Sine and Cosine Transforms

Continuous sine series: $$u(t) = 2\sum_{k=1}^{\infty} b_k \sin(kt), \quad \text{where} \quad b_k = \frac{1}{\pi}\int_0^{\pi} u(t)\sin(kt)\,dt$$

Comes from Fourier series, when $u(t)$ is an odd function on $[-\pi,\pi]$

$\Rightarrow \quad u(-t) = -u(t)$.

Discrete sine transform:

Let $t_j = \boxed{\frac{\pi}{N}}j$, $j = 0, 1, \ldots, N$, Apply trapezoidal rule to

h

$$b_k \approx \tilde{b}_k = \hat{u}_k = \frac{1}{\pi} h \sum_{j=1}^{N-1} u(t_j)\sin(kt_j) + \frac{h}{2}\underbrace{\left[u(0)\sin(k0)^{=0} + u(\pi)\sin(k\pi)^{=0}\right]}_{\text{end points}}$$

$$\underbrace{\hat{u}_k = \frac{1}{N}\sum_{j=1}^{N-1} u_j \sin\left(\frac{\pi}{N}kj\right), \quad k = 1, 2, \ldots, N-1}$$

(Forward) Discrete sine transform (DST)

Inverse DST:

$$u_j = 2 \sum_{k=1}^{N-1} \hat{u}_k \sin\left(\frac{\pi}{N} kj\right), \quad j = 1, 2, \dots, N-1$$

These form a transform pair.

- There are fast ways to compute the DST:

  FST

  fast

- Based on the FFT.

- Code on Blackboard that does this (Matlab)

Continuous cosine series: $u(t) = a_0 + 2\sum_{k=0}^{\infty} a_k \cos(kt)$, where $a_k = \frac{1}{\pi}\int_0^{\pi} u(t)\cos(kt)dt$

Comes from the Fourier series when $u(t)$ is even: $u(-t) = u(t)$

Discrete cosine transform:

Apply trapezoidal rule to integral, like the DST:

$$a_k \approx \hat{u}_k = \frac{h}{\pi}\sum_{j=1}^{N-1} u_j \cos\left(\frac{\pi}{N}jk\right) + \frac{h}{2\pi}\left[u_0\cos(0) + u_N \cos(\pi k)\right]$$

$$= \frac{1}{N}\left[\frac{u_0}{2} + \sum_{j=1}^{N-1} u_j \cos\left(\frac{\pi}{N}jk\right) + \frac{u_N}{2}\cos(\pi k)\right], \quad k=0,1,\dots,N$$

$$= \frac{1}{N}\sum_{j=0}^{N}{}'' u_j \cos\left(\frac{\pi}{N}jk\right)$$

Forward Discrete Cosine Transform (DCT)

Inverse transform:

$$u_j = 2\sum_{k=0}^{N}{}'' \hat{u}_k \cos\left(\frac{\pi}{N}jk\right), \quad j=0,1,\dots,N$$

- Form a transform pair.

Computing derivatives using the DFT (FFT)

Recall the discrete Fourier series (trigonometric interpolant) obtained from samples u_j at the location $t_j = \frac{2\pi j}{N}$, $j = 0, 1, \ldots, N-1$ (N is odd):

$$p_N(t) = \sum_{k=-\frac{N-1}{2}}^{\frac{N-1}{2}} \underline{\tilde{c}_k} e^{ikt}, \quad \text{where} \quad \tilde{c}_k = \frac{1}{N} \sum_{j=0}^{N-1} u_j e^{-\frac{2\pi i}{N} jk}, \ k = -\frac{N-1}{2}, \ldots, \frac{N-1}{2}.$$

$$\frac{d}{dt} e^{ikt} = ike^{ikt}$$

Also recall the discrete Fourier transforms (DFT) for u_j (N is odd):

$$\text{forward DFT:} \quad \hat{u}_k = \sum_{j=0}^{N-1} u_j e^{-2\pi ijk/N}, \ k = 0, \ldots, N-1$$

$$\text{inverse DFT:} \quad u_j = \frac{1}{N} \sum_{k=0}^{N-1} \hat{u}_k e^{2\pi ijk/N}, \ j = 0, \ldots, N-1$$

$$\underline{\hat{u}} = \frac{1}{N} \left[\tilde{c}_0 \ \ \hat{c}_1 \ \ \cdots\cdots \ \ \hat{c}_{\frac{N-1}{2}} \ \ \hat{c}_{-\frac{(N-1)}{2}} \ \cdots \ \hat{c}_{-2} \ \ \hat{c}_{-1} \right]$$

