Finite Difference Method

KONRAD LORENZ
FUNDACIÓN UNIVERSITARIA

Simple Finite Difference Approximation to a Derivative

Truncating (1) after the first derivative term gives,

$$U(x_0+h)=U(x_0)+hU_x(x_0)+O(h^2)$$

Rearranging gives,

$$U_x(x_0)=\frac{U(x_0+h)-U(x_0)}{h}-\frac{O(h^2)}{h}$$

Neglecting the *O(h)* term gives,

$$U_x(x_0)\approx\frac{U(x_0+h)-U(x_0)}{h}$$

The las Eq. is called a first order FD approximation to U x (x o) since the approximation error=O(h) which depends on the first power of h.

This approximation is called a forward FD approximation since we start at x_o and step forwards to the point x_o+h. *h* is called the **step size** (h > 0).

Simple Finite Difference Approximation to a Derivative

As an example we choose a simple function for *U*. Let

$$U(x) = x^3$$

We will find the first order forward FD approximation to $U_x(1)$ using step size $h = 0.1$

Since

$$U_x(x_0) \approx \frac{U(x_0 + h) - U(x_0)}{h}$$

Substituting for U gives

$$U_x(x_0) \approx \frac{(x_0 + h)^3 - x_0^3}{h}$$

Replacing x_o by *1* and *h* by *0.1* gives,

What if h=0.05?

$$U_x(1) \approx \frac{(1 + 0.1)^3 - 1^3}{0.1} = 1{,}10333$$

Lets do it by hand

Finite Difference Approximations

For simplicity we suppose that *U* is a function of **only two variables**, t and x.

We will approximate the partial derivatives of U with respect to *x*.

As *t* is held constant U is effectively a function of the single variable *x* so we can use Taylor's formula (1) where the ordinary derivative terms are now partial derivatives and the arguments are *(t, x)* instead of *x*.

Finally we will replace the step size h by **Δx** (to indicate a change in x) so that (1) becomes,

$$U(t, x_0+\Delta x)=U(t, x_0)+\Delta x U_x(t, x_0)+\frac{\Delta x^2}{2!}U_{xx}(t, x_0)+\ldots+\frac{\Delta x^{n-1}}{(n-1)!}U_{n-1}(t, x_0)+O(\Delta x^n) \quad \textbf{(2)}$$

Truncating it to O(Δx^2) gives,

$$U(t, x_0+\Delta x)=U(t, x_0)+\Delta x U_x(t, x_0)+O(\Delta x^2)$$

Now we derive some FD approximations to partial derivatives. Rearranging it gives,

$$U_x(t, x_0)=\frac{U(t, x_0+\Delta x)-U(t, x_0)}{\Delta x}-O(\Delta x)$$

Finite Difference Approximations

In numerical schemes for solving PDEs we are restricted to a grid of discrete *x* values, x_1, x_2 ,...,x_N , and discrete *t* levels t_0 , t_1 , ...

We will assume a **constant** grid spacing, Δx, in x, so that $x_{i+1} = x_i + \Delta x$.

Evaluating the last equation for a point, (t_n , x_i), on the grid gives,

$$U_x(t_n, x_i) = \frac{U(t_n, x_{i+1}) - U(t_n, x_i)}{\Delta x} - O(\Delta x)$$

We will use the common *subscript/superscript* notation

$$U_i^n = U(t_n, x_i)$$

so that dropping the O(Δx) error term,

$$U_x(t_n, x_i) \approx \frac{U_{i+1}^n - U_i^n}{\Delta x}$$

Finite Difference Approximations

We now derive another FD approximation to $U_x(t_n, x_i)$. Replacing *Δx* by -*Δx*

$$U(t, x_0 - \Delta x) = U(t, x_0) - \Delta x U_x(t, x_0) + O(\Delta x^2)$$

Evaluating it at (t_n, x_i) and rearranging as previously gives,

$$U_x(t_n, x_i) \approx \frac{U_i^n - U_{i-1}^n}{\Delta x}$$

And it is the first order backward difference approximation to $U_x(t_n, x_i)$.

Our first two FD approximations are first order in *x* but we can increase the order (and so make the approximation more **accurate**) by taking more terms in the Taylor series as follows.

