \times 0.238514 + 0.227753) = 2.095379 \end{aligned}$$

Predictor-Corrector Methods

32.3

Introduction

In this final Section on numerical approximations for initial value problems involving ordinary differential equations we consider **predictor-corrector** methods. These methods are a way of getting around the difficulties inherent in implementing certain implicit numerical schemes.

Prerequisites

Before starting this Section you should ...

- review the preceding material in this Workbook

Learning Outcomes

On completion you should be able to ...

- implement simple predictor-corrector methods

1. Predictor-corrector methods

We have seen that when using an implicit linear multistep method there is an additional difficulty because we cannot, in general, solve simply for the newest approximate y-value y_{n+k}. A general k-step implicit method involves, at the k^{th} time step,

$$\underset{\substack{\uparrow \\ \text{the} \\ \text{unknown}}}{\alpha_k y_{n+k}} + \cdots + \alpha_1 y_1 + \alpha_0 y_0 = h(\underset{\substack{\uparrow \\ y_{n+k} \text{ occurs} \\ \text{here too}}}{\beta_k f_{n+k}} + \cdots + \beta_1 f_1 + \beta_0 f_0)$$

and if f depends on y in a complicated way then it is not obvious how to dig y_{n+k} out of $f_{n+k} = f((n+k)h, y_{n+k})$.

One solution to this problem would be to only ever use **explicit** methods in which $\beta_k = 0$. But this is not a good solution, for **implicit** methods generally have better properties than the explicit ones (for example, the implicit trapezium is second order while the explicit Euler is only first order).
Another solution involves a so-called **predictor-corrector** method. This involves

1. The predictor step. We use an explicit method to obtain an approximation y^P_{n+k} to y_{n+k}.

2. The corrector step. We use an implicit method, but with the predicted value y^P_{n+k} on the right-hand side in the evaluation of f_{n+k}. We use f^P_{n+k} to denote this approximate (predicted) value of f_{n+k}.

3. We can then go on to correct again and again. At each step we put the latest approximation to y_{n+k} in the right-hand side of the scheme (*via* f) to generate a new approximation from the left-hand side.

(This is not unlike an implementation of Newton-Raphson. In that method we require an initial guess (we "predict") and then the Newton-Raphson approach tells us how to iterate (or "correct") our latest approximation. The main difference here is that we have a systematic way of obtaining the initial prediction.)

It is sufficient for our purposes to illustrate the idea of a predictor-corrector method using the simplest possible pair of methods. We use Euler's method to predict and the trapezium method to correct.

Example 12

Suppose that $y = y(t)$ is the solution to the initial value problem

$$\frac{dy}{dt} = t + y, \qquad y(0) = 3$$

Use Euler's method and the trapezium method as a predictor-corrector pair (with one correction at each time step). Take the time step to be $h = 0.05$ so as to obtain approximations to $y(0.05)$ and $y(0.1)$.

Solution

Euler's method, $y_{n+1} = y_n + hf_n$, is the explicit method so we use that to **predict**. For the first time step we require $f_0 = f(0, y_0) = f(0, 3) = 3$ and therefore

$$y_1^P = y_0 + hf_0 = 3 + 0.05 \times 3 = 3.15$$

We now use this predicted value of y_1 to obtain a "predicted" value for f_1 which we can use in the implicit trapezium method. We find $f_1^P = f(h, y_1^P) = f(0.05, 3.15) = 3.2$. We now **correct** using the trapezium method in the form

$$y_1 = y_0 + \frac{h}{2}\left(f_0 + f_1^P\right) = 3 + \frac{1}{2}(0.05)(3 + 3.2) = 3.155$$

This completes prediction and one correction for the first time step.
For the second time step we require $f_1 = f(h, y_1) = f(0.05, 3.155) = 3.205$ and therefore

$$y_2^P = y_1 + hf_1 = 3.155 + 0.05 \times 3.205 = 3.31525$$

which is the predicted value for y_2. We now correct it with

$$y_2 = y_1 + \frac{h}{2}\left(f_1 + f_2^P\right) = 3.155 + \frac{1}{2}(0.05)(3.205 + 3.41525) = 3.320506$$

We conclude that

$$\begin{aligned} y(0.05) &\approx 3.155 \\ y(0.1) &\approx 3.320506 \end{aligned}$$

If correction is repeated until the corrected values settle down to a converged number then the approximation inherits all the (nice) properties of the implicit scheme. So, in the example above we would have second order accurate results obtained by a procedure which gets around the implicit nature of the trapezium method. Of course in the hand-calculations done above we only corrected once, rather than repeatedly to convergence.

The example above is such that the dependence of $f(t, y)$ on y is very simple and we could use the approach seen in Section 32.1 to implement the trapezium method. It turns out that the true trapezium method approximations to $y(0.05)$ and $y(0.1)$ are $y_1 = 3.155128$ and $y_2 = 3.320776$ respectively, to 6 decimal places. The predictor-corrector method will produce these values if enough corrections are taken.

As noted in the last paragraph, the example above was one in which it is possible to get around the

implicit nature of the trapezium method easily because of the simple way in which the right-hand side of the differential equation depends on y. This is not true of the next example.

Example 13

Suppose that $y = y(t)$ is the solution to the initial value problem

$$\frac{dy}{dt} = -\tan(y) \qquad y(0) = 1$$

Use Euler's method and the trapezium method as a predictor-corrector pair (with one correction at each time step). Take the time step to be $h = 0.2$ so as to obtain approximations to $y(0.2)$ and $y(0.4)$.

Solution

Euler's method, $y_{n+1} = y_n + hf_n$, is the explicit method so we use that to predict. For the first time step we require $f_0 = f(0, y_0) = f(0, 1) = -1.55741$ and therefore

$$y_1^P = y_0 + hf_0 = 1 + 0.2 \times -1.55741 = 0.688518$$

We now use this predicted value to obtain a "predicted" value for f_1 which we can use in the implicit trapezium method. We find $f_1^P = f(h, y_1^P) = f(0.2, 0.688518) = -0.82285$. We now correct using the trapezium method in the form

$$y_1 = y_0 + \frac{h}{2}\left(f_0 + f_1^P\right) = 1 + \frac{1}{2}(0.2)(-1.55741 - 0.822848) = 0.761974$$

This completes prediction and one correction for the first time step.

For the second time step we require $f_1 = f(h, y_1) = f(0.2, 0.761974) = -0.95422$ and therefore

$$y_2^P = y_1 + hf_1 = 0.76194 + 0.2 \times -0.95422 = 0.571131$$

which is the predicted value for y_2. We now correct it with

$$y_2 = y_1 + \frac{h}{2}\left(f_1 + f_2^P\right) = 0.761974 + \frac{1}{2}(0.2)(-0.95422 - -0.64257) = 0.602296$$

We conclude that

$$\begin{aligned} y(0.2) &\approx 0.761974 \\ y(0.4) &\approx 0.602296 \end{aligned}$$

Suppose that $y = y(t)$ is the solution to the initial value problem

$$\frac{dy}{dt} = \cos(y), \qquad y(0) = 0$$

Use Euler's method and the trapezium method as a predictor-corrector pair (with one correction at each time step). Take the time step to be $h = 0.1$ so as to obtain approximations to $y(0.1)$ and $y(0.2)$.

Your solution

Answer

Euler's method, $y_{n+1} = y_n + hf_n$, is the explicit method so we use that to predict. For the first time step we require $f_0 = f(0, y_0) = f(0, 0) = 1$ and therefore $y_1^P = y_0 + hf_0 = 0 + 0.1 \times 1 = 0.1$ We now use this predicted value to obtain a "predicted" value for f_1 which we can use in the implicit trapezium method. We find $f_1^P = f(h, y_1^P) = f(0.1, 0.1) = 0.995004$. We now correct using the trapezium method in the form $y_1 = y_0 + \frac{h}{2}\left(f_0 + f_1^P\right) = 0 + \frac{1}{2}(0.1)(1 + 0.995004) = 0.099750$ which completes the prediction and one correction for the first time step.

For the second time step we require $f_1 = f(h, y_1) = f(0.1, 0.099750) = 0.995029$ and therefore

$$y_2^P = y_1 + hf_1 = 0.099750 + 0.1 \times 0.995029 = 0.199253$$

which is the predicted value for y_2. We now correct it with

$$y_2 = y_1 + \frac{h}{2}\left(f_1 + f_2^P\right) = 0.099750 + \frac{1}{2}(0.1)(0.995029 + 0.980215) = 0.198512$$

We conclude that $y(0.1) \approx 0.099750, \quad y(0.2) \approx 0.198512$ to six decimal places.

Exercise

Suppose that $y = y(t)$ is the solution to the initial value problem

$$\frac{dy}{dt} = 1/(1+y^2) \qquad y(0) = 1$$

Use Euler's method and the trapezium method as a predictor-corrector pair (with one correction at each time step). Take the time step to be $h = 0.25$ so as to obtain approximations to $y(0.25)$ and $y(0.5)$.

Answer

Euler's method, $y_{n+1} = y_n + hf_n$, is the explicit method so we use that to predict. For the first time step we require $f_0 = f(0, y_0) = f(0, 1) = 0.5$ and therefore

$$y_1^P = y_0 + hf_0 = 1 + 0.25 \times 0.5 = 1.125$$

We now use this predicted value to obtain a "predicted" value for f_1 which we can use in the implicit trapezium method. We find $f_1^P = f(h, y_1^P) = f(0.25, 1.125) = 0.441379$. We now correct using the trapezium method in the form

$$y_1 = y_0 + \frac{h}{2}\left(f_0 + f_1^P\right) = 1 + \frac{1}{2}(0.25)(0.5 + 0.441379) = 1.117672$$

This completes prediction and one correction for the first time step.

For the second time step we require $f_1 = f(h, y_1) = f(0.25, 1.117672) = 0.444604$ and therefore

$$y_2^P = y_1 + hf_1 = 1.125 + 0.25 \times 0.444604 = 1.228823$$

which is the predicted value for y_2. We now correct it with

$$y_2 = y_1 + \frac{h}{2}\left(f_1 + f_2^P\right) = 1.117672 + \frac{1}{2}(0.25)(0.444604 + 0.398405) = 1.223049$$

We conclude that $y(0.25) \approx 1.117672, \quad y(0.5) \approx 1.223049$ to six decimal places.

Parabolic PDEs

32.4

Introduction

Second-order partial differential equations (PDEs) may be classified as parabolic, hyperbolic or elliptic. Parabolic and hyperbolic PDEs often model time dependent processes involving initial data.

In this Section we consider numerical solutions of parabolic problems.

Prerequisites

Before starting this Section you should ...

- review difference methods for first and second derivatives (HELM 31.3)

Learning Outcomes

On completion you should be able to ...

- implement simple methods to obtain approximate solutions of the heat diffusion equation

1. Definitions

We begin by giving some definitions.
Suppose that $u = u(x,t)$ satisfies the second order partial differential equation

$$Au_{xx} + Bu_{xt} + Cu_{tt} + Du_x + Eu_t + Fu = G$$

in which $A, \ldots, G$ are given functions. This equation is said to be

parabolic if $B^2 - 4AC = 0$

hyperbolic if $B^2 - 4AC > 0$

elliptic if $B^2 - 4AC < 0$

These may look like rather abstract definitions at this stage, but we will see that equations of different types give rise to mathematical models of different physical situations. In this Section we will consider equations only of the parabolic type. The hyperbolic type is dealt with later in this Workbook and the elliptic type is discussed in HELM 33.

