

Επιστημονικός Υπολογισμός 2023-24

ΣΟΥΛΙ ΣΠΥΡΟ

A.M. 1070263

Πάτρα 2023/01/10

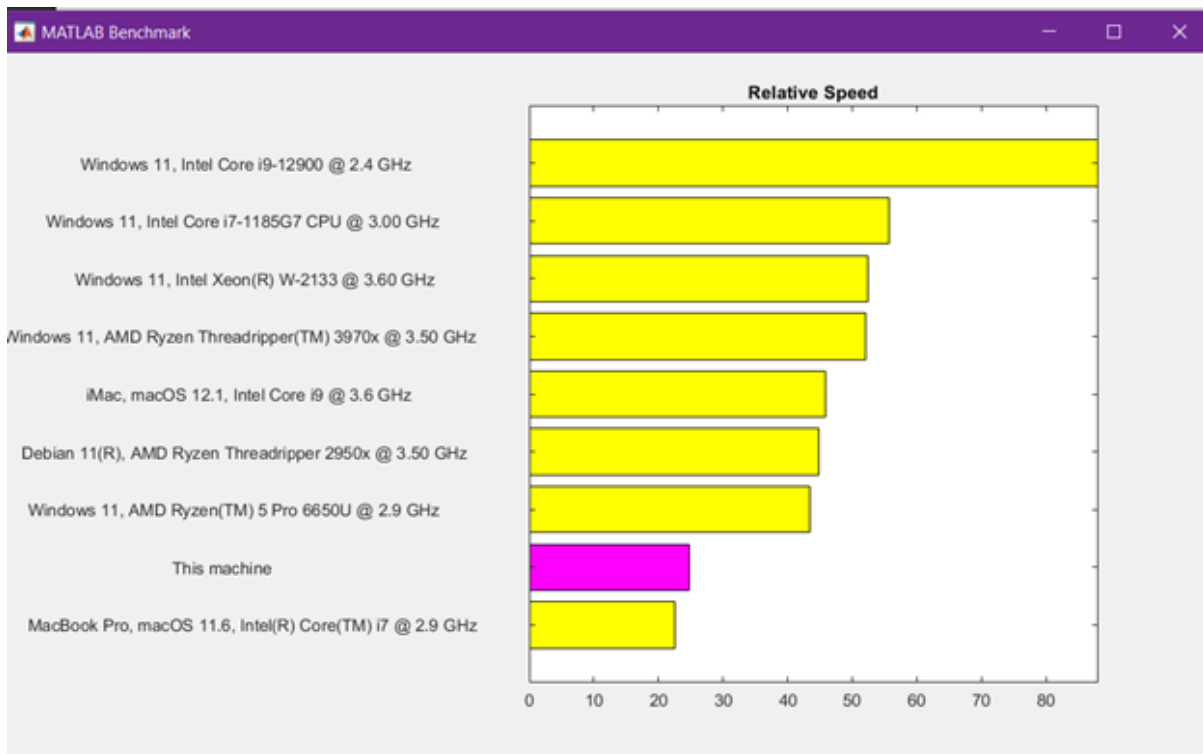
Περιεχόμενα

1	Στοιχεία υπολογιστικού συστήματος	2
2	Χρονομετρήσεις	4
2.1	4
2.2	5
2.3	6
2.4	8
3	Ειδικοί επιλυτές και αραιά μητρώα	10
3.1	10
3.2	11
3.3	12
4	Τανυστές	15
4.1	15
4.2	15
5	Επίλυση ΣΘΟ συστημάτων με PCG	16
5.1	poison_test_1070263.m	16
5.2	pcg_1070263	16
5.3	constructPoissonMatrix_1070263	18
5.4	customPreconditioner	19
5.5	incompleteCholesky	19
5.6	solveExactPoisson	19
5.7	SuiteSparse.m	19

1 Στοιχεία υπολογιστικού συστήματος

Χαρακτηριστικά	Απάντηση
Έναρξη/λήξη εργασίας	1/9/2024 – 1/13/2024
model	ASUS G531GU
O/S	Windows 10 Pro
processor name	4-Core Intel Core i5 9300H
processor speed	3.9 GHz
number of processors	1
total # cores	4
total # threads	8
FMA instruction	yes
L1 cache	128KB Instruction, 128KB Data write-back
L2 cache	128 write-back
L3 cache	(shared)8MB write-back
Gflops/s	301.0
Memory	16GB
Memory Bandwidth	40.5 Gb/s
MATLAB Version	(9.14.0.2206163) R2023a Last Update February 22, 2023
BLAS	Intel(R) oneAPI Math Kernel Library Version 2021.3
LAPACK	Intel(R) oneAPI Math Kernel Library Version 2021.3 supporting Linear Algebra PACKage (LAPACK 3.9.0)

Table 1: Στοιχεία υπολογιστικού συστήματος



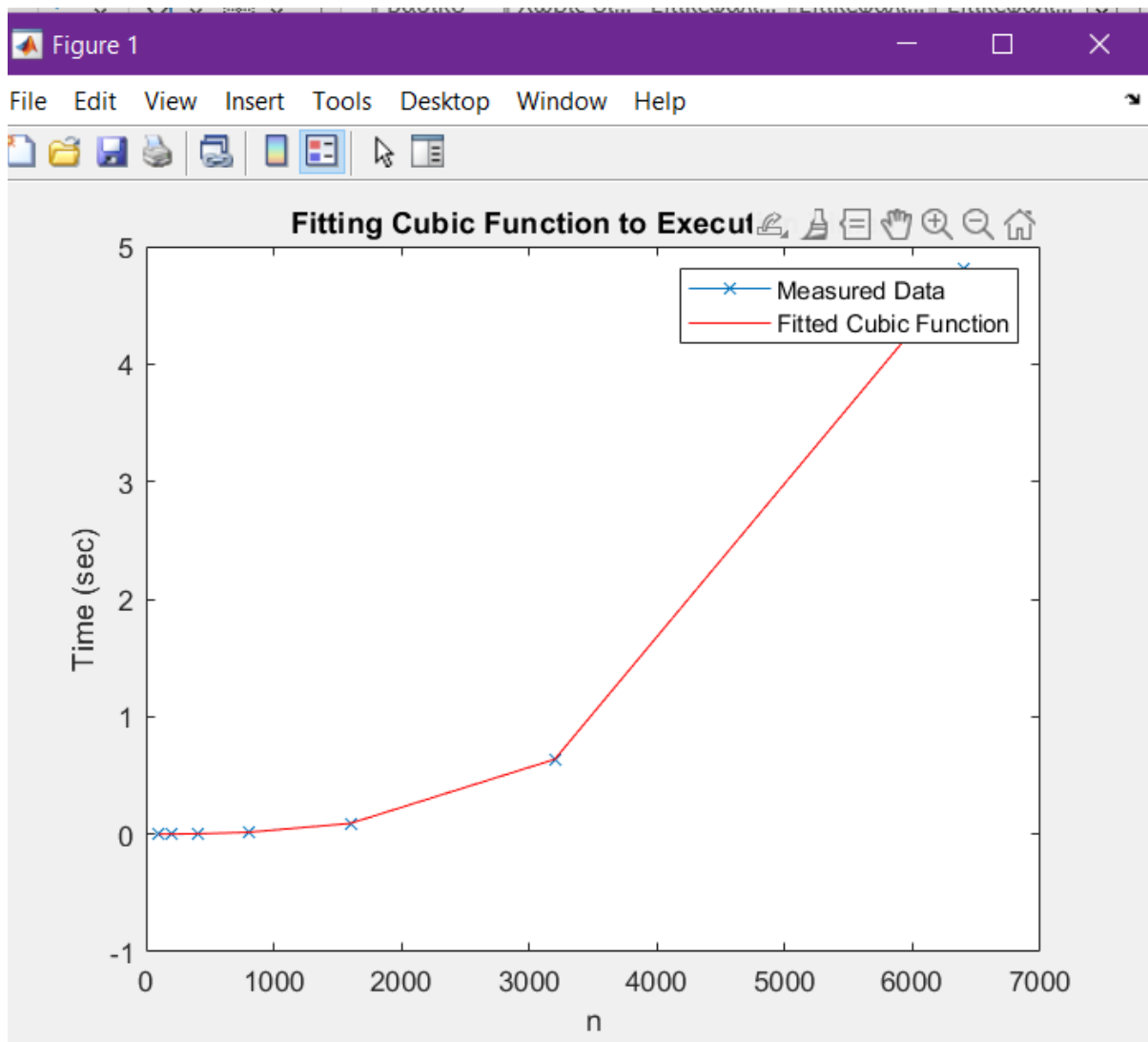
MATLAB Benchmark (times in seconds)

