Finite differences

Real space approximations

18.303 Linear Partial Differential Equations: Analysis and Numerics

Finite differences

What is the simplest way to do differentials on a computer?

Finite differences

What is the simplest way to do differentials on a computer?

For the sake of simplicity let's assume we have $u : [0,1] \to \mathbb{R}$ and we want to make sense of $\frac{\mathrm{d}u(x)}{\mathrm{d}x}$.

Finite differences

What is the simplest way to do differentials on a computer?

For the sake of simplicity let's assume we have $u : [0,1] \to \mathbb{R}$ and we want to make sense of $\frac{\mathrm{d}u(x)}{\mathrm{d}x}$.

We discretize the variable x, the function u, and the derivative $u'(x)$. Formally this means that

$$u(x) \to u_n = u(x_n), \quad u'(x) \to u'_n,$$

where $n = 1, 2, ..., N$.

Finite differences

What is the simplest way to do differentials on a computer?

For the sake of simplicity let's assume we have $u : [0,1] \to \mathbb{R}$ and we want to make sense of $\frac{\mathrm{d}u(x)}{\mathrm{d}x}$.

We discretize the variable x, the function u, and the derivative $u'(x)$. Formally this means that

$$u(x) \to u_n = u(x_n), \quad u'(x) \to u'_n,$$

where $n = 1, 2, ..., N$.

For now, let's assume that $x_n = n\Delta x$, where $\Delta x = 1/N$ (we will later talk about non-evenly spaced discretization). We can *approximate* $u'(x)$ by writing the Taylor series for u:

$$u(x_n + \Delta x) = u(x_n) + u'(x_n)\Delta x + \mathcal{O}(\Delta x^2).$$

(Here we use "Big-$\mathcal{O}$ notation": the term $\mathcal{O}(\Delta x^2)$ scales as Δx^2 approaching zero as $\Delta x \to 0$. Frankly it means that for small Δx this term is very small.)

Finite differences

We can solve this for $u'(x_n)$ giving

$$u'(x_n) = \frac{u(x_n + \Delta x) - u(x_n)}{\Delta x} + \mathcal{O}(\Delta x) \tag{1}$$

i.e.

$$u'_n \approx \frac{u_{n+1} - u_n}{\Delta x}.$$

We can solve this for $u'(x_n)$ giving

$$u'(x_n) = \frac{u(x_n + \Delta x) - u(x_n)}{\Delta x} + \mathcal{O}(\Delta x) \tag{1}$$

i.e.

$$u'_n \approx \frac{u_{n+1} - u_n}{\Delta x}.$$

We notice that this is just a linear combination of the terms in the vector u_n i.e. it can be expressed as a linear operation

$$\mathbf{u}' \approx \delta_+ \mathbf{u}.$$

Finite differences

We can solve this for $u'(x_n)$ giving

$$u'(x_n) = \frac{u(x_n + \Delta x) - u(x_n)}{\Delta x} + \mathcal{O}(\Delta x) \tag{1}$$

i.e.

$$u'_n \approx \frac{u_{n+1} - u_n}{\Delta x}.$$

We notice that this is just a linear combination of the terms in the vector u_n i.e. it can be expressed as a linear operation

$$\mathbf{u}' \approx \delta_+ \mathbf{u}.$$

We call the operator δ_+ forward difference operator. Changing the sign of Δx in Eq. (1) gives the backward difference operation

$$u'_n \approx \frac{u_n - u_{n-1}}{\Delta x} = (\delta_- \mathbf{u})_n.$$

Finite differences

The fact that we have $\mathcal{O}(\Delta x)$ in Eq. (1) tells that the difference operation is *first order* i.e. the error of the operation when approximating the actual differential scales linearly with Δx.

Finite differences

The fact that we have $\mathcal{O}(\Delta x)$ in Eq. (1) tells that the difference operation is *first order* i.e. the error of the operation when approximating the actual differential scales linearly with Δx.

We can do a bit better. Calculating

$$\mathbf{u}' \approx \left(\frac{\delta_+ + \delta_-}{2}\right)\mathbf{u} = \delta_0 \mathbf{u}$$

gives in indices

$$u_n' = \frac{u_{n+1} - u_{n-1}}{2\Delta x} + \mathcal{O}(\Delta x^2).$$

We call δ_0 the central difference operator and, as we can see, it's second order in Δx.

Exercise 1

Derive a formula for the second order derivative u_n'' by using the Taylor series for u_{n+1} and u_{n-1}.

Operators δ

Note that in the discrete setting our vector space is just $\mathbb{R}^n$. What are the difference operators δ then? They are linear maps i.e. matrices.

Operators δ

Note that in the discrete setting our vector space is just $\mathbb{R}^n$. What are the difference operators δ then? They are linear maps i.e. matrices.

To calculate a matrix for the second differential, let's write

$$\delta^{(2)}\mathbf{u} = \frac{1}{\Delta x^2}\begin{pmatrix} u_0 - 2u_1 + u_2 \\ u_1 - 2u_2 + u_3 \\ \vdots \\ u_{N-1} - 2u_N + u_{N+1} \end{pmatrix}.$$

Operators δ

Note that in the discrete setting our vector space is just $\mathbb{R}^n$. What are the difference operators δ then? They are linear maps i.e. matrices.

To calculate a matrix for the second differential, let's write

$$\delta^{(2)}\mathbf{u} = \frac{1}{\Delta x^2}\begin{pmatrix} u_0 - 2u_1 + u_2 \\ u_1 - 2u_2 + u_3 \\ \vdots \\ u_{N-1} - 2u_N + u_{N+1} \end{pmatrix}.$$

We immediately notice that we need values u_0 and u_{N+1}. In order to define the operator properly, we need a notion of boundary conditions.

