Αριθμητική Ανάλυση

Τι προβλήματα αντιμετωπίζουμε;

- Επίλυση Γραμμικών και μη-Γραμμικών Εξισώσεων ή Συστημάτων
- Θεωρία Προσεγγίσεων
- Αριθμητική Επίλυση Διαφορικών Εξισώσεων

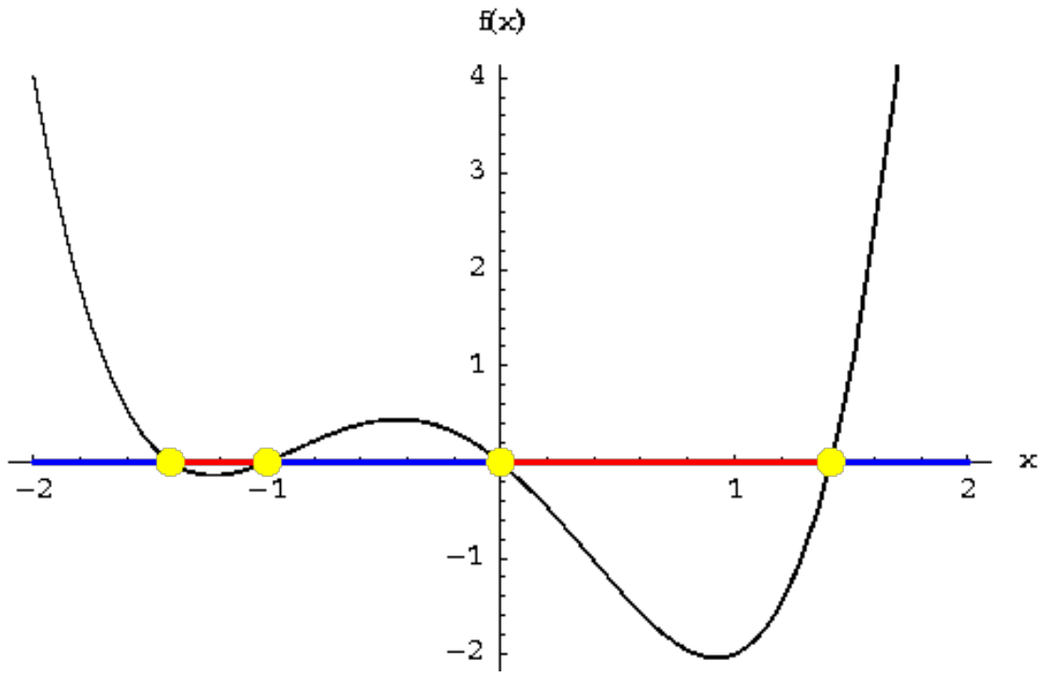
Επίλυση Γραμμικών και μη-Γραμμικών Εξισώσεων ή Συστημάτων

Γραμμικά συστήματα $Ax = b$, όπου αναζητούμε το διάνυσμα x, με A ένα τετραγωνικός πίνακα

