Αριθμητική Ανάλυση Εργασία 2

Λογοθέτης Φακίνος

Μάιος 2024

1 Άσκηση 1

1.1 Μέρος Α

Εφόσον θέλουμε αλγόριθμο LU παραγοντοποίσης σε πρώτη φάση, μπορούμε να χρησιμοποίησουμε μια παραλλαγή της απαλλοιφής Gauss, που θα κάνει οικονομία σε πράξεις. Θα εκμεταλλευτούμε το ότι χρείαζεται μόνο μια γραμμοπράξη ανα γραμμή του πίνακα, για να κάνουμε τον πίνακα μας άνω τριγωνικό με τη μέθοδο Gauss. Σε κάθε βήμα i θα εξαλείφουμε το c_{i+1} με κατάλληλο συντελεστή στην αφαίρεση γραμμών και αφού ο πίνακας μας A είναι τριδιαγώνιος $\forall j > i+1$, $A_{ji} = 0$.

Αν ο πίνακας A', είναι στο $i+1$ βήμα της μεθόδου μας:

$$A' = \begin{pmatrix} a_1 & b_1 & 0 & \cdots & & & & & & 0 \\ 0 & a_2^{(1)} & b_2 & 0 & \cdots & & & & & 0 \\ 0 & \cdots & & & & & & & & 0 \\ 0 & \cdots & & & a_i^{(i-1)} & b_i & 0 & \cdots & & 0 \\ 0 & \cdots & & & c_{i+1} & a_{i+1} & b_{i+1} & 0 & \cdots & 0 \\ 0 & \cdots & & & & & & & & 0 \\ 0 & \cdots & & & & & & c_{n-1} & a_{n-1} & b_{n-1} \\ 0 & \cdots & & & & & & & c_n & a_n \end{pmatrix}$$

Παρατηρούμε ότι αρκεί να υπολογίσουμε τον συντελεστή $m_{(i+1)i} = \dfrac{c_{i+1}}{a_i^{(i-1)}}$, όλοι οι υπόλοιποι συντελεστές m_{ji}, $\forall j > i+1$ του L είναι 0, καθώς ήδη

$A_{ji} = 0$. Μετά, μας απομένει να μηδενίσουμε το c_{i+1} και να θέσουμε $a_{i+1}^{(i)} = a_{i+1} - m_{(i+1)i}a_i^{(i-1)}$, προχορώντας στην επόμενη γραμμή.

Έτσι ο L θα είναι ο εξής:

$$L = \begin{pmatrix} 1 & 0 & \cdots & & 0 \\ m_{21} & 1 & \cdots & & 0 \\ 0 & m_{32} & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & \cdots & 0 & \cdots & 1 \end{pmatrix}$$

Παρατηρούμε, τότε ότι τα στοιχεία b_i μένουν ίδια (καθώς απο πάνω τους έχουν 0 και δεν επηρεάζονται απο την αφαίρεση γραμμών), ενώ $a_{i+1}^{(i)} = a_{i+1} - m_{(i+1)i}b_i$. Άρα:

$$U = \begin{pmatrix} a_1 & b_1 & \cdots & & 0 \\ 0 & a_2^{(1)} & b_2 & \cdots & 0 \\ 0 & 0 & a_3^{(2)} & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & \cdots & 0 & \cdots & a_n^{(n-1)} \end{pmatrix}$$

Τώρα προχωράμε στο diagLsol, δηλαδή την επίλυση του $Ly = b$

$$\begin{pmatrix} 1 & 0 & \cdots & & 0 \\ m_{21} & 1 & \cdots & & 0 \\ 0 & m_{32} & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & \cdots & 0 & \cdots & 1 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix} \implies$$

$$\begin{pmatrix} y_1 \\ m_{21}y_1 + y_2 \\ \vdots \\ m_{n(n-1)}y_{n-1} + y_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix}$$

Τώρα προχωράμε στο diagUsol, δηλαδή την επίλυση του $Ux = y$.

