PDE Spring 2016

A. Donev

Lecture 19

Discrete Sine/Fourier Transform

Recall the Fourier series for Dirichlet homogeneous BCs

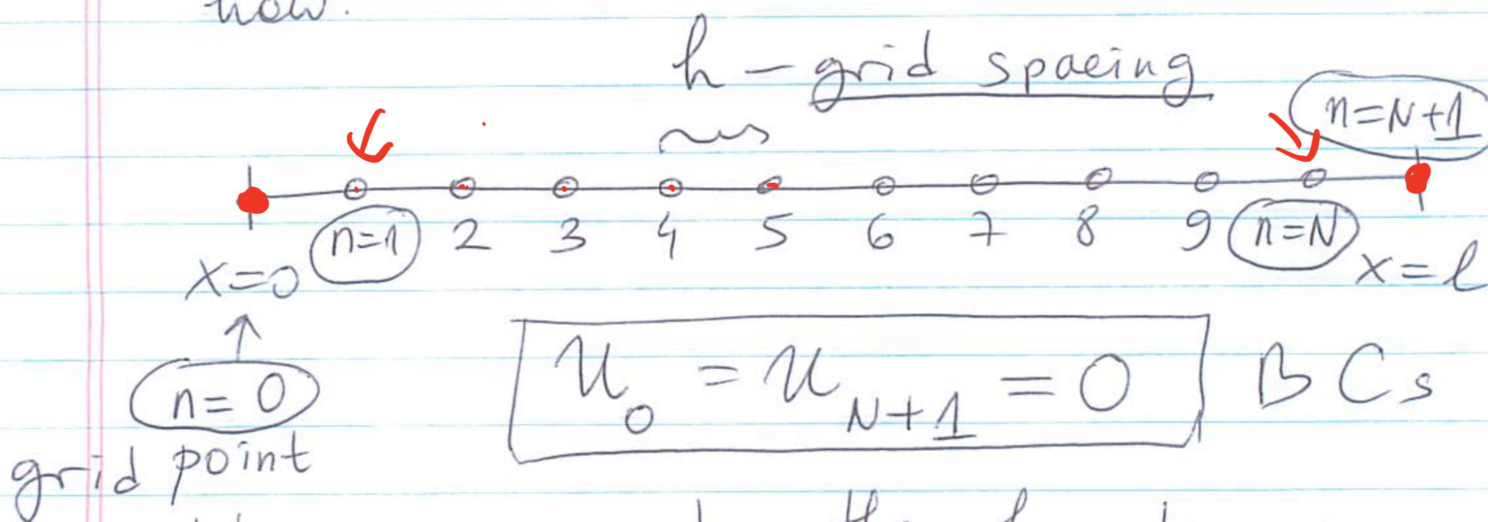
$$\begin{cases} \varphi(x) = \sum_{n=1}^{\infty} A_n \sin\left(\frac{n\pi x}{\ell}\right) \\ \varphi(0) = \varphi(\ell) = 0 \end{cases}$$

We know that

$$A_n = \frac{2}{\ell} \int_0^{\ell} \varphi(x) \sin\left(\frac{n\pi x}{\ell}\right) dx$$

But what if we cannot compute these integrals analytically?

In numerical methods, we must use a finite number of points to represent the solution, and we must replace sums with integrals. Here is how.



$$\boxed{u_0 = u_{N+1} = 0} \quad \text{BCs}$$

We approximate the function on a grid of points with spacing

$$h = \frac{\ell}{N+1} \qquad x_k = k \cdot h$$

$$\boxed{u(x_k, t) = u(k \cdot h, t) \approx u_k(t)}$$

This is the most obvious approach and is related to finite-difference methods, which we won't cover

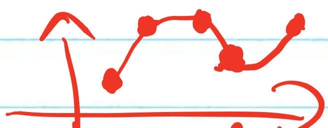
Now let's expand our function in a Fourier series

$$u(x,t) = \sum_{n=1}^{\cancel{\infty}\, N} \underline{\underline{A_n}}(t) \sin\left(\frac{n\pi x}{\ell}\right)$$