Computer Type	LU	FFT	ODE	Sparse	2-D	3-D
Windows 11, Intel Core i9-12900 @ 2.4 GHz	0.2412	0.1537	0.0938	0.2071	0.1285	0.2198
Windows 11, Intel Core i7-1185G7 CPU @ 3.00 GHz	0.5330	0.3032	0.1488	0.3094	0.2628	0.1627
Windows 11, Intel Xeon(R) W-2133 @ 3.60 GHz	0.3983	0.2755	0.2231	0.4463	0.2517	0.1754
Windows 11, AMD Ryzen Threadripper(TM) 3970x @ 3.50 GHz	0.2049	0.1730	0.1858	1.2342	0.1952	0.1211
iMac, macOS 12.1, Intel Core i9 @ 3.6 GHz	0.3410	0.2640	0.1514	0.2883	0.6711	0.3848
Debian 11(R), AMD Ryzen Threadripper 2950x @ 3.50 GHz	0.3386	0.2451	0.1778	1.2104	0.2420	0.1982
Windows 11, AMD Ryzen(TM) 5 Pro 6650U @ 2.9 GHz	0.6737	0.3595	0.1375	0.4712	0.4325	0.2418
This machine	1.1190	0.5135	0.2858	0.4214	0.6194	0.7787
MacBook Pro, macOS 11.6, Intel(R) Core(TM) i7 @ 2.9 GHz	0.7366	0.5500	0.2211	0.4432	1.7170	0.7573

Place the cursor near a computer name for system and version details. Before using this data to compare different versions of MATLAB, or to download an updated timing data file, see the help for the bench function by typing "help bench" at the MATLAB prompt.

2 Χρονομετρήσεις

2.1



Command Window

Cubic function coefficients:

alpha3: 0.21185

alpha2: 0.52412

alpha1: 0.44688

alpha0: 0.12925

Σύντομη περιγραφή όσων έκανα και σχολιασμός των αποτελεσμάτων: Παραγωγή δεδομένων: Δημιουργήσαμε δεδομένα εκτελώντας τη λειτουργία αποσύνθεσης Cholesky για πίνακες μεγεθών $n = 100 \times 2^{[0:6]}$. Για κάθε μέγεθος πίνακα, μετρήσαμε το χρόνο εκτέλεσης χρησιμοποιώντας τη συνάρτηση `timeit`. Προσαρμογή κυβικής συνάρτησης: Χρησιμοποιήσαμε τη συνάρτηση `polyfit` για να προσαρμόσουμε μια κυβική συνάρτηση στα παραγόμενα δεδομένα. Ενσωματώσαμε τη συμβουλή να χρησιμοποιούμε βάρη (`mu`) για πιο αξιόπιστα αποτελέσματα. Αποτελέσματα και συντελεστές: Εξήχθησαν οι συντελεστές της προσαρμοσμένης κυβικής συνάρτησης ($\alpha_3 n^3 + \alpha_2 n^2 + \alpha_1 n + \alpha_0$). Οι συντελεστές παρέχουν πληροφορίες σχετικά με τους ρυθμούς ανάπτυξης και τις σταθερές που συμμετέχουν στην κυβική συνάρτηση. Απεικόνιση: Δημιουργήσαμε μια γραφική παράσταση για την απεικόνιση των μετρούμενων δεδομένων και της προσαρμοσμένης κυβικής συνάρτησης. Σχολιασμός των αποτελεσμάτων: Η συμπερίληψη βαρών στη συνάρτηση `polyfit` με την παράμετρο `mu` ενισχύει την αξιοπιστία της διαδικασίας προσαρμογής. Λαμβάνει υπόψη την κλίμακα των δεδομένων και την κεντρώνει κατάλληλα, οδηγώντας σε ακριβέστερες εκτιμήσεις των συντελεστών. Η καμπύλη που σχεδιάζεται παρέχει μια οπτική αναπαράσταση του πόσο καλά η κυβική συνάρτηση προσαρμόζεται στα μετρούμενα δεδομένα.

2.2

Predicted execution times for n in [100:100:2000]:

Columns 1 through 8

0.0022	0.0007	-0.0002	-0.0003	0.0005	0.0021	0.0048	0.0086
--------	--------	---------	---------	--------	--------	--------	--------

Columns 9 through 16

0.0136	0.0200	0.0277	0.0369	0.0478	0.0603	0.0746	0.0908
--------	--------	--------	--------	--------	--------	--------	--------

Columns 17 through 20

0.1090	0.1292	0.1516	0.1762
--------	--------	--------	--------

Predicted execution times for n in [150:100:1550]:

Columns 1 through 8

0.0014	0.0002	-0.0003	-0.0000	0.0012	0.0033	0.0066	0.0110
--------	--------	---------	---------	--------	--------	--------	--------

