Matrix form

$\vec{u}_j$

k/h^2

$u(x_0, t_j)$

$u_{0,j}$

$$\begin{bmatrix} u_{1,j+1} \\ \vdots \\ u_{N,j+1} \end{bmatrix} = \begin{bmatrix} u_{1,j} \\ \vdots \\ u_{N,j} \end{bmatrix} + \lambda \begin{bmatrix} -2 & 1 & 0 & \cdots & 0 \\ 1 & -2 & 1 & \cdots & 0 \\ 0 & 1 & -2 & 1 & 0 \cdots 0 \\ \vdots & & & & \\ 0 & \cdots & 0 & 1 & -2 \end{bmatrix} \begin{bmatrix} u_{1,j} \\ \vdots \\ u_{N,j} \end{bmatrix} + \begin{bmatrix} f_{1,j} \\ \vdots \\ f_{N,j} \end{bmatrix}$$

with $u_{0,j} = 0$, $u_{N+1,j} = 0$ always

$1 \quad u_{N+1,j}$

$\vec{u}_{j+1}$

Matrix form

$$\begin{bmatrix} u_{1,j+1} \\ \vdots \\ u_{N,j+1} \end{bmatrix} = \begin{bmatrix} u_{1,j} \\ \vdots \\ u_{N,j} \end{bmatrix} + \lambda \begin{bmatrix} -2 & 1 & 0 & \cdots & 0 \\ 1 & -2 & 1 & \cdots & 0 \\ 0 & 1 & -2 & 1 & 0 \cdots 0 \\ \vdots & & & & \\ 0 & \cdots & 0 & 1 & -2 \end{bmatrix} \begin{bmatrix} u_{1,j} \\ \vdots \\ u_{N,j} \end{bmatrix} + k \begin{bmatrix} f_{1,j} \\ \vdots \\ f_{N,j} \end{bmatrix}$$

with $u_{0,j} = 0$, $u_{N+1,j} = 0$ always

Why is this OK?

Matrix form

$$\begin{bmatrix} u_{1,j+1} \\ \vdots \\ u_{N,j+1} \end{bmatrix} = \begin{bmatrix} u_{1,j} \\ \vdots \\ u_{N,j} \end{bmatrix} + \lambda \begin{bmatrix} -2 & 1 & 0 & \cdots & 0 \\ 1 & -2 & 1 & \cdots & 0 \\ 0 & 1 & -2 & 1 & 0 \cdots 0 \\ \vdots & & & & \\ 0 & \cdots & 0 & 1 & -2 \end{bmatrix} \begin{bmatrix} u_{1,j} \\ \vdots \\ u_{N,j} \end{bmatrix} + k \begin{bmatrix} f_{1,j} \\ \vdots \\ f_{N,j} \end{bmatrix}$$

with $u_{0,j} = 0$, $u_{N+1,j} = 0$ always

$$\vec{u}_{j+1} = \underbrace{\begin{bmatrix} 1-2\lambda & \lambda & 0 & \cdots & 0 \\ \lambda & 1-2\lambda & \lambda & 0 \cdots 0 & \\ & \ddots & & & \\ & & 0 & \lambda & 1-2\lambda \end{bmatrix}}_{A} \vec{u}_j + k\vec{f}_j$$

Direct Method:

$$\vec{u}_{j+1} = A\,\vec{u}_j + h\vec{f}_j$$

$4 \cdot 9$ doubles (64 bit)

$N \times N$

Full vs Tridiagonal

$n \times n$ $\quad$ $n \times 1$

Ax

	Full	Tridiagonal
Ax	$O(n^2)$	$O(n)$
$Ax = b$	$O(n^3)$ (LU factorization)	$O(n)$ (Tridiagonal solver)

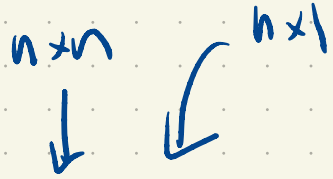
Full vs Tridiagonal

$n \times n$ $\downarrow$ A x $\leftarrow$ $n \times 1$

	Full	Tridiagonal
Ax	$O(n^2)$	$O(n)$
$Ax = b$	$O(n^3)$ (LU factorization)	$O(n)$ (Tridiagonal solver)

Relative labor if $n = 1000$:

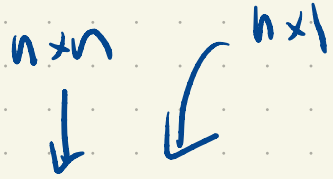
Full vs Tridiagonal



	Full	Tridiagonal	
Ax	$O(n^2)$	$O(n)$	$\times 1000$
$Ax=b$	$O(n^3)$ (LU factorization)	$O(n)$ (Tridiagonal solver)	

Relative labor if $n = 1000$:

Full vs Tridiagonal



	Full	Tridiagonal	
Ax	$O(n^2)$	$O(n)$	$\times 1000$
$Ax=b$	$O(n^3)$ (LU factorization)	$O(n)$ (Tridiagonal solver)	$\times 1000000$

Relative labor if $n = 1000$:

MATLAB

sparse, spzeros

A\b will notice the matrix is tridiagonal. Easy!

