# Επιστημονικοί Υπολογισμοί

## Αριθμητική Επίλυση συστημάτων μη-γραμμικών εξισώσεων

Σταμάτιος-Άγγελος Ν. Αλεξανδρόπουλος
e-mail: stalexan@ee.duth.gr

https://www.researchgate.net/profile/Stamatios_Aggelos_Alexandropoulos
https://scholar.google.gr/citations?user=mht7W_YAAAAJ&hl=el
http://cilab.math.upatras.gr

Τμήμα Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών
Δημοκρίτειο Πανεπιστήμιο Θράκης
Κιμμέρια 67100, Ξάνθη

23 Μαρτίου 2021

ΔΗΜΟΚΡΙΤΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΡΑΚΗΣ

# Περιεχόμενα I

# Περιεχόμενα II

## Λίγα λόγια για τη μέθοδο

Η μέθοδος Newton είναι μια καθιερωμένη μεθόδος αριθμητικής προσέγγισης λύσεων της συνάρτησης $F_n(x) = \mathcal{O}_n$, όπου $\mathcal{O}_n$ δηλώνει το μηδενικό διάνυσμα.

Πρόκειται για μια γενίκευση της μεθόδου των Newton-Raphson για συναρτήσεις πολλών μεταβλητών.

Για την παραγωγή του επαναληπτικού σχήματος αξιοποιούμε και πάλι το ανάπτυγμα Taylor, αυτή τη φορά για συναρτήσεις πολλών μεταβλητών.

Προσεγγίζει τις λύσεις που ψάχνουμε ανάγοντας την επίλυση ενός συστήματος μη-γραμμικών εξισώσεων σε ένα πρόβλημα επίλυσης γραμμικών συστημάτων.

Υπό προϋποθέσεις δύναται να συγκλίνει με γρήγορο ρυθμό προς την επιθυμητή προσέγγιση με μια δεδομένη ακρίβεια.

# Η μέθοδος Newton

Υποθέτουμε ότι $F_n \in C^n$ και επιπλέον, ότι $x^i \in D_n \subset \mathbb{R}^n$ είναι μια προσέγγιση της ρίζας, έστω $\xi$.

Εφόσον η συνάρτηση $F_n$ είναι συνεχώς παραγωγίσιμη κοντά στη ρίζα $\xi$, τότε για κάθε $x^i$, μπορούμε να ορίσουμε το Ιακωβιανό του μητρώο, ως εξής:

$$J_{F_n}(x^i) = \begin{pmatrix} \frac{\partial f_1(x^i)}{\partial x_1} & \frac{\partial f_1(x^i)}{\partial x_2} & \dots & \frac{\partial f_1(x^i)}{\partial x_n} \\ \frac{\partial f_2(x^i)}{\partial x_1} & \frac{\partial f_2(x^i)}{\partial x_2} & \dots & \frac{\partial f_2(x^i)}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n(x^i)}{\partial x_1} & \frac{\partial f_n(x^i)}{\partial x_2} & \dots & \frac{\partial f_n(x^i)}{\partial x_n} \end{pmatrix}$$

όπου $x^i = (x_1^i, x_2^i, \dots, x_n^i)^\top$ η $i-$οστή προσέγγιση της λύσης.

ΔΗΜΟΚΡΙΤΕΙΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΘΡΑΚΗΣ

# Η μέθοδος Newton

Το γραμμικό σύστημα που επιλύει επαναληπτικά είναι της μορφής $Ax = b$ και περιγράφεται από την ακόλουθη σχέση:

$$J_{F_n}(x^i)s^i = -F_n(x^i), \quad i = 0, 1, 2, \ldots$$

όπου $s^i = (s_1^i, s_2^i, \ldots, s_n^i)^\top$ δηλώνει τη διόρθωση της προσέγγισης στην $i$ επανάληψη και $F_n(x^i) = (f_1(x^i), f_2(x^i), \ldots, f_n(x^i))^\top$ τη συναρτησιακή τιμή της $F_n$ στην $i$ επανάληψη.

Επιλύοντας, το γραμμικό σύστημα ως προς τη διόρθωση $s^i$ και κάνοντας αντικατάσταση στην επαναληπτική σχέση

$$x^{i+1} = x^i + s^i, \quad i = 0, 1, 2, 3, \ldots$$

παίρνουμε την προσέγγιση της λύσης στην $i + 1$ επανάληψη, με $x^0 = (x_1^0, x_2^0, \ldots, x_n^0)^\top$ να παριστάνει μια αρχική προσέγγιση (αυθαίρετα επιλεγμένη).

ΔΗΜΟΚΡΙΤΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΡΑΚΗΣ

# Επαναληπτικό σχήμα

Θεωρώντας, το γραμμικό σύστημα που επιλύει επαναληπτικά η μέθοδος Newton και κάνοντας αντικατάσταση τη διόρθωση, παίρνουμε:

$$J_{F_n}(x^i)(x^{i+1} - x^i) = -F_n(x^i), \quad i = 0, 1, 2, \dots$$

Πολλαπλασιάζοντας, την τελευταία σχέση από αριστερά με το αντίστροφο του Ιακωβιανού μητρώου, παίρνουμε

$$J_{F_n}^{-1} J_{F_n}(x^i)(x^{i+1} - x^i) = -J_{F_n}^{-1} F_n(x^i)$$

Ισοδύναμα, και εκτελώντας τις σχετικές απλοποιήσεις, έχουμε:

$$x^{i+1} = x^i - J_{F_n}^{-1} F_n(x^i), \quad i = 0, 1, 2, 3, \dots$$

το οποίο αποτελεί το επαναληπτικό σχήμα της μεθόδου του Newton.

# Ερμηνεία

Γεωμετρικά, λαμβάνοντας υπόψιν την πληροφορία της προσέγγισης στην $i$ επανάληψη, η μέθοδος Newton υπολογίζει το σημείο τομής των επιπέδων που εφάπτονται στις επιφάνειες

$$x_{n+1} = f_j(x), \quad j = 1, 2, \ldots, n$$

στα σημεία

$$\{(x_1^i, x_2^i, \ldots, x_n^i, f_j(x^i))\}, \quad j = 1, 2, \ldots, n$$

με το επίπεδο

$$x_{n+1} = 0$$

# Αλγόριθμος υλοποίησης

1. Δίνουμε στον αλγόριθμο τους πόρους που χρειάζεται για να εκκινήσει η διαδικασία: Είσοδος $= \{F_n, J_{F_n}, x^0, \varepsilon_1, \varepsilon_2, max\}$
2. Θέτουμε $i = -1$.
3. Αν $i \leq max$, κάνε το $i$ ίσο με $i + 1$ και πήγαινε στο βήμα 4, αλλιώς δώσε την Έξοδο στο βήμα 8.
4. Ελέγχουμε αν $||F_n(x^i)|| \leq \varepsilon_1$. Αν ναι, πήγαινε στο βήμα 8 και δώσε την έξοδο, αλλιώς πήγαινε στο βήμα 5.
5. Επιλύουμε το γραμμικό σύστημα $J_{F_n}(x^i)s^i = -F_n(x^i)$ ως προς τη διόρθωση και πήγαινε στο βήμα 6.
6. Ελέγχουμε αν $||s^i|| \leq \varepsilon_2$. Αν ισχύει πήγαινε στο βήμα 8, αλλιώς πήγαινε στο βήμα 7.
7. Θέτουμε $x^{i+1} = x^i + s^i$ και πήγαινε στο βήμα 3.
8. Δώσε την έξοδο του αλγορίθμου: Έξοδος $= \{i, x^i, F_n(x^i)\}$

ΔΗΜΟΚΡΙΤΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΡΑΚΗΣ

# Κριτήρια τερματισμού του αλγορίθμου

- Τυπικά δε θα ελέγχουμε εάν $F_n(x^i) = 0$.
- Αντί αυτού του ελέγχου, εφόσον ζητείται συγκεκριμένη ακρίβεια $\varepsilon$, μπορούμε να ελέγχουμε ένα από τα ακόλουθα κριτήρια:
  1. $||x^i - x^{i-1}|| < \varepsilon$
  2. $||F(x^i)|| < \varepsilon$
  3. $\dfrac{||x^i - x^{i-1}||}{||x^i||} < \varepsilon$
  4. $||s^i|| < \varepsilon$
  5. $||s^i - s^{i-1}|| < \varepsilon$

# Πλεονεκτήματα - Μειονεκτήματα

- Η μέθοδος απαιτεί μόνο μία αρχική προσέγγιση για την εκκίνηση του επαναληπτικού σχήματος.
- Εφόσον η συνάρτηση πληροί κάποιες προϋποθέσεις συγκλίνει με τετραγωνικό ρυθμό.
- Είναι σχετικά απλή και εύκολα ερμηνεύσιμη με απλό τρόπο υλοποίησης.
- Είναι μέθοδος τοπικής σύγκλισης, καθώς παρουσιάζει μεγάλη ευαισθησία στην αρχική προσέγγιση της λύσης.
- Δε συγκλίνει πάντα.
- Απαιτεί πληροφορία υπολογισμού που είναι δαπανηρή (μερικές παράγωγοι) ή και αδύνατο να υπολογιστεί σε κάποιες περιπτώσεις.
- Απαιτεί συνολικά $n^2 + n$ συναρτησιακούς υπολογισμούς σε κάθε επανάληψη.

## Άσκηση

Να προσεγγισθεί με τη μέθοδο του Newton η ρίζα της συνάρτησης

$$F(x,y) = \begin{pmatrix} 4x^2 - y^3 + 28 \\ 3x^3 + 4y^2 - 145 \end{pmatrix}$$

με αρχική προσέγγιση $x = 1, y = 1$ για 1 επανάληψη

## Άσκηση

Να προσεγγισθεί με τη μέθοδο του Newton η ρίζα της συνάρτησης

$$F(x,y) = \begin{pmatrix} 4x^2 - y^3 + 28 \\ 3x^3 + 4y^2 - 145 \end{pmatrix}$$

με αρχική προσέγγιση $x = 1, y = 1$ για 1 επανάληψη

- Ο επαναληπτικός τύπος της μεθόδου είναι

$$z^{k+1} = z^k - J_{F_n}(z^k)^{-1} F_n(z^k) \quad z^k = (x^k, y^k)$$

## Άσκηση

Να προσεγγισθεί με τη μέθοδο του Newton η ρίζα της συνάρτησης

$$F(x,y) = \begin{pmatrix} 4x^2 - y^3 + 28 \\ 3x^3 + 4y^2 - 145 \end{pmatrix}$$

με αρχική προσέγγιση $x = 1, y = 1$ για 1 επανάληψη

- Ο επαναληπτικός τύπος της μεθόδου είναι

$$z^{k+1} = z^k - J_{F_n}(z^k)^{-1} F_n(z^k) \quad z^k = (x^k, y^k)$$

- Αρχικά θα υπολογίσουμε το Ιακωβιανό μητρώο $J_{F_n}(z^k)$

$$J_{F_n}(x^k) = \begin{pmatrix} \frac{\partial f_1}{\partial x} & \frac{\partial f_1}{\partial y} \\ \frac{\partial f_2}{\partial x} & \frac{\partial f_2}{\partial y} \end{pmatrix} \quad \begin{pmatrix} \quad & \quad \\ \quad & \quad \end{pmatrix}$$

### Άσκηση

Να προσεγγισθεί με τη μέθοδο του Newton η ρίζα της συνάρτησης

$$F(x,y) = \begin{pmatrix} 4x^2 - y^3 + 28 \\ 3x^3 + 4y^2 - 145 \end{pmatrix}$$

με αρχική προσέγγιση $x = 1, y = 1$ για 1 επανάληψη

- Ο επαναληπτικός τύπος της μεθόδου είναι

$$z^{k+1} = z^k - J_{F_n}(z^k)^{-1} F_n(z^k) \quad z^k = (x^k, y^k)$$

- Αρχικά θα υπολογίσουμε το Ιακωβιανό μητρώο $J_{F_n}(z^k)$

$$J_{F_n}(x^k) = \begin{pmatrix} \frac{\partial f_1}{\partial x} & \frac{\partial f_1}{\partial y} \\ \frac{\partial f_2}{\partial x} & \frac{\partial f_2}{\partial y} \end{pmatrix} = \begin{pmatrix} 8x & \\ & \end{pmatrix}$$

ΔΗΜΟΚΡΙΤΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΡΑΚΗΣ

## Άσκηση

Να προσεγγισθεί με τη μέθοδο του Newton η ρίζα της συνάρτησης

$$F(x,y) = \begin{pmatrix} 4x^2 - y^3 + 28 \\ 3x^3 + 4y^2 - 145 \end{pmatrix}$$

με αρχική προσέγγιση $x = 1, y = 1$ για 1 επανάληψη

- Ο επαναληπτικός τύπος της μεθόδου είναι

$$z^{k+1} = z^k - J_{F_n}(z^k)^{-1} F_n(z^k) \quad z^k = (x^k, y^k)$$

- Αρχικά θα υπολογίσουμε το Ιακωβιανό μητρώο $J_{F_n}(z^k)$

$$J_{F_n}(x^k) = \begin{pmatrix} \frac{\partial f_1}{\partial x} & \frac{\partial f_1}{\partial y} \\ \frac{\partial f_2}{\partial x} & \frac{\partial f_2}{\partial y} \end{pmatrix} = \begin{pmatrix} 8x & -3y^2 \\ & \end{pmatrix}$$

## Άσκηση

Να προσεγγισθεί με τη μέθοδο του Newton η ρίζα της συνάρτησης

$$F(x,y) = \begin{pmatrix} 4x^2 - y^3 + 28 \\ 3x^3 + 4y^2 - 145 \end{pmatrix}$$

με αρχική προσέγγιση $x = 1, y = 1$ για 1 επανάληψη

- Ο επαναληπτικός τύπος της μεθόδου είναι
$$z^{k+1} = z^k - J_{F_n}(z^k)^{-1} F_n(z^k) \quad z^k = (x^k, y^k)$$
- Αρχικά θα υπολογίσουμε το Ιακωβιανό μητρώο $J_{F_n}(z^k)$
$$J_{F_n}(x^k) = \begin{pmatrix} \frac{\partial f_1}{\partial x} & \frac{\partial f_1}{\partial y} \\ \frac{\partial f_2}{\partial x} & \frac{\partial f_2}{\partial y} \end{pmatrix} = \begin{pmatrix} 8x & -3y^2 \\ 9x^2 & \end{pmatrix}$$

## Άσκηση

Να προσεγγισθεί με τη μέθοδο του Newton η ρίζα της συνάρτησης

$$F(x,y) = \begin{pmatrix} 4x^2 - y^3 + 28 \\ 3x^3 + 4y^2 - 145 \end{pmatrix}$$

με αρχική προσέγγιση $x = 1, y = 1$ για 1 επανάληψη

- Ο επαναληπτικός τύπος της μεθόδου είναι

$$z^{k+1} = z^k - J_{F_n}(z^k)^{-1} F_n(z^k) \quad z^k = (x^k, y^k)$$

- Αρχικά θα υπολογίσουμε το Ιακωβιανό μητρώο $J_{F_n}(z^k)$

$$J_{F_n}(x^k) = \begin{pmatrix} \frac{\partial f_1}{\partial x} & \frac{\partial f_1}{\partial y} \\ \frac{\partial f_2}{\partial x} & \frac{\partial f_2}{\partial y} \end{pmatrix} = \begin{pmatrix} 8x & -3y^2 \\ 9x^2 & 8y \end{pmatrix}$$

- Σαν κριτήριο τερματισμού, θα χρησιμοποιήσω τη διαφορά δύο διαδοχικών προσεγγίσεων να είναι μικρότερη από 0.001

$$\|x^{k+1} - x^k\| < \epsilon$$

- Σαν κριτήριο τερματισμού, θα χρησιμοποιήσω τη διαφορά δύο διαδοχικών προσεγγίσεων να είναι μικρότερη από 0.001

$$\|x^{k+1} - x^{k}\| < \epsilon$$

- Ελέγχω αν $F(1,1) = (0,0)^T$,

- Σαν κριτήριο τερματισμού, θα χρησιμοποιήσω τη διαφορά δύο διαδοχικών προσεγγίσεων να είναι μικρότερη από 0.001

$$\|x^{k+1} - x^k\| < \epsilon$$

- Ελέγχω αν $F(1,1) = (0,0)^T$, παρατηρώ ότι $F(1,1) = (31, -138)^T$

- Σαν κριτήριο τερματισμού, θα χρησιμοποιήσω τη διαφορά δύο διαδοχικών προσεγγίσεων να είναι μικρότερη από 0.001

$$\|x^{k+1} - x^k\| < \epsilon$$

- Ελέγχω αν $F(1,1) = (0,0)^T$, παρατηρώ ότι $F(1,1) = (31,-138)^T$
- Ξεκινώ με την πρώτη επανάληψη $k = 0, x^0 = (1,1)^T$

- Σαν κριτήριο τερματισμού, θα χρησιμοποιήσω τη διαφορά δύο διαδοχικών προσεγγίσεων να είναι μικρότερη από 0.001

$$\|x^{k+1} - x^k\| < \epsilon$$

- Ελέγχω αν $F(1,1) = (0,0)^T$, παρατηρώ ότι $F(1,1) = (31, -138)^T$
- Ξεκινώ με την πρώτη επανάληψη $k = 0, x^0 = (1,1)^T$
    1. Υπολογίζω το Ιακωβιανό μητρώο

$$J_{F_n}(1,1) = \begin{pmatrix} 8 & -3 \\ 9 & 8 \end{pmatrix}$$

- Σαν κριτήριο τερματισμού, θα χρησιμοποιήσω τη διαφορά δύο διαδοχικών προσεγγίσεων να είναι μικρότερη από 0.001

$$\|x^{k+1} - x^k\| < \epsilon$$

- Ελέγχω αν $F(1,1) = (0,0)^T$, παρατηρώ ότι $F(1,1) = (31, -138)^T$
- Ξεκινώ με την πρώτη επανάληψη $k = 0, x^0 = (1,1)^T$
  1. Υπολογίζω το Ιακωβιανό μητρώο

$$J_{F_n}(1,1) = \begin{pmatrix} 8 & -3 \\ 9 & 8 \end{pmatrix}$$

  2. Υπολογίζω το αντίστροφο του Ιακωβιανού μητρώου

$$J_{F_n}^{-1}(1,1) = \begin{pmatrix} 0.0879 & 0.0330 \\ -0.0989 & 0.0879 \end{pmatrix}$$

- Σαν κριτήριο τερματισμού, θα χρησιμοποιήσω τη διαφορά δύο διαδοχικών προσεγγίσεων να είναι μικρότερη από 0.001

$$\|x^{k+1} - x^k\| < \epsilon$$

- Ελέγχω αν $F(1,1) = (0,0)^T$, παρατηρώ ότι $F(1,1) = (31, -138)^T$
- Ξεκινώ με την πρώτη επανάληψη $k = 0, x^0 = (1,1)^T$
  1. Υπολογίζω το Ιακωβιανό μητρώο

$$J_{F_n}(1,1) = \begin{pmatrix} 8 & -3 \\ 9 & 8 \end{pmatrix}$$

  2. Υπολογίζω το αντίστροφο του Ιακωβιανού μητρώου

$$J_{F_n}^{-1}(1,1) = \begin{pmatrix} 0.0879 & 0.0330 \\ -0.0989 & 0.0879 \end{pmatrix}$$

- Υπολογίζω την επόμενη προσέγγιση μέσω του τύπου

$$x^1 = x^0 - J_{F_n}^{-1}(x^0) F_n(x^0)$$

$$x^1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix} - \begin{pmatrix} 0.0879 & 0.0330 \\ -0.0989 & 0.0879 \end{pmatrix} \cdot \begin{pmatrix} 31 \\ -138 \end{pmatrix}$$

$$x^1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix} - \begin{pmatrix} 0.0879 & 0.0330 \\ -0.0989 & 0.0879 \end{pmatrix} \cdot \begin{pmatrix} 31 \\ -138 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix} - \begin{pmatrix} -1.8242 \\ -15.1978 \end{pmatrix}$$

$$x^1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix} - \begin{pmatrix} 0.0879 & 0.0330 \\ -0.0989 & 0.0879 \end{pmatrix} \cdot \begin{pmatrix} 31 \\ -138 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix} - \begin{pmatrix} -1.8242 \\ -15.1978 \end{pmatrix} = \begin{pmatrix} 2.8242 \\ 16.1978 \end{pmatrix}$$

$$x^1 = \begin{pmatrix}1\\1\end{pmatrix} - \begin{pmatrix}0.0879 & 0.0330\\-0.0989 & 0.0879\end{pmatrix} \cdot \begin{pmatrix}31\\-138\end{pmatrix} = \begin{pmatrix}1\\1\end{pmatrix} - \begin{pmatrix}-1.8242\\-15.1978\end{pmatrix} = \begin{pmatrix}2.8242\\16.1978\end{pmatrix}$$

- Ελέγχω το κριτήριο τερματισμού

$$\|x^1 - x^0\| = \left\| \begin{pmatrix}2.8242\\16.1978\end{pmatrix} - \begin{pmatrix}1\\1\end{pmatrix} \right\| =$$

$$x^1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix} - \begin{pmatrix} 0.0879 & 0.0330 \\ -0.0989 & 0.0879 \end{pmatrix} \cdot \begin{pmatrix} 31 \\ -138 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix} - \begin{pmatrix} -1.8242 \\ -15.1978 \end{pmatrix} = \begin{pmatrix} 2.8242 \\ 16.1978 \end{pmatrix}$$

- Ελέγχω το κριτήριο τερματισμού

$$\|x^1 - x^0\| = \left\| \begin{pmatrix} 2.8242 \\ 16.1978 \end{pmatrix} - \begin{pmatrix} 1 \\ 1 \end{pmatrix} \right\| = \left\| \begin{pmatrix} 1.8242 \\ 15.1978 \end{pmatrix} \right\|$$

$$x^1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix} - \begin{pmatrix} 0.0879 & 0.0330 \\ -0.0989 & 0.0879 \end{pmatrix} \cdot \begin{pmatrix} 31 \\ -138 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix} - \begin{pmatrix} -1.8242 \\ -15.1978 \end{pmatrix} = \begin{pmatrix} 2.8242 \\ 16.1978 \end{pmatrix}$$

- Ελέγχω το κριτήριο τερματισμού

$$\|x^1 - x^0\| = \left\| \begin{pmatrix} 2.8242 \\ 16.1978 \end{pmatrix} - \begin{pmatrix} 1 \\ 1 \end{pmatrix} \right\| = \left\| \begin{pmatrix} 1.8242 \\ 15.1978 \end{pmatrix} \right\| = 15.3069 > 0.001$$

# Να δημιουργήσετε συνάρτηση στο MATLAB για τη μέθοδο Newton

# Να δημιουργήσετε συνάρτηση στο MATLAB για τη μέθοδο Newton

- Θέλουμε συνάρτηση άρα θα έχουμε function m-file

```
function [επιστρεφόμενες τιμές] = όνομα(ορίσματα)
Εντολές
end
```

# Να δημιουργήσετε συνάρτηση στο MATLAB για τη μέθοδο Newton

- Θέλουμε συνάρτηση άρα θα έχουμε function m-file

  function [επιστρεφόμενες τιμές] = όνομα(ορίσματα)
  Εντολές
  end

- ορίσματα

| | | | | |
|---|---|---|---|---|
| | | | | |
| | | | | |

# Να δημιουργήσετε συνάρτηση στο MATLAB για τη μέθοδο Newton

- Θέλουμε συνάρτηση άρα θα έχουμε function m-file

  function [επιστρεφόμενες τιμές] = όνομα(ορίσματα)
  Εντολές
  end

- ορίσματα

| | | | | |
|---|---|---|---|---|
| αρχικό σημείο | | | | |
| $p0$ | | | | |

# Να δημιουργήσετε συνάρτηση στο MATLAB για τη μέθοδο Newton

- Θέλουμε συνάρτηση άρα θα έχουμε function m-file

  function [επιστρεφόμενες τιμές] = όνομα(ορίσματα)
  Εντολές
  end

- ορίσματα

| αρχικό σημείο | επαναλήψεις | | | |
|---|---|---|---|---|
| *p0* | *maxiter* | | | |

# Να δημιουργήσετε συνάρτηση στο MATLAB για τη μέθοδο Newton

- Θέλουμε συνάρτηση άρα θα έχουμε function m-file

  function [επιστρεφόμενες τιμές] = όνομα(ορίσματα)
  Εντολές
  end

- ορίσματα

| αρχικό σημείο | επαναλήψεις | δεκαδικά | | |
|---|---|---|---|---|
| *p0* | *maxiter* | *acc* | | |

# Να δημιουργήσετε συνάρτηση στο MATLAB για τη μέθοδο Newton

- Θέλουμε συνάρτηση άρα θα έχουμε function m-file

function [επιστρεφόμενες τιμές] = όνομα(ορίσματα)
Εντολές
end

- ορίσματα

| αρχικό σημείο | επαναλήψεις | δεκαδικά | συνάρτηση | |
|---|---|---|---|---|
| *p0* | *maxiter* | *acc* | *Func* | |

