CHAPTER 13

PARTIAL DIFFERENTIAL EQUATIONS I: PARABOLIC EQUATIONS

In Chapters 5-7, we determined functions of a single variable, such as velocity as a function of time, with ordinary differential equations. However, many physical quantities depend on more than one variables. For example, the particle density in the three-dimensional space depends on three coordinates x, y, and z. Electric field $\mathbf{E}(x,t)$ is another example, which depend on space-time coordinates x and t, . Equations which determine functions of multi-dimensional variables are known as partial differential equation (PDE). Perhaps, you already encounter such equations in other physics courses. Diffusion equation, Maxwell equations, and Schrödinger equation are all PDE.

PDE is both mathematically and computationally more challenging than ODE. One numerical method that works well for one type of PDE may fail for another type of PDE. From mathematical point of view, there are three different types of PDE for a second-order PDE with two-variable function $F(x,y)$. Its general form can be written as

$$a\frac{\partial^2 F}{\partial x^2} + b\frac{\partial^2 F}{\partial x \partial y} + c\frac{\partial^2 F}{\partial y^2} + d\frac{\partial F}{\partial x} + e\frac{\partial F}{\partial y} + fF + g = 0 \tag{13.1}$$

where coefficients a through g are constant. The variables x and y are not necessarily indicating spacial coordinates. One of them can be time. When $b^2 - 4ac = 0$, the PDE is said to be *parabolic*. Similarly the PDE is *hyperbolic* for $b^2 - 4ac > 0$, and *elliptic* for $b^2 - 4ac < 0$. Various numerical methods have been developed but they are suitable usually only for one type of PDE and unfortunately there is no single method that works for all three types.

Parabolic equations popular in physics are heat equation for temperature $T(x,t)$

$$\frac{\partial}{\partial t}T(x,t) = \kappa\frac{\partial^2}{\partial x^2}T(x,t) \tag{13.2}$$

and diffusion equation for particle density $\rho(x,t)$

$$\frac{\partial}{\partial t}\rho(x,t) = D\frac{\partial^2}{\partial x^2}\rho(x,t). \tag{13.3}$$

The thermal diffusion coefficient κ and particle diffusion constant D are both positive. Clearly these two equations are mathematically identical. Letting $y = t$ and $b = c = d = f = g = 0$ in Eq. (13.1), we obtain these two equations. Since $b^2 - 4ac = 0$, they are parabolic.

Schrödinger equation

$$i\hbar\frac{\partial}{\partial t}\psi(x,t) = -\frac{\hbar^2}{2m}\frac{\partial^2}{\partial x^2}\psi(x,t) + V(x)\psi(x,t) \tag{13.4}$$

is slightly different from the previous two equations ($f \neq 0$ and e is pure imaginary) but it is another example of parabolic PDE.

When $ac < 0$ and all other coefficients vanish, Eq. (13.1) becomes wave equation

$$\frac{\partial^2}{\partial t^2}\phi(x,t) = v^2\frac{\partial^2}{\partial x^2}\phi(x,t) \tag{13.5}$$

where v is the velocity of wave. In this expression $a = v^2$ and $c = -1$ and thus it is an example of hyperbolic equation.

If $a = c = 1$ (thus $ac > 0$), and all other coefficients vanish, Eq. (13.1) leads to Laplace's equation

$$\frac{\partial^2}{\partial x^2}\phi(x,y) + \frac{\partial^2}{\partial y^2}\phi(x,y) = 0 \tag{13.6}$$

which is an example of elliptic PDE. The Laplace's equation is one of the most important equations in physics and appears in many fields of physics, including, electromagnetism, fluid dynamics, thermodynamics, ... When an inhomogeneous term is added to the Laplace equation, we have Poisson's equation

$$\frac{\partial^2}{\partial x^2}\phi(x,y) + \frac{\partial^2}{\partial y^2}\phi(x,y) = -\frac{1}{\epsilon_0}\rho(x,y) \tag{13.7}$$

where $\phi(x,y)$ and $\rho(x,y)$ are the electrostatic potential and the charge density, respectively. This equation is also a family of elliptic PDE.

In the present chapter we focus on parabolic equations such as diffusion/heat equations and Sch$\acute{'}$odinfer equations. In the next chapter, the wave equation is discussed as an example of hyperbolic equation. The elliptic equation is investigated in the following chapter using Laplace's/Poisson's equations as example.

13.1 Diffusion Equation

To begin with, we look for a numerical method for a simple diffusion equation (**??**). While the development of numerical algorithms is purely mathematical procedure, actually consideration of physical processes described by the equation helps to find a good numerical approach. Let us consider a free diffusion of a particle. How fast does the particle diffuses from x_0 to another position x_1? The mean square displacement $\left\langle (x_1 - x_0)^2 \right\rangle$ is known to be proportional to time, or more precisely $\left\langle (x_1 - x_0)^2 \right\rangle = 2Dt$ where D is the diffusion constant. This means that the typical time to travel over the distance L is given by

$$\tau \approx \frac{L^2}{2D}. \tag{13.8}$$

The important thing is that time scale of the process τ is related to the spacial scale L. Any numerical method must be consistent with this physical condition.

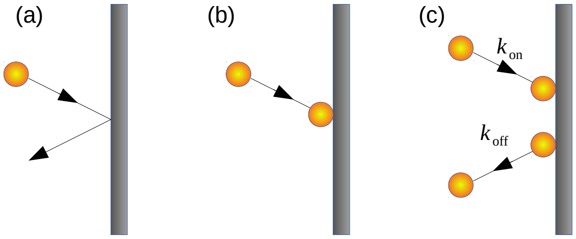


Figure 13.1: Three different types of boundary conditions for diffusion equations. (a) The particle is reflected by the wall [Neumann boundary]. (b) The particle is perfectly absorbed on the wall [Dirichlet boundary]. (c) Some particles are reflected and others absorbed on the wall with a transition rate k_{on}. The particles on the wall can desorb with a transition rate k_{off}. This situation can be dealt with the Robin boundary condition.

