MATH662: L16 Gaussian elimination experiments

Overview

- `f2py`: efficient computational kernel (Fortran) called from Python

f2py

- Write computational kernel in Fortran, e.g. `MATH662/examples/f2py/add2`

```
subroutine add2(a,b,c)
  integer, parameter :: dp = 8
  real (dp), intent(in) :: a,b
  real (dp), intent(out) :: c
end subroutine add2
```

- Compile with f2py wrapper around Fortran compiler, use Makefile:

```
# Simple f2py usage example: c=a+b
add2.so: add2.f90
        f2py3 -c add2.f90 -m add2
```

LU algorithm with pivoting

- 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 $m$
            $a_{p(i)\,j} = a_{p(i)\,j} + \ell\, a_{p(s)\,j}$
```

-

LU factorization with pivoting for banded matrices

- $\boldsymbol{A} \in \mathbb{C}^{m \times m}$,$\boldsymbol{A}_{ij} \in \mathbb{C}^{q \times q}$, 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 $\min(s + 2q - 1, m)$
            $a_{p(i)\,j} = a_{p(i)\,j} + \ell\, a_{p(s)\,j}$
```