Η μέθοδος IDO
και
η Υλοποίησή της σε CUDA C

Αντωνιάδης Παναγιώτης - Δημήτριος

επιβλέπων καθηγητής: κ. Στεργιούλας Νικόλαος

Οι εξισώσεις του Euler

$$\varrho_t+(\varrho u)_x+(\varrho v)_y+(\varrho \omega)_z=0$$ διατήρηση μάζας

$$\left.\begin{aligned}(\varrho u)_t+(\varrho u^2+p)_x+(\varrho u v)_y+(\varrho u \omega)_z=0\\(\varrho v)_t+(\varrho u v)_x+(\varrho v^2+p)_y+(\varrho v \omega)_z=0\\(\varrho \omega)_t+(\varrho u \omega)_x+(\varrho v \omega)_y+(\varrho \omega^2+p)_z=0\end{aligned}\right\}$$ διατήρηση ορμής

$$E_t+[u(E+p)]_x+[v(E+p)]_y+[\omega(E+p)]_z=0$$ διατήρηση ενέργειας

+ εξίσωση κατάστασης (π.χ.) p = p (ρ, e)

1D

$$\boldsymbol{U}_t+\boldsymbol{F}(\boldsymbol{U})_x=0$$

$$\boldsymbol{U}=\begin{bmatrix}\varrho\\ \varrho u\\ E\end{bmatrix},\ \boldsymbol{F}=\begin{bmatrix}\varrho u\\ \varrho u^2+p\\ u(E+p)\end{bmatrix}$$

διατηρητική

$$\boldsymbol{W}_t+\boldsymbol{A}(\boldsymbol{W})\boldsymbol{W}_x=0$$

$$\boldsymbol{W}=(\varrho, u, p)^T$$

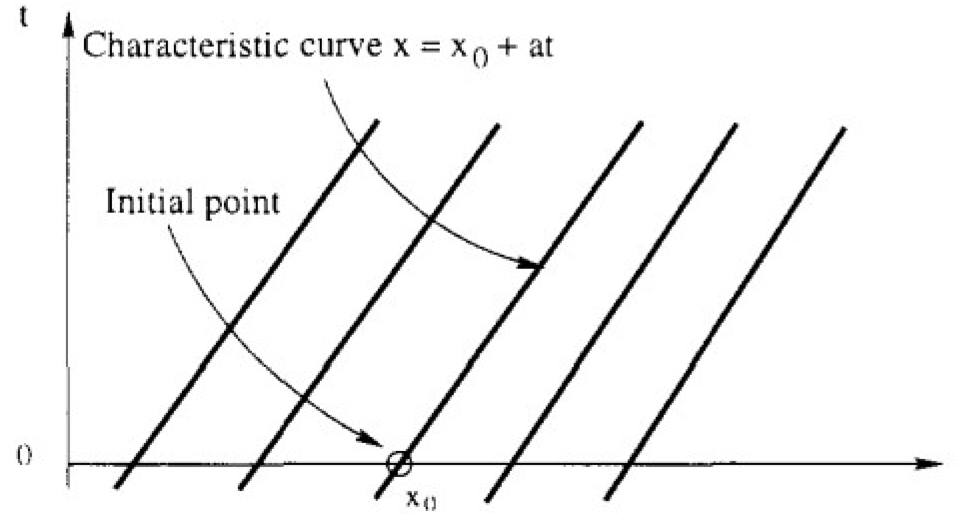
$$\boldsymbol{A}=\begin{bmatrix}u & \varrho & 0\\ 0 & u & 1/\varrho\\ 0 & \varrho a^2 & u\end{bmatrix}$$

μη - διατηρητική

1ης τάξης (γραμμική) υπερβολική εξίσωση

$$u_t + a\,u_x = 0\,,\ \ a = \sigma\tau\alpha\theta.$$

$$u(x,0) = u_0(x) = \begin{cases} u_L\,,\ \ x<0 \\ u_R\,,\ \ x>0 \end{cases}$$



$$u = u(x(t),t)$$

$$\frac{du}{dt} = \frac{\partial u}{\partial t} + \frac{\partial u}{\partial x}\frac{dx}{dt} \xrightarrow{\frac{dx}{dt}=a} u = \sigma\tau\alpha\theta.$$

$$u(x,t) = u_0(x_0) = u_o(x-at) = \begin{cases} u_L\,,\ x-at<0 \\ u_R\,,\ x-at>0 \end{cases}$$

Σύστημα γραμμικών υπερβολικών εξισώσεων

$$\boldsymbol{U}_t + \boldsymbol{A}\boldsymbol{U}_x = 0 \xrightarrow[\boldsymbol{W}=\boldsymbol{K}^{-1}\boldsymbol{U}]{} \boldsymbol{W}_t + \boldsymbol{\Lambda}\boldsymbol{W}_x = 0 \longrightarrow w_i(x,t) = w_i^{(0)}(x-\lambda_i t)$$

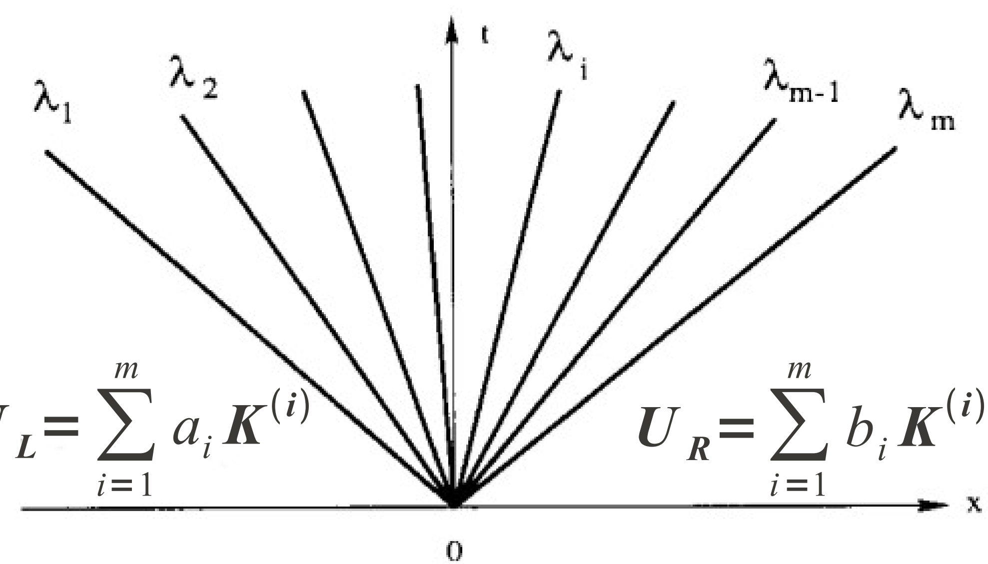
$$= \begin{cases} a_i, & x-\lambda_i t < 0 \\ b_i, & x-\lambda_i t > 0 \end{cases}$$

και αντίστροφα

$$w_i(x,t) \xrightarrow[\boldsymbol{U}=\boldsymbol{K}\boldsymbol{W}]{} \boldsymbol{U}(x,t) = \sum_{i=1}^{m} w_i^{(0)}(x-\lambda_i t)\,\boldsymbol{K}^{(i)}$$

$$\boldsymbol{U}(x,t) = \sum_{i=1}^{I} b_i \boldsymbol{K}^{(i)} + \sum_{i=I+1}^{m} a_i \boldsymbol{K}^{(i)}$$

$$\lambda_I < \frac{x}{t} < \lambda_{I+1}$$

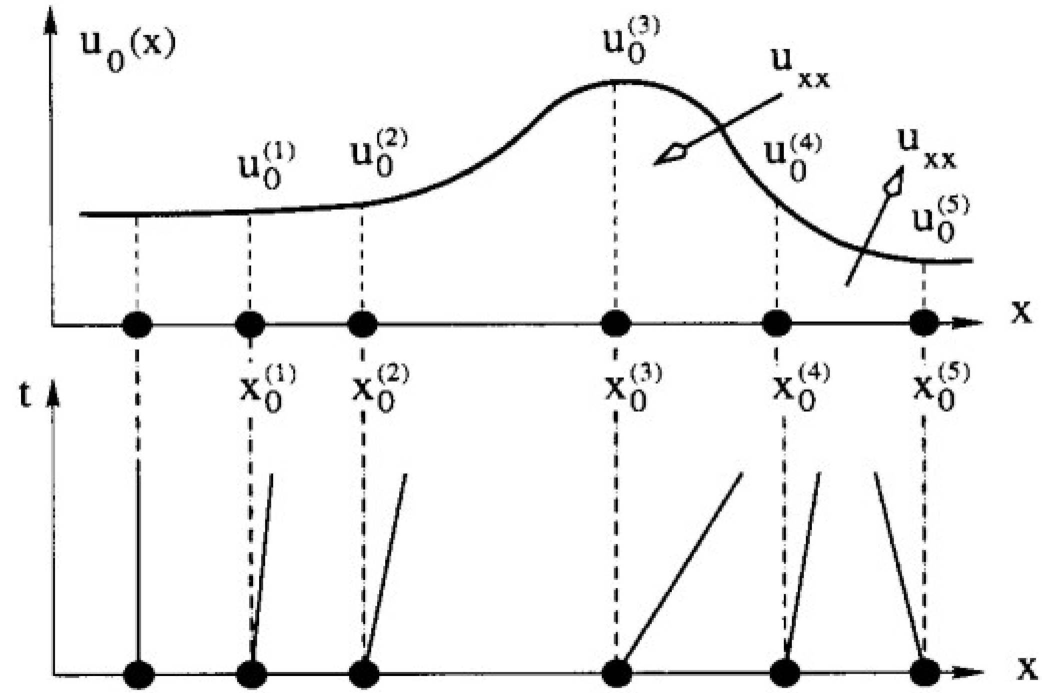


$$\boldsymbol{U}_L = \sum_{i=1}^{m} a_i \boldsymbol{K}^{(i)} \qquad \boldsymbol{U}_R = \sum_{i=1}^{m} b_i \boldsymbol{K}^{(i)}$$