Compute derivatives of $p_N(t)$ and evaluate at t_j

$$\frac{d}{dt} p_N(t) = \sum_{k=-\frac{(N-1)}{2}}^{\frac{N-1}{2}} \hat{c}_k ik e^{ikt}$$

Evaluate at t_j

$$\frac{d}{dt} P_N(t)\Big|_{t=t_j} = P_N'(t_j) = \sum_{k=-\frac{(N-1)}{2}}^{\frac{N-1}{2}} \underbrace{\hat{c}_k ik}_{\hat{d}_k} e^{2\pi i jk/N}$$

$$= \sum_{k=-\frac{(N-1)}{2}}^{\frac{N-1}{2}} \hat{d}_k e^{2\pi i jk/N}$$

Compute derivatives using the DFT (and inverse)

Step 1: Apply DFT to $\{u_j\}_{j=0}^{N-1}$ and get $\{\hat{u}_k\}_{k=0}^{N-1}$

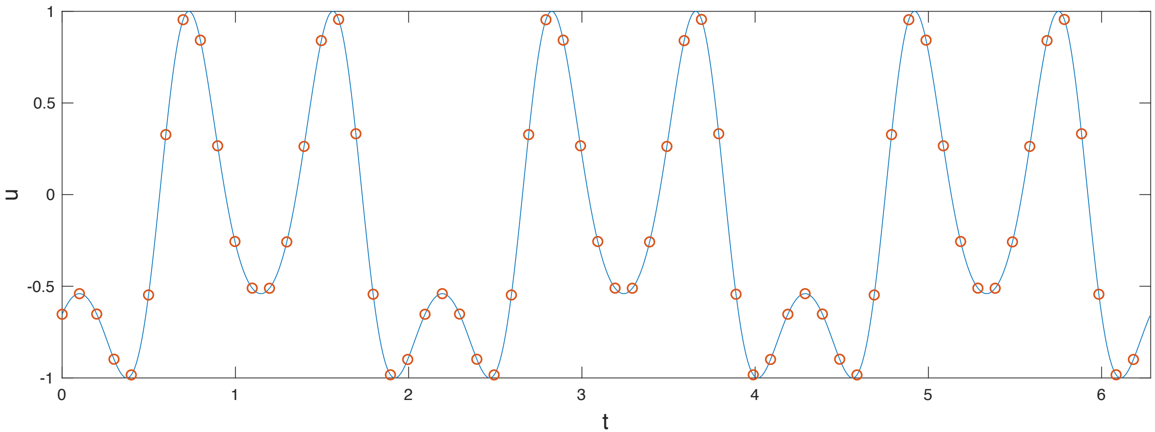
Step 2: Compute $\hat{d}_k = ik\hat{c}_k$, $k = -\frac{N-1}{2}, \dots, \frac{N-1}{2}$

$$\hat{u}_k = [\hat{c}_0 \quad \hat{c}_1 \quad \dots \quad \hat{c}_{\frac{N-1}{2}} \quad \hat{c}_{-\frac{(N-1)}{2}} \quad \dots \quad \hat{c}_{-1}]$$

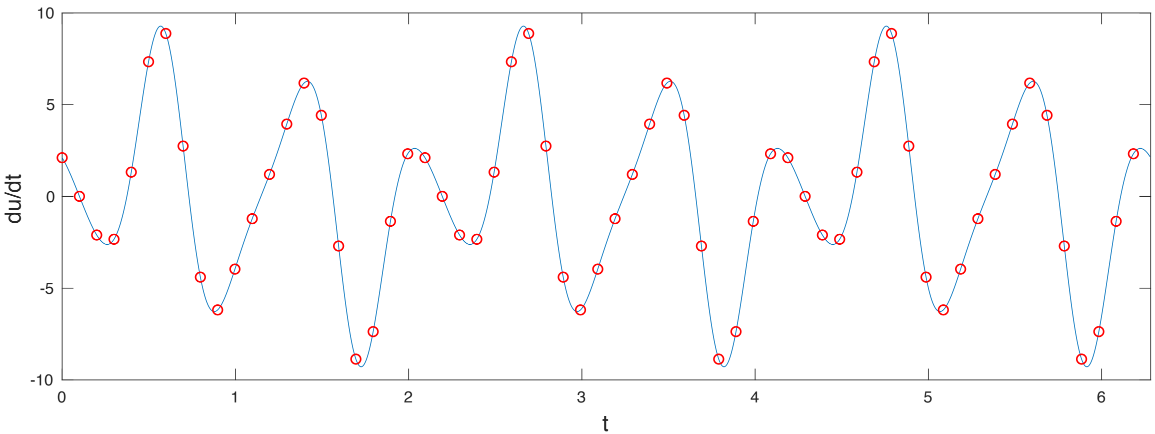
$$\hat{u}_k' = [0 \quad i\hat{c}_1 \quad 2i\hat{c}_2 \quad \dots \quad i\tfrac{N-1}{2}\hat{c}_{\frac{N-1}{2}} \quad -i\tfrac{(N-1)}{2}\hat{c}_{-\frac{(N-1)}{2}} \quad \dots \quad -2i\hat{c}_{-2} \quad -i\hat{c}_{-1}]$$

Step 3: Apply inverse DFT on $\hat{u}_k'$.

