Chapter 4

Preliminary Computational Techniques

4.1 Discretization

A numerical (approximate) solution to a given problem requires to consider carefully a number of steps through which one arrives at the solution. It is fairly obvious that it must be possible to formulate the problem appropriately in terms of mathematical equations which are usually partial differential equations. The steps summarized in Figure 4.1. Quantities of the differential equations must be represented in a discrete form. Then the resulting algebraic equations have to be solved numerically.

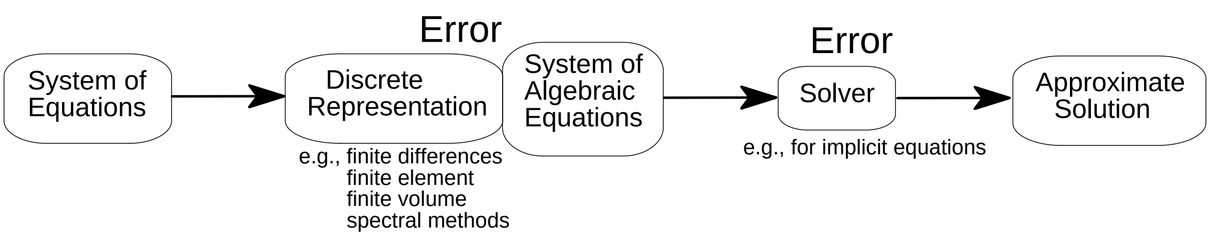


Figure 4.1: Steps to arrive at the numerical solution .

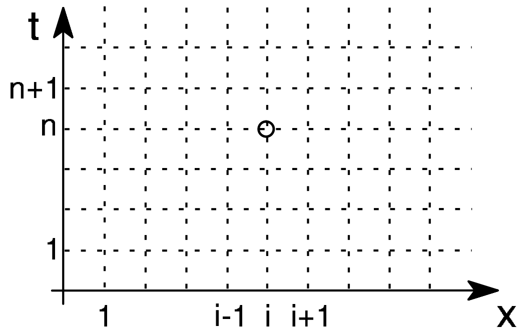
To illustrate the process consider the following equation

$$\frac{\partial \tilde{f}}{\partial t} = \alpha \frac{\partial^2 \tilde{f}}{\partial x^2} \tag{4.1}$$

which represents for instance one-dimensional heat conduction or diffusion. $\tilde{f}$ is the exact temperature or concentration of a medium and α is the diffusion coefficient. Boundary and initial conditions are

$$\tilde{f}(0,t) = b, \quad \tilde{f}(1,t) = d, \quad and$$
$$\tilde{f}(x,0) = f_0(x), \qquad for \qquad 0 \le x \le 1$$

The most obvious discretization is to store the temperature $\tilde{f}$ on a discrete spatial grid and to advance the temperature in discrete time increments. Assuming the spatial grid separation Δx, a total number of N_x grid point for the range $0 \leq x \leq 1$ and a time increment of Δt implies to evolve the temperature on a grid as shown in Figure.



This method implies to discretize the the diffusion equation 4.1 for instance in the following manner

$$\frac{f_i^{n+1} - f_i^n}{\Delta t} = \alpha \frac{f_{i-1}^n - 2f_i^n + f_{i+1}^n}{\Delta x^2} \tag{4.2}$$

with f_i^n representing the value of the temperature at the (i,n)th grid point. Note that the solution to f_i^n is different from the exact solution f' such that the prime is dropped from the numerical solution. The algebraic equation can be used to advance the the solution f_i^n explicit in time

$$f_i^{n+1} = f_i^n + \frac{\alpha \Delta t}{\Delta x^2} \left(f_{i-1}^n - 2f_i^n + f_{i+1}^n \right) \tag{4.3}$$

to determine the solution at time t_{n+1} and from there to t_{n+2} and so forth.

4.1.1 Spatial derivatives

Equation 4.2 provides already one example for the discretization of the second derivative. In the prior example the second derivative is evaluated at time t_n. However, another possibility would be to evaluated the derivative at time t_{n+1}. It will be shown later that there are many different ways to use finite differences to represent the second derivative operator. Examples for the discretization of the first order derivative $\partial f / \partial x$ are

$$\left.\frac{\partial f}{\partial x}\right|_{forward} = \frac{f_{i+1} - f_i}{\Delta x} \tag{4.4}$$

$$\left.\frac{\partial f}{\partial x}\right|_{centered} = \frac{f_{i+1} - f_{i-1}}{2\Delta x} \tag{4.5}$$

$$\left.\frac{\partial f}{\partial x}\right|_{backward} = \frac{f_i - f_{i-1}}{\Delta x} \tag{4.6}$$

Another discrete representation of derivatives can be obtained by approximation through a suitable set of orthonormal functions, i.e.,

$$f = \sum_{i=1}^{N} a_i(t)\phi_i(x) \tag{4.7}$$

the second derivatives can be expressed through appropriate recursion relations such that

$$\frac{\partial^2 \phi_i}{\partial x^2} = \sum_{j=1}^{N} b_{ij}\phi_j(x) \tag{4.8}$$

This expression has to be combined with the time derivative to update the coefficients $a_i(t)$. Such an approach is called a **spectral** method. Similar to the second derivative there are multiple ways to express the first or any other higher order derivative by finite differences or

4.1.2 Time derivatives

In the prior example we used a one sided difference approximation $\left(f_i^{n+1} - f_i^n\right)/\Delta t$ to represent the time derivative. Similarly we could have chosen $\left(f_i^{n+1} - f_i^{n-1}\right)/(2\Delta t)$. Again there are different ways to approximate this derivative and it will be illustrated that the particular choice combined with the choice for the spatial derivative will determine accuracy and stability of the numerical solution.