# Να δημιουργήσετε συνάρτηση στο MATLAB για τη μέθοδο Newton

- Θέλουμε συνάρτηση άρα θα έχουμε function m-file

  function [επιστρεφόμενες τιμές] = όνομα(ορίσματα)
  Εντολές
  end

- ορίσματα

| αρχικό σημείο | επαναλήψεις | δεκαδικά | συνάρτηση | Ιακωβιανό μητρώο |
|---|---|---|---|---|
| *p0* | *maxiter* | *acc* | *Func* | *Jac* |

# Να δημιουργήσετε συνάρτηση στο MATLAB για τη μέθοδο Newton

- Θέλουμε συνάρτηση άρα θα έχουμε function m-file

  function [επιστρεφόμενες τιμές] = όνομα(ορίσματα)
  Εντολές
  end

- ορίσματα

| αρχικό σημείο | επαναλήψεις | δεκαδικά | συνάρτηση | Ιακωβιανό μητρώο |
|---|---|---|---|---|
| $p0$ | $maxiter$ | $acc$ | $Func$ | $Jac$ |

- επιστρεφόμενες τιμές

| | |
|---|---|
| | |

# Να δημιουργήσετε συνάρτηση στο MATLAB για τη μέθοδο Newton

- Θέλουμε συνάρτηση άρα θα έχουμε function m-file

```
function [επιστρεφόμενες τιμές] = όνομα(ορίσματα)
Εντολές
end
```

- ορίσματα

| αρχικό σημείο | επαναλήψεις | δεκαδικά | συνάρτηση | Ιακωβιανό μητρώο |
|---|---|---|---|---|
| *p0* | *maxiter* | *acc* | *Func* | *Jac* |

- επιστρεφόμενες τιμές

| | |
|---|---|
| ρίζα | |
| *res_newton* | |

ΔΗΜΟΚΡΙΤΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΡΑΚΗΣ

# Να δημιουργήσετε συνάρτηση στο MATLAB για τη μέθοδο Newton

- Θέλουμε συνάρτηση άρα θα έχουμε function m-file

```
function [επιστρεφόμενες τιμές] = όνομα(ορίσματα)
Εντολές
end
```

- ορίσματα

| αρχικό σημείο | επαναλήψεις | δεκαδικά | συνάρτηση | Ιακωβιανό μητρώο |
|---|---|---|---|---|
| *p0* | *maxiter* | *acc* | *Func* | *Jac* |

- επιστρεφόμενες τιμές

| ρίζα | πλήθος επαναλήψεων |
|---|---|
| *res_newton* | *tot_iter* |

function $[res\_newton, tot\_iter] = \text{newton}(p0, maxiter, acc, Func, Jac)$
Εντολές
end

function $[res\_newton, tot\_iter] = \text{newton}(p0, maxiter, acc, Func, Jac)$
Εντολές
end

- Αρχικοποίηση με βάση την είσοδο

| | |
|---|---|
| | |

function $[res\_newton, tot\_iter] = \text{newton}(p0, maxiter, acc, Func, Jac)$
Εντολές
end

- Αρχικοποίηση με βάση την είσοδο

| | |
|---|---|
| ρίζα | |

function $[res\_newton, tot\_iter] = \text{newton}(p0, maxiter, acc, Func, Jac)$
Εντολές
end

- Αρχικοποίηση με βάση την είσοδο

| | |
|---|---|
| ρίζα | $res\_newton = X0;$ |

function $[res\_newton, tot\_iter] = \text{newton}(p0, maxiter, acc, Func, Jac)$
Εντολές
end

- Αρχικοποίηση με βάση την είσοδο

| | |
|---|---|
| ρίζα<br>ακρίβεια | $res\_newton = X0;$ |

function $[res\_newton, tot\_iter] = \text{newton}(p0, maxiter, acc, Func, Jac)$
Εντολές
end

- Αρχικοποίηση με βάση την είσοδο

| | |
|---|---|
| ρίζα | $res\_newton = X0;$ |
| ακρίβεια | $accur = 0.5 * 10\hat{(} - acc);$ |

function $[res\_newton, tot\_iter] = \text{newton}(p0, maxiter, acc, Func, Jac)$
Εντολές
end

- Αρχικοποίηση με βάση την είσοδο

| | |
|---|---|
| ρίζα<br>ακρίβεια<br>αρχικοποίηση μετρητή επαναλήψεων | $res\_newton = X0;$<br>$accur = 0.5 * 10\hat{(} - acc);$ |

function $[res\_newton, tot\_iter] = \text{newton}(p0, maxiter, acc, Func, Jac)$
Εντολές
end

- Αρχικοποίηση με βάση την είσοδο

| | |
|---|---|
| ρίζα | $res\_newton = X0;$ |
| ακρίβεια | $accur = 0.5 * 10\hat{(} - acc);$ |
| αρχικοποίηση μετρητή επαναλήψεων | $k = 0;$ |

function $[res\_newton, tot\_iter]$ = newton($p0, maxiter, acc, Func, Jac$)
Εντολές
end

- Αρχικοποίηση με βάση την είσοδο

| | |
|---|---|
| ρίζα<br>ακρίβεια<br>αρχικοποίηση μετρητή επαναλήψεων<br>δείκτης για τερματισμό | $res\_newton = X0;$<br>$accur = 0.5 * 10\hat{(} - acc);$<br>$k = 0;$ |

function $[res\_newton, tot\_iter]$ = newton($p0, maxiter, acc, Func, Jac$)
Εντολές
end

- Αρχικοποίηση με βάση την είσοδο

| | |
|---|---|
| ρίζα | $res\_newton = X0;$ |
| ακρίβεια | $accur = 0.5 * 10\hat{(} - acc);$ |
| αρχικοποίηση μετρητή επαναλήψεων | $k = 0;$ |
| δείκτης για τερματισμό | $stop = false;$ |

- Η μέθοδος τερματίζει όταν ισχύει κάποιο κριτήριο τερματισμού ή φτάσουμε το μέγιστο πλήθος επαναλήψεων

- Η μέθοδος τερματίζει όταν ισχύει κάποιο κριτήριο τερματισμού ή φτάσουμε το μέγιστο πλήθος επαναλήψεων
- Χρειαζόμαστε τη δομή while

$$while(stop == false)$$

$$end$$

Όσο ισχύει η συνθήκη,

- Η μέθοδος τερματίζει όταν ισχύει κάποιο κριτήριο τερματισμού ή φτάσουμε το μέγιστο πλήθος επαναλήψεων
- Χρειαζόμαστε τη δομή while
- Υπολογισμός νέας προσέγγισης

$$
\begin{array}{l}
\mathit{while}(\mathit{stop} == \mathit{false}) \\
\quad k = k + 1; \\
\\
\mathit{end}
\end{array}
$$

Όσο ισχύει η συνθήκη, ανανέωσε επαναλήψεις

- Η μέθοδος τερματίζει όταν ισχύει κάποιο κριτήριο τερματισμού ή φτάσουμε το μέγιστο πλήθος επαναλήψεων
- Χρειαζόμαστε τη δομή while
- Υπολογισμός νέας προσέγγισης

```
while(stop == false)
    k = k + 1;
    F = feval(Func, X0);

end
```

Όσο ισχύει η συνθήκη, ανανέωσε επαναλήψεις υπολόγισε τιμή συνάρτησης

- Η μέθοδος τερματίζει όταν ισχύει κάποιο κριτήριο τερματισμού ή φτάσουμε το μέγιστο πλήθος επαναλήψεων
- Χρειαζόμαστε τη δομή while
- Υπολογισμός νέας προσέγγισης

```
while(stop == false)
    k = k + 1;
    F = feval(Func, X0);
    Jacob = feval(Jacobian, X0);

end
```

Όσο ισχύει η συνθήκη, ανανέωσε επαναλήψεις υπολόγισε τιμή συνάρτησης υπολόγισε τιμή ιακωβιανού

- Η μέθοδος τερματίζει όταν ισχύει κάποιο κριτήριο τερματισμού ή φτάσουμε το μέγιστο πλήθος επαναλήψεων
- Χρειαζόμαστε τη δομή while
- Υπολογισμός νέας προσέγγισης

$$
\begin{array}{l}
\textit{while}(\textit{stop} == \textit{false}) \\
\quad k = k + 1; \\
\quad F = \textit{feval}(\textit{Func}, X0); \\
\quad \textit{Jacob} = \textit{feval}(\textit{Jacobian}, X0); \\
\quad \textit{Sp} = -\textit{Jacob} \backslash F; \\
\\
\textit{end}
\end{array}
$$

Όσο ισχύει η συνθήκη,
ανανέωσε επαναλήψεις
υπολόγισε τιμή συνάρτησης
υπολόγισε τιμή ιακωβιανού
επίλυση συστήματος

- Η μέθοδος τερματίζει όταν ισχύει κάποιο κριτήριο τερματισμού ή φτάσουμε το μέγιστο πλήθος επαναλήψεων
- Χρειαζόμαστε τη δομή while
- Υπολογισμός νέας προσέγγισης

$$
\begin{array}{l}
\textit{while}(\textit{stop} == \textit{false}) \\
\quad k = k + 1; \\
\quad F = \textit{feval}(\textit{Func}, X0); \\
\quad \textit{Jacob} = \textit{feval}(\textit{Jacobian}, X0); \\
\quad \textit{Sp} = -\textit{Jacob} \backslash F; \\
\quad \textit{Sp} = \textit{Sp}'; \\
\\
\textit{end}
\end{array}
$$

Όσο ισχύει η συνθήκη,
ανανέωσε επαναλήψεις
υπολόγισε τιμή συνάρτησης
υπολόγισε τιμή ιακωβιανού
επίλυση συστήματος
μετατροπή στον ανάστροφο

- Η μέθοδος τερματίζει όταν ισχύει κάποιο κριτήριο τερματισμού ή φτάσουμε το μέγιστο πλήθος επαναλήψεων
- Χρειαζόμαστε τη δομή while
- Υπολογισμός νέας προσέγγισης

| | |
|---|---|
| $while(stop == false)$ | Όσο ισχύει η συνθήκη, |
| $k = k + 1;$ | ανανέωσε επαναλήψεις |
| $F = feval(Func, X0);$ | υπολόγισε τιμή συνάρτησης |
| $Jacob = feval(Jacobian, X0);$ | υπολόγισε τιμή ιακωβιανού |
| $Sp = -Jacob \backslash F;$ | επίλυση συστήματος |
| $Sp = Sp';$ | μετατροπή στον ανάστροφο |
| $X1 = X0 + Sp;$ | εύρεση νέου σημείου |
| $end$ | |

- Η μέθοδος τερματίζει όταν ισχύει κάποιο κριτήριο τερματισμού ή φτάσουμε το μέγιστο πλήθος επαναλήψεων
- Χρειαζόμαστε τη δομή while
- Υπολογισμός νέας προσέγγισης
- Έλεγχος κριτηρίου τερματισμού

```
while(stop == false)
    k = k + 1;
    F = feval(Func, X0);
    Jacob = feval(Jacobian, X0);
    Sp = -Jacob\F;
    Sp = Sp';
    X1 = X0 + Sp;
    δομή if
end
```

Όσο ισχύει η συνθήκη,
ανανέωσε επαναλήψεις
υπολόγισε τιμή συνάρτησης
υπολόγισε τιμή ιακωβιανού
επίλυση συστήματος
μετατροπή στον ανάστροφο
εύρεση νέου σημείου

- Έλεγχος κριτηρίου τερματισμού

- Έλεγχος κριτηρίου τερματισμού
- θα χρησιμοποιήσουμε

$$\|F\|_2 \leq \epsilon$$

$$\|Sp\|_2 \leq \epsilon$$

- Έλεγχος κριτηρίου τερματισμού
- θα χρησιμοποιήσουμε

$$\|F\|_2 \leq \epsilon$$

$$\|Sp\|_2 \leq \epsilon$$

- Έλεγχος αν ισχύει κάποιο κριτήριο

```
if((                    )||(                    ))

elseif(          )

end
```

- Έλεγχος κριτηρίου τερματισμού
- θα χρησιμοποιήσουμε

$$\|F\|_2 \leq \epsilon$$

$$\|Sp\|_2 \leq \epsilon$$

- Έλεγχος αν ισχύει κάποιο κριτήριο

```
if((norm(F,2) <= accur)||(norm(Sp,2) <= accur))

elseif(                    )

end
```

