

Physical Processes in the Geosciences

LAST REVISED
September 25, 2020

Joe LaCasce
Department for Geosciences
University of Oslo
P.O. Box 1022 Blindern
0315 Oslo, Norway
j.h.lacasce@geo.uio.no

Contents

1	Introduction	7
2	Math review	11
2.1	Functions	11
2.2	Derivatives	13
2.3	Ordinary Differential Equations	14
2.4	Partial differential equations	17
2.5	Linear algebra	19
3	Difference equations	23
3.1	Finite differences	23
3.2	Integration schemes	27
3.2.1	FTCS	27
3.2.2	Leapfrog scheme	28
3.2.3	Runge-Kutta methods	29
3.2.4	iFTCS	31
3.2.5	The Crank-Nicholson (CN) scheme	33
3.3	Summary	34
4	Time dependent systems	37
4.1	Single volume systems	37
4.1.1	Radioactive decay	38
4.1.2	Time dependent forcing	41
4.1.3	Numerical solutions	43
4.2	Multiple volumes	44
4.2.1	Analytical solution	46
4.2.2	Numerical solutions	50
4.3	A two mass model	51
4.4	A simple climate model	54
4.4.1	Black body radiation	55
4.4.2	A latitudinally-varying model	60
4.5	Summary	66
5	Advection/transport	69

5.1	The advection equation	70
5.1.1	Eulerian and Lagrangian formulations	74
5.1.2	Advection with a constant velocity	75
5.1.3	Wave equation	78
5.1.4	Numerical solutions	79
5.1.5	Advection with a variable velocity	86
5.2	Summary	89
6	Diffusion	91
6.1	The diffusion equation	91
6.2	Analytical solutions	94
6.2.1	Self-similar solution	96
6.2.2	A point source: an oceanic spill	99
6.2.3	A boundary source: flow in an aquifer	103
6.2.4	Erosion	108
6.3	Numerical solutions	112
6.3.1	Forward Euler	113
6.3.2	Implicit Euler	117
6.3.3	Initial point source	119
6.3.4	A diffusive climate model	124
6.4	Summary	129
7	Advection-Diffusion	131
7.1	Advection-diffusion equation	132
7.2	Analytical solution: point source	135
7.3	Numerical solutions	138
7.3.1	Non-constant diffusivity	142
7.3.2	The climate model with storm tracks	145
7.4	Summary	149
8	Momentum Advection	151
8.1	The momentum equation	152
8.2	Burger's equation	155
8.3	Analytical solutions	158
8.3.1	Solution for a point source	161
8.4	Numerical solutions	163
8.4.1	FTCS	163
8.4.2	iFTCS	166
8.4.3	RK2	168
8.5	Summary	170
9	Shallow water flows	173
9.1	Navier-Stokes Equations	174
9.1.1	Gravity	174

9.1.2	Pressure gradients	174
9.2	Shallow water equations	177
9.2.1	Incompressibility	178
9.2.2	Hydrostatic balance	179
9.2.3	Linear shallow water equations	180
9.3	Surface waves	183
9.3.1	Analytical solution (1D)	184
9.3.2	Numerical solution (1D)	187
9.4	Summary	191
10	Appendix A: Diffusion example with Python	193
11	Appendix B: Numerical stability	199

Chapter 1

Introduction

The Earth has a wide range of physical systems, displaying a wealth of complex phenomena. These phenomena can greatly impact human life as well. So there is a pressing need to be able to study, understand and model them.

A few examples of these phenomena are:

- **Climate models**, to understand changes due to excess atmospheric CO₂
- **Volcanoes**, such as the Öræfajökull volcano on Iceland, which erupted in 2010 and shut down the European airspace
- **Meteor impacts**, such as the Chicxulub meteor which struck off the coast of Mexico and wiped out the dinosaurs 66 million years ago
- **Glaciers**, whose increased melting under global warming is leading to significant sea level rise
- **Hurricanes**, which form over the ocean and then impact land, causing billions of dollars in damage every year
- **Tectonic plates**, whose motion and collisions produce the mountain

ranges

- **Pollutants**, such as the oil spilled in the Deepwater Horizon accident in the Gulf of Mexico in 2011.

While each exhibits very complex dynamics, they have common features.

In this course, we will study the physical processes which are common across the geosciences. These processes include the conservation of mass, advection and diffusion. We will develop simple models, to see how these processes function. The initial models are highly simplified (for example, on an infinite plane). But using the same concepts, we increase the complexity, to approach that of the modelled system.

If the model is simple enough, one can find *analytical solutions*. The advantage of such a mathematical solution is that you can see how the system evolves with different choices of parameters. No lengthy computer calculations are required! The disadvantage is that the solution only works in very idealized cases, for instance an oil spill in a homogeneous, infinite aquifer.

Once we have derived an analytical solution, we will write a *numerical code* to solve the same problem. In the course we will do this in Python, although any other language will do. The first step is then to *test the code against the analytical solution*. So we run the model to simulate the oil spill mentioned above and compare the evolving distribution at different times with that from the analytical model. This should be done *extensively*, until you are convinced the model is behaving correctly. It is common to have small bugs in the code which make the numerical solutions incorrect. The numerical codes can also be used to test for errors in the analytical

solutions, which can be quite complex. When the models agree, you have confidence that they are probably both correct.

Once the model works for the simplified case, we can use it to study more complicated situations (e.g. a distributed oil leak in a inhomogeneous, three dimensional aquifer). This is the great advantage of the numerical models, that they have much greater flexibility than the analytical solutions. However, numerical codes have limitations, in particular *resolution* and *numerical stability*. We'll learn more about these hereafter.

In the rest of this chapter, we review basic concepts in math and programming which will be useful later on. Hopefully you will have seen some of these before, but we will become more familiar with the techniques as we use them in the course.

Chapter 2

Math review

The following are concepts which we will use in developing our analytical solutions. This introduction is very superficial; for more details, you should consult math textbooks. But these concepts are central to the subsequent development.

2.1 Functions

Much analysis is based on *functions*. A function is like a machine, converting *input* variables to *output*. A function of one variable can be written $y = f(x)$, where x is the *independent variable*, y is the *dependent variable* and f is the function which converts x to y . An example is:

$$y = x^2 \tag{2.1}$$

This is plotted in Fig. (2.1).

The variable x has a *range* from -2 to 2, while the resulting y ranges from 0 to 4. We see that y has a minimum at $x = 0$ and increases for both negative and positive x . This is a simple two dimensional (2D) system, because we have two variables total.

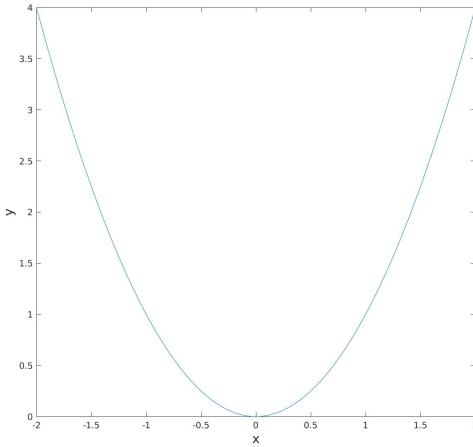


Figure 2.1: The function $y = x^2$.

We can also define functions of multiple variables, for example:

$$z = f(x, y) = x^2 + y^2 - 2 \quad (2.2)$$

This is shown in Fig. (2.2). Now both x and y are independent variables

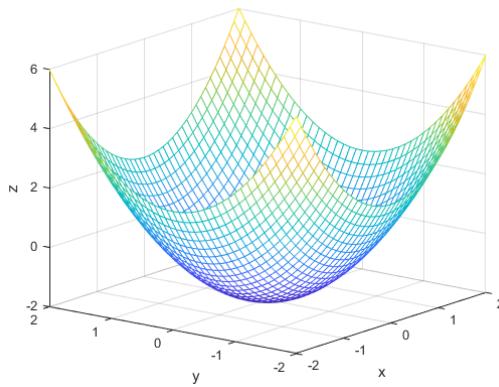


Figure 2.2: The function $z = x^2 + y^2 - 2$.

and z is the dependent variable. This is a three dimensional (3D) system, because we have three variables. Here, x and y both range from -2 to 2 , while z ranges from -2 to 6 .

Higher dimensions are of course possible. But these are much harder to visualize! We will be mostly concerned with 2D systems, to keep things

simple. But the same approaches that we use will be applicable to three and higher dimensions as well.

2.2 Derivatives

Once we have a function, we can consider how it changes as the independent variables change. This is a central concept in calculus. We quantify this change with *derivatives*. The derivative is essentially the tangent to a curve or surface at a given point.

For the function $y = x^2$, the derivative is given by:

$$\frac{dy}{dx} = 2x \quad (2.3)$$

This is positive for positive x and negative for negative x . The slope is exactly zero at $x = 0$, meaning the function doesn't change going from just below zero to just above zero.

Note too that the slope changes x . Thus the slope is greater at $x = 4$ than at $x = 2$. When the slope changes like this, we say that the function is *nonlinear*. The function $y = 2x$ on the other hand is *linear*, because the derivative (2) is constant for all x .

One can also examine higher order derivatives. The second derivative is one that we'll use frequently hereafter. The second derivative measures the curvature of a function. For the function $y = x^2$, we have:

$$\frac{d^2y}{dx^2} = 2 \quad (2.4)$$

which is constant. So even though the slope changes for all x , the rate that it changes is constant.

One can also discuss *extreme values* of a function, i.e. the maximum and minimum values exhibited. As seen in Fig. (2.1), the function $y = x^2$ has a single minimum, at $x = 0$. As we have seen, the first derivative ($2x$) vanishes at $x = 0$. Indeed, this is how we look for extreme values, by finding where the first derivative vanishes. The same concept applies to functions of multiple variables. The function $z = x^2 + y^2 - 2$ has a minimum at $x = y = 0$, which is where the derivatives with respect to x and y both vanish.

The vanishing of the first derivative indicates an extreme value, but how do we know whether this is a minimum or a maximum? This is determined by the second derivative. If the curvature at the extremum is positive, the function increases on either side, indicating the value is a minimum. If the curvature is negative, the function decreases and the extremum is a maximum. Extreme values are very useful in physics. We'll see an example later where we use an extremum to relate the color of a star to its surface temperature.

2.3 Ordinary Differential Equations

When a derivative appears in an equation, it is known as a *differential equation*. When there is only one independent variable, this is an *ordinary differential equation* (ODE). An example is:

$$m \frac{d^2x}{dt^2} + kx = 0 \quad (2.5)$$

This describes the motion of a mass, m , connected to a spring. The position, $x(t)$, is a function of time only. The equation involves the second derivative, $\frac{d^2x}{dt^2}$, so we call this a “second order” equation. Moreover, since

there is no forcing term on the right hand side, this is called a *homogeneous equation*.

The equation has a general solution:

$$x = A \cos(\omega t) + B \sin(\omega t) \quad (2.6)$$

where $\omega = \sqrt{k/m}$ is the frequency of the mass-spring system. You can confirm this by substituting the solution into the equation. A and B are two unknown amplitudes. This is typically the case; a second order homogeneous equation has two unknowns.

To determine the unknowns, we require *initial conditions*. For instance, with the initial position, $x(0) = x_0$, and the initial velocity, $dx(0)/dt = u_0$, we can find A and B :

$$A = x_0, \quad B = \frac{u_0}{\omega} \quad (2.7)$$

Initial conditions are useful when you have an ODE with time as the independent variable.

If instead x is the independent variable, we usually require *boundary conditions*. Consider the second order ODE:

$$\frac{d^2y}{dx^2} = 1 \quad (2.8)$$

This is an *inhomogeneous* ODE, because of the 1 on the right hand side. The general solution for this is:

$$y = \frac{1}{2}x^2 + Ax + B \quad (2.9)$$

To determine the constants, A and B , we need information about y at certain locations. Say we knew:

$$y(0) = 3, \quad y(2) = -1 \quad (2.10)$$

then we would find $B = 3$, $A = -3$. This is an example of *Dirichlet* boundary conditions, when the values of the function are specified on the boundary.

Alternately we could specify the derivatives, for example:

$$\frac{dy}{dx}(0) = 0, \quad y(2) = 0 \quad (2.11)$$

The first is known as a *Neumann* boundary condition. Applying these conditions, we find $A = 0$ and $B = -2$.

As will be seen later, the Neumann condition is related to the *flux* at a boundary. Here there is no flux of y at $x = 0$ or at $x = 2$. This implies the total y , integrated over the domain is *conserved*, because there is no “leakage” of y at the ends of the domain. This is an important concept that we’ll see again.

It is also possible to have a combination of derivatives and values, such as:

$$\frac{dy}{dx}(0) = 2(x + 1) \quad (2.12)$$

This is known as a “Robin” condition. This is much less common though than the Dirichlet and Neumann conditions, and we won’t consider it further in the course.

Equations (2.5) and (2.8) are *linear* ODEs. This is because only linear functions of the dependent variable, y , appear. Another example is:

$$\frac{du}{dt} + Au = F \quad (2.13)$$

This is an inhomogeneous, first order ODE. It is linear because both terms on the LHS are linear in u .

When an equation has powers or functions of the dependent variable, it is a *nonlinear* equation. Some examples are:

$$\frac{du}{dt} + Au^2 = F \quad (2.14)$$

and:

$$u \frac{du}{dt} + Au = F \quad (2.15)$$

and:

$$\frac{du}{dt} + Ae^u = F \quad (2.16)$$

Nonlinear equations are usually harder to solve than linear equations. Indeed, with many nonlinear equations, it isn't possible to find an analytical solution at all. In such cases, numerical solutions are required.

2.4 Partial differential equations

Usually, we work with functions with multiple independent variables. An example is $z(x, y)$, shown previously in Fig. (2.2). Functions of multiple variables are common in the geosciences. The temperature in a volcano varies in the three spatial directions, (x, y, z) and in time, t . The salt in the ocean is a function of the same four independent variables.

When such a function appears in an equation, we can have derivatives with respect to all the dependent variables, such as:

$$\frac{\partial \rho}{\partial x}, \frac{\partial^2 \rho}{\partial y^2}$$

So an equation might look like:

$$\frac{\partial^2 \rho}{\partial x^2} + \frac{\partial^2 \rho}{\partial y^2} = 0 \quad (2.17)$$

This is an example of a *partial differential equation* (PDE). PDEs are central in the geosciences. Our weather models, ocean models, volcano models, etc. are all based on PDEs.

PDEs are more complicated than ODEs, but the same concepts apply. Solving PDEs requires boundary and (if time is involved) initial conditions. In the equation above, we could have Dirichlet and/or Neumann conditions in x and in y . And we can have linear and nonlinear PDEs as well.

When multiple dependent variables appear in the same equations, the equations are *coupled*. For example:

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 \quad (2.18)$$

involves two dependent variables, u and v .

If there are N dependent variables, we require N equations to determine them. The equation above gives us a relation between u and v , but the values are not determined uniquely from this alone. We require an additional equation to determine them.

An important example of coupled PDEs are the *Navier-Stokes equations*. These are central equations to weather forecasting. The equations are nonlinear and have no known analytical solutions. Indeed, there is a one million dollar prize on offer for anyone who can demonstrate that such solutions even exist!¹

Linear PDEs are usually classified, as follows. Consider the following generic linear PDE:

$$A \frac{\partial^2 u}{\partial x^2} + B \frac{\partial^2 u}{\partial x \partial y} + C \frac{\partial^2 u}{\partial y^2} + D \frac{\partial u}{\partial x} + E \frac{\partial u}{\partial y} + F = 0 \quad (2.19)$$

¹See “Millennium prize problems” in Wikipedia.

If $B^2 - 4AC < 0$, the equation is *elliptic*. An example of an elliptic equation is Laplace's equation:

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

This is a familiar equation in fluid flow, electrostatics and in gravitation.

If $B^2 - 4AC > 0$, the equation is *hyperbolic*. The most common example is the wave equation, for example:

$$\frac{\partial^2 \phi}{\partial t^2} - c^2 \frac{\partial^2 \phi}{\partial x^2} = 0$$

A range of different waves are found in the atmosphere, the ocean, in lakes and rivers, and even in the solid earth.

Lastly, if $B^2 - 4AC = 0$, the equation is *parabolic*. The “diffusion equation” is parabolic:

$$\frac{\partial \phi}{\partial t} = \kappa \left(\frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} \right)$$

The diffusion equation can be used to model the spreading of heat and also of pollutants.

Many geophysical processes can be described by precisely these equations. We'll look at analytical and numerical solutions of equations like these. In so doing, you'll develop intuition about how they work.

2.5 Linear algebra

When we convert equations like the diffusion equation to numerical form, we usually obtain a large set of *coupled equations*. Linear algebra is a branch of math developed specifically for dealing with coupled linear equations. A central concept in linear algebra is the *matrix*, which is an array

of numbers. A matrix might be written like this:

$$A = \begin{bmatrix} A_{11} & A_{12} & A_{13} & A_{14} & A_{15} \\ A_{21} & A_{22} & A_{23} & A_{24} & A_{25} \\ A_{31} & A_{32} & A_{33} & A_{34} & A_{35} \\ A_{41} & A_{42} & A_{43} & A_{44} & A_{45} \\ A_{51} & A_{52} & A_{53} & A_{54} & A_{55} \end{bmatrix}$$

This is a 5×5 matrix, with 25 elements. The matrix has *5 rows* (the horizontal lines) and *5 columns* (the vertical lines). The elements with equal indices (A_{11} , A_{22} , etc.) make up the *diagonal* of the matrix.

If we have two matrices, we can multiply them, for example:

$$\begin{bmatrix} A_{11} & A_{12} & A_{13} & A_{14} & A_{15} \\ A_{21} & A_{22} & A_{23} & A_{24} & A_{25} \\ A_{31} & A_{32} & A_{33} & A_{34} & A_{35} \\ A_{41} & A_{42} & A_{43} & A_{44} & A_{45} \\ A_{51} & A_{52} & A_{53} & A_{54} & A_{55} \end{bmatrix} \begin{bmatrix} B_{11} \\ B_{21} \\ B_{31} \\ B_{41} \\ B_{51} \end{bmatrix} = \begin{bmatrix} C_{11} \\ C_{21} \\ C_{31} \\ C_{41} \\ C_{51} \end{bmatrix} \quad (2.20)$$

When multiplying a (5×5) matrix, we require a matrix (or vector) which has $(5 \times n)$ elements. In this example, multiplying a (5×5) matrix with a (5×1) matrix yields another (5×1) matrix.

For example, consider:

$$\begin{bmatrix} 3 & -1 & 2 \\ -1 & 0 & 4 \\ 2 & 1 & 2 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} = \begin{bmatrix} 7 \\ 11 \\ -2 \end{bmatrix} \quad (2.21)$$

We get this by multiplying the (3×1) vector with each row of the (3×3) matrix. The first element on the RHS (7) is obtained by multiplying sequentially the terms in the top row of the first matrix by the elements in the column vector. So:

$$3 * 1 - 1 * 2 + 2 * 3 = 7$$

This is repeated with the second row, etc. to obtain the vector on the right.

To illustrate why this is so useful, consider the system of equations:

$$2x + 3y + 3z = 3 \quad (2.22)$$

$$-x + 2y - z = 2 \quad (2.23)$$

$$3x + 3z = 1 \quad (2.24)$$

This is a coupled set of three equations for the three variables, (x, y, z) .

We can write this as a matrix equation:

$$\begin{bmatrix} 2 & 3 & 3 \\ -1 & 2 & -1 \\ 3 & 0 & 3 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 3 \\ 2 \\ 1 \end{bmatrix} \quad (2.25)$$

We can express this in compact form thus:

$$\mathbf{A}\mathbf{x} = \mathbf{b} \quad (2.26)$$

We can then solve for \mathbf{x} by writing:

$$\mathbf{x} = \mathbf{A}^{-1}\mathbf{b} \quad (2.27)$$

Here \mathbf{A}^{-1} is the *inverse* of \mathbf{A} . The inverse is defined such that:

$$\mathbf{A}^{-1}\mathbf{A} = \mathbf{I} \quad (2.28)$$

where:

$$\mathbf{I} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.29)$$

is the *identity matrix* (the 3x3 version, in this case).

The inverse of a matrix can be calculated thus:

$$\mathbf{A}^{-1} = \frac{\text{adj}(\mathbf{A})}{\det(\mathbf{A})} \quad (2.30)$$

where \mathbf{A}^T is the matrix *adjugate* and where $\det(\mathbf{A})$ is its *determinant*. For a 2x2 matrix:

$$\mathbf{A} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

the adjugate is:

$$\text{adj}(\mathbf{A}) = \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$$

The determinant on the other hand is the difference of the products along the diagonals:

$$\det(\mathbf{A}) = ad - bc$$

Multiplying the inverse by the original matrix, we get:

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \frac{1}{ad - bc} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix} = \frac{1}{ad - bc} \begin{bmatrix} ad - bc & 0 \\ 0 & ad - bc \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

This procedure is easily done by hand for 2x2 matrices, but quickly gets more cumbersome for larger matrices. However, the inverse is easily found using Matlab, Python or Julia, which have built-in linear algebra functionality. For the 3x3 example given above, the inverse of \mathbf{A} is found to be:

$$\mathbf{A}^{-1} = \begin{bmatrix} -1 & 1/2 & 7/6 \\ 0 & 1/2 & 1/6 \\ 1 & -1/2 & -5/6 \end{bmatrix}$$

So we obtain the solution:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} -1 & 1/2 & 7/6 \\ 0 & 1/2 & 1/6 \\ 1 & -1/2 & -5/6 \end{bmatrix} \begin{bmatrix} 3 \\ 2 \\ 1 \end{bmatrix} = \begin{bmatrix} -5/6 \\ 7/6 \\ 7/6 \end{bmatrix} \quad (2.31)$$

We use matrices extensively when we write our numerical models hereafter. These are based on *difference equations*, discussed in the next chapter.

Chapter 3

Difference equations

3.1 Finite differences

We will be solving PDEs like this:

$$\frac{\partial}{\partial t}\phi = \kappa \frac{\partial^2}{\partial x^2}\phi \quad (3.1)$$

numerically. How do we do this?

The central idea in numerical solutions is that derivatives are replaced by *finite differences*. These occur both in space (x) and time (t). For both space and time, we assume we have a set of discrete points, usually separated by small amounts. For example:

$$x = [0 \ 0.1 \ 0.2 \ 0.3 \ 0.4 \ 0.5 \ 0.6 \ 0.7 \ 0.8 \ 0.9 \ 1.0]$$

This constitutes a *grid* of values. Here x ranges from 0 to 1, with a step size of $dx = 0.1$.

The grid can be uniform (with the same step size, dx) or non-uniform (with different grid spacings). Uniform grids are very common and used in basic applications. Non-uniform grids are used when you need to resolve rapid changes, for instance near a boundary in a flow. We'll use uniform grids hereafter, but the ideas are easily generalized to non-uniform grids.

At each grid point, we have a value of the dependent variable, such as $\phi(x, t)$ in the example above. When we run the model, we obtain new values of ϕ at the grid points.

To solve the equation, we require derivatives. We obtain these by using *Taylor series expansions*. For instance:

$$\phi(x + dx) = \phi(x) + \frac{\partial\phi}{\partial x}dx + \frac{1}{2}\frac{\partial^2\phi}{\partial x^2}dx^2 + O|dx^3| \quad (3.2)$$

where the expression $O|dx^3|$ means that the remaining terms are at most proportional to dx^3 . This approximates $\phi(x + dx)$ by quantities defined at x . The Taylor Series is accurate if dx is small. Then the terms multiplied by higher powers of dx are very small.

Say for example, $dx = 0.1$. Then we could have something like:

$$\phi(0.6) = \phi(0.5) + \frac{\partial\phi}{\partial x}(0.1) + \frac{1}{2}\frac{\partial^2\phi}{\partial x^2}(0.1)^2 + O|(0.1)^3| \quad (3.3)$$

Note the derivatives are evaluated at $x = 0.5$. Because $dx = 0.1$, the higher order terms are vanishingly small and we can ignore them.

Going backwards, we can also write:

$$\phi(x - dx) = \phi(x) - \frac{\partial\phi}{\partial x}dx + \frac{1}{2}\frac{\partial^2\phi}{\partial x^2}dx^2 + O|dx^3| \quad (3.4)$$

The sign of the second term on the RHS is negative because dx is negative.

For the grid above, we would have:

$$\phi(0.4) = \phi(0.5) - \frac{\partial\phi}{\partial x}(0.1) + \frac{1}{2}\frac{\partial^2\phi}{\partial x^2}(0.1)^2 + O|(0.1)^3| \quad (3.5)$$

Now, if we take the difference of (3.2) and (3.4), we get:

$$\phi(x + dx) - \phi(x - dx) = 2\frac{\partial\phi}{\partial x}dx + O|dx^3|$$

Note that both the $\phi(x)$ and the second derivative terms have vanished.

Solving this for the derivative, we get:

$$\frac{\partial \phi}{\partial x} = \frac{\phi(x + dx) - \phi(x - dx)}{2dx} + O|dx^2| \quad (3.6)$$

This is the second order center difference form of the derivative. It is “centered” because it involves the values of ϕ on either side of (x) . The expression is second order accurate because the excluded terms are $O|dx^2|$ (because we neglected terms of size dx^3 , but then divided by dx).

We could obtain more accuracy if we save more terms in the Taylor series expansion. For instance, if we exclude terms of $O|dx^5|$, we would have obtained the four order center difference, accurate to $O|dx^4|$. But the second order expression will suffice for the examples in this course.

What happens at the boundaries? If we want the centered difference at $x = 0$, we would have to use the value at $x = -0.1$, which doesn’t exist. So instead we use the following:

$$\phi(x + dx) - \phi(x) = \frac{\partial \phi}{\partial x} dx + O|dx^2|$$

or:

$$\frac{\partial \phi}{\partial x} = \frac{\phi(x + dx) - \phi(x)}{dx} + O|dx| \quad (3.7)$$

This is the forward difference version of the derivative. It is only first order accurate, because the excluded terms are $O|dx|$.

Likewise, we can’t use the forward or centered difference at $x = 1$, because we don’t have $\phi(1.1)$. So instead we use the backward difference:

$$\frac{\partial \phi}{\partial x} = \frac{\phi(x) - \phi(x - dx)}{dx} + O|dx| \quad (3.8)$$

This also is first order accurate. The Python function “numpy.gradient” uses centered differences for the interior points, but a forward difference

for the first two elements of x and a backward difference for the last two.

Using the Taylor series, we can also obtain an expression for the second derivative. First we add the expressions:

$$\phi(x + dx) + \phi(x - dx) = 2\phi(x) + \frac{\partial^2 \phi}{\partial x^2} dx^2 + O|dx^3|$$

Then we solve for the second derivative:

$$\frac{\partial^2 \phi}{\partial x^2} = \frac{\phi(x + dx) - 2\phi(x) + \phi(x - dx)}{dx^2} + O|dx^2| \quad (3.9)$$

Like the centered difference derivative (3.6), this is second order accurate because the excluded terms are $O|dx^2|$. Now we have a finite difference expression for the RHS of our equation (3.1).

It is convenient to use a short-hand notation for these terms. We label the x values by the index i . So $i = 1$ is the first value (0), $i = 2$ is the second value (0.1), and so forth. Then the second derivative can be written:

$$\frac{\partial^2 \phi}{\partial x^2} = \frac{\phi_{i+1} - 2\phi_i + \phi_{i-1}}{dx^2} \quad (3.10)$$

But what about the LHS of equation (3.1)? For that, we need a finite difference in time. Imagine we have a set of discrete times, for example:

$$t = [0 \ 0.01 \ 0.02 \ 0.03 \ 0.04 \dots 9.99 \ 10]$$

Thus we have a uniform grid of time values, with a grid spacing of $dt = 0.01$.

We can write derivatives in exactly the same way as we did in space. The first order, forward difference is:

$$\frac{\partial \phi}{\partial t} = \frac{\phi(t + dt) - \phi(t)}{dt} + O|dt| \quad (3.11)$$

For the short-hand version of this, we'll use superscripts:

$$\frac{\partial \phi}{\partial t} = \frac{\phi^{n+1} - \phi^n}{dt} + O|dt| \quad (3.12)$$

This is the forward Euler derivative, and it is first order accurate. This will be our go-to time derivative in many of the examples that follow. As we will see though, the forward Euler scheme can be unstable, meaning we have to be careful about how large dt is.

One could also write a second order time step:

$$\frac{\partial \phi}{\partial t} = \frac{\phi^{n+1} - \phi^{n-1}}{2dt} + O|dt^2| \quad (3.13)$$

This is the basis of the leapfrog scheme. We could also write:

$$\frac{\partial \phi}{\partial t} = \frac{\phi^n - \phi^{n-1}}{dt} + O|dt| \quad (3.14)$$

which is the backward Euler derivative. We use this when *implicit time stepping*, as described below.

3.2 Integration schemes

3.2.1 FTCS

With these expressions, we can write the finite difference version of equation (3.1), for example:

$$\frac{\phi_j^{n+1} - \phi_j^n}{dt} = \kappa \frac{\phi_{j+1}^n - 2\phi_j^n + \phi_{j-1}^n}{dx^2} \quad (3.15)$$

We can solve this for $\phi(t + dt)$:

$$\phi_j^{n+1} = \phi_j^n + s(\phi_{j+1}^n - 2\phi_j^n + \phi_{j-1}^n) \quad (3.16)$$

or:

$$\phi_j^{n+1} = s\phi_{j-1}^n + (1 - 2s)\phi_j^n + s\phi_{j+1}^n \quad (3.17)$$

where:

$$s = \frac{\kappa dt}{dx^2} \quad (3.18)$$

is a parameter determined by the time step, the grid spacing and the constant, κ .

This is an example of a “forward in time, centered difference in space” version of the equation, or “FTCS” for short. This difference equation is first order accurate in time (due to the Euler step) and second order accurate in space. The FTCS scheme gives us a recipe for calculating ϕ at subsequent times.

For example, if we know the initial distribution, $\phi(x, 0) = \phi_j^0$, we can calculate the RHS of (3.12) at all the grid points. Multiplying by the factors with s and combining we obtain the solution at the next time step, i.e. ϕ_j^1 . Then we repeat, calculating the expression on the RHS to obtain the solution at the next step, ϕ_j^2 . It may sound exhausting, but it is easy work for a computer.

The FTCS scheme can be *numerically unstable*. In this example, one can show (Appendix B) that instability occurs when the parameter $s > 0.5$. As we will see, the solutions tend to infinity when s is too large. In practice, this means that you have to use a small enough value of dt to obtain solutions.

3.2.2 Leapfrog scheme

The FTCS scheme is only first order accurate in time, as noted, but we can obtain second order accuracy by using a centered time difference, as

in (3.13). To use this, we write the difference at n thus:

$$\frac{\partial \phi}{\partial t} = \frac{\phi^{n+1} - \phi^{n-1}}{2dt} + O|dt^2| \quad (3.19)$$

Using this with eq. (3.1), and solving for ϕ^{n+2} , we get:

$$\phi_j^{n+1} = \phi_j^{n-1} + s\phi_{j-1}^n - 2s\phi_j^n + s\phi_{j+1}^n \quad (3.20)$$

But how would we use this? If we start at $n = 0$, we need ϕ^{-1} , i.e. at a time prior to the initial condition. This requires an additional boundary condition, for example:

$$\frac{\partial \phi}{\partial t}(0) = A$$

If, in this case, $A = 0$, then we would have $\phi^{-1} = \phi^0$.

Lacking such a boundary condition, we could simply start the integration with an Euler step, i.e. we would obtain ψ^1 from ϕ^0 using the FTCS formula (3.17). After, we would use (3.20) to find ψ^2 . Then we would use ψ^0 and ψ^1 to obtain ψ^2 , and so on.

The leap frog scheme is often used and, as noted, more accurate than the FTCS scheme. It turns out though that the centered time difference is *unstable* (Appendix B). But the instability can be damped if one inserts an Euler step occasionally.

3.2.3 Runge-Kutta methods

The forward Euler step can be imprecise because functions are evaluated at the present point and extrapolated forward. This is shown graphically in left panel of Fig. (3.1). The slope of the curve (the tangent) at $t = t_0$ is used to step to the next point. This results in an error at $t = t_1$, because

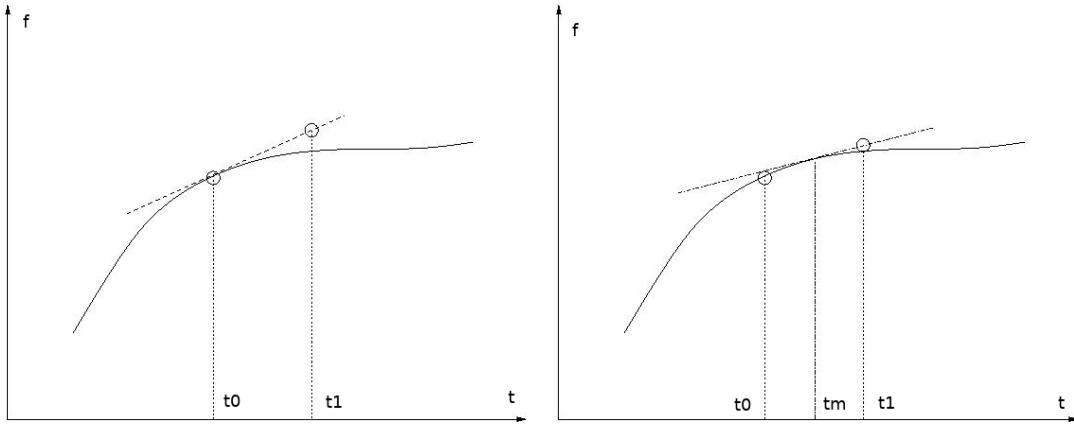


Figure 3.1: Left panel: the forward Euler time step. Left panel: the second order Runge-Kutta time step.

the curve swings away from the tangent line. Moreover, the larger the time step, the worse the error.

One way to achieve higher precision is to use the slope at the midpoint between $t = t_0$ and $t = t_1$, as shown in the right panel of Fig. (3.1). We do this by first stepping to the intermediate point, $t = t_m$, and evaluating the function there. Then we step forward from $t = t_0$ using the tangent at $t = t_m$.

Say the equation is given by:

$$\frac{df}{dt} = g(t, f) \quad (3.21)$$

The forward Euler finite difference version is then:

$$f^{n+1} = f^n + dt \ g(t^n, f^n) \quad (3.22)$$

Thus the function, $g(t, f)$ is evaluated at the n th step to step forward.

With the second order Runge-Kutta (RK2) scheme, we first step forward half-way:

$$f^{n+1/2} = f^n + \frac{dt}{2} g(t^n, f^n) \quad (3.23)$$

Then we use the value at the half-way point to obtain next value:

$$f^{n+1} = f^n + dt g\left(t^n + \frac{dt}{2}, f^{n+1/2}\right) \quad (3.24)$$

In this way, we use the function at the half-way point to determine the slope.

The RK2 method is second order accurate and thus a substantial improvement over the forward Euler time step. However, this comes at a cost: we require an additional calculation at each time step. Thus the RK2 scheme takes twice as long as the FTCS to run. Also, because the scheme is explicit, like the Euler step, it becomes unstable if the time step is too big.

Nevertheless, Runge-Kutta schemes are widely used. Higher order Runge-Kutta schemes are also popular, each being more accurate than the preceding order. A “gold standard” in numerical analysis is the fourth-order scheme (RK4), which involves three intermediate steps before the final step forward. We’ll only consider the second order scheme in this course, but the approach is (relatively) easily generalized to fourth order.

3.2.4 iFTCS

A scheme which doesn’t have a problem with numerical instability is the *implicit* forward in time scheme, or iFTCS. This is based on the backward time difference, given in (3.14). Thus, like the FTCS scheme, it is first order accurate in time. Using this, we have:

$$\frac{\phi_j^n - \phi_j^{n-1}}{dt} = \kappa \frac{\phi_{j+1}^n - 2\phi_j^n + \phi_{j-1}^n}{dx^2} \quad (3.25)$$

You can see right away though that there is a problem with this, because we again need information at a time before the beginning of the integration.

So instead we write:

$$\frac{\phi_j^{n+1} - \phi_j^n}{dt} = \kappa \frac{\phi_{j+1}^{n+1} - 2\phi_j^{n+1} + \phi_{j-1}^{n+1}}{dx^2} \quad (3.26)$$

Now we can integrate forward from the initial time, at n=1.

But how do we know what to use on the RHS? This is written in terms of ϕ^{n+1} , which is unknown! This is where the “implicit” part of the scheme comes in. First we rewrite the equation:

$$-s\phi_{j-1}^{n+1} + (1 + 2s)\phi_j^{n+1} - s\phi_{j+1}^{n+1} = \phi_j^n \quad (3.27)$$

Now the RHS is determined by data at the present time step. But the equation has three unknowns on the LHS; with one equation, we can't solve for them!

But in fact, we have a whole set of equations. Consider a simple set up, with 5 grid points in the x direction, so that $j = 1, 2, 3, 4, 5$. The values at $j = 1$ and $j = 5$ are on the boundaries, so we will assume that we know them (as under Dirichlet boundary conditions). Say, for example, that $\phi_1 = a$ and $\phi_5 = f$. Then the unknowns are the values at the three interior grid points, with $j = 2, 3, 4$. Then we have 3 equations:

$$-s\phi_1^{n+1} + (1 + 2s)\phi_2^{n+1} - s\phi_3^{n+1} = \phi_2^n \quad (3.28)$$

$$-s\phi_2^{n+1} + (1 + 2s)\phi_3^{n+1} - s\phi_4^{n+1} = \phi_3^n \quad (3.29)$$

$$-s\phi_3^{n+1} + (1 + 2s)\phi_4^{n+1} - s\phi_5^{n+1} = \phi_4^n \quad (3.30)$$

or:

$$(1 + 2s)\phi_2^{n+1} - s\phi_3^{n+1} = \phi_2^n + sa \quad (3.31)$$

$$-s\phi_2^{n+1} + (1 + 2s)\phi_3^{n+1} - s\phi_4^{n+1} = \phi_3^n \quad (3.32)$$

$$-s\phi_3^{n+1} + (1 + 2s)\phi_4^{n+1} = \phi_4^n + sf \quad (3.33)$$

Thus we have three unknowns, ϕ_2^{n+1} , ϕ_3^{n+1} and ϕ_4^{n+1} , but also three equations. So the system is closed; we can find all the unknowns.

We do this by writing the equation in matrix form:

$$\mathbf{A}\phi^{n+1} = \phi^n + \mathbf{b} \quad (3.34)$$

where:

$$\mathbf{A} = \begin{bmatrix} 1+2s & -s & 0 \\ -s & 1+2s & -s \\ 0 & -s & 1+2s \end{bmatrix}, \mathbf{b} = \begin{bmatrix} sa \\ 0 \\ sf \end{bmatrix} \quad (3.35)$$

We then obtain ϕ^{n+1} by inverting the relation:

$$\phi^{n+1} = \mathbf{A}^{-1}(\phi^n + \mathbf{b}) \quad (3.36)$$

Even though this is only first order accurate, implicit time stepping is *unconditionally stable*. As such, we can use as large values of dt as we like. However, we will see that using larger dt yields less accurate estimates.

3.2.5 The Crank-Nicholson (CN) scheme

A particularly good scheme comes from combining the FTCS and iFTCS schemes. Using equal parts of each, we would write:

$$\phi_j^{n+1} = \phi_j^n + \frac{s}{2}\phi_{j-1}^n - s\phi_j^n + \frac{s}{2}\phi_{j+1}^n + \frac{s}{2}\phi_{j-1}^{n+1} - s\phi_j^{n+1} + \frac{s}{2}\phi_{j+1}^{n+1} \quad (3.37)$$

Re-arranging, this is:

$$-\frac{s}{2}\phi_{j-1}^{n+1} + (1+s)\phi_j^{n+1} - \frac{s}{2}\phi_{j+1}^{n+1} = \frac{s}{2}\phi_{j-1}^n + (1-s)\phi_j^n + \frac{s}{2}\phi_{j+1}^n \quad (3.38)$$

Note that this still uses the first order time derivative, and so is first order accurate in time.

We can also write this as a matrix equation. To see how, we again use the case of 5 grid points in x direction:

$$-\frac{s}{2}a + (1+s)\phi_2^{n+1} - \frac{s}{2}\phi_3^{n+1} = \frac{s}{2}a + (1+s)\phi_2^n + \frac{s}{2}\phi_3^n \quad (3.39)$$

$$-\frac{s}{2}\phi_2^{n+1} + (1+s)\phi_3^{n+1} - \frac{s}{2}\phi_4^{n+1} = \frac{s}{2}\phi_2^n + (1+s)\phi_3^n + \frac{s}{2}\phi_4^n \quad (3.40)$$

$$-\frac{s}{2}\phi_3^{n+1} + (1+2s)\phi_4^{n+1} - \frac{s}{2}f = \phi_4^n + sf = \frac{s}{2}\phi_3^n + (1+s)\phi_4^n \quad (\text{3.41})$$

This is the same as:

$$\mathbf{A}\phi^{n+1} = \mathbf{B}\phi^{n+1} + \mathbf{b} \quad (3.42)$$

where:

$$\mathbf{A} = \begin{bmatrix} 1+s & -s/2 & 0 \\ -s/2 & 1+s & -s/2 \\ 0 & -s/2 & 1+s \end{bmatrix}, \mathbf{B} = \begin{bmatrix} 1-s & s/2 & 0 \\ s/2 & 1+s & s/2 \\ 0 & s/2 & 1+s \end{bmatrix}, \mathbf{b} = \begin{bmatrix} sa \\ 0 \\ sf \end{bmatrix} \quad (3.43)$$

We then solve this by inverting \mathbf{A} :

$$\phi^{n+1} = \mathbf{A}^{-1}(\mathbf{B}\phi^{n+1} + \mathbf{b}) \quad (3.44)$$

In practice, the Crank-Nicholson scheme isn't more complicated than the implicit scheme, iFTCS. Like iFTCS, the CN scheme is unconditionally stable, so it won't blow up if the time step is too large. It also often gives somewhat better results than either FTCS or iFTCS, as we'll see.