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

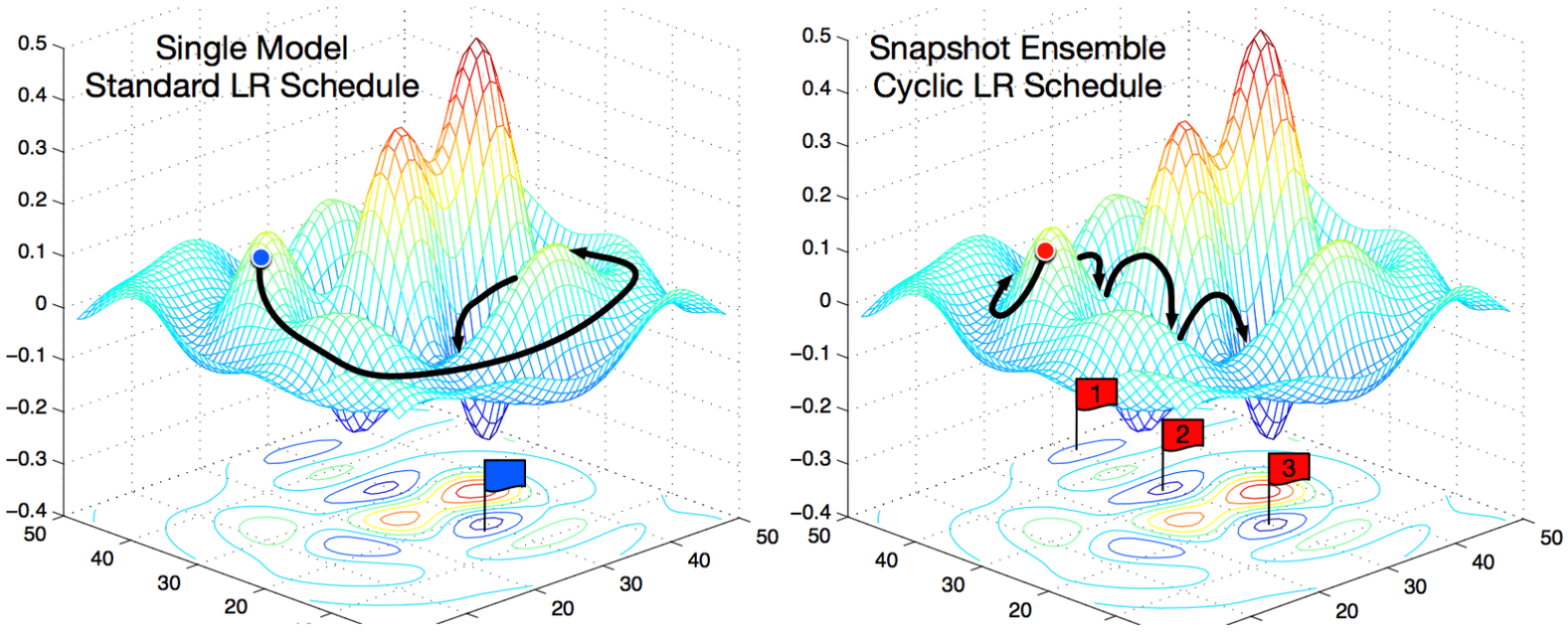
Επίλυση Γραμμικών και μη-Γραμμικών Εξισώσεων ή Συστημάτων

Μη γραμμικά συστήματα. Αναζητούμε τη ρίζα της εξίσωσης $f(x) = 0$ ή του συστήματος $F(x) = 0$



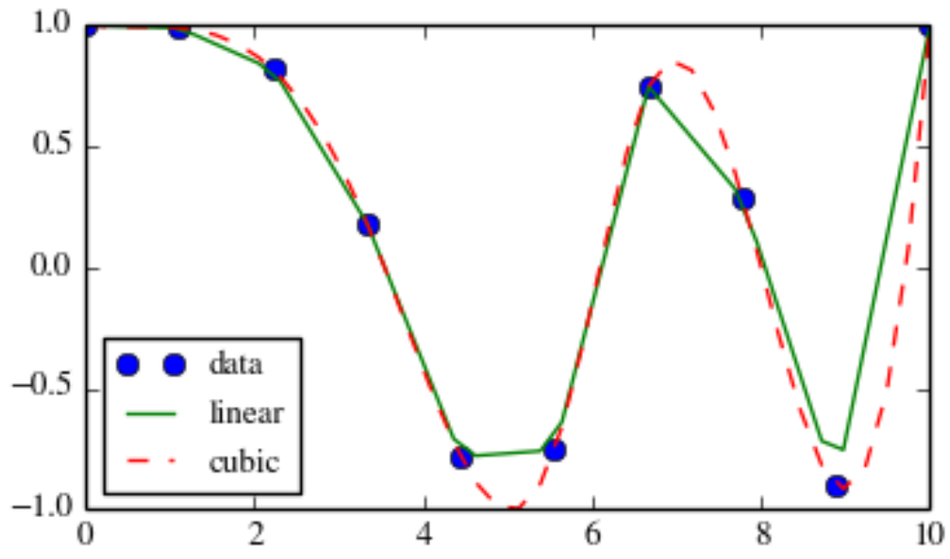
Επίλυση Γραμμικών και μη-Γραμμικών Εξισώσεων ή Συστημάτων