2. Motivation

Consider an example of the type seen in the earlier material concerning separable solutions of the heat conduction equation. Suppose that $u = u(x,t)$ is the temperature of a metal bar a distance x from one end and at time t. For the sake of argument let us suppose that the metal bar has length equal to ℓ and that the ends are held at constant temperatures u_L at the left and u_R at the right.

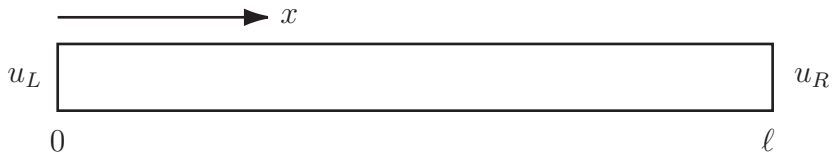


Figure 2

We also suppose that the temperature distribution at the initial time is known to be $f(x)$, with $f(0) = u_L$ and $f(\ell) = u_R$ so that the initial and boundary conditions do not give rise to a conflict at the ends of the bar at the initial time.

This physical situation may be modelled by

$$\left.\begin{array}{rcll} u_t & = & \alpha u_{xx} & (0 < x < \ell, \quad t > 0) \\ u(0,t) & = & u_L & (t > 0) \\ u(\ell,t) & = & u_R & (t > 0) \\ u(x,0) & = & f(x) & (0 < x < \ell) \end{array}\right\}$$

in which $\alpha > 0$ is a constant called the **thermal diffusivity** or simply the **diffusivity** of the metal. If the bar is made of aluminium then $\alpha = 0.86$ cm^2 s^{-1}, and if made of copper then $\alpha = 1.14$ cm^2 s^{-1}.

Using separation of variables and Fourier series (neither of which are required for the remainder of this Section) it can be shown that the solution to the above problem (in the case where $u_L = u_R = 0$) is

$$u(x,t) = \sum_{m=1}^{\infty} B_m e^{-m^2\alpha\pi^2 t/\ell^2} \sin(m\pi x/\ell), \qquad \text{where} \qquad B_m = \frac{2}{\ell}\int_0^{\ell} f(s)\sin(m\pi s/\ell)\, ds.$$

Now, let us be realistic. Any evaluation of u for particular choices of x and t must involve approximating the infinite series that defines u (that is, just taking the first few terms - and care is required if we are to be sure that we have taken enough). Also, in each of the terms we retain in the sum, we need to find B_m by integration. It is not surprising that computation of this procedure is a common approach. So if we (eventually) resort to computation in order to find u, why not *start* with a computational approach?

(This is not to say that there is no value in the analytic solution involving the B_m. The solution above is of great value, but we simply observe here that there are times when a computational approach is all we may end up needing.)

So, the aim of this Section is to derive methods for obtaining numerical solutions to parabolic problems of the type above. In fact, it is sufficient for our present purposes to restrict attention to that particular problem.

3. Approximating partial derivatives

Earlier, in HELM 31.3, we saw methods for approximating first and second derivatives of a function of one variable. We review some of that material here. If $y = y(x)$ then the forward and central difference approximations to the first derivative are:

$$\frac{dy}{dx} \approx \frac{y(x+\delta x) - y(x)}{\delta x}, \qquad \frac{dy}{dx} \approx \frac{y(x+\delta x) - y(x-\delta x)}{2\delta x}$$

and the central difference approximation to the second derivative is:

$$\frac{\mathrm{d}^2 y}{dx^2} \approx \frac{y(x+\delta x) - 2y(x) + y(x-\delta x)}{(\delta x)^2}$$

in which δx is a small x-increment. The quantity δx is what we previously referred to as h, but it is now convenient to use a notation which is more closely related to the independent variable (in this case x). (Examples implementing the difference approximations for derivatives can be found in HELM 31.)

We now return to the subject of this Section, that of *partial* derivatives. The PDE $u_t = \alpha u_{xx}$ involves the first derivative $\dfrac{\partial u}{\partial t}$ and the second derivative $\dfrac{\partial^2 u}{\partial x^2}$. We now adapt the ideas used for functions of one variable to the present case involving $u = u(x,t)$.

Let δt be a small increment of t, then the partial derivative $\dfrac{\partial u}{\partial t}$ may be approximated by:

$$\frac{\partial u}{\partial t} \approx \frac{u(x,t+\delta t) - u(x,t)}{\delta t}$$

Let δx be a small increment of x, then the partial derivative $\dfrac{\partial^2 u}{\partial x^2}$ may be approximated by:

$$\frac{\partial^2 u}{\partial x} \approx \frac{u(x+\delta x,t) - 2u(x,t) + u(x-\delta x,t)}{(\delta x)^2}$$

The two difference approximations above are the ones we will use later in this Section. Example 14 below refers to these and others.

Example 14

Consider the function u defined by

$$u(x,t) = \sin(x^2 + 2t)$$

Using increments of $\delta x = 0.004$ and $\delta t = 0.04$, and working to 8 decimal places, approximate

(a) $u_x(2,3)$ with a one-sided forward difference

(b) $u_{xx}(2,3)$ with a central difference

(c) $u_t(2,3)$ with a one-sided forward difference

(d) $u_t(2,3)$ with a central difference.

Enter your approximate derivatives to 3 decimal places.

Solution

The evaluations of u we will need are $u(x,t) = -0.54402111$, $u(x+\delta x,t) = -0.55738933$, $u(x-\delta x,t) = -0.53054047$, $u(x,t+\delta t) = -0.60933532$, $u(x,t-\delta t) = -0.47522703$. It follows that

(a) $u_x(2,3) \approx \dfrac{-0.55738933 + 0.54402111}{0.004} = -3.342$

(b) $u_{xx}(2,3) \approx \dfrac{-0.55738933 + 2 \times 0.54402111 - 0.53054047}{0.004^2} = 7.026$

(c) $u_t(2,3) \approx \dfrac{-0.60933532 + 0.54402111}{0.04} = -1.633$

(d) $u_t(2,3) \approx \dfrac{-0.60933532 + 0.47522703}{2 \times 0.04} = -1.676$

to 3 decimal places. (Workings shown to 8 decimal places.)

4. An explicit numerical method for the heat equation

The approximations used above for approximating partial derivatives can now be applied in order to derive a numerical method for solving the heat conduction problem

$$\begin{aligned} u_t &= \alpha u_{xx} && (0 < x < \ell, \quad t > 0) \\ u(0,t) &= 0 && (t > 0) \\ u(\ell,t) &= 0 && (t > 0) \\ u(x,0) &= f(x) && (0 < x < \ell). \end{aligned}$$

In order to specify the numerical method we choose values for δt and δx and use these in approximations of the two derivatives in the partial differential equation. It is convenient to divide the

interval $0 < x < \ell$ into equally spaced subintervals so, in effect, we choose a whole number J so that $\delta x = \dfrac{\ell}{J}$.

Key Point 15

In order to specify the numerical procedure for solving the heat conduction equation

$$\frac{\partial u}{\partial t} = \alpha \frac{\partial^2 u}{\partial x^2}$$

we need to choose

δt — the time step

δx — the space step

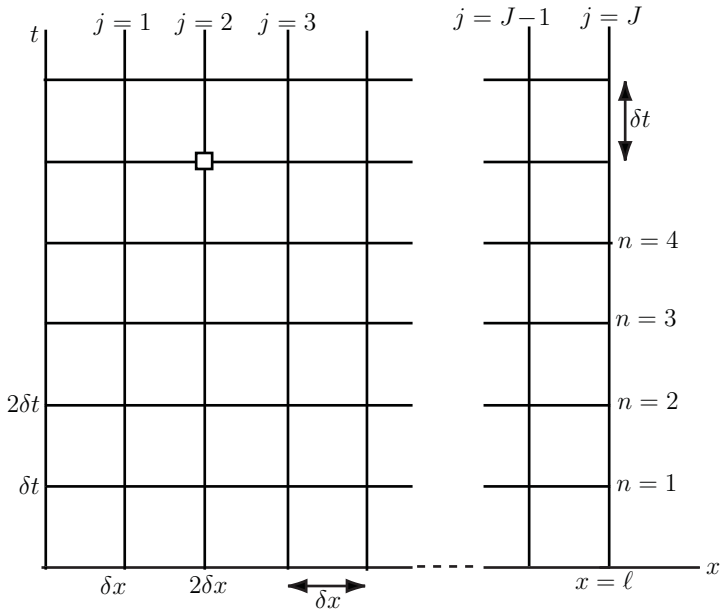


Figure 3

The diagram above shows the independent variables x and t at which we seek the function u. The numerical solution we shall find is a sequence of numbers which approximate u at a sequence of (x, t) points.

Key Point 16

The numerical approximations to $u(x,t)$ that we will find will be approximations to u at (x,t) values where the horizontal and vertical lines cross in the above diagram (Figure 3).

The notation we use is that

$$\underset{\substack{\uparrow \\ \text{numerical} \\ \text{approximation}}}{u_j^n} \quad \approx \quad \underbrace{u(j\ \delta x\ ,\ n\ \delta t)}_{\substack{\uparrow \\ \text{exact (i.e., unknown) solution} \\ \text{evaluated at } x = j \times \delta x,\ t = n \times \delta t}}$$

The idea is that the subscript j counts how many "steps" to the right we have taken from the origin and the superscript n counts how many time-steps (up, on the diagram) we have taken. To say this another way

u_j^n — the superscript counts up the t values; the subscript counts across the x values

For example, consider the point on Figure 3 which is highlighted with a small square. This point is two steps to the right of the origin (so that $j = 2$) and five steps up (so that $n = 5$). The exact solution evaluated at this point is $u(2\delta x, 5\delta t)$ and our numerical approximation to that value is u_2^5. Combining this new notation with the familiar idea for approximating derivatives we obtain the following approximation to the PDE

$$\frac{u_j^{n+1} - u_j^n}{\delta t} = \alpha\ \frac{u_{j-1}^n - 2u_j^n + u_{j+1}^n}{(\delta x)^2}$$

Key Point 17

The exact solution $u = u(x,t)$ satisfies the partial differential equation

$$u_t = \alpha u_{xx}$$

The approximate (numerical) solution satisfies the difference equation

$$\frac{u_j^{n+1} - u_j^n}{\delta t} = \alpha\ \frac{u_{j-1}^n - 2u_j^n + u_{j+1}^n}{(\delta x)^2}$$

The difference between the unknown exact solution and the numerical solution will be governed by how well the one-sided and central differences approximate the partial derivatives in the PDE.

To simplify (the appearance of) the numerical method we define a new quantity $r = \dfrac{\alpha\delta t}{(\delta x)^2}$ so that our numerical procedure can be written

$$u_j^{n+1} = u_j^n + r(u_{j-1}^n - 2u_j^n + u_{j+1}^n) = ru_{j-1}^n + (1-2r)u_j^n + ru_{j+1}^n$$

This equation defines a numerical "stencil" which allows us to find one of the values at the $n+1$ time level in terms of values at the previous level, n. In Figure 4 we envisage terms on the right-hand side of the above equation leading towards a result equal to the left-hand side, and the arrows therefore point towards the point at which u_j^{n+1} approximates u.