Next we derive the diffusion equation. The Fick's law tells that the flux of the particles is given by

$$j(x,t) = -D\frac{\partial}{\partial x}\rho(x,t) \tag{13.9}$$

Substituting this flux into the continuity equation

$$\frac{\partial}{\partial t}\rho(x,t) + \frac{\partial}{\partial x}j(x,t) = 0 \tag{13.10}$$

we obtain the diffusion equation (**??**). The Fick's law (13.9) is essential when we construct boundary condition.

13.2 Boundary Conditions

The parabolic PDEs common in physics has the first order derivative with respect to time. Therefore, we need only one boundary condition for time (initial condition)

$$f(x,t_0) = g(x) \tag{13.11}$$

where t_0 is the starting time. The initial function $g(x)$ must satisfy the boundary condition for x which we discuss next.

The derivative with respect to the spacial coordinates is second order and thus we need two boundary conditions. What happens on the boundary is not determined by the PDE itself. Separate physical processes on the boundary determine the boundary conditions. There are many different types of boundary conditions depending on the physical situations. Among them, four types of boundary conditions are common in physics. We use the diffusion equation (13.3) as example.

When particles diffusing in a container reach the wall, four different kinds of boundary conditions are commonly used in physics . In one case, particles which hit the wall are perfectly reflected back from the wall. The particle flux going to the wall and the flux coming from the wall must be canceled out. Hence, the net flux at the boundary must vanish.* When the particles are reflected back at $x=a$, $j(a,t)=0$. Based

*Vanishing flux does not mean that nothing is moving. It simply means that the number of particles moving to the left and to the right is equal on average.

on the Fick's law (13.9), this condition implies that

$$j(x=a,t) = \frac{\partial}{\partial x}\rho(x,t)\bigg|_{x=a} = 0 \tag{13.12}$$

which is the *reflective* boundary condition (also known as no-flux condition). In mathematics, the boundary condition given by the derivative is known as Neumann boundary condition. Note that the number of particles conserves in this boundary condition.

In another scenario, the particles are absorbed on the wall and do not come back to the system. Since the particles disappear at the boundary, the boundary condition is simply

$$\rho(a,t) = 0. \tag{13.13}$$

This is the *absorbing* boundary condition. In mathematics, this is known as Dirichlet boundary condition. The number of particles in the system decreases with this boundary condition.

The third possibility corresponds to the situation between the reflective and absorbing boundary conditions. Th particles is partly absorbed with a certain rate k_{on}. The particles absorbed on the wall desorbe from the wall with a different rate k_{off}. The particle flux at the boundary is now defined by

$$D\,\frac{\partial}{\partial x}\rho(x,t)\bigg|_{x=a} = k_{\text{on}}\,\rho(a,t) - k_{\text{off}}\,\sigma(t) \tag{13.14}$$

where $\sigma(t)$ is the number density of the particle on the wall and it satisfies the following ODE

$$\frac{\mathrm{d}}{\mathrm{d}t}\sigma(t) = k_{\text{on}}\,\rho(a,t) - k_{\text{off}}\,\sigma(t). \tag{13.15}$$

In mathematics, the boundary condition given by

$$\alpha f(a,t) + \beta\,\frac{\partial}{\partial x}f(x,t)\bigg|_{a} = g(t) \tag{13.16}$$

is known as the Robin boundary condition.

We need the boundary condition at two different boundaries. We don't have to use the same type of boundary conditions. We can use one of the three types at one boundary and another type at the other boundary. Sometime, this type of setting is called mixed boundary value problem.

Finally, we consider a system has ”no boundary”. Consider a field $F(\rho,\theta)$ on two-dimensional space expressed with polar coordinates; i.e., radial coordinate ρ and angular coordinate θ. The radial coordinate ρ is defined in $[0,\infty)$ and thus regular boundary conditions are usually specified at $\rho=0$ and ∞. However, the angular coordinate θ defined in $[0,2\pi)$ does not have a boundary since $\theta=0$ and $\theta=2\pi$ correspond to the same point on the space. Hence, we have $F(\rho,0)=F(\rho,2\pi)$, $\forall\rho$. Instead of limiting θ in $[0,2\pi)$, we often use $\theta\in\mathbb{R}$ and require

$$F(\rho,\theta+2\pi) = F(\rho,\theta), \quad \forall\theta\in\mathbb{R}.$$

Then, F is a periodic function with respect to θ. This is a kind of ”boundary condition” called periodic boundary condition.

There are other cases where the periodic boundary is used. For example, consider an infinitely extended system filled with infinite number of particles. Since computers cannot deal with infinity, we limit the size of the system. For example, we consider only the regions between $x=-L/2$ and $x=L/2$. However, there is no wall at the boundary. A common trick is to use a periodic boundary condition. We assume that the system of size L repeats infinitely many times by using the condition

$$F(x+L) = F(x), \quad \forall x \tag{13.17}$$

which is equivalent to the periodic boundary condition. We can consider a ring-like space. For two-dimensional cases, we can have peridic boundary in both dimensions,

$$F(x+L,y)=F(x,y) \quad \text{and} \quad F(x,y+M)=F(x,y) \quad \forall x,y$$

where L and M are period in each direction. The space is a torus for this case.

