1 General Comments

In addressing a new problem in fluid mechanics, one has the choice of many approaches, depending on the types and quality of the information needed, the accuracy required, the costs, the time available, and other related factors. If rough estimates are needed, then 'back-of-the-envelop' estimates can sometimes be made, with such approaches as using Bernoulli's equation, control-volume analysis, potential flow analysis, and other related methods. Other problems, for example determining the dynamic stability of a certain flow, might require somewhat sophisticated mathematical analysis coupled with numerical solutions of the resulting equations. When more detailed information on a particular flow is required, then either laboratory experiments or computer (numerical) simulations are usually employed.

For most applications in fluid mechanics, the equations needed to be solved in a numerical solution, the Navier-Stokes equations plus possibly thermo-chemical equations, are highly nonlinear (except for very low Reynolds number flows), and the geometries can be very complex. For such problems analytical solutions are out of the question, and numerical methods are required. As numerical methods have improved, and computers have become faster, cheaper, and larger (in both processor memory and hard-drive or solid-state storage), numerical solution has become more feasible for complex problems, and are becoming quite common. Some examples of such problems are the following.

- Aircraft design, including both the aerodynamics and the ventilation systems, among other things.
- Auto and trucking industry, including drag forces (even the drag forces on the wheels), and engine design to decrease pollutants and increase performance.
- Power generation, e.g., flow in an industrial burner, or flow in a fuel cell.
- Biomedical, e.g., cardiovascular flows, the pulmonary system.
- Weather prediction.

In fact computational fluid mechanics is now used in many applications involving fluid mechanics.

There are a number of advantages to using computational fluid dynamics.

- Compared to laboratory experiments, there is not the need to build a new model for each test case. Generally the approach is then less expensive, and the design or research process is much faster.
- Compared to laboratory experiments, some issues can be studied experimentally that cannot be studied in the laboratory, e.g.,
  - fire on a spacecraft (at zero gravity so that there is no buoyancy);
  - hypersonic flows.
- The results from a numerical simulation are very comprehensive, e.g.,

– results are obtained at each point of interest in space and time, which is generally not possible to do experimentally;
– results are obtained for all the variables of interest, e.g., pressure and specie concentrations; some of these are impossible to measure experimentally.

However, there are also some disadvantages in using numerical simulations. These include the following.

- Models for the physics and chemistry used in the simulations may not be very accurate; this is especially true for turbulent flows, and can tremendously limit the quality of the results of a simulation.
- Numerical errors – there might not be enough resolution in the simulations, or the method of integration might not be sufficiently accurate, or both.

Because both numerical simulations and laboratory experiments have significant strengths and weaknesses, it is important to understand these strengths and weaknesses, and to consider which is most appropriate for a given problem. And sometimes, when considering a complex problem such as the design of airplanes, automobiles, trucks, etc., both methods can be employed synergistically, taking advantage of the strengths of each.

The partial differential equations describing fluid flows cannot be directly solved by modern computers. Numerical methods, such as finite-volume methods, are used to approximate the partial differential equations by systems of algebraic equations, which can then be solved by computers. In these notes the basic ideas involved in using finite-volume numerical methods to solve flow problems are introduced, using as an example the one-dimensional heat equation. Finite-volume methods are used in popular computer codes such as ANSYS Fluent, STAR-CCM+, and OpenFOAM.

2 Spatial Discretization

Consider the one-dimensional heat conduction problem satisfying the following:

$$\frac{\partial T}{\partial t} = k\frac{\partial^2 T}{\partial y^2}\,, \quad 0 \le y \le L\,, \quad t \ge 0 \tag{1}$$

$$T(y,0) = F(y) \quad \text{initial conditions}$$

$$\text{boundary conditions at } y = 0, L\,.$$

Here k is the thermal diffusivity, assumed constant, L is the length of the interval of interest, and $F(y)$ is the initial temperature, assumed to be known. This overall mathematical problem is called an initial-value, boundary-value problem, and the objective here is to develop a finite-volume algorithm, implementable on the computer, to solve this for fairly arbitrary initial and boundary conditions.

To develop this algorithm, we go back to the control volume form of the equation, and consider the flow field as consisting of a number of small control volumes, subdivided by a grid, as shown in Figure 1 (in two dimensions). Note that the control volumes do not have to be the same size, nor even be rectangular, although we will assume that they are here for convenience. The variables of interest, here the temperature $T(y,t)$, are assumed to be defined at the center of each control volume, at 'nodes'. If the control volumes are small enough, then we can use interpolation to find the temperature at any other point in space.

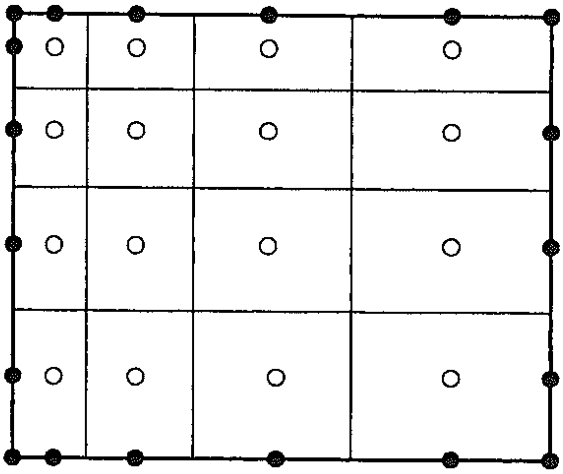

Figure 1: Problem subdivided into control volumes.

We are starting by considering the heat equation in one dimension with no flow field, as given by Equation (1). For this one-dimensional problem, we select a computational grid, control volumes, and nodes as shown in Figure 2. Note that in Equation (1) the unknown temperature $T(y,t)$ is defined continuously over the interval $[0, L]$, i.e., at each point in y. Of course each point cannot be treated directly, since there is an uncountable infinity of points, so a discrete set of control volumes is selected, defined by the points $y_0, y_1, y_2, \ldots, y_N, y_{N+1}$; this divides the interval up into N control volumes, as seen in the figure. Nodes are defined at the centers of the control volumes and at the boundary points; it is at the nodes that we would like to determine the temperature. The width of the control volumes is taken to be $\Delta y = L/N$, which is assumed to be constant here for simplicity, although this assumption can be easily relaxed. The area of each side of each control volume is taken to be A. Below we will assume that Δy is 'small' in some sense. This will be shown below to be necessary so that the numerical errors involved are small enough, and so that values of T can be accurately interpolated between nodal points.

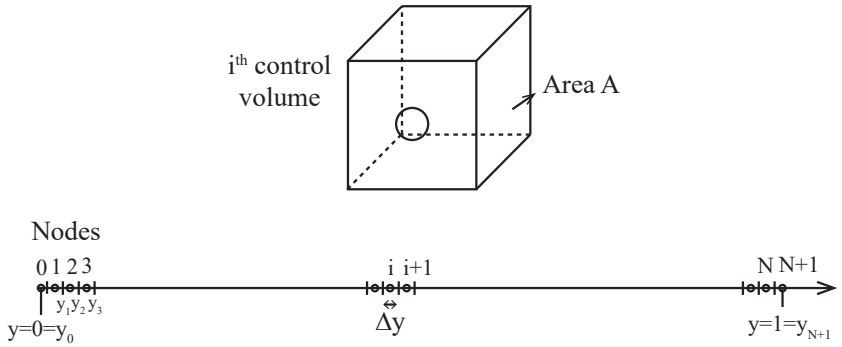

Figure 2: Computational grid with nodes.

It is convenient to define the location of the nodes as $y_0 = 0, y_i = \Delta y/2 + \Delta y(i-1)$ for $i = 1, 2, \ldots, N$, and $y_{N+1} = L$, so that we can use either y_i or i to refer to the position of the i^{th}

node on the computational mesh. We also introduce the notation $T(y_i,t) = T_i(t)$ to denote the temperature of the i^{th} node at time t.

The control volume form of the heat equation, applied to a particular small control volume, say the i^{th}, is given as:

$$\frac{d}{dt}\int_{\mathrm{CV}_i} T(y,t)d\mathcal{V} = -\int_{\mathrm{CS}_i} \mathbf{q}\cdot\mathbf{n}d\mathcal{A} = \int_{\mathrm{CS}_i} k\nabla T\cdot\mathbf{n}d\mathcal{A}\,. \tag{2}$$

Note that control volumes are not defined for the two boundary points, y_0 and y_{N+1}. We would like to evaluate this equation in terms of the nodal values of T. To evaluate the volume integral on the left-hand side of the equation we expand $T(y,t)$ about the nodal point y_i using a Taylor series expansion. With the area of the control volume in the (x,z) direction defined as A, then

$$\begin{aligned}
\int_{\mathrm{CV}_i} T(y,t)d\mathcal{V} &= A\int_{y_i-\Delta y/2}^{y_i+\Delta y/2} T(y,t)dy \\
&= A\int_{y_i-\Delta y/2}^{y_i+\Delta y/2}\left\{T(y_i,t) + \left.\frac{\partial T}{\partial y}\right|_{y_i}(y-y_i) + \left.\frac{\partial^2 T}{\partial y^2}\right|_{y_i}\frac{(y-y_i)^2}{2} + \ldots\right\}dy \\
&\underbrace{=}_{\zeta=y-y_i} A\int_{-\Delta y/2}^{\Delta y/2}\left\{T_i(t) + \left.\frac{\partial T}{\partial y}\right|_{y_i}\zeta + \left.\frac{\partial^2 T}{\partial y^2}\right|_{y_i}\frac{\zeta^2}{2} + \ldots\right\}d\zeta \\
&= A\left\{T_i(t)\zeta + \left.\frac{\partial T}{\partial y}\right|_{y_i}\frac{\zeta^2}{2} + \left.\frac{\partial^2 T}{\partial y^2}\right|_{y_i}\frac{\zeta^3}{6} + \ldots\right\}\Bigg|_{-\Delta y/2}^{\Delta y/2} \\
&= A\Delta y\left\{T_i(t) + 0 + \left.\frac{\partial^2 T}{\partial y^2}\right|_{y_i}\frac{(\Delta y)^2}{24} + \ldots\right\}.
\end{aligned} \tag{3}$$

With the volume defined by $V_i = A\Delta y$, we will then approximate this integral by

$$\int_{\mathrm{CV}_i} T(y,t)d\mathcal{V} = V_i T_i(t)\,. \tag{4}$$

The error in making this approximation,

$$\mathcal{T} = V_i\left\{\left.\frac{\partial^2 T}{\partial y^2}\right|_{y_i}\frac{(\Delta y)^2}{24} + \ldots\right\}, \tag{5}$$

is called the truncation error. We would like to keep this error as small as necessary so that the error in the numerical solution will be acceptable.