Using the one-sided time derivative formula in the case of the spectral method we obtain

$$\frac{f_i^{n+1} - f_i^n}{\Delta t} = \sum_{i=1}^{N} \frac{a_i^{n+1} - a_i^n}{\Delta t}\phi(x). \tag{4.9}$$

Combining this with the spatial derivative leads to relations such as

$$\frac{a_i^{n+1} - a_i^n}{\Delta t} = \sum_{j=1}^{N} c_{ij} a_j^n \tag{4.10}$$

In general if there is an equation involving the first order time derivative we may write this as

$$\frac{\partial \widetilde{f}}{\partial t} = L\widetilde{f}$$

where L is the derivative operator, for instance the second derivative in the prior example. After choosing an appropriate discretization the equations becomes

$$\frac{\partial f}{\partial t} = L_a f$$

where L_a is now the corresponding algebraic operator. We can now treat this equation as an ordinary differential equation and integrate this equation in time

$$f_i^{n+1} = f_i^n + \int_{t_n}^{t_{n+1}} [L_a f_j] \, dt$$

or with the Euler integration scheme

$$f_i^{n+1} = f_i^n + [L_a f_j]^n \, \Delta t \tag{4.11}$$

which is the same as derived in equation 4.3 if we use the same second derivative operator as before.

4.2 Approximation to derivatives

4.2.1 Taylor series expansion

To approximate derivatives it is intuitive to consider first Taylor series expansions for the value of $\tilde{f}$

$$\widetilde{f}_{i+1}^n = \sum_{m=0}^{\infty} \frac{\Delta x^m}{m!} \left[\frac{\partial^m \tilde{f}}{\partial x^m} \right]_i^n \tag{4.12}$$

and

$$\tilde{f}_i^{n+1} = \sum_{m=0}^{\infty} \frac{\Delta t^m}{m!} \left[\frac{\partial^m \tilde{f}}{\partial t^m} \right]_i^n . \tag{4.13}$$

The series can be truncated for any number of terms with the error determined by the next higher order term in the expansion. Note that this usually requires $\Delta x \ll 1$ and $\Delta t \ll 1$, i.e., smaller than the respective radius of convergence. his also implies that the function $\widetilde{f}$ is analytic. For second order accuracy we obtain

$$\widetilde{f}_{i+1}^n = \widetilde{f}_i^n + \Delta x \left[\frac{\partial \widetilde{f}}{\partial x} \right]_i^n + \frac{\Delta x^2}{2} \left[\frac{\partial^2 \widetilde{f}}{\partial x^2} \right]_i^n + O\left(\Delta x^3\right)$$

This truncation implies that the error of the approximation is smaller than $k\Delta x^3$ for a suitable finite value of k. Note that this is strictly true if the radius of convergence is larger than the choice for Δx.

Using the Taylor expansion one can estimate the error from finite differences

$$\left[\frac{\partial \widetilde{f}}{\partial x}\right]_i^n = \frac{\widetilde{f}_{i+1}^n - \widetilde{f}_i^n}{\Delta x} - 0.5\Delta x \left[\frac{\partial^2 \widetilde{f}}{\partial x^2}\right]_i^n + ... \tag{4.14}$$

which demonstrates that the forward differencing is accurate to $O(\Delta x)$. The additional terms in the finite difference approximation are the truncation error. Similar to the forward differencing the backward spatial difference also generates an error of $O(\Delta x)$.

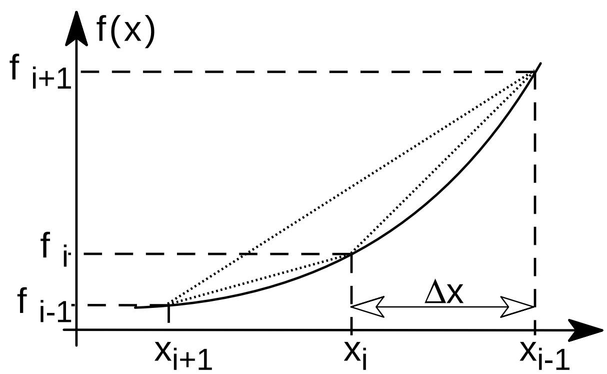


Figure 4.2: Finite difference approximation of $\partial f/\partial x$.

For the temporal derivative the same considerations apply, e.g.,

$$\left[\frac{\partial \widetilde{f}}{\partial t}\right]_i^n = \frac{\widetilde{f}_i^{n+1} - \widetilde{f}_i^n}{\Delta t} - 0.5\Delta t \left[\frac{\partial^2 \widetilde{f}}{\partial t^2}\right]_i^n + ... \tag{4.15}$$

implying an error of $O(\Delta t)$.

4.2.2 General technique

One can use the Taylor series expansion in a more systematic way to construct a finite difference approximation from a general expression for instance

$$\left[\frac{\partial \widetilde{f}}{\partial x}\right]_i^n = a\widetilde{f}_{i-1}^n + b\widetilde{f}_i^n + c\widetilde{f}_{i+1}^n + O(\Delta x^m) \tag{4.16}$$

Substituting the Taylor expansion yields

$$\begin{aligned}
a\widetilde{f}_{i-1}^{n} + b\widetilde{f}_{i}^{n} + c\widetilde{f}_{i+1}^{n} &= a\sum_{m=0}^{\infty}\frac{(\Delta x)^m}{m!}\left[\frac{\partial^m \widetilde{f}}{\partial x^m}\right]_i^n + b\tilde{f}_i^n + c\sum_{m=0}^{\infty}\frac{\Delta x^m}{m!}\left[\frac{\partial^m \widetilde{f}}{\partial x^m}\right]_i^n \\
&= (a+b+c)\,\widetilde{f}_i^n + (-a+c)\,\Delta x\left[\frac{\partial \widetilde{f}}{\partial x}\right]_i^n \\
&\quad + (a+c)\,\frac{\Delta x^2}{2}\left[\frac{\partial^2 \widetilde{f}}{\partial x^2}\right]_i^n + (-a+c)\,\frac{\Delta x^3}{6}\left[\frac{\partial^3 \widetilde{f}}{\partial x^3}\right]_i^n + \ldots
\end{aligned}$$

