—Fundamentals of FD methods

We will be computing derivatives using sampled values of the field variables. First we must introduce and discuss the concepts of numerical stability, numerical dispersion, and numerical dissipation.

- Model Problem:

$$u_t = u_x \qquad -\infty < x < \infty,\ t > 0,$$

$$u(x,0) = g(x) \qquad -\infty < x < \infty.$$

- Solution of Model Problem:

$$u(x,t) = g(x+t); \quad \textit{left moving wave},$$

$$u(x,t) = \mathcal{F}^{-1}\{\widehat{g}(k)e^{i(kx+\omega t)}\}; \quad k = \omega.$$

NOTE: $||u|| = ||g||$ for all time, $\frac{\omega}{k} = 1$, $\frac{d\omega}{dk} = 1$, and $\widehat{g}(k) = \mathcal{F}\{g(x)\}$.

Approximation of space derivatives

Let h be a small spatial increment. We wish to compute u_x at x, using information about u at nearby points $x \pm h$. Employ Taylor series:

$$u(x+h) = u(x) + hu_x + \frac{h^2}{2!}u_{xx} + \frac{h^3}{3!}u_{xxx} + O(h^4),$$

$$u(x-h) = u(x) - hu_x + \frac{h^2}{2!}u_{xx} - \frac{h^3}{3!}u_{xxx} + O(h^4).$$

Subtracting we get

$$u_x(x) = \frac{u(x+h)-u(x-h)}{2h} - \frac{h^2}{6}u_{xxx} + O(h^4).$$

To implement on a computer we drop the $O(h^2)$ term and introduce a grid, $x_j = jh$, $u_j = u(x_j)$, $j = ..., -2, -1, 0, 1, 2, ...,$ so that

$$u_x|_{x_j} = \frac{u_{j+1}-u_{j-1}}{2h}.$$

This is what we call the $O(h^2)-$accurate central difference approximation to the first derivative.

Second-order accurate means that $e(h) = 4e(\frac{h}{2})$ ($e(h)$ =error at mesh size h) provided u_{xxx} is bounded. Second-order central differences are *exact* for quadratics but we are interested in computing waves.

A basic element out of which waves are contructed (via Fourier synthesis) is $u(x) = e^{ikx}$, where $i = \sqrt{-1}$, and k is the Fourier variable (wavenumber) conjugate to x. For such functions:

$u_x = iku$ in continuous space, while

$u_x|_{x_j} = ik\frac{\sin(kh)}{kh}u_j$ in discrete space.

In Fourier space, the effect of the finite differencing is to replace the symbol ik, of $\frac{\partial}{\partial x}$, with the symbol $ik\frac{\sin(kh)}{kh}$.

NOTE: For fixed k we get the right answer as $h \to 0$, but with finite kh we do not.

The concept of points per wavelength

In Fourier space the relative error for $O(h^2)$ centered differentiation is

$$e_2(kh) = 1 - \frac{\sin(kh)}{kh}.$$

NOTE:

a) $e_2(kh) \to 0$ as $h \to 0$, k fixed
b) $e_2(kh) = O(k^2 h^2)$
c) $e_2(kh)$ depends only on kh

The parameter kh is non-dimensional:

$$kh = \frac{2\pi}{\lambda} h = \frac{2\pi}{N_{ppw}},$$

where $N_{ppw} = \lambda/h$ is what we will refer to as the "number of points per wavelength".

d) $e_2(kh) \sim \frac{1}{N_{ppw}^2}$

A more accurate approximation

Back to Taylor series on the grid:

$$u_x|_{x_j} = \frac{u_{j+1}-u_{j-1}}{2h} - \frac{h^2}{6} u_{xxx}|_{x_j} + O(h^4)$$

$$u_x|_{x_j} = \frac{u_{j+2}-u_{j-2}}{4h} - \frac{(2h)^2}{6} u_{xxx}|_{x_j} + O((2h)^4).$$