In the case of Equation (5), it is said that '$\mathcal{T}$ is of order $(\Delta y)^2$', which is written as $\mathcal{T} = \mathcal{O}(\Delta y)^2$, where the symbol $\mathcal{O}$ is defined by

$$|\mathcal{T}| \le \mathcal{K}(\Delta y)^2 \text{ as } \Delta y \to 0\,,$$

where $\mathcal{K}$ is a positive constant; i.e., $|\mathcal{T}|$ goes to 0 as fast as, or faster than $(\Delta y)^2$. In the case of Equation (3), $\mathcal{K} \ge \left|\frac{1}{24}\frac{\partial^2 T}{\partial y^2}|_{y_i}\right|$. In general, $\mathcal{S} = \mathcal{O}(\Delta y^n)$ means that

$$|\mathcal{S}| \le \mathcal{K}'(\Delta y)^n \text{ as } \Delta y \to 0\,,$$

for $\mathcal{K}'$ a positive real number.

Note that just as 0.1^2 is smaller than 0.1, something of $\mathcal{O}(\Delta y^2)$ is usually smaller than something else of $\mathcal{O}(\Delta y)$ as $\Delta y \to 0$. Therefore, something with truncation error $\mathcal{T} = \mathcal{O}(\Delta y^2)$ is a better approximation than something with truncation error $\mathcal{T} = \mathcal{O}(\Delta y)$. STAR CCM+ generally has $\mathcal{T}(\Delta y)^2$ (said to be a 'second order scheme'); most commercial finite volume codes have this same property. Some research codes, however, have $\mathcal{O}(\Delta y)^4$, $\mathcal{O}(\Delta y)^6$, $\mathcal{O}(\Delta y)^8$ or even higher. These higher-order schemes have very good accuracy, but generally require significantly more programming, and maybe not be as flexible, e.g., in treating complex geometry.

Next we consider the right-hand side of Equation (2), which can be integrated to give:

$$\int_{CS_i} k\nabla T \cdot \mathbf{n}\, d\mathcal{A} = kA\left\{ -\left.\frac{\partial T}{\partial y}\right|_\ell + \left.\frac{\partial T}{\partial y}\right|_r \right\}, \tag{6}$$

where we have used the fact that $\mathbf{n} = -\mathbf{j}$ on the left face (ℓ) of the control volume, whereas $\mathbf{n} = \mathbf{j}$ on the right face (r), where $\mathbf{j}$ is a unit vector in the y-direction; consistent with the assumption of one-dimensionality, there is no heat flux across the front and back, and top and bottom faces in this one-dimensional problem.

We now need to write $\left.\dfrac{\partial T}{\partial y}\right|_{\ell,r}$ in terms of the nodal values of T. Consider the forwards and backwards Taylor series expansions of the temperature T about the left face (ℓ) , i.e.,

$$T_i = T_\ell + \left.\frac{\partial T}{\partial y}\right|_\ell \frac{\Delta y}{2} + \frac{1}{2}\left.\frac{\partial^2 T}{\partial y^2}\right|_\ell \frac{(\Delta y)^2}{4} + \frac{1}{6}\left.\frac{\partial^3 T}{\partial y^3}\right|_\ell \frac{(\Delta y)^3}{8} + \cdots \tag{7}$$

$$T_{i-1} = T_\ell + \left.\frac{\partial T}{\partial y}\right|_\ell \frac{(-\Delta y)}{2} + \frac{1}{2}\left.\frac{\partial^2 T}{\partial y^2}\right|_\ell \frac{(-\Delta y)^2}{4} + \frac{1}{6}\left.\frac{\partial^3 T}{\partial y^3}\right|_\ell \frac{(-\Delta y)^3}{8} + \cdots \tag{8}$$

Subtracting Equation (8) from Equation (7) gives

$$T_i - T_{i-1} = 0 + 2\left.\frac{\partial T}{\partial y}\right|_\ell \frac{\Delta y}{2} + 0 + \mathcal{O}(\Delta y)^3, \text{ so, solving for } \left.\frac{\partial T}{\partial y}\right|_\ell, \tag{9}$$

$$\left.\frac{\partial T}{\partial y}\right|_\ell = \frac{T_i - T_{i-1}}{\Delta y} + \mathcal{O}(\Delta y)^2, \tag{10}$$

where the truncation error $\mathcal{T}$ is given by

$$\mathcal{T} = -\left.\frac{\partial^3 T}{\partial y^3}\right|_\ell \frac{\Delta y^2}{24} + \cdots = \mathcal{O}(\Delta y^2). \tag{11}$$

This is called a 'centered difference' formula for the derivate, i.e., for the slope (see Figure 3).

Similarly, approximating the derivative on the right face gives

$$\left.\frac{\partial T}{\partial y}\right|_r = \frac{T_{i+1} - T_i}{\Delta y} + \mathcal{O}(\Delta y)^2, \tag{12}$$

with the truncation error

$$\mathcal{T} = -\left.\frac{\partial^3 T}{\partial y^3}\right|_r \frac{\Delta y^2}{24} + \cdots = \mathcal{O}(\Delta y^2). \tag{13}$$

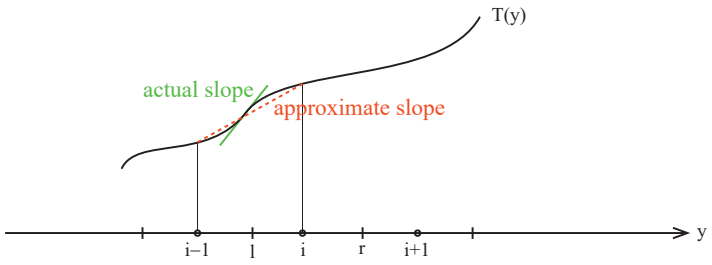

Figure 3: Comparing the true and the approximate slopes.

Next, using Equations (10) and (12) in Equation (6) results in:

$$\begin{aligned}\int_{CS_i} k\nabla T\cdot\mathbf{n}\,\mathcal{A} &= kA\left\{-\frac{T_i-T_{i-1}}{\Delta y}+\frac{T_{i+1}-T_i}{\Delta y}\right\}+\mathcal{O}(\Delta y)^2\\ &= k\underbrace{A\Delta y}_{V_i}\left\{\frac{T_{i+1}-2T_i+T_{i-1}}{(\Delta y)^2}\right\}+\mathcal{O}(\Delta y)^2\,. \end{aligned}\tag{14}$$

Finally, using the results in Equations (4) and (14) in the control volume equation, Equation (2), and dividing out the volume V_i, gives an approximation for the heat equation evaluated at the nodes:

$$\frac{dT_i}{dt}=k\,\frac{T_{i+1}-2T_i+T_{i-1}}{(\Delta y)^2}+\mathcal{O}(\Delta y)^2\quad i=1,2,\ldots,N\tag{15}$$

As we will find below, the equations for the boundary nodes will depend on the boundary conditions, and so $i=0$ and $i=N+1$ are not included for this equation. Note that if we compare Equation (15) with the origin heat equation, Equation (1), we see that the right-hand side has been approximated by

$$\left.\frac{\partial^2 T}{\partial y^2}\right|_i=\frac{T_{i+1}-2T_i+T_{i-1}}{(\Delta y)^2}+\mathcal{O}(\Delta y)^2\,,\tag{16}$$

which could have been obtained by other methods. In this particular case, although the approach is different, using finite-different methods results in the same approximation. In more general cases, however, finite-difference methods and finite-volume methods give different results.

3 Temporal Discretization – Explicit Methods

We have taken the partial differential equation for $T(y,t)$ and converted it into $N+2$ (the number of nodes) ordinary differential equations in time for the value of the temperature at the $N+2$ nodes. We now have to deal with the time dependence. This will involve approximating $\dfrac{dT_i}{dt}$. There are a number of methods to approximate the time derivative, e.g., Runga-Kutta, Adams-Bashforth, leap-frog, etc. The choice depends on the accuracy required, the programming difficulty, the numerical stability of the method, the method of parallelization of the resulting equations (if needed), and possible other requirements. We will start by using Euler's method to approximate the time difference.

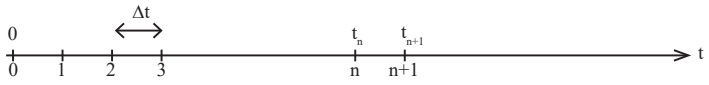

Figure 4: Discretization of time.