3.3 Summary

We've considered four different time stepping schemes: three explicit schemes (the forward Euler (FTCS), the leapfrog and the Runge-Kutta), and two implicit (iFTCS and the Crank-Nicholson (CN)). Which is best?

When choosing a numerical scheme, there are three main points to consider:

- Accuracy
- Numerical stability
- Speed of execution

The FTCS, iFTCS and CN schemes are first order in time and second order in space, while the leapfrog and RK2 are second order accurate in time and space. So the leapfrog and RK2 schemes are the best choices in terms of accuracy.

However, the explicit schemes are numerically unstable if the time step is too large. The iFTCS and CN are unconditionally stable, so there is no limit on the time step. But, as we will see, these schemes become less and less accurate as the time step increases. So there is a limit to what time steps we actually use.

Then there is the question of speed. The explicit schemes are the fastest, as the steps forward (in eqs. 3.17 and 3.20) are executed using the present values. Among these the FTCS and leapfrog schemes are faster; the RK2 scheme, which requires an additional line at each time step, is only half as fast. The implicit schemes require an additional step, which is inverting the difference matrix A . If this is only done once, at the beginning of the code, then the increase in computation time is minimal. But we will see examples where the inversion has to be repeated at every time step; this *greatly* slows the execution. In such cases, an explicit scheme is definitely preferable.

Thus there isn't a single “best” scheme. What we generally do is to start with FTCS and then compare the results with the other schemes. In some cases, FTCS is perfectly adequate; it's also the easiest to implement and is

generally the fastest to run. But in other cases, the more elaborate schemes are superior. You'll begin to get a feel for which is best, as we study different problems.

Moreover, this is absolutely *not* the only set of possibilities available! There are many other schemes as well.

Next we begin considering different physical systems that we can model numerically. The approach is often the same, as follows:

- we construct an equation which simulates the dynamics
- we obtain an *analytical* solution, usually in a simplified case
- we write the finite difference version of the equation
- we write a code to solve the difference equation
- we test the finite difference code against the analytical solution

After that, we can use the code to examine cases which are more complicated than what we can study analytically.

Chapter 4

Time dependent systems

We begin with the simplest case, a system which only depends on time. An example is the amount of carbon in a rock (which can be used to determine its age), or the volume of water in a lake. Such systems can be modelled with ODEs, with time as the only independent variable. Interesting behavior can result, nonetheless, particularly in coupled systems.

4.1 Single volume systems

A typical example of a purely time-dependent system is one which has been integrated in space. Consider the total mass of a lake, M . If there are no mass sources or sinks, the total mass should be conserved in time.

So:

$$\frac{d}{dt}M = 0 \quad (4.1)$$

But with sources (for example, precipitation over a lake) or sinks (evaporation over the lake), the total mass can change.

4.1.1 Radioactive decay

Another example is the amount of carbon 14 (C^{14}) in a rock sample. Carbon 14 is radioactive and decays with a known rate. The “half-life” of Carbon 14 is 5730 years, meaning that after that period of time, only half of the original Carbon 14 will remain. Thus radioactive decay is a sink, because it reduces the integrated amount.

The time rate of change of the carbon is just proportional to the amount of carbon:¹

$$\frac{dC}{dt} = -\lambda C \quad (4.2)$$

The equation implies that the decay rate is greatest when C is largest. The rate then decreases as the carbon is depleted.

The equation is easily solved analytically. Re-arranging, we have:

$$\frac{dC}{C} = -\lambda dt \quad (4.3)$$

Then integrating both sides we have:

$$\ln(C) = -\lambda t + A \quad (4.4)$$

where A is a constant of integration. Taking the exponential of both sides, we get:

$$C = \exp(-\lambda t + A) = B \exp(-\lambda t) \quad (4.5)$$

where $B = \exp(A)$ is another constant.

How do we find the constant B ? If we know how much carbon there is initially, at $t = 0$, then we write:

$$C(0) = B \exp(-\lambda \cdot 0) = B \quad (4.6)$$

¹A linear decay term like this is referred to as “Rayleigh damping”, after Lord Rayleigh (1842-1919). This is the simplest form of damping that can be imposed in a system.

So:

$$C = C(0) \exp(-\lambda t) \quad (4.7)$$

Thus the amount of carbon in the sample decays *exponentially in time*.

This is shown in Fig. (4.1). The amount decays from the initial value, here set to $C(0) = 100$, with the characteristic exponential decay.

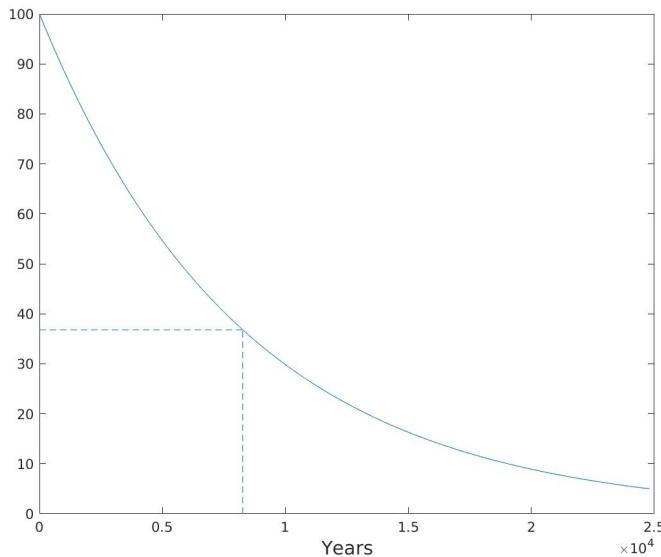


Figure 4.1: The exponential decay of carbon 14. The initial value is 100, and the e-folding time scale (indicated by the dashed lines) is 8267 years.

Just how quickly does it decay? When $t = t_e = \lambda^{-1}$, then:

$$C = C(0) \exp(-1) = 0.37C(0) \quad (4.8)$$

So the amount remaining is 37% of the initial value. We call the time $t_e = \lambda^{-1}$ the “e-folding time scale”, a typical measure with exponential systems. This is 8267 years for Carbon 14.

Another commonly-used time is the half-life, mentioned above. This is:

$$C = C(0)/2 = C(0) \exp(-\lambda t_{1/2}) \quad (4.9)$$

Solving for $t_{1/2}$, we get:

$$t_{1/2} = -\ln(0.5)/\lambda = -\ln(0.5)t_e \quad (4.10)$$

For Carbon 14, this is 5730 years, as noted before. The half-life is often used with radioactive substances. For instance, the half-life of Uranium 235 is 704 million years. Thus if a nuclear power plant explodes, as the Chernobyl plant in Russia did in 1986, one would have to wait over 700 million years for the U_{235} fall-out to decay by half.

But we also have sources of carbon. In the atmosphere, cosmic rays convert nitrogen (N^{14}) to Carbon 14. We can model this by adding a forcing term to the RHS of the equation:

$$\frac{dC}{dt} = P - \lambda C \quad (4.11)$$

Radioactive decay removes carbon, but the cosmic rays constantly replenish it. Let's assume that P is a constant, i.e. the cosmic rays produce carbon at a fixed rate.

How do we solve (4.11)? After some time, the total carbon will cease to change. Then the LHS of the equation goes to zero, leaving:

$$P - \lambda C = 0 \quad \rightarrow \quad C = \frac{P}{\lambda} \quad (4.12)$$

This is known as the *steady state solution*. This is a familiar concept in geosciences, and in physics generally, and one we will refer to frequently hereafter. After some period of time, the solution relaxes to this steady state.

In the preceding example, with carbon or uranium simply decaying, the steady state is zero. If one waits long enough, there is eventually nothing left. But with a source, the carbon decays to a non-zero value.

The full analytical solution to (4.11) can be shown to be:

$$C = \frac{P}{\lambda} + (C(0) - \frac{P}{\lambda}) \exp(-\lambda t) \quad (4.13)$$

Notice that at $t = 0$, $C = C(0)$. But as $t \rightarrow \infty$, $C \rightarrow \frac{P}{\lambda}$, the steady state solution.

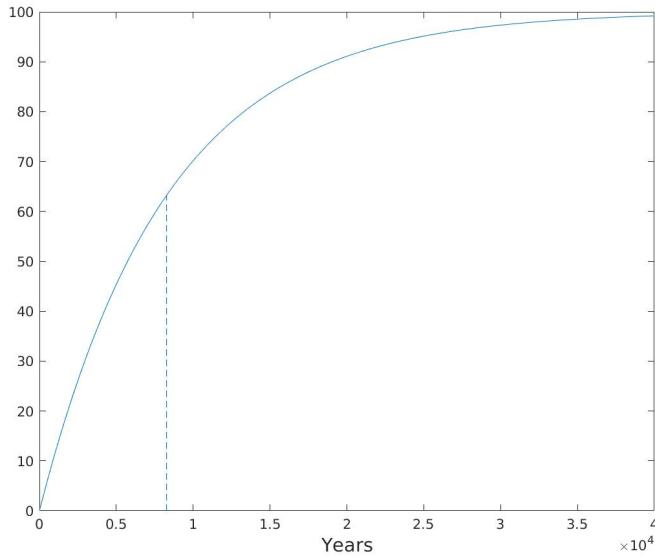


Figure 4.2: The amount of Carbon with constant forcing. The forcing, P , is such that the steady state solution is 100.

The solution is plotted in Fig. (4.2), for the case when the amount of Carbon is initially zero, $C(0) = 0$. We've also set $P = 100\lambda$, so that the steady state solution is 100. The amount of carbon grows from zero, and eventually levels off at 100. At the e-folding time, $t = \lambda^{-1} = 8267$ years, the carbon has reached $1 - 0.37 = 0.63$ times the steady state value.

4.1.2 Time dependent forcing

Now consider a more complicated case. Diverse human activities lead to pollution in rivers. This include wastewater discharge, run-off from cities into storm drains and from agricultural activity, in particular fertilizer and

pesticides. The pollutants flow in the rivers to lakes and also to the ocean, where they can have significant negative impacts on fish and other marine life.

Consider the total amount of pollutants in a lake. The supply, which enters via rivers feeding the lake, varies in time. There is a constant background supply (due to sewerage, for example) and a seasonal variation (due to agriculture). We can express the total mass of the pollutants in the lake as $M(t)$, governed by the following equation:

$$\frac{dM}{dt} = P + b \sin(\omega t) - kM \quad (4.14)$$

P is the yearly average supply while the $b \sin(\omega t)$ is the seasonally varying supply. We'll assume this term is a maximum in spring, when the crops are being fertilized. We assume too that the pollutants decay in time, with a rate proportional to the total amount, as in the previous section.

The solution is somewhat more complicated:

$$M = \frac{P}{k} + (M(0) - \frac{P}{k} + \frac{b\omega}{k^2 + \omega^2})e^{-kt} + \frac{bk}{k^2 + \omega^2} \sin(\omega t) - \frac{b\omega}{k^2 + \omega^2} \cos(\omega t) \quad (4.15)$$

The first term on the RHS is the steady state solution. If there were no seasonal variation, the amount of fertilizer would rise until that supplied by the rivers equaled the amount lost to decay. The second term is a *transient* term, representing the change from the initial condition. The last two *oscillatory* terms occur because of the seasonal change. These cause the total amount to rise and fall.

The solution is plotted in Fig. (4.3), for a chosen set of parameters. The total forcing (constant + seasonal) is shown by the dashed curve, and the total amount, M , by the solid curve. The seasonal variation causes the wa-

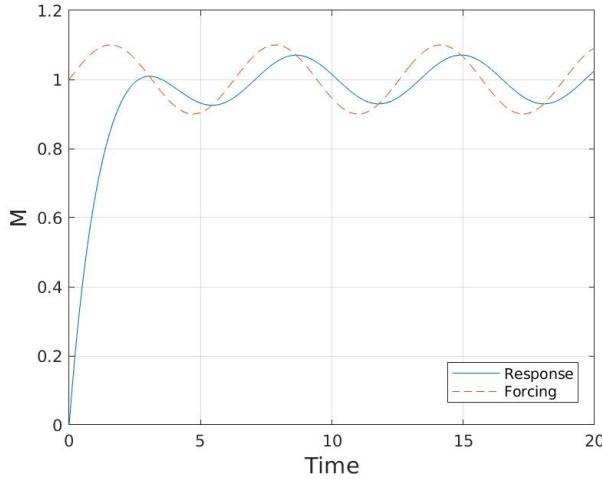


Figure 4.3: The amount of fertilizer in a lake as a function of time (solid curve) with the total forcing (dashed curve).

ter level to oscillate. But notice that the level is *out of phase* with the forcing; the maximum level is reached later. So the fertilizer level in the lake reaches its maximum level in early summer, rather than in the spring.

4.1.3 Numerical solutions

Purely time dependent problems like those above yield ODEs, which are relatively straightforward to solve analytically. But when applying them to more realistic cases, they can become intractable. Imagine for example that the fertilizer input to the lake was a more complicated function of time, even varying from year to year. Then an analytical solution would be much more difficult. In such cases, numerical solutions are very useful.

ODEs like (4.14) are relatively simple to solve numerically, since there is only one equation and the derivative is first order. First we'll use the forward Euler time step to solve this. The finite difference version of the equation is:

$$\frac{M^{n+1} - M^n}{dt} = P + b \sin(\omega t^n) - k M^n \quad (4.16)$$

or:

$$M^{n+1} = M^n + dt(P + b \sin(\omega t^n) - kM^n) \quad (4.17)$$

For comparison, we'll also calculate the second order Runge-Kutta version of the equation, as in sec. (3.2.3). This is:

$$M^{n+1/2} = M^n + \frac{dt}{2}(P + b \sin(\omega t^n) - kM^n) \quad (4.18)$$

$$M^{n+1} = M^n + dt\left(P + b \sin\left(\omega\left(t^n + \frac{dt}{2}\right)\right) - kM^{n+1/2}\right) \quad (4.19)$$

Note that the sine term in the second equation is evaluated at $t^n + dt/2$.

The solutions are shown in Fig. (4.4) for two values of the time step, dt . With $dt = 0.1$, both numerical solutions are close to the analytical solution (4.15). With $dt = 0.5$, the solutions are less precise, but the Runge-Kutta solution is better, deviating less from the actual solution. Keep in mind though that the Runge-Kutta solution, with its two steps, takes twice as long to execute.

The fourth order Runge-Kutta is even better with this time step, but also takes four times longer than the Euler step. Thus it sometimes is equally attractive to take the Euler step and reduce the time step, dt .

4.2 Multiple volumes

More complex behavior can be observed with multiple, connected domains. Consider a simple model of the Earth's carbon budget.² In this, carbon can reside in the atmosphere and also in plants at the surface. The plants take up CO₂ and water, under photosynthesis, but also release carbon when they die. Carbon is also produced in the atmosphere as cosmic

²This example is taken from Slingerland and Kump (2011).

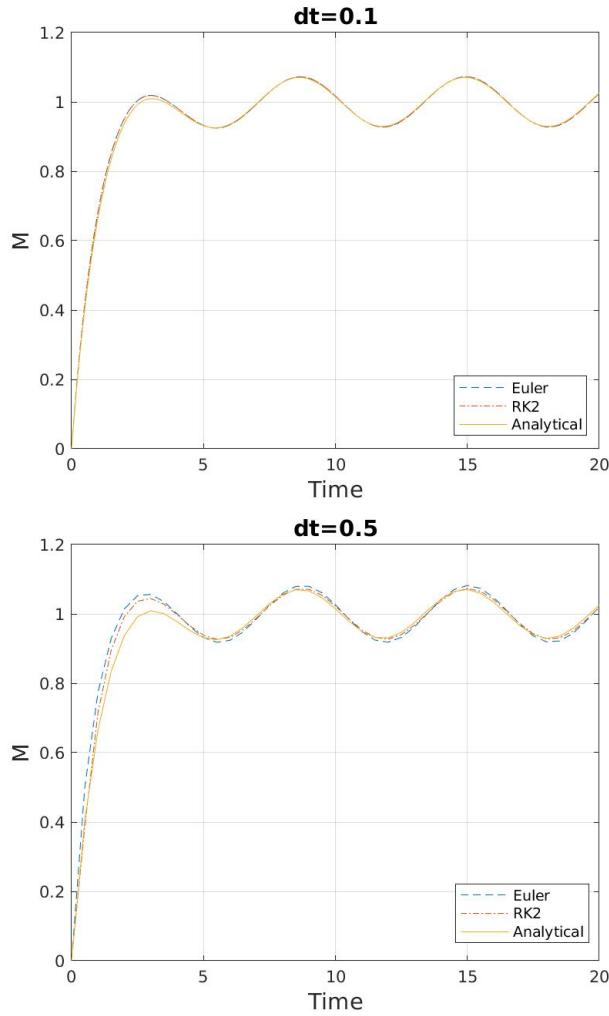


Figure 4.4: Numerical solutions to the lake problem, with two time steps, $dt = 0.1$ (left panel) and $dt = 0.5$ (right panel).

rays convert Nitrogen to Carbon 14. But we'll ignore the source for the moment and focus on the atmosphere-plant interaction.

Say that the total mass of carbon in the atmosphere is C_1 , and the total in the plants is C_2 . We represent the uptake of CO₂ by the plants as a *flux* from the atmosphere. Assume this flux is proportional to the total amount in the atmosphere, so that:

$$F_{12} = k_{12}C_1$$

where F_{12} is the carbon flux from the atmosphere to the plants and k_1 is a constant of proportionality. Likewise, the loss of CO₂ back to the atmosphere depends on the total mass of plants, so:

$$F_{21} = k_{21}C_2$$

Then the total mass budgets can be written as follows:

$$\frac{dC_1}{dt} = k_{21}C_2 - k_{12}C_1 \quad (4.20)$$

$$\frac{dC_2}{dt} = k_{12}C_1 - k_{21}C_2 \quad (4.21)$$

The flux to the plants causes the atmospheric mass (C_1) to decrease and the carbon in the plants (C_2) to increase. Likewise, the flux from the plants increases C_1 and decreases C_2 . As such, the atmosphere and plant concentrations are *coupled*.

However, the total mass of carbon is conserved. If we add the two equations together, we obtain:

$$\frac{d}{dt}(C_1 + C_2) = 0 \quad (4.22)$$

There is no creation or destruction of carbon; it's simply passed between the two systems.

We can write the system (4.21) as a matrix equation:

$$\frac{d}{dt} \begin{bmatrix} C_1 \\ C_2 \end{bmatrix} = \begin{bmatrix} -k_{12} & k_{21} \\ k_{12} & -k_{21} \end{bmatrix} \begin{bmatrix} C_1 \\ C_2 \end{bmatrix} \quad (4.23)$$

4.2.1 Analytical solution

We can treat (4.23) as an *eigenvalue problem*, which will reduce the solution to a useful form. We do this by looking for solutions which have:

$$\frac{d}{dt} \begin{bmatrix} C_1 \\ C_2 \end{bmatrix} = \lambda \begin{bmatrix} C_1 \\ C_2 \end{bmatrix} \quad (4.24)$$

where λ is a constant, the “eigenvalue”.

When you have an equation like:

$$\frac{d}{dt}\phi = \lambda\phi \quad (4.25)$$

the solution is an exponential:

$$\phi = \phi(0)e^{\lambda t} \quad (4.26)$$

For the case with a 2x2 matrix, as above, there are two different eigenvalues. Thus we expect two types of exponential behavior.

Given this, we can re-write the (4.23) thus:

$$\begin{bmatrix} -k_{12} & k_{21} \\ k_{12} & -k_{21} \end{bmatrix} \begin{bmatrix} C_1 \\ C_2 \end{bmatrix} = \lambda \begin{bmatrix} C_1 \\ C_2 \end{bmatrix} \quad (4.27)$$

or:

$$\begin{bmatrix} -k_{12} - \lambda & k_{21} \\ k_{12} & -k_{21} - \lambda \end{bmatrix} \begin{bmatrix} C_1 \\ C_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (4.28)$$

In order for eq. (4.28) to be satisfied, the determinant of the matrix on the LHS has to vanish. That yields a quadratic equation for λ :

$$(k_{12} + \lambda)(k_{21} + \lambda) - k_{21}k_{12} = 0 \quad (4.29)$$

or:

$$\lambda^2 + (k_{12} + k_{21})\lambda + (k_{21}k_{12} - k_{12}k_{21}) = 0 \quad (4.30)$$

Because the last term vanishes, we have:

$$\lambda(\lambda + k_{12} + k_{21}) = 0 \quad (4.31)$$

Thus:

$$\lambda = 0, \quad \lambda = -(k_{12} + k_{21}) \quad (4.32)$$

So we can expect two types of solution, one which is constant in time and one which decays like $\exp(-(k_{12} + k_{21})t)$. Thus the system will decay from its initial condition to a steady state solution.

To finish the problem, we need the *eigenvectors* associated with the eigenvalues. We obtain these by setting λ to each of the eigenvalues and then solving the resulting equation. Take $\lambda = 0$. Then we have:

$$\begin{bmatrix} -k_{12} & k_{21} \\ k_{12} & -k_{21} \end{bmatrix} \begin{bmatrix} C_1 \\ C_2 \end{bmatrix} = 0 * \begin{bmatrix} C_1 \\ C_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (4.33)$$

The first equation is:

$$-k_{12}C_1 + k_{21}C_2 = 0 \quad (4.34)$$

The second equation is just the negative of this. This is typical of eigenvalue problems; the resulting equations are the same (they are “degenerate”). We are free to pick values that will satisfy this equation, for example:

$$\begin{bmatrix} C_1 \\ C_2 \end{bmatrix} = \begin{bmatrix} k_{21} \\ k_{12} \end{bmatrix} \quad (4.35)$$

Setting $\lambda = -(k_{12} + k_{21})$, we obtain two identical equations for the other eigenvector. You can show the answer is:

$$\begin{bmatrix} C_1 \\ C_2 \end{bmatrix} = \begin{bmatrix} 1 \\ -1 \end{bmatrix} \quad (4.36)$$

Putting this all together, the solution is:

$$\begin{bmatrix} C_1 \\ C_2 \end{bmatrix} = A \begin{bmatrix} k_{21} \\ k_{12} \end{bmatrix} + B \begin{bmatrix} 1 \\ -1 \end{bmatrix} e^{-(k_{12}+k_{21})t} \quad (4.37)$$

The first term on the RHS is constant – this is the steady state solution. The second is decaying in time, and this allows an adjustment from the initial condition to the steady state.

Whe constants, A and B are unknown and we find them from the initial conditions. After a bit of algebra, the result is:

$$C_1 = \frac{k_{21}(C_{10} + C_{20})}{k_{12} + k_{21}} + \frac{k_{12}C_{10} - k_{21}C_{20}}{k_{12} + k_{21}}e^{-(k_{12}+k_{21})t} \quad (4.38)$$

$$C_2 = \frac{k_{12}(C_{10} + C_{20})}{k_{12} + k_{21}} - \frac{k_{12}C_{10} - k_{21}C_{20}}{k_{12} + k_{21}}e^{-(k_{12}+k_{21})t} \quad (4.39)$$

where C_{10} and C_{20} are the initial values of C_1 and C_2 respectively.

The solution is plotted in Fig. (4.5). We've assumed the carbon is initially in the atmosphere ($C_{10} = 100$, $C_{20} = 0$). The constants are set as $k_{12} = 1/13\text{yr}^{-1}$ and $k_{21} = 1/10\text{yr}^{-1}$, following bulk estimates from observations (Kump and Slingerland, 2011). Carbon is removed from the atmosphere and is taken up by plants, and then reaches a steady state, with more residing in the atmosphere.

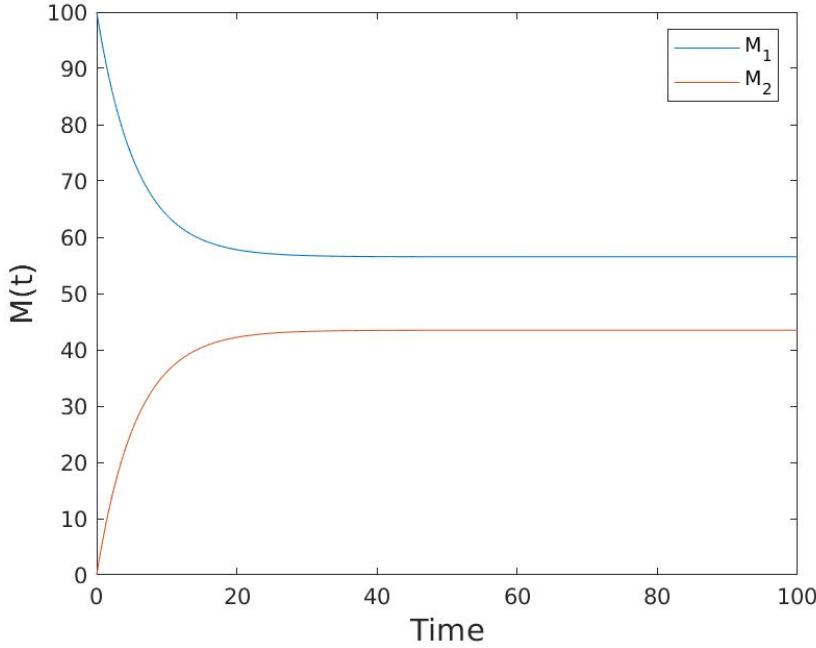


Figure 4.5: The carbon concentration in the atmosphere and on land (in plants) from the two box model.

An interesting aspect here is that the decay time for the transient response

differs from the two time scales in the problem. This is given by:

$$(k_{12} + k_{21})^{-1} = \frac{130}{13 + 10} = 5.65 \text{ yr}$$

So the combination of the two effects yields a much *faster* decay than expected from one of the systems alone.

4.2.2 Numerical solutions

Now let's examine the numerical solution. What's different this time is that we have two components, C_1 and C_2 . But we can treat equation (4.23) exactly as written, as a matrix equation. The finite difference version, with the forward Euler step, is just:

$$\begin{aligned} \begin{bmatrix} C_1 \\ C_2 \end{bmatrix}^{n+1} &= \begin{bmatrix} C_1 \\ C_2 \end{bmatrix}^n + dt \begin{bmatrix} -k_{12} & k_{21} \\ k_{12} & -k_{21} \end{bmatrix} \begin{bmatrix} C_1 \\ C_2 \end{bmatrix}^n \\ &= \begin{bmatrix} 1 - k_{12}dt & k_{21}dt \\ k_{12}dt & 1 - k_{21}dt \end{bmatrix} \begin{bmatrix} C_1 \\ C_2 \end{bmatrix}^n \end{aligned} \quad (4.40)$$

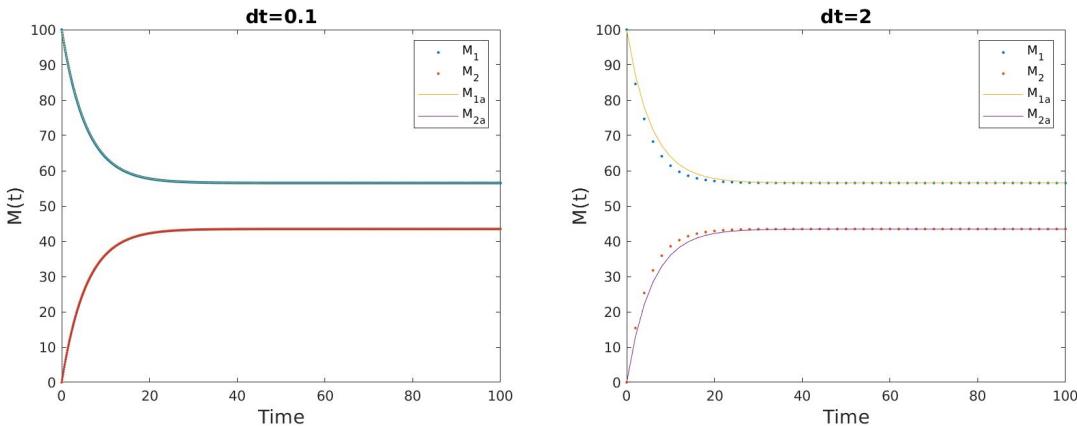


Figure 4.6: Numerical solutions for two box model using the forward Euler time step. In the left panel, $dt = 0.1$; in the right panel, $dt = 2$.

The Euler solution is plotted in the left panel of Fig. (4.6) the numerical solution (in dots) is plotted over the analytical solution (solid curves). The two are indistinguishable.

To obtain this, we had to use a small time step ($dt = 0.1$). As such, running to $t = 100$ requires 1000 steps. What if we use a larger time step? An example with $dt = 2$ is shown in the right panel. Now the numerical solution differs from the analytical one. But the two have the same steady state.

If we use an implicit time step instead, we would write:

$$\begin{bmatrix} C_1 \\ C_2 \end{bmatrix}^{n+1} = \begin{bmatrix} C_1 \\ C_2 \end{bmatrix}^n + dt \begin{bmatrix} -k_{12} & k_{21} \\ k_{12} & -k_{21} \end{bmatrix} \begin{bmatrix} C_1 \\ C_2 \end{bmatrix}^{n+1} \quad (4.41)$$

or:

$$\begin{bmatrix} 1 + k_{12}dt & -k_{21}dt \\ -k_{12}dt & 1 + k_{21}dt \end{bmatrix} \begin{bmatrix} C_1 \\ C_2 \end{bmatrix}^{n+1} = \begin{bmatrix} C_1 \\ C_2 \end{bmatrix}^n \quad (4.42)$$

Using the matrix inverse, we get:

$$\begin{bmatrix} C_1 \\ C_2 \end{bmatrix}^{n+1} = \begin{bmatrix} 1 + k_{12}dt & -k_{21}dt \\ -k_{12}dt & 1 + k_{21}dt \end{bmatrix}^{-1} \begin{bmatrix} C_1 \\ C_2 \end{bmatrix}^n \quad (4.43)$$

The result is shown in Fig. (4.7). With $dt = 0.1$, the solution is as good as with the forward Euler and matches the analytical solution. With $dt = 2$, the numerical solution again differs from the analytical, despite that the steady solution is captured correctly. In contrast to the Euler solution, this under-predicts the change at early times. But otherwise the two routines yield similar results (and take roughly the same amount of time computationally).

4.3 A two mass model

Burridge and Knopoff (1967) proposed an idealized model for an earthquake, involving a set of masses connected by springs. Oscillations in one mass are transferred to the others via the springs. Such models are still

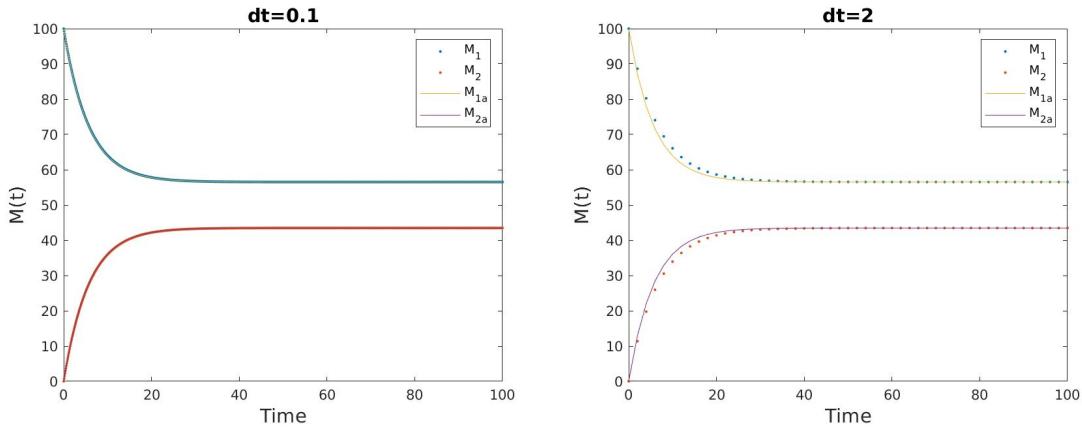


Figure 4.7: Numerical solutions for two box model using the implicit Euler time step. In the left panel, $dt = 0.1$; in the right panel, $dt = 1$.

used as idealized representations of seismic disturbances. We'll consider a highly simplified version of this with two equal masses, connected by springs to each other and to the adjacent walls (Fig. 4.8). The displacement of the left mass is x and of the right mass is y .

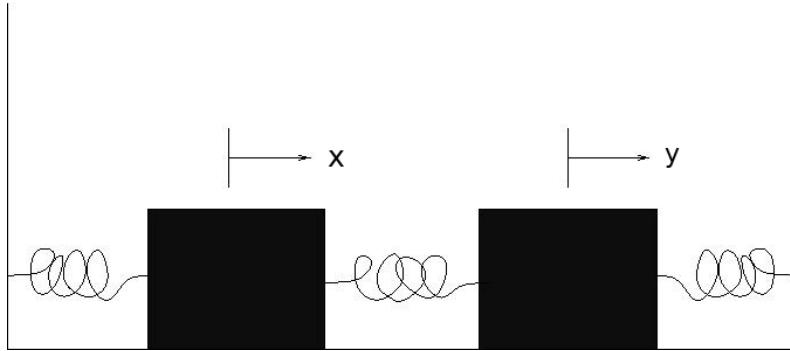


Figure 4.8: The two mass-spring system.

Newton's 2nd law when applied to the masses yields two equations:

$$m \frac{d^2x}{dt^2} = -2kx + ky \quad (4.44)$$

$$m \frac{d^2y}{dt^2} = kx - 2ky \quad (4.45)$$

Where do the terms on the RHS come from? If the left mass is moved a distance x , both the spring on the wall and the one on the right mass

oppose the displacement. But if the right mass is moved a distance y , this will accelerate the left mass to the right. Similar arguments apply to the other mass.

If the masses oscillate, the positions will vary sinusoidally in time:

$$(x, y) \propto \cos(\omega t) \rightarrow \left(\frac{d^2x}{dt^2}, \frac{d^2y}{dt^2} \right) = -\omega^2(x, y) \quad (4.46)$$

Combining this with the equations of motion and rearranging gives:

$$\begin{bmatrix} -2 & 1 \\ 1 & -2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \frac{m\omega^2}{k} \begin{bmatrix} x \\ y \end{bmatrix} \quad (4.47)$$

We see immediately that this is another eigenvalue problem, with an eigenvalue of $\lambda = m\omega^2/k$

Diagonalizing and taking the determinant yields another quadratic equation:

$$\lambda^2 - 4\lambda + 3 = (\lambda - 3)(\lambda - 1) = 0 \quad (4.48)$$

This has two solutions, $\lambda = 3$ and $\lambda = 1$. Thus there are two frequencies:

$$\omega_1 = \sqrt{\frac{k}{m}}, \quad \omega_2 = \sqrt{\frac{3k}{m}} \quad (4.49)$$

Thus the first corresponds to a slow oscillation and the second to a fast one.

The eigenvectors can be found as before. These are:

$$\omega_1 \rightarrow \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad \omega_2 \rightarrow \begin{bmatrix} 1 \\ -1 \end{bmatrix} \quad (4.50)$$

Thus for the slow oscillation, the masses move together, but for the fast oscillation, they move towards and away from each other. If you think about it, the latter involves more opposition from the springs (you're compressing the middle one while pulling the other two). Thus the masses will return to their equilibrium positions faster.

Even this simplest possible earthquake system has two frequencies. You can imagine if the earthquake had many masses, it would have a much richer range of motion.

4.4 A simple climate model

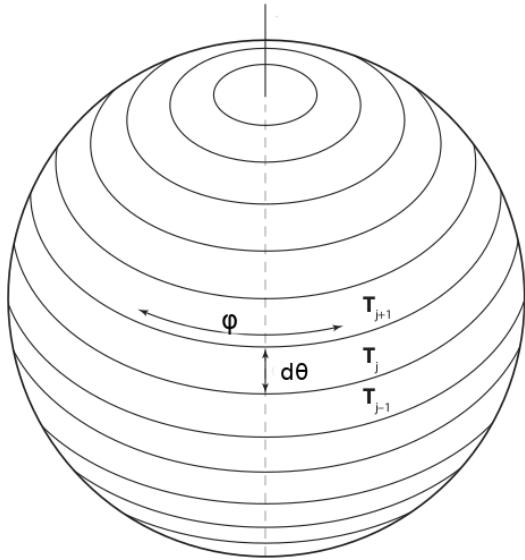


Figure 4.9: The energy balance model for the atmosphere. There is a single box at each latitude, with a temperature, T_j .

As a last example of a multi-box model, we'll look at a simple “energy balance” model of the atmosphere. We divide the atmosphere into boxes with a latitudinal width, $d\theta$, of one degree (Fig. 4.9). Each box is vertically integrated, and so has a single temperature, T_j . We'll assume for now that the only forcing is the sun heating the box and radiative cooling back to space. Later we'll see what happens when the boxes can exchange heat as well.

4.4.1 Black body radiation

To formulate the forcing terms, we need to consider how heat is transferred from the sun or from the planet. There are basically three methods of heat transfer relevant for us:

- *Conduction* – for example of heat transferred along a solid metal bar
- *Convection* – this occurs in the mantle, when plumes of molten rock rise up towards the surface, but also in the atmosphere, where rising warm air forms clouds
- *Radiation* – transferred by electromagnetic waves

Of the three types, radiation is the only one that occurs in space. Convection typically happens in fluids while conduction most often happens in solids.

We refer to a radiating body as a “black body”. This curious nomenclature comes about because a “black” surface is opaque or non-reflecting. The radiation of a black body depends only on its temperature. The black body concept comes from thermodynamics and is important in a range of applications.

The energy associated with black body radiation is conserved. Thus the integrated amount of radiation leaving the surface of the black body is the same as that passing a sphere centered on the black body. Thus we can express this as:

$$Q_s A_S = Q_r A_r$$

where Q_s and A_s are the radiation and area at the black body surface and Q_r and A_r are the values at a distance r from its center. Substituting in for

the (spherical) surface areas, we get:

$$Q_r = Q_b \frac{4\pi r_b^2}{4\pi r^2} = Q_b \frac{r_b^2}{r^2} \quad (4.51)$$

Thus the radiation decreases as $1/r^2$ away from the black body. This is referred to as the “inverse square law” of radiation.

To determine the radiation arriving at earth from the sun, we set $r_b = r_{sun}$, and $r = r_{se}$, the distance between the sun and earth. But how do we determine Q_s , the sun’s radiation?

The light coming from a black body has a certain *spectrum*, which means there are different intensities at different wavelengths. The sun is yellow, meaning the peak intensity has the wavelength of yellow light.

The spectral intensity at different wavelengths for a radiating body is given by *Planck’s function*:

$$B_\lambda(T) = \frac{c_1}{\lambda^5(\exp(c_2/(\lambda T)) - 1)} \quad (4.52)$$

where T is the temperature, λ is the wavelength and c_1 and c_2 are two constants which depend on the speed of light and Boltzmann’s and Planck’s constants. The function is plotted in Fig. (4.10), for three different temperatures.

Notice that the peak intensity shifts slightly in wavelength as the temperature changes. The higher the temperature, the shorter the peak wavelength. Given Planck’s law (4.52), one can derive a relation for the maximum intensity (by taking the derivative of $B_\lambda(T)$ with respect to T and setting that to zero). The result is Wien’s law:

$$\lambda_{max} = \frac{b}{T} \quad (4.53)$$

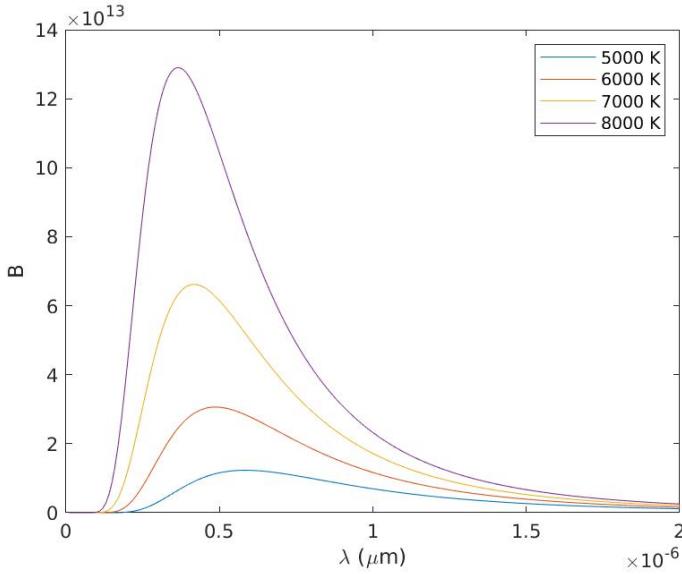


Figure 4.10: Planck’s law for three different temperatures.

where $b = 2897 \mu\text{m K}$. Wien’s law is useful because we can determine the temperature of a radiating body by finding the peak wavelength.

Because the sun is yellow, the peak wavelength is $0.475 \mu\text{m}$. Thus we can estimate the temperature at the surface of the sun as:

$$T = \frac{b}{\lambda_{max}} = \frac{2897}{0.475} = 6100K \quad (4.54)$$

In fact, the temperature is slightly lower, as the sun differs somewhat from a blackbody. But this estimate is still quite good.

If we integrate Planck’s law (4.52) over all wavelengths, we get the total energy flux from the black body. This yields the *Stefan-Boltzmann Law*:

$$F = \sigma T^4 \quad (4.55)$$

where $\sigma = 5.67 \times 10^{-8} \text{ W}/(\text{m}^2 \text{ K}^4)$ is the Stefan-Boltzmann constant. We see that the energy flux varies strongly with temperature. Doubling the temperature increases the flux by a factor of 16.

Using the temperature above for the sun, 6100 K, yields:

$$F = \sigma(6100)^4 = 7.85 \times 10^7 \text{ W/m}^2$$

The actual value is somewhat less ($6.28 \times 10^7 \text{ W/m}^2$), again due to deviations from the black body law. But this estimate is fairly close.

The earth is also warm, as the incoming solar radiation heats it up. But being warm, it also radiates energy. If we were to suddenly switch on the sun, the earth would warm until the outgoing radiation exactly balanced the incoming solar radiation. Using the Stefan-Boltzmann law, we can then estimate the equilibrium temperature for the planet.

