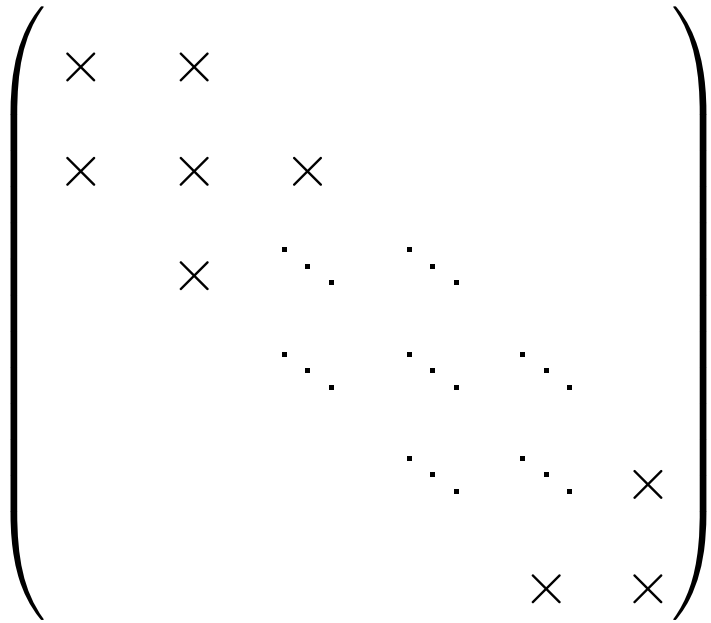
Tridiagonal systems I

Sparse matrices are ones whose entries are mostly zeros. A simple type of sparse matrices is a tridiagonal matrix like the following



where each $\times$ denotes a nonzero entry and the rest is zero. Gaussian elimination is simpler for tridiagonal systems, because there is only one row below the main diagonal that needs zeroing and that row has only few nonzero entries.

FLOPs

In general, processing the n-th column an $n \times n$ tridiagonal system requires $(3\mathrm{M} + 2\mathrm{A})$ FLOPs, and we need to process from the second row to the last ($n - 1$ rows in total).
Hence the total number of FLOPs for triangularization is $\mathcal{O}(n)$.

After triangularizing, the back substitution also requires $\mathcal{O}(n)$ ($2\mathrm{M} + 1\mathrm{A}$ per row except for the last row).

1-D two-point boundary value problem I

Consider the equation

$$-cu_{xx}(x) + u(x) = f(x), \qquad u(0) = b_0, u(1) = b_1. \qquad (*)$$

First, we divide (grid) up the domain $[0, 1]$ into n segments each with length $h = \frac{1}{n}$ and denote the endpoints by $x_0 = 0, x_1 = \frac{1}{n}, \cdots, x_k = \frac{k}{n}$ and $x_n = 1$.

1-D two-point boundary value problem II

Recall the second derivative approximation

$$u_{xx}(x) = \frac{u(x-h) + u(x+h) - 2u(x)}{h^2} + \mathcal{O}(h^2).$$

Therfore,

$$u_{xx}(x_k) \approx \frac{u_{k-1} + u_{k+1} - u_k}{h^2},$$

where $u_k = u(x_k)$. Using this approximation in $(*)$, we have

$$-c\frac{u_{k-1} + u_{k+1} - 2u_k}{h^2} + u_k = f(x_k).$$

The equation we just derived applies to u_2 through u_{n-2}.

1-D two-point boundary value problem III

For u_1 and u_{n-1}, we note that u_0 and u_n are specified as the boundary conditions of $(*)$ and therefore,

$$\begin{aligned} -\tfrac{c}{h^2} b_0 + (1 + \tfrac{2u_k}{h^2}) u_1 - \tfrac{c}{h^2} u_2 &= f_1, \\ -\tfrac{c}{h^2} u_{n-2} + (1 + \tfrac{2u_k}{h^2}) u_{n-1} - \tfrac{c}{h^2} b_1 &= f_{n-1}, \end{aligned}$$

may be rearranged as

$$\begin{aligned} (1 + \tfrac{2u_k}{h^2}) u_1 - \tfrac{c}{h^2} u_2 &= f_1 + \tfrac{c}{h^2} b_0, \\ -\tfrac{c}{h^2} u_{n-2} + (1 + \tfrac{2u_k}{h^2}) u_{n-1} &= f_{n-1} + \tfrac{c}{h^2} b_1. \end{aligned}$$

Putting into the matrix form, we get a tridiagonal linear system that corresponds to $(*)$.

Evolution PDEs I

Consider the PDE

$$u_t(t,x) = c_1 u_{xx}(t,x) + c_2 u(t,x) + \hat{f}(t,x),$$

with boundary conditions

$$u(0,x) = w(x), \quad u(t,0) = v_0, \quad u(t,1) = v_1.$$

Evolution PDEs II

In this settings, we divide both the spatial domain $[0, 1]$ as before and the time domain $[0, T]$. Let $h = \frac{1}{n}$ and $\delta = \frac{1}{K}$. Let $t_k = k\delta$ and $x_l = kh$ and denote $u_{k,l} = u(t_k, x_l)$ and $f_{k,l} = \hat{f}(t_k, x_l)$.
Using the following derivative approximations from before,

$$u_{xx}(t_k, x_l) = \frac{1}{h^2}(u_{k,l-1} - 2u_{k,l} + u_{k,l+1}) + \mathcal{O}(h^2),$$
$$u_t(t_{k+1}, x_l) = \frac{1}{\delta}(u_{k+1,l} - u_{k,l}) + \mathcal{O}(\delta),$$

we get

$$\frac{1}{\delta}(u_{k+1,l} - u_{k,l}) = \frac{c_1}{h^2}(u_{k,l-1} - 2u_{k,l} + u_{k,l+1}) + c_2 u_{k,l} + f_{k,l}.$$

Evolution PDEs III

The boundary conditions can be treated the same way as for the boundary value problem before.
If we put $u_{k,1}, u_{k,2}, \cdots$ into a vector $\vec{u}_k$, then the formula above is in the form

$$\vec{u}_{k+1} = A\vec{u}_k + \vec{b}_k,$$

where A is a tridiagonal matrix and $\vec{b}_k$ is a vector that depends on $\hat{f}$ and the boundary conditions.