Μη-γραμμικές εξισώσεις

ή $$u_t + f(u)_x = 0$$
$$u_t + \lambda(u) u_x = 0$$

- $\lambda'(u) > 0$ κυρτή f
- $\lambda'(u) < 0$ κοίλα f
- $\lambda'(u) = 0$ -



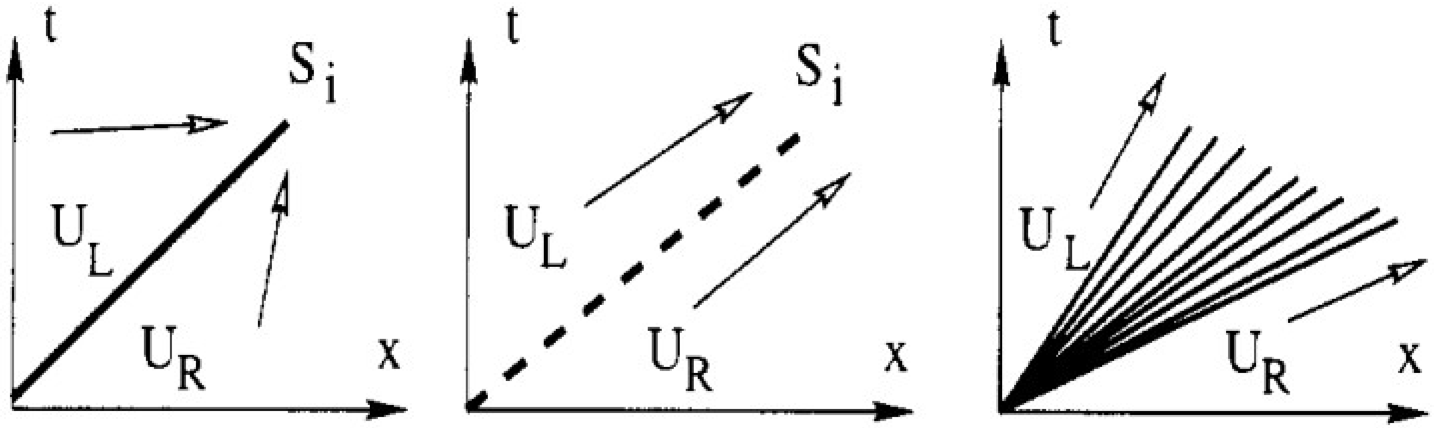
$I_E = [x_0^{(1)}, x_0^{(3)}]$ $\quad$ $I_C = [x_0^{(3)}, x_0^{(5)}]$

...η έννοια της ασυνέχειας

Στοιχειώδεις λύσεις μη-γραμμικών συστημάτων

<u>*χαρακτηριστικά πεδία*</u> $\nabla \lambda_i(\boldsymbol{U}) \cdot \boldsymbol{K}^{(i)}(\boldsymbol{U}) \begin{cases} = 0 \\ \neq 0 \end{cases}$

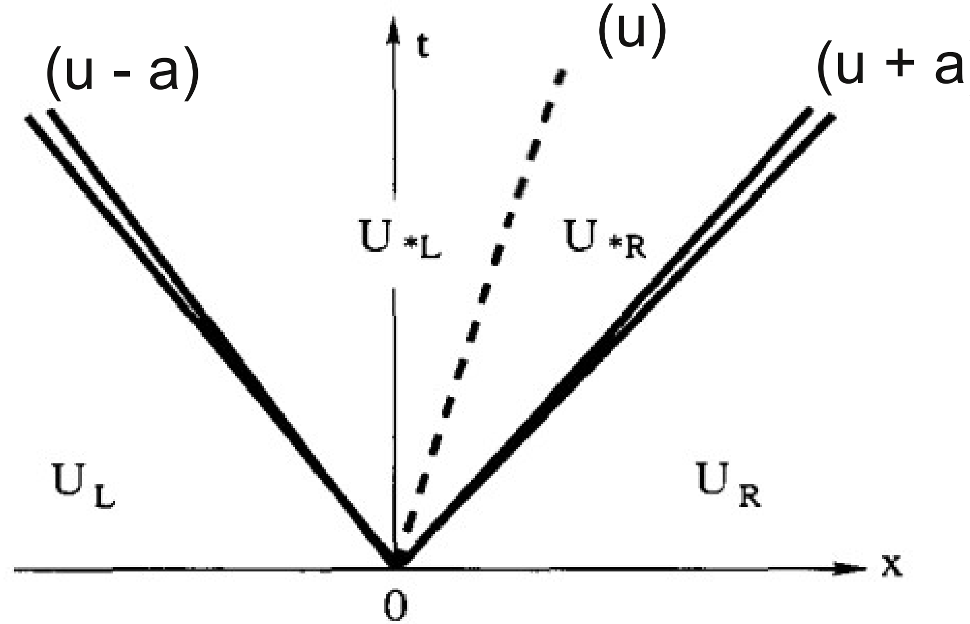
$$\boldsymbol{U}_t + \boldsymbol{F}(\boldsymbol{U})_x = 0$$
$$\boldsymbol{W}_t + \boldsymbol{A}(\boldsymbol{W})\boldsymbol{W}_x = 0$$



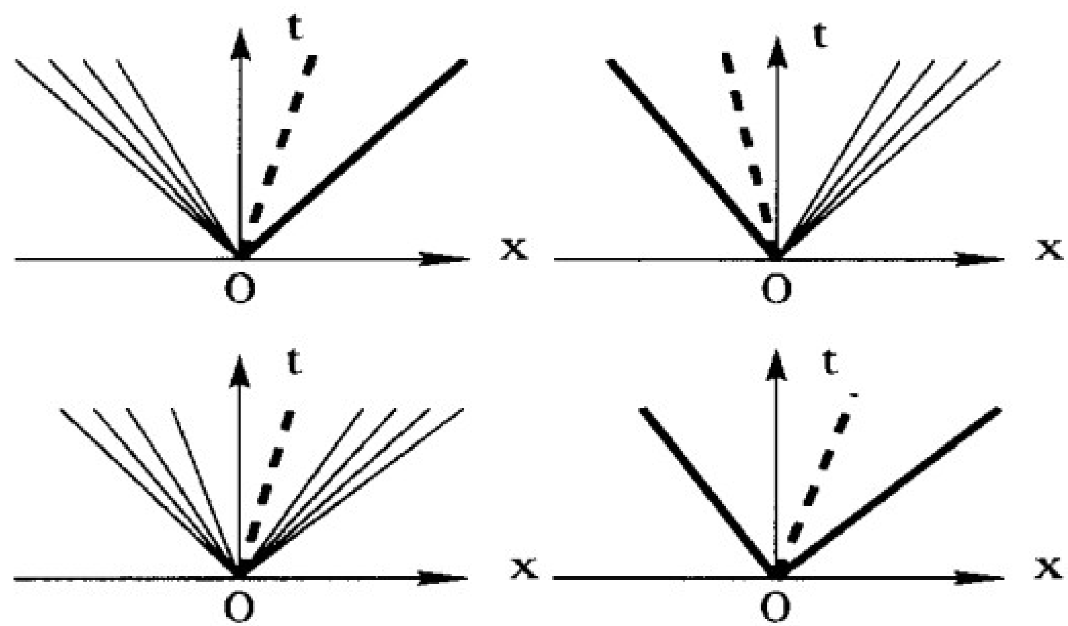
κρουστικό κύμα	ασυνέχεια επαφής	κύμα αραίωσης
$\Delta \boldsymbol{F} = S_i \Delta \boldsymbol{U}$	$\Delta \boldsymbol{F} = S_i \Delta \boldsymbol{U}$	—
—	$\dfrac{d w_1}{k_1^{(i)}} = \dfrac{d w_2}{k_2^{(i)}} = \ldots = \dfrac{d w_m}{k_m^{(i)}}$	$\dfrac{d w_1}{k_1^{(i)}} = \dfrac{d w_2}{k_2^{(i)}} = \ldots = \dfrac{d w_m}{k_m^{(i)}}$
$\lambda_i(\boldsymbol{U}_L) > S_i > \lambda_i(\boldsymbol{U}_R)$	$\lambda_i(\boldsymbol{U}_L) = S_i = \lambda_i(\boldsymbol{U}_R)$	$\lambda_i(\boldsymbol{U}_L) < \lambda_i(\boldsymbol{U}_R)$

