



image credits Jason Allen

Υπολογιστική Φυσική — 10ΥΚΟ21

Τμήμα Φυσικής - ΕΚΠΑ

2025-2026

Κώστας Θεοφιλάτος & Δ. Φασουλιώτης [Τμήμα Β]



γιατί;

Τα περισσότερα προβλήματα δεν λύνονται αναλυτικά

δηλαδή σε “κλειστή μορφή”, ως μια πεπερασμένη ακολουθία από βασικές πράξεις (+ - * /) μεταξύ γνωστών αριθμών και “γνωστών” συναρτήσεων (**e^x , $\sin x$, $\cos x$, $\log x$...**)

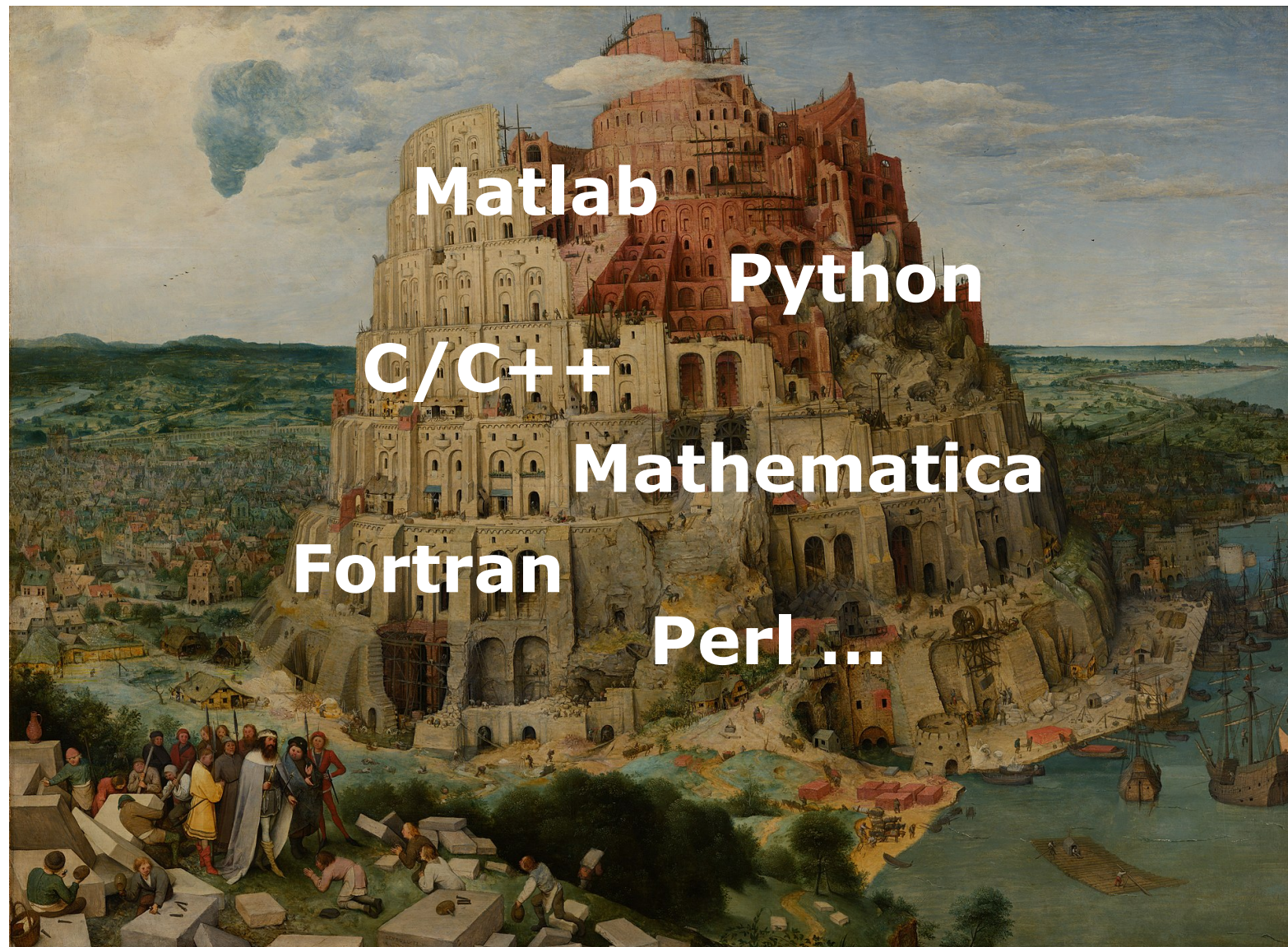
πχ

$$ax^2 + bx + c = 0 \rightarrow x = [-b \pm (b^2 - 4ac)^{0.5}]/2a$$

$$dN/dt = -N\lambda \rightarrow N(t) = N_0 e^{-\lambda t}$$

Αλήθεια πως θα υπολογίζεται το **e^x** για κάποια τιμή του **x** , αν είχατε στην διαθεσή σας μόνο μολύβι και χαρτί ?

ποια γλώσσα;



https://en.wikipedia.org/wiki/Category:Numerical_programming_languages

https://en.wikipedia.org/wiki/List_of_numerical-analysis_software

>60 υπολογιστικά
περιβάλλοντα
εργασίας

οποιαδήποτε γλώσσα

- 1) **ασκήσεις σε H/Y:** μπορείτε να χρησιμοποιήσετε όποια γλώσσα προγραμματισμού θέλετε
- 2) **εξεταστική:** οποιαδήποτε γλώσσα προγραμματισμού ή ψευδοκώδικα

δε “μιλάω” καμία γλώσσα

τι κάνω?

μαθαίνω μια γλώσσα (!)

προσπαθώντας να λύσω απλά προβλήματα

ανακυκλώνω + προσαρμόζω κώδικα από άλλους

αναζήτηση στην google

πχ

<https://github.com/theofil/CompPhysics/tree/master/examples>



python στο νέφος

☰ README.md



Computational Physics -- Υπολογιστική Φυσική 10ΥΚΟ21 [↗](#)

Cloud-based **interactive** computing environment for the Computational Physics [Υπολογιστική Φυσική 10ΥΚΟ21](#) course (3rd semester) at National and Kapodistrian University of Athens (2019-2024).

Run Python without installing it [↗](#)

- Use the [binder enviroment created for this course](#) [↖](#)
- Use <http://colab.research.google.com> (recommended) [↖](#)

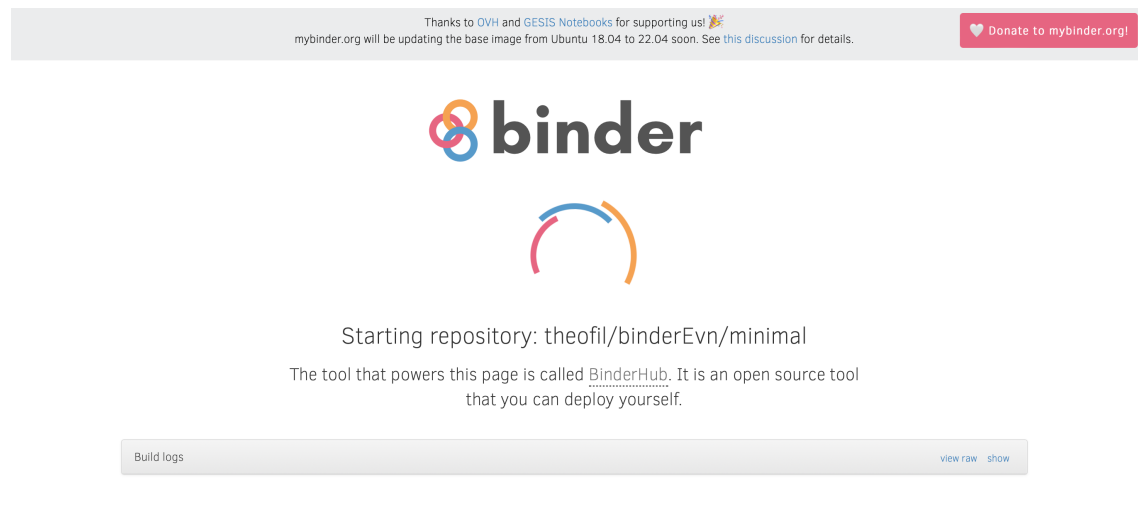
δοκιμάστε αυτά τα 2
links στο github του
μαθήματος

Visit <http://theofil.web.cern.ch/theofil/courses/compPhysics/> for slides, notes, YouTube and other resources related to this course.

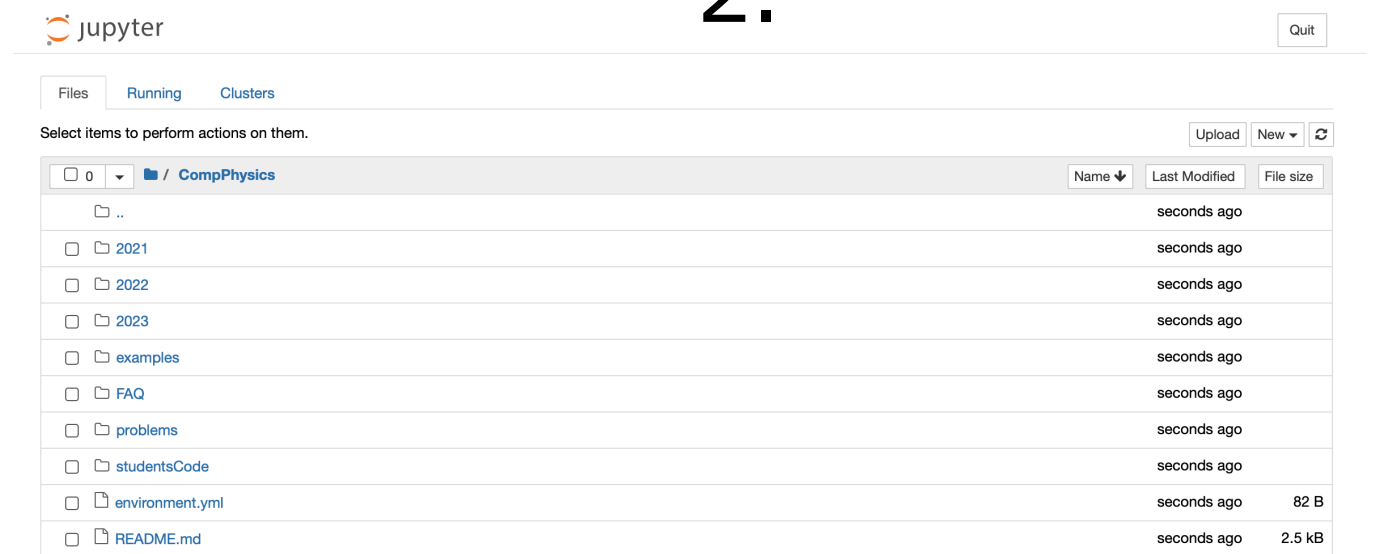
<https://github.com/theofil/CompPhysics>

python σε 4 βήματα

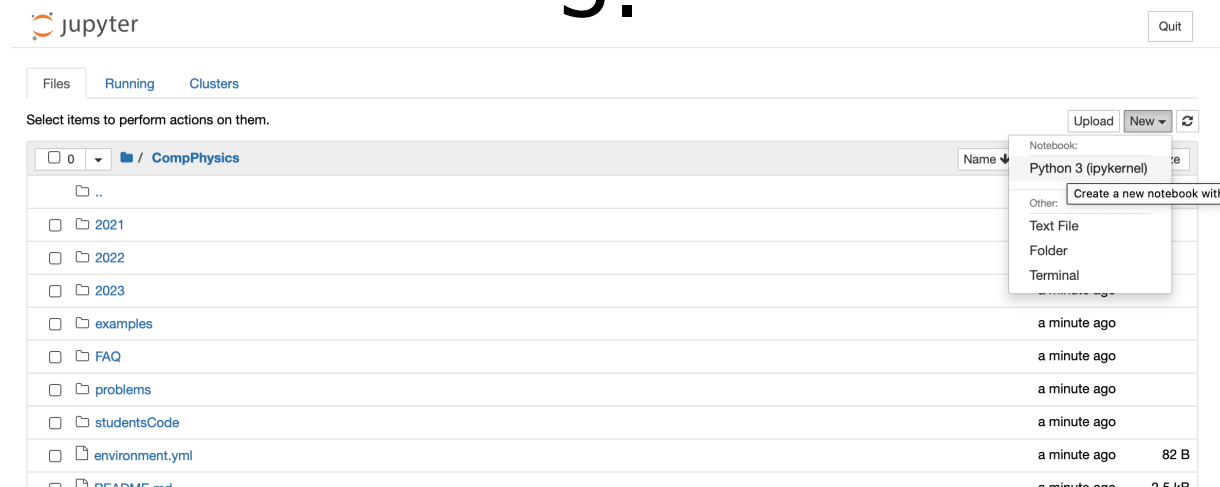
1. (περιμένουμε να φορτώσει)



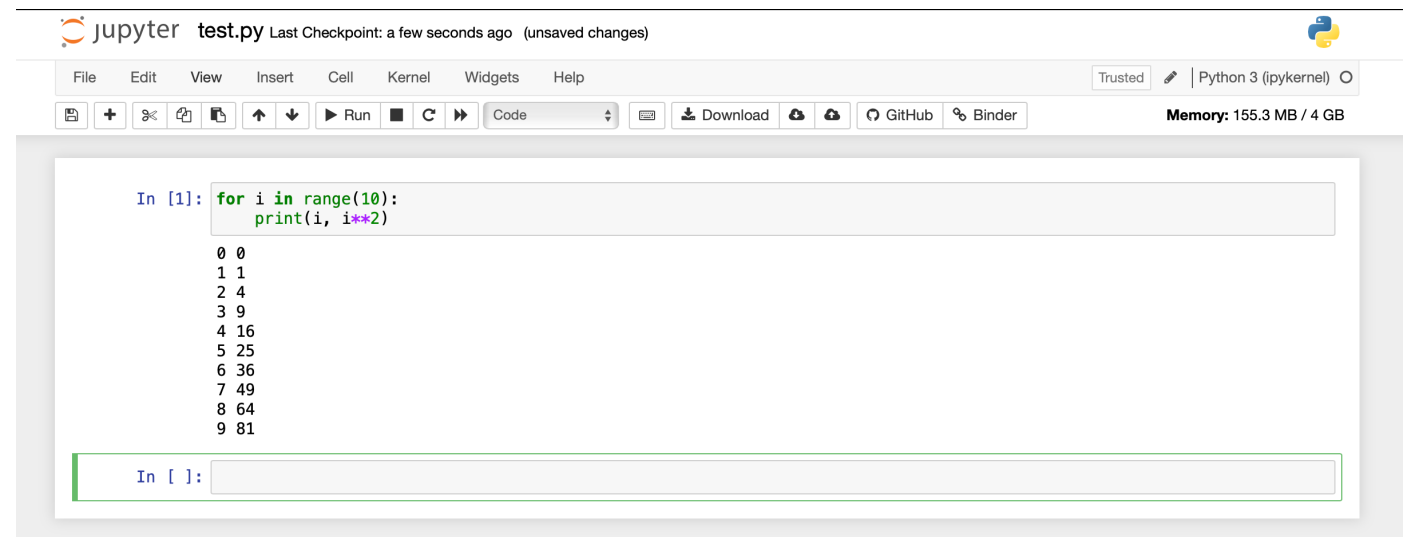
2.