Columns 9 through 15

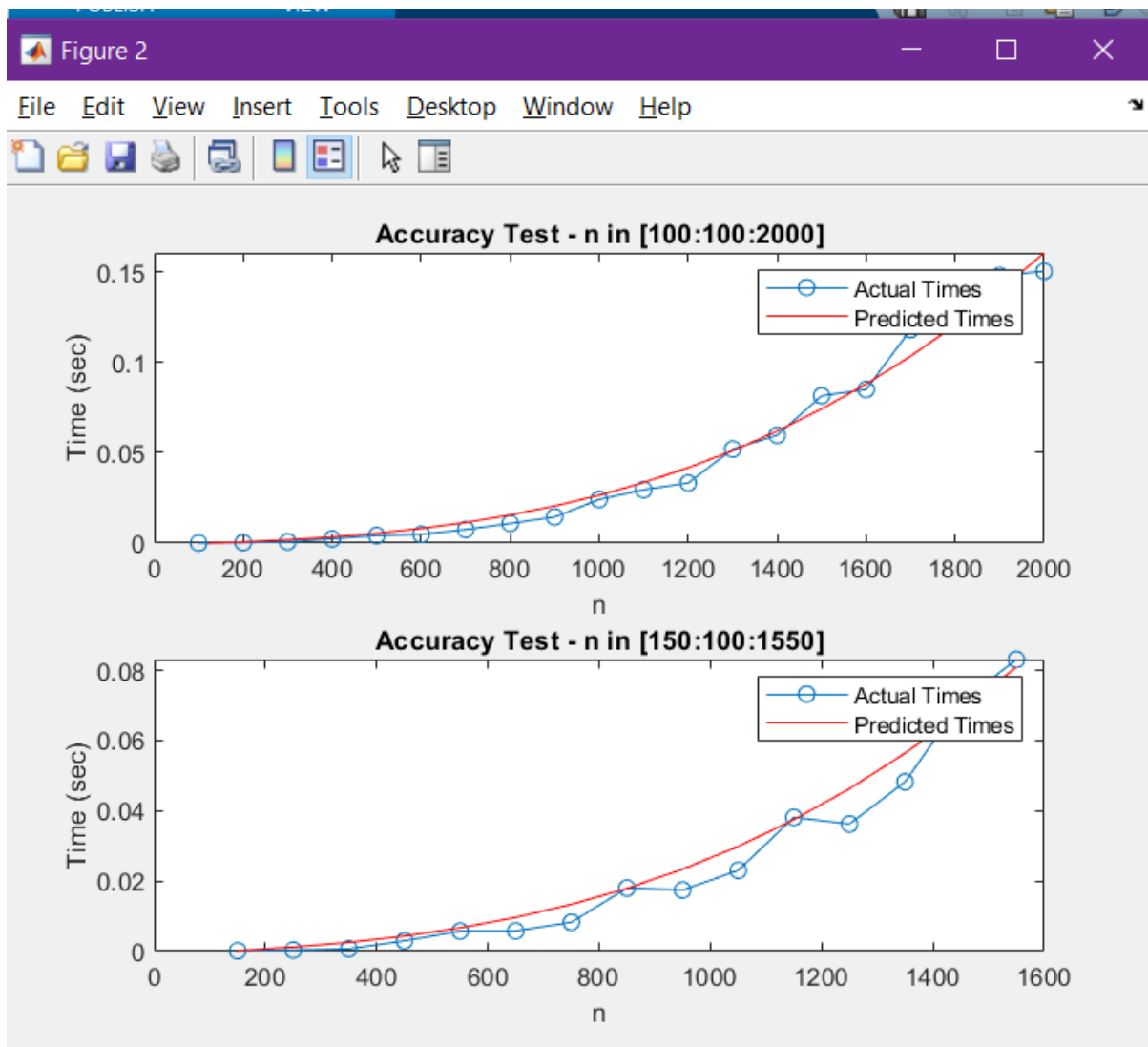
0.0166	0.0236	0.0321	0.0421	0.0538	0.0672	0.0825
--------	--------	--------	--------	--------	--------	--------

Σύντομη περιγραφή όσων έκανα και σχολιασμός των αποτελεσμάτων:

Εξαγωγή συντελεστών: Οι συντελεστές $\alpha_3, \alpha_2, \alpha_1, \alpha_0$ προέκυψαν από την πράξη `polyfit`. Προβλεπόμενοι χρόνοι εκτέλεσης: Η λειτουργία `polyval` χρησιμοποιήθηκε για τον υπολογισμό των προβλεπόμενων χρόνων εκτέλεσης για δύο σύνολα τιμών n : `[100:100:2000]`: Ένα εύρος που συνάδει με τα αρχικά δεδομένα. `[150:100:1550]`: Ένα διαφορετικό υποσύνολο τιμών n για πρόσθετες προβλέψεις. Εμφάνιση των αποτελεσμάτων: Εμφανίστηκαν οι προβλεπόμενοι χρόνοι εκτέλεσης και για τα δύο σύνολα τιμών n .

Σχολιασμός των αποτελεσμάτων: Οι προβλεπόμενοι χρόνοι εκτέλεσης παρέχουν εκτιμήσεις με βάση την κυβική συνάρτηση που λαμβάνεται από τα μετρούμενα δεδομένα. Το πρώτο σύνολο τιμών n (`[100:100:2000]`) αντιστοιχεί στο αρχικό εύρος που χρησιμοποιήθηκε για την προσαρμογή της κυβικής συνάρτησης. Οι προβλέψεις για αυτό το εύρος θα πρέπει να είναι πιο αξιόπιστες, καθώς η συνάρτηση προσαρμόστηκε ειδικά σε αυτά τα δεδομένα. Το δεύτερο σύνολο τιμών n (`[150:100:1550]`) αντιπροσωπεύει ένα διαφορετικό υποσύνολο για το οποίο γίνονται προβλέψεις. Αυτές οι προβλέψεις αποτελούν προεκτάσεις πέραν του αρχικού εύρους και θα πρέπει να ερμηνεύονται με προσοχή. Η προεκβολή εισάγει αβεβαιότητες και η ακρίβεια των προβλέψεων μπορεί να μειωθεί καθώς απομακρύνεστε από το εύρος που χρησιμοποιήθηκε για την προσαρμογή.

2.3



Αυτός ο κώδικας δημιουργεί ένα υποδιάγραμμα με δύο γραφήματα. Το πρώτο υποδιάγραμμα δείχνει τον έλεγχο ακρίβειας για το n στο $[100:100:2000]$ και το δεύτερο υποδιάγραμμα δείχνει τον έλεγχο ακρίβειας για το n στο $[150:100:1550]$.

Για κάθε υποδιάγραμμα: Οι μπλε δείκτες και γραμμές αντιπροσωπεύουν τους πραγματικούς χρόνους εκτέλεσης. Η κόκκινη γραμμή αντιπροσωπεύει τους προβλεπόμενους χρόνους εκτέλεσης. Σύντομη περιγραφή όσων έκανα και σχολιασμός των αποτελεσμάτων: Υπολογισμός προβλέψεων και πραγματικών χρόνων: Οι προβλεπόμενοι χρόνοι εκτέλεσης υπολογίστηκαν για δύο σύνολα τιμών n χρησιμοποιώντας την κυβική συνάρτηση που λαμβάνεται από το `polyfit`. Οι πραγματικοί χρόνοι εκτέλεσης μετρήθηκαν για τα ίδια σύνολα τιμών n . Απεικόνιση: Δημιουργήθηκαν δύο υποδιαγράμματα για την απεικόνιση του ελέγχου ακρίβειας για διαφορετικά εύρη n . Σε κάθε υποδιάγραμμα, οι μπλε δείκτες και γραμμές αντιπροσώπευαν τους πραγματικούς χρόνους εκτέλεσης, ενώ η κόκκινη γραμμή αντιπροσώπευε τους προβλεπόμενους χρόνους εκτέλεσης.