Οι στοιχειώδεις λύσεις για τις εξισώσεις του Euler

$K^{(1)}, K^{(3)}$:   αμιγώς μη-γραμμικά
$K^{(2)}$:   γραμμικά εκφυλισμένο

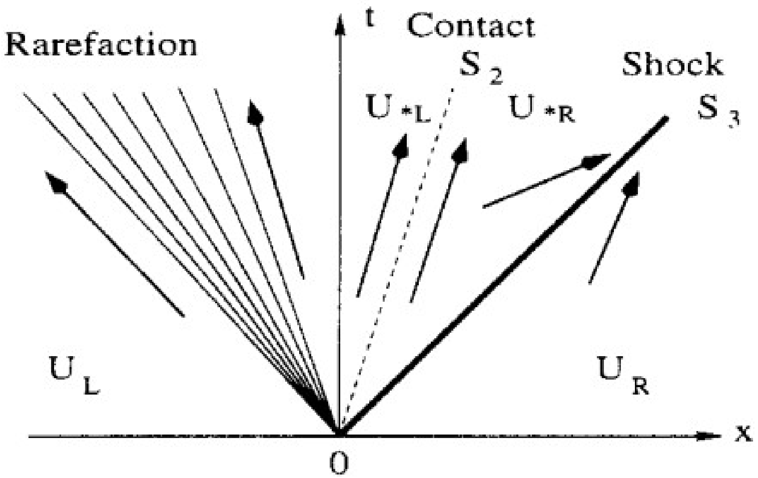


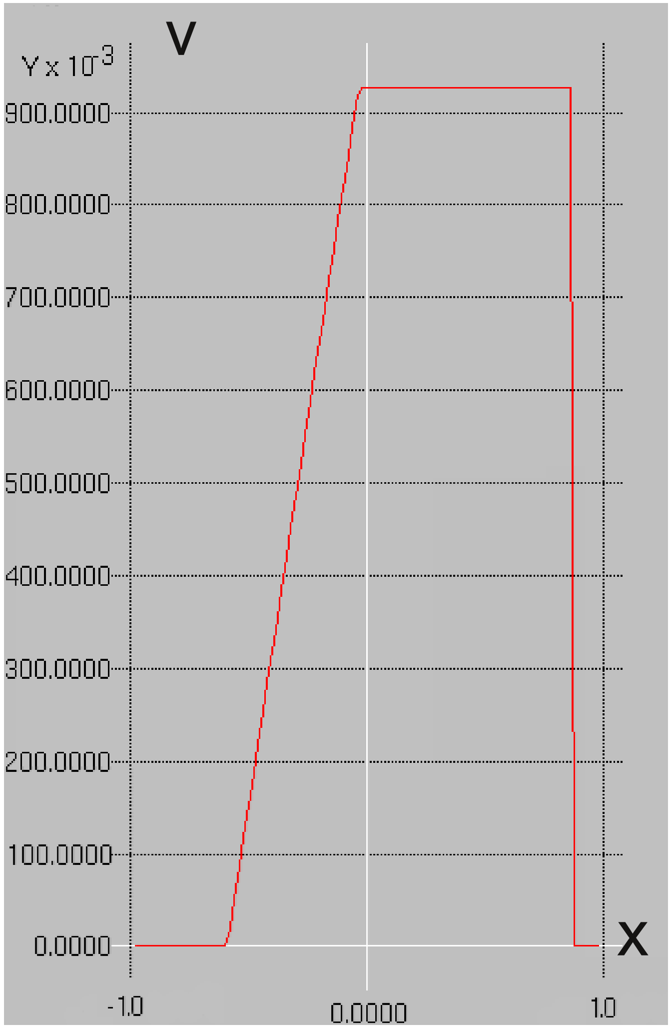
Περιοχές			
L	*L	*R	R
ρ_L	ρ_{*L}	ρ_{*R}	ρ_R
u_L	u_*	u_*	u_R
p_L	p_*	p_*	p_R

