**ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ**

**ΠΑΝΕΠΙΣΤΗΜΙΟ ΙΩΑΝΝΙΝΩΝ**

**ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ**

**ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΕΠΙΣΤΗΜΗΣ ΥΛΙΚΩΝ**

# ΕΡΓΑΣΤΗΡΙΑΚΕΣ ΣΗΜΕΙΩΣΕΙΣ ΤΟΥ ΠΡΟΠΤΥΧΙΑΚΟΥ ΜΑΘΗΜΑΤΟΣ ΑΡΙΘΜΗΤΙΚΗ ΑΝΑΛΥΣΗ

**ΕΚΔΟΣΗ 1η**

**ΙΩΑΝΝΗΣ Ο. ΑΝΔΡΙΚΟΣ**

**ΕΛΕΝΗ Ι. ΓΕΩΡΓΑ**

**ΔΗΜΗΤΡΙΟΣ Ι. ΦΩΤΙΑΔΗΣ**

**ΙΩΑΝΝΙΝΑ, 2018**

## Περιεχόμενα:

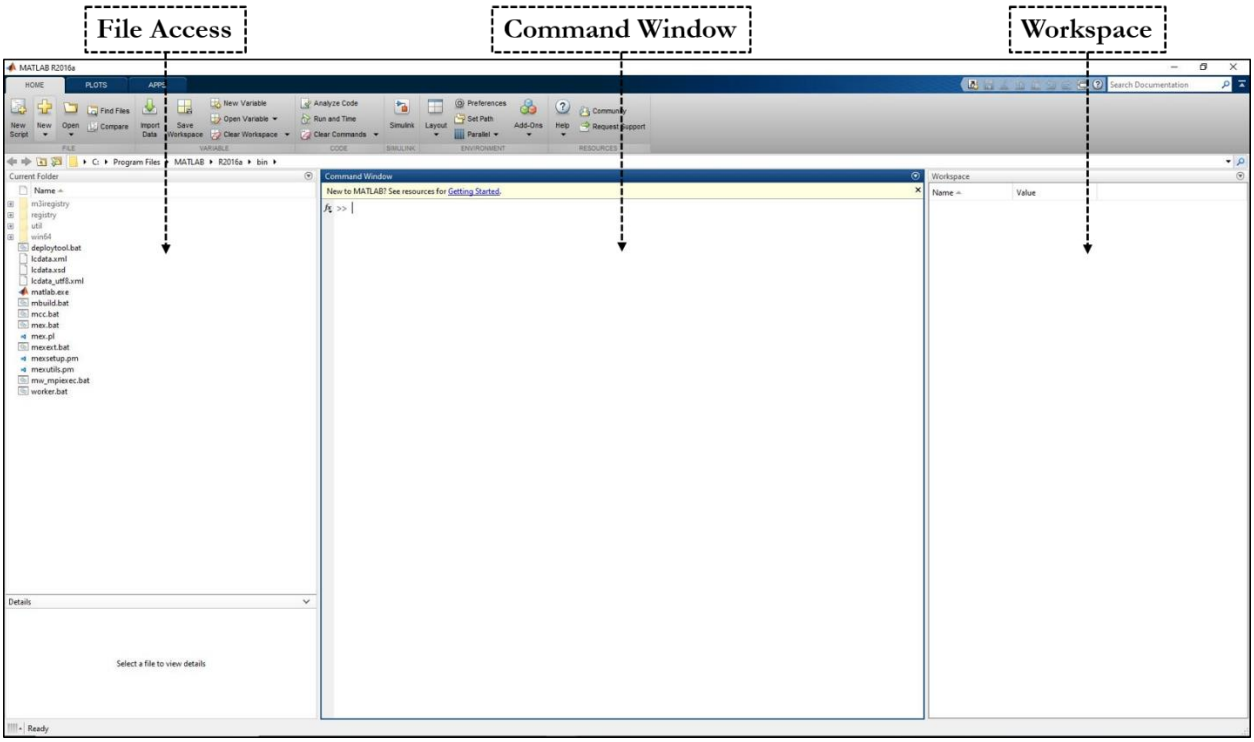
# 1 Εισαγωγή στη MATLAB

Η MATLAB είναι ένα σύγχρονο ολοκληρωμένο μαθηματικό λογισμικό πακέτο που χρησιμοποιείται σε πανεπιστημιακά μαθήματα αλλά και ερευνητικές και άλλες εφαρμογές με επιστημονικούς υπολογισμούς (scientific computing). Το όνομά του προέρχεται από τα αρχικά γράμματα των λέξεων MATtrix LABoratory (εργαστήριο πινάκων).

Όπως υποδηλώνεται και από το όνομά της, η MATLAB είναι ειδικά σχεδιασμένη για υπολογισμούς με πίνακες, όπως η επίλυση γραμμικών συστημάτων, η εύρεση ιδιοτιμών και ιδιοδιανυσμάτων, η αντιστροφή τετραγωνικών πινάκων κλπ. Επιπλέον το πακέτο αυτό είναι εφοδιασμένο με πολλές επιλογές για γραφικά (δηλ. την κατασκευή γραφικών παραστάσεων) και προγράμματα γραμμένα στη δική του γλώσσα προγραμματισμού για την επίλυση άλλων προβλημάτων όπως η εύρεση των ριζών μη γραμμικής εξίσωσης, η επίλυση μη γραμμικών συστημάτων, η επίλυση προβλημάτων αρχικών τιμών με συνήθεις διαφορικές εξισώσεις, κ.α. Το λογισμικό MATLAB είναι σχεδιασμένο για την αριθμητική επίλυση προβλημάτων σε αριθμητική πεπερασμένης ακρίβειας (finite-precision arithmetic), δηλαδή δεν βρίσκει την ακριβή αλλά μια προσεγγιστική λύση ενός προβλήματος. Επιπλέον, εκτός από τις έτοιμες βιβλιοθήκες και συναρτήσεις, η γλώσσα προγραμματισμού MATLAB δίνει την ευχέρεια στον χρήστη να το επεκτείνει με δικά του προγράμματα [1].

## 1.1 Το γραφικό περιβάλλον της MATLAB

Το γραφικό περιβάλλον της MATLAB χωρίζεται σε 3 μέρη: 1) Το File Access, 2) Το Command Window, 3) Το Workspace.

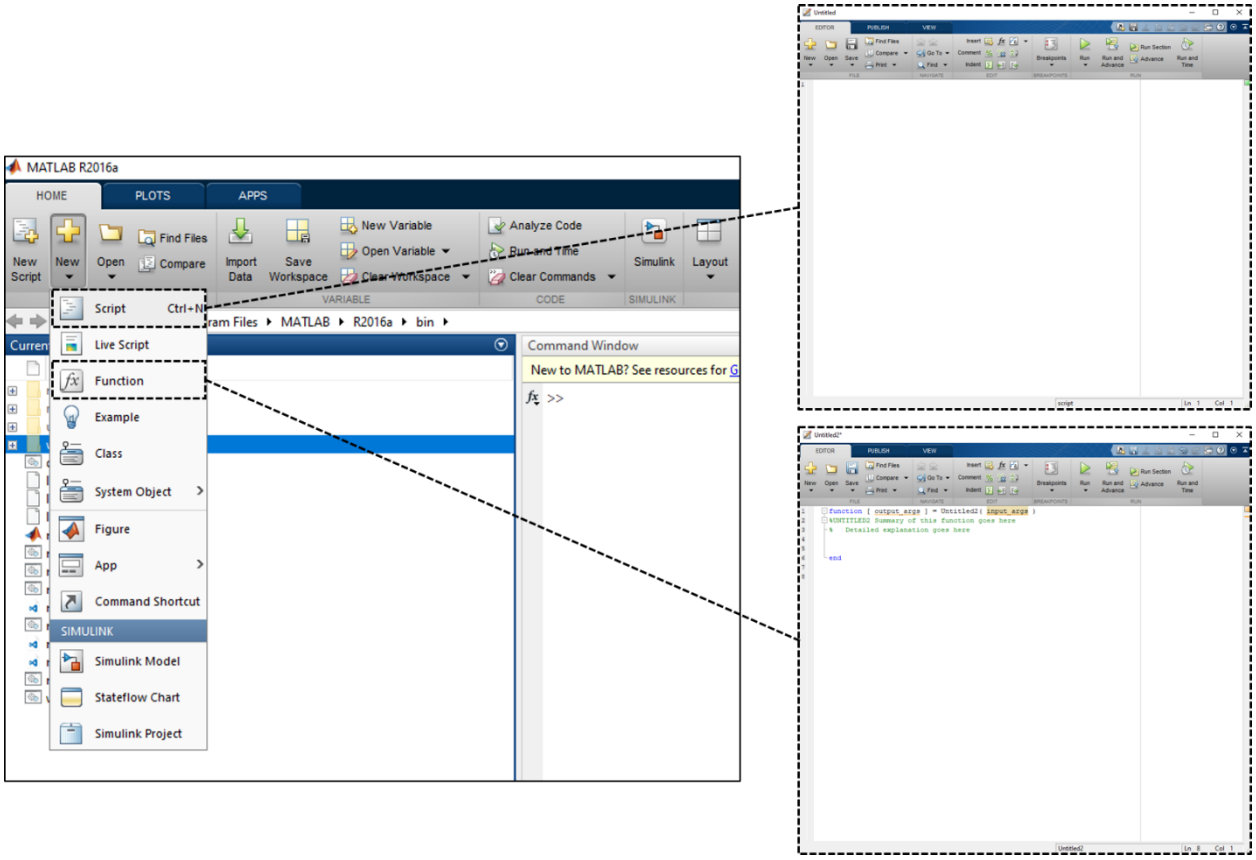



- ✓ Μέσα από το File Access του MATLAB ο χρήστης μπορεί να πλοηγηθεί στο σύστημα αρχείων του λειτουργικού συστήματος, να αναζητήσει καθώς και να επιλέξει φακέλους που περιέχουν εκτελέσιμα αρχεία (Scripts) και συναρτήσεις (functions). Και τα δύο ήδη αρχείων έχουν την κατάληξη (*.m).

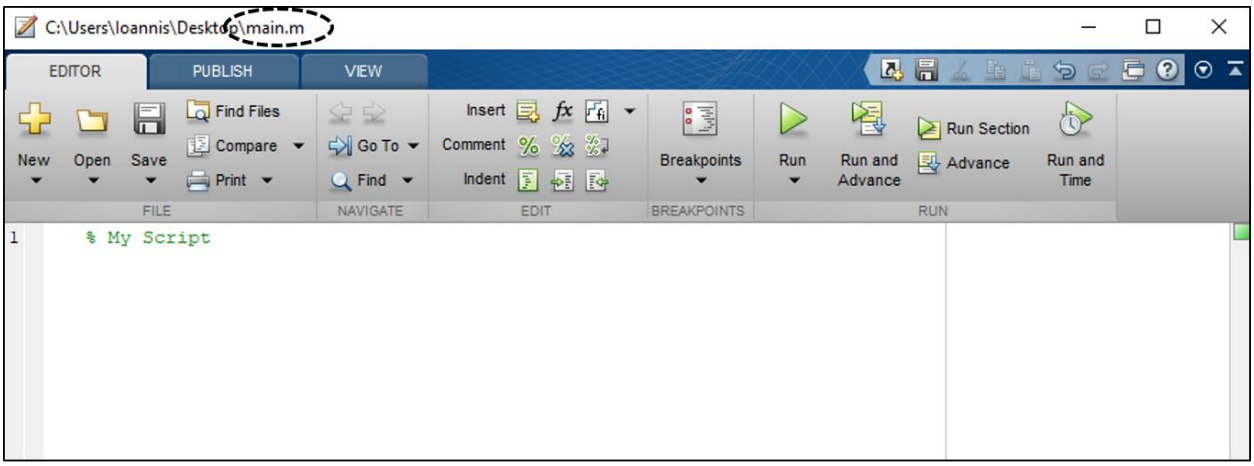
- ✓ Στο command Window ο χρήστης μπορεί απευθείας να γράψει κώδικα και να τον εκτελέσει γραμμή προς γραμμή.
- ✓ Τέλος, στο Workspace ο χρήστης μπορεί να παρατηρήσει τις αναθέσεις των μεταβλητών καθώς και τα αποτελέσματα των πράξεων που πραγματοποίησε.

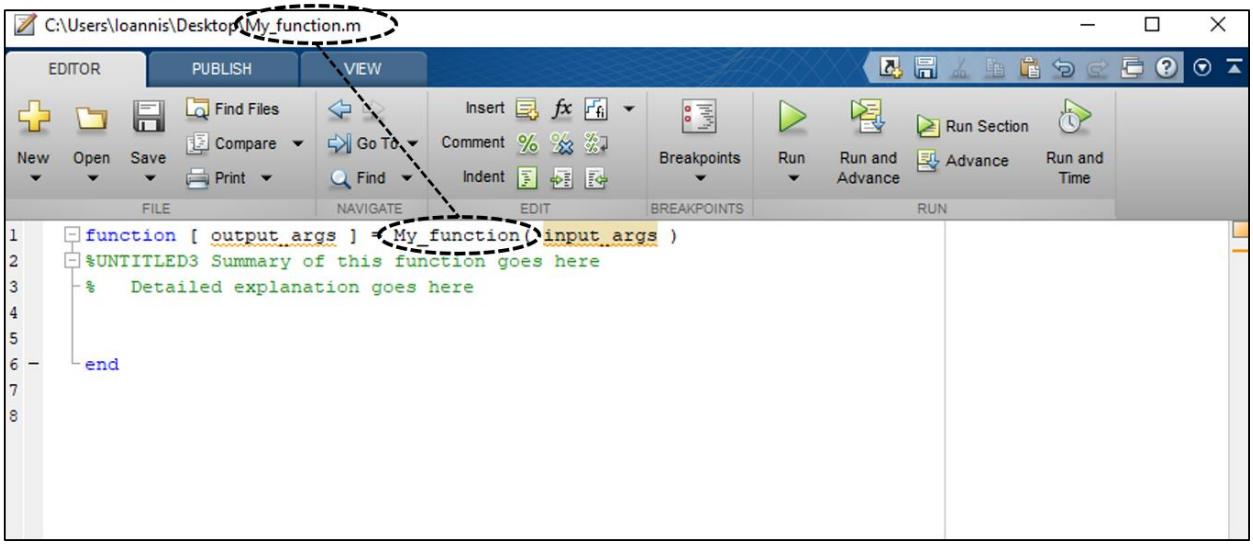
## 1.2 Δημιουργία εκτελέσιμου κώδικα (Script) και συνάρτησης (function)

- ➢ Αρχικά ο χρήστης δημιουργεί έναν φάκελο στο σύστημα αρχείων του υπολογιστή.
- ➢ Έπειτα, μέσω του File Acesss της MATLAB επιλέγει αυτό το φάκελο προκειμένου να αποθηκεύσει τα εκτελέσιμα (*.m) που θα δημιουργήσει.
- ➢ Στην μπάρα εργαλείων της MATLAB επιλέγει NEW → Script για να δημιουργήσει εκτελέσιμο αρχείο script ή NEW → Function για να δημιουργήσει μια συνάρτηση.




Στο εκτελέσιμο αρχείο (.m), ο χρήστη μπορεί να δώσει οποιοδήποτε όνομα (πχ. main.m), ενώ το όνομα της συνάρτησης ως αρχείο (.m) πρέπει να ταυτίζεται με το όνομα που δηλώθηκε στον κώδικα της συνάρτησης.



