Finite difference method for solving differential equations

Sergiy Bubin

Department of Physics
Nazarbayev University

Discretization in space

A **grid** consists of a finite set of points in space and/or time. It provides a way to obtain a discrete sampling of continuous quantities.

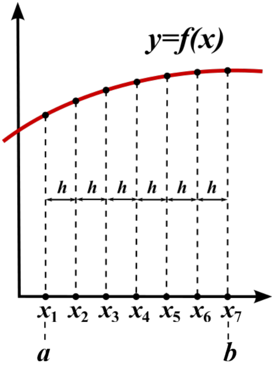
1D grid:

$$x_i = a + (i-1)h, \qquad i = 1, 2, \ldots, N. \tag{1}$$

In some applications it is convenient to specify N instead. In this case

$$h = \frac{b-a}{N-1}. \tag{2}$$

Here a and b are the end points of the simulation interval.



Finite differences

Recall how we can approximate the first derivative of a function.

Forward difference:

$$f'(x) = \frac{f(x+h) - f(x)}{h}, \qquad h \to 0. \tag{3}$$

We can also use the backward difference:

$$f'(x) = \frac{f(x) - f(x-h)}{h}, \qquad h \to 0. \tag{4}$$

The central difference gives a more accurate approximation:

$$f'(x) = \frac{f(x+h) - f(x-h)}{2h}, \qquad h \to 0. \tag{5}$$

Multipoint expressions for finite differences

Yet, we can do a better job if we use function values at more points.

Example: 5 points centered around x

$$\begin{aligned}
f(x-2h) &= f(x)-f'(x)2h+\frac{1}{2}f''(x)(2h)^2-\frac{1}{6}f'''(x)(2h)^3+\frac{1}{24}f''''(x)(2h)^4 \\
f(x-h) &= f(x)-f'(x)h+\frac{1}{2}f''(x)h^2-\frac{1}{6}f'''(x)h^3+\frac{1}{24}f''''(x)h^4, \\
f(x) &= f(x), \\
f(x+h) &= f(x)+f'(x)h+\frac{1}{2}f''(x)h^2+\frac{1}{6}f'''(x)h^3+\frac{1}{24}f''''(x)h^4, \\
f(x+2h) &= f(x)+f'(x)2h+\frac{1}{2}f''(x)(2h)^2+\frac{1}{6}f'''(x)(2h)^3+\frac{1}{24}f''''(x)(2h)^4.
\end{aligned}$$

Multipoint expressions for finite differences

That gives us a system of 5 linear equations for the derivatives of f:

$$\underbrace{\begin{pmatrix} f(x-2h) \\ f(x-h) \\ f(x) \\ f(x+h) \\ f(x+2h) \end{pmatrix}}_{\mathbf{b}} = \underbrace{\begin{pmatrix} 1 & -2 & 2 & -4/3 & 2/3 \\ 1 & -1 & 1/2 & -1/6 & 1/24 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1/2 & 1/6 & 1/24 \\ 1 & 2 & 2 & 4/3 & 2/3 \end{pmatrix}}_{A} \underbrace{\begin{pmatrix} f(x) \\ f'(x)h \\ f''(x)h^2 \\ f'''(x)h^3 \\ f''''(x)h^4 \end{pmatrix}}_{\mathbf{y}} \tag{6}$$

or

$$\mathbf{b} = A\mathbf{y}. \tag{7}$$

Multipoint expressions for finite differences

By solving the system of linear equations we get:

$$
\begin{aligned}
f(x) &= f(x), \\
f'(x)h &= \frac{1}{12}f(x-2h)-\frac{2}{3}f(x-h)+\frac{2}{3}f(x+h) - \frac{1}{12}f(x+2h), \\
f''(x)h^2 &= -\frac{1}{12}f(x-2h)+\frac{4}{3}f(x-h)-\frac{5}{2}f(x)+\frac{4}{3}f(x+h)-\frac{1}{12}f(x+2h), \\
f'''(x)h^3 &= -\frac{1}{2}f(x-2h)+f(x-h)-f(x+h)+\frac{1}{2}f(x+2h), \\
f''''(x)h^4 &= f(x-2h)-4f(x-h)+6f(x)-4f(x+h)+f(x+2h),
\end{aligned}
\tag{8}
$$

Multipoint expressions for finite differences

In matrix form it looks as follows:

$$\underbrace{\begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 1/12 & -2/3 & 0 & 2/3 & -1/12 \\ -1/12 & 4/3 & -5/2 & 4/3 & -1/12 \\ -1/2 & 1 & 0 & -1 & 1/2 \\ 1 & -4 & 6 & -4 & 1 \end{pmatrix}}_{A^{-1}} \underbrace{\begin{pmatrix} f(x-2h) \\ f(x-h) \\ f(x) \\ f(x+h) \\ f(x+2h) \end{pmatrix}}_{\mathbf{b}} = \underbrace{\begin{pmatrix} f(x) \\ f'(x)h \\ f''(x)h^2 \\ f'''(x)h^3 \\ f''''(x)h^4 \end{pmatrix}}_{\mathbf{y}} \tag{9}$$

or

$$A^{-1}\mathbf{b} = \mathbf{y}. \tag{10}$$

Multipoint expressions for finite differences

We can generalize this approach to higher orders:

$$f(x+nh) = \sum_{i=0}^{\infty} \frac{1}{i!} f^{(i)}(x)(nh)^i, \quad n = -m, \ldots, m, \tag{11}$$

where we truncate the series at m-th order

Multipoint expressions for finite differences

Using the notation

$$y_j = f^{(j-1)}(x)h^{j-1}, \quad b_k = f(x+(k-m-1)h), \quad j,k = 1,\ldots,2m+1, \tag{12}$$

and

$$A_{kj} = \frac{(k-m-1)^{j-1}}{(j-1)!} \tag{13}$$

By inverting A we obtain the desired derivatives:

$$f^{(j-1)}(x) = \frac{1}{h^{j-1}} \sum_{k=1}^{2m+1} C_{jk} f(x+(k-m-1)h), \tag{14}$$

where

$$C_{jk} = (A^{-1})_{jk}. \tag{15}$$

Solving differential equations and eigenvalue problems on a grid

Finite difference formulae allow to represent differential operators on a grid. Thus, we can reduce a linear differential equation and the boundary/initial conditions to a system of N linear algebraic equations (N is the number of grid points). By solving this system of linear equations we can obtain an approximation (given on the same grid) to the solution of the original problem.

