# Boundary value problems

Consider $-u''(x) = f(x, u, u')$ for $x \in [a, b]$ where $u(a) = u_a$, $u(b) = u_b$ in 1D

and

$-u_{xx}(\bar{x}) - u_{yy}(\bar{x}) = f(\bar{x})$ for $x \in \Omega = [0, 1]^2$ with $u(\bar{x}) = u_0$ on $\delta\Omega$ in 2D.

# Approximation of 2nd order derivatives

Taylor series (write $h$ for $\delta x$):

$$u(x+h) = u(x) + u'(x)h + u''(x)\frac{h^2}{2!} + u'''(x)\frac{h^3}{3!} + u^{(4)}(x)\frac{h^4}{4!} + u^{(5)}(x)\frac{h^5}{5!} + \cdots$$

and

$$u(x-h) = u(x) - u'(x)h + u''(x)\frac{h^2}{2!} - u'''(x)\frac{h^3}{3!} + u^{(4)}(x)\frac{h^4}{4!} - u^{(5)}(x)\frac{h^5}{5!} + \cdots$$

Add:

$$u(x+h) + u(x-h) = 2u(x) + u''(x)h^2 + u^{(4)}(x)\frac{h^4}{12} + \cdots$$

so

$$u''(x) = \frac{u(x+h) - 2u(x) + u(x-h)}{h^2} - u^{(4)}(x)\frac{h^4}{12} + \cdots$$

Numerical scheme:

$$-\frac{u(x+h) - 2u(x) + u(x-h)}{h^2} = f(x, u(x), u'(x))$$

(2nd order PDEs are very common!)

# This leads to linear algebra

$$-\frac{u(x+h) - 2u(x) + u(x-h)}{h^2} = f(x, u(x), u'(x))$$

Equally spaced points on $[0, 1]$: $x_k = kh$ where $h = 1/n$, then

$$-u_{k+1} + 2u_k - u_{k-1} = h^2 f(x_k, u_k, u'_k) \quad \text{for } k = 1, \ldots, n-1$$

Written as matrix equation:

$$\begin{pmatrix} 2 & -1 & & \\ -1 & 2 & -1 & \\ & \ddots & \ddots & \ddots \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ \vdots \end{pmatrix} = \begin{pmatrix} h^2 f_1 + u_0 \\ h^2 f_2 \\ \vdots \end{pmatrix}$$

# Matrix properties

- Very sparse, banded
- Symmetric (only because 2nd order problem)
- Sign pattern: positive diagonal, nonpositive off-diagonal (true for many second order methods)
- Positive definite (just like the continuous problem)

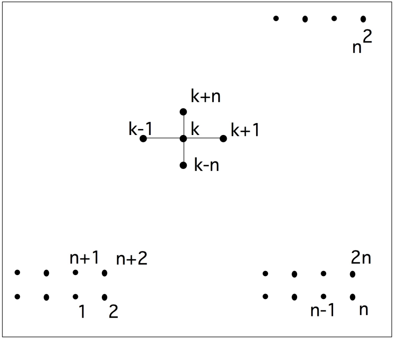
# Sparse matrix in 2D case

Sparse matrices so far were tridiagonal: only in 1D case.

Two-dimensional: $-u_{xx} - u_{yy} = f$ on unit square $[0,1]^2$

Difference equation:

$$4u(x,y) - u(x+h,y) - u(x-h,y) - u(x,y+h) - u(x,y-h) = h^2 f(x,y)$$




This is a graph!