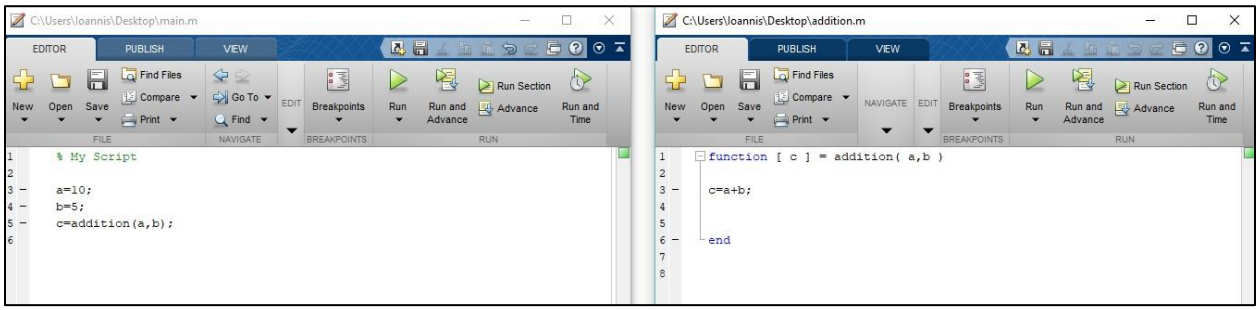




## 1.3 Κλήση συνάρτησης από εκτελέσιμο αρχείο

Σε ένα εκτελέσιμο αρχείο (πχ. main.m) εκτός από το σύνολο των εντολών μπορεί να γίνει και κλήση της συνάρτησης. Η κλήση της συνάρτησης πραγματοποιείται με το όνομα, το σύνολο των ορισμάτων που δέχεται καθώς και με το σύνολό των μεταβλητών που επιστρέφει.

Παράδειγμα:

Θεωρούμε ότι θέλουμε να υλοποιήσουμε την πρόσθεση των αριθμών 10 και 5 με χρήση συνάρτησης. Το εκτελέσιμο αρχείο main.m καθώς και η συνάρτηση addition.m δημιουργούνται ως εξής:



## 1.4 Βασικές πράξεις

Οι βασικές πράξεις [2] του MATLAB χρησιμοποιούν τα σύμβολα που φαίνονται στον πίνακα:

| Σύμβολο | Πράξη |
|---|---|
| + | Πρόσθεση |
| - | Αφαίρεση |
| * | Πολλαπλασιασμός |
| / | Διαίρεση |
| ^ | Ύψωση σε δύναμη |

## 1.5 Πράξεις υλοποιήσιμες από συναρτήσεις βιβλιοθήκης

Οι πράξεις που είναι υλοποιήσιμες από συναρτήσεις βιβλιοθήκης [1] του MATLAB φαίνονται στον πίνακα:

| Συνάρτηση | Ερμηνεία |
|---|---|
| **sin** | ημίτονο |
| **cos** | συνημίτονο |
| **tan** | εφαπτομένη |
| **asin** | τόξο ημιτόνου |
| **acos** | τόξο συνημίτονου |
| **atan** | τόξο εφαπτομένης |
| **exp** | εκθετική συνάρτηση |
| **log** | φυσικός λογάριθμος |
| **log10** | λογάριθμος με βάση το 10 |
| **abs** | απόλυτη τιμή |
| **sqrt** | τετραγωνική ρίζα |
| **mod** | προσημασμένο υπόλοιπο διαίρεσης (modulus) |
| **rem** | υπόλοιπο διαίρεσης |
| **round** | στρογγυλοποίηση στον πλησιέστερο ακέραιο |
| **ceil** | στρογγυλοποίηση στον αμέσως μεγαλύτερο ακέραιο |
| **floor** | στρογγυλοποίηση προς το μείον άπειρο |
| **fix** | στρογγυλοποίηση προς το μηδέν |

## 1.6 Ειδικές σταθερές και μεταβλητές

Οι ειδικές σταθερές και μεταβλητές [2] του MATLAB φαίνονται στον πίνακα:

| Σταθερά ή μεταβλητές | Ερμηνεία |
|---|---|
| **ans** | η πιο πρόσφατη τιμή |
| **eps** | σχετική ακρίβεια κινητής υποδιαστολής |
| **i** | φανταστική μονάδα |
| **inf** | άπειρο |
| **NaN** | μη αριθμός (not a number) |
| **pi** | π |

## 1.7 Η εντολή if

Η εντολή if μας επιτρέπει να ελέγξουμε αν μια (ή περισσότερες) συνθήκες ισχύουν και να εκτελέσουμε σε κάθε περίπτωση την επιθυμητή ακολουθία εντολών και πράξεων [2]. Η εντολή έχει τη γενική μορφή:

```
if relation_1
statement(s)
elseif relation_2
 statement(s)
else
 statement(s)
end
```

Οι συνθήκες ελέγχονται με τη χρήση σχεσιακών και λογικών τελεστών. Σημειώνουμε επίσης ότι η εντολή elseif γράφεται σαν μια λέξη (δεν πρέπει να υπάρχει κενό μεταξύ του else και του if). Η απλούστερη μορφή της εντολής if είναι η πιο κάτω:

```
if relation
 statement(s)
end
```

## 1.8 Η εντολή switch

Η εντολή switch-case μας δίνει τη δυνατότητα να επιλέξουμε για εκτέλεση μια ομάδα εντολών από άλλες πιθανές ομάδες [2]. Η γενική της δομή έχει ως εξής:

```
switch switch_expression
 case value_1
 statement(s)
 case value_2
 statement(s)
 case value_3
 statement(s)
.
.
.
otherwise
 statement(s)
end
```

Η πρώτη γραμμή περιέχει την λέξη κλειδί switch, ακολουθούμενη από το όνομα switch_expression, που θα δώσουμε εμείς, το οποίο μπορεί να είναι βαθμωτή ποσότητα, αλφαριθμητικό, ή ακόμα και μαθηματική παράσταση με προκαθορισμένες μεταβλητές που μπορεί να πάρει μια τιμή.

Μετά από το switch, ακολουθούν οι διάφορες εντολές case. Η κάθε μια έχει ένα όνομα (π.χ. value _1, value_2 κλπ) το οποίο μπορεί να είναι βαθμωτή ποσότητα ή αλφαριθμητικό, και μετά ακολουθούν οι εντολές που θα εκτελεστούν αν βρεθούμε στη συγκεκριμένη περίπτωση.

Μετά την τελευταία περίπτωση/εντολή case, ακολουθεί η προαιρετική περίπτωση/εντολή otherwise της οποίας οι εντολές θα εκτελεστούν αν καμιά από τις προηγούμενες περιπτώσεις δεν ισχύει.

Σε αντίθεση με άλλες γλώσσες προγραμματισμού (όπως, π.χ. η C), στη MATLAB δεν χρειάζεται να διακόψουμε τη ροή της δομής μετά από κάθε case, μια και αυτό θα γίνει αυτόματα αφού μια από τις περιπτώσεις έχει επαληθευτεί.

## 1.9 Βρόχοι for

Οι βρόχοι for έχουν την εξής δομή [2]:

```
for index = initial value (: step) : final value
 statements
end
```

Οι λέξεις "for" και "end" χρησιμοποιούνται στην αρχή και στο τέλος του βρόχου, ο μετρητής index παίρνει τις τιμές από initial value μέχρι final value με βήμα step, και οι εντολές (statements) εκτελούνται για όλες τις τιμές του μετρητή index. Αν παραλείψουμε το βήμα, τότε η MATLAB χρησιμοποιεί το 1 σαν βήμα.

## 1.10 Βρόχοι while

Οι βρόχοι while είναι της μορφής [2]:

```
while relation
 statements
end
```

Οι λέξεις "while" και "end" χρησιμοποιούνται στην αρχή και στο τέλος του βρόχου. Η ακολουθία εντολών «statements» εκτελούνται εφόσον η συνθήκη relation ικανοποιείται (δηλ. είναι αληθής) και σταματούν όταν αυτή παύει να ισχύει

# 2 Συστήματα Γραμμικών Εξισώσεων

## 2.1 Εισαγωγή

Ένα σύστημα γραμμικών εξισώσεων γράφεται σε αναλυτική μορφή ως εξής:

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= b_1, \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &= b_2, \\ \vdots \quad\quad & \vdots \quad\quad\quad\quad \vdots \quad\quad \vdots \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n &= b_n. \end{aligned} \tag{1.1}$$

Όπου, οι πραγματικοί ή μιγαδικοί αριθμοί $a_{ij}, i = 1,...,n,\ j = 1,...,n$ του συστήματος ονομάζονται συντελεστές των αγνώστων ή μεταβλητών και Οι πραγματικοί ή μιγαδικοί αριθμοί $b_i, i = 1,...,n$ ονομάζονται δεύτερα μέλη του συστήματος [3].

Χρησιμοποιώντας μητρώα, το σύστημα γραμμικών εξισώσεων (1.1), διάστασης n γράφεται ως:

$$Ax = b, \tag{1.2}$$

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}, \quad x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \quad b = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}. \tag{1.3}$$

### 2.1.1 Παράδειγμα:

Θεωρούμε το παρακάτω σύστημα γραμμικών εξισώσεων 3 διαστάσεων:

$$\begin{aligned} 1x_1 + 2x_2 + 2x_3 &= 2, \\ 4x_1 + 5x_2 + 6x_3 &= 4, \\ 7x_1 + 8x_2 + 9x_n &= 5 \end{aligned}$$

τότε το γραμμικό σύστημα μπορεί να αναπαρασταθεί στην μορφή $Ax = b$, ως εξής:

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}, \quad x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}, \quad b = \begin{bmatrix} 2 \\ 4 \\ 5 \end{bmatrix}.$$

**Υλοποίηση σε MATLAB:**

```
>> A=[1,2,3;4,5,6;7,8,9];
>> syms x1;
>> syms x3;
>> x=[x1,x2,x3];
>> b=[2,4,5];
```

## 2.2 Ανάστροφο μητρώο

Ανάστροφο μητρώο [3] ενός μητρώου $A=\left(a_{ij}\right)_{i,j}^{m,n}$ ορίζεται το μητρώο $A^T=\left(a_{ji}\right)_{j,i}^{n,m}$, δηλαδή, για ένα πίνακα διάστασης 4,

$$\mathrm{A}=\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \rightarrow \mathrm{A}^{\mathrm{T}}=\begin{bmatrix} a_{11} & a_{21} & a_{31} & a_{41} \\ a_{12} & a_{22} & a_{32} & a_{42} \\ a_{13} & a_{23} & a_{33} & a_{43} \\ a_{14} & a_{24} & a_{34} & a_{44} \end{bmatrix}, \tag{1.4}$$

Τα στοιχεία των γραμμών του $A$ γίνονται στοιχεία των στηλών του $A^T$ ενώ τα στοιχεία των στηλών του $A^T$ γίνονται στοιχεία των γραμμών του $A$.

Σημείωση:

- ✓ Το μητρώο $A$ για το οποίο ισχύει η ισότητα $A=A^T$ ονομάζεται συμμετρικό μητρώο.

### 2.2.1 Παράδειγμα:

Βρίσκουμε το ανάστροφο μητρώο του μητρώο $A$ του προηγούμενου παραδείγματος:

$$A=\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \rightarrow A^{\mathrm{T}}=\begin{bmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{bmatrix}$$

**Υλοποίηση σε MATLAB:**

```
>> A=A';
ή
>> A=transpose(A);
```

## 2.3 Άθροισμα μητρώων

Ως άθροισμα δύο μητρώων $A=\left(a_{ij}\right), B=\left(b_{ij}\right)$ ίδιου μεγέθους $m\times n$ ορίζεται το μητρώο $C=\left(c_{ij}\right)$ μεγέθους $m\times n$ του οποίου τα στοιχεία ορίζονται ως εξής:

$$c_{ij}=a_{ij}+b_{ij}, \quad i=1,...,m,\ j=1,...,n \tag{1.5}$$

### 2.3.1 Παράδειγμα:

Θεωρούμε τα μητρώα $A$ και $B$:

$$A=\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}, B=\begin{bmatrix} 5 & 2 & 2 \\ 4 & 8 & 1 \\ 3 & 5 & 2 \end{bmatrix},$$

τότε,

$$C = A + B = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} + \begin{bmatrix} 5 & 2 & 2 \\ 4 & 8 & 1 \\ 3 & 5 & 2 \end{bmatrix} = \begin{bmatrix} 6 & 4 & 5 \\ 8 & 13 & 7 \\ 10 & 13 & 11 \end{bmatrix}.$$

**Υλοποίηση σε MATLAB:**

```
>> A=[1,2,3;4,5,6;7,8,9];
>> B=[5,2,2;4,8,1;3,5,2];
>> C=A+B;
```

## 2.4 Γινόμενο μητρώων

Αν το πρώτο μητρώο $A$ είναι μεγέθους $m \times p$ και πολλαπλασιαστεί με το δεύτερο μητρώο μεγέθους $p \times n$, το γινόμενο των μητρώων [3] συμβολίζεται ως $C = AB$ και το μητρώο $C$ μεγέθους $m \times n$ με στοιχεία:

$$c_{ij} = g_i(A)\, s_j(B) = \sum_{k=1}^{p} a_{ik} b_{kj},\ i = 1(1)m,\ j = 1,...,n, \tag{1.6}$$

όπου το διάνυσμα $g_i(A)$ συμβολίζει την $i$-οστή γραμμή του μητρώου $A$ ενώ το διάνυσμα $s_j(B)$ συμβολίζει την $j$-οστή στήλη του μητρώου $B$. Επομένως, το στοιχείο $c_{ij}$ του γινομένου προκύπτει αν αθροίσουμε τα επιμέρους γινόμενα της [γραμμής $i$ του $A$] επί τα αντίστοιχα στοιχεία της [στήλης $j$ του $B$].

Παράδειγμα:

Θεωρούμε τα μητρώα $A$ και $B$:

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}, \mathrm{B} = \begin{bmatrix} 1 & 4 \\ 5 & 8 \\ 6 & 3 \end{bmatrix},$$

τότε,

$$C = AB \to C = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \begin{bmatrix} 1 & 4 \\ 5 & 8 \\ 6 & 3 \end{bmatrix} = \begin{bmatrix} (1*1+2*5+3*6) & (1*4+2*8+3*3) \\ (4*1+5*5+6*6) & (4*4+5*8+6*3) \\ (7*7+8*5+9*6) & (7*4+8*8+9*3) \end{bmatrix} = \begin{bmatrix} 29 & 29 \\ 65 & 74 \\ 101 & 119 \end{bmatrix}.$$

**Υλοποίηση σε MATLAB:**

```
>> A=[1,2,3;4,5,6;7,8,9];
>> B=[1,4;5,8;6,3];
>> C=A*B;
```

## 2.5 Τάξη μητρώου

Το πλήθος των γραμμικών ανεξάρτητων διανυσμάτων τα οποία αποτελούν ένα μητρώο $A$ λέγεται τάξη του μητρώου [3] και συμβολίζεται ως $rank(A)$.

Θεωρούμε τα μητρώο $A$:

$$A=\begin{bmatrix}1 & 0 & 3\\ 0 & 1 & 2\\ 0 & 0 & 0\end{bmatrix},$$

τότε η τάξη του μητρώου $A$ είναι 2 εφόσον $[\text{στήλη } 3]=3[\text{στήλη } 1]+2[\text{στήλη } 2]$

**Υλοποίηση σε MATLAB:**

```
>> rank(A);
```

## 2.6 Ορίζουσα μητρώου

Η ορίζουσα [3] ενός τετραγωνικού μητρώου $A$, συμβολίζεται ($\det(A)$) και ορίζεται ως το άθροισμα όλων των προσημασμένων στοιχειωδών γινομένων από το $A$.