3.



4.



πατήστε σε αυτό το [link](#) για να μεταβείτε στο binder με το περιβάλλον του μαθήματος της ΥΦ (μη ξεχάσετε να σώσετε στο σκληρό σας δίσκο το πρόγραμμα που θα γράψετε File → Download As → Notebook (*.ipynb))

μόνιμες λύσεις

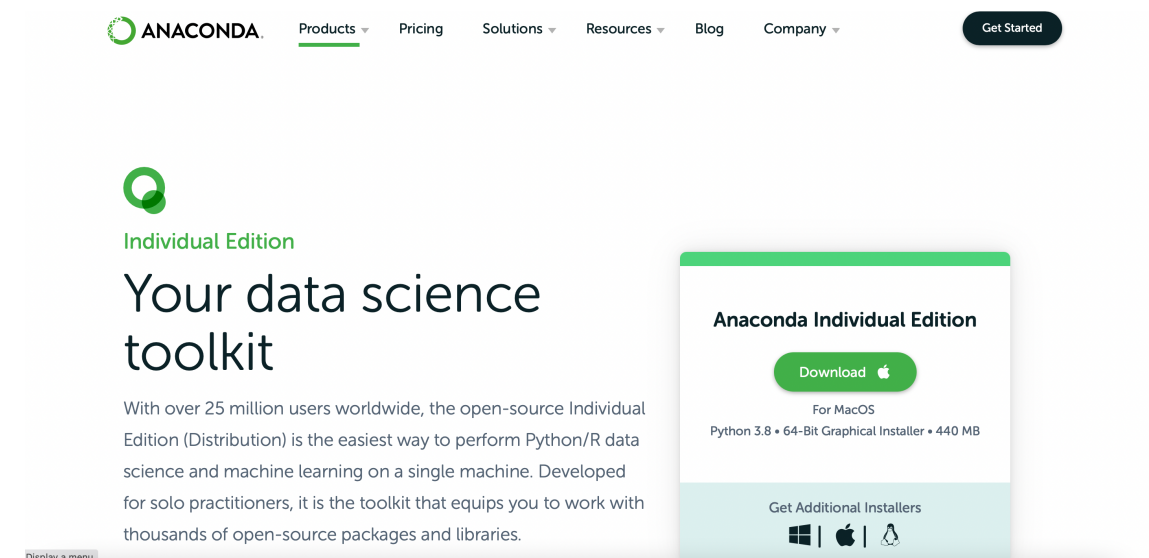
εγκατάσταση της “γλώσσας” στον Η/Υ σας —> δε χρειάζεται σύνδεση στο internet για να τρέξετε τα αγαπημένα σας προγράμματα

>1 τρόποι:

python και **C/C++** (built-in σε ~Unixοείδη συστήματα πχ linux, MacOS, BSD)

εγκατάσταση **python, jupyter lab, C++/C, ROOT** ... μέσω ΤΟΥ **anaconda**

<https://www.anaconda.com>



ROOT data analysis (CERN)



ROOT
Data Analysis Framework

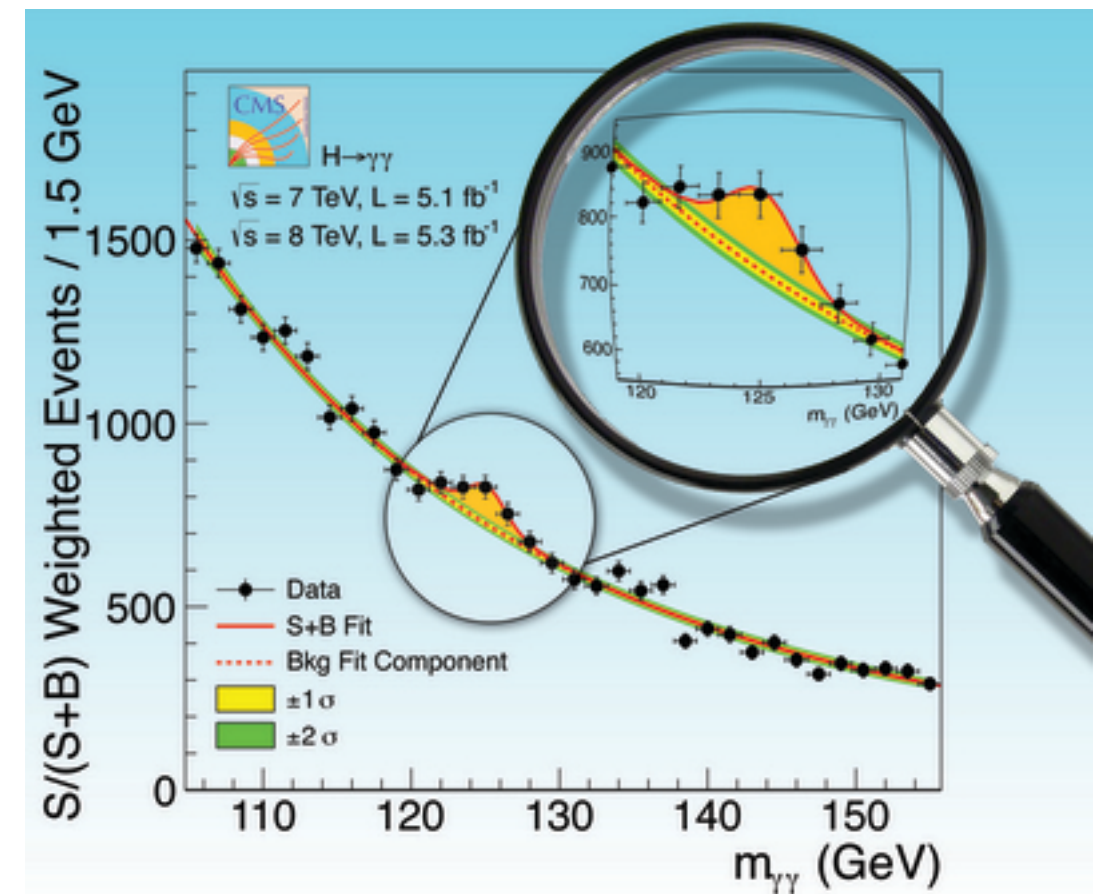
[About](#)[Install](#)[Get Started](#)[Forum & Help](#)[Manual](#)[Blog Posts](#)[Contribute](#)[For Developers](#)

ROOT: analyzing petabytes of data, scientifically.

An open-source data analysis framework used by high energy physics and others.

[Learn more](#)[Install v6.24/06](#)

ανοιχτό λογισμικό
υποστηρίζει **C/C++** και **python**
<https://root.cern.ch/>
διαθέσιμο και στο  ANACONDA.



στόχοι της ~εβδομάδας

εξοικείωση με H/Y και συγγραφή κώδικα

βρόγχοι, γεννήτριες τυχαίων αριθμών, κατανομές

2D διαγράμματα [(x,y) scatter plots]

1D ιστογράμματα/κατανομές πυκν. πιθανότητας [histogram]

αναμενόμενες τιμές κάποιας τυχαίας μεταβλητής

μέση τιμή (μ), τετραγωνική διασπορά (σ^2), τυπική απόκλιση (σ)

εκτιμητές — estimators

αριθμητικός μ.ο., (διορθωμένη) διακύμανση δείγματος

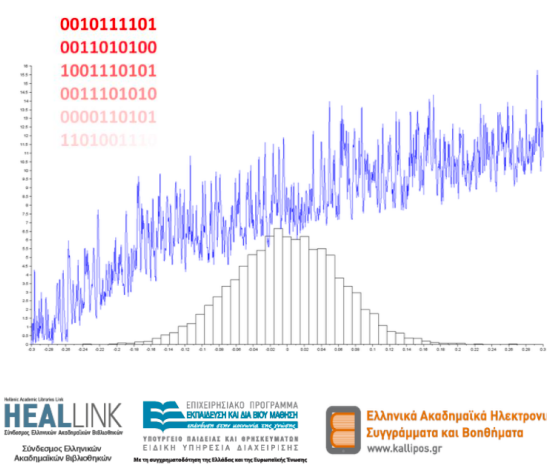
για το επόμενο μάθημα (03/10)

Ανασκόπηση των:

6, 7, 10, 11 και 12

ΣΤΟΙΧΕΙΑ
ΠΙΘΑΝΟΤΗΤΩΝ
ΜΕ ΕΦΑΡΜΟΓΕΣ ΣΤΗ ΣΤΑΤΙΣΤΙΚΗ
ΚΑΙ ΤΗΝ ΠΛΗΡΟΦΟΡΙΚΗ

Γιάννης Κοντογιάννης
Σταύρος Τουμπής



vi ΠΕΡΙΕΧΟΜΕΝΑ

6 Διακριτές τυχαίες μεταβλητές	63
6.1 Ορισμός και βασικές ιδιότητες	63
6.2 Μέση τιμή, διασπορά, ανεξαρτησία	70
6.3 * Μετρησιμότητα και άπειρες τιμές	77
6.3.1 Ορισμός τυχαίας μεταβλητής	77
6.3.2 Ορισμοί $E(X)$, $\text{Var}(X)$, και επιπλέον συνθήκες	78
6.4 Ασκήσεις	80
7 Διακριτές κατανομές	85
7.1 Κατανομές Bernoulli, διωνυμική και γεωμετρική	85
7.2 Υπεργεωμετρική και Poisson κατανομή	94
7.3 * Η γεωμετρική και συναφείς σειρές	101
7.4 * Η συνάρτηση e^x , δυναμοσειρές, τύπος του Stirling	102
7.5 Ασκήσεις	106
8 Παραδείγματα πιθανοκρατικής ανάλυσης αλγορίθμων	111
8.1 Επαλήθευση ισότητας πολυωνύμων	112
8.2 Εύρεση ελαχιστιαίου cut set σε γράφους	114
8.3 String matching	121
8.4 Γρήγορη ταξινόμηση δεδομένων	123
8.5 Ασκήσεις	129
9 Από κοινού κατανομή και ο N.M.A.	133
9.1 Ανισότητες Markov και Chebychev	133
9.2 Από κοινού κατανομή και συνδιακύμανση	135
9.3 Ο Νόμος των Μεγάλων Αριθμών	144
9.4 Ασκήσεις	150
10 Συνεχείς τυχαίες μεταβλητές	155
10.1 Συνεχείς T.M. και συνεχής πυκνότητα	156
10.2 Μέση τιμή και διασπορά	163
10.3 * Μετρησιμότητα και άπειρες τιμές	168
10.3.1 Ορισμός μιας συνεχούς T.M.	168
10.3.2 Ορισμοί $E(X)$, $\text{Var}(X)$, και επιπλέον συνθήκες	170
10.4 Ασκήσεις	171
11 Συνεχείς κατανομές και ο N.M.A.	175
11.1 Ομοιόμορφη και εκθετική κατανομή	175
11.2 Μετασχηματισμοί	181
11.3 Ανεξαρτησία, ανισότητες και ο N.M.A.	183
11.4 Ασκήσεις	187

Θεωρία Πιθανοτήτων - 10ΥΚΟ13

<https://eclass.uoa.gr/courses/PHYS369/>

το πρόβλημα της ημέρας

Έστω το δείγμα δεδομένων

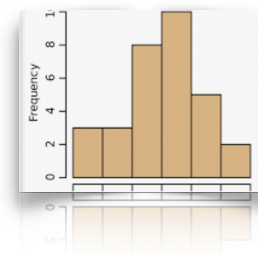
$$x = \{3, 2, 4, 1, 2, 3, 4, 3, 4, 4\}$$