Consider time as discretized as shown in Figure 4. We start by using a Taylor series expansion in time, with the notation $t_n = n\Delta t$,

$$T_i(t_{n+1}) = T_i(t_n) + \left.\frac{dT_i}{dt}\right|_n \Delta t + \left.\frac{d^2T}{dt^2}\right|_n \frac{(\Delta t)^2}{2} + \dots , \tag{17}$$

or, solving for $\left.\frac{dT_i}{dt}\right|_n$,

$$\left.\frac{dT_i}{dt}\right|_n = \frac{T_i(t_{n+1}) - T_i(t_n)}{\Delta t} + \mathcal{O}(\Delta t) . \tag{18}$$

For convenience we introduce the notation $T_i(t_n) = T_i^n$. Using this notation, considering Equation (15) at time t_n, and plugging Equation (18) into Equation (15), we obtain

$$\underbrace{\frac{T_i^{n+1} - T_i^n}{\Delta t} = k\frac{T_{i+1}^n - 2T_i^n + T_{i-1}^n}{(\Delta y)^2}}_{\text{finite-volume approximation}} + \underbrace{\mathcal{O}(\Delta y)^2 + \mathcal{O}(\Delta t)}_{\text{truncation error}} . \tag{19}$$

Assuming that the truncation error is small, the original partial differential equation can then be approximated by

$$\frac{T_i^{n+1} - T_i^n}{\Delta t} = k\frac{T_{i+1}^n - 2T_i^n + T_{i-1}^n}{(\Delta y)^2} , \text{ or} \tag{20}$$

$$T_i^{n+1} = T_i^n + R(T_{i+1}^n - 2T_i^n + T_{i-1}^n) , \; R = \frac{k\Delta t}{(\Delta y)^2} , \; i = 1, 2, \dots N . \tag{21}$$

This is our first finite-volume approximation to the heat equation. The equations for the boundary points $i = 0$ and $i = N + 1$ will depend on the boundary conditions.

Consider Equation (21) and note the following:

- This equation holds for each $i = 1, 2, \dots, N$.
- If T_i^n is known for $i = 0, 1, 2, \dots, N + 1$, i.e., if we know T at time step n, this equation plus the equations for the boundary points will allow the computation of T at time step $n + 1$. Knowing T at time step $n + 1$, then the equation can be used to compute T at time step $n + 2$, etc. This is known as time stepping, or time marching. It is described by the grid in Figure 5.
- For the scheme to work, we will need to select Δt and Δy small enough so that the truncation error is acceptably small. This will be discussed in more depth later.
- The accuracy is not as good in t as in y.

The type of time-stepping expression given by Equation (21) is called 'explicit', and is of the form

$$T_i^{n+1} = F(T_i^n, T_{i+1}^n, T_{i-1}^n, \dots) , \tag{22}$$

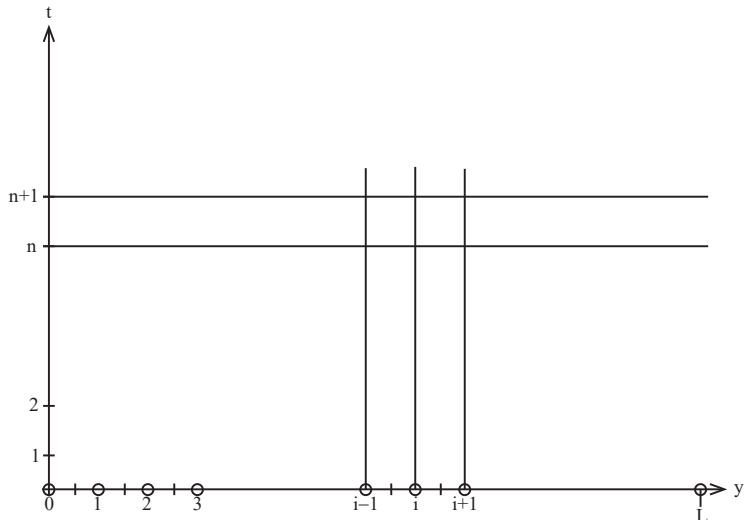

Figure 5: Grid showing the time-stepping from time t_n to t_{n+1} for the i^{th} node.

where F is a known function. In an explicit expression, if T_i^n is given for all i, then one can directly solve for T_i^{n+1}. However, we will find in general that the time-stepping expression can be of the form

$$T_i^{n+1} = G(T_{i+1}^{n+1}, T_i^{n+1}, T_{i-1}^{n+1}, T_i^n, T_{i+1}^n, T_{i-1}^n, \ldots)\,, \tag{23}$$

where again G is a known function. Now there are unknowns on the right-hand side, so that one cannot directly solve for T_i^{n+1}. This latter expression is called 'implicit', and is usually harder to treat from a programming point of view, and usually requires more computational time, but can have better numerical properties. We will first consider the explicit scheme given by Equation (21) above, and then later consider an implicit scheme which will be shown to have better numerical properties.

In Equation (21) we now have a finite-volume approximation to Equation (1) that we can use, under certain restrictions, to obtain an accurate solution for $T(y,t)$. But in order to obtain solutions to a particular problem, we still must consider initial conditions and boundary conditions.

Initial Conditions. To solve Equation (1) or Equation (21), $T(y,t)$ needs to be defined at some point in time, which we will call $t = 0$. We will assume that we can obtain $T(y,0)$ at the nodes on the computational mesh, usually in one of two ways,

1. computing the values from an analytical functions, or
2. reading in data values obtained in some other way, e.g., from a laboratory experiment or another simulation.

For example, suppose that for a particular problem,

$$T(y,0) = T_m \exp(-y^2/d^2)\,, \quad 0 \le y \le L\,, \tag{24}$$

where T_m and d are known constants. Then we would compute $T(y_i,0) = T_i^0$ as

$$\begin{aligned} T_0^0 &= T_m \exp(0) = T_m \\ T_i^0 &= T_m \exp(-y_i^2/d^2) = T_m \exp\{-[\Delta y/2 + (i-1)\Delta y]^2/d^2\}\,, \quad i = 1,2,\ldots,N \\ T_{N+1}^0 &= T_m \exp(-L^2/d^2)\,. \end{aligned} \tag{25}$$

The finite-difference scheme would then be used to compute T_i^1 for $i = 0, 1, 2, \ldots, N+1$, then T_i^2, etc.

Boundary Conditions. The manner in which temperature changes with time usually depends strongly on what happens at the boundaries, i.e., on the boundary conditions. In the numerical scheme given by Equation (21), the 1^{st} point depends on the 0^{th} (boundary point) and 2^{nd} points, the 2^{nd} point depends on the 1^{st} and 3^{rd} points, etc. Information therefore 'propagates' away from the boundaries. This is of course representing the molecular conduction (or diffusion for mass) of internal energy away from (or to) the boundaries.

The boundary conditions depend on the problem under consideration, and it is up to the engineer or scientist to determine the proper boundary conditions for a problem. In this discussion we will consider two types of boundary conditions, (i) constant temperature, and (ii) insulated boundaries.

Constant Temperature. If there is (in the idealized case) a constant temperature reservoir at a boundary, e.g., a constant temperature heat source at the left boundary ($y = 0$) at temperature T_ℓ, then the appropriate boundary condition is: $T_0^n = T_\ell$ for $n \geq 0$ ($t \geq 0$). In addition, if there is a constant temperature heat source of temperature T_r at the right boundary ($y = L$), then the appropriate boundary condition there is: $T_{N+1}^n = T_r$ for $n \geq 0$. The equation for the first nodes away from the boundary points, nodes 1 and N, will have to be modified because of the structure of the grid near the boundary. Consider node 1 with a constant temperature boundary condition. Equation (6) still holds, but now a new estimate is needed for $\left.\dfrac{\partial T}{\partial y}\right|_0$. This can be obtained by considering a Taylor series expansion of T_1 about $y = 0$:

$$T_1 = T_0 + \left.\frac{\partial T}{\partial y}\right|_0 \frac{\Delta y}{2} + \mathcal{O}(\Delta y)^2 \,. \tag{26}$$

Solving this for $\left.\dfrac{\partial T}{\partial y}\right|_0$ gives

$$\frac{\partial T}{\partial y} = \frac{T_1 - T_0}{\Delta y/2} + \mathcal{O}(\Delta y) \,. \tag{27}$$

Note that this expression is only accurate to $\mathcal{O}(\Delta y)$. The accuracy could be improved by including three points, y_0, y_1, and y_2 in the expression, but this will not be done here. Using Equation (27) in Equation (6) then gives:

$$\begin{aligned} \int_{CS_1} k\nabla T \cdot \mathbf{n}\, \mathcal{A} &= kA\left\{-\frac{2(T_1 - T_0)}{\Delta y} + \frac{T_2 - T_1}{\Delta y}\right\} + \mathcal{O}(\Delta y) \\ &= k\underbrace{A\Delta y}_{V_i}\left\{\frac{T_2 - 3T_1 + 2T_0}{(\Delta y)^2}\right\} + \mathcal{O}(\Delta y) \,. \end{aligned} \tag{28}$$

Therefore, for node 1 Equation (21) is modified to give:

$$T_1^{n+1} = T_1^n + R(T_2^n - 3T_1^n + 2T_2^n) \,. \tag{29}$$

Assuming a constant temperature boundary condition at $y = L$ leads to, with similar analysis,

$$T_N^{n+1} = T_N^n + R(2T_{N+1}^n - 3T_N^n + T_{N-1}^n) \,. \tag{30}$$

Therefore, the system of finite-difference equations would be the following.

$$\begin{aligned}
T_0^{n+1} &= T_\ell \\
T_1^{n+1} &= T_1^n + R(T_2^n - 3T_1^n + 2T_0^n)\,,\ T_0^n = T_\ell \\
T_2^{n+1} &= T_2^n + R(T_3^n - 2T_2^n + T_1^n) \\
&\vdots \\
T_i^{n+1} &= T_i^n + R(T_{i+1}^n - 2T_i^n + T_{i-1}^n) \\
&\vdots \\
T_N^{n+1} &= T_N^n + R(2T_{N+1}^n - 3T_N^n + T_{N-1}^n)\,,\ T_{N+1}^n = T_r \\
T_{N+1}^{n+1} &= T_r
\end{aligned}$$

Therefore, given T_i^0, $i = 0, 1, 2, \ldots, N, N+1$, this scheme could be used to compute T_i^n for latter times.