Multiply first eq. by $\frac{4}{3}$ and second eq. by $\frac{1}{3}$ and subtract:

$$u_x|_{x_j} = \frac{-u_{j+2}+8u_{j+1}-8u_{j-1}+u_{j-2}}{12h}.$$

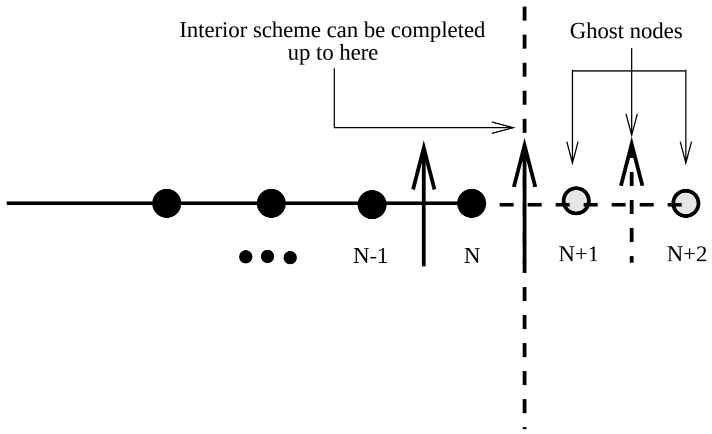
The error now is $O(h^4)$ (exact for quartics). How good is this for waves ?

The Fourier symbol for the $O(h^4)-$accurate central difference formula is

$$ik[\frac{4}{3}\frac{\sin(kh)}{kh} - \frac{1}{3}\frac{\sin(2kh)}{2kh}]$$

NOTE: $e_4(kh) \approx (kh)^4$ *thus* $\frac{e_2(kh)}{e_4(kh)} \approx (kh)^{-2}$!

Central differencing and boundaries



The $O(h^2)-$accurate formula can be applied on the last node next to a boundary node. Then, the appropriate boundary condition is applied to the boundary node thus completing the approximation of the derivative. However, the $O(h^4)-$accurate formula is not so straightforwardly applied near a boundary. Now, a *numerical* boundary condition is required at the node next to the boundary node, and that is in addition to the physical boundary condition one would want to impose at the boundary node.

It makes sense then to look at one-sided approximations to u_x (for use near boundaries).

One-sided differencing

- Forward difference

$$u_x|_{x_j} = \frac{u_{j+1} - u_j}{h}$$

with symbol: $ik(\frac{\sin(kh)}{kh} + i\frac{1-\cos(kh)}{kh})$

- Backward difference

$$u_x|_{x_j} = \frac{u_j - u_{j-1}}{h}$$

with symbol: $ik(\frac{\sin(kh)}{kh} - i\frac{1-\cos(kh)}{kh})$

Both of these formulas are only $O(h)-$accurate for $\frac{\partial}{\partial x}$. High-order one-sided differences can also be derived (via Taylor series).

NOTE: The symbol of $\frac{\partial}{\partial x}$ is no longer strictly complex.

Approximation of time derivatives

As a first step towards solving $u_t = u_x$ we considered

$$u_x = D_2(h)u = \frac{u(x+h)-u(x-h)}{2h}.$$

For a fully numerical solution we must also approximate u_t in the *new* problem

$$u_t = D_2(h)u.$$

Two major issues arise then:
—Stability
—Accuracy

Numerical stability is required of a useful code. Numerical dispersion and dissipation affect the accuracy of a wave computation.

Numerical dispersion

A partial differential equation, whose solutions are synthesized from harmonics of the form

$$e^{i\omega(k)t+ikx},$$

is non-dispersive if $\frac{d\omega}{dk} = constant$. Otherwise it is dispersive.

In other words, if the phase velocity $v_p = \frac{\omega(k)}{k}$ is a function of k, then the problem is dispersive. Clearly, $u_t = u_x$ with $\omega = k$ is *not* dispersive.

The numerical approximation of the space derivative now has us facing two problems: the one we want to solve, and the one we actually end up solving using the approximation.

These two problems are:

$$u_t = u_x \to u(x,t) = a(t)e^{ikx}$$

and

$$v_t = D_2(h)v \to v(x,t) = b(t)e^{ikx},$$

together with $u(x,0) = v(x,0) = e^{ikx}$, $a(0) = b(0) = 1$. Thus,

$$a(t) = e^{ikt}$$

$$b(t) = e^{ik\frac{\sin(kh)}{kh}t}.$$

We see the numerical approximation v to u is a dispersive solution with $\omega(k) = k\frac{\sin(kh)}{kh}$ instead of the correct $\omega(k) = k$.

This is no surprise because

$$D_2(h)v = v_x - \frac{h^2}{6} v_{xxx} + O(h^4).$$

Thus, v really solves *not* $v_t = v_x$ but

$$v_t = v_x - \frac{h^2}{6} v_{xxx} + O(h^4)$$

which is a dispersive partial differential equation with $\omega(k) \approx k + \frac{h^2}{6} k^3$.

With these simple notions we can now determine how many points per wavelength we really need when solving time-dependent equations.

Considering the exact solution

$$u = e^{ikt+ikx},$$

and the approximate solution

$$v = e^{ik\frac{\sin(kh)}{kh}t+ikx},$$

we see only phase errors due to the central differencing ($||u|| = ||v|| = 1$). Hence,

$$e_\phi = kt(1 - \frac{\sin(kh)}{kh}) \approx kt\frac{(kh)^2}{6}.$$

Normalizing time by the number of time periods $\frac{2\pi}{k} \to t = J\frac{2\pi}{k}$, we find that

$$N_{ppw} \sim \sqrt{\frac{J}{e_\phi}}$$

NOTE: the celebrated “rule of thumb” (10-20 points per wavelength) is not valid ! And, for fixed e_ϕ the N_{ppw} requirement grows with time.

Similar arguments for the $O(h^4)-$accurate central difference formula ($D_4(h)$) gives

$$N_{ppw} \sim (\frac{J}{e_\phi})^{\frac{1}{4}}$$

In general, $N_{ppw} \sim J^{\frac{1}{p}}$, where p is the order of accuracy of the FD method.

There are methods (spectral) for which

$$\lim_{p\to\infty} N_{ppw} = constant.$$

Numerical dissipation

By replacing $D_2(h)v$, in the previous discussion, with the forward difference

$$D_+(h)v = \frac{v_{j+1} - v_j}{h},$$

and repeating the calculations, we obtain

$$\omega(k) = k\frac{\sin(kh)}{kh} + i\frac{1 - \cos(kh)}{kh}.$$

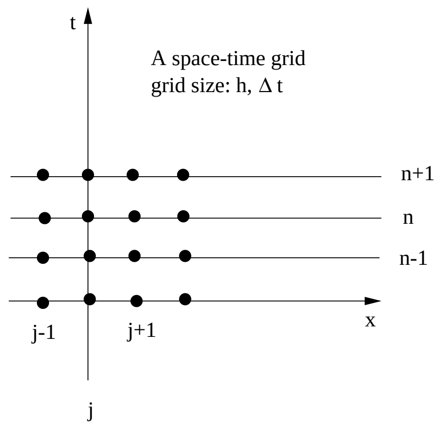
Thus, ω is no longer real and now the approximate solution decays in time (in addition to being dispersed) since

$$||v|| = e^{-\frac{k^2 h}{2}t},$$

while the true solution has $||u|| = 1$.

NOTE: if $D_-(h)$ (backward difference) is used instead of $D_+(h)$ then $Im\{\omega\} < 0$, and the approximate solution blows up as $e^{\frac{k^2 h}{2}t}$.

Analysis of some model schemes



- Forward Euler (ν =CFL number= $\frac{\Delta t}{h}$)

$$u_j^{n+1} = u_j^n + \frac{\nu}{2}(u_{j+1}^n - u_{j-1}^n)$$

Accuracy is $O(\Delta t) + O(h^2)$, i.e., it is a first-order scheme since $\Delta t \sim h$. Stability is determined by the DFT in j and $z-$transform in n: $u_j^n = z^n e^{ikhj}$ for $|kh| \leq \pi$.