το "x" συνοψίζει τα αποτελέσματα των διαδοχικών ρίψεων ενός ζαριού

α) να βρεθεί ο αριθμητικός μ.ο. του δείγματος $\bar{x}$

β) να βρεθεί η τυπική απόκλιση του δείγματος $\sqrt{\hat{\sigma}_x^2}$

γ) να φτιαχτεί ιστόγραμμα του x



το πρόγραμμα

```
x = [3, 2, 4, 1, 2, 3, 4, 3, 4, 4 ]
```

```
Σx = 0
```

```
for i in range(10):
```

```
    Σx = Σx + x[i]
```

```
    print(i, x[i], Σx)
```

όποια γραμμή έχει στοίχιση με κενά (ή tabs) στην αρχή

θεωρείται από τον “ερμηνευτή” εντός του

βρόγχου (for loop)

google search for [python indentation](https://www.w3schools.com/python/gloss_python_indentation.asp)

https://www.w3schools.com/python/gloss_python_indentation.asp

καλύτερο πρόγραμμα

```
xs = [3, 2, 4, 1, 2, 3, 4, 3, 4, 4 ]
```

```
Σx = 0
```

```
for x in xs:
```

```
    Σx = Σx + x
```

```
    print(i, x, Σx)
```

όποια γραμμή έχει στοίχιση με κενά (ή tabs) στην αρχή

θεωρείται από τον “ερμηνευτή” εντός του

βρόγχου (for loop)

google search for [python indentation](https://www.w3schools.com/python/gloss_python_indentation.asp)

https://www.w3schools.com/python/gloss_python_indentation.asp

```
#include <stdio.h>      /* printf, scanf, puts, NULL */
#include <math.h>        /* for the sqrt funcion */

int main ()
{
    float x [] = {3, 2, 4, 1, 2, 3, 4, 3, 4, 4};
    unsigned int N = sizeof(x)/sizeof(x[1]); // N = 10, but in that way we can add/remove elements

    float sumX  = 0;
    float sumX2 = 0;

    for(unsigned int i = 0 ; i < N; ++i)
    {
        float z= x[i];
        sumX  += z;
        sumX2 += z*z;
    }

    float meanX  = sumX/N;
    float sigma2 = sumX2/N - meanX*meanX;
    float sigma  = sqrt(sigma2);

    printf("N = %d    mean = %2.3f    sigma = %2.3f \n", N, meanX, sigma);

    return 0;
}

/*
outputs
N = 10    mean = 3.000    sigma = 1.000
*/
```

Βρείτε τα παραδείγματα που συζητήσαμε στην τάξη σε python καθώς και υλοποιήσεις σε C (με ή χωρίς ROOT) στο <https://github.com/theofil/CompPhysics/tree/master/>

theofil / **CompPhysics** Public

Notifications Fork 13 Star 13

Code Issues Pull requests Actions Projects Security Insights

master 2 Branches 0 Tags Go to file Code

Konstantinos Theofilatos adding lec1 2024 ✓ c20bdb8 · 3 minutes ago 252 Commits

2021	adding crudeMCexample from lecture 8	3 years ago
2022	adding graphics to the 3D sphere	2 years ago
2023	problem 22	last year
2024/lec1	adding lec1 2024	3 minutes ago
FAQ	updating readme	last year
examples	ensuring float arrayas in LU	9 months ago
problems	fixing filenames of 2022 problems	9 months ago
studentsCode	Add files via upload	last year
README.md	updating readme	last year

About

Material for my course of Computational Physics (3rd semester, obligatory), in National and Kapodistrian University of Athens

Readme Activity 13 stars 4 watching 13 forks Report repository

Releases

No releases published

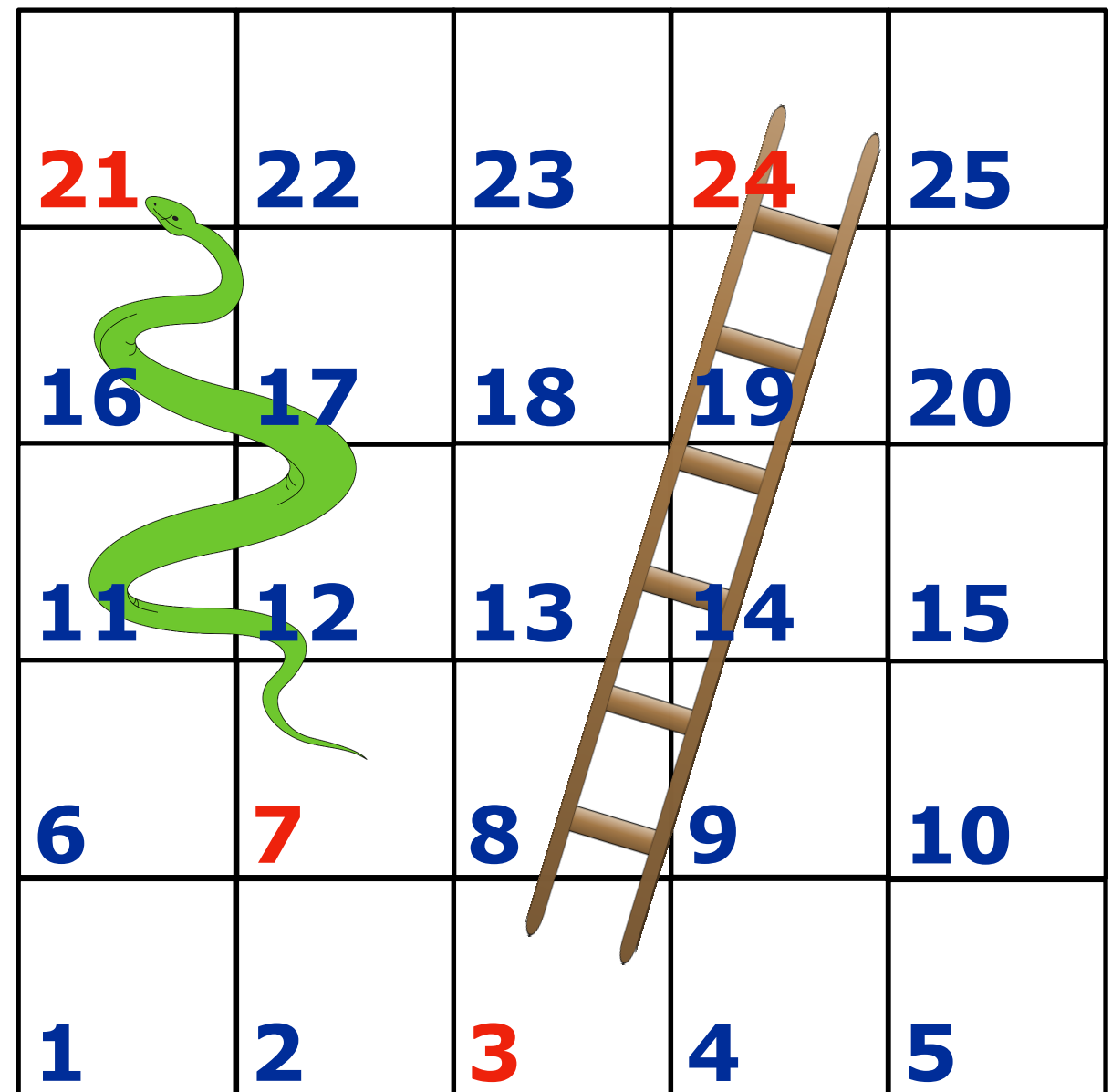
Packages

φιδάκι

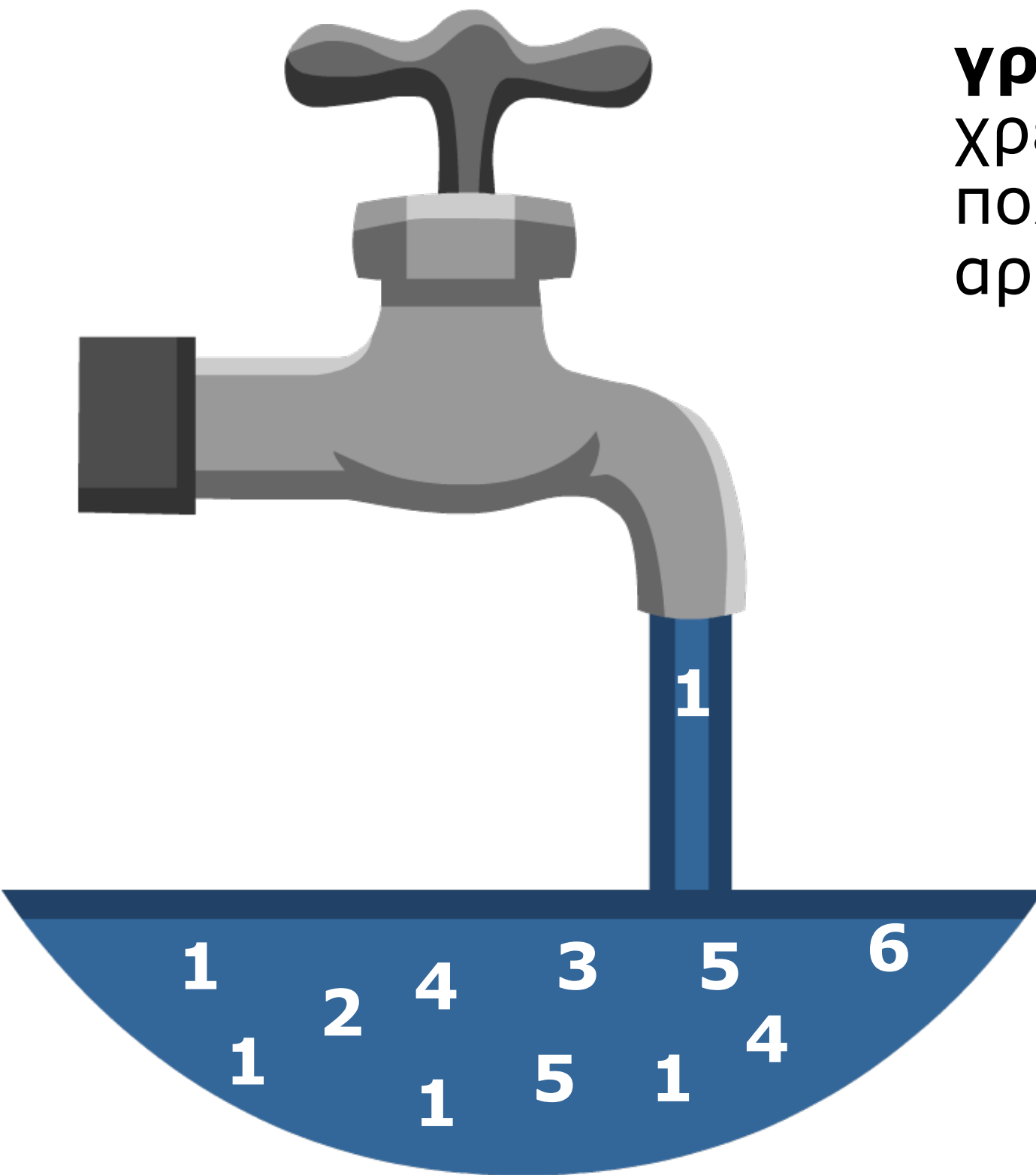
πόσες ζαριές $n \pm \delta n$ πρέπει να ρίξουμε για να κερδίσουμε ?

απάντηση: φτιάχνουμε ένα πρόγραμμα που να "παίζει" φιδάκι και το βάζουμε να παίξει πολλές φορές ...

πόσο μεγάλη είναι η πιθανότητα να κερδίσει αυτός που ξεκινά δεύτερος, όταν παίζουν δύο παίκτες ? όταν παίζουν δέκα παίκτες ?



γεννήτριες τυχαίων αριθμών



γρήγορες μηχανές
χρειάζονται (γρήγορα)
πολλούς τυχαίους
αριθμούς

τυχαίους αριθμούς
με κατά παραγγελία
συχνότητα εμφάνισης
(πιθανότητα
παραγωγής)

