Chapter 7

partial differential equations: boundary value + initial value problems

7.1 definitions

Partial differential equations (PDEs) form the basis of very many quantitative problems in Earth Science. In some cases, the problem can be simplified such that an ordinary differential equation (ODE), or series of coupled ODEs, will suffice but more generally, the quantity in which we have interest varies with respect to more than one independent variable.

7.1.1 general form: second-order linear PDE in two dimensions

Consider a partial differential equation with independent variables x and y :

$$a\frac{\partial^2 u}{\partial x^2} + b\frac{\partial^2 u}{\partial x \partial y} + c\frac{\partial^2 u}{\partial y^2} + d\frac{\partial u}{\partial x} + e\frac{\partial u}{\partial y} + fu + g = 0 \tag{7.1}$$

where the coefficients a, b, c, d, e, f and g may be functions of x and y but are not functions of the dependent variable u.

Equation (7.1) is the general form of a linear second-order PDE, meaning that all possible terms are included. It is second-order because the highest-order derivative is a second derivative. It is linear because none of the coefficients depend on the dependent variable u. The *behavior* of equation (7.1) depends on the relationships amongst its coefficients and various PDEs are classified using the characteristic $(b^2 - 4ac)$

- elliptic $b^2 - 4ac < 0$
- parabolic $b^2 - 4ac = 0$

- hyperbolic $b^2 - 4ac > 0$

The classification scheme is handy for mathematicians because they express something about the expected behavior of the equation. This may, in turn, inform us regarding a numerical integration scheme. Elliptic equations tend to have smooth solutions and parabolic equations tend to smooth over time while hyperbolic equations tend to preserve and propagate large gradients. The diffusion equation discussed later in this chapter is parabolic. The equation for flow in a parallel-sided channel in the previous chapter was elliptic. Advective transport is described by a hyperbolic equation. Computationally, what we want to know is whether there are any derivatives in time. That is, are we solving a boundary value problem or an initial value and boundary value problem?

As we begin to consider differential equations in multiple space dimensions, it is worthwhile to review a few concepts from vector calculus. For a more thorough discussion, consult your calculus book. Simply defined, a vector is a quantity that has both magnitude and direction. Some vector-valued quantities are velocity, acceleration, force, stress, and strain.

7.1.2 unit vectors

A unit vector has magnitude of 1. That is, its magnitude has been normalized using the L_2 norm (or Euclidean norm)

$$\hat{u} = \frac{u}{\|u\|}$$

Here, we will use unit vectors in our usual cartesian coordinate system with axes x, y and z:

$$\hat{\imath}, \hat{\jmath}, \hat{k}$$

The unit vectors can be used to write the vector u in terms of its three components in space:

$$\boldsymbol{u} = u_x\hat{\imath} + u_y\hat{\jmath} + u_z\hat{k}$$

If one of the three unit vectors is zero, the vector exists on a two-dimensional plane instead of in a three-dimensional volume.

7.1.3 scalar fields and vector fields

The variables we use in constructing equations to simulate physical processes contain both scalar and vector-valued quantities. When we expand our range of view beyond just one point in space, we begin to deal with “fields” of scalars and vectors.

Consider the two-dimensional representation of Earth’s surface: a map (or gridded data set) with x and y axes with corresponding land surface elevations zs. At every location in the plane (on the map), $zs(x, y)$ has some magnitude. The surface elevations are a scalar field.

Suppose instead we instead are interested in the surface slope α at each (x, y). Slope has both magnitude and direction. Thus, $\alpha(x, y)$ is a vector field. Another example of a vector field is the velocity in a moving fluid observed in an Eulerian frame.

7.1.4 dot and cross products of vectors

Vector products were introduced in the introduction to linear algebra in lab 6. Here, the concepts of dot and cross products are reviewed. The immediate interest is is their use in the following section on gradient, divergence, and curl.

The **dot product** of two vectors is the sum of the products of the i'th elements of the vectors. Consider two vectors a and b:

$$a \cdot b = \sum_{i=1}^{n} a_i b_i \tag{7.2}$$

where $i = x, y, z$. The dot products of unit vectors are an important identity:

$$\hat{i} \cdot \hat{i} = \hat{j} \cdot \hat{j} = \hat{k} \cdot \hat{k} = 1$$

The **cross product** of two vectors is another vector. The magnitude is the product of the magnitudes of the two vectors and the sine of the angle between them. The direction of the vector is perpendicular to the plane defined by the two original vectors:

$$a \times b = \|a\| \, \|b\| \sin\theta \hat{n} \tag{7.3}$$

where θ is the angle between the two vectors and

$$\hat{n}$$

is a unit vector in the direction of the cross product. This is perhaps best understood with an example. Suppose:

$$a = 2\hat{i} + 5\hat{j} + 1\hat{k} \quad \text{and} \quad b = 1\hat{i} - 3\hat{j} + 2\hat{k}$$

then a cross b is found using the determinant:

$$\begin{aligned} &= \begin{vmatrix} \hat{i} & \hat{j} & \hat{k} \\ 2 & 5 & 1 \\ 1 & -3 & 2 \end{vmatrix} \\ &= \hat{10}i + 1\hat{j} - 6\hat{k} - 5\hat{k} + 3\hat{i} - 4\hat{j} \\ &= \hat{13}i - 3\hat{j} - \hat{11}k \end{aligned}$$

7.1.5 vector differentials

PDEs in more than one space dimension are the result of differentiation of a either a scalar, such as temperature, or vector, such as velocity, with respect to more than one dimension. Mathematically, these operations are represented using the vector differential operator ("del"):

$$\nabla = \frac{\partial}{\partial x}\hat{\imath} + \frac{\partial}{\partial y}\hat{\jmath} + \frac{\partial}{\partial z}\hat{k}$$

The vector differential can be used in many ways.

