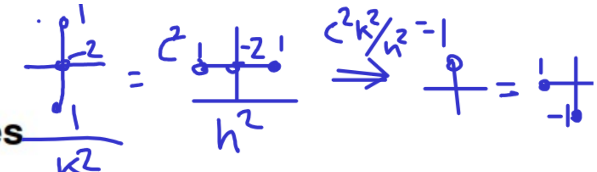


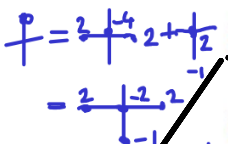
1 Wave equation by finite differences

(a) Using the standard 3-point differences for u_{tt} and u_{xx}, write down the time stepping formula for the wave equation $u_{tt} = c^2 u_{xx}$ for the case that the time step is as large as it can be without instability. I'm asking for an explicit formula for u_{j+1} in terms of other grid values.

(b) Entirely by hand, use your formula from part (a) to fill in the blanks below.

j+1	2	2+4-4 = 2	2+4-4 = 2	4+2-2 = 4	4+2-2 = 4	2+2-2 = 2	2
j	2	2	4	4	2	2	2
j−1	2	4	4	2	2	2	2

nice!



(c) Repeat parts (a) and (b) for the case that k^2 is twice as large. (k is the time step.) i.e. $\sigma = 2$ instead of 1.

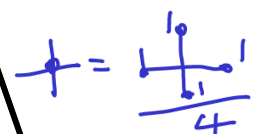
2	4	0	2	6	2	2
2	2	4	4	2	2	2
2	4	4	2	2	2	2

not nice.

2 Laplace via SOR

Grid (equal horizontal and vertical spacings) and BCs:

	4	3	
4	3	2	1
	2	1	

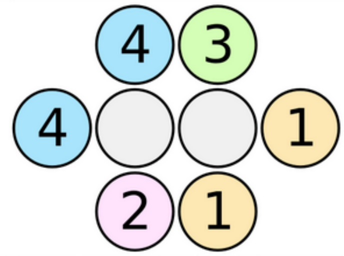


HW9

2 Laplace via SOR

(a)

$$\begin{array}{cccc} & 4 & 3 & \\ 4 & u_1 & u_2 & 1 \\ & 2 & 1 & \end{array}$$



$$-4u_1 + u_2 = -(4+4+2)$$

$$u_1 - 4u_2 = -(3+1+1)$$

$$\begin{bmatrix} -4 & 1 \\ 1 & -4 \end{bmatrix}\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} -10 \\ -5 \end{bmatrix}$$

Solution is $\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} 3 \\ 2 \end{bmatrix}$ by eyeball!

(b) $A = \begin{bmatrix} -4 & 1 \\ 1 & -4 \end{bmatrix} \rightarrow D = \begin{bmatrix} -4 & 0 \\ 0 & -4 \end{bmatrix}, L = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}, U = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}.$

$$T_\omega = (D+\omega L)^{-1}\left[(1-\omega)D - \omega U\right]$$

$$= \begin{bmatrix} -4 & 0 \\ \omega & -4 \end{bmatrix}^{-1}\left[(1-\omega)\begin{bmatrix} -4 & 0 \\ 0 & -4 \end{bmatrix} - \omega\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}\right]$$

$$= \frac{-1}{16}\begin{bmatrix} -4 & 0 \\ -\omega & -4 \end{bmatrix}\begin{bmatrix} -4+4\omega & -\omega \\ 0 & -4+4\omega \end{bmatrix} = \begin{bmatrix} 1-\omega & \omega/4 \\ \frac{\omega(1-\omega)}{4} & \frac{\omega^2}{16} - \omega + 1 \end{bmatrix}$$

Characteristic eqn

$$\lambda^2 + \left(\frac{-\omega^2}{16} + 2\omega - 2\right)\lambda + (\omega-1)^2 = 0$$

Eigenvalues

$$\lambda = \frac{\omega^2}{32} - \omega + 1 \pm \frac{\omega\sqrt{\omega^2 - 64\omega + 64}}{32}$$

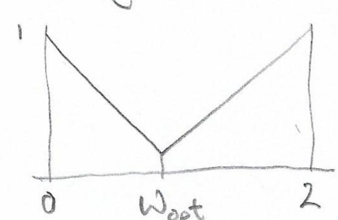
Optimal where there's a double root:

$$\omega^2 - 64\omega + 64 = 0 \longrightarrow \omega_{opt} = 32 \pm 8\sqrt{15}$$

$-$ gives value in $(0,2)$.

$$\omega_{opt} = 32 - 8\sqrt{15} = 1.0161\ldots$$

$$\lambda\Big|_{\omega=\omega_{opt}} = 0.0161\ldots$$



2(e) At ω_{opt}, $\rho(T) = 0.0161\ldots$

If $|e_0| = 0.1$

then $|e_k| \leq \rho(T)^k |e_0|$

$$\leq (0.0161\ldots)^k \cdot 0.1$$

For $|e_k| < 10^{-8}$, suffices to have

$$(0.0161\ldots)^k < 10^{-7}$$

$$k \log_{10}(0.0161\ldots) < -7$$

$$k > \frac{-7}{\log_{10}(0.0161)} = 3.9\ldots$$

So 4 iterations would suffice.