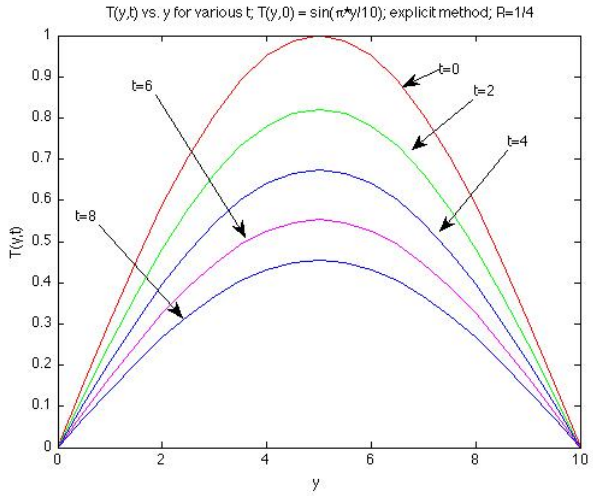

Figure 6: $T(y,t)$ versus y for five different times. Explicit method. $R = 1/4$.

An example problem was run using this algorithm in Matlab. Equation 1 was solved over the domain $0 \le y \le 10$ subject to the initial and boundary conditions:

$$T(y,0) = \sin(\pi y/10)\,, \quad T(0,t) = T(10,t) = 0\,. \tag{31}$$

There were 20 control volumes used, giving $\Delta y = 1/2$, the thermal diffusivity k was taken to be 1, R was taken to be $1/4$, giving $\Delta t = R(\Delta y)^2/k = 1/16$, and the time interval was chosen as $0 \le t \le 8$, giving $n_s = 8/(1/16) = 128$ time steps.

Figure 6 gives a plot of $T(y,t)$ for five different times over the time period of interest. The temperature decreases monotonically in time, as there is heat flux at both boundaries, decreasing the internal energy over the domain. Figure 7 gives a comparison at $t = 8$ of the computed solution with the exact solution,

$$T(y,t) = \exp\left\{-\left(\frac{\pi}{10}\right)^2 t\right\} \sin(\pi y/10)\,.$$

The agreement is excellent. The problem was rerun, however, with $R = 1$, resulting in $\Delta t = 1/4$. Figure 8 is a plot of $T(y,t)$ for five different times over the period of interest. It is seen that, by

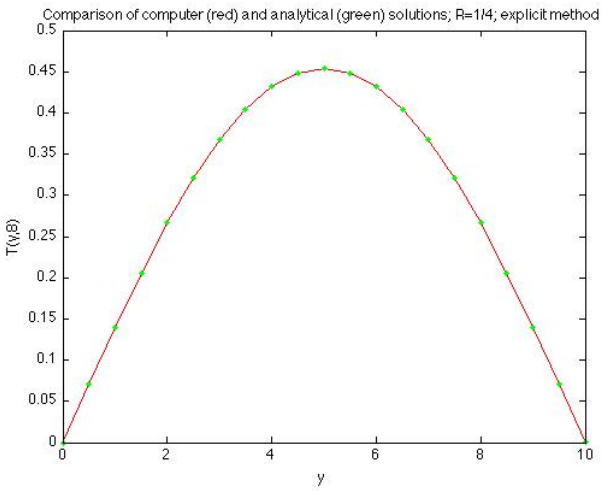

Figure 7: Comparison of the exact solution with the computed solution at $t = 8$. Explicit method.

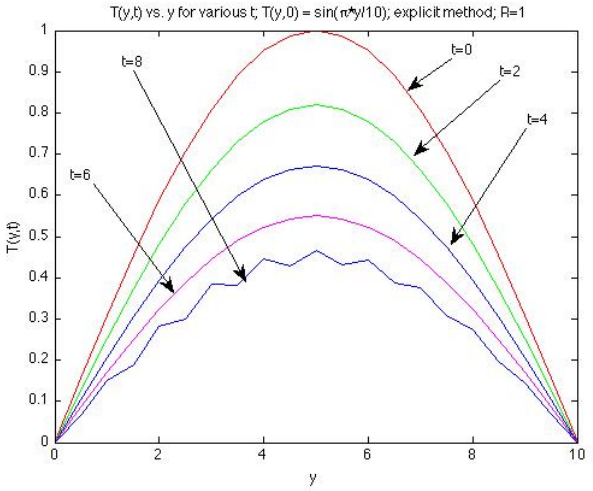

Figure 8: $T(y,t)$ versus y for five different times. Explicit method. $R = 1$.

$t = 8$, the solution is breaking down. We will find out below that the explicit scheme is unstable for $R \geq 1/2$.

Derivative Boundary Conditions. Often the temperature is not known at the boundary, but the heat flux is. For example, if the boundary is insulated, then the heat flux is zero, i.e.,

$$q_y = -\kappa \frac{\partial T}{\partial y}\bigg|_0 = 0\,, \text{ or } \frac{\partial T}{\partial y}\bigg|_0 = 0\,. \tag{32}$$

Another common situation is when the heat flux is related to the difference between the temperature at the boundary and the ambient temperature, i.e.,

$$\begin{aligned} -q_y|_0 &= hA(T|_0 - T_{\text{amb}})\,, \text{ or} \\ \\ \kappa \frac{\partial T}{\partial y}\bigg|_0 &= hA(T|_0 - T_{\text{amb}})\,. \end{aligned}$$

where A is the area of the boundary, h is a coefficient of heat transfer, and T_{amb} is the ambient temperature. We will only consider the insulated case, although the more general case is a straight forward generalization.

Suppose that $\left.\frac{\partial T}{\partial y}\right|_0 = 0$. The the RHS of the equation for node 1 becomes

$$\begin{aligned}\int_{CS_1} k\nabla T\cdot\mathbf{n}\,d\mathcal{A} &= kA\Big\{-\underbrace{\left.\frac{\partial T}{\partial y}\right|_L}_{=0}+\left.\frac{\partial T}{\partial y}\right|_R\Big\}\\ &= kA\,\frac{T_2-T_1}{\Delta y}+\mathcal{O}(\Delta y)^2\\ &= k\,\underbrace{A\Delta y}_{V_1}\,\frac{T_2-T_1}{(\Delta y)^2}+\mathcal{O}(\Delta y)^2\,. \end{aligned}\tag{33}$$

Combining this with the time derivative gives:

$$V_1\frac{T_i^{n+1}-T_i^n}{\Delta t} = kV_1\Big\{\frac{T_2-T_1}{(\Delta y)^2}\Big\}\,.$$

The finite-volume expression for the node 1 is then:

$$T_1^{n+1} = T_1 + R(T_2^n - T_1^n)\,.\tag{34}$$

Note that the boundary point, node 0, isn't directly computed. Its value can be obtained by extrapolation in the following way. Writing a Taylor series about $y=0$,

$$T(y) = T_0 + \underbrace{\left.\frac{\partial T}{\partial y}\right|_0}_{=0} y + \frac{1}{2}\left.\frac{\partial^2 T}{\partial y^2}\right|_0 y^2 + \mathcal{O}(\Delta y)^3\,.$$

Evaluating this at nodes 1 and 2 gives:

$$T_1 = T_0 + \frac{1}{2}\left.\frac{\partial^2 T}{\partial y^2}\right|_0\left(\frac{\Delta y}{2}\right)^2 + \mathcal{O}(\Delta y)^3\,,\text{ and}\tag{35}$$

$$T_2 = T_0 + \frac{1}{2}\left.\frac{\partial^2 T}{\partial y^2}\right|_0\left(\frac{3\Delta y}{2}\right)^2 + \mathcal{O}(\Delta y)^3\,.\tag{36}$$

Subtracting Equation (35) from Equation (36) gives:

$$T_2 - T_1 = \frac{1}{2}\left.\frac{\partial^2 T}{\partial y^2}\right|_0 2(\Delta y)^2 + \mathcal{O}(\Delta y)^3\,,\text{ so that}$$

$$\left.\frac{\partial^2 T}{\partial y^2}\right|_0 = \frac{T_2-T_1}{(\Delta y)^2} + \mathcal{O}(\Delta y)^1\,.$$

Plugging this back into Equation (35) gives

$$\begin{aligned} T_1 &= T_0 + \frac{1}{2}\Big[\frac{T_2-T_1}{(\Delta y)^2}+\mathcal{O}(\Delta y)\Big]\left(\frac{\Delta y}{2}\right)^2 + \mathcal{O}(\Delta y)^3\\ &= T_0 + \frac{1}{8}(T_2-T_1) + \mathcal{O}(\Delta y)^3\,,\end{aligned}$$

so the equation for node 0 is:

$$T_0^{n+1} = \frac{9}{8}T_1^{n+1} - \frac{1}{8}T_2^{n+1}\,, \tag{37}$$

with error $\mathcal{O}(\Delta y)^3$. So T_0^{n+1} can be computed, once T_1^{n+1} and T_2^{n+1} are known.