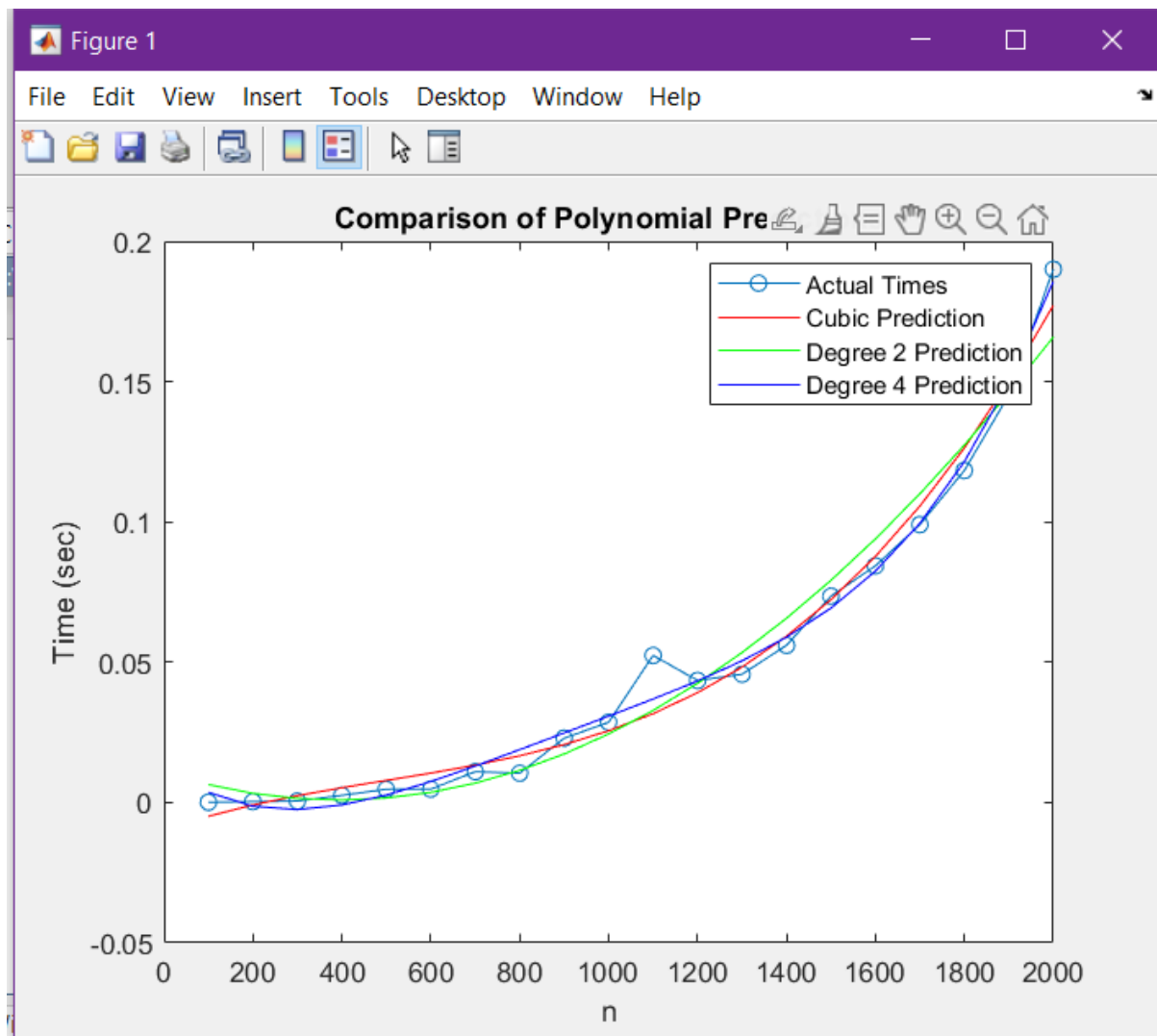
Σχολιασμός αποτελεσμάτων:

Υποδιάγραμμα 1 ($n = [100 : 100 : 2000]$): Οι μπλε δείκτες και γραμμές δείχνουν τους πραγματικούς χρόνους εκτέλεσης που προέκυψαν από μετρήσεις. Η κόκκινη γραμμή αντιπροσωπεύει τους προβλεπόμενους χρόνους εκτέλεσης με βάση την κυβική συνάρτηση. Ιδανικά, η κόκκινη γραμμή θα πρέπει να ακολουθεί στενά την τάση των μπλε δεικτών, υποδεικνύοντας ακριβείς προβλέψεις. Τυχόν αποκλίσεις μεταξύ των προβλεπόμενων και των πραγματικών χρόνων μπορούν να παρέχουν πληροφορίες σχετικά με την ακρίβεια του μοντέλου και τους πιθανούς περιορισμούς του.

Υποδιάγραμμα 2 ($n = [150 : 100 : 1550]$): Παρόμοια με το πρώτο υποδιάγραμμα, αυτό το υποδιάγραμμα επικεντρώνεται σε ένα διαφορετικό εύρος τιμών n για πρόσθετες δοκιμές. Η σύγκριση μεταξύ των πραγματικών και των προβλεπόμενων χρόνων είναι ζωτικής σημασίας για την αξιολόγηση της αξιοπιστίας των προβλέψεων πέραν του αρχικού συνόλου δεδομένων που χρησιμοποιήθηκε για την προσαρμογή.

Παρατηρήσεις: Η στενή ευθυγράμμιση μεταξύ της κόκκινης γραμμής και των μπλε δεικτών υποδηλώνει καλή ακρίβεια στις προβλέψεις. Τυχόν σημαντικές αποκλίσεις ενδέχεται να υποδεικνύουν περιορισμούς ή σφάλματα στη μοντελοποίηση της κυβικής συνάρτησης. Πρέπει να λάβουμε υπόψη ότι η πρόβλεψη των χρόνων εκτέλεσης για τιμές n πέραν του αρχικού συνόλου δεδομένων περιλαμβάνει παρέκταση και τα αποτελέσματα πρέπει να ερμηνεύονται με προσοχή.

2.4



Εδώ προσαρμόζει πολυώνυμα βαθμού 2, 3 (κυβικά) και 4 στα δεδομένα του χρόνου εκτέλεσης και συγκρίνει τις προβλέψεις τους. Τα αποτελέσματα απεικονίζονται σε ένα γράφημα όπου οι πραγματικοί χρόνοι εκτέλεσης αναπαρίστανται με μπλε δείκτες και οι προβλέψεις με κόκκινες, πράσινες και μπλε γραμμές για κυβικά, πολυώνυμα 2ου και 4ου βαθμού αντίστοιχα.

Σύντομη περιγραφή όσων έκανα και σχολιασμός των αποτελεσμάτων:

Παραγωγή δεδομένων: Εκτελέστηκε η λειτουργία αποσύνθεσης Cholesky για πίνακες μεγεθών ($n = [100 : 100 : 2000]$). Μετρήθηκαν οι χρόνοι εκτέλεσης για κάθε μέγεθος πίνακα.

Προσαρμογή πολυωνύμων: Προσαρμόστηκαν πολυώνυμα βαθμού 2, 3 (κυβικά) και 4 στους μετρούμενους χρόνους εκτέλεσης με τη χρήση της συνάρτησης polyfit. Χρησιμοποίησα βάρη για όλες τις προσαρμογές πολυωνύμων για να ενισχύσει την αξιοπιστία.

Υπολογισμός πρόβλεψης: Υπολογισμός προβλεπόμενων χρόνων εκτέλεσης για όλες τις τιμές n χρησιμοποιώντας τους συντελεστές που προέκυψαν για κάθε βαθμό πολυωνύμου.

Απεικόνιση αποτελεσμάτων: Δημιουργήθηκε ένα διάγραμμα για τη σύγκριση των πραγματικών χρόνων εκτέλεσης (μπλε δείκτες) με τους προβλεπόμενους χρόνους από τα πολυώνυμα κυβικού (κόκκινη γραμμή), 2ου βαθμού (πράσινη γραμμή) και 4ου βαθμού (μπλε γραμμή).

Σχολιασμός των αποτελεσμάτων: Η γραφική παράσταση συγκρίνει οπτικά την ακρίβεια των προβλέψεων που γίνονται από πολυώνυμα διαφορετικού βαθμού. Το κυβικό πολυώνυμο προσδιορίστηκε προηγουμένως ως η καλύτερη προσαρμογή για τα αρχικά δεδομένα στο μέρος 1 της ανάλυσης. Συγκρίνοντας τις προβλέψεις των κυβικών πολυωνύμων, των πολυωνύμων 2ου και 4ου βαθμού, μπορούμε να αξιολογήσουμε πόσο καλά κάθε βαθμός αποτυπώνει την υποκείμενη τάση στους χρόνους εκτέλεσης.