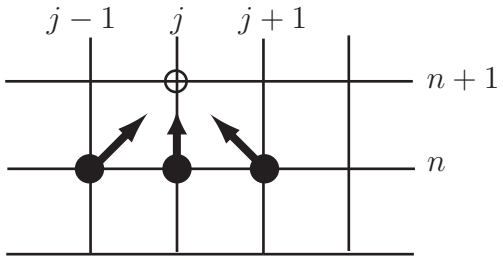


Figure 4

At the stage of the process depicted above, the solid circles represent points in the (x,t) plane where we have already found our numerical approximation. The unfilled circle is the point for which the new approximation u_j^{n+1} is being found.

Implementation

The initial condition gives u at $t=0$, and this information can be used to find

$$u_0^0, \quad u_1^0, \quad u_2^0, \quad \ldots, \quad u_J^0$$

that is, the numerical solution at all the selected x values and at $t=0$. In general

$$u_j^0 = f(j \times \delta x) = f_j$$

where f_j is a shorthand notation for $f(j \times \delta x)$.
Then we use the boundary conditions and numerical method

$$u_j^{n+1} = u_j^n + r(u_{j-1}^n - 2u_j^n + u_{j+1}^n)$$

(with $n=0$) to work out u_j^1 for $j = 0, 1, 2, \ldots, J$. (This completes the first time-step.)

The time-stepping procedure is then used repeatedly to find u_j^{n+1} in terms of the u_j^n, which are known either from the last time-step or (at the beginning) from the initial condition.

The time-stepping procedure is summarised in the following Key Point.

Key Point 18

Here the step-by-step process used to implement the numerical procedure is presented.

1. The initial condition implies that

$$u_j^0 = f_j \qquad (j = 0, 1, 2, \ldots, J)$$

(the boundary conditions could be used to find u_0^0 and u_J^0, but our supposition is that this is consistent with taking f_0 and f_J).

2. The first time-step

Here we find u_j^1 for $j = 0, 1, \ldots, J$.

(a) The boundary condition at $x = 0$ is $u(0, t) = u_L$. It follows that $u_0^1 = u_L$.

(b) The boundary condition at $x = \ell$ is $u(\ell, t) = u_R$. It follows that $u_J^1 = u_R$.

(c) Now we work from left to right finding u_j^1 at the interior points. This is achieved by repeatedly applying the general numerical scheme:

$$\begin{aligned} u_1^1 &= u_1^0 + r(u_0^0 - 2u_1^0 + u_2^0) \\ u_2^1 &= u_2^0 + r(u_1^0 - 2u_2^0 + u_3^0) \\ &\vdots \\ u_{J-1}^1 &= u_{J-1}^0 + r(u_{J-2}^0 - 2u_{J-1}^0 + u_J^0) \end{aligned}$$

This completes the first time-step. We have taken the initial data and used our approximation to the PDE to obtain an approximate solution at time $t = \delta t$.

3. The second time-step

Here we find u_j^2 for $j = 0, 1, \ldots, J$.

(a) The boundary condition at $x = 0$ is $u(0, t) = u_L$. It follows that $u_0^2 = u_L$.

(b) The boundary condition at $x = \ell$ is $u(\ell, t) = u_R$. It follows that $u_J^2 = u_R$.

(c) Now we work from left to right finding u_j^2 at the interior points. This is achieved by repeatedly applying the general numerical scheme:

$$\begin{aligned} u_1^2 &= u_1^1 + r(u_0^1 - 2u_1^1 + u_2^1) \\ u_2^2 &= u_2^1 + r(u_1^1 - 2u_2^1 + u_3^1) \\ &\vdots \\ u_{J-1}^2 &= u_{J-1}^1 + r(u_{J-2}^1 - 2u_{J-1}^1 + u_J^1) \end{aligned}$$

This completes the second time-step. We now have an approximation to u at time $t = 2\delta t$.

4. And so on ....

The following is a concrete example of the time-stepping procedure.

Example 15

The temperature $u(x,t)$ of a metal bar of length $\ell = 2$ at a distance x from one end and at time t is modelled by the partial differential equation

$$u_t = \alpha u_{xx} \qquad (0 < x < \ell, \quad t > 0)$$

It is given that the metal has diffusivity $\alpha = 4$, that the two ends of the bar are kept at temperature $u = 0$ and that the initial temperature distribution is

$$u(x,0) = f(x) = x(\ell - x)$$

Use the explicit difference scheme with $\delta x = 0.5$ and $\delta t = 0.01$ to approximate $u(x,t)$ at $t = \delta t$ and $t = 2\delta t$.

Solution

In this case $r = \alpha\delta t/(\delta x)^2 = 0.16$ so that the numerical method can be written

$$u_j^{n+1} = u_j^n + 0.16(u_{j-1}^n - 2u_j^n + u_{j+1}^n) = 0.68u_j^n + 0.16(u_{j-1}^n + u_{j+1}^n)$$

We now find u_j^0

$$\begin{array}{lcll} u_0^0 & = & 0 & \text{from the left-hand boundary condition} \\ u_1^0 & = & f(\delta x) = 0.75 & \text{from the initial condition} \\ u_2^0 & = & f(2\delta x) = 1 & \text{from the initial condition} \\ u_3^0 & = & f(3\delta x) = 0.75 & \text{from the initial condition} \\ u_4^0 & = & 0 & \text{from the boundary condition at the right hand end} \end{array}$$

The first time-step will find u_j^1, but first we note that $u_0^1 = u_4^1 = 0$ from the two boundary conditions. Now

$$\begin{array}{rcl} u_1^1 & = & 0.68u_1^0 + 0.16(u_0^0 + u_2^0) = 0.68 \times 0.75 + 0.16(0 + 1) = 0.670 \\ u_2^1 & = & 0.68u_2^0 + 0.16(u_1^0 + u_3^0) = 0.68 \times 1 + 0.16(0.75 + 0.75) = 0.920 \\ u_3^1 & = & 0.68u_3^0 + 0.16(u_2^0 + u_4^0) = 0.68 \times 0.75 + 0.16(1 + 0) = 0.670 \end{array}$$

The second time-step will find u_j^2, but first we note that $u_0^2 = u_4^2 = 0$ from the two boundary conditions. Now

$$\begin{array}{rcl} u_1^2 & = & 0.68u_1^1 + 0.16(u_0^1 + u_2^1) = 0.68 \times 0.67 + 0.16(0 + 0.92) = 0.603 \\ u_2^2 & = & 0.68u_2^1 + 0.16(u_1^1 + u_3^1) = 0.68 \times 0.92 + 0.16(0.67 + 0.67) = 0.84 \\ u_3^2 & = & 0.68u_3^1 + 0.16(u_2^1 + u_4^1) = 0.68 \times 0.67 + 0.16(0.92 + 0) = 0.603 \end{array}$$

(Quantities have been rounded to three decimal places here.)

Figure 5 plots the numerical solutions found in the example above. The initial condition is shown as circles. Results of the first time-step appear as squares and the second time-step is shown as stars. The line joining the values we found are not part of the numerical solution and are included only as

an aid to clarity.

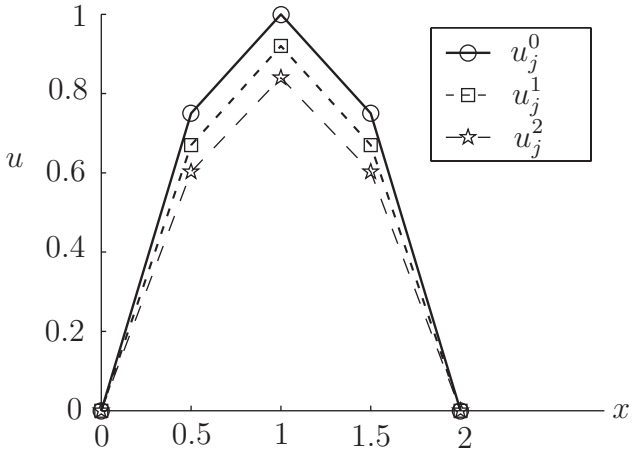


Figure 5

Notice how the numerical results are behaving as they should. The temperature decreases slightly at each time-step.

The temperature $u(x,t)$ of a metal bar of length $\ell = 2$ at a distance x from one end and at time t is modelled by the partial differential equation

$$u_t = \alpha u_{xx} \qquad (0 < x < \ell, \quad t > 0)$$

It is given that the metal has diffusivity $\alpha = 2.25$, that the two ends of the bar are kept at temperature $u = 0$ and that the initial temperature distribution is

$$u(x,0) = f(x) = \sin(\pi x/\ell)$$

Use the explicit difference scheme with $\delta x = 0.5$ and $\delta t = 0.05$ to approximate $u(x,t)$ at $t = \delta t$ and $t = 2\delta t$.

Your solution

Initial condition and first time-step:

Answer

In this case $r = \alpha\delta t/(\delta x)^2 = 0.45$ so that the numerical scheme can be written

$$u_j^{n+1} = u_j^n + 0.45(u_{j-1}^n - 2u_j^n + u_{j+1}^n) = 0.1u_j^n + 0.45(u_{j-1}^n + u_{j+1}^n)$$

The first stage is to use the given data to find u_j^0

$$\begin{array}{rcll} u_0^0 & = & 0 & \text{from the boundary condition} \\ u_1^0 & = & f(\delta x) = f(0.5) = 0.707 & \text{from the initial condition} \\ u_2^0 & = & f(2\delta x) = f(1) = 1 & \text{from the initial condition} \\ u_3^0 & = & f(3\delta x) = f(1.5) = 0.707 & \text{from the initial condition} \\ u_4^0 & = & 0 & \text{from the boundary condition} \end{array}$$

The first time-step will find u_j^1. First we note that the boundary condition implies that $u_0^1 = u_4^1 = 0$.

$$\begin{array}{rcl} u_1^1 & = & 0.1u_1^0 + 0.45(u_0^0 + u_2^0) = 0.1 \times 0.71 + 0.45(0 + 1) = 0.521 \\ u_2^1 & = & 0.1u_2^0 + 0.45(u_1^0 + u_3^0) = 0.1 \times 1 + 0.45(0.71 + 0.71) = 0.736 \\ u_3^1 & = & 0.1u_3^0 + 0.45(u_2^0 + u_4^0) = 0.1 \times 0.71 + 0.45(1 + 0) = 0.521 \end{array}$$

Your solution

Second time-step:

Answer

The second time-step will find u_j^2. First we note that the boundary condition implies that $u_0^2 = u_4^2 = 0$. Now

$$\begin{array}{rcl} u_1^2 & = & 0.1u_1^1 + 0.45(u_0^1 + u_2^1) = 0.1 \times 0.52 + 0.45(0 + 0.74) = 0.383 \\ u_2^2 & = & 0.1u_2^1 + 0.45(u_1^1 + u_3^1) = 0.1 \times 0.74 + 0.45(0.52 + 0.52) = 0.542 \\ u_3^2 & = & 0.1u_3^1 + 0.45(u_2^1 + u_4^1) = 0.1 \times 0.52 + 0.45(0.74 + 0) = 0.383 \end{array}$$

5. Stability of the simple explicit scheme

The purpose of the time-stepping scheme is to approximate $u(x,t)$ at later and later times t. It is clear that the larger we take the time step δt, the fewer steps will be necessary to reach a particular time t. One constraint on the size of δt is that we know from our earlier look at difference methods that derivative approximations are most accurate when *small* increments are used. However, as we will see in the next couple of pages, a far more telling constraint on the size of δt arises on consideration of **stability**. We begin with an Example.