- Έλεγχος κριτηρίου τερματισμού
- θα χρησιμοποιήσουμε

$$\|F\|_2 \leq \epsilon$$

$$\|Sp\|_2 \leq \epsilon$$

- Έλεγχος αν ισχύει κάποιο κριτήριο

$if((norm(F,2) <= accur)||(norm(Sp,2) <= accur))$
$\quad stop = true;$
$elseif(\qquad)$

$end$

Τερματισμός
από νόρμα

- Έλεγχος κριτηρίου τερματισμού
- θα χρησιμοποιήσουμε

$$\|F\|_2 \leq \epsilon$$

$$\|Sp\|_2 \leq \epsilon$$

- Έλεγχος αν ισχύει κάποιο κριτήριο

$$\begin{aligned}
&if((norm(F,2) <= accur)\|(norm(Sp,2) <= accur))\\
&\quad stop = true;\\
&elseif(k >= maxit)\\
&\\
&end
\end{aligned}$$

Τερματισμός από νόρμα

- Έλεγχος κριτηρίου τερματισμού
- θα χρησιμοποιήσουμε

$$\|F\|_2 \leq \epsilon$$

$$\|Sp\|_2 \leq \epsilon$$

- Έλεγχος αν ισχύει κάποιο κριτήριο

$$if((norm(F,2) <= accur)\|(norm(Sp,2) <= accur))$$
$$stop = true;$$
$$elseif(k >= maxit)$$
$$stop = true;$$
$$end$$

Τερματισμός από νόρμα

Τερματισμός από επαναλήψεις

- Έλεγχος κριτηρίου τερματισμού
- θα χρησιμοποιήσουμε

$$\|F\|_2 \leq \epsilon$$

$$\|Sp\|_2 \leq \epsilon$$

- Έλεγχος αν ισχύει κάποιο κριτήριο

$$
\begin{array}{l}
if((norm(F,2) <= accur)\|(norm(Sp,2) <= accur)) \\
\quad stop = true; \\
elseif(k >= maxit) \\
\quad stop = true; \\
end \\
X0 = X1;
\end{array}
$$

Τερματισμός από νόρμα
Τερματισμός από επαναλήψεις

- Έλεγχος κριτηρίου τερματισμού
- θα χρησιμοποιήσουμε

$$\|F\|_2 \leq \epsilon$$

$$\|Sp\|_2 \leq \epsilon$$

- Έλεγχος αν ισχύει κάποιο κριτήριο

```
if((norm(F,2) <= accur)||(norm(Sp,2) <= accur))
   stop = true;
elseif(k >= maxit)
   stop = true;
end
X0 = X1;
res_newton = [res_newton; X0];
```

Τερματισμός από νόρμα
Τερματισμός από επαναλήψεις

- Έλεγχος κριτηρίου τερματισμού
- θα χρησιμοποιήσουμε

$$\|F\|_2 \leq \epsilon$$

$$\|Sp\|_2 \leq \epsilon$$

- Έλεγχος αν ισχύει κάποιο κριτήριο

$$
\begin{array}{l}
if((norm(F,2) <= accur)\|(norm(Sp,2) <= accur)) \\
\quad stop = true; \\
elseif(k >= maxit) \\
\quad stop = true; \\
end \\
X0 = X1; \\
res\_newton = [res\_newton; X0];
\end{array}
$$

Τερματισμός από νόρμα

Τερματισμός από επαναλήψεις

έξω από τη δομή *while*

- Έλεγχος κριτηρίου τερματισμού
- θα χρησιμοποιήσουμε

$$\|F\|_2 \leq \epsilon$$

$$\|Sp\|_2 \leq \epsilon$$

- Έλεγχος αν ισχύει κάποιο κριτήριο

```
if((norm(F,2) <= accur)||(norm(Sp,2) <= accur))
   stop = true;
elseif(k >= maxit)
   stop = true;
end
X0 = X1;
res_newton = [res_newton; X0];
```

Τερματισμός από νόρμα
Τερματισμός από επαναλήψεις

έξω από τη δομή *while*

```
tot_iter = k;
```

## Άσκηση

Να εφαρμόσετε τη μέθοδο Newton για την επίλυση της συνάρτησης

$$F(x_1, x_2) = \begin{array}{l} X_1^3 - 3X_1X_2^2 - 1 \\ 3 * (X_1^2X_2 - X_2^3) \end{array}$$

### Άσκηση

Να εφαρμόσετε τη μέθοδο Newton για την επίλυση της συνάρτησης

$$F(x_1, x_2) = \begin{matrix} X_1^3 - 3X_1X_2^2 - 1 \\ 3*(X_1^2X_2 - X_2^3) \end{matrix}$$

- Θα χρειαστούμε ένα function m-file για τη συνάρτηση

```
function [έξοδος] = όνομα(είσοδος)
Εντολές
end
```

### Άσκηση

Να εφαρμόσετε τη μέθοδο Newton για την επίλυση της συνάρτησης

$$F(x_1, x_2) = \begin{array}{l} X_1^3 - 3X_1X_2^2 - 1 \\ 3 * (X_1^2 X_2 - X_2^3) \end{array}$$

- Θα χρειαστούμε ένα function m-file για τη συνάρτηση

  function [έξοδος] = όνομα(είσοδος)
  Εντολές
  end

- ορίσματα εισόδου και εξόδου

| | |
|---|---|
| | |

## Άσκηση

Να εφαρμόσετε τη μέθοδο Newton για την επίλυση της συνάρτησης

$$F(x_1, x_2) = \begin{array}{l} X_1^3 - 3X_1X_2^2 - 1 \\ 3 * (X_1^2 X_2 - X_2^3) \end{array}$$

- Θα χρειαστούμε ένα function m-file για τη συνάρτηση

```
function [έξοδος] = όνομα(είσοδος)
Εντολές
end
```

- ορίσματα εισόδου και εξόδου

| σημείο | |
|---|---|
| | |

## Άσκηση

Να εφαρμόσετε τη μέθοδο Newton για την επίλυση της συνάρτησης

$$F(x_1, x_2) = \begin{matrix} X_1^3 - 3X_1X_2^2 - 1 \\ 3 * (X_1^2X_2 - X_2^3) \end{matrix}$$

- Θα χρειαστούμε ένα function m-file για τη συνάρτηση

```
function [έξοδος]    = όνομα(είσοδος)
Εντολές
end
```

- ορίσματα εισόδου και εξόδου

| σημείο | |
|---|---|
| $x$ | |

ΔΗΜΟΚΡΙΤΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΡΑΚΗΣ

## Άσκηση

Να εφαρμόσετε τη μέθοδο Newton για την επίλυση της συνάρτησης

$$F(x_1, x_2) = \begin{array}{l} X_1^3 - 3X_1X_2^2 - 1 \\ 3 * (X_1^2 X_2 - X_2^3) \end{array}$$

- Θα χρειαστούμε ένα function m-file για τη συνάρτηση

  function [έξοδος] = όνομα(είσοδος)
  Εντολές
  end

- ορίσματα εισόδου και εξόδου

| σημείο | τιμή |
|---|---|
| $x$ | |

### Άσκηση

Να εφαρμόσετε τη μέθοδο Newton για την επίλυση της συνάρτησης

$$F(x_1, x_2) = \begin{matrix} X_1^3 - 3X_1X_2^2 - 1 \\ 3 * (X_1^2X_2 - X_2^3) \end{matrix}$$

- Θα χρειαστούμε ένα function m-file για τη συνάρτηση

```
function [έξοδος] = όνομα(είσοδος)
Εντολές
end
```

- ορίσματα εισόδου και εξόδου

| σημείο | τιμή |
|---|---|
| $x$ | $w$ |

### Άσκηση

Να εφαρμόσετε τη μέθοδο Newton για την επίλυση της συνάρτησης

$$F(x_1, x_2) = \begin{array}{l} X_1^3 - 3X_1X_2^2 - 1 \\ 3 * (X_1^2 X_2 - X_2^3) \end{array}$$

- Θα χρειαστούμε ένα function m-file για τη συνάρτηση

```
function [έξοδος] = όνομα(είσοδος)
Εντολές
end
```

- ορίσματα εισόδου και εξόδου

| σημείο | τιμή |
|---|---|
| $x$ | $w$ |

- όνομα συνάρτησης $f$

## Άσκηση

Να εφαρμόσετε τη μέθοδο Newton για την επίλυση της συνάρτησης

$$F(x_1, x_2) = \begin{matrix} X_1^3 - 3X_1X_2^2 - 1 \\ 3 * (X_1^2X_2 - X_2^3) \end{matrix}$$

- Θα χρειαστούμε ένα function m-file για τη συνάρτηση

  function [έξοδος] = όνομα(είσοδος)
  Εντολές
  end

- ορίσματα εισόδου και εξόδου

| σημείο | τιμή |
|---|---|
| $x$ | $w$ |

- όνομα συνάρτησης $f$
- Εντολές

### Άσκηση

Να εφαρμόσετε τη μέθοδο Newton για την επίλυση της συνάρτησης

$$F(x_1, x_2) = \begin{array}{l} X_1^3 - 3X_1X_2^2 - 1 \\ 3 * (X_1^2 X_2 - X_2^3) \end{array}$$

- Θα χρειαστούμε ένα function m-file για τη συνάρτηση

function [έξοδος] = όνομα(είσοδος)
Εντολές
end

- ορίσματα εισόδου και εξόδου

| σημείο | τιμή |
|---|---|
| $x$ | $w$ |

- όνομα συνάρτησης $f$
- Εντολές

## Άσκηση

Να εφαρμόσετε τη μέθοδο Newton για την επίλυση της συνάρτησης

$$F(x_1, x_2) = \begin{matrix} X_1^3 - 3X_1X_2^2 - 1 \\ 3*(X_1^2X_2 - X_2^3) \end{matrix}$$

- Θα χρειαστούμε ένα function m-file για τη συνάρτηση

  function [έξοδος] = όνομα(είσοδος)
  Εντολές
  end

- ορίσματα εισόδου και εξόδου

| σημείο | τιμή |
|---|---|
| $x$ | $w$ |

- όνομα συνάρτησης $f$
- Εντολές

| | |
|---|---|
| Υπολογισμος συναρτησιακής τιμής 1 | $f1 = (X(1)\hat{}3 - 3*X(1)*X(2)\hat{}2 - 1);$ |

## Άσκηση

Να εφαρμόσετε τη μέθοδο Newton για την επίλυση της συνάρτησης

$$F(x_1, x_2) = \begin{array}{l} X_1^3 - 3X_1X_2^2 - 1 \\ 3 * (X_1^2 X_2 - X_2^3) \end{array}$$

- Θα χρειαστούμε ένα function m-file για τη συνάρτηση

  function [έξοδος] = όνομα(είσοδος)
  Εντολές
  end

- ορίσματα εισόδου και εξόδου

| σημείο | τιμή |
|---|---|
| $x$ | $w$ |

- όνομα συνάρτησης $f$
- Εντολές

| | |
|---|---|
| Υπολογισμος συναρτησιακής τιμής 1<br>Υπολογισμος συναρτησιακής τιμής 2 | $f1 = (X(1)^3 - 3 * X(1) * X(2)^2 - 1);$ |

## Άσκηση

Να εφαρμόσετε τη μέθοδο Newton για την επίλυση της συνάρτησης

$$F(x_1, x_2) = \begin{array}{l} X_1^3 - 3X_1X_2^2 - 1 \\ 3 * (X_1^2 X_2 - X_2^3) \end{array}$$

- Θα χρειαστούμε ένα function m-file για τη συνάρτηση

function [έξοδος] = όνομα(είσοδος)
Εντολές
end

- ορίσματα εισόδου και εξόδου

| σημείο | τιμή |
|---|---|
| $x$ | $w$ |

- όνομα συνάρτησης $f$
- Εντολές

| | |
|---|---|
| Υπολογισμος συναρτησιακής τιμής 1 | $f1 = (X(1)^3 - 3 * X(1) * X(2)^2 - 1);$ |
| Υπολογισμος συναρτησιακής τιμής 2 | $f2 = (3 * (X(1)^2) * X(2) - X(2)^3)$ |

## Άσκηση

Να εφαρμόσετε τη μέθοδο Newton για την επίλυση της συνάρτησης

$$F(x_1, x_2) = \begin{matrix} X_1^3 - 3X_1X_2^2 - 1 \\ 3 * (X_1^2 X_2 - X_2^3) \end{matrix}$$

- Θα χρειαστούμε ένα function m-file για τη συνάρτηση

```
function [έξοδος] = όνομα(είσοδος)
Εντολές
end
```

