Lecture 2: Numerical Methods for Differentiations and Integrations

As we have discussed in Lecture 1 that numerical simulation is a set of carefully planed numerical schemes to solve an initial value problem numerically. Let us consider the following three types of differential equations.

$$\frac{dy(t)}{dt} = f(t) \tag{2.1}$$

$$\frac{dy(t)}{dt} = f(y,t) \tag{2.2}$$

$$\frac{\partial y(x,t)}{\partial t} = f(t, y, \frac{\partial y}{\partial x}, \frac{\partial^2 y}{\partial x^2}, ..., \int y dx, ...) \tag{2.3}$$

The numerical methods for time integration of these equations will be discussed in the Lecture 3. Before we conduct the time integration, we need to determine the differentiations $\frac{\partial y}{\partial x}$, $\frac{\partial^2 y}{\partial x^2}$, … and the integrations $\int y dx,...$ on the right-hand side of the equation (2.3) at each grid point.

In this section, we are going to discuss the following three types of numerical methods, which are commonly used in spatial differentiations and integrations.

1. Finite Differences (based on Taylor series expansion)
2. FFT (Fast Fourier Transform)
3. Cubic Spline

2.1. Finite Differences

For convenience, we shall use the following notation in the rest of this lecture notes.

$$f_{ijk}^n = f(x = i\Delta x, y = j\Delta y, z = k\Delta z, t = n\Delta t) = f(x_i, y_j, z_k, t^n)$$

For a given tabulate function

$$\begin{array}{cccccc} i & : & 1 & 2 & ... & N \\ x_i & : & x_1 & x_2 & ... & x_N \\ f_i & : & f_1 & f_2 & ... & f_N \end{array}$$

the finite-difference expression of the n-th order derivatives of the given function f can be obtained from the Taylor series expression of f. Table 2.1 lists examples of

finite-difference expression of $[df/dx]_{x=x_i}$, $[d^2f/dx^2]_{x=x_i}$, and $[d^3f/dx^3]_{x=x_i}$, where the central-difference expressions are of the first-order accuracy, while the forward-difference and backword-difference expressions are of the zeroth-order accuracy. Complete derivations of the higher-order finite-difference schemes are given in the Appendix B.
Table 2.2 lists examples of the finite-difference expressions of spatial integrations $y(x) = \int_{x_i}^{x_i+\Delta x} f(x)dx$. Complete derivations of the higher-order finite-difference scheme are given in the Appendix C.1.

Table 2.1. The numerical differentiations based on finite difference method

Derivatives	Central Difference (First-order scheme)	Forward Difference (Zeroth-order scheme)	Backward Difference (Zeroth-order scheme)
$\left.\frac{df}{dx}\right\|_{x=x_i}$	$\delta f_i = \frac{f_{i+1} - f_{i-1}}{2\Delta x}$	$\Delta f_i = \frac{f_{i+1} - f_i}{\Delta x}$	$\nabla f_i = \frac{f_i - f_{i-1}}{\Delta x}$
$\left.\frac{d^2f}{dx^2}\right\|_{x=x_i}$	$\delta^2 f_i = \frac{f_{i+1} - 2f_i + f_{i-1}}{(\Delta x)^2}$	$\Delta^2 f_i = \frac{f_{i+2} - 2f_{i+1} + f_i}{(\Delta x)^2}$	$\nabla^2 f_i = \frac{f_i - 2f_{i-1} + f_{i-2}}{(\Delta x)^2}$
$\left.\frac{d^3f}{dx^3}\right\|_{x=x_i}$	$\delta^3 f_i =$ $\frac{f_{i+2} - 2f_{i+1} + 2f_{i-1} - f_{i-2}}{2(\Delta x)^3}$	$\Delta^3 f_i =$ $\frac{f_{i+3} - 3f_{i+2} + 3f_{i+1} - f_i}{(\Delta x)^3}$	$\nabla^3 f_i =$ $\frac{f_i - 3f_{i-1} + 3f_{i-2} - f_{i-3}}{(\Delta x)^3}$

Exercise 2.1.

(a) Please prove that the central-difference expressions shown in Table 2.1 are of first-order accuracy.

(b) Please prove that the forward-difference and backward-difference expressions shown in Table 2.1 are of zeroth-order accuracy.

Exercise 2.2.

Determine df/dx, and d^2f/dx^2 of a given analytical function $f(x)$ numerically based on the first-order, the third-order, and the fifth-order finite-difference expressions listed in the Appendix B. The differences between the numerical solutions and the analytical solutions are called the numerical errors. Determine the numerical errors of a given grid size Δx. Show (or plot) that the numerical error is a function of position and also a function of Δx.