Example 16

The temperature $u(x,t)$ of a metal bar of length $\ell = 1$ at a distance x from one end and at time t is modelled by the partial differential equation

$$u_t = \alpha u_{xx} \qquad (0 < x < \ell, \quad t > 0)$$

It is given that the metal has diffusivity $\alpha = 1$, that the two ends of the bar are kept at temperature $u = 0$ and that the initial temperature distribution is

$$u(x,0) = f(x) = x(\ell - x)$$

Use the explicit difference scheme with $\delta x = 0.25$ and $\delta t = 0.075$ to approximate $u(x,t)$ at $t = \delta t$ and $t = 2\delta t$.

Solution

In this case $r = \alpha\delta t/(\delta x)^2 = 1.2$ so that the numerical scheme can be written

$$u_j^{n+1} = u_j^n + 1.2(u_{j-1}^n - 2u_j^n + u_{j+1}^n) = -1.4u_j^n + 1.2(u_{j-1}^n + u_{j+1}^n)$$

The first stage is to use the given data to find u_j^0

$$\begin{array}{lcll} u_0^0 & = & 0 & \text{from the boundary condition} \\ u_1^0 & = & f(\delta x) = f(0.25) = 0.188 & \text{from the initial condition} \\ u_2^0 & = & f(2\delta x) = f(0.5) = 0.25 & \text{from the initial condition} \\ u_3^0 & = & f(3\delta x) = f(0.75) = 0.188 & \text{from the initial condition} \\ u_4^0 & = & 0 & \text{from the boundary condition} \end{array}$$

The first time-step will find u_j^1. First we note that the boundary condition implies that $u_0^1 = u_4^1 = 0$.

$$\begin{array}{lcl} u_1^1 & = & -1.4u_1^0 + 1.2(u_0^0 + u_2^0) = -1.4 \times 0.19 + 1.2(0 + 0.25) = 0.038 \\ u_2^1 & = & -1.4u_2^0 + 1.2(u_1^0 + u_3^0) = -1.4 \times 0.25 + 1.2(0.188 + 0.188) = 0.1 \\ u_3^1 & = & -1.4u_3^0 + 1.2(u_2^0 + u_4^0) = -1.4 \times 0.19 + 1.2(0.25 + 0) = 0.038 \end{array}$$

The second time-step will find u_j^2. First we note that the boundary condition implies that $u_0^2 = u_4^2 = 0$. Now

$$\begin{array}{lcl} u_1^2 & = & -1.4u_1^1 + 1.2(u_0^1 + u_2^1) = -1.4 \times 0.04 + 1.2(0 + 0.1) = 0.067 \\ u_2^2 & = & -1.4u_2^1 + 1.2(u_1^1 + u_3^1) = -1.4 \times 0.1 + 1.2(0.038 + 0.038) = -0.05 \\ u_3^2 & = & -1.4u_3^1 + 1.2(u_2^1 + u_4^1) = -1.4 \times 0.04 + 1.2(0.1 + 0) = 0.067 \end{array}$$

Figure 6 shows the results found in Example 16.

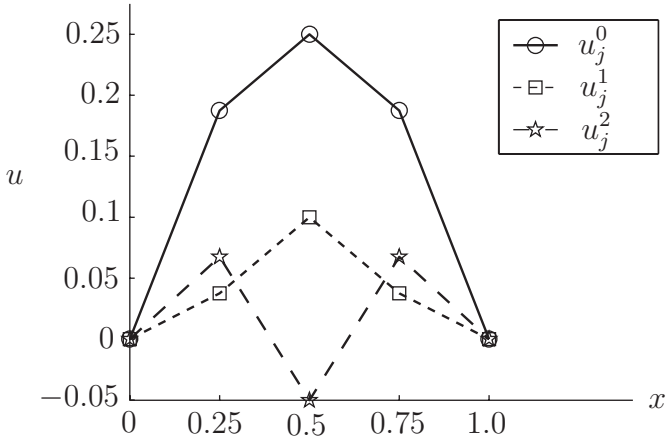


Figure 6

Something has gone wrong here. And it only gets worse in subsequent time-steps. After 9 time-steps the numerical solution approximating $u(x,t)$ at $t = 9\delta t$ is

$$\begin{aligned} u(0.25, 9\delta t) &\approx u_1^9 = -140.5531 \\ u(0.50, 9\delta t) &\approx u_2^9 = 198.7722 \\ u(0.75, 9\delta t) &\approx u_3^9 = -140.5531 \end{aligned}$$

(to 4 decimal places). This is an example of **instability**. A part of the numerical solution wants to keep growing and growing in a way that is not a part of the engineering application being modelled. There are many different definitions of (in)stability, and they often depend on the specific application in mind. For the heat conduction problem under discussion here, the following definition is sufficient.

Key Point 19

The explicit difference scheme $u_j^{n+1} = ru_{j-1}^n + (1-2r)u_j^n + ru_{j+1}^n \qquad \left(r = \dfrac{\alpha\delta t}{(\delta x)^2}\right)$

$$\begin{aligned} u_0^n &= 0 && (n > 0) \\ u_J^n &= 0 && (n > 0) \\ u_j^0 &= f(j\ \delta x) && (j = 1, 2, \ldots, J-1) \end{aligned}$$

where $J\delta x = \ell$, approximating the heat conduction problem

$$\left.\begin{aligned} u_t &= \alpha u_{xx} && (0 < x < \ell, \quad t > 0) \\ u(0,t) &= 0 && (t > 0) \\ u(\ell,t) &= 0 && (t > 0) \\ u(x,0) &= f(x) && (0 < x < \ell). \end{aligned}\right\}$$

is said to be **stable** if the approximations u_j^n do not grow in magnitude with n.

(Of course, there are applications where the principal quantity of interest *does* grow with time, and in these cases other definitions of stability are appropriate.)

The main stability result for the explicit scheme is proved in many textbooks on the subject, but for this Workbook it is sufficient to simply state it.

Key Point 20

The explicit scheme is stable if and only if

$$r \leq \tfrac{1}{2}$$

Writing this another way we see that the restriction on the time-step is that

$$\delta t \leq \frac{\delta x^2}{2\alpha}$$

Why is the stability constraint a problem?

In the above account it has been stated that the stability constraint is a severe restriction on the time-step δt. Here we discuss why this is the case.

For sake of argument let us take an example where $\alpha = 1$ and choose $\delta x = \dfrac{1}{10}$. The stability requirement insists that we must choose

$$\delta t \leq \frac{1}{2}\delta x^2 = \frac{1}{200},$$

which is much smaller than δx. If we require an even smoother approximation in the x direction we could halve δx taking it to be equal to $\dfrac{1}{20}$. It is now necessary that

$$\delta t \leq \frac{1}{2}\delta x^2 = \frac{1}{800}.$$

Decreasing δx by a factor of 2 causes δt to decrease by a factor of 4. The problem is that the upper bound on δt involves the *square* of δx, which is likely to be very small.

The following method overcomes the requirement of tiny time-steps.

6. The Crank-Nicolson method

In the notation established for the explicit method, the so-called Crank-Nicolson scheme can be written

$$u_j^{n+1} = u_j^n + \tfrac{1}{2}r\left(\underbrace{u_{j-1}^n - 2u_j^n + u_{j+1}^n}_{\dagger} \quad + \quad \underbrace{u_{j-1}^{n+1} - 2u_j^{n+1} + u_{j+1}^{n+1}}_{\ddagger}\right)$$

which might, at first glance, look off-puttingly complicated. To aid clarity, certain groups of terms have been gathered together in the above:

† these are the terms that appeared on the right hand side of the explicit method and are involved with approximating u_{xx} at time $t = n\ \delta t$

‡ these are very similar to the † terms, but all the superscripts are $n+1$ instead of n, that is the terms ‡ approximate u_{xx} at time $t = (n+1)\ \delta t$

(the factor of $\frac{1}{2}$ outside the large bracket shows that we take the *average* of † and ‡)

Figure 7 shows another way of thinking of this numerical method. As in the earlier diagram of this type, arrows point away from positions relating to terms on the right-hand side of the numerical scheme.

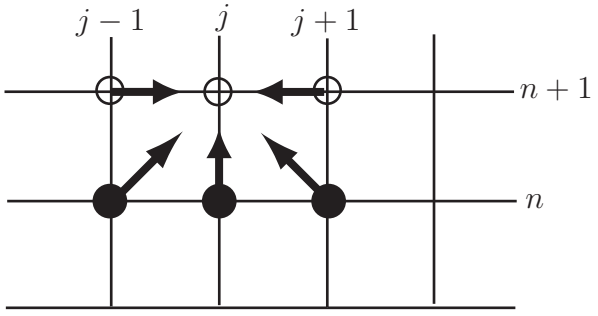


Figure 7

The new terms in the Crank-Nicolson method, as compared with the explicit method, give rise to two new unfilled circles on the diagram and the horizontal arrows.

The implementation of this method is similar to that used for the explicit method, but there is a key difference. The Crank-Nicolson scheme is **implicit**, for consider its use in the first time-step when finding u_j^1,

$$u_j^1 = \underbrace{u_j^0}_{\checkmark} \ + \ \tfrac{1}{2}r\left(\underbrace{u_{j-1}^0}_{\checkmark} - 2\underbrace{u_j^0}_{\checkmark} + \underbrace{u_{j+1}^0}_{\checkmark} \quad + \quad \underbrace{u_{j-1}^1 - 2u_j^1 + u_{j+1}^1}_{?}\right)$$

The terms labelled ✓ are known from the initial condition. But there are other unknown terms on the right-hand side. We cannot simply "read off" the values at the new time-step as we did using the explicit scheme. Instead we have to store all of the equations given by the stencil at a particular time-step and then solve them as a system of simultaneous equations. The following Example illustrates this point.

Example 17

The temperature $u(x,t)$ of a metal bar of length $\ell = 1.2$ at a distance x from one end and at time t is modelled by the partial differential equation

$$u_t = \alpha u_{xx} \qquad (0 < x < \ell, \quad t > 0).$$

It is given that the metal has diffusivity $\alpha = 1$, that the two ends of the bar are kept at temperature $u = 0$ and that the initial temperature distribution is

$$u(x,0) = f(x) = x\sqrt{(\ell - x)^3}$$

Use the Crank-Nicolson difference scheme with $\delta x = 0.4$ and $\delta t = 0.1$ to approximate $u(x,t)$ at $t = \delta t$ and $t = 2\delta t$.

Solution

In this case $r = \alpha\delta t/(\delta x)^2 = 0.62500$ so that the numerical scheme can be written

$$u_j^{n+1} = u_j^n + \frac{0.62500}{2}(u_{j-1}^n - 2u_j^n + u_{j+1}^n + u_{j-1}^{n+1} - 2u_j^{n+1} + u_{j+1}^{n+1})$$

Moving the unknowns to the left of the equation we obtain

$$-0.31250u_{j-1}^{n+1} + 1.62500u_j^{n+1} - 0.31250u_{j+1}^{n+1} = 0.37500u_j^n + 0.31250(u_{j-1}^n + u_{j+1}^n)$$

The first stage is to use the given data to find u_j^0

u_0^0	$=$	0	from the boundary condition
u_1^0	$=$	$f(\delta x) = f(0.4) = 0.28622$	from the initial condition
u_2^0	$=$	$f(2\delta x) = f(0.8) = 0.20239$	from the initial condition
u_3^0	$=$	0	from the boundary condition

The first time-step will find u_j^1. First we note that the boundary condition implies that $u_0^1 = u_3^1 = 0$. Two uses of the stencil give

$$\begin{aligned} -0.31250u_0^1 + 1.62500u_1^1 - 0.31250u_2^1 = 0.37500u_1^0 + 0.31250(u_0^0 + u_2^0) &= 0.17058 \\ -0.31250u_1^1 + 1.62500u_2^1 - 0.31250u_3^1 = 0.37500u_2^0 + 0.31250(u_1^0 + u_3^0) &= 0.16534 \end{aligned}$$

The implicit nature of this method means that we have to do some extra work to complete the time-step. We must now solve the simultaneous equations

$$\begin{pmatrix} 1.62500 & -0.31250 \\ -0.31250 & 1.62500 \end{pmatrix} \begin{pmatrix} u_1^1 \\ u_2^1 \end{pmatrix} = \begin{pmatrix} 0.17058 \\ 0.16534 \end{pmatrix}$$

In this case there are only two unknowns and it is a simple matter to solve the pair of equations to give $u_1^1 = 0.12932$ and $u_2^1 = 0.12662$.