For the first order derivative the coefficients are

$$\begin{aligned}
a+b+c &= 0 \\
-a+c &= \frac{1}{\Delta x} \\
a+c &= 0
\end{aligned}$$

Here we take the first three terms of the expansion because we have chosen three coefficients a, b, and c. With $a = -c$ the solution is

$$\begin{aligned}
c = -a &= \frac{1}{2\Delta x} \\
b &= 0
\end{aligned}$$

Substituting these coefficient back into our general expression yields

$$\left[\frac{\partial \widetilde{f}}{\partial x}\right]_i^n = \frac{\widetilde{f}_{i+1}^n - \widetilde{f}_{i-1}^n}{2\Delta x} + \frac{\Delta x^2}{6}\left[\frac{\partial^3 \widetilde{f}}{\partial x^3}\right]_i^n + \ldots \tag{4.17}$$

such that the truncation error is $O\left(\Delta x^2\right)$. Closer inspection of the expansion shows that all even terms of the expansion are zero.

Similar to this example for the first order derivative one can use the general approach for the second order derivative which yields

$$\left[\frac{\partial^2 \widetilde{f}}{\partial x^2}\right]_i^n = \frac{\tilde{f}_{i+1}^n - 2\tilde{f}_i^n + \tilde{f}_{i-1}^n}{\Delta x^2} + O\left(\Delta x^2\right) \tag{4.18}$$

Note that this approach is also particularly helpful for nonuniform grids.

4.2.3 Asymmetric formula

Let us now consider an example which uses an asymmetric approximation for a second derivative.

$$\left[\frac{\partial^2 \widetilde{f}}{\partial x^2}\right]_i^n = a\widetilde{f}_i^n + b\widetilde{f}_{i+1}^n + c\widetilde{f}_{i+2}^n + O\left(\Delta x^m\right) \tag{4.19}$$

Using the Taylor expansion for each term yields

$$\begin{aligned}\left[\frac{\partial^2 \widetilde{f}}{\partial x^2}\right]_i^n &= (a+b+c)\,\widetilde{f}_i^n + (b+2c)\,\Delta x\left[\frac{\partial \widetilde{f}}{\partial x}\right]_i^n \\ &+ (b+4c)\,\frac{\Delta x^2}{2}\left[\frac{\partial^2 \widetilde{f}}{\partial x^2}\right]_i^n + (b+8c)\,\frac{\Delta x^3}{6}\left[\frac{\partial^3 \widetilde{f}}{\partial x^3}\right]_i^n + \ldots\end{aligned}$$

which gives the following relations for the coefficients

$$\begin{aligned} a+b+c &= 0 \\ b+2c &= 0 \\ b+4c &= \frac{2}{\Delta x^2} \end{aligned}$$

with $b = -2c$ one obtains

$$c = \frac{1}{\Delta x^2},\ b = -\frac{2}{\Delta x^2},\ \text{and } a = \frac{1}{\Delta x^2}$$

such that the approximation is

$$\left[\frac{\partial^2 \widetilde{f}}{\partial x^2}\right]_i^n = \frac{\widetilde{f}_{i+1}^n - 2\widetilde{f}_i^n + \widetilde{f}_{i-1}^n}{\Delta x^2} + O\left(\Delta x\right) \tag{4.20}$$

4.3 Accuracy

The accuracy of a finite difference approximation depends on a number of aspects. The prior discussions show that higher accuracy can be achieved by the inclusion of more terms in the Taylor expansion and by using preferably centered difference schemes rather then one sided differencing such as backward or forward differencing. In fact the one sided schemes above show only $O\left(\Delta x\right)$ accuracy. Accurate representation is possible only if $\widetilde{f}$ is a low order polynomial and in the case of the one sided example either liner or quadratic polynomials.

Table 4.1: Leading truncation error term for $d\tilde{f}/dx$.

Case	Difference formula	Leading truncation error term
3pt sym	$\left(\tilde{f}_{i+1} - \tilde{f}_{i-1}\right)/2\Delta x$	$\Delta_x^2 f_{xxx}/6$
Forw diff	$\left(\tilde{f}_{i+1} - \tilde{f}_{i}\right)/\Delta x$	$\Delta_x f_{xx}/2$
Back diff	$\left(\tilde{f}_{i} - \tilde{f}_{i-1}\right)/\Delta x$	$-\Delta_x f_{xx}/2$
3pt asym	$\left(-1.5\tilde{f}_{i} + 2\tilde{f}_{i+1} - 0.5\tilde{f}_{i+2}\right)/\Delta x$	$-\Delta_x^2 f_{xxx}/3$
5pt sym	$\left(\tilde{f}_{i-2} - 8\tilde{f}_{i-1} + 8\tilde{f}_{i+1} - \tilde{f}_{i+2}\right)/12\Delta x$	$-\Delta_x^4 f_{xxxxx}/30$

Table 4.2: Leading truncation error term for $d^2\tilde{f}/dx^2$.

Case	Difference formula	Leading error term
3pt sym	$\left(\tilde{f}_{i-1} - 2\tilde{f}_{i} + \tilde{f}_{i+1}\right)/\Delta x^2$	$\Delta_x^2 f_{xxxx}/12$
3pt asym	$\left(\tilde{f}_{i} - 2\tilde{f}_{i+1} + \tilde{f}_{i+2}\right)/\Delta x^2$	$\Delta_x f_{xxx}$
5pt sym	$\left(-\tilde{f}_{i-2} + 16\tilde{f}_{i-1} - 30\tilde{f}_{i} + 16\tilde{f}_{i+1} - \tilde{f}_{i+2}\right)/12\Delta x^2$	$\Delta_x^4 f_{xxxxxx}/90$

The truncation error leading terms for the varies difference approximations are shown in Table 4.1 for $d\tilde{f}/dx$ and in Table 4.2 for $d^2\tilde{f}/dx^2$. In these tables the indices x such as $\tilde{f}_{xxx}$ denote the degree of the x derivative.