- ορίσματα εισόδου και εξόδου

| σημείο | τιμή |
|---|---|
| $x$ | $w$ |

- όνομα συνάρτησης $f$
- Εντολές

| | |
|---|---|
| Υπολογισμος συναρτησιακής τιμής 1 | $f1 = (X(1)^3 - 3 * X(1) * X(2)^2 - 1);$ |
| Υπολογισμος συναρτησιακής τιμής 2 | $f2 = (3 * (X(1)^2) * X(2) - X(2)^3);$ |
| Δημιοργία ζεύγους | |

### Άσκηση

Να εφαρμόσετε τη μέθοδο Newton για την επίλυση της συνάρτησης

$$F(x_1, x_2) = \begin{array}{l} X_1^3 - 3X_1X_2^2 - 1 \\ 3 * (X_1^2 X_2 - X_2^3) \end{array}$$

- Θα χρειαστούμε ένα function m-file για τη συνάρτηση

  function [έξοδος] = όνομα(είσοδος)
  Εντολές
  end

- ορίσματα εισόδου και εξόδου

| σημείο | τιμή |
|---|---|
| $x$ | $w$ |

- όνομα συνάρτησης $f$
- Εντολές

| | |
|---|---|
| Υπολογισμος συναρτησιακής τιμής 1 | $f1 = (X(1)\hat{}3 - 3 * X(1) * X(2)\hat{}2 - 1);$ |
| Υπολογισμος συναρτησιακής τιμής 2 | $f2 = (3 * (X(1)\hat{}2) * X(2) - X(2)\hat{}3);$ |
| Δημιοργία ζεύγους | $w = [f1\ f2]';$ |

- Θα χρειαστούμε ένα function m-file για το Ιακωβιανό μητρώο της συνάρτησης

$$Jacob(X_1, X_2) = \begin{matrix} 3X_1^2 - 3X_2^2 & -6X_1X_2 \\ 6X_1X_2 & 3X_1^2 - 3X_2^2 \end{matrix}$$

function [έξοδος] = όνομα(είσοδος)
Εντολές
end

- Θα χρειαστούμε ένα function m-file για το Ιακωβιανό μητρώο της συνάρτησης

$$Jacob(X_1, X_2) = \begin{matrix} 3X_1^2 - 3X_2^2 & -6X_1X_2 \\ 6X_1X_2 & 3X_1^2 - 3X_2^2 \end{matrix}$$

function [έξοδος] = όνομα(είσοδος)
Εντολές
end

- ορίσματα εισόδου και εξόδου

| | |
|---|---|
| | |

- Θα χρειαστούμε ένα function m-file για το Ιακωβιανό μητρώο της συνάρτησης

$$Jacob(X_1, X_2) = \begin{matrix} 3X_1^2 - 3X_2^2 & -6X_1X_2 \\ 6X_1X_2 & 3X_1^2 - 3X_2^2 \end{matrix}$$

function [έξοδος] = όνομα(είσοδος)
Εντολές
end

- ορίσματα εισόδου και εξόδου

| τετμημένη | |
|---|---|
| | |

- Θα χρειαστούμε ένα function m-file για το Ιακωβιανό μητρώο της συνάρτησης

$$Jacob(X_1, X_2) = \begin{matrix} 3X_1^2 - 3X_2^2 & -6X_1X_2 \\ 6X_1X_2 & 3X_1^2 - 3X_2^2 \end{matrix}$$

```
function [έξοδος]    = όνομα(είσοδος)
Εντολές
end
```

- ορίσματα εισόδου και εξόδου

| τετμημένη | |
|---|---|
| $x$ | |

- Θα χρειαστούμε ένα function m-file για το Ιακωβιανό μητρώο της συνάρτησης

$$Jacob(X_1, X_2) = \begin{matrix} 3X_1^2 - 3X_2^2 & -6X_1X_2 \\ 6X_1X_2 & 3X_1^2 - 3X_2^2 \end{matrix}$$

function [έξοδος] = όνομα(είσοδος)
Εντολές
end

- ορίσματα εισόδου και εξόδου

| τετμημένη | τεταγμένη |
|---|---|
| $x$ | |

- Θα χρειαστούμε ένα function m-file για το Ιακωβιανό μητρώο της συνάρτησης

$$Jacob(X_1, X_2) = \begin{matrix} 3X_1^2 - 3X_2^2 & -6X_1X_2 \\ 6X_1X_2 & 3X_1^2 - 3X_2^2 \end{matrix}$$

function [έξοδος] = όνομα(είσοδος)
Εντολές
end

- ορίσματα εισόδου και εξόδου

| τετμημένη | τεταγμένη |
|---|---|
| $x$ | $w$ |

- Θα χρειαστούμε ένα function m-file για το Ιακωβιανό μητρώο της συνάρτησης

$$Jacob(X_1, X_2) = \begin{matrix} 3X_1^2 - 3X_2^2 & -6X_1X_2 \\ 6X_1X_2 & 3X_1^2 - 3X_2^2 \end{matrix}$$

function [έξοδος] = όνομα(είσοδος)
Εντολές
end

- ορίσματα εισόδου και εξόδου

| τετμημένη | τεταγμένη |
|---|---|
| $x$ | $w$ |

- όνομα συνάρτησης *Jacob*

- Θα χρειαστούμε ένα function m-file για το Ιακωβιανό μητρώο της συνάρτησης

$$Jacob(X_1, X_2) = \begin{matrix} 3X_1^2 - 3X_2^2 & -6X_1X_2 \\ 6X_1X_2 & 3X_1^2 - 3X_2^2 \end{matrix}$$

function [έξοδος] $=$ όνομα(είσοδος)
Εντολές
end

- ορίσματα εισόδου και εξόδου

| τετμημένη | τεταγμένη |
|---|---|
| $x$ | $w$ |

- όνομα συνάρτησης *Jacob*

- Εντολές

| | |
|---|---|
| | |

- Θα χρειαστούμε ένα function m-file για το Ιακωβιανό μητρώο της συνάρτησης

$$Jacob(X_1, X_2) = \begin{matrix} 3X_1^2 - 3X_2^2 & -6X_1X_2 \\ 6X_1X_2 & 3X_1^2 - 3X_2^2 \end{matrix}$$

function [έξοδος] = όνομα(είσοδος)
Εντολές
end

- ορίσματα εισόδου και εξόδου

| τετμημένη | τεταγμένη |
|---|---|
| $x$ | $w$ |

- όνομα συνάρτησης *Jacob*

- Εντολές

| Υπολογισμος $\frac{\partial f_1}{\partial x_1}$ | |
|---|---|
| | |

- Θα χρειαστούμε ένα function m-file για το Ιακωβιανό μητρώο της συνάρτησης

$$Jacob(X_1, X_2) = \begin{matrix} 3X_1^2 - 3X_2^2 & -6X_1X_2 \\ 6X_1X_2 & 3X_1^2 - 3X_2^2 \end{matrix}$$

function [έξοδος] = όνομα(είσοδος)
Εντολές
end

- ορίσματα εισόδου και εξόδου

| τετμημένη | τεταγμένη |
|---|---|
| $x$ | $w$ |

- όνομα συνάρτησης *Jacob*

- Εντολές

| | |
|---|---|
| Υπολογισμος $\frac{\partial f_1}{\partial x_1}$ | $d1f1 = (3 * X(1)\hat{}2 - 3 * X(2)\hat{}2);$ |
| | |

- Θα χρειαστούμε ένα function m-file για το Ιακωβιανό μητρώο της συνάρτησης

$$Jacob(X_1, X_2) = \begin{matrix} 3X_1^2 - 3X_2^2 & -6X_1X_2 \\ 6X_1X_2 & 3X_1^2 - 3X_2^2 \end{matrix}$$

function [έξοδος] = όνομα(είσοδος)
Εντολές
end

- ορίσματα εισόδου και εξόδου

| τετμημένη | τεταγμένη |
|---|---|
| $x$ | $w$ |

- όνομα συνάρτησης *Jacob*
- Εντολές

| | |
|---|---|
| Υπολογισμος $\frac{\partial f_1}{\partial x_1}$ | $d1f1 = (3 * X(1)\char94 2 - 3 * X(2)\char94 2);$ |
| Υπολογισμος $\frac{\partial f_1}{\partial x_2}$ | |

ΔΗΜΟΚΡΙΤΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΡΑΚΗΣ

- Θα χρειαστούμε ένα function m-file για το Ιακωβιανό μητρώο της συνάρτησης

$$Jacob(X_1, X_2) = \begin{matrix} 3X_1^2 - 3X_2^2 & -6X_1X_2 \\ 6X_1X_2 & 3X_1^2 - 3X_2^2 \end{matrix}$$

function [έξοδος] = όνομα(είσοδος)
Εντολές
end

- ορίσματα εισόδου και εξόδου

| τετμημένη | τεταγμένη |
|---|---|
| $x$ | $w$ |

- όνομα συνάρτησης *Jacob*

- Εντολές

| | |
|---|---|
| Υπολογισμος $\frac{\partial f_1}{\partial x_1}$ | $d1f1 = (3 * X(1)^2 - 3 * X(2)^2);$ |
| Υπολογισμος $\frac{\partial f_1}{\partial x_2}$ | $d2f1 = (-6 * X(1) * X(2));$ |
| | |

- Θα χρειαστούμε ένα function m-file για το Ιακωβιανό μητρώο της συνάρτησης

$$Jacob(X_1, X_2) = \begin{matrix} 3X_1^2 - 3X_2^2 & -6X_1X_2 \\ 6X_1X_2 & 3X_1^2 - 3X_2^2 \end{matrix}$$

function [έξοδος] = όνομα(είσοδος)
Εντολές
end

- ορίσματα εισόδου και εξόδου

| τετμημένη | τεταγμένη |
|---|---|
| $x$ | $w$ |

- όνομα συνάρτησης *Jacob*
- Εντολές

| | |
|---|---|
| Υπολογισμος $\frac{\partial f_1}{\partial x_1}$ | $d1f1 = (3 * X(1)\hat{\ }2 - 3 * X(2)\hat{\ }2);$ |
| Υπολογισμος $\frac{\partial f_1}{\partial x_2}$ | $d2f1 = (-6 * X(1) * X(2));$ |
| Υπολογισμος $\frac{\partial f_2}{\partial x_1}$ | |

- Θα χρειαστούμε ένα function m-file για το Ιακωβιανό μητρώο της συνάρτησης

$$Jacob(X_1, X_2) = \begin{array}{ll} 3X_1^2 - 3X_2^2 & -6X_1X_2 \\ 6X_1X_2 & 3X_1^2 - 3X_2^2 \end{array}$$

function [έξοδος] = όνομα(είσοδος)
Εντολές
end

- ορίσματα εισόδου και εξόδου

| τετμημένη | τεταγμένη |
|---|---|
| $x$ | $w$ |

- όνομα συνάρτησης *Jacob*
- Εντολές

| | |
|---|---|
| Υπολογισμος $\frac{\partial f_1}{\partial x_1}$ | $d1f1 = (3 * X(1)^2 - 3 * X(2)^2);$ |
| Υπολογισμος $\frac{\partial f_1}{\partial x_2}$ | $d2f1 = (-6 * X(1) * X(2));$ |
| Υπολογισμος $\frac{\partial f_2}{\partial x_1}$ | $d1f2 = (6 * X(1) * X(2));$ |
| | |

- Θα χρειαστούμε ένα function m-file για το Ιακωβιανό μητρώο της συνάρτησης

$$Jacob(X_1, X_2) = \begin{matrix} 3X_1^2 - 3X_2^2 & -6X_1X_2 \\ 6X_1X_2 & 3X_1^2 - 3X_2^2 \end{matrix}$$

function [έξοδος] = όνομα(είσοδος)
Εντολές
end

- ορίσματα εισόδου και εξόδου

| τετμημένη | τεταγμένη |
|---|---|
| $x$ | $w$ |

- όνομα συνάρτησης *Jacob*
- Εντολές

| | |
|---|---|
| Υπολογισμος $\frac{\partial f_1}{\partial x_1}$ | $d1f1 = (3 * X(1)\hat{}2 - 3 * X(2)\hat{}2);$ |
| Υπολογισμος $\frac{\partial f_1}{\partial x_2}$ | $d2f1 = (-6 * X(1) * X(2));$ |
| Υπολογισμος $\frac{\partial f_2}{\partial x_1}$ | $d1f2 = (6 * X(1) * X(2));$ |
| Υπολογισμος $\frac{\partial f_2}{\partial x_2}$ | |

ΔΗΜΟΚΡΙΤΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΡΑΚΗΣ

- Θα χρειαστούμε ένα function m-file για το Ιακωβιανό μητρώο της συνάρτησης

$$Jacob(X_1, X_2) = \begin{matrix} 3X_1^2 - 3X_2^2 & -6X_1X_2 \\ 6X_1X_2 & 3X_1^2 - 3X_2^2 \end{matrix}$$

```
function [έξοδος] = όνομα(είσοδος)
Εντολές
end
```