ντετερμινιστικούς
(ψευδο-τυχαίους) με
ταυτόσημες ακολουθίες
TA για ίδιο random seed

linear congruential method

$$\mathbf{X}_n = (c\mathbf{X}_{n-1} + a) \bmod \mathbf{N}_{\max}$$

γεννήτρια (ψευδο)τυχαίων ακεραίων αριθμών στο $[1, N_{\max})$

πχ

γραμμικός μετασχηματισμός ισοδυναμίας υπολοίπου

$$\mathbf{X}_{n+1} = (16807 * \mathbf{X}_n) \bmod (2^{31} - 1)$$

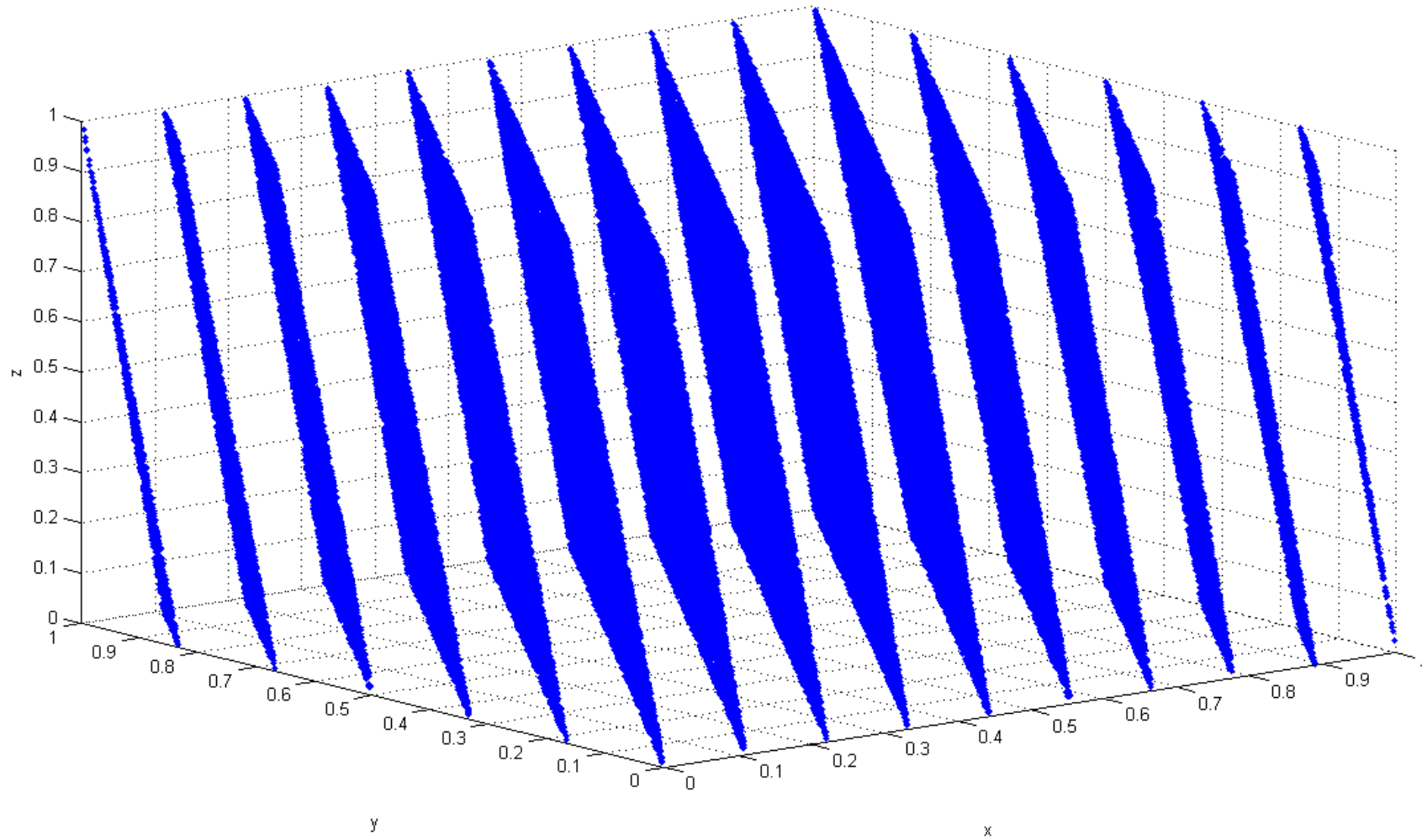
παράγει ΤΑ στο διάστημα $[0, 2^{31} - 1)$

διαιρώντας με $2^{31} - 1$ το $[0, 2^{31} - 1)$

“συμπιέζεται” στο $[0, 1)$

https://en.wikipedia.org/wiki/Linear_congruential_generator

πόσο τυχαία είναι η ακολουθία?



IBM RANDU

$$\mathbf{X}_{n+1} = (65539 * \mathbf{X}_n) \bmod (2^{31})$$

`/dev/random`

~Unix συστήματα

random seed (= τυχαίος σπόρος, εναρκτήριο λάκτισμα)

γεννήτρια τυχαίων στο $[0,1)$

python:

```
import numpy as np  
U = np.random.rand()
```

C/C++:

```
#include <stdlib.h> /* srand, rand */  
U = (double) rand() / (RAND_MAX);
```



**ΤΥΧΑΙΟΚΡΑΤΙΚΟΙ
ΑΛΓΟΡΙΘΜΟΙ
Monte-Carlo**
σύνοψη & εφαρμογές

Monte Carlo

Μέθοδοι που κάνουν χρήση **τυχαίων αριθμών**:

εξομοίωση πειραμάτων φυσικής σε H/Y πχ:

εκτίμηση αβεβαιότητας (μέτρηση του **g** στο εργαστήριο)

ραδιενεργές διασπάσεις

ρυθμός παραγωγής Higgs boson στο LHC

επίλυση προβλημάτων που δεν έχουν τίποτα τυχαίο πχ:

μάζας σώματος δοσμένης της πυκνότητάς του **$\rho(\mathbf{x}, \mathbf{y}, \mathbf{z})$**

εύρεση του ηλεκτρικού δυναμικού στο χώρο

Εάν μπορούμε να εξομοιώσουμε την φυσική διεργασία
δε χρειάζεται να λύσουμε τις εξισώσεις που διέπουν την
χρονική εξέλιξη του συστήματος

στατιστική ανάλυση

Η πυκνότητα πιθανότητας **pdf** δεν είναι πάντα **a-priori** γνωστή, μπορούμε όμως να μάθουμε για αυτήν αρκεί να έχουμε δυνατότητα να κάνουμε **δειγματοληψία**

1. μέσω εξομοίωσης σε H/Y (**MC simulation**)
2. μέσω επανάληψης του πραγματικού πειράματος (**data**)

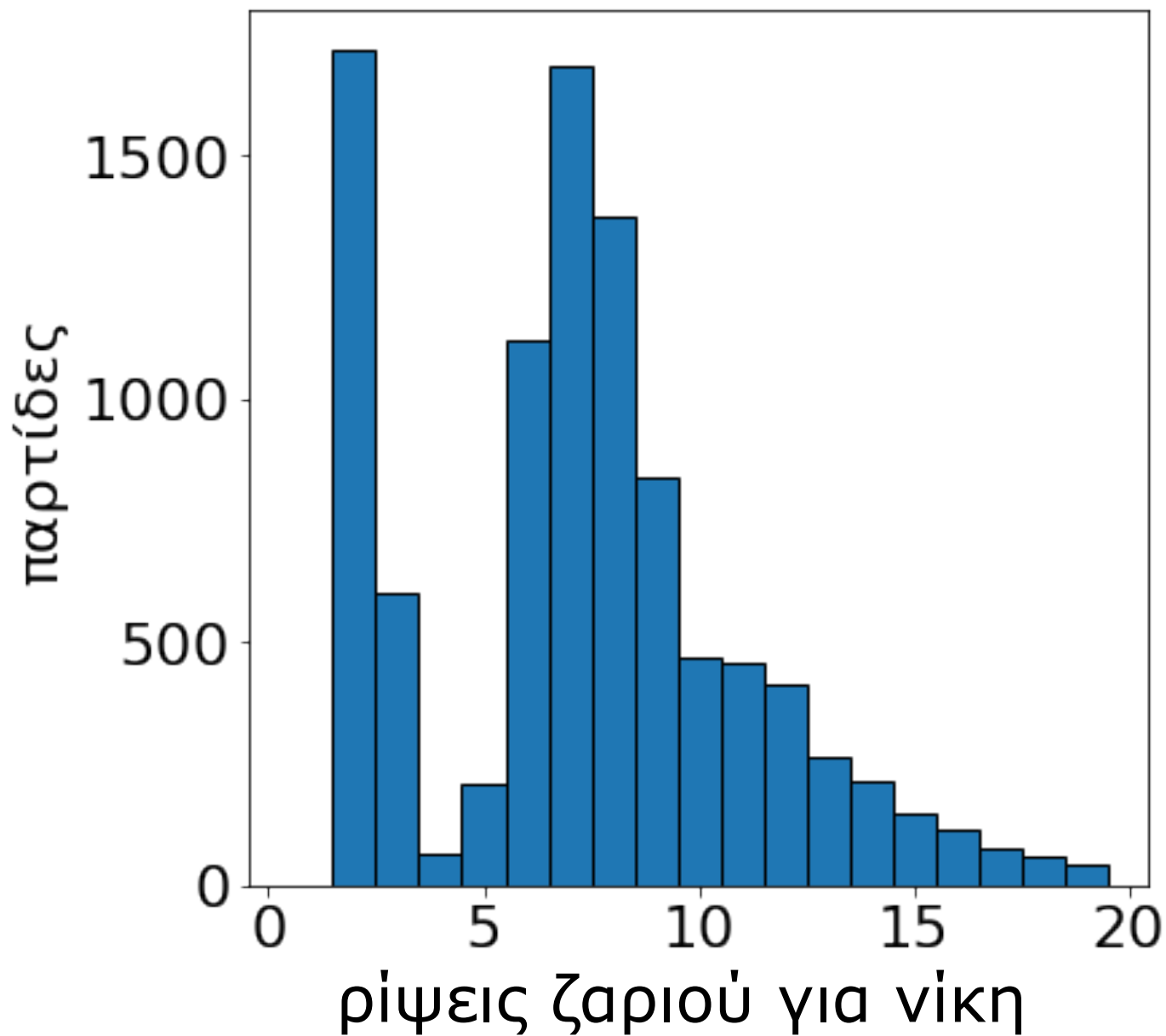
MC εξομοίωση

κατανοώ → μπορώ και προβλέπω τι θα γίνει μέσω υπολογιστικής εξομοίωσης

βρίσκω την λύση → κατασκευάζω αλγόριθμο που βρίσκει τη λύση

δεν υπάρχει τίποτα πιο διασκεδαστικό από την εξομοίωση ενός φυσικού συστήματος και την εξέτασή του υπό ακραίες συνθήκες

ΠΥΚΝΟΤΗΤΑ ΠΙΘΑΝΟΤΗΤΑΣ



$n = 10000$ (παιχνίδια)

x = ρίψεις ζαριού για νίκη

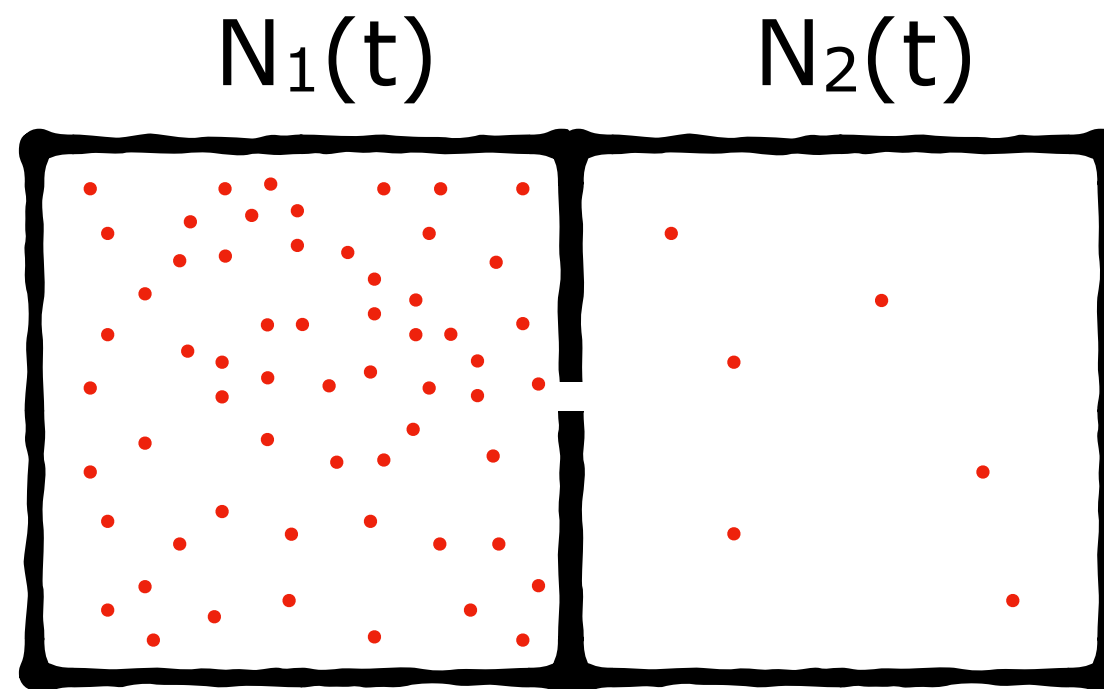
$\bar{x} = 7.5 \pm 4.2$ ζαριές

$$\bar{x} = \frac{1}{10000} \sum_{i=1}^{10000} x_i = 7.5$$

$$\hat{\sigma}_x^2 = \frac{1}{10000 - 1} \sum_{i=1}^{10000} (x_i - \bar{x})^2 = 17.8$$

$$\hat{\sigma}_x = 4.2$$

1. Ο μέσος και η τυπική απόκλιση του δείγματος, υπολογίζονται πάντα (αλλά δεν έχουν πάντα νόημα)
2. η πυκνότητα πιθανότητας "κουβαλά" ό,τι θα θέλαμε να ξέρουμε
3. ιστόγραμμα $\sim$ ο ποιο απλός εκτιμητής πυκνότητας πιθανότητας



