Parallel Iterative solution of the Hermite Collocation Equations on GPUs

Emmanuel N. Mathioudakis

Co-Authors : Elena Papadopoulou – Yiannis Saridakis – Nikolaos Vilanakis

TECHNICAL UNIVERSITY OF CRETE
DEPARTMENT OF SCIENCES
APPLIED MATHEMATICS AND COMPUTERS LABORATORY
73100 CHANIA - CRETE - GREECE

Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης
ΕΣΠΑ 2007-2013
πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

Talk Overview

- Hermite Collocation for elliptic BVPs &
  Derivation of the Collocation linear system
- Development of a parallel algorithm for the Schur Complement method on Shared Memory Parallel Architectures
- Parallel implementation on multicore computers with GPUs

Hermite Collocation Method

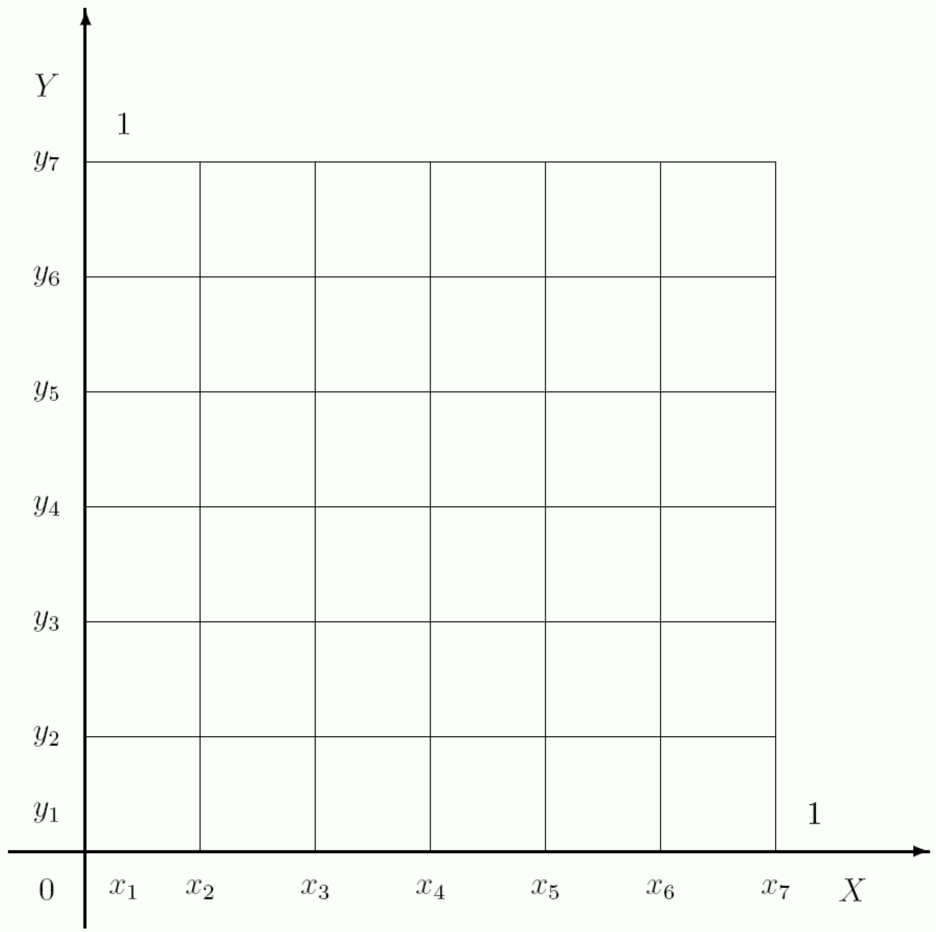
BVP

$$\mathcal{L}u(x,y) = f(x,y) \quad , \quad (x,y) \in \Omega$$

$$\mathcal{B}u(x,y) = g(x,y) \quad , \quad (x,y) \in \partial\Omega$$

$$u(x,y) \sim u_n(x,y) = \sum_{i=1}^{\tilde{n}} \sum_{j=1}^{\tilde{n}} a_{i,j} \underbrace{\phi_i(x)\phi_j(y)}_{Hermite\ Cubics}$$

$$a_{i,j} \; : \begin{cases} \mathcal{L}u(\sigma_i^x, \sigma_j^y) - f(\sigma_i^x, \sigma_j^y) = 0 \quad , \quad (\sigma_i^x, \sigma_j^y) \in \Omega \\ \mathcal{B}u(\xi_i^x, \xi_j^y) - g(\xi_i^x, \xi_j^y) = 0 \quad , \quad (\xi_i^x, \xi_j^y) \in \partial\Omega \end{cases}$$