To do this, we first need the amount of radiation reaching earth from the sun, the “solar constant”. This is given by:

$$Q_0 = Q_s \frac{r_{sun}^2}{r_{se}^2}$$

where r_{se} is again the distance from the sun to earth. Plugging in values, we get:

$$Q_0 = 6.28 \times 10^7 \frac{(7 \times 10^8)^2}{(1.5 \times 10^{11})^2} = 1368 \text{ W/m}^2$$

To obtain the total radiance warming the planet, we integrate over the area facing the sun. This is the circular area with the earth’s radius:

$$F_{in} = Q_0 \pi r_{earth}^2$$

However, not all the incoming radiation is absorbed by the planet. Some of it is reflected back to space, from ice and snow for example. This *albedo* effect is represented by a reflection fraction, α . So we add a factor to reduce the incoming radiation:

$$F_{in} = Q_0 \pi (1 - \alpha) r_{earth}^2$$

The albedo can be measured directly by satellite.³ The average albedo for the planet is about 0.3, meaning 70% of the incoming radiation is absorbed.

For the outgoing radiation, we assume the earth radiates uniformly over it's whole surface area:

$$F_{out} = Q_e(4\pi r_{earth}^2) = \sigma T_{earth}^4(4\pi r_{earth}^2)$$

Because the earth's temperature is much less than the sun's, the outgoing radiation has a longer wavelength. Indeed, it is primarily in the “infrared”, and hence is not visible to us. As such, the incoming radiation is referred to as the “shortwave” radiation and the outgoing as “longwave” radiation.

If we equate the incoming and outgoing radiation, we get:

$$F_{in} = F_{out} \rightarrow Q_0\pi(1 - \alpha)r_{earth}^2 = \sigma T_{earth}^4(4\pi r_{earth}^2)$$

or:

$$T_{earth} = \left(\frac{1368(1 - 0.3)}{4(5.67 \times 10^{-8})}\right)^{1/4} = 255 \text{ K}$$

This is -18C, which is rather chilly! It is also quite a bit cooler than the average temperature on earth.

What are we missing? Due to characteristics of the earth's surface and the atmosphere, the emitted radiance is less than that of a black body. Thus the *emissivity*, the ratio of the emitted radiation to that of a black body, is less than one. An example of this is a thermally insulated window, which emits less heat than a regular pane of glass. On earth, heat is trapped in

³Interestingly, it can also be inferred by looking at how bright the moon is when the moon is in the earth's shadow, so that it is illuminated solely by the earth.

the atmosphere – the *greenhouse effect* – and this reduces the outgoing radiation. The average emissivity on earth is about $\epsilon = 0.6$.

Using this, the balance is:

$$F_{in} = F_{out} \rightarrow Q_0\pi(1 - \alpha)r_{earth}^2 = \epsilon\sigma T_{earth}^4(4\pi r_{earth}^2)$$

This yields a higher estimate for the temperature:

$$T_{earth} = \left(\frac{1368(1 - 0.3)}{4(0.6)(5.67 \times 10^{-8})}\right)^{1/4} = 290 \text{ K}$$

or 17C, which is a much more pleasant temperature. Thus the greenhouse effect makes the planet more habitable than it would otherwise be.

What if the incoming and outgoing radiation aren't in balance? This is actually the case right now, due to humans adding CO₂ to the atmosphere. CO₂ is a potent absorber of longwave radiation, trapping heat on the planet. The radiative imbalance results in a temperature rise – *global warming*. This warming is throwing the biosphere out of balance, and threatening the survival of species. This simple model illustrates how important the effect is to the planet's energy balance.

4.4.2 A latitudinally-varying model

The foregoing is the simplest form of climate model, known as an “energy balance model” (EBM). We can make our EBM more realistic if we allow for *variations with latitude*. The incoming radiation is not uniform with latitude, because the planet is a sphere not a disc, and because it tilts on its axis of rotation. The albedo also varies strongly with latitude, as snow and ice at the poles reflect more sunlight than the rain forests near the equator. Thus we can write:

$$F_{in}(\theta) = S(\theta)(1 - \alpha(\theta)) \quad (4.56)$$

where θ is the latitude, $S(\theta)$ is the incoming shortwave radiation and $\alpha(\theta)$ is the albedo. The outgoing radiation also varies with latitude (since the poles are colder than the equator), so we write:

$$F_{out}(\theta) = \epsilon\sigma T(\theta)^4 \quad (4.57)$$

A given latitude box has a mass of air. This is:

$$m = \rho V = \rho A H$$

where ρ is the air density, V is the volume, A the area and H is the height of the box. We'll assume all boxes have the same area, volume and height.⁴ Each box is exposed to incoming radiation and radiates heat to space. Thus its temperature changes, following this formula (which comes from the first law of thermodynamics):

$$\frac{d}{dt}(\rho c_p A H T(\theta)) = (F_{in}(\theta) - F_{out}(\theta))A \quad (4.58)$$

Here c_p is the *specific heat* which links how much the temperature for given substance rises in response to heating. If we label the latitude with the subscript, j , we can write this as:

$$\frac{d}{dt}(\rho c_p A H T_j) = (F_{ij} - F_{oj})A \quad (4.59)$$

If we divide through by the constants and insert the expressions for the radiative terms, we have:

$$\frac{d}{dt}T_j = \frac{1}{\rho c_p H} \left(\frac{S_j}{4} (1 - \alpha_j) - \epsilon \sigma T_j^4 \right) \quad (4.60)$$

⁴Including the latitudinal variation in volume introduces small but non-negligible changes.

Numerical solution

We'll use equation (4.60) to model the atmospheric temperature. We'll use the forward Euler scheme to do this. Thus we rewrite (4.60) in finite difference form:

$$\frac{T_j^{n+1} - T_j^n}{dt} = \frac{1}{\rho c_p H} \left(\frac{S_j}{4} (1 - \alpha_j) - \epsilon \sigma (T_j^n)^4 \right) \quad (4.61)$$

We then collect terms with the same power of n on the same sides:

$$T_j^{n+1} = T_j^n + \frac{S_j (1 - \alpha_j) dt}{4 \rho c_p H} - \frac{\epsilon \sigma (T_j^n)^4 dt}{\rho c_p H} \quad (4.62)$$

This is straightforward to code.

For this, we need the solar insolation, $S_j/4$, the albedo and the emissivity. Using published curves, we use the following function to estimate the annually-averaged insolation:

$$\frac{S}{4} = (350 \cos(\theta) + 150) \text{W/m}^2$$

Thus the incoming radiation is greatest at the equator and decreases toward the poles. However, the insolation doesn't vanish at the poles – remember that this is the annual average insolation, so it includes the summers as well, when the poles have 24 hours of sunlight.

Similarly, we represent the average albedo with the following:

$$\alpha_j = 0.7 - 0.45 \cos(\theta)$$

The albedo is largest, 0.7, near the poles, which are covered with ice and snow for much of the year. It decreases toward the equator. But it is not zero there either, due to cloud cover. We also set the emissivity to $\epsilon = 0.6$ as before.

What is a good choice for the time step? We can find out by scaling the time derivative term in (4.60) and equating it with the earth's radition term. This yields:

$$\frac{\mathcal{T}}{T_{rad}} \propto \epsilon\sigma\mathcal{T}^4\rho c_p H$$

or:

$$T_{rad} = \frac{\rho c_p H}{\epsilon\sigma\mathcal{T}^3} \quad (4.63)$$

Assuming a mean tropospheric height of $H=10$ km, this is approximately:

$$T_{rad} \approx \frac{(1)(1004)(10^4)}{0.6(5.67 \times 10^{-8})(290)^3} = 12 \times 10^6 \text{ sec} = 140 \text{ days} \quad (4.64)$$

This implies we don't need to use a 1 second time step – that would entail much too many time steps. A good choice of dt is $T_{rad}/100$, or about 1.4 days.

Then we are ready to step (4.62) forward in time. We take the initial condition to be such that $T_j = 0$ for all j .

The results are shown in Fig. (4.11). After 14 days, the equatorial region is warming, but only with a maximum value of -22C. After 42 days though, the equator is past freezing and the warming has spread to higher latitudes. After roughly 6 months or so, the curves converge to a single curve, which is the steady state solution.

Despite that we can't solve (4.60) analytically, we can obtain the steady state solution. We do this by setting the LHS of the equation to zero. This yields:

$$T_j = \left(\frac{S_j(1 - \alpha_j)}{4\epsilon\sigma}\right)^{1/4} \quad (4.65)$$

Using the expressions for S_j , α_j and ϵ above, we can evaluate this numerically. The result agrees quite well with the numerical solution, except

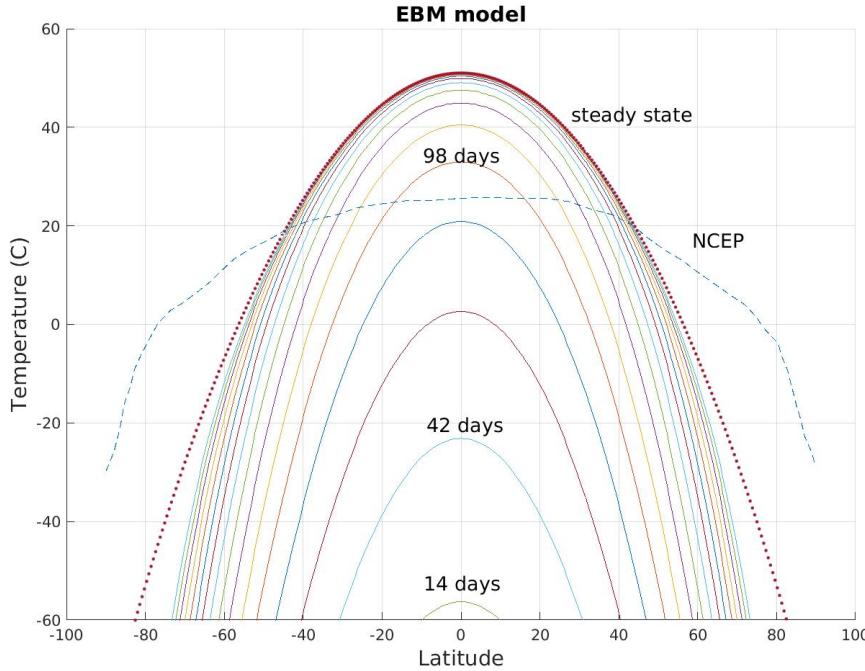


Figure 4.11: The equilibrium temperature profile from the simple energy balance model. The National Center for Environmental Prediction (NCEP) mean atmospheric temperature is shown for comparison.

at the highest latitudes where the numerical solution exhibits colder values. This is because the convergence to the steady state is slower at high latitudes, as the damping term is weaker with colder temperatures.

How does the steady state profile compare to observed temperatures? The answer is: not very well! Also shown in Fig. (4.11) is the annual and latitudinal mean temperature curve from an atmospheric reanalysis by the National Center for Environmental Prediction (NCEP). The temperature is the right order of magnitude, but the gradients are much too sharp in the model solution. As a result, the equator is too warm (50C as opposed to 25C) and the poles are *much* too cold (-80C!). So our first attempt at a climate model was not a big success.

What are we missing? In this model, there is no heat exchange between

latitude boxes. In reality there is a lot of heat exchange, mediated mostly by *storms*. Storms move heat north and south, cooling the equator and warming the poles, and thereby reducing the temperature gradients.

We'll take this up later, when we study diffusion processes in Chap. (6). But we can still use the model to test different dependencies. For example, what happens if the emissivity is 1.0, so that the earth is radiating as a perfect black body? Then the temperature is much colder than observed, at all latitudes (upper panel of Fig. 4.12). This again illustrates the greenhouse effect, which keeps temperatures more habitable.

But increasing greenhouse gases leads to excess warming, as seen by changing the emissivity only slightly, from 0.6 to 0.55 (lower panel of Fig. 4.12). Then the equatorial temperatures increase by about 4C, which is a substantial amount. The higher latitudes see less warming, which is actually unrealistic. We know that under global warming, the high latitudes warm faster – producing so-called “arctic amplification”. But our model is too simple to capture this.

What happens if the albedo is zero, i.e. if all the ice, snow and clouds disappeared? Then the temperatures are much *warmer* at every latitude (not shown). So these features are also very important for maintaining a livable environment on the planet.

This gives you a basic idea of how we can test different climate sensitivities. But the present model is deficient and needs to be made more realistic. We'll do this in the subsequent chapters. In the end, we'll be able to get a fairly reasonable estimate of the annual mean temperature.

4.5 Summary

We have examined a number of time dependent problems. These can represent volume-averaged systems, for example the entire atmosphere or ocean. We can express such problems as single or coupled ordinary differential equations (ODEs). Systems that can be represented by linear equations are usually solvable, using matrices and/or ODE techniques. The nonlinear problems are less tractable and generally require numerical solutions.

We tested several types of numerical representation. For these simple problems, even the simple forward Euler representation often gives good results. Higher accuracy can be obtained with Runge-Kutta methods, but these also take longer to run. And the implicit forward Euler method yielded results comparable to the forward Euler, and thus weren't necessary.

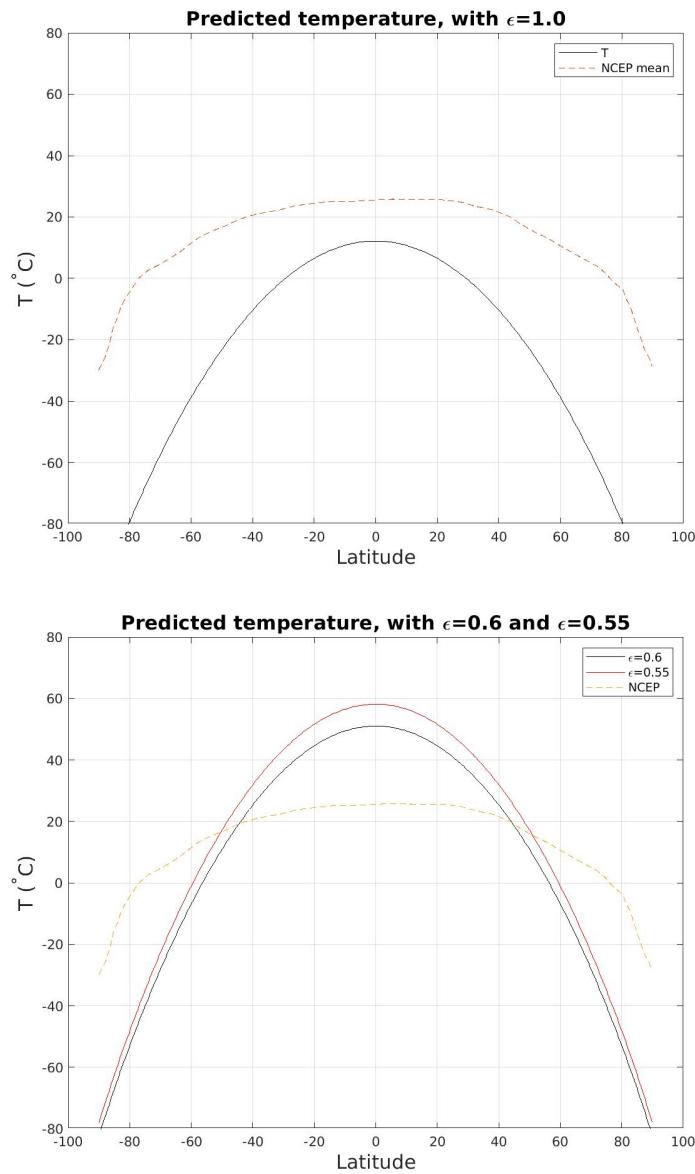


Figure 4.12: Equilibrium temperature profiles with from the simple EBM. The National Center for Environmental Prediction (NCEP) mean atmospheric temperature is shown for comparison.

Chapter 5

Advection/transport



Figure 5.1: Pollution from the Grand River flowing into Lake Michigan in the United States. Courtesy of *The Rapidian*.

Now we consider systems with variations in both space and time. *Advection* is the transport of a substance by the motion of a medium. In the geosciences, the medium is usually a fluid.

An example is a river, which sweeps leaves and debris downstream. Winds

are another example, advecting smoke over a city. And ocean currents transport phytoplankton from one region to another.

There are a wide range of advected substances in such systems. We distinguish “passive” advected quantities, whose motion does not change the system itself. Pollutants are a good example. When oil is advected away from a sinking tanker, the oil itself does not change the ocean currents. The other possibility is an “active” tracer, whose motion does alter the system. Heat and energy are two active tracers (in fluids, though not necessarily in solids). Transporting energy from a region with high energy to one with low energy causes the latter to “spin up”.

In this chapter, we’ll focus on passive tracers. This has applications to pollution modelling, for example of particles from car exhaust in cities, and in many other areas. The same type of equation can also be applied to wave propagation, when the waves move at a fixed speed. In chapter (8) we’ll examine the more complicated problem of an active tracer, which usually involves nonlinear equations.

5.1 The advection equation

To write an equation for an advection, it’s useful to think in terms of *fluxes*. Consider a small volume, like that shown in Fig. (5.2). The box has sides dx , dy and dz . On the left side, some material, ϕ , is carried in by the flow. On the right side, material is carried out. But the flow and the amount of material on the left and right hand sides need not be the same. If they are not the same, the total amount of ϕ in the box will change.

Let’s express this mathematically. The total amount of material in the box

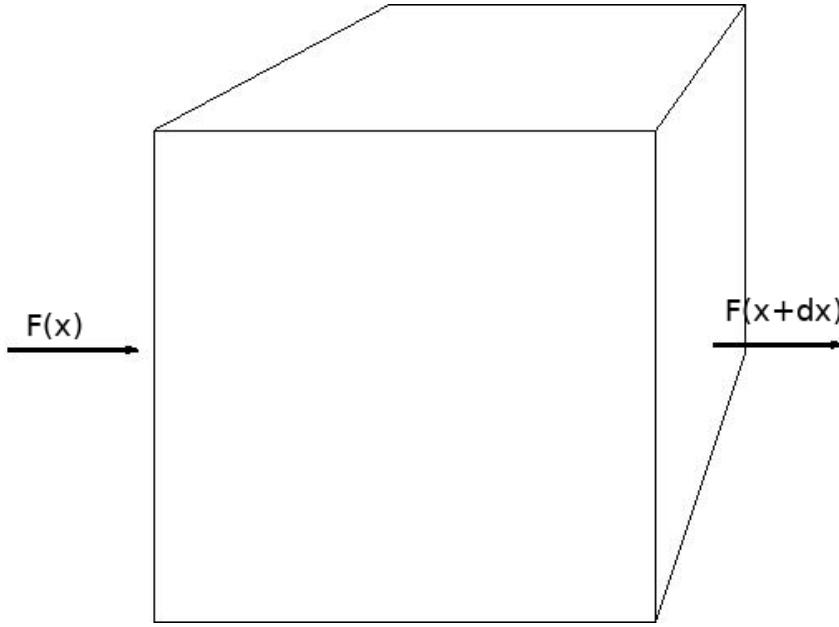


Figure 5.2: The flux of some material, ϕ , into and out of a box, with a size, dx .

is:

$$\iiint \phi \, dx \, dy \, dz$$

If the box is small enough, then we can assume ϕ doesn't vary in it. Then we can write:

$$\iiint \phi \, dx \, dy \, dz \approx \phi(x, y, z, t) V = \phi A dx$$

where $A = dydz$ is the area of the side and dx is the width. The time rate of change of this is then:

$$\frac{d}{dt} \iiint \phi \, dx \, dy \, dz = Adx \frac{\partial}{\partial t} \phi$$

We move the volume through the derivative because it is a fixed quantity, but we require a partial time derivative for ϕ now, since $\phi = \phi(x, y, z, t)$.

The total flux of material in through the left side is given by:

$$\iint u(x)\phi(x) \, dy \, dz \approx u(x)\phi(x)A$$

The flux is $u(x)\phi(x)$, but we need to integrate over the area to get the total transport. Notice this quantity has units of $\phi m^3/sec$, which is the same as the time derivative of ϕ integrated over the box. Similarly, the flux out the right side is:

$$\iint u(x+dx)\phi(x+dx) dy dz \approx u(x+dx)\phi(x+dx)A$$

Putting this all together, we get:

$$Adx \frac{\partial}{\partial t} \phi = u(x)\phi(x)A - u(x+dx)\phi(x+dx)A \quad (5.1)$$

or:

$$\frac{\partial}{\partial t} \phi = \frac{1}{dx} [u(x)\phi(x) - u(x+dx)\phi(x+dx)] \quad (5.2)$$

Now, since the box (and hence dx) is small, we can expand the last term using a Taylor series:

$$u(x+dx)\phi(x+dx) = u(x)\phi(x) + \frac{\partial}{\partial x} [u(x)\phi(x)] dx + O|dx^2|$$

We ignore all the higher order terms, which are of order dx^2 or smaller. Putting this into the expression above, and canceling the $u(x)\phi(x)$ and dx terms, we get:

$$\frac{\partial}{\partial t} \phi = -\frac{\partial}{\partial x} [u(x)\phi(x)] \quad (5.3)$$

The last term is the “advective term”. We usually put this on the left side of the equation:

$$\frac{\partial}{\partial t} \phi + \frac{\partial}{\partial x} (u\phi) = 0 \quad (5.4)$$

Equation (5.4) is the advection equation, in one dimension, for the passive tracer, ϕ . If we include advection through the other sides of the box, we get the advection equation in 3 dimensions:

$$\frac{\partial}{\partial t}\phi + \nabla \cdot (\vec{u}\phi) = \frac{\partial}{\partial t}\phi + \frac{\partial}{\partial x}(u\phi) + \frac{\partial}{\partial y}(v\phi) + \frac{\partial}{\partial z}(w\phi) = 0 \quad (5.5)$$

where (u, v, w) are the velocities in the (x, y, z) directions. Equation (5.5) states that if there is a net inflow or outflow of ϕ into the volume, the total amount of ϕ in the volume will change.

The same derivation can be used to derive a conservation for the fluid density:

$$\frac{\partial}{\partial t}\rho + \nabla \cdot (\vec{u}\rho) = 0 \quad (5.6)$$

An interesting distinction here is the density is an active tracer. Thus the divergence term in (5.6) involves two dependent variables, the velocity and the density, and as such, the term is *nonlinear*. This is the *continuity equation*, and it represents the conservation of mass. It is one of the fundamental equations in many numerical models, for the weather, the ocean and groundwater movement.

The equation also applies for momentum, another active tracer. Thus for example, the advection of velocity in the x direction is given by:

$$\frac{\partial}{\partial t}u + \nabla \cdot (\vec{u}u) = 0 \quad (5.7)$$

As with the density, the advective term is nonlinear and results in a change in the flow.

The advection equation takes a simpler form if the velocity is *incompressible*. Incompressibility applies to water, which approximately conserves its volume (under normal pressures). As discussed in sec. (9.2.1), an in-

compressible fluid has:

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = \nabla \cdot \vec{u} = 0 \quad (5.8)$$

Expanding (5.5), we have:

$$\frac{\partial}{\partial t} \phi + \vec{u} \cdot \nabla \phi + \phi (\nabla \cdot \vec{u}) = 0 \quad (5.9)$$

using a vector identity for the divergence term. The second term vanishes for an incompressible flow, leaving:

$$\frac{\partial}{\partial t} \phi + \vec{u} \cdot \nabla \phi = 0 \quad (5.10)$$

This is somewhat simpler to work with than (5.5).

5.1.1 Eulerian and Lagrangian formulations

The preceding equations apply to flow at a fixed point, i.e. $\phi(x, y, z, t)$. But what if we were interested in the changes following a volume advected by the flow, for example a balloon in the atmosphere?

Consider an infinitesimal change in ϕ , which we write as $d\phi$. From the chain rule, we can write:

$$d\phi(x, y, z, t) = \frac{\partial \phi}{\partial t} dt + \frac{\partial \phi}{\partial x} dx + \frac{\partial \phi}{\partial y} dy + \frac{\partial \phi}{\partial z} dz \quad (5.11)$$

If we divide through by dt , we get:

$$\frac{d\phi}{dt} = \frac{\partial \phi}{\partial t} + \frac{\partial \phi}{\partial x} \frac{dx}{dt} + \frac{\partial \phi}{\partial y} \frac{dy}{dt} + \frac{\partial \phi}{\partial z} \frac{dz}{dt} \quad (5.12)$$

Using the definitions of the velocities, this is:

$$\frac{d\phi}{dt} = \frac{\partial \phi}{\partial t} + u \frac{\partial \phi}{\partial x} + v \frac{\partial \phi}{\partial y} + w \frac{\partial \phi}{\partial z} \quad (5.13)$$

or, in vector form:

$$\frac{d\phi}{dt} = \frac{\partial \phi}{\partial t} + \vec{u} \cdot \nabla \phi \quad (5.14)$$

The LHS of equation (5.14) is called the *Lagrangian* derivative. This signifies the change in ϕ for a particle advected by the flow. The RHS is the local time derivative and the advective terms, and this applies at a fixed point. We call the RHS the *Eulerian* formulation.

What if we use this in the incompressible version of the advection equation (5.10)? Then we have simply that:

$$\frac{d\phi}{dt} = 0 \quad (5.15)$$

So, the material ϕ is *conserved following the flow*.

Most of the time, we will use the Eulerian formulation, since it is more natural in modeling. But it's worth knowing about the Lagrangian formulation, which can be useful in analyzing data. Some more sophisticated numerical models use a combination of the two as well, for improved accuracy and speed of execution.

5.1.2 Advection with a constant velocity

A very simple case is when the advecting velocity, u , is constant. In one dimension, equation (5.10) is:

$$\frac{\partial}{\partial t} \phi + u \frac{\partial}{\partial x} \phi = 0 \quad (5.16)$$

This could represent a (very) idealized river, flowing with the velocity, u . ϕ could be a pollutant, spilled in the river.

Equation (5.16) is a *hyperbolic* PDE. Equations like this can be solved using the “method of characteristics”. We won’t go into this here, but simply state the solution:

$$\phi = F(x - ut) \quad (5.17)$$

where F is an unspecified function.

First, let’s check that this solution works. We can write:

$$\frac{\partial}{\partial t} \phi = F' \frac{\partial}{\partial t} (x - ut) = -uF'$$

The prime indicates that we take the derivative with respect to the argument. Say, for example, we had:

$$\phi = \phi_0(x - ut)^2$$

Using the chain rule, we have:

$$\frac{\partial}{\partial t} \phi = F' \frac{\partial}{\partial t} (x - ut) = -uF'$$

With the form above, we have:

$$\frac{\partial}{\partial t} \phi = -2u\phi_0(x - ut)$$

Similarly, we have:

$$\frac{\partial}{\partial x} \phi = F' \frac{\partial}{\partial x} (x - ut) = F'$$

So we have:

$$u \frac{\partial}{\partial x} \phi = 2u\phi_0(x - ut)$$

Putting these two expressions into (5.16), we get:

$$\frac{\partial}{\partial t} \phi + u \frac{\partial}{\partial x} \phi = -2u\phi_0(x - ut) + 2u\phi_0(x - ut) = 0$$

This works for *any* smooth function, F :

$$\frac{\partial}{\partial t}\phi + u\frac{\partial}{\partial x}\phi = -uF' + uF' = 0$$

The function F is usually determined by the starting condition. Then we just replace the x in the initial condition by $x - ut$. For example, if the concentration is initially a Gaussian function:

$$\phi(x, 0) = \phi_0 e^{-(x-x_0)^2/L^2}$$

then the solution is:

$$\phi(x, t) = \phi_0 e^{-(x-ut-x_0)^2/L^2} \quad (5.18)$$

Or if the initial condition is a sine function:

$$\phi(x, 0) = \phi_0 \sin(kx)$$

then the solution is:

$$\phi(x, t) = \phi_0 \sin(k(x - ut))$$

The Gaussian solution is plotted in Fig. (5.3). The Gaussian has a width, $L = 1$ and is initially centered at $x=2.2$. It shifts to the right at a constant speed. If this were an oil spill in a river, the distribution would maintain its shape as it moved downriver.

Alternately, $\phi(x, t)$ could be the *probability* of finding a person who has fallen in the river. If there is some uncertainty in where she fell in, that would yield a distribution of possible locations, like a Gaussian. The moving distribution would then indicate where we should look for the person.

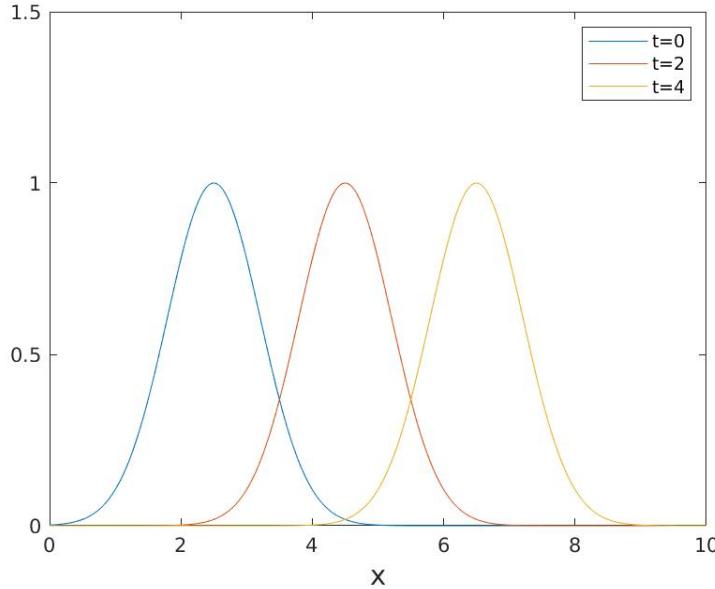


Figure 5.3: An initially Gaussian distribution advected at a constant velocity, $u = 1$.

5.1.3 Wave equation

Equation (5.16) is also a one dimensional wave equation. A wave, propagating at a constant speed, obeys:

$$\frac{\partial}{\partial t} h \pm c \frac{\partial}{\partial x} h = 0 \quad (5.19)$$

where h is the wave height and c is the *phase speed*. One example is sound waves, for which $c = 343$ m/sec in air.

Equation (5.19) has the same type of solution as the 1D advection equation, specifically:

$$h = F(x \mp ct) \quad (5.20)$$

where F is some type of function, usually determined by the initial conditions, and where the sign in front of the c is determined by that in the equation. So if the wave equation is:

$$\frac{\partial}{\partial t} h - c \frac{\partial}{\partial x} h = 0 \quad (5.21)$$

the solution is:

$$h = F(x + ct) \quad (5.22)$$

In this case, the wave is propagating to the right, because as time increases, x must increase to keep the argument of F the same.

We'll study waves more in sec. (9.3). But keep in mind that the techniques for solving the advection equation also apply in some wave cases.

5.1.4 Numerical solutions

FTCS

We start with the simplest time stepping routine, the forward Euler. We also use the centered difference for the spatial derivative (hence this is a FTCS scheme). The finite difference equation with constant velocity becomes:

$$\frac{\phi_j^{n+1} - \phi_j^n}{dt} + u \frac{\phi_{j+1}^n - \phi_{j-1}^n}{2dx} = 0 \quad (5.23)$$

Re-arranging:

$$\phi_j^{n+1} = \frac{Co}{2} \phi_{j-1}^n + \phi_j^n - \frac{Co}{2} \phi_{j+1}^n \quad (5.24)$$

where:

$$Co = \frac{udt}{dx} \quad (5.25)$$

is the *Courant number*. Equation (5.24) gives us a prescription for predicting ϕ at the next time step, given the values at the present time step.

We can rewrite this as a matrix equation thus:

$$\phi^{n+1} = \mathbf{A}\phi^n \quad (5.26)$$

The matrix, \mathbf{A} , is:

$$\mathbf{A} = \begin{bmatrix} 1 & -Co/2 & 0 & 0 & 0 \\ Co/2 & 1 & -Co/2 & 0 & 0 \\ 0 & Co/2 & 1 & -Co/2 & 0 \\ 0 & 0 & Co/2 & 1 & -Co/2 \\ 0 & 0 & 0 & Co/2 & 1 \end{bmatrix} \quad (5.27)$$

in the 5x5 case, and:

$$\phi = \begin{bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \phi_4 \\ \phi_5 \end{bmatrix} \quad (5.28)$$

Then it's a simple matter to code up the advection equation. We specify the initial condition, ϕ^0 , and then multiply by \mathbf{A} , to obtain ϕ^1 . We then repeat until the chosen number of steps is completed.

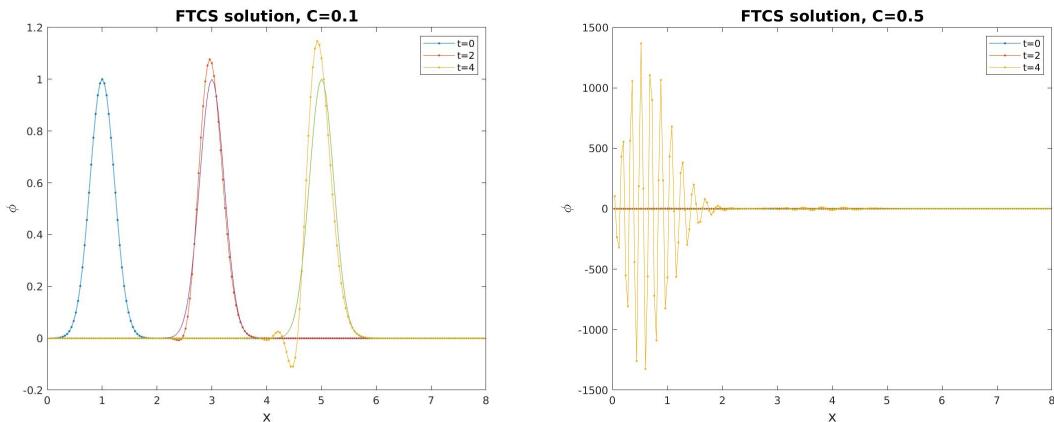


Figure 5.4: FTCS solutions for an initially Gaussian distribution advected at a constant velocity, $u = 1$. In the left panel, $C = 0.1$, and in the right, $C = 0.5$.

Examples are shown in Fig. (5.4). In the left panel, $Co = 0.1$. The numerical solutions are plotted over the analytical solution (5.18) at three different times ($t = 0, 2, 4$). The numerical solution tracks the analytical approximately, but the amplitude is growing, and there are wiggles devel-

oping on the trailing side of the Gaussian. If you run the code longer, these discrepancies get worse.

In the right panel is the solution with $Co = 0.5$. This is a disaster from the outset; noise develops and quickly grows. At $t = 4$, the amplitude exceeds 1000 (compared to the analytical solution, with a constant amplitude of 1).

The solution with $Co = 0.5$ is numerically unstable. But interestingly, the solution with $Co = 0.1$ is *also* unstable. If we run that case long enough, the amplitude gets very large. Indeed, the FTCS scheme is always unstable for simple advection. If you're interested in seeing this, the details are given in Appendix B.

iFTCS

We can avoid numerical instability by using the iFTCS scheme instead. Then we write the finite difference equation as:

$$\frac{\phi_j^{n+1} - \phi_j^n}{dt} + u \frac{\phi_{j+1}^{n+1} - \phi_{j-1}^{n+1}}{2dx} = 0 \quad (5.29)$$

or:

$$-\frac{Co}{2}\phi_{j-1}^{n+1} + \phi_j^{n+1} + \frac{Co}{2}\phi_{j+1}^{n+1} = \phi_j^n \quad (5.30)$$

with the Courant number, Co , as defined before. The matrix equation is then:

$$\mathbf{A}\phi^{n+1} = \phi^n \quad (5.31)$$

with:

$$\mathbf{A} = \begin{bmatrix} 1 & Co/2 & 0 & 0 & 0 \\ -Co/2 & 1 & Co/2 & 0 & 0 \\ 0 & -Co/2 & 1 & Co/2 & 0 \\ 0 & 0 & -Co/2 & 1 & Co/2 \\ 0 & 0 & 0 & -Co/2 & 1 \end{bmatrix} \quad (5.32)$$

We then take the inverse of \mathbf{A} and write:

$$\phi^{n+1} = \mathbf{A}^{-1} \phi^n \quad (5.33)$$

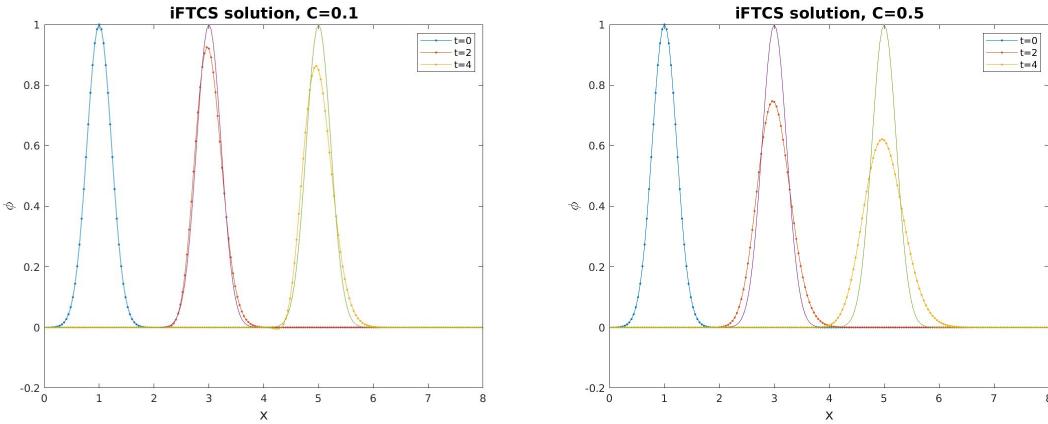


Figure 5.5: iFTCS solutions for the initially Gaussian distribution, with $Co = 0.1$ (left panel) and $Co = 0.5$ (right panel).

Solutions for the same two Courant numbers are shown in Fig. (5.5). As before, the numerical curves track the analytical ones with $Co = 0.1$, but now the amplitude is decreasing. Note too that the wiggles on the trailing edge are absent. With $Co = 0.5$, the numerical solutions are now stable, as expected. But the decrease in amplitude is more pronounced than with $Co = 0.1$. Now it's clear that the Gaussian signal is spreading out in time. Increasing Co further causes the spreading to happen faster. So while the iFTCS is numerically stable, it is less accurate with larger values of Co

The spreading is linked to *diffusion*, which is implicit in the numerical scheme. We'll see in Chapter (7) that this type of evolution is typical of advective-diffusive systems.

Crank-Nicholson

A good compromise between stability and accuracy can be had with the Crank-Nicholson scheme. This involves evaluating the advection term at the present step *and* the next step. So we write:

$$\frac{\phi_j^{n+1} - \phi_j^n}{dt} + u \frac{\phi_{j+1}^{n+1} - \phi_{j-1}^{n+1}}{4dx} + u \frac{\phi_{j+1}^n - \phi_{j-1}^n}{4dx} = 0 \quad (5.34)$$

or, rearranging:

$$-\frac{Co}{4}\phi_{j-1}^{n+1} + \phi_j^{n+1} + \frac{Co}{4}\phi_{j+1}^{n+1} = \frac{Co}{4}\phi_{j-1}^n + \phi_j^n - \frac{Co}{4}\phi_{j+1}^n \quad (5.35)$$

The matrix version is just:

$$\mathbf{A}\boldsymbol{\phi}^{n+1} = \mathbf{B}\boldsymbol{\phi}^n \quad (5.36)$$

with:

$$\mathbf{A} = \begin{bmatrix} 1 & Co/4 & 0 & 0 & 0 \\ -Co/4 & 1 & Co/4 & 0 & 0 \\ 0 & -Co/4 & 1 & Co/4 & 0 \\ 0 & 0 & -Co/4 & 1 & Co/4 \\ 0 & 0 & 0 & -Co/4 & 1 \end{bmatrix} \quad (5.37)$$

and

$$\mathbf{B} = \begin{bmatrix} 1 & -Co/4 & 0 & 0 & 0 \\ Co/4 & 1 & -Co/4 & 0 & 0 \\ 0 & Co/4 & 1 & -Co/4 & 0 \\ 0 & 0 & Co/4 & 1 & -Co/4 \\ 0 & 0 & 0 & Co/4 & 1 \end{bmatrix} \quad (5.38)$$

Taking the inverse of \mathbf{A} , we get:

$$\boldsymbol{\phi}^{n+1} = \mathbf{A}^{-1}\mathbf{B}\boldsymbol{\phi}^n \quad (5.39)$$

This is only slightly more complicated to set up than the iFTCS scheme. We calculate the inverse of \mathbf{A} once, multiply by \mathbf{B} , and then we're off.

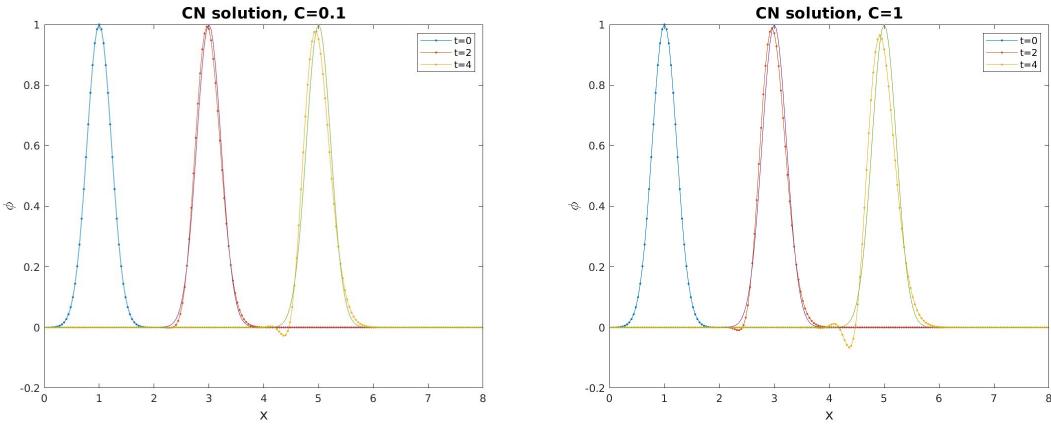


Figure 5.6: Crank-Nicholson solutions for the initially Gaussian distribution, with $Co = 0.1$ (left panel) and $Co = 1.0$ (right panel).