It should also be noted that the 3pt asymmetric scheme yields an accuracy of order Δ_x^2 for the first order derivative while it yields only linear accuracy in the case of the second derivative. Thus higher order derivatives require usually to consider more terms in the finite difference approach.

One possibility to illustrate accuracy is to consider a general function such as $f = \sin x$. Table 4.3 shows the values and error terms associated with different difference approximations for $\tilde{f}$ at $x = 1$ and for $\Delta_x = 0.1$. The error E is computed as

$$E_{1st} = \frac{\partial f}{\partial x} - \left[\frac{\partial \tilde{f}}{\partial x}\right]_i^n \tag{4.21}$$

$$E_{2nd} = \frac{\partial^2 f}{\partial x^2} - \left[\frac{\partial^2 \tilde{f}}{\partial x^2}\right]_i^n \tag{4.22}$$

for a particular value x_0 and grid resolution Δ_x. The index n could actually be dropped if no time dependence is considered.

The results listed in the tables show that the approximation is better for higher order approximations involving more terms and for better centered approximations. The results can be understood

Table 4.3: Comparison of errors for $d\tilde{f}/dx$ for various difference schemes for $\tilde{f} = \sin x$ at $x = 1/2$.

Case	Difference formula	$\left.\frac{d\tilde{f}}{dx}\right\|_i$	Error	Leading error term
Exact	-	0.87757	-	-
3pt sym	$\left(\tilde{f}_{i+1} - \tilde{f}_{i-1}\right)/2\Delta x$	0.87612	-1.462×10^{-3}	-1.463×10^{-3}
Forw diff	$\left(\tilde{f}_{i+1} - \tilde{f}_{i}\right)/\Delta x$	0.85217	-2.541×10^{-2}	-2.397×10^{-2}
Back diff	$\left(\tilde{f}_{i} - \tilde{f}_{i-1}\right)/\Delta x$	0.90007	2.249×10^{-2}	2.397×10^{-1}
3pt asym	$\left(-1.5\tilde{f}_i + 2\tilde{f}_{i+1} - 0.5\tilde{f}_{i+2}\right)/\Delta x$	0.88038	2.795×10^{-3}	2.925×10^{-3}
5pt sym	$\left(\tilde{f}_{i-2} - 8\tilde{f}_{i-1} + 8\tilde{f}_{i+1} - \tilde{f}_{i+2}\right)/2\Delta x$	0.87758	-2.922×10^{-6}	-2.925×10^{-6}

in terms of the convergence of the Taylor series. Higher order expansion generates better accuracy. An important role plays the the radius of convergence. A small radius of convergence for the series expansion implies that the suitable values for Δ_x should be smaller for a good approximation.

Table 4.4: Comparison of errors for $d^2\tilde{f}/dx^2$ for various difference schemes for $\tilde{f} = \sin x$ at $x = 1/2$.

Case	Difference formula	$\left.\frac{d^2\tilde{f}}{dx^2}\right\|_i$	Error	Leading error term
Exact	-	-0.47943	-	-
3pt sym	$\left(\tilde{f}_{i-1} - 2\tilde{f}_i + \tilde{f}_{i+1}\right)/\Delta x^2$	-0.47903	3.994×10^{-4}	3.995×10^{-4}
3pt asym	$\left(\tilde{f}_{i} - 2\tilde{f}_{i+1} + \tilde{f}_{i+2}\right)/\Delta x^2$	0.56417	-8.475×10^{-2}	-8.776×10^{-2}
5pt sym	$\left(-\tilde{f}_{i-2} + 16\tilde{f}_{i-1} - 30\tilde{f}_i + 16\tilde{f}_{i+1} - \tilde{f}_{i+2}\right)/12\Delta x^2$	0.47943	5.322×10^{-7}	5.327×10^{-7}

Table 4.4 shows similar results for the second order derivative approximations for $\tilde{f} = \exp x$, at $x = 1$ and for $\Delta_x = 0.1$. Note that the radius of convergence has also an impact for higher order expansions because these can involve points such as x_{i+2} or x_{i+3} effectively assuming an expansion at $x + 2\Delta_x$ or $x + 3\Delta_x$.

The comparison of the real error with the leading error term shows that the real truncation error is well represented by the leading error term.

The different convergence for various difference schemes is illustrated in Figure (4.3) showing the total error for the first and second spatial derivatives as a function of grid spacing Δ_x.

convergence for backward, 3point symmetric, 3point asymmetric, and 5point symmetric differences for the first order derivative (for $\tilde{f} = \sin x$ at $x = 1$ and $\Delta_x = 0.1$) in Figure (a) and for 3point symmetric, 3point asymmetric, and 5point symmetric differences for the second order

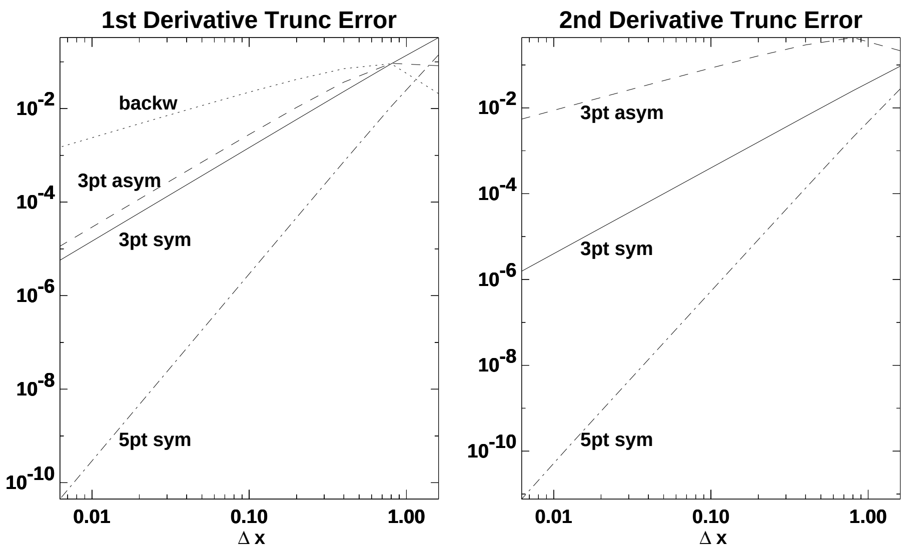


Figure 4.3: Total error for first and second derivative difference schemes as a function of Δ_x.

derivative in Figure (b). The results demonstrate that the symmetric 5point difference achieves the highest accuracy and fastest convergence to the accurate result.