3 (a) $(F_4)_{ij} = \frac{1}{\sqrt{n}}\left(e^{-2\pi \hat{i}/4}\right)^{ij} = (-\hat{i})^{ij}$, $i,j \in \{0,1,2,3\}$.

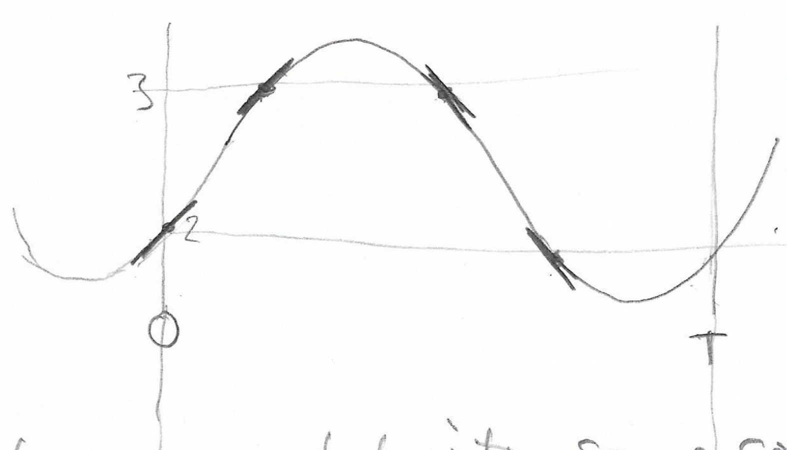
$$F_4 = \frac{1}{2}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -\hat{i} & -1 & \hat{i} \\ 1 & -1 & 1 & -1 \\ 1 & \hat{i} & -1 & -\hat{i} \end{bmatrix}$$

(b) $$F_4 \begin{bmatrix} 2 \\ 3 \\ 3 \\ 2 \end{bmatrix} = \frac{1}{2}\begin{bmatrix} & & '' & & \end{bmatrix}\begin{bmatrix} 2 \\ 3 \\ 3 \\ 2 \end{bmatrix} = \frac{1}{2}\begin{bmatrix} 10 \\ -1-i \\ 0 \\ -1+i \end{bmatrix}$$

(c) $$x' = \underset{= F_4^*}{F_4^{-1}} \cdot \left(\frac{1}{2}\begin{bmatrix} 10 \\ -1-i \\ 0 \\ -1+i \end{bmatrix} * \begin{array}{c} \begin{bmatrix} 0 \\ 1 \\ 0 \\ -1 \end{bmatrix} \end{array} \begin{array}{l} \leftarrow k=0 \\ \leftarrow k=n/2-1 \\ \leftarrow k=n/2, -n/2 \\ \leftarrow k=-\frac{n}{2}+1 \end{array} \cdot \frac{2\pi \hat{i}}{\underset{\text{length of interval}}{T}} \right)$$

$$= \frac{2\pi \hat{i}}{4T}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & \hat{i} & -1 & -\hat{i} \\ 1 & -1 & 1 & -1 \\ 1 & -\hat{i} & -1 & -\hat{i} \end{bmatrix}\begin{bmatrix} 0 \\ -1-i \\ 0 \\ 1-i \end{bmatrix} = \frac{2\pi \hat{i}}{2T}\begin{bmatrix} -\hat{i} \\ -\hat{i} \\ \hat{i} \\ \hat{i} \end{bmatrix}$$

$$= \frac{\pi}{T}\begin{bmatrix} 1 \\ 1 \\ -1 \\ -1 \end{bmatrix}$$



See plot below, generated with same code we used in class.

Using the same code we used in class for trig interpolation and differentiation, we can check that our spectral derivative vector above is correct. Here T=1. Note the values at the gridpoints are +-pi, as computed above.

```
for example in [1]:

        #figure(example,figsize=(15,10))
        fig, axes = plt.subplots(2,1,figsize=(15,10))

        if example==1:
                mytitle = 'vector [2,3,3,2]'
                c = 0.
                d = 1.
                L = d-c
                x = np.array([2,3,3,2.])
                n = len(x)
                t = L*np.arange(n)/float(n)
```