- ορίσματα εισόδου και εξόδου

| τετμημένη | τεταγμένη |
|---|---|
| $x$ | $w$ |

- όνομα συνάρτησης *Jacob*
- Εντολές

| | |
|---|---|
| Υπολογισμος $\frac{\partial f_1}{\partial x_1}$ | $d1f1 = (3 * X(1)^2 - 3 * X(2)^2);$ |
| Υπολογισμος $\frac{\partial f_1}{\partial x_2}$ | $d2f1 = (-6 * X(1) * X(2));$ |
| Υπολογισμος $\frac{\partial f_2}{\partial x_1}$ | $d1f2 = (6 * X(1) * X(2));$ |
| Υπολογισμος $\frac{\partial f_2}{\partial x_2}$ | $d2f2 = (3 * X(1)^2 - 3 * X(2)^2);$ |

ΔΗΜΟΚΡΙΤΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΡΑΚΗΣ

- Θα χρειαστούμε ένα function m-file για το Ιακωβιανό μητρώο της συνάρτησης

$$Jacob(X_1, X_2) = \begin{matrix} 3X_1^2 - 3X_2^2 & -6X_1X_2 \\ 6X_1X_2 & 3X_1^2 - 3X_2^2 \end{matrix}$$

function [έξοδος] = όνομα(είσοδος)
Εντολές
end

- ορίσματα εισόδου και εξόδου

| τετμημένη | τεταγμένη |
|---|---|
| $x$ | $w$ |

- όνομα συνάρτησης *Jacob*
- Εντολές

| | |
|---|---|
| Υπολογισμος $\frac{\partial f_1}{\partial x_1}$ | $d1f1 = (3 * X(1)\hat{}2 - 3 * X(2)\hat{}2);$ |
| Υπολογισμος $\frac{\partial f_1}{\partial x_2}$ | $d2f1 = (-6 * X(1) * X(2));$ |
| Υπολογισμος $\frac{\partial f_2}{\partial x_1}$ | $d1f2 = (6 * X(1) * X(2));$ |
| Υπολογισμος $\frac{\partial f_2}{\partial x_2}$ | $d2f2 = (3 * X(1)\hat{}2 - 3 * X(2)\hat{}2);$ |
| Επιστροφή μητρώου | |

- Θα χρειαστούμε ένα function m-file για το Ιακωβιανό μητρώο της συνάρτησης

$$Jacob(X_1, X_2) = \begin{matrix} 3X_1^2 - 3X_2^2 & -6X_1X_2 \\ 6X_1X_2 & 3X_1^2 - 3X_2^2 \end{matrix}$$

function [έξοδος] = όνομα(είσοδος)
Εντολές
end

- ορίσματα εισόδου και εξόδου

| τετμημένη | τεταγμένη |
|---|---|
| $x$ | $w$ |

- όνομα συνάρτησης *Jacob*

- Εντολές

| | |
|---|---|
| Υπολογισμος $\frac{\partial f_1}{\partial x_1}$ | $d1f1 = (3 * X(1)^2 - 3 * X(2)^2);$ |
| Υπολογισμος $\frac{\partial f_1}{\partial x_2}$ | $d2f1 = (-6 * X(1) * X(2));$ |
| Υπολογισμος $\frac{\partial f_2}{\partial x_1}$ | $d1f2 = (6 * X(1) * X(2));$ |
| Υπολογισμος $\frac{\partial f_2}{\partial x_2}$ | $d2f2 = (3 * X(1)^2 - 3 * X(2)^2);$ |
| Επιστροφή μητρώου | $w = [d1f1 \quad d2f1; d1f2 \quad d2f2]$ |

# Συμπερασματικά...

1. Πρόκειται για μία από τις γνωστότερες και ευρέως χρησιμοποιούμενες μεθόδους για την επίλυση συστημάτων μη-γραμμικών εξισώσεων.

# Συμπερασματικά...

1. Πρόκειται για μία από τις γνωστότερες και ευρέως χρησιμοποιούμενες μεθόδους για την επίλυση συστημάτων μη-γραμμικών εξισώσεων.
2. Είναι μια μέθοδος που απαιτεί την προσεκτική επιλογή της αρχικής προσέγγισης.

# Συμπερασματικά...

1. Πρόκειται για μία από τις γνωστότερες και ευρέως χρησιμοποιούμενες μεθόδους για την επίλυση συστημάτων μη-γραμμικών εξισώσεων.
2. Είναι μια μέθοδος που απαιτεί την προσεκτική επιλογή της αρχικής προσέγγισης.
3. Απαιτεί τον υπολογισμό μερικών παραγώγων, μια πληροφορία που είναι δύσκολη ή/και αδύνατο να υπολογσιτεί στην πλειοψηφία των περιπτώσεων.

# Συμπερασματικά...

1. Πρόκειται για μία από τις γνωστότερες και ευρέως χρησιμοποιούμενες μεθόδους για την επίλυση συστημάτων μη-γραμμικών εξισώσεων.
2. Είναι μια μέθοδος που απαιτεί την προσεκτική επιλογή της αρχικής προσέγγισης.
3. Απαιτεί τον υπολογισμό μερικών παραγώγων, μια πληροφορία που είναι δύσκολη ή/και αδύνατο να υπολογσιτεί στην πλειοψηφία των περιπτώσεων.
4. Πρόκειται για μια μέθοδο που υπό προϋποθέσεις συγκλίνει γρήγορα.

## Λίγα λόγια για τη μέθοδο

Η μέθοδος Broyden είναι μια γνωστή και συχνά χρησιμοποιούμενη μεθόδος αριθμητικής προσέγγισης λύσεων της συνάρτησης $F_n(x) = \mathcal{O}_n$, όπου $\mathcal{O}_n$ δηλώνει το μηδενικό διάνυσμα.

Πρόκειται για μια γενίκευση της μεθόδου της Τέμνουσας για συναρτήσεις πολλών μεταβλητών.

Ομοιάζει με τη μέθοδο Newton, καθώς και οι δυο αποτελούν μεθόδους τοπικής σύγκλισης.

Επιλύει επαναληπτικά ισοδύναμα γραμμικά συστήματα, οι λύσεις των οποίων αποτελούν καλές προσεγίσεις για το πρόβλημα εύρεσης λύσεων συστημάτων μη-γραμμικών εξισώσεων.

Διαφοροποιείται από τη μέθοδο Newton, αναφορικά με τον υπολογισμό του Ιακωβιανού μητρώου.

# Περιγραφή της μεθόδου

## Βασική ιδέα

- Η μέθοδος του Broyden αποτελεί γενίκευση της μεθόδου της τέμνουσας (secant method) στις πολλές διαστάσεις.
- Η μέθοδος της τέμνουσας αντικαθιστά την τιμή της πρώτης παραγώγου $f'(x_n)$ με την πεπερασμένη διαφορά:

$$f'(x_n) \simeq \frac{f(x_n) - f(x_{n-1})}{x_n - x_{n-1}} \qquad (1)$$

- Η τελευταία ποσότητα αντικαθίσταται στη μέθοδο του Newton:

$$x_{n+1} = x_n - \frac{1}{f'(x_n)} f(x_n) \simeq x_n - \frac{1}{\frac{f(x_n)-f(x_{n-1})}{x_n-x_{n-1}}} f(x_n) \qquad (2)$$

- Ο Broyden γενίκευσε τον παραπάνω τύπο για συστήματα εξισώσεων της μορφής $F(x) = 0$, αντικαθιστώντας την παράγωγο με το Ιακωβιανό μητρώο $J$ της συνάρτησης.
- Το Ιακωβιανό μητρώο καθορίζεται με χρήση της εξίσωσης της χορδής:

$$J_n \cdot (x_n - x_{n-1}) \simeq F(x_n) - F(x_{n-1}) \qquad (3)$$

# Περιγραφή

## Παρατηρήσεις

- Αντί για τη χρήση του Ιακωβιανού μητρώου ο Broyden πρότεινε τη χρησιμοποίηση του μητρώου $A_n$ που ικανοποιεί τον ίδιο τύπο.
- Έτσι, θέτοντας $s_n = x_n - x_{n-1}, \; y_n = F(x_n) - F(x_{n-1})$ προκύπτει το μητρώο του Broyden:

$$A_n = A_{n-1} + \frac{y_n - A_{n-1}s_n}{||s_n||_2^2}(s_n)^T \tag{4}$$

- Το επαναληπτικό σχήμα της μεθόδου Broyden θα είναι:

$$x_{n+1} = x_n - (A_n)^{-1}F(x_n) \tag{5}$$

# Περιγραφή

## Προσοχή:

- Για την αποφυγή του υπολογισμού του αντίστροφου μητρώου, μπορούμε να επιλύσουμε το γραμμικό σύστημα $A_n \Delta_n = -F(x_n)$ και ύστερα να υπολογίσουμε τη νέα προσέγγιση $x_{n+1}$, καθώς $x_{n+1} = x_n + \Delta_n$
- Η μέθοδος του Broyden απαιτεί λιγότερους συναρτησιακούς υπολογισμούς ανά επανάληψη συγκριτικά με τη μέθοδο του Newton, καθώς δεν υπολογίζει σε κάθε επανάληψη το Ιακωβιανό μητρώο.

# Αλγόριθμος του Broyden

0: Αρχικοποιούμε το σημείο $x(0)$ και αντιστοίχως την εκτίμηση του Ιακωβιανού μητρώου $A_0 = J(x_0)$

1: Για την πρώτη προσέγγιση του μητρώου του Broyden χρησιμοποιείται το Ιακωβιανό μητρώο, οπότε για την πρώτη εκτέλεση έχουμε τη μέθοδο του Newton

$$x_1 = x_0 - A_0^{-1} F(x_0) \tag{6}$$

2: Για $k \geq 1$ επαναλαμβάνουμε τα παρακάτω βήματα:

2.1: Εκτιμούμε τη συνάρτηση $F(x_k)$

2.2: Ενημερώνουμε το Ιακωβιανό μητρώο χρησιμοποιώντας τους τύπους: $s_k = x_k - x_{k-1}$ και $y_k = F(x_k) - F(x_{k-1})$. Συνεπώς, το μητρώο του Broyden:

$$A_k = A_{k-1} + \frac{y_k - A_{k-1} s_k}{||s_k||_2^2} (s_k)^T \tag{7}$$

2.3: Επιλύουμε το γραμμικό σύστημα $A_k \Delta x = -F(x_k)$ ως προς $\Delta x$

2.4: Υπολογίζουμε την επόμενη προσέγγιση: $x_{k+1} = x_k + \Delta x$

3: Ελέγχουμε κάποιο κριτήριο τερματισμού. Εάν δεν ικανοποιείται πηγαίνουμε στο Βήμα 2

# Παράδειγμα - Εφαρμογή 1

Να εφαρμόσετε τη μέθοδο του Broyden (για δύο βήματα) για την επίλυση του παρακάτω συστήματος:

$$\begin{cases} x+y &= 2 \\ x-y &= 0 \end{cases} \tag{8}$$

για $(x_0, y_0) = (0, 0)$ και $A_0 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$.

Λύση:

Η συνάρτηση $F$, θα είναι η $F_2 = (f_1, f_2) = (x+y-2, x-y)$. Στο πρώτο βήμα θα έχουμε:

$$\begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} - A_0^{-1} F(x_0, y_0) = \begin{bmatrix} 0 \\ 0 \end{bmatrix} - \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} -2 \\ 0 \end{bmatrix} = \begin{bmatrix} 2 \\ 0 \end{bmatrix}.$$

Οπότε: το $s_1 = x_1 - x_0 = \begin{bmatrix} 2 \\ 0 \end{bmatrix}$ και $F(x_1, y_1) = (0, 2)$. Χρησιμοποιώντας το μητρώο ανανέωσης του Broyden, παίρνουμε:

# Παράδειγμα - Εφαρμογή 1

$$A_1 = A_0 + \frac{(F(x_1,y_1) - F(x_0,y_0)) - A_0 s_1}{\|s_1\|^2} s_1^\top = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \frac{\left(\begin{bmatrix} 0 \\ 2 \end{bmatrix} - \begin{bmatrix} -2 \\ 0 \end{bmatrix}\right) - \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} 2 \\ 0 \end{bmatrix}}{\| \begin{bmatrix} 2 \\ 0 \end{bmatrix} \|^2} \begin{bmatrix} 2 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$$

**Στο δεύτερο βήμα, έχουμε:** $\begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} - A_1^{-1} F(x_1,y_1) = \begin{bmatrix} 2 \\ 0 \end{bmatrix} - \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 2 \end{bmatrix} = \begin{bmatrix} 2 \\ -2 \end{bmatrix}$.