Truncating to $O(\Delta x^3)$, then replacing *Δx* by -*Δx* and subtracting this new expression from (2) and evaluating at (t_n, x_i) gives, after some algebra,

$$U_x(t_n, x_i) \approx \frac{U_{i+1}^n - U_{i-1}^n}{2\Delta x}$$

And is called the second order central difference FD approximation to $U_x(t_n, x_i)$.

Finite Difference Approximations

Many PDEs of interest contain second order (and higher) partial derivatives so we need to derive approximations to them.

We will restrict our attention to second order *unmixed* partial derivatives i.e. U_{xx} .
Truncating (2) to $O(\Delta x^4)$ gives

$$U(t,x_0+\Delta x)=U(t,x_0)+\Delta x U_x(t,x_0)+\frac{\Delta x^2}{2!}U_{xx}(t,x_0)+\frac{\Delta x^3}{3!}U_{xxx}(t,x_0)+O(\Delta x^4) \qquad \textbf{(3)}$$

Replacing Δx by -Δx gives

$$U(t,x_0-\Delta x)=U(t,x_0)-\Delta x U_x(t,x_0)+\frac{\Delta x^2}{2!}U_{xx}(t,x_0)-\frac{\Delta x^3}{3!}U_{xxx}(t,x_0)+O(\Delta x^4) \qquad \textbf{(4)}$$

Adding (3) and (4) gives

$$U(t,x_0+\Delta x)+U(t,x_0-\Delta x)=2U(t,x_0)+\Delta x^2 U_{xx}(t,x_0)+O(\Delta x^4)$$

Evaluating at (t_n, x_i)

$$U_{i+1}^n+U_{i-1}^n=2U_i^n+\Delta x^2 U_{xx}(t_n,x_i)+O(\Delta x^4)$$

Finite Difference Approximations

Rearranging it and dropping the $O(\Delta x^2)$ error term gives

$$U_{xx}(t_n, x_i) \approx \frac{U_{i+1}^n - 2U_i^n + U_{i-1}^n}{\Delta x^2} \quad \textbf{(5)}$$

And it is the second order symmetric difference FD approximation to $U_{xx}(t_n,x_i)$

partial derivative	finite difference approximation	type	order
$\frac{\partial U}{\partial x} = U_x$	$\frac{U_{i+1}^n - U_i^n}{\Delta x}$	forward	first in x
$\frac{\partial U}{\partial x} = U_x$	$\frac{U_i^n - U_{i-1}^n}{\Delta x}$	backward	first in x
$\frac{\partial U}{\partial x} = U_x$	$\frac{U_{i+1}^n - U_{i-1}^n}{2\Delta x}$	central	second in x
$\frac{\partial^2 U}{\partial x^2} = U_{xx}$	$\frac{U_{i+1}^n - 2U_i^n + U_{i-1}^n}{\Delta x^2}$	symmetric	second in x

Finite Difference Approximations

Approximations to partial derivatives with respect to *t* are derived in a similar manner

partial derivative	finite difference approximation	type	order
$\frac{\partial U}{\partial t} = U_t$	$\frac{U_i^{n+1} - U_i^n}{\Delta t}$	forward	first in t
$\frac{\partial U}{\partial t} = U_t$	$\frac{U_i^{n} - U_i^{n-1}}{\Delta t}$	backward	first in t
$\frac{\partial U}{\partial t} = U_t$	$\frac{U_i^{n+1} - U_i^{n-1}}{2\Delta t}$	central	second in t
$\frac{\partial^2 U}{\partial t^2} = U_{tt}$	$\frac{U_i^{n+1} - 2U_i^n + U_i^{n-1}}{\Delta t^2}$	symmetric	second in t

Example

The 1D linear advection equation is

$$U_t + vU_x = 0$$

where the independent variables are *t* (time) and *x* (space). x is restricted to the finite interval *[p, q]* which is called the computational domain.

v is a constant and the dependent variable, U = U(t,x).

Let the initial conditions be,

Known!

$$U(0,x) = f(x) \qquad p \leqslant x \leqslant q$$

A solution is a function U = U(t, x) which satisfies the PDE at all points *x* in the computational domain and all times *t* and the initial conditions.