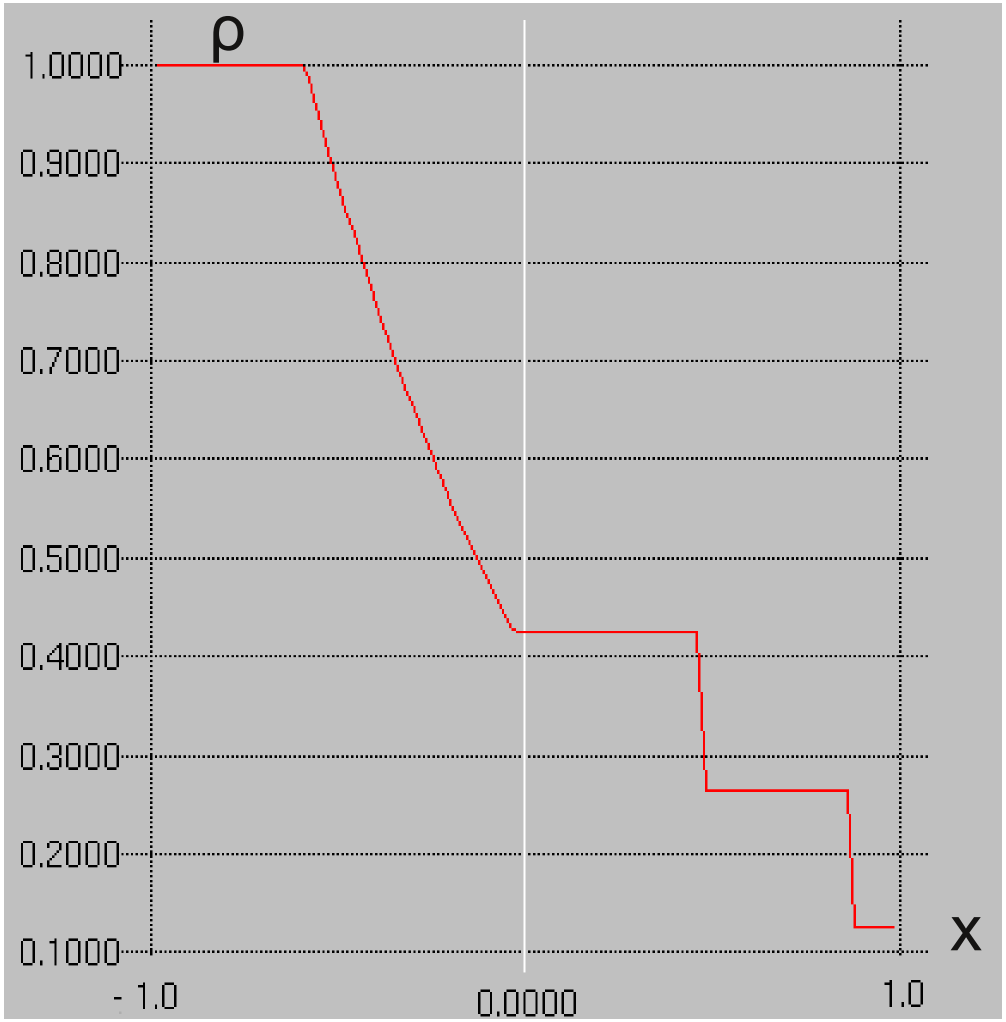


αρχικές συνθήκες

x < 0	x > 0
$\varrho = 1$	$\varrho = 0.125$
$p = 1$	$p = 0.1$
$u = 0$	$u = 0$







Η μέθοδος IDO

$$u_t = -a\,u_x$$

t = n

U

0	1	2	3	...	k

+

t = n

Ux

0	1	2	3	...	k

$$u^{n+1} = u^n + u_t^n \Delta t + u_{tt}^n \frac{\Delta t^2}{2}$$

$$u_x^{n+1} = u_x^n + u_{tx}^n \Delta t + u_{ttx}^n \frac{\Delta t^2}{2}$$

$$u_t = -a\,u_x$$

$$u_{ttx} = -a\,u_{txx}$$

$$u_{tt} = -a\,u_{tx}$$

$$u_{txx} = -a\,u_{xxx}$$

$$u_{tx} = -a\,u_{xx}$$

t = n + 1

U

0	1	2	3	...	k

+

t = n + 1

Ux

0	1	2	3	...	k

Μόνον χωρικές παράγωγοι $\longrightarrow$ Συμπτωτικό πολυώνυμο

Η μέθοδος IDO

Πολυώνυμο Hermite

upwind παρεμβολή

$a>0 \quad (i-1)\rightarrow i$

u	0	1	2	3	...	k

$a<0 \quad i\leftarrow(i+1)$

2 σημεία

κεντρική παρεμβολή

$i-1\leftarrow i\rightarrow i+1$

u	0	1	2	3	...	k

3 σημεία

$$F(x)=a_u x^3+b_u x^2+f_{x,i}x+f_i$$

$$F(x)=a_c x^5+b_c x^4+\ldots+f_{x,i}x+f_i$$

$$u^{n+1}=u^n+u_t^n\Delta t+u_{tt}^n\frac{\Delta t^2}{2}+\ldots+O(\Delta t^?)$$

Η μέθοδος IDO

Τεχνητό ιξώδες

$$q = -a\varrho C_s u_x \Delta x + b\varrho u_x^2 \Delta x^2$$

όρος α

+ για μικρά μέτωπα
- πρώτης τάξης

όρος b

+ για μεγάλα μέτωπα
- μείωση ακρίβειας

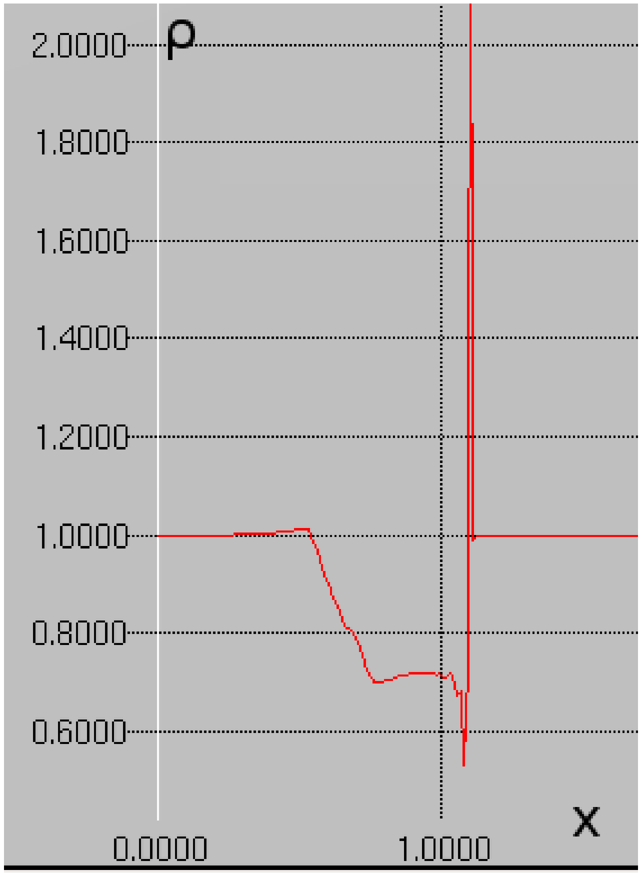
Euler 1D

$$\varrho_t = -(\varrho u)_x$$

$$u_t = -u u_x - \frac{p_x + q_x}{\varrho}$$

$$e_t = -u e_x - \frac{p+q}{\varrho} u_x$$

Λάθος επιλογή α, b



Η μέθοδος IDO

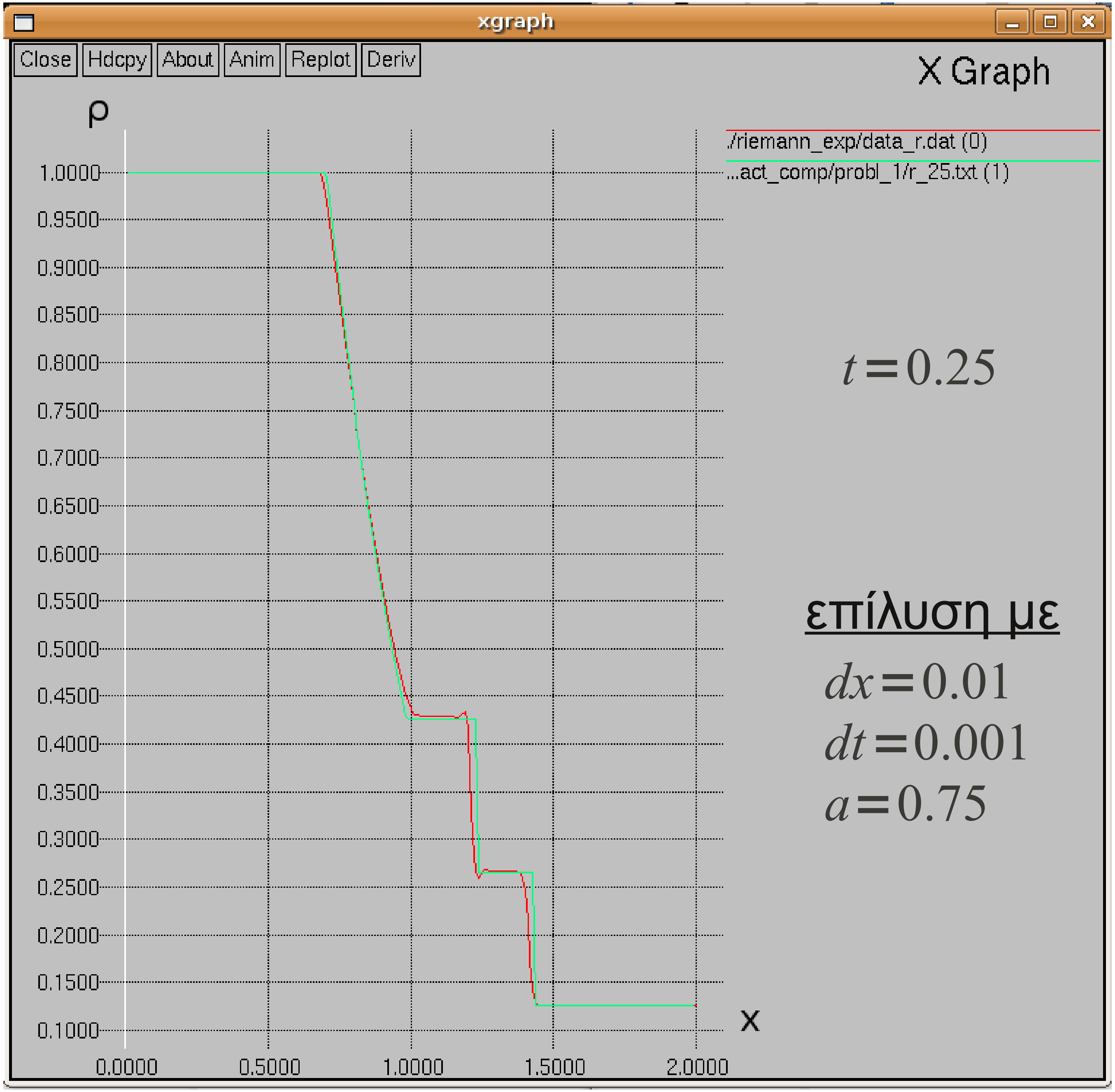
πρόβλημα του Sod
(ήπιο)

<u>αρχικές συνθήκες</u>

- U_L: $\varrho=1$
  $p=1$
  $u=0$

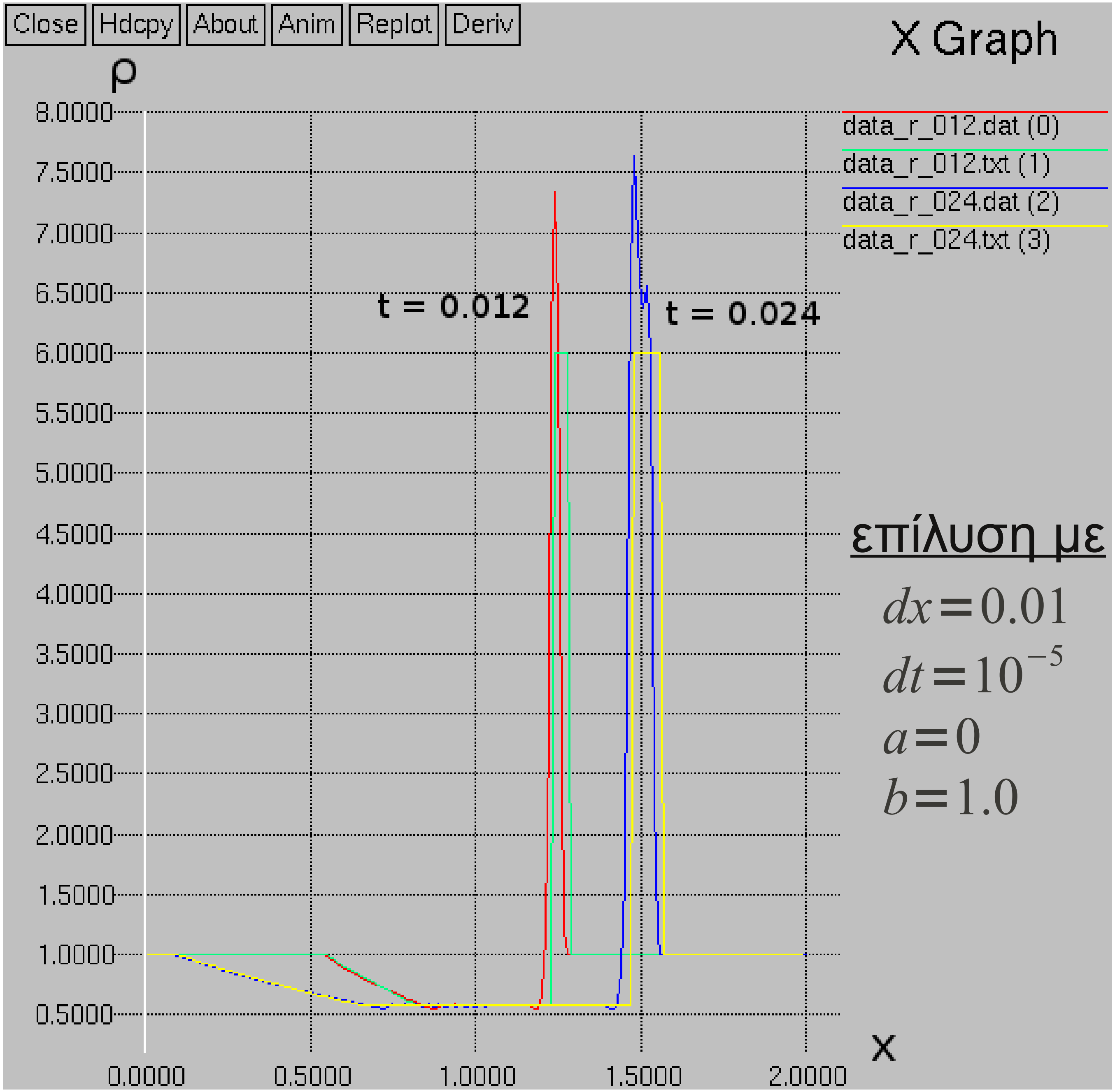
- U_R: $\varrho=0.125$
  $p=0.1$
  $u=0$

$\gamma=1.4$



Η μέθοδος IDO

two interacting blast waves (μισό)