13.3 Forward Time Centered Space method

Now, we solve simple diffusion equation (13.3) numerically. Other parabolic PDE can be solved in the same way. Consider free diffusion of particles in a one-dimensional box of size L. The position coordinate x covers the space from $x=0$ to $X=L$. WThe particle density $\rho(x,t)$ evolves in time from $t=0$. We discretize space and time as $x_i = i\Delta x$, $i=0,\cdots,N$ and $t_j = j\Delta t$. The initial time is $t_0=0$ and the boundary for the space coordinate are $x_0=0$ and $x_N=L$. The function value at time t_j and position x_i are stored in an array as

$$\rho_i^j \equiv \rho(x_i,t_j). \tag{13.18}$$

Using finite difference methods (see Chapter 2),

$$\frac{\partial}{\partial t}\rho(x,t) \approx \frac{\rho_i^{j+1}-\rho_i^j}{\Delta t} \tag{13.19a}$$

$$\frac{\partial^2}{\partial x^2}\rho(x,t) \approx \frac{\rho_{i+1}^j+\rho_{i-1}^j-2\rho_i^j}{\Delta x^2} \tag{13.19b}$$

Eq (13.3) becomes

$$\rho_i^{j+1} \approx \rho_i^j + \frac{D\Delta t}{\Delta x^2}\left(\rho_{i+1}^j+\rho_{i-1}^j-2\rho_i^j\right), \quad i=1,\cdots,N-1 \tag{13.20}$$

If we knows the density at time t_j, the density at the next time t_{j+1} is obtained by this recursive equation. Note that $i=0$ and $i=N$ are not included in the evolution since they are fixed by boundary conditions. This is one of the simplest method, known as *forward time centered space* (FTCS) method.

Next we set up the boundary conditions. The initial condition is given by

$$\rho_i^0 = g(x_i). \tag{13.21}$$

Using the Euler method, the reflective boundary condition (13.12) is given by

$$\frac{\rho_1^j-\rho_0^j}{\Delta x}=0 \quad \rightarrow \quad \rho_0^j=\rho_1^j \tag{13.22}$$

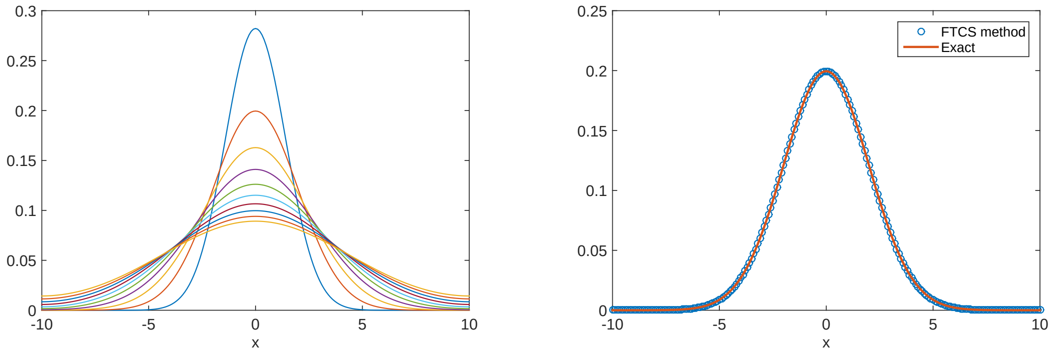
and similarly at the other boundary

$$\frac{\rho_N^j-\rho_{N-1}^j}{\Delta x}=0 \quad \rightarrow \quad \rho_N^j=\rho_{N-1}^j \tag{13.23}$$

Substituting the boundary conditions to Eq. (13.20), the function values at adjacent to the boundary evolves by

$$\rho_1^{j+1} = \rho_1^j + \frac{D\Delta t}{\Delta x^2}\left(\rho_2^j-\rho_1^j\right) \tag{13.24a}$$

$$\rho_{N-1}^{j+1} = \rho_{N-1}^j + \frac{D\Delta t}{\Delta x^2}\left(\rho_{N-2}^j-\rho_{N-1}^j\right) \tag{13.24b}$$



(a) Time evolution of the probability density.

(b) Comparison with the exact distribution.

Figure 13.2: A solution to the diffusion equation with the Neumann boundary at $x = \pm 10$. The left panel shows the time-evolution of the density at from $t = 10$ to $t = 100$, starting with an initial condition, $\rho(x, 0) = \delta(x)$. The right panel shows the density at $t = 20$, which is in good agreement with the exact solution.

The Dirichlet boundary is simply

$$\rho_0^j = 0 \qquad \rho_N^j = 0 \tag{13.25}$$

The evolution of the function values at adjacent to the boundary is explicitly given by

$$\rho_1^{j+1} = \rho_1^j + \frac{D\Delta t}{\Delta x^2}\left(\rho_2^j - 2\rho_1^j\right) \tag{13.26a}$$

$$\rho_{N-1}^{j+1} = \rho_{N-1}^j + \frac{D\Delta t}{\Delta x^2}\left(\rho_{N-2}^j - 2\rho_{N-1}^j\right) \tag{13.26b}$$

The finite difference method is accurate when δt and δx are sufficiently small. We tend to believe that any smaller values generates more accurate results. However, we cannot chose Δx and Δt independently. Numerically, it is clear that the factor $\dfrac{D\Delta t}{\Delta x^2}$ in Eq. (13.20) must be smaller than 1. Actually, this limitation is also clear from physics. Recall that the mean square displacement of the Brownian particles is proportional to time, or more precisely $\langle x^2 \rangle = 2Dt$, which suggest that the time a particle travels from x_i won't reach x_{i+1} is about $\dfrac{\Delta x^2}{2D}$ on average. The time step must be much smaller than that. Therefore, we require

$$\Delta t \ll \frac{\Delta x^2}{2D}. \tag{13.27}$$

EXAMPLE 13.1 Free Diffusion

Initially a particle is located at $x = 0$ and it freely diffuses at a diffusion rate D. We want to know how the probability distribution $p(x, t)$ changes in time. If N non-interacting particles diffuse, the particle density is given by $\rho(x, t) = Np(x, t)$. Dividing Eq. (13.3), it is easy to find that the probability density satisfies the same diffusion equation (13.3). The difference is only their normalization, $\displaystyle\int_{-\infty}^{\infty} = \rho(x, t)\, dx = N$ for particle density and $\int_{-\infty}^{\infty} p(x,t)\,\mathrm{d}x = 1$ for the probability density. We assume that the space is infinitely large and the particle diffuses freely for ever. Then, the boundary condition is $\lim_{|x|\to\infty} p(x,t) = 0$.

