Computational Physics- Assignment I

Jay Deshmukh

1)
$$\begin{bmatrix} 1 & 0.67 & 0.33 \\ 0.45 & 1 & 0.55 \\ 0.67 & 0.33 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 2 \\ 2 \\ 2 \end{bmatrix}$$

$$C_2 \longrightarrow C_2 - 0.45 \times C_1$$

$$0.45 \times 10^0 - 0.45 \times 10^0 = 0$$

$$0.1 \times 10^1 - (0.45)(0.67 \times 10^0)$$

$$= 0.1 \times 10^1 - 0.30 \times 10^0$$

$$= 0.1 \times 10^1 - 0.03 \times 10^1 = 0.7 \times 10^0$$

$$0.55 \times 10^0 - (0.45)(0.33 \times 10^0)$$

$$= (0.55 - 0.15) \times 10^0 = 0.40 \times 10^0$$

$$0.2 \times 10^1 - 0.09 \times 10^1 = 0.11 \times 10^1$$

$$\begin{bmatrix} 1 & 0.67 & 0.33 \\ 0 & 0.70 & 0.40 \\ 0.67 & 0.33 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 2 \\ 1.1 \\ 2 \end{bmatrix}$$

$$C_3 \rightarrow C_3 - (0.67) C_1$$

$$\begin{bmatrix} 1 & 0.67 & 0.33 \\ 0 & 0.70 & 0.40 \\ 0 & -0.12 & 0.80 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 2 \\ 1.1 \\ 0.7 \end{bmatrix}$$

$$0.1 \times 10^1 - 0.22 \times 10^0$$

$$= 0.8 \times 10^0 = 0.80$$

$$C_2 \longrightarrow C_2 / 0.70$$

$$\begin{bmatrix} 1 & 0.67 & 0.33 \\ 0 & 1 & 0.57 \\ 0 & -0.12 & 0.80 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 2 \\ 1.6 \\ 0.7 \end{bmatrix}$$

$$C_3 \longrightarrow C_3 + (0.12) C_2$$

$$\begin{bmatrix} 1 & 0.67 & 0.33 \\ 0 & 1 & 0.57 \\ 0 & 0 & 0.87 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 2 \\ 1.6 \\ 0.89 \end{bmatrix}$$

$$C_3 \longrightarrow C_3 / 0.87$$

$$\begin{bmatrix} 1 & 0.67 & 0.33 \\ 0 & 1 & 0.57 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 2 \\ 1.6 \\ 1.0 \end{bmatrix}$$

$$\Rightarrow \quad x_3 = 1.0 \quad , \quad x_2 = 1.6 - 0.57 = (0.16 - 0.06) \times 10^1$$
$$= 1.0$$

$$x_1 = 2 - 0.67 - 0.33 = (0.2 - 0.07 - 0.03) \times 10^1 = 1.0$$

$$\boxed{x_1 = 1.0 \quad , \quad x_2 = 1.0 \quad , \quad x_3 = 1.0}$$

2) np.linalg.solve gives $x_1 = x_2 = x_3 = 1$. We seem to have gotten lucky!

4)

$$\underbrace{\begin{bmatrix} a_{11} & & \\ a_{21} & & \\ a_{31} & & \\ \vdots & & \\ a_{n1} & & \end{bmatrix}_{n\times n}}_{A} \underbrace{\begin{bmatrix} \\ \\ \\ \\ \\ \end{bmatrix}}_{x} = \underbrace{\begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_n \end{bmatrix}_{n\times 1}}_{b}$$

Divide first row by a_{11} $\Rightarrow$ $n+1$ divisions
↳ including b

Multiply first row by a_{i1} and subtract from i^{th} row $\Rightarrow$ $n+1$ multiplications, $n+1$ subtractions

Do this for all the $n-1$ rows

$\Rightarrow$ No. of steps : $n+1 + (n+1+n+1)(n-1)$

Now repeat the above procedure for inner $(n-1)$ dimensional matrix, and so on

$n+1$ div., $(n+1$ mul., $n+1$ subtractions$) \times (n-1)$

n div., $(n$ mul., n sub$) \times (n-2)$

$\vdots$

2 div

No. of steps to conver to upper-triangular form -

$$\sum_{j=1}^{n} \left\{ j+1 + 2(j+1)(j-1) \right\} = \sum_{j=1}^{n} \left(2j^2 + j - 1 \right)$$

$$= \frac{n(n+1)}{2} - n + \frac{2}{6} n(n+1)(2n+1)$$

$j=1 \Rightarrow$ last row, $j=n \Rightarrow$ first row

No. of back substitution steps $= \sum_{j=1}^{n} 2(j-1) = n(n+1) - 2n$

1 mul. + 1 sub.