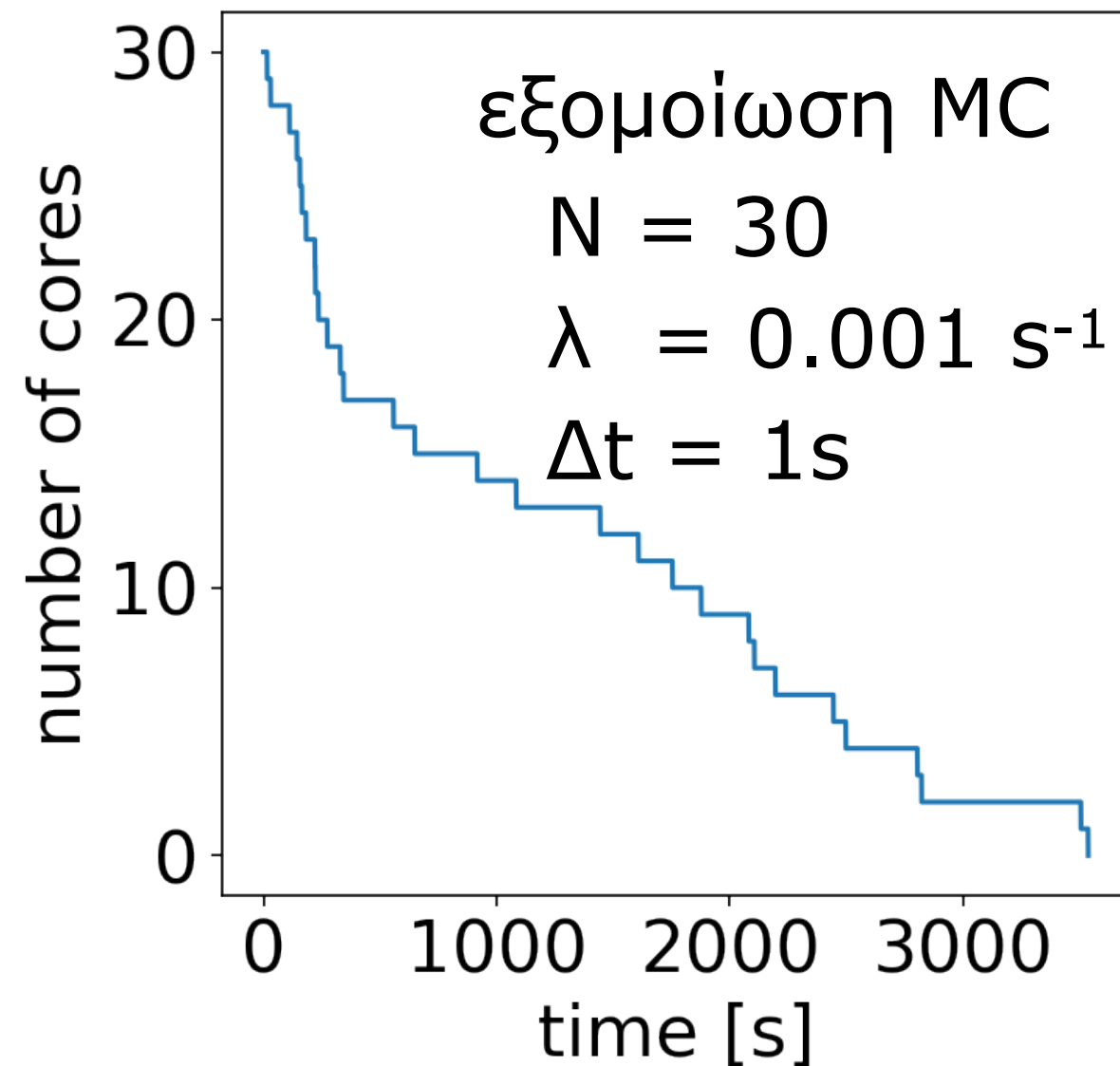
να βρεθεί η χρονική εξέλιξη των δύο πληθυσμών $N_1(t)$ και $N_2(t)$ αν:

$$N_1(0) = 1000$$

$$N_2(0) = 0$$

$p = \lambda \Delta t = 1\%$ πιθανότητα για κάθε μόριο του αερίου να διασχίσει την τρύπα και να περάσει από το ένα δοχείο στο άλλο στο χρονικό διάστημα $[t, t + \Delta t]$

διάσπαση πυρήνων



$\lambda \Delta t$ = πιθανότητα διάσπασης
ενός πυρήνα στο διάστημα Δt

ανά μονάδα χρόνου, κάθε πυρήνας περνάει από δοκιμή
Bernoulli (ανεξαρτησία διασπάσεων, άνευ μνήμης)

Ολοκλήρωση MC

Υπολογισμός ορισμένων ολοκληρωμάτων με MC:

Απλοϊκό MC (crude MC)

MC Απόρριψης (Accept-Reject MC, hit-or-miss)

Έξυπνο MC (σημαντική δειγματοληψία / importance sampling)

εὕρεση ριζῶν της
 $f(x) = 0$

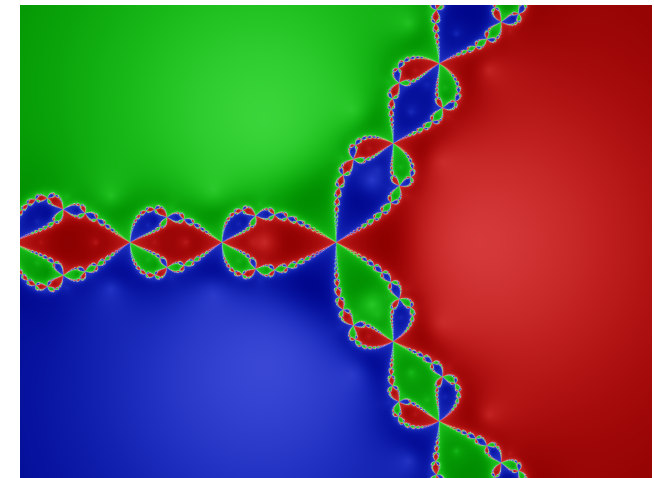
αλγόριθμοι με ιστορία



Secant
~17ος πΧ



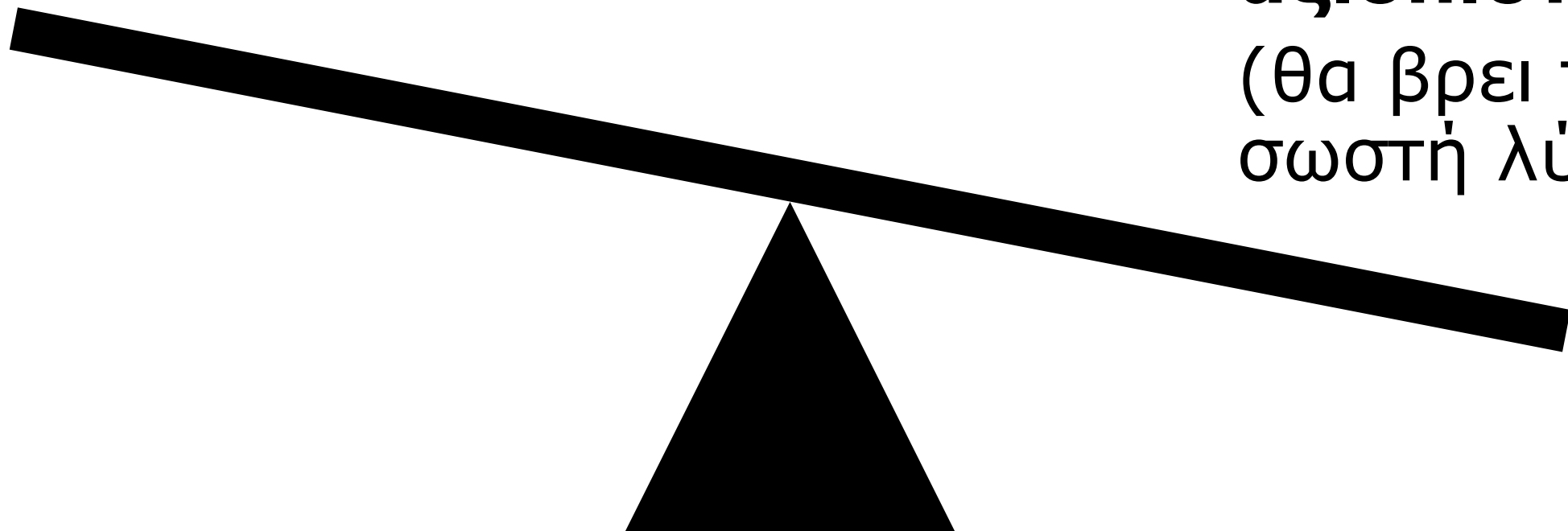
Babylonian
~600 πΧ



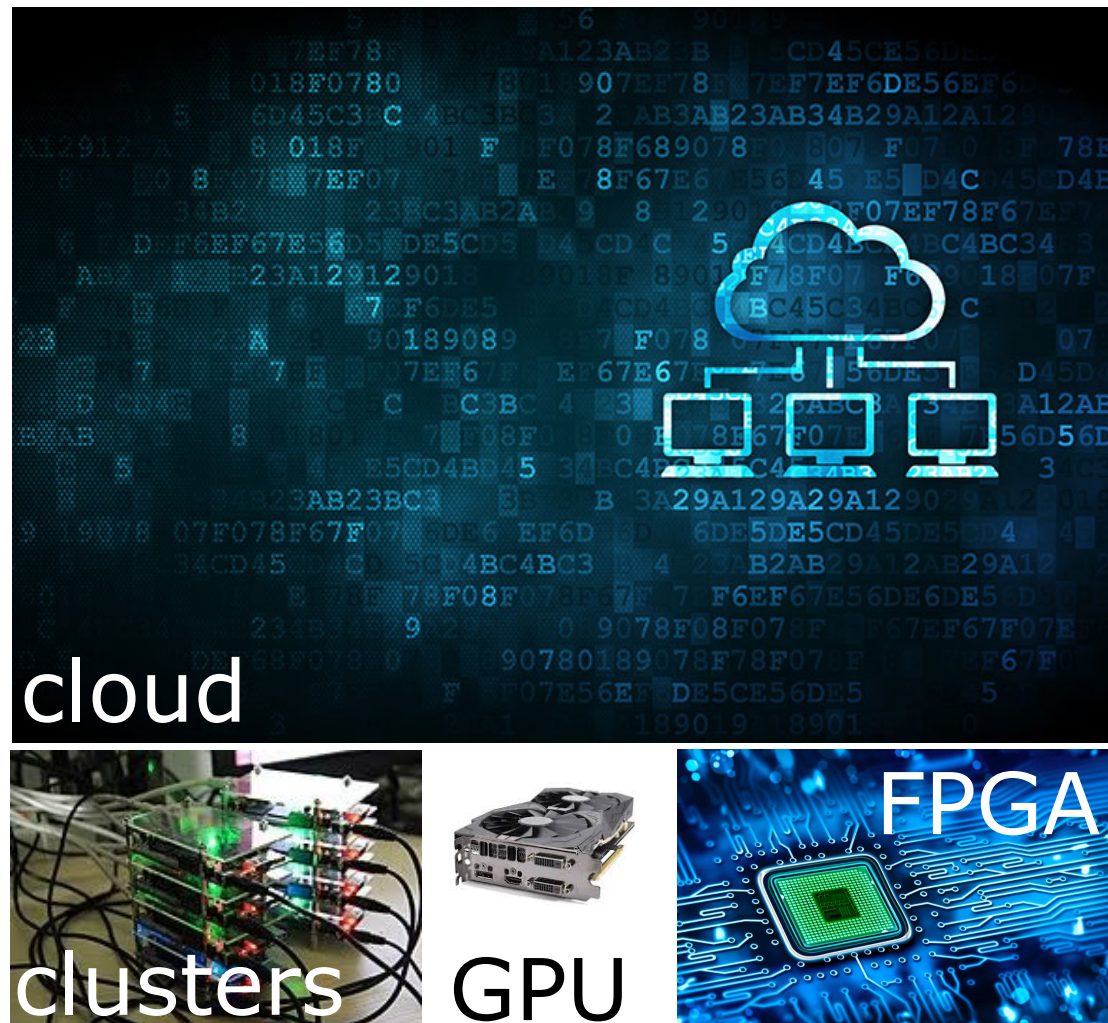
Newton-Raphson
~17ος μΧ

ταχύτητα
αλγορίθμου

αξιοπιστία
(θα βρει τη
σωστή λύση;)



υπολογιστική ισχύς



γίνεται όλο και πιο φθηνή & προσβάσιμη, μας προσκαλεί να αλλάξουμε τον τρόπο υλοποίησης και επιλογής κατάλληλων αλγορίθμων

επίλυση της $f(x) = 0$

Όταν η $f(x)$ είναι μη-γραμμική, τα πράγματα
δυσκολεύουν

διχοτόμηση (Bisection)

εσφαλμένη θέση (Regula Falsi)

διατέμνουσας (Secant)

σταθερού σημείου

Newton-Raphson

*Υποθέτουν την ύπαρξη "καλής" αρχικής εκτίμησης από τον
χρήστη την οποία θα (προσπαθήσουν) να βελτιώσουν
επαναληπτικά.*

επίλυση της $f(x) = 0$

Όταν η $f(x)$ είναι μη-γραμμική, τα πράγματα
δυσκολεύουν

διχοτόμηση (Bisection)

εσφαλμένη θέση (Regula Falsi)

διατέμνουσας (Secant)

σταθερού σημείου

Newton-Raphson

Παρενθετικές μέθοδοι [bracketing methods]

$a < \text{ρίζα} < b$, συρρίκνωση του διαστήματος $b - a$

επίλυση της $f(x) = 0$

Όταν η $f(x)$ είναι μη-γραμμική, τα πράγματα
δυσκολεύουν

διατέμνουσας (Secant)

σταθερού σημείου

Newton-Raphson

Fixed-point iterations [επαναλήψεις σταθερού σημείου]

$\mathbf{x}_{n+1} = \mathbf{g}(\mathbf{x}_n)$ [αναδρομική σχέση ενός σημείου]

$\mathbf{x}_{n+1} = \mathbf{g}(\mathbf{x}_n, \mathbf{x}_{n-1}, \mathbf{x}_{n-2} \dots)$ [αναδρομική σχέση πολλών
σημείων]

$$\lim_{n \rightarrow \infty} x_n = \rho$$



επίλυση της $f(x) = 0$

*Συνήθως, δεν επισημαίνεται όσο
πρέπει ότι η **γραφική μέθοδος** είναι
αναντικατάστατη, για μια καλή αρχή*

$$\tan x = \frac{x}{1 - x^2}$$

ποια είναι μια καλή αρχική εκτίμηση;