Θεωρούμε τα μητρώο $A$:

$$A=\begin{bmatrix}a_{11} & a_{12} & a_{13}\\ a_{21} & a_{22} & a_{23}\\ a_{31} & a_{32} & a_{33}\end{bmatrix},$$

τότε,

$$\det(A)=a_{11}(a_{22}a_{33}-a_{23}a_{32})-a_{12}(a_{21}a_{33}-a_{23}a_{31})+a_{13}(a_{21}a_{32}-a_{22}a_{31}) \tag{1.7}$$

**Υλοποίηση σε MATLAB:**

```
>> det(A);
```

## 2.7 Ιδιοτιμές και ιδιοδιανύσματα μητρώου

Αν ο $A$ είναι ένα $n\times n$ μητρώο, τότε ένα μη μηδενικό διάνυσμα $x$ αποτελεί ιδιοδιάνυσμα [3] του $A$ αν το $Ax$ είναι βαθμωτό πολλαπλάσιο του $x$:

$$Ax=\lambda x\text{, όπου }\lambda\text{ κάποια ιδιοτιμή του } A$$

$$\text{ή } Ax=\lambda x\rightarrow Ax=\lambda \mathrm{I}x\rightarrow(\lambda \mathrm{I}-A)x=0 \tag{1.8}$$

### 2.7.1 Παράδειγμα:

Θεωρούμε τα μητρώο $A$:

$$A=\begin{bmatrix}3 & 0\\ 8 & -1\end{bmatrix}$$

τότε το $x=\begin{bmatrix}1\\2\end{bmatrix}$ είναι ιδιοδιάνυσμα του $A$ εφόσον:

$$Ax=\begin{bmatrix}3 & 0\\8 & -1\end{bmatrix}\begin{bmatrix}1\\2\end{bmatrix}=\begin{bmatrix}3\\6\end{bmatrix}=3x,\ \lambda=3$$

Θεωρούμε τα μητρώο $A$:

$$A=\begin{bmatrix}3 & 2\\-1 & 0\end{bmatrix},$$

τότε,

$$\lambda \mathrm{I}-\mathrm{A}=\lambda\begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}-\begin{bmatrix}3 & 2\\-1 & 0\end{bmatrix}=\begin{bmatrix}\lambda-3 & -2\\1 & \lambda\end{bmatrix},$$

το χαρακτηριστικό πολυώνυμο του $A$ είναι:

$$\det(\lambda \mathrm{I}-\mathrm{A})=\det\left(\begin{bmatrix}\lambda-3 & -2\\1 & \lambda\end{bmatrix}\right)=\lambda^2-3\lambda+2.$$

Συνεπώς οι λύσεις τις εξίσωσης είναι $\lambda_1=1$ και $\lambda_2=2$ αποτελούν και τις ιδιοτιμές του $A$.

**Υλοποίηση σε MATLAB:**

```
>> A=[3,2;-1,0];
>> eig(A);
```

## 2.8 Αντίστροφο μητρώο

Για ένα τετραγωνικό μητρώο $A$ όπου $\det(A)\neq 0$ υπάρχει το $A^{-1}$, τέτοιο ώστε $\mathrm{A}^{-1}\mathrm{A}=\mathrm{I}$. Το $A^{-1}$ ονομάζεται αντίστροφο μητρώο [3] του $A$. Το μητρώο $\mathrm{I}$ ονομάζεται μοναδιαίο μητρώο και έχει την τιμή 1 στα στοιχεία τις κύρια διαγώνιου του και σε όλα τα υπόλοιπα μηδενικές τιμές.

Σημείωση:

- ✓ Για διαγώνιο μητρώο ο αντίστροφος ορίζεται το διαγώνιο μητρώο με αντίστροφα τα στοιχεία της διαγώνιου.

$$A=\begin{bmatrix}2 & 0 & 0\\0 & 3 & 0\\0 & 0 & 4\end{bmatrix}\rightarrow A^{-1}=\begin{bmatrix}1/2 & 0 & 0\\0 & 1/3 & 0\\0 & 0 & 1/4\end{bmatrix}.$$

**Υλοποίηση σε MATLAB:**

```
>> inv(A);
```

## 2.9 Επίλυση συστήματος γραμμικών εξισώσεων

Δεδομένου του γραμμικού συστήματος της μορφής $Ax = b,$ μπορούμε να ορίσουμε τη λύση του ως εξής [3]:

$$Ax = b \xrightarrow{A^{-1}} A^{-1}Ax = A^{-1}b \xrightarrow{A^{-1}A=I} x = A^{-1}b. \tag{1.9}$$

**Υλοποίηση σε MATLAB:**

```
>> x = A\B;
ή
>> x = mldivide(A,B)
```

## 2.10 Απαλοιφή Gauss

Δεδομένου του προβλήματος $Ax = b$, η απαλοιφή Gauss είναι μια πεπερασμένη επαναληπτική διαδικασία απλοποίησης που επιδρά πάνω στο γραμμικό σύστημα προκειμένου να το φέρει σε μια κατάσταση από την οποία είναι αρκετά «εύκολο» να βρεθεί το διάνυσμα της λύσης $x$ [4].

Η μέθοδος απαλοιφής του Gauss βασίζεται στις εξής ιδιότητες των γραμμικών συστημάτων:

- ✓ αν αντιμετατεθούν δύο οποιεσδήποτε εξισώσεις του συστήματος
- ✓ αν πολλαπλασιαστεί μία εξίσωση του συστήματος επί έναν αριθμό $c \in \mathbb{R}$ τέτοιον ώστε $c \neq 0,$
- ✓ αν αντικατασταθεί μία εξίσωση του συστήματος με το άθροισμα αυτής και μιας άλλης πολλαπλασιασμένης επί έναν αριθμό $c \in \mathbb{R}$ τέτοιον ώστε $c \neq 0,$

τότε προκύπτει ένα σύστημα:

$$A'x = b', \tag{1.10}$$

το οποίο είναι ισοδύναμο με το αρχικό σύστημα, με την έννοια ότι έχουν την ίδια λύση, αλλά η εξαγωγή της λύσης $x$ είναι προφανής σε σχέση με την αρχική κατάσταση του γραμμικού συστήματος $Ax = b$.

Για το γραμμικό σύστημα $Ax = b$ ισχύει:

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}, \quad x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \quad b = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}, \tag{1.11}$$

η απαλοιφή Gauss φέρει το γραμμικό σύστημα στην απλοποιημένη μορφή $A'x = b'$ :

$$A' = \begin{bmatrix} a'_{11} & a'_{12} & \cdots & a'_{1n} \\ 0 & a'_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & a'_{nn} \end{bmatrix}, \quad x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \quad b' = \begin{bmatrix} b'_1 \\ b'_2 \\ \vdots \\ b'_n \end{bmatrix}, \tag{1.12}$$

Όπου ο $A'$ είναι άνω τριγωνικό (όλα τα στοιχεία κάτω από την κύρια διαγώνιο είναι μηδέν).

**Αλγόριθμος απαλοιφής Gauss:**

Ο αλγόριθμος δέχεται σαν είσοδο το μητρώο $A$ μεγέθους $(n\times n)$, καθώς και το διάνυσμα στήλης $b$ μεγέθους $(n\times 1)$.

1. Ορίζουμε το επαυξημένο μητρώο $n\times(m+1)$ ως εξής:

$$\tilde{A}=[A\,|\,b]=\left[\begin{array}{cccc|c} a_{11} & a_{12} & \cdots & a_{1n} & b_1 \\ a_{21} & a_{22} & \cdots & a_{2n} & b_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} & b_n \end{array}\right],$$

2. Για $i=1$ έως $n-1$ επαναλήψεις πραγματοποιείται:
   - ✓ Οδήγηση, κατά την οποία επιλέγεται ως στοιχείο «οδηγός» το μέγιστο στοιχείο της $i$ - οστής στήλης από την $i$ έως $n$ γραμμή του $\tilde{A}$. Το μητρώο πλέον αναδιατάσσεται με $i$ - οστή γραμμή τη γραμμή που περιέχει τον οδηγό $a'_{ii}$.
   - ✓ Κάθε στοιχείο των γραμμών από $k=i+1$ έως $n$ του τροποποιημένου μητρώου $A$ (εξαιρείται αυτή με τον οδηγό) γίνεται:

$$a_{kj}=a_{kj}-\frac{a_{k1}}{a_{ii}}a_{kj}.$$

   - ✓ Αυτή η διαδικασία επαναλαμβάνεται έως ότου το μητρώο $A$ του συστήματος να μετασχηματιστεί σε ένα άνω τριγωνικό μητρώο $A'$.

### 2.10.1 Παράδειγμα:

Θεωρούμε το γραμμικό σύστημα $Ax=b$:

$$A=\begin{bmatrix}2&2&2\\3&4&5\\4&6&7\end{bmatrix},\mathrm{x}=\begin{bmatrix}x_1\\x_2\\x_3\end{bmatrix},b=\begin{bmatrix}1\\2\\3\end{bmatrix}$$

**Λύση:**

Για $i=1$

$$\tilde{A}=[A\,|\,b]=\left[\begin{array}{ccc|c}2&2&2&1\\3&4&5&2\\4&6&7&3\end{array}\right]\xrightarrow{\text{οδήγηση}}\left[\begin{array}{ccc|c}\underline{4}&6&7&3\\3&4&5&2\\2&2&2&1\end{array}\right]$$

Για $k=i+1=2$

$$\left[\begin{array}{ccc|c}4&6&7&3\\0&-1/2&-1/4&5/4\\2&2&2&1\end{array}\right]$$

Για $k=i+2=3$

$$\left[\begin{array}{ccc|c}4&6&7&3\\0&-1/2&-1/4&-1/4\\0&-1&-3/2&-1/2\end{array}\right]$$

Για $i=2$

$$\left[\begin{array}{ccc|c} 4 & 6 & 7 & 3 \\ 0 & -1/2 & -1/4 & 1/4 \\ 0 & -1 & -3/2 & -1/2 \end{array}\right] \xrightarrow{\text{οδήγηση}} \left[\begin{array}{ccc|c} 4 & 6 & 7 & 3 \\ 0 & \underline{-1} & -3/2 & -1/2 \\ 0 & -1/2 & -3/2 & 1/4 \end{array}\right]$$

Για $k=i+1=2$

$$\left[\begin{array}{ccc|c} 4 & 6 & 7 & 3 \\ 0 & -1 & -3/2 & -1/2 \\ 0 & 0 & 1/2 & 0 \end{array}\right]$$

Συνεπώς το γραμμικό σύστημα έχει έρθει στη μορφή $A'x=b'$:

$$A=\begin{bmatrix} 4 & 6 & 7 \\ 0 & -1 & -3/2 \\ 0 & 0 & 1/2 \end{bmatrix}, \mathrm{x}=\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}, b=\begin{bmatrix} 3 \\ -1/2 \\ 0 \end{bmatrix}$$

Πίσω αντικατάσταση:

Η λύση του παραπάνω γραμμικού συστήματος προκύπτει ως εξής:

$$\begin{pmatrix} 4x_1+6x_2+7x_3=3, \\ -x_2-\frac{3}{2}x_3=-\frac{1}{2}, \\ \frac{1}{2}x_3=0 \end{pmatrix} \xrightarrow{x_3=0} \begin{pmatrix} 4x_1+6x_2+7x_3=3, \\ -x_2=-\frac{1}{2} \end{pmatrix} \xrightarrow[x_2=\frac{1}{2}]{x_3=0} \left(4x_1-3=3,\right) \rightarrow x_1=0$$

Συνεπώς η λύση του συστήματος είναι:

$$x=\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}=\begin{bmatrix} 0 \\ 1/2 \\ 0 \end{bmatrix}$$

**Υλοποίηση σε MATLAB:**

**main.m**

```
A = [2 2 2;3 4 5;4 6 7];
b = [1;2;3];
x = gausselim(A,b);
```

**gausselim.m**

```
function [x, err] = gausselim(A, b)
% Gaussian Elimination routine

err = 0;
```

```
% check that input is legal
[n,m] = size(A);
if n ~= m
  disp('Error in gauselim: Matrix must be square.')
  err = 2; return
end

%set up space for solution vector
x = zeros(size(b));

% form augmented matrix by adding b to last column of A
m = n+1;
A(:,m) = b;

% FORWARD ELIMINATION
for k = 1:n-1
  [A, err] = gepivot(A, k);
  for i = k+1:n
    mm = A(i,k)/A(k,k);
    for j = k:m
       A(i,j) = A(i,j) - mm*A(k,j);
    end

%     A(i,k:m) = A(i,k:m) - mm*A(k,k:m);
  end

  A

end

% BACK SUBSTITUTION
if A(n,n) == 0
     err = 1; disp('Matrix is singular'); return
end

x(n) = A(n,m)/A(n,n);
for i = n-1:-1:1
   sum = 0;
   for j = i+1:n
      sum = sum + A(i,j)*x(j);
   end
   x(i) = (A(i,m) - sum) / A(i,i);
end
```

**gepivot.m**

```
function [A, err] = gepivot(A, k)

    err = 0;

    [piv,piv_index]=max(abs(A(k:end,k)));

    if piv == 0              % trouble - can't avoid 0 pivot
```

```
        err = 1;
    elseif piv_index+k-1 ~= k      % best pivot elsewhere, swap rows
      t=A(piv_index+k-1,:);
      A(piv_index+k-1,:)=A(k,:);
      A(k,:)=t;
    end

end
```

## 2.11 Απαλοιφή Gauss-Jordan

Η μέθοδος απαλοιφής των Gauss – Jordan [5] αποτελεί μία απλοποίηση της κλασικής μεθόδου απαλοιφής του Gauss. Η μέθοδος αυτή συνίσταται σε μία συστηματική εφαρμογή των μετασχηματισμών γραμμών, όπως και στην απαλοιφή Gauss, έτσι ώστε το μητρώο των συντελεστών των αγνώστων του τελικού μετασχηματιζόμενου συστήματος να είναι διαγώνιο, αντί να είναι άνω τριγωνικό.

Για το γραμμικό σύστημα $Ax = b$ ισχύει:

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}, \quad x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \quad b = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}, \tag{1.13}$$

η απαλοιφή Gauss – Jordan φέρει το γραμμικό σύστημα στην απλοποιημένη μορφή $A'x = b'$ :

$$A' = \begin{bmatrix} a'_{11} & 0 & \cdots & 0 \\ 0 & a'_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & a'_{nn} \end{bmatrix}, \quad x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \quad b' = \begin{bmatrix} b'_1 \\ b'_2 \\ \vdots \\ b'_n \end{bmatrix}. \tag{1.14}$$

Όπου ο $A'$ είναι διαγώνιο (όλα τα στοιχεία πάνω και κάτω από την κύρια διαγώνιο είναι μηδέν).

**Αλγόριθμος απαλοιφής Gauss-Jordan:**

---

Ο αλγόριθμος δέχεται σας είσοδο το μητρώο $A$ μεγέθους $(n\times n)$, καθώς και το διάνυσμα στήλης $b$ μεγέθους $(n\times 1)$.

1. Ορίζουμε το επαυξημένο μητρώο $n\times(m+1)$ ως εξής:

$$\tilde{A}=[A\,|\,b]=\left[\begin{array}{cccc|c} a_{11} & a_{12} & \cdots & a_{1n} & b_1 \\ a_{21} & a_{22} & \cdots & a_{2n} & b_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} & b_n \end{array}\right],$$

2. Για $i=1,...,n$ πραγματοποιείται:
   - ✓ Οδήγηση, κατά την οποία επιλέγεται ως στοιχείο «οδηγός» το μέγιστο στοιχείο της $i$ - οστής στήλης από την $i$ έως τη $n$ γραμμή του $\tilde{A}$. Το μητρώο πλέον αναδιατάσσεται με $i$ - οστή γραμμή τη γραμμή που περιέχει τον οδηγό $a'_{ii}$.
   - ✓ Κάθε στοιχείο των όλων γραμμών του μητρώου $\tilde{A}$, $(k=1,...,n)$ εκτός από αυτά της $i$ - οστής γραμμής $(k\neq i)$, δηλαδή εξαιρείται από τη διαδικασία η γραμμή που περιέχει τον οδηγό, τροποποιείται ως εξής:

$$a_{kj}=a_{kj}-\frac{a_{k1}}{a_{ki}}a_{kj}.$$

   - ✓ Αυτή η διαδικασία επαναλαμβάνεται έως ότου το μητρώο $A$ του συστήματος να μετασχηματιστεί σε ένα διαγώνιο μητρώο $A'$.

---

Σημείωση:

- ✓ Η διαφορά της μεθόδου απαλοιφής Gauss – Jordan από τη απλή μέθοδο τη απαλοιφής Gauss έγκειται στο γεγονός ότι κατά το $i$ - οστό βήμα της διαδικασίας ο άγνωστος $x_i$ απαλείφεται από τις τελευταίες $(n-i)$ καθώς και από τις $(i-1)$ πρώτες εξισώσεις.