<u>αρχικές συνθήκες</u>

- U_L: $\varrho = 1$
  $p = 1000$
  $u = 0$

- U_R: $\varrho = 1$
  $p = 0.01$
  $u = 0$

$\gamma = 1.4$

<u>επίλυση με</u>

$dx = 0.01$

$dt = 10^{-5}$

$a = 0$

$b = 1.0$

Η μέθοδος IDO

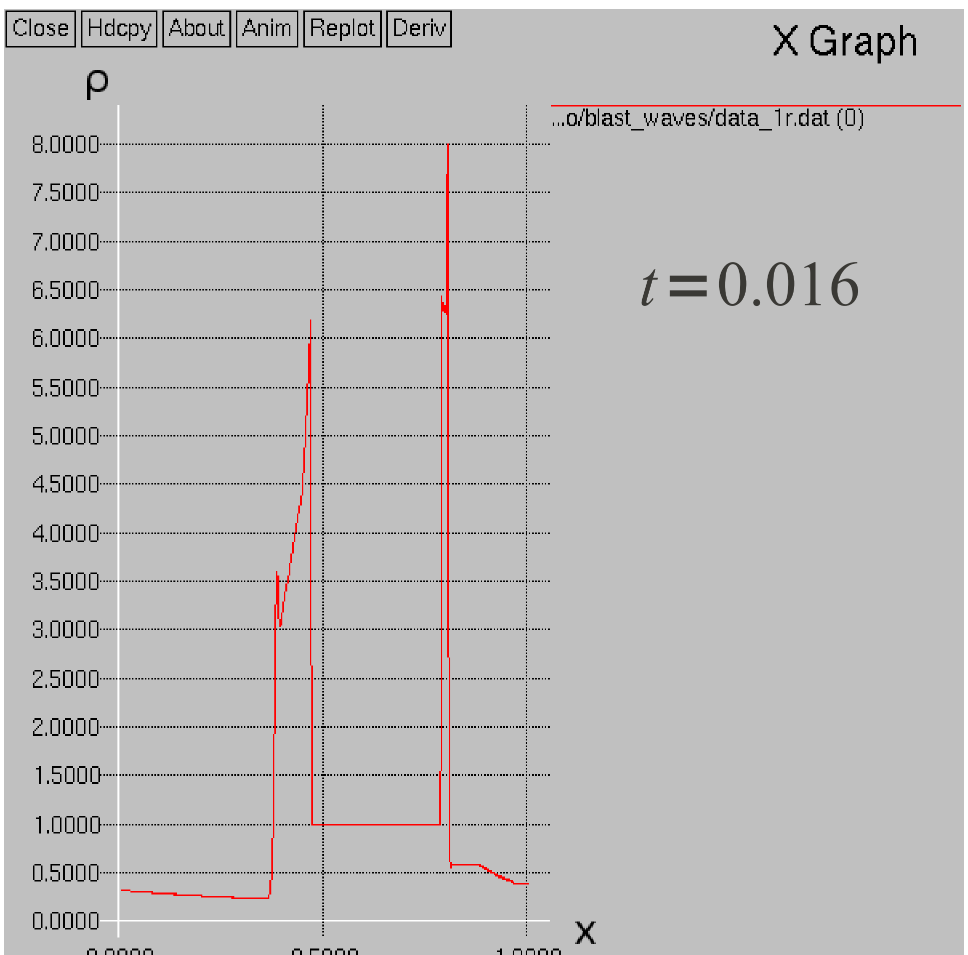
two interacting blast waves

<u>αρχικές συνθήκες:</u>

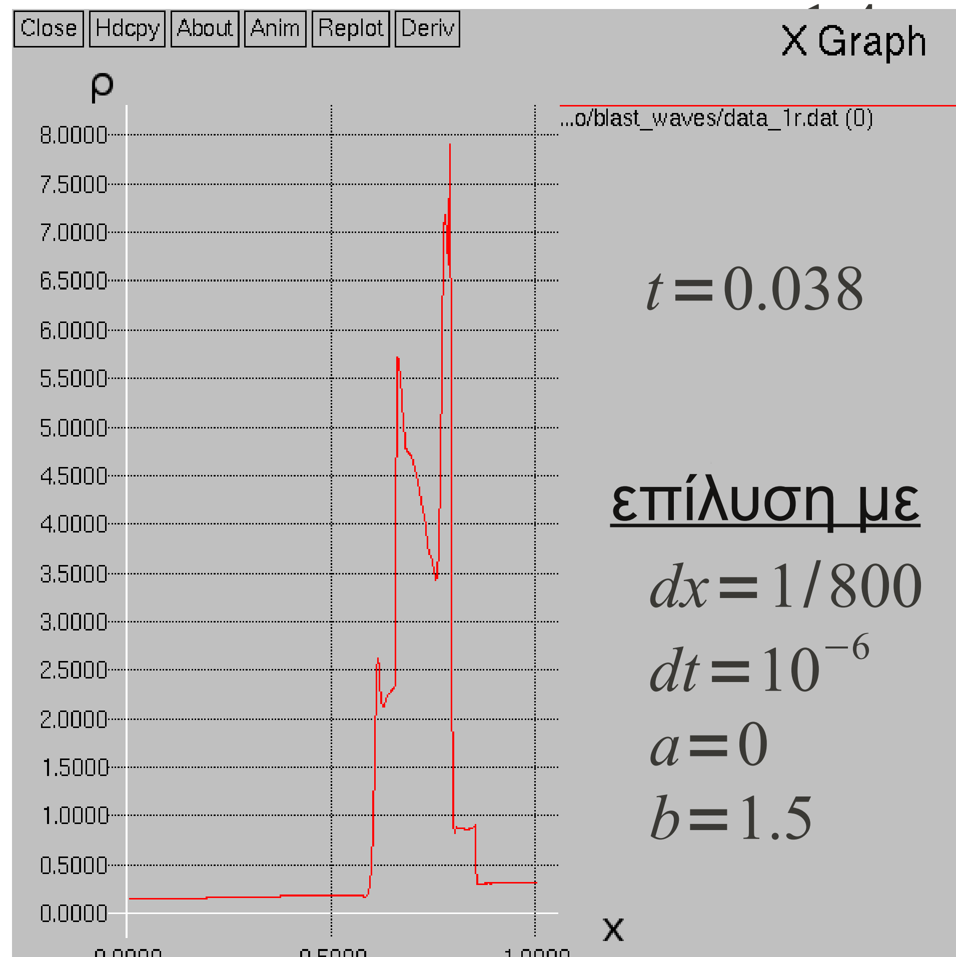
αριστερό δέκατο : $\varrho=1\,,\quad p=1000\,,\quad u=0$