The **gradient** of a scalar field $\phi(x, y, z)$ is the product of the vector differential operator and the scalar:

$$\begin{aligned} &= \left(\tfrac{\partial}{\partial x}\hat{\imath} + \tfrac{\partial}{\partial y}\hat{\jmath} + \tfrac{\partial}{\partial z}\hat{k}\right)\phi \\ &= \tfrac{\partial \phi}{\partial x}\hat{\imath} + \tfrac{\partial \phi}{\partial y}\hat{\jmath} + \tfrac{\partial \phi}{\partial z}\hat{k} \end{aligned} \tag{7.4}$$

The gradient of the scalar field is a vector field. If we take ϕ to represent surface elevation of a landform, then $\nabla\phi$ is the surface slope, a vector-valued quantity. When problems are reduced to one or two dimensions, the relevant terms in the differential operator are reduced accordingly.

The **divergence** of a vector field $u(x, y, z)$ is the dot product of the vector differential operator and the vector:

$$\begin{aligned} &= \left(\tfrac{\partial}{\partial x}\hat{\imath} + \tfrac{\partial}{\partial y}\hat{\jmath} + \tfrac{\partial}{\partial z}\hat{k}\right)\cdot\left(u_x\hat{\imath} + u_y\hat{\jmath} + u_z\hat{k}\right) \\ &= \tfrac{\partial u_x}{\partial x} + \tfrac{\partial u_y}{\partial y} + \tfrac{\partial u_z}{\partial z} \end{aligned} \tag{7.5}$$

where the identity involving the dot products of unit vectors is used. If u is a velocity field in a fluid, then $\nabla \cdot u$ is the strain rate. It is important to note that u must be differentiable at each (x, y, z) in the region of interest. The same must have been true for ϕ in the case of the gradient, and must be true for the following definition as well.

The **curl** of a vector field $u(x, y, z)$ is the cross product of the vector differential operator and the vector:

$$
\begin{aligned}
&= \left(\frac{\partial}{\partial x}\hat{i} + \frac{\partial}{\partial y}\hat{j} + \frac{\partial}{\partial z}\hat{k}\right) \times \left(u_x\hat{i} + u_y\hat{j} + u_z\hat{k}\right) \\
&= \begin{vmatrix} \hat{i} & \hat{j} & \hat{k} \\ \frac{\partial}{\partial x} & \frac{\partial}{\partial y} & \frac{\partial}{\partial z} \\ u_x & u_y & u_z \end{vmatrix} \qquad (7.6) \\
&= \left(\frac{\partial u_z}{\partial y} - \frac{\partial u_y}{\partial z}\right)\hat{i} + \left(\frac{\partial u_x}{\partial z} - \frac{\partial u_z}{\partial x}\right)\hat{j} + \left(\frac{\partial u_y}{\partial x} - \frac{\partial u_x}{\partial y}\right)\hat{k}
\end{aligned}
$$

The curl is often called the rotation of the vector field. In the case where u is a continuous velocity field, its curl would represent the turbulence in the flow.

7.2 finite differences for PDEs

The schemes developed for two-point boundary value problems are applied in the same way for partial differential equations. In the case of multidimensional problems, the differences are formed on a discrete grid that represents the model domain. As before, we require an appropriate number of boundary conditions and in the case of time-dependent problems, an initial condition for every node in the space domain.

7.2.1 time domain

When the PDE is time dependent, a decision must be made regarding the interval over which to apply the difference. Different schemes have different stability characteristics with different types of equations. Recall that by their nature, numerical integration introduces error into our calculations. Schemes for which the error (that is, the difference between the numerical solution and the exact solution) is uniformly bounded for successive steps in the integration are called numerically stable. it is often useful to think about the way in which information *flows* through the solution scheme when considering numerical stability.

A forward difference in time, in which the superscript on y is used to indicate the step in time,

$$
\left.\frac{\partial y}{\partial t}\right|_{t_n} = \frac{y^{n+1} - y^n}{\Delta t} + error
$$

yields an **explicit** (or *forward Euler*) scheme which is easy to implement but not always stable. The scheme is easy to implement because the function evaluated at the present time step, t_n, is used to compute the dependent variable at the next time step y^{n+1}.

A backward difference in time

$$
\left.\frac{\partial y}{\partial t}\right|_{t_{n+1}} = \frac{y^{n+1} - y^n}{\Delta t} + error
$$

yields what is called an **implicit** (or *backward Euler*) scheme. Here, the function evaluated at t_{n+1} is used to compute the value of the dependent variable y^{n+1}. We have more information on the far side of the interval in time, which is good news for stability but results in a more complicated computation than in the explicit scheme. We may also write this using n and $n-1$ for the function evaluated at t_n.