πρώτο βήμα

κατασκευάζουμε την γραφική παραστασή της $f(x)$

μελετάμε μονοτονία

εντοπίζουμε διαστήματα $[x_{\min}, x_{\max}]$ με ρίζες

— και το πλήθος αυτών —

ακρίβεια της γραφικής μεθόδου : $|x_{\max} - x_{\min}| / ((N-1)\sqrt{12})$
όπου N ο αριθμός των σημείων που χρησιμοποιήσαμε στο
διάγραμμα

παραδείγματα

bisection

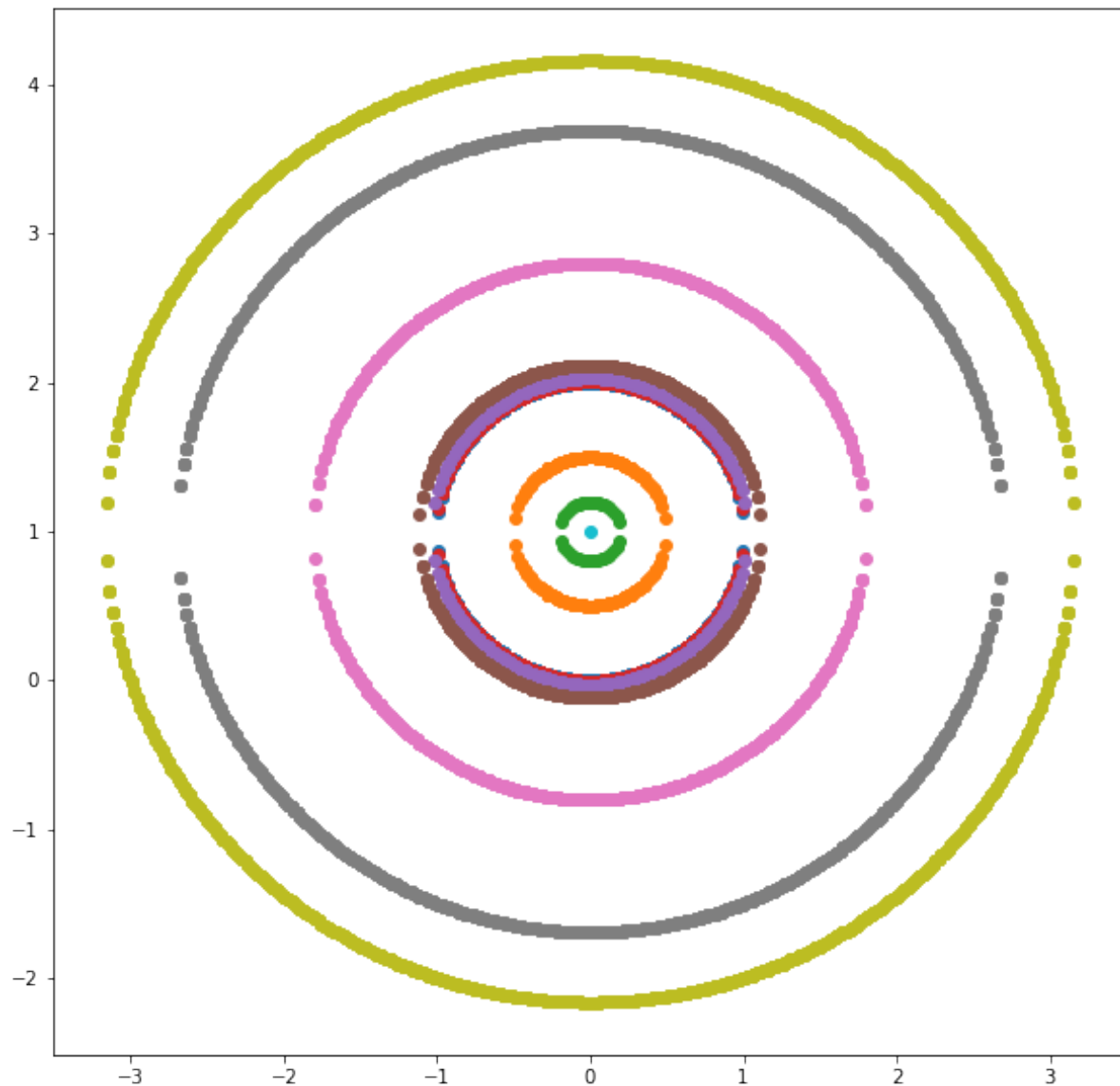
newton

fixedpoint

electrostatic_potential

**εφαρμογή φυσικής –
εύρεση ισοδυναμικών
γραμμών ηλεκτρικού πεδίου**

εφαρμογή φυσικής



$$V(r) = kq/r$$

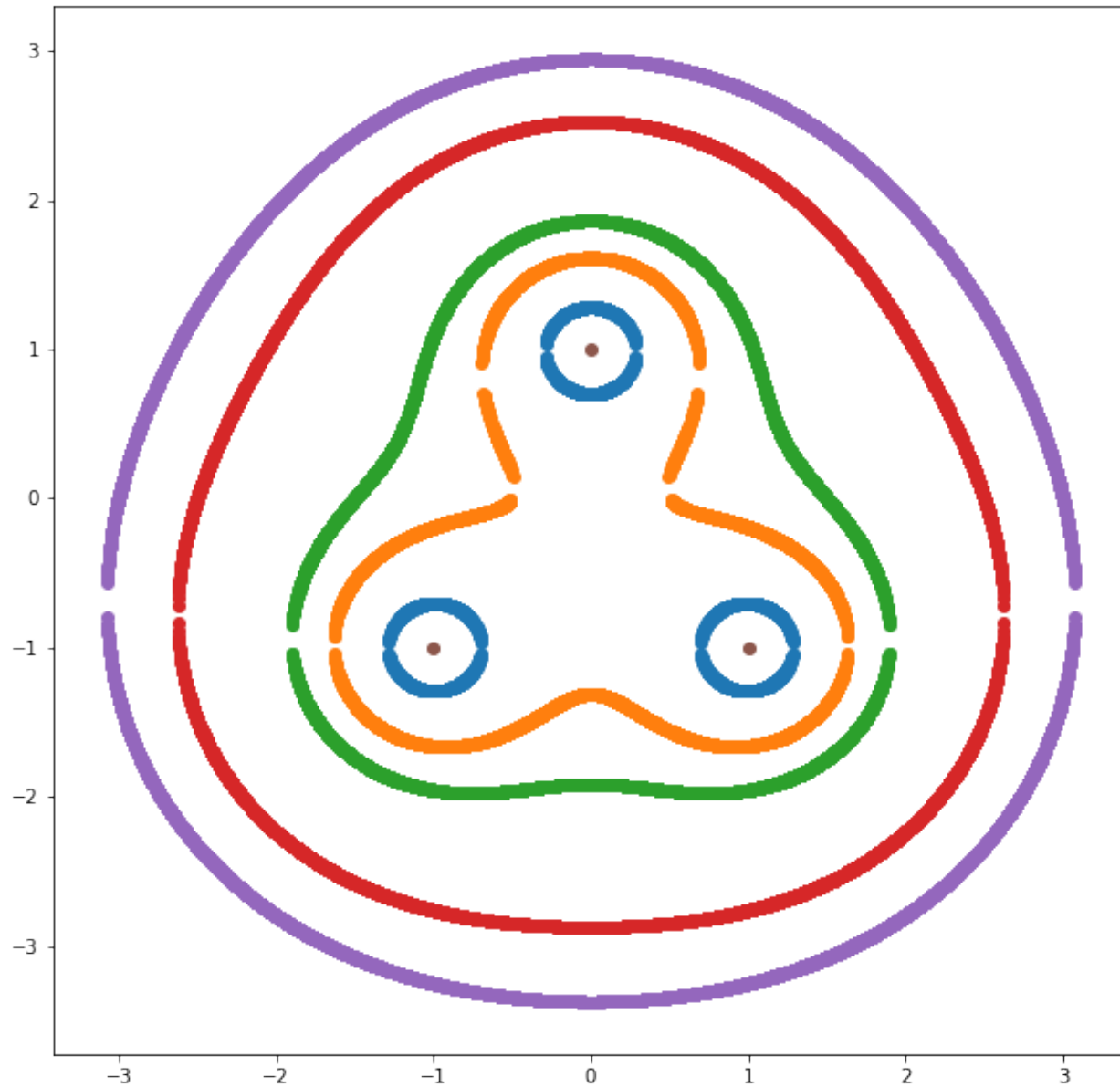
εύρεση ισοδυναμικών
γραμμών με την λύση
αλγεβρικής εξίσωσης, αντί
σχεδιασμού του πιο απλού

$$\mathbf{r}(\theta) = r\cos\theta \mathbf{i} + r\sin\theta \mathbf{j}$$

for $\mathbf{x}$ in range(-5,5):
 find $\mathbf{y}$, for which $\mathbf{V}(\mathbf{x}, \mathbf{y}) = \mathbf{Vref}$

brute force -
ωμή δύναμη

εφαρμογή φυσικής



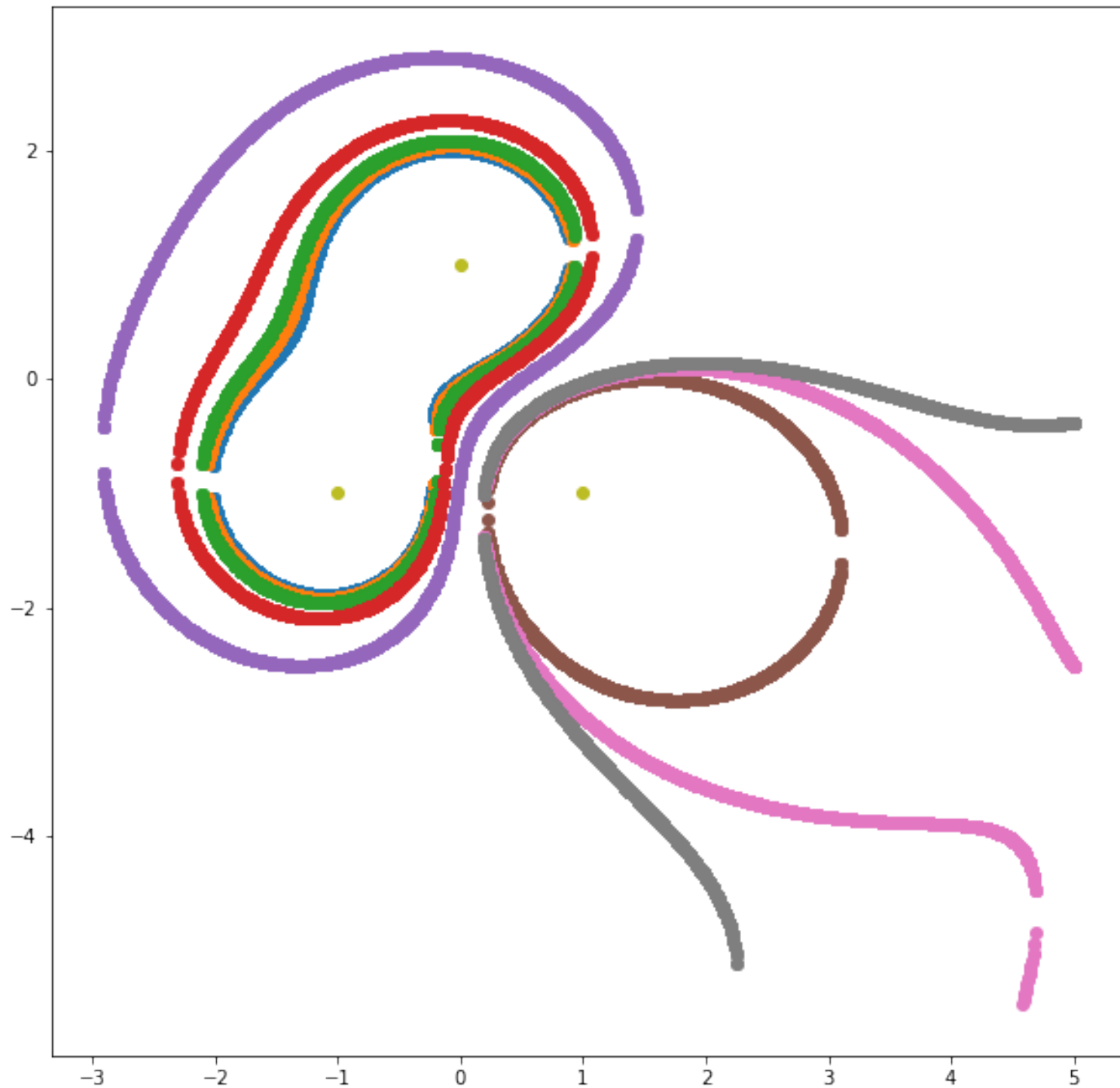
$$V(r) = \sum kq/r$$

το όφελος του να
συνδυάζουμε άπλες
οδηγίες (λύσε την $f(y) = 0$) με την ωμή δύναμη
του H/Y , είναι η
δυνατότητα γενίκευσης
για προβλήματα πιο
δύσκολα, πχ 3 φορτία

for $\mathbf{x}$ in range(-5,5):
 find $\mathbf{y}$, for which $\mathbf{V}(\mathbf{x}, \mathbf{y}) = \mathbf{Vref}$