Εύκολα λοιπόν λύνεται, το παραπάνω σύστημα υπολογίζοντας το $y_i = b_i - m_{i(i-1)}y_{i-1}$, χρησιμοποιώντας το προηγούμενο με $y_1 = b_1$.

$$\begin{pmatrix} a_1 & b_1 & \cdots & & & 0 \\ 0 & a_2^{(1)} & b_2 & \cdots & & 0 \\ 0 & 0 & a_3^{(2)} & \cdots & & 0 \\ \vdots & \vdots & \vdots & & & \vdots \\ 0 & & & & a_{n-1}^{(n-2)} & b_{n-1} \\ 0 & \cdots & 0 & \cdots & 0 & a_n^{(n-1)} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} \implies$$

$$\begin{pmatrix} a_1x_1 + b_1x_2 \\ a_2^{(1)}x_2 + b_2x_3 \\ \vdots \\ a_n^{(n-1)}x_n \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}$$

Εύκολα λοιπόν λύνεται, το παραπάνω σύστημα υπολογίζοντας το $x_i = \dfrac{y_i - b_ix_{i+1}}{a_i^{(i-1)}}$, χρησιμοποιώντας το επόμενο με $x_n = \dfrac{y_n}{a_n^{(n-1)}}$.

Ύστερα απο τις παραπάνω τροποποιήσεις, επέρχεται τεράστια βελτίωση σε θέμα χρόνου (ασυμπτωτικά) σε όλες τις φάσεις του αλγορίθμου επίλυσης του συστήματος.

Ο καινούργιος μας αλγόριθμος LUdec, είναι της τάξης $O(n)$, δηλαδή γραμμικός, όσο αφορά τον χρόνο (αφού έχει ουσιαστικά ένα for, το εσωτερικό for συνεισφέρει κατά μια σταθερά 2, αφού τρέχει απο το i στο $i+1$) και $O(n^2)$ όσο αφορά τον χώρο, καθώς αποθηκεύουμε πίνακες στη μνήμη μεγέθους n^2.

Προηγουμένως, ο αλγόριθμος LUdec ήταν της τάξης $O(\frac{n^3}{3})$ όσο αφορά τη χρονική πολυπλοκότητα (όσο και η μέθοδο απαλλοιφής του Gauss, στην οποία βασίζεται) και $O(n^2)$ όσο αφορά τον χώρο, για τον ίδιο λόγο.

Ο καινούργιος μας αλγόριθμος diagLsol, είναι της τάξης $O(n)$, γραμμικός όσο αφορά τον χρόνο (ένα for) και $O(n^2)$ όσο αφορά τον χώρο. (Απλά εκτελούμε n υπολογισμούς).

Ο προηγούμενος αλγόριθμος Lsol είναι της τάξης $O(\frac{n^2-n}{2}) = O(n^2)$, στον χρόνο (περίπου $n + n - 1 + \cdots + 1$ επαναλήψεις) και $O(n^2)$ όσο αφορά τον χώρο.

Ο καινούργιος μας αλγόριθμος diagUsol, είναι της τάξης $O(n)$, γραμμικός όσο αφορά τον χρόνο (ένα for) και $O(n^2)$ όσο αφορά τον χώρο.

Ο προηγούμενος αλγόριθμος Usol είναι της τάξης $O(\frac{n^2-n}{2}) = O(n^2)$, στον χρόνο (περίπου $n + n - 1 + \cdots + 1$ επαναλήψεις) και $O(n^2)$ όσο αφορά τον χώρο.

Συνδυάζοντας τα παραπάνω για ένα σύστημα με τριδιαγώνιο πίνακα ($Ax = b$), αν χρησιμοποιήσουμε την γενική μέθοδο LUdec $\rightarrow$ Lsol $\rightarrow$ Usol, έχουμε χρονική πολυπλοκότητα $O(\frac{n^3}{3}) + O(n^2) + O(n^2) = O(\frac{n^3}{3})$, ενώ με την ειδική μέθοδο μας diagLUdec $\rightarrow$ diagLsol $\rightarrow$ diagUsol έχουμε χρονική πολυπλοκότητα $O(n) + O(n) + O(n) = O(n)$, ενώ όσο αφορά τον χώρο και οι δύο μέθοδοι είναι της τάξης $O(n^2)$.