**Οπότε: το** $s_2 = x_2 - x_1 = \begin{bmatrix} 0 \\ -2 \end{bmatrix}$ **και** $F(x_2,y_2) = (-2,4)$. **Χρησιμοποιώντας το μητρώο ανανέωσης του** Broyden, **παίρνουμε:**

$$A_2 = A_1 + \frac{(F(x_2,y_2) - F(x_1,y_1)) - A_1 s_2}{\|s_2\|^2} s_2^\top = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} + \frac{1}{4} \begin{bmatrix} -2 \\ 4 \end{bmatrix} \begin{bmatrix} 0 & -2 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

# Παράδειγμα - Εφαρμογή 1

- Όσο προχωράει η διαδικασία αποκτούμε πληροφορία σε όλο και περισσότερα σημεία, οπότε τα μητώρα $A_k$ προσεγγίζουν τις κατάλληλες τιμές που ανταποκρίνονται στο Ιακωβιανό μητρώο της συνάρτησης $F$.
- Στην επόμενη επανάληψη, πράγματι, η τιμή του $A_3$ θα ανταποκρίνεται στην τιμή του Ιακωβιανού, ώστε η προσέγγιση $(x_3, y_3)$ να είναι η λύση του συστήματος.
- Άσκηση: Να κάνετε τον σχετικό έλεγχο ώστε να επιβεβαιώσετε τον παραπάνω ισχυρισμό.

# Παράδειγμα - Εφαρμογή 2

Να εφαρμόσετε τον αλγόριθμο της μεθόδου του Broyden για την προσέγγιση μιας λύσης του συστήματος εξισώσεων:

$$F_2 = \begin{cases} x_1 + x_2 - 3 = & 0 \\ x_1^2 + x_2^2 - 9 = & 0 \end{cases} \tag{9}$$

Λύση:

(Το σύστημα έχει ως λύσεις τις $(0,3)^\top$ και $(3,0)^\top$.) Παίρνουμε ως αρχική προσέγγιση της λύσης την τιμή $x^0 = (1,5)^\top$ και ως αρχικό μητρώο $A_0$ το Ιακωβιανό μητρώο $J_{F_2}(x^0)$.

Το Ιακωβιανό μητρώο της συνάρτησης $F_2 = (f_1, f_2) : \mathbb{R}^2 \to \mathbb{R}^2$ θα είναι:

$$J_{F_2}(x) = \begin{bmatrix} 1 & 1 \\ 2x_1 & 2x_2 \end{bmatrix}$$

Χρησιμοποιώντας την αρχική εκτίμηση $x^0$, υπολογίζουμε τις αντίστοιχες τιμές της συνάρτησης και του Ιακωβιανού μητρώου:

# Παράδειγμα - Εφαρμογή 2

$$F_2(x^0) = \begin{pmatrix} f_1(x_1^0, x_2^0) \\ f_2(x_1^0, x_2^0) \end{pmatrix} = \begin{pmatrix} x_1^0 + x_2^0 - 3 \\ (x_1^0)^2 + (x_2^0)^2 - 9 \end{pmatrix} = \begin{pmatrix} 3 \\ 17 \end{pmatrix} \quad \text{και}$$

$$J_{F_2}(x^0) = \begin{pmatrix} \frac{\partial f_1(x_1^0, x_2^0)}{\partial x_1} & \frac{\partial f_1(x_1^0, x_2^0)}{\partial x_2} \\ \frac{\partial f_2(x_1^0, x_2^0)}{\partial x_1} & \frac{\partial f_2(x_1^0, x_2^0)}{\partial x_2} \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 2x_1^0 & 2x_2^0 \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 2 & 10 \end{pmatrix}$$

ΔΗΜΟΚΡΙΤΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΡΑΚΗΣ

- Θέτουμε

$$A_0 \equiv J_{F_2}(x^0) = \begin{pmatrix} 1 & 1 \\ 2 & 10 \end{pmatrix}$$

- Θέτουμε

$$A_0 \equiv J_{F_2}(x^0) = \begin{pmatrix} 1 & 1 \\ 2 & 10 \end{pmatrix}$$

- Επιλύουμε το παρακάτω γραμμικό σύστημα ως προς $s^0$

$$A_0 s^0 = -F_n(x_0)$$

- Θέτουμε

$$A_0 \equiv J_{F_2}(x^0) = \begin{pmatrix} 1 & 1 \\ 2 & 10 \end{pmatrix}$$

- Επιλύουμε το παρακάτω γραμμικό σύστημα ως προς $s^0$

$$A_0 s^0 = -F_n(x_0)$$

- Θα έχει τη μορφή

$$\begin{pmatrix} 1 & 1 \\ 2 & 10 \end{pmatrix} \begin{pmatrix} s_1^0 \\ s_2^0 \end{pmatrix} = - \begin{pmatrix} 3 \\ 17 \end{pmatrix}$$

- Θέτουμε

$$A_0 \equiv J_{F_2}(x^0) = \begin{pmatrix} 1 & 1 \\ 2 & 10 \end{pmatrix}$$

- Επιλύουμε το παρακάτω γραμμικό σύστημα ως προς $s^0$

$$A_0 s^0 = -F_n(x_0)$$

- Θα έχει τη μορφή

$$\begin{pmatrix} 1 & 1 \\ 2 & 10 \end{pmatrix} \begin{pmatrix} s_1^0 \\ s_2^0 \end{pmatrix} = - \begin{pmatrix} 3 \\ 17 \end{pmatrix}$$

- Η λύση του συστήματος είναι

$$\begin{pmatrix} s_1^0 \\ s_2^0 \end{pmatrix} = -\frac{1}{8} \begin{pmatrix} 13 \\ 11 \end{pmatrix}$$

- Θέτουμε

$$A_0 \equiv J_{F_2}(x^0) = \begin{pmatrix} 1 & 1 \\ 2 & 10 \end{pmatrix}$$

- Επιλύουμε το παρακάτω γραμμικό σύστημα ως προς $s^0$

$$A_0 s^0 = -F_n(x_0)$$

- Θα έχει τη μορφή

$$\begin{pmatrix} 1 & 1 \\ 2 & 10 \end{pmatrix} \begin{pmatrix} s_1^0 \\ s_2^0 \end{pmatrix} = - \begin{pmatrix} 3 \\ 17 \end{pmatrix}$$

- Η λύση του συστήματος είναι

$$\begin{pmatrix} s_1^0 \\ s_2^0 \end{pmatrix} = -\frac{1}{8} \begin{pmatrix} 13 \\ 11 \end{pmatrix} = \begin{pmatrix} -1.625 \\ -1.375 \end{pmatrix}$$

ΔΗΜΟΚΡΙΤΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΡΑΚΗΣ

- Η επόμενη εκτίμηση της λύσης θα είναι

$$\begin{pmatrix} x_1^1 \\ x_2^1 \end{pmatrix} = \begin{pmatrix} x_1^0 \\ x_2^0 \end{pmatrix} + \begin{pmatrix} s_1^0 \\ s_2^0 \end{pmatrix}$$

- Η επόμενη εκτίμηση της λύσης θα είναι

$$\begin{pmatrix} x_1^1 \\ x_2^1 \end{pmatrix} = \begin{pmatrix} x_1^0 \\ x_2^0 \end{pmatrix} + \begin{pmatrix} s_1^0 \\ s_2^0 \end{pmatrix} = \begin{pmatrix} 1 \\ 5 \end{pmatrix} + \begin{pmatrix} -1.625 \\ -1.375 \end{pmatrix}$$

- Η επόμενη εκτίμηση της λύσης θα είναι

$$\begin{pmatrix} x_1^1 \\ x_2^1 \end{pmatrix} = \begin{pmatrix} x_1^0 \\ x_2^0 \end{pmatrix} + \begin{pmatrix} s_1^0 \\ s_2^0 \end{pmatrix} = \begin{pmatrix} 1 \\ 5 \end{pmatrix} + \begin{pmatrix} -1.625 \\ -1.375 \end{pmatrix} = \begin{pmatrix} -0.625 \\ 3.625 \end{pmatrix}$$

ΔΗΜΟΚΡΙΤΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΡΑΚΗΣ

- Η επόμενη εκτίμηση της λύσης θα είναι

$$\begin{pmatrix} x_1^1 \\ x_2^1 \end{pmatrix} = \begin{pmatrix} x_1^0 \\ x_2^0 \end{pmatrix} + \begin{pmatrix} s_1^0 \\ s_2^0 \end{pmatrix} = \begin{pmatrix} 1 \\ 5 \end{pmatrix} + \begin{pmatrix} -1.625 \\ -1.375 \end{pmatrix} = \begin{pmatrix} -0.625 \\ 3.625 \end{pmatrix}$$

- Υπολογίζουμε τις τιμές της συνάρτησης $F_2 = (f_1, f_2)^T$ στο σημείο $x^1$

$$F_2(x^1) = \begin{pmatrix} f_1(x_1^1, x_2^1) \\ f_2(x_1^1, x_2^1) \end{pmatrix} = \begin{pmatrix} x_1^1 + x_2^1 - 3 \\ (x_1^1)^2 + (x_2^1)^2 - 9 \end{pmatrix}$$

ΔΗΜΟΚΡΙΤΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΡΑΚΗΣ

- Η επόμενη εκτίμηση της λύσης θα είναι

$$\begin{pmatrix} x_1^1 \\ x_2^1 \end{pmatrix} = \begin{pmatrix} x_1^0 \\ x_2^0 \end{pmatrix} + \begin{pmatrix} s_1^0 \\ s_2^0 \end{pmatrix} = \begin{pmatrix} 1 \\ 5 \end{pmatrix} + \begin{pmatrix} -1.625 \\ -1.375 \end{pmatrix} = \begin{pmatrix} -0.625 \\ 3.625 \end{pmatrix}$$

- Υπολογίζουμε τις τιμές της συνάρτησης $F_2 = (f_1, f_2)^T$ στο σημείο $x^1$

$$F_2(x^1) = \begin{pmatrix} f_1(x_1^1, x_2^1) \\ f_2(x_1^1, x_2^1) \end{pmatrix} = \begin{pmatrix} x_1^1 + x_2^1 - 3 \\ (x_1^1)^2 + (x_2^1)^2 - 9 \end{pmatrix} = \begin{pmatrix} 0 \\ 4.53125 \end{pmatrix}$$

- Η επόμενη εκτίμηση της λύσης θα είναι

$$\begin{pmatrix} x_1^1 \\ x_2^1 \end{pmatrix} = \begin{pmatrix} x_1^0 \\ x_2^0 \end{pmatrix} + \begin{pmatrix} s_1^0 \\ s_2^0 \end{pmatrix} = \begin{pmatrix} 1 \\ 5 \end{pmatrix} + \begin{pmatrix} -1.625 \\ -1.375 \end{pmatrix} = \begin{pmatrix} -0.625 \\ 3.625 \end{pmatrix}$$

- Υπολογίζουμε τις τιμές της συνάρτησης $F_2 = (f_1, f_2)^T$ στο σημείο $x^1$

$$F_2(x^1) = \begin{pmatrix} f_1(x_1^1, x_2^1) \\ f_2(x_1^1, x_2^1) \end{pmatrix} = \begin{pmatrix} x_1^1 + x_2^1 - 3 \\ (x_1^1)^2 + (x_2^1)^2 - 9 \end{pmatrix} = \begin{pmatrix} 0 \\ 4.53125 \end{pmatrix}$$

- και θέτουμε

$$y^0 = F_2(x^1) - F_2(x^0)$$

- Η επόμενη εκτίμηση της λύσης θα είναι

$$\begin{pmatrix} x_1^1 \\ x_2^1 \end{pmatrix} = \begin{pmatrix} x_1^0 \\ x_2^0 \end{pmatrix} + \begin{pmatrix} s_1^0 \\ s_2^0 \end{pmatrix} = \begin{pmatrix} 1 \\ 5 \end{pmatrix} + \begin{pmatrix} -1.625 \\ -1.375 \end{pmatrix} = \begin{pmatrix} -0.625 \\ 3.625 \end{pmatrix}$$

- Υπολογίζουμε τις τιμές της συνάρτησης $F_2 = (f_1, f_2)^T$ στο σημείο $x^1$

$$F_2(x^1) = \begin{pmatrix} f_1(x_1^1, x_2^1) \\ f_2(x_1^1, x_2^1) \end{pmatrix} = \begin{pmatrix} x_1^1 + x_2^1 - 3 \\ (x_1^1)^2 + (x_2^1)^2 - 9 \end{pmatrix} = \begin{pmatrix} 0 \\ 4.53125 \end{pmatrix}$$