Solutions are plotted in Fig. (5.6), now with $Co = 0.1$ and $Co = 1.0$. With $Co = 0.1$ (left panel), the solution tracks the analytical one fairly well, although the profile differs somewhat by $t = 4$ and oscillations are developing on the trailing edge. The result is actually not much worse with $Co = 1.0$ (right panel), though the deviations are somewhat more pronounced. But remember: the code with $Co = 1$ runs *10 times faster* than that with $Co = 0.1$. So though we lose a bit with accuracy, we gain a lot with speed.

Running the code again with a wider initial Gaussian, the result is much improved—the profile maintains its shape better and the trailing oscillations are absent. Conversely, a narrower initial distribution is harder to capture, and requires a smaller time step.

RK2

We'll try one more explicit scheme. As noted, the forward Euler time step is only first order accurate in time. We can obtain higher accuracy using the second order Runge-Kutta scheme (sec. 3.2.3). With this, the finite

differencing happens in two stages:

$$\phi_j^{n+1/2} = \phi_j^n + \frac{Co}{4} \phi_{j-1}^n - \frac{Co}{4} \phi_{j+1}^n \quad (5.40)$$

$$\phi_j^n = \phi_j^n + \frac{Co}{2} \phi_{j-1}^{n+1/2} - \frac{Co}{2} \phi_{j+1}^{n+1/2} \quad (5.41)$$

As a matrix equation, this is:

$$\phi^{n+1/2} = \phi^n + \frac{1}{2} \mathbf{A} \phi^n \quad (5.42)$$

$$\phi^n = \phi^n + \mathbf{A} \phi^{n+1/2} \quad (5.43)$$

$$(5.44)$$

with the matrix, \mathbf{A} , now given by:

$$\mathbf{A} = \begin{bmatrix} 0 & -Co/2 & 0 & 0 & 0 \\ Co/2 & 0 & -Co/2 & 0 & 0 \\ 0 & Co/2 & 0 & -Co/2 & 0 \\ 0 & 0 & Co/2 & 0 & -Co/2 \\ 0 & 0 & 0 & Co/2 & 0 \end{bmatrix} \quad (5.45)$$

in the 5x5 case.

Numerical solutions with $Co = 0.1$ and $Co = 1$ are shown in Fig. (5.7). With $Co = 0.1$, the result is as good as with the Crank-Nicholson scheme (Fig. 5.6), and more accurate than with FTCS (Fig. 5.4) or iFTCS (Fig. 5.5). With $Co = 1$, the solution is nearly as good again as with the Crank-Nicholson scheme. But it is weakly unstable, as can be seen from the oscillations forming near $x=1$ at $t=4$. Increasing Co further confirms this, with the solution blowing up earlier.

Thus the RK2 time step is more accurate than the forward Euler for advection problems. We must have $Co < 1$ though to maintain numerical stability. But the higher accuracy comes at a cost: we require two calculations at every time, which doubles the computational time. Indeed, if we

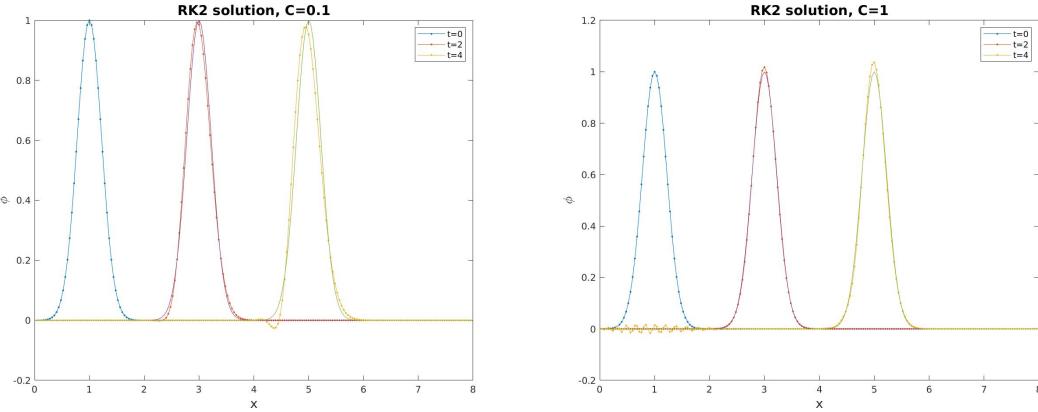


Figure 5.7: Second order Runge-Kutta solutions for the initially Gaussian distribution, with $Co = 0.1$ (left panel) and $Co = 1$ (right panel).

had used the fourth order Runge-Kutta scheme, which is very accurate, the computational time increases by a factor of four.

For simple calculations like this, the increase in computational time isn't a problem. But for more complex calculations, this might be enough of a reason not to use the Runge-Kutta algorithm.

5.1.5 Advection with a variable velocity

The case with a constant velocity is not very realistic (how many rivers do you know which have a constant speed?). But even with simple, variable velocity profiles, an analytic solution can be difficult to find. But this is not a problem at all with a numerical solution.

To illustrate, let's consider the following profile:

$$u = U_0 - \frac{3U_0}{4} \exp\left(-\left(\frac{x - x_s}{L}\right)^2\right) \quad (5.46)$$

This is plotted in Fig. (5.8) for the case with $U_0 = 1$, $x_s = 4$ and $L = 0.5$. The velocities weaken near x_s to 25% of their upstream value, and then strengthen again downstream. This could represent a “stagnation” point in

a river, where the flow swirls and advection downstream is hindered.

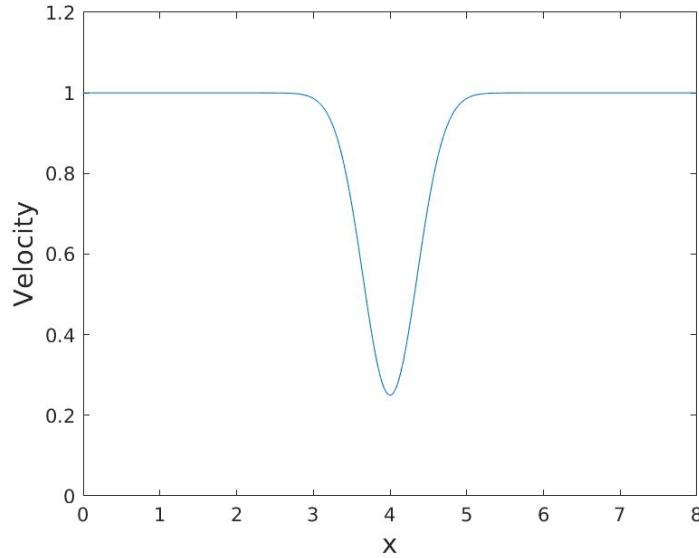


Figure 5.8: A constant flow with a slower velocities near $x = 4$.

Now that u is variable, we'll use the original advection equation (5.4), in divergence form. We'll simulate this with the Crank-Nicholson scheme. With a variable velocity, the difference equation is now:

$$\frac{\phi_j^{n+1} - \phi_j^n}{dt} + \frac{u_{j+1}\phi_{j+1}^{n+1} - u_{j-1}\phi_{j-1}^{n+1}}{4dx} + \frac{u_{j+1}\phi_{j+1}^n - u_{j-1}\phi_{j-1}^n}{4dx} = 0 \quad (5.47)$$

Thus the velocity coefficient is no longer constant, but varying in x . Rearranging, we get:

$$-\frac{Co_{j-1}}{4}\phi_{j-1}^{n+1} + \phi_j^{n+1} + \frac{Co_{j+1}}{4}\phi_{j+1}^{n+1} = \frac{Co_{j-1}}{4}\phi_{j-1}^n + \phi_j^n - \frac{Co_{j+1}}{4}\phi_{j+1}^n \quad (5.48)$$

where:

$$Co_j = \frac{u_j dt}{dx} \quad (5.49)$$

is the (now variable) Courant number.

The matrix version is again:

$$\mathbf{A}\phi^{n+1} = \mathbf{B}\phi^n \quad (5.50)$$

with matrices:

$$\mathbf{A} = \begin{bmatrix} 1 & Co_2/4 & 0 & 0 & 0 \\ -Co_1/4 & 1 & Co_3/4 & 0 & 0 \\ 0 & -Co_2/4 & 1 & Co_4/4 & 0 \\ 0 & 0 & -Co_3/4 & 1 & Co_5/4 \\ 0 & 0 & 0 & -Co_4/4 & 1 \end{bmatrix} \quad (5.51)$$

and

$$\mathbf{B} = \begin{bmatrix} 1 & -Co_2/4 & 0 & 0 & 0 \\ Co_1/4 & 1 & -Co_3/4 & 0 & 0 \\ 0 & Co_2/4 & 1 & -Co_4/4 & 0 \\ 0 & 0 & Co_3/4 & 1 & -Co_5/4 \\ 0 & 0 & 0 & Co_4/4 & 1 \end{bmatrix} \quad (5.52)$$

The case with an initial Gaussian distribution is shown in Fig. (5.9). The initial distribution is given by:

$$p = \exp\left(-\frac{(x' - 1)^2}{0.15}\right)$$

We see that as the tracer approaches the stagnation area, the amplitude grows and the leading edge steepens. When it is in the region, the distribution is very narrow and the amplitude is high, indicating the concentration is large at this point. Then the tracer moves beyond and returns to its original amplitude.

If this had been a toxic chemical, we would conclude the concentrations in the stagnation region might reach dangerous levels. This is due to the combination of a locally slow velocity, and the need to conserve the total concentration. This sort of thing happens routinely in flows, and accounts for local variations in pollutant levels, for example in city air.

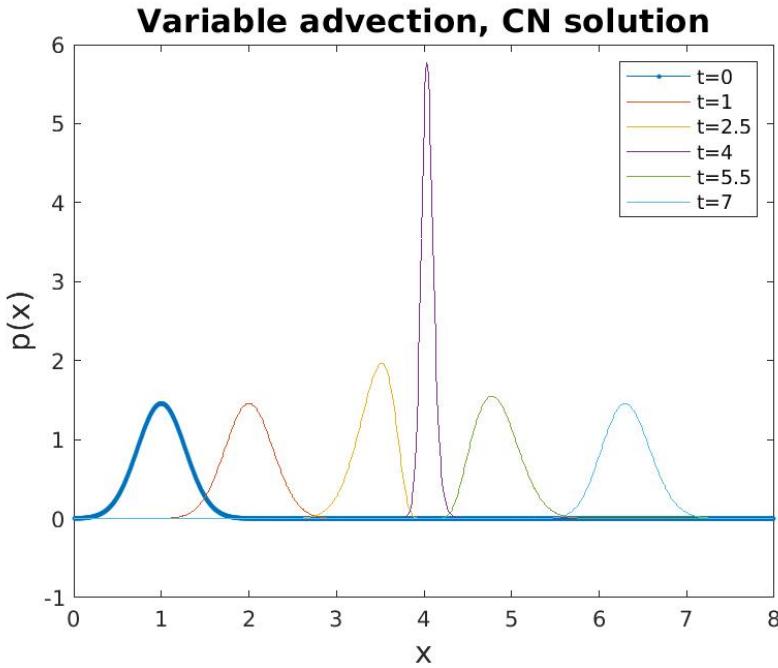


Figure 5.9: Advection of a tracer with a Gaussian distribution and the velocity shown in Fig. 5.8).

5.2 Summary

Advection is very common in geophysical systems, from rivers to the atmosphere, to lava flow below the surface. Advection results in *transport* of anomalies. If the fluid velocity is constant, the anomalies translate without losing their shape. But most often the advecting velocity is not constant, and then the anomalies can be compressed and stretched. In 2D and 3D flows, this can be complex and even beautiful, as the tracer is contorted into complex structures.

Pure advection, even with a constant velocity, can be demanding on numerical schemes. The FTCS scheme is unstable, developing oscillations for small values of the Courant number, Co , and blowing up if $Co > 1$. The iFTCS scheme is numerically stable, but anomalies tend to spread out in time. The second order Runge-Kutta scheme is superior in terms of

accuracy, but also requires $Co < 1$ for numerical stability. The Crank-Nicholson scheme offered the best combination of stability and accuracy of the methods tested here.

Chapter 6

Diffusion

Next we consider diffusive processes. Diffusion is the spreading of material from a region of high concentration to one of low concentration. This could be a passive tracer, like a pollutant, or an active tracer, like momentum.

Diffusion is illustrated in Fig. (6.1). The molecules on the left side of the box are bouncing into each other and against the walls of the container. Occasionally they cross the center line and move into the region on the right, where there were no molecules initially. Eventually we expect the molecules will uniformly fill the whole container.

Diffusion processes are widespread in the geosciences, from the spreading of pollutants in an aquifer, of rocks in a landslide, to the dispersion of radioactive elements in the atmosphere. Many of the equations used in geophysical systems have diffusive terms in them.

6.1 The diffusion equation

When there are only a few particles, as in the figure, the process of spreading out proceeds randomly, with a few particles crossing the center line,

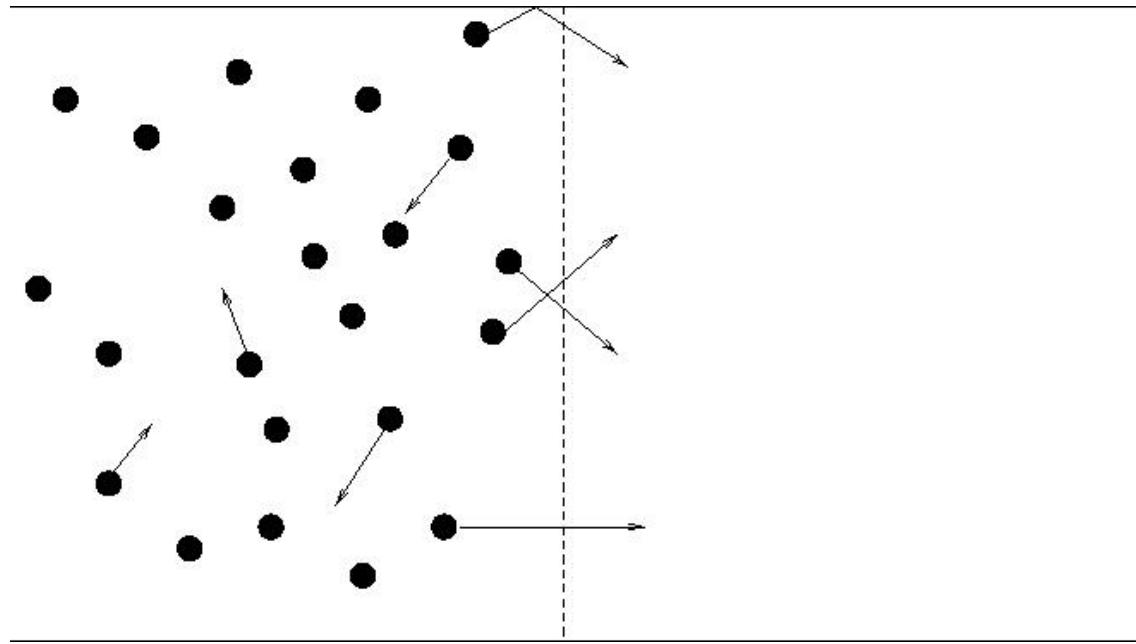


Figure 6.1: Diffusion of particles from a region of high concentration to one of low concentration.

then a few more, etc. But where there are many, many particles, the spreading is more uniform. Observations and experiments suggest that the *flux of particles is proportional to the gradient of the particle concentration*:

$$F = -k\nabla C \quad (6.1)$$

This is known as ‘‘Fick’s Law’’ after Fick (1855). In one dimension, this is just:

$$F = -k \frac{dC}{dx} \quad (6.2)$$

In Fig. (6.1), the concentration gradient is negative, because there are fewer particles moving towards positive x . Thus the flux in (6.2) is positive, so that the particles are spreading to the right.

There are similar laws for other diffusing fields. For instance, when the diffusing substance is temperature, we have *Fourier’s law* (1822):

$$F = -k \frac{dT}{dx} \quad (6.3)$$

When the substance is momentum in a porous medium (like an aquifer), this is *Darcy's law* (1856):

$$F = -\frac{k}{\mu} \frac{dp}{dx} \quad (6.4)$$

Here p is the pressure and k and μ are constants.

We can apply the flux concept with a control volume, as that in Fig. (5.2). But now the fluxes are diffusive rather than advective. The diffusive flux on the left side is:

$$F_l = -k \frac{\partial}{\partial x} C(x)$$

while on the right side, it is:

$$F_r = -k \frac{\partial}{\partial x} C(x + dx)$$

Expressing this as a Taylor series, we get:

$$F_r = -k \frac{\partial}{\partial x} C(x + dx) \approx -k \frac{\partial}{\partial x} C(x) - \frac{\partial}{\partial x} (k \frac{\partial}{\partial x} C(x)) dx - \dots$$

The dots indicate higher order terms, which we neglect. Using these fluxes, we can write an equation for the total amount of C in the box:

$$\begin{aligned} \frac{d}{dt} \iiint C dV &= Adx \frac{\partial}{\partial t} C = F(x)A - F(x + dx)A \\ &= -kA \frac{\partial}{\partial x} C(x) + kA \frac{\partial}{\partial x} C(x) + \frac{\partial}{\partial x} (k \frac{\partial}{\partial x} C(x)) Adx \end{aligned} \quad (6.5)$$

As before, we assume the volume is small enough so that the concentration inside is approximately constant.

Canceling terms, we obtain the 1D diffusion equation:

$$\frac{\partial}{\partial t} C = \frac{\partial}{\partial x} (k \frac{\partial}{\partial x} C) \quad (6.6)$$

If we include diffusive fluxes on the other sides of the volume, we would get the 3D version of the equation:

$$\frac{\partial}{\partial t}C = \frac{\partial}{\partial x}(k\frac{\partial}{\partial x}C) + \frac{\partial}{\partial y}(k\frac{\partial}{\partial y}C) + \frac{\partial}{\partial z}(k\frac{\partial}{\partial z}C) \quad (6.7)$$

or, in more compact form:

$$\frac{\partial}{\partial t}C = \nabla \cdot (k\nabla C) \quad (6.8)$$

This is the full diffusion equation.

The factor, k , is called the “diffusivity”. This determines how quickly the material spreads out. The diffusivity may not be constant, i.e. it can vary in (x, y, z) . If it is constant, the equations take a somewhat simpler form:

$$\frac{\partial}{\partial t}C = k\frac{\partial^2}{\partial x^2}C \quad (6.9)$$

and:

$$\frac{\partial}{\partial t}C = k\frac{\partial^2}{\partial x^2}C + k\frac{\partial^2}{\partial y^2}C + k\frac{\partial^2}{\partial z^2}C = k\nabla^2C \quad (6.10)$$

6.2 Analytical solutions

Consider temperature diffusing along a metal rod (Fig. 6.2). The rod is initially 10C. Then the left end is suddenly cooled to 0C by putting it in contact with an ice bath. At the same time, the right side is heated, increasing the temperature to 30C (right panel). Heat then diffuses along the rod, changing the temperature profile (right panel).

We can model the temperature as a 1D diffusion process:

$$\frac{\partial}{\partial t}T = k\frac{\partial^2}{\partial x^2}T \quad (6.11)$$

Initially, the temperature is nearly uniform along the rod, except near the ends. But then rod warms near the right end, and cools near the left. The change migrates toward the middle of the rod. Eventually the temperature has a linear profile, changing from 0C on the left to 30C on the right.

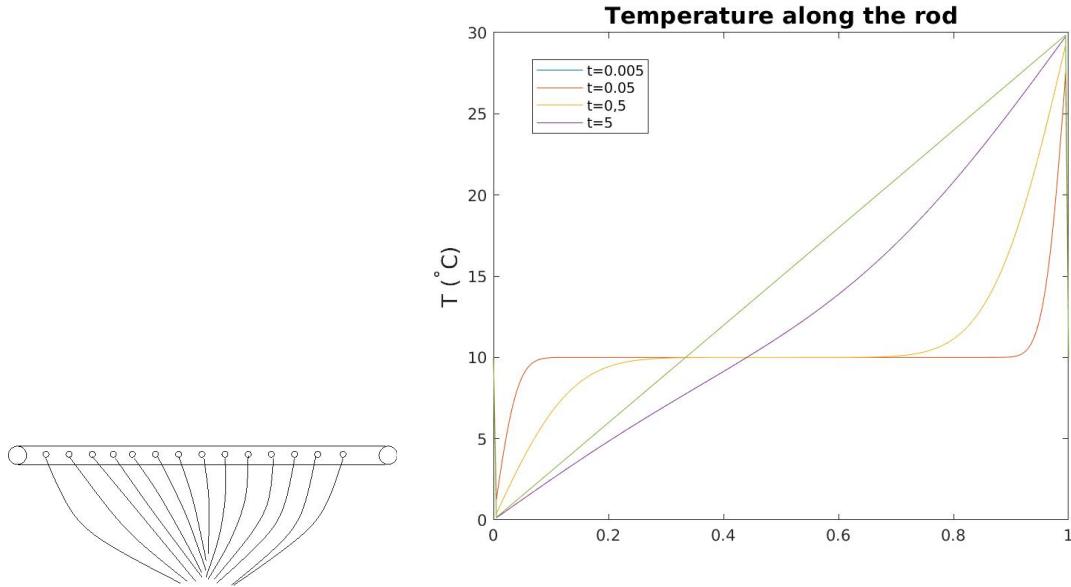


Figure 6.2: Temperature diffusion along a metal rod. Left panel: the temperature is initially 10C, then the temperature on left side is suddenly reduced to 0C (from an ice bath, for example) and the temperature on the right increased to 30C. The sensors then record how the temperature along the rod changes in time. Right panel: the rod temperature at different times.

The linear profile is the steady state solution for the problem. To see that, we set the time derivative to zero in eq. (6.11). Then we have:

$$\frac{\partial^2}{\partial x^2} T = 0 \quad (6.12)$$

The solution is:

$$T = Ax + B \quad (6.13)$$

We obtain the constants by matching the two boundary conditions, $T(0) = 0$ and $T(1) = 30$, to obtain:

$$T = 30x \quad (6.14)$$

Notice this profile is determined solely by the boundary conditions at the end. The initial temperature is completely lost. This means that a diffusion process cannot be run backwards in time, because many different initial conditions yield the same final state.

This evolution is common with all diffusion problems. There is an initial adjustment, as the system responds to changes in forcing or in the boundary conditions. Eventually the system settles into a steady state.

6.2.1 Self-similar solution

The diffusion equation can be solved using a “self-similar solution”.¹ This is a useful technique to know, as it is applicable in a wide range of problems. With it, we can convert a PDE into an ODE, and the ODE is usually much easier to solve. Significantly, a self-similar solution *retains its shape* as it evolves. Initially the solution has some form, determined by the initial conditions. But after some time, the solution settles into the self-similar form.

To see how this works, let’s *scale* the diffusion equation, thus:

$$\frac{\partial}{\partial t} C = D \frac{\partial^2}{\partial x^2} C$$

$$\frac{\Delta C}{T} = \frac{D \Delta C}{L^2} \quad (6.15)$$

The left side scales as the change in concentration divided by time. For example, T could be the amount of time required for C to decrease to half its initial value. The right side scales as the change in concentration multiplied by the diffusivity and divided by L^2 (because there are two derivatives). Here L is for example the width of a pollutant plume. Since the two

¹This section is more mathematical than most in this course(!) We include it to demonstrate how to solve a diffusion equation. But don’t be too worried about the details.

sides are equal, we obtain a scaling for the spatial scale, X :

$$L \propto (DT)^{1/2} \quad (6.16)$$

This indicates how much the tracer spreads in a given time, T . Notice that the increase is less than linear. For the pollutant cloud to double in size requires the time to increase by a factor of four. This square root dependence is a characteristic of diffusion problems.

Given this, we can define a *non-dimensional parameter*, $\eta = x/(Dt)^{1/2}$. We see that η has no units (because Dt has units of m^2 , so that its root has the same units as x in the numerator). For a self-similar evolution, the concentration function should only be a function of η :

$$C = f(\eta) = f\left(\frac{x}{\sqrt{Dt}}\right) \quad (6.17)$$

where f is some function, to be determined later.

What does this mean? The concentration, C , is spreading out in time. But if we plot C as a function of η , it will retain its shape. That's why we say it is "self-similar". $f(\eta)$ remains unchanged.

The solution as it stands is incomplete, because C may have a prefactor as well. To find this, consider the total concentration in the domain:

$$\int_{-\infty}^{\infty} C(x, t) dx = const. \quad (6.18)$$

This is constant if there are no sources. Changing variables to η , we obtain:

$$\sqrt{Dt} \int_{-\infty}^{\infty} C(\eta) d\eta = const. \quad (6.19)$$

In order for the whole LHS to be constant, we require:

$$C \propto (Dt)^{-1/2}$$

Thus the self-similar form of C is:

$$C = \frac{1}{\sqrt{Dt}} f(\eta) \quad (6.20)$$

Using this, we can convert the diffusion equation, a PDE, into an ODE.

First, we have:

$$\frac{\partial}{\partial t} C = -\frac{1}{2\sqrt{Dt^{3/2}}} f - \frac{1}{\sqrt{Dt}} f' \frac{\partial \eta}{\partial t} = -\frac{1}{2\sqrt{Dt^{3/2}}} f - \frac{x}{2Dt^2} f' \quad (6.21)$$

The prime again indicates a derivative with respect to the argument, in this case η . We also have:

$$\frac{\partial}{\partial x} C = \frac{1}{\sqrt{Dt}} f' \frac{\partial \eta}{\partial x} = \frac{1}{Dt} f' \quad (6.22)$$

and:

$$\frac{\partial^2}{\partial x^2} C = \frac{1}{(Dt)^{3/2}} f'' \quad (6.23)$$

Thus the diffusion equation is:

$$-\frac{1}{2\sqrt{Dt^{3/2}}} f - \frac{x}{2Dt^2} f' = \frac{D}{(Dt)^{3/2}} f''$$

or:

$$f'' = -\frac{1}{2} f - \frac{x}{2\sqrt{Dt}} f' = -\frac{1}{2} f - \frac{\eta}{2} f'$$

The RHS is just a total derivative, so we have:

$$f'' = -\frac{d}{d\eta} \left(\frac{\eta}{2} f \right) \quad (6.24)$$

This is equivalent to the diffusion equation. But, as advertised, it's an ODE. The solution will depend on the boundary conditions and the initial condition. We'll look at a couple of examples next.

6.2.2 A point source: an oceanic spill

In 2010, there were two catastrophic pollution events. One was the eruption of the Icelandic volcano, Eyjafjallajökull. Ash spewed into the atmosphere and subsequently spread out, preventing airlines from flying over western Europe. The second was when British Petroleum's Deepwater Horizon oil rig exploded, releasing large amounts of oil into the Gulf of Mexico (Fig. 6.3).



Figure 6.3: The explosion at the Deep Water Horizon oil rig in the Gulf of Mexico in 2010 (upper panel). A satellite photo of the plume subsequently spreading out in the northern Gulf. From Wikipedia.

We will represent such an event as one dimensional spreading from an initial “point source”. The spill will have occurred at $x = 0$, and we’ll plot

how the slick spreads in time. To solve such an initial value problem, we integrate equation (6.24) once in space, yielding:

$$\frac{df}{d\eta} = -\frac{\eta}{2}f(\eta) + C \quad (6.25)$$

where C is a constant of integration. We can set this to zero, because we don't have any oil at infinity.

Then we integrate the equation again, thus:

$$\frac{df}{f} = -\frac{\eta}{2}d\eta \quad (6.26)$$

This yields:

$$\ln(f) = -\frac{\eta^2}{4} + A \quad (6.27)$$

where A is another constant of integration. Taking the exponential of both sides, we have:

$$f = B \exp\left(-\frac{\eta^2}{4}\right) \quad (6.28)$$

where $B = \exp(A)$ is another constant. Then, using this in (6.20), we obtain:

$$C = \frac{B}{\sqrt{Dt}} \exp\left(-\frac{\eta^2}{4}\right) = \frac{B}{\sqrt{Dt}} \exp\left(-\frac{x^2}{4Dt}\right) \quad (6.29)$$

We finish the problem by finding the constant, B . We do this by integrating over the whole area, and matching to the initial concentration. As noted, we'll assume the spill occurred at a point. Then we could represent the initial concentration as a *delta function*, $\delta(x)$. A delta function is non-zero only where the argument is zero, here at $x = 0$. The function has infinitesimal thickness, but it also has an infinite amplitude(!). The amplitude is such that the area under the function is identically one:

$$\int_{-\infty}^{\infty} \delta(x) dx = 1 \quad (6.30)$$

Thus we write the initial concentration as:

$$C(x, 0) = C_0 \delta(x) \quad (6.31)$$

Integrating this over all space yields:

$$C_0 \int_{-\infty}^{\infty} \delta(x) dx = C_0 \quad (6.32)$$

Now we can determine the constant, B , in the solution. We write:

$$\int_{-\infty}^{\infty} \frac{B}{\sqrt{Dt}} \exp\left(-\frac{x^2}{4Dt}\right) dx = C_0 \quad (6.33)$$

Making the substitutions:

$$u = \frac{x}{2\sqrt{Dt}}, \quad du = \frac{dx}{2\sqrt{Dt}}$$

we have:

$$2B \int_{-\infty}^{\infty} \exp(-u^2) du = 2B\sqrt{\pi} = C_0$$

Thus:

$$B = \frac{C_0}{2\sqrt{\pi}}$$

So the total solution is:

$$C = \frac{C_0}{2\sqrt{\pi Dt}} \exp\left(-\frac{x^2}{4Dt}\right) \quad (6.34)$$

The solution is plotted in Fig. (6.4). We've set the diffusivity, $D = 1$. The tracer distribution has the familiar bell-shape of a Gaussian function. It spreads rapidly from the initial delta distribution at $x = 0$, becoming flatter as time goes by.

Note that the steady state solution in this case, is just $C = 0$. Diffusion acts to weaken the tracer gradients. So the tracer spreads out to infinity, leaving only a vanishingly thin film behind.

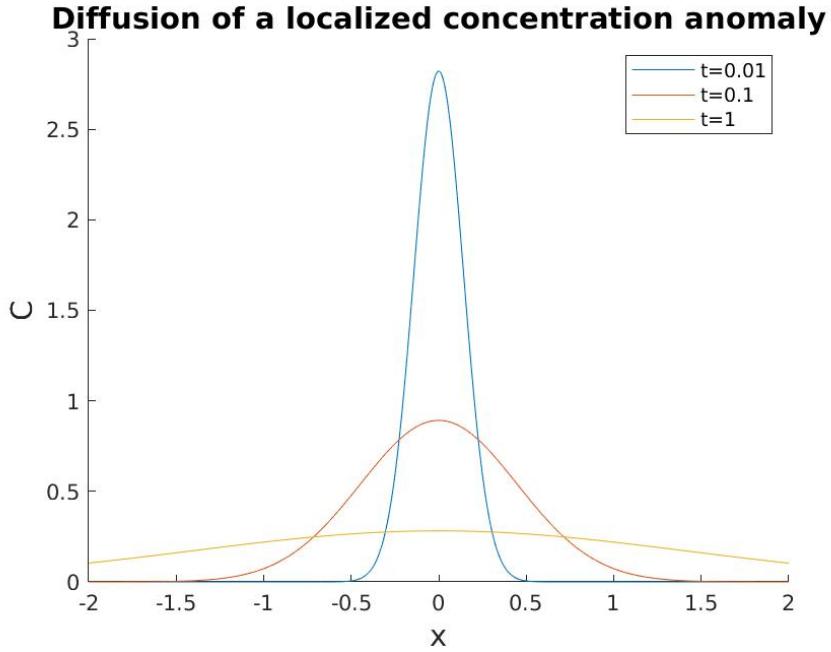


Figure 6.4: The diffusive solution from an initial point source at various times, with a diffusivity, $D = 1$.

Despite appearances, this solution is self-similar. We have:

$$C = \frac{C_0}{2\sqrt{\pi D t}} \exp\left(-\frac{x^2}{4Dt}\right) = \frac{C_0 \eta}{2\sqrt{\pi} x} \exp\left(-\frac{\eta^2}{4}\right) \quad (6.35)$$

At each position x , this has the same profile, when plotted against η .

A valuable metric is how long it takes for the concentration to increase at a given distance from the source. For example, how long would it take for the Icelandic ash to reach London, and how high will the concentrations get there? We can see this by plotting the solution at fixed values of x as a function of time (Fig. 6.5).

At $x = 2$, closest to the spill (at $x = 0$), the concentration rises rapidly, reaching a value of 0.12 near $t = 2$. Thereafter the concentrations fall.

At $x = 4$, the maximum is reached later, nearer $t = 6$, and reaches a maximum which is only half as large. This is the effect of having a finite amount of tracer – the concentration is weaker as it spreads over a larger

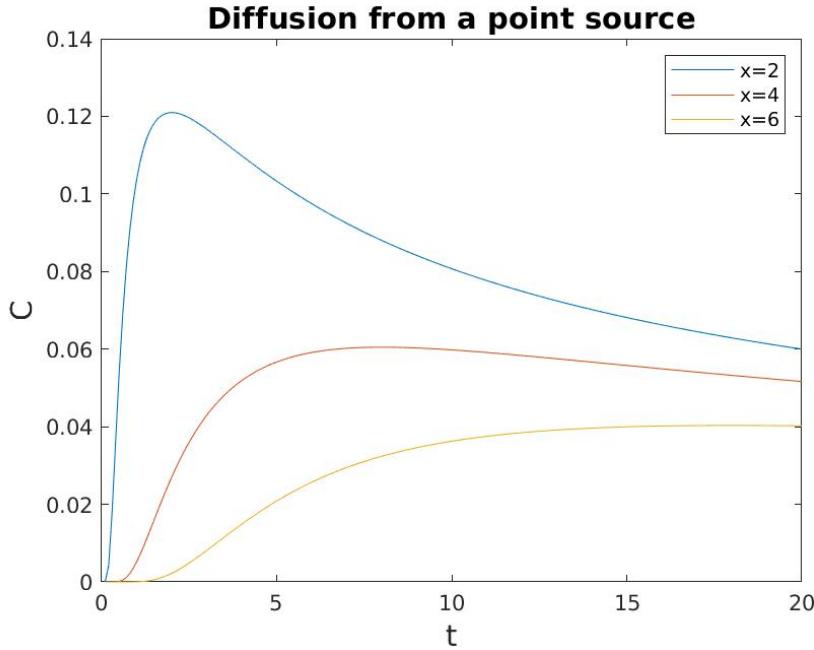


Figure 6.5: The diffusive solution at several distances from the release point (at $x = 0$) as functions of time. Again, $D = 1$.

distance. And the changes at $x = 6$ are consistent, reaching a maximum after $t = 12$ and only attaining a value of $C = 0.04$

6.2.3 A boundary source: flow in an aquifer

On Cape Cod, in the American state of Massachusetts, they discovered that jet fuel was leaking into the ground water around the Otis National Guard Air Base, in the early 1990s. People were obviously concerned, since this was the drinking water for the area. The spill was estimated to have the distribution shown in the right panel of Fig. (6.6). You can see the oil was spreading out in all directions from the central location, the air base.

We'll model this (very approximately) as shown in Fig. (6.7). The oil is introduced with a certain concentration, C_0 (e.g. 100%), on the left side of the aquifer and spreads to the right. The concentration obeys a diffusion

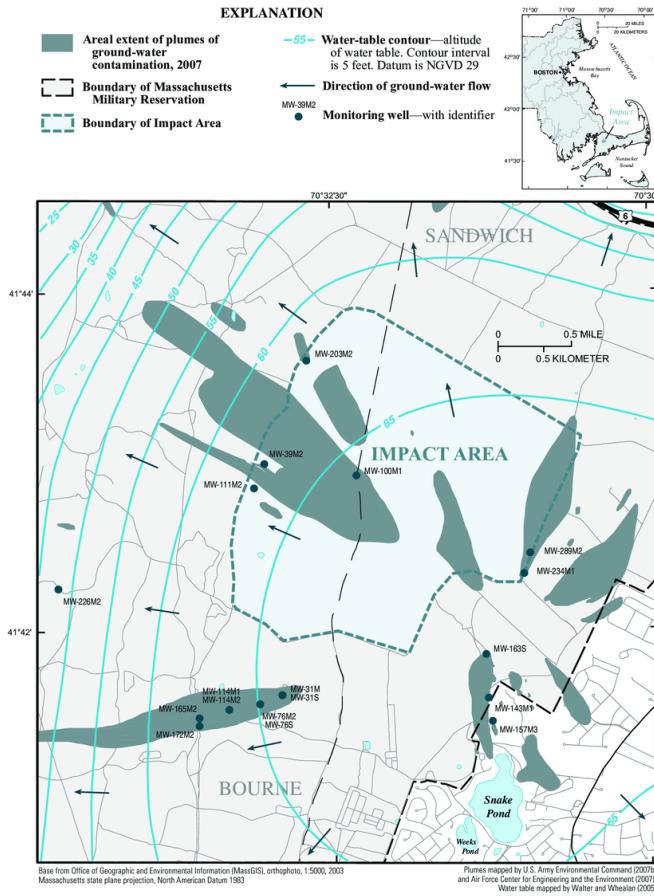


Figure 6.6: Jet fuel spreading in the ground from the Otis Air National Guard base in Massachusetts, USA, in the early 1990s. Courtesy of USGS.

equation, and we again assume the spreading is one dimensional:

$$\frac{\partial}{\partial t} C = D \frac{\partial^2}{\partial x^2} C, \quad C(0, t) = C_0 \quad (6.36)$$

We'll also assume that the aquifer is semi-infinite, so that the oil can continue spreading to the right indefinitely.

The 1D self-similar diffusion equation (6.24) still applies, but the solution is slightly different due to the domain and boundary conditions. It's simpler in this case to define another function:

$$g = \frac{df}{d\eta}$$

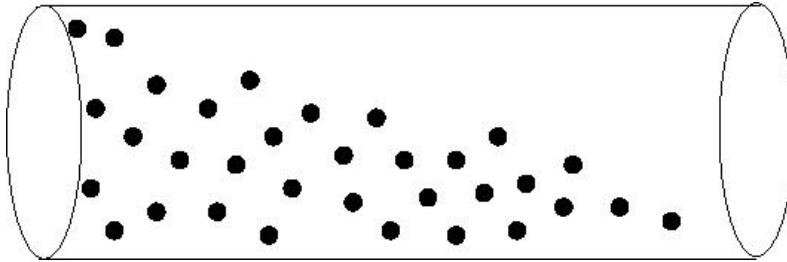


Figure 6.7: An idealized aquifer. Oil introduced at the left end diffuses to the right.

Then equation (6.24) is:

$$\frac{dg}{d\eta} = -\frac{\eta}{2}g$$

This integrates the same as before, leaving:

$$g = A \exp\left(-\frac{\eta^2}{4}\right) = \frac{df}{d\eta}$$

Now, let's integrate this from $\eta = 0$ to η :

$$\int_0^\eta \frac{df}{d\eta} d\eta = f(\eta) - f(0) = \int_0^\eta A \exp\left(-\frac{\eta^2}{4}\right) d\eta$$

Thus:

$$f(\eta) = f(0) + \int_0^\eta A \exp\left(-\frac{\eta^2}{4}\right) d\eta$$

or:

$$f(\eta) = f(0) + A \operatorname{erf}\left(\frac{\eta}{2}\right) \quad (6.37)$$

The function, $\operatorname{erf}(x)$, is the *error function*, defined as:²

$$\operatorname{erf}(x) = \int_0^x e^{-t^2} dt \quad (6.38)$$

This is often used in probability. It is indicated by the blue curve in Fig.(6.8). The function increases from zero and asymptotes to one.

Now that we have $f(\eta)$, we can find the concentration:

$$C = C_0 + A \operatorname{erf}\left(\frac{x}{2\sqrt{Dt}}\right)$$

²The error function is often defined with a prefactor of $2/\sqrt{\pi}$, but we'll use the simpler definition here. Any prefactor would be absorbed into the constant A in any case.

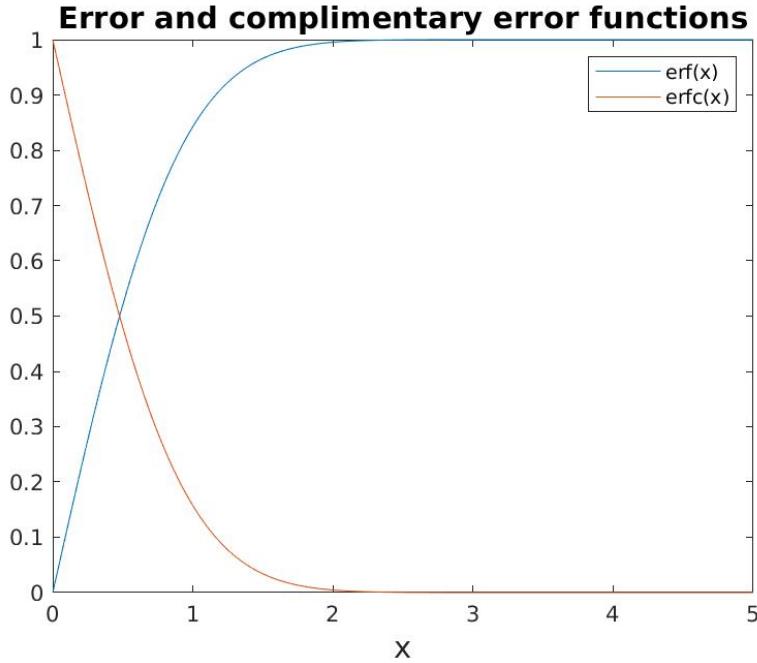


Figure 6.8: The error and complimentary error functions.

where C_0 is the concentration at $x = 0$. We determine the constant A by demanding that the concentration go to zero as $x \rightarrow \infty$. Clearly we must have $A = -C_0$. So the full solution is:

$$C = C_0 \left(1 - \operatorname{erf}\left(\frac{x}{2\sqrt{Dt}}\right)\right) = C_0 \operatorname{erfc}\left(\frac{x}{2\sqrt{Dt}}\right) \quad (6.39)$$

where $\operatorname{erfc}(x)$ is the *complimentary error function* (Fig. 6.8).