Similarly, with an insulated right boundary,

$$\begin{aligned}\int_{CS_N} k\nabla T\cdot\mathbf{n}\,d\mathcal{A} &= kA\Big\{-\frac{\partial T}{\partial y}\Big|_L + \underbrace{\frac{\partial T}{\partial y}\Big|_R}_{=0}\Big\}\\ &= -kA\,\frac{T_N - T_{N-1}}{\Delta y} + \mathcal{O}(\Delta y)^2\\ &= -k\underbrace{A\Delta y}_{V_1}\frac{T_N - T_{N-1}}{(\Delta y)^2} + \mathcal{O}(\Delta y)^2\,. \end{aligned}\tag{38}$$

Bringing in the LHS then gives

$$\frac{T_N^{n+1} - T_N^n}{\Delta t} = k\frac{T_{N-1}^n - T_N^n}{(\Delta y)^2}\,,$$

giving the expression for node N as

$$T_N^{n+1} = T_N^n + R(T_{N-1}^n - T_N^n)\,.$$

Again the value of node (N+1) can be estimated by extrapolation, given the values of T_N^{n+1} and T_{N-1}^{n+1}. The result is

$$T_{N+1}^{n+1} = \frac{9}{8}T_N^{n+1} - \frac{1}{8}T_{N-1}^{n+1}\,.$$

An example problem was also run in Matlab for the case of derivative boundary conditions. Equation 1 was again solved over the domain $0 \le y \le 10$, but now with the following initial and boundary conditions:

$$T(y,0) = \cos(\pi y/10)\,, \quad \frac{\partial T}{\partial y}\Big|_{y=0} = \frac{\partial T}{\partial y}\Big|_{y=10} = 0\,. \tag{39}$$

Again 20 control volumes were used, giving $\Delta y = 1/2$, the thermal diffusivity k was taken to be 1, R was taken to be 1/4, giving $\Delta t = 1/16$, and the time interval was chosen to be $0 \le t \le 8$, giving 128 time steps.

Figure 9 gives a plot of the solution $T(y,t)$ for five times over the time interval of interest. In this case, with no heat loss at the boundaries, there is conduction of temperature (internal energy) from the left side of the domain to the right side. Figure 10 gives a comparison of the computed solution with the exact solution,

$$T(y,t) = \exp\Big\{-\Big(\frac{\pi}{10}\Big)^2 t\Big\}\cos(\pi y/10)\,.$$

The agreement is again excellent.

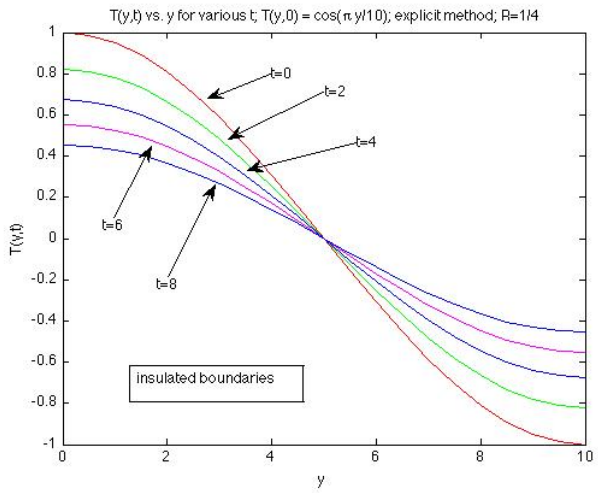

Figure 9: $T(y,t)$ versus y for five different times. Explicit method.

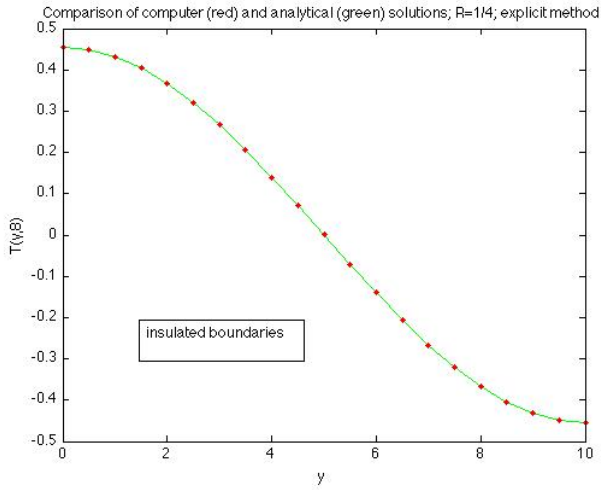

Figure 10: Comparison of the exact solution with the computed solution at $t = 8$. Explicit method.

4 Temporal Discretization – Implicit Methods

The previous explicit method has several limitations, namely,

- the truncation error is $\mathcal{O}(\Delta t)$ in time, but $\mathcal{O}((\Delta y)^2)$ in space. Therefore one has to take considerably smaller Δt compared to Δy for the same accuracy;
- related to this, as we will find below, the method is unstable for $R > 1/2$. Since then $R = \dfrac{k\Delta t}{(\Delta y)^2} \leq 1/2$ for stability, then $\Delta t \leq \dfrac{(\Delta y)^2}{2k}$, i.e., the time step is severely restricted. For example, if Δy is decreased by a factor of 2, then Δt must be decreased by a factor of 4.

This latter fact is not important in the simpler problems discussed here, but becomes crucial in larger-scale problems, especially in fluid mechanics.

A method to avoid these problems, an implicit methods called the "Crank-Nicolson" method can be used, and goes as follows. Assuming that the values of T_i^n are given at time step n, we would like to develop an implicit algorithm to compute its values at time step $n+1$. For the moment, consider the artificial (fictitious) point in time $n+1/2$, equidistant in time between n and $n+1$. The central difference approximation for the time derivative at this point in time is given by

$$\left.\frac{\partial T}{\partial t}\right|_j^{n+1/2} = \frac{T_j^{n+1} - T_j^n}{\Delta t} + \mathcal{O}(\Delta t)^2\,, \tag{40}$$

which is more accurate than the forward difference formula used in the explicit scheme. We would now like to consider the heat equation at the point $(j, n+1/2)$, i.e.,

$$\left.\frac{\partial T}{\partial t}\right|_j^{n+1/2} = k\left.\frac{\partial^2 T}{\partial y^2}\right|_j^{n+1/2}\,.$$

Unfortunately, assuming values of y and t at the nodes and times n and $n+1$, we only have the values at (j,n) and $(j,n+1)$. To complete the scheme using Equation (40), we also need to approximate the right-hand side of Equation (1), $k\dfrac{\partial^2 T}{\partial y^2}$, at $(j, n+1/2)$. We can approximate a function at the time step $n+1/2$ in terms of its values at n and $n+1$ by averaging its values at time steps n and $n+1$. Consider the Taylor series expansions in time of the function $F(t)$ at n and $n+1$ about the time $n+1/2$.

$$F^{n+1} = F^{n+1/2} + \left.\frac{\partial F}{\partial t}\right|_{n+1/2}\frac{\Delta t}{2} + \left.\frac{\partial^2 F}{\partial t^2}\right|_{n+1/2}\frac{(\Delta t)^2}{8} + \mathcal{O}(\Delta t)^3 \tag{41}$$

$$F^{n} = F^{n+1/2} + \left.\frac{\partial F}{\partial t}\right|_{n+1/2}\frac{(-\Delta t)}{2} + \left.\frac{\partial^2 F}{\partial t^2}\right|_{n+1/2}\frac{(-\Delta t)^2}{8} + \mathcal{O}(\Delta t)^3 \tag{42}$$

Adding, dividing by 2, and rearranging gives

$$F^{n+1/2} = \frac{1}{2}(F^n + F^{n+1}) + \mathcal{O}(\Delta t)^2\,. \tag{43}$$

This could be called a 'centered sum'. Note that the accuracy is $\mathcal{O}(\Delta t)^2$, the same as desired for the time-stepping.

Replacing $F(t)$ in Equation (43) by $k\dfrac{\partial^2 T}{\partial y^2}$, and using its centered difference approximation in space,

$$k\left.\frac{\partial^2 T}{\partial y^2}\right|_i^{n+1/2} = \frac{k}{2}\left\{\frac{T_{i+1}^n - 2T_i^n + T_{i-1}^n}{(\Delta y)^2} + \frac{T_{i+1}^{n+1} - 2T_i^{n+1} + T_{i-1}^{n+1}}{(\Delta y)^2}\right\} + \mathcal{O}((\Delta t)^2, \Delta y)^2)\,. \tag{44}$$

Finally, plugging Equations (40) and (44) into Equation 1, the heat equation, results in

$$\frac{T_i^{n+1} - T_i^n}{\Delta t} = \frac{k}{2(\Delta y)^2}[(T_{i+1}^n - 2T_i^n + T_{i-1}^n) + (T_{i+1}^{n+1} - 2T_i^{n+1} + T_{i-1}^{n+1})]\,, \tag{45}$$

with truncation error $\mathcal{O}((\Delta t)^2, (\Delta y)^2)$. Multiplying through by Δt, moving the unknowns (values at time step $n+1$) to the left-hand side, and using $R = \dfrac{k\Delta}{(\Delta y)^2}$ gives

$$-\frac{R}{2}T_{i+1}^{n+1} + (1+R)T_i^{n+1} - \frac{R}{2}T_{i-1}^{n+1} = \frac{R}{2}T_{i+1}^n + (1-R)T_i^n + \frac{R}{2}T_{i-1}^n\,, \tag{46}$$

an equation for each $i = 1, 2, 3, \ldots, N$ (except possibly at boundaries). For a given value of i, say $i = 3$, the equation is the following:

$$-\frac{R}{2}T_4^{n+1} + (1+R)T_3^{n+1} - \frac{R}{2}T_2^{n+1} = \frac{R}{2}T_4^n + (1-R)T_3^n + \frac{R}{2}T_2^n\,,$$

where the right-hand side is considered to be known. The next equation is

$$-\frac{R}{2}T_5^{n+1} + (1+R)T_4^{n+1} - \frac{R}{2}T_3^{n+1} = \frac{R}{2}T_5^n + (1-R)T_4^n + \frac{R}{2}T_3^n\,,$$

and so on. Note again that the i^{th} equation contains values of temperature at adjacent grid points, again due to the conductive properties of the heat equation.