Operators δ

We can write this e.g. for Dirichlet boundaries defining $u_0 = u_{N+1} = 0$.

Operators δ

We can write this e.g. for Dirichlet boundaries defining $u_0 = u_{N+1} = 0$.

Now,

$$\delta_{\mathrm{D}}^{(2)}\mathbf{u} = \frac{1}{\Delta x^2}\begin{pmatrix} -2u_1 + u_2 \\ u_1 - 2u_2 + u_3 \\ \vdots \\ u_{N-1} - 2u_N \end{pmatrix}$$

and we can write

$$\delta_{\mathrm{D}}^{(2)} = \frac{1}{\Delta x^2}\begin{pmatrix} -2 & 1 & & & & \\ 1 & -2 & 1 & & & \\ & \ddots & \ddots & \ddots & & \\ & & 1 & -2 & 1 & \\ & & & 1 & -2 & \end{pmatrix}.$$

Operators δ

We can write this e.g. for Dirichlet boundaries defining $u_0 = u_{N+1} = 0$.

Now,

$$\delta_{\mathrm{D}}^{(2)}\mathbf{u} = \frac{1}{\Delta x^2}\begin{pmatrix} -2u_1 + u_2 \\ u_1 - 2u_2 + u_3 \\ \vdots \\ u_{N-1} - 2u_N \end{pmatrix}$$

and we can write

$$\delta_{\mathrm{D}}^{(2)} = \frac{1}{\Delta x^2}\begin{pmatrix} -2 & 1 & & & & \\ 1 & -2 & 1 & & & \\ & \ddots & \ddots & \ddots & & \\ & & 1 & -2 & 1 & \\ & & & 1 & -2 & \end{pmatrix}.$$

This matrix takes vectors from $\mathbb{R}^N$ to $\mathbb{R}^N$ i.e. it's a square matrix.

Poisson equation revisited

We can write our Poisson equation in the discretized setting as

$$\delta_{\mathsf{D}}^{(2)}\mathbf{u} = \mathbf{f}.$$

This is a *sparse* linear system that can be solved for any $\mathbf{f}$ ($\delta_{\mathsf{D}}^{(2)}$ is invertible).

We can also formulate an eigenvalue problem

$$\delta_{\mathsf{D}}^{(2)}\boldsymbol{\phi} = \lambda\boldsymbol{\phi}.$$

Exercise 2

Try the ansatz $\phi_k^{(n)} = \sin(a_n k)$, $k = 1, \ldots, N$ for solving the eigenvalue problem

$$\delta_{\mathrm{D}}^{(2)} \boldsymbol{\phi}^{(n)} = \lambda_n \boldsymbol{\phi}^{(n)}.$$

Finite differences from polynomial interpolation

Assume we are given three points 0, Δx, and $2\Delta x$ with corresponding values for u: u_0, u_1, and u_2.

We can fit a second order polynomial

$$p_2(x) = a_0 + a_1 x + a_2 x^2$$

by requiring that $p_2(x_i) = u_i$.

Finite differences from polynomial interpolation

Assume we are given three points 0, Δx, and $2\Delta x$ with corresponding values for u: u_0, u_1, and u_2.

We can fit a second order polynomial

$$p_2(x) = a_0 + a_1 x + a_2 x^2$$

by requiring that $p_2(x_i) = u_i$.

This gives a linear system

$$\begin{aligned} u_0 &= p_2(0) = a_0, \\ u_1 &= p_2(\Delta x) = a_2 \Delta x^2 + a_1 \Delta x + a_0, \\ u_2 &= p_2(2\Delta x) = 4a_2 \Delta x^2 + 2a_1 \Delta x + a_0. \end{aligned}$$

Finite differences from polynomial interpolation

Solving for a_k gives

$$\begin{aligned} a_0 &= u_0, \\ a_1 &= \frac{-3u_0 + 4u_1 - u_2}{2\Delta x}, \\ a_2 &= \frac{u_0 - 2u_1 + u_2}{2\Delta x^2}. \end{aligned}$$

Finite differences from polynomial interpolation

Solving for a_k gives

$$a_0 = u_0,$$
$$a_1 = \frac{-3u_0 + 4u_1 - u_2}{2\Delta x},$$
$$a_2 = \frac{u_0 - 2u_1 + u_2}{2\Delta x^2}.$$

We can evaluate

$$p'(\Delta x) = \frac{u_2 - u_0}{2\Delta x}$$

and

$$p''(\Delta x) = \frac{u_2 - 2u_1 + u_0}{\Delta x^2}.$$

Finite differences from polynomial interpolation

This gives us a systematic way of deriving stencils (operators) for finite differences on any number of points (we could have chosen e.g. 5 points and a 4th order polynomial).

Finite differences from polynomial interpolation

This gives us a systematic way of deriving stencils (operators) for finite differences on any number of points (we could have chosen e.g. 5 points and a 4th order polynomial).

Moreover, we could have chosen our points with a non-equal spacing and could have still solved for the coefficients a_k. This shows that this method can be used for non-uniform grids.

Finite differences from polynomial interpolation

This gives us a systematic way of deriving stencils (operators) for finite differences on any number of points (we could have chosen e.g. 5 points and a 4th order polynomial).

Moreover, we could have chosen our points with a non-equal spacing and could have still solved for the coefficients a_k. This shows that this method can be used for non-uniform grids.

The algorithm that automatically generates stencils from the interpolating polynomial forms is known as the Fornberg algorithm.