### 2.11.1 Παράδειγμα:

Θεωρούμε το γραμμικό σύστημα $Ax=b$:

$$A=\begin{bmatrix} 2 & 2 & 2 \\ 3 & 4 & 5 \\ 4 & 6 & 7 \end{bmatrix}, \mathrm{x}=\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}, b=\begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$$

**Λύση:**

Για $i=1$

$$\tilde{A}=[A\,|\,b]=\left[\begin{array}{ccc|c} 2 & 2 & 2 & 1 \\ 3 & 4 & 5 & 2 \\ 4 & 6 & 7 & 3 \end{array}\right] \xrightarrow{\text{οδήγηση}} \left[\begin{array}{ccc|c} \underline{4} & 6 & 7 & 3 \\ 3 & 4 & 5 & 2 \\ 2 & 2 & 2 & 1 \end{array}\right]$$

Για $k=1$ Για $k=2$ Για $k=3$

$k=i$

$$\left[\begin{array}{ccc|c} 4 & 6 & 7 & 3 \\ 0 & -1/2 & -1/4 & -1/4 \\ 2 & 2 & 2 & 1 \end{array}\right] \qquad \left[\begin{array}{ccc|c} 4 & 6 & 7 & 3 \\ 0 & -1/2 & -1/4 & -1/4 \\ 0 & -1 & -3/2 & -1/2 \end{array}\right]$$

Για $i=2$

$$\tilde{A}=[A\,|\,b]=\left[\begin{array}{ccc|c} 4 & 6 & 7 & 3 \\ 0 & -1/2 & -1/4 & -1/4 \\ 0 & -1 & -3/2 & -1/2 \end{array}\right] \xrightarrow{\text{οδήγηση}} \left[\begin{array}{ccc|c} 4 & 6 & 7 & 3 \\ 0 & -1 & -3/2 & -1/2 \\ 0 & -1/2 & -1/4 & -1/4 \end{array}\right]$$

Για $k=1$ Για $k=2$ Για $k=3$

$$\left[\begin{array}{ccc|c} 4 & 0 & -2 & 0 \\ 0 & -1 & -3/2 & -1/2 \\ 0 & -1/2 & -1/4 & -1/4 \end{array}\right] \qquad k=i \qquad \left[\begin{array}{ccc|c} 4 & 0 & -2 & 0 \\ 0 & -1 & -3/2 & -1/2 \\ 0 & 0 & 1/2 & 0 \end{array}\right]$$

Για $i=3$

$$\tilde{A}=[A\,|\,b]=\left[\begin{array}{ccc|c} 4 & 0 & -2 & 0 \\ 0 & -1 & -3/2 & -1/2 \\ 0 & 0 & 1/2 & 0 \end{array}\right] \xrightarrow{\text{οδήγηση}} \left[\begin{array}{ccc|c} 4 & 0 & -2 & 0 \\ 0 & -1 & -3/2 & -1/2 \\ 0 & 0 & 1/2 & 0 \end{array}\right]$$

Για $k=1$ Για $k=2$ Για $k=3$

$$\left[\begin{array}{ccc|c} 4 & 0 & 0 & 0 \\ 0 & -1 & -3/2 & -1/2 \\ 0 & 0 & -1/2 & 0 \end{array}\right] \qquad \left[\begin{array}{ccc|c} 4 & 0 & 0 & 0 \\ 0 & -1 & 0 & -1/2 \\ 0 & 0 & -1/2 & 0 \end{array}\right] \qquad k=i$$

Εφαρμόζουμε την πίσω αντικατάσταση:

Τότε η λύση του παραπάνω γραμμικού συστήματος προκύπτει ως εξής:

$$\begin{pmatrix} 4x_1 = 0, \\ -1x_2 = -\dfrac{1}{2}, \\ \dfrac{1}{2}x_3 = 0 \end{pmatrix}$$

Συνεπώς η λύση του συστήματος είναι:

$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 1/2 \\ 0 \end{bmatrix}$$

**Υλοποίηση σε MATLAB:**

**main.m**

```
A = [2 2 2;3 4 5;4 6 7];
b = [1;2;3];
x = gaussjord(A, b);
```

**gaussjord.m**

```
function [x, err] = gaussjord(A, b)
% Gauss - Jordan Routine for solution of single system

err = 0;

%set up space for solution vector
x = zeros(size(b));

% check that input is legal
[n,m] = size(A);
if n ~= m
  disp('Error in gauselim: Matrix must be square.')
  err = 2; return
end

% form augmented matrix by adding b to last column of A
m = n+1; A(:,m) = b;

% ELIMINATION

for k = 1:n
  if k < n
     [A, err] = gepivot(A, k);    % partial pivoting
  end

  if err ~= 0
     disp('Matrix is singular'); return
  end

  for i = 1:n
    if i ~= k
       mm = A(i,k)/A(k,k);
       for j = k:m
          A(i,j) = A(i,j) - mm*A(k,j);
```

```
        end
    end
  end
end

x = A(:,m)./diag(A);
```

## 2.12 Μέθοδος του Jacobi και μέθοδος των Gauss - Seidel

Δοθέντος του προβλήματος $Ax = b$ οι δύο επαναληπτικές μέθοδοι συνίστανται στη διάσπαση του μητρώου των συντελεστών των αγνώστων $A$ ως εξής:

$$A = D - L - U, \tag{1.15}$$

όπου το $D$ είναι το διαγώνιο μητρώο με διαγώνια στοιχεία τα αντίστοιχα του $A$, το $L$ είναι κάτω τριγωνικό και το $U$ είναι άνω τριγωνικό:

$$D = \begin{bmatrix} a_{11} & 0 & \cdots & 0 \\ 0 & a_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \cdots & a_{nn} \end{bmatrix}, \quad L = \begin{bmatrix} 0 & & & \\ -a_{21} & 0 & & \\ \vdots & \vdots & \ddots & \\ -a_{n1} & -a_{n2} & \cdots & 0 \end{bmatrix}, \quad U = \begin{bmatrix} 0 & -a_{12} & \cdots & -a_{1n} \\ & 0 & \cdots & -a_{2n} \\ & & \ddots & \vdots \\ & & & 0 \end{bmatrix}. \tag{1.16}$$

Ικανή συνθήκη για τη σύγκλιση την μεθόδων του Jacobi και των Gauss – Seidel [6] είναι το μητρώο $A$ να έχει αυστηρά διαγώνια κυριαρχία κατά γραμμές ή κατά στήλες:

$$|a_{ii}| > \sum_{\substack{j=1 \\ j \neq i}}^{n} |a_{ij}|, \quad i = 1,...,n, \quad \text{ή} \quad |a_{ii}| > \sum_{\substack{j=1 \\ j \neq i}}^{n} |a_{ji}|, \quad i = 1,...,n. \tag{1.16}$$

## 2.13 Μέθοδος του Jacobi

Για να μπορέσει η μέθοδος του Jacobi [6] να οριστεί θα πρέπει το μητρώο $D$ να είναι αντιστρέψιμο, δηλαδή:

$$\det(D) = a_{11}a_{22}...a_{nn} \neq 0. \tag{1.17}$$

Τότε αν υπάρχει το $D^{-1}$, το επαναληπτικό σχήμα της μεθόδου Jacobi είναι:

$$x^{(k+1)} = D^{-1}\left(L + U\right)x^{(k)} + D^{-1}b, \quad k = 0,1,2,\ldots, \tag{1.18}$$

για οποιοδήποτε αρχική εκτίμηση $x^{(0)}$.

**Αλγόριθμος του Jacobi:**

---

Ο αλγόριθμος δέχεται σαν είσοδο τετραγωνικό μητρώο $A$ μεγέθους $(n\times n)$, το διάνυσμα στήλης $b$ μεγέθους $(n\times 1)$ και την αρχική εκτίμηση της λύσης $x = x^{(0)}$.

1. Υπολογίζονται τα μητρώα $D$, $L$, $U$ και $\mathrm{T} = D^{-1}(L+U)$.
2. Εάν ικανοποιείται το κριτήριο $\|\mathrm{T}\| = \|D^{-1}(L+U)\| < 1 \xrightarrow{\text{αρκεί}} |\lambda_{\max}| < 1$ τότε ο αλγόριθμος συγκλίνει και συνεχίζει:
3. Οι συνιστώσες $x_i^{(k)}$, $i = 1$ έως $n$ του διανύσματος της $k$-οστής επανάληψης υπολογίζονται ως εξής:

$$x_i^{(k+1)} = \frac{1}{a_{ii}}\left[ b_i - \sum_{\substack{j=1 \\ j\neq i}}^{n} a_{ij} x_j^{(k)} \right], \quad i = 1...n, \quad k = 0,1,2,\ldots$$

4. Ο αλγόριθμος επαναλαμβάνεται έως ότου η $x^{(k+1)}$ προσέγγιση της λύσης δεν διαφέρει από την $x^{(k)}$ στα $m$ σημαντικά ψηφία. Δηλαδή όταν ικανοποιείται η εξής σχέση:

$$\frac{\| x^{(k+1)} - x^{(k)} \|}{\| x^{(k+1)} \|} \leq \frac{1}{2} 10^{-m}$$

Όπου $m$ είναι ο αριθμός των σημαντικών ψηφίων με βάση τα οποία ο χρήστης επιθυμεί την ακρίβεια της λύσης.

---

### 2.13.1 Παράδειγμα:

Θεωρούμε το γραμμικό σύστημα $Ax^{(0)} = b$:

$$A = \begin{bmatrix} 1 & 2 & -2 \\ 1 & 1 & 1 \\ 2 & 2 & 1 \end{bmatrix}, \mathrm{x}^{(0)} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, b = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

Χρησιμοποιώντας τη μέθοδο του Jacobi θέλουμε να βρούμε την προσεγγιστική λύση για ακρίβεια 3 σημαντικών ψηφίων.

**Λύση:**

Υπολογίζουμε τα μητρώα $D$, $L$, $U$ και $\mathrm{T} = D^{-1}(L+U)$:

$$D = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \mathrm{L} = \begin{bmatrix} 0 & 0 & 0 \\ -1 & 0 & 0 \\ -2 & -2 & 0 \end{bmatrix}, \quad \mathrm{U} = \begin{bmatrix} 0 & -2 & 2 \\ 0 & 0 & -1 \\ 0 & 0 & 0 \end{bmatrix}, \quad T = \begin{bmatrix} 0 & -2 & 2 \\ -1 & 0 & -1 \\ -2 & -2 & 0 \end{bmatrix}$$

Εξετάζουμε εάν το μητρώο $\mathrm{T}$ είναι αντιστρέψιμο. Αρκεί να αποδείξουμε ότι η μεγαλύτερη ιδιοτιμή είναι μικρότερη από 1 ($|\lambda_{\max}| < 1$).

Εύρεση ιδιοτιμών του $\mathrm{T}$:

$$\det(T - \mathrm{I}\lambda) = 0,$$

δηλαδή:

$$\det\left(\begin{bmatrix} 0-\lambda & -2 & 2 \\ -1 & 0-\lambda & -1 \\ -2 & -2 & 0-\lambda \end{bmatrix}\right)=0,$$

άρα το χαρακτηριστικό πολυώνυμο που προκύπτει είναι:

$$-\lambda(\lambda^2-2)+(2\lambda+4)-2(2+2\lambda)=0 \rightarrow -\lambda^3=0,$$

τότε

$$\lambda_1=0, \quad \lambda_2=0, \quad \lambda_3=0 \quad \rightarrow \quad |\lambda_{\max}|=0<1,$$

Άρα η μέθοδος Jacobi συγκλίνει.

Για $k=1 \rightarrow x_i^{(1)}=\frac{1}{a_{ii}}\left[b_i-\sum_{\substack{j=1\\ j\neq i}}^{n} a_{ij}x_j^{(0)}\right]$, δηλαδή:

$$\begin{aligned} x_1^{(1)} &= 1-2x_2^{(0)}+2x_3^{(0)}, & x_1^{(1)} &= 1-2\cdot 1+2\cdot 1, & x_1^{(1)} &= 1, \\ x_2^{(1)} &= 1-x_1^{(0)}-x_3^{(0)}, & \rightarrow\ x_2^{(1)} &= 1-1-1, & \rightarrow\ x_2^{(1)} &= -1, \\ x_3^{(1)} &= 1-2x_1^{(0)}-2x_2^{(0)} & x_3^{(1)} &= 1-2\cdot 1-2\cdot 1 & x_3^{(1)} &= -3 \end{aligned}$$

$\frac{\| x^{(1)}-x^{(0)} \|}{\| x^{(1)} \|}=\frac{\sqrt{(1-1)^2+(-1-1)^2+(-3-1)^2}}{\sqrt{1^2+(-1)^2+(-3)^2}}=1.35 \rightarrow 1.35>\frac{1}{2}10^{-3}$, άρα ο αλγόριθμος συνεχίζει.

Για $k=2 \rightarrow x_i^{(2)}=\frac{1}{a_{ii}}\left[b_i-\sum_{\substack{j=1\\ j\neq i}}^{n} a_{ij}x_j^{(1)}\right]$, δηλαδή:

$$\begin{aligned} x_1^{(2)} &= 1-2x_2^{(1)}+2x_3^{(1)}, & x_1^{(1)} &= 1-2\cdot(-1)+2\cdot(-3), & x_1^{(2)} &= -3, \\ x_2^{(2)} &= 1-x_1^{(1)}-x_3^{(1)}, & \rightarrow\ x_2^{(2)} &= 1-1-(-3), & \rightarrow\ x_2^{(2)} &= 3, \\ x_3^{(2)} &= 1-2x_1^{(1)}-2x_2^{(1)} & x_3^{(2)} &= 1-2\cdot 1-2\cdot(-1) & x_3^{(2)} &= 1 \end{aligned}$$

$\frac{\| x^{(2)}-x^{(1)} \|}{\| x^{(2)} \|}=\frac{\sqrt{(-3-1)^2+(3-(-1))^2+(1-(-3))^2}}{\sqrt{(-3)^2+3^2+1^2}}=1.59 \rightarrow 1.59>\frac{1}{2}10^{-3}$, άρα ο αλγόριθμος συνεχίζει.

Για $k=3 \rightarrow x_i^{(2)}=\frac{1}{a_{ii}}\left[b_i-\sum_{\substack{j=1\\ j\neq i}}^{n} a_{ij}x_j^{(1)}\right]$, δηλαδή:

$$x_1^{(3)} = 1 - 2x_2^{(2)} + 2x_3^{(2)}, \quad x_1^{(3)} = 1 - 2\cdot 3 + 2\cdot 1, \quad x_1^{(3)} = -3,$$
$$x_2^{(3)} = 1 - x_1^{(2)} - x_3^{(2)}, \quad \rightarrow x_2^{(3)} = 1 - (-3) - 1, \quad \rightarrow x_2^{(3)} = 3,$$
$$x_3^{(3)} = 1 - 2x_1^{(2)} - 2x_2^{(2)} \quad x_3^{(3)} = 1 - 2\cdot(-3) - 2\cdot 3 \quad x_3^{(3)} = 1$$

$$\frac{\| x^{(3)} - x^{(2)} \|}{\| x^{(3)} \|} = \frac{\sqrt{(-3-(-3))^2 + (3-3)^2 + (1-1)^2}}{\sqrt{(-3)^2 + 3^2 + 1^2}} = 0 \rightarrow 0 < \frac{1}{2}10^{-3}$$, άρα ο αλγόριθμος τερματίζει.