7.3 boundary and initial conditions

7.3.1 initial condition

Each node in the space domain requires an initial value at n=1. The values may be from a set of observations, a steady-state calculation, or may be chosen arbitrarily. Finding the right initial condition for a time-dependent model is important. Each new calculation uses values of the dependent variable from the last time step so the initial condition is carried forward in time as a model runs. The numerical model will retain a "memory" of the initial condition for a number of time steps that depends on the rate of change in the dependent variable. Additionally, approximations in the model equations and numerics may cause the calculated state of a system to differ from its natural state for a particular set of boundary conditions and material properties.

For some studies, memory of an initial state is important. For example, you may be interested in how a change in a boundary condition or coefficient in the governing equation affects a known state of a system. In that case, the model would be initialized with the known state and then used to track how the dependent variable(s) in responds to the change. It is challenging to initialize from a known (observed) state because numerical models never represent *exactly* the real world in which the observations were made. This is a common issue in physical oceanographic modeling, where observations made at large intervals and mostly on the surface of the ocean must be used to produce values for all variables in the model domain (the whole ocean) yet remain true to the observations. Statistical techniques are applied to ensure that fidelity to observations in a process called *data assimilation*.

For other studies, the goal is to either find a steady state solution or start some other calculation from a steady state initialization. In the case of boundary conditions that are constant in time, the model steady state would be one in which the value of the dependent variable at any node j does not change from time step to time step. In the case of boundary conditions that make one complete cycle over some period τ, the model steady state would be one in which the value of the dependent variable at any node j is constant at integer multiples of τ. The model steady state may be found by running the model with fixed boundary conditions (or with repeated cycles of the same time-varying conditions) until changes in values of the unknown dependent variable fall below some threshold, either from n to $n+1$ or from τ to $m\ \tau$ where m is an integer. As in other iterative calculations, the modeler selects a tolerance criterion at which the calculation should stop. This process is often called "spinning up" the model.

7.3.2 Dirichlet boundary conditions

Boundary conditions in which values of the dependent variable specified at every time step are called Dirichlet conditions, after the German mathematician Johann Peter Gustav Lejeune Dirichlet (1805 to 1859). Lejeune Dirichlet worked in the field of number theory and developed the modern definition of a function.

Dirichlet conditions can be handled in more than one way in an implicit numerical scheme. The simplest approach is to include statements of boundary values θ as the first and last equations in the system of equations we wish to solve in an implicit scheme.

$$y_1^n = \theta_1^n \tag{7.7}$$

and

$$y_{N+1}^n = \theta_{N+1}^n \tag{7.8}$$

7.3.3 Neumann boundary conditions

It is also possible to use known derivatives of the dependent variable as boundary conditions. Derivative conditions are called Neumann conditions, after the German mathematician Carl Gottfried Neumann (1832 to 1925), who worked on the theory of integral equations.

In many circumstances, we do not know the value of the dependent variable at a boundary but we may either know or be able to assume something useful about its gradient. For example, if we are interested in heat flow through a soil, we may specify a geothermal gradient at $j = N + 1$. We would need to be sure that the boundary at which the gradient condition is applied is far enough away from the region of interest that uncertainty in this condition does not dominate the rest of the model domain.

Applying such a condition,

$$\phi_{N+1}^n = \left.\frac{\partial y}{\partial x}\right|_{x_{N+1},t_n} \tag{7.9}$$

where ϕ is the magnitude of the gradient. The problem now has an additional unknown, the value of y at $j = N + 1$. The derivative may be approximated as either a backward difference:

$$\phi_{N+1}^n \approx \frac{y_{N+1}^n - y_N^n}{\Delta x} \tag{7.10}$$

(forward at $j = 1$) or a double-interval centered difference:

$$\phi_{N+1}^n \approx \frac{y_{N+2}^n - y_N^n}{2\Delta x} \tag{7.11}$$

(at either $j = 1$ or $j = N + 1$). In the latter case, a fictitious node $N + 2$ is used for the purpose of generating the final equation. Writing the finite difference equation for $j = N + 1$ would yield an unknown term y_{N+2}. The boundary condition is applied by replacing y_{N+2} with its equivalent from equation (7.11).

7.4 example: thermal diffusion in one space dimension

Diffusion is a fundamental concept for many problems in Earth science. Most simply, we can think of diffusion as the time-dependent change in the distribution of particles in a region by random motion from regions of higher concentration to regions of lower concentration. Often, we are concerned with a **flux**, the amount of a quantity that flows through a unit area in a unit time, due to a gradient in that quantity. *We would do well to keep in mind that advective fluxes are also important, but that is a topic for another day.*

Because diffusion involves both time and space dimensions, it is described by a partial differential equation.