- και θέτουμε

$$y^0 = F_2(x^1) - F_2(x^0) = \begin{pmatrix} 0 \\ 4.53125 \end{pmatrix} - \begin{pmatrix} 3 \\ 17 \end{pmatrix}$$

ΔΗΜΟΚΡΙΤΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΡΑΚΗΣ

- Η επόμενη εκτίμηση της λύσης θα είναι

$$\begin{pmatrix} x_1^1 \\ x_2^1 \end{pmatrix} = \begin{pmatrix} x_1^0 \\ x_2^0 \end{pmatrix} + \begin{pmatrix} s_1^0 \\ s_2^0 \end{pmatrix} = \begin{pmatrix} 1 \\ 5 \end{pmatrix} + \begin{pmatrix} -1.625 \\ -1.375 \end{pmatrix} = \begin{pmatrix} -0.625 \\ 3.625 \end{pmatrix}$$

- Υπολογίζουμε τις τιμές της συνάρτησης $F_2 = (f_1, f_2)^T$ στο σημείο $x^1$

$$F_2(x^1) = \begin{pmatrix} f_1(x_1^1, x_2^1) \\ f_2(x_1^1, x_2^1) \end{pmatrix} = \begin{pmatrix} x_1^1 + x_2^1 - 3 \\ (x_1^1)^2 + (x_2^1)^2 - 9 \end{pmatrix} = \begin{pmatrix} 0 \\ 4.53125 \end{pmatrix}$$

- και θέτουμε

$$y^0 = F_2(x^1) - F_2(x^0) = \begin{pmatrix} 0 \\ 4.53125 \end{pmatrix} - \begin{pmatrix} 3 \\ 17 \end{pmatrix} = \begin{pmatrix} -3 \\ -12.46875 \end{pmatrix}$$

- Υπολογίζουμε το μητρώο $U_0$

$$U_0 = \frac{(y^0 - A_0 s^0)(s^0)^T}{(s^0)^T s^0}$$

- Υπολογίζουμε το μητρώο $U_0$

$$U_0 = \frac{(y^0 - A_0 s^0)(s^0)^T}{(s^0)^T s^0}$$

### Αριθμητής

$$\begin{aligned}(y^0 - A_0 s^0)(s^0)^T &= \begin{pmatrix} 0 \\ 4.53125 \end{pmatrix} \begin{pmatrix} -1.625 & -1.375 \end{pmatrix} \\ &= \begin{pmatrix} 0 & 0 \\ -7.3632812 & -6.2304687 \end{pmatrix}\end{aligned}$$

- Υπολογίζουμε το μητρώο $U_0$

$$U_0 = \frac{(y^0 - A_0 s^0)(s^0)^T}{(s^0)^T s^0}$$

**Αριθμητής**

$$\begin{aligned}(y^0 - A_0 s^0)(s^0)^T &= \begin{pmatrix} 0 \\ 4.53125 \end{pmatrix} \begin{pmatrix} -1.625 & -1.375 \end{pmatrix} \\ &= \begin{pmatrix} 0 & 0 \\ -7.3632812 & -6.2304687 \end{pmatrix}\end{aligned}$$

**Παρονομαστής**

$$\begin{aligned}(s^0)^T s^0 &= \begin{pmatrix} -1.625 & -1.375 \end{pmatrix} \begin{pmatrix} -1.625 \\ -1.375 \end{pmatrix} \\ &= (-1.625)^2 + (-1.375)^2 = 4.53125\end{aligned}$$

$$U_0 = \frac{1}{4.53125}\begin{pmatrix} 0 & 0 \\ -7.3632812 & -6.2304687 \end{pmatrix}$$
$$= \begin{pmatrix} 0 & 0 \\ -1.625 & -1.375 \end{pmatrix}$$

$$U_0 = \frac{1}{4.53125}\begin{pmatrix} 0 & 0 \\ -7.3632812 & -6.2304687 \end{pmatrix}$$
$$= \begin{pmatrix} 0 & 0 \\ -1.625 & -1.375 \end{pmatrix}$$

- Υπολογίζουμε το μητρώο $A_1$

$$A_1 = A_0 + U_0$$

$$U_0 = \frac{1}{4.53125}\begin{pmatrix} 0 & 0 \\ -7.3632812 & -6.2304687 \end{pmatrix}$$
$$= \begin{pmatrix} 0 & 0 \\ -1.625 & -1.375 \end{pmatrix}$$

- Υπολογίζουμε το μητρώο $A_1$

$$A_1 = A_0 + U_0$$
$$= \begin{pmatrix} 0 & 0 \\ -1.625 & -1.375 \end{pmatrix} + \begin{pmatrix} 1 & 1 \\ 2 & 10 \end{pmatrix}$$

$$U_0 = \frac{1}{4.53125}\begin{pmatrix} 0 & 0 \\ -7.3632812 & -6.2304687 \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ -1.625 & -1.375 \end{pmatrix}$$

- Υπολογίζουμε το μητρώο $A_1$

$$A_1 = A_0 + U_0 = \begin{pmatrix} 0 & 0 \\ -1.625 & -1.375 \end{pmatrix} + \begin{pmatrix} 1 & 1 \\ 2 & 10 \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 0.375 & 8.625 \end{pmatrix}$$

$$\begin{aligned} U_0 &= \frac{1}{4.53125}\begin{pmatrix} 0 & 0 \\ -7.3632812 & -6.2304687 \end{pmatrix} \\ &= \begin{pmatrix} 0 & 0 \\ -1.625 & -1.375 \end{pmatrix} \end{aligned}$$

- Υπολογίζουμε το μητρώο $A_1$

$$\begin{aligned} A_1 &= A_0 + U_0 \\ &= \begin{pmatrix} 0 & 0 \\ -1.625 & -1.375 \end{pmatrix} + \begin{pmatrix} 1 & 1 \\ 2 & 10 \end{pmatrix} \\ &= \begin{pmatrix} 1 & 1 \\ 0.375 & 8.625 \end{pmatrix} \end{aligned}$$

Επαναλαμβάνοντας την παραπάνω διαδικασία, μέχρι να ενεργοποιηθεί ένα κριτήριο τερματισμού, αποκτούμε καλύτερες προσσεγγίσεις της αρχικής λύσης

# Ασκήσεις

1. Να επιλυθεί με τη μέθοδο Broyden το ακόλουθο μη-γραμμικό σύστημα:

$$\left\{\begin{array}{rcl} 9x^2+36y^2+4z^2-36 & = & 0 \\ x^2-2y^2-20z & = & 0 \\ x^2-y^2+z^2 & = & 0 \end{array}\right. \tag{10}$$

Υπόδειξη: Το σύστημα έχει 4 λύσεις. Μια καλή αρχική προσέγγιση είναι η $X0 = [\pm 1 \;\; \pm 1 \;\; 0]$

2. Να επιλυθεί με τη μέθοδο Broyden το ακόλουθο μη-γραμμικό σύστημα:

$$\left\{\begin{array}{rcl} w^2+x^2+3y^2-z^2-5 & = & 0 \\ w+x^3-2y^2-10z & = & 0 \\ 20-w+x^2+y^3+z^2 & = & 0 \\ w^3+x-y^3+z-10 & = & 0 \end{array}\right. \tag{11}$$

Υπόδειξη: Το σύστημα έχει μόνο μία πραγματική λύση. Μια καλή αρχική προσέγγιση είναι η $X0 = [-4 \;\; 4 \;\; -4 \;\; 4]$

# Newton-Broyden

- Και οι δύο μέθοδοι επιλύουν ένα σύστημα μη-γραμμικών εξισώσεων, επιλύοντας επαναληπτικά κατάλληλα συστήματα γραμμικών εξισώσεων.
- Η μέθοδος Newton απαιτεί σε κάθε επανάληψη τον υπολογισμό μερικών παραγώγων λόγω του Ιακωβιανού μητρώου.
- Η μέθοδος Broyden συνήθως αξιοποιεί μόνο στην πρώτη επανάληψη την τιμή του Ιακωβιανού και ακολούθως, αναπροσαρμόζει το μητρώο με τη βοήθεια του μητρώου Broyden.
- Το υπολογιστικό κόστος που απαιτεί η μέθοδος Newton είναι μεγαλύτερο σε σχέση με αυτό της Broyden.
- Η μέθοδος Newton υπερισχύει στην ταχύτητα σύγκλισης έναντι της Broyden.
- Το πλήθος επαναλήψεων που απαιτεί η μέθοδος Newton μπορεί να είναι μικρότερο, όμως η μέθοδος Broyden «στοιχίζει» λιγότερους συναρτησιακούς υπολογισμούς.

Πλήθος συναρτησιακών υπολογισμών: Newton: $(n^2 + n) * ITER_N$, Broyden: $n * ITER_B + n^2$

Συμπέρασμα: Όσο αυξάνεται η διαστασιμότητα του προβλήματος προτιμούμε τη μέθοδο του Broyden, καθώς «απαιτεί» μικρότερο πλήθος συναρτησιακών υπολογισμών ανά επανάληψη.

ΔΗΜΟΚΡΙΤΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΡΑΚΗΣ

# Βιβλιογραφία - Αναφορές


- Γραββάνης Γ., Επιστημονικοί υπολογισμοί: Υπολογιστικές μέθοδοι και μαθηματικό λογισμικό, Παπασωτηρίου, 2013.
- Ακρίβης Γ.Δ., Δουγάλης Β.Α., Εισαγωγή στην αριθμητική ανάλυση, Πανεπιστημιακές εκδόσεις Κρήτης, Ηράκλειο, 1997.
- Βραχάτης Μ.Ν., Αριθμητική ανάλυση: Εισαγωγή, Εκδόσεις Κλειδάριθμος, Αθήνα, 2011.
- Βραχάτης Μ.Ν., Αριθμητική Ανάλυση: Υπερβατικές Εξισώσεις, Εκδόσεις Κλειδάριθμος, Αθήνα, 2012.
- Νικόλαος Μισυρλής. Αριθμητική Ανάλυση: Μια Αλγοριθμική Προσέγγιση. Εκδόσεις: Εκδοτική ΕΚΠΑ 2017. Εύδοξος Αριθμητική Ανάλυση


ΔΗΜΟΚΡΙΤΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΡΑΚΗΣ

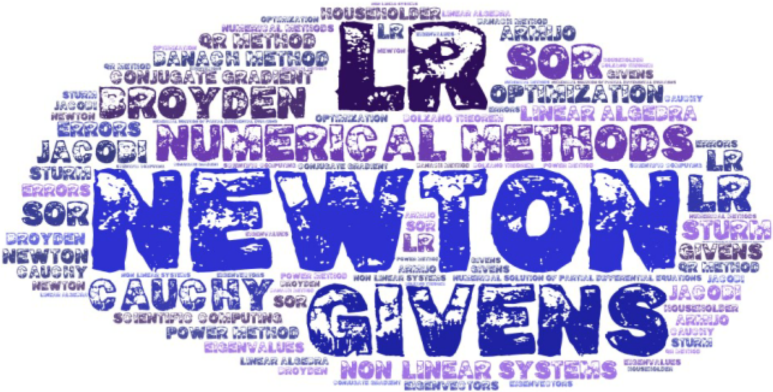
# Βιβλιογραφία - Αναφορές


- Gerald, Curtis F, Applied numerical analysis. Pearson Education India, 2004.
- Atkinson, Kendall E, An introduction to numerical analysis. John wiley & sons, 2008.
- Burden, Richard L and Faires, J Douglas, Numerical Analysis, Brooks, Cole, Belmont, CA, 1997.
- Phillips, George M and Taylor, Peter J, Theory and applications of numerical analysis. Elsevier, 1996.
- Davis, Philip J and Rabinowitz, Philip, Methods of numerical integration. Courier Corporation, 2007.

# Βιβλιογραφία - Αναφορές


- Herniter, Marc E, Programming in MATLAB. Brooks/Cole Publishing Co., 2000.
- Wolfram, Stephen and others, The MATHEMATICA® book, version 4. Cambridge university press, 1999.
- Gene H. Golub, Charles F. Van Loan, Θεωρία και Υπολογισμοί Μητρώων, Εκδόσεις Πεδίο, 2015.
- Λ. Trefethen, D. Bau, Numerical Linear Algebra, SIAM 1997.
- Β. Δουγαλής, Δ. Νούτσος, Α. Χατζηδήμος, Αριθμητική Γραμμική Άλγεβρα, Πανεπιστημιακές σημειώσεις, Πανεπιστήμιο Ιωαννίνων.

# Σας ευχαριστώ πολύ για την προσοχή σας!




ΔΗΜΟΚΡΙΤΕΙΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΘΡΑΚΗΣ