2η Σειρά Ασκήσεων

1. Γράψτε συνάρτηση που να υπολογίζει το γινόμενο πίνακα–διανύσματος $\boldsymbol{A} * \boldsymbol{x}$.

   i. Θεωρείστε ότι ο πίνακας **A** είναι αποθηκευμένος σε πλήρη μορφή ως ενα 2D array της C#, δηλαδή double[,].

   ii. Γραψτε μια δεύτερη συνάρτηση που να ελέγχει την πρώτη

2. Έστω οι πίνακες $\boldsymbol{A} = \begin{bmatrix} 1.4 & 6.4 & 9.1 & 0 \\ 3.2 & 2.1 & 3.6 & 0 \\ -1.8 & 0.3 & 0.1 & 0 \\ 5.4 & -0.2 & -0.5 & 1.0 \end{bmatrix}$ και $\boldsymbol{B} = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$.

   Η εξίσωση $\boldsymbol{A} * \boldsymbol{X} = \boldsymbol{B}$ μπορεί να θεωρηθεί ως ένα σύνολο γραμμικών συστημάτων, όπου πίνακας είναι ο $\boldsymbol{A}$, δεξιά μέλη οι στήλες του $\boldsymbol{B}$ και άγνωστοι οι στήλες του $\boldsymbol{X}$ (η στηλη j του $\boldsymbol{X}$ αντιστοιχεί στη στήλη j του $\boldsymbol{B}$):

$$\begin{bmatrix} 1.4 & 6.4 & 9.1 & 0 \\ 3.2 & 2.1 & 3.6 & 0 \\ -1.8 & 0.3 & 0.1 & 0 \\ 5.4 & -0.2 & -0.5 & 1.0 \end{bmatrix} * \begin{bmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \\ x_{31} & x_{32} & x_{33} \\ x_{41} & x_{42} & x_{43} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

   i. Γράψτε μια συνάρτηση που να λύνει ταυτόχρονα πολλά γραμμικά συστήματα με τη μέθοδο απαλοιφής Gauss (Gauss elimination). Φροντίστε να εκτελείται partial-pivoting.

   ii. Θεωρείστε ότι οι πίνακες **A, B** είναι αποθηκευμένοι σε πλήρη μορφή ως 2D arrays της C#, δηλαδή double[,]

   iii. Γράψτε μια δεύτερη συνάρτηση που να ελέγχει την πρώτη με τους πίνακες $\boldsymbol{A}, \boldsymbol{B}$ που δόθηκαν. Στο τέλος θα πρέπει ο πολλαπλασιασμός $\boldsymbol{A} * \boldsymbol{X}$ να επιστρέφει $\boldsymbol{B}$.

<u>Σημείωση</u>: μπορείτε να χρησιμοποιήσετε τα απλά 2D arrays του Matlab αντί για κάποια από τις μορφές αποθήκευσης πινάκων που διδαχθηκαν στο μάθημα.

3. Έστω ένας αντιστρέψιμος πίνακας $\boldsymbol{A}$.

   i. Γράψτε μία συνάρτηση που να εκτελεί την παραγοντοποίηση $\boldsymbol{A} = \boldsymbol{L} * \boldsymbol{U}$ (χωρίς pivoting**).** Οι παράγοντες $\boldsymbol{L}, \boldsymbol{U}$ θα αποθηκεύονται σε 2 διαφορετικά arrays.

   ii. Γράψτε μία συνάρτηση που να δέχεται ως όρισμα τους παράγοντες $\boldsymbol{L}, \boldsymbol{U}$ (και το διάνυσμα $\boldsymbol{b}$) και να λύνει το γραμμικό σύστημα $\boldsymbol{A} * \boldsymbol{x} = \boldsymbol{b}$ χρησιμοποιώντας τους.

iii. Γράψτε μια τρίτη συνάρτηση που θα ελέγχει τις προηγούμενες με τον αντιστρέψιμο πίνακα

$$\boldsymbol{A} = \begin{bmatrix} 2 & 2 & 4 & 9 \\ 1 & 2.6667 & 71 & 5 \\ 1 & 1 & 3 & 5 \\ 0 & 0.3333 & 1 & -2.5 \end{bmatrix}$$

και δεξί μέλος της αρεσκείας σας. Ο πίνακας αυτός δεν θα χρειαστεί pivoting. Στο τέλος θα πρέπει ο πολλαπλασιασμός $\boldsymbol{A} * \boldsymbol{x}$ να επιστρέφει $\boldsymbol{b}$.

Σημείωση: μπορείτε να χρησιμοποιήσετε τα απλά 2D arrays της C# αντί για κάποια από τις μορφές αποθήκευσης πινάκων που διδαχθηκαν στο μάθημα.

4. Επαναλάβετε την **άσκηση 3**, αλλά τώρα αποθηκεύστε τους παράγοντες $\boldsymbol{L}, \boldsymbol{U}$ γράφοντας πάνω στο array που έχει αποθηκευτεί ο πίνακας $\boldsymbol{A}$. Κατα τη διαρκεια της LU παραγοντοποίησης τα στοιχεία του $\boldsymbol{A}$ μπορούν σταδιακά να αντικαθίστανται με στοιχεία των $\boldsymbol{L}, \boldsymbol{U}$.

