

**ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΗΛΕΚΤΡΟΝΙΚΩΝ
ΥΠΟΛΟΓΙΣΤΩΝ ΚΑΙ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΠΑΝΕΠΙΣΤΗΜΙΟΥ ΠΑΤΡΩΝ**

ΕΠΙΣΤΗΜΟΝΙΚΟΣ ΥΠΟΛΟΓΙΣΜΟΣ

ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ 2018-19

ΚΟΥΛΟΥΡΗΣ ΓΡΗΓΟΡΗΣ Α.Μ. 6092 (1041782 ΝΕΟ Α.Μ.)

1. ΠΑΡΟΥΣΙΑΣΗ ΥΠΟΛΟΓΙΣΤΙΚΟΥ ΠΕΡΙΒΑΛΛΟΝΤΟΣ

ΣΥΣΤΗΜΑ	ΕΠΕΞΕΡΓΑΣΤΗΣ	ΕΠΙΠΕΔΑ ΜΝΗΜΗΣ	LAPACK	FMA	ΕΚΔΟΣΗ MATLAB
<i>Windows 10 Pro 64-Bit</i>	<i>Intel i5 2 x 2.3 GHz</i>	<i>4 GB RAM 250 GB ROM</i>	<i>3.7.0</i>	<i>OXI</i>	<i>R2018a academic use</i>

2. ΕΠΙΛΥΣΗ ΣΥΣΤΗΜΑΤΟΣ ΜΕ ΑΝΑΝΕΩΣΕΙΣ ΜΕΙΩΜΕΝΗΣ ΤΑΞΗΣ

(Διευκρινίζω ότι στα ονόματα των συναρτήσεων έχω χρησιμοποιήσει τα 4 τελευταία ψηφία του νέου Α.Μ.)

2.1. Η συνάρτηση *SMW_solve_1782* σε κώδικα MATLAB

```
function [ x ] = SMW_solve_1782a ( A, b, M, P, Q, sdir)
n = length(A);
if isequal( sdir , 'colwise' )
    M = diag(diag(A));
    C = A-M;
    P = [];
    for i = 1:n
        P = [P C(:,i)];
    end
    Q = (inv(P)*C)';

elseif isequal( sdir , 'rowwise' )
    M = diag(diag(A));
    C = A-M;
    Q = [];
    for i = 1:n
        Q = [Q ;C(i,:)];
    end
    P = C * inv(Q');

end

m = length(P(:,1));
A0 = M;
x0 = inv(A0)*b;

Pi = cell(m,m);
for i = 1:m
    Pi{1,i} = inv(A0)*P(:,i); % y0,i Mhtrwo dianismatwn sthlhs
end

for i = 2 : m
    for j = i : m
        Pi{i,j} = Pi{i-1,j} - ((Q(:,i-1)' * (Pi{i-1,j}))) / (1 + (Q(:,i-1)' *
Pi{i-1,i-1}))) * (Pi{i-1,i-1});
    end
end

for(i=2:m)
    x0 = x0 - ((Q(:,i-1)' * (x0))/(1 + (Q(:,i-1)'*Pi{i-1,i-1}))) * (Pi{i-1,i-
1});
end
[ x ] = x0 - ((Q(:,m)')*(x0) / (1 + (Q(:,m)' * Pi{m,m}))) * (Pi{m,m});
end
```

2.2 Περιγραφή του αλγορίθμου και συζήτηση σχετικά με τον σχεδιασμό του.

Βασμισμένος στον αλγόριθμο 1 από το paper του P.Maroni “ The solution of linear systems by using the Sherman–Morrison formula”. Ο αλγόριθμος παίρνει σαν είσοδο ένα $n \times n$ μητρώο A το οποίο κατασκευάζουμε με την συνάρτηση *Mx_Make_1782(mx_id,n)* , ένα

διάνυσμα b το οποίο το έχω κατασκευάσει όπως διευκρινίζεται στην εκφώνηση, τρία μητρώα M , P , Q τα οποία ικανοποιούν την σχέση $(M + PQ^T)x = b$ καθώς και μία μεταβλητή $sdir$ σύμφωνα με την οποία υπολογίζονται τα μητρώα αυτά.

Αρχικά ελέγχουμε την τιμή της μεταβλητής $sdir$ και ανάλογα με την τιμή της κατασκευάζουμε τα μητρώα M , P και Q . Στη συνέχεια δημιουργώ ένα cell και σε κάθε κελί αποθηκεύω ένα διάνυσμα γραμμής P_i . Τέλος υπολογίζω αναδρομικά το x το οποίο και επιστρέφω.

Σχετικά με τον σχεδιασμό του αλγορίθμου, αυτό που με δυσκόλεψε ήταν η υλοποίηση του cell για να αποθηκεύω τα P_i καθώς και ο συσχετισμός του l με το k στο βήμα (III) του αλγορίθμου (1).