Note that we no longer have an explicit expression for any of the unknowns, T_i^{n+1}, but instead T_i^{n+1} depends on T_{i+1}^{n+1} and T_{i-1}^{n+1}, other unknowns. This is characteristic of 'implicit' numerical methods. This method will have better numerical properties than the explicit method introduced above, but the resulting equations can be more difficult to program and take more computational power to solve.

For the implicit scheme, the initial conditions are applied in the same manner as for the explicit scheme, as discussed above. It is useful, however, to discuss the boundary conditions for the implicit scheme.

Constant Temperature. The constant temperature boundary conditions are straight-forward. For example, assume that at the left boundary, $T(0,t) = T_\ell$. So $T_0^{n+1} = T_\ell$. The equation for T_1^{n+1} can be found from the evaluation of the RHS of the heat equation, given by Equation (28), and is

$$\frac{T_1^{n+1} - T_1^n}{\Delta t} = \frac{k}{2}\left\{\frac{T_2^n - 3T_1^n + 2T_o^n}{(\Delta y)^2} + \frac{T_2^{n+1} - 3T_1^{n+1} + 2T_0^{n+1}}{(\Delta y)^2}\right\}, \text{ or}$$

$$-\frac{R}{2}T_2^{n+1} + (1+\frac{3}{2}R)T_1^{n+1} - RT_0^{n+1} = \frac{R}{2}T_2^n + (1-\frac{3}{2}R)T_1^n + RT_0^n\,. \tag{47}$$

With a constant temperature boundary condition at $y = L$, a similar approach leads to:

$$-\frac{R}{2}T_{N+1}^{n+1} + (1+\frac{3}{2}R)T_N^{n+1} - \frac{R}{2}T_{N-1}^{n+1} = RT_{N+1}^n + (1-\frac{3}{2}R)T_N^n + \frac{R}{2}T_{N-1}^n\,, \text{ and} \tag{48}$$

$$T_{N+1}^{n+1} = T_r\,. \tag{49}$$

So, assuming constant temperature boundary conditions on both boundaries, the system of equations to be solved is the following.

$$\begin{aligned}
T_0^{n+1} &= T_\ell \\
-\frac{R}{2}T_2^{n+1} + (1+\frac{3}{2}R)T_1^{n+1} - RT_0^{n+1} &= \frac{R}{2}T_2^n + (1-\frac{3}{2}R)T_1^n + \frac{R}{2}T_0^n \\
&\vdots \\
-\frac{R}{2}T_{i+1}^{n+1} + (1+R)T_i^{n+1} - \frac{R}{2}T_{i-1} &= \frac{R}{2}T_{i+1}^n + (1-R)T_i^n + \frac{R}{2}T_{i-1} \\
&\vdots \\
-\frac{R}{2}T_{N+1}^{n+1} + (1+\frac{3}{2}R)T_N^{n+1} - RT_{N-1}^{n+1} &= \frac{R}{2}T_{N+1}^n + (1-\frac{3}{2}R)T_N^n + \frac{R}{2}T_{N-1}^n \\
T_{N+1}^{n+1} &= T_r\,.
\end{aligned} \tag{50}$$

Insulated Boundaries. For an insulated boundary at $y = 0$, using Equations (33) and (43), for node 1 we have

$$\frac{T_1^{n+1} - T_1^n}{\Delta t} = \frac{k}{2}\left\{\frac{T_2^{n+1} - T_1^{n+1}}{(\Delta y)^2} + \frac{T_2^n - T_1^n}{(\Delta y)^2}\right\},$$

which leads to

$$-\frac{R}{2}T_2^{n+1} + \left(1 + \frac{R}{2}\right)T_1^{n+1} = \frac{R}{2}T_2^n + \left(1 - \frac{R}{2}\right)T_1^n. \tag{51}$$

Again extrapolation can be used to determine T_0^{n+1} in terms of T_1^{n+1} and T_2^{n+1}, leading to Equation (37). Given an insulated boundary at $y = L$, an equation similar to Equation (51) can be obtained for node N, and then the value for node $(N + 1)$ can be obtained by extrapolation.

Both constant temperature and insulated boundaries lead to a system of equations of the form (here only the equations for the former case will be written out, although the equations for insulated boundary conditions are very similar):

$$\begin{array}{ccccccccccc}
(1+R)T_1 & -\frac{R}{2}T_2 & +0 & \dots & +0 & +0 & +0 & +0 & = & Q_1 \\
-\frac{R}{2}T_1 & +(1+R)T_2 & -\frac{R}{2}T_3 & +0 & \dots & +0 & +0 & +0 & = & Q_2 \\
 & & & \vdots & & & & & & \\
0 & +0 & \dots & -\frac{R}{2}T_{i-1} & +(1+R)T_i & -\frac{R}{2}T_{i+1} & \dots & +0 & = & Q_i \\
 & & & \vdots & & & & & & \\
0 & +0 & +0 & +0 & +0 & \dots & -\frac{R}{2}T_{N-1} & +(1+R)T_N & = & Q_N
\end{array}$$

Here the superscripts have been dropped for the moment. There are N algebraic equations in N unknowns. The one-dimensional array $(T_1, T_2, \dots, T_N)$ is the unknown, given in terms of T_i^{n+1}, while the one-dimensional array $(Q_1, Q_2, \dots, Q_N)$ is known, given in terms of the T_i^n. In matrix form this can be written as

$$\left|\begin{array}{cccccccc||c|c|c}
(1+R) & \frac{R}{2} & 0 & \dots & 0 & 0 & 0 & 0 & T_1 & = & Q_1 \\
-\frac{R}{2} & (1+R) & -\frac{R}{2} & 0 & \dots & 0 & 0 & 0 & T_2 & = & Q_2 \\
 & & & \vdots & & & & & & & \\
0 & 0 & \dots & -\frac{R}{2} & +(1+R) & -\frac{R}{2} & \dots & 0 & T_i & = & Q_i \\
 & & & \vdots & & & & & & & \\
0 & 0 & 0 & 0 & 0 & \dots & -\frac{R}{2} & +(1+R) & T_N & = & Q_N
\end{array}\right| \tag{52}$$

Or $\mathbf{MT} = \mathbf{Q}$ in matrix form.

The matrix $\mathbf{M}$ has non-zero components only along the diagonal and each side of the diagonal; such a matrix is called "tridiagonal". For $N = 3$, $\mathbf{M}$ is a 3x3 matrix, and the solution to the equations comes from well-known formulae involving determinants, etc. In general, however, the matrix $\mathbf{M}$ is an NxN matrix where N is very much larger than 3 (perhaps in the 100's or 1,000's or more). Solving these equation for large N and arbitrary $\mathbf{M}$ is a formidable task. Fortunately, for tridiagonal matrices, the task is much simpler.

Consider the following example:

$$\begin{aligned} 3x_1 - x_2 + 2x_3 &= 12 \quad (53) \\ x_1 + 2x_2 + 3x_3 &= 11 \quad (54) \\ 2x_1 - 2x_2 - x_3 &= 2 \quad (55) \end{aligned}$$

Note these two rules regarding the manipulation of these equations:

i. if we multiply a row by a constant, the new system has the same solution as the original system (unless we multiply by 0);

ii. If we add (or subtract) two rows, and replace one of the rows with the resulting sum, then again the solution to the new set of equations is the same as the solution of the original system.

In both cases we are not gaining or losing information.

We can solve the system of equations by a judicious choice of adding, subtracting, and multiplying of rows. Consider Equations 53, 54, and 55, and the following operations on these equations.

i. Multiply Equation 53 by -1 and add to Equation 54 multiplied by 3, and then using this sum to replace Equation 54;

ii. Multiply Equation 53 by -2 and add this to Equation 55 multiplied by 3, then using the result to replace Equation 55.

The result is:

$$\begin{aligned} 3x_1 - x_2 + 2x_3 &= 12 \quad (56) \\ 0x_1 + 7x_2 + 7x_3 &= 21 \quad (57) \\ 0x_1 - 4x_2 - 7x_3 &= -18 \quad (58) \end{aligned}$$

Finally, multiply Equation 57 by 4 and add it to Equation 58 multiplied by 7; the result is:

$$\begin{aligned} 3x_1 - x_2 + 2x_3 &= 12 \quad (59) \\ 0x_1 + 7x_2 + 7x_3 &= 21 \quad (60) \\ 0x_1 + 0x_2 - 21x_3 &= -42 \quad (61) \end{aligned}$$

Note that we now have a matrix which has non-zero elements only along the above the diagonal. Such a matrix is called an "upper triangular" matrix, and can be solved by back substitution in the following way.

i. Solve Equation 61, giving $x_3 = 2$.

ii. Plug in this value for x_3 into Equation 60 and solve for x_2, giving $x_2 = 1$.

iii. Plug the values for x_3 and x_2 into Equation 59 and solve for x_1 giving $x_1 = 3$.

This method, called "Gaussian elimination", can be easily implemented on the computer, although there can be accuracy problems if the matrices have certain properties, e.g., a small coefficient or 0 multiplying one or more of the diagonal terms, a situation in which the matrix is called "ill-conditioned". For the tridiagonal matrices resulting from the Crank-Nicolson method, however,

these ill-conditioning problems do not exist; the method is efficient and accurate. A popular method to solve the tridiagonal system of equations given by Equation 52 is called "Thomas' method', and is given in various textbooks on numerical analysis and on various websites (see, e.g., the *Wikipedia* version at 'http://en.wikipedia.org/wiki/Tridiagonal_matrix_algorithm').

Matlab uses Gaussian elimination to solve the matrix equation. For example, given the matrix equation

$$\mathbf{MT} = \mathbf{Q}\,,$$

then the following Matlab operation can be used to solve for $\mathbf{T}$, given $\mathbf{M}$ and $\mathbf{Q}$:

$$\mathbf{T} = \mathbf{M}\backslash\mathbf{Q}\,.$$

Therefore, to use Matlab to go from time step n to $n+1$ in the Crank-Nicolson method, the following steps are required, given $(T_1^n, T_2,^n, \ldots, T_N^n)$.