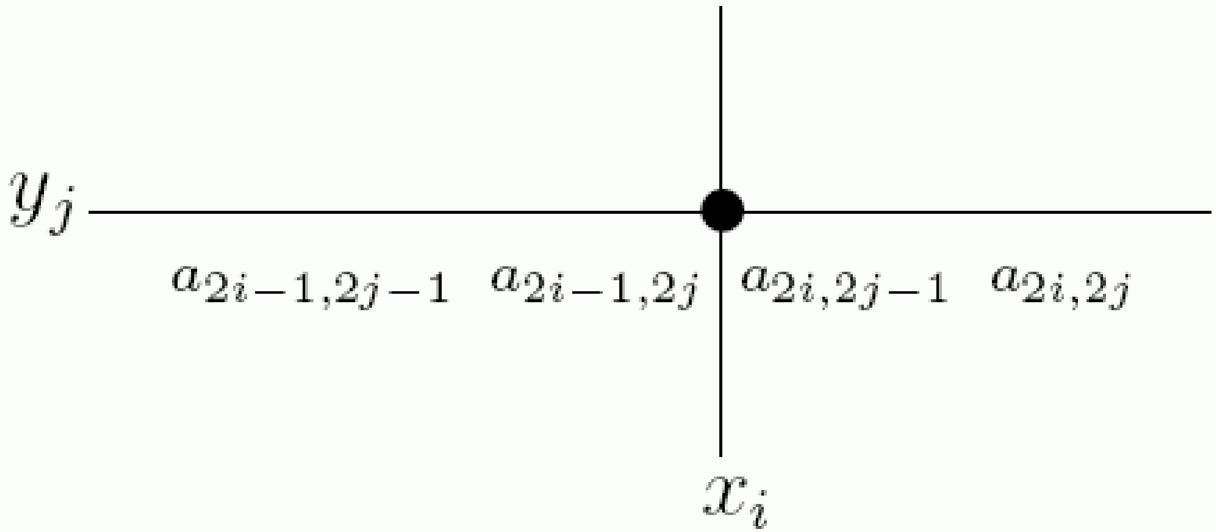
$$h = \frac{1}{n_s}$$

$$x_i = (i-1)h$$

$$y_j = (j-1)h$$

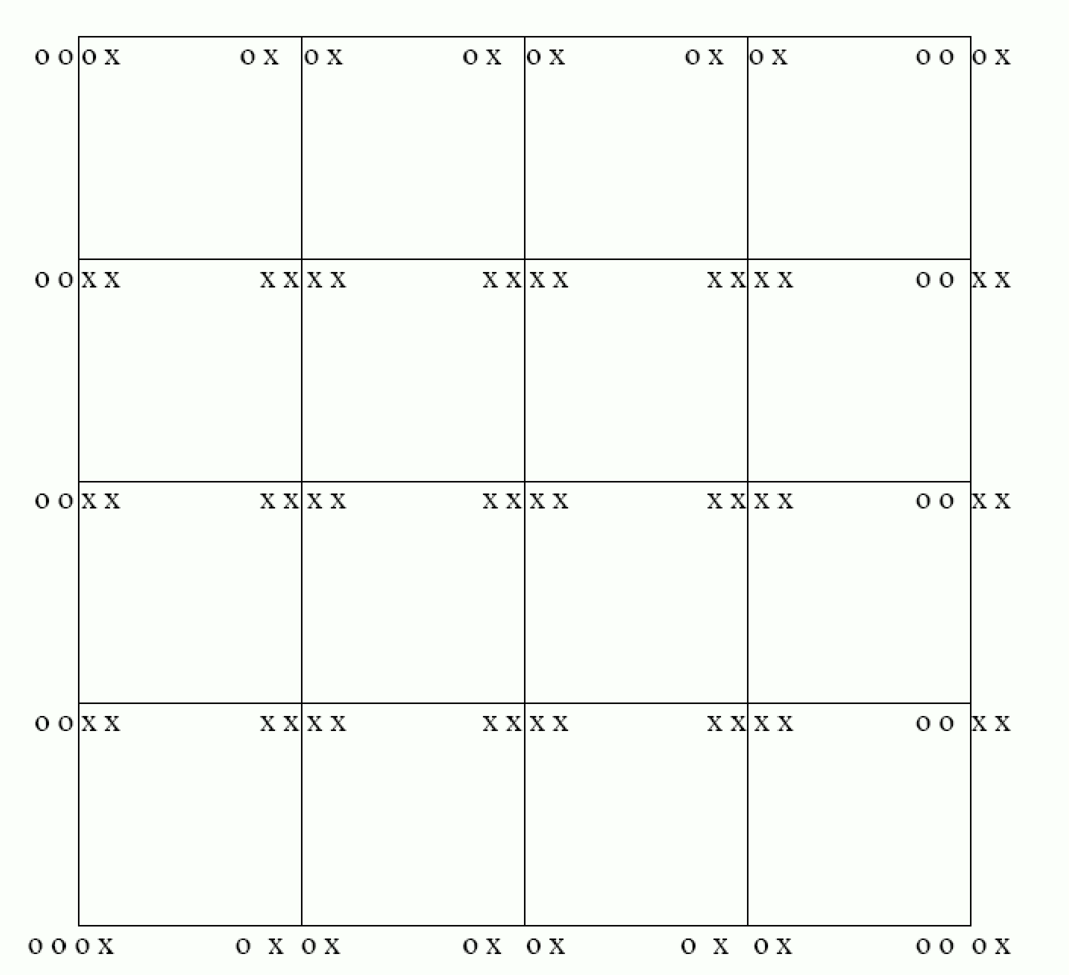
$$i, j = 1, \ldots, (n_s+1)$$

Hermite Collocation Method…

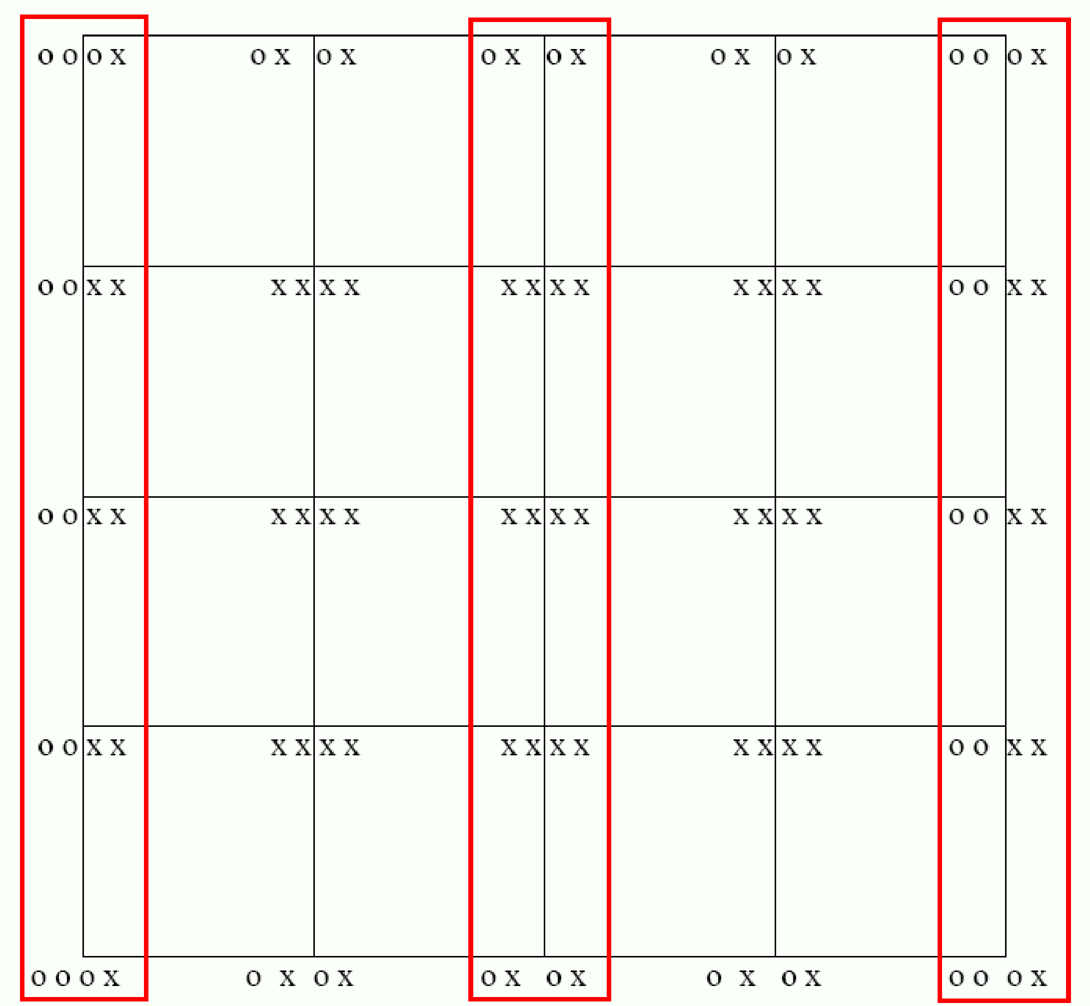


$$\begin{cases} u_n(x_i,y_j) = a_{2i-1,2j-1} \quad , \quad h\frac{\partial}{\partial y}u_n(x_i,y_j) = a_{2i-1,2j} \\ h\frac{\partial}{\partial x}u_n(x_i,y_j) = a_{2i,2j-1} \quad , \quad h^2\frac{\partial^2}{\partial x \partial y}u_n(x_i,y_j) = a_{2i,2j} \end{cases}$$