Συνεπώς προσεγγιστική λύση του δοθέντος γραμμικού συστήματος με ακρίβεια 3 σημαντικών σημείων είναι:

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} -3 \\ 3 \\ 1 \end{bmatrix}$$

**Υλοποίηση σε MATLAB:**

**main.m**

```
A = [1 2 -2;1 1 1;2 2 1];
b = [1;1;1];
x = [1;1;1];
x = Jacobi(A, b, x, 3);
```

**Jacobi.m**

```
function x = Jacobi(A, b, x, m)

D = diag(diag(A))
L = diag(diag(A))-tril(A)
U = diag(diag(A))-triu(A)

d = diag(A);

inv(D)

T = inv(D)*(L+U)

sr = max(abs(eig(T)))

disp('Jacobi method:');
if (sr < 1)
    iter = 0;

    while (true)
        iter = iter + 1;
        fprintf('Iteration: %d',iter)
        prevx = x;

        %x = inv(D)*(L+U)*x + inv(D)*b
```

```
        for i = 1:length(A)
            sum = 0;
            for j = 1:length(A)
              if (j ~= i), sum = sum + A(i,j)*prevx(j); end
            end
            x(i) = (1/A(i,i))*[b(i) - sum];
        end
        x

        if (norm(prevx-x)/norm(x))< 0.5*10^(-m), return; end
    end
else
    disp('ERROR: The Jacobi method is diverging');
    return;
end
```

## 2.14 Μέθοδος των Gauss - Seidel

Για να μπορέσει η μέθοδος των Gauss-Seidel [6] να οριστεί θα πρέπει το μητρώο $D$ να είναι αντιστρέψιμο, δηλαδή:

$$\det(D-L)\neq 0 \tag{1.20}$$

Τότε αν υπάρχει το $(D-L)^{-1}$, το επαναληπτικό σχήμα της μεθόδου Jacobi είναι:

$$x^{(k+1)}=(D-L)^{-1}Ux^{(k)}+(D-L)^{-1}b,\quad k=0,1,2,\ldots, \tag{1.21}$$

για οποιοδήποτε αρχική εκτίμηση της λύσης $x^{(0)}$.

**Αλγόριθμος Gauss - Seidel:**

---

Ο αλγόριθμος δέχεται ως είσοδο το γραμμικού συστήματος $Ax=b$, όπου $A$ τετραγωνικό μητρώο μεγέθους $(n\times n)$, το διάνυσμα στήλης $b$ μεγέθους $(n\times 1)$ και $x=x^{(0)}$ την αρχική εκτίμηση της λύσης.

1. Υπολογίζονται τα μητρώα $D$, $L$, $U$ και $\mathrm{T}=(D-L)^{-1}U$
2. Εάν ικανοποιείται το κριτήριο $\|\mathrm{T}\|=\|D^{-1}(L+U)\|<1 \xrightarrow{\text{αρκεί}} |\lambda_{\max}|<1$ τότε ο αλγόριθμος συγκλίνει και συνεχίζεται ως εξής:
3. Οι συνιστώσες $x_i^{(k)}$, $i=1,\ldots,n$ του διανύσματος της $k$ -οστής επανάληψης υπολογίζονται ως εξής:

   $$x^{(k+1)}=\left(D-L\right)^{-1}Ux^{(k)}+\left(D-L\right)^{-1}b,\quad k=0,1,2,\ldots,$$

   ή πιο αναλυτικά:

   $$x_i^{(k+1)}=\frac{1}{a_{ii}}\left[b_i-\sum_{j=1}^{i-1}a_{ij}x_j^{(k+1)}-\sum_{j=i+1}^{n}a_{ij}x_j^{(k)}\right],\quad i=1,\ldots,n,\quad k=0,1,2,\ldots$$

4. Ο αλγόριθμος επαναλαμβάνεται έως ότου η $x^{(k+1)}$ προσέγγιση της λύσης δεν διαφέρει από την $x^{(k)}$ στα $m$ σημαντικά ψηφία. Δηλαδή όταν ικανοποιείται η εξής σχέση:

---

$$\frac{\| x^{(k+1)} - x^{(k)} \|}{\| x^{(k+1)} \|} \leq \frac{1}{2} 10^{-m}$$

Όπου $m$ είναι ο αριθμός των σημαντικών ψηφίων με βάση τα οποία ο χρήστης επιθυμεί την ακρίβεια της λύσης.

---

### 2.14.1 Παράδειγμα:

Θεωρούμε το γραμμικό σύστημα $Ax^{(0)} = b$:

$$A = \begin{bmatrix} 3 & 2 & 0 \\ 0 & 2 & 1 \\ 1 & 0 & 2 \end{bmatrix}, \quad \mathrm{x}^{(0)} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \quad b = \begin{bmatrix} 5 \\ 3 \\ 3 \end{bmatrix}$$

Χρησιμοποιώντας τη μέθοδο των Gauss-Seidel θέλουμε να βρούμε την προσεγγιστική λύση για ακρίβεια $m = 2$ σημαντικά ψηφία.

**Λύση:**

Υπολογίζουμε τα μητρώα $D$, $L$, $U$ και $\mathrm{T} = (D - L)^{-1} U$:

$$D = \begin{bmatrix} 3 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{bmatrix}, \quad \mathrm{L} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix}, \quad \mathrm{U} = \begin{bmatrix} 0 & -2 & 0 \\ 0 & 0 & -1 \\ 0 & 0 & 0 \end{bmatrix}, \quad T = \begin{bmatrix} 0 & -0.67 & 0 \\ 0 & 0 & -0.5 \\ 0 & 0.33 & 0 \end{bmatrix},$$

Εξετάζουμε εάν το μητρώο $\mathrm{T}$ είναι αντιστρέψιμο. Αρκεί να αποδείξουμε ότι η μεγαλύτερη ιδιοτιμή είναι μικρότερη από 1 ($| \lambda_{\max} | < 1$).

Εύρεση ιδιοτιμών του $\mathrm{T}$:

$$\det(T - \mathrm{I}\lambda) = 0,$$

δηλαδή,

$$\det\left( \begin{bmatrix} 0-\lambda & -0.67 & 0 \\ 0 & 0-\lambda & -0.5 \\ 0 & 0.33 & 0-\lambda \end{bmatrix} \right) = 0.$$

Από την επίλυση του χαρακτηριστικού πολυωνύμου προκύπτει ότι:

$$| \lambda_{\max} | = 0 < 1.$$

Συνεπώς η μέθοδος συγκλίνει.

Για $k = 1 \quad \rightarrow \quad \begin{bmatrix} x_1^{(1)} \\ x_2^{(1)} \\ x_3^{(1)} \end{bmatrix} = \begin{bmatrix} 1.6667 \\ 1.5000 \\ 0.6667 \end{bmatrix}$, $\frac{\| x^{(1)} - x^{(0)} \|}{\| x^{(1)} \|} = 1 > \frac{1}{2} 10^{-2}$, ο αλγόριθμος συνεχίζει,

Για $k=2 \quad \rightarrow \quad \begin{bmatrix} x_1^{(2)} \\ x_2^{(2)} \\ x_3^{(2)} \end{bmatrix} = \begin{bmatrix} 0.6667 \\ 1.1667 \\ 1.1667 \end{bmatrix}$, $\frac{\| x^{(2)} - x^{(1)} \|}{\| x^{(2)} \|} = 0.6556 > \frac{1}{2}10^{-2}$, ο αλγόριθμος συνεχίζει,

Για $k=3 \quad \rightarrow \quad \begin{bmatrix} x_1^{(3)} \\ x_2^{(3)} \\ x_3^{(3)} \end{bmatrix} = \begin{bmatrix} 0.8889 \\ 0.9167 \\ 1.0556 \end{bmatrix}$, $\frac{\| x^{(3)} - x^{(3)} \|}{\| x^{(2)} \|} = 0.2128 > \frac{1}{2}10^{-2}$, ο αλγόριθμος συνεχίζει,

Για $k=4 \quad \rightarrow \quad \begin{bmatrix} x_1^{(4)} \\ x_2^{(4)} \\ x_3^{(4)} \end{bmatrix} = \begin{bmatrix} 1.0556 \\ 0.9722 \\ 0.9722 \end{bmatrix}$, $\frac{\| x^{(4)} - x^{(3)} \|}{\| x^{(4)} \|} = 0.1122 > \frac{1}{2}10^{-2}$, ο αλγόριθμος συνεχίζει,

Για $k=5 \quad \rightarrow \quad \begin{bmatrix} x_1^{(5)} \\ x_2^{(5)} \\ x_3^{(5)} \end{bmatrix} = \begin{bmatrix} 1.0185 \\ 1.0139 \\ 0.9907 \end{bmatrix}$, $\frac{\| x^{(5)} - x^{(4)} \|}{\| x^{(5)} \|} = 0.0337 > \frac{1}{2}10^{-2}$, ο αλγόριθμος συνεχίζει,

Για $k=6 \quad \rightarrow \quad \begin{bmatrix} x_1^{(6)} \\ x_2^{(6)} \\ x_3^{(6)} \end{bmatrix} = \begin{bmatrix} 0.9907 \\ 1.0046 \\ 1.0046 \end{bmatrix}$, $\frac{\| x^{(6)} - x^{(5)} \|}{\| x^{(6)} \|} = 0.0187 > \frac{1}{2}10^{-2}$, ο αλγόριθμος συνεχίζει,

Για $k=7 \quad \rightarrow \quad \begin{bmatrix} x_1^{(7)} \\ x_2^{(7)} \\ x_3^{(7)} \end{bmatrix} = \begin{bmatrix} 0.9969 \\ 0.9977 \\ 1.0015 \end{bmatrix}$, $\frac{\| x^{(7)} - x^{(6)} \|}{\| x^{(7)} \|} = 0.0057 > \frac{1}{2}10^{-2}$, ο αλγόριθμος συνεχίζει,

Για $k=8 \quad \rightarrow \quad \begin{bmatrix} x_1^{(8)} \\ x_2^{(8)} \\ x_3^{(8)} \end{bmatrix} = \begin{bmatrix} 1.0015 \\ 0.9992 \\ 0.9992 \end{bmatrix}$, $\frac{\| x^{(8)} - x^{(7)} \|}{\| x^{(8)} \|} = 0.0031 < \frac{1}{2}10^{-2}$, ο αλγόριθμος τερματίζει.

Συνεπώς η προσεγγιστική λύση του δοθέντος γραμμικού συστήματος με βάση την αρχικής εκτίμηση της λύσης και ακρίβεια 2 σημαντικών ψηφίων είναι:

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1.0015 \\ 0.9992 \\ 0.9992 \end{bmatrix}$$

**Υλοποίηση σε MATLAB:**

**main.m**

```
A = [3 2 0;0 2 1;1 0 2];
b=[5; 3; 3];
x = [0;0;0];
x = Jacobi(A, b, x, 2);
x = GaussSeidel(A, b, x, 2);
```

**GaussSeidel.m**

```
function x = GaussSeidel(A, b, x, m)

D = diag(diag(A))
L = diag(diag(A))-tril(A)
U = diag(diag(A))-triu(A)

d = diag(A);

D-L

inv(D-L)


T = inv(D-L)*U

sr = max(abs(eig(T)))

disp('Gauss Seidel method:');
if (sr < 1)
    iter = 0;

    while (true)
        iter = iter + 1;
        fprintf('Iteration: %d',iter)
        prevx = x;

        x = inv(D-L)*U*x + inv(D-L)*b

       if (norm(prevx-x)/norm(x))< 0.5*10^(-m), return; end
    end
else
    disp('ERROR: The Gauss Seidel method is diverging');
    return;
end
```

## 3 Πεπερασμένες και διαιρεμένες διαφορές

Έστω ότι μετράμε την απόσταση $y$ που διανύει ένα αυτοκίνητο ξεκινώντας από ένα δεδομένο σταθερό σημείο σε διάφορες χρονικές στιγμές. Γνωρίζουμε ότι ο ρυθμός της μεταβολής της απόστασης ως προς το χρόνο εκφράζει την ταχύτητα του αυτοκινήτου, ενώ ο ρυθμός της μεταβολής της ταχύτητας ως προς το χρόνο εκφράζει την επιτάχυνση του αυτοκινήτου. Η απόσταση $y$ που διανύει το κινούμενο αυτοκίνητο εξαρτάται από το χρόνο $x$ κατά τον οποίο το αυτοκίνητο κινείται. Έτσι αφού για μια δεδομένη χρονική στιγμή το αυτοκίνητο έχει διανύσει μια μοναδική απόσταση $y$, τότε η απόσταση $y$ είναι μια συνάρτηση $y = f(x)$

Υποθέτουμε ότι όταν η διανυόμενη απόσταση σε μέτρα μετριέται σε χρονικά διαστήματα των δέκα δευτερολέπτων, έχουμε τα ακόλουθα αποτελέσματα:

| $x$ | 0 | 10 | 20 | 30 | 40 | 50 | 60 |
|---|---|---|---|---|---|---|---|
| $y = f(x)$ | 0 | 212 | 740 | 1430 | 2280 | 3172 | 4114 |

Για να εξάγουμε περισσότερες πληροφορίες από τα παραπάνω δεδομένα γράφουμε τον πίνακα σε πιο εκτεταμένη μορφή, που ονομάζεται πίνακας πεπερασμένων διαφορών [6], όπως παρακάτω:

| $x$ | $f(x)$ | 1ες διαφ. | 2ες διαφ. | 3ες διαφ. | 4ες διαφ. | 5ες διαφ. | 6ες διαφ. |
|---|---|---|---|---|---|---|---|
| 0 | 0 | | | | | | |
| | | 212 | | | | | |
| 10 | 212 | | 316 | | | | |
| | | 528 | | −154 | | | |
| 20 | 740 | | 162 | | 152 | | |
| | | 690 | | −2 | | −268 | |
| 30 | 1430 | | 160 | | −116 | | 510 |
| | | 850 | | −118 | | 242 | |
| 40 | 2280 | | 42 | | 126 | | |
| | | 892 | | 8 | | | |
| 50 | 3172 | | 50 | | | | |
| | | 942 | | | | | |
| 60 | 4114 | | | | | | |

Παρατηρήσεις:

- ✓ Οι πρώτες διαφορές ή διαφορές πρώτης τάξης προκύπτουν αφαιρώντας κάθε συναρτησιακή τιμή από εκείνη την τιμή που βρίσκεται μια θέση ακριβώς αποκάτω της στον πίνακα των πεπερασμένων διαφορών.
- ✓ Με τον ίδιο τρόπο μπορούμε να αποκτήσουμε δεύτερης και υψηλότερης τάξης διαφορές.
- ✓ Γενικά, ένας πίνακας διαφορών που δημιουργείται με βάση $(k+1)$ τιμές εξαντλείται στη στήλη των διαφορών $k$ τάξης.

**Αλγόριθμος εύρεσης πεπερασμένων διαφορών:**

Ο αλγόριθμος δέχεται σαν είσοδο τα διανύσματα $x(i), i=1,...,n$ και $y(i), i=1,...,n$ καθώς και έναν βαθμωτό $k$ που δηλώνει τη μέγιστης τάξης πεπερασμένες διαφορές που θα υπολογιστούν.

- Για $j=1,...,k$ πραγματοποιείται:
  - ✓ Υπολογισμός του $y_j(m)=y_{j-1}(i+1)-y_{j-1}(i),$ όπου $m=1,...,n-j$ και $i=1,...,m$

### 3.1.1 Παράδειγμα:

Θεωρούμε ως είσοδο στον αλγόριθμο τα εξής δεδομένα:

Το διάνυσμα $x=\{0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8\}$, τον κλειστό τύπο της συνάρτησης $y=x^4-2x+10$, καθώς και $k=5$, δηλαδή εύρεση μέχρι 5ης τάξης πεπερασμένων διαφορών.