Plug in scheme and obtain

$$z(\nu, kh) = 1 + i\nu \sin(kh),\ |z| > 1 \rightarrow \underline{\text{unstable}}$$

- Backward Euler

$$u_j^{n+1} = u_j^n + \tfrac{\nu}{2}(u_{j+1}^{n+1} - u_{j-1}^{n+1})$$

Accuracy is again $O(\Delta t) + O(h^2)$, but now the scheme is <u>implicit</u> (inversion of a tri-diagonal matrix is required at each timestep) and unconditionally stable since

$$z = \tfrac{1}{1 - i\nu \sin(kh)},\ |z| < 1 \text{ for any } \nu.$$

• Lax-Friedrich

$$u_j^{n+1} = \frac{1}{2}(u_{j+1}^n + u_{j-1}^n) + \frac{\nu}{2}(u_{j+1}^n - u_{j-1}^n)$$

Accuracy is still $O(\Delta t) + O(h^2)$ but now the scheme is conditionally stable since

$$z = \cos(kh) + i\nu \sin(kh),\ |z| \leq 1 \text{ for } \nu \leq 1.$$

NOTE: This scheme is exact for $\nu = 1$ (it reduces to $u_j^{n+1} = u_{j+1}^n$) !

- Leapfrog

$$u_j^{n+1} = u_j^{n-1} + \nu(u_{j+1}^n - u_{j-1}^n)$$

This is a two-level scheme. Accuracy now is $O(\Delta t^2) + O(h^2)$, i.e., it is a second-order scheme. It is conditionally stable, and non-dissipative (good for propagation problems) since

$$z = i\nu \sin(kh) \pm \sqrt{1 - \nu^2 \sin^2(kh)},\ |z| = 1 \text{ for } \nu \leq 1.$$

NOTE: To start this scheme you need data at the $n = 0$ and $n = 1$ time levels (use a first-order scheme to get the data at $n = 1$ from the initial condition at $n = 0$).

Numerical dispersion of fully discrete schemes

On the space-time grid let $u_j^n = e^{i\tilde{\omega} n\Delta t + ikhj}$. Then, for the Leapfrog scheme we find that the numerical frequency is

$$\tilde{\omega} = \frac{1}{\Delta t}\sin^{-1}(\nu \sin(kh)).$$

NOTE: Leapfrog is non-dispersive for $\nu = 1$, but for $\nu < 1$ we have a phase error per timestep that is given by

$$e_\phi(N_{ppw}, \nu)/\Delta t = \left|\frac{2\pi\nu}{N_{ppw}} - \sin^{-1}\left(\nu \sin\left(\frac{2\pi}{N_{ppw}}\right)\right)\right|.$$