ενδιάμεσος χώρος : $\varrho=1\,,\quad p=0.01\,,\quad u=0$

δεξί δέκατο: $\varrho=1\,,\quad p=100\,,\quad u=0$



$t=0.016$



$t=0.038$

<u>επίλυση με</u>

$dx=1/800$

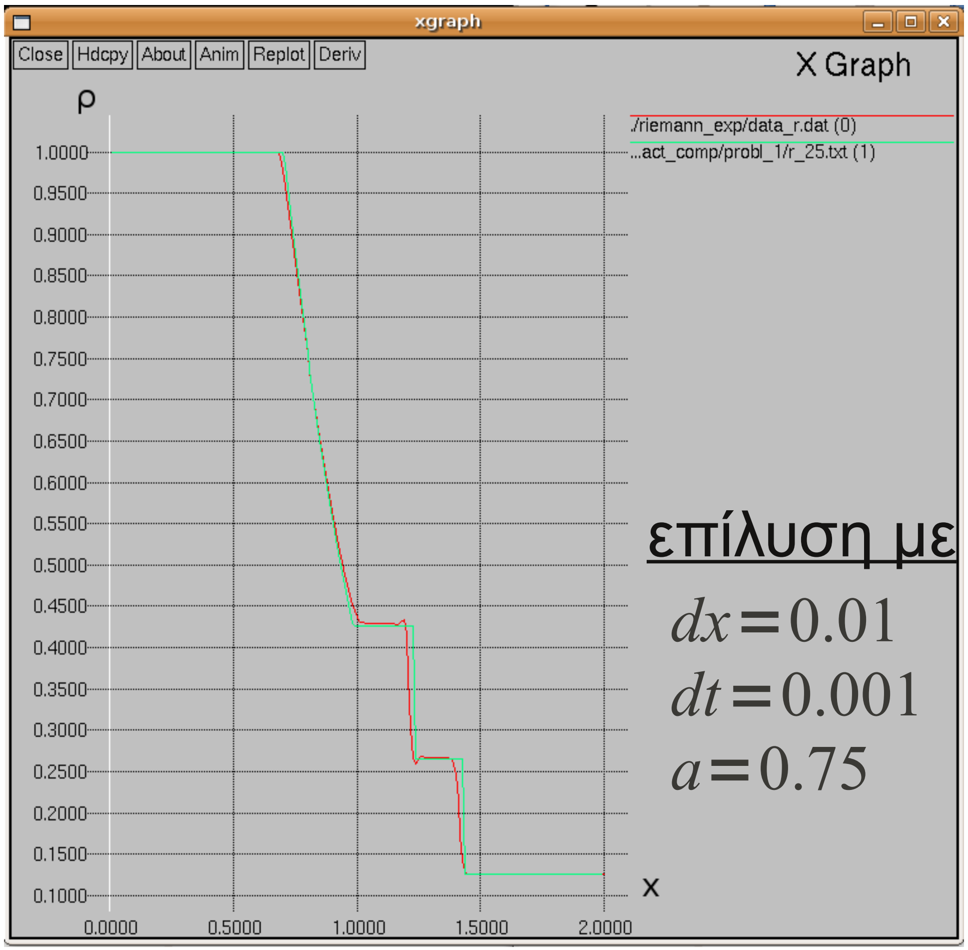
$dt=10^{-6}$

$a=0$

$b=1.5$

Η μέθοδος IDO

Taylor

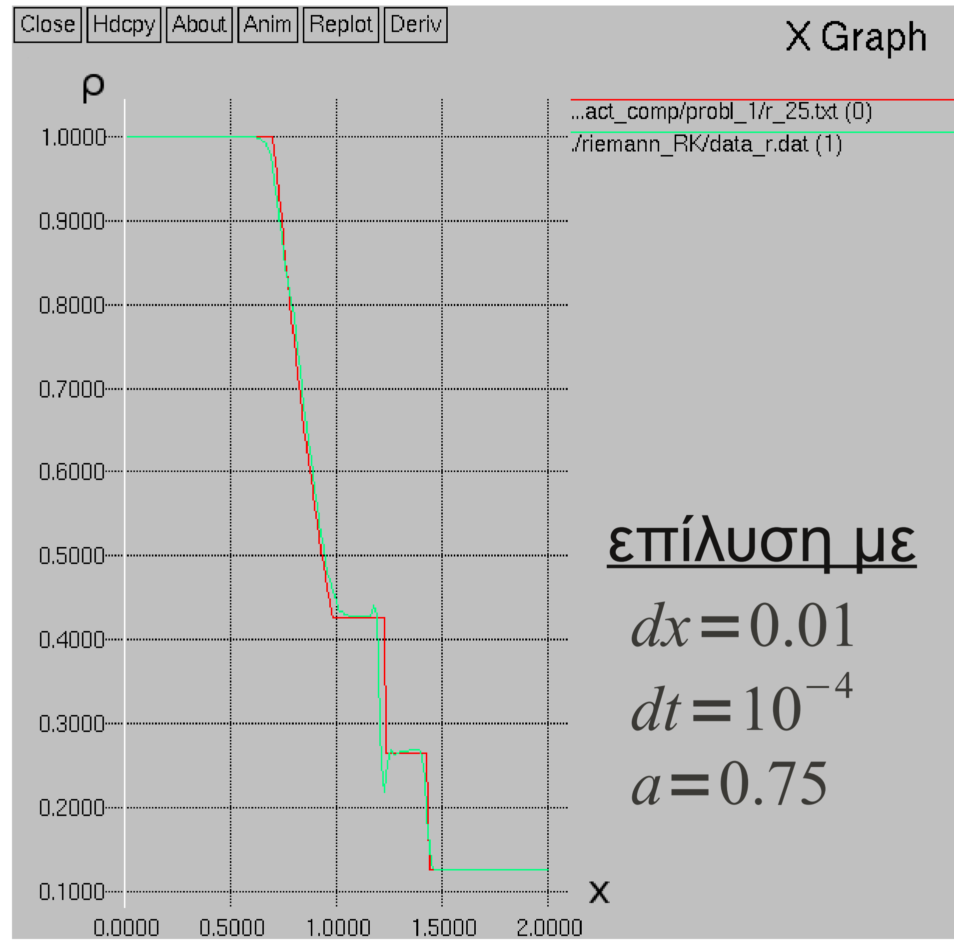


<u>επίλυση με</u>

$$dx = 0.01$$
$$dt = 0.001$$
$$a = 0.75$$

για Euler **1D** !!