7.4.1 a statement of conservation

The first law of thermodynamics tells us that the temperature of any particular parcel of space (say a unit volume of soil or of air) changes if the heat flowing into the parcel is different than the heat flowing out of the parcel. We can write this casually as:

$$\frac{dT}{dt} = -\nabla Q \tag{7.12}$$

where T represents temperature, t represents time, Q represents a vector-valued heat flow and ∇ (*The symbol ∇ is called "del" or sometimes "nabla"*) is a vector differential operator that tells us to take the first derivative of Q in each of its dimensions. In a cartesian coordinate system, this vector-valued gradient of Q is:

$$\begin{aligned} \nabla Q &= \left(\frac{\partial}{\partial x} + \frac{\partial}{\partial y} + \frac{\partial}{\partial z}\right) Q \\ \nabla Q &= \frac{\partial Q}{\partial x} + \frac{\partial Q}{\partial y} + \frac{\partial Q}{\partial z} \end{aligned} \tag{7.13}$$

Considering only one space dimension, z, equation (7.12) may be written:

$$\frac{dT}{dt} = -\frac{dQ}{dz} \tag{7.14}$$

Equation (7.14) is written such that the heat flow is positive in the positive z direction.

7.4.2 a constitutive relation

Heat flow through a material depends on the temperature gradient and a material property K, the thermal diffusivity. Intuitively, we can think of this as "excess" heat in a relatively warm region diffusing toward adjacent, relatively cooler regions, simply because of the temperature difference. In our one space dimension this is:

$$Q = -\mathrm{K}\frac{dT}{dz} \tag{7.15}$$

The dimensions of Q are MASS $\times$ LENGTH^2/TIME^3, which in MKS units is Watts. The dimension of K is LENGTH^2/TIME .

Combining equations (7.14) and (7.15) yields the thermal diffusion equation in one dimension.

$$\begin{aligned} \frac{\partial T}{\partial t} &= -\frac{\partial}{\partial z}\left(-K\frac{\partial T}{\partial z}\right) \\ \frac{\partial T}{\partial t} &= K\frac{\partial^2 T}{\partial z^2} \end{aligned} \tag{7.16}$$

This is a partial differential equation involving derivatives of the dependent variable in both time and space.

7.4.3 steady state

A system at steady state does not change with respect to time. For our problem, this is:

$$0 = \mathrm{K}\frac{\partial^2 T}{\partial z^2} \tag{7.17}$$

Equation (7.17) states that the temperature gradient $\partial T/\partial z$ has a constant value between the boundaries of the problem domain (because its second derivative is zero). Equation (7.17) could be solved analytically for $T(z)$ by integrating twice using two boundary conditions. The result is not very exciting.

7.5 finite difference approximation of the 1D diffusion equation

The numerical solution to equation (7.16) requires us to approximate both the space and time derivatives. We will use centered differences for the space domain and consider two different options for the time derivative. In the following, a subscript j will be used to represent the space coordinate and a superscript n will be used to represent time:

$$T_j^n \equiv T\left(z(j), t(n)\right)$$

7.5.1 space domain

Using centered differences, the second derivative of T with respect to z at time t_n and depth z_j is written:

$$\left.\frac{\partial^2 T}{\partial z^2}\right|_{z_j,t_n} = \mathrm{K}\frac{T^n_{j+1} - 2T^n_j + T^n_{j-1}}{\Delta z^2} + error \tag{7.18}$$

in which Δz represents the interval between successive z_j. The bar on the left-hand side indicates that the derivative is evaluated at the specified z and t.

7.5.2 time domain

The time derivative may be discretized in several ways. We will consider two single-step schemes called explicit (forward looking) and implicit (backward looking) schemes. The consequences for selecting one method or the other are in the numerics required to solve the resulting equations and in the stability of the numerical model.

7.5.2.1 explicit

The explicit time step uses a forward difference:

$$\left.\frac{\partial T}{\partial t}\right|_{z_j,t_n} = \frac{T^{n+1}_j - T^n_j}{\Delta t} + error \tag{7.19}$$

7.5.2.2 implicit

The implicit scheme uses a backward difference:

$$\left.\frac{\partial T}{\partial t}\right|_{z_j,t_n} = \frac{T^n_j - T^{n-1}_j}{\Delta t} + error \tag{7.20}$$

7.5.3 complete finite difference equations

The finite difference model is classified according to the nature of the differences used in its construction. Here, we will produce centered-difference explicit and centered-difference implicit models.

7.5.3.1 explicit

Using equations (7.18) and (7.19) to re-write (7.16) yields:

$$\frac{T_j^{n+1} - T_j^n}{\Delta t} \approx \mathrm{K}\frac{T_{j+1}^n - 2T_j^n + T_{j-1}^n}{\Delta z^2} \tag{7.21}$$

which can be re-arranged to group terms involving unknown quantities (left-hand side) and known quantities (right-hand side) at the time step t_{n+1}:

$$T_j^{n+1} \approx T_j^n + \frac{\mathrm{K}\Delta t}{\Delta z^2}\left(T_{j+1}^n - 2T_j^n + T_{j-1}^n\right) \tag{7.22}$$

No worries. Because equation (7.22) contains only one unknown value of the dependent variable, it can be solved directly. There is no need to build a linear algebra solution. That simplicity makes the explicit solution attractive but it comes at a price. Explicit solutions are only stable for certain combinations of step sizes. For our diffusion equation, stability is ensured when:

$$0 < 4\frac{\mathrm{K}\Delta t}{\Delta z^2} < 1 \tag{7.23}$$

This expression is the result of a "von Neumann" stability analysis, a topic best left to another class.