**Λύση:**

| Αρχικοποίηση | | $j=1$ | $j=2$ | $j=3$ | $j=4$ | $j=5$ |
|---|---|---|---|---|---|---|
| $x$ | $y$ | $y_1$ | $y_2$ | $y_3$ | $y_4$ | $y_5$ |
| –0.3 | 10.6081 | –0.207 | 0.005 | –0.004 | 0.002 | 0.000 |
| –0.2 | 10.4016 | –0.202 | 0.001 | –0.001 | 0.002 | 0.000 |
| –0.1 | 10.2001 | –0.200 | 0.000 | 0.001 | 0.002 | |
| 0 | 10.000 | –0.200 | 0.001 | 0.004 | | |
| 0.1 | 9.8001 | –0.199 | 0.005 | | | |
| 0.2 | 9.6016 | –0.194 | | | | |
| 0.3 | 9.4081 | | | | | |

**Υλοποίηση σε MATLAB:**

**main.m**

```
clear;
clc;
close all;

% Example 1
x = 0:0.1:0.9;
y = x.^3-8*x.^2-4*x-1;

fprintf('Example 1\n');
fdiff = DifferenceMatrix(x',y',4);

%Example 2
x=-0.3:0.1:0.3;
y = x.^4-2*x+10;

fprintf('\n\nExample 2\n');
fdiff = DifferenceMatrix(x',y',5);
```

**DifferenceMatrix.m**

```
function diff = DifferenceMatrix(x,y,k)

    n = length(x);
    m = length(y);

% Compute difference matrix
    if (n~=m)
        disp('The two vectors should be of equal size.');
        return;
    end

    diff(:,1) = y(2:end)-y(1:end-1);
    for i=2:k
        prev_end = length(y)-i+1;
        diff(1:length(y)-i,i) = diff(2:prev_end,i-1)-diff(1:prev_end-1,i-1);
    end


% Print matrix
    for i=1:length(y)-k
        for j=1:k
            fprintf('%6.3f ',diff(i,j))
        end
        fprintf('\n');
    end

    ind = 1;
    for i=length(y)-k+1 :length(y)-1
        for j=1:k-ind
            fprintf('%6.3f ',diff(i,j))
        end
        ind = ind + 1;
        fprintf('\n');
    end

end
```

# 4 Αριθμητική επίλυση εξισώσεων

## 4.1 Η μέθοδος διχοτόμησης

Δοθείσας της συνάρτησης $y = f(x)$ καθώς και της πληροφορίας ότι το διάστημα $x \in (a,b)$ περιέχει μια λύση της συνάρτησης, η μέθοδος της διχοτόμησης [6, 7] αποτελεί μια επαναληπτική μέθοδο για την ανίχνευση της λύσης $f(x) = 0$. Στη πραγματικότητα η μέθοδος της διχοτόμησης χρησιμοποιείται για να προσεγγίσει με μία επιθυμητή ακρίβεια $\varepsilon$ τη λύση $r$ της εξίσωσης $f(x) = 0$, η οποία είναι γνωστό ότι βρίσκεται στο διάστημα $(a,b)$.

Πιο συγκεκριμένα, έστω ότι έχουμε την εξίσωση $f(x) = 0$ όπου η συνάρτηση $f(x)$ είναι συνεχής στο κλειστό διάστημα $[a,b]$ για το οποίο ισχύει ότι $f(a)f(\mathrm{b}) < 0$. Σύμφωνα με το κριτήριο του Bolzano, θα υπάρχει μια λύση $r$ της $f(x) = 0$ στο διάστημα $(a,b)$. Στόχος είναι να βρεθεί μια προσεγγιστική λύση $x^*$ της πραγματικής λύσης $r$ με εφαρμογή της μεθόδου της διχοτόμησης.

**Αλγόριθμος διχοτόμησης:**

---

Ο αλγόριθμος δέχεται ως είσοδο τη συνάρτηση $f(x)$, καθώς και τα άκρα του διαστήματος $(a,b)$ στο οποίο εμπεριέχεται η πραγματική λύση $r$ της συνάρτησης $f(x) = 0$.

- Αρχικά εξετάζεται εάν ικανοποιείται το κριτήριο του Bolzano στο διάστημα $(a,b)$, δηλαδή εάν ικανοποιείται η σχέση: $f(a)\,f(b) < 0$. Εάν ικανοποιείται γίνεται εύρεση του μέσου του διαστήματος $(a,b)$, δηλαδή $x_0 = \dfrac{a+b}{2}$. Εάν για το σημείο αυτό ισχύει ότι $f(x_1) = 0$, τότε έχουμε υπολογίσει την πραγματική λύση η οποία είναι το $r = x_1$ και ο αλγόριθμος τερματίζει. Εάν όχι ο αλγόριθμος συνεχίζει.
- Για $i = 1, \ldots$ πραγματοποιείται:
  - ✓ Εξετάζουμε εάν $f(a)\,f(x_{i-1}) < 0$ και θέτουμε $a = a$ και $b = x_{i-1}$ ή εάν $f(x_{i-1})\,f(b) < 0$ και θέτουμε $a = x_{i-1}$ και $b = b$,
  - ✓ Βρίσκουμε την $i$ -οστή προσεγγιστική λύση: $x_i = \dfrac{a+b}{2}$,

    Εξετάζουμε εάν ικανοποιείται το κριτήριο τερματισμού της μεθόδου. Εάν ικανοποιείται το κριτήριο τερματισμού τότε ο αλγόριθμος τερματίζει στο $i$ - οστό βήμα και η προσεγγιστική λύση είναι η $x_i$, διαφορετικά ο αλγόριθμος επαναλαμβάνεται.

---

Κριτήριο τερματισμού μεθόδου διχοτόμησης:

Η μέθοδος της διχοτόμησης αποτελεί μια επαναληπτική διαδικασία για την εύρεση μιας προσεγγιστικής λύσης $x^*$ της πραγματικής λύσης $r$. Επειδή στην πράξη είναι αδύνατο να συμβεί $f(x_k) = 0$, μπορεί, αντί αυτού, να εξεταστεί κατά πόσο το $x_k$ είναι μια ικανοποιητική προσέγγιση της λύσης. Συνεπώς για τα κριτήρια τερματισμού της μεθόδου της διχοτόμησης χρησιμοποιούνται οι εξής σχέσεις:

$$
\begin{aligned}
&(a)\ \ |x_k - x_{k-1}| < \varepsilon, \\
&(b)\ \ |f(x_k)| < \varepsilon, \\
&(c)\ \ \frac{|x_k - x_{k-1}|}{|x_k|} < \varepsilon,\ \ x_k \neq 0.
\end{aligned}
\tag{3.1}
$$

### 4.1.1 Παράδειγμα:

Θεωρούμε τη συνάρτηση $f(x) = x^3 - 7$. Θέλουμε να προσεγγίσουμε τη λύση στο διάστημα $(0,3)$ με μέγιστο σφάλμα $\varepsilon = 0.1$ με βάση το $(a)$ κριτήριο τερματισμού.

**Λύση:**

1η επανάληψη:

$$f(0) f(3) = -7 * 20 < 0,$$

τότε: $x_1 = \frac{0+3}{2} = 1.5$

εξετάζεται κριτήριο σύγκλισης $|x_1 - x_0| = |1.5 - 0| = 1.5 > 0.1$, άρα ο αλγόριθμος συνεχίζει

2η επανάληψη:

$$f(0) f(1.5) = -7 * -3.6250 > 0,$$

$$f(1.5) f(3) = -3.6250 * 20 > 0,$$

τότε: $x_2 = \frac{1.5+3}{2} = 2.25$

εξετάζεται κριτήριο σύγκλισης $|x_2 - x_1| = |2.25 - 1.5| = 0.75 > 0.1$, άρα ο αλγόριθμος συνεχίζει

3η επανάληψη:

$$f(1.5) f(2.25) = -3.6250 * 4.3906 < 0,$$

τότε: $x_3 = \frac{1.5+2.25}{2} = 1.875$

εξετάζεται κριτήριο σύγκλισης $|x_3 - x_2| = |1.875 - 2.25| = 0.3750 > 0.1$, άρα ο αλγόριθμος συνεχίζει

4η επανάληψη:

$$f(1.5) f(1.875) = -3.6250 * -0.4082 > 0,$$

$$f(1.875) f(2.25) = -0.4082 * 4.3906 < 0,$$

τότε: $x_4 = \frac{1.875+2.25}{2} = 2.0625$

εξετάζεται κριτήριο σύγκλισης $|x_4 - x_3| = |2.0625 - 1.875| = 0.1875 > 0.1$, άρα ο αλγόριθμος συνεχίζει

5η επανάληψη:

$$f(1.875)\,f(2.0625) = -0.4082 * 1.7737 < 0,$$

τότε: $x_5 = \dfrac{1.875 + 2.0625}{2} = 1.96875$

εξετάζεται κριτήριο σύγκλισης $|x_5 - x_4| = |1.96875 - 2.0625| = 0.0938 < 0.1$, άρα ο αλγόριθμος τερματίζει.

Συνεπώς η μέθοδος της διχοτόμησης χρειάστηκε 5 επαναλήψεις για την εύρεση της προσεγγιστικής λύσης $x^* = 1.96875$ με μέγιστο αποδεκτό σφάλμα $\varepsilon = 0.1$.

**Υλοποίηση σε MATLAB:**

**main.m**

```
clear; clc; close all;
bisection('x^3-7',0, 3, 20, 0.1, true);
```

**bisection.m**

```
function [ a, b, k ] = bisection(f, a, b, N, Tol, verb)
%
% bisection(f, a, b, N, Tol)
%
% BISECTION
%   Bisection Method.
%
% Input:
%   f - function given as a string
%   a - lower bound
%   b - upper bound
%   N - number of iterations
%   Tol - error tolerance
%   verb - verbose mode
%
% Output:
%   a - lower bound
%   b - upper bound
%   k - number of iterations performed
%
% Examples:
%   [ a, b, k ] = bisection( '1/x', -1, 1, 1e2, 1e-5, true )
%   [ a, b, k ] = bisection( '1/x', -1, 1, 1e2, 1e-5, 1 )
%   [ a, b, k ] = bisection( 'abs(log(x))-0.2*sin(x)', 1, pi/2 )
%   [ a, b, k ] = bisection( 'abs(log(x))-0.2*sin(x)', .5, 1, 1e2, 1e-5 )
%


    if nargin == 3
        N = 1e4;
        Tol = 1e-4;
```

```
        verb = false;
    elseif nargin == 4
        Tol = 1e-4;
        verb = false;
    elseif nargin == 5
        verb = false;
    elseif nargin ~= 6
        error('bisection: invalid input parameterss');
    end

    f = inline(f);
    %fplot(f,[a b]); grid on; hold on;
    if (f(a) * f(b) >= 0) || (a >= b)
        error('bisection: condition f(a)*f(b)<0 or a<b didn''t apply');
    end

    k = 1;
    x(k) = (a + b) / 2;
    if verb == true
%         fprintf('\na = %d\nb = %d\nx(%d) = %d\n', a, b, k, x(k));
        fprintf('%3g %10g %10g %10g %10.4f %10.4f\n',k,a,b,x(k), f(x(k)),((b
- a) / 2))
    end
    plot(x(k),f(x(k)),'ro');

    while ((k <= N) && ((b - a) / 2) >= Tol)
        if f(x(k)) == 0
            error([ 'bisection: condition f(' num2str(x(k)) ...
                ')~=0 didn''t apply' ]);
        end
        if (f(x(k)) * f(a)) < 0
            b = x(k);
        else
            a = x(k);
        end
        k = k + 1;
        x(k) = (a + b) / 2;
        if verb == true
%             fprintf('\na = %d\nb = %d\nx(%d) = %d\n', a, b, k, x(k));
            fprintf('%3g %10g %10g %10g %10.4f
%10.4f\n',k,a,b,x(k),f(x(k)),((b - a) / 2))
        end
        %plot(x(k),f(x(k)),'ro');
    end

end
```

## 4.2 Η μέθοδος των Newton-Raphson

Η μέθοδος των Newton-Raphson [6, 8] περιγράφεται από ένα επαναληπτικό σχήμα που παράγεται από το ανάπτυγμα σε σειρά Taylor της συνάρτησης $y = f(x)$. Αν $r \in (a,b)$ είναι η πραγματική ρίζα της συνάρτησης

$y = f(x)$, για την οποία υποθέτουμε ότι είναι δύο φορές συνεχώς παραγωγίσιμη στο διάστημα $[a,b]$, τότε για κάθε $x_k \in (a,b)$ ισχύει ότι:

$$0 = f(r) = f(x_k) + (r - x_k) f'(x_k) + \frac{1}{2}(r - x_k)^2 f''(\xi), \tag{3.2}$$

όπου $\xi$ είναι ένα σημείο μεταξύ των $x_k$ και $r$.

Από την παραπάνω σχέση προκύπτει ότι:

$$r = x_k - \frac{f(x_k)}{f'(x_k)} - \frac{1}{2}(r - x_k)^2 \frac{f''(\xi)}{f'(x_k)}. \tag{3.3}$$

Επομένως, αν το σημείο $x_k$ είναι αρκετά κοντά στο σημείο $r$, τότε το σημείο:

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}. \tag{3.4}$$

Αποτελεί καλύτερη προσέγγιση της λύσης $r$ από ότι το σημείο $x_k$.

Με βάση τα παραπάνω η μέθοδος των Newton-Raphson είναι μια επαναληπτική μέθοδος για την προσέγγιση της $f(x) = 0$ που δίνεται από την εξής αναδρομική σχέση:

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}, \quad k = 0, 1, 2, \ldots \tag{3.5}$$

Με τη προϋπόθεση ότι η παράγωγος $f'(x)$ δε μηδενίζεται για κάθε $k$ και ότι το $x_0$ είναι μια δεδομένη αρχική προσέγγιση της λύσης $r$.

Για να μπορέσει η μέθοδος των Newton-Raphson να συγκλίνει θα πρέπει η συνάρτηση $f(x)$ να ικανοποιεί τις εξής συνθήκες:

- ✓ Είναι ορισμένη και δύο φορές συνεχώς παραγωγίσιμη στο $[a,b]$: $f(x) \in \mathbb{C}^2[a, b]$,
- ✓ Ικανοποιεί το κριτήριο Bolzano: $f(a) f(b) < 0$,
- ✓ Η $f'(x) \neq 0$ για κάθε $x \in [a, b]$,
- ✓ Η $f''(x)$ διατηρεί σταθερό πρόσημο στο $[a, b]$,
- ✓ Αν το $z$ είναι το σημείο όπου η $|f'(x)|$ παίρνει ελάχιστη τιμή στο $[a, b]$, τότε ισχύει $|f(z)/f'(z)| \leq b - a$.

Κριτήρια τερματισμού:

Όπως και στην επαναληπτική μέθοδο της διχοτόμησης, έτσι και στην μέθοδο των Newton-Raphson υπάρχει κάποιο κριτήριο τερματισμού το οποίο επιβάλλεται από το χρήστη. Συγκεκριμένα δοθέντος του μέγιστου αποδεκτού σφάλματος $\varepsilon$ η μέθοδος συγκλίνει όταν:

$$|x_k - x_{k-1}| < \varepsilon, \quad k = 1, \ldots \tag{3.6}$$

**Αλγόριθμος των Newton-Raphson:**

Ο αλγόριθμος δέχεται ως είσοδο τη συνάρτηση $f(x)$, την παράγωγο της συνάρτησης $f'(x)$, τα άκρα του διαστήματος $(a,b)$ στο οποίο εμπεριέχεται η πραγματική λύση $r$ της συνάρτησης $f(x)=0$, την αρχική εκτίμηση της λύσης $x_0$, καθώς και μέγιστό επιθυμητό σφάλμα $\varepsilon$ της προσεγγιστικής λύσης $x_i$ από την αμέσως προηγούμενη προσεγγιστική λύση $x_{i-1}$.

- Αρχικά πραγματοποιείται έλεγχος εάν η μέθοδος συγκλίνει. Εάν ναι ο αλγόριθμος συνεχίζει, εάν όχι ο αλγόριθμος τερματίζει.
- Για $i=1,...$ πραγματοποιείται:
  - ✓ Έλεγχος για μη-μηδενική παράγωγο στην $i$ - οστή προσεγγιστική λύση $x_i$: $f'(x_i)\neq 0$. Εάν είναι μη-μηδενική η τιμή της παραγώγου τότε ο αλγόριθμος συνεχίζει, διαφορετικά η μέθοδος δε συγκλίνει και η διαδικασία διακόπτεται.
  - ✓ Εύρεση $i$ - οστής προσεγγιστικής λύσης: $x_{i+1}=x_i-\dfrac{f(x_i)}{f'(x_i)}$,
  - ✓ Έλεγχος κριτηρίου σύγκλισης: $|x_i-x_{i-1}|<\varepsilon$. Εάν ικανοποιείται το κριτήριο σύγκλισης τότε ο αλγόριθμος τερματίζει στο $i$ - οστο βήμα και η προσεγγιστική λύση είναι η $x_i$, διαφορετικά ο αλγόριθμος επαναλαμβάνεται.