An analytic solution is well-known:

$$\rho(x,t) = \frac{1}{2\pi}\frac{1}{\sqrt{2Dt}}\mathrm{e}^{-x^2/4Dt}. \tag{13.28}$$

prog:diffusion First we define the computational boundary since the infinitely large space cannot be used in the numerical method. We replace $\pm\infty$ with $x = \pm L$ as usual and use the Neumann boundary condition $\left.\frac{\partial}{\partial x}\rho(x,t)\right|_{\pm L} = 0$, which implies that the particle will be reflected back if it ever reaches the boundary. The initial condition is mathematically $\rho(x,0) = \delta(x)$ which will be replaced with

$$p_i^0 = \begin{cases} \frac{1}{\Delta x} & x_i = 0 \\ 0 & \text{otherwise} \end{cases},$$

which satisfies the normalization $\int_{-\infty}^{\infty} p(x,t)\,\mathrm{d}x = \sum_i p_i \Delta x = 1$.

Program 13.1 solves this problem and the results are plotted in Fig. 13.2. The agreement between the numerical result and the exact solution is quite good at $t = 0$ (right panel). However, as time increases, the particle hits the artificial boundary at $L = 10$ where the probability does not vanish. L must be increased to see the correct tail.

13.4 Runge-Kutta time evolution

The forward time finite difference method used in the FTCS scheme is equivalent to the Euler method (see Section 4.2.1), which is not accurate. We can improve the accuracy with respect to time evolution using the 2nd order Runge-Kutta method. For simple diffusion equation, we first use the Euler scheme with a half time step

$$\rho_i^{j+\frac{1}{2}} = \rho_i^j + \frac{D\Delta t}{2\Delta x^2}\left(\rho_{i+1}^j + \rho_{i-1}^j - 2\rho_i^j\right). \tag{13.29}$$

Then, the Runge-Kutta step is given by

$$\rho_i^{j+1} = \rho_i^j + \frac{D\Delta t}{\Delta x^2}\left(\rho_i^{j+\frac{1}{2}} + \rho_{i-1}^{j+\frac{1}{2}} - 2\rho_i^{j+\frac{1}{2}}\right). \tag{13.30}$$

This method is more accurate than the FTCS method.

13.5 Higher spatial dimensions

In the above, the particles diffuse along a line and thus the diffusion equation (13.3) has only two variables, t and x. For a higher dimension, the diffusion equation becomes

$$\frac{\partial}{\partial t}\rho(t,\mathbf{r}) = \boldsymbol{\nabla}\cdot\rho(t,\mathbf{r}) \tag{13.31}$$

The extension of Eq. (13.20) to a higher dimensional space is straight forward. For a two-dimensional space, we discretize the space by $x_i = i\Delta x$ and $y_j = j\Delta y$. The density is denoted as $\rho_{i,j}^k \equiv \rho(t_k, x_i, y_j)$.

Using the3-point finite difference approximation to the second order derivative for each direction, the discrete version of Eq. (13.31) is given by

$$\rho_{i,j}^{k+1} \approx \rho_{i,j}^{k} + \frac{D\Delta t}{\Delta x^2}\left(\rho_{i+1,j}^{k} + \rho_{i-1,j}^{k} - 2\rho_{i,j}^{k}\right) + \frac{D\Delta t}{\Delta y^2}\left(\rho_{i,j+1}^{k} + \rho_{i,j-1}^{k} - 2\rho_{i,j}^{k}\right) \tag{13.32}$$

13.6 Schrödinger Equations

The Schrödinger equation for a particle of mass m in a one-dimensional space is given by

$$i\hbar\frac{\partial\psi(x,t)}{\partial t} = H\psi(x,t) \tag{13.33}$$

where a typical form of the Hamiltonian is the sum of kinetic and potential energy operators:

$$H = -\frac{\hbar^2}{2m}\frac{\partial^2}{\partial x^2} + V(x) \tag{13.34}$$

A major difference from the diffusion equation is that the solution to this equation is inherently complex. It is possible to write a set of partial differential equations separately for real and complex parts.

$$\hbar\frac{\partial}{\partial t}u(x,y) \quad = \quad Hw(x,t) \tag{13.35a}$$

$$\hbar\frac{\partial}{\partial t}w(x,t) \quad = \quad -Hu(x,t) \tag{13.35b}$$

where $u(x,t)$ and $w(x,t)$ are real and imaginary part of $\psi(x,t)$, respectively. These partial differential equations are coupled and must be solved simultaneously. By deferentiating both side with respect to time, we can make two independent PDEs:

$$\frac{\partial^2}{\partial t^2}u(x,t) \quad = \quad -\left(\frac{H}{\hbar}\right)^2 u(x,t) \tag{13.36a}$$

$$\frac{\partial^2}{\partial t^2}w(x,t) \quad = \quad -\left(\frac{H}{\hbar}\right)^2 w(x,t) \tag{13.36b}$$

These are more complicated than the original equation since H^2 involves fourth order derivative. We will look for other methods.

The Schrödinger equation is linear and its solution can be formally written with a time evolution operator as

$$\psi(x,t) = \mathrm{e}^{-iH(t-t_0)/\hbar}\psi(x,t_0). \tag{13.37}$$

However, since an operator H is in the exponential function, the numerical evaluation of this solution is still difficult. Introducing discrete time $t_n = t_0 + n\Delta t, n = 0, \cdots, N$ where Δt is a small time step, the time evolution can be expressed as a product of step operator

$$\psi(x,t_N) = \mathrm{e}^{-iHN\Delta t/\hbar}\,\psi(x,t_0) = \left(\mathrm{e}^{-iH\Delta t/\hbar}\right)^N\,\psi(x,t_0) \tag{13.38}$$

and a single time step as

$$\psi(x,t_n+\Delta t) = \mathrm{e}^{-iH\Delta t/\hbar}\,\psi(x,t_n). \tag{13.39}$$

To evaluate the right hand side of this equation, we may expand the exponential function up to the order of Δt. Then, the single step is

$$\psi(x, t_n + \Delta t) = \left(1 - \frac{i}{\hbar} H \Delta\right) \psi(x, t_n) \tag{13.40}$$

which is equivalent to the Euler method for ODEs. This approach is not only inaccurate (order of Δt) but also does not conserve the norm of the wavefunction.