Πράξεις:

diagLUdec: $2(n-1) + 2(n-1) = 4n$ flops

diagUsol: $1 + n - 1 = n$ flops

diagLsol: $1 + n - 1 = n$ flops

diagLUdec $\rightarrow$ diagUsol $\rightarrow$ diagLsol: $6n$ flops

Ludec: Για κάθε $1 \leq i \leq n-1$, έχουμε $(n-i) + (n-i)(n-i+1) = (n-i)(n-i+2)$ flops, άρα συνολικά $\displaystyle\sum_{i=1}^{n-1}(n-i)(n-i+2) = \sum_{i=1}^{n}(n-i)^2 + 2(n-i) = \sum_{i=1}^{n} i^2 + 2i = \frac{n(n+1)(2n+1)}{6} + \frac{2n(n+1)}{2} = n(n+1)(\frac{2n+1}{6} + 1) = (n^2+n)(\frac{2n+7}{6}) = \frac{2n^3 + 9n^2 + 7n}{6}$ flops.

Lsol: Για κάθε $1 \leq i \leq n$, έχουμε $2 + i - 1 = i + 1$ flops, άρα συνολικά $\dfrac{n(n+1)}{2} + 1 = \dfrac{n^2 + n + 2}{2}$ flops.

Usol: Ομοίως, $\dfrac{n^2 + n + 2}{2}$ flops.

LUdec → Usol → Lsol: $\dfrac{2n^3+9n^2+7n}{6}+\dfrac{6n^2+6n+12}{6}=\dfrac{2n^3+15n^2+13n+12}{6}$ flops.

1.2 Μέρος Β

$$A=\begin{pmatrix}5 & 2 & 0 & \cdots & & \\ 2 & 5 & 2 & 0 & \cdots & \\ 0 & 2 & 5 & 2 & 0 & \cdots \\ \vdots & & \ddots & \ddots & \ddots & \\ 0 & \cdots & & & 2 & 5\end{pmatrix}$$

Υπολογίζουμε την ορίζουσα του $A=LU$, μέσω του άνω τριγωνικού U, συγκεκριμένα όπως γνωρίζουμε ο U προκύπτει απο τον A, με αφαιρέσεις γραμμών, οι οποίες δεν επηρεάζουν την τιμή της ορίζουσας $detA$, επίσης ο U είναι άνω τριγωνικός επομένως $detA=detU=\prod_{i=1}^{11}u_{ii}=5592405$.

Λύνουμε τα συστήματα $Ax_i=e_i$, με $e_j=\delta_{ij}$, $1\leq i,j\leq 11$, με τη μέθοδο που αναπτύξαμε στο μέρος A (diagLUdec → diagLsol → diagUsol) και φτιάχνουμε τον πίνακα $B=(x_1,x_2,\cdots,x_{11})$, τότε $A(x_1,x_2,\cdots,x_{11})=(Ax_1,Ax_2,\cdots,Ax_{11})=(e_1,e_2,\cdots,e_{11})=I_{11}\implies B=A^{-1}$.

2 Άσκηση 2

2.1 Μέρος Β

Επιλύοντας το σύστημα:

$$Ax=b\leftrightarrow\begin{pmatrix}2 & 1 & 3\\ 4 & 2 & 1\\ 1 & 1 & 1\end{pmatrix}\begin{pmatrix}x_1\\ x_2\\ x_3\end{pmatrix}=\begin{pmatrix}13\\ 11\\ 6\end{pmatrix}$$

Με τη μέθοδο απαλλοιφής του Gauss (LUdec → Lsol → Usol), οδηγούμαστε στη λύση του συστήματος:

$$x=\begin{pmatrix}NaN\\ NaN\\ NaN\end{pmatrix}$$

Προφανώς, δεν είναι λύση, καθώς κάπου προέκυψε σφάλμα. Πράγματι, ύστερα απο διερεύνηση, διαπιστώνει κανείς, ότι έγινε διαίρεση με το 0 κατά τον υπολογισμό (συγκεκριμένα το 2ο διαγώνιο στοιχείο του A μηδενίστηκε στην πρώτη αφαίρεση γραμμών και ύστερα χρησιμοποιήθηκε στην διαίρεση για τον συντελεστή m_{32} του L) και επομένος η μέθοδος απαλλοιφής του Gauss χωρίς οδήγηση, απέτυχε.