ο αλγόριθμος
επίλυσης δεν
άλλαξε

εφαρμογή φυσικής



το σημαντικό είναι, ότι
ο υπολογιστής "έμαθε"
να ζωγραφίζει το
σωστό, χωρίς να έχει
προγραμματιστεί ρητά
να το κάνει για τις
συγκεκριμένες
περιπτώσεις μέσω
κάποιας $r(\theta)$ οδηγίας

**άρα μπορούμε να
μάθουμε πίσω από
αυτόν, τι γίνεται σε
προβλήματα που δε
μπορούμε να λύσουμε
με χαρτί και μολύβι**

εφαρμογή φυσικής

~80 γραμμές (python)

```
# written by Konstantinos Theofilatos 05.10.19
import numpy as np
import matplotlib.pyplot as plt

def bisection(f, a, b, tol = 1.e-7, Nmax = 10000, n = 0):
    c = -9.999e+300
    while abs(b-a) > tol and n < Nmax:
        if np.sign(f(a))*np.sign(f(b)) > 0:
            # print('f(%2.4f)*f(%2.4f) = %2.4f thus it's problematic'%(a, b, f(a)*f(b)))
            break
        c = a + (b-a)*0.5
        if f(c)==0: break
        if np.sign(f(c))*np.sign(f(b)) < 0:
            a = c
        if np.sign(f(a))*np.sign(f(c)) < 0:
            b = c
        n = n + 1
    #print('root at best effort is %2.6f [n = %d]%(c, n))
    return c

def distance(r1, r2):
    res = (((r1-r2).T@(r1-r2))**0.5)[0,0]
    if res < 1.e-9:
        res = 1.e-9
    print('divergence occurred for \n r1 = \n', r1, '\n r2 = \n', r2)
    return res

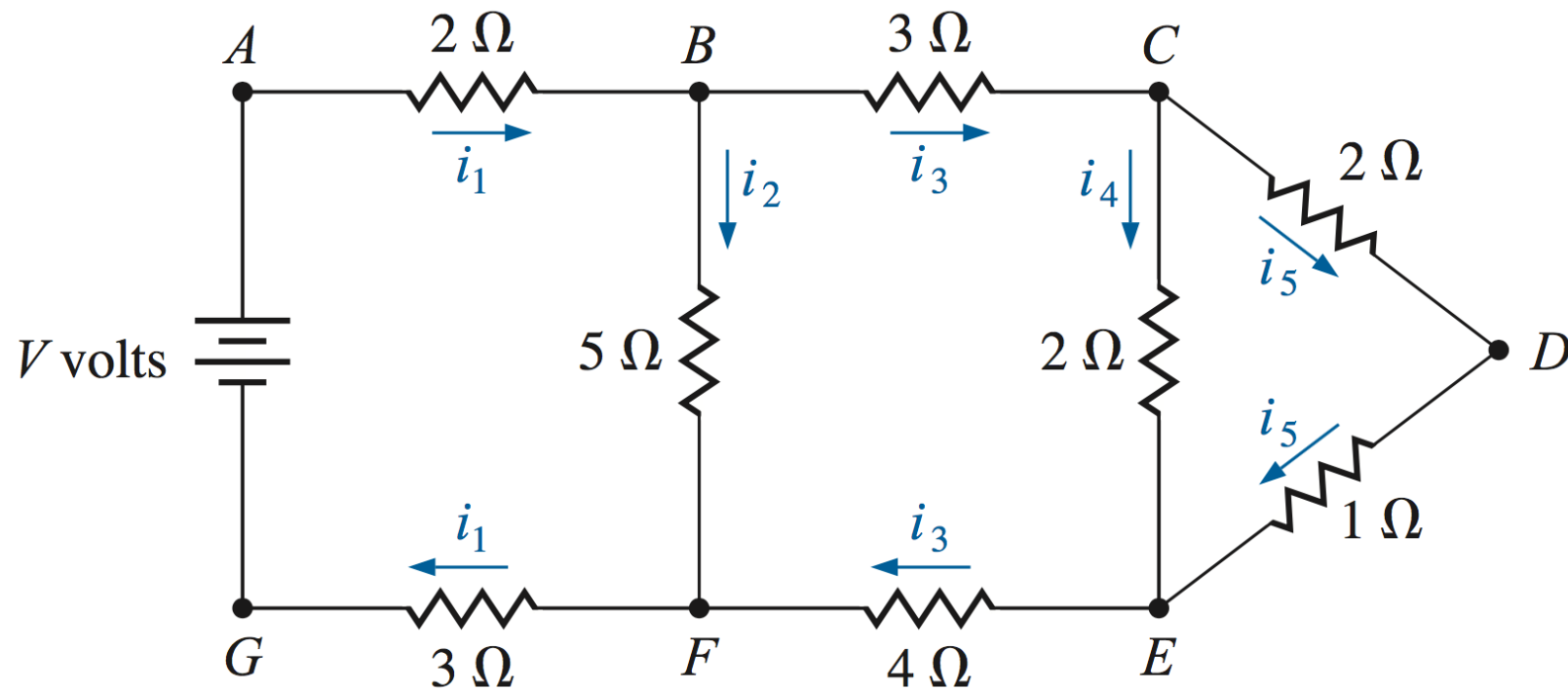
class particle:
    '''particles live in 2D, having x, y coordinates and charge q'''
    def __init__(self, x, y, q):
        self.x = x
        self.y = y
        self.q = q
        self.r = np.array([[x], [y]])

def V(x, y):
```

https://github.com/theofil/CompPhysics/blob/master/examples/electrostatic_potential.ipynb

γραμμικά συστήματα εξισώσεων

ηλεκτρικά κυκλώματα



$$5i_1 + 5i_2 = V,$$

$$i_3 - i_4 - i_5 = 0,$$

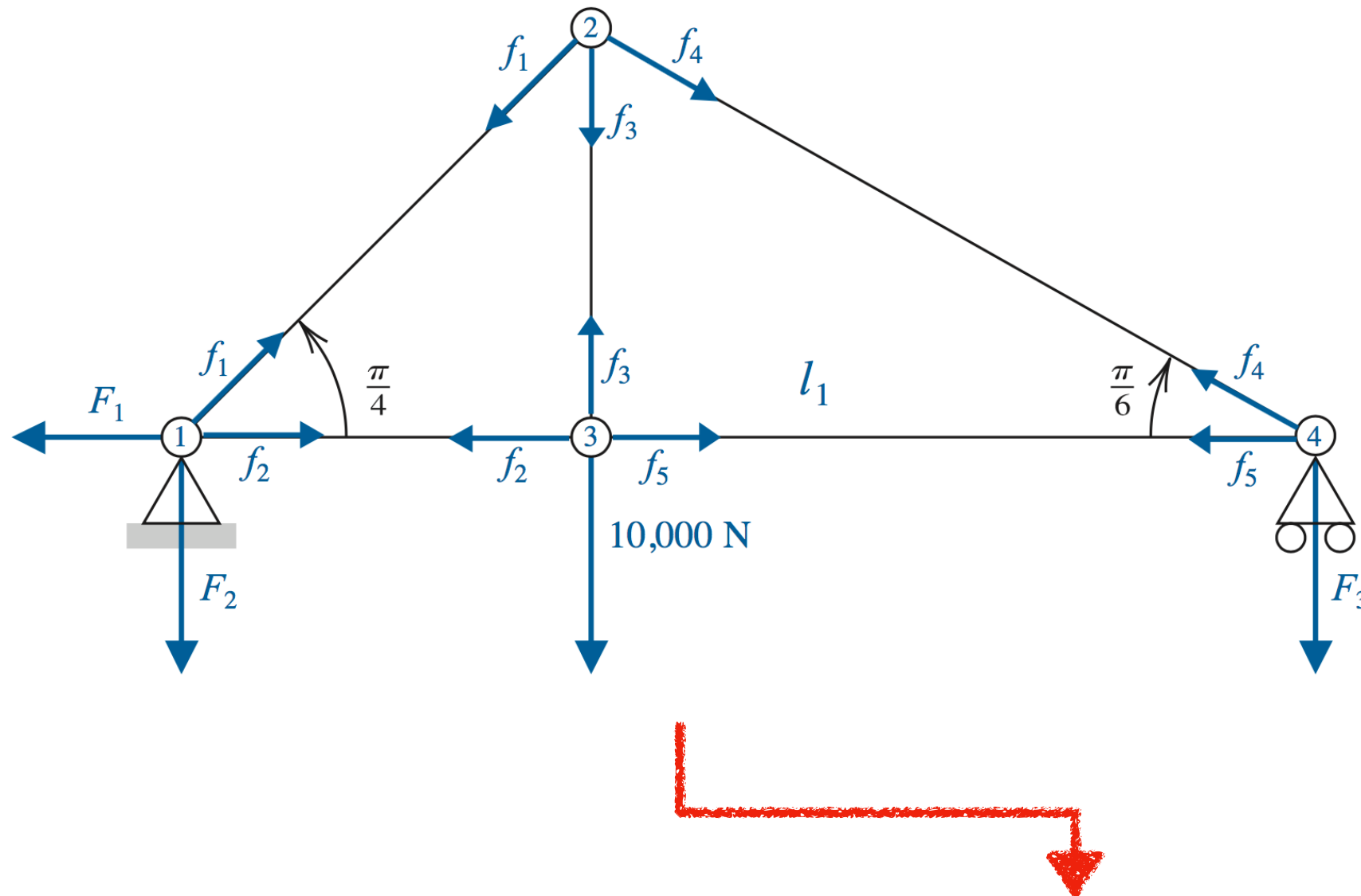
$$2i_4 - 3i_5 = 0,$$

$$i_1 - i_2 - i_3 = 0,$$

$$5i_2 - 7i_3 - 2i_4 = 0.$$

R. L. Burden

J. D. Faires



Joint	Horizontal Component	Vertical Component
①	$-F_1 + \frac{\sqrt{2}}{2} f_1 + f_2 = 0$	$\frac{\sqrt{2}}{2} f_1 - F_2 = 0$
②	$-\frac{\sqrt{2}}{2} f_1 + \frac{\sqrt{3}}{2} f_4 = 0$	$-\frac{\sqrt{2}}{2} f_1 - f_3 - \frac{1}{2} f_4 = 0$
③	$-f_2 + f_5 = 0$	$f_3 - 10,000 = 0$
④	$-\frac{\sqrt{3}}{2} f_4 - f_5 = 0$	$\frac{1}{2} f_4 - F_3 = 0$

R. L. Burden

J. D. Faires

σύνθετα προβλήματα

πχ

μη-γραμμικά συστήματα εξισώσεων

διαφορικές εξισώσεις

συχνά ανάγονται, έπειτα από προσπάθεια + καλές
προσεγγίσεις, σε (απλά) γραμμικά συστήματα εξισώσεων

Βασικές μέθοδοι επίλυσης

ακριβείς — **[μικρά + πυκνά συστήματα $\sim O(2n^3/3)$]**

απαλοιφή Gauss (Gauss elimination) + [Gauss-Jordan]

διάσπαση LU

επαναληπτικές — **[μεγάλα + αραιά σύστημα]**

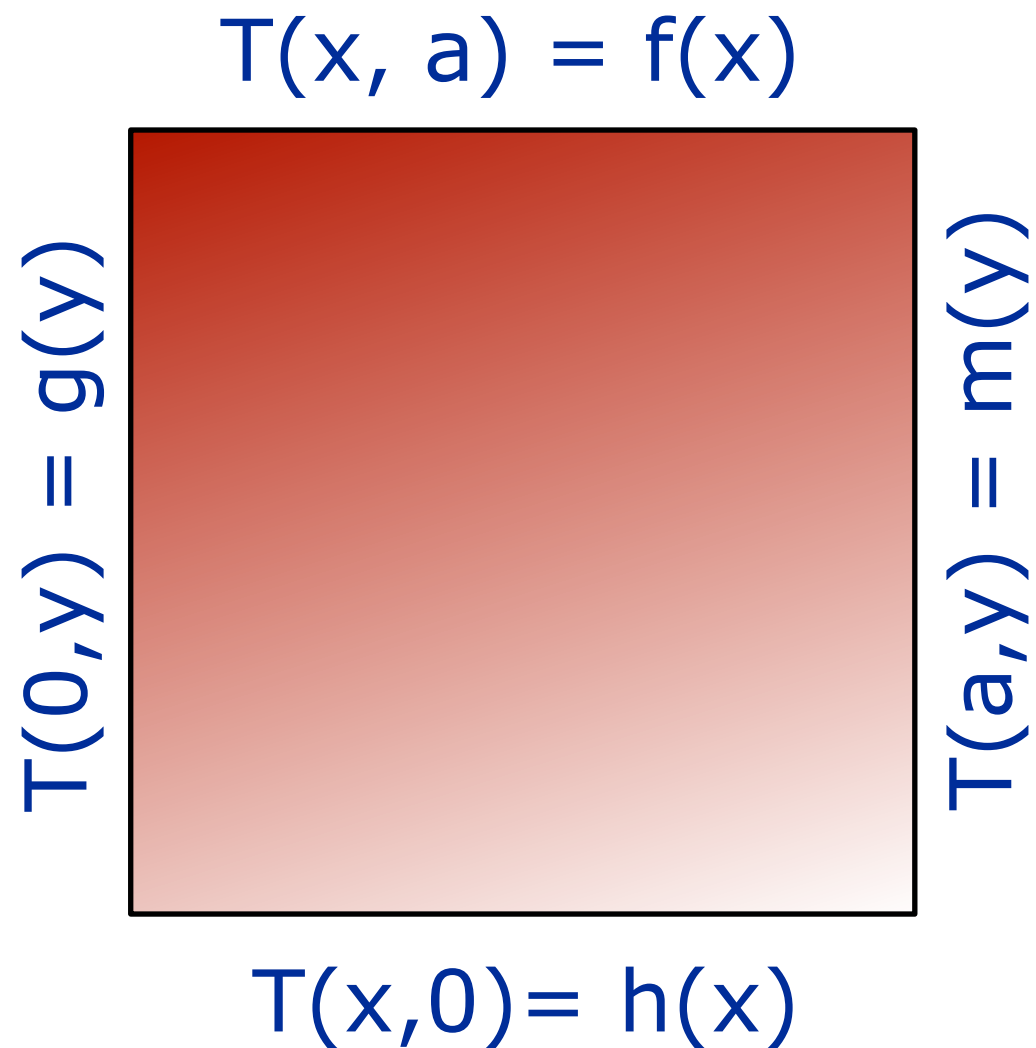
προσεγγιστική λύση (σύγκλιση υπό συνθήκες)