There are several numerical algorithms specifically suitable for the Schrödinger equation, which conserves the norm and correct up to the order of Δt^2 (higher than the Euler method).

13.6.1 Crank-Nicolson method

Consider a half forward step from t_n and a half backward step from t_{n+1},

$$\psi(x, t_n + \Delta t/2) = \mathrm{e}^{-iH\Delta t/2\hbar}\, \psi(x, t_n) \tag{13.41}$$

$$\psi(x, t_n + \Delta t/2) = \mathrm{e}^{iH\Delta t/2\hbar}\, \psi(x, t_n + \Delta t) \tag{13.42}$$

and thus

$$\mathrm{e}^{iH\Delta t/2\hbar}\, \psi(x, t_{n+1}) = \mathrm{e}^{-iH\Delta t/2\hbar}\, \psi(x, t_n) \tag{13.43}$$

which is still exact. Now, we expand the exponential function up to the order of Δt and obtain

$$\left(1 + \frac{i}{2\hbar} H \Delta t\right) \psi(x, t_n + \Delta t) = \left(1 - \frac{i}{2\hbar} H \Delta t\right) \psi(x, t_n). \tag{13.44}$$

Unlike the previous expansion in Eq. (13.40), this expression is correct upto the order of Δt^2. Noting that $\|1 + \frac{i}{2\hbar} H \Delta t\| = \|1 - \frac{i}{2\hbar} H \Delta t\|$, the norm conserves. Rearranging the equation

$$\frac{1}{2}\left(1 + \frac{i}{2\hbar} H \Delta t\right) \left[\psi(x, t_n + \Delta t) + \psi(x, t_n)\right] = \psi(x, t_n). \tag{13.45}$$

which is a linear equation

$$A \chi = \psi(x, t_n) \tag{13.46}$$

where $A = \frac{1}{2}\left(1 + \frac{i}{2\hbar} H \Delta t\right)$ and $\chi = \psi(x, t_{n+1}) + \psi(x, t_n)$. We solve this equation for χ and the solution is $\chi = A^{-1} \psi(x, t_n)$. Once χ is obtained, the wavefunction at next time is given by

$$\psi(x, t_n + \Delta t) = \chi - \psi(x, t_n) \tag{13.47}$$

To solve Eq. (13.46), as usual we discretize the space by $x_j = x_0 + jh, j = 0, \cdots, M$. We discussed a discrete version of the Hamiltonian in CHap 6 which is given as a matrix

$$H \doteq \begin{bmatrix}
\frac{\hbar^2}{mh^2} + \tilde{U}_1 & -\frac{\hbar^2}{2mh^2} & 0 & 0 & 0 & 0 & \cdots \\
-\frac{\hbar^2}{2mh^2} & \frac{\hbar^2}{mh^2} + U_2 & -\frac{\hbar^2}{2mh^2} & 0 & 0 & 0 & \cdots \\
0 & -\frac{\hbar^2}{2mh^2} & \frac{\hbar^2}{mh^2} + U_3 & -\frac{\hbar^2}{2mh^2} & 0 & 0 & \cdots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \cdots \\
0 & 0 & 0 & \cdots & -\frac{\hbar^2}{2mh^2} & \frac{\hbar^2}{mh^2} + U_{M-1} & -\frac{\hbar^2}{2mh^2} \\
0 & 0 & 0 & 0 & \cdots & -\frac{\hbar^2}{2mh^2} & \frac{\hbar^2}{mh^2} + U_M
\end{bmatrix} \tag{13.48}$$

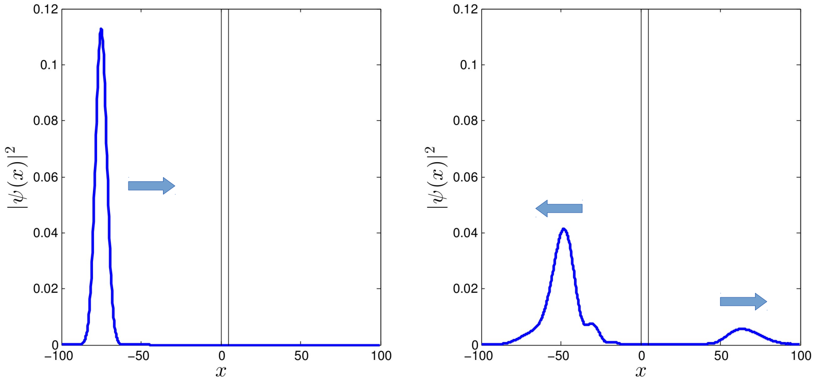


Figure 13.3: Quantum tunneling through the square potential barrier. The left panel shows the probability density of the initial wave packet moving toward the potential barrier. The right panel shows the probability density after the collision with the potential barrier. A broad peak in the right side of the potential barrier indicates that the fraction of the packet tunnels through the barrier.

and the matrix A is a tridiagonal matrix with the matrix elements

$$\begin{aligned} A_{i\,i} &= \frac{1}{2}\left[1 + \frac{i\Delta t}{2\hbar}\left(\frac{\hbar^2}{2m}\cdot\frac{2}{h^2} + U_i\right)\right] && (13.49\text{a}) \\ A_{i\,i+1} &= -\frac{i\Delta t}{4\hbar}\cdot\frac{\hbar^2}{2m}\cdot\frac{1}{h^2} && (13.49\text{b}) \\ A_{i\,i-1} &= A_{i\,i+1} && (13.49\text{c}) \end{aligned}$$

where $U_i = U(x_i)$. Now, $\psi(x, t_n)$ is a column vector with the component $\psi_i(t_n) = \psi(x_i, t_n)$. Then, we can solve Eq. (13.46) by the Gaussian elimination/backsubstitution method or other methods discussed in Chap 7.

