Introduction to Numerical Relativity

1 Finite difference methods

In numerical analysis one can often approximately solve continuum systems, typically differential equations, through a process known as discretization. In the continuum case, the unknown functions, will typically be defined on some interval $0 < x < x_{max}$ of the real line and will thus constitute an infinite number of values. In the discrete case, however, the unknown functions will typically be defined only at a finite number of values $x_n, \quad n = 0, 1, 2, 3...N$.

Computational resources are finite, then the purpose of discretization is to reduce the infinite number of degrees of freedom of the continuum system to finite number. This is usually achieved by replacing differential equations with algebraic equations. The reason for doing this is because we can solve algebraic equations with computers.

Finite difference approximation

Finite difference approximation (FDA) is one specific approach to the discretization of continuum systems such as differential equations. The basic idea is that derivatives are replaced with algebraic difference quotients, very similar in spirit to algebraic expressions that are encountered in the standard definition of derivatives in ordinary calculus

$$\frac{d}{dx}f(x) = \lim_{h\to 0} \frac{f(x+h) - f(x)}{h} \ , \tag{1}$$

we say that $\frac{f(x+h)-f(x)}{h}$ is a finite difference approximation of the derivative of f in the point x.

In order to find the solution of a differential systems using FDA it is necessary to have a complete mathematical description of the problem, this includes: The specification of dependent and independent variables, the domain in terms of independent variables (coordinates), the differential equations governing dependent variables and the specification of sufficient initial and/or boundary conditions to ensure that the problem has a unique solution.

The starting point to work with a FDA is to define a finite difference grid that replaces the continuum domain with a finite set of grid points at which discrete solution will be computed. The grid will be characterized by a set of spacings between adjacent points in each of the coordinate directions, in the forthcoming discussion we will assume that these are constants. As we will see, mesh spacings constitute fundamental parameters that control accuracy of particular FDA.

Once we have defined the numerical domain, we need to replace all the differential operators with finite difference approximations, this include any derivative involved in the initial or boundary condition. This process yields a set of algebraic equations for the discrete unknowns. The solution of the algebraic equations is then performed computationally. Depending on the nature of the differential equations as well as the FDA used, the complexity of the algorithms required to accomplish this task can vary substantially.

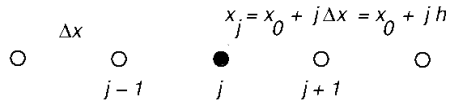

Figure 1: schematic of typical uniform Finite Difference Mesh.

One of the most important parts of the solution process is to perform an error analysis. The simplest one is to repeat calculations using same parameters, initial data, boundary conditions etc., but with varying mesh sizes. Investigation of behaviour of finite difference solution as a function of mesh size allows us to estimate the accuracy of the solution, and to establish that the solution is converging to the desired continuum limit. This is a necessary condition but as we will see is not always sufficient because this is only a measure of how the discrete operator approaches the continuous one.

Derivation of FDAs

The straightforward procedure is using a Taylor series. Given grid function value $f_j := f(x_j)$, and two nearest neighbours, $f_{j-1} := f(x_{j-1})$ and $f_{j+1} := f(x_{j+1})$. The Taylor expansion about $x = x_j$ with the mesh spacing $h = x_{j+1} - x_j$ is

$$f_{j-1} = f(x_{j-1}) = f_j - h[f']_j + \frac{1}{2}h^2[f'']_j - \frac{1}{6}h^3[f''']_j + \frac{1}{24}h^4[f'''']_j + O(h^5) \ , \tag{2}$$

$$f_{j+1} = f(x_{j+1}) = f_j + h[f']_j + \frac{1}{2}h^2[f'']_j + \frac{1}{6}h^3[f''']_j + \frac{1}{24}h^4[f'''']_j + O(h^5) \ , \tag{3}$$

We can obtain an approximation for the first derivative from (2) as

$$[f']_j = \frac{f_j - f_{j-1}}{h} + O(h) \ . \tag{4}$$

Since the leading order error term is linear in the mesh spacing, the approximation is said to be *first order accurate* or simply first order.