Βελτιστοποίηση



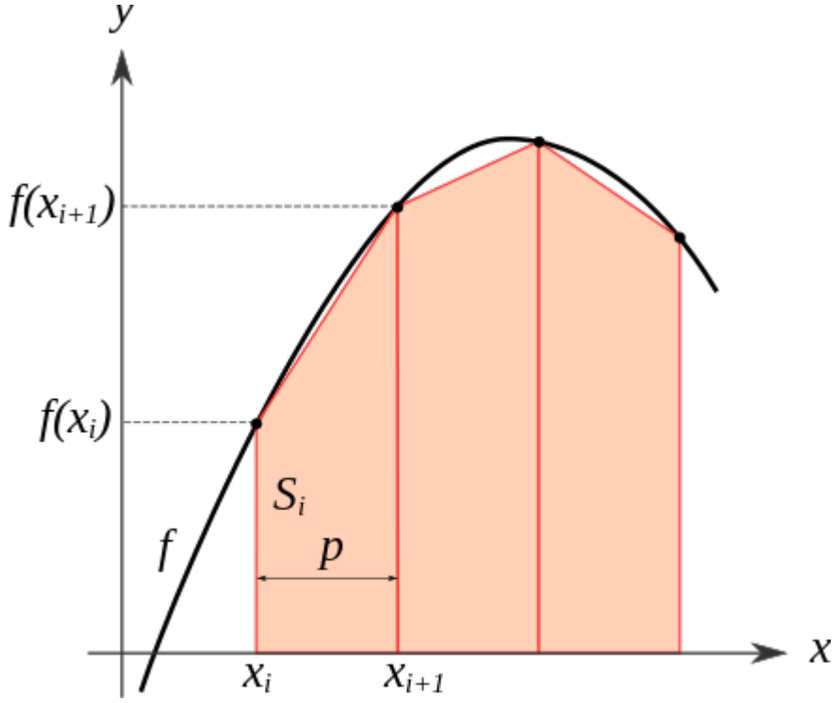
Θεωρία Προσεγγίσεων

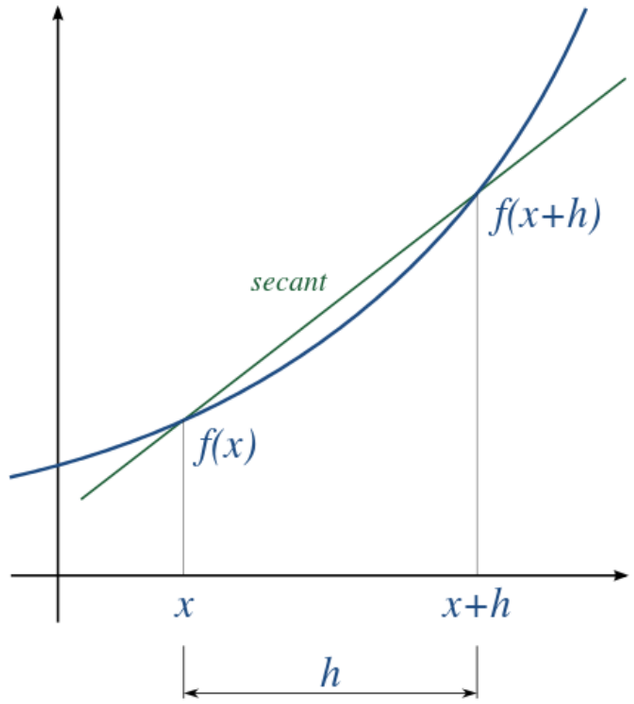
- Παρεμβολή (π.χ. κατασκευή ενός πολυώνυμου p το οποίο να συμφωνεί σε ορισμένα σημεία με μια συνάρτηση f(x)