2.3 Δίκτης κατάστασης $k_{(inf)}$, Σχετικό πίσω σφάλμα, Φράγμα για το συνεπαγόμενο εμπρός σφάλμα, Ακριβές σχετικό εμπρός σφάλμα για:

A) Την `SMW_solve_AM` με την επιλογή 'colwise'

B) $x = A \backslash b$

```
disp('Δίκτης Κατάστασης k inf')
k = norm(A,inf) * norm(inv(A),inf); % dikths katastashs k inf
disp(k)
r = b - A*x;
disp('Σχετικό πίσω σφάλμα')
posteriori = norm(r) / (norm(A)*norm(x) + norm(b)); %sxetiko pishw sfalma
disp(posteriori)
disp('Άνω φράγμα για σχετικό εμπρός σφάλμα')
sxetiko_empros_sfalma = (2*k*posteriori)/(1-(k*posteriori));
disp(sxetiko_empros_sfalma)

x_abs = A\b;
r = b - A*x_abs;
posteriori_abs = norm(r) / (norm(A)*norm(x_abs) + norm(b));
disp('Άνω φράγμα για ακριβές σχετικό εμπρός σφάλμα')
emprios_sfalma_abs = (2*k*posteriori_abs)/(1-(k*posteriori_abs));
disp(emprios_sfalma_abs)
```

Με τον παραπάνω κώδικα υπολογίζω τα ζητούμενα για ένα μητρώο A αρχικά για την λύση της συνάρτησης που υλοποίησα και στην συνέχεια για την συνάρτηση της MATLAB $x = A \backslash b$. Παρακάτω σας παραθέτω τα

αποτελέσματα για κάθε ένα από τα 6 κατασκευασμένα μητρώα που χρησιμοποιήσα.

1. $A = \text{MxMake_1782}('had', 64)$

Δίκτης Κατάστασης $k \inf$
64

Σχετικό πίσω σφάλμα
4.0723e-16

Άνω φράγμα για σχετικό εμπρός σφάλμα
5.2126e-14

Άνω φράγμα για ακριβές σχετικό εμπρός σφάλμα
3.2334e-14

2. $A = \text{MxMake_1782}('trihad', 64)$

Στο συγκεκριμένο μητρώο η συνάρτηση μου επιστρέφει το διάνυσμα χ σαν NaN .

Δίκτης Κατάστασης $k \inf$

Warning: Matrix is close to singular or badly scaled. Results may be inaccurate.

> In [query3](#) (line 2)

7.8541e+15

Σχετικό πίσω σφάλμα
NaN

Άνω φράγμα για σχετικό εμπρός σφάλμα
NaN

Άνω φράγμα για ακριβές σχετικό εμπρός σφάλμα
0.0385

3. $A = \text{MxMake_1782}('toep', 64)$

Δίκτης Κατάστασης $k \inf$
3

Σχετικό πίσω σφάλμα
1.3621e-16

Άνω φράγμα για σχετικό εμπρός σφάλμα
8.1724e-16

Άνω φράγμα για ακριβές σχετικό εμπρός σφάλμα
3.3597e-16

4. `A = MxMake_1782('mc' , 400)`

Δίκτης Κατάστασης `k inf`
335.5220

Σχετικό πίσω σφάλμα
8.1684e-17

Άνω φράγμα για σχετικό εμπρός σφάλμα
5.4813e-14

Άνω φράγμα για ακριβές σχετικό εμπρός σφάλμα
2.5271e-14

|

5. `A = MxMake_1782('wathen' , 12)`

Εδώ χρησιμοποίησα την συνάρτηση `normest()`.

Δίκτης Κατάστασης `k inf`
2.6754e+04

Σχετικό πίσω σφάλμα

Warning: NORMEST did not converge for 100 iterations with tolerance 1e-06

> In `normest` (line 44)

In `query3` (line 6)

1.2281e-16

Άνω φράγμα για σχετικό εμπρός σφάλμα
6.5714e-12

Warning: NORMEST did not converge for 100 iterations with tolerance 1e-06

> In `normest` (line 44)

In `query3` (line 14)

Άνω φράγμα για ακριβές σχετικό εμπρός σφάλμα
1.0765e-12

6. `A = eye(1899) - 0.85 * MxMake_1782('CollegeMsg' , 1)`

Στο συγκεκριμένο μητρώο η Matlab μου πετάει out of Memory error στην διπλά εμφολευμένη for της συνάρτησής μου.

Διαχείριση αρχείων

Στο αρχείο Main.m περιέχονται η δημιουργία των μητρώων A και του διανύσματος b καθώς και η κλίση της συνάρτησης `SMW_solve_AM()` .

Στο αρχείο Query3.m περιέχετε η λύση του ερωτήματος 3 του 2^{ου} μέρους της άσκησης.