Note: This is actually not a terribly efficient approach to solve linear differential equations in 1D. However, it can be easily generalized to the case of 2D and 3D partial differential equations, where it often works beautifully.

Let us now see how it works...

The Schrödinger equation in 1D

With the discretization of differential operators we can also solve eigenvalue problems. For example we solve the Schrödinger equation:

$$-\frac{\hbar^2}{2m}\frac{d^2\psi(x)}{dx^2} + V(x)\psi(x) = E\psi(x) \tag{16}$$

Here $\hbar$, and m are physical constants, $V(x)$ is the potential, and E is an energy eigenvalue.

Solving the Schrödinger equation in 1D using three-point differences

$$\psi''(x) = \frac{\psi(x-h) - 2\psi(x) + \psi(x+h)}{h^2}. \tag{17}$$

Define

$$\psi_i \equiv \psi(i) \equiv \psi(x_i) = \psi(a + ih), \qquad V_i = V(i) = V(x_i). \tag{18}$$

The Schrödinger equation can be written as

$$-\frac{\hbar^2}{2m}\frac{\psi_{i-1} - 2\psi_i + \psi_{i+1}}{h^2} + V_i\psi_i = E\psi_i, \quad i = 1, \dots, N. \tag{19}$$

There are two exterior points, x_{-1} and x_N. For bound states

$$\psi_0 = \psi_{N+1} = 0. \tag{20}$$

Solving the Schrödinger equation in 1D using three-point differences

Equation (19) can be rewritten as an eigenvalue problem:

$$Hc = Ec, \tag{21}$$

where

$$H_{ij} = \begin{cases} \frac{\hbar^2}{2m}\frac{2}{h^2} + V_i, & i = j \\ -\frac{\hbar^2}{2m}\frac{1}{h^2}, & i = j \pm 1 \\ 0, & \text{otherwise} \end{cases} \tag{22}$$

and

$$c = \begin{pmatrix} \psi_1 \\ \psi_2 \\ \vdots \\ \psi_N \end{pmatrix}. \tag{23}$$

Solving the Poisson equation in 2D using three-point differences

The Poisson equation:

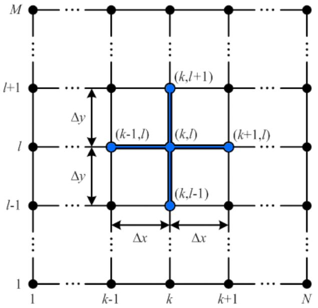
$$\nabla^2 \phi(x, y) = -4\pi\rho(x, y), \tag{24}$$

where ρ is a given charge distribution and ϕ is an unknown function (electrostatic potential) that is subject to boundary conditions

$$\phi|_{\text{boundary}} = f(x, y).$$

When discretized using the simplest three-point approximation, it has the following form:

$$\frac{\phi(k+1,l)-2\phi(k,l)+\phi(k-1,l)}{\Delta x^2} + \frac{\phi(k,l+1)-2\phi(k,l)+\phi(k,l-1)}{\Delta y^2} = -4\pi\rho(k,l). \tag{25}$$



Solving the Poisson equation in 2D using three-point differences

If we assume that our simulation volume has a shape of a rectangular box, the boundary conditions on the surface of the box are

$$\phi(0, l) = g_{x0}(l), \tag{26}$$

$$\phi(N+1, l) = g_{xN}(l), \tag{27}$$

$$\phi(k, 0) = g_{y0}(k), \tag{28}$$

$$\phi(k, M+1) = g_{yM}(k), \tag{29}$$

Solving the Poisson equation in 2D using three-point differences

The finite difference Poisson equation with these boundary conditions can be rewritten as a system of $P = N \times M$ linear equations

$$Ax = b, \tag{30}$$

where the unknown vector x is

$$x = \begin{pmatrix} \phi(1,1) \\ \phi(2,1) \\ \vdots \\ \phi(N,1) \\ \phi(1,2) \\ \phi(2,2) \\ \vdots \\ \phi(N-1,M) \\ \phi(N,M) \end{pmatrix}. \tag{31}$$

Solving the Poisson equation in 2D using three-point differences

Matrix A lists coefficients of equation (25) according to the map

$$p = N(l-1) + k, \qquad k = 1 \dots N, \quad l = 1 \dots M, \tag{32}$$

so that index p ranges from 1 to $P = N \times M$.
Vector b is given by

$$b_p = \begin{cases} -4\pi\rho(k,l) - g_{x0}(l)/\Delta x^2, & k = 1, \\ -4\pi\rho(k,l) - g_{xN}(l)/\Delta x^2, & k = N, \\ -4\pi\rho(k,l) - g_{y0}(l)/\Delta y^2, & l = 1, \\ -4\pi\rho(k,l) - g_{yM}(l)/\Delta y^2, & l = M, \\ -4\pi\rho(k,l), & \text{otherwise.} \end{cases} \tag{33}$$