High order vs low order approximation

The prior results show that higher order schemes are a better approximation to the actual result. However, how high does an approximation need to be for numerical purposes. It is important to consider that higher order approximations also require more computational effort. Therefore the choice of the approximation order is also one of computational efficiency.

A second consideration is how well a chosen grid resolves the actual solution. It requires a good resolution to realize the advantage of a high order scheme. However, a lower order scheme may already provide reasonable results for a good grid resolution. Note that the choice of Δ_x usually assumes that the corresponding solution varies slowly with Δ_x.

There are many fluid and gas dynamic problems which involve discontinuities. In such cases higher order approximations such as the 5point symmetric formula offer little advantage over a second order scheme such as the 3 point symmetric scheme. To illustrate this point consider the function $\tilde{f} = \tanh\left[kx\right]$ at $x = 0.04$. With increasing k the function approaches a step-like change at $x = 0$ as illustrated in Figure (4.4).

Figure (4.5) shows the discretization error for different discretization and $k = 5$ and 20 as a function of Δ_x. The figure shows the relative error, i.e.,

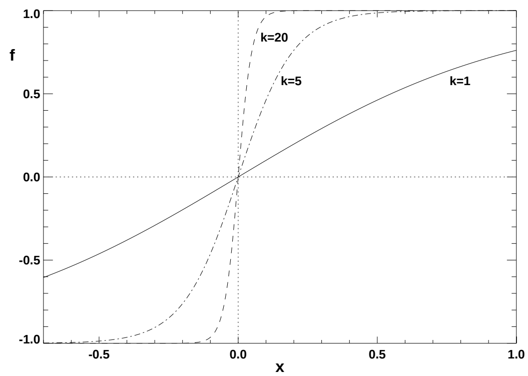


Figure 4.4: Illustration of $\tanh kx$ for $k = 1$, $k = 5$, and $k = 20$.

$$
\begin{aligned}
E_{r,\,1st} &= 1 - \left[\frac{\partial \tilde{f}}{\partial x}\right]_i^n \Big/ \frac{\partial f}{\partial x} \\
E_{r,\,2nd} &= 1 - \left[\frac{\partial^2 \tilde{f}}{\partial x^2}\right]_i^n \Big/ \frac{\partial^2 f}{\partial x^2}
\end{aligned}
$$

Especially for the first order derivative the results indicate no particular advantage of the 5point scheme for $k = 20$ and $\Delta_x \geq 0.025$.

There is a final important consideration regarding the basic discretization. First order schemes should always be avoided. In many cases problems require to use ore compute integral properties (such as conservation laws). An error which is linear in Δ_x will yield errors of order unity when it accumulates through integration.

In many cases it is sufficient to use a scheme of second order accuracy. This avoids the problem of linear schemes but will usually provide good approximations. In terms of numerical efficiency one often has the choice between a simple second order scheme and larger number of grid points or a more complex higher order approximation with less grid points (and thus larger grid spacing). Advantages and disadvantages should be carefully considered, i.e., numerical efficiency to achieve a particular accuracy. The complexity of higher order schemes usually requires more effort in terms of programming and programs are of higher complexity. Summary of arguments:

Higher order schemes provide higher accuracy for sufficiently small Δ_x

- Minimum accuracy should be second order
- High order schemes have only limited advantage for discontinuities
- High order schemes require more operations
- High order schemes require more complex programs
  - Boundary conditions or modifications to include other physics are more difficult for high order schemes (more error prone and larger programming effort)

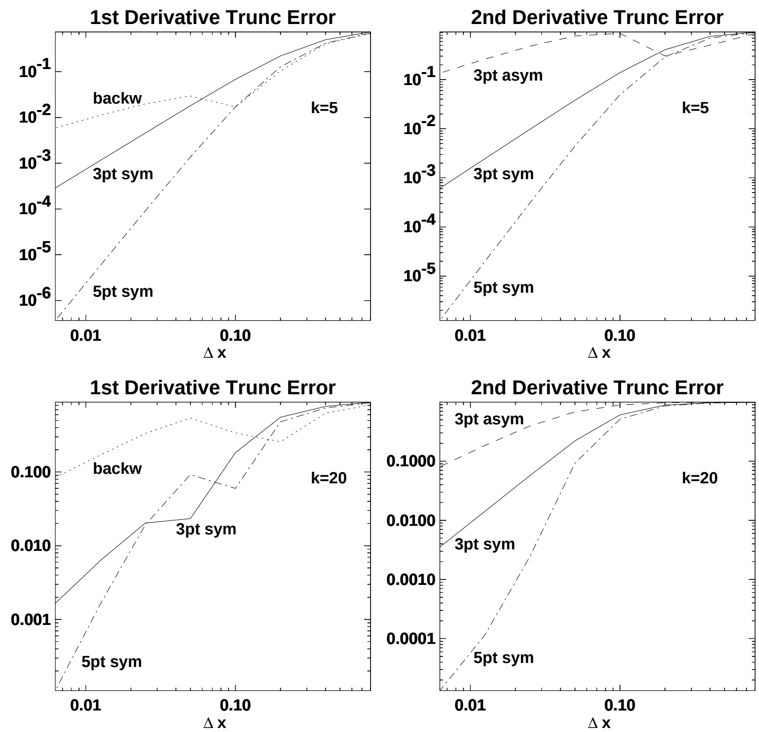


Figure 4.5: Error for first and second derivative finite difference approximations of $\tanh kx$ at $x = 0.04$.