13.7 Applications in Physics

13.7.1 Quantum Tunneling

A quantum particle can tunnel through a potential barrier. Consider a quantum particle of mass m colliding with a square potential barrier

$$U(x) = \begin{cases} 0 & x < 0 \\ U_0 & 0 < x < L \\ 0 & L < x \end{cases} \qquad (13.50)$$

Th corresponding Schrödinger equation is

$$i\hbar\frac{\partial}{\partial t}\psi(x,t) = -\frac{\hbar^2}{2m}\frac{\partial^2}{\partial x^2}\psi(x,t) + U(x)\psi(x,t). \qquad (13.51)$$

Before writing a program, we will simplify the mathematical expression by introducing normalizing energy, time and wave number as $\tilde{E} = E/U_0$, $\tilde{t} = t/(\hbar/U_0)$ and $\tilde{k} = k/\sqrt{2mU_0/\hbar^2}$. Accordingly, distance is measured in $\tilde{x} = x/\sqrt{\hbar^2/2mU_0}$. For simplicity, we omit the tilde in the normalized expression

$$i\frac{\partial}{\partial t}\psi(x,t) = -\frac{\partial^2}{\partial x^2}\psi(x,t) + U(x)\psi(x,t) \tag{13.52}$$

where the normalized potential has the height 1 and width L measured in the unit of $\sqrt{\hbar^2/2mU_0}$.

The initial wavefunction is a Gaussian packet

$$\psi(x,0) = \sqrt{\frac{a}{\sqrt{\pi}}}\mathrm{e}^{-(x-x_0)^2/2a^2}\mathrm{e}^{ikx} \tag{13.53}$$

where x_0 and a are the initial position and width of the packet. The wave number k is determined by the speed v of the packet as $k = mv/\hbar$. The transmission probability is determined by

$$T = \lim_{t\to\infty}\int_L^\infty |\psi(x,t)|^2 \mathrm{d}x \tag{13.54}$$

and the reflection probability by

$$R = \lim_{t\to\infty}\int_{-\infty}^0 |\psi(x,t)|^2 \mathrm{d}x \tag{13.55}$$

Program 13.2 computes the time evolution of wavefunction using the Crank-Nicolson method and computes the transmission/reflection probabilities. Figure 13.3 shows the initial and final probability densities. A smaller peak is seen in the right side of the potential barrier, indicating that a fraction o the packet tunnels through the potential barrier. The transmission probability is 0.15.

13.7.2 Pattern Formation

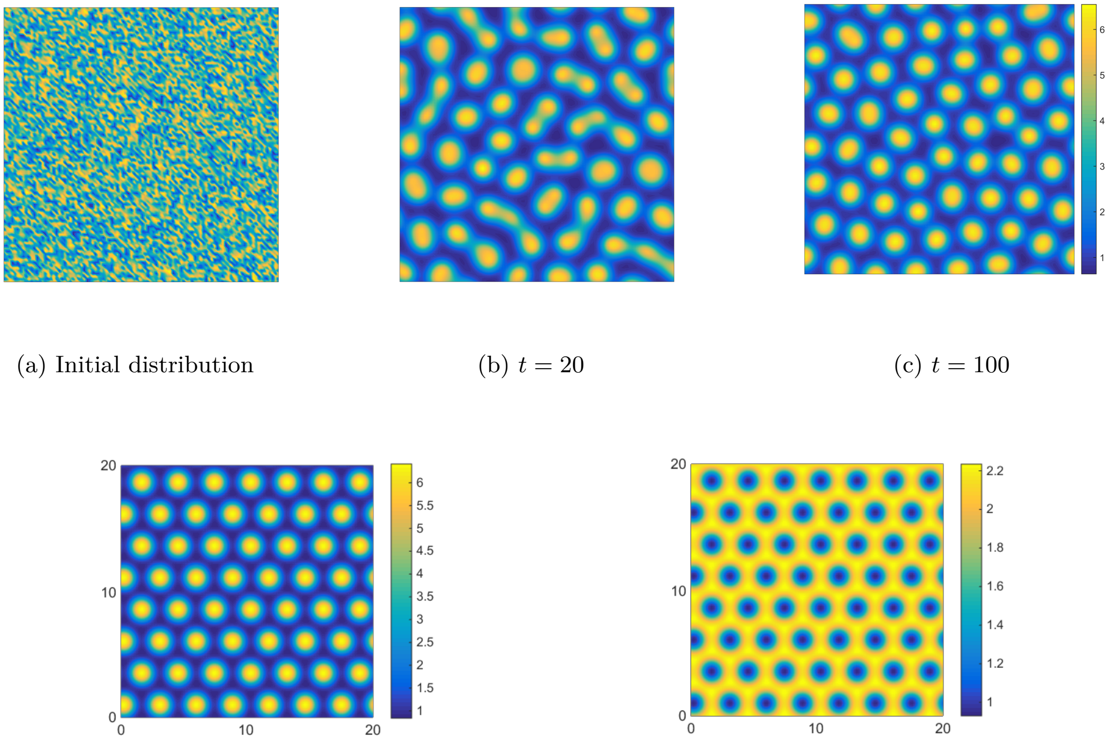
The first chemical model to show oscillations and traveling waves was proposed by Prigogine and Lefever[1] in 1968. The model is called the "Brusselator" because it was discovered in the city of Brussels. The Brusselator system is the following sequence of reaction:

$$A \longrightarrow X \tag{13.56a}$$
$$B + X \longrightarrow Y + D \tag{13.56b}$$
$$2X + Y \longrightarrow 3X \tag{13.56c}$$
$$X \longrightarrow E \tag{13.56d}$$

where the species A and B are sources whose concentration are kept constant, and D and E are products which are extracted from the system at a constant rate. The species X and Y are intermediate products. It is important to note that both X and Y are produced and consumed during the sequence of reactions in such a way that X produces Y and Y produces X.