7.5.3.2 implicit

Using equations (7.18) and (7.20) to re-write (7.16) yields:

$$\frac{T_j^{n} - T_j^{n-1}}{\Delta t} \approx \mathrm{K}\frac{T_{j+1}^n - 2T_j^n + T_{j-1}^n}{\Delta z^2} \tag{7.24}$$

which can be re-arranged to group terms involving unknown quantities (left-hand side) and known quantities (right-hand side) at the time step t_n:

$$T_j^n - \frac{\mathrm{K}\Delta t}{\Delta z^2}\left(T_{j+1}^n - 2T_j^n + T_{j-1}^n\right) \approx T_j^{n-1} \tag{7.25}$$

Yikes. Equation (7.25) requires a bit more effort to implement than does equation (7.22) because we must solve the system of equations representing T^n at all unknown j simultaneously. Lucky for us, we learned how to do that in the last chapter. The implicit scheme has important advantages that make it worth the effort. It is unconditionally stable and for large problems, the matrix inversion is much more efficient than the direct calculation required by the explicit scheme.

7.6 boundary and initial conditions

7.6.1 initial condition

Each T_j requires an initial condition at n=1. The T_j^1 may be from a set of observations, a steady-state calculation, or may be chosen arbitrarily. Here, the goal is to find a steady state solution using a time varying upper surface boundary condition. The model steady state may be found by running the model with repeated cycles of the same time-varying T_1^n until changes in values of the unknown dependent variable fall below some threshold, either from n to $n+1$ or from τ to $m\ \tau$ where m is an integer.

7.6.2 Dirichlet boundary conditions

$$T_1^n = \theta_1^n \tag{7.26}$$

and

$$T_{N+1}^n = \theta_{N+1}^n \tag{7.27}$$

7.6.3 Neumann boundary conditions

In the case of heat balance in a surface soil layer, we may not know the temperature at the base of the soil layer or we might not want to be bound to the assumption that the 10-meter temperature is equal to the mean annual temperature. As an alternative, we may specify a geothermal gradient at $j = N + 1$. Applying such a condition,

$$\phi_{N+1}^n = \left.\frac{\partial T}{\partial z}\right|_{z_{N+1},t_n} \tag{7.28}$$

where ϕ is the magnitude of the gradient. The problem now has an additional unknown, the value of T at $j = N + 1$. The derivative may be approximated as either a backward difference:

$$\phi_{N+1}^n \approx \frac{T_{N+1}^n - T_N^n}{\Delta z} \tag{7.29}$$

(forward at $j = 1$) or a double-interval centered difference:

$$\phi_{N+1}^n \approx \frac{T_{N+2}^n - T_N^n}{2\Delta z} \tag{7.30}$$

(at either $j = 1$ or $j = N + 1$). In the latter case, a fictitious node $N + 2$ is used for the purpose of the mathematical development. It will not appear in the final equation.

Using the double-interval central difference with an implicit scheme, and

$$\gamma = \mathrm{K}\frac{\Delta t}{\Delta z^2}$$

for simplicity of notation, equation (7.16) is written for $j = N + 1$:

$$-\gamma T_N^n + (1 + 2\gamma) T_{N+1}^n - \gamma T_{N+2}^n = T_{N+1}^{n-1} \tag{7.31}$$

Equation (7.30) is used to replace T_{N+2}:

$$-\gamma T_N^n + (1 + 2\gamma) T_{N+1}^n - \gamma \left(2\Delta z \phi_{N+1}^n + T_N^n\right) = T_{N+1}^{n-1} \tag{7.32}$$

which rearranged, yields:

$$-2\gamma T_N^n + (1 + 2\gamma) T_{N+1}^n = T_{N+1}^{n-1} + 2\gamma \Delta z \phi_{N+1}^n \tag{7.33}$$

7.7 creating the numerical model

7.7.1 model domain

The numerical model requires both space and time domains. We will define the space domain such that z is positive downward into the subsurface. The time domain will begin at $ti = 0$ and proceed with a specified step size through a specified number of time steps.

The span of the space domain is $\{z_1 : z_{N+1}\}$ where N is the number of intervals and $N + 1$ is the number of nodes in the domain. We will specify boundary values for T at z_1 and z_{N+1}. Let $z_1 = 0$ represent the ground surface. The maximum depth to which we expect the surface temperature signal to penetrate is a reasonable value to select for z_{N+1}. That depth depends on the thermal diffusivity of the soil. Typically, the temperature at about 10 meters depth is equal to the mean annual temperature. (*The USDA Soil Survey Manual includes a nice summary of soil temperature characteristics. You can find the manual online at* `http://soils.usda.gov/procedures/ssm/main.htm`). The domain will be divided into intervals of size dz.