4.4 Wave representation

In many cases of fluid dynamics information is transported by wave type motion. Thus it is important to understand how well different wavelength waves are represented by a certain grid resolution.

The effect of grid coarseness is illustrated in Figure (4.6). The two plots show the same values for g_i with i being the grid index along the x direction. While the function $g(x)$ in plot (a) shows many small scale oscillation the function $g(x)$ in (b) is smooth. Since the the g_i are identical the discretized function cannot distinguish between the two cases and the grid is too coarse too resolve the smaller scale structure of the plot in (a).

The Fourier representation of $g(x)$ is

$$g(x) = \sum_{k=-\infty}^{\infty} h_k \exp{(ikx)} \tag{4.23}$$

with the wave number k for a wavelength of $\lambda = 2\pi/k$. The amplitude h_k is given by

$$h_k = \int_0^{2\pi} g(x) \exp{(-ikx)}\, dx \tag{4.24}$$

For a finite number of discrete nodal values g_l such as given by a grid representation the Fourier representation is

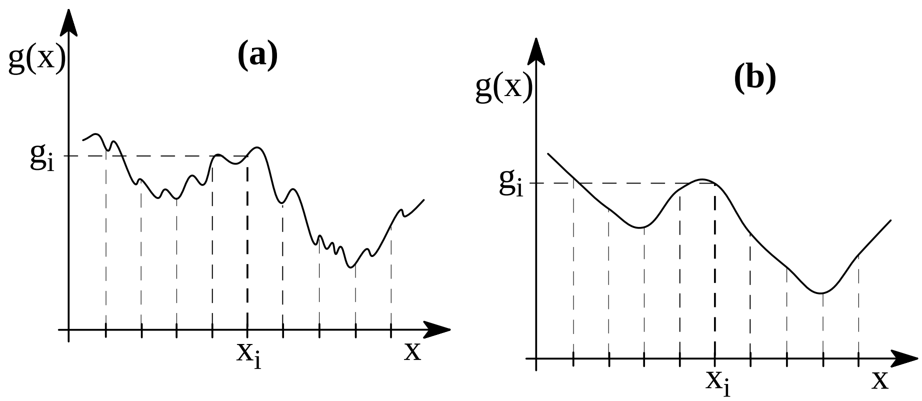


Figure 4.6: Discrete representation of $g(x)$. The discrete series g_i is identical for the two functions but the resolution is insufficient to represent the small scale structure in (a).

$$g_l = \sum_{k=1}^{N} h_k \exp\left(ikl\Delta x\right) \tag{4.25}$$

with the amplitudes

$$h_k = \Delta x \sum_{l=1}^{N} g_l \exp\left(-ikl\Delta x\right) \tag{4.26}$$

Thus a grid with N grid points limits the range of wavelength which can be resolved. In particular wavelength shorter than $\lambda = 2\Delta x$ cannot be resolved.

Finite difference approximation of waves

Consider a wave given by

$$f(x,t) = \cos\left[k\left(x - ut\right)\right] \tag{4.27}$$

where u is the propagation speed and k is the wavenumber. The exact spatial derivatives are

$$\frac{\partial f}{\partial x} = -k \sin\left[k\left(x - ut\right)\right] \tag{4.28}$$

$$\frac{\partial^2 f}{\partial x^2} = -k^2 \cos\left[k\left(x - ut\right)\right] \tag{4.29}$$

Finite difference of the first order derivative:

$$\begin{aligned}
\left.\frac{\partial f}{\partial x}\right|_i^n &= \frac{\tilde{f}_{i+1}^n - \tilde{f}_{i-1}^n}{2\Delta x} \\
&= \frac{1}{2\Delta x}\left\{\cos\left[k\left(x_i - ut_n\right) + k\Delta x\right] - \cos\left[k\left(x_i - ut_n\right) - k\Delta x\right]\right\} \\
&= \frac{-k\sin\left[k\left(x - ut\right)\right]\sin\left(k\Delta x\right)}{k\Delta x}
\end{aligned}$$

Thus the amplitude of the approximated first derivative to the actual amplitude is

$$R_{1,\,3pt} = \left.\frac{\partial f}{\partial x}\right|_i^n \Big/ \frac{\partial f}{\partial x} = \frac{\sin\left(k\Delta x\right)}{k\Delta x} \tag{4.30}$$

For the shortest possible wavelength of $\lambda = 2\Delta x$ with $k = \pi/\Delta x$ one obtains $R_{1,\,3pt}(\pi/\Delta x) = 0$. Similarly the 3pt central difference approximation for the second derivative yields

$$\frac{\tilde{f}_{i+1}^n - 2\tilde{f}_i^n + \tilde{f}_{i-1}^n}{\Delta x^2} = -k^2\left[\frac{\sin\left(k\Delta x/2\right)}{\left(k\Delta x/2\right)}\right]^2 \cos\left[k\left(x - ut\right)\right]$$

or an amplitude ratio of

$$R_{2,\,3pt} = \left[\frac{\sin\left(k\Delta x/2\right)}{\left(k\Delta x/2\right)}\right] \tag{4.31}$$

which for the shortest possible wavelength is $R_{2,\,3pt}(\pi/\Delta x) = 2/\pi$. For wavelength larger than $20\Delta x$ the amplitude ratio is larger than 0.98 for the first order derivative and better than 0.99 for the second derivative. It is easy to see that the approximated derivatives approximate the actual derivative for $\lambda \to \infty$ or for $\Delta x \to 0$.

For the corresponding 5pt approximations the amplitude ratios become

$$R_{1,\,5pt} = \left(\frac{4}{3} - \frac{1}{3}\cos\left(k\Delta x\right)\right)\frac{\sin\left(k\Delta x\right)}{k\Delta x} \tag{4.32}$$

$$R_{2,\,5pt} = \frac{4}{3}\left\{1 - \frac{1}{4}\cos^2\left(k\Delta x/2\right)\right\}\left[\frac{\sin\left(k\Delta x/2\right)}{\left(k\Delta x/2\right)}\right]^2 \tag{4.33}$$

The 5pt approximations yield somewhat more accurate results for short wavelength. However, the 5pt formula yield a much faster convergence of the amplitude ratio to unity. For instance for wavelength larger than $20\Delta x$ the amplitude ratios are better than 0.9999.