Παρατηρήσεις: Οι μπλε δείκτες αντιπροσωπεύουν τους πραγματικούς χρόνους εκτέλεσης, ενώ οι γραμμές αντιπροσωπεύουν τις προβλέψεις. Μια καλή προσαρμογή θα έχει τις προβλεπόμενες γραμμές να ακολουθούν στενά την τάση των πραγματικών σημείων δεδομένων. Οι διαφορές μεταξύ των πραγματικών και των προβλεπόμενων τιμών μπορούν να αναδείξουν τα πλεονεκτήματα και τους περιορισμούς κάθε πολυωνυμικού βαθμού στη μοντελοποίηση των χρόνων εκτέλεσης.

3 Ειδικοί επιλυτές και αραιά μητρώα

3.1

```
function experimentMatrixProperties(n)
    % Initialize the diagonal components with random values
    D = diag(randi([1, 5], n, 1)); % Main diagonal
    L = diag(randi([1, 5], n-1, 1), -1); % Lower diagonal
    U = diag(randi([1, 5], n-1, 1), 1); % Upper diagonal
    A = D - L - U;

    % Check if the matrix is positive definite
    try
        chol(A);
        disp('Matrix is positive definite.');
```

```
    catch
        disp('Matrix is not positive definite.');
```

```
    end

    % Initialize a vector b with random elements
    b = randi([1, 10], n, 1);

    % Calculate A1 and b1 for the system solution
    A1 = D - L * inv(D) * U - U * inv(D) * L - L * inv(D) * L - U * inv(D) * U;
    b1 = b + L * (inv(D) * b) + U * (inv(D) * b);

    % Solve the system
    x = A1 \ b1;
```

```

% Compute A2 and A3 based on A1
A2 = (diag(diag(A1)) + diag(diag(A1)) - A1 * inv(diag(diag(A1)))) * A1;
A3 = (diag(diag(A2)) + diag(diag(A2)) - A2 * inv(diag(diag(A2)))) * A2;

% Display results
disp('Solution x:');
disp(x);
disp('Matrix A:');
disp(A);
disp('Matrix A1:');
disp(A1);
disp('Matrix A2:');
disp(A2);
disp('Matrix A3:');
disp(A3);

% Calculate and display norms
norm_main_diag = norm(diag(D), 2);
norm_sub_diag = norm(diag(L, -1), 2);
norm_hyper_diag = norm(diag(U, 1), 2);
fprintf('Norm of main diagonal: %f\n', norm_main_diag);
fprintf('Norm of sub-diagonal: %f\n', norm_sub_diag);
fprintf('Norm of hyper-diagonal: %f\n', norm_hyper_diag);

% Simplified computational cost for Hadamard products
% Assuming hypothetical direct Hadamard product operations
cost_hadamard = n + 2 * (n - 1); % n for D.*b, (n-1) for L.*b and U.*b
fprintf('Total computational cost estimate for Hadamard products: %d operations\n', cost_hadamard);
end

%Call this function like this
experimentMatrixProperties(6);

```

3.2

Για το συγκεκριμένο ερώτημα βασίστηκα στο σετ 13 διαφάνειες 13 κ.ο. πιο συγκεκριμένα, ως αναλύσουμε τον κώδικα για να δούμε πως δουλεύει.

Αρχικοποίηση πινάκων:

Δημιουργεί έναν τετραγωνικό πίνακα A μεγέθους $n \times n$ ρυθμίζοντας τη διαγώνιο, την κάτω διαγώνιο (μία κάτω από την κύρια διαγώνιο) και την άνω διαγώνιο (μία πάνω από την κύρια διαγώνιο) με τυχαίους ακέραιους αριθμούς μεταξύ 1 και 5. Ο πίνακας A σχηματίζεται στη συνέχεια αφαιρώντας την κάτω και την άνω διαγώνιο από την κύρια διαγώνιο.

Έλεγχος θετικής οριστικότητας:

Προσπαθεί να εκτελέσει μια αποσύνθεση Cholesky στο A για να ελέγξει αν ο A είναι θετικά ορισμένος. Εάν η αποσύνθεση πετύχει, εκτυπώνει ότι ο πίνακας είναι θετικά ορισμένος-

διαφορετικά, εκτυπώνει ότι δεν είναι.

Επίλυση συστήματος:

Η συνάρτηση αρχικοποιεί ένα διάνυσμα β με τυχαίους ακέραιους αριθμούς μεταξύ 1 και 10 και στη συνέχεια υπολογίζει έναν τροποποιημένο πίνακα συστήματος $A1$ και ένα τροποποιημένο διάνυσμα $b1$ της δεξιάς πλευράς. Αυτές οι τροποποιήσεις ενσωματώνουν αντίστροφες πράξεις στο D και αλληλεπιδράσεις με τα L και U . Το σύστημα $A1x=b1$ επιλύεται στη συνέχεια για το διάνυσμα x .

Υπολογισμός τροποποιημένων πινάκων $A2$ και $A3$:

Υπολογίζει δύο περαιτέρω τροποποιημένους πίνακες, τους $A2$ και $A3$, με βάση τον $A1$. Αυτοί οι υπολογισμοί περιλαμβάνουν πράξεις με τα διαγώνια στοιχεία του $A1$ και $A2$, αφαίρεση από αυτούς τους πίνακες και πολλαπλασιασμό με τους αντίστροφους τους. Εμφάνιση αποτελεσμάτων: Η συνάρτηση εξάγει το διάνυσμα της λύσης x , τον αρχικό πίνακα A , και τους τροποποιημένους πίνακες $A1$, $A2$ και $A3$. Υπολογισμός και εμφάνιση προτύπων: Υπολογίζει τα 2-norms της κύριας διαγωνίου του D , της υποδιαγωνίου του L και της υπερδιαγωνίου του U και εμφανίζει αυτές τις νόρμες.

Εκτίμηση υπολογιστικού κόστους για υποθετικά προϊόντα Hadamard:

Παρέχει μια απλουστευμένη εκτίμηση του υπολογιστικού κόστους για την εκτέλεση προϊόντων Hadamard (πολλαπλασιασμός κατά στοιχείο) μεταξύ του πίνακα D και του διανύσματος b , και ομοίως για τα L , U με το b , υποθέτοντας ότι είναι δυνατές οι άμεσες πράξεις.