Runge Kutta



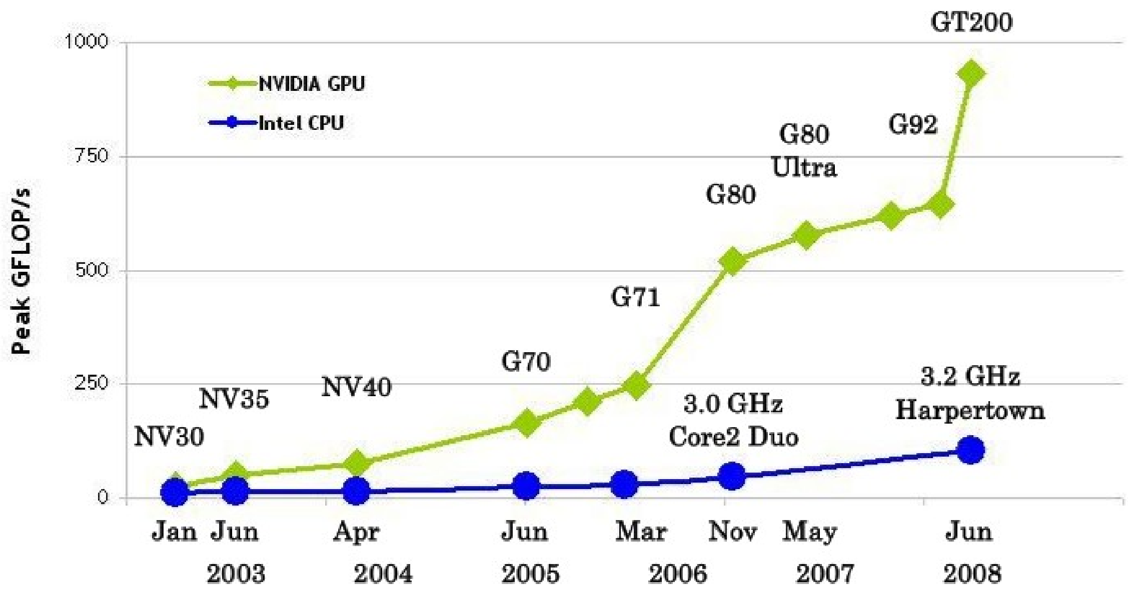
<u>επίλυση με</u>

$$dx = 0.01$$
$$dt = 10^{-4}$$
$$a = 0.75$$

\+ ταχύτερος αλγόριθμος

\- σταθερότητα / ακρίβεια

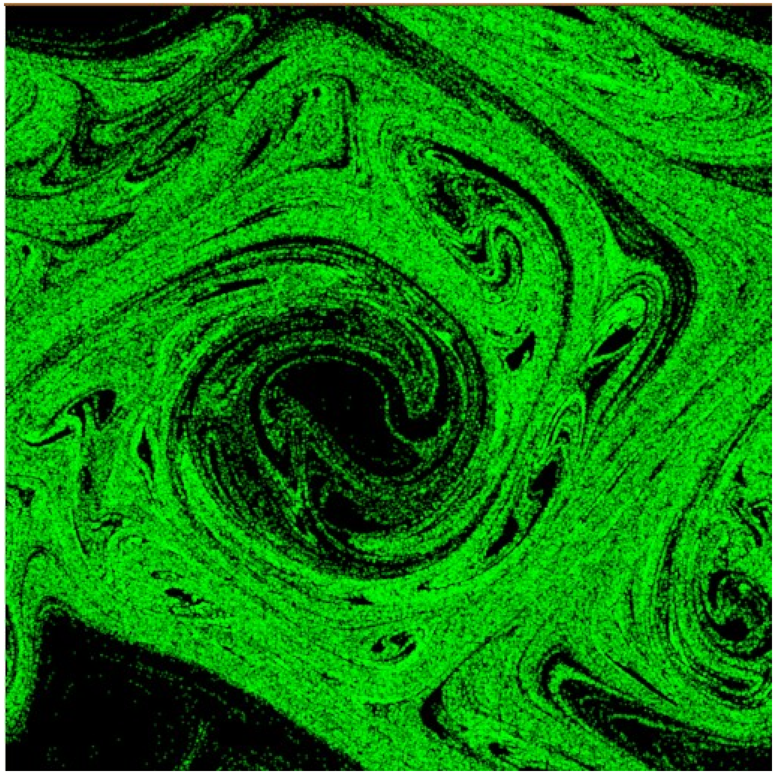
CUDA



παράλληλη επεξεργασία

CPUs - 6 cores

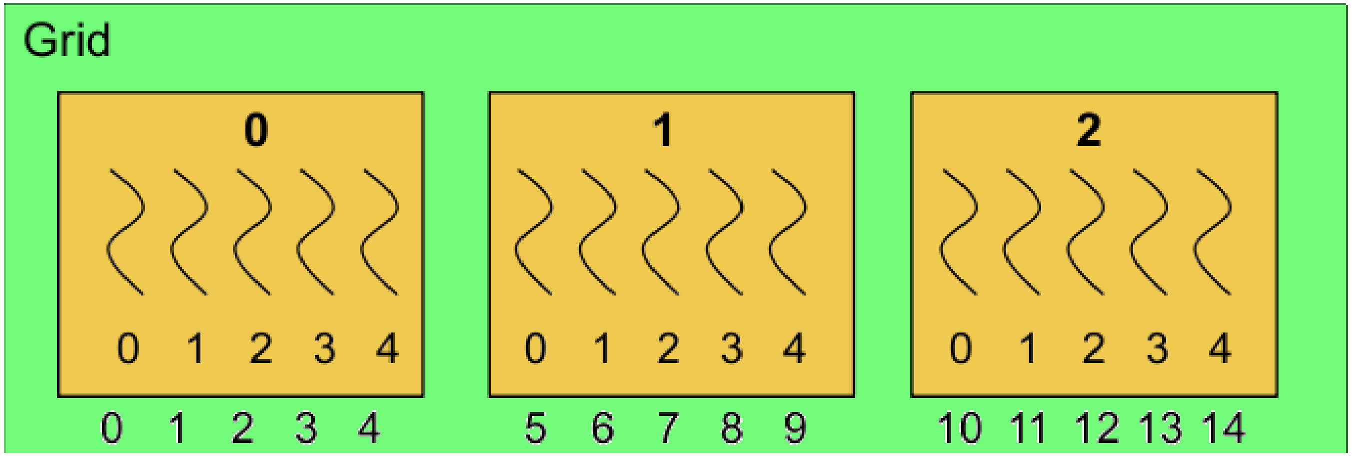
GPUs - 448 cores



CUDA

Προγραμματιστικό Μοντέλο

dim3 var type:
- gridDim
- blockDim
- blockIdx
- threadIdx

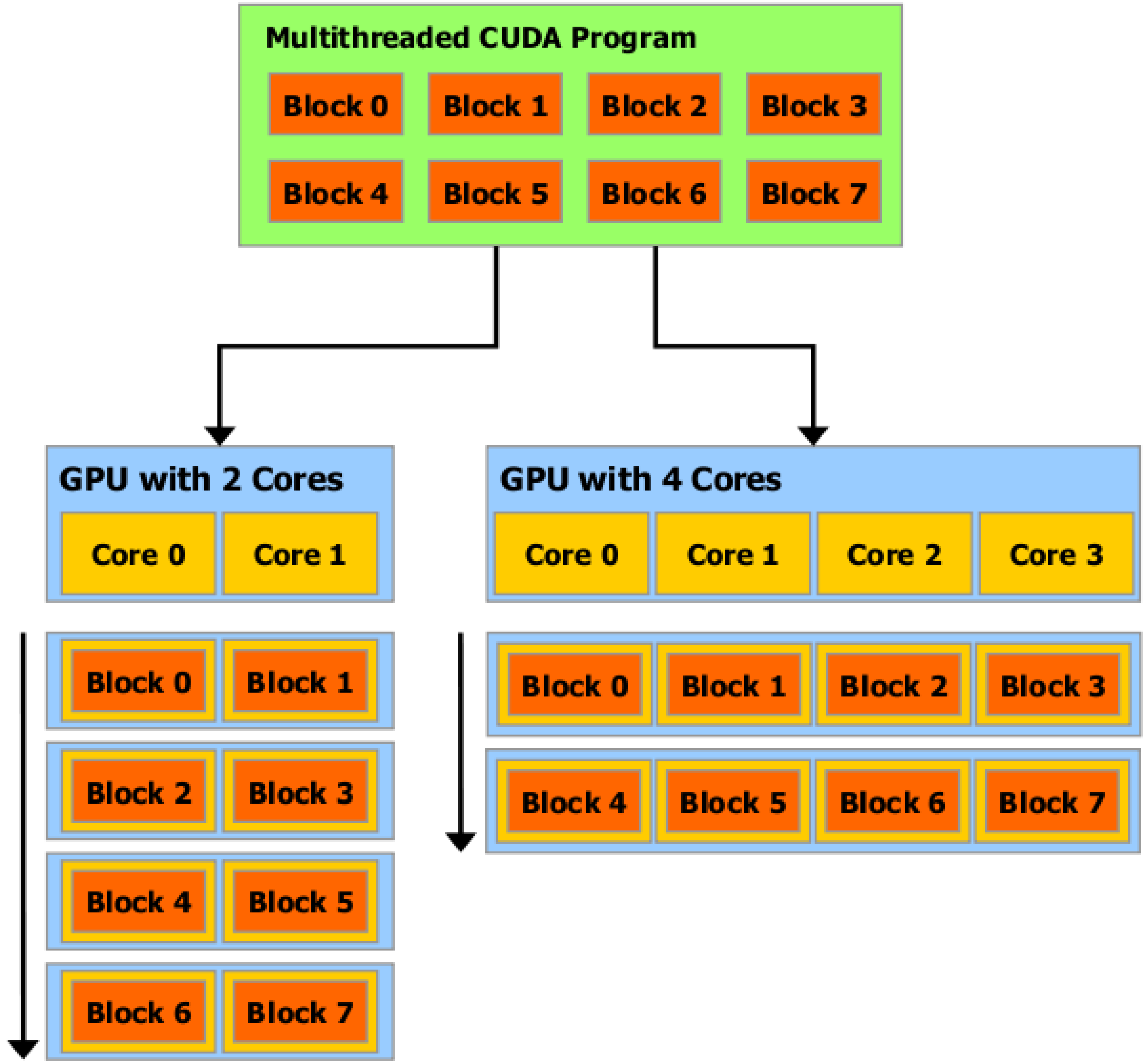



```
__global__ void parallelfunc(int var1, float var2);

int main() {
dim3  blocks_in_grid(3),  threads_per_block(5) ;
....
parallelfunc <<< blocks_in_grid, threads_per_block >>> (a, b)  ;
...
return 0;
}
```

CUDA

Αυτόματα κλιμακούμενη απόδωση



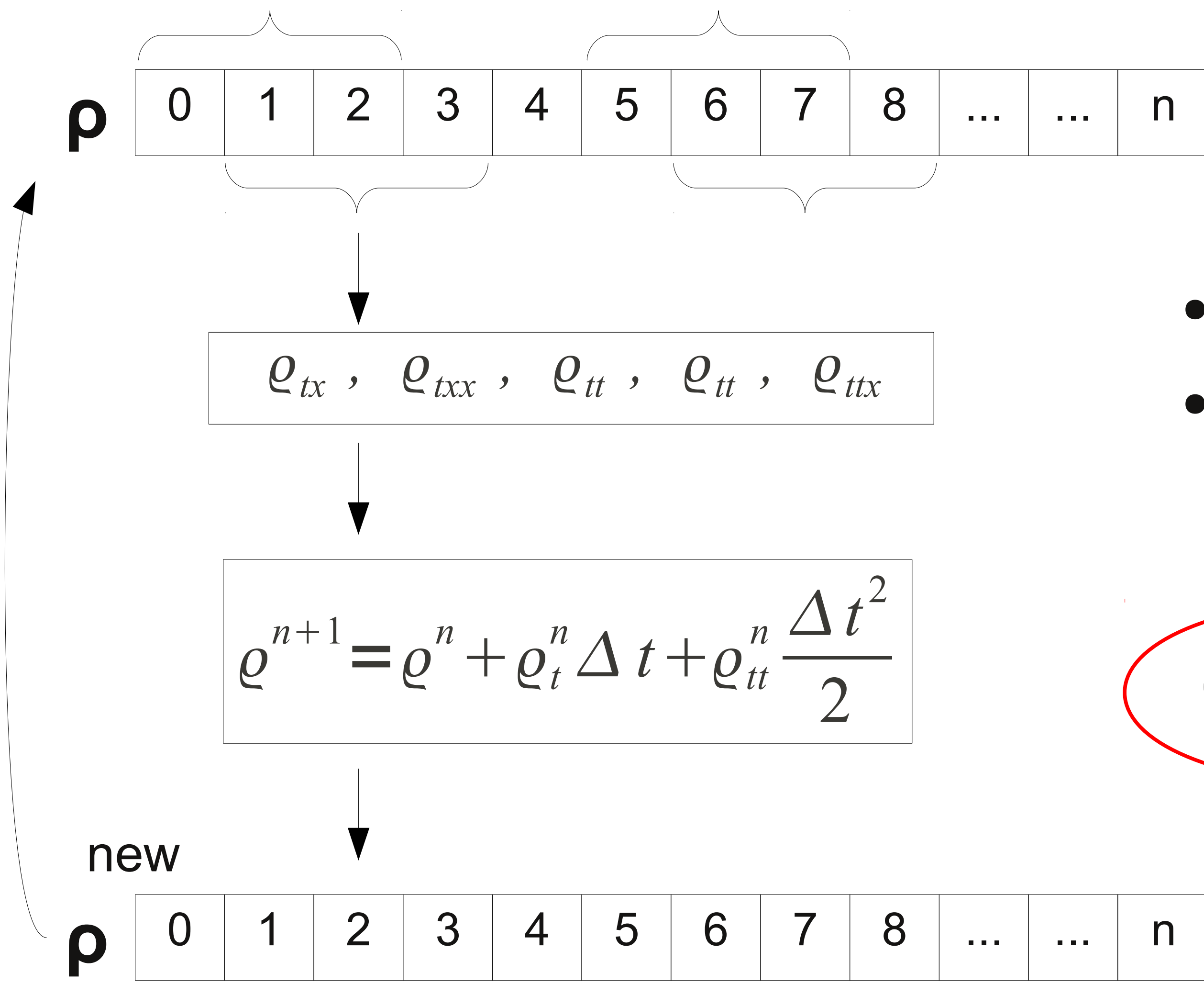
<u>Διαθέσιμοι πόροι:</u>

threads
registers
shared memory

occupancy ~50%

CUDA

<u>IDO on CUDA C</u>



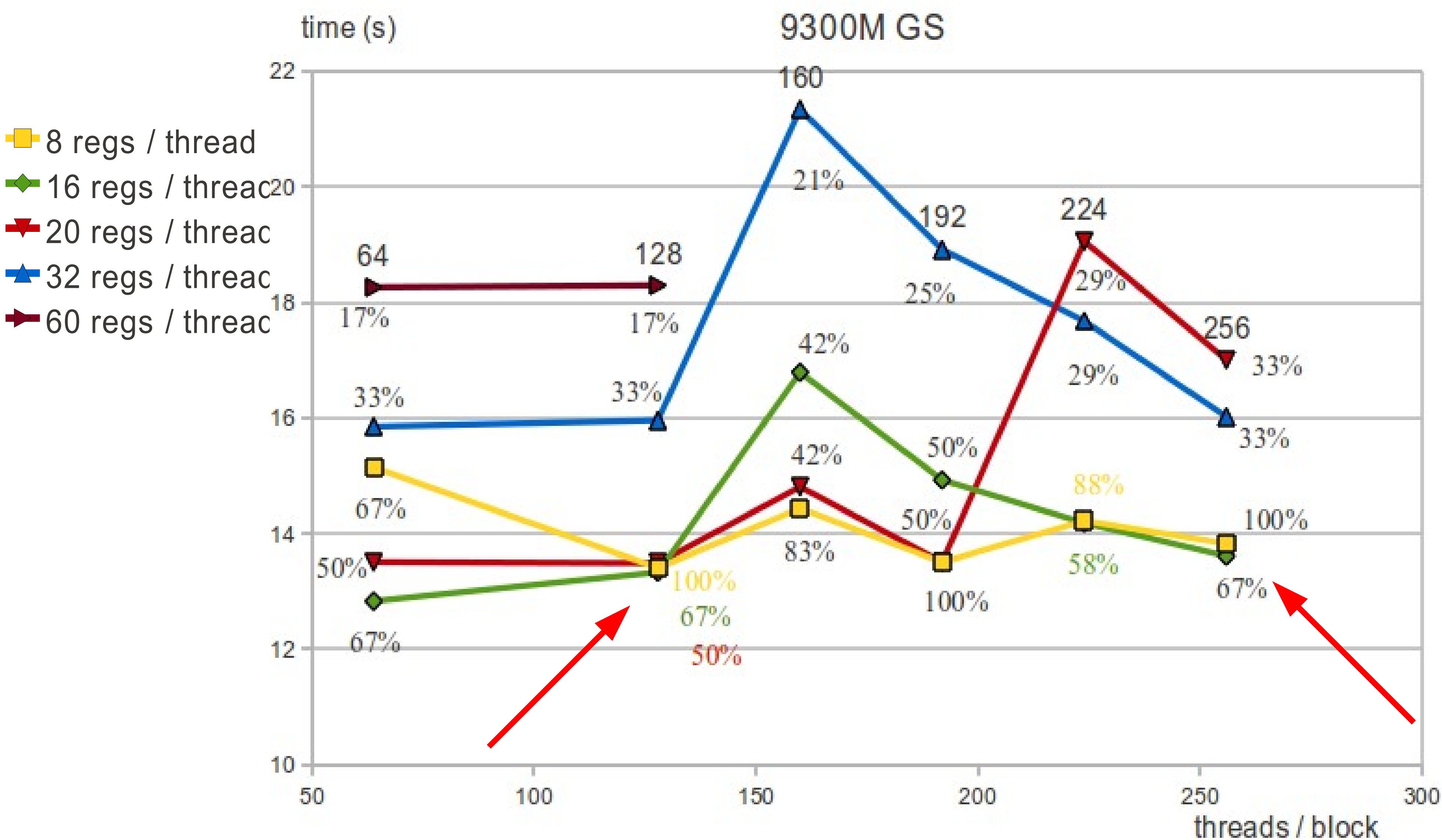
- Ανεξάρτητες διαδικασίες
- shared memory ??