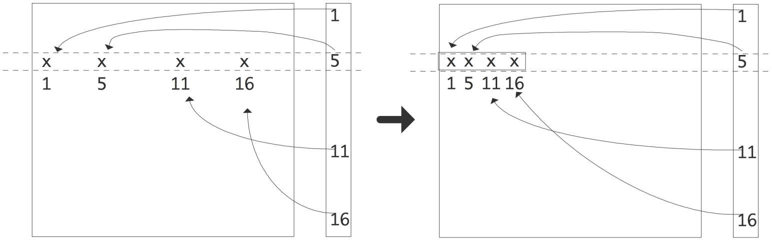
# Sparse matrix from 2D equation

$$
\left(\begin{array}{ccccc|ccccc|cc}
4 & -1 & & & \emptyset & -1 & & & & \emptyset & & \\
-1 & 4 & 1 & & & & -1 & & & & & \\
 & \ddots & \ddots & \ddots & & & & \ddots & & & & \\
 & & \ddots & \ddots & -1 & & & & \ddots & & & \\
\emptyset & & & -1 & 4 & \emptyset & & & & -1 & & \\
\hline
-1 & & & & \emptyset & 4 & -1 & & & & -1 & \\
 & -1 & & & & -1 & 4 & -1 & & & & -1 \\
 & \uparrow & \ddots & & & \uparrow & \uparrow & \uparrow & & & & \uparrow \\
 & k-n & & & & k-1 & k & k+1 & & -1 & & k+n \\
 & & & & -1 & & & & -1 & 4 & & \\
\hline
 & & & & & \ddots & & & & & \ddots & \\
\end{array}\right)
$$

# Sparse matrix storage

Matrix above has many zeros: $n^2$ elements but only $O(n)$ nonzeros. Big waste of space to store this as square array.

Matrix is called 'sparse' if there are enough zeros to make specialized storage feasible.



# Compressed Row Storage

$$A = \begin{pmatrix} 10 & 0 & 0 & 0 & -2 & 0 \\ 3 & 9 & 0 & 0 & 0 & 3 \\ 0 & 7 & 8 & 7 & 0 & 0 \\ 3 & 0 & 8 & 7 & 5 & 0 \\ 0 & 8 & 0 & 9 & 9 & 13 \\ 0 & 4 & 0 & 0 & 2 & -1 \end{pmatrix} . \qquad (1)$$

Compressed Row Storage (CRS): store all nonzeros by row, their column indices, pointers to where the columns start (1-based indexing):

| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| val | 10 | -2 | 3 | 9 | 3 | 7 | 8 | 7 | 3 ··· 9 | 13 | 4 | 2 | -1 |
| col_ind | 1 | 5 | 1 | 2 | 6 | 2 | 3 | 4 | 1 ··· 5 | 6 | 2 | 5 | 6 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| row_ptr | 1 | 3 | 6 | 9 | 13 | 17 | 20 |

.

# Sparse matrix operations

Most common operation: matrix-vector product

```
for (row=0; row<nrows; row++) {
  s = 0;
  for (icol=ptr[row]; icol<ptr[row+1]; icol++) {
    int col = ind[icol];
    s += a[aptr] * x[col];
    aptr++;
  }
  y[row] = s;
}
```

Operations with changes to the nonzero structure are much harder!

Indirect addressing of `x` gives low spatial and temporal locality.

# The graph view of things

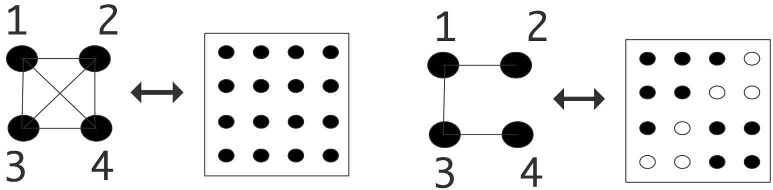
Poisson eq:

$$4u_k - u_{k-1} - u_{k+1} - u_{k-n} - u_{k+n} = f_k$$

Consider a graph where $\{u_k\}_k$ are the edges
and $(u_i, u_j)$ is an edge iff $a_{ij} \neq 0$.

This is the (adjacency) graph of a sparse matrix.

# Graph theory of sparse matrices



# Initial Boundary value problem

Heat conduction in a rod $T(x,t)$ for $x \in [a,b]$, $t > 0$:

$$\frac{\partial}{\partial t} T(x,t) - \alpha \frac{\partial^2}{\partial x^2} T(x,t) = q(x,t)$$

- Initial condition: $T(x,0) = T_0(x)$
- Boundary conditions: $T(a,t) = T_a(t)$, $T(b,t) = T_b(t)$
- Material property: $\alpha > 0$ is thermal diffusivity
- Forcing function: $q(x,t)$ is externally applied heating.

The equation $u''(x) = f$ above is the steady state.