Step by Step

Step 1: Spatial Discretization

The computational domain contains an infinite number of x values so first we must replace them by a finite set. This process is called spatial discretization

For simplicity the computational domain is replaced by a grid of **N** *equally spaced grid points.* Starting with the first grid point at $x=p$ and ending with the last grid point at $x=q$, the constant grid spacing, Δx, is,

$$\Delta x=\frac{q-p}{N-1}$$

The values of x in the discretized computational domain are indexed by subscripts to give,

$$x_1=p, x_2=p+\Delta x, \ldots, x_i=p+(i-1)\Delta x, \ldots, x_N=p+(N-1)\Delta x=q$$

So the grid spacing is constant,

$$x_{i+1}=x_i+\Delta x$$

Fixing t at $t = t_n$ we approximate the spatial partial derivative, U_x at each point (t_n,x_i) using the forward difference formula

$$U_t+vU_x=0 \longrightarrow U_t+v\frac{U_{i+1}^n-U_i^n}{\Delta x}=0$$

semi-discrete form since only the spatial derivative has been discretized

Step 2: Time Discretization

Fixing *x* at $x=x_i$ we approximate the temporal partial derivative, U_t at each point (t_n, x_i) using the first order forward difference formula

$$U_t \approx \frac{U_i^{(n+1)} - U_i^n}{\Delta t}$$

and

$$U_t + v\frac{U_{i+1}^n - U_i^n}{\Delta x} = 0 \longrightarrow \frac{U_i^{(n+1)} - U_i^n}{\Delta t} + v\frac{U_{i+1}^n - U_i^n}{\Delta x} = 0$$

which rearranges to give

$$U_i^{(n+1)} = U_i^{(n)} - \frac{v\,\Delta t}{\Delta x}\left(U_{i+1}^{(n)} - U_i^{(n)}\right)$$

It is an example of a FDS to approximate the solution of the PDE. Is a so-called time-marching scheme which enables U values at time level n+1 to be approximated from U values at the previous time level n.

Since all U values are only known exactly at the initial time level is rewritten as

$$u_i^{(n+1)} = u_i^{(n)} - \frac{v\,\Delta t}{\Delta x}\left(u_{i+1}^{(n)} - u_i^{(n)}\right)$$

:$u(t_n,x_i)$ is a numerical approximation to $U(t_n,x_i)$

Step 2: Time Discretization

Remarks

- $u(0,x_i) = U(0,x_i)$ but this will **not** be true in general for later times.

- **u** values on the right hand side of are all at time t_n whereas on the left hand side **u** values are all at the **next** timed level $t_n + \Delta t = t_{n+1}$
- It is an example of a time-marching scheme in that (known) data for each grid point at time t_n is used to find data at each grid point at the future time $t_n + \Delta t$. This is called an **iteration** of the scheme. After an iteration of the scheme all *u* values at each grid point are known at time $t_n + \Delta t$.

- These new values can be used as known data for another iteration of the scheme to give data for each grid point at the next time level.

- This process can be repeated until the required future time is attained.

- The errors in approximating the spatial and temporal derivatives which are used are $O(\Delta x)$ and $O(\Delta t)$ respectively and so it is said to be (formally) first order in space (*x*) and first order in time (*t*).

- The grid spacing, Δx, was determined by choosing the number of grid points, N. A larger N gives a smaller Δx and a (hopefully) more accurate solution as spatial derivatives are more accurately approximated. However as N increases compute time increases so there is a trade off between **accuracy and speed**.

Step 2: Time Discretization
Remarks

- The time step, Δt, is for the moment, chosen arbitrarily. However a smaller time step will mean that more iterations are needed to reach a stated future time which will obviously increase the compute time.

- In addition, since the result of each iteration is an approximation to the required solution, more iterations could cause the build up of more error.

$$u_i^{(n+1)} = u_i^{(n)} - \frac{v\,\Delta t}{\Delta x}\left(u_{i+1}^{(n)} - u_i^{(n)}\right)$$

is said to be an *explicit method* since the value of *u* at the next time level is given by an explicit formula for each grid point.

MATHEMATICAL WARNING:

It does not work for v>0...

PEN AND PAPER CALCULATION

Example

Board II

References

D. M. Causon, C. G. Mingham, Introductory Finite Difference Methods for PDEs Ventus Publishing ApS, ISBN 978-87-7681-642-1 (2010).