Θεωρία Προσεγγίσεων

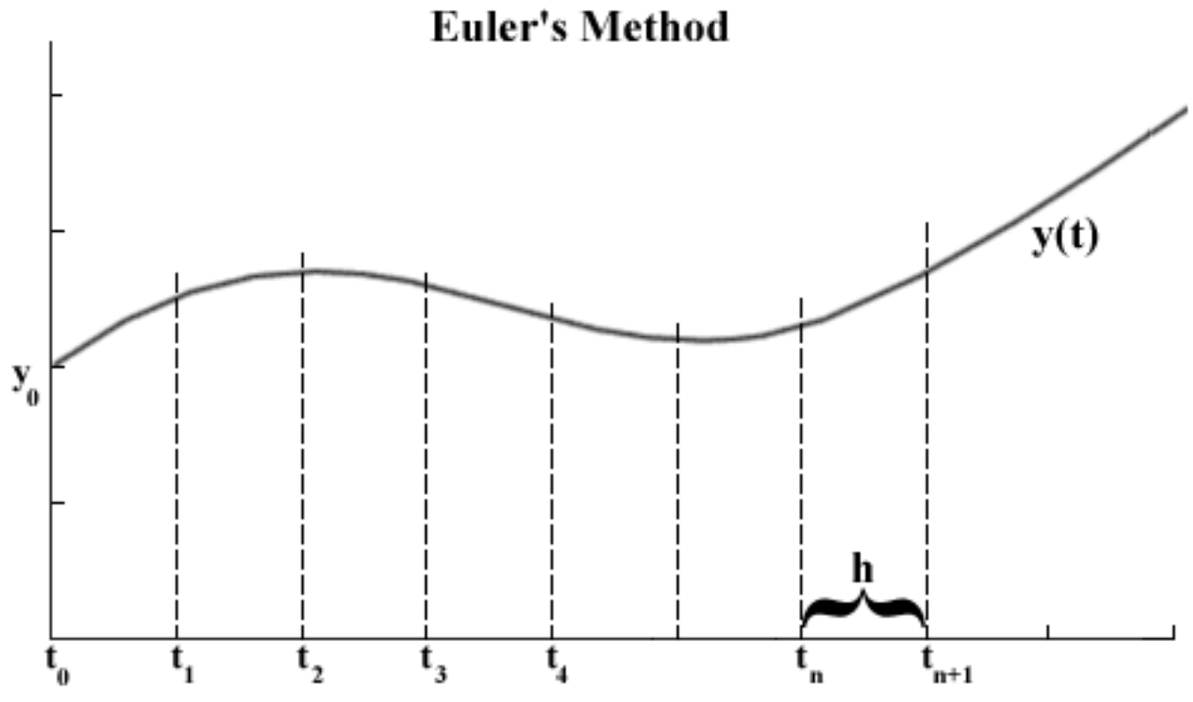
- Αριθμητική ολοκλήρωση και παραγώγιση





Αριθμητική Επίλυση Διαφορικών Εξισώσεων

- Αριθμητική Επίλυση ΣΔΕ



$$\frac{dy}{dt} = f(t,y)$$

$$y(t_0) = y_0$$

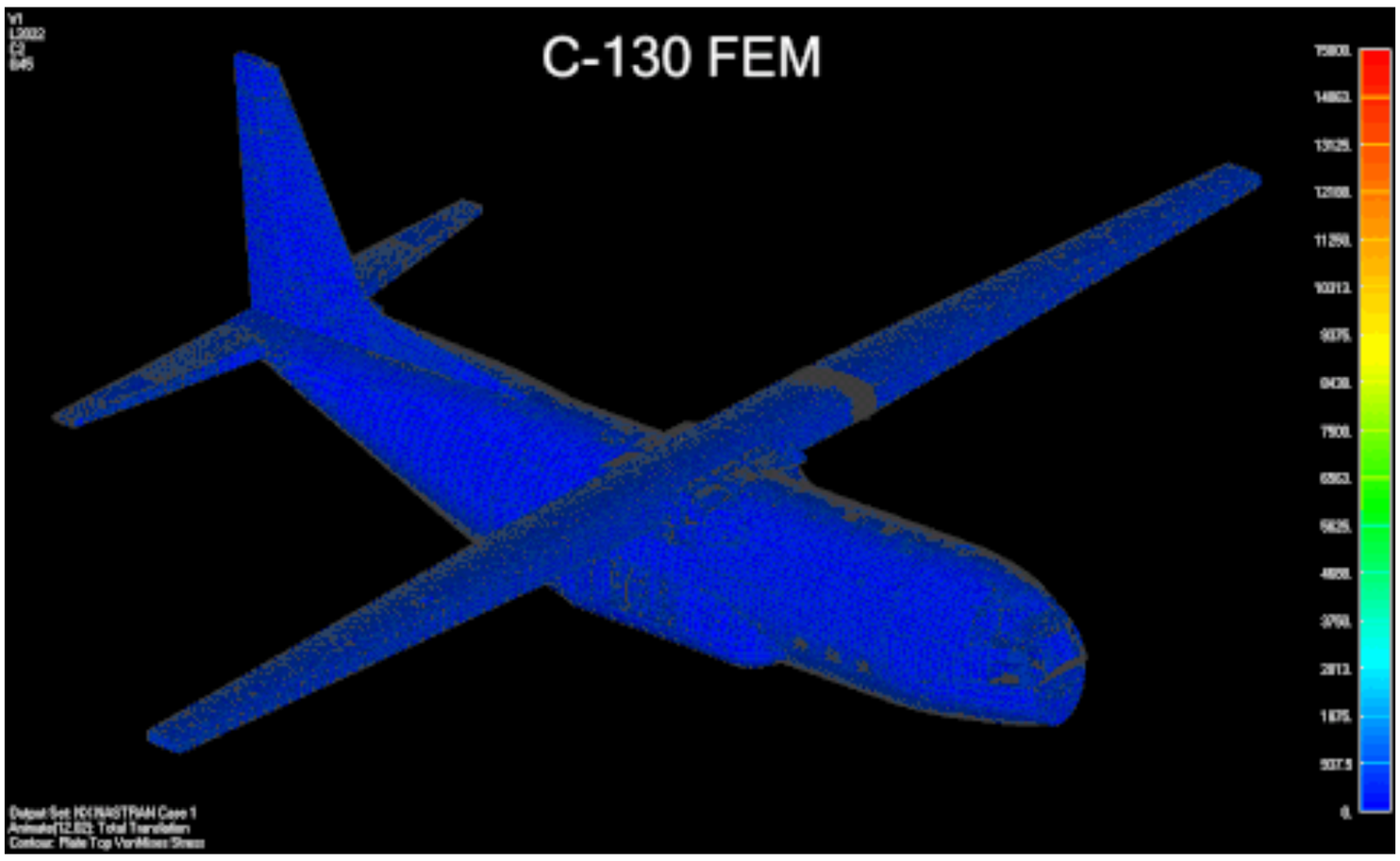
y(t) is the solution of this differential equation

This is Euler's formula to approximate the solutions.

$$y_{n+1} = y_n + hf(t_n, y_n)$$

Αριθμητική Επίλυση ΣΔΕ και ΜΔΕ

Αριθμητική επίλυση ΜΔΕ



Μπορούμε να περιγράψουμε τι εννοούμε Αριθμητική Ανάλυση;

Αριθμητική Ανάλυση *είναι η μελέτη των αλγορίθμων για προβλήματα των "συνεχών μαθηματικών"*

Αλγόριθμοι

Οι αλγόριθμοι βρίσκονται στον πυρήνα της Αριθμητικής Ανάλυσης.

Σκοπός της Αριθμητικής Ανάλυσης είναι η ***κατασκευή*** και ***ανάλυση*** αλγορίθμων για την επίλυση προβλημάτων (μιας ορισμένης κατηγοριάς).

Η λέξη αλγόριθμος

Muḥammad Khwārizmī
(780μ.χ.-850μ.χ.)