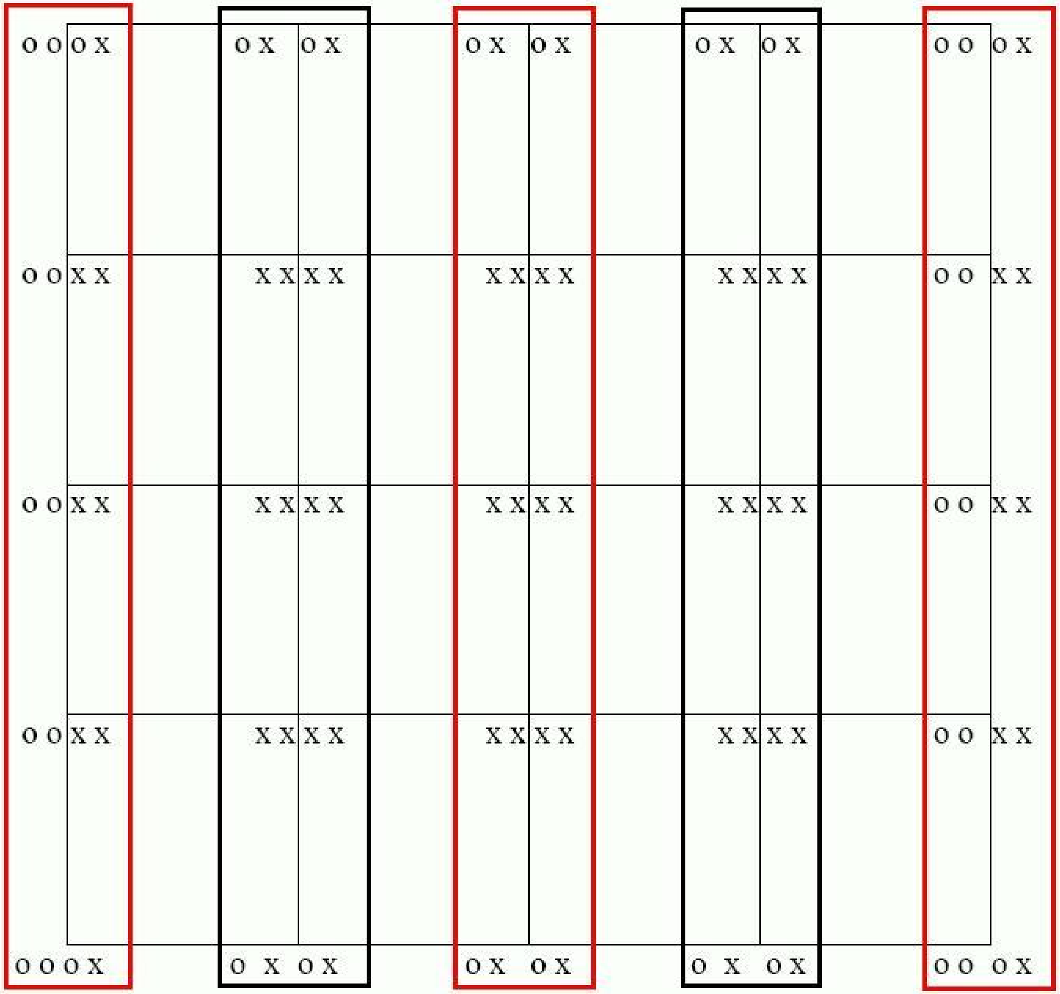
Red – Black Collocation Linear system