5. Έστω ένας θετικά ορισμένος (positive definite) πίνακας $\boldsymbol{A}$.

i. Γράψτε μία συνάρτηση που να εκτελεί την παραγοντοποίηση Cholesky ($\boldsymbol{A} = \boldsymbol{L} * \boldsymbol{L}^{\boldsymbol{T}}$).

ii. Γράψτε μία συνάρτηση που να δέχεται ως όρισμα τον παράγοντα $\boldsymbol{L}$ (και το διάνυσμα $\boldsymbol{b}$) και να λύνει το γραμμικό σύστημα $\boldsymbol{A} * \boldsymbol{x} = \boldsymbol{b}$ χρησιμοποιώντας τον.

iii. Γράψτε μια τρίτη συνάρτηση που θα ελέγχει τις προηγούμενες σας με τον θετικά ορισμένο πίνακα

$$\boldsymbol{A} = \begin{bmatrix} 8.9156 & 0.4590 & 0.0588 & 0.5776 \\ & 7.5366 & 0.4276 & 0.3282 \\ & & 6.4145 & 0.4144 \\ sym & & & 7.7576 \end{bmatrix}$$

και δεξί μέλος της αρεσκείας σας. Στο τέλος θα πρέπει ο πολλαπλασιασμός $\boldsymbol{A} * \boldsymbol{x}$ να επιστρέφει $\boldsymbol{b}$.

Σημείωση: μπορείτε να χρησιμοποιήσετε τα απλά 2D arrays της C# αντί για κάποια από τις μορφές αποθήκευσης πινάκων που διδαχθηκαν στο μάθημα.

6. Επαναλάβετε την **άσκηση 5**, αλλά τώρα αποθηκεύστε τον παράγοντες $\boldsymbol{L}$ γράφοντας πάνω στο array που έχει αποθηκευτεί ο πίνακας $\boldsymbol{A}$. Κατα τη διαρκεια της παραγοντοποίησης Cholesky τα στοιχεία του $\boldsymbol{A}$ μπορούν σταδιακά να αντικαθίστανται με του $\boldsymbol{L}$.

7. Έστω ο πίνακας $\boldsymbol{A} = \begin{bmatrix} 0 & 0 & 2.09065 & 0 \\ 0.76202 & 3.72194 & 2.74205 & 0 \\ 1.6819 & 0 & 0 & 0.02756 \end{bmatrix}$.

i. Γράψτε μία συνάρτηση που να εκτελεί τον πολλαπλασιασμό $\boldsymbol{A} * \boldsymbol{x}$ με τον πίνακα σε μορφή αποθήκευσης CSR.

ii. Γράψτε μία συνάρτηση που να εκτελεί τον πολλαπλασιασμό $\boldsymbol{A}^{\boldsymbol{T}} * \boldsymbol{x}$ με τον πίνακα σε μορφή αποθήκευσης CSR.

iii. Σε μία τρίτη συνάρτηση, γράψτε τον παραπάνω πίνακα σε μορφή αποθήκευσης CSR και ελέγξτε τις συναρτήσεις σας, με διάνυσμα $\boldsymbol{x}$ της αρεσκείας σας.

8. Επαναλάβετε την **άσκηση 7**, αλλά τώρα χρησιμοποιήστε μορφή αποθήκευσης COO. Επιλέξτε εσείς αν θα είναι row major ή column major, αλλά δηλώστε την επιλογή σας στον κώδικα.

9. Έστω ο πίνακας $\boldsymbol{A} = \begin{bmatrix} 21 & 1 & 0 & 4 & 0 \\ & 22 & 2 & 0 & 0 \\ & & 23 & 1 & 3 \\ & & & 24 & 2 \\ sym & & & & 25 \end{bmatrix}$.

i. Γράψτε μία συνάρτηση που να δέχεται ως ορίσματα α) τον πίνακα $\boldsymbol{A}$ σε banded μορφή αποθήκευσης, β) τη γραμμή και στήλη ενός στοιχείου (i, j) και να επιστρέφει την τιμή του στοιχείου A_{ij}.

ii. Σε μια δεύτερη συνάρτηση, γράψτε τον παραπάνω πίνακα σε μορφή αποθήκευσης banded (ταινιωτή). Δοκιμάστε την πρώτη συναρτήση σας για 2-3 στοιχεία του πίνακα. Ελέγξτε τα αποτελέσματα χρησιμοποιώντας αποθήκευση σε απλό 2D array.