Python

scipy.linalg.solve_banded

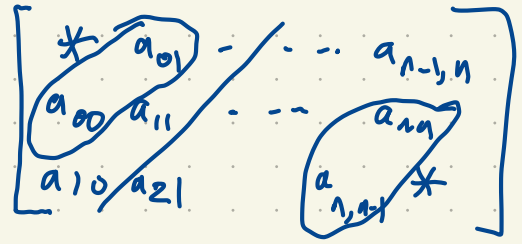
Python

scipy.linalg.solve_banded

(l, u)

l = # bands below

u = # above



$A = \begin{bmatrix} a_{00} & a_{01} & 0 & \cdots \\ a_{10} & a_{11} & a_{12} & \cdots \\ & \ddots & \ddots & \ddots \\ & & a_{n,n-1} & a_{nn} \end{bmatrix}$

* is ignored

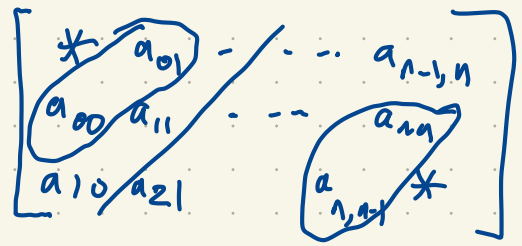
Python

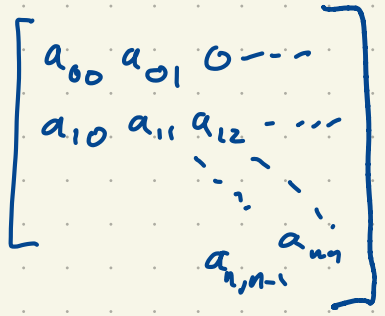
scipy.linalg.solve_banded

(l, u)

↑ # bands below

↑ # above



$A =$ 

* is ignored

$(1, 1)$

$$\begin{bmatrix} * & \lambda & \cdots & \lambda \\ 1-2\lambda & 1-2\lambda & \cdots & 1-2\lambda \\ \lambda & \cdots & \cdots & \lambda \; * \end{bmatrix}$$

super easy to construct.

Ad

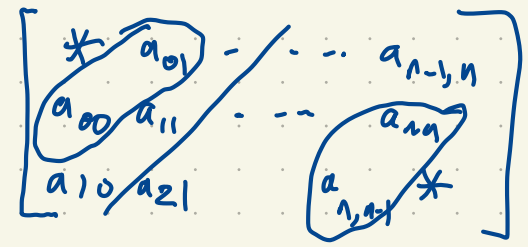
Python

scipy.linalg.solve_banded

(l, u)

l = # bands below

u = # above



* is ignored

$$A = \begin{bmatrix} a_{00} & a_{01} & 0 & \cdots \\ a_{10} & a_{11} & a_{12} & \cdots \\ & \ddots & \ddots & \ddots \\ & & a_{n,n-1} & a_{nn} \end{bmatrix}$$

$(1, 1)$

$$\begin{bmatrix} * & \lambda & \cdots & \lambda \\ 1-2\lambda & 1-2\lambda & \cdots & 1-2\lambda \\ \lambda & \cdots & \lambda & * \end{bmatrix}$$

super easy to construct.

Ad

scipy.sparse.spdiags(Ad, (1, 0, -1), N, N)

A.dot(b)

Demo

Direct Method

$$\vec{u}_{j+1} = (I - \lambda D)\vec{u}_j + k\vec{f}_j$$

is <u>exactly</u> Euler's Method applied to

$$u' = \frac{1}{h^2} D u + f$$

$$D = \begin{bmatrix} -2 & 1 & & 0 \\ 1 & -2 & \ddots & \\ & \ddots & \ddots & 1 \\ 0 & & 1 & -2 \end{bmatrix}$$

Now $\frac{1}{h^2} D$ is supposed to approximate ∂_x^2

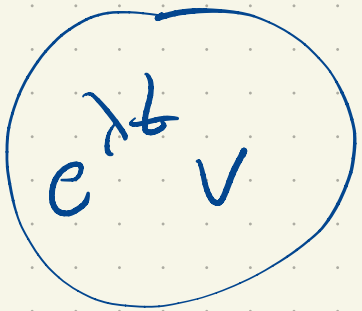
And $\partial_x^2 v_j = -j^2 \pi^2 v_j$ $\qquad v_j = \sin(j\pi x)$

So reasonable to expect $\frac{1}{h^2} D$ has eigenvalues

$\sim -\pi^2, -2^2\pi^2, \ldots, -N^2\pi^2$

$\hookrightarrow$ large, negative.