1. Define the matrix $\mathcal{M}$, which is the same for all time steps, and so is only needed to be done once.

$$\mathbf{M} = \begin{vmatrix} (1+R) & \frac{R}{2} & 0 & \ldots & 0 & 0 & 0 & 0 \\ -\frac{R}{2} & (1+R) & -\frac{R}{2} & 0 & \ldots & 0 & 0 & 0 \\ & & & \vdots & & & & \\ 0 & 0 & \ldots & -\frac{R}{2} & +(1+R) & -\frac{R}{2} & \ldots & 0 \\ & & & \vdots & & & & \\ 0 & 0 & 0 & 0 & 0 & \ldots & -\frac{R}{2} & +(1+R) \end{vmatrix} \tag{62}$$

2. Define the vector $\mathbf{Q}$ which will be new for each time step.

$$\mathbf{Q} = \begin{vmatrix} \frac{R}{2}T_2^n + \left(1 - \frac{3}{2}R\right)T_1^n + R(T_0^{n+1} + T_0^n) \\ \vdots \\ \frac{R}{2}T_{i+1}^n + (1+R)T_i^n + \frac{R}{2}T_{i-1}^n \\ \vdots \\ \frac{R}{2}T_{N-1}^n + \left(1 - \frac{R}{2}\right)T_N^n + \frac{R}{2}T_{N+1}^n + RT_{N+1}^{n+1} \end{vmatrix} \tag{63}$$

3. Solve for $\mathbf{T}^{n+1}$ as $\mathbf{T}^{n+1} = \mathbf{M}\backslash\mathbf{Q}$.

5 Numerical Stability

Explicit scheme. Consider again the heat equation, Equation 1. If we set up a finite-volume grid with spacing Δy in y, and with time step Δt in time t, we can use the explicit scheme given by Equation 21, which is accurate to $\mathcal{O}((\Delta t), (\Delta y)^2)$, to solve for $T(y,t)$. If T_s is the actual solution to the heat equation, and T_f is the solution to the explicit finite-volume equation, then we expect the discretization error,

$$\epsilon_d = T_s - T_f\,, \tag{64}$$

to become smaller and smaller as $\Delta t, \Delta y \to 0$, so that the finite-volume solution approaches the exact solution.

But as seen in the example problem for $R = 1$ given in Figure 8 above, this is not the whole story. The finite-volume scheme represents a dynamic system in its own right. It is possible that the solutions are unstable and, because of this instability, do not represent the exact solution no matter how small Δt and Δy are taken.

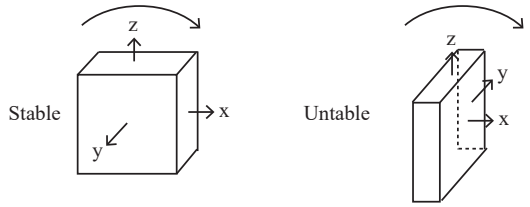

Figure 11: Examples of stable and unstable dynamic motion.

An example of a dynamic instability is seen in Figure 11, in which a rectangular parallelepiped is rotated first about its x-axis, and then about its y-axis. It can easily be demonstrated that the rotation about the x-axis produces a stable motion, while rotation about the y-axis is unstable, so that the object begins to wobble as it is rotated. In each case you can think about the object as spinning through the air about about the appropriate axis, but being subject to fluctuations due to (i) air currents and drag forces, and (ii) the fact that the object may not have been released with exactly the proper spin. These perturbations to its trajectory do not grow in the first case, but do grow in the second, i.e., the motion is unstable.

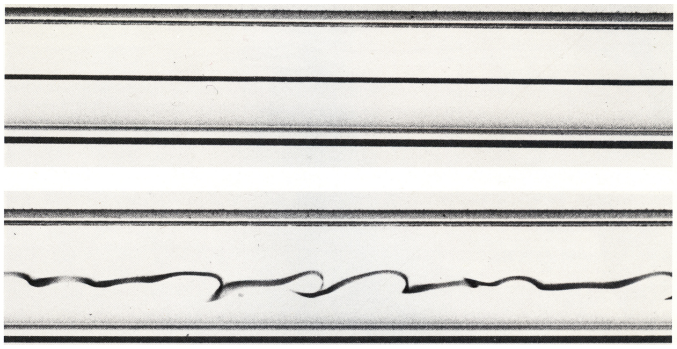

Figure 12: A visualization of an stable (laminar) and unstable flow in a pipe.

Another example of dynamical instability is fluid flow at high enough Reynolds numbers. Consider, for example, flow in a circular pipe. It is well-known that, for flow of a fluid of average speed U, density ρ, viscosity μ, and pipe radius R, the flow is laminar for small enough Reynolds number, where the Reynolds number is defined by $Re = \dfrac{\rho U R}{\mu}$. In this case, if the flow is fully-developed, the profile of the axial velocity is given by

$$u(r) = U_m\left(1 - \frac{r^2}{R^2}\right),$$

where U_m is the maximum flow speed. The perturbations to the flow, due, e.g., to vibrations in the system, roughness of the pipe walls, etc., will not grow and the flow will remain laminar. If

the Reynolds number is large enough, however, small perturbations will grow downstream, and the flow will ultimately become turbulence. Figure 12, taken from Van Dyke, *An Album of Fluid Motion*, gives two images of the flow of water through a glass tube; the flow is visualized by dye introduced upstream of the photograph. It is seen that in the top image, run at smaller Reynolds number, the flow is laminar; in the bottom image, however, run at much higher Reynolds number, the flow is unstable and oscillations in the flow are appearing.

An analogous type of instability is possible in the numerical scheme. Even if proper values are used as initial conditions, computer roundoff will produce a perturbation to the system. Again calling T_f the solution of the finite-volume equation, and call T_a the actual solution obtained from a calculation, then the error coming from the roundoff is given by

$$\epsilon = T_a - T_f \,. \tag{65}$$

This error should be smaller on computers and software that retains more significant digits in a calculation. For example, in a 32-bit calculation, about 7 to 8 significant digits are retained, while for a 64-bit calculation, 14 to 16 significant digits are retained. For a particular finite-volume scheme, if ϵ grows in time, the scheme is considered unstable, while if it decays or remains neutral in time, the scheme is considered stable. Note that in considering numerical instabilities it is important to distinguish two types of instabilities, (i) instabilities in the original dynamical system itself, e.g., turbulent flow, and (ii) instabilities in the numerical approximation to the solution.

We can obtain the equation for the error ϵ as follows. From Equation 65, we can write the computed solution T_a, containing roundoff error, as the sum of the exact solution to the finite-volume equation and the error due to roundoff:

$$T_a = T_f + \epsilon \,.$$

For example, ϵ at time $t = 0$ could be the roundoff error in the initial conditions, and we would like to determine how this error evolves in time. Plugging the expression for T_a into the explicit finite-difference equation, Equation 21, one obtains

$$\frac{T_{f_i}^{n+1} + \epsilon_i^{n+1} - T_{f_i}^{n+1} - \epsilon_i^n}{\Delta t} = \frac{k}{(\Delta y)^2}\{T_{f_{i+1}}^n + \epsilon_{i+1}^n - 2T_{f_i}^n - 2\epsilon_i^n + T_{f_{i-1}}^n - \epsilon_{i-1}^n\} \,, \tag{66}$$

or, since T_f is an exact solution to the explicit finite-volume equation, so that it can be cancelled out in Equation 66, then ϵ satisfies

$$\frac{\epsilon_i^{n+1} - \epsilon_i^n}{\Delta t} = \frac{k}{(\Delta y)^2}(\epsilon_{i+1}^n - 2\epsilon_i^n + \epsilon_{i-1}^n) \,. \tag{67}$$

Therefore, <u>in this case</u>, the error evolves in time in the same manner as the solution itself.

Let us assume that, at $t = 0$, there is some roundoff error, and determine whether it grows or decays. To determine the error evolution, use is made of a Fourier series. It is found that almost any function on an interval $0 \le y \le L$ can be represented as a summation of the form

$$a_0 + a_1 \exp\left\{\frac{2\pi i}{L}x\right\} + a_2 \exp\left\{\frac{2\pi i}{L}2x\right\} + \cdots + a_n \exp\left\{\frac{2\pi i}{L}nx\right\} + \cdots \,.$$

Since $e^{i\alpha} = \cos\alpha + i\sin\alpha$, this could also be considered as a series in terms of sines and cosines. This expansion is analogous to the frequency decomposition of a time-varying signal (using a time

series, e.g., in the same way you would think of a frequency distribution of a radio signal). The more terms in the series, the better the representation of the signal. The coefficients a_i are related to the original signal as follows:

$$\epsilon(y,0) = \sum_{n=-\infty}^{\infty} a_n \exp\left\{\frac{2\pi i}{L} nx\right\}, \tag{68}$$

where the a_n are found from the formula

$$a_n = \frac{1}{L}\int_{-L/2}^{L/2} \epsilon(y,0) \exp\left\{-\frac{2\pi i}{L} nx\right\} dx \tag{69}$$

The Fourier series theorem says that, under mild restrictions on $\epsilon(y,0)$, the series converges to $\epsilon(y,0)$, i.e., it gets to be a better and better representation of $\epsilon(y,0)$ as $n \to \infty$.