4.5 First example for a finite difference simulation

Following a first example to simulate the non-steady diffusion equation is presented. The basic equation is

$$\frac{\partial T'}{\partial t} = \alpha \frac{\partial^2 T'}{\partial x^2} \tag{4.34}$$

The application is represented schematically in Figure (4.7). Assume a metal rod which is insulate along it length and connected at both ends to a medium with fixed temperature. Initial the temperature of the rod is 0° C. At time $t = 0$ the ends are brought in contact with media of $T = 100°$C.

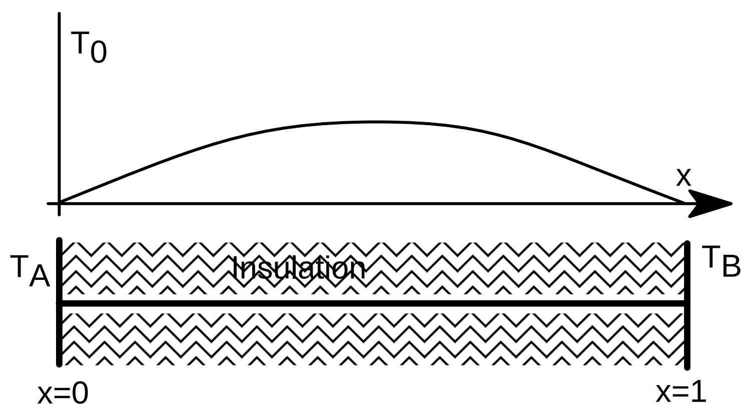


Figure 4.7: Unsteady heat conduction along a metal rod.

The goal is to find the variation of the temperature along the rod as a function of time. The numerical solution of the problem requires a numerical scheme as illustrated in Figure (4.8).

Initially the grid which is used for the finite difference scheme must be generated and the initial condition must be assigned to the variable T. The solution proceeds by solving the algebraic equations for all interior grid points, adjusting the boundary conditions for the two boundary grid points, test whether or not the final time is reached, and if not the time step must be updated. These steps are repeated up to the point the final time and solution has been reached.

The present example will use a forward time, centered space difference approximation equation (4.3). The corresponding scheme is

$$T_i^{n+1} = (1 - 2s)T_i^n + s\left(T_{i-1}^n + T_{i+1}^n\right) \tag{4.35}$$

with $s = \alpha \Delta t / \Delta x^2$. The numerical integration uses the parameter s as a parameterization for time which may appear odd at first sight. However, this parameterization can be understood by a suitable normalization of the diffusion equation. Measuring time in units of t_0 and distance in units of x_0 where x_0 is a typical length scale of the system (typical gradient, distance, etc) such that $t = t_0 \widehat{t}$ and $x = x_0 \widehat{x}$ one obtains

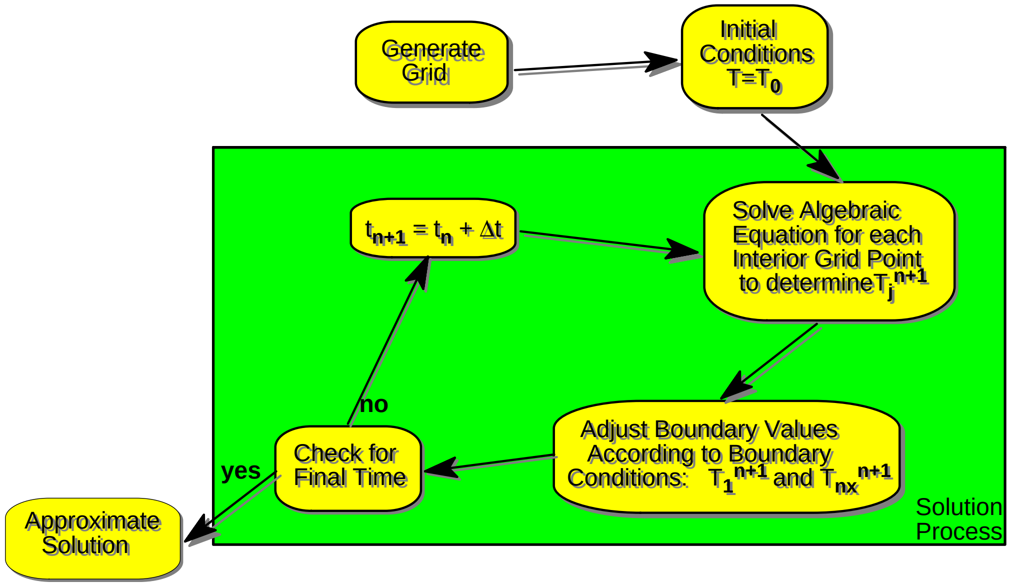


Figure 4.8: Schematic of the finite difference solution process.

$$\frac{\partial T'}{\partial \hat{t}} = \frac{\alpha t_0}{x_0^2} \frac{\partial^2 T'}{\partial \hat{x}^2}$$

In other words $t_0 = x_0^2/\alpha$ is a natural normalization for time. In this sense we can understand s to be a normalized time where the typical dimension is given by the grid distance.

The chosen problem has an analytic solution

$$T_{ex}(x_i, t_n) = 100 - \sum_{k=1}^{K_{max}} \frac{400}{(2k-1)\,\pi} \sin\left[(2k-1)\,\pi x_i\right] \exp\left[-\alpha\,(2k-1)^2\,\pi^2 t_n\right] \tag{4.36}$$

which is obtained by the separation of variables technique from the diffusion equation for the given initial and boundary conditions. The analytic solution can be used to determine the root mean square (RMS) error which is

$$RMS = \left[\left(\sum_{i=1}^{N_x} (T_i - T_{ex,i})^2\right) / N_x\right] \tag{4.37}$$

Parameters and variables used in the program are provided in Table 4.5. The program sim1.f provided on the course website uses two additional files one of which is an include file that declares all variables, constants, and common statements the other file is a parameter file which is read at the start of the program to provide all adjustable parameters. The web site will also provide an IDL file to read the data and to generate plots from the data generated in the simulation.