### 4.2.1 Παράδειγμα:

Θεωρούμε τη συνάρτηση $f(x)=x^3-7$. Θέλουμε να προσεγγίσουμε τη λύση $f(x)=0$ με τη μέθοδο των Newton-Raphson στο διάστημα $(0,3)$, θεωρώντας αρχική εκτιμώμενη λύση $x_0=1.5$ και ακρίβεια ενός δεκαδικού ψηφίου, δηλαδή μέγιστο αποδεκτό σφάλμα $\varepsilon=0.01$.

**Λύση:**

Για τη δοθείσα συνάρτηση ισχύει:

- ✓ $f(0)f(3)=-7*20<0$
- ✓ $f'(x)=3x^2\neq 0, x\in[0,3]$
- ✓ Η $f''(x)=6x$ διατηρεί σταθερό πρόσημο $\exists\, x\in[0,3]$

Συνεπώς η μέθοδος συγκλίνει και μπορεί να εφαρμοστεί.

1η επανάληψη:

$$x_1=x_0-\frac{f(x_0)}{f'(x_0)}=1.5-\frac{f(1.5)}{f'(1.5)}=2.0370$$

εξετάζεται κριτήριο το σύγκλισης: $|x_1-x_0|=|2.0370-1.5|=0.5370>0.01$, άρα ο αλγόριθμος συνεχίζει

2η επανάληψη:

$$x_2 = x_1 - \frac{f(x_1)}{f'(x_1)} = 2.037 - \frac{f(2.037)}{f'(2.037)} = 1.9203,$$

εξετάζεται το κριτήριο σύγκλισης: $|x_2 - x_1| = |1.9203 - 2.0370| = 0.1167 > 0.01$, άρα ο αλγόριθμος συνεχίζει

3η επανάληψη:

$$x_3 = x_2 - \frac{f(x_2)}{f'(x_2)} = 1.9203 - \frac{f(1.9203)}{f'(1.9203)} = 1.9130,$$

εξετάζεται το κριτήριο σύγκλισης: $|x_2 - x_1| = |1.9130 - 1.9203| = 0.0073 < 0.01$, άρα ο αλγόριθμος τερματίζει.

Συνεπώς η μέθοδος Newton-Raphson χρειάστηκε 3 επαναλήψεις για την εύρεση της προσεγγιστικής λύσης $x^* = 1.9130$ με ακρίβεια ενός δεκαδικού ψηφίου.

**Υλοποίηση σε MATLAB:**

**main.m**

```
clear; clc; close all;
new_raph('x^3-7','3*x^2', 1.5, 30, 0.01, 0, 3);
```

**new_raph.m**

```
function xcur = new_raph( f, df, x0, N, tol, a, b )

% This routine implements the Newton-Raphson's method of rootfinding.
% The following are the input parameters to the routine :
%    f : it is any function involving one independent variable of the form
f( variables ) = 0.
%           i.e. all the terms of the function have to be transferred to LHS
making RHS as 0.
%    df : the derivative of f.
%    x0 : initial guess of the root / starting value.
%    N : maximum number of iterations needed to be performed.
%    tol : allowable deviation / tolerance.

   k = 1;         % iteration counter
   x(1) = x0;     % some random initialisations
   xcur = x(1);

   f = inline(f);
   df = inline(df);

%   fplot(f,[a b]);
   grid on; hold on;
   fprintf('%3g %10.4f %10.4f\n',k, x(k), f(x(k)))
```

```
   while( k <= N )
      fx = f(x(k));
      dfx = df(x(k));

      if( dfx == 0 )
         error( 'Derivative has become 0' );
      else
         k = k + 1;
         x(k) = x(k-1) - fx / dfx
         xcur = x(k);
         plot(x(k),f(x(k)),'ro');
         fprintf('%3g %10.4f %10.4f\n',k, x(k), f(x(k)))
      end

      if (abs(x(k-1)-x(k)) <= tol)
          return;
      end

   end

return;
```

# 5 Παρεμβολή και Παρεκβολή

**Ορισμός παρεμβολής:**

Αν έχουμε στη διάθεση μας τις τιμές μιας συνάρτησης $f(x)$ που αντιστοιχούν σε διάφορες τιμές $x_o, x_1, x_2,..., x_n$ της ανεξάρτητης μεταβλητής $x$, τότε παρεμβολή είναι η διαδικασία με την οποία αποκτούμε, από τις δοθείσες συναρτησιακές τιμές, προσεγγίσεις της $f(x)$ για τιμές της ανεξάρτητης μεταβλητής $x$ που βρίσκονται ενδιάμεσα στις δεδομένες τιμές $x_o, x_1, x_2,..., x_n$ [6].

**Ορισμός παρεκβολής:**

Αν έχουμε στη διάθεση μας τις τιμές μιας συνάρτησης $f(x)$ που αντιστοιχούν σε διάφορες τιμές $x_o, x_1, x_2,..., x_n$ της ανεξάρτητης μεταβλητής $x$, τότε παρεκβολή είναι η διαδικασία με την οποία αποκτούμε, από τις δοθείσες συναρτησιακές τιμές, προσεγγίσεις της $f(x)$ για τιμές της ανεξάρτητης μεταβλητής $x$ που βρίσκονται εκτός στις δεδομένες τιμές $x_o, x_1, x_2,..., x_n$ [6].

**Ορισμός αντίστροφης παρεμβολής:**

Αν έχουμε στη διάθεση μας τις τιμές μιας συνάρτησης $f(x)$ που αντιστοιχούν σε διάφορες τιμές $x_o, x_1, x_2,..., x_n$ της ανεξάρτητης μεταβλητής $x$, τότε αντίστροφη παρεμβολή είναι η διαδικασία με την οποία αποκτούμε, από τις δοθείσες συναρτησιακές τιμές, προσεγγίσεις της ανεξάρτητης μεταβλητής $x$ για τις οποίες η $f(x)$ παίρνει μια συγκεκριμένη τιμή [6].

Γενικά, ένας τρόπος για να πραγματοποιηθεί η παρεμβολή και η παρεκβολή βασίζεται στην κατασκευή ενός πολυωνύμου που διέρχεται από όλα τα δοθέντα σημεία $(x, f(x))$. Τότε μπορούμε να κάνουμε παρεμβολή ή παρεκβολή υπολογίζοντας την τιμή του πολυωνύμου για οποιαδήποτε τιμή της ανεξάρτητης μεταβλητής $x$.

## 5.1 Παρεμβολή Lagrange

Έστω ότι έχουμε στη διάθεση μας τις τιμές τις συνάρτησης $f(x)$ σε $(n+1)$ διακεκριμένα σημεία $x_i, i = 0,...,n$ τα οποία εν γένει δεν ισαπέχουν. Υποθέτουμε ότι η συνάρτηση αυτή παρεμβάλλεται με ένα πολυώνυμο $P_n(x)$ το πολύ βαθμού $n$, έτσι ώστε οι τιμές του πολυωνύμου να συμπίπτουν με τις αντίστοιχες τιμές της συνάρτησης $f_i = f(x_i)$ στα $(n+1)$ σημεία παρεμβολής $x_i, i = 0,...,n$. Το πρόβλημα της παρεμβολής έγκειται στην εύρεση του πολυωνύμου $P_n(x)$ χωρίς την επίλυση του αντίστοιχου γραμμικού συστήματος [6].

Εξετάζουμε πρώτα την περίπτωση που τα σημεία παρεμβολής είναι δύο, $x_0$ και $x_1$, με $x_0 \neq x_1$.

Θεωρούμε το πολυώνυμο πρώτου βαθμού: $P_1(x) = a_0 + a_1 x$ και ζητούμε τον προσδιορισμό των συντελεστών $a_0$ και $a_1$.

Συγκεκριμένα οι συντελεστές $a_i, i = 0,1$ πρέπει να ικανοποιούν το σύστημα:

$$\begin{aligned} a_0 + x_0 a_1 &= f(x_0), \\ a_0 + x_1 a_1 &= f(x_1). \end{aligned} \tag{4.1}$$

Το σύστημα αυτό έχει μία μοναδική λύση, αφού η ορίζουσα των συντελεστών των αγνώστων $a_i, i = 0,1$ είναι διάφορη του μηδενός.

Επομένως, οι συντελεστές $a_i$, $i=0,1$ δίνονται από τις σχέσεις $a_i = \det V_i / \det V$ όπου οι ορίζουσες $\det V_i$ για $i=0,1$, προσδιορίζονται από την ορίζουσα $\det V$ αν αντικαταστήσουμε την $(i+1)$ στήλη αυτής με το διάνυσμα $\left| f(x_0), f(x_1) \right|$, δηλαδή από τις παρακάτω ορίζουσες:

$$\det V_0 = \begin{vmatrix} f(x_0) & x_0 \\ f(x_1) & x_1 \end{vmatrix} \text{ και } \det V_1 = \begin{vmatrix} 1 & f(x_0) \\ 1 & f(x_1) \end{vmatrix}. \tag{4.2}$$

Επομένως, έχουμε:

$$a_0 = \frac{\det V_0}{\det V} = \frac{x_1 f(x_0) - x_0 f(x_1)}{x_1 - x_0} \text{ και } a_1 = \frac{\det V_1}{\det V} = \frac{f(x_1) - f(x_0)}{x_1 - x_0}. \tag{4.3}$$

Με βάση αυτά, το γραμμικό πολυώνυμο που αναζητούμε γίνεται διαδοχικά:

$$\begin{aligned} P_1(x) = a_0 + a_1 x &= \frac{x_1 f(x_0) - x_0 f(x_1)}{x_1 - x_0} + \frac{f(x_1) - f(x_0)}{x_1 - x_0} x \\ &= \frac{(x - x_0) f(x_1) - (x - x_1) f(x_0)}{x_1 - x_0}, \end{aligned} \tag{4.4}$$

από την οποία με διαχωρισμό των όρων του αριθμητή προκύπτει:

$$P_1(x) = \frac{x - x_1}{x_0 - x_1} f(x_0) + \frac{x - x_0}{x_1 - x_0} f(x_1), \tag{4.5}$$

όπου η παραπάνω σχέση παριστά την ευθεία που διέρχεται από τα σημεία $(x_0, f(x_0))$ και $(x_1, f(x_1))$.

Αν η συνάρτηση $f(x)$ είναι συνεχής στο κλειστό διάστημα $[x_0, x_1]$ και αν είναι παραγωγίσιμη στο ανοικτό διάστημα $(x_0, x_1)$, τότε το πολυώνυμο παρεμβολής $P_1(x)$ αποτελεί το πολυώνυμο Taylor πρώτου βαθμού. Το πολυώνυμο παρεμβολής $P_1(x)$ μπορεί να γραφεί και με την παρακάτω μορφή:

$$P_1(x) = f(x_0) + (x - x_0) \frac{f(x_1) - f(x_0)}{x_1 - x_0}. \tag{4.6}$$

Αφού η συνάρτηση $f(x)$ είναι συνεχής στο κλειστό διάστημα $[x_0, x_1]$ και αν είναι παραγωγίσιμη στο ανοικτό διάστημα $(x_0, x_1)$, τότε από το θεώρημα της μέσης τιμής υπάρχει ένα σημείο $\xi$ μεταξύ των σημείων $x_0$ και $x_1$ για το οποίο ισχύει ότι:

$$f'(\xi) = \frac{f(x_1) - f(x_0)}{x_1 - x_0}. \tag{4.7}$$

Με βάση αυτή τη σχέση, το πολυώνυμο (4.6) μπορεί να γραφεί και ως εξής:

$$P_1(x) = f(x_0) + (x - x_0) f'(\xi), \tag{4.8}$$

το οποίο αποτελεί το πολυώνυμο Taylor πρώτου βαθμού

Το πολυώνυμο παρεμβολής $P_1(x)$ μπορεί να γραφεί ως $P_1(x) = L_0(x) f(x_0) + L_1(x) f(x_1)$ όπου:

$$L_0(x)=\frac{x-x_1}{x_0-x_1} \text{ και } L_1(x)=\frac{x-x_0}{x_1-x_0}. \tag{4.9}$$

Παρατήρηση:

- ✓ Αν στα παραπάνω πολυώνυμα αν θέσουμε $x=x_0$ τότε θα έχουμε:

$$L_0(x_0)=1 \text{και } L_1(x_0)=0 \text{, οπότε } P_1(x_0)=f(x_0), \tag{4.10}$$

ενώ αν θέσουμε $x=x_1$ τότε θα έχουμε

$$L_0(x_1)=0 \text{ και } L_1(x_1)=1 \text{ οπότε } P_1(x_1)=f(x_1). \tag{4.11}$$

Σχετικά με το γενικευμένο πρόβλημα στον προσδιορισμό του πολυωνύμου παρεμβολής βαθμού $n$ το οποίο να ικανοποιεί τις συνθήκες ταύτισης:

$$P_n(x_i)=f_i, i=0,...,n \text{ .} \tag{4.12}$$

Παρατήρηση:

- ✓ Για τα πολυώνυμα πρώτου βαθμού $L_0(x)=1$ και $L_1(x)$ ισχύει ότι στο σημείο $x_j$ η τιμή του πολυωνύμου $L_i(x_j)$ είναι ένα όταν $i=j$ και μηδέν όταν $i\neq j$.

Δηλαδή, για $i,j=1,2$ ισχύει ότι:

$$L_i(x_j)=\delta_i^j, \tag{4.13}$$

Όπου $\delta_i^j$ είναι το γνωστό δέλτα του Kronecker που ορίζεται ως εξής:

$$\delta_i^j=\begin{cases}1 & \text{αν} \quad i=j\\ 0 & \text{αν} \quad i\neq j\end{cases} \tag{4.14}$$

Σχετικά με το γενικευμένο πρόβλημα στον προσδιορισμό του πολυωνύμου παρεμβολής βαθμού $n$ το οποίο να ικανοποιεί τις συνθήκες ταύτισης:

$$P_n(x_i)=f_i, i=0,...,n, \tag{4.15}$$

το ερώτημα που τίθεται είναι αν μπορούν να οριστούν εκφράσεις των πολυωνύμων $L_i(x), i=0,...,n$ τέτοιες ώστε να ισχύει:

$$P_n(x)=L_0(x)f_0+L_1(x)f_1+\cdots+L_n(x)f_n, \tag{4.16}$$

όπου τα πολυώνυμα $L_i(x), i=0,...,n$ έχουν την ιδιότητα:

$$L_i(x_j)=\delta_i^j=\begin{cases}1 & \text{αν} \quad i=j\\ 0 & \text{αν} \quad i\neq j\end{cases} \; i,j=0,...,n. \tag{4.17}$$

Τότε το πολυώνυμο $P_n(x)$ ικανοποιεί όλες τις συνθήκες ταύτισης.

Αφού ισχύει ότι $L_i(x_j)=0$ για $i\neq j$ στο πολυώνυμο $L_i(x)$ μηδενίζεται για κάθε τιμή του $x=x_j$ για όλα τα $j=0,...,n$ με $j\neq i$.

Επομένως, το πολυώνυμο $L_i(x)$ διαιρείται με κάθε παράγοντα $(x-x_j)$ για όλα τα $j=0,...,n$ με $j\neq i$ και κατά συνέπεια διαιρείται και με το γινόμενο των όρων αυτών.