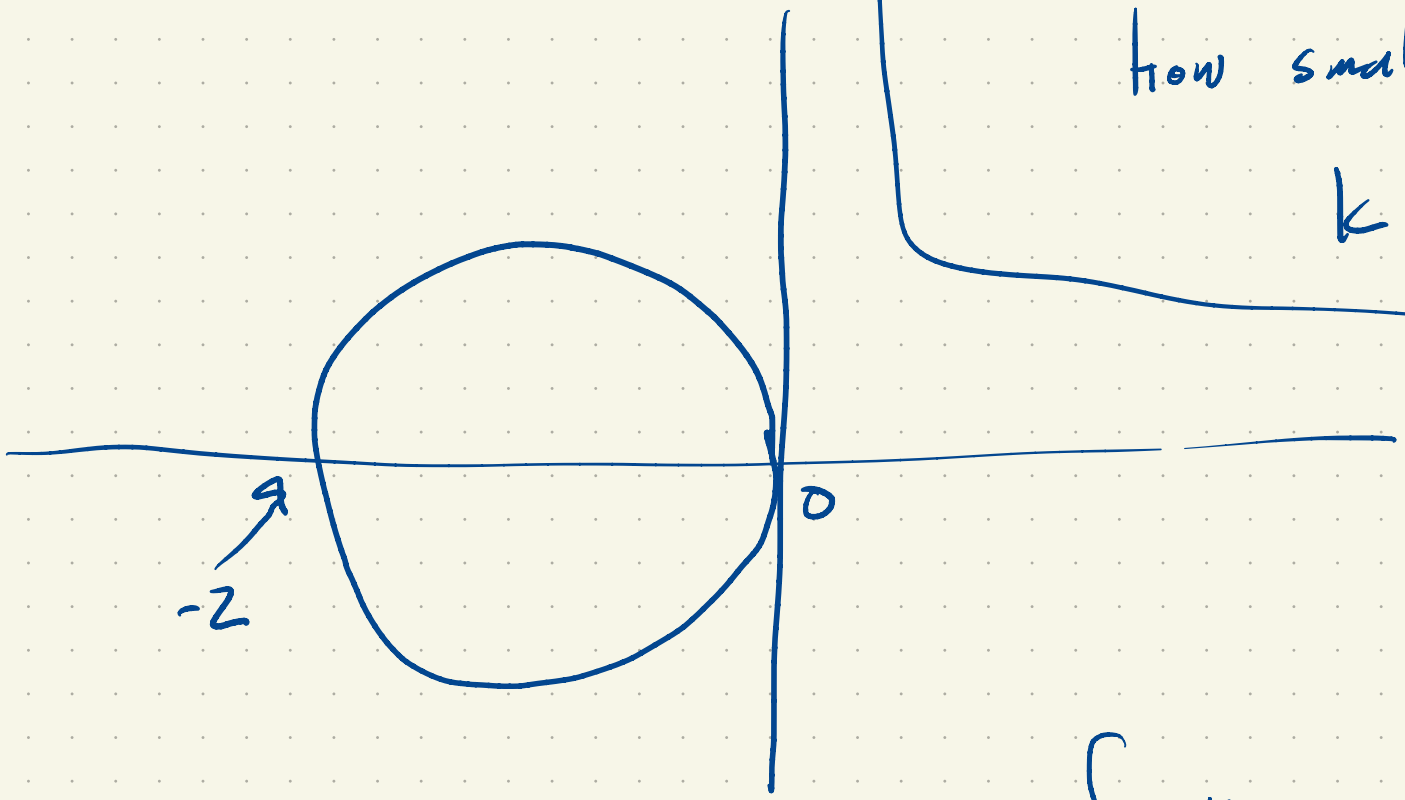
Transients!



Region of abs stability:

$-\pi^2, -4\pi^2, \ldots, -N^2\pi^2$

how small should k be?



$u_t u_{xt}$ $\quad - u_{tx} u_t$

$\quad - u_{xx} u_{tx}$

$\int u u_t$

$\int u u_{xx} = - \int u_x u_x$