# Discretization

Space discretization: $x_0 = a$, $x_n = b$, $x_{j+1} = x_j + \Delta x$
Time discretiation: $t_0 = 0$, $t_{k+1} = t_k + \Delta t$
Let $T_j^k$ approximate $T(x_j, t_k)$

Space:

$$\frac{\partial}{\partial t} T(x_j, t) - \alpha \frac{T(x_{j-1}, t) - 2T(x_j, t) + T(x_{j+1}, t)}{\Delta x^2} = q(x_j, t)$$

Explicit time stepping:

$$\frac{T_j^{k+1} - T_j^k}{\Delta t} - \alpha \frac{T_{j-1}^k - 2T_j^k + T_{j+1}^k}{\Delta x^2} = q_j^k$$

Implicit time stepping:

$$\frac{T_j^{k+1} - T_j^k}{\Delta t} - \alpha \frac{T_{j-1}^{k+1} - 2T_j^{k+1} + T_{j+1}^{k+1}}{\Delta x^2} = q_j^{k+1}$$

# Computational form: explicit

$$T_j^{k+1} = T_j^k + \frac{\alpha \Delta t}{\Delta x^2}(T_{j-1}^k - 2T_j^k + T_{j+1}^k) + \Delta t q_j^k$$

This has an explicit form:

$$\underset{\sim}{T}^{k+1} = \left( I + \frac{\alpha \Delta t}{\Delta x^2} \right) K \underset{\sim}{T}^k + \Delta t \underset{\sim}{q}^k$$

# Computational form: implicit

$$T_j^{k+1} - \frac{\alpha \Delta t}{\Delta x^2}(T_{j-1}^{k+1} - 2T_j^{k+1} + T_{j+1}^{k+1}) = T_j^k + \Delta t q_j^k$$

This has an implicit form:

$$\left(I - \frac{\alpha \Delta t}{\Delta x^2} K\right) \underset{\sim}{T}^{k+1} = \underset{\sim}{T}^k + \Delta t \underset{\sim}{q}^k$$

Needs to solve a linear system in every time step

# Stability of explicit scheme

Let $q \equiv 0$; assume $T_j^k = \beta^k e^{i\ell x_j}$; for stability we require $|\beta| < 1$:

$$\begin{aligned}
T_j^{k+1} &= T_j^k + \frac{\alpha \Delta t}{\Delta x^2}(T_{j-1}^k - 2T_j^k + T_{j+1}^k) \\
\Rightarrow \beta^{k+1} e^{i\ell x_j} &= \beta^k e^{i\ell x_j} + \frac{\alpha \Delta t}{\Delta x^2}(\beta^k e^{i\ell x_{j-1}} - 2\beta^k e^{i\ell x_j} + \beta^k e^{i\ell x_{j+1}}) \\
\Rightarrow \beta &= 1 + 2\frac{\alpha \Delta t}{\Delta x^2}[\frac{1}{2}(e^{i\ell \Delta x} + e^{-\ell \Delta x}) - 1] \\
&= 1 + 2\frac{\alpha \Delta t}{\Delta x^2}(\cos(\ell \Delta x) - 1)
\end{aligned}$$

$$\frac{\beta^{k+1}}{\beta^k} = 1 + 2\frac{\alpha\Delta t}{\Delta x^2}(\cos(\ell\Delta x) - 1)$$

To get $|\beta| < 1$:

- $2\frac{\alpha\Delta t}{\Delta x^2}(\cos(\ell\Delta x) - 1) < 0$: automatic
- $2\frac{\alpha\Delta t}{\Delta x^2}(\cos(\ell\Delta x) - 1) > -2$: needs $2\frac{\alpha\Delta t}{\Delta x^2} < 1$, that is

$$\Delta t < \frac{\Delta x^2}{2\alpha}$$

big restriction on size of time steps

# Stability of implicit scheme

$$
\begin{aligned}
T_j^{k+1} - \frac{\alpha \Delta t}{\Delta x^2}\left(T_{j_1}^{k+1} - 2T_j^{k+1} + T_{j+1}^{k+1}\right) &= T_j^k \\
\Rightarrow \beta^{k+1} e^{i\ell\Delta x} - \frac{\alpha \Delta t}{\Delta x^2}\left(\beta^{k+1} e^{i\ell x_{j-1}} - 2\beta^{k+1} e^{i\ell x_j} + \beta^{k+1} e^{i\ell x_{j+1}}\right) &= \beta^k e^{i\ell x_j}
\end{aligned}
$$

$$
\begin{aligned}
\Rightarrow \beta^{-1} &= 1 + 2\frac{\alpha \Delta t}{\Delta x^2}(1 - \cos(\ell \Delta x)) \\
\beta &= \frac{1}{1 + 2\frac{\alpha \Delta t}{\Delta x^2}(1 - \cos(\ell \Delta x))}
\end{aligned}
$$

Noting that $1 - \cos(\ell \Delta x) > 0$, the condition $|\beta| < 1$ always satisfied: method always stable.