Με τη μέθοδο απαλλοιφής του Gauss, με μερική οδήγηση (pdLUdec → Lsol → Usol), οδηγούμαστε στη λύση του συστήματος $PAx = Pb \implies Ax = b$, $(P^T P = I_3 \implies P^{-1} = P^T)$:

$$x = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}, P = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}$$

Τελικά βρίσκουμε ότι η λύση του συστήματος είναι η:

$$x = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}$$

Σε αυτό το παράδειγμα η μερική οδήγηση στη μέθοδο απαλλοιφής του Gauss, απέτρεψε την επικείμενη διαίρεση με το 0, εναλλάσοντας τις κατάλληλες γραμμές.

3 Άσκηση 3

$$Hil = \begin{pmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \frac{1}{4} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \frac{1}{6} \\ \frac{1}{4} & \frac{1}{5} & \frac{1}{6} & \frac{1}{7} \end{pmatrix}$$

$$Had = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix}$$

$$b_1 = \begin{pmatrix} \frac{25}{12} \\ \frac{77}{60} \\ \frac{19}{20} \\ \frac{319}{420} \end{pmatrix}, b_2 = \begin{pmatrix} 4 \\ 0 \\ 0 \\ 0 \end{pmatrix}, x = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

$$b_1^* = \begin{pmatrix} \frac{25}{12} \\ \frac{77}{60} \\ \frac{19}{20} \\ \frac{320}{420} \end{pmatrix}, b_2^* = \begin{pmatrix} 4 \\ 0 \\ 0 \\ \frac{1}{420} \end{pmatrix}$$

$$Hilx_1 = b_1^* \implies x_1 = \begin{pmatrix} 0.6667 \\ 5.0000 \\ -9.0000 \\ 7.6667 \end{pmatrix}$$

$$Hadx_2 = b_2^* \implies x_2 = \begin{pmatrix} 1.0006 \\ 0.9994 \\ 0.9994 \\ 1.0006 \end{pmatrix}$$

$$\frac{||x - x_1||_\infty}{||x||_\infty} = 10, \frac{||b_1 - b_1^*||_\infty}{||b_1||_\infty} = 1.1429 \cdot 10^{-3}, \kappa(Hil)_\infty = 2.8375 \cdot 10^4$$

$$\frac{||x - x_2||_\infty}{||x||_\infty} = 5.9524 \cdot 10^{-4}, \frac{||b_2 - b_2^*||_\infty}{||b_2||_\infty} = 5.9524 \cdot 10^{-4}, \kappa(Had)_\infty = 4$$

Απο θεωρία γνωρίζουμε ότι $\frac{||\Delta x||_\infty}{||x||_\infty} \leq \kappa(A)\frac{||\Delta b||_\infty}{||b||_\infty}$, $\forall A \in R^{n \times n}$ αντιστρέψιμο. Δηλαδή, το $\kappa(A)$ καθορίζει το άνω φράγμα της μεταβολής της λύσης του συστήματος, δεδομένου μιας μεταβολής του b. Επομένως, αν αυτός ο αριθμός είναι μικρός, τότε μικρές μεταβολές στο b, προκαλούν μικρές μεταβολές στο x και αντίστροφα μεγάλο $\kappa(A)$ σημαίνει ότι μια μικρή μεταβολή στο b, μπορεί να προκαλέσει μεγάλη μεταβολή στη λύση x του συστήματος.

Πράγματι, παρατηρούμε ότι για τον πίνακα Hilbert, $\kappa(Hil)_\infty >> 1$ και ότι μια μικρή μεταβολή στο b, της τάξης του 10^{-3}, προκαλεί συγριτικά μεγάλη μεταβολή στη λύση του συστήματος, της τάξης του 10^{1}, το οποίο βγάζει νοήμα, σύμφωνα με την παραπάνω ανισότητα. (ill-conditioned matrix).

Αντιθέτως, παρατηρούμε ότι για τον πίνακα Hadamard, $\kappa(Had)_\infty \approx 1$ και ότι μια μικρή μεταβολή στο b, της τάξης του 10^{-4}, προκαλεί συγκριτικά μια μικρή μεταβολή στη λύση του συστήματος, της τάξης του 10^{-4}, όπως και περιμέναμε απο την ανισότητα. (well-conditioned matrix).