3.3

```
>> experimentMatrixProperties(6);
Matrix is not positive definite.
Solution x:
  1.6667
  0.6667
 -2.0000
 -4.0000
 -4.0000
 -3.0000

Matrix A:
   5   -5    0    0    0    0
  -2    5   -5    0    0    0
   0   -3    1   -3    0    0
   0    0   -5    5   -5    0
   0    0    0   -5    4   -1
   0    0    0    0   -1    1

Matrix A1:
  3.0000    0   -5.0000    0    0    0
   0  -12.0000    0  -15.0000    0    0
 -1.2000    0   -5.0000    0  -3.0000    0
   0  -15.0000    0  -16.2500    0  -1.2500
   0    0   -5.0000    0  -2.0000    0
   0    0    0   -1.2500    0   0.7500
```

```

Matrix A2:
  4.2000      0      0      0      3.0000      0
      0      1.8462      0      0      0      1.1538
 -0.0000      0      0.5000      0      0      0
      0      0      0      0.4167      0      0
  1.2000      0     -0.0000      0      1.0000      0
      0      1.1538      0      0.0000      0      0.8462

Matrix A3:
  0.6000      0      0.0000      0      0      0
      0      0.2727      0     -0.0000      0      0
 -0.0000      0      0.5000      0      0.0000      0
      0      0      0      0.4167      0      0
 -0.0000      0     -0.0000      0      0.1429      0
      0     -0.0000      0     -0.0000      0      0.1250

Norm of main diagonal: 9.643651
Norm of sub-diagonal: 8.000000
Norm of hyper-diagonal: 9.219544
Total computational cost estimate for Hadamard products: 16 operations

```

```

>> experimentMatrixProperties(5);
Matrix is not positive definite.
Solution x:
 -13.5147
 -17.5147
  -4.4853
   6.1912
   6.9412

```

```

Matrix A:
  1  -1   0   0   0
 -4   3  -1   0   0
  0  -2   3  -3   0
  0   0  -4   4  -5
  0   0   0  -4   4

```

```

Matrix A1:
 -0.3333      0     -0.3333      0      0
      0     -1.6667      0     -1.0000      0
 -2.6667      0     -0.6667      0     -3.7500
      0     -2.6667      0     -5.0000      0
      0      0     -4.0000      0     -1.0000

```

```

Matrix A2:
  1.0000      0      0      0      1.8750
      0     -1.1333      0      0      0
      0      0     17.0000      0      0
      0      0      0     -3.4000      0
  16.0000      0      0      0     21.5000

```

```

Matrix A3:
 -0.3953      0      0      0      0
      0     -1.1333      0      0      0
      0      0     17.0000      0      0
      0      0      0     -3.4000      0
      0      0      0      0     -8.5000

```

```

Norm of main diagonal: 7.141428
Norm of sub-diagonal: 7.211103
Norm of hyper-diagonal: 6.000000
Total computational cost estimate for Hadamard products: 13 operations

```

```
>> experimentMatrixProperties(3);
Matrix is positive definite.
Solution x:
    3.8077
    8.4231
    5.6538
```

```
Matrix A:
    3    -1     0
   -5     4    -1
    0    -3     5
```

```
Matrix A1:
    1.7500         0   -0.2500
         0    1.7333         0
   -3.7500         0    4.2500
```

```
Matrix A2:
    1.5294         0         0
         0    1.7333         0
         0         0    3.7143
```

```
Matrix A3:
    1.5294         0         0
         0    1.7333         0
         0         0    3.7143
```

```
Norm of main diagonal: 7.071068
```

```
Norm of sub-diagonal: 5.830952
```

```
Norm of hyper-diagonal: 1.414214
```

```
Total computational cost estimate for Hadamard products: 7 operations
```

4 Τανυστές

4.1

```
>> test_tensor

ans =

    'Input error: X must be a matrix, and V must be a column vector.'

ans =

    'Input error: X must be a matrix, and V must be a column vector.'

ans =

    'Input error: A and B must be matrices.'

ans =

    'Input error: A and B must be matrices.'
```

4.2

`ttv_myid.m`

Αυτή η συνάρτηση εκτελεί τροπικό πολλαπλασιασμό ενός πολυδιάστατου πίνακα X με ένα διάνυσμα στήλης V κατά μήκος της καθορισμένης διάστασης N . Οι τύποι εισόδου και οι διαστάσεις ελέγχονται για να διασφαλιστεί ότι ο X είναι πίνακας και το V διάνυσμα στήλης. Επιπλέον, οι διαστάσεις ελέγχονται για συμβατότητα με τη λειτουργία τροπικού πολλαπλασιασμού.

`ttm_myid.m`

Παρόμοια με την `ttv_myid.m`, αυτή η συνάρτηση εκτελεί τροπικό πολλαπλασιασμό, αλλά με μια μικρή διαφοροποίηση στη λειτουργία. Πολλαπλασιάζει τον πολυδιάστατο πίνακα X με ένα διάνυσμα στήλης V κατά μήκος της καθορισμένης διάστασης N . Οι τύποι και οι διαστάσεις εισόδου ελέγχονται για να διασφαλιστεί ότι ο X είναι πίνακας και το V διάνυσμα στήλης. Οι διαστάσεις επαληθεύονται επίσης για συμβατότητα με την πράξη τροπικού πολλαπλασιασμού.

`ttt_myid.m`

Αυτή η συνάρτηση παρέχει μια απλή μορφή εξωτερικών και εσωτερικών παραγώγων τανυστών. Για το εξωτερικό γινόμενο, επιστρέφει ένα MDA που περιέχει τις τιμές και τη σωστή διάταξη του τανυστικού γινομένου των A και B . Για το εσωτερικό γινόμενο, επιστρέφει το κλιμακωτό αποτέλεσμα του στοιχειομετρικού πολλαπλασιασμού και του αθροίσματος των A και B . Οι τύποι εισόδου ελέγχονται για να διασφαλιστεί ότι τόσο ο A όσο και ο B είναι

πίνακες. Το προαιρετικό όρισμα `mod` επιτρέπει στο χρήστη να καθορίσει αν επιθυμεί το εξωτερικό ή το εσωτερικό γινόμενο.

`test_tensor.m`

Αυτή η συνάρτηση αρχικοποιεί τους τανυστές A και B , μαζί με άλλες παραμέτρους, και στη συνέχεια ελέγχει την ορθότητα των υλοποιημένων συναρτήσεων (`ttv_myid.m`, `ttm_myid.m`, `ttt_myid.m`) συγκρίνοντας τα αποτελέσματά τους με αντίστοιχες λειτουργίες στην εργαλειοθήκη Tensor Toolbox. Η συνάρτηση ελέγχει για τυχόν σφάλματα ή αναντιστοιχίες και αναφέρει τα αποτελέσματα. Στόχος είναι να διασφαλιστεί ότι οι προσαρμοσμένες υλοποιήσεις παράγουν τα ίδια αποτελέσματα με τις εντολές της Tensor Toolbox.

5 Επίλυση ΣΘΟ συστημάτων με PCG

Στην συγκεκριμένη ερώτηση δεν μπόρεσα να βγάλω σωστά αποτελέσματα, παρόλα αυτά επειδή ασχολήθηκα πολύ έφτιαξα αρκετά αξιόλογο κώδικα για να μπορώ να εξηγήσω το σκεπτικό πίσω από τον κώδικα.

5.1 `poison_test_1070263.m`

Ρύθμιση παραμέτρων: Όρισα παραμέτρους όπως ο αριθμός των εσωτερικών κόμβων (n), ο αριθμός των οριακών συνθηκών (m) και η απόσταση μεταξύ διαδοχικών κόμβων (h).

Ορισμός του πλέγματος: Δημιούργησα ένα πλέγμα (X and Y) χρησιμοποιώντας το `meshgrid` για να αναπαραστήσω το χωρικό πεδίο. Κατασκεύασα τον πίνακα Poisson και το διάνυσμα της δεξιάς πλευράς: Ύστερα χρησιμοποίησα μια συνάρτηση `constructPoissonMatrix_1070263` για να κατασκευάσω τον πίνακα Poisson A και να δημιουργήσω το διάνυσμα b της δεξιάς πλευράς.

Ρύθμιση δοκιμών προαπαιτούμενων συνθηκών (Preconditioning Tests): Δοκίμασα τον επαναληπτικό επιλύτη (μέθοδος συζυγών βαθμίδων με προϋποθέσεις) χωρίς προϋποθέσεις ("none"). Επίσης έκανα δοκιμή με ελλιπή προρύθμιση Cholesky ('ichol') και δοκιμή με προσαρμοσμένη προεπεξεργασία ('custom').