Recall that since this is a self-similar solution, i.e.

$$C = C_0 \operatorname{erfc}\left(\frac{\eta}{2}\right)$$

the concentration retains this shape if we plot it as a function of η . But because η itself changes in time, so does the concentration.

The solution is plotted in Fig. (6.9) for different times. We've set the diffusivity to $D = 1$ and plot the solution from $x = 0$ to $x = 2$. At the first time ($t = 0.005$), the oil is concentrated near $x = 0$. But as time increases,

the oil spreads to the right. At long times $t > 5$, the curve is almost linear over the region (which is basically filling with oil).

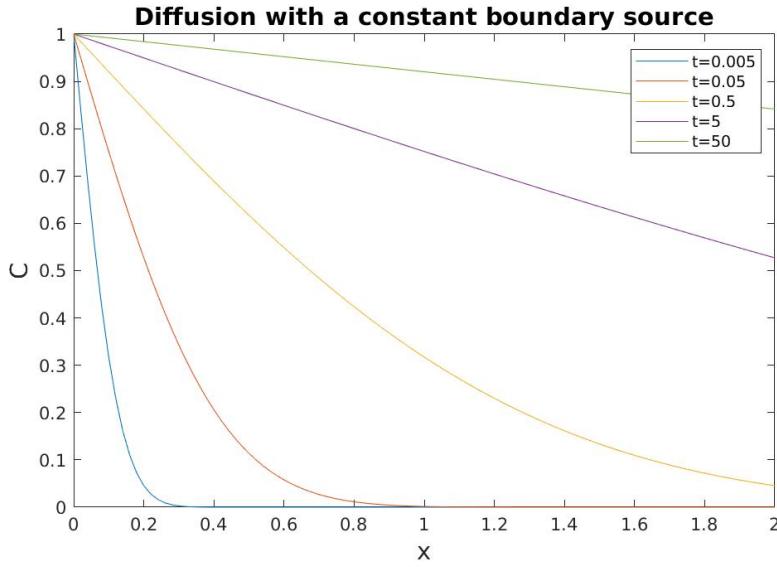


Figure 6.9: The complementary error function solution to the diffusion equation, with $C_0 = 1$. In the left panel is the solution plotted against x at different times.

We can also plot the concentrations at chosen locations, as we did before. This is shown in Fig. (6.10), for the locations $x = 2$, $x = 4$, and $x = 6$. The concentration is initially zero at all three locations but increases steadily thereafter. It increases fastest at $x = 2$, being nearest the source, and slower at the locations further away. In contrast to the concentrations from a point source spill (Fig. 6.5), here the values continue to increase indefinitely. This is because the source at $x = 0$ continually supplies more oil, so eventually the concentration will be 1.0 at all locations. Thus the steady solution (at infinite times) is actually $C = 1.0$ for all positive x .

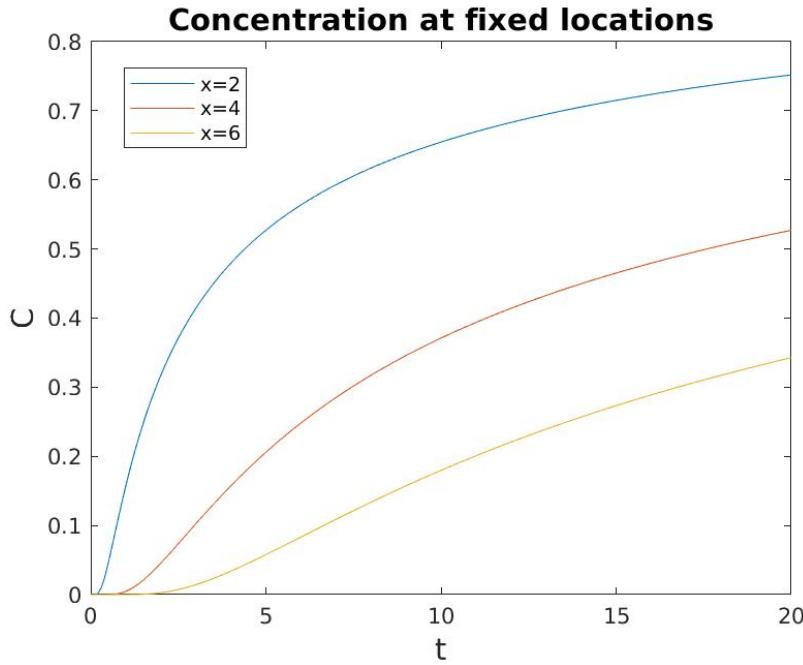


Figure 6.10: The increase in concentration at fixed locations, as a function of time.

6.2.4 Erosion

Mountains are formed by tectonic motion. As the continental plates collide, land is lifted up in dramatic fashion. The Himalayas and the Tibetan plateau formed as the Indian and Eurasian plates collided 50 million years ago, and the uplift continues even today. After the uplift ceases, weathering and erosion smooth out the mountain contours. This is why the Appalachian mountains in North America are smoother than the Rocky mountains, because they were raised earlier and have eroded more.

We'll model erosion also as a diffusive process. Again, we represent this as a 1D process, with the mountain height given by $h(x)$ and governed by a diffusion equation:

$$\frac{\partial}{\partial t} h = D \frac{\partial^2}{\partial x^2} h \quad (6.40)$$

Again, we assume the diffusivity, D , is constant, for simplicity.

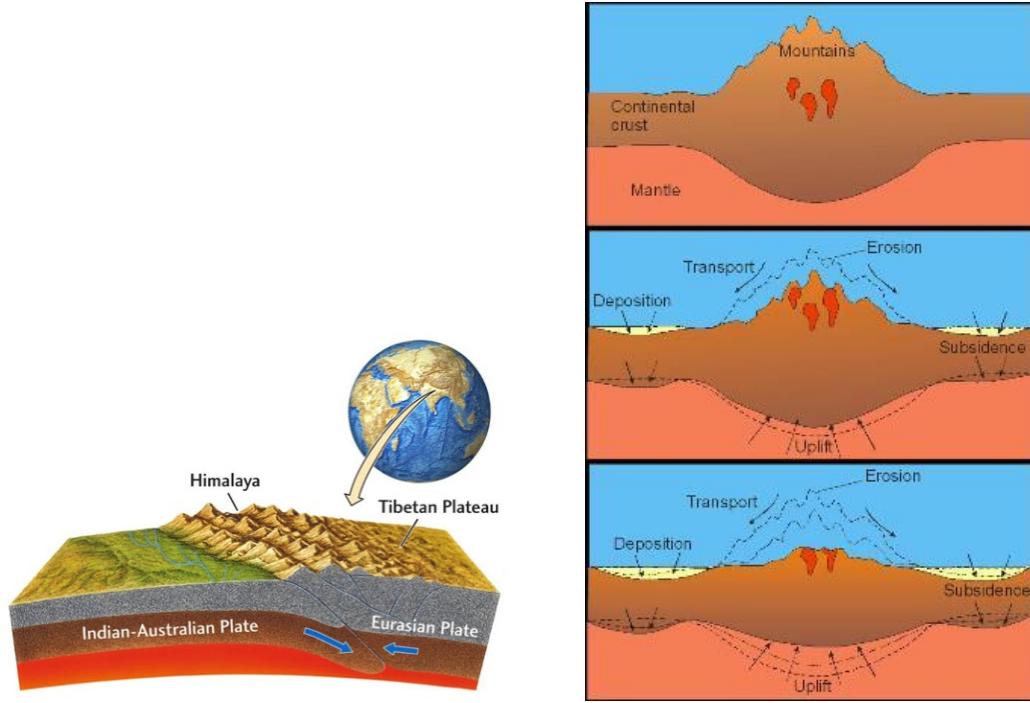


Figure 6.11: The formation of the Himalayas (left), and general uplift and erosion of a mountain range (right). From TCS Academy and the Freie Universität Berlin.

This is somewhat different than the previous two cases, because we have an initial distribution (the uplifted mountain range) which extends over the whole domain. Thus a self-similar solution is less appropriate, because the mountains have a specific shape initially which need not be even close to self-similar. Thus we adopt a different approach.

Let's assume the height is *separable*, specifically that it can be expressed as a product of two functions, one which depends on x and the other on t :

$$h(x, t) = X(x)T(t)$$

Note that X and T aren't scales, as used previously, but functions. Substituting this into the diffusion equation yields:

$$X \frac{\partial}{\partial t} T = DT \frac{\partial^2}{\partial x^2} X \quad (6.41)$$

Dividing both sides by $h = XT$, we get:

$$\frac{1}{T} \frac{\partial}{\partial t} T = \frac{D}{X} \frac{\partial^2}{\partial x^2} X \quad (6.42)$$

Notice that the LHS is only a function of time, while the RHS is a function of x . The only way this can work is if both sides are constant, independent of their respective independent variables. So we can write:

$$\frac{1}{T} \frac{\partial}{\partial t} T = \frac{D}{X} \frac{\partial^2}{\partial x^2} X = -\lambda^2 D \quad (6.43)$$

The reason for choosing this particular constant will become clear below.

The advantage of this is that we now have two ODEs, rather than a PDE, and we can easily solve both. The x equation is:

$$\frac{\partial^2}{\partial x^2} X = -\lambda^2 X \rightarrow X = A \sin(\lambda x) + B \cos(\lambda x) \quad (6.44)$$

While the t equation is:

$$\frac{\partial}{\partial t} T = -\lambda^2 D T \rightarrow T = T_0 e^{-\lambda^2 D t} \quad (6.45)$$

So the total solution for the height, h , is:

$$h(x, t) = (C \sin(\lambda x) + D \cos(\lambda x)) e^{-\lambda^2 D t} \quad (6.46)$$

Note we have absorbed the various constants into C and D .

Since the diffusion equation is linear, we can combine solutions like that above with different values of λ :

$$h(x, t) = \sum_n (C_n \sin(\lambda_n x) + D_n \cos(\lambda_n x)) e^{-\lambda_n^2 D t} \quad (6.47)$$

Then we can determine the constants and the λ_n from the initial condition:

$$h(x, 0) = \sum_n (C_n \sin(\lambda_n x) + D_n \cos(\lambda_n x)) \quad (6.48)$$

This is an example of a *Fourier transform*: we represent the initial condition, $h(x, 0)$, as a sum of sines and cosines. The Fourier transform is a very common technique in physics, geosciences and many other fields.

Consider a simple (idealized) example. Say that the mountain range initially was given by the following function:

$$h(x, 0) = \cos(x) + \cos(3x) + \cos(5x) \quad (6.49)$$

By inspection, we see that $C_n = 0$ for all values of n , and that:

$$D_1 = D_3 = D_5 = 1 \quad (6.50)$$

The other values of D_n are all zero. Thus the full solution is:

$$h(x, t) = \cos(x)e^{-Dt} + \cos(3x)e^{-9Dt} + \cos(5x)e^{-25Dt} \quad (6.51)$$

This gives us an expression for how the mountains evolve in time.

At $t = 0$, the solution reproduces the initial distribution. But as time increases, the three terms decay. However, they decay at different rates, with the smaller scale components decaying more rapidly.

This can be seen in Fig. (6.12), for the case with $D = 1$. The initial range (in blue) has high mountains and valleys, but also smaller scale wiggles. As time progresses, the smaller scale features gradually disappear, leaving the gravest mode, $\cos(x)$. This too is decaying though, and eventually the mountain range will be flat, as erosion carries off all the soil. So as with the oil spill at a point, the steady state solution is just a distribution with $h = 0$.

Thus diffusion *selectively damps smaller scale features*. This can be understood from scaling the diffusion equation, as we did when discussing

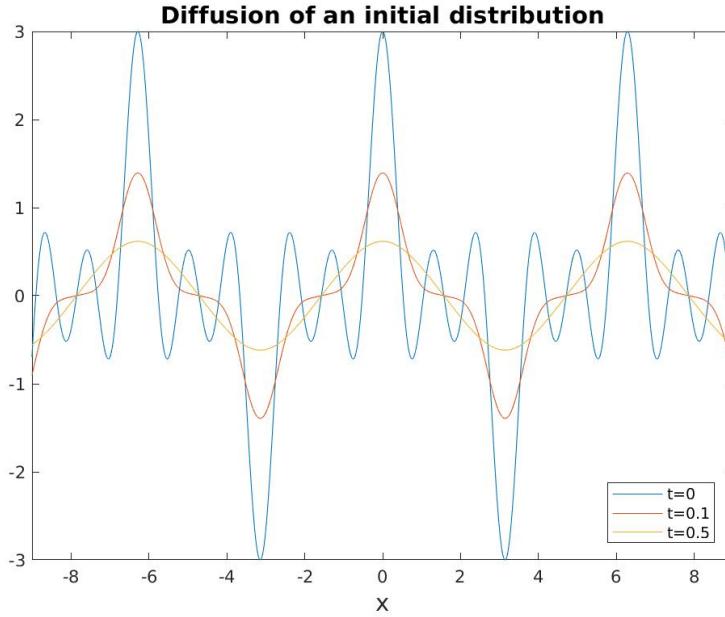


Figure 6.12: Diffusive weathering of an idealized mountain range, with a diffusivity $D = 1$.

self-similar solutions in sec. (6.2.1). There we found that the spatial scale, L , scaled as:

$$L \propto \sqrt{DT}$$

If we invert this, we see the diffusive time scale, T , scales as:

$$T \propto \frac{L^2}{D} \quad (6.52)$$

Thus small scales have short times while large scales have long times. And since the diffusive time scale increases as the length scale squared, the small scales reduce faster. Such selective damping is a characteristic of diffusive systems.

6.3 Numerical solutions

Now we examine numerical solutions of diffusion examples. We'll begin with the second example above, of diffusion in an aquifer with a source on

the left boundary (sec. 6.2.3). Then we'll look at the initial point source solution (sec. 6.2.2), before closing with a diffusive version of the climate model considered in sec. (4.4).

6.3.1 Forward Euler

We'll start again with the forward Euler time step with centered space differences (FTCS). The difference equations have actually been presented previously, in sec. (3.2.1). For the 1D diffusion equation:

$$\frac{\partial}{\partial t}C = D \frac{\partial^2}{\partial x^2}C \quad (6.53)$$

we have:

$$\frac{C_j^{n+1} - C_j^n}{dt} = D \frac{C_{j+1}^n - 2C_j^n + C_{j-1}^n}{dx} \quad (6.54)$$

or:

$$C_j^{n+1} = sC_{j-1}^n + (1 - 2s)C_j^n + sC_{j+1}^n \quad (6.55)$$

with

$$s = \frac{Ddt}{dx^2} \quad (6.56)$$

The Dirichlet boundary conditions, $C(0) = A$ and $C(L) = B$, affect the first and last equations. We'll define $j = 0$ for $x = 0$, and $j = N + 1$ for $x = L$. Then we have:

$$C_1^{n+1} = sC_0^n + (1 - 2s)C_1^n + sC_2^n = sA + (1 - 2s)C_1^n + sC_2^n \quad (6.57)$$

Also:

$$C_N^{n+1} = sC_{N-1}^n + (1 - 2s)C_N^n + sC_{N+1}^n = sC_{N-1}^n + (1 - 2s)C_N^n + sB \quad (6.58)$$

We can then write the system as a matrix equation:

$$\mathbf{C}^{n+1} = \mathbf{AC}^n + \mathbf{b} \quad (6.59)$$

where:

$$\mathbf{A} = \begin{bmatrix} 1 - 2s & s & 0 & 0 \\ s & 1 - 2s & s & 0 \\ 0 & s & 1 - 2s & s \\ 0 & 0 & s & 1 - 2s \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} sa \\ 0 \\ 0 \\ sf \end{bmatrix} \quad (6.60)$$

For the case with four interior grid points. This can be easily generalized to any number of interior grid points.

Let's examine the aquifer solution, shown in Fig. (6.9). Recall this analytical solution is self-similar, defined by the complimentary error function. The concentration is $C = 1$ at $x = 0$ at all times, and the domain shown extends to $x = 2$. As with the analytical solution, we set the diffusivity to $D = 1$.

The FTCS solution is shown in Fig. (6.13). The dots indicate the numerical solutions at different times, while the solid curves show the analytical solutions. At the first time, $t = 0.005$, the two solutions agree well. However by $t = 0.05$, the numerical solution differs, having larger concentrations. The discrepancies grow at $t = 0.5$ and are severe at $t \geq 5$. The $t = 5$ and $t = 50$ numerical solutions actually overly one another here, in contrast to the analytical solutions which differ.

The failure of the FTCS scheme here is not due to numerics. Rather, it is the Dirichlet boundary condition at $x = 2$, where we demand that $C = 0$. The numerical solution thus reaches a steady state with a linear distribution between $x = 0$, where $C = 1$, and $x = 2$, where $C = 0$. The analytical solution in contrast has the concentration increasing at $x = 2$, as oil fills the aquifer

This suggests that the numerical solution might have been better had we used a larger domain. Such a solution, with a domain width of 10, is

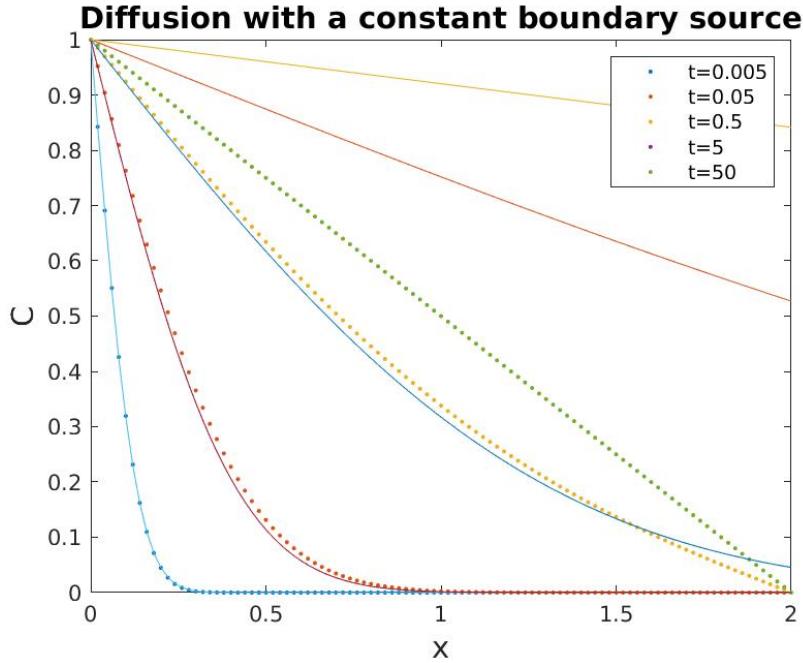


Figure 6.13: FTCS solution for diffusive spreading in a 1D aquifer, with $D = 1$ and $s = 0.4$.

shown in Fig. (6.14). Now the numerical solution is more accurate to nearly $t = 5$, and only shows serious discrepancies by $t = 50$. The latter time is when the concentration at the right boundary again greatly exceeds zero. So this confirms our suspicion that the boundary is the major problem here.

However, the solution at $t = 0.005$ now also differs from the analytical solution. This is due to having a grid spacing dx which is too large to capture the rapid change in the analytical solution. Had we used a smaller grid spacing, we would have resolved this better. But that would also require using a smaller time step, dt , in order to keep s small. Indeed, if the grid spacing is halved, we must decrease dt by a factor of four to maintain the same value of s . Thus the integration is four times longer.

What if s is larger? Shown in Fig. (6.15) is the diffusive solution with

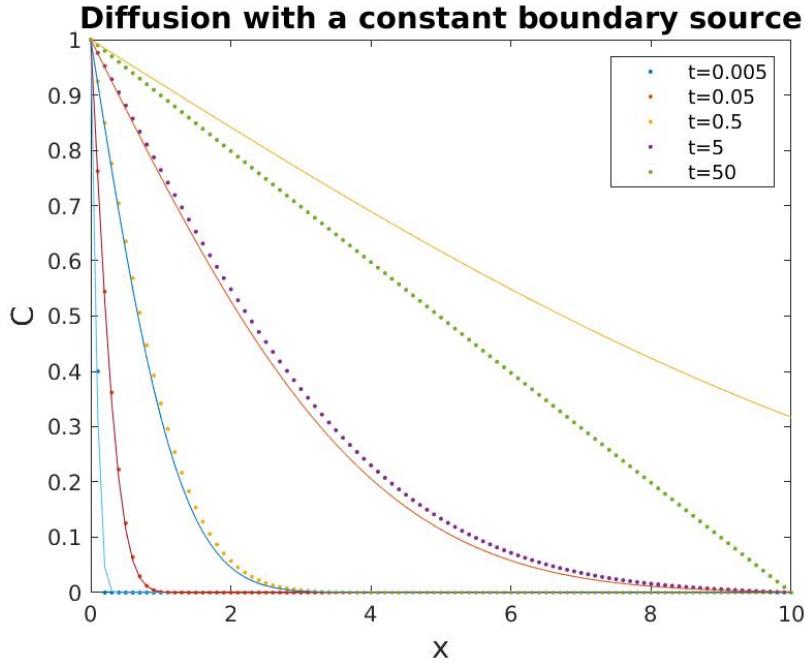


Figure 6.14: The FTCS solution for diffusive spreading with a larger domain. Again, $s = 0.4$.

$s = 0.55$. At the short times, there are several dots near the analytical solution. But by $t = 0.5$, these are mostly gone. The reason is that the numerical values no longer lie in the domain, indicating the solution has “blown up”. The transition to numerical instability is fairly sharp. The solution is stable with s just below 0.5, but unstable if s is slightly greater than 0.5.

Recall what happened when we used the FTCS scheme with pure advection (Chapter 5). Then the solution was *always* unstable – the instability simply grew faster if the Courant number, C , was larger. The FTCS scheme with diffusion is provably stable, so long as $s < 0.5$. For details, see Appendix B.

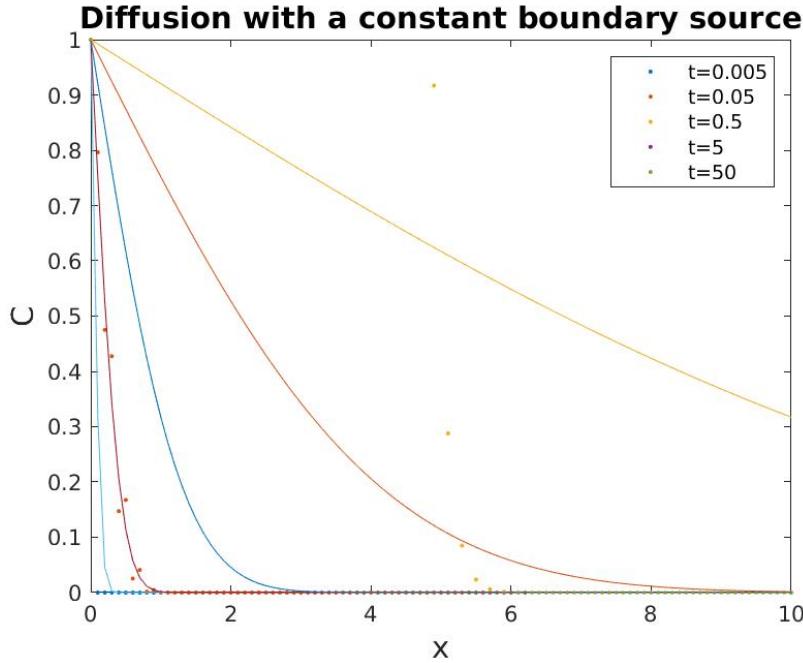


Figure 6.15: FTCS solution for diffusive spreading with $s = 0.55$.

6.3.2 Implicit Euler

Numerical instability can again be avoided using an implicit time step (sec. 3.2.4). Then we write the finite difference equation as in eq. (3.27):

$$-s\phi_{j-1}^{n+1} + (1 + 2s)\phi_j^{n+1} - s\phi_{j+1}^{n+1} = \phi_j^n \quad (6.61)$$

with a corresponding matrix equation:

$$\mathbf{A}\mathbf{C}^{n+1} = \mathbf{C}^n + \mathbf{b} \quad (6.62)$$

Here:

$$\mathbf{A} = \begin{bmatrix} 1 + 2s & -s & 0 & 0 \\ -s & 1 + 2s & -s & 0 \\ 0 & -s & 1 + 2s & -s \\ 0 & 0 & -s & 1 + 2s \end{bmatrix} \mathbf{b} = \begin{bmatrix} sa \\ 0 \\ 0 \\ sf \end{bmatrix} \quad (6.63)$$

assuming again 4 interior grid points. Notice that \mathbf{A} differs from before, but the \mathbf{b} is the same. The last step is then to invert \mathbf{A} , leaving:

$$\mathbf{C}^{n+1} = \mathbf{A}^{-1}\mathbf{C}^n + \mathbf{A}^{-1}\mathbf{b} \quad (6.64)$$

This equation can then be stepped forward in time.

The result, with $s = 0.5$ and the larger domain (from $x = 0$ to $x = 10$) is shown in the left panel of Fig. (6.16). The solutions are very similar to those with the forward Euler scheme (Fig. 6.14). The solution at the earliest time ($t = 0.005$) is poor, because the resolution is insufficient. But at $t = 0.05$, the numerical solution agrees closely with the analytical one. At $t = 0.5$, differences become apparent and are worse by $t = 5$. And at $t = 50$, the iFTCS solution again shows the linear steady state, going to zero at $x = 10$, in contrast with the analytical solution which has a non-zero value there.

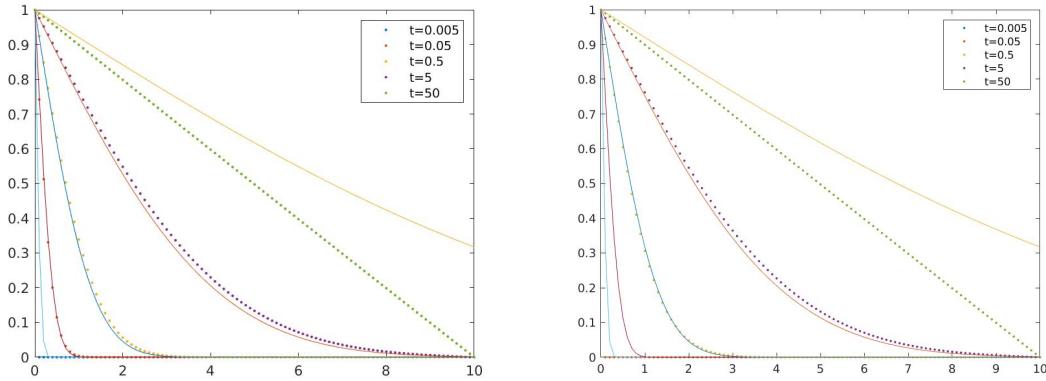


Figure 6.16: The iFTCS solution for diffusion with a boundary source with $s = 0.5$ (left panel) and $s = 5$ (right panel).

What's remarkable about the implicit solution is that we obtain nearly the same results with $s = 5$ (right panel of Fig. 6.14)! In particular, the results at $t = 0.5$ and $t = 5$ are nearly identical to before. What differs is the solution at $t = 0.05$, which is unresolved with this grid spacing ($dx = 0.1$). However, like the $s = 0.5$ solution, the solution deviates from the analytical one when the curve reaches $x = 10$.

The solutions with different values of s are compared in Fig. (6.17). For

the most part, the solutions are very similar. Only the $s = 0.5$ solution resolves the behavior at $t = 0.05$ correctly, but otherwise the agreement is good. Significant deviations at $t = 5$ are found with the $s = 50$ solution. But remember that the time step is now 100 times bigger than with $s = 0.5$, meaning the calculation takes 1/100th the amount of time!

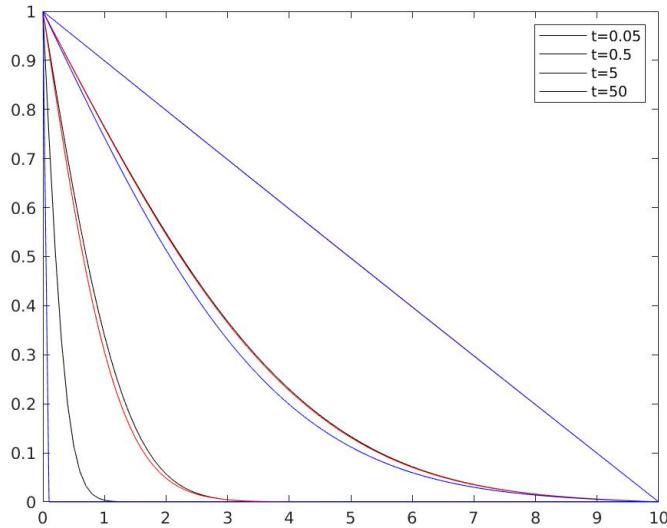


Figure 6.17: The iFTCS solution for boundary diffusion with three values of s : $s = 0.5$ (in black), $s = 5$ (in red) and $s = 50$ (in blue).

Thus the iFTCS solution is stable for all values of s . It is however less accurate for larger values of s . If speed is of the essence, then the iFTCS solution is preferred because of the larger time step. But we have to be careful about the loss of accuracy.

6.3.3 Initial point source

Let's now test the iFTCS method on the case with a spill in the middle of the domain. The analytical solution was given in sec. (6.2.2) and plotted in Fig. (6.4). The tracer had a self-similar Gaussian form, and spread rapidly in time.

In the analytical solution we assumed the initial condition had a delta function form, for example:

$$C = C_0 \delta(x - 1)$$

We will approximate this numerically by the following form:

$$C = \begin{cases} C_0/dx & \text{if } 1 - dx < x < 1 \\ 0 & \text{if } x < 1 - dx \text{ or } x > 1 \end{cases} \quad (6.65)$$

Then note that the domain-integrated concentration is:

$$\int_0^L C dx = C_0/dx \int_{1-dx}^1 dx = C_0 \quad (6.66)$$

Thus our initial condition is actually a rectangular block rather than a delta function (perhaps a Lego version of a delta function). As seen though, this will suffice.

The iFTCS scheme is exactly as in sec. (6.3.2), except for the boundary conditions. We'll test two, one Dirichlet and one Neumann. Assume the domain extends from $x = 0$ to $x = 2$, as shown in the figure. For the Dirichlet, we'll take $C = 0$ at the end points. Then eq. (6.63) applies, but with $a = f = 0$. Thus the vector, \mathbf{b} , is simply:

$$\mathbf{b} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (6.67)$$

and we simply calculate:

$$\mathbf{C}^{n+1} = A^{-1} \mathbf{C}^n \quad (6.68)$$

at each time step.

The result is shown in Fig. (6.18), with $s = 0.5$. The solutions at $t = 0.005$ and $t = 0.05$ are in excellent agreement with the analytical solution.

Note that this is despite the difference in initial conditions (block vs. delta function).

At the later times though, discrepancies appear. Differences at the boundaries are evident at $t = 0.5$, but by $t = 5$, the analytical solution is greater than the numerical one over the whole domain. At $t = 50$, the numerical solution is nearly zero while the analytical one (the lower blue curve) is positive, with $C \approx 0.1$.

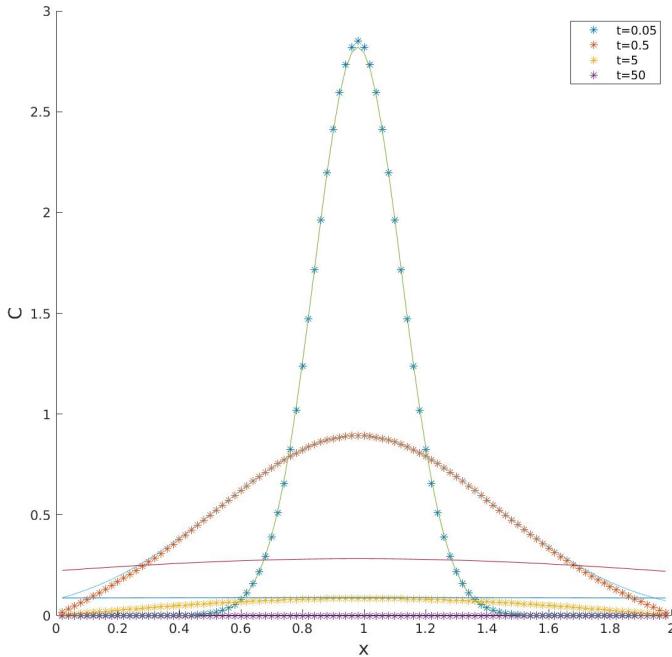


Figure 6.18: The iFTCS solution with $s = 0.5$ for an initial point distribution, at $x = 0$, and Dirichlet boundary conditions. The numerical solutions are indicated by dots and the analytical ones by the solid curves.

The discrepancies again stem from the boundary conditions. The numerical solution has $C = 0$ at both $x = 0$ and $x = 2$, and as such, evolves towards zero over the whole domain. The analytical solution is not so constrained and thus C can take on any value at the ends. At very long times, the distribution becomes thinner and thinner, and spreads over a larger

range of x . But the numerical solution reaches zero sooner, because of the boundary conditions.

For contrast, we impose Neumann conditions instead. In particular, we'll set:

$$\frac{dC}{dx} = 0 \quad \text{at } x = 0, 2$$

This is actually a very different boundary condition to the Dirichlet.

How do we implement this? As always, it helps to write out the equations at the ends. The finite difference equation (eq. 3.27) with $j = 1$ is:

$$-s\phi_0^{n+1} + (1 + 2s)\phi_1^{n+1} - s\phi_2^{n+1} = \phi_1^n \quad (6.69)$$

If the gradient at $x = 0$ is zero, then $\phi_0^n = \phi_1^n$. So the equation becomes:

$$(1 + s)\phi_1^{n+1} - s\phi_2^{n+1} = \phi_1^n \quad (6.70)$$

Similarly, at the last grid point:

$$-s\phi_{N-1}^{n+1} + (1 + 2s)\phi_N^{n+1} - s\phi_{N+1}^{n+1} = \phi_N^n \quad (6.71)$$

or:

$$-s\phi_{N-1}^{n+1} + (1 + s)\phi_N^{n+1} = \phi_N^n \quad (6.72)$$

Thus the matrix A changes to:

$$A = \begin{bmatrix} 1+s & -s & 0 & 0 \\ -s & 1+2s & -s & 0 \\ 0 & -s & 1+2s & -s \\ 0 & 0 & -s & 1+s \end{bmatrix} \quad (6.73)$$

for the case with four interior grid points. The vector, b , is also zero. So we invert the new A and step forward as before:

$$C^{n+1} = A^{-1}C^n \quad (6.74)$$

The result is shown in Fig. (6.19). Again, the analytical and numerical solutions agree well at the first two times, but differ at the last two times. While the analytical solution decays slowly to $C = 0$, the numerical one flattens out to a value of $C = 0.3$. This happens relatively early as well. The solution at $t = 0.5$ is the nearly same as that at $t = 5$.

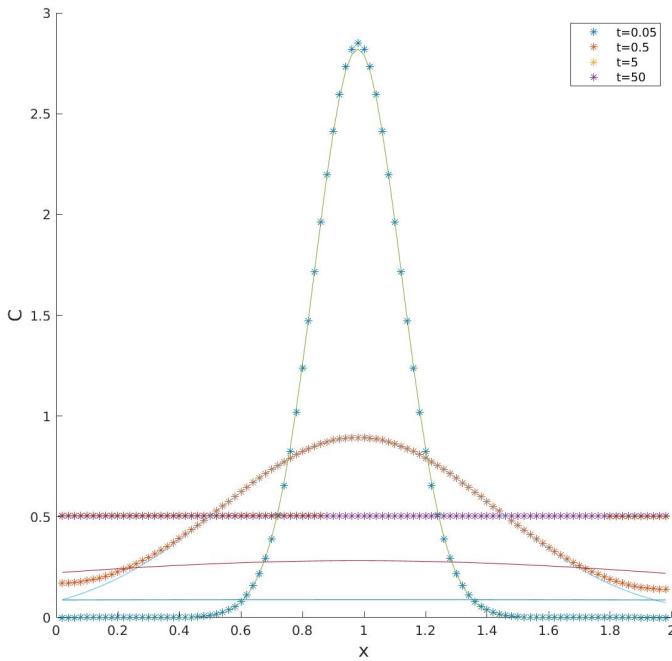


Figure 6.19: The iFTCS solution with $s = 0.5$ for an initial point distribution, at $x = 0$, with Neumann boundary condition. The numerical solutions are indicated by circles and the analytical ones by the solid curves.

What's happening here? With the Neumann condition $\frac{dC}{dx} = 0$ at the end, there is *zero flux* of tracer through the ends. That means that the total tracer in the domain is *conserved*. To see this, we integrate the diffusion equation across the domain:

$$\int_0^2 \frac{\partial}{\partial t} C dx = D \int_0^2 \frac{\partial^2}{\partial x^2} C dx = D \frac{\partial}{\partial x} C|_0^2 = 0 \quad (6.75)$$

So we have:

$$\frac{d}{dt} \int_0^2 C dx = 0 \quad (6.76)$$

which implies the total concentration in the domain doesn't change.

In the numerical solution, diffusion flattens the distribution, but since no tracer can leave the domain, it asymptotes to C_0 , the domain-average concentration (which is 0.5 in this case). The analytical solution in contrast can flux material out of the domain. So it obviously does not conserve the total concentration in the domain.

To summarize, the boundary conditions matter for these simulations. This is not the case before the tracer reaches the walls. But thereafter, the numerical solutions differ markedly from the analytical. The Dirichlet conditions prevent the concentration from being non-zero at the walls, while the Neumann conditions artificially cause the total concentration to be conserved.

This is worth knowing. In practice, we can always use a larger domain, to prolong the period before the tracer reached the boundaries. On the other hand, if the spreading actually occurs in a region with boundaries – for example, in a lake – then it would be desirable to include boundary conditions. In any case, it's worth testing the solutions carefully, to make sure they are physical.

6.3.4 A diffusive climate model

Now we return to the climate model, discussed in sec. (4.4). Recall what we found before. While the model was able to obtain a reasonable estimate for the mean atmospheric temperature, it did poorly at representing the latitudinal variation. The equator was too warm, and the poles were

much too cold (Fig. 4.11). We suggested that the failure was due to a lack of storms, which act to transport heat from low to high latitudes. We will not explicitly model storms (which requires much more complicated equations), but rather we will represent their effect via diffusion.

The original model is given in eq. (4.60). The continuous (non-box) version of that is:

$$\frac{\partial}{\partial t}T = \frac{1}{\rho c_p H} \left(\frac{S}{4}(1 - \alpha) - \epsilon\sigma T^4 \right) \quad (6.77)$$

where $T = T(\theta, t)$ is the temperature at latitude θ . The two terms on the RHS represent heating by incoming shortwave radiation, corrected for surface albedo, and cooling by outgoing longwave radiation. We now add a diffusive term:

$$\frac{\partial}{\partial t}T = \frac{1}{\rho c_p H} \left(\frac{S}{4}(1 - \alpha) - \epsilon\sigma T^4 \right) + \frac{D}{R_e^2} \frac{\partial^2}{\partial \theta^2} T \quad (6.78)$$

Note that:

$$\frac{\partial}{\partial y} = \frac{1}{R_e} \frac{\partial}{\partial \theta}$$

if R_e is the radius of the earth. The diffusive term will allow heat to spread across latitudes.

Thus there are two damping terms in the equation, one due to longwave cooling and one from diffusion. The relative importance of each can be assessed by looking at the associated time scales. The diffusive time (eq. 6.52), using the earth's radius as the length scale, is:

$$T_D = \frac{R_e^2}{D} \quad (6.79)$$

This can have a range of values. If $D = 10^6 \text{ m}^2/\text{sec}$, then $T_D \approx 4 \times 10^7 \text{ sec}$, which is roughly 16 months. If $D = 10^7 \text{ m}^2/\text{sec}$, the diffusive time is 10 times less, or about a month and a half.

The other time scale is that associated with the radiation, T_{rad} . We calculated this in sec. (4.4.2), and found a value of about 4.5 months. Thus the diffusive and radiative terms will be of equal size if the diffusivity is $D \approx 3 \times 10^6 \text{ m}^2/\text{sec}$.

Equation (6.78) is difficult to solve, even for the steady state. This is due to the nonlinear radiative term. So we'll turn immediately to a numerical solution. We'll use the FTCS formulation, since it's the simplest. The iFTCS form is less straightforward, also because of the nonlinear radiative term. The FTCS version of the equation is:

$$T_j^{n+1} = T_j^n + s(T_{j+1}^n - 2T_j^n + T_{j-1}^n) + \frac{S_j(1 - \alpha_j)dt}{4\rho c_p H} - \frac{\epsilon\sigma dt}{\rho c_p H}(T_j^n)^4 \quad (6.80)$$

or:

$$T_j^{n+1} = sT_{j-1}^n + (1 - 2s)T_j^n + sT_{j+1}^n + \frac{S_j(1 - \alpha_j)dt}{4\rho c_p H} - \frac{\epsilon\sigma dt}{\rho c_p H}(T_j^n)^4 \quad (6.81)$$

We have the radiation, S_j , and the albedo, α_j , from before. The only difference is the diffusion terms on the RHS.