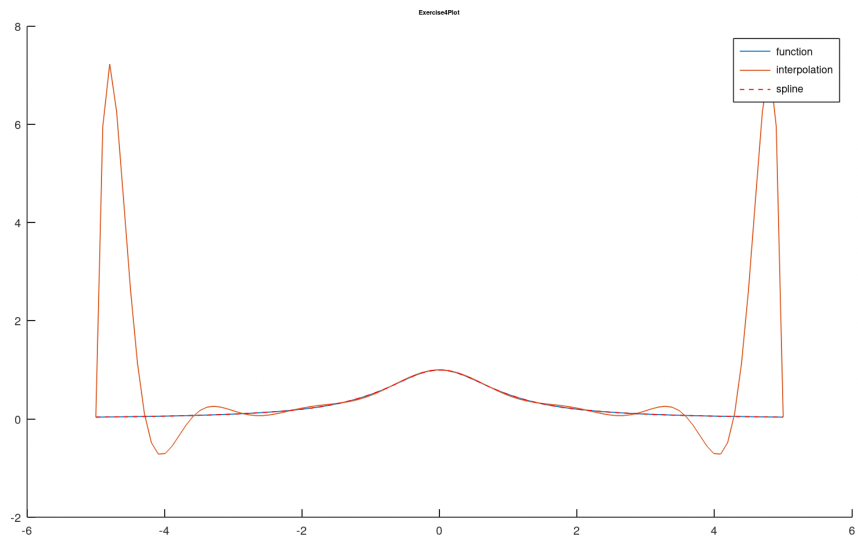
4 Άσκηση 4

Θεωρούμε:

$$f(x) = \frac{1}{1+x^2}, \, x \in [-5, 5]$$

Και τα σημεία $(x_i, f(x_i))$, ομοιόμορφα κατανεμημένα στο $[-5, 5]$ και 15 στο πλήθος.

Παρατηρούμε και απο το γράφημα, ότι η μέθοδος Spline, προσεγγίζει την f



καλύτερα, ενώ η μέθοδος Langrange είναι ασταθής κοντά στα άκρα του διαστήματος $[-5, 5]$, αυτό ήταν και αναμενόμενο, καθώς:

Για τη μέθοδο splines, γνωρίζουμε απο θεωρία ότι $||f-s||_\infty \leq \frac{5}{384}h^4||f^{(4)}||_\infty$, όπου h η λεπτότητα της διαμέρισης, εδώ $h = \frac{10}{15} = \frac{2}{3}$, εφόσον η διαμέριση μας είναι ομοιόμορφη στο $[-5, 5]$. Υπολογίζουμε επίσης με βοήθεια (Wolfram Alpha), $||f^{(4)}||_\infty = 24$, άρα $||f-s||_\infty \leq \frac{5}{384}\frac{16}{81} \cdot 24 \approx 0.06172839506$, το οποίο είναι αρκετά καλό.

Απο την άλλη, για τη μέθοδο Langrange έχουμε απο θεωρία $||f-p_n||_\infty \leq \frac{h^{n+1}}{4(n+1)}||f^{(n+1)}||_\infty \implies ||f-p_n||_\infty \leq \frac{h^{16}}{64}||f^{(16)}||_\infty$. Με βοήθεια υπολογίζουμε (Wolfram Alpha) $||f^{(16)}||_\infty = 20922789888000$, άρα $||f-p_n||_\infty \leq \frac{2^{16}}{64 \cdot 3^{16}} \cdot 20922789888000 \approx 497713562.092$.

Διαισθητικά, αυτό οφείλεται στο ότι έχουμε υπερβολικά πολλά σημεία παρεμβολής και το πολυώνυμο παρεμβολής γίνεται 15ου βαθμού, που το κάνει δύσχρηστο και ασταθές, όσο $|x| \uparrow$, απο την το spline συμπεριφέρεται καλύτερα ώς πολυώνυμο τρίτου βαθμού και τα πολλά σημεία επιδρόυν αποκλειστικά θετικά, σε αυτό, οπώς παρατηρούμε και απο την ανισότητα.