Απεικόνιση: Σχεδίασα το σχετικό υπόλειμμα για κάθε επανάληψη και για τις τρεις στρατηγικές προετοιμασίας. Έκανα απεικόνιση του σχετικού σφάλματος για κάθε επανάληψη και για τις τρεις στρατηγικές προετοιμασίας. Τέλος, γίνεται η εμφάνιση των παραγόμενων γραφικών παραστάσεων.

5.2 `pcg_1070263`

Η συνάρτηση του MATLAB είναι μια τροποποιημένη έκδοση της μεθόδου Preconditioned Conjugate Gradients (PCG), με την ονομασία `pcg_1070263`. Έχει σχεδιαστεί για την επίλυση ενός συστήματος γραμμικών εξισώσεων $Ax = b$ με τη δυνατότητα χρήσης ενός προρρυθμιστή.

Παράμετροι εισόδου:

A : Ο πίνακας συντελεστών ή μια λαβή συνάρτησης που αναπαριστά έναν γραμμικό τελεστή.
 b : Το διάνυσμα της δεξιάς πλευράς. `tol`: Ανοχή σύγκλισης. `maxit`: Μέγιστος αριθμός

επαναλήψεων.

preconditioner: Συμβολοσειρά που καθορίζει τον τύπο του preconditioner που θα χρησιμοποιηθεί ('none', 'ichol', ή 'custom').

x0: Αρχική εικασία για τη λύση.

xsol: Ακριβής λύση (για τον υπολογισμό σφαλμάτων).

varargin: Πρόσθετα ορίσματα που μεταβιβάζονται στις συναρτήσεις του γινόμενου μήτρας-διανύσματος και του preconditioner.

Παράμετροι εξόδου:

x: Η υπολογισμένη λύση.

flag: Σημαία σύγκλισης (0 για σύγκλιση).

relres: Σχετικό υπόλειμμα.

iter: Αριθμός επαναλήψεων που εκτελούνται.

resvec: Διάνυσμα που αποθηκεύει τη νόρμα του υπολοίπου σε κάθε επανάληψη.

errvec: Διάνυσμα που αποθηκεύει το σχετικό σφάλμα σε κάθε επανάληψη.

Προσδιορισμός τύπου λειτουργίας:

Η συνάρτηση προσδιορίζει αν ο πίνακας συντελεστών A είναι ένας αριθμητικός πίνακας ή μια συνάρτηση, ελέγχοντας τον τύπο του (πίνακας ή λαβή συνάρτησης). Ορίζει ανάλογα μια κατάλληλη λαβή συνάρτησης (afun).

Επικύρωση παραμέτρων εισόδου:

Ελέγχει τα μεγέθη και τους τύπους των παραμέτρων εισόδου, όπως A , b , tol , $maxit$, $preconditioner$, $x0$ $xsol$.

Παρέχει προεπιλεγμένες τιμές για ορισμένες παραμέτρους εάν δεν έχουν καθοριστεί.

Εφαρμογή προαπαιτούμενου:

Εάν έχει καθοριστεί ένας προκλιματιστής ('ichol' ή 'custom'), εφαρμόζει την αντίστοιχη λειτουργία προκλιματισμού. Εάν δεν έχει καθοριστεί κανένας προκαθοριστής, χρησιμοποιεί τον πίνακα ταυτότητας ως προκαθοριστή τοποθέτησης.

Κύριος επαναληπτικός βρόχος:

Εφαρμόζει τη μέθοδο Preconditioned Conjugate Gradients, επαναλαμβάνοντας μέχρι τη σύγκλιση ή τη συμπλήρωση του μέγιστου αριθμού επαναλήψεων. Υπολογίζει την κατεύθυνση αναζήτησης (p) και το γινόμενο μήτρας-διανύσματος (q) σε κάθε επανάληψη.

Έλεγχοι σύγκλισης και στασιμότητας:

Παρακολουθεί τη σύγκλιση συγκρίνοντας τη νόρμα του υπολείμματος ($normr_act$) με την καθορισμένη ανοχή (tol). Ελέγχει τη στασιμότητα εξετάζοντας το λόγο της νόρμας της κατεύθυνσης αναζήτησης προς τη νόρμα της λύσης.

Αποθήκευση αποτελεσμάτων:

Καταγράφει τις σχετικές πληροφορίες, όπως τις υπολειμματικές νόρμες σε κάθε επανάληψη ($resvec$), το σφάλμα σε κάθε επανάληψη ($errvec$) και τον αριθμό των επαναλήψεων που εκτελέστηκαν.

Χειρισμός ακραίων περιπτώσεων:

Αντιμετωπίζει ειδικές περιπτώσεις, όπως μια αρχική εικασία που είναι ήδη μια καλή λύση ή

ένα διάνυσμα της δεξιάς πλευράς με όλα τα μηδενικά.

Έξοδος και εμφάνιση:

Επιστρέφει την υπολογισμένη λύση (x), τη σημαία σύγκλισης ($flag$), το σχετικό υπόλοιπο (ρελρες), τον αριθμό των επαναλήψεων ($iter$) και τα διανύσματα που αποθηκεύουν τις πληροφορίες για το υπόλοιπο και το σφάλμα. Εμφανίζει πληροφορίες σύγκλισης εάν δεν ζητείται ρητά η σημαία εξόδου.

Χειρισμός σφαλμάτων και προειδοποιήσεων:

Ελέγχει για πιθανά προβλήματα κατά τη διάρκεια των επαναλήψεων, όπως άπειρες τιμές, και θέτει τις κατάλληλες σημαίες ($flag$) και προειδοποιήσεις.

Ευελιξία:

Επιτρέπει στους χρήστες να ορίζουν προσαρμοσμένες στρατηγικές προκλιμάκωσης παρέχοντας τη δική τους συνάρτηση προκλιμάκωσης. Αυτή η συνάρτηση έχει σχεδιαστεί για να είναι ευέλικτη, να χειρίζεται διαφορετικούς τύπους πινάκων εισόδου, να παρέχει επιλογές για την προαποτύπωση και να προσφέρει ευελιξία στο χειρισμό διαφόρων σεναρίων σύγκλισης. Στόχος της είναι να είναι ισχυρή στην επίλυση γραμμικών συστημάτων με τη μέθοδο Preconditioned Conjugate Gradients.

5.3 `constructPoissonMatrix_1070263`

Αυτή η συνάρτηση παράγει τον πίνακα Poisson για δεδομένο πλέγμα και οριακές συνθήκες.

Πιο αναλυτικά:

Παράμετροι εισόδου:

X : Συντεταγμένες πλέγματος.

(αχρησιμοποίητο): Παράμετρος που αγνοείται.

h : Απόσταση μεταξύ διαδοχικών κόμβων.

Κατασκευή πινάκων:

Αρχικοποιεί έναν κενό αραιό πίνακα A με διαστάσεις $n \times m$ όπου n και m είναι οι διαστάσεις του πλέγματος X . Επαναλαμβάνει κάθε σημείο του πλέγματος και κατασκευάζει τις καταχωρήσεις του πίνακα Poisson. Ορίζει το γριδ ποιντ σε $-4/h^2$ Θέτει τα στοιχεία εκτός διαγωνίου που αντιστοιχούν στα γειτονικά σημεία σε $1/h^2$

Χειρισμός οριακών συνθηκών:

Μηδενίζει τα στοιχεία της πρώτης και της τελευταίας γραμμής (που αντιπροσωπεύουν τα άνω και κάτω όρια) και της πρώτης και της τελευταίας στήλης (που αντιπροσωπεύουν τα αριστερά και τα δεξιά όρια). Αυτό εφαρμόζει αποτελεσματικά τις συνοριακές συνθήκες Dirichlet, καθορίζοντας τις τιμές της λύσης στα συνοριακά σημεία στο μηδέν.