Because of these, we need boundary conditions, which occur at the poles. We'll use zero flux (Neumann) conditions there, because no heat can escape into space. If we evaluate (6.81) at the first latitude, which we set as the one next to the South Pole, we have:

$$T_1^{n+1} = sT_0^n(1 - 2s)T_1^n + sT_2^n + \frac{S_1(1 - \alpha_1)dt}{4\rho c_p H} - \frac{\epsilon\sigma dt}{\rho c_p H}(T_1^n)^4 \quad (6.82)$$

or, setting $T_1 = T_0$:

$$T_1^{n+1} = (1 - s)T_1^n + sT_2^n + \frac{S_1(1 - \alpha_1)dt}{4\rho c_p H} - \frac{\epsilon\sigma dt}{\rho c_p H}(T_1^n)^4 \quad (6.83)$$

since $T_0 = T_1$. Likewise, at the last grid point (next to the North Pole):

$$T_N^{n+1} = sT_{N-1}^n + (1 - s)T_N^n + \frac{S_1(1 - \alpha_1)dt}{4\rho c_p H} - \frac{\epsilon\sigma dt}{\rho c_p H}(T_N^n)^4 \quad (6.84)$$

Then we code this with matrices, so that:

$$\mathbf{T}^{n+1} = A\mathbf{T}^n + \mathbf{F} \quad (6.85)$$

where:

$$\mathbf{A} = \begin{bmatrix} 1-s & s & 0 & 0 \\ s & 1-2s & s & 0 \\ 0 & s & 1-2s & s \\ 0 & 0 & s & 1-s \end{bmatrix} \quad (6.86)$$

and where \mathbf{F} are the two forcing terms.

We set the latitude spacing at 1 degree, and the time step so that $s = 0.4$. We then integrate out to 10 times the radiative time scale, given above. The radiative forcing was set to:

$$\frac{S}{4} = 350 * \cos(\theta) + 150$$

which is somewhat larger than is realistic, but necessary to obtain reasonable temperature profiles.

The solutions with $D = 10^6$ m²/sec and $D = 5 \times 10^6$ m²/sec are shown in Fig. (6.20). With $D = 10^6$ m²/sec, the profile is more spread out than in the zero diffusion case (shown in dots), but not a lot. So this level of diffusion is too little to move the profile towards the observations (shown by the dashed curve). But the solution with $D = 5 \times 10^6$ m²/sec is definitely better. The temperature is still too warm at the equator, but the curve is much closer to the observations.

Two more cases are shown in Fig. (6.21), $D = 10^7$ m²/sec and $D = 5 \times 10^7$ m²/sec. The first (left panel) is now fairly close to the observations, with the temperature more accurate at the equator. However, the temperatures (as with $D = 5 \times 10^6$ m²/sec) are too warm at the poles.

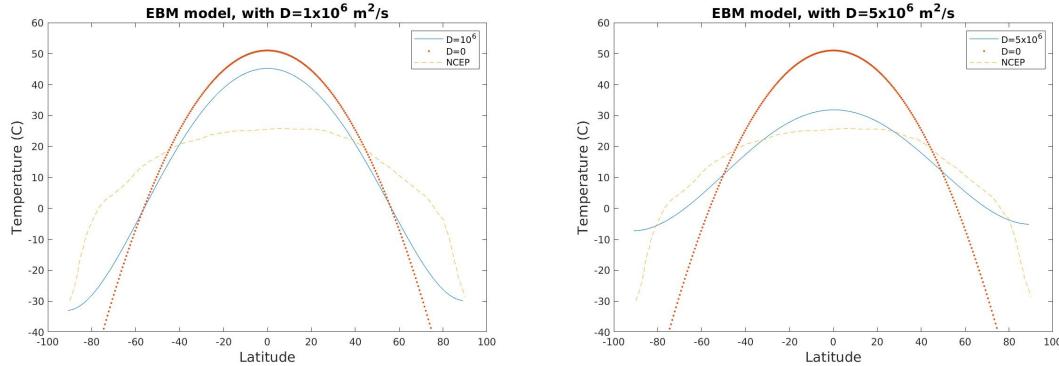


Figure 6.20: The diffusive climate model with $D = 10^6 \text{ m}^2/\text{sec}$ (left panel) and $D = 5 \times 10^6 \text{ m}^2/\text{sec}$ (right panel). The zero diffusion solution and the NCEP reanalysis profiles are shown for comparison.

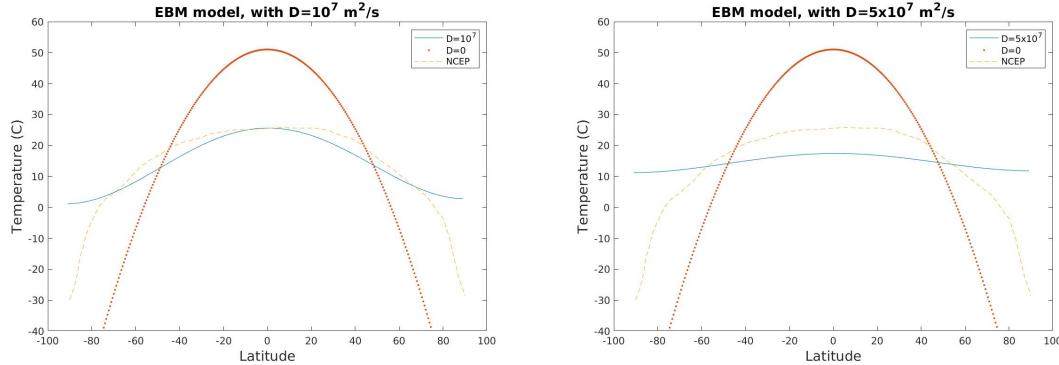


Figure 6.21: The diffusive climate model with $D = 10^7 \text{ m}^2/\text{sec}$ (left panel) and $D = 5 \times 10^7 \text{ m}^2/\text{sec}$ (right panel).

The temperature curve with $D = 5 \times 10^6 \text{ m}^2/\text{sec}$ (right panel) however is much too flat. The atmosphere in this simulation is nearly *isothermal*, with little difference between the poles and the equator. Thus the “storms” are too energetic, mixing heat too efficiently to the poles. Such is the case on Venus, where the atmosphere is also nearly isothermal (and hot enough at the surface to melt lead!).

Thus the most realistic solution obtains with $D \approx 10^7 \text{ m}^2/\text{sec}$. Then the diffusive and radiative time scales are of the same order of magnitude. We have few direct observations of the lateral diffusivity in the atmo-

sphere, but one study (using weather balloons) estimated $D \approx 3 \times 10^7$ m²/sec (Theibaux, 1976). So our estimate is of the correct order of magnitude.

Nevertheless, the model exhibits one aspect which is unrealistic: while the modelled temperature gradient is zero at the poles (due to the no flux boundary conditions), it definitely isn't in the observations. It is unlikely there is a heat flux to space, so something else is going on. The forcing terms are weaker at the poles, but not small enough to account for the difference. The problem more likely is that the storm activity is weak at the highest latitudes, meaning the diffusivity is also weak there.

As we have taken the diffusivity to be constant, we cannot capture this effect. We must make the diffusivity vary with latitude. This introduces an additional complication, so we'll wait to look at this in Chapter (7).

6.4 Summary

Many problems can be modelled as diffusive processes, from the spreading of pollution from a point source or from a steady boundary source, to modelling the atmospheric temperature. Diffusion comes about because many fields (temperature, particles, tracer) spread down the mean gradients. Diffusion thus acts to reduce gradients, driving the distribution to linear or flat profiles.

For the analytical solutions, we used “self-similar” solutions, which retain their shape when plotted as a function of the variable $\eta = x/\sqrt{Dt}$. Thus while the distribution spreads out in x as time progresses, it retains its form when plotted against η . This is a useful technique for reducing PDEs to

ODEs and has wide applicability. It can also be used in solving nonlinear PDEs.³

For the numerical solutions, we tested the FTCS and iFTCS schemes. The FTCS scheme yields fairly accurate solutions as long as $s = Ddt/dx^2 < 0.5$. For larger values, the scheme was numerically unstable. The iFTCS scheme is numerically stable for all s , and thus is potentially faster, because you can use larger time steps. However, it is also less accurate for larger values of s .

Lastly, we found that *boundary conditions* are very important for the solutions, and the solutions depend on the choices made. Sometimes the choice is obvious, for example when there are solid boundaries or there is no place for the diffused quantity to go (as at the poles in the climate model). But other times, for instance when the modelled domain is smaller than the actual region, the boundary conditions can produce unrealistic features. So some care is required when dealing with these.

³An alternative approach, which works only for linear problems but can also capture the initial behavior, before the distribution is self-similar, uses the so-called “Laplace transform”.

Chapter 7

Advection-Diffusion



Figure 7.1: A spill of toxic waste in the Colorado river. From the New York Times.

We considered advection and diffusion separately in the previous two chapters. But most often, both processes occur simultaneously. This can introduce interesting effects.

Some examples of advection and diffusion are:

- When Eyjafjallajökull erupted in 2010, the ash was advected toward Europe by the large scale winds. This caused the ash to stretch out into filaments, with large concentrations. But the filaments at the

same time diffused laterally, which increased their width.

- In a famous oceanographic experiment in the North Atlantic (the NATRE experiment), dye was released below the surface. The dye spread out laterally, due to large scale advection, and diffused *vertically*. The resulting values of the vertical diffusivity had a huge impact on oceanography.
- Mantle plumes occur when hot magma rises toward the surface, causing the tectonic plates to shift. This is inherently an advective-diffusive process, since the Reynolds number in the mantle is relatively low.
- A chemical spilled in a river also experiences advection-diffusion as it moves downstream. At some point, the plume can appear nearly uniform across the river (and in the worst cases, changing even the rivers color).

As usual, we'll examine a simple analytical solution, and then move on to numerical solutions, testing them against the analytical solution. Then we consider the case of pure diffusion with a spatially-variable diffusivity. Surprisingly, this also acts as an advective-diffusive process. And lastly, we return to the energy-balance climate model, allowing for variable diffusion.

7.1 Advection-diffusion equation

First, we'll derive the equation for advection. For this, we'll take the general case where both the advecting velocity and the diffusivity vary in space. Consider the volume shown in Fig. (5.2). The total (advective

+ diffusive) flux into the left side of the box is:

$$F(x) = u(x)C(x) - D(x)\frac{\partial}{\partial x}C(x)$$

. Recall the diffusive flux is negative because it is down the gradient. Similarly, the flux on the right side of the box is:

$$F(x + dx) = u(x + dx)C(x + dx) - D(x + dx)\frac{\partial}{\partial x}C(x + dx)$$

. We expand this using a Taylor series, assuming that dx is small:

$$\begin{aligned} F(x+dx) \approx & u(x)C(x) - D(x)\frac{\partial}{\partial x}C(x) + \frac{\partial}{\partial x}(u(x)C(x))dx - \frac{\partial}{\partial x}(D\frac{\partial}{\partial x}C)dx \\ & . \end{aligned}$$

Thus the volume-integrated concentration evolves as:

$$\begin{aligned} \frac{d}{dt} \iiint C dV \approx & dx \frac{\partial}{\partial t} CA = \\ F_l A - F_r A = & -\frac{\partial}{\partial x}(u(x)C(x))Adx + \frac{\partial}{\partial x}(D\frac{\partial}{\partial x}C)Adx \quad (7.1) \end{aligned}$$

In the first line, we assumed the volume was small, so that we could take the concentration is approximately constant inside. In the second line, we multiply the fluxes by the side area, A , to obtain the total tracer transport.

Canceling the A 's and dx 's, we get:

$$\frac{\partial}{\partial t}C + \frac{\partial}{\partial x}(uC) = \frac{\partial}{\partial x}(D\frac{\partial}{\partial x}C) \quad (7.2)$$

This is the 1D advection-diffusion equation with a spatially-variable velocity and diffusivity. Including the other sides of the control volume, we obtain:

$$\frac{\partial}{\partial t} C + \nabla \cdot (\vec{u}C) = \nabla \cdot (D \nabla C) \quad (7.3)$$

the 3D advective-diffusive equation.

Equations like this are widely used in the geosciences. And again, C doesn't need to be a passive tracer. As noted previously, the same equation (eq. 5.6) applies for fluid density, ρ , an active tracer.

In this chapter though, we'll focus on the 1D version of the equation (7.2). And we'll start by taking the velocity and diffusivity to be constant, so that:

$$\frac{\partial}{\partial t} C + u \frac{\partial}{\partial x} C = D \frac{\partial^2}{\partial x^2} C \quad (7.4)$$

In such processes, there is a “competition” between advection and diffusion. If advection dominates, the solutions will resemble those in Chapter (5), while if diffusion dominates, they will resemble those in Chapter (6).

To see which dominates, we scale the equation:

$$\begin{aligned} \frac{\partial}{\partial t} C + u \frac{\partial}{\partial x} C &= D \frac{\partial^2}{\partial x^2} C \\ \frac{C}{T} &\quad \frac{UC}{L} \quad \frac{DC}{L^2} \\ \frac{L^2}{DT} &\quad \frac{UL}{D} \quad 1 \end{aligned} \quad (7.5)$$

In the third line, we've divided through by the scale of the third term, leaving two non-dimensional parameters. The first is the ratio of the diffusive and actual time scales:

$$\frac{T_D}{T}$$

where again $T_D = L^2/D$.

The second parameter is the *Peclet number*, $Pe = UL/D$. This measures the relative importance of advection and diffusion. If $Pe \gg 1$, advection dominates, while if $Pe \ll 1$, diffusion does. If $Pe \approx 1$, both terms are equally important.

Note the Peclet number can also be seen as a ratio of time scales:

$$Pe = \frac{UL}{D} = \frac{L^2 U}{D L} = \frac{T_D}{T_a}$$

Here T_a is the *advection time scale*. This represents how long it takes tracer to be carried a distance L by the flow. If the diffusive time is much longer than the advective time, the Peclet number is large and advection dominates.

Note that the Lagrangian form (sec. 5.1.1) of the advection diffusion equation is given by:

$$\frac{dC}{dt} = \kappa \frac{\partial^2}{\partial x^2} C \quad (7.6)$$

Thus the concentration on a fluid parcel advected by the flow changes only by diffusion.

Equation (7.6) seems to be much simpler to solve than the corresponding Eulerian equation (7.3), because there is no advective term. However, equation (7.6) can be difficult to handle because the gradients on the RHS must be evaluated where the fluid parcel is. In cases when the velocity is variable or even unknown, this can be an enormous problem. So most often we model the Eulerian equation instead.

7.2 Analytical solution: point source

Now we'll consider a simple analytical solution. This is actually a combination of two solutions obtained earlier, in Chapters (5) and (6). The tracer

(a pollutant, for example) is spilled exactly at a point. It is then swept along by the flow, diffusing as it goes.

A (horrifying) example of this is shown in Fig. (7.1). In 2015, the US Environmental Protection Agency (EPA) attempted to drain a pond near the Gold King Mine near Silverton, Colorado. In so doing, they inadvertently spilled a large amount of toxic waste into the river, which subsequently spread downstream and turned the river orange. The site was declared a disaster, and they obtained money for the government for the subsequent clean-up.

As before, we'll assume the initial tracer distribution is a delta function:

$$C = C_0 \delta(x) \quad (7.7)$$

Under pure diffusion, the solution is given by eq. (6.34):

$$C = \frac{C_0}{2\sqrt{\pi Dt}} \exp\left(-\frac{x^2}{4Dt}\right) \quad (7.8)$$

Recall that this is a self-similar solution, in the form of a spreading Gaussian function.

With a constant velocity, the solution retains its shape and is carried downstream such that:

$$C = F(x - ut) \quad (7.9)$$

To obtain the advective-diffusive solution, we simply combine these two, replacing the x in the diffusive solution by $x - ut$:

$$C = \frac{C_0}{2\sqrt{\pi Dt}} \exp\left(-\frac{(x - ut)^2}{4Dt}\right) \quad (7.10)$$

Two examples are plotted in Fig. (7.2). The evolution depends on the Peclet number, and this in turn depends on the length scale, L , which is

changing in time. But we see the length scale is of order one, so we can set $L = 1$, yielding $Pe = U/D$. In the left panel is the case with $Pe = 10$, and in the right, $Pe = 2$. In both cases, the concentration spreads as it moves to the right. But in the right panel, the spreading is much more pronounced. From this, we can infer what would happen with much larger Pe (weak spreading) and smaller (diffusively dominated).

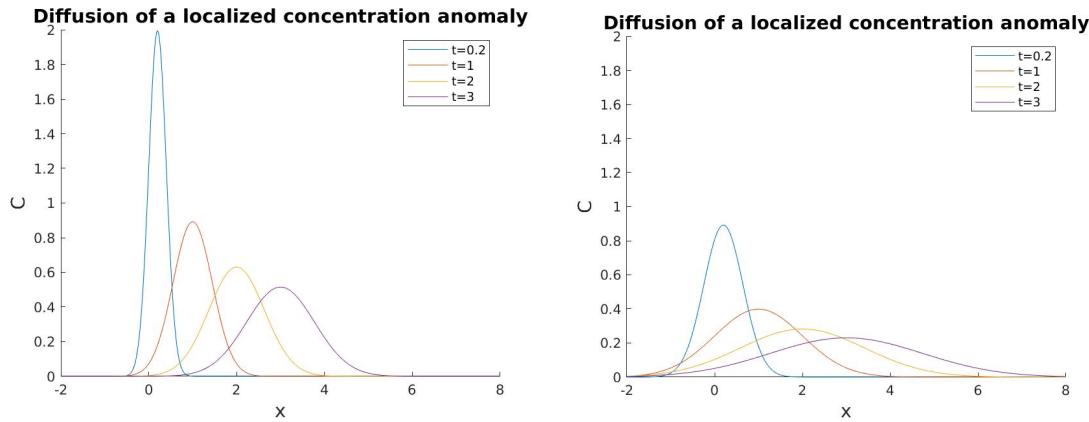


Figure 7.2: Analytical solutions to the advective-diffusive equation with a point source initial condition. In the left panel, $Pe = 10$ and in the right, $Pe = 2$.

Recall that diffusion acts most effectively when there are strong gradients. This occurs at the earliest times with a delta function initial condition. Increasing the diffusivity by a factor of 5 (as done here), has a large impact on the initial development.

As discussed with advection, the concentration can also be thought of as a *probability* of observing tracer. Thus a similar figure would describe the likely position of a person falling into the river at the initial location ($x = 0$). Diffusion spreads the probability distribution in time over ever larger areas, making it more difficult to locate the person as he/she floats downstream.

7.3 Numerical solutions

Now we turn to numerical solutions to the advection-diffusion equation. We'll keep the velocity and diffusivity constant at first, in line with the preceding analytical solution. Then we'll relax the second constraint and consider a spatially-variable diffusivity.

As usual, we start with the FTCS scheme, the simplest of all. The finite difference equation is:

$$\frac{C_j^{n+1} - C_j^n}{dt} + u \frac{C_{j+1}^n - C_{j-1}^n}{2dx} = D \frac{C_{j+1}^n - 2C_j^n + C_{j-1}^n}{dx} \quad (7.11)$$

or:

$$C_j^{n+1} = \left(\frac{C_0}{2} + s\right)C_{j-1}^n + (1 - 2s)C_j^n + \left(-\frac{C_0}{2} + s\right)C_{j+1}^n \quad (7.12)$$

where again:

$$Co = \frac{udt}{dx}, \quad s = \frac{Ddt}{dx^2} \quad (7.13)$$

are the Courant and diffusion numbers, the two important non-dimensional parameters in the problem.

We'll use Dirichlet boundary conditions, $C(0) = 0$ and $C(L) = 0$. Then the first equation is:

$$C_1^{n+1} = (1 - 2s)C_1^n + \left(-\frac{C_0}{2} + s\right)C_2^n \quad (7.14)$$

and the last equation is:

$$C_N^{n+1} = \left(\frac{C_0}{2} + s\right)C_{N-1}^n + (1 - 2s)C_N^n \quad (7.15)$$

So the problem can be written as a matrix equation:

$$\mathbf{C}^{n+1} = A\mathbf{C}^n \quad (7.16)$$

with:

$$\mathbf{A} = \begin{bmatrix} 1 - 2s & -\frac{Co}{2} + s & 0 & 0 \\ \frac{Co}{2} + s & 1 - 2s & -\frac{Co}{2} + s & 0 \\ 0 & \frac{Co}{2} + s & 1 - 2s & -\frac{Co}{2} + s \\ 0 & 0 & \frac{Co}{2} + s & 1 - 2s \end{bmatrix} \quad (7.17)$$

for the case with four interior grid points.

The case with $Pe = 2$ is shown in the left panel of Fig. (7.3) and that with $Pe = 10$ in the right panel. The numerical solutions are shown by the dots, and the analytical solution curves from Fig. (7.2) are superimposed. The numerical solutions are initialized again with an approximation of the delta function, with $C = 1/dx$ for $0 < x < dx$ and zero otherwise. We set $dx = 10/200 = 0.5$.

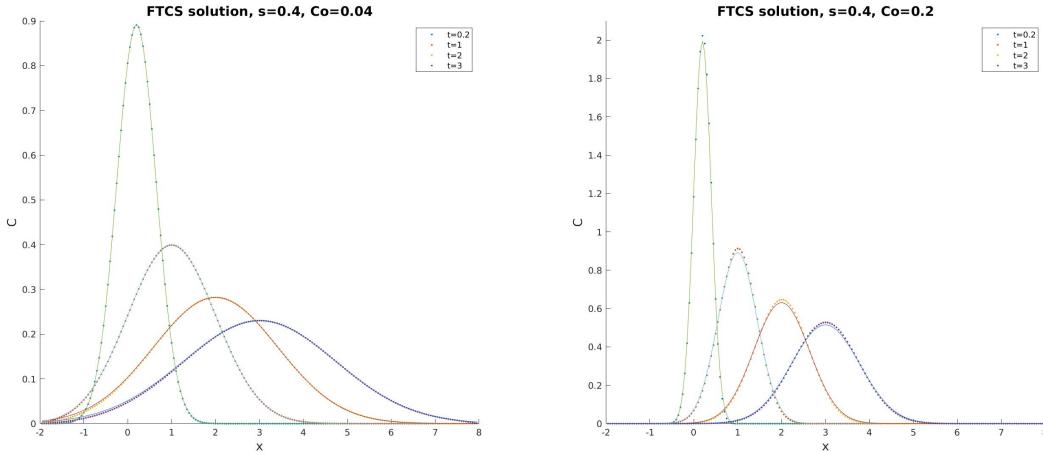


Figure 7.3: FTCS solutions to the advective-diffusive equation with a point source initial condition. In the left panel, $Pe = 2$ and in the right, $Pe = 10$.

The FTCS solutions are remarkably similar to the analytical solutions, particularly with $Pe = 2$. There are deviations with $Pe = 10$, near the tops of the distributions. This is because the narrower (less diffusive) solution is harder to simulate. We'll see this again with propagating fronts. Increasing the resolution to $dx = 0.25$, essentially fixes this; then the agreement

is excellent.

Then we can try the iFTCS scheme. Now the finite difference equation is:

$$\frac{C_j^{n+1} - C_j^n}{dt} + u \frac{C_{j+1}^{n+1} - C_{j-1}^{n+1}}{2dx} = D \frac{C_{j+1}^{n+1} - 2C_j^{n+1} + C_{j-1}^{n+1}}{dx} \quad (7.18)$$

or:

$$\left(-\frac{C_0}{2} - s\right)C_{j-1}^{n+1} + (1 + 2s)C_j^{n+1} + \left(\frac{C_0}{2} - s\right)C_{j+1}^{n+1} = C_j^n \quad (7.19)$$

With zero boundary conditions, the first equation is:

$$(1 + 2s)C_1^{n+1} + \left(\frac{C_0}{2} - s\right)C_2^{n+1} = C_1^n \quad (7.20)$$

and the last is:

$$\left(-\frac{C_0}{2} - s\right)C_{N-1}^{n+1} + (1 + 2s)C_N^{n+1} = C_N^n \quad (7.21)$$

So the matrix equation is:

$$\mathbf{A}C^{n+1} = C^n \quad (7.22)$$

with:

$$\mathbf{A} = \begin{bmatrix} 1 + 2s & \frac{C_0}{2} - s & 0 & 0 \\ -\frac{C_0}{2} - s & 1 + 2s & \frac{C_0}{2} - s & 0 \\ 0 & -\frac{C_0}{2} - s & 1 + 2s & \frac{C_0}{2} - s \\ 0 & 0 & -\frac{C_0}{2} - s & 1 + 2s \end{bmatrix} \quad (7.23)$$

We then step forward with the inverse:

$$C^{n+1} = \mathbf{A}^{-1}C^n \quad (7.24)$$

The solutions (Fig. 7.4) are as good as the FTCS solutions. However, we have used $s = 1.0$; this would be unstable with the FTCS scheme. Thus the time step is 2.5 times larger, meaning the runs are 2.5 times faster.

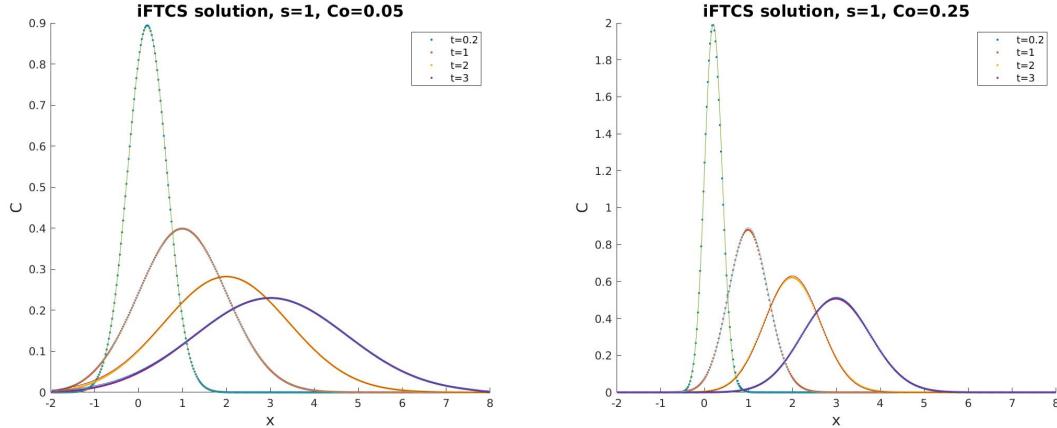


Figure 7.4: iFTCS solutions to the advective-diffusive equation with a point source initial condition, with $Pe = 2$ (left panel) and $Pe = 10$ (right panel). Note that $s = 1$ in both cases.

Larger values of s can be used as well, although the accuracy gradually worsens.

Lastly, consider the Crank-Nicholson scheme. Following the other two examples, we write the matrix equation as:

$$\mathbf{A}C^{n+1} = \mathbf{B}C^n \quad (7.25)$$

with:

$$\begin{aligned} \mathbf{A} &= \begin{bmatrix} 1+s & \frac{Co}{4} - \frac{s}{2} & 0 & 0 \\ -\frac{Co}{4} - \frac{s}{2} & 1+s & \frac{Co}{4} - \frac{s}{2} & 0 \\ 0 & -\frac{Co}{4} - \frac{s}{2} & 1+s & \frac{Co}{4} - \frac{s}{2} \\ 0 & 0 & -\frac{Co}{4} - \frac{s}{2} & 1+s \end{bmatrix} \\ \mathbf{B} &= \begin{bmatrix} 1-s & -\frac{Co}{4} + \frac{s}{2} & 0 & 0 \\ \frac{Co}{4} + \frac{s}{2} & 1-s & -\frac{Co}{4} + \frac{s}{2} & 0 \\ 0 & \frac{Co}{4} + \frac{s}{2} & 1-s & -\frac{Co}{4} + \frac{s}{2} \\ 0 & 0 & \frac{Co}{4} + \frac{s}{2} & 1-s \end{bmatrix} \end{aligned} \quad (7.26)$$

And we step forward thus:

$$C^{n+1} = \mathbf{A}^{-1} \mathbf{B} C^n \quad (7.27)$$

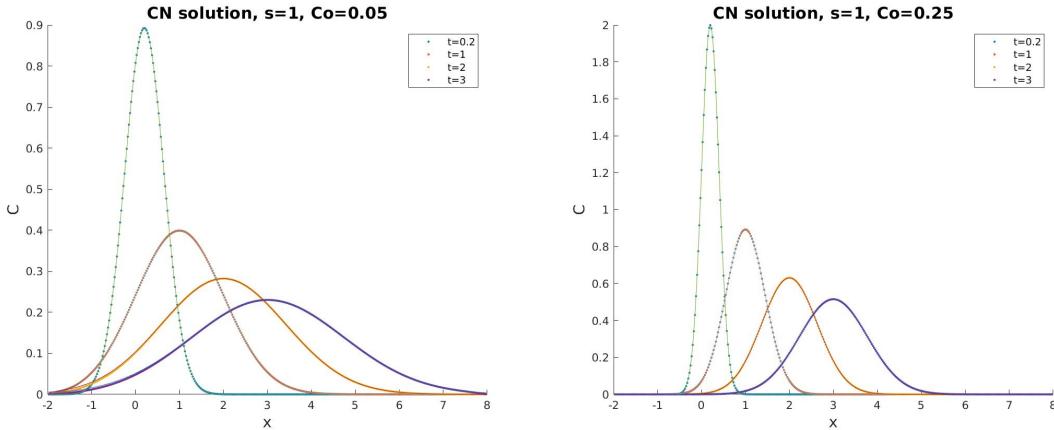


Figure 7.5: Crank-Nicholson solutions to the advective-diffusive equation with a point source initial condition, with $Pe = 2$ (left panel) and $Pe = 10$ (right panel). Again, $s = 1$ in both cases.

The results with the Crank-Nicholson scheme are the best of the three schemes (Fig. 7.5). The scheme is stable, so we can run it with $s = 1$, as with the iFTCS scheme, but the accuracy is also higher. With both values of the Peclet number, the agreement with the analytical solution is excellent. Thus in this (relatively simple) case with constant advection and diffusivity, the Crank-Nicholson scheme is the best choice, but either of the other schemes is reasonable as well.

7.3.1 Non-constant diffusivity

Up to now, we've only considered diffusion with a constant diffusivity. This is applicable for molecular motion, but when modelling geophysical flow, the diffusivity is rarely constant. For example, the lateral mixing happening in the western part of the North Atlantic, near the Gulf Stream, greatly exceeds that happening in the eastern Atlantic. Interestingly, variable diffusion acts as an advective-diffusive process, as we will see.

Consider the 1D advection-diffusion equation (7.2), without explicit ad-

vectation ($u = 0$):

$$\frac{\partial}{\partial t}C = \frac{\partial}{\partial x}(D\frac{\partial}{\partial x}C) \quad (7.28)$$

Expanding the RHS, we get:

$$\frac{\partial}{\partial t}C = \frac{\partial}{\partial x}D\frac{\partial}{\partial x}C + D\frac{\partial^2}{\partial x^2}C \quad (7.29)$$

or:

$$\frac{\partial}{\partial t}C - \frac{\partial}{\partial x}D\frac{\partial}{\partial x}C = D\frac{\partial^2}{\partial x^2}C \quad (7.30)$$

This is the same as the advection-diffusion equation, but with a velocity given by:

$$u = -\frac{\partial}{\partial x}D$$

So having a non-constant diffusivity implies *advection from regions of high diffusivity to low*.

For example, consider the simple case where the diffusivity is given by:

$$D = D_0x^2$$

The advecting velocity is then:

$$u = -2D_0x$$

This is positive for negative x and negative for positive x . Thus advection acts to move tracer towards $x = 0$.

Let's code this, using the Crank-Nicholson scheme. The matrices are the same as in (7.26), except that now the Courant number and the diffusion parameter vary in x :

$$Co(x) = -\frac{\partial}{\partial x}D\frac{dt}{dx}, \quad s(x) = D(x)\frac{dt}{dx^2}$$

Thus the matrices should be written:

$$\mathbf{A} = \begin{bmatrix} 1 + s_1 & \frac{Co_1}{4} - \frac{s_1}{2} & 0 & 0 \\ -\frac{Co_2}{4} - \frac{s_2}{2} & 1 + s_2 & \frac{Co_2}{4} - \frac{s_2}{2} & 0 \\ 0 & -\frac{Co_3}{4} - \frac{s_3}{2} & 1 + s_3 & \frac{Co_3}{4} - \frac{s_3}{2} \\ 0 & 0 & -\frac{Co_4}{4} - \frac{s_4}{2} & 1 + s_4 \end{bmatrix} \quad (7.31)$$

$$\mathbf{B} = \begin{bmatrix} 1 - s_1 & -\frac{Co_1}{4} + \frac{s_1}{2} & 0 & 0 \\ \frac{Co_2}{4} + \frac{s_2}{2} & 1 - s_2 & -\frac{Co_2}{4} + \frac{s_2}{2} & 0 \\ 0 & \frac{Co_3}{4} + \frac{s_3}{2} & 1 - s_3 & -\frac{Co_3}{4} + \frac{s_3}{2} \\ 0 & 0 & \frac{Co_4}{4} + \frac{s_4}{2} & 1 - s_4 \end{bmatrix} \quad (7.32)$$

We then proceed as before, inverting \mathbf{A} and stepping forward in time using eq. (7.27).

An example is shown in Fig. (7.6). We initialize with a delta function centered at $x = 3$, i.e.

$$C(x, 0) = C_0 \delta(x - 3)$$

Why is this preferable to initializing at $x = 0$?

Shown in the left panel is the full solution. In the right panel is the solution without the advection term, obtained by setting $u = 0$. The solutions are broadly similar, in that the tracer spreads more rapidly at larger values of x . But this is suppressed in the full solution, as the advection shifts the concentration back toward $x = 0$. Thus the concentration at, for example, $x = 9$ increases more slowly with the advective term than without it.

Hence a variable diffusivity has two effects. In regions of large diffusivity, the spreading is more rapid. But the non-uniform diffusivity also has an advective effect, which shifts tracer away from the highly diffusive regions.

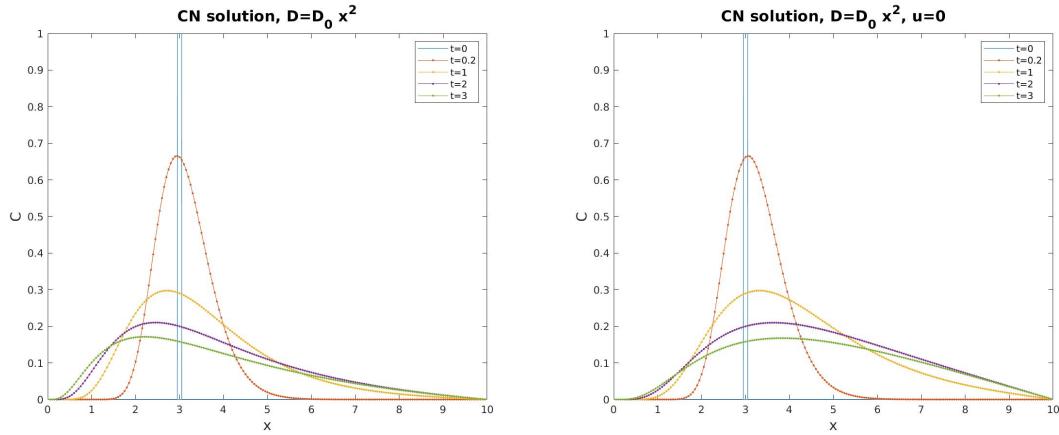


Figure 7.6: Crank-Nicholson solutions to the diffusive equation with a point source initial condition and a variable diffusivity, $D = 0.1x^2$. Shown in the left panel is the full solution, and in the right with the advective term set to zero.

7.3.2 The climate model with storm tracks

Now we return to the simple EBM of the climate. The model was introduced in sec. (4.4), and we included a diffusion term in sec. (6.3.4) to simulate heat transfer between different latitudes due to storms. We used a constant diffusivity when doing this, so that the “storms” were the same everywhere.

But this isn’t realistic. Storms in the atmosphere are enhanced in “storm tracks”, which occur at mid-latitudes. This is because storms derive from instability in the “Jet Streams”, which also are at mid-latitudes. In the tropics and at high latitudes, there is much less storm activity. It would be more realistic to have a diffusivity which reflects this.

Thus we’ll model this with non-constant diffusivity. As we have seen, this will also yield an advection term, causing heat to spread from the regions of high diffusivity to regions with low. The code follows closely from sec.

(6.3.4) and (7.3.1). First we define:

$$Co(j) = -\frac{\partial \kappa}{\partial y_j} \frac{dt}{dy}, \quad s(j) = \kappa_j \frac{dt}{dy^2}$$

With the FTCS scheme, we have:

$$\begin{aligned} T_j^{n+1} &= (s_j - \frac{Co_j}{2})T_{j-1}^n + (1 - 2s_j)T_j^n + (s_j + \frac{Co_j}{2})T_{j+1}^n + \\ &\quad \frac{S_j(1 - \alpha_j)dt}{4\rho c_p H} - \frac{\epsilon \sigma dt}{\rho c_p H} (T_j^n)^4 \end{aligned} \quad (7.33)$$

With the zero flux conditions at the poles, the first and last equations are:

$$\begin{aligned} T_1^{n+1} &= (1 - s_1 - \frac{Co_1}{2})T_1^n + (s_1 + \frac{Co_1}{2})T_2^n + \\ &\quad \frac{S_1(1 - \alpha_1)dt}{4\rho c_p H} - \frac{\epsilon \sigma dt}{\rho c_p H} (T_1^n)^4 \end{aligned} \quad (7.34)$$

and

$$\begin{aligned} T_N^{n+1} &= (s_N - \frac{Co_N}{2})T_{N-1}^n + (1 - s_N + \frac{Co_N}{2})T_N^n + \\ &\quad \frac{S_N(1 - \alpha_N)dt}{4\rho c_p H} - \frac{\epsilon \sigma dt}{\rho c_p H} (T_N^n)^4 \end{aligned} \quad (7.35)$$

The matrix set-up follows that in the previous section. We only need remember to use the spatially variable s and Co .

In the first attempt with the model, we use a bimodal (two peaked) diffusivity, as shown in the left panel of Fig. (7.7). The diffusivity is given by:

$$\kappa = 2 \times 10^7 [sech((y + L/2)/Re * 3) + sech((y - L/2)/Re * 3)] \text{ m}^2/\text{sec}$$

This yields a profile with peaks at 45S and 45N, roughly where the storm tracks lie.

The solution is shown in the right panel, in blue. This is certainly better than the non-diffusive solution (red dots), but is not necessarily closer to

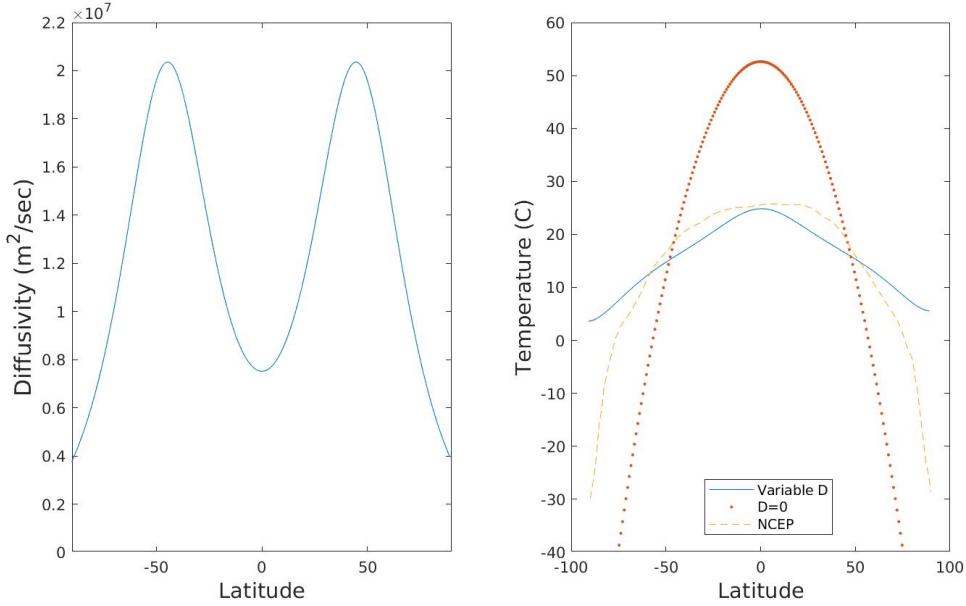


Figure 7.7: The simple climate model with a variable diffusivity. The diffusivity (left panel) peaks at 40S and 40N, roughly in accord with the storm tracks. The resulting temperature profile is plotted in blue in the right panel, along with the zero diffusion equilibrium solution (dots) and the NCEP reanalysis annual mean temperature (dashed).

the observations (dashed curve) than the solution with $\kappa = 10^7 \text{ m}^2/\text{sec}$ (left panel of Fig. 6.21). This solution is more “peaked” at low latitudes and is still much warmer at the poles.

After some trial-and-error experimenting with parameters, we arrive at the solution in Fig. (7.8). The diffusivity now is given by:

$$\kappa = 2 \times 10^7 [\cos(y/Re)]^{1.5} + 2 \times 10^5 \text{ m}^2/\text{sec}$$

This is similar to a simple cosine profile, but the exponent of 1.5 sharpens the distribution slightly. The diffusivity is now intensified near the equator, rather than in the storm tracks.

The solution (right panel) is now fairly close to the observations. The flattening seen in the temperature profile at the equator is more nearly captured than in the previous case, and the temperature decreases rapidly at higher

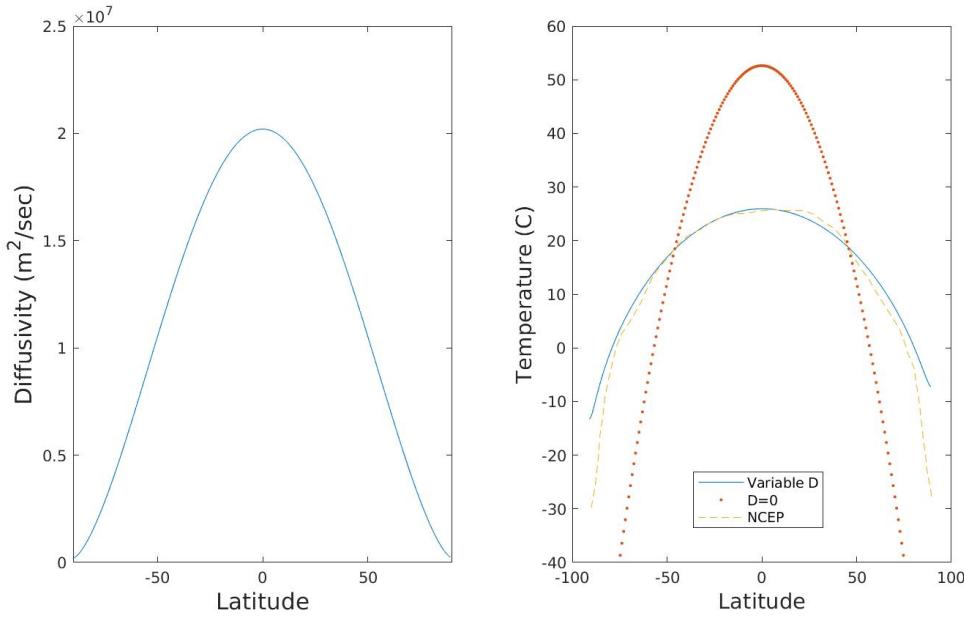


Figure 7.8: The EBM climate model with a variable diffusivity. The diffusivity (left panel) peaks at 45S and 45N, roughly in accord with the storm tracks. The resulting temperature profile is plotted in blue in the right panel, along with the zero diffusion equilibrium solution (dots) and the NCEP reanalysis annual mean temperature (dashed).

latitudes, as observed. The only difference really is the temperature at the poles, which is over-estimated in the model. This is probably because the code required having a small constant background value (2×10^5 m²/sec) to remain numerically stable. The fact that the observed temperature decreases more suggests the actual diffusivity (and hence storm activity) is very weak indeed at the poles.