The amplitudes in Equation 68 depend on the function that the series is describing and, for the case of the time evolution of the error, are expected to change with time. Therefore we can write

$$\epsilon(y,t) = \sum_{n=-\infty}^{\infty} a_n(t) \exp\left\{\frac{2\pi i}{L} nx\right\}. \tag{70}$$

Now let's consider, without loss of generalization, an arbitrary member of this series, say $a_n(t)$, and determine how it behaves in time. Furthermore, let's assume that its growth or decay is of the form

$$a_n(t) = a_n(0)e^{\sigma t}. \tag{71}$$

With this time dependence, then

$$a_n(t+\Delta t) = a_n(0)e^{\sigma(t+\Delta t)} = e^{\sigma \Delta t} a_n(0) e^{\sigma t} = e^{\sigma \Delta t} a_n(t). \tag{72}$$

Therefore, $|a_n(t+\Delta t)| = |e^{\sigma \Delta t}||a_n(t)|$, so that

$$\frac{|a_n(t+\Delta t)|}{|a_n(t)|} = |e^{\sigma \Delta t}|. \tag{73}$$

Therefore if $|e^{\sigma \Delta t}| \leq 1$, the error solution will not grow from one time step to the next. We therefore need to find the properties of $e^{\sigma \Delta t}$ to find the stability properties of the explicit numerical scheme. In particular, we want to find $|e^{\sigma \Delta t}|$.

Let $\alpha_n = \frac{2\pi}{L} n$ (to simplify the writing), and assume that

$$a_n(t) e^{\frac{2\pi i}{L} ny} = A e^{\sigma t} e^{i\alpha_n y} \quad \text{where } A = a_n(0). \tag{74}$$

Plugging this into the finite-volume equation for ϵ, Equation 67, gives, for a time step going from t to $t + \Delta t$,

$$\begin{aligned} A e^{\sigma(t+\Delta t)} e^{i\alpha_n y} - A e^{\sigma t} e^{i\alpha_n y} &= \\ R(A e^{\sigma t} e^{i\alpha_n (y+\Delta y)} &- 2A e^{\sigma t} e^{i\alpha_n y} + A e^{\sigma t} e^{i\alpha_n (y-\Delta y)}, \end{aligned}$$

with $R = k\Delta t/(\Delta y)^2$ as above. Note that there will be an equation like this for each term in the Fourier series for $\epsilon(y,t)$, Equation 70. Dividing out the common term $A e^{\sigma t} e^{i\alpha_n y}$ gives

$$e^{\sigma \Delta t} - 1 = R(e^{i\alpha_n \Delta y} - 2 + e^{i\alpha_n \Delta y}), \text{ or}$$

$$e^{\sigma t} = 1 + 2R(\cos\beta - 1)\,,$$

with $\beta = \alpha_n \Delta y$, and using the identity,

$$\cos\beta = \frac{e^{i\beta} + e^{-i\beta}}{2}\,.$$

With the additional trigonometric identy

$$\sin^2\frac{\beta}{2} = \frac{1-\cos\beta}{2}\,,$$

the final expression is:

$$e^{\sigma\Delta t} = 1 - 4R\sin^2\frac{\beta}{2} \equiv G(R,\beta)\,. \tag{75}$$

The function $G(R,\beta)$ is often called the amplification factor. For a given R and a particular $\beta = \alpha_n\Delta y = \frac{2\pi}{L}n\Delta y$, it determines whether there is growth or decay in the n^{th} term in the series for ϵ.

For stability,

$$|e^{\sigma\Delta t}| = \underbrace{|1 - 4R\sin^2\frac{\beta}{2}|}_{|G(R,\beta)|} \le 1\,. \tag{76}$$

When is $|G| \le 1$? There are two possible cases, depending on the sign of G.

1. Suppose $G = 1 - \sin^2\frac{\beta}{2} > 0$. Then, since $|G| = G$, the condition is:

$$1 - 4R\sin^2\frac{\beta}{2} \le 1\,, \text{ or } 4R\sin^2\frac{\beta}{2} \ge 0\,.$$

   But since $R > 0$, then this condition is always satisfied.

2. Suppose $G = 1 - \sin^2\frac{\beta}{2} < 0$, or $|G| = -G$. Then

$$\begin{aligned} -(1 - 4R\sin^2\frac{\beta}{2}) &\le 1\,, \text{ or} \\ 4R\sin^2\frac{\beta}{2} &\le 2\,, \text{ or} \\ R\sin^2\frac{\beta}{2} &\le \frac{1}{2}\,. \end{aligned}$$

Remember that $\beta = \alpha_n\Delta y = \frac{2\pi}{L}n\Delta y$. Now

$$0 \le \sin^2\frac{\beta}{2} \le 1\,.$$

For stability this condition is to be satisfied for a particular value of β. In general, however, all wavelengths will be contained in the noise, so that we want to choose a value of R that holds for all possible values of the β's. Clearly since $\max(\sin^2\frac{\beta}{2}) = 1$ (the best chance for instability), then

$$R \le \frac{1}{2}\,, \text{ i.e., } \frac{k\Delta t}{(\Delta y)^2} \le \frac{1}{2} \tag{77}$$

will guarantee stability. On the other hand if $R > 1/2$, then there will undoubtably be values of β such that $|e^{\sigma\Delta t}| > 1$ and the noise will blow up, i.e., the scheme is unstable.

This condition means that, for the explicit scheme, if small errors are introduced into the scheme, they will grow exponentially if $R > 1/2$. This of course imposes a strong constraint on the time step Δt, given Δy. This analysis is called "von Neumann" or Fourier stability analysis.

Note that, even if you choose Δy and Δt to both be 'very small', unless the ratio $R = k\Delta t/(\Delta y)^2$ is kept less that $1/2$, the scheme is unstable. The instability is not in the original heat equations, but in this finite-volume approximation to it.

Implicit scheme. Next consider the same analysis, but applied to the implicit scheme given by Equation 46. Again selecting a term from the Fourier series of the form

$$a_n(t)\exp\left\{\frac{2\pi i}{L}ny\right\} = Ae^{\sigma t}e^{i\alpha_n y} \quad (\alpha_n = \frac{2\pi}{L}n)\,,$$

and plugging this into Equation 46 gives

$$\begin{aligned}
\frac{Ae^{\sigma(t+\Delta t)}e^{i\alpha_n y} - Ae^{\sigma t}e^{i\alpha_n y}}{\Delta t} &= \\
\frac{k}{2(\Delta y)^2}\{Ae^{\sigma(t+\Delta t)}e^{i\alpha_n(y+\Delta y)} &- 2Ae^{\sigma(t+\Delta t)}e^{i\alpha_n y} + Ae^{\sigma(t+\Delta t)}e^{i\alpha_n(y+\Delta y)} \\
+Ae^{\sigma t}e^{i\alpha_n(y+\Delta y)} &- 2Ae^{\sigma t}e^{i\alpha_n y} + Ae^{\sigma t}e^{i\alpha_n(y-\Delta y)}\}\,.
\end{aligned}$$

Dividing out the common term $Ae^{\sigma t}e^{i\alpha_n y}$, and using the definition of R gives

$$e^{\sigma\Delta t} - 1 = \frac{R}{2}\{e^{\sigma\Delta t}e^{i\alpha_n\Delta y} - 2e^{\sigma\Delta t} + e^{\sigma\Delta t}e^{-i\alpha_n\Delta y} + e^{i\alpha_n\Delta y} - 2 + e^{-i\alpha_n\Delta y}\}\,.$$

With $\beta = \alpha_n y$ and $\cos\beta = \dfrac{e^{i\beta} + e^{-i\beta}}{2}$, then

$$\begin{aligned}
e^{\sigma\Delta t} - 1 &= R\{e^{\sigma\Delta t}(\cos\beta - 1) + (\cos\beta - 1)\} \\
&= R\{e^{\sigma\Delta t)} + 1)(\cos\beta - 1)\}\,.
\end{aligned}$$

Solving for $e^{\sigma\Delta t}$,

$$e^{\sigma\Delta t}\{1 - R(\cos\beta - 1)\} = 1 + R(\cos\beta - 1)\,, \text{ or, finally}$$

$$e^{\sigma\Delta t} = \frac{1 + R(\cos\beta - 1)}{1 - R(\cos\beta - 1)} \equiv G(R,\beta)\,, \tag{78}$$

the amplification factor for the implicit method. Note that it is similar to that for the explicit case, except for the factor in the denominator.

Let $S = R(\cos\beta - 1)$. Then

$$e^{\sigma\Delta t} = \frac{1 + S}{1 - S}\,.$$

Note that $R > 0$ and $\cos\beta - 1 \le 0$, so that $S \le 0$. Also note that $G(0) = 1$, $G(-1) = 0$, and $G(S) \to -1$ from above as $S \to -\infty$. In Figure 13 a plot of $G(S)$ is given for the allowable values of S. It is seen that, over this range of S, $|G| \le 1$ so that $|e^{\sigma\Delta t}| \le 1$. Therefore the scheme is unconditionally stable, and R can be made arbitrarily large without the scheme going unstable. On the other hand, if R is made too large for a fixed Δy, then Δt will be too large, and the scheme will be inaccurate. Finally note the distinction between (i) numerical accuracy, and (ii) numerical stability. And note that the implicit scheme is much more flexible in the choice of Δt.

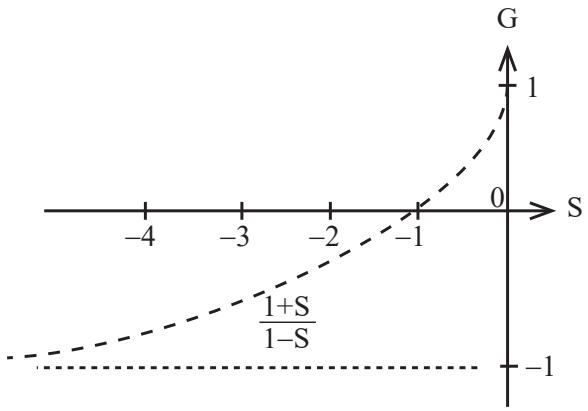

Figure 13: A plot of $\dfrac{1+S}{1-S}$ for $S \leq 0$.