10. Έστω ο θετικά ορισμένος πίνακας $\boldsymbol{A} = \begin{bmatrix} 21 & 1 & 0 & 4 & 0 \\ & 22 & 2 & 0 & 0 \\ & & 23 & 1 & 3 \\ & & & 24 & 2 \\ sym & & & & 25 \end{bmatrix}$.

i. Γράψτε μία συνάρτηση που να εκτελεί την παραγοντοποίηση Cholesky ($\boldsymbol{A} = \boldsymbol{L} * \boldsymbol{L}^{\boldsymbol{T}}$) με τον πίνακα σε μορφή αποθήκευσης skyline. Ο παράγοντας $\boldsymbol{L}$ θα πρέπει να αποθηκευτεί σε μορφή skyline. Θυμηθείτε ότι το sparsity pattern του $\boldsymbol{L}$ είναι ίδιο με του $\boldsymbol{A}$. Επίσης ο $\boldsymbol{L}$ θα πρέπει να γραφτεί στο ίδιο array (αλλά σε ποιό?) με τον $\boldsymbol{A}$, αντικαθιστώντας τον.

ii. Γράψτε μία συνάρτηση που να δέχεται ως όρισμα τον παράγοντα $\boldsymbol{L}$ σε μορφή αποθήκευσης skyline (και το διάνυσμα $\boldsymbol{b}$) και να λύνει το γραμμικό σύστημα $\boldsymbol{A} * \boldsymbol{x} = \boldsymbol{b}$ χρησιμοποιώντας τον.

iii. Σε μια τρίτη συνάρτηση, γράψτε τον παραπάνω πίνακα σε μορφή αποθήκευσης skyline (γραμμής ορίζοντα) και ελέγξτε τις προηγούμενες δυο συναρτήσεις σας με δεξί μέλος της αρεσκείας σας. Στο τέλος θα πρέπει ο πολλαπλασιασμός $\boldsymbol{A} * \boldsymbol{x}$ να επιστρέφει $\boldsymbol{b}$ (αν ο $\boldsymbol{A}$ είναι αποθηκευμένος σαν 2D array).

11. Γράψτε συνάρτηση που να λύνει το γραμμικό σύστημα $\boldsymbol{A} * \boldsymbol{x} = \boldsymbol{b}$ χρησιμοποιώντας τη μέθοδο **Jacobi**:
    i. Η συνάρτηση θα δέχεται ως όρισμα τον τετραγωνικό πίνακα $\boldsymbol{A}$ και το διάνυσμα $\boldsymbol{b}$.
    ii. Ο πίνακας θα είναι αποθηκευμένος σε πλήρη μορφή, οπότε μπορούν να χρησιμοποιήθούν τα απλά 2D arrays της C#.
    iii. Η συνάρτηση θα πρέπει επίσης να δέχεται έναν πραγματικό αριθμό που δείχνει την επιθυμητή ακρίβεια προσέγγισης της λύσης και έναν ακέραιο αριθμό που δείχνει τον μέγιστο αριθμό επαναλήψεων. Εντός της συνάρτησης αυτά θα χρησιμοποιούνται για να ελεγχθεί η σύγκλιση της μεθόδου (έχει προταθεί στις διαφάνειες ένας τρόπος ελέγχου της σύγκλισης).
    iv. Η συνάρτηση θα επιστρέφει την λύση του συστήματος $\boldsymbol{x}$, τον αριθμό τοων επαναλήψεων που τελικά έγιναν και την νόρμα του τελικού υπολειμματικού υπολοίπου (residual) $\|\boldsymbol{r}\| = \|\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}\|$
    v. Σε μια δεύτερη συναρτηση, ελέγξτε την πρώτη με πίνακα και δεξί μέλος της αρεσκείας σας. Τι παρατηρήτε ως προς τη σύγκλιση της μεθόδου?

12. Επαναλάβετε την άσκηση 11, αλλά τώρα αντί για Jacobi χρησιμοποιείστε τις μεθόδους
    i. Gauss-Seidel,
    ii. SOR (διαλέξτε εσείς συντελεστή χαλάρωσης)

13. Επαναλάβετε την άσκηση 11, αλλά τώρα διαλέξτε μια μορφή αποθήκευσης αραιών μητρώων από αυτές που διδάχθηκαν (π.χ. COO, CSR, CSC, Banded, Skyline) αντί για τα 2D arrays της C# και αντί για Jacobi χρησιμοποιείστε τις μεθόδους:
    i. Gradient Descent,
    ii. Conjugate Gradient

Σημείωση: Διαλέξτε καποια μορφή αποθήκευσης που να βολεύει για τις πράξεις των αλγορίθμων επίλυσης. Αν έχετε έτοιμες συναρτήσεις από προηγούμενες ασκήσει που να κάνουν αυτές τις πράξεις, καλέστε εκείνες.

14. Επαναλάβετε την άσκηση 13, αλλά τώρα ο πίνακας θα είναι αποθηκευμένος σε μορφή CSR.

<u>Σημείωση για ασκήσεις που ζητείται παραγοντοποίηση και αντικατάσταση του πίνακα με τους παράγοντες:</u>

Σε μια συνάρτηση που εκτελεί παραγοντοποίηση, ο παράγοντας μπορεί να αποθηκεύεται στο ίδιο array με τον αρχικό πίνακα $\boldsymbol{A}$, αντικαθιστώντας τον.