Matlab code to set up the model domain:

```
%* 1-D thermal diffusion model
clear

%* define some constants
K=2.8e-7;                 % thermal diffusivity of dry sand, m^2/s
day=24*60*60;             % seconds in one day
year=365;                 % days in one year

%* model domain
zsurf=0;                  % surface coordinate, m
zbase=10;                 % base coordinate, m "deep"
dz=0.1;                   % vertical interval size, m
z=[zsurf:dz:zbase]';      % vertical coordinates, m
N=length(z)-1;            % intervals in model domain
```

```
ti=0;                    % initial time
tf=1;                    % final time in years
nD=1;                    % time step size in days
dt=nD*day;               % time step size in seconds
M=tf*year/nD;            % number of time steps, days
```

7.7.2 Dirichlet boundary conditions

The space-domain part of the model requires two boundary conditions for T at each time step, one at T_1 (zsurf) and one at T_{N+1} (zbase). Here, the surface temperature will be specified as a sinusoidal variation over 365 days. Beginning at the minimum temperature and using the mean Tsm and annual range about that mean Tr the surface temperature cycle is

$$Tsm \; + \; \frac{Tr}{2} \; \sin\phi$$

in which $\phi = \{-1/2\pi : 3/2\pi\}$. The basal temperature will be taken as the mean annual temperature. In MATLAB we could set this up as follows:

```
%* boundary conditions
% start model at coldest part of temperature cycle, -1/2 pi

Tsurfmean=280;           % mean annual temperature, K
Tdiff=30;                % annual temperature range, K
Tsurf=Tsurfmean + Tdiff/2*sin(linspace(-1/2*pi, (tf*2-1/2)*pi, M));
Tbase=Tsurfmean;
```

7.7.3 initial condition on T

The time derivative requires an initial condition for every $T(z,t)$. A simple approach is to specify a constant slope between the initial T_1 and T_{N+1}, that is, a steady state dT/dz for $T(z_1,0)$.

```
%* variables
T=zeros(N+1,M);          % T history

%* insert boundary and initial conditions in matrix T
% IC: use first Tsurf and monotonic dT/dz to Tbase

T(1,1:M)=Tsurf;
T(N+1,1:M)=Tbase;
T(:,1)=linspace(Tsurf(1),Tbase, N+1)';
```

Through this section, the setup is the same for either an explicit or implicit scheme.

7.7.4 explicit, centered-difference scheme

The direct calculation is very simple to program. This model would be written using the MATLAB code in sections 7.7.1, 7.7.2, and the two nested **for** loops, the outer loop over time and the inner loop over space.

```
for n=1:M-1                   % time steps, n=1 is initial condition
        for j=2:N             % space steps, N+1 nodes
         T(j, n+1) = T(j,n) + K*dt/dz^2 * (T(j+1,n) - 2*T(j,n) + T(j-1,n));
        end
end
```

While simple, the explicit model has an important limitation: it is only conditionally stable (equation (7.23)). The peril comes when you try to run the explicit model with an inappropriately large Δt for your chosen Δz, as is the case with the Δt defined in section 7.7.2. The time step size would need to be closer to an hour than a day to run this model successfully.

7.7.5 implicit, centered-difference scheme

The implicit model requires us to build a matrix $\boldsymbol{A}$ for all the coefficients of all unknown T_j^n and a vector b to hold the known $T_j^{n-1}$and the boundary conditions at each time t_n. The dimensions of $\boldsymbol{A}$ are $[N+1, N+1]$, and the dimension of b is $[N+1, 1]$.

This particular model setup is relatively simple because none of the coefficients of T involve the independent variable t. Thus, we need only construct $\boldsymbol{A}$ once. The elements of b include the modeled T_j^{n-1}so this vector must be re-created at each time step.

7.7.6 one Neumann condition

The model setup for the derivative boundary condition version of our problem is similar to that for the implicit model with two Dirichlet conditions except that an additional constant must be defined, the number of unknowns increases to N and the last rows in the matrix of coefficients $\boldsymbol{A}$ and the vector of known values and constants b are constructed in a slightly different manner (according to equation (7.33)). The lower boundary of the model domain must be moved far away from the region of interest, perhaps $z_{N+1} = 50$. A new scheme is needed for initializing T. One possibility is to specify T_j^1 using the mean annual surface temperature and the geothermal gradient. The model must be spun up over many years before a steady state is attained.

```
%** define constants
K=2.8e-7;                     % thermal diffusivity of dry sand, m^2/s
Ggrad=25/1000;                % geothermal gradient K/m
```

The initialization could be

```
T(1,1:M)=Tsurf;
T(N+1,1)=Tsurfmean+Ggrad*zbase;
T(:,1)=linspace(Tsurfmean,T(N+1,1), N+1)';
```

7.7.7 plot some results

Our simulation of the time-variation in subsurface temperature may be plotted in many different ways. Two possibilities are scripted below. The second produces a three-dimensional view of the $T(z,t)$ parameter space using MATLAB's **surf** function.

```
%** plot result

%* a family of curves representing T(z) at selected time steps
Tmin=min(min(T));
Tmax=max(max(T));

figure(1)
clf
axis([Tmin Tmax -zbase zsurf])  % use -z to plot in a perspective
                                % that makes physical sense

hold on
plot(T(:,2:30:M-1), -z, 'c-')   % plot every 30th day
plot(T(:,1), -z, 'b-')          % plot initial condition

xlabel('depth (m)')
ylabel('T (K)')

%* the complete T(z,t) space
figure(2)
surf(z,[ti:M-1],T')
shading interp
xlabel('depth (m)')
ylabel('day of year')
zlabel('T (K)')
```

7.8 exercises

1. Implement both the implicit and explicit models of subsurface thermal diffusion using Dirichlet boundary conditions as described in section 7.7.

    (a) Run the implicit model with $\Delta z = 0.1$ meter, $\Delta t = 1$ day, $\kappa = 2.8 \times 10^{-7}$ m^2 s^{-1}, and $z_{N+1} = 10$ m. Plot the result as suggested in section 7.7.7, figure (1). Run the explicit model with the same step sizes and make a second figure. What went wrong with the explicit model?