Έχοντας ως βάση αυτό, συμπεραίνουμε ότι η έκφραση του πολυωνύμου $L_i(x)$ θα είναι η εξής:

$$L_i(x)=c(x-x_0)\cdots(x-x_{i-1})(x-x_{i+1})\cdots(x-x_j), \tag{4.18}$$

όπου $c$ είναι μια σταθερά

Αφού ισχύει ότι $L_i(x_i)=1$ για όλα τα $i=0,...,n,$ τότε:

$$L_i(x_i)=c(x_i-x_0)\cdots(x_i-x_{i-1})(x_i-x_{i+1})\cdots(x_i-x_n)=1, \tag{4.19}$$

οπότε έχουμε ότι η σταθερά $c$ έχει την τιμή:

$$c=\frac{1}{(x_i-x_0)\cdots(x_i-x_{i-1})(x_i-x_{i+1})\cdots(x_i-x_n)}. \tag{4.20}$$

Επομένως, η ζητούμενη έκφραση των πολυωνύμων $L_i(x)$ είναι η εξής:

$$L_i(x)=\frac{\prod_{\substack{j=o\\ j\neq i}}^{n}(x-x_j)}{\prod_{\substack{j=0\\ j\neq 1}}^{n}(x_i-x_j)},\ i=0,...,n. \tag{4.21}$$

Τα πολυώνυμα $L_i(x)$ ονομάζονται συντελεστές παρεμβολής του Lagrange ή θεμελιώδη πολυώνυμα σημειακής παρεμβολής. Ενώ το γενικευμένο πολυώνυμο παρεμβολής του Lagrange δίνεται από τη σχέση:

$$P_n(x)=\sum_{i=0}^{n}L_i(x)\,f_i \Rightarrow P_n(x)=\sum_{i=0}^{n}\frac{\prod_{\substack{j=o\\ j\neq i}}^{n}(x-x_j)}{\prod_{\substack{j=0\\ j\neq 1}}^{n}(x_i-x_j)}\,f_i\,. \tag{4.22}$$

Οι συντελεστές του Lagrange $L_i(x)$ μπορούν να γραφούν με την παρακάτω μορφή:

$$L_i(x)=\frac{L(x)}{(x-x_i)L'(x_i)},\ i=0,...,n\,, \tag{4.22}$$

όπου,

$$L(x)=(x-x_0)(x-x_1)\cdots(x-x_n)=\prod_{j=0}^{n}(x-x_j), \tag{4.23}$$

και,

$$L'(x) = L(x)\sum_{j=0}^{n}\frac{1}{(x-x_j)}. \tag{4.24}$$

Άρα, για το πολυώνυμο παρεμβολής του Lagrange προκύπτει ως:

$$P_n(x) = L(x)\sum_{j=0}^{n}\frac{f_i}{(x-x_i)L'(x_i)}. \tag{4.25}$$

Για τους συντελεστές $L_i(x)$ του Lagrange ισχύουν οι παρακάτω σχέσεις του Cauchy:

$$\begin{aligned} &(a) \quad \sum_{i=0}^{n}(x_i - x)^k L_i(x) = 0, \text{ για όλα τα } k = 1,...,n, \\ &(b) \quad \sum_{i=0}^{n} L_i(x) = 1. \end{aligned} \tag{4.26}$$

Υποθέτουμε ότι $(x_i, f_i), i = 0,...,n$ είναι τα $(n+1)$ σημεία παρεμβολής της συνάρτησης $f(x)$. Τότε από το πολυώνυμο παρεμβολής $P_n(x)$ του Lagrange έχουμε ότι:

$$P_n(x) = \sum_{i=0}^{n} L_i(x)\, f_i\,. \tag{4.26}$$

**Αλγόριθμος παρεμβολής Lagrange:**

---

Ο αλγόριθμος δέχεται ως είσοδο $(n+1)$ παρατηρήσεις της συνάρτησης $f(x)$: δηλαδή $(x(i), y(i)), i = 0,...,n$.

Για $i = 0,...,n$ πραγματοποιείται:

Για $j = 0,...,n$ πραγματοποιείται:

Εάν $i \neq j$ πραγματοποιείται:

$$L = L\frac{x - x(j)}{x(i) - x(j)},$$

$P = P + L\, y(i)$

$L = 0$

---

### 5.1.1 Παράδειγμα:

Δίνονται τα σημεία $(x, f(x)), i = 0,1,2$. Να υπολογιστεί το πολυώνυμο Lagrange που διέρχεται από τα σημεία των παρατηρήσεων της $f(x)$ όπως φαίνονται στο πίνακα:

| $x$ | -1 | 0 | 1 |
|---|---|---|---|
| $y = f(x)$ | -3 | -2 | -1 |

**Λύση:**

Έχοντας στη διάθεση μας τις τρεις παρατηρήσεις της συνάρτησης $f(x)$, το πολυώνυμο Lagrange αναπτύσσεται ως εξής:

$$P_2(x) = \sum_{i=0}^{2} f_i L_i(x) = -3L_0(x) - 2L_1(x) - L_2(x),$$

όπου

$$L_0(x) = \frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)} = \frac{(x-0)(x-1)}{(-1-0)(-1-1)} = \frac{x^2-x}{2},$$

$$L_1(x) = \frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)} = \frac{(x+0)(x-1)}{(0+1)(0-1)} = \frac{x^2-1}{-1},$$

$$L_2(x) = \frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)} = \frac{(x+1)(x-0)}{(1+1)(1-0)} = \frac{x^2+x}{2}.$$

Αντικαθιστώντας το πολυώνυμο Lagrange που διέρχεται από τα δοθέντα σημεία είναι:

$$P_2(x) = -3\frac{x^2-x}{2} - 2\frac{x^2-1}{-1} - \frac{x^2+x}{2} = x-2.$$

**Υλοποίηση σε MATLAB:**

**main.m**

```
clear;
clc;
close all;

pointx = [-2 0 1];
x = [-2:0.01:1];

% Example1
pointy = pointx.^3;
y = lagrange(x,pointx,pointy);
figure;
plot(pointx, pointy,'ro',x,y,x,x.^3,'r');
% hold on;
% plot(x,-x.^2+2*x,'c');
% hold off;
xlabel('x');
ylabel('y');
legend('(x_i,f(x_i))','-x^2+2*x','x^3','Location','SouthEast');
legend('boxoff')
title('Example1: f(x) = x^3');



% Example2
pointy = pointx.^4;
y = lagrange(x,pointx,pointy);
figure;
```

```
plot(pointx, pointy,'ro',x,y,x,x.^4,'r');
% hold on;
% plot(x,3*x.^2-2*x,'c');
% hold off;
xlabel('x');
ylabel('y');
legend('(x_i,f(x_i))','3*x^2-2*x','x^4');
legend('boxoff')
title('Example2: f(x) = x^4');

% Example3
pointy = pointx.^5;
y = lagrange(x,pointx,pointy);
figure;
plot(pointx, pointy,'ro',x,y,x,x.^5,'r');
% hold on;
% plot(x,-5*x.^2+6*x,'c');
% hold off;
xlabel('x');
ylabel('y');
legend('(x_i,f(x_i))','-5*x^2+6*x','x^5','Location','SouthEast');
legend('boxoff')
title('Example3: f(x) = x^5');
```

**lagrange.m**

```
function y=lagrange(x,pointx,pointy)
%
%LAGRANGE   approx a point-defined function using the Lagrange polynomial
interpolation
%
%      LAGRANGE(X,POINTX,POINTY) approx the function definited by the
points:
%      P1=(POINTX(1),POINTY(1)), P2=(POINTX(2),POINTY(2)), ...,
PN(POINTX(N),POINTY(N))
%      and calculate it in each elements of X
%
%      If POINTX and POINTY have different number of elements the function
will return the NaN value
%
%      function wrote by: Calzino
%      7-oct-2001
%
n=size(pointx,2);
L=ones(n,size(x,2));
if (size(pointx,2)~=size(pointy,2))
   fprintf(1,'\nERROR!\nPOINTX and POINTY must have the same number of
elements\n');
   y=NaN;
else
   for i=1:n
      for j=1:n
         if (i~=j)
            L(i,:)=L(i,:).*(x-pointx(j))/(pointx(i)-pointx(j));
```

```
        end
      end
   end
   y=0;
   for i=1:n
      y=y+pointy(i)*L(i,:);
   end
end
```

# 6 Αριθμητική Παραγώγιση και Ολοκλήρωση

## 6.1 Αριθμητική Παραγώγιση

Αν $y = f(x)$ είναι μια συνάρτηση ορισμένη σε ένα διάστημα $[a,b]$, τότε σε κάθε εσωτερικό σημείο $x$, μπορούμε να δώσουμε στο $x$ μια θετική ή αρνητική αύξηση $\Delta x = h$ και, κατά συνέπεια, η αντίστοιχη αύξηση του $y$ θα είναι $\Delta y = f(x+h) - f(x)$.

Όταν το $x$ είναι σταθερό, το πηλίκο των διαφορών:

$$\frac{\Delta y}{\Delta x} = \frac{f(x+h) - f(x)}{h}, \tag{5.1}$$

Είναι μια συνάρτηση του $h$.

Αν ο παραπάνω λόγος πλησιάζει στο ίδιο όριο καθώς το $h$ τείνει στο μηδέν με οποιοδήποτε τρόπο αποφεύγοντας την τιμή $h = 0$, τότε το όριο αυτό ονομάζεται παράγωγος της συνάρτησης $f(x)$ και συμβολίζεται ως εξής:

$$\frac{df(x)}{dx} \equiv \frac{d}{dx} f(x) \equiv y' \equiv f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}, \tag{5.2}$$

Αν η συνάρτηση $y = f(x)$ έχει παράγωγο $f'(x)$ σε κάποια περιοχή του σημείου $x$, και αν η συνάρτηση της παραγώγου έχει μία παράγωγο στο $x$, τότε συμβολίζουμε αυτή τη δεύτερη παράγωγο με $f''(x)$.

Συνεπώς με διαδοχικές παραγωγίσεις μπορούμε να πάρουμε $f'(x), f''(x), f'''(x), \ldots$ όπου $n-$τάξης παράγωγο συμβολίζεται ως εξής:

$$\frac{d^n f(x)}{dx^n} \equiv \frac{d^n}{dx^n} f(x) \equiv y^{(n)} \equiv f^{(n)}(x), \tag{5.3}$$

**Παράδειγμα:**

Θεωρούμε τον κλειστό τύπο της συνάρτηση $f(x) = \frac{x^2 - 5x + 6}{x - 1}$. Θέλουμε να βρούμε τον τύπο της πρώτη παραγώγου της $f(x)$.

**Λύση:**

Από τα μαθηματικά γνωρίζουμε ότι: $f'(x) = \frac{2x-5}{x-1} - \frac{x^2 - 5x + 6}{(x-1)^2}$

Αναθέτοντας $x = 5$ βρίσκουμε: $f'(5) = 0.8750$

**Υλοποίηση σε MATLAB:**

```
syms x;
f=(x^2-5*x+6)/(x-1);
df=diff(f);
pretty(f); pretty(df);
x=5;
subs(df);
```

### 6.1.1 Παράδειγμα:

Θεωρούμε δεδομένες τις τιμές των παρατηρήσεων της συνάρτησης $f(x)=\frac{x^2-5x+6}{x-1}$ για $x=5,...,10$ με βήμα ολοκλήρωσης $h=1$, δηλαδή:

| $x$ | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|
| $f(x)$ | 1.50 | 2.40 | 3.33 | 4.29 | 5.25 | 6.22 |

Θέλουμε να υπολογίσουμε τις τιμές της πρώτης παραγώγου.

**Λύση:**

Με βάση τον τύπο (5.1) υπολογίζουμε:

$$\frac{\Delta y_1}{\Delta x}=\frac{f(5+1)-f(5)}{1}=0.90,$$

$$\frac{\Delta y_2}{\Delta x}=\frac{f(6+1)-f(6)}{1}=0.93,$$

$$\frac{\Delta y_3}{\Delta x}=\frac{f(7+1)-f(7)}{1}=0.95,$$

$$\frac{\Delta y_4}{\Delta x}=\frac{f(8+1)-f(8)}{1}=0.96,$$

$$\frac{\Delta y_5}{\Delta x}=\frac{f(9+1)-f(9)}{1}=0.97.$$

**Υλοποίηση σε MATLAB:**

```
a=5;
b=10;
h=1;
x=[a:h:b];
f=(x.^2-5.*x+6)./(x-1);
df=diff(f)/h
```

## 6.2 Αριθμητική Ολοκλήρωση

Για να υπολογιστεί το ολοκλήρωμα $\int_a^b f(x)dx$, όπου η $f(x)$ είναι μία πραγματική φραγμένη και ολοκληρώσιμη συνάρτηση, χρησιμοποιώντας αναλυτικές μεθόδους, χρειάζεται να βρεθεί σε αναλυτική (κλειστή μορφή) η αντίστοιχη έκφραση της παράγουσας $\mathrm{F}(x)$.

Ουσιαστικά η αριθμητική ολοκλήρωση προσεγγίζει το ολοκλήρωμα $I$ με τύπους ολοκλήρωσης της μορφής:

$$Q_{n+1}(f) = \sum_{i=0}^{n} w_i f(w_i). \tag{5.4}$$

Όπου τα $w_i$ είναι πραγματικοί αριθμοί, ενώ ο τύπος (5.4) ονομάζεται κανόνας υπολογισμού.

Δοθέντων των παρατηρήσεων της συνάρτησης $y = f(x)$ στο διάστημα $[a,b]$ καθώς και του βήματος ολοκλήρωσης $h$, η αριθμητική ολοκλήρωση υπολογίζεται σύμφωνα με το γενικευμένο κανόνα του τραπεζίου ως εξής:

$$\int_{x_0}^{x_k} f(x)dx \approx \frac{h}{2}\left[ f_0 + 2f_1 + 2f_2 + ... + 2f_{k-1} + f_k \right]. \tag{5.4}$$

### 6.2.1 Παράδειγμα:

Θεωρούμε τον κλειστό τύπο της συνάρτηση $f(x) = \sin(x)$. Θέλουμε να υπολογίσουμε τον τύπο του ολοκληρώματος της $f(x)$.

**Λύση:**

Από τα μαθηματικά γνωρίζουμε ότι: $\mathrm{F}(x) = \int_{-1}^{1} \sin(x)\,dx = -\cos(x)$.

Αναθέτοντας $x = 2\pi$ βρίσκουμε: $\mathrm{F}(2\pi) = -\cos(2\pi) = 0$

**Υλοποίηση σε MATLAB:**

```
syms x;
f=sin(x);
ezplot(f);
hold on;
fint=int(f);
pretty(f);
pretty(fint);
h=ezplot(fint);
set(h,'Color','r');
grid on;
axis([-4 4 -4 4]);
hold off;
```

### 6.2.2 Παράδειγμα:

Θεωρούμε δεδομένες τις τιμές των παρατηρήσεων της συνάρτησης $f(x)=\sin(x)$ για $x=0,...,2\pi$ με βήμα $h=\frac{\pi}{2}$, δηλαδή:

| $x$ | 0 | π/2 | π | 3π/2 | 2π |
|---|---|---|---|---|---|
| $f(x)$ | 0 | 1 | 0 | -1 | 0 |

**Λύση:**

Με βάση τύπο (5.4) υπολογίζουμε:

$$\mathrm{F}(x)=\frac{h}{2}\left[ f(0)+f(2\pi)+2\left[ f(\frac{\pi}{2})+f(\pi)+f(\frac{3\pi}{2}) \right] \right]=\frac{1}{2}\left[0+0+2\left[1+0+(-1)\right]\right]=0$$

**Υλοποίηση σε MATLAB:**

```
a=0;
b=2*pi;
N=100;
h=pi/2;
x=[a:h:b];
y=sin(x);
Q=h/2*(sin(a)+sin(b)+2*sum(sin(a+[1:N-1]*h)))
```

## Αναφορές:


[1] H. Moore, MATLAB for Engineers: Pearson, 2017.

[2] C. B. Moler, Numerical Computing with MATLAB: Revised Reprint vol. 87: Siam, 2008.

[3] G. Strang, G. Strang, G. Strang, and G. Strang, Introduction to linear algebra vol. 3: Wellesley-Cambridge Press Wellesley, MA, 1993.

[4] L. Richard and J. Burden, "Douglas faires, numerical analysis," ed: PWS Publishing Co., Boston, MA, 1988.

[5] W. Gautschi, Numerical analysis: Springer Science & Business Media, 2011.

[6] Μ. Βραχάτης, "Αριθμητική Ανάλυση," Εκδ.«Ελληνικά Γράμματα, 2002.

[7] Γ. Παπαγεωργίου and Χ. Τσιπούρας, "Αριθμητική Ανάλυση," Εκδ. Συμεών, Αθήνα, 2000.

[8] Γ. Παπαγεωργίου and Τ. Χ. Γρ, "Αριθμητική ανάλυση με εφαρμογές σε Matlab και Mathematica," ed: Εκδόσεις Συμεών, Αθήνα, 1997.