When the reaction takes place in a well stirred container, the concentration of chemicals are uniform and does not depend on the position. We have studied such a case in Section 4.4.1. If the system is no stirred, the chemicals are not well mixed and the concentration becomes position-dependent. The diffusion becomes the main mechanism of the mixing of the chemicals. Then, the dynamics of the reaction is described by a pair of reaction-diffusion equations:

$$\frac{\partial}{\partial t}u(\mathbf{r},t) = D_u\nabla^2 u(\mathbf{r},t) + a - (b+1)u(\mathbf{r},t) + u^2(\mathbf{r},t)\,w(\mathbf{r},t) \tag{13.57a}$$
$$\frac{\partial}{\partial t}w(\mathbf{r},t) = D_w\nabla^2 w(\mathbf{r},t) + b\,u(\mathbf{r},t) - u^2(\mathbf{r},t)\,w(\mathbf{r},t) \tag{13.57b}$$



(a) Initial distribution

(b) $t = 20$

(c) $t = 100$

(d) Steady state concentration of X

(e) Steady state concentration of Y

Figure 13.4: Time evolution of pattern formation. Initially, the chemicals are randomly distributed. As time goes, a pattern begins to appear. By $t = 100$, a two dimensional crystal like structure is formed. However, the pattern does not have a precise periodicity or symmetry yet. At $t = 2000$, the system reaches a steady state. The spot size is now identical and they form a hexagonal close-packing structure. Parameter values are $a = 2.5$, $b = 5.0$, $D_u = 0.2$, and $D_w = 1.6$. Periodic boundary condition with $L = 20$ is used. The discretization parameters are $h = 1$, and $\Delta t = 0.125 \times 10^{-2}$.

where $u(\mathbf{r},t)$ and $w(\mathbf{r},t)$ are the concentration of chemicals X and Y, and D_u and D_w are their diffusion constants, respectively. The parameters a and b remain constant both in space and time as before (see Section 4.4.1).

The reaction-diffusion equations (13.57) has many different types of solution depending parameter values, initial conditions and boundary conditions, for examples, pattern formation, traveling wave, and spiral waves.

Equations (13.57) are essentially the diffusion equation with additional terms. Discretizing he time and space, we denote the two function as $u^k_{i,j} = u(t_k, x_i, y_j)$ and $w^k_{i,j} = w(t_k, x_i, y_j)$. Similarly to the 2-dimensional diffusion equation (**??**), the discrete version of the diffusion-reaction equations are

$$\begin{aligned} u^{k+1}_{i,j} &\approx \rho^k_{i,j} + \frac{D_u \Delta t}{\Delta x^2}\left(u^k_{i+1,j} + u^k_{i-1,j} - 2u^k_{i,j}\right) + \frac{D_u \Delta t}{\Delta y^2}\left(u^k_{i,j+1} + u^k_{i,j-1} - 2u^k_{i,j}\right) \\ &\quad + a - (b+1)u^k_{i,j} + (u^k_{i,j})^2 w^k_{i,j} \qquad (13.58a) \\ w^{k+1}_{i,j} &\approx w^k_{i,j} + \frac{D_w \Delta t}{\Delta x^2}\left(w^k_{i+1,j} + w^k_{i-1,j} - 2w^k_{i,j}\right) + \frac{D_u \Delta t}{\Delta y^2}\left(w^k_{i,j+1} + w^k_{i,j-1} - 2w^k_{i,j}\right) \\ &\quad + bu^k_{i,j} - (u^k_{i,j})^2 w^k_{i,j} \qquad (13.58b) \end{aligned}$$

Program 13.3 implements Eq. (13.7.2), with periodic boundary conditions in both directions.

We assume that the space is isotropic ($h = \Delta x = \Delta y$). Initially, both u and w take independent random values between 0 and 1 at each point. Figure 13.4 shows the time evolution of the concentration of X. Initially (Fig. 13.4a) no simple pattern is seen but by time $t = 20$ (Fig. 13.4b) the high concentration regions is formed. At $t = 100$ (Fig. 13.4c), many circular spots with nearly the same radius are distributed with nearly equal distance between them. The spots size is not exactly the same and they are not exactly aligned, it is clear that the spots are not randomly placed. The reaction-diffusion system is organizing itself and forming a distinct order. Figures 13.4d and 13.4e show the final steady state. Each spot has the same size and they form hexagonal close packed structure like a two-dimensional crystal.

The patterns depend on the boundary conditions. In the present example, the close-packing is formed due to the periodical boundary condition. Other types of boundary conditions generates different patters. It is also known that the same reaction-diffusion equation can generate spiral waves with appropriate boundary conditions and initial conditions.

Program Lists

Program 13.1

```
%***************************************************************************
%*     Example 13.1                                                        *
%*     filename: ch13pr01.m                                                *
%*     program listing number: 13.1                                        *
%*                                                                         *
%*     This program solves a diffusion equation using the forward time    *
%*     centered space method.                                              *
%*                                                                         *
%*     Programed by Ryoichi Kawai for Computational Physics Course.        *
%*     Last modification:  02/15/2014.                                     *
%***************************************************************************
close all;
clear all;
% parameters
D=0.1; % diffusion constant
N=201; % number of grids
N1=(N-1)/2;
dx=0.1; % spacial step
R = 0.1; % D*dt/dx^2
dt=dx^2/D *R;  % time step

x=(-N1:N1)*dx; % spatial coordinates

M=10; % number of sample point.
tmax=10000; % total time steps
MS=tmax/M;

rho0=zeros(N,1);  % initial density profile
rho0(N1+1)=1.0/dx;

% allocate arrays
rho1=zeros(N,1);
rho=zeros(N,M);
t=zeros(M,1);

k=0;
for j=1:tmax
    rho1(1)=(1-R)*rho0(1)+R*rho0(2);  % left boundary
    rho1(N)=(1-R)*rho0(N)+R*rho0(N-1);  % right boundary
    for i=2:N-1
        rho1(i)=rho0(i)*(1-2*R)+R*(rho0(i+1)+rho0(i-1));
    end
    rho0=rho1;
    if mod(j,MS)==0  % record the results
        k=k+1;
        t(k)=j*dt;
        rho(:,k)=rho0(:);
    end
end

figure(1)
surf(t,(-N1:N1),rho)

figure(2)
for i=1:M
    plot(x,rho(:,i))
    hold on
end
```

```
hold off

figure(3)
f=1/sqrt(2*pi)*1/sqrt(2*D*t(2))*exp(-x.^2/(4*D*t(2))); % exact
plot(x,rho(:,2),'o',x,f)
legend('FTCS method','Exact')
```