We now seek a linear combination of f_{j-1}, f_j and f_{j+1} which yields $[f']_j$ to $O(h^2)$ accuracy. We are looking constants $c_-,\ c_0,\ c_+$ such that

$$[f']_j = c_- f_{j-1} + c_0 f_j + c_+ f_{j+1} + O(h^2) \ , \tag{5}$$

Substituting the Taylor expansion, order by order, results in a system of three linear equations

$$c_- + c_0 + c_+ = 0 \ , \tag{6}$$

$$-hc_- + hc_+ = 1 \ , \tag{7}$$

$$h^2 c_- + h^2 c_+ = 0 \ , \tag{8}$$

which has the solution $c_+ = -c_- = \frac{1}{2h}$ and $c_0 = 0$. Thus, the second order finite difference approximation for the first derivative is

$$[f']_j = \frac{f_{j+1} - f_{j-1}}{2h} + O(h^2) \ , \tag{9}$$

For this particular example the formula can be obtained by subtracting (2) from (3)

$$[f']_j = \frac{f_{j+1} - f_{j-1}}{2h} - \frac{1}{6}h^2[f''']_j + O(h^4) \ . \tag{10}$$

The fact that the error of order $O(h^3)$ cancels is a typical characteristic of central differences.

Difference approximations for higher derivatives can be obtained in a similar fashion. For example, the central difference approximation of the second derivative can be obtained adding (2) and (3)

$$[f'']_j = \frac{f_{j+1} - 2f_j + f_{j-1}}{h} + \frac{1}{12}h^2[f'''']_j + O(h^4) \ . \tag{11}$$

Higher order approximations, while more accurate for a given grid spacing, in general require more work both for their evaluation and for the solution of the algebraic solution resulting from their use.

In order to calculate the derivatives with higher accuracy, more points are required. One has to find linear combinations of the form: $af_{j-2} + bf_{j-1} + cf_j + df_{j+1} + ef_{j+2}$ The relevant Taylor expansions

$$f_{j+2} = f_j + 2h[f']_j + 2h^2[f'']_j + \frac{4h^3}{3}[f''']_j + \frac{2h^4}{3}[f'''']_j + O(h^5) \ , \tag{12}$$

$$f_{j-2} = f_j - 2h[f']_j + 2h^2[f'']_j - \frac{4h^3}{3}[f''']_j + \frac{2h^4}{3}[f'''']_j + O(h^5) \ , \tag{13}$$

Sometimes this method is referred as *method of undetermined coefficients*.

Polynomial interpolation

A systematic way to construct FDAs with an arbitrary distribution of points -which are centred or one sided, and any desired convergence order is through interpolation. The idea is to use a local polynomial interpolant to approximate the function f at the desired points. For this purpose we use the Lagrange polynomials. For a given set of distinct points x_j and values of f_j, the Lagrange polynomial is the polynomial of the least degree that at each point x_j the polynomial and the function coincide.

$$L^N[f(x)] = \sum_{j=0}^{N} f(x_j) l_j^N(x) \ , \tag{14}$$

where $l_j^N(x)$ is a polynomial of degree (at most) N such that

$$l_j^N(x_i) = \delta_{ij} \ . \tag{15}$$

The FDA is defined as the exact derivative of the interpolant. For instance the first order derivative is

$$[f'](x) = \frac{d}{dx}L^N[f](x) = \sum_{i=1}^{N} f(x_i)\frac{d}{dx}l_i^N(x) \ . \tag{16}$$

The explicit form of the Lagrange basis polynomials is

$$l_j^N(x) = \prod_{0\le m\le N, m\neq j} \frac{x - x_m}{x_j - x_m} = \frac{x - x_0}{x_j - x_0} \cdots \frac{(x - x_{j-1})}{x_j - x_{j-1}} \frac{(x - x_{j+1})}{(x_j - x_{j+1})} \cdots \frac{(x - x_N)}{(x_j - x_N)} \ . \tag{17}$$

Here we show a couple of examples.
Using two points x_{j-1}, x_j, the first degree interpolant allow us to construct a first order FDA for $f'(x)$ at $x = x_j$

$$L^1[f](x) = f_{j-1} l_{j-1}^1(x) + f_j l_j^1(x) \ , \tag{18}$$

with

$$l_{j-1}^1(x) = \frac{x - x_j}{x_{j-1} - x_j} \ , \quad l_j^1 = \frac{x - x_{j-1}}{x_j - x_{j-1}} \ , \tag{19}$$

then

$$[f']_j = \frac{d}{dx} L^1[f](x) \mid_{x=x_j} = \frac{f_j - f_{j-1}}{h} \ , \quad h = x_j - x_{j-1}. \tag{20}$$