Solution (contd.)

The second time-step will find u_j^2. First we note that the boundary condition implies that $u_0^2 = u_3^2 = 0$. Two uses of the stencil give

$$\begin{aligned} -0.31250u_0^2 + 1.62500u_1^2 - 0.31250u_2^2 = 0.37500u_1^1 + 0.31250(u_0^1 + u_2^1) &= 0.08806 \\ -0.31250u_1^2 + 1.62500u_2^2 - 0.31250u_3^2 = 0.37500u_2^1 + 0.31250(u_1^1 + u_3^1) &= 0.08789 \end{aligned}$$

The implicit nature of this method means that we have to do some extra work to complete the time-step. We must now solve the simultaneous equations

$$\begin{pmatrix} 1.62500 & -0.31250 \\ -0.31250 & 1.62500 \end{pmatrix} \begin{pmatrix} u_1^2 \\ u_2^2 \end{pmatrix} = \begin{pmatrix} 0.08806 \\ 0.08789 \end{pmatrix}$$

In this case there are only two unknowns and it is a simple matter to solve the pair of equations to give $u_1^2 = 0.06707$ and $u_2^2 = 0.06699$.

Figure 8 depicts the numerical solutions found in Example 17 above. (Again, the dotted lines are intended to aid clarity, they are not part of the numerical solution.)

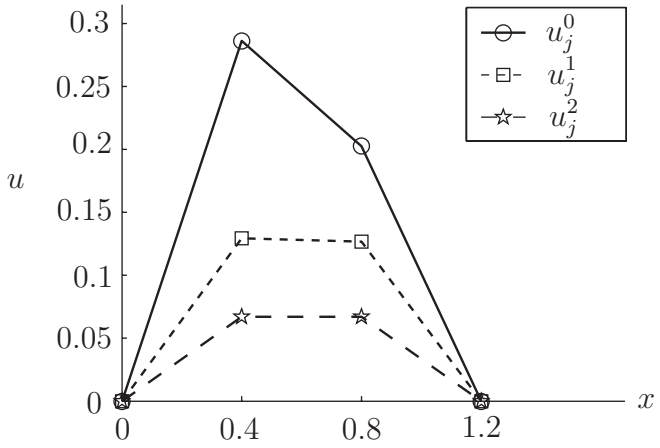


Figure 8

The temperature $u(x,t)$ of a metal bar of length $\ell = 0.9$ at a distance x from one end and at time t is modelled by the partial differential equation

$$u_t = \alpha u_{xx} \qquad (0 < x < \ell, \quad t > 0).$$

It is given that the metal has diffusivity $\alpha = 0.25$, that the two ends of the bar are kept at temperature $u = 0$ and that the initial temperature distribution is

$$u(x,0) = f(x) = \sin(\pi x/\ell)$$

Use the Crank-Nicolson difference scheme with $\delta x = 0.3$ and $\delta t = 0.2$ to approximate $u(x,t)$ at $t = \delta t$ and $t = 2\delta t$.

Your solution

Initial condition and first time-step:

Answer

In this case $r = \alpha\delta t/(\delta x)^2 = 0.55556$ so that the numerical scheme can be written

$$u_j^{n+1} = u_j^n + \frac{0.55556}{2}(u_{j-1}^n - 2u_j^n + u_{j+1}^n + u_{j-1}^{n+1} - 2u_j^{n+1} + u_{j+1}^{n+1})$$

Moving the unknowns to the left of the equation we obtain

$$-0.27778u_{j-1}^{n+1} + 1.55556u_j^{n+1} - 0.27778u_{j+1}^{n+1} = 0.44444u_j^n + 0.27778(u_{j-1}^n + u_{j+1}^n)$$

The first stage is to use the given data to find u_j^0

u_0^0	$=$	0	from the boundary condition
u_1^0	$=$	$f(\delta x) = f(0.3) = 0.86603$	from the initial condition
u_2^0	$=$	$f(2\delta x) = f(0.6) = 0.86603$	from the initial condition
u_3^0	$=$	0	from the boundary condition

The first time-step will find u_j^1. First we note that the boundary condition implies that $u_0^1 = u_3^1 = 0$.

Two uses of the stencil give

$$\begin{aligned} -0.27778u_0^1 + 1.55556u_1^1 - 0.27778u_2^1 = 0.44444u_1^0 + 0.27778(u_0^0 + u_2^0) &= 0.62546 \\ -0.27778u_1^1 + 1.55556u_2^1 - 0.27778u_3^1 = 0.44444u_2^0 + 0.27778(u_1^0 + u_3^0) &= 0.62546 \end{aligned}$$

The implicit nature of this method means that we have to do some extra work to complete the time-step. We must now solve the simultaneous equations

$$\begin{pmatrix} 1.55556 & -0.27778 \\ -0.27778 & 1.55556 \end{pmatrix} \begin{pmatrix} u_1^1 \\ u_2^1 \end{pmatrix} = \begin{pmatrix} 0.62546 \\ 0.62546 \end{pmatrix}$$

In this case there are only two unknowns and it is a simple matter to solve the pair of equations to give $u_1^1 = 0.48949$ and $u_2^1 = 0.48949$.

Your solution

Second time-step:

Answer
The second time-step will find u_j^2. First we note that the boundary condition implies that $u_0^2 = u_3^2 = 0$. Two uses of the stencil give

$$\begin{aligned} -0.27778u_0^2 + 1.55556u_1^2 - 0.27778u_2^2 = 0.44444u_1^1 + 0.27778(u_0^1 + u_2^1) &= 0.35352 \\ -0.27778u_1^2 + 1.55556u_2^2 - 0.27778u_3^2 = 0.44444u_2^1 + 0.27778(u_1^1 + u_3^1) &= 0.35352 \end{aligned}$$

The implicit nature of this method means that we have to do some extra work to complete the time-step. We must now solve the simultaneous equations

$$\begin{pmatrix} 1.55556 & -0.27778 \\ -0.27778 & 1.55556 \end{pmatrix} \begin{pmatrix} u_1^2 \\ u_2^2 \end{pmatrix} = \begin{pmatrix} 0.35352 \\ 0.35352 \end{pmatrix}$$

In this case there are only two unknowns and it is a simple matter to solve the pair of equations to give $u_1^2 = 0.27667$ and $u_2^2 = 0.27667$.

In general

Having now seen some instances with a relatively large δx, we now look at the general case where the space step may be much smaller. In this case there will be a larger system of equations to solve at each time-step than was the case above.

In general, the procedure of moving the unknowns to the left hand side of the equation leads to

$$-\frac{r}{2}u_{j-1}^{n+1} + (1+r)u_j^{n+1} - \frac{r}{2}u_{j+1}^{n+1} = \frac{r}{2}u_{j-1}^n + (1-r)u_j^n + \frac{r}{2}u_{j+1}^n$$

which we apply all the way along the x-axis. That is, we put $j = 1, 2, 3, \ldots, J-1$ in the above expression and hence derive a system of equations for all the u with superscript $n+1$.

$$\begin{pmatrix} 1+r & -\frac{r}{2} & 0 & \ldots & \ldots & 0 \\ -\frac{r}{2} & 1+r & -\frac{r}{2} & & & \\ 0 & -\frac{r}{2} & 1+r & -\frac{r}{2} & & \\ \vdots & & \ddots & \ddots & \ddots & \\ & & & & & -\frac{r}{2} \\ 0 & \ldots & \ldots & 0 & -\frac{r}{2} & 1+r \end{pmatrix} \begin{pmatrix} u_1^{n+1} \\ u_2^{n+1} \\ u_3^{n+1} \\ \vdots \\ u_{J-1}^{n+1} \end{pmatrix} = \begin{pmatrix} \frac{r}{2}\underline{u_0^n} + (1-r)u_1^n + \frac{r}{2}u_2^n + \frac{r}{2}\underline{\underline{u_0^{n+1}}} \\ \frac{r}{2}u_1^n + (1-r)u_2^n + \frac{r}{2}u_3^n \\ \frac{r}{2}u_2^n + (1-r)u_3^n + \frac{r}{2}u_4^n \\ \vdots \\ \frac{r}{2}u_{J-2}^n + (1-r)u_{J-1}^n + \frac{r}{2}\underline{u_J^n} + \frac{r}{2}\underline{\underline{u_J^{n+1}}} \end{pmatrix}$$

The underlined terms on the right-hand side will be known from the boundary conditions. The doubly underlined quantities are "new" at the current time-step and involve the only appearances of $n+1$ on the right-hand side. All the other u approximations at time level $n+1$ are unknown at this stage and appear on the left.

The matrix on the left-hand side of the system has the following properties

- It is **independent of n**. In other words, the same matrix appears at each time-step. (We saw this in the example and exercise above in which the same 2×2 matrix appeared at each of the two time-steps carried out).

- It is **tridiagonal**. That is, the only non-zero entries are either on, or adjacent to, the diagonal. Furthermore, there are only two different values ($\frac{r}{2}$ and $1+r$) which appear. This is good news as far as storage is concerned. Gaussian elimination (seen in HELM 30, for example) works extremely well on tridiagonal matrices.

It is also true that the matrix is **strictly diagonally dominant**. (That is, the diagonal element on each row is greater in size than the sum of the absolute values of the off-diagonal elements on that row.) This means that methods such as Jacobi and Gauss Seidel (see HELM 30 for details) would work very well.

Stability of the Crank-Nicolson scheme

This is the big pay-off when using the Crank-Nicolson method.

Key Point 21

The Crank-Nicolson method is stable for all values of r.

This is excellent news. It means that there is no hideously restrictive constraint on the size of δt.

7. Cost -v- benefit

At a first reading of this Section, it might be tempting to think that the extra effort involved in using Crank-Nicolson (we have to store a set of simultaneous equations, we have to solve them and we have to do this at every time-step) is enough to make the explicit method the winner in a cost-benefit analysis. **But this would be wrong.**

In practical problems involving numerical approximations to parabolic problems the explicit method is rarely good enough. The stability constraint ($r \leq \frac{1}{2}$) imposes such tiny time-steps that it takes a great deal of time for a computer to produce approximations corresponding to even fairly modest values of t. If efficiency is what matters, then Crank-Nicolson beats the explicit approach, and it is worth the extra initial effort formulating a solver (such as those we saw in HELM 30) for the system of equations.

Exercises

1. Consider the function u defined by

   $$u(x,t) = x^3\cos(xt)$$

   Using increments of $\delta x = 0.005$ and $\delta t = 0.01$, and working to 8 decimal places, approximate

   (a) $u_x(2,3)$ with a one-sided forward difference

   (b) $u_{xx}(2,3)$ with a central difference

   (c) $u_t(2,3)$ with a one-sided forward difference

   (d) $u_t(2,3)$ with a central difference.