- Πέρσης Μαθηματικός
- Θεωρείται ο θεμελιωτής της Άλγεβρας
- Έγραψε το βιβλίο “*Al-jabr*”
- Το όνομα του αραβοποιήθηκε σε **al-Khwarizmi**
- Στη συνέχεια λατινοποιήθηκε σε **Algorithmi**, οπότε προκύπτει και ο όρος **Αλγόριθμος**

Προβλήματα “συνεχών μαθηματικών”

Προβλήματα που αφορούν πραγματικές ή μιγαδικές μεταβλητές

Προβλήματα “συνεχών μαθηματικών”

- Να βρούμε τη λύση ενός γραμμικού συστήματος, $Ax = b$
- Να βρούμε τη ρίζα μιας συνάρτησης, $f(x) = 0$
- Να λύσουμε ένα πρόβλημα ελαχιστοποίησης (ή μεγιστοποίησης) (Βελτιστοποίηση)
- Και άλλα ….

Πραγματικοί Αριθμοί

- Οι αριθμοί **δεν** μπορούν να αναπαρασταθούν **ακριβώς** στον Η/Υ

- Ένα κομμάτι της δουλειάς της Αριθμητικής Ανάλυσης είναι η προσέγγιση των πραγματικών αριθμών (σφάλματα αναπαράστασης) και η διάδοση του σφάλματος μέσω των πράξεων που υλοποιούνται σε έναν αλγόριθμο

Οι αλγόριθμοι διαχωρίζονται σε αυτούς που ολοκληρώνονται σε

- Πεπερασμένο πλήθος βημάτων
- Μη-πεπερασμένο πλήθος βημάτων

Αλγόριθμοι πεπερασμένων βημάτων

- Η επίλυση ενός γραμμικού συστήματος $Ax = b$, με τη μέθοδο απαλοιφής Gauss (Γραμμική Άλγεβρα), ολοκληρώνετε σε ορισμένο αριθμό βημάτων που εξαρτάται από το μέγεθος του πίνακα A

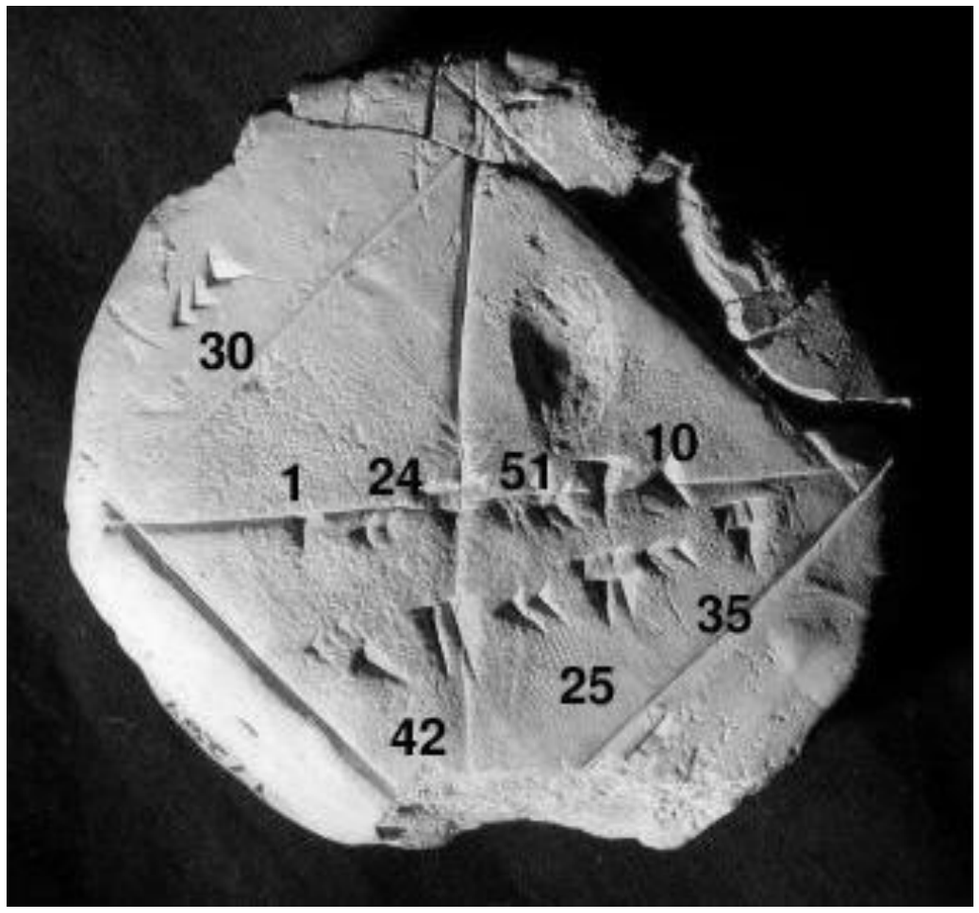
- Τα προβλήματα που αντιμετωπίζουμε εδώ είναι το πλήθος των πράξεων, η αρχιτεκτονική των Η/Υ (παράλληλοι ή όχι), η διάδοση των σφαλμάτων αναπαράστασης αριθμών, κ.α.