Table 2.2. The spatial integrations based on finite difference method

	$\frac{dy(x)}{dx} = f(x), \quad h = \Delta x$
1st order integration	$y_{i+1} = y_i + hf_i + O(h^2 f^{(1)})$
2nd order integration Trapezoidal rule	$y_{i+1} = y_i + h\frac{f_{i+1} + f_i}{2} + O(h^3 f^{(2)})$
4th order integration Simpson's rule	$y_{i+1} = y_i + h(\frac{1}{6} f_i + \frac{4}{6} f_{i+(1/2)} + \frac{1}{6} f_{i+1}) + O(h^5 f^{(4)})$
4th order integration Simpson's $3/8$ rule	$y_{i+1} = y_i + \frac{h}{3}(\frac{3}{8} f_i + \frac{9}{8} f_{i+\frac{1}{3}} + \frac{9}{8} f_{i+\frac{2}{3}} + \frac{3}{8} f_{i+1}) + O(h^5 f^{(4)})$

Exercise 2.3.

Use the first-order, the second-order, and the forth-order integration expressions listed in Table 2.2 to determine $y(x = \pi / 4)$ with $dy(x) / dx = \cos(x)$ and boundary condition $y(x = 0) = 0$. Determine the numerical errors in your results. Compare the numerical errors obtained from different spatial integration expressions.

2.2. FFT (Fast Fourier Transform)

A function can be expanded by a complete set of sine and cosine functions. In the Fast Fourier Transform, the sine and cosine tables are calculated in advance to save the CPU time of the simulation.

For a periodic function f, one can use FFT to determine its spatial *differentiations* and *integrations,* i.e.,

$$\frac{df}{dx} = FFT^{-1}\{ik[FFT(f)]\}$$

$$\int f\, dx = FFT^{-1}\{\frac{1}{ik}[FFT(f)]\} \quad \text{for} \quad k > 0.$$

Exercise 2.4.

Use an FFT subroutine to determine the first derivatives of a periodic analytical function f. Determine the numerical errors in your results.

Exercise 2.5.

Use an FFT subroutine to determine the first derivatives of a non-periodic analytical function f. Determine the numerical errors in your results.

2.3. Cubic Spline

A tabulate function can be fitted by a set of piece-wise continuous functions, in which the first and the second derivatives of the fitting functions are continuous at each grid point. One need to solve a tri-diagonal matrix to determine the piece-wise continuous cubic spline functions. The inversion of the tri-diagonal matrix depends only on the position of grid points. Thus, for simulations with fixed grid points, one can evaluate the inversion of the tri-diagonal matrix in advance to save the CPU time of the simulation.

For a non-periodic function f, it is good to use the cubic spline method to determine its spatial *differentiations* and *integrations* at each grid point. Results of the spatial *differentiations* obtained from the cubic spline show the same order of accuracy as the results obtained from the fifth order finite differences scheme.

The piece-wise continuous function in the cubic spline can be written in the following form.

$$f(x_k \le x \le x_{k+1}) = \frac{f(x_k)(x-x_{k+1})}{(x_k - x_{k+1})} + \frac{f(x_{k+1})(x-x_k)}{(x_{k+1}-x_k)} + [a_k \frac{(x-x_k)}{(x_{k+1}-x_k)} + b_k]\frac{(x-x_k)(x-x_{k+1})}{(x_{k+1}-x_k)^2}$$

The constants $\{a_k, b_k, \text{for } k = 1 \to n-1\}$ are chosen such that the matching conditions for cubic spline can be fulfilled, i.e.,

$$\left.\frac{df(x_{k-1} \le x \le x_k)}{dx}\right|_{x=x_k} = \left.\frac{df(x_k \le x \le x_{k+1})}{dx}\right|_{x=x_k}$$

and

$$\left.\frac{d^2 f(x_{k-1} \le x \le x_k)}{dx^2}\right|_{x=x_k} = \left.\frac{d^2 f(x_k \le x \le x_{k+1})}{dx^2}\right|_{x=x_k}$$

One can obtain the following two types of recursion formula

$$f'(x_{k-1}) + f'(x_k)[2 + 2(\frac{h_{k-1}}{h_k})] + f'(x_{k+1})(\frac{h_{k-1}}{h_k}) = 3f_0'(x_{k-1}) + 3f_0'(x_k)(\frac{h_{k-1}}{h_k})$$

$$f''(x_{k-1}) + f''(x_k)[2 + 2(\frac{h_k}{h_{k-1}})] + f''(x_{k+1})(\frac{h_k}{h_{k-1}}) = \frac{6}{h_{k-1}}[f_0'(x_k) - f_0'(x_{k-1})]$$

where $f_0'(x_k) = \dfrac{f(x_{k+1}) - f(x_k)}{x_{k+1} - x_k}$ and $h_k = x_{k+1} - x_k$.

Detail derivations of the Cubic Spline with different boundary conditions are given in Appendix A.

Exercise 2.6.

Use a Cubic Spline subroutine to determine the first derivatives of an analytical function f . Determine the numerical errors in your results.

References

Hildebrand, F. B., *Advanced Calculus for Applications, 2nd edition,* Prentice-Hall, Inc., Englewood, Cliffs, New Jersey, 1976.

Press, W. H., B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes (in C or in FORTRAN and Pascal), Cambridg*e University Press, Cambridge, 1988.

System/360 Scientific Subroutine Package Version III, Programmer's Manual, 5th edition, IBM, New York, 1970.