   State the approximate derivatives to 3 decimal places.

2. The temperature $u(x,t)$ of a metal bar of length $\ell = 3$ at a distance x from one end and at time t is modelled by the partial differential equation

   $$u_t = \alpha u_{xx} \qquad (0 < x < \ell, \quad t > 0)$$

   It is given that the metal has diffusivity $\alpha = 1.6$, that the two ends of the bar are kept at temperature $u = 0$ and that the initial temperature distribution is

   $$u(x,0) = f(x) = x(\ell - x)$$

   Use the explicit difference scheme with $\delta x = 0.75$ and $\delta t = 0.08$ to approximate $u(x,t)$ at $t = \delta t$ and $t = 2\delta t$.

3. The temperature $u(x,t)$ of a metal bar of length $\ell = 1.2$ at a distance x from one end and at time t is modelled by the partial differential equation

   $$u_t = \alpha u_{xx} \qquad (0 < x < \ell, \quad t > 0).$$

   It is given that the metal has diffusivity $\alpha = 2.25$, that the two ends of the bar are kept at temperature $u = 0$ and that the initial temperature distribution is

   $$u(x,0) = f(x) = \sin(\pi x/\ell)$$

   Use the Crank-Nicolson difference scheme with $\delta x = 0.4$ and $\delta t = 0.06$ to approximate $u(x,t)$ at $t = \delta t$ and at $t = 2\delta t$.

Answers

1. The evaluations of u we will need are $u(x,t) = -0.41614684$, $u(x+\delta x,t) = -0.43162908$, $u(x-\delta x,t) = -0.40095819$, $u(x,t+\delta t) = -0.42521885$, $u(x,t-\delta t) = -0.40703321$. It follows that

   (a) $u_x(1,2) \approx \dfrac{-0.43162908 + 0.41614684}{0.005} = -3.096$

   (b) $u_{xx}(1,2) \approx \dfrac{-0.43162908 + 2 \times 0.41614684 - 0.40095819}{0.005^2} = -11.744$

   (c) $u_t(1,2) \approx \dfrac{-0.42521885 + 0.41614684}{0.01} = -0.907$

   (d) $u_t(1,2) \approx \dfrac{-0.42521885 + 0.40703321}{2 \times 0.01} = -0.909$

   to 3 decimal places. (Workings shown to 8 decimal places.)

2. In this case $r = \alpha^2 \delta t/(\delta x)^2 = 0.227556$ so that the numerical scheme can be written

$$u_j^{n+1} = u_j^n + 0.227556(u_{j-1}^2 - 2u_j^n + u_{j+1}^n) = 0.544889u_j^n + 0.227556(u_{j-1}^2 + u_{j+1}^n)$$

   The first stage is to use the given data to find u_j^0

u_0^0	$=$	0	from the boundary condition
u_1^0	$=$	$f(\delta x) = f(0.75) = 1.6875$	from the initial condition
u_2^0	$=$	$f(2\delta x) = f(1.5) = 2.25$	from the initial condition
u_3^0	$=$	$f(3\delta x) = f(2.25) = 1.6875$	from the initial condition
u_4^0	$=$	0	from the boundary condition

   The first timestep will find u_j^1. We note that the boundary condition implies that $u_0^1 = u_4^1 = 0$.

$$u_1^1 = 0.544889u_1^0 + 0.227556(u_0^0 + u_2^0) = 0.544889 \times 1.6875 + 0.227556(0 + 2.25) = 1.4315$$
$$u_2^1 = 0.544889u_2^0 + 0.227556(u_1^0 + u_3^0) = 0.544889 \times 2.25 + 0.227556(1.688 + 1.688) = 1.994$$
$$u_3^1 = 0.544889u_3^0 + 0.227556(u_2^0 + u_4^0) = 0.544889 \times 1.6875 + 0.227556(2.25 + 0) = 1.4315$$

   The second timestep will find u_j^2. First we note that the boundary condition implies that $u_0^2 = u_4^2 = 0$.

$$u_1^2 = 0.544889u_1^1 + 0.227556(u_0^1 + u_2^1) = 0.544889 \times 1.4315 + 0.227556(0 + 1.994) = 1.233754$$
$$u_2^2 = 0.544889u_2^1 + 0.227556(u_1^1 + u_3^1) = 0.544889 \times 1.994 + 0.227556(1.432 + 1.432) = 1.738$$
$$u_3^2 = 0.544889u_3^1 + 0.227556(u_2^1 + u_4^1) = 0.544889 \times 1.4315 + 0.227556(1.994 + 0) = 1.233754$$

   where some quantities have been rounded to 6 decimal places.

Answers

3. In this case $r = \alpha\delta t/(\delta x)^2 = 0.84375$ so that the numerical scheme can be written

$$u_j^{n+1} = u_j^n + \frac{0.84375}{2}(u_{j-1}^n - 2u_j^n + u_{j+1}^n + u_{j-1}^{n+1} - 2u_j^{n+1} + u_{j+1}^{n+1})$$

Moving the unknowns to the left of the equation we obtain

$$-0.42188u_{j-1}^{n+1} + 1.84375u_j^{n+1} - 0.42188u_{j+1}^{n+1} = 0.15625u_j^n + 0.42188(u_{j-1}^n + u_{j+1}^n)$$

The first stage is to use the given data to find u_j^0

$$\begin{array}{lcll} u_0^0 & = & 0 & \text{from the boundary condition} \\ u_1^0 & = & f(\delta x) = f(0.4) = 0.86603 & \text{from the initial condition} \\ u_2^0 & = & f(2\delta x) = f(0.8) = 0.86603 & \text{from the initial condition} \\ u_3^0 & = & 0 & \text{from the boundary condition} \end{array}$$

The first time-step will find u_j^1. First we note that the boundary condition implies that $u_0^1 = u_3^1 = 0$. Two uses of the stencil give

$$\begin{array}{rcl} -0.42188u_0^1 + 1.84375u_1^1 - 0.42188u_2^1 = 0.15625u_1^0 + 0.42188(u_0^0 + u_2^0) & = & 0.50067 \\ -0.42188u_1^1 + 1.84375u_2^1 - 0.42188u_3^1 = 0.15625u_2^0 + 0.42188(u_1^0 + u_3^0) & = & 0.50067 \end{array}$$

The implicit nature of this method means that we have to do some extra work to complete the time-step. We must now solve the simultaneous equations

$$\begin{pmatrix} 1.84375 & -0.42188 \\ -0.42188 & 1.84375 \end{pmatrix} \begin{pmatrix} u_1^1 \\ u_2^1 \end{pmatrix} = \begin{pmatrix} 0.50067 \\ 0.50067 \end{pmatrix}$$

In this case there are only two unknowns and it is a simple matter to solve the pair of equations to give $u_1^1 = 0.35212$ and $u_2^1 = 0.35212$.
The second time-step will find u_j^2. First we note that the boundary condition implies that $u_0^2 = u_3^2 = 0$. Two uses of the stencil give

$$\begin{array}{rcl} -0.42188u_0^2 + 1.84375u_1^2 - 0.42188u_2^2 = 0.15625u_1^1 + 0.42188(u_0^1 + u_2^1) & = & 0.20357 \\ -0.42188u_1^2 + 1.84375u_2^2 - 0.42188u_3^2 = 0.15625u_2^1 + 0.42188(u_1^1 + u_3^1) & = & 0.20357 \end{array}$$

The implicit nature of this method means that we have to do some extra work to complete the time-step. We must now solve the simultaneous equations

$$\begin{pmatrix} 1.84375 & -0.42188 \\ -0.42188 & 1.84375 \end{pmatrix} \begin{pmatrix} u_1^2 \\ u_2^2 \end{pmatrix} = \begin{pmatrix} 0.20357 \\ 0.20357 \end{pmatrix}$$

In this case there are only two unknowns and it is a simple matter to solve the pair of equations to give $u_1^2 = 0.14317$ and $u_2^2 = 0.14317$.

Hyperbolic PDEs

32.5

Introduction

In the preceding Section we looked at parabolic partial differential equations. Another class of PDE modelling initial value problems are of the hyperbolic type.

In this Section we will concentrate on the wave equation, which was introduced in HELM 25.

Prerequisites

Before starting this Section you should ...

- revise those aspects of HELM 25 which deal with the wave equation
- familiarise yourself with difference methods for approximating first and second derivatives
- be familiar with the numerical methods used for parabolic equations

Learning Outcomes

On completion you should be able to ...

- obtain simple numerical solutions of the wave equation

1. The (one-dimensional) wave equation

The wave equation is a PDE which (as its name suggests) models wave-like phenomena. It is a model of waves on water, of sound waves, of waves of reactant in chemical reactions and so on. For the purposes of most of the following examples we may think of the application in hand as that of being a length of string tightly stretched between two points. Let $u = u(x,t)$ be the displacement from rest of the string at time t and distance x from one end. Oscillations in the string may be modelled by the **wave equation**

$$u_{tt} = c^2 u_{xx} \qquad (0 < x < \ell, \quad t > 0)$$

where ℓ is the length of the string, $t = 0$ is some initial time and $c > 0$ is a constant (the **wave speed**) dependent on the material properties of the string. (Further discussion of the constant c is given in HELM 25.2.)

The wave equation is hyperbolic, as we can readily verify on recalling the definitions at the beginning of Section 32.4. Extra information is needed to specify the initial value problem. The initial position and initial velocity are given as

$$\left.\begin{array}{rcl} u(x,0) & = & f(x) \\ u_t(x,0) & = & g(x) \end{array}\right\} \quad 0 \leq x \leq \ell$$

Finally, we need boundary conditions specifying how the ends of the string are held. For example

$$u(0,t) = u(\ell,t) = 0 \qquad (t > 0)$$

models the situation where the string is fixed at each end.
(We will suppose that $f(0) = f(\ell) = 0$ so that there is no apparent conflict at the ends of the string at the initial time.)

2. Numerical solutions

The approach we will adopt is similar to that seen in Section 32.4 where we looked at parabolic equations. We use the notation

$$u_j^n$$

to denote an approximation to u evaluated at $x = j \times \delta x$, $t = n \times \delta t$. Approximating the derivatives in the PDE

$$u_{tt} = c^2 u_x$$

by central differences we obtain the numerical difference equation

$$\frac{u_j^{n+1} - 2u_j^n + u_j^{n-1}}{(\delta t)^2} = c^2 \, \frac{u_{j+1}^n - 2u_j^n + u_{j-1}^n}{(\delta x)^2}.$$

Multiplying through by $(\delta t)^2$ this can be rearranged to give

$$u_j^{n+1} = 2u_j^n - u_j^{n-1} + \mu^2(u_{j+1}^n - 2u_j^n + u_{j-1}^n)$$

in which $\mu = \dfrac{c\delta t}{\delta x}$ is called the **Courant number.**
The equation above gives u_j^{n+1} in terms of u-approximations at earlier time-steps (that is, all the appearances of u on the right-hand side have a superscript smaller than $n+1$).

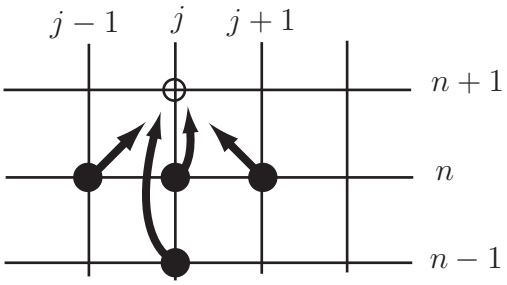


Figure 9

Thinking of the numerical stencil graphically we have the situation shown above. We may think of the values on the right-hand side of the equation "pointing to" a new value on the left-hand side.