The program sim1.f provided on the course website uses two additional files one of which is an include file that declares all variables, constants, and common statements the other file is a

Table 4.5: Variables and parameters used in the program sim1.f

Parameter	Description
nx	number of gridpoints in x
x	x coordinate array
f,fold	dependent variable, here: temperature
xmin, xmax	boundary coordinates in x
ntmax	max number of time steps
tmax	max time
nt	time index
alpha, s	diffusion coefficient and time stepping parameter s
fini, f0	Initial temperature, normalization for temperature - here: 100
fb0, fb1	boundary conditions: here different for initial and later integration steps
nout	parameter to generate a data output every nout time steps
nterm(maxex)	number of terms in the exact solutio
fex	exact solution

parameter file which is read at the start of the program to provide all adjustable parameters. The web site will also provide an IDL file to read the data and to generate plots from the data generated in the simulation.

The output generated by the program sim1.f includes an ASCII file which summarizes program parameters and provides an tabulated version of the temperature at selected times.

```
 FTCS scheme (explicit)

 nx =  21   ntmax =  500     tmax = 6000.00  xmin =  0.0  xmax =  1.0
  s =0.500  alpha = 0.100E-04  dt = 125.000    dx =0.0500 nout =    8
 f0 = 100.0  fini =  0.00  fxmin = 1.00 fxmax = 1.00 fxmin0 = 0.50 fxmax0 = 0.50
 No of terms in exact solution (nterm) =   20

 Time =      0.
   T:  50.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00
   T:   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00  50.00
 Time =   1000.
   T: 100.00  72.66  48.05  28.91  15.23   7.03   2.73   0.78   0.20   0.00   0.00
   T:   0.00   0.20   0.78   2.73   7.03  15.23  28.91  48.05  72.66 100.00
 Time =   2000.
   T: 100.00  80.36  61.81  45.45  31.70  21.01  13.11   7.73   4.38   2.55   2.01
   T:   2.55   4.38   7.73  13.11  21.01  31.70  45.45  61.81  80.36 100.00
 Time =   3000.
   T: 100.00  83.89  68.39  54.15  41.53  30.90  22.31  15.82  11.31   8.66   7.80
   T:   8.66  11.31  15.82  22.31  30.90  41.53  54.15  68.39  83.89 100.00
 Time =   4000.
   T: 100.00  86.05  72.53  59.87  48.39  38.41  30.07  23.54  18.84  16.03  15.09
   T:  16.03  18.84  23.54  30.07  38.41  48.39  59.87  72.53  86.05 100.00
 Time =   5000.
   T: 100.00  87.62  75.57  64.21  53.80  44.61  36.84  30.67  26.17  23.45  22.54
   T:  23.45  26.17  30.67  36.84  44.61  53.80  64.21  75.57  87.62 100.00
 Time =   6000.
   T: 100.00  88.89  78.06  67.80  58.37  50.00  42.87  37.17  33.01  30.48  29.63
   T:  30.48  33.01  37.17  42.87  50.00  58.37  67.80  78.06  88.89 100.00

No of terms in exact solution (nterm) =   20
 time =   6000.
  T: 100.00  88.89  78.07  67.82  58.41  50.06  42.96  37.28  33.14  30.62  29.78
```

```
T:  30.62  33.14  37.28  42.96  50.06  58.41  67.82  78.07  88.89 100.00
rms dif =  0.8574E-01
```

The second output file is a binary file which contains the essential program parameters and the solution at selected times. This file serves as input for other graphics software to generate plots of the temperature at the certain times. The ASCII file serves the purpose to provide a fast overview of program parameters and essential results. The binary file is easier to read for graphics software and provide a more compact and more accurate (machine accuracy) of the results. The provided IDL program sim1.pro generates for the present example the plot in Figure (4.9).

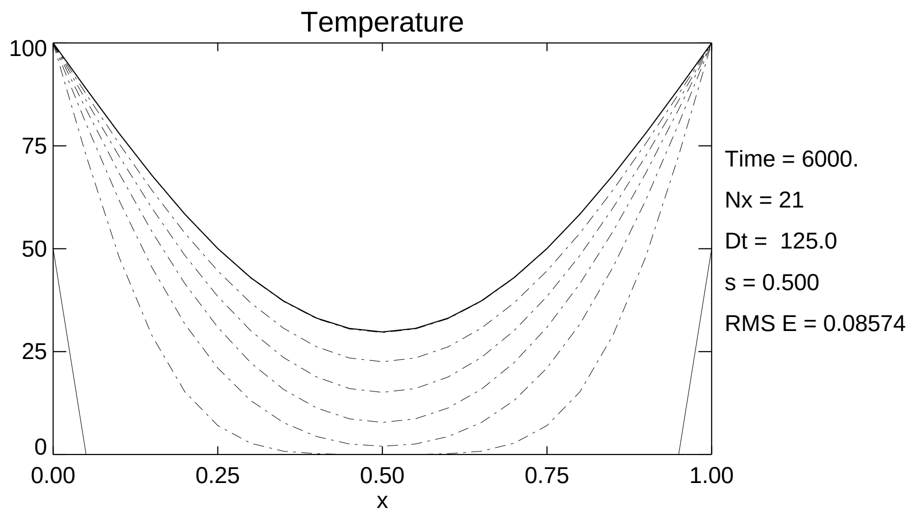


Figure 4.9: Numerical solution of the diffusion equation for specific parameters, boundary, and initial conditions.

The dotted lines represent solutions at intermediate times and the solid line shows the exact final (specified by tmax or ntmax) solution for the temperature profile.