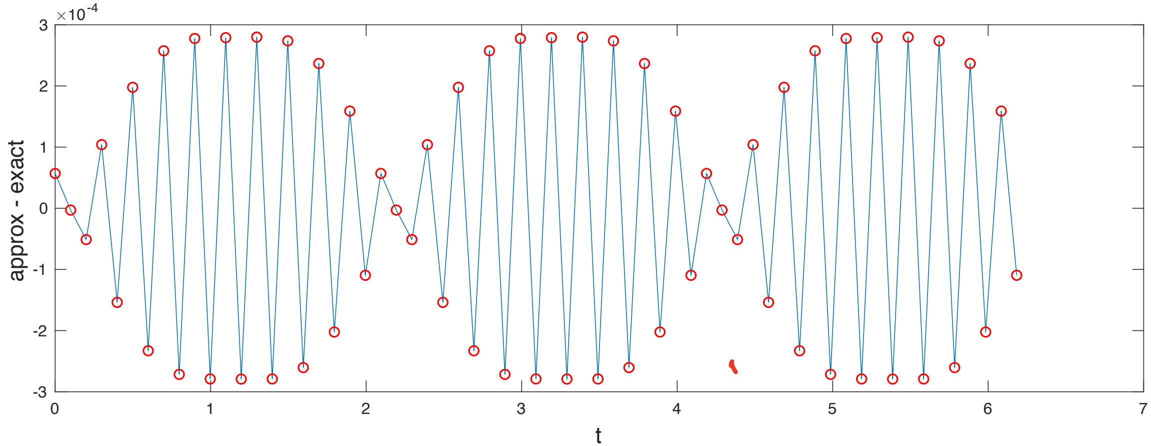
Example: $u(t) = \cos(1 + \pi \cos(3(t - 0.1)))$ and choose $N = 1\cancel{27}$ 63



Approximation of the derivative:



Error in the approximation:



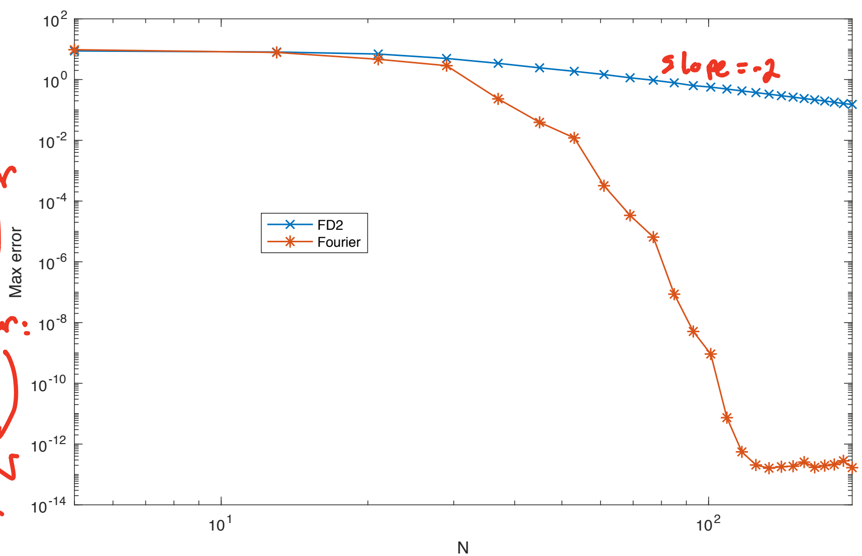
Convergence: ~~relative~~ max-norm of the error vs. N

$h = \frac{2\pi}{N}$

FD2: Truncation error $O(h^2) = O\left(\frac{1}{N^2}\right)$

Fourier: Truncation error: $O(e^{-cN}) = O\left(e^{-c\frac{2\pi}{h}}\right)$

Plot $\underline{\log(\text{error}) \text{ vs. } N}$



$\text{error} \approx B e^{-CN}$

$$\underbrace{\log(\text{error})}_{y} \approx \underbrace{\log(B)}_{b} - C N$$

slope

$u''(x) = f(x)$ periodic over $[0, 2\pi]$

Given $f(x)$ find $u(x)$?