Using a second order interpolant with three points $\{x_{j-1},\, x_j,\, x_{j+1}\}$

$$L^2[f](x) = f_j \frac{(x-x_{j+1})(x-x_{j-1})}{(x_j - x_{j+1})(x_j - x_{j-1})} + f_{j-1} \frac{(x-x_j)(x-x_{j+1})}{(x_{j-1} - x_j)(x_{j-1} - x_{j+1})} + f_{j+1} \frac{(x-x_{j-1})(x-x_j)}{(x_{j+1} - x_{j-1})(x_{j+1} - x_j)} \ . \tag{21}$$

The derivative of $L^2[f](x)$ and can be evaluated at the desired point to obtain the approximate derivative of $f(x)$.

$$\begin{aligned}
[f']_{j-1} &= \frac{d}{dx} L^2[f](x) \mid_{x=x_{j-1}} = \frac{-3f_{j-1} + 4f_j - f_{j+1}}{2h} + O(h^2) \ , \\
[f']_j &= \frac{d}{dx} L^2[f](x) \mid_{x=x_j} = \frac{f_{j+1} - f_{j-1}}{2h} + O(h^2) \ , \\
[f']_{j+1} &= \frac{d}{dx} L^2[f](x) \mid_{x=x_{j+1}} = \frac{3f_{j+1} - 4f_j + f_{j-1}}{2h} + O(h^2) \ .
\end{aligned}$$

After a renaming of variables we get the one sided-derivatives

$$\begin{aligned}
[f']_k &= \frac{-3f_k + 4f_{k+1} - f_{k+2}}{2h} + O(h^2) \ , \\
[f']_k &= \frac{3f_k - 4f_{k-1} + f_{k-2}}{2h} + O(h^2) \ .
\end{aligned}$$

These are the usual expressions for the centred and one-sided second order derivatives. One can proceed in this way to systematically construct any FDA to any derivative with any desired convergence order and distribution of points.

Fornberg algorithm

Classical techniques for determining weights in finite difference formulas might be slow computationally. Surprisingly, simple recursions exist for calculating the weights in finite difference formulas for any order of derivative and to any order of accuracy well suited to generate tables of weights.

The Fornberg algorithm was introduced in the 80's and improved in the late 90's. It is based on the Padé approximation of a function [1].

With the following example we will illustrate both, how this algorithm works and how it is used.

The basic idea is to find the coefficients which make the stencil

$$b_1 f''(x) \approx c_0 f(x-h) + c_1 f(x) + c_2 f(x+h) \ , \tag{22}$$

as accurate as possible. Substituting $f(x) = e^{i\omega x}$ gives

$$-\omega^2 \ (\, b_1 \,) e^{i\omega x} \approx (c_0 e^{-i\omega h} + c_1 + c_2 e^{i\omega h}) e^{i\omega x} \ , \tag{23}$$

The goal is to make the approximation accurate if expanded locally around $h = 0$. Substituting $z = e^{i\omega h}$ and cancelling the factors $e^{i\omega x}$

$$\left(\frac{\ln z}{h}\right)^2 (b_1) \approx \frac{c_0}{z} + c_1 + c_2 z \ , \quad \Rightarrow \quad \left(\frac{\ln z}{h}\right)^2 \approx \frac{c_0 + c_1 z + c_2 z^2}{b_1 z} \ . \tag{24}$$

[1] The Padé approximation of order $[m, n]$ to an analytic function $f(x)$ at x_0 is the rational function $R(x) = \frac{p(x)}{q(x)}$ where $p(x)$ is a polynomial of degree m , $q(x)$ is a polynomial of degree n, such that $f(x_0) = R(x_0), f'(x_0) = R'(x_0) ... f^{(n+m)}(x_0) = R^{(n+m)}(x_0)$.

Padé approximation of $\left(\frac{\ln z}{h}\right)^2$ around $z = 1$ to order $[2, 1]$ yields

$$\left(\frac{\ln z}{h}\right)^2 \approx \frac{1 - 2z + z^2}{h^2 z} \ , \tag{25}$$

thus, $c_0 = 1, c_1 = -2, c_2 = 1$ and $b_1 = h^2$. Substituting these results in (22) we finally get

$$f''(x) \approx \frac{f(x-h) - 2f(x) + f(x+h)}{h^2} \ . \tag{26}$$

As we will see, this approach and it generalizations can be implemented in Mathematica in straightforward way.