Why would a solution without storm tracks yield better results than one with them? The answer is perhaps related to the seasons. The storm tracks intensify in winter and are weak in summer, in each hemisphere. Thus the diffusivity profile in effect shifts back and forth across the equator. When averaging over the entire year, the result may be an effective diffusivity which is greatest at the equator. That's what we would infer from our

model, in any case.

This example illustrates a couple of points. For one, the application with a non-constant diffusivity and the nonlinear radiation term is too complex to solve analytically. This is where the numerical solution really comes to the fore. Secondly, we can learn about mixing occurring in the atmosphere by adjusting the model parameters. The annual mean temperature profile is best captured using a diffusivity on the order of $10^7 \text{ m}^2/\text{sec}$ which is intensified near the equator. Such a conclusion could be checked in experiments, either with balloons or with a full dynamical model. And lastly, a model like this could be used to study simple changes due to global warming, but systematically altering parameters like the emissivity.

7.4 Summary

We have examined processes under which both advection and diffusion occur simultaneously. Advection results in transport of the tracer and diffusion causes the distribution to spread out. We illustrated this with an analytical solution for a delta function initial distribution, advected by a constant flow with a constant diffusivity.

For the numerical solutions, we tested the FTCS, iFTCS and Crank-Nicholson schemes. As with pure diffusion, the FTCS scheme was often sufficient, though the CN scheme was better when sharp gradients exist. We also showed that having pure diffusion with a *non-constant* diffusivity yields advection as well, from regions of high mixing to regions with low. And we found that we could approximately capture the annual mean temperature in the atmosphere using such a model, with the radiation terms included.

Chapter 8

Momentum Advection

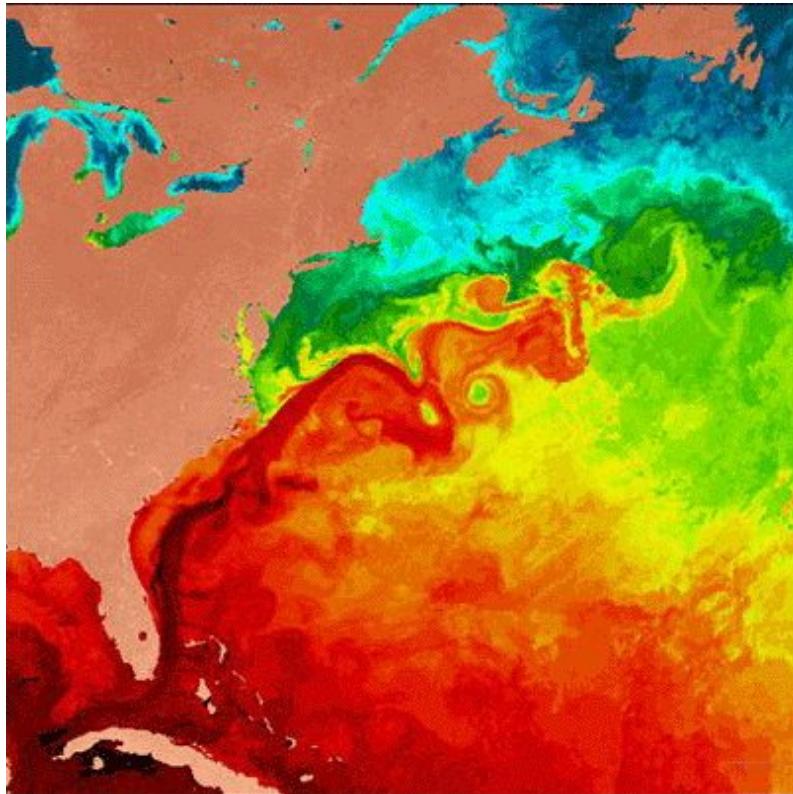


Figure 8.1: Sea surface temperature (SST) from satellite-derived, infrared image of the Gulf Stream. The current is seen as a boundary between the warm waters of the Sargasso Sea and the colder waters of the Gulf of Maine to the north. Courtesy of NASA.

In the last three chapters, we examined advection and diffusion of a passive tracer. These have many important applications, such as for pollution in the

atmosphere or in groundwater. But as noted, it is also possible to advect fluid density, which is an *active* tracer. It is active because moving the density can alter the flow.

Another example is of *momentum*. Momentum is obviously an active tracer, and its redistribution will change the flow. Momentum advection exhibits a rich diversity of phenomena, from breaking waves to fully-developed turbulence. And since the advection of momentum by the fluid velocity involves the product of two dependent variables, the problem is nonlinear. As such, few analytical solutions exist and numerical solutions are generally required. But even numerical modelling can be challenging.

A geophysical example of nonlinear advection is shown in Fig. (8.1). This is the Gulf Stream, separating from the North American coast, as seen in a satellite infrared image. The red colors correspond to warm water and the blue and green to colder waters. The Gulf Stream is a distinct boundary between the warm and cold waters. Moreover, it is meandering and pinching off large “eddies”, essentially oceanic storms. Both temperature and momentum are being advected here, and both are active tracers. Thus modelling the Gulf Stream is a demanding task, requiring high resolution and advanced numerics.

8.1 The momentum equation

Consider advection and diffusion of momentum with a fluid volume, as shown in Fig. (5.2). Momentum is normally defined as a mass times a velocity, but for fluids it is more convenient to work with mass per unit volume, or density. We’ll focus first on the momentum in the x -direction,

i.e. ρu .

Again, we can consider how the momentum changes in a small volume.

The flux on the left side of the box is:

$$F_l = u(x)[\rho(x)u(x)] - \nu \frac{\partial \rho u}{\partial x}(x)$$

The first term is the advective flux and the second is the diffusive flux.¹

The coefficient, ν , is the *viscosity*. The flux on the right side of the box is:

$$F_r = u(x + dx)[\rho(x + dx)u(x + dx)] - \nu \frac{\partial \rho u}{\partial x}(x + dx)$$

Expanding in a Taylor series, we get:

$$F_r = \rho u^2 - \nu \frac{\partial \rho u}{\partial x} + \frac{\partial}{\partial x}[\rho u^2 - \nu \frac{\partial \rho u}{\partial x}]dx + O|dx^2|$$

with all terms evaluated at x , i.e. at the LHS of the box. Using these fluxes, the conservation equation for the x-momentum is:

$$Adx \frac{\partial}{\partial t} \rho u = F_l A - F_r A = - \frac{\partial}{\partial x}(\rho u^2 - \nu \frac{\partial \rho u}{\partial x}) Adx \quad (8.1)$$

Cancelling constants and rearranging, we get:

$$\frac{\partial}{\partial t} \rho u + \frac{\partial}{\partial x}(\rho u^2) = \nu \frac{\partial^2}{\partial x^2}(\rho u) \quad (8.2)$$

Including the fluxes through the other sides of the box yields the full 3D equation for the x-momentum:

$$\frac{\partial}{\partial t} \rho u + \nabla \cdot (\rho u^2) = \nu \nabla^2(\rho u) \quad (8.3)$$

There are similar equations for the y- and z-momenta as well. We can write all three at once using a vector version of the equation:

¹Momentum diffusion, which is the result of viscous forces, is normally introduced as a *stress* (force) acting on the fluid volume, rather than an actual momentum flux. But the end result is the same as the derivation here.

$$\frac{\partial}{\partial t} \rho \vec{u} + \nabla \cdot (\rho \vec{u} \otimes \vec{u}) = \nu \nabla^2 (\rho \vec{u}) \quad (8.4)$$

This is the 3D momentum equation. Recall that we use the tensor product, \otimes , between the vector velocity and itself (see below).

We obtain a simpler version of the equation if we combine it with the continuity equation (for the density, ρ), given in eq. (5.6). We modify that slightly, by adding a diffusive term on the RHS, with the same diffusivity, ν .

$$\frac{\partial}{\partial t} \rho + \nabla \cdot (\vec{u} \rho) = \nu \nabla^2 \rho \quad (8.5)$$

Then we re-write equation (8.4), using the chain rule:

$$\vec{u} \left[\frac{\partial}{\partial t} \rho + \nabla \cdot (\vec{u} \rho) \right] + \rho \left(\frac{\partial}{\partial t} \vec{u} + \vec{u} \otimes \nabla \vec{u} \right) = \rho \nu \nabla^2 \vec{u} + \vec{u} [\nu \nabla^2 \rho] \quad (8.6)$$

Using the continuity equation, we can eliminate the terms in brackets. That leaves:

$$\frac{\partial}{\partial t} \vec{u} + \vec{u} \otimes \nabla \vec{u} = \nu \nabla^2 \vec{u} \quad (8.7)$$

Again, equation (8.7) is three equations, one for each component of the velocity. The tensor product in the advective term represents *nine* different terms:

$$\vec{u} \otimes \nabla \vec{u} = \begin{bmatrix} u \frac{\partial}{\partial x} u & v \frac{\partial}{\partial y} u & w \frac{\partial}{\partial z} u \\ u \frac{\partial}{\partial x} v & v \frac{\partial}{\partial y} v & w \frac{\partial}{\partial z} v \\ u \frac{\partial}{\partial x} w & v \frac{\partial}{\partial y} w & w \frac{\partial}{\partial z} w \end{bmatrix} \quad (8.8)$$

Thus if we take the x -component of the velocity, we have:

$$\frac{\partial}{\partial t} u + u \frac{\partial}{\partial x} u + v \frac{\partial}{\partial y} u + w \frac{\partial}{\partial z} u = \nu \nabla^2 u \quad (8.9)$$

Similar equations obtain for the other velocity components, v and w .

8.2 Burger's equation

The momentum equations are nonlinear PDEs and are difficult to solve. We'll focus instead on a simpler version, of flow in one dimension. Then the momentum equation is:

$$\frac{\partial}{\partial t}u + u\frac{\partial}{\partial x}u = \nu\frac{\partial^2}{\partial x^2}u \quad (8.10)$$

This is *Burger's equation*, a well-known equation in fluid mechanics. It exhibits some interesting effects, as we will see, and can be applied to a number of different flows.

Consider the *inviscid* version of the equation, i.e. the equation with $\nu = 0$:

$$\frac{\partial}{\partial t}u + u\frac{\partial}{\partial x}u = 0 \quad (8.11)$$

The equation looks like the standard advection equation (5.4) discussed in Chapter (5), except that that the advected quantity is now the velocity itself.

The effect that this has is illustrated in Fig. (8.2). This shows a velocity with an initially sinusoidal distribution:

$$u = \sin(x)$$

Advection causes the distribution to move, but the advecting velocity depends on the value of u itself. As such, the top of the sine wave moves to the right *faster than the rest of the wave*. The middle of the wave lags behind and the points where $u = 0$ don't move at all. Conversely, the trough near $x = 5$ moves *to the left*, because u is negative here. The net result

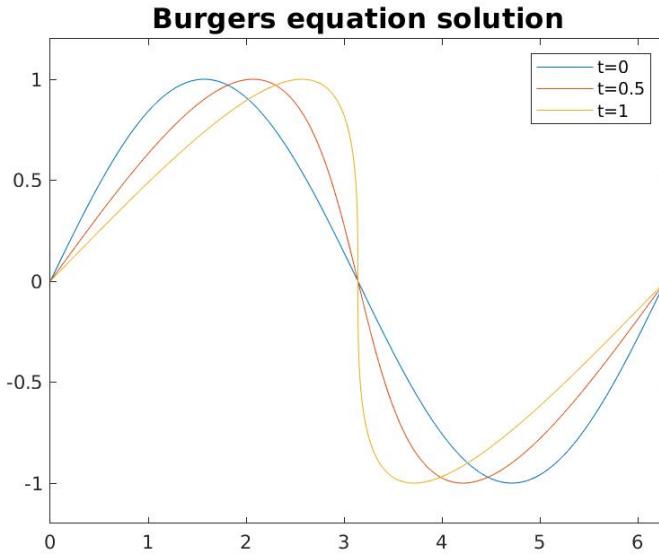


Figure 8.2: The evolution of a sinusoidal distribution of u under the inviscid Burger's equation.

is that the wave *steepens* near $x = \pi$. Indeed, the “wave” will eventually spill over as the top passes the bottom.

As you might guess, this is an approximate model of a breaking wave at a beach. In that case too, the wave moves at a speed which depends on its height, and thus eventually falls over or “breaks”. This is also a simple model of *frontal formation* in the atmosphere. A front is a region between warm and cold air, where the temperature changes very rapidly. Frontal formation also involves a process by which the gradients steepen. We refer to such fronts as *shocks*.

What happens though if we ignore advection? Then equation (8.10) becomes:

$$\frac{\partial}{\partial t} u = \nu \frac{\partial^2}{\partial x^2} u \quad (8.12)$$

This of course is just a 1D diffusion equation. So an initial distribution, for example a delta function, will spread out in time.

So what happens when we combine advection and diffusion? Advection acts to steepen gradients, while diffusion works to smooth them out. As before, the competition between advection and diffusion can be quantified by scaling the equation. Using typical scales, the equation scales as:

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

$$\frac{U}{T} \quad \frac{U^2}{L} \quad \frac{\nu U}{L^2}$$

$$\frac{L^2}{\nu T} \quad \frac{UL}{\nu} \quad 1 \quad (8.13)$$

The first term is the ratio of two time scales, that of the motion, T , and that of diffusion:

$$T_d = \frac{L^2}{\nu}$$

The second parameter measures the relative importance of advection and diffusion:

$$Re = \frac{UL}{\nu} \quad (8.14)$$

This is the *Reynolds number*. It is analogous to the Peclet number for a passive tracer (sec. 7.1). If the Reynolds number is large, advection dominates diffusion.

The Reynolds number is frequently used to categorize flows. Molasses, with a viscosity 5000-10,000 times that of water, is an example of a low Reynolds number fluid. Geophysical examples of low Reynolds number flows include **magma**, with a viscosity $10^4 - 10^8$ times that of water, and **glaciers**, with a viscosity 10^{15} times that of water (Fig. 8.3).

High Reynolds number flows on the other hand are usually strongly time dependent, and often *turbulent*. A river in early spring can be very turbu-



Figure 8.3: Low Reynolds number flows: a glacier (left) and magma (right). Courtesy of the University of Oslo and Wikipedia.

lent, with eddies and white water and $Re \approx 10^6$ (Fowler, 2011) (Fig. 8.4). A hurricane is another example, with violent winds and $Re \approx 10^{11}$.



Figure 8.4: High Reynolds number flows: a river (left) and a hurricane (right). Courtesy of Natalia Strandet and Freightwaves.com.

Even if the viscosity is very small though, in the high Re case, diffusion will eventually act to smooth out shocks. This is because diffusion also depends on the velocity gradient. Eventually the gradients become so large, under the effect of advection, that diffusion will act, limiting the further steepening.

8.3 Analytical solutions

Burger's equation is nonlinear, but because it is one-dimensional, it is possible to find solutions. We can do this using something known as the “Cole-

Hopf transformation”, after Hopf (1950) and Cole (1951). Say that we have a variable, w , that obeys the corresponding diffusion equation:

$$\frac{\partial}{\partial t}w = \nu \frac{\partial^2}{\partial x^2}w \quad (8.15)$$

Then the variable:

$$u = -2\nu \frac{w_x}{w} \quad (8.16)$$

is a solution to Burger’s equation. Thus if we have a solution to the diffusion equation, we can derive a solution to Burger’s equation. This doesn’t guarantee that the solution is actually relevant (or even very interesting), but it’s a solution.

For example, the solution to the diffusion equation with an initial delta function distribution is:

$$w = \frac{w_o}{2} \frac{1}{\sqrt{\nu\pi t}} \exp\left(-\frac{x^2}{4\nu t}\right) \quad (8.17)$$

You can show that this obeys:

$$\frac{\partial}{\partial x}w = -\frac{2x}{4\nu t}w \quad (8.18)$$

So we find:

$$u = -2\nu \frac{w_x}{w} = \frac{x}{t} \quad (8.19)$$

a rather simple solution.

We can check that this satisfies Burger’s equation. We have:

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

and:

$$\frac{\partial}{\partial x}u = \frac{1}{t}$$

and also:

$$\frac{\partial^2}{\partial x^2} u = 0$$

Thus:

$$\frac{\partial}{\partial t} u + u \frac{\partial}{\partial x} u = -\frac{x}{t^2} + \frac{x}{t^2} = 0$$

But is this solution useful? Not really. This is a linear solution, with a slope which decreases in time (left panel of Fig. 8.5). Worse, the slope is *infinite* at $t = 0$. It would be hard to imagine an initial condition which fit this description.

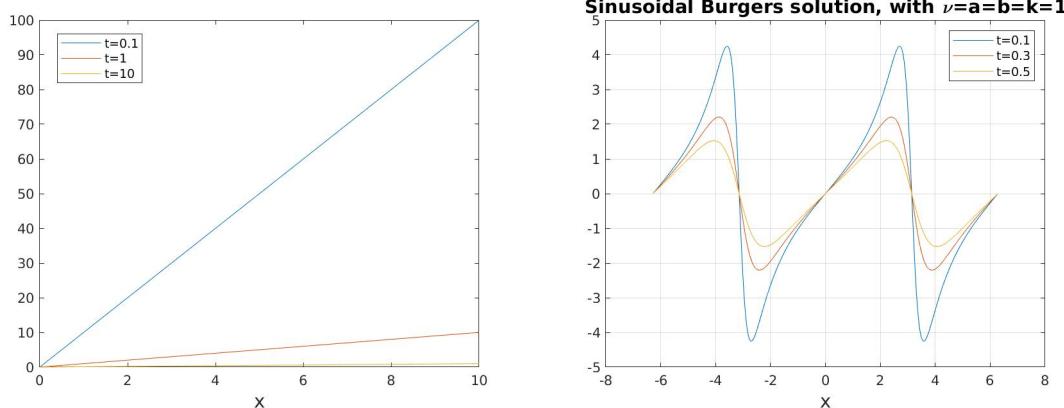


Figure 8.5: Two solutions to Burger's equation, $u = x/t$ (left panel) and one deriving from a sinusoidal solution to the diffusion equation (right panel).

Consider another example, with diffusion of an initial sinusoidal distribution:

$$w(x, 0) = a + b \cos(kx)$$

The solution (sec. 6.2.4) is:

$$w(x, t) = a + b e^{-\nu k^2 t} \cos(kx)$$

Using (8.16), we obtain:

$$u = -2\nu \frac{w_x}{w} = \frac{2\nu b k e^{-\nu k^2 t} \sin(kx)}{a + b e^{-\nu k^2 t} \cos(kx)} \quad (8.20)$$

This is plotted in the right panel of Fig. (8.5), using the parameters $\nu = a = b = k = 1$. The solution at the earliest time ($t = 0.1$) has sharp gradients, near $x = \pm 3$. But as time progresses, the gradients weaken and the amplitude of the disturbance decreases. This illustrates the effect of diffusion on strong gradients, which are present from the beginning with these parameters.

8.3.1 Solution for a point source

But these solutions don't help us unless the initial condition takes exactly the form represented by the solution. It turns out though that the Cole-Hopf transformation can be used to solve certain initial value problems. For example, with an initial delta-function condition:

$$u(x, 0) = A\delta(x)$$

The solution can be shown to be:

$$u = \sqrt{\frac{\nu}{\pi t}} \frac{(exp(A/(2\nu)) - 1)exp(-x^2/(4\nu t))}{1 + 1/2(exp(A/(2\nu)) - 1)erfc(x/sqrt(4\nu t))} \quad (8.21)$$

(Whitham, 1974). This is not a very pleasant-looking solution(!) But it behaves the right way. If the viscosity is large, then:

$$(exp(A/(2\nu)) - 1) \approx \frac{A}{2\nu} \ll 1$$

so that:

$$u \approx \sqrt{\frac{\nu}{\pi t}} \left(\frac{A}{2\nu}\right) exp\left(-\frac{x^2}{4\nu t}\right) = \frac{A}{\sqrt{4\pi\nu t}} exp\left(-\frac{x^2}{4\nu t}\right)$$

which is exactly the diffusive solution with a delta function initial condition.

The solution is plotted for different values of ν in Fig. (8.6). With $\nu = 0.5$ (upper left panel), the familiar Gaussian profile emerges by $t = 0.1$.

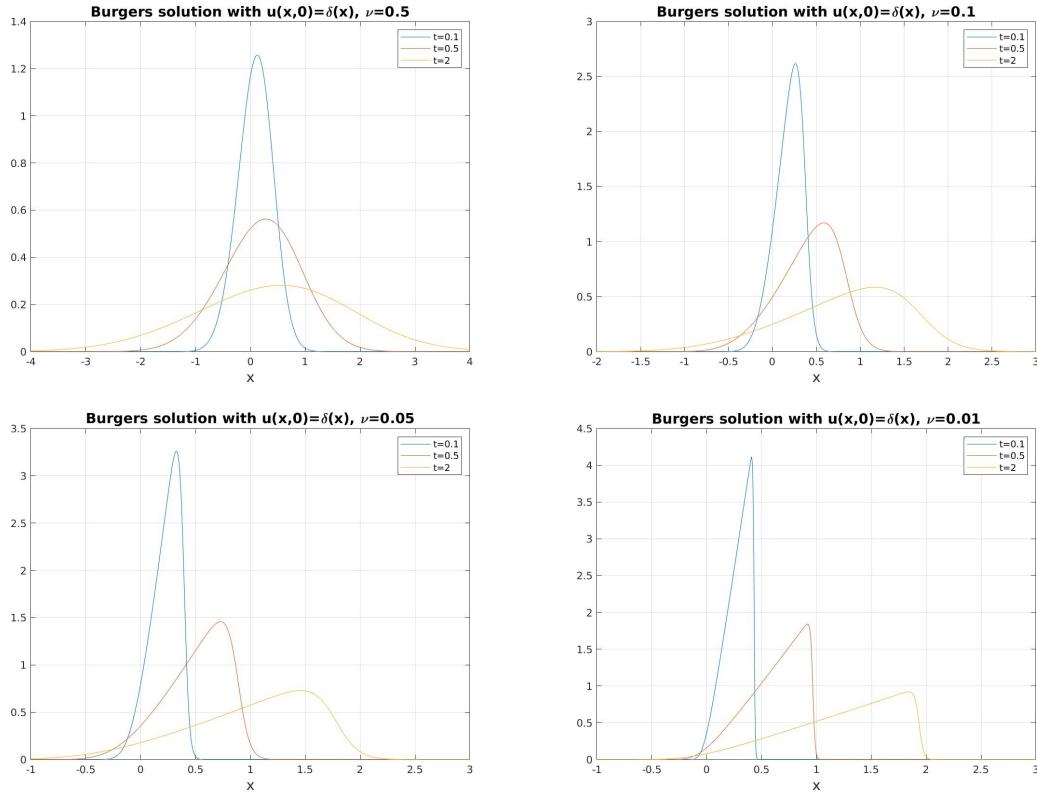


Figure 8.6: Solutions to Burger’s equation with a delta function initial condition, with increasing values of the viscosity, ν .

This advects to the right, but also spreads rapidly. Except for the slight deflection to the right, this looks like simple diffusion.

As we decrease the viscosity though, the spreading happens more slowly. Then, as the disturbance advects to the right, the leading edge becomes sharper and sharper. Indeed, it almost looks like a “sawtooth” disturbance with $\nu = 0.01$. But even with weak diffusion, the gradient at the leading edge is weakening in time. Thus diffusion, even with a small viscosity, eventually smooths strong gradients.

The solution is reminiscent of a “tidal bore”, a propagating wave found in some parts of the world, such as the Bay of Fundy in eastern Canada or off the coast of Normandy in France (Fig. 8.7). If the tide is strong enough,



Figure 8.7: Examples of tidal bores. Courtesy of the Truro News and Wikipedia.

and confined to a relative narrow bay or crossing relatively flat ground, the leading wave can be very steep. In the Bay of Fundy, people routinely surf the tidal wave as it makes its way through the bay. The bore can also propagate fast. They say that you need to ride a horse to outrun the tidal wave off Normandy. Each year, some unlucky tourists are swept out to sea.

8.4 Numerical solutions

Coding Burger's equation presents some challenges, again because the advection term is nonlinear. We'll use different approaches to handling this, depending on which scheme we use.

8.4.1 FTCS

As usual, we'll start with the FTCS scheme. But first, we re-write the equation slightly, thus:

$$\frac{\partial}{\partial t} u + \frac{\partial}{\partial x} E = \nu \frac{\partial^2}{\partial x^2} u \quad (8.22)$$

where E is the kinetic energy, defined:

$$E \equiv \frac{u^2}{2}$$

Then the FTCS difference equation is:

$$\frac{u_j^{n+1} - u_j^n}{dt} + \frac{E_{j+1}^n - E_{j-1}^n}{2dx} = \nu \frac{u_{j+1}^n - 2u_j^n + u_{j-1}^n}{dx^2} \quad (8.23)$$

or:

$$u_j^{n+1} = cE_{j-1}^n - cE_{j+1}^n + su_{j-1}^n + (1 - 2s)u_j^n + su_{j+1}^n \quad (8.24)$$

with:

$$c = \frac{dt}{2dx}, \quad s = \frac{\nu dt}{dx^2}$$

We write this as a matrix equation thus:

$$\mathbf{u}^{n+1} = \mathbf{A}\mathbf{E}^n + \mathbf{B}\mathbf{u}^n \quad (8.25)$$

and

$$\mathbf{E}^{n+1} = \frac{1}{2}(\mathbf{u}^{n+1})^t \mathbf{u}^{n+1} \quad (8.26)$$

where the t superscript indicates the transpose. Here:

$$\mathbf{A} = \begin{bmatrix} 0 & -c & 0 & 0 \\ c & 0 & -c & 0 \\ 0 & c & 0 & -c \\ 0 & 0 & c & 0 \end{bmatrix} \quad (8.27)$$

and:

$$\mathbf{B} = \begin{bmatrix} 1 - 2s & s & 0 & 0 \\ s & 1 - 2s & s & 0 \\ 0 & s & 1 - 2s & s \\ 0 & 0 & s & 1 - 2s \end{bmatrix} \quad (8.28)$$

for the case of four interior grid points. We execute this by first calculating \mathbf{E}^0 from \mathbf{u}^0 . Then we obtain \mathbf{u}^1 , from which we calculate \mathbf{E}^1 , and then so on, with \mathbf{u}^2 , etc.

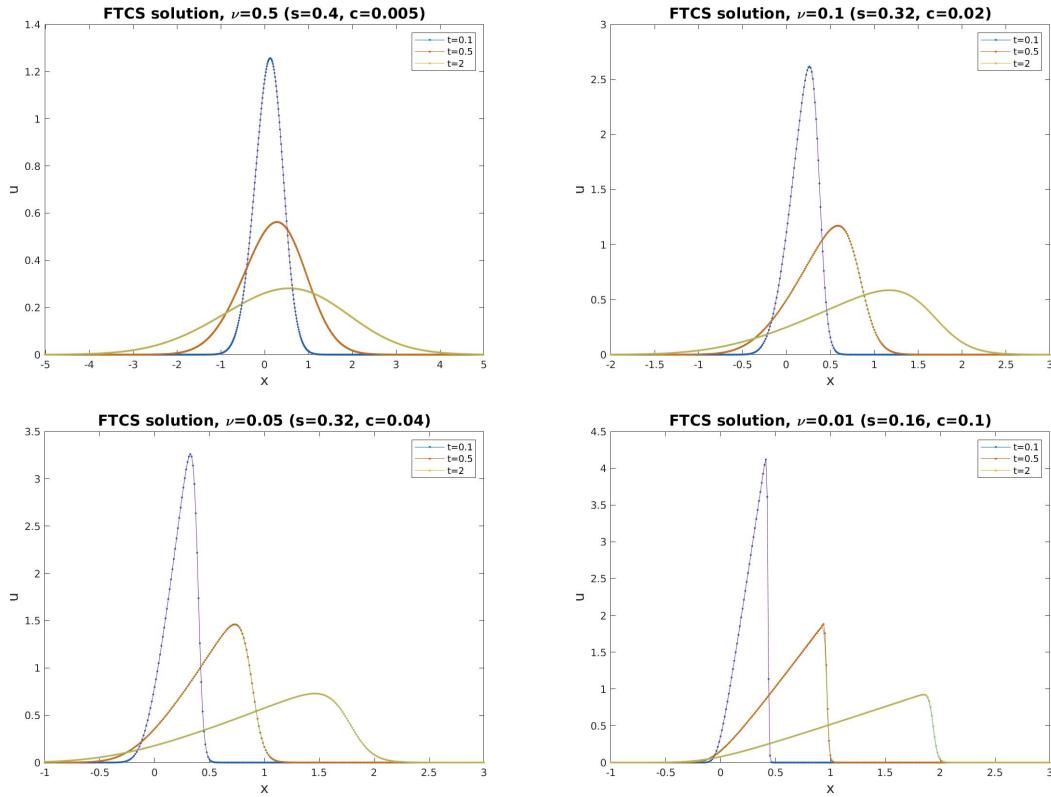


Figure 8.8: FTCS solutions (dots) to Burger's equation with a delta function initial condition, with increasing values of the viscosity, ν . The analytical solutions are indicated by the solid curves.

The solutions are shown in Fig. (8.8) for the same viscosities as in Fig. (8.6). Note that we use various time steps, so s varies. But it is less than 0.5 in all cases. The results are remarkably good, with the numerical solution tracking the analytical for all values of ν , even with $\nu = 0.01$.

Again we see that diffusion works in favor of the FTCS scheme. In practice, we might just stop here; if the simplest scheme functions as well as this, we don't need a more complicated scheme. But we'll consider two alternatives, just for practice.

8.4.2 iFTCS

The time step is limited with FTCS, as we require $s < 0.5$. How well does the iFTCS scheme do, as this is always stable?

The iFTCS scheme is a bit more complicated here, again because of the nonlinear advection term. Using the kinetic energy doesn't help, because of the need to invert the matrices. The problem can be seen if we first write the difference equation thus:

$$\frac{u_j^{n+1} - u_j^n}{dt} + u_j^{n+1} \frac{u_{j+1}^{n+1} - u_{j-1}^{n+1}}{2dx} = \nu \frac{u_{j+1}^n - 2u_j^n + u_{j-1}^n}{dx^2} \quad (8.29)$$

or:

$$-(cu_j^{n+1} + s)u_{j-1}^{n+1} + (1 - 2s)u_j^{n+1} + (cu_j^{n+1} - s)u_{j+1}^{n+1} = u_j^n \quad (8.30)$$

with c and s defined as before. The coefficients on the LHS, which would make up the matrix multiplying \mathbf{u}^{n+1} , depend on \mathbf{u}^{n+1} itself. Since we don't know that yet, we can't invert the matrix to obtain the solution.

What we will do is to *linearize* the equation, by replacing the u^{n+1} terms in the coefficients by u^n :

$$-(cu_j^n + s)u_{j-1}^{n+1} + (1 - 2s)u_j^{n+1} + (cu_j^n - s)u_{j+1}^{n+1} = u_j^n \quad (8.31)$$

This is obviously not correct, but it allows us to proceed by writing the matrix equation:

$$\mathbf{A}\mathbf{u}^{n+1} = \mathbf{u}^n \quad (8.32)$$

with:

$$\mathbf{A} = \begin{bmatrix} 1 + 2s & cu_1^n - s & 0 & 0 \\ -cu_2 - s & 1 + 2s & cu_2 - s & 0 \\ 0 & -cu_3 - s & 1 + 2s & cu_3 - s \\ 0 & 0 & -cu_4 - s & 1 + 2s \end{bmatrix} \quad (8.33)$$

Then we invert \mathbf{A} and write:

$$\mathbf{u}^{n+1} = \mathbf{A}^{-1} \mathbf{u}^n \quad (8.34)$$

The catch though is that we must invert \mathbf{A} at every time step. When we calculate \mathbf{u}^1 , we add the u^0 terms in \mathbf{A} . Then, to obtain \mathbf{u}^2 , we put the u^1 terms in \mathbf{A} and invert again. The inversions at each time step slow the calculation down. However, we can use larger values of s , which will speed things up.

Consider the same case, starting from a delta function initial condition (Fig. 8.9). The solution with $\nu = 0.5$ and $s = 0.4$ (upper left panel) agrees well with the analytic solution and with the FTCS solution (Fig. 8.8). But the calculation took 80 times as long as the FTCS solution!

So we use a larger value of s (upper right panel of Fig. 8.9). The numerical solution is essentially as good as it was before, and the calculation now is only 4 times slower than the FTCS. This is not terrific, but certainly better than 80 times slower.

However, the solutions are less accurate with a smaller viscosity. With $\nu = 0.01$ and $s = 0.32$ (lower left panel of Fig. 8.9), the iFTCS solution lags behind the analytical solution, and this lag increases with time. This is due to using the velocity from the previous time step when calculating the matrix \mathbf{A} . The discrepancy worsens with a larger time step (and larger s), as seen in the lower right panel. This is because the velocity \mathbf{u}^n differs more from \mathbf{u}^{n+1} , making the substitution less accurate. So we are required to use a smaller time step with iFTCS as well, to obtain accuracy.

Thus the iFTCS is much less attractive here. The need to invert the matrix \mathbf{A} at each time step greatly slows execution, and using the velocity from

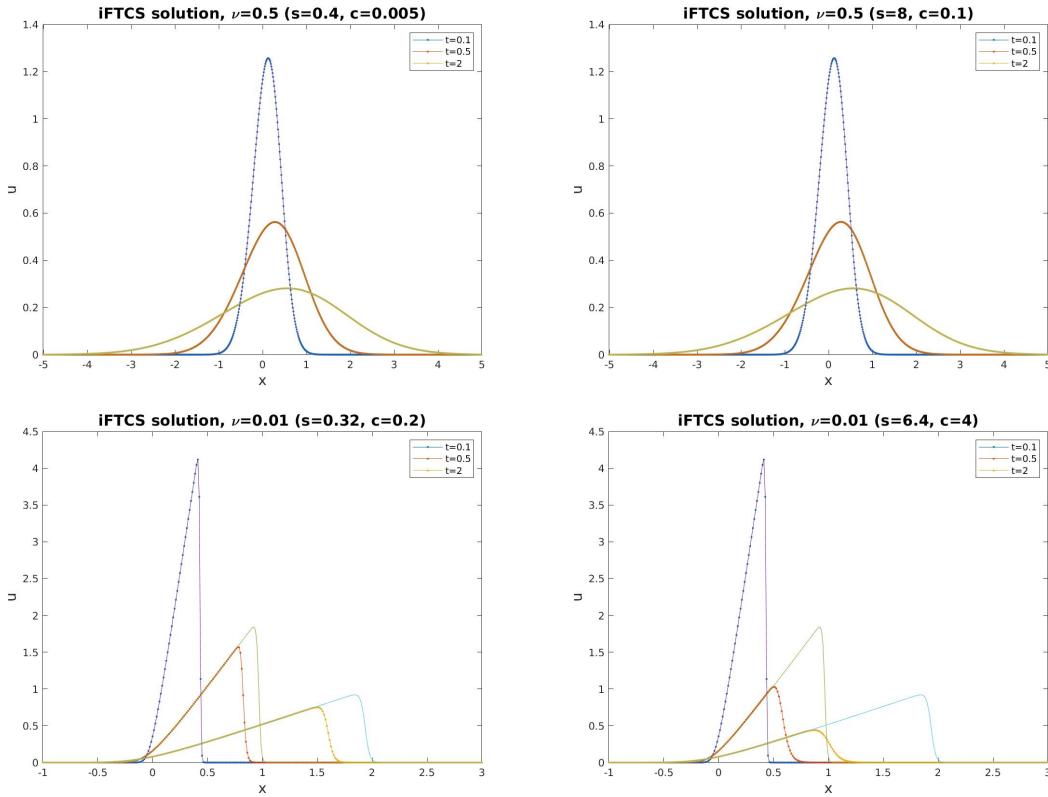


Figure 8.9: iFTCS solutions (dots) to Burger’s equation with a delta function initial condition, with $\nu = 0.5$ (upper panels) and $\nu = 0.01$ (lower panels). In the left panel, $s = 0.4$ and in the right, $s = 8$.

the previous time step in A causes the signal to propagate at the wrong speed. Given the success of the simpler FTCS scheme, we should stick with that. But let’s try one more explicit scheme first.

8.4.3 RK2

The Runge-Kutta schemes are also explicit, and of higher order accuracy in time than the FTCS scheme (sec. 3.2.3). We’ll consider the second order scheme, which is second order accurate in time. The main idea is that we take a half step first, to obtain $u^{n+1/2}$, the value at the mid-point. Then we use that to estimate the change from step n to $n + 1$.

We'll use the energy formulation of Burger's equation (8.22). Taking the FTCS finite difference equation (8.24), but evaluated at a half time step, we get:

$$u_j^{n+1/2} = u_j^n + \frac{1}{2}[cE_{j-1}^n - cE_{j+1}^n + su_{j-1}^n - 2su_j^n + su_{j+1}^n] \quad (8.35)$$

Then we evaluate the energy at the half step:

$$E_j^{n+1/2} = \frac{1}{2}u_j^{n+1/2} * u_j^{n+1/2} \quad (8.36)$$

We next use the half-step values to find u^{n+1} :

$$u_j^{n+1} = u_j^n + [cE_{j-1}^{n+1/2} - cE_{j+1}^{n+1/2} + su_{j-1}^{n+1/2} - 2su_j^{n+1/2} + su_{j+1}^{n+1/2}] \quad (8.37)$$

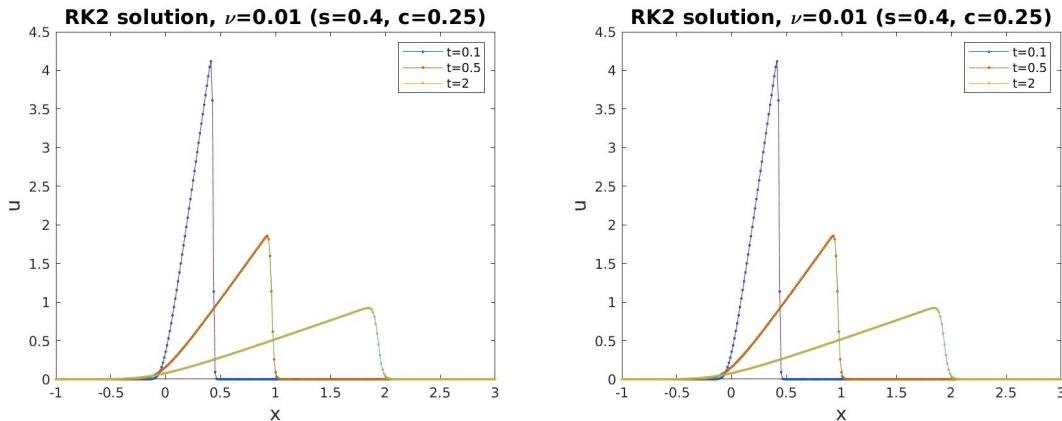


Figure 8.10: RK2 solutions (dots) to Burger's equation with a delta function initial condition, with $\nu = 0.1$ (left panel) and $\nu = 0.01$ (right panel).

Two examples are shown in Fig. (8.10), with $\nu = 0.1$ and $\nu = 0.01$. The agreement with the analytical solutions are excellent, and slightly better than with the FTCS solutions (although it's difficult to see in the figures). Close examination shows that the RK2 solution is closer, particularly near the peaks.

Again, this precision comes at a cost, as the calculation takes twice as long. But if accuracy is a concern, the RK2 scheme is preferable. Remember

too that because the scheme is explicit, we must keep the time step small enough so that $s < 0.5$. Using $s = 0.6$ in the above calculation yields a disastrous result.

8.5 Summary

Momentum advection yields interesting effects, because momentum is an *active* tracer. The one dimensional version of this is Burger's equation, which includes both advection and diffusion of momentum. Advection causes gradients to steepen, because wave crests move faster than the rest of the wave. This in turn leads to a tendency for waves to "fall over". Diffusion on the other hand weakens gradients, spreading the distributions out. This competition between advection and diffusion is also seen in higher dimensional problems, for example in the formation of fronts in the atmosphere and ocean.

The one dimensional Burger's equation can be solved analytically using the Cole-Hopf transformation. With this, we can convert the equation to a purely diffusive one, which is also linear. We considered several examples, with the most useful being a solution to an initial delta-function distribution.

We then simulated that case using three different numerical schemes. The FTCS scheme works remarkably well and is also the fastest. The iFTCS scheme, though unconditionally stable, is much slower because the mapping matrix, \mathbf{A} , must be inverted at each time step. To facilitate the inversion, we linearized \mathbf{A} , which led to inaccuracies in the solution. The second order Runge-Kutta scheme yielded the best results, but also required twice as long to execute as the FTCS scheme. Thus the explicit

schemes were best, with the trade-off between accuracy and speed of execution.

Chapter 9

Shallow water flows



Figure 9.1: Waves on the surface of Store Åklungen, north of Oslo.