$$\Rightarrow \text{Total steps} = \frac{n^2+n}{2} + \frac{2n^3+3n^2+n}{3} - n + n^2 - n$$

$$= \frac{4n^3 + 15n^2 + 7n}{6} \sim O(n^3)$$

5)

We can always express the complex matrix as $A + iB$, where A, B are real matrices

Let $C + iD$ be the inverse, where C, D are real matrices

$$\Rightarrow (C+iD)(A+iB) = CA - DB + i(CB + DA) = I$$

$$\Rightarrow CB + DA = 0 \Rightarrow D = -CBA^{-1}$$ } equating imaginary parts

Now equating the real part,

$$CA + CBA^{-1}B = I \Rightarrow \boxed{\begin{aligned} C^{-1} &= A + BA^{-1}B \\ D &= -CBA^{-1} \end{aligned}}$$

We can find C^{-1} and then invert it. Thefore we have reduced the problem of computing the inverse of a complex matrix to that of inverting real matrices

6)

$$\begin{bmatrix} 3 & -1 & 1 \\ 3 & 6 & 2 \\ 3 & 3 & 7 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 4 \end{bmatrix}$$

$$x_i = -\sum_{\substack{j=1 \\ j \neq i}}^{n} \frac{a_{ij} x_j}{a_{ii}} + \frac{b_i}{a_{ii}}$$

If we start with x as the zero vector,

$$x = \begin{bmatrix} 1/3 \\ 0 \\ 4/7 \end{bmatrix} \quad \text{after 1 iteration}$$

2^{nd} iteration -

$$x_1 = -\left[\frac{(-1)(0)}{3} + \frac{(1)(4/7)}{3}\right] + \frac{1}{3} = \frac{1}{7}$$

$$x_2 = -\left[\frac{(3)(1/3)}{6} + \frac{(2)(4/7)}{6}\right] = -\frac{15}{6(7)} = -\frac{5}{14}$$

$$x_3 = -\left[\frac{(3)(1/3)}{7} + 0\right] + \frac{4}{7} = \frac{3}{7}$$

7) $$\begin{bmatrix} 3 & -1 & 1 \\ 3 & 6 & 2 \\ 3 & 3 & 7 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 4 \end{bmatrix}$$

$$x_i^{(k)} = \frac{1}{a_{ii}}\left[-\sum_{j=1}^{i-1} a_{ij}\, x_j^{(k)} - \sum_{j=i+1}^{n} a_{ij}\, x_j^{(k-1)} + b_i \right]$$

Again starting from zero vector,

$x_1^{(1)} = 1/3$ $\qquad$ $x_2^{(1)} = \frac{1}{6}\left[-3\left(\frac{1}{3}\right) + 0\right] = -\frac{1}{6}$

$$x_3^{(1)} = \frac{1}{7}\left[-3\left(\frac{1}{3}\right) - 3\left(-\frac{1}{6}\right) + 4\right] = \frac{1}{2}$$

$$\frac{1}{2} - \frac{1}{6} = \frac{3-1}{6} = \frac{1}{3}$$

$$-\frac{1}{3} - 1 = -\frac{\cancel{4}2}{3} \cdot \frac{1}{\cancel{6}3}$$

$$-\frac{1}{3} + \frac{2}{3} \qquad \frac{1}{3} + 4$$

$$\frac{13}{3}$$

$$x_1^{(2)} = \frac{1}{3}\left[-(-1)\left(-\frac{1}{6}\right) - (1)\left(\frac{1}{2}\right) + 1\right] = \frac{1}{9}$$

$$x_2^{(2)} = \frac{1}{6}\left[-3\left(\frac{1}{9}\right) - 2\left(\frac{1}{2}\right)\right] = -\frac{2}{9}$$

$$x_3^{(2)} = \frac{1}{7}\left[-3\left(\frac{1}{9}\right) - 3\left(-\frac{2}{9}\right) + 4\right] = \frac{13}{21}$$