Jacobi

Gauss-Seidel

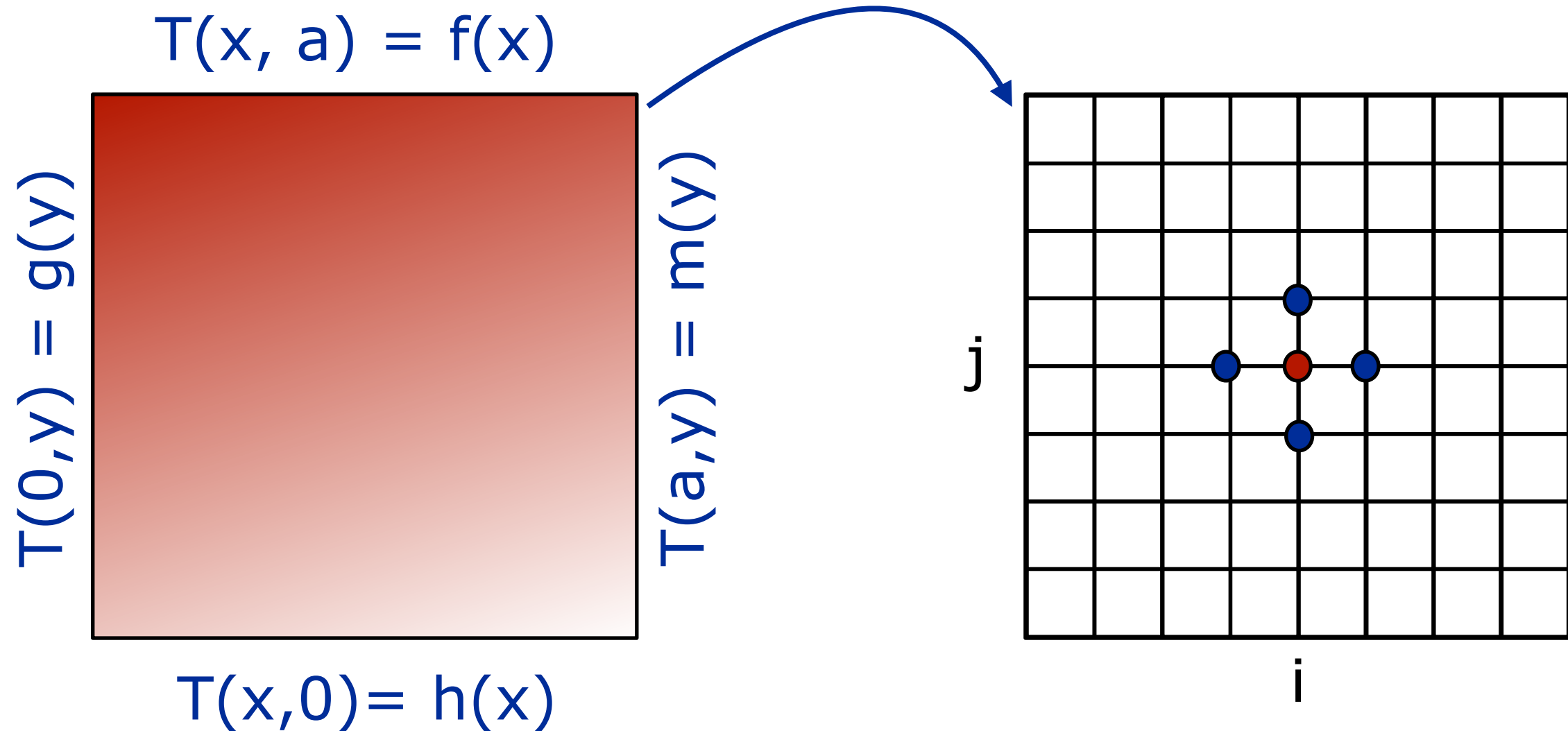
σημειώσεις στην η-τάξη

εξίσωση Poisson



$$\frac{\partial^2 T(x, y)}{\partial x^2} + \frac{\partial^2 T(x, y)}{\partial y^2} = g(x, y)$$

Εξίσωση Poisson



$$\frac{\partial^2 T(x, y)}{\partial x^2} + \frac{\partial^2 T(x, y)}{\partial y^2} = g(x, y)$$

$$T_{i+1,j} + T_{i-1,j} + T_{i,j+1} + T_{i,j-1} - 4T_{i,j} = g_{i,j}$$

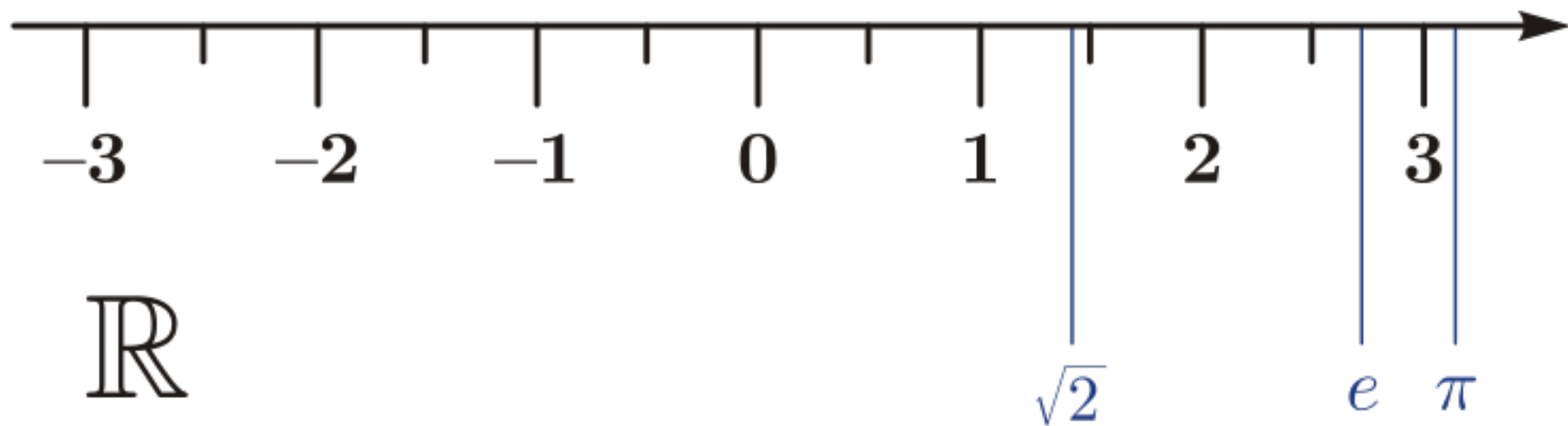
feedback form

Η φόρμα αφορά την χρήση Η/Υ και την εξοικείωση σας με τον προγραμματισμό στα επίπεδα που χρειάζονται για το μάθημα

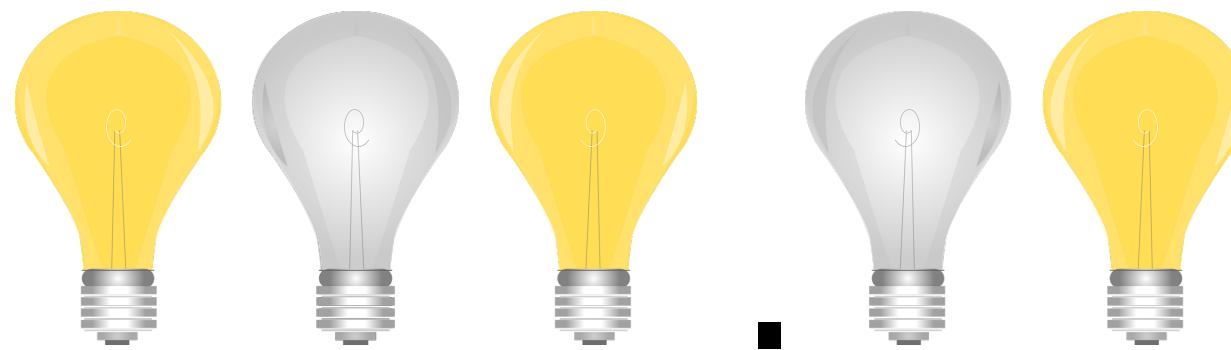
<https://forms.gle/ZAHshuyiSL4p3nJd6>

παρακαλώ στείλτε μου ό,τι σχόλια έχετε
επώνυμα ή ανώνυμα

πραγματικοί αριθμοί

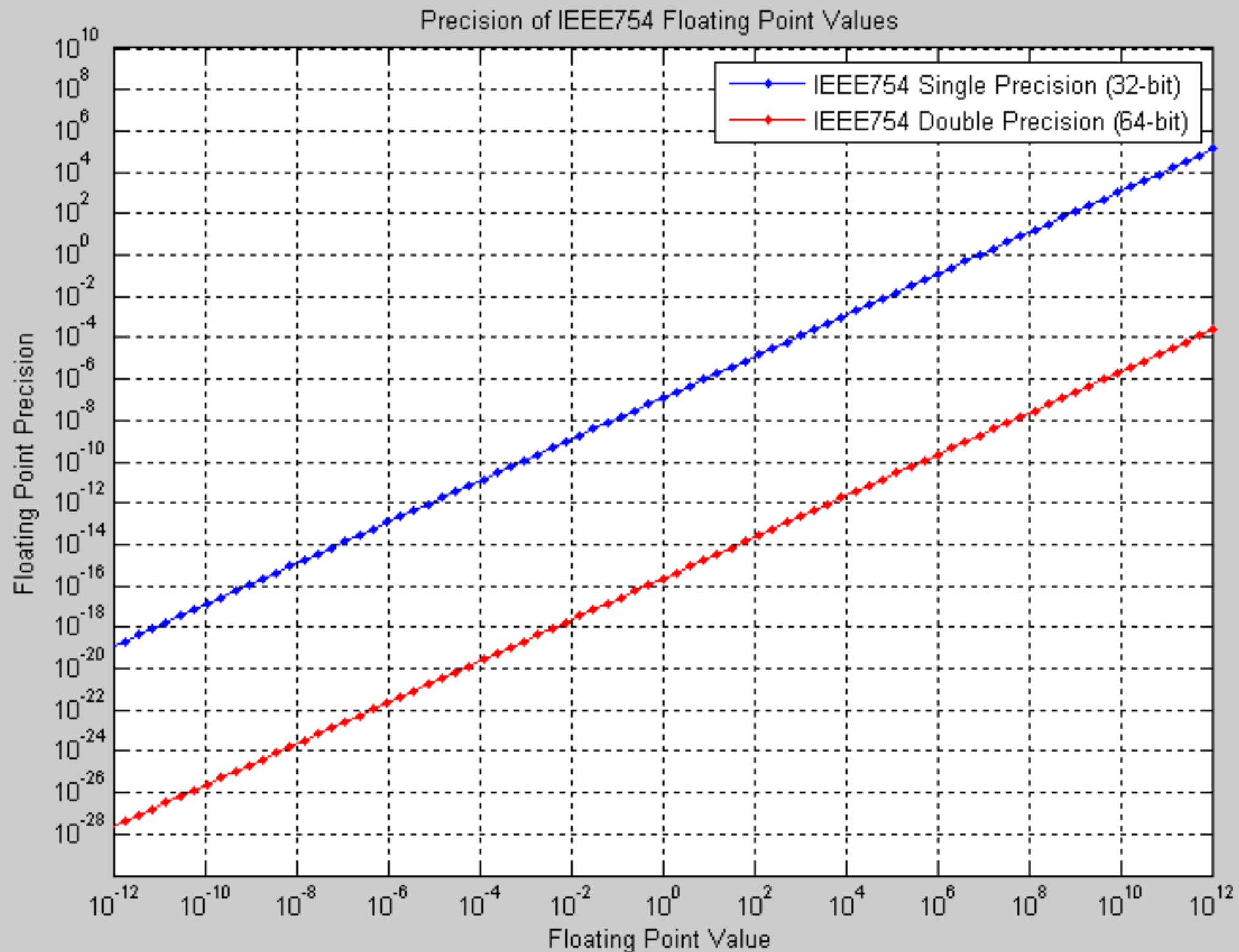


αριθμοί μηχανής



■ ■ ■ . ■ ■

$$= 101.01$$



rounding, ceiling, floor

Example of rounding to integers using the IEEE 754 rules

Mode / Example Value	+11.5	+12.5	-11.5	-12.5
to nearest, ties to even	+12.0	+12.0	-12.0	-12.0
to nearest, ties away from zero	+12.0	+13.0	-12.0	-13.0
toward 0	+11.0	+12.0	-11.0	-12.0
toward $+\infty$	+12.0	+13.0	-11.0	-12.0
toward $-\infty$	+11.0	+12.0	-12.0	-13.0