Numerics (Part III)

[Lecture 1] Triangular Linear Systems

Alessandro Antonucci
alessandro.antonucci@supsi.ch

Matrix Representation of a Linear System

$$\hat{A} = \begin{bmatrix} 1 & -1 & 2 \\ 2 & -1 & -1 \\ 1 & 1 & 1 \end{bmatrix} \qquad \vec{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \qquad \vec{b} = \begin{bmatrix} 2 \\ 0 \\ 3 \end{bmatrix}$$

$$\begin{cases} x_1 - x_2 + 2x_3 = 2 \\ 2x_1 - x_2 - x_3 = 0 \\ x_1 + x_2 + x_3 = 3 \end{cases} \qquad \hat{A} \cdot \vec{x} = \vec{b}$$

Matrix Representation of a Triangular System

$$\hat{A} = \begin{bmatrix} 1 & -1 & 2 \\ 0 & -1 & -1 \\ 0 & 0 & 1 \end{bmatrix} \quad \vec{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \quad \vec{b} = \begin{bmatrix} 2 \\ 0 \\ 3 \end{bmatrix}$$

$$\begin{cases} x_1 - x_2 + 2x_3 = 2 \\ \quad\;\; - x_2 - x_3 = 0 \\ \qquad\qquad\; x_3 = 3 \end{cases} \quad \begin{cases} x_1 = \; 2 + x_2 - 2x_3 = 2 - 3 - 6 = -7 \\ x_2 = \; -x_3 = -3 \\ x_3 = \; 3 \end{cases}$$

Solving by Substitution

- INPUT: triangular system $\hat{A}, \vec{b}$ ($\hat{A}$ triangular)
- OUTPUT: solution $\vec{x}^*$ such that $\hat{A} \cdot \vec{x}^* = \vec{b}$

Bottom-up substitution + right-left

(stop just before the diagonal)

`n` matrix dimension

```
for i=n .. 1
    for j=n .. i
        x(i)= ...
    end
end
```

solution vector $\vec{x}^*$ obtained by starting from the last element

Manual 5×5 in Python . . .

```
x[4] = b[4]/A[4][4]
tmp = A[3][4]*x[4]
x[3] = (b[3]-tmp)/A[3][3]
tmp = A[2][4]*x[4] + A[2][3]*x[3]
x[2] = (b[2]-tmp)/A[2][2]
tmp = A[1][4]*x[4] + A[1][3]*x[3] + A[1][2]*x[2]
x[1] = (b[1]-tmp)/A[1][1]
tmp = A[0][4]*x[4] + A[0][3]*x[3] + A[0][2]*x[2] +
x[0] = (b[0]-tmp)/A[0][0]
```

Substitution in Python ...

```
def triangular_solver(A,b):

  n = len(b)
  x = numpy.array([0.0 for _ in range(n)])
  for i in range(n-1,-1,-1):
    tmp = 0.0
    for j in range(n-1,i,-1):
      tmp += A[i][j]*x[j]
    x[i] = (b[i]-tmp)/A[i][i]
  return x
```

Computational Complexity of Substitution

- One division for each equation (n divisions)
- One subtraction and one multiplication for each upper-diagonal element
- Upper-diagonal Elements $= \frac{n^2-n}{2}$
- Number of Basic Operations $= n + 2 \cdot \frac{n^2-n}{2} = n^2$
- Quadratic Complexity $O(n^2)$