Αλγόριθμοι μη πεπερασμένων βημάτων

- Τα πιο πολλά προβλήματα των “συνεχών μαθηματικών”, δεν αντιμετωπίζονται με πεπερασμένων βημάτων αλγορίθμους, ακόμα και χρησιμοποιήσουμε ακριβείς αριθμούς.

- Υπολογισμός: ριζών συναρτήσεων, ολοκληρωμάτων, παραγώγων, βέλτιστων λύσεων

Προσέγγιση άρρητων αριθμών



Η προσέγγιση της $\sqrt{2}$ είναι τέσσερα εξηνταδικά στοιχεία,
1,24,51,10

$$1 + \frac{24}{60} + \frac{51}{60^2} + \frac{10}{60^3} = 1{,}41421296$$

Ηρων ο Αλεξανδρεύς (10μχ-75μχ)

- Γεωμέτρης και μηχανικός

- Επαναληπτική διαδικασία για τον υπολογισμό της τετραγωνικής ρίζας κάποιου αριθμού.

Μέθοδος του Ήρων

- Κατασκευή ακολουθίας αριθμών $x_n \approx \sqrt{A}$
- Αρχικά επιλέγει x_0 τέτοιο ώστε $x_0^2 \approx A$
- Στη συνέχεια κάθε επόμενος όρος $x_{n+1} = \frac{1}{2}(x_n + \frac{A}{x_n})$
- Αν $x_n \to x^\star$ τότε $(x^\star)^2 = A$

Υλοποίηση στον Η/Υ

Συνάρτηση Python

```
def heron(x,y):
    x=(x+y/x)*0.5
    return x
```

Κλήση της heron

```
x=1
y=2
for i in range(5):
   x=heron(x,y)
```

```
Approximation of square root of 2 at step 0: 1.0000000000000000
Approximation of square root of 2 at step 1: 1.5000000000000000
Approximation of square root of 2 at step 2: 1.4166666666666665
Approximation of square root of 2 at step 3: 1.4142156862745097
Approximation of square root of 2 at step 4: 1.4142135623746899
Approximation of square root of 2 at step 5: 1.4142135623730949
```

Στάδια

Υπάρχουν δύο διαφορετικά στάδια που πρέπει να θεωρήσουμε στην Αριθμητική Ανάλυση

- Την **ανάπτυξη** αλγορίθμων

- Την **ανάλυση** αλγορίθμων

Φαίνονται ανεξάρτητα στάδια, όμως πολλές φορές δρουν παράλληλα

Χαρακτηριστικά αλγορίθμων

- Η **ακρίβεια** (ή συνέπεια) του αλγορίθμου
- Η **ευστάθεια** του αλγορίθμου
- Η **επιρροή της πεπερασμένης αριθμητικής** (σφάλματα στρογγύλευσης)

Σφάλματα (Απόλυτο)

Συνάρτηση Python

```
import numpy as np

def heron(x,y):
    x=(x+y/x)*0.5
    return x
def errheron(x,y):
    z=x-np.sqrt(y)
    return z
x=1.
    y=2.
```

Απόλυτο Σφάλμα: $|x_n - \sqrt{(y)}|$

```
Approximation of square root of 2 at step 0: 1.0000000000000000
Absolute error  :-0.4142135623730951
Approximation of square root of 2 at step 0: 1.5000000000000000
Absolute error  :0.085786437626904 9
Approximation of square root of 2 at step 1: 1.4166666666666665
Absolute error  :0.002453104293571 4
Approximation of square root of 2 at step 2: 1.4142156862745097
Absolute error  :0.000002123901414 5
Approximation of square root of 2 at step 3: 1.4142135623746899
Absolute error  :0.000000000001594 7
Approximation of square root of 2 at step 4: 1.4142135623730949
Absolute error  :-0.000000000000000 2
```