    (b) How would you fix the problem with the explicit model in part 1a?

2. Finding the right initial condition for a time-dependent model can be a tricky business, as discussed in section 7.6.1. A one dimensional soil profile is relatively straightforward but image initializing an ocean circulation model. A common technique is to run the model with fixed boundary conditions (or with repeated cycles of the same time-varying conditions) until

changes in values of the unknown dependent variables fall below some threshold. This is often referred to as "spinning up" the model.

To spin our thermal diffusion model with Dirichlet boundary conditions through multiple years, we need only change one variable, the final time (in years):

```
%** model domain

zsurf=0;                 % surface coordinate, m
zbase=10;                % base coordinate, m
dz=0.1;                  % vertical interval size, m
z=[zsurf:dz:zbase]';     % vertical coordinates, m
N=length(z)-1;           % intervals in model domain

ti=0;                    % initial time
tf=5;                    % final time, in years
nD=1;                    % time step size in days
dt=nD*day;               % time step size in seconds
M=tf*365/nD;             % number of time steps
```

tf automatically tells our program to use multiples of the fundamental number of time steps (which here is 365 days) in constructing the surface boundary condition.

```
%* boundary conditions
% start model at coldest part of temperature cycle, -1/2 pi

Tsurfmean=280;           % mean annual temperature, K
Tdiff=30;                % annual temperature range, K
Tsurf=Tsurfmean + Tdiff/2*sin(linspace(-1/2*pi, (tf*2-1/2)*pi, M));
Tbase=Tsurfmean;
```

How many years are required to spin the model, as configured here, up to a steady state?

3. Replace the basal boundary condition in your implicit model with a Neumann condition using the geothermal gradient of 25 K km^{-1}. Spin the model up for 5 years.

   (a) Was the assumption about the temperature at 10 meters depth being equal to the mean annual temperature valid?

   (b) Don't trust the initial condition? Try another. This initialization uses the coldest surface temperature and the geothermal gradient:

   ```
   %* insert boundary and initial conditions into matrix T
   % initial condition: use coldest Tsurf and monotonic trend to Tbase

   T(1,1:M)=Tsurf;
   T(N+1,1)=Tsurf(1)+Ggrad*zbase;
   T(:,1)=linspace(Tsurf(1),T(N+1,1), N+1)';
   ```

   Again, spin the model up over 5 years. Explain what's happening to the temperature of the subsurface and why it's happening.

Appendix A

Conservation equations

A.1 introduction

The mathematical descriptions of physical systems we encounter in Earth science problems have at their hearts statements requiring the conservation of energy, momentum, or mass. We see this fundamental principle written in many different forms for many different quantities but strictly, they all derive from one place. In this chapter, the origin of conservation equations is described.

A.2 conservation of ...

Conservation involves the change in a quantity ϕ (with dimension of *something*/L^3) in an arbitrary inertial volume V enclosed by a surface S. A unit vector $\hat{n}$ normal to S, defined to be positive outward, is used to identify the regions of space within and outside of S.

The value of ϕ within V may change over time t if there is a flux through S or creation of ϕ within V. The flux may have two parts, one due to diffusion and another due to advection. The change in ϕ within V is written:

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_V \phi \mathrm{d}V \;=\; -\int_S \mathbf{F}\cdot\hat{n}\;\mathrm{d}S \;-\; \int_S \phi\mathbf{V}\cdot\hat{n}\;\mathrm{d}S \;+\; \int_V H \mathrm{d}V \tag{A.1}$$

where $\mathbf{F}$ represents the flux due to diffusion, $\phi\mathbf{V}$ represents advection of ϕ in a fluid flow with velocity $\mathbf{V}$, H represents a source (or sink) of ϕ within V and boldface indicates a vector-valued quantity. The negative signs in front of the first two terms on the right-hand side of (A.1) indicate that an outward flux results in a decrease of ϕ in the volume enclosed by S.

Equation (A.1) is a statement of conservation of ϕ for the unit volume V. This statement is always true, independent of the size of V and even if the fields enclosed by S are not continuous. This is the case because we integrate over the whole volume. It is important to note, however, that the integration also means that information on spatial scales smaller than V is not available to us.

One way to visualize the importance of the integration in equation (A.1) is to consider rice being poured from a bag. The rice is a discontinuous mixture of grains and air. If we focus in on a small volume of the flowing rice, in any instant of time we might see rice, we might see air, or we might see a mixture of the two. If we zoom out, and look again, we see the flow. If we integrate over that larger volume, the pouring rice looks like a fluid flow but we can't say anything about what is happening to individual grains.

A.2.1 reduced dimensionality

Reduced spatial detail is often beneficial, as was the case in the zero-dimension energy balance model in chapter 4. By considering average (or "integrated") properties of Earth's surface and atmosphere, we were able to avoid complicated equations for radiative transfer and build an efficient model that let us explore some fundamental aspects of Earth's mean climate. Another example where reduced dimensionality is beneficial are linear reservoir problems such as the ocean chemistry "box" model in chapter 3.

The zero-dimension approximation dictates that we consider only mean quantities. This is expressed:

$$\bar{f} \;=\; \frac{1}{V}\int_V f \;\mathrm{d}V \tag{A.2}$$

where f is any function (ϕ or H in equation (A.1)) and the overbar indicates a mean. In our polluted lake example, we assume that the reservoir is well-mixed and f corresponds to the contaminant in the lake and V represents the lake volume. We would use a similar statement to address any sources or sinks within the lake.

The flux terms are grouped into two terms, one each for the rates at which the conserved quantity is going in and coming out through S:

$$\int_S (\mathbf{F} \;+\; \phi\mathbf{V}) \cdot \mathrm{d}S \;=\; \dot{\phi}_{out} - \;\dot{\phi}_{in} \tag{A.3}$$

where the dot indicates a rate. The ϕ rate is equivalent to the product of water flow rate and contaminant concentration in our lake pollution problem.

Using these definitions, equation (A.1) is simplified:

$$\tfrac{\mathrm{d}}{\mathrm{d}t}\tfrac{1}{\mathbf{V}}\textstyle\int_V \phi \;\mathrm{d}V \;=\; \dot{\phi}_{in} \;-\dot{\phi}_{out} \;+\tfrac{1}{\mathbf{V}}\textstyle\int_V H \;\mathrm{d}V \tag{A.4}$$

and becomes:

$$\frac{\mathrm{d}\phi}{\mathrm{d}t} \;=\; \dot{\Phi}_{in} \;-\; \dot{\Phi}_{out} \;+\; \bar{H} \tag{A.5}$$

in which Φ represents the volumetric rate of ϕ,

$$\dot{\Phi} = \frac{\dot{\phi}}{\mathbf{V}} \quad \text{(A.6)}$$

Equation (A.5) is an ordinary differential equation, which may be solved exactly, by integration, or numerically, as an initial value problem. Equations derived in this manner are often linked together to form what are called *box models.* As we saw in the planetary energy balance problem, such models are quite powerful for conceptual studies but interpretation of the results derived from such models must be done with care. Additionally, it is often the case that the fluxes depend on the concentrations (as was the case with the temperature-dependent albedo in the energy balance model), in which case the equations are nonlinear and must be solved iteratively.

The zero dimension conservation equation is elegant but its use is restricted. Spatial, as well as temporal, variations are important to may problems in Earth science. Thus, we are also interested in applying equation (A.1) to cases in which ϕ is not constant within V.

A.2.2 (locally) continuous fields

In order to make progress for situations in which ϕ varies in space, we must be able to write equation (A.1) as a partial differential equation. This requires the derivatives of ϕ to exist within V. That is, our spatial field ϕ must be nearly constant within within smaller, subdivided regions of V and vary smoothly from little region to little region. It is important to recognize that there is still some lower limit at which our assumption of a locally continuous field breaks down (recall the example of rice pouring from a bag). That limit varies from system to system and among quantities within a given system. When we discuss the variation of ϕ in space, we are really talking about the average ϕ's in small regions of space. We can make no assertions about anything that happens on a smaller scale and we assume that no smaller scale processes are important to the problem we wish to address. (The assumption about "sub-grid" processes is sometimes a poor one, in which case we might attempt to parameterize them, as we did for radiative transfer in the energy balance model.)

With the assumption of a locally continuous field, we can rewrite equation (A.1) as a (local) partial differential equation. We begin by applying the (very handy) divergence theorem which, for a vector-valued quantity, states:

$$\int_S \mathbf{F}\cdot n \, \mathrm{d}S = \int_V \nabla\cdot\mathbf{F} \, \mathrm{d}V \quad \text{(A.7)}$$

Using index notation, equation (A.7) is written:

$$\int_S \mathbf{F}_j \, n_j \, \mathrm{d}S = \int_V \frac{\partial \mathbf{F}_j}{\partial x_j} \, \mathrm{d}V \quad \text{(A.8)}$$

Using equation (A.7), the surface integrals in equation (A.1) may be replaced:

$$-\int_S \mathbf{F}\cdot \mathrm{d}S \ - \ \int_S \phi \mathbf{V}\cdot \mathrm{d}S \ = \ -\int_V \nabla\cdot (\mathbf{F} \ + \ \phi \ \mathbf{V})\, \mathrm{d}V \tag{A.9}$$

In our Eulerian reference frame it must be the case that

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_V \phi \ \mathrm{d}V \ = \ \int_V \frac{\partial \phi}{\partial t} \ \mathrm{d}V \tag{A.10}$$

Substituting (A.9) and (A.10) into (A.1), we have

$$\int_V \left\{ \frac{\partial \phi}{\partial t} \ + \ \nabla\cdot (\mathbf{F} \ + \ \phi \mathbf{V}) \ - \ H \right\} \mathrm{d}V \ = \ 0 \tag{A.11}$$

Because V is an arbitrary volume, equation (A.11) can only be true if the term in brackets is zero for our little subdivided volumes (note that this "*zero everywhere*" does not necessarily apply at scales smaller than the limiting resolution discussed earlier). At scales we care about, we can write:

$$\frac{\partial \phi}{\partial t} \ + \ \nabla\cdot (\mathbf{F} \ + \ \phi \mathbf{V}) \ - \ H \ = \ 0 \tag{A.12}$$

This is the general form for all conservation laws in continuum mechanics.