Έξοδος:

Επιστρέφει τον κατασκευασμένο πίνακα Poisson A . Αυτή η συνάρτηση είναι ειδικά προσαρμοσμένη για την κατασκευή του πίνακα Poisson για ένα δισδιάστατο πλέγμα με οριακές συνθήκες Dirichlet.

5.4 customPreconditioner

Η συνάρτηση, παράγει έναν προσαρμοσμένο ατελή προρυθμιστή Cholesky (IC) για έναν δεδομένο πίνακα A χρησιμοποιώντας τη συνάρτηση ichol.

Παράμετρος εισόδου:

A: Ο πίνακας για τον οποίο κατασκευάζεται ο preconditioner.

Κατασκευή preconditioner: Ο πίνακας που χρησιμοποιείται για την κατασκευή του προκοντίζιονερ χρησιμοποιεί τη συνάρτηση ichol, η οποία υπολογίζει μια ατελή παραγοντοποίηση Cholesky του πίνακα εισόδου.

Οι επιλογές για την ichol καθορίζονται μέσω μιας δομής struct:

'type': Καθορίζει τον τύπο της ατελούς παραγοντοποίησης ('ict' για IC με κατώφλι).

'droptol': Ορίζει την ανοχή πτώσης για μικρά στοιχεία κατά την παραγοντοποίηση.

'shape': Καθορίζει εάν θα υπολογιστεί το άνω ή το κάτω τριγωνικό μέρος της παραγοντοποίησης.

5.5 incompleteCholesky

Παράμετρος εισόδου:

A: Ο πίνακας για τον οποίο εκτελείται η ατελής παραγοντοποίηση Cholesky.

Παραγοντοποίηση:

Χρησιμοποιεί τη συνάρτηση cholinc για την εκτέλεση ατελούς παραγοντοποίησης Cholesky με μηδενική πλήρωση ('U'). Η παραγοντοποίηση εκτελείται με επίπεδο μηδενικής πλήρωσης, πράγμα που σημαίνει ότι δεν προστίθενται πρόσθετα στοιχεία στον παραγοντοποιημένο πίνακα για να διασφαλιστεί η αραιότητα.

Έξοδος:

Επιστρέφει τον κατώτερο τριγωνικό παράγοντα L και τον ανώτερο τριγωνικό παράγοντα U της ατελούς παραγοντοποίησης Cholesky.

5.6 solveExactPoisson

Ψπολογίζει την ακριβή λύση για την εξίσωση Ποισσον σε ένα δεδομένο πλέγμα που ορίζεται από τα X και Y

5.7 SuiteSparse.m

Αυτός ο κώδικας επιλύει ένα γραμμικό σύστημα που ορίζεται από Το μητρώο υπ' αριθμ. 1 του SuiteSparse, δηλ. το 1138.βυς και το διάνυσμα της δεξιάς πλευράς b χρησιμοποιώντας τη μέθοδο Preconditioned Conjugate Gradients (PCG) με την τροποποιημένη συνάρτηση (pcg_1070263).

Ορισμός μητρώου:

A = Πρόβλημα: Αρχικοποιεί τον πίνακα συντελεστών A με βάση το μητρώο Problem.

Διάνυσμα δεξιάς πλευράς:

b = rand(size(A, 1), 1); Δημιουργεί ένα τυχαίο διάνυσμα δεξιάς πλευράς β με τον ίδιο

αριθμό γραμμών με τον πίνακα A.

Προκαθοριστής και όριο επανάληψης:

preconditioner = 'ichol';: Καθορίζει τον τύπο του preconditioner που θα χρησιμοποιηθεί στη μέθοδο PCG. Σε αυτή την περίπτωση, ορίζεται σε ατελή παραγοντοποίηση Cholesky ('ichol'). Μπορείτε να το αλλάξετε σε άλλες επιλογές όπως 'none', 'custom', κ.λπ.

maxiter = 4 * size(A, 1);: Ορίζει ένα ανώτερο όριο στον αριθμό των επαναλήψεων για τη μέθοδο PCG.

PCG Solver:

x, flag, relres, iter, resvec, errvec

= pcg_myid_1070263(A, b, 1e-6, maxiter, preconditioner, [], [], 'resvec', [], 'errvec', []);: Καλεί τον τροποποιημένο επιλυτή PCG (pcg_myid_1070263) για την επίλυση του γραμμικού συστήματος. Επιστρέφει τη λύση x, τη σημαία της σημαίας σύγκλισης, το σχετικό υπόλοιπο relres, τον αριθμό επαναλήψεων iter και τα διανύσματα που αποθηκεύουν τις πληροφορίες για το υπόλοιπο (resvec) και το σφάλμα (errvec).

Αυτός ο κώδικας συγκρίνει την απόδοση της μεθόδου (PCG) με την τροποποιημένη συνάρτηση (pcg_1070263). Για την επίλυση ενός γραμμικού συστήματος χρησιμοποιώντας διαφορετικές στρατηγικές προετοιμασίας.

Αριθμός διαδρομών:

Αριθμός διαδρομών = 10:: Καθορίζει τον αριθμό των εκτελέσεων για τη συγκριτική αξιολόγηση. Συγκριτική αξιολόγηση χωρίς προαπαιτούμενα: Χρησιμοποιεί έναν βρόχο για την εκτέλεση της μεθόδου PCG χωρίς προαπαιτούμενα για πολλαπλές εκτελέσεις. Μετρά το χρόνο εκτέλεσης για κάθε εκτέλεση και υπολογίζει το μέσο χρόνο.

Benchmarking With Incomplete Cholesky Preconditioning:

Παρόμοια με την περίπτωση χωρίς προαπαιτούμενα, αλλά με τον προαπαιτούμενο 'ichol'. Μετρά το χρόνο εκτέλεσης για κάθε εκτέλεση και υπολογίζει το μέσο χρόνο. Συγκριτική αξιολόγηση με προσαρμοσμένη προεπεξεργασία: Παρόμοια με τις προηγούμενες περιπτώσεις, αλλά με έναν προσαρμοσμένο preconditioner. Μετρά το χρόνο εκτέλεσης για κάθε εκτέλεση και υπολογίζει το μέσο χρόνο.

Εμφάνιση του μέσου χρόνου εκτέλεσης:

Εκτυπώνει τους μέσους χρόνους εκτέλεσης για κάθε στρατηγική προκλιμάκωσης.

Αρχικοποίηση πίνακα και συλλογή δεδομένων:

Αρχικοποιεί πίνακες για την αποθήκευση των αποτελεσμάτων για κάθε στρατηγική προκλιμάκωσης. Συμπληρώνει τους πίνακες με πληροφορίες όπως η μέθοδος που χρησιμοποιήθηκε, η σημαία σύγκλισης, το τελικό σχετικό υπόλειμμα, το τελικό σχετικό σφάλμα και ο μέσος χρόνος εκτέλεσης.

Εμφάνιση πινάκων:

Εμφανίζει τους πίνακες που περιέχουν τα αποτελέσματα για κάθε στρατηγική προκλιμάκωσης.

Ευχαριστώ για τον χρόνο σας!