$\varphi''(x)$

From now on I will drop the time t to shorten the notation

If we want this to match our <u>discrete</u> solution $\{u_k \mid k = 1, \ldots, N\}$



interpolation

we need to <u>truncate</u> to N terms (N unknown coefficients) and solve the <u>linear system</u> of N equations for the unknown coefficients $a_1, \ldots, a_N$:

$$u_k = u(kh,t) = \sum_{n=1}^{N} a_n \sin\left(\frac{n\pi k \cancel{h}}{(N+1)\cancel{h}}\right)$$

for $k = 1, \ldots, N$

trigonometric polynomial

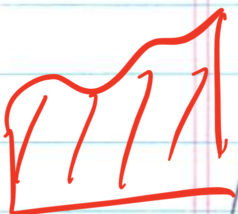
Linear system

$$\boxed{\forall k: \sum_{n=1}^{N} a_n \sin\left(\frac{n\pi k}{(N+1)}\right) = u_k}$$

$$\overset{[N\times N]}{M}\;\overset{[N\times 1]}{\vec{a}} = \overset{[N\times 1]}{\vec{u}}$$

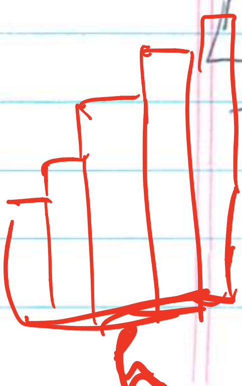
It is no accident that as it turns out the matrix M is <u>orthogonal</u> (unitary but real)! This is similar to how the eigenfunctions are orthogonal but now we have ordinary linear algebra! $M^{-1} \equiv M^T$

$$\vec{a} = M^T \vec{u}$$



$$\boxed{a_n = \frac{2}{N+1} \sum_{k=1}^{N} u_k \sin\left(\frac{n\pi k}{N+1}\right)}$$

very good approx for modest N



This should remind us of

$$A_n = \frac{2}{\ell} \int_0^{\ell} u(x) \sin\left(\frac{n\pi x}{\ell}\right) dx$$

this combination of formulas is called the <u>discrete sine transform</u>

<u>Backward</u> transform (inverse)
(convert "function" to Fourier coefficients)

IDST: $$a_n = \frac{2}{N+1} \sum_{k=1}^{N} u_k \sin\left(\frac{n\pi k}{N+1}\right)$$

<u>Forward</u> transform
(convert coefficients to "function")

DST: $$u_k = \sum_{n=1}^{N} a_n \sin\left(\frac{n\pi k}{N+1}\right)$$

Note that

$$a_n \approx A_n \quad , \quad n = 1, \ldots, N$$

but they are <u>not equal</u>.
We have made some approximations and thus introduced <u>numerical error</u>, called

truncation error.

For other boundary conditions, one just switches the basis functions, e.g., for periodic BCs replace $\sin\left(\frac{n\pi x}{\ell}\right)$ with $\exp\left(\frac{i n \pi x}{\ell}\right)$ and the same works, leading to the

<u>Discrete Fourier Transform</u>

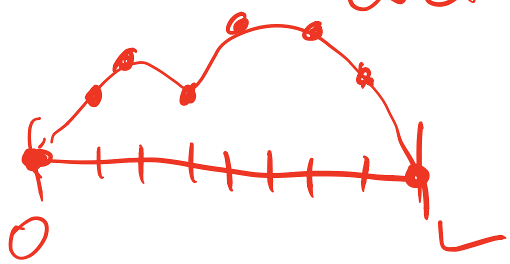
which can be computed <u>very</u> rapidly even for millions of points using the <u>Fast Fourier Trans</u> (FFT), one of the most important algorithms in use.

Following the <u>change of basis</u> with the DST, we can solve PDEs. For example, for <u>heat eq</u>

$$\frac{du_k(t)}{dt} = u_{xx}\Big|_{x=kh} = -\sum_{n=1}^{N} a_n(t)\left(\frac{n\pi}{\ell}\right)^2 \sin\left(\frac{n\pi x_k}{\ell}\right)$$

$$= \sum_{n=1}^{N} a_n'(t) \sin\left(\frac{n\pi x_k}{\ell}\right) \Rightarrow \boxed{a_n' = -\lambda_n a_n}$$

$$a_n = \underbrace{a_n(t=0)}_{\text{Fourier coeff of IC}} e^{-\lambda_n t}$$



$\varphi(x) \rightarrow FFT \rightarrow a_n(t=0) * e^{-\lambda_n t} \xrightarrow[iFFT]{} u(x,t)$

ODE on computer

$$\frac{du}{dt} = f(u,t)$$

$$u(t=0) = u_0$$

Δt small instead $\Delta t \to 0$

$$\frac{u(t+\Delta t) - u(t)}{\Delta t} = f(u(t),t)$$

$$u(t+\Delta t) \approx \underline{u(t)} + \underline{\Delta t} \cdot \underline{f(u(t), t)}$$

Euler's method