The momentum equations discussed in Chapter (8) are missing an important ingredient: *forcing*. We'll add two. The first is *gravity*, which exerts a dominant force in the vertical direction. And the second is *pressure gradient*.

ents, which are a central element in the horizontal direction.

We first demonstrate how to include these effects, leading to the forced momentum equations. When combined with the continuity (density) equation, we obtain a complete set called the *shallow water equations* which can be used to model flow in lakes, rivers and the ocean. We'll then consider one particular phenomenon, that of surface waves on a homogeneous (single density) fluid. These are familiar to anyone who has been to the beach.

9.1 Navier-Stokes Equations

9.1.1 Gravity

We begin with gravity, which is the easiest force to incorporate. Gravity causes vertical acceleration, and hence enters in the equation for the vertical velocity, w . For a mass, m , the gravitational force is given by:

$$m \frac{dw}{dt} = -mg \hat{k} \quad (9.1)$$

or just:

$$\frac{dw}{dt} = -g \hat{k} \quad (9.2)$$

Including momentum advection and diffusion, the vertical momentum equation would then be:

$$\frac{\partial}{\partial t} w + \vec{u} \cdot \nabla w = -g \hat{k} + \nu \nabla^2 w \quad (9.3)$$

Note that gravity doesn't affect the horizontal velocity components.

9.1.2 Pressure gradients

Then there is pressure. In a gas, pressure results from the motion of gas molecules. If the gas is in a container, the molecules rebound off the con-

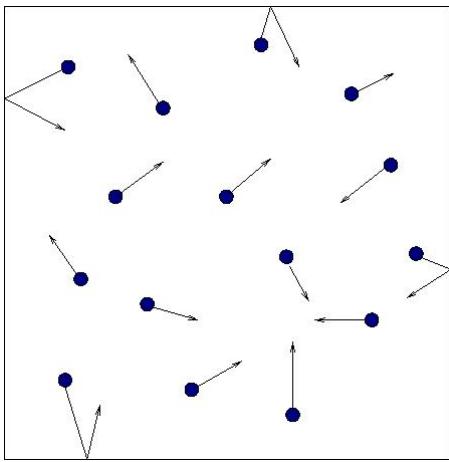


Figure 9.2: Gas molecules rebounding off container walls, exerting pressure on them.

tainer, exerting a force on the container walls (Fig. 9.2). The pressure is defined as the force per unit area exerted on the walls.

The higher the gas temperature, the higher the pressure is. This is because molecules in a hot gas are travelling faster than those in a slow gas, and exert more pressure on the container walls. Pressure and temperature are related by the *ideal gas law*:

$$p = \rho RT \quad (9.4)$$

where R is the universal gas constant, given by:

$$R = 287 \text{ J kg}^{-1}\text{K}^{-1}$$

Thus doubling the gas temperature results in a doubling of the pressure.

Differences in pressure produce fluid flow. Consider water in a sink, with a drain in the middle of the basin. When the plug is removed from the drain at the bottom, the water flows out and the surface is lowered over the drain (Fig. 9.3). This results in a difference in pressure – low pressure in the middle where there is less fluid, and high pressure on the sides. The water

then flows from the high pressure regions to the low, to replace the water draining out.

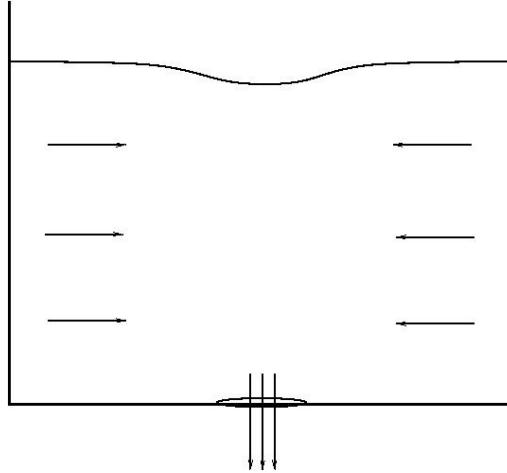


Figure 9.3: Flow in a sink. Water leaves via the drain at the bottom, lowering the fluid surface at the top. This causes a lateral pressure gradient, which drives fluid in towards the center.

To add the pressure term, let's return to our control volume (Fig. 5.2). If the pressure on the left side is $p(x)$ and the pressure on the right side is $p(x + dx)$, the volume will accelerate:

$$\frac{d}{dt} \iiint \rho u \, dV = Adx \frac{\partial}{\partial t} (\rho u) = p(x)A - p(x + dx)A \quad (9.5)$$

Expanding $p(x + dx)$ in a Taylor series, we get:

$$Adx \frac{\partial}{\partial t} (\rho u) = -\frac{\partial p}{\partial x} dx A \quad (9.6)$$

or:

$$\frac{\partial}{\partial t} (\rho u) = -\frac{\partial p}{\partial x} \quad (9.7)$$

If we include advection and diffusion, as in eq. (8.4), we have:

$$\frac{\partial}{\partial t} \rho u + \nabla \cdot (\rho \vec{u} u) = -\frac{\partial p}{\partial x} + \nu \frac{\partial^2}{\partial x^2} (\rho u) \quad (9.8)$$

We can simplify this as before, using the diffusive continuity equation (8.5), which yields:

$$\frac{\partial}{\partial t} u + \vec{u} \cdot \nabla u = -\frac{1}{\rho} \frac{\partial p}{\partial x} + \nu \nabla^2 u \quad (9.9)$$

after through by ρ .

Eq. (9.9) is x component of the full momentum equation. The equation for all three components is given by:

$$\frac{\partial}{\partial t} \vec{u} + \vec{u} \otimes \nabla \vec{u} = -\frac{1}{\rho} \nabla p - g \hat{k} + \nu \nabla^2 \vec{u} \quad (9.10)$$

This is the *Navier-Stokes equation*, named after physicists Claude-Louis Navier and George Stokes. It describes the motion of fluids with viscosity. It is the basis of equations used in modelling flow in the atmosphere, ocean, lakes, magma, etc.

9.2 Shallow water equations

But this is a complicated set of equations! In fact, there is actually an award (the Millennium Prize) on offer for anyone who can solve these equations in closed form (the prize is actually for anyone who can show that solutions exist!). Nobody has won this prize so far, and some believe that nobody will ever do so. Most often, people use numerical models to solve the equations instead.

We'll consider a simpler set of equations. These apply at intermediate scales, particularly at scales of 10s of kilometers and are relevant for large lakes and enclosed seas. But they can also be used to model glaciers, as discussed later. We'll make three assumptions: 1) that the density is constant, 2) that the flow is *hydrostatic* and 3) that the motion is *linear*.

9.2.1 Incompressibility

Take the density first. If $\rho = \text{const.}$, the continuity equation becomes simply:

$$\nabla \cdot (\rho \vec{u}) = \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0 \quad (9.11)$$

Equation (9.11) is known as the *incompressibility* condition. This is a much simpler relation than the full continuity equation, because it is linear. Many ocean and hydrological models use the incompressibility condition.

The condition implies that volume is conserved in the flow. Any inflow into a closed region must be compensated by an equal outflow. You can see this mathematically by integrating the non-divergence statement over a volume:

$$\int \nabla \cdot \vec{u} \, dV = \oint \vec{u} \cdot \hat{n} \, dS = 0 \quad (9.12)$$

which follows from Gauss' theorem. Thus the integral of the velocities over the sides of the box must sum to zero, and the net flow in or out of the box is zero.

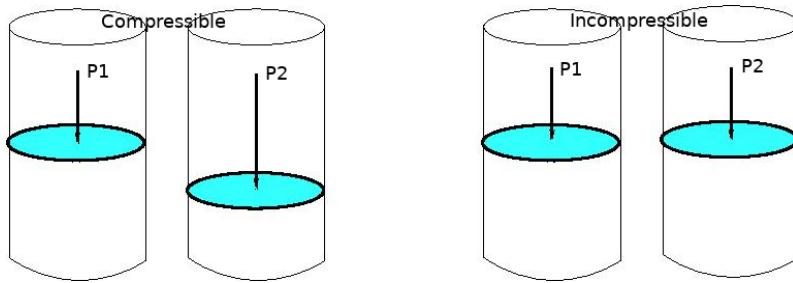


Figure 9.4: A schematic illustration of a compressible and an incompressible fluid. A compressible fluid (left), like air, shrinks in volume when exposed to a greater pressure. But an incompressible fluid (right) maintains its volume, regardless of the pressure.

The volume of an incompressible fluid does not change when exposed to different pressures (Fig. 9.4). Water is an example of an (approximately)

incompressible fluid. If water is placed in a piston, it is very difficult to force the lid down. This is the basis of a hydraulic pump – you can lift a car without greatly compressing the fluid.

A compressible fluid on the other hand shrinks depending on the applied pressure. Air is a good example. By pushing down on a piston full of air, we can easily move the lid down, and the extent it goes down determines the pressure (left panel of Fig. 9.4).

9.2.2 Hydrostatic balance

The hydrostatic assumption comes about from using scaling to evaluate the sizes of various terms in the vertical momentum equation at larger scales. It turns out that the two largest terms, by far, are gravity and the pressure gradient. Thus, to a good approximation, we can replace the equation with:

$$\frac{\partial}{\partial z} p = -\rho g \quad (9.13)$$

Notice that the same balance applies when the fluid is at rest, i.e. if we set $u = v = w = 0$ in the vertical momentum equation. This is why this is called the “hydrostatic” balance, since “hydro” is water and “static” is not moving.

Consider a layer of fluid at rest in a container (Fig. 9.5). The fluid is in a cylinder with area of A . The region indicated in the middle of the cylinder, with a thickness dz , has a mass:

$$m = \rho V = \rho A dz$$

and a weight, mg . The fluid underneath exerts a pressure upwards on the element, $p(z)$, while the fluid over exerts a pressure downwards, $p(z + dz)$.

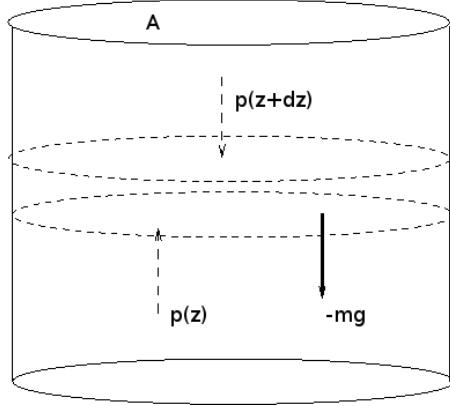


Figure 9.5: The hydrostatic balance.

The corresponding forces are the pressures times the area, A . Since the fluid is at rest, the forces must sum to zero:

$$p(z)A - p(z + dz)A - mg = 0$$

or

$$p(z + dz) - p(z) = -\frac{mg}{A} = -\rho g dz$$

Letting the height, dz , go to zero, we obtain (9.13). The hydrostatic balance thus explains why the ocean doesn't collapse under the force of gravity.

Hydrostatic balance is a significant simplification of the vertical momentum equation. If the density is constant, the equation is linear. Moreover it is a *diagnostic* relation, meaning that if we know the density, we can integrate to get the pressure. As such, we don't have to step the equation forward in time.

9.2.3 Linear shallow water equations

The third assumption is that the motion is linear. You can argue this by scaling the equations. But the result is simply that we ignore the nonlinear

advection terms on the LHS of the momentum equations. Doing this, we obtain a much simpler set:

$$\frac{\partial}{\partial t} u = -\frac{1}{\rho} \frac{\partial}{\partial x} p + \nu \nabla^2 u \quad (9.14)$$

$$\frac{\partial}{\partial t} v = -\frac{1}{\rho} \frac{\partial}{\partial y} p + \nu \nabla^2 v \quad (9.15)$$

$$\frac{\partial}{\partial z} p = -\rho g \quad (9.16)$$

$$\frac{\partial}{\partial x} u + \frac{\partial}{\partial y} v + \frac{\partial}{\partial z} w = 0 \quad (9.17)$$

In addition, we ignore the diffusive terms, since viscous effects at the scales of kilometers are very small.

We'll make an additional alteration. From the hydrostatic relation (the third equation), the pressure change with depth is constant if the density is constant. This implies the pressure gradients are the same at all depths. Specifically:

$$\frac{\partial}{\partial z} \left(\frac{\partial}{\partial x} p \right) = \frac{\partial}{\partial x} \left(\frac{\partial}{\partial z} p \right) = \frac{\partial}{\partial x} (-\rho g) = 0 \quad (9.18)$$

Thus a fluid parcel at depth will accelerate under pressure exactly as a parcel at the surface will, and as such, u and v are the same at all depths. Thus fluid moves in columns in this constant density fluid.

Furthermore, the pressure gradients are set by the height of the surface of the fluid. To see this, integrate the hydrostatic relation from a depth z to the surface, at $z = \eta$:

$$\int_z^\eta \frac{\partial}{\partial z} p dz = p(\eta) - p(z) = -\rho g(\eta - z) \quad (9.19)$$

If we ignore the pressure at the surface (which usually varies over much larger scales), we have:

$$p(z) = \rho g(\eta - z) \quad (9.20)$$

Thus we have:

$$\frac{\partial}{\partial x} p = \rho g \frac{\partial}{\partial x} \eta, \quad \frac{\partial}{\partial y} p = \rho g \frac{\partial}{\partial y} \eta \quad (9.21)$$

Thus we can rewrite the horizontal momentum equations as:

$$\frac{\partial}{\partial t} u = -g \frac{\partial}{\partial x} \eta \quad (9.22)$$

$$\frac{\partial}{\partial t} v = -g \frac{\partial}{\partial y} \eta \quad (9.23)$$

These are fairly simple equations and appealing to work with. But we only have two of them, while we have three unknowns, (u, v, η) . So we need an additional equation.

We get this by integrating the incompressibility condition over the entire fluid depth:

$$\int_{-H}^{\eta} \left(\frac{\partial}{\partial x} u + \frac{\partial}{\partial y} v + \frac{\partial}{\partial z} w \right) dz = (\eta + H) \left(\frac{\partial}{\partial x} u + \frac{\partial}{\partial y} v \right) + w(\eta) - w(-H) \quad (9.24)$$

The horizontal velocities move through the integral since they don't vary with z . Now we just need the vertical velocities at the upper and lower surfaces. We'll assume the bottom is flat, neglecting bottom topography.

Then we have:

$$w(-H) = 0$$

For the vertical velocity at the surface, consider a fluid parcel on the surface. This has:

$$z = \eta$$

Taking the Lagrangian derivative of both sides, we have:

$$\frac{dz}{dt} = w(\eta) = \frac{d\eta}{dt}$$

Putting this all together, the integrated incompressibility condition is:

$$\frac{d\eta}{dt} + (\eta + H)\left(\frac{\partial}{\partial x}u + \frac{\partial}{\partial y}v\right) = 0 \quad (9.25)$$

This is our third equation, since it also involves η , u and v . In keeping with our other equations, we'll linearize this, thus:

$$\frac{\partial}{\partial t}\eta + H\left(\frac{\partial}{\partial x}u + \frac{\partial}{\partial y}v\right) = 0 \quad (9.26)$$

Notice we've neglected the advective term implicit in the Lagrangian derivative, and products of η , u and v .

This leaves us with our full set, the linear *shallow water equations*:

$$\frac{\partial}{\partial t}u = -g\frac{\partial}{\partial x}\eta \quad (9.27)$$

$$\frac{\partial}{\partial t}v = -g\frac{\partial}{\partial y}\eta \quad (9.28)$$

$$\frac{\partial}{\partial t}\eta + H\left(\frac{\partial}{\partial x}u + \frac{\partial}{\partial y}v\right) = 0 \quad (9.29)$$

This is a relatively simple system, and one which can be used to model different ocean and lake flows. If one retains the variable depth, i.e. $H(x, y)$, one has essentially the same linear equations used to model the barotropic tide, i.e. the tidal component which doesn't vary with water depth.

9.3 Surface waves

Waves familiar to anyone who has been in a boat or at the beach. Regular undulations on the surface can propagate over long distances. They are generated by the winds, but also by earthquakes which can produce large

and destructive *tsunamis*. A tsunami is a surface wave, but can grow to very large amplitude when it propagates in to the shore.

The shallow water system has wave solutions. We can easily reduce the three equations (9.29) to a single one for the surface height. We do this by taking a time derivative of the third equation, and substituting in using the first two equations:

$$\begin{aligned} \frac{\partial^2}{\partial t^2}\eta + H\frac{\partial}{\partial x}\left(\frac{\partial}{\partial t}u\right) + H\frac{\partial}{\partial y}\left(\frac{\partial}{\partial t}v\right) &= \\ \frac{\partial^2}{\partial t^2}\eta - gH\frac{\partial^2}{\partial x^2}\eta - gH\frac{\partial^2}{\partial y^2}\eta &= 0 \end{aligned} \quad (9.30)$$

Or:

$$\frac{\partial^2}{\partial t^2}\eta - c_g^2\nabla^2\eta = 0 \quad (9.31)$$

where:

$$c_g = \sqrt{gH} \quad (9.32)$$

is the *gravity wave speed*. Equation (9.31) is a hyperbolic equation (sec. 2.4). It differs from those we have studied before because the time derivative is second order. This changes slightly the analytical and numerical solutions.

9.3.1 Analytical solution (1D)

Imagine waves propagating in a single direction (for example, towards a long beach). We can orient the coordinates so that y is parallel to the crests and x is parallel to the direction of propagation. Then the wave equation (9.31) can be written as a one dimensional equation:

$$\frac{\partial^2}{\partial t^2}\eta - c_g^2\frac{\partial^2}{\partial x^2}\eta = 0 \quad (9.33)$$

Because of the second order time derivative, we can't use the techniques we've used previously. But notice that we can *factor* the derivatives on the LHS:

$$\left(\frac{\partial}{\partial t} - c_g \frac{\partial}{\partial x}\right) \left(\frac{\partial}{\partial t} + c_g \frac{\partial}{\partial x}\right) \eta = 0 \quad (9.34)$$

As such, the equation is equivalent to the product of two first order wave equations:

$$\frac{\partial}{\partial t} \eta - c_g \frac{\partial}{\partial x} \eta = 0 \quad (9.35)$$

and

$$\frac{\partial}{\partial t} \eta + c_g \frac{\partial}{\partial x} \eta = 0 \quad (9.36)$$

We examined these equations in sec. (5.1). Following that discussion, we know the solution is:

$$\eta(x, t) = F_l(x + c_g t) + F_r(x - c_g t) \quad (9.37)$$

Here, F_l is a function which propagates to the left at the gravity wave, c_g , and F_r is another function which propagates to the right at the same speed. Both functions can be determined from the initial conditions.

For example, consider that a meteor strikes the ocean. The surface of the ocean is depressed in response, and we'll assume that depression is Gaussian-shaped. Shortly after impact, the depression splits in two, one going left at a speed c_g and one going right at the same speed.

How do we solve this problem? From expression (9.37), the initial height is given by:

$$\eta(x, 0) = F_l(x) + F_r(x) \quad (9.38)$$

So the left and right parts must add up to the initial Gaussian. Let's call that $G(x)$, such that:

$$G(x) = -Ae^{-(x-x_0)^2/L^2} \quad (9.39)$$

But notice we can't solve the problem, because we have two unknowns (F_l and F_r) but only one equation.

Let's say then that the initial condition isn't moving left or right initially, so that:

$$\frac{\partial}{\partial t}\eta(x, 0) = 0 \quad (9.40)$$

This would happen for example if the meteor fell straight down, rather than at an angle. Taking the derivative of (9.37), we get:

$$\frac{\partial}{\partial t}\eta(x, t) = \frac{\partial}{\partial t}(F_l + F_r) = c_g F'_l - c_g F'_r \quad (9.41)$$

Here the prime means a derivative with respect to the argument, as discussed in sec. (5.1.2). For example, if we write:

$$\theta = x + c_g t$$

then:

$$\frac{\partial}{\partial t}F_l(\theta) = c_g \frac{\partial}{\partial \theta}F_l \equiv c_g F'_l$$

If the initial depression isn't moving, then:

$$c_g F'_l - c_g F'_r = 0 \quad \rightarrow \quad F'_l = F'_r \quad (9.42)$$

We can integrate this once to get:

$$F_l = F_r \quad (9.43)$$

after discarding the constant of integration (why can we do this?).

Taken together, this says:

$$\begin{aligned} \eta(x, t) &= \frac{1}{2}G(x + c_g t) + \frac{1}{2}G(x - c_g t) \\ &= -\frac{A}{2}e^{-(x+c_g t-x_0)^2/L^2} - \frac{A}{2}e^{-(x-c_g t-x_0)^2/L^2} \end{aligned} \quad (9.44)$$

Physically, the initial depression splits into two equal depressions, each half the amplitude of the first, which propagate left and right at speed c_g .

How fast is this actually? If the ocean were a single density fluid, we could estimate H with the mean ocean depth, which is approximately 4000 m. Then the gravity wave speed would be:

$$c_g = \sqrt{9.8(4000)} \approx 200 \text{ m/sec} \quad (9.45)$$

which is fast!

This is also a simple model of the ocean response to an earthquake. In that case, the ocean bottom violently shifts up or down. That produces a surface disturbance, like a step, which is lower on one side and higher on the other. The step then disperses into waves, which move away at the gravity wave speed. The waves move quickly, and so pose a risk to surrounding coastal areas.

A key point about gravity waves in the shallow water system is that they are *non-dispersive*. This means that long and short waves all move at the same speed. Thus a Gaussian-shaped wave can maintain its form. The Gaussian represents a superposition of different sized waves, but these all move at the same speed, c_g , preserving the original shape. Thus even if the meteor had been *square* (like a Lego meteor), the resulting half-square waves would also propagate away at the gravity wave speed, c_g .

9.3.2 Numerical solution (1D)

Now we'll simulate the surface waves numerically. Equation (9.33) differs from the equations we've examined before because it has a second

order difference in time. But we know how to write a second order spatial difference, and the time difference is coded the same way. Using center differences for both t and x , we have:

$$\frac{\eta_j^{n+1} - 2\eta_j^n + \eta_j^{n-1}}{dt^2} - c_g^2 \frac{\eta_{j+1}^n - 2\eta_j^n + \eta_{j-1}^n}{dx^2} = 0 \quad (9.46)$$

Rearranging, this is:

$$\eta_j^{n+1} = C^2 \eta_{j-1}^n + 2(1 - C^2) \eta_j^n + C^2 \eta_{j+1}^n - \eta_j^{n-1} \quad (9.47)$$

where $C = c_g dt/dx$ is the Courant number for gravity waves.

As with the Leapfrog scheme (sec. 3.2.2), the center differenced scheme poses a problem because we require η^{n-1} to begin the integration. This follows because two time derivatives requires two initial conditions. For the same reason, we had two unknowns for the analytical solution for the meteor impact; we obtained those by imposing a second initial condition:

$$\frac{\partial}{\partial t} \eta(x, 0) = 0 \quad (9.48)$$

Written in difference form, this is:

$$\frac{\eta_j^1 - \eta_j^0}{dt} = 0 \quad (9.49)$$

which implies:

$$\eta_j^0 = \eta_j^1 \quad (9.50)$$

Thus for the first step, we use:

$$\eta_j^2 = C^2 \eta_{j-1}^1 + (1 - 2C^2) \eta_j^1 + C^2 \eta_{j+1}^1 - \eta_j^1 \quad (9.51)$$

Thereafter, we continue with (9.47). Thus the next step would be:

$$\eta_j^3 = C^2 \eta_{j-1}^2 + 2(1 - C^2) \eta_j^2 + C^2 \eta_{j+1}^2 - \eta_j^1 \quad (9.52)$$

and so on.

Written in matrix form, this is:

$$\mathbf{N}^{n+1} = A\mathbf{N}^n - \mathbf{N}^{n-1} \quad (9.53)$$

with:

$$A = \begin{bmatrix} 2(1 - C^2) & C^2 & 0 & 0 \\ C^2 & 2(1 - C^2) & C^2 & 0 \\ 0 & C^2 & 2(1 - C^2) & C^2 \\ 0 & 0 & C^2 & 2(1 - C^2) \end{bmatrix} \quad (9.54)$$

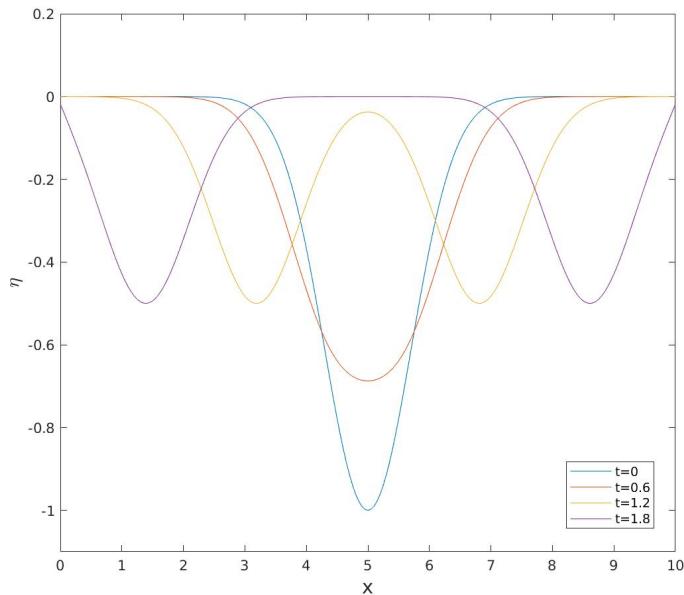


Figure 9.6: An initial Gaussian depression. The Gaussian (in blue) is initially centered at $x = 5$ and has a half-width of $L = 1$. The Courant number is $C = 0.5$ and the gravity wave speed is $c_g = 1$. The depression splits in two, with one half moving left and the other right.

The initial Gaussian depression case is shown in Fig. (9.6). The domain goes from $x = [0, 10]$, and the Gaussian, with a half-width of one, sits initially at $x = 5$. We've set the Courant number to 0.5 and the gravity wave speed, $c_g = 1$. At $t = 0.6$, the Gaussian has slumped and spread. By

$t = 1.2$ it has clearly separated into two waves; by $t = 1.8$, the waves no longer overlap and are propagating left and right.

As discussed in Chp. (5), the solution is unstable if the Courant number is too large. Using $C = 1$ yields nearly identical results, but with $C = 1.2$, the simulation explodes, with the amplitudes reaching 10^{18} by $t = 1.8$. But using C smaller than 0.5 yields nearly identical results to those with $C = 0.5$.

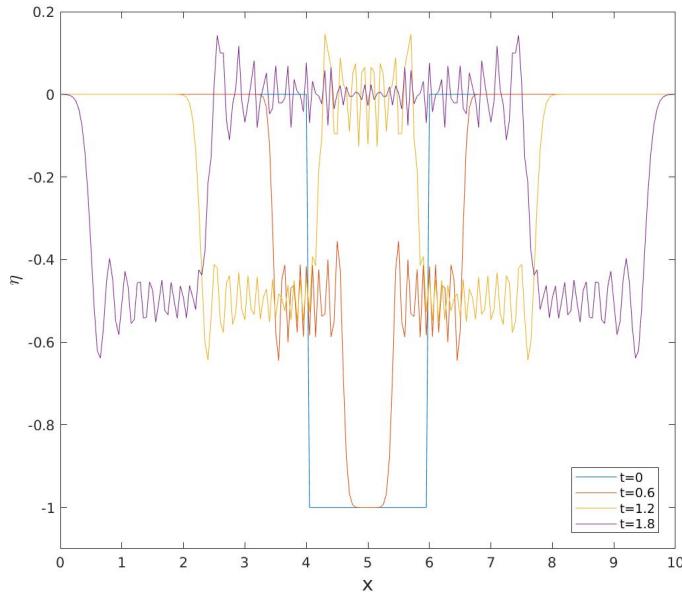


Figure 9.7: An square depression. The initial wave (in blue) is centered at $x = 5$ and has a width of 2. The Courant number is again $C = 0.5$ and the gravity wave speed is $c_g = 1$.

We can also test the Lego meteor (Fig. 9.7). The initial depression is square, with an amplitude of one and a width of two. The wave also splits in two, with each daughter wave moving away at c_g . But pronounced oscillations develop almost immediately on the wave. These are known as “Gibbs phenomenon” and occur because the gradients at the edges of the initial wave are too steep. Decreasing the Courant number to 0.01 doesn’t

help matters, yielding nearly the same result.

Thus the center differenced scheme yields excellent results, so long as the Courant number is small enough to ensure stability and the gradients aren't too sharp. Steep gradients require more sophisticated numerical techniques, as are also required in modelling propagating fronts. But this gives you an idea about how the system behaves.

9.4 Summary

We derived the shallow water equations, a simplified set of equations for a constant density fluid. The equations can be used to simulate a wide range of phenomena, from surface waves to depth-invariant tides in the ocean. We considered one of the simplest solutions to the equations, of surface waves in one dimension. These propagate non-dispersively at the “gravity wave speed”, $c_g = \sqrt{gH}$. But many more solutions are possible, as discussed for example in courses on oceanography.

We then simulated the 1D waves numerically. The second order time derivative necessitated using a different time step, in this case a center differenced one. With a centered difference spatial derivative, the model is fully second order. The model worked well, as long as the Courant number, $C = c_g dt / dx$ was one or less, and if the initial gradients weren't too steep.

Chapter 10

Appendix A: Diffusion example with Python

An Explicit Scheme for a 1D Diffusion Equation¹

In this section, we will in detail go through all the required steps to translate a physical problem (Fig. 10.1) to Python code (version 2.x).

Let's consider a chamber of fluid bounded by two walls (Fig. 10.1). Initially ($t = 0$), the fluid is at rest, but at $t > 0$ the left wall starts moving with velocity $V_{wall} = 40$ m/s. The evolution of the fluid velocity $V(x, t)$ perpendicular to the x -axis can be modeled by a 1D diffusion equation.

What is the velocity of the fluid at $t = 0.2$ s?

Step 1: Formulate the mathematical problem

Based on the information in the text and figure above, the mathematical formulation of this physical problem reads:

$$\begin{aligned} V &= 0 \quad x \in [0, L] \quad \text{for } t = 0, \\ \frac{\partial V}{\partial t} &= \nu \frac{\partial^2 V}{\partial x^2} \quad x \in (0, L) \end{aligned}$$

¹The following is an example in Python coding written by Krister Karlsen.

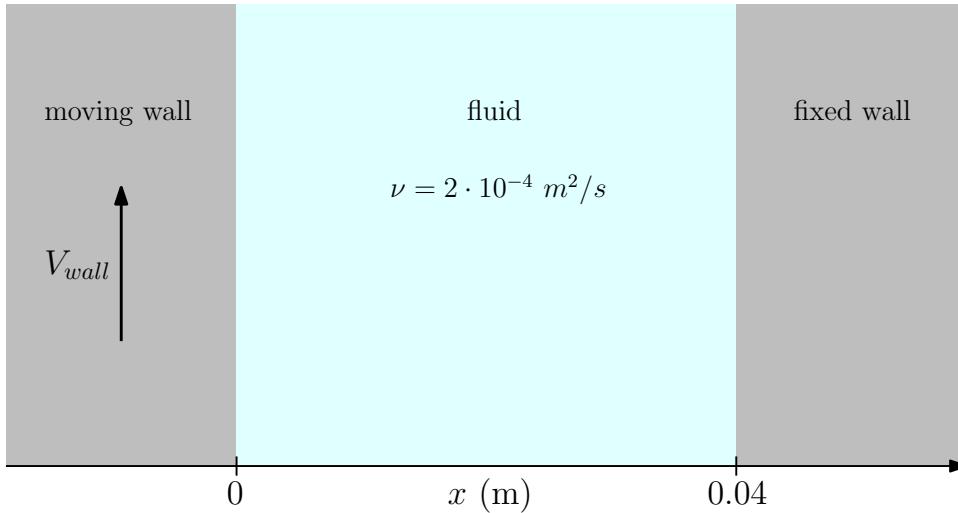


Figure 10.1: Sketch of a fluid chamber with one moving, and one fixed wall.

subject to the following BCs:

$$V(x = 0) = V_{wall}, \quad V(x = L) = 0 \quad \text{for } t > 0 \quad (10.1)$$

Implementation: We start out by importing our favorite Python libraries and defining the domain and various constants. Hot tips: when dealing with explicit numerical schemes for diffusion equations, always print out the diffusion number $s = \nu\Delta t/\Delta x^2$. to see if the stability requirement is met (See Section 3.2.1)!

```

import numpy as np
import matplotlib.pyplot as plt

# Define problem and constants
x0 = 0.0                      # Domain start (m)
x1 = 0.04                       # Domain end (m)
N_x = int(100)                   # Numbers of points in space
dt = 0.0004                      # Time-step (s)
D = 2e-4                         # Diffusivity (m^2/s)
V_wall = 40.0                     # Boundary condition "left", moving wall (m/s)
T = 0.2                           # End time (s)
x = np.linspace(x0,x1,N_x)       # Grid (m)
dx = x[1]-x[0]                   # Grid spacing (m)
s = D*dt/dx**2                    # Diffusion number
print "Diffusion number s is now:", s

```

Step 2: Discretize the equations

The continuous mathematical problem above has to be formulated on a finite number of points in space and time to be solvable on a computer, this is called *discretization*. Here, we will consider a grid consisting of N_x points in space, and N_t points in time (Fig. 10.2). Note that if we stick with the same indexing convention as Python, the index of the first points are 0, while the index of the last points are $N_x - 1$ and $N_t - 1$ respectively.

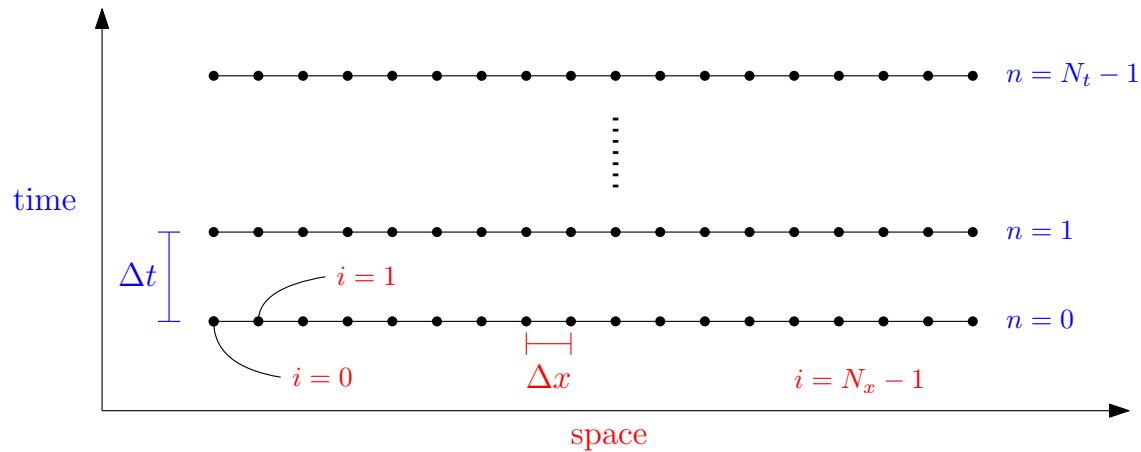


Figure 10.2: Indexing of variables in space and time.

Continuing with the convection of index notation, we substitute the derivatives with finite differences, *forward* in time and *centered* in space (see Section 3.1), to obtain the following equation:

$$\frac{V_i^{n+1} - V_i^n}{\Delta t} = \nu \frac{V_{i+1}^n - 2V_i^n + V_{i-1}^n}{\Delta x^2}$$

The initial condition can now be formulated as $V_i^0 = 0$ for $i = 0, 1, 2, \dots, N_x - 1$, and the boundary conditions are $V_0^n = V_{wall}$ and $V_{N_x-1}^n = 0$, for $n = 1, 2, \dots, N_t - 1$.

Step 3: Formulate equations for the unknowns

Solving the discretized equation for the unknown V_i^{n+1} yields

$$V_i^{n+1} = sV_{i-1}^n + (1 - 2s)V_i^n + sV_{i+1}^n,$$

This equation provides the solution for one single point space, meaning that we need to solve N_x such equations per time step (if we include the boundaries). To solve multiple equations at the same time we can formulate a matrix-vector equation. Just to get a sense of what the matrix-vector system we are dealing with looks like, it is useful write out the equations by hand for a small number of grid points in space. Here we will use $N_x = 5$.

$$\begin{aligned} i = 0 \quad & V_0^{n+1} = V_{wall} \\ i = 1 \quad & V_1^{n+1} = sV_0^n + (1 - 2s)V_1^n + sV_2^n \\ i = 2 \quad & V_2^{n+1} = sV_1^n + (1 - 2s)V_2^n + sV_3^n \\ i = 3 \quad & V_3^{n+1} = sV_2^n + (1 - 2s)V_3^n + sV_4^n \\ i = 4 \quad & V_4^{n+1} = 0 \end{aligned}$$

Now, from the equations above, we recognize that this can be written as a matrix-vector equation $\mathbf{V}^{n+1} = \mathbf{A}\mathbf{V}^n$, where

$$\mathbf{A} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ s & 1 - 2s & s & 0 & 0 \\ 0 & s & 1 - 2s & 1 & 0 \\ 1 & 0 & s & 1 - 2s & s \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \quad \mathbf{V}^{n+1} = \begin{pmatrix} V_0^{n+1} \\ V_1^{n+1} \\ V_2^{n+1} \\ V_3^{n+1} \\ V_4^{n+1} \end{pmatrix}, \quad \mathbf{V}^n = \begin{pmatrix} V_{wall} \\ V_1^n \\ V_2^n \\ V_3^n \\ V_4^n \\ 0 \end{pmatrix}.$$

We immediately observe that the matrix \mathbf{A} takes special form, i.e. $A_{0,0} = 1$ and $A_{N_x-1,N_x-1} = 1$, and $A_{i,i-1} = s$, $A_{i,i} = 1 - 2s$ and $A_{i,i+1} = s$ for $i = 1, 2, \dots, N_x - 1$, while the remaining elements are zero.

Implementation: It's time to assemble the matrix A and the initial vector \mathbf{V}^n .

```
# Assemble matrix A
A = np.zeros([N_x,N_x])
A[0,0] = 1                      # Top "left corner" of matrix
A[N_x-1,N_x-1] = 1              # Bottom "right corner" of matrix
for i in range(1,N_x-1):        # Loop over rows of A (excluding i=0 and i=N_x-1)
    A[i,i-1] = s
    A[i,i] = 1-2*s
    A[i,i+1] = s
print "A =",A                  # Check that A is assembled correctly

# Assemble vector V_n
V_n = np.zeros(N_x)             # All elements initially zero except boundaries
V_n[0] = V_wall                 # Boundary condition "left", moving wall
print "V_n =",V_n              # Check that V_n is assembled correctly
```

Notice that the Python code for assembling this matrix elements $A_{i,j}$ is almost identical to the mathematical notation used above! Hot tips: always print out the matrix and vector to see that they are assembled correctly, like shown in the example.

Next, we need to solve the matrix equation $\mathbf{V}^{n+1} = \mathbf{AV}^n$ at each time-step, until we reach the "end time" $t = 0.2$ s (referred to as T in the code). This is done by simply multiplying the matrix A by \mathbf{V}^n to obtain \mathbf{V}^{n+1} (referred to as \mathbf{V} in the code). NB! Before we move to the next time-step we need to update "the old" \mathbf{V}^n to take the value of \mathbf{V}^{n+1} .

```
# Loop forward in time
time = 0 # from time=0 ..
while time <= T: # .. to time = T
    time += dt # Add dt to current time
    print "Time =", time # Check that time is updated correctly
    V = np.dot(A,V_n) # Compute V_{n+1} = A*V_n
    V_n = V # Update V_n
```

Finally, we can run the code and plot the result (Fig. 10.3).

```
# Plot solution
plt.title("Solution at time="+str(time))
plt.plot(x,V)
plt.xlabel("x (m)")
plt.ylabel("V (m/s)")
plt.grid()
plt.show()
```

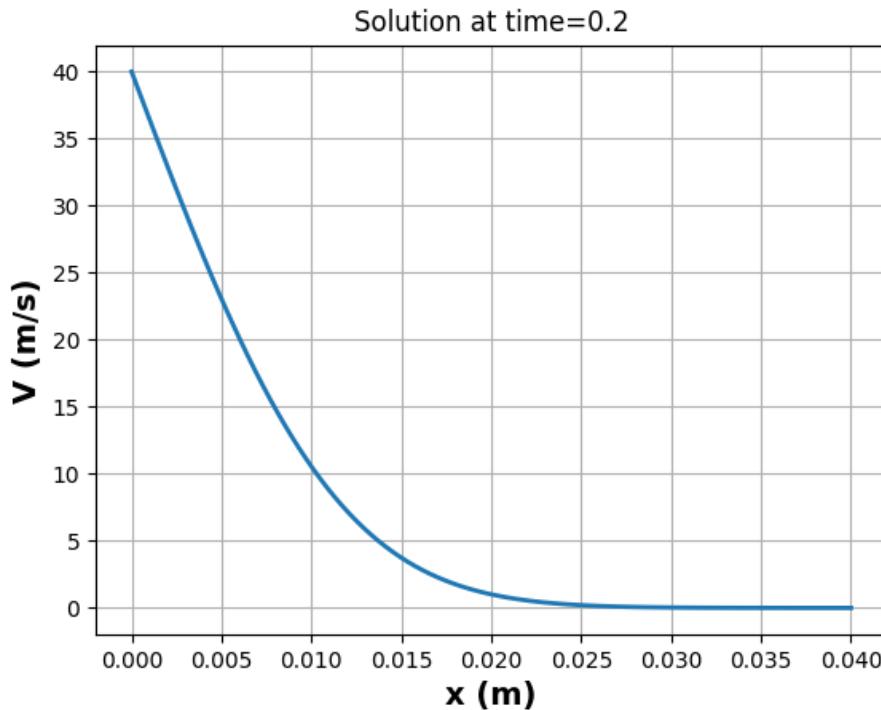


Figure 10.3: Numerical solution at time = 0.2s.

Chapter 11

Appendix B: Numerical stability

Under construction...