Leapfrog for the 1D Maxwell equations

- E =electric field, H =magnetic field

$$E_t = H_x \to E_j^{n+1} = E_j^{n-1} + \nu(H_{j+1}^n - H_{j-1}^n)$$

$$H_t = E_x \to H_j^{n+1} = H_j^{n-1} + \nu(E_{j+1}^n - E_{j-1}^n)$$

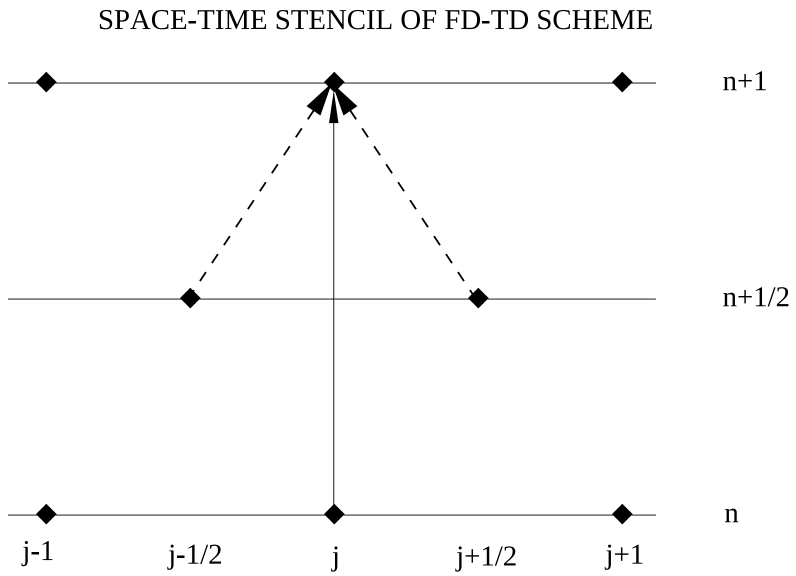
NOTE: From above, $E_{tt} = E_{xx} \to$

$$\frac{E_j^{n+2} - 2E_j^n + E_j^{n-2}}{(2\Delta t)^2} = \frac{E_{j+2}^n - 2E_j^n + E_{j-2}^n}{(2h)^2}.$$

This not nice because the best second-order accurate centered discretization of the wave equation is

$$\frac{E_j^{n+1} - 2E_j^n + E_j^{n-1}}{\Delta t^2} = \frac{E_{j+1}^n - 2E_j^n + E_{j-1}^n}{h^2} \ !$$

The 1D Staggered Leapfrog scheme



$$E_j^{n+1} = E_j^n + \nu(H_{j+\frac{1}{2}}^{n+\frac{1}{2}} - H_{j-\frac{1}{2}}^{n+\frac{1}{2}})$$

$$H_{j+\frac{1}{2}}^{n+\frac{1}{2}} = H_{j+\frac{1}{2}}^{n-\frac{1}{2}} + \nu(E_{j+1}^n - E_j^n)$$

NOTE: Now, eliminating the magnetic field variable results in the desired discrete wave equation for the electric field

$$\frac{E_j^{n+1} - 2E_j^n + E_j^{n-1}}{\Delta t^2} = \frac{E_{j+1}^n - 2E_j^n + E_{j-1}^n}{h^2}.$$

This particular staggered scheme is the best non-dissipative, second-order accurate, conditionally stable scheme that requires only one level of storage. It can also be viewed as a finite-volume scheme, and in that parlance the electric field is considered to be a cell-centered quantity while the magnetic field is a cell-edge quantity.

It holds a large advantage over the standard Leapfrog scheme when it comes to programming it. Also, it facilitates the implementation of electromagnetic boundary conditions that involve field components tangential to a curve and/or surface. For example, if the last node is on the surface of a PEC then we set $e_{NX}^{n+1} = 0$.

Most of the fundamental issues that affect the successful application of the FD-TD method can be understood by analyzing equivalent one-dimensional problems.

Bonus topic: Infinite-order accuracy

Most periodic functions do not look like polynomials. A natural representation for them is

$$u(x) = \textstyle\sum_{-\infty}^{\infty} a_\ell e^{i\ell x}.$$

In practice

$$u(x) \approx u_N(x) = P_N u(x) = \textstyle\sum_{-N}^{N} a_\ell e^{i\ell x}.$$

Suppose we approximate the derivative as

$$u_x \approx u_{N,x} = \textstyle\sum_{-N}^{N} i\ell a_\ell e^{i\ell x}.$$

This approximation uses $2N+1$ degrees of freedom *but* is exact for trigonometric polynomials of degree N. Also, now there is no fixed order of accuracy.

Let us see how this is possible (let $\ell = k$):

$$a_k = \frac{1}{2\pi} \int_0^{2\pi} u(x')e^{-ikx'} dx'.$$

Given an integer r, integrate by parts r times:

$$a_k = \frac{1}{2\pi} \frac{(-1)^r}{(-ik)^r} \int_0^{2\pi} u^{(r)}(x')e^{-ikx'} dx'.$$

Thus $|a_k| \leq \frac{C}{|k|^r}$, where C depends on the $r-th$ derivative of u. Then:

$$|u - P_N u|_\infty \leq \sum_{|k|>N} |a_k| \leq C \sum_{|k|>N} \frac{1}{|k^r|}$$

Since $\sum_{|k|>N} \frac{1}{|k^r|} = O(\frac{1}{N^{r-1}})$, the error goes like $1/N^r$ for any r, provided u is smooth enough. This is infinite order accuracy. The error of the second-order accurate FD derivative, on the other hand, goes like $1/N^2$!