8)

$$\begin{bmatrix} 4 & -1 & 0 & -1 & 0 & 0 \\ -1 & 4 & -1 & 0 & -1 & 0 \\ 0 & -1 & 4 & 0 & 0 & -1 \\ -1 & 0 & 0 & 4 & -1 & 0 \\ 0 & -1 & 0 & -1 & 4 & -1 \\ 0 & 0 & -1 & 0 & -1 & 4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} = \begin{bmatrix} 0 \\ 5 \\ 0 \\ 6 \\ -2 \\ 6 \end{bmatrix}$$

Jacobi- $x_1^{(1)} = 0$, $x_2^{(1)} = \frac{5}{4}$, $x_3^{(1)} = 0$, $x_4^{(1)} = \frac{3}{2}$, $x_5^{(1)} = \frac{-1}{2}$, $x_6^{(1)} = \frac{3}{2}$

$$x_1^{(2)} = \frac{1}{4}\left[(+1)\left(\frac{5}{4}\right) + (1)\left(\frac{3}{2}\right)\right] = \frac{11}{16}$$

$$x_2^{(2)} = \frac{1}{4}\left[(1)\left(-\frac{1}{2}\right) + 5\right] = \frac{9}{8}$$

$$x_3^{(2)} = \frac{1}{4}\left[(1)\left(\frac{5}{4}\right) + (1)\left(\frac{3}{2}\right)\right] = \frac{11}{16}$$

$$x_4^{(2)} = \frac{1}{4}\left[(1)\left(-\frac{1}{2}\right) + 6\right] = \frac{11}{8}$$

$$x_5^{(2)} = \frac{1}{4}\left[(1)\left(\frac{5}{4}\right) + (1)\left(\frac{3}{2}\right) + (1)\left(\frac{3}{2}\right) - 2\right] = \frac{9}{16}$$

$$x_6^{(2)} = \frac{1}{4}\left[-\frac{1}{2} + 6\right] = \frac{11}{8}$$

Gauss- Seidel

$$x_1^{(1)} = 0 \,, \quad x_2^{(1)} = \frac{5}{4} \,, \quad x_3^{(1)} = \frac{5}{16} \,, \quad x_4^{(1)} = \frac{3}{2}$$

$$x_5^{(1)} = \frac{1}{4}\left[(1)\left(\frac{5}{4}\right) + (1)\frac{3}{2} - 2\right] = \frac{3}{16} \,, \quad x_6^{(1)} = \frac{1}{4}\left[\frac{5}{16} + \frac{3}{16} + 6\right] = \frac{13}{8}$$

$$x_1^{(2)} = \frac{1}{4}\left[\frac{5}{4} + \frac{3}{2}\right] = \frac{11}{16} \,, \quad x_2^{(2)} = \frac{1}{4}\left[\frac{11}{16} + \frac{5}{16} + \frac{3}{16} + 5\right] = \frac{99}{64}$$

$$x_3^{(2)} = \frac{1}{4}\left[\frac{99}{64} + \frac{13}{8}\right] = \frac{203}{256} \,, \quad x_4^{(2)} = \frac{1}{4}\left[\frac{11}{16} + \frac{3}{16} + 6\right] = \frac{55}{32}$$

$$x_5^{(2)} = \frac{1}{4}\left[\frac{99}{64} + \frac{55}{32} + \frac{13}{8} - 2\right] = \frac{185}{256} \,, \quad x_6^{(2)} = \frac{1}{4}\left[\frac{203 + 185}{256} + 6\right] = \frac{1995}{1024}$$

10)

Eigenvalues using np.linalg.qr (12 iterations) : 8.99999993
4.00000007

Using eigh - 9., 4.

12) The time taken by the codes to run varies with each execution.

Manual - 0.002996 s

Numpy - 0.001999 s

9) No. of iterations needed-

Jacobi- 36

Gauss-Seidel- 17

Relaxation- 6

Conjugate Gradient- 5

13)

i)	sgesv	gsl_linalg_LU_solve	numpy.linalg.solve
ii)			
iii)			
iv)			
v)			
vi)	sgetrf	gsl_linalg_LU_decomp	scipy.linalg.lu
vii)	sgeqrf	gsl_linalg_QR_decomp	numpy.linalg.qr
viii)	sgesvd	gsl_linalg_SV_decomp	numpy.linalg.svd
ix)	sy/heev	gsl_eigen_symm	numpy.linalg.eigh
x)	sy/heev	gsl_eigen_herm	numpy.linalg.eigh
xi)	sgeev	gsl_eigen_nonsym	numpy.linalg.eig