Program 13.2

```
%**************************************************************************
%*     Example 13.2                                                       *
%*     filename: ch13pr02.m                                               *
%*     program listing number: 13.2                                       *
%*                                                                        *
%*     This program calculates quantum tunneling by solving a Shrodinger  *
%*     equation.  The Crank-Nicolson method is used.                      *
%*                                                                        *
%*     Programed by Ryoichi Kawai for Computational Physics Course.       *
%*     Last modification:  02/15/2014.                                    *
%**************************************************************************

clear all
close all

% system parameters
E=0.9;
k=sqrt(2*E);
a=5.0;
dt=0.1;

% control parameters
L=100;
h=0.05*min(a,2*pi/k);
N=2*round(L/h)+1;

% initial condition
x0=-L+5*a;
for j=1:N;
    x(j)=(j-(N+1)/2)*h;
    psi(j,1)=exp(-(x(j)-x0)^2/(2*a^2))*exp(i*k*x(j));
end
c=sum(abs(psi).^2)*h;
psi=psi/sqrt(c);

% construct matrix
A=zeros(N,N);
A(1,1)=complex(1,dt/h^2/2)/2;
A(1,2)=-i*dt/h^2/8;
for n=2:N-1
    A(n,n)=A(1,1);
    A(n,n-1)=A(1,2);
    A(n,n+1)=A(1,2);
end
A(N,N)=A(1,1);
A(N,N-1)=A(1,2);

% add potential barrier
V=complex(0,dt/4);
for n=(N+1)/2:(N+1)/2+5/h;
    A(n,n)=A(n,n)+V;
end
```

```
% solve Schrodinger equation.
for I=1:1000
    chi = A\psi;
    psi = chi - psi;
    rho = abs(psi).^2;
    p=plot(x,rho);
    set(p,'linewidth',2);
    hold on
    r=plot([0,0],[0,0.12],[5,5],[0,0.12],[0,5],[0.12,0.12]);
    set(r,'color','black');
    axis([-L L 0 0.12]);
    xlabel('$x$','interpreter','latex','fontsize',16)
ylabel('$|\psi(x)|^2$','interpreter','latex','fontsize',16)
hold off; drawnow;
end

% check the normalization
c=sum(abs(psi).^2)*h;
fprintf('Final Norm=%.6f\n',c)

% compute transmission/reflection probability
T=sum(abs(psi((N+1)/2+int32(5/h):N-1).^2))*h;
R=sum(abs(psi(1:(N-1)/2).^2))*h;
fprintf('Transmission Probability=%.6f\n',T)
fprintf('Reflection   Probability=%.6f\n',R)
```

Program 13.3

```
%**************************************************************************
%*     Example 13.5.2                                                     *
%*     filename: ch13pr03.m                                               *
%*     program listing number: 13.3                                       *
%*                                                                        *
%*     This program solves a coupled reaction-diffusion systems based on  *
%*     the Brusselator model. The parameters are chosen to form spots.    *
%*                                                                        *
%*     Programed by Ryoichi Kawai for Computational Physics Course.       *
%*     Last modification:  02/15/2014.                                    *
%**************************************************************************
clear all
close all

% system parameters
a=2.5; b=5; % parameters for spots
Du=0.2; % diffusion constant for u
Dw=1.6; % duiffusion constant for w
L=20.0; % the size of the system (periodic boundary condition.

% control parameters
NL=100; % number of grid points
dx=L/NL; % step length
Du=Du/dx^2;
Dw=Dw/dx^2;
T=100; % total time (takes a long time to reach the final patttern)
dt=0.1/max(Du,Dw); % time step
NT=int32(T/dt);

% initial condition
u0=rand(NL,NL);
w0=rand(NL,NL);
pcolor(u0); axis equal tight; shading interp; drawnow;
```

```
laplace_u=zeros(NL,NL);
laplace_w=zeros(NL,NL);

for k=1:NT
    t=(k-1)*dt;
    % Laplacian with periodic boundary
    laplace_u = circshift(u0,1,1)+circshift(u0,-1,1) ...
              + circshift(u0,1,2)+circshift(u0,-1,2) - 4*u0;
    laplace_w = circshift(w0,1,1)+circshift(w0,-1,1) ...
              + circshift(w0,1,2)+circshift(w0,-1,2) - 4*w0;
    % Euler step
    fu=a-(b+1)*u0+u0.^2.*w0+Du*laplace_u;
    fw=b*u0-u0.^2.*w0+Dw*laplace_w;
    u1=u0+fu*dt/2;
    w1=w0+fw*dt/2;

    % Laplacian at the mid time.
    laplace_u = circshift(u1,1,1)+circshift(u1,-1,1) ...
              + circshift(u1,1,2)+circshift(u1,-1,2) - 4*u1;
    laplace_w = circshift(w1,1,1)+circshift(w1,-1,1) ...
              + circshift(w1,1,2)+circshift(w1,-1,2) - 4*w1;

    % Runge-Kutta step
    fu=a-(b+1)*u1+u1.^2.*w1+Du*laplace_u;
    fw=b*u1-u1.^2.*w1+Dw*laplace_w;
    u0=u0+fu*dt;
    w0=w0+fw*dt;

    pcolor(u0); axis equal tight; shading interp; drawnow;
end
colorbar


figure(2)
pcolor(w0); axis equal tight; shading interp; drawnow;
colorbar
```

Bibliography

[1] I. Prigogine and R. Lefever. Symmetry breaking instabilities in dissipative systems. ii. *J. Chem. Phys.*, 48:1695, 1968.