Key Point 22

Timesteps (other than the first one) are carried out by using the numerical stencil

$$u_j^{n+1} = \underbrace{2u_j^n - u_j^{n-1} + \mu^2(u_{j+1}^n - 2u_j^n + u_{j-1}^n)}$$

↑ "new" approximation at $(n+1)^{\text{th}}$ time-step

↑ "old" approximations at earlier time-steps

(We will deal with how to carry out the *first* time-step shortly.)

The time-stepping process has much in common with the corresponding procedure for parabolic problems. The following Example will help establish the general idea.

Example 18

Given that $u = u(x,t)$ satisfies the wave equation $u_{tt} = c^2 u_{xx}$ in $t > 0$ and $0 < x < 1$ with boundary conditions $u(0,t) = u(1,t) = 0$ $(t > 0)$ with wave speed $c = 1.2$.
The numerical method $u_j^{n+1} = 2u_j^n - u_j^{n-1} + \mu^2(u_{j+1}^n - 2u_j^n + u_{j-1}^n)$ where $\mu = c\ \delta t/\delta x$, is implemented using $\delta x = 0.25$ and $\delta t = 0.1$.
Suppose that, after 5 time-steps, the following data forms part of the numerical solution:

$$\begin{array}{lll} u_0^4 = 0.0000 & & u_0^5 = 0.0000 \\ u_1^4 = 0.9242 & & u_1^5 = 0.7110 \\ u_2^4 = -0.0020 & & u_2^5 = -0.0059 \\ u_3^4 = -0.9624 & & u_3^5 = -0.7409 \\ u_4^4 = 0.0000 & & u_4^5 = 0.0000 \end{array}$$

Carry out the next time-step so as to find an approximation to u at $t = 6\delta t$.

Solution

In this case $\mu = 1.2 \times 0.1/0.25 = 0.48$ and the required time-step is carried out as follows:

$$\begin{aligned} u_0^6 &= 0 \quad \text{from the boundary condition} \\ u_1^6 = 2u_1^5 - u_1^4 + \mu^2(u_2^5 - 2u_1^5 + u_0^5) &= -0.1689 \\ u_2^6 = 2u_2^5 - u_2^4 + \mu^2(u_3^5 - 2u_2^5 + u_1^5) &= -0.0140 \\ u_3^6 = 2u_3^5 - u_3^4 + \mu^2(u_4^5 - 2u_3^5 + u_2^5) &= -0.1794 \\ u_4^6 &= 0 \quad \text{from the boundary condition} \end{aligned}$$

to 4 decimal places and these are the approximations to $u(0, 6\delta t)$, $u(0.25, 6\delta t)$, $u(0.5, 6\delta t)$, $u(0.75, 6\delta t)$ and $u(1, 6\delta t)$, respectively.

The diagram below shows the numerical results that appeared in the example above. It can be seen that the example was a (rather coarse) model of a standing wave with two antinodes.

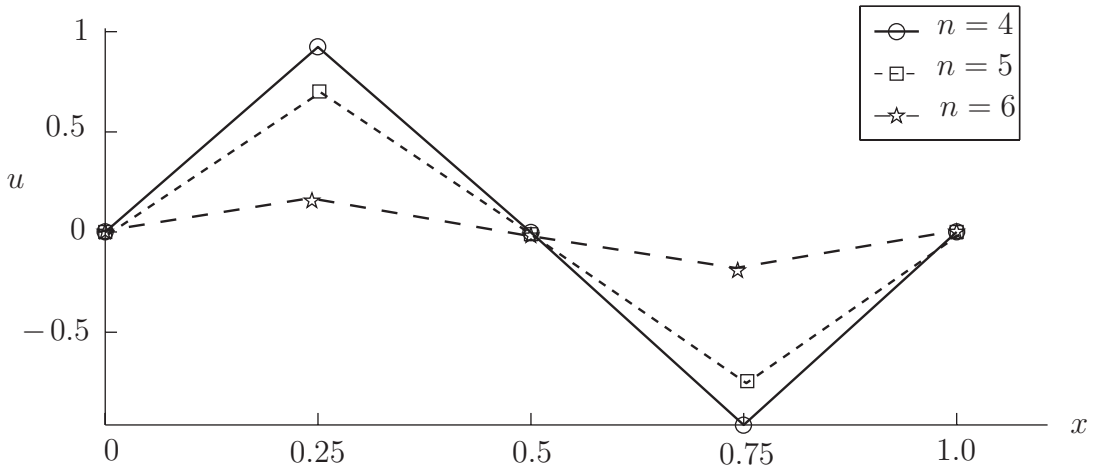


Figure 10

Task

Suppose that $u = u(x,t)$ satisfies the wave equation $u_{tt} = c^2 u_{xx}$ in $t > 0$ and $0 < x < 1$. It is given that u satisfies boundary conditions $u(0,t) = u(1,t) = 0$ $(t > 0)$ and initial conditions that need not be stated for the purposes of this question. The application is such that the wave speed $c = 1.2$.
The numerical method $u_j^{n+1} = 2u_j^n - u_j^{n-1} + \mu^2(u_{j+1}^n - 2u_j^n + u_{j-1}^n)$ where $\mu = c\ \delta t/\delta x$, is implemented using $\delta x = 0.25$ and $\delta t = 0.2$.
Suppose that, after 8 time-steps, the following data forms part of the numerical solution:

$$\begin{array}{lcllcl} u_0^7 &=& 0.0000 & u_0^8 &=& 0.0000 \\ u_1^7 &=& 0.6423 & u_1^8 &=& 0.4640 \\ u_2^7 &=& 0.8976 & u_2^8 &=& 0.6792 \\ u_3^7 &=& 0.6789 & u_3^8 &=& 0.4668 \\ u_4^7 &=& 0.0000 & u_4^8 &=& 0.0000 \end{array}$$

Carry out the next time-step so as to find an approximation to u at $t = 9\delta t$.

Your solution

Answer

In this case $\mu = 1.2 \times 0.2/0.25 = 0.96$ and the required time-step is carried out as follows:

$$\begin{aligned} u_0^9 &= 0 \quad \text{from the boundary condition} \\ u_1^9 = 2u_1^8 - u_1^7 + \mu^2(u_2^8 - 2u_1^8 + u_0^8) &= 0.0564 \\ u_2^9 = 2u_2^8 - u_2^7 + \mu^2(u_3^8 - 2u_2^8 + u_1^8) &= 0.0667 \\ u_3^9 = 2u_3^8 - u_3^7 + \mu^2(u_4^8 - 2u_3^8 + u_2^8) &= 0.0202 \\ u_4^9 &= 0 \quad \text{from the boundary condition} \end{aligned}$$

to 4 decimal places and these are the approximations to $u(0, 9\delta t)$, $u(0.25, 9\delta t)$, $u(0.5, 9\delta t)$, $u(0.75, 9\delta t)$ and $u(1, 9\delta t)$, respectively.

The above Task concerns a stretched string oscillating in such a way that at the 9th time-step the string is approximately flat. The motion continues with u taking negative values. Figure 11 below uses data calculated above, and also data for the next two time-steps so as to show subsequent progress of the solution.

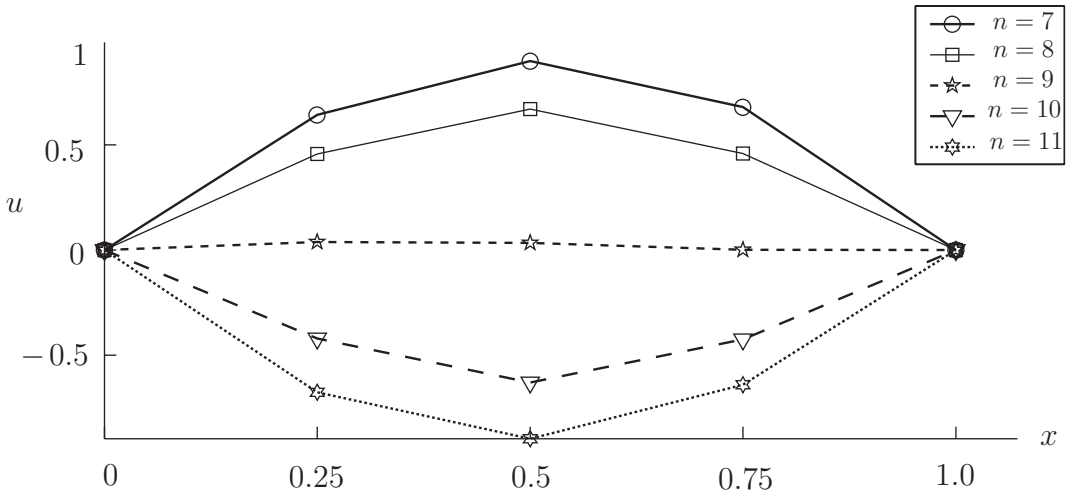


Figure 11

3. The first time-step

In the Example and Task above we have seen how time-steps can be carried out using the numerical stencil

$$u_j^{n+1} = 2u_j^n - u_j^{n-1} + \mu^2(u_{j+1}^n - 2u_j^n + u_{j-1}^n),$$

but there remains one issue which, so far, we have neglected. How do we carry out the *first* time-step?

Initial conditions

The initial time-step must use information from the two initial conditions

$$\left.\begin{array}{rcl} u(x,0) & = & f(x) \\ u_t(x,0) & = & g(x) \end{array}\right\} \quad 0 \leq x \leq \ell$$

The first initial condition is easy enough to interpret. It gives u_j^n in the case where $n = 0$. In fact

$$u_j^0 = f_j$$

where f_j is simply shorthand for $f(j \times \delta x)$.

The second initial condition, the one involving g, gives information about $u_t = \dfrac{\partial u}{\partial t}$ at $t = 0$. We can approximate the t-derivative of u at $t = 0$ and $x = j \times \delta x$ by a central difference to write

$$\frac{u_j^1 - u_j^{-1}}{2\delta t} = g_j$$

in which g_j is shorthand for $g(j \times \delta x)$.