60 registers!

CUDA

Two Interacting Blast Waves / Runge Kutta version

Υπολογιστικό πλέγμα : 20000 σημεία
χρονικά βήματα : 3000

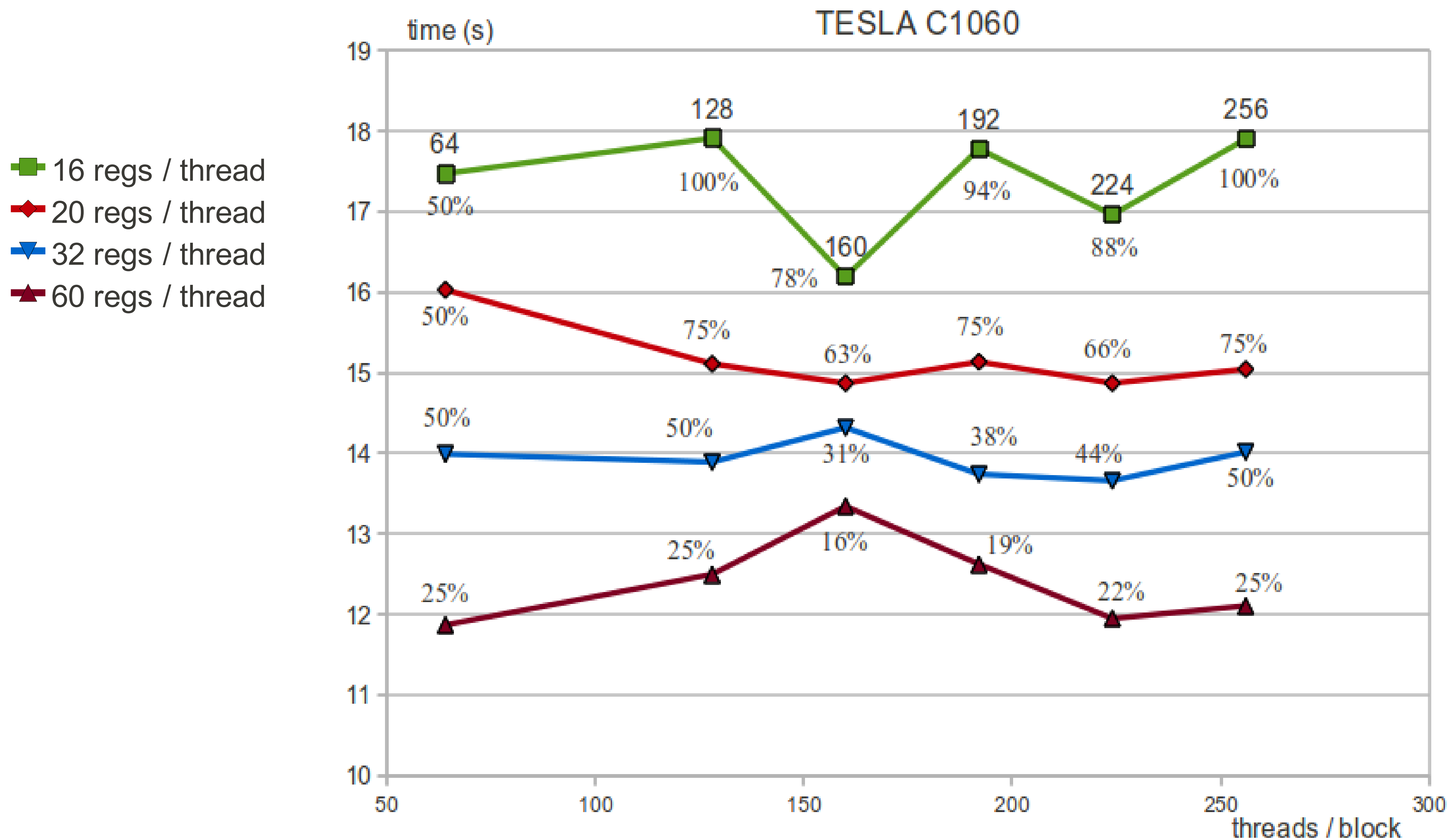


CUDA

Two Interacting Blast Waves / Runge Kutta version

Υπολογιστικό πλέγμα : 40000 σημεία

χρονικά βήματα : 20000



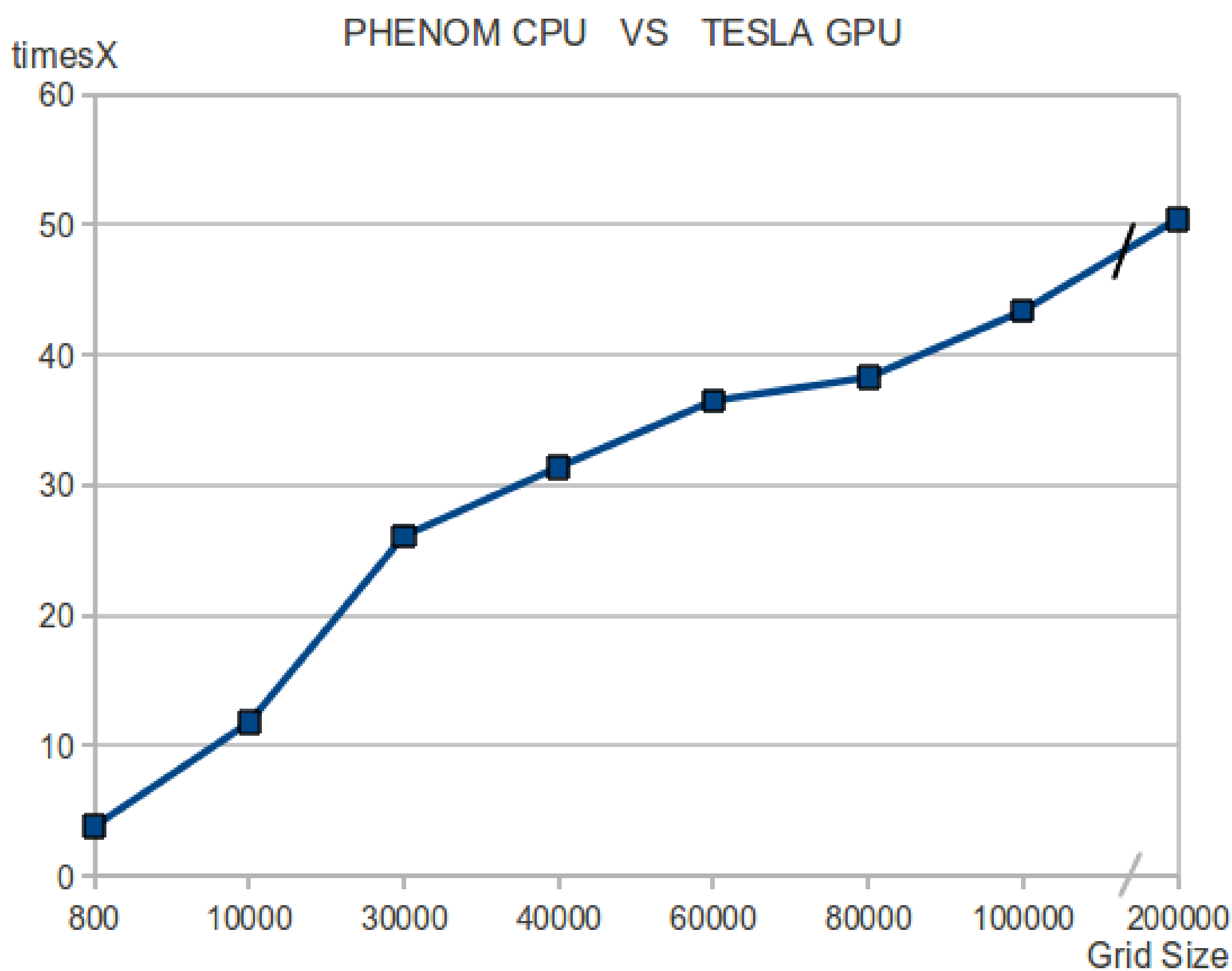
CUDA

Two Interacting Blast Waves / <u>Taylor</u> version

1000 χρονικά βήματα

και όμως:

- μόνο 25% occupancy
- αναξιοποίητα threads
- χωρίς βελτιώσεις κώδικα



CUDA

+	-
• ταχύτατη εκτέλεση • προγραμματιστικό μοντέλο • και ακόμα είναι μόνο η αρχή	• υποστήριξη • debugger (υπάρχει) • περιορισμένη βιβλιογραφία • λοιπά bugs

---

CUDA FORTRAN   **Open CL**   DirectCompute

Ευχαριστώ