

# PDE Spring 2016

A. DONEV

①

## Lecture 19

### Discrete Sine / Fourier Transform

Recall the Fourier series for Dirichlet homogeneous BCs

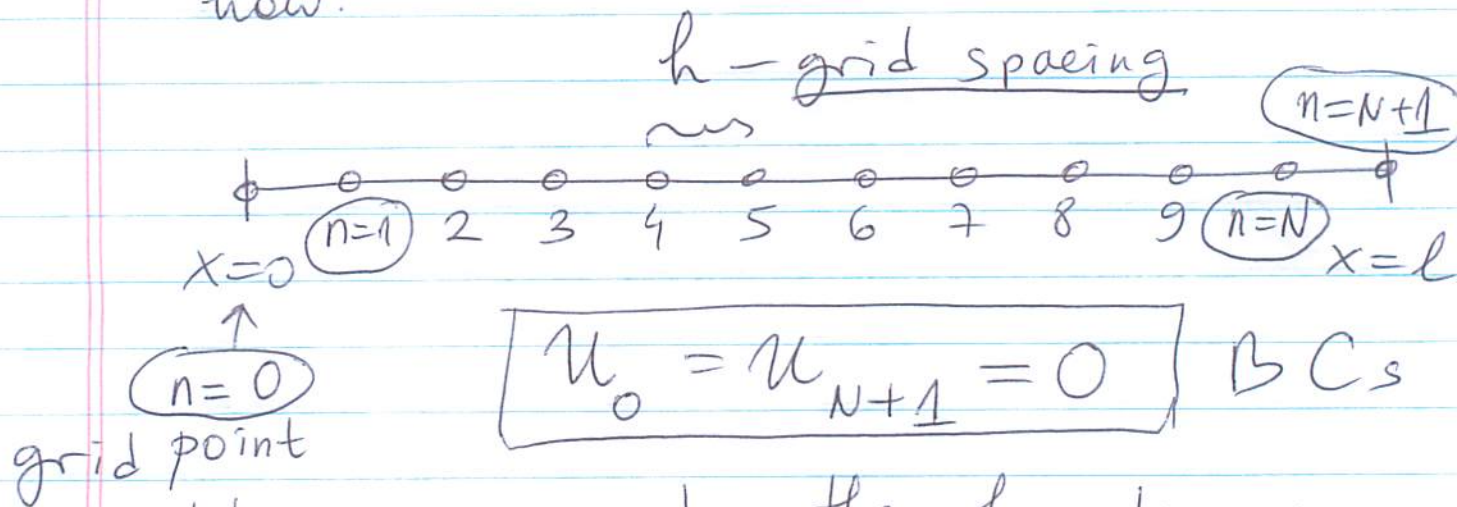
$$\left\{ \begin{array}{l} \varphi(x) = \sum_{n=1}^{\infty} A_n \sin\left(\frac{n\pi x}{l}\right) \\ \varphi(0) = \varphi(l) = 0 \end{array} \right.$$

We know that

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

But what if we cannot compute these integrals analytically?

In numerical methods, we <sup>(2)</sup> must use a finite number of points to represent the solution, and we must replace sums with integrals. Here is how.



We approximate the function on a grid of points with spacing

$$h = \frac{l}{N+1}$$

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

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

(3)

Now let's expand our function in a Fourier series

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



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

If we want this to match our discrete solution

$$\{u_k \mid k=1, \dots, N\}$$

we need to truncate to  $N$  terms ( $N$  unknown coefficients)

and solve the linear system of  $N$  equations for the unknown coefficients  $a_1, \dots, a_N$ :

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

for  $k=1, \dots, N$



Linear system

(4)

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

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

It is no accident that as it turns out the matrix  $M$  is orthogonal (unitary but real)!

This is similar to how the eigenfunctions are orthogonal but now we have ordinary linear algebra!  $M^{-1} \equiv M^T$

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

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

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 discrete sine transform (5)

Backward transform (inverse)  
(convert "function" to Fourier coefficients)

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

Forward transform  
(convert coefficients to "function")

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

Note that

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

but they are not equal.

We have made some approximations and thus introduced numerical error, called truncation error.



For other boundary conditions, (6) one just switches the basis functions, e.g., for periodic BCs replace  $\sin\left(\frac{n\pi x}{l}\right)$  with  $\exp\left(\frac{in\pi x}{l}\right)$  and the same works, leading to the Discrete Fourier Transform

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

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

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

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