Overview

- Special cases of LU factorization
  - banded matrices
  - tridiagonal matrices, Richardson algorithm
  - block form, Schur complement
  - incomplete factorization of matrices from PDE finite difference methods
- Stability
  - theoretical results
  - experiments
- `f2py`: efficient computational kernel (Fortran) called from Python

- Partial pivoting (within the current column)

for $i = 1$ to m: $p_i = i$
for $s = 1$ to $m - 1$
  $\boldsymbol{p}$=pivot(s)
  for $i = s + 1$ to m
    $\ell = -a_{p(s)\,i} / a_{p(s)\,s}$
    for $j = s + 1$ to m
      $a_{p(i)\,j} = a_{p(i)\,j} + \ell\, a_{p(s)\,j}$

- $\boldsymbol{A} \in \mathbb{C}^{m \times m}$, $\boldsymbol{A}_{ij} \in \mathbb{C}^{q \times q}$, bandwidth $B = 3q$

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{A}_{11} & \boldsymbol{A}_{12} & \boldsymbol{0} & \dots & \boldsymbol{0} \\ \boldsymbol{A}_{21} & \boldsymbol{A}_{22} & \boldsymbol{A}_{23} & \dots & \boldsymbol{0} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \boldsymbol{0} & \dots & \boldsymbol{A}_{b-1,b-2} & \boldsymbol{A}_{b-1,b-1} & \boldsymbol{A}_{b-1,b} \\ \boldsymbol{0} & \dots & \boldsymbol{0} & \boldsymbol{A}_{b,b-1} & \boldsymbol{A}_{bb} \end{bmatrix}$$

- Partial pivoting (within the current column, componentwise algorithm)

for $i = 1$ to m: $p_i = i$
for $s = 1$ to $m - 1$
  $\boldsymbol{p}$=pivot(s)
  for $i = s + 1$ to $\min(s + 2q - 1, m)$
    $\ell = -a_{p(s)\,i} / a_{p(s)\,s}$
    for $j = s + 1$ to $\min(s + 2q - 1, m)$
      $a_{p(i)\,j} = a_{p(i)\,j} + \ell\, a_{p(s)\,j}$

- Consider scalar Gauss multiplier $\ell = -a_{p(s)\,i} / a_{p(s)\,s}$. For $m = 2$, no pivot

$$a_{22} \leftarrow a_{22} - \frac{a_{21}}{a_{11}} a_{12}, \begin{bmatrix} 1 & 0 \\ -\frac{a_{21}}{a_{11}} & 1 \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ 0 & a_{22} - \frac{a_{21}}{a_{11}} a_{12} \end{bmatrix}$$

- Extend to block structured matrices

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{A}_{11} & \boldsymbol{A}_{12} \\ \boldsymbol{A}_{21} & \boldsymbol{A}_{22} \end{bmatrix}$$

- The Gauss multiplier is now (Schur's formula)

$$\boldsymbol{L}\,\boldsymbol{A} = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ -\boldsymbol{A}_{21}\boldsymbol{A}_{11}^{-1} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{A}_{11} & \boldsymbol{A}_{12} \\ \boldsymbol{A}_{21} & \boldsymbol{A}_{22} \end{bmatrix} = \begin{bmatrix} \boldsymbol{A}_{11} & \boldsymbol{A}_{12} \\ \boldsymbol{0} & \boldsymbol{A}_{22} - \boldsymbol{A}_{21}\boldsymbol{A}_{11}^{-1}\boldsymbol{A}_{12} \end{bmatrix}$$

- $\boldsymbol{A} \in \mathbb{C}^{m \times m}$,$\boldsymbol{A}_{ij} \in \mathbb{C}^{q \times q}$, bandwidth $B = 3q$

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{A}_{11} & \boldsymbol{A}_{12} & \mathbf{0} & \ldots & \mathbf{0} \\ \boldsymbol{A}_{21} & \boldsymbol{A}_{22} & \boldsymbol{A}_{23} & \ldots & \mathbf{0} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \mathbf{0} & \ldots & \boldsymbol{A}_{b-1,b-2} & \boldsymbol{A}_{b-1,b-1} & \boldsymbol{A}_{b-1,b} \\ \mathbf{0} & \ldots & \mathbf{0} & \boldsymbol{A}_{b,b-1} & \boldsymbol{A}_{bb} \end{bmatrix}$$

- Partial pivoting (within the current column, componentwise algorithm)

for $s = 1$ to $b - 1$

$$\boldsymbol{A}_{s+1,s+1} = \boldsymbol{A}_{s+1,s+1} - \boldsymbol{A}_{s+1,s}\boldsymbol{A}_{s,s}^{-1}\boldsymbol{A}_{s,s+1}$$

- For $q = 1 \rightarrow$ tridiagonal matrix, LU becomes Richardson algorithm

- Complexity $\mathcal{O}(2bq^3) = \mathcal{O}\left(2\frac{m}{q}q^3\right) = \mathcal{O}(2mq^2)$

- Poisson equation $\nabla^2 u = f$ in $(0,1)\times(0,1)$, $h=1/(n-1)$, $x_i = ih$, $y_j = jh$

$$\nabla^2 u = \frac{1}{h^2}(u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1} - 4u_{i,j}) = f_{i,j}$$

- Above is expressed $\boldsymbol{A}\boldsymbol{u} = \boldsymbol{f}$, Gaussian elimination does not preserve sparsity

```
octave> n=3; m=n^2;
        d=-4*ones(m,1); sd=ones(m-1,1); ssd=ones(m-n,1);
```

```
octave> A=diag(ssd,-n)+diag(sd,-1)+diag(d,0)+diag(sd,1)+diag(ssd,n)
```

```
octave> A
```

$$\left(\begin{array}{ccccccccc}
-4 & 1 & 0.0 & 1 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\
1 & -4 & 1 & 0.0 & 1 & 0.0 & 0.0 & 0.0 & 0.0 \\
0.0 & 1 & -4 & 1 & 0.0 & 1 & 0.0 & 0.0 & 0.0 \\
1 & 0.0 & 1 & -4 & 1 & 0.0 & 1 & 0.0 & 0.0 \\
0.0 & 1 & 0.0 & 1 & -4 & 1 & 0.0 & 1 & 0.0 \\
0.0 & 0.0 & 1 & 0.0 & 1 & -4 & 1 & 0.0 & 1 \\
0.0 & 0.0 & 0.0 & 1 & 0.0 & 1 & -4 & 1 & 0.0 \\
0.0 & 0.0 & 0.0 & 0.0 & 1 & 0.0 & 1 & -4 & 1 \\
0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 1 & 0.0 & 1 & -4
\end{array}\right)$$

- Since $\boldsymbol{L}\boldsymbol{U} = \boldsymbol{A}$ loses sparsity pattern, construct $\tilde{\boldsymbol{L}}\tilde{\boldsymbol{U}} \cong \boldsymbol{A}$ with $\tilde{\boldsymbol{L}}, \tilde{\boldsymbol{U}}$ with sparsity as the corresponding lower and upper parts of $\boldsymbol{A}$