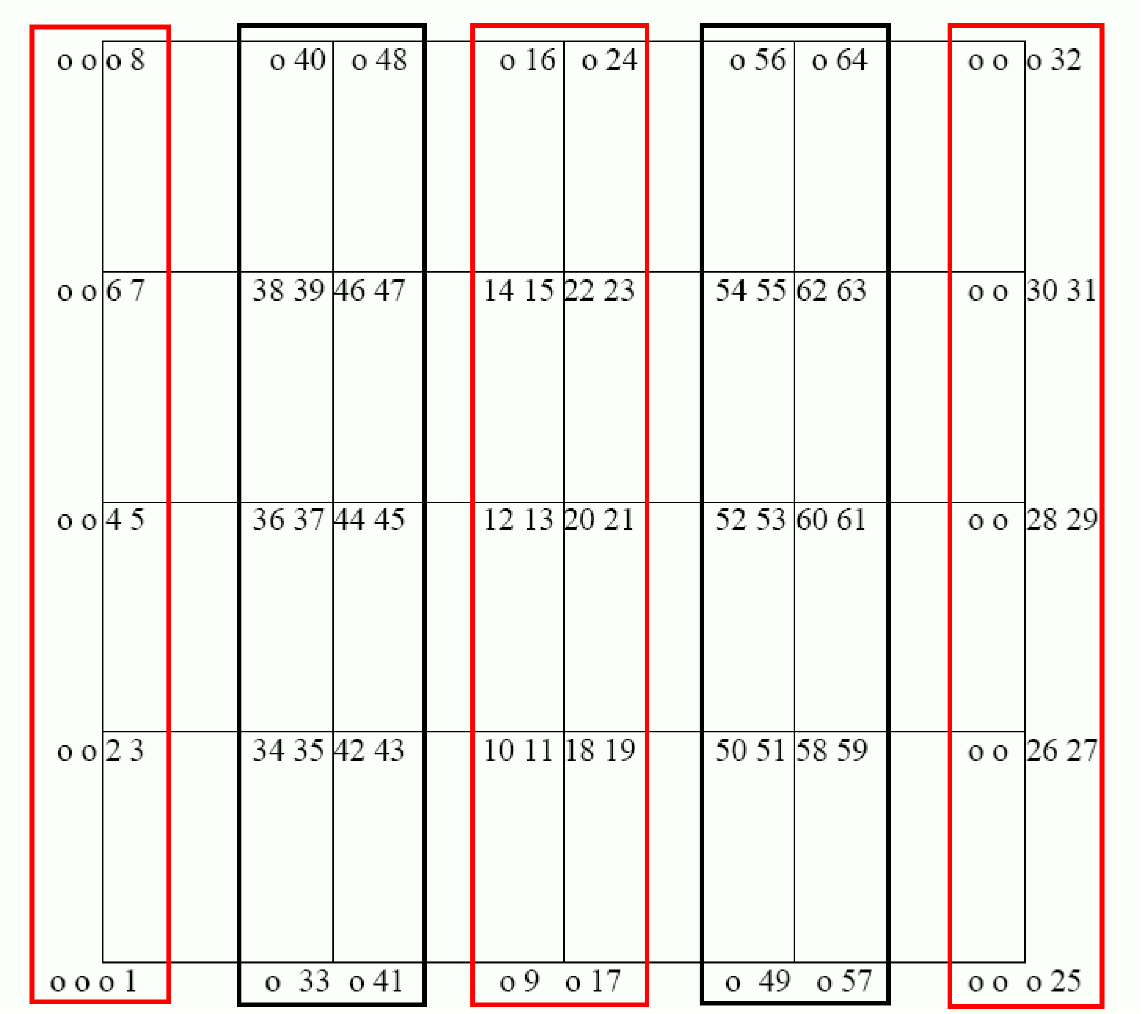
Red – Black Collocation Linear system