This last expression involves u_j^{-1} which, if it has a meaning at all, refers to u at time $t = -\delta t$, that is, *before* the initial time $t = 0$. One way to think of u_j^{-1} is simply as an artificial quantity which proves useful later on. The equation above, rearranged for u_j^{-1} is

$$u_j^{-1} = u_j^1 - 2\delta t \times g_j$$

Key Point 23

A central difference used to approximate the first derivative in the condition defining initial speed gives rise to the following useful equation

$$u_j^{-1} = u_j^1 - 2\delta t \times g_j$$

The first time-step

To carry out the first time-step we put $n = 0$ in the numerical stencil

$$u_j^{n+1} = 2u_j^n - u_j^{n-1} + \mu^2\left(u_{j+1}^n - 2u_j^n + u_{j-1}^n\right),$$

to give

$$u_j^1 = 2u_j^0 - u_j^{-1} + \mu^2\left(u_{j+1}^0 - 2u_j^0 + u_{j-1}^0\right).$$

Those terms on the right-hand side with a 0 superscript are known *via* the function f since we know that $u_j^0 = f_j$. Hence

$$u_j^1 = 2f_j - u_j^{-1} + \mu^2\left(f_{j+1} - 2f_j + f_{j-1}\right).$$

And the u_j^{-1} term is dealt with using the Key Point above to give

$$u_j^1 = 2f_j - u_j^1 + 2\delta t \times g_j + \mu^2\left(f_{j+1} - 2f_j + f_{j-1}\right).$$

and therefore, moving the latest appearance of u_j^1 over to the left-hand side and dividing by 2,

$$\begin{aligned} u_j^1 &= f_j + \delta t \times g_j + \tfrac{1}{2}\mu^2\left(f_{j+1} - 2f_j + f_{j-1}\right) \\ &= \tfrac{1}{2}\mu^2\left(f_{j-1} + f_{j+1}\right) + (1-\mu^2)f_j + \delta t \times g_j \end{aligned}$$

Key Point 24

The first time-step is carried out by using the initial data and can be summarised as

$$u_j^1 = \tfrac{1}{2}\mu^2\left(f_{j-1} + f_{j+1}\right) + (1-\mu^2)f_j + \delta t \times g_j$$

Example 19

Suppose that $u = u(x,t)$ satisfies the wave equation $u_{tt} = c^2 u_{xx}$ in $t > 0$ and $0 < x < 1$. It is given that u satisfies boundary conditions $u(0,t) = u(1,t) = 0$ $(t > 0)$ and initial conditions that may be summarised as

$$\begin{array}{lcllcl} f_0 &=& 0.0000 & g_0 &=& 0.0000 \\ f_1 &=& 0.6000 & g_1 &=& 0.1000 \\ f_2 &=& 0.0000 & g_2 &=& 0.2000 \\ f_3 &=& -0.5000 & g_3 &=& 0.1000 \\ f_4 &=& 0.0000 & g_4 &=& 0.0000 \end{array}$$

The application is such that the wave speed $c = 1$.
Carry out the first two time-steps of the numerical method

$$u_j^{n+1} = 2u_j^n - u_j^{n-1} + \mu^2(u_{j+1}^n - 2u_j^n + u_{j-1}^n)$$

where $\mu = c\ \delta t/\delta x$ in which $\delta x = 0.25$ and $\delta t = 0.2$.

Solution

In this case $\mu = 1 \times 0.2/0.25 = 0.8$ and the first time-step is carried out as follows (to 4 d.p.):

$$\begin{array}{rcl} u_0^1 &=& 0 \quad \text{from the boundary condition} \\ u_1^1 = \frac{1}{2}\mu^2(f_0 + f_2) + (1-\mu^2)f_1 + \delta t g_1 &=& 0.2360 \\ u_2^1 = \frac{1}{2}\mu^2(f_1 + f_3) + (1-\mu^2)f_2 + \delta t g_2 &=& 0.0720 \\ u_3^1 = \frac{1}{2}\mu^2(f_2 + f_4) + (1-\mu^2)f_3 + \delta t g_3 &=& -0.0160 \\ u_4^1 &=& 0 \quad \text{from the boundary condition} \end{array}$$

The second time-step is as follows (to 4 d.p.):

$$\begin{array}{rcl} u_0^2 &=& 0 \quad \text{from the boundary condition} \\ u_1^2 = 2u_1^1 - u_1^0 + \mu^2(u_2^1 - 2u_1^1 + u_0^1) &=& -0.3840 \\ u_2^2 = 2u_2^1 - u_2^0 + \mu^2(u_3^1 - 2u_2^1 + u_1^1) &=& 0.1005 \\ u_3^2 = 2u_3^1 - u_3^0 + \mu^2(u_4^1 - 2u_3^1 + u_2^1) &=& 0.4309 \\ u_4^2 &=& 0 \quad \text{from the boundary condition} \end{array}$$

Suppose that $u = u(x,t)$ satisfies the wave equation $u_{tt} = c^2 u_{xx}$ in $t > 0$ and $0 < x < 0.8$. It is given that u satisfies boundary conditions $u(0,t) = u(0.8,t) = 0$ $(t > 0)$ and initial conditions that may be summarised as

$$\begin{array}{lcllcl} f_0 &=& 0.0000 & g_0 &=& 0.0000 \\ f_1 &=& 0.1703 & g_1 &=& 0.4227 \\ f_2 &=& 0.2364 & g_2 &=& 0.5417 \\ f_3 &=& 0.1703 & g_3 &=& 0.4227 \\ f_4 &=& 0.0000 & g_4 &=& 0.0000 \end{array}$$

The application is such that the wave speed $c = 1$.
Carry out the first two time-steps of the numerical method

$$u_j^{n+1} = 2u_j^n - u_j^{n-1} + \mu^2(u_{j+1}^n - 2u_j^n + u_{j-1}^n)$$

where $\mu = c\ \delta t/\delta x$ in which $\delta x = 0.2$ and $\delta t = 0.11$.

Your solution

Answer
In this case $\mu = 1 \times 0.11/0.2 = 0.55$ and the first time-step is carried out as follows:

$$\begin{aligned} u_0^1 &= 0 \quad \text{from the boundary condition} \\ u_1^1 = \tfrac{1}{2}\mu^2(f_0 + f_2) + (1-\mu^2)f_1 + \delta t g_1 &= 0.2010 \\ u_2^1 = \tfrac{1}{2}\mu^2(f_1 + f_3) + (1-\mu^2)f_2 + \delta t g_2 &= 0.2760 \\ u_3^1 = \tfrac{1}{2}\mu^2(f_2 + f_4) + (1-\mu^2)f_3 + \delta t g_3 &= 0.2010 \\ u_4^1 &= 0 \quad \text{from the boundary condition} \end{aligned}$$

The second time-step is as follows:

$$\begin{aligned} u_0^2 &= 0 \quad \text{from the boundary condition} \\ u_1^2 = 2u_1^1 - u_1^0 + \mu^2(u_2^1 - 2u_1^1 + u_0^1) &= 0.1936 \\ u_2^2 = 2u_2^1 - u_2^0 + \mu^2(u_3^1 - 2u_2^1 + u_1^1) &= 0.2702 \\ u_3^2 = 2u_3^1 - u_3^0 + \mu^2(u_4^1 - 2u_3^1 + u_2^1) &= 0.1936 \\ u_4^2 &= 0 \quad \text{from the boundary condition} \end{aligned}$$

4. Stability

There is a stability constraint that is common to many methods for obtaining numerical solutions of the wave equation. Issues relating to stability of numerical methods can be extremely complicated, but the following Key Point is enough for our purposes.

Key Point 25

The numerical method seen in this Section requires that

$$\mu \leq 1 \qquad \text{that is,} \quad \frac{c\delta t}{\delta x} \leq 1$$

for solutions not to grow unrealistically with n.

This is called the CFL condition (named after an acronym of three mathematicians Courant, Friedrichs and Lewy).

Exercises

1. Suppose that $u = u(x,t)$ satisfies the wave equation $u_{tt} = c^2 u_{xx}$ in $t > 0$ and $0 < x < 0.6$. It is given that u satisfies boundary conditions $u(0,t) = u(0.6,t) = 0$ $(t > 0)$ and initial conditions that need not be stated for the purposes of this question. The application is such that the wave speed $c = 1.4$.
   The numerical method

   $$u_j^{n+1} = 2u_j^n - u_j^{n+1} + \mu^2(u_{j+1}^n - 2u_j^n + u_{j-1}^n)$$

   where $\mu = c\ \delta t/\delta x$, is implemented using $\delta x = 0.15$ and $\delta t = 0.1$.
   Suppose that, after 7 time-steps, the following data forms part of the numerical solution:

   $$\begin{array}{lcllcl} u_0^6 &=& 0.0000 & u_0^7 &=& 0.0000 \\ u_1^6 &=& 0.1024 & u_1^7 &=& 0.0997 \\ u_2^6 &=& 0.1986 & u_2^7 &=& 0.1730 \\ u_3^6 &=& 0.2361 & u_3^7 &=& 0.1169 \\ u_4^6 &=& 0.0000 & u_4^7 &=& 0.0000 \end{array}$$

   Carry out the next time-step so as to find an approximation to u at $t = 8\delta t$.

2. Suppose that $u = u(x,t)$ satisfies the wave equation $u_{tt} = c^2 u_{xx}$ in $t > 0$ and $0 < x < 1$. It is given that u satisfies boundary conditions $u(0,t) = u(1,t) = 0$ $(t > 0)$. The initial elevation may be summarised as

   $$\begin{array}{lcllcllcl} f_0 &=& 0.0000 & f_1 &=& 0.7812 & f_2 &=& 0.2465 \\ f_3 &=& -0.1209 & f_4 &=& 0.0000 & & & \end{array}$$

   and the string is initially at rest (that is, $g(x) = 0$). The application is such that the wave speed $c = 1$.
   Carry out the first two time-steps of the numerical method

   $$u_j^{n+1} = 2u_j^n - u_j^{n-1} + \mu^2(u_{j+1}^n - 2u_j^n + u_{j-1}^n)$$

   where $\mu = c\ \delta t/\delta x$ in which $\delta x = 0.25$ and $\delta t = 0.2$.

Answers

1. In this case $\mu = 1.4 \times 0.1/0.15 = 0.93333$ and the required time-step is carried out as follows:

$$\begin{aligned} u_0^8 &= 0 \quad \text{from the boundary condition} \\ u_1^8 = 2u_1^7 - u_1^6 + \mu^2(u_2^7 - 2u_1^7 + u_0^7) &= 0.0740 \\ u_2^8 = 2u_2^7 - u_2^6 + \mu^2(u_3^7 - 2u_2^7 + u_1^7) &= 0.0347 \\ u_3^8 = 2u_3^7 - u_3^6 + \mu^2(u_4^7 - 2u_3^7 + u_2^7) &= -0.0552 \\ u_4^8 &= 0 \quad \text{from the boundary condition} \end{aligned}$$

to 4 decimal places and these are the approximations to $u(0, 8\delta t)$, $u(0.15, 8\delta t)$, $u(0.3, 8\delta t)$, $u(0.45, 8\delta t)$ and $u(0.6, 8\delta t)$, respectively.

2. In this case $\mu = 1 \times 0.2/0.25 = 0.8$ and the first time-step is carried out as follows:

$$\begin{aligned} u_0^1 &= 0 \quad \text{from the boundary condition} \\ u_1^1 = \tfrac{1}{2}\mu^2(f_0 + f_2) + (1 - \mu^2)f_1 + \delta t g_1 &= 0.3601 \\ u_2^1 = \tfrac{1}{2}\mu^2(f_1 + f_3) + (1 - \mu^2)f_2 + \delta t g_2 &= 0.3000 \\ u_3^1 = \tfrac{1}{2}\mu^2(f_2 + f_4) + (1 - \mu^2)f_3 + \delta t g_3 &= 0.0354 \\ u_4^1 &= 0 \quad \text{from the boundary condition} \end{aligned}$$

to 4 decimal places.

The second time-step is as follows:

$$\begin{aligned} u_0^2 &= 0 \quad \text{from the boundary condition} \\ u_1^2 = 2u_1^1 - u_1^0 + \mu^2(u_2^1 - 2u_1^1 + u_0^1) &= -0.3299 \\ u_2^2 = 2u_2^1 - u_2^0 + \mu^2(u_3^1 - 2u_2^1 + u_1^1) &= 0.2226 \\ u_3^2 = 2u_3^1 - u_3^0 + \mu^2(u_4^1 - 2u_3^1 + u_2^1) &= 0.3384 \\ u_4^2 &= 0 \quad \text{from the boundary condition} \end{aligned}$$

to 4 decimal places.

Index for Workbook 32

Workbook 32

HELM®

HELM: Helping Engineers Learn Mathematics

http://helm.lboro.ac.uk