Σφάλματα (Απόλυτο)

Συνάρτηση Python

```
import numpy as np

def heron(x,y):
    x=(x+y/x)*0.5
    return x
def errheron(x,y):
    z=x-np.sqrt(y)
    return z
x=1.*10**10
    y=2.*10**20
```

Απόλυτο Σφάλμα: $|x_n - \sqrt{(y)}|$

```
Approximation of square root of 2.00000e+20 at step 0: 10000000000.000000000000000
Absolute error  :-4142135623.730951309204101 6
Approximation of square root of 2.00000e+20 at step 0: 15000000000.000000000000000
Absolute error  :857864376.2690486907958984
Approximation of square root of 2.00000e+20 at step 1: 14166666666.666667938232421 9
Absolute error  :24531042.935716629028320 3
Approximation of square root of 2.00000e+20 at step 2: 14142156862.745098114013671 9
Absolute error  :21239.014146804809570 3
Approximation of square root of 2.00000e+20 at step 3: 14142135623.746898651123046 9
Absolute error  :0.015947341918945 3
Approximation of square root of 2.00000e+20 at step 4: 14142135623.730949401855468 8
Absolute error  :-0.000001907348632 8
```

Σφάλματα (Σχετικό)

Συνάρτηση Python

```
import numpy as np

def heron(x,y):
    x=(x+y/x)*0.5
    return x
def relerrheron(x,y):
    z=x/np.sqrt(y)-1
    return z
x=1.*10**10
y=2.*10**20
```

Σχετικό Σφάλμα: $\frac{|x_n - \sqrt{y}|}{|\sqrt{y}|}$

```
Approximation of square root of 2e+20 at step 0: 10000000000.000000000000000
Relative error  :-0.2928932188134525
Approximation of square root of 2e+20 at step 0: 15000000000.000000000000000
Relative error  :0.060660171779821 2
Approximation of square root of 2e+20 at step 1: 14166666666.666667938232421 9
Relative error  :0.001734606680942 3
Approximation of square root of 2e+20 at step 2: 14142156862.745098114013671 9
Relative error  :0.000001501825093 0
Approximation of square root of 2e+20 at step 3: 14142135623.746898651123046 9
Relative error  :0.000000000001127 5
Approximation of square root of 2e+20 at step 4: 14142135623.730949401855468 8
Relative error  :-0.000000000000000 1
```

Άλλη Μέθοδος για $\sqrt{A}$

- Κατασκευή ακολουθίας αριθμών $x_n \approx \sqrt{A}$
- Αρχικά επιλέγει x_0 τέτοιο ώστε $x_0^2 \approx A$
- Στη συνέχεια κάθε επόμενος όρος $x_{n+1} = x_n + x_n^2 - A$
- Αν $x_n \to x^\star$ τότε $(x^\star)^2 = A$

Σφάλματα (Απόλυτο)

Συνάρτηση Python

```
import numpy as np

def other(x,y):
    x=x+x**2-y
    return x
def errheron(x,y):
    z=x-np.sqrt(y)
    return z
x=1.5
y=2.
```

Απόλυτο Σφάλμα: $|x_n - \sqrt{(y)}|$

Η μέθοδος δεν συγκλίνει!

```
Approximation of square root of 2 at step 0: 1.5000000000000000
Absolute error  :0.0857864376269049
Approximation of square root of 2 at step 1: 1.7500000000000000
Absolute error  :0.3357864376269049
Approximation of square root of 2 at step 2: 2.8125000000000000
Absolute error  :1.3982864376269049
Approximation of square root of 2 at step 3: 8.7226562500000000
Absolute error  :7.3084426876269051
Approximation of square root of 2 at step 4: 82.8073883056640625
Absolute error  :81.3931747432909702
Approximation of square root of 2 at step 5: 6937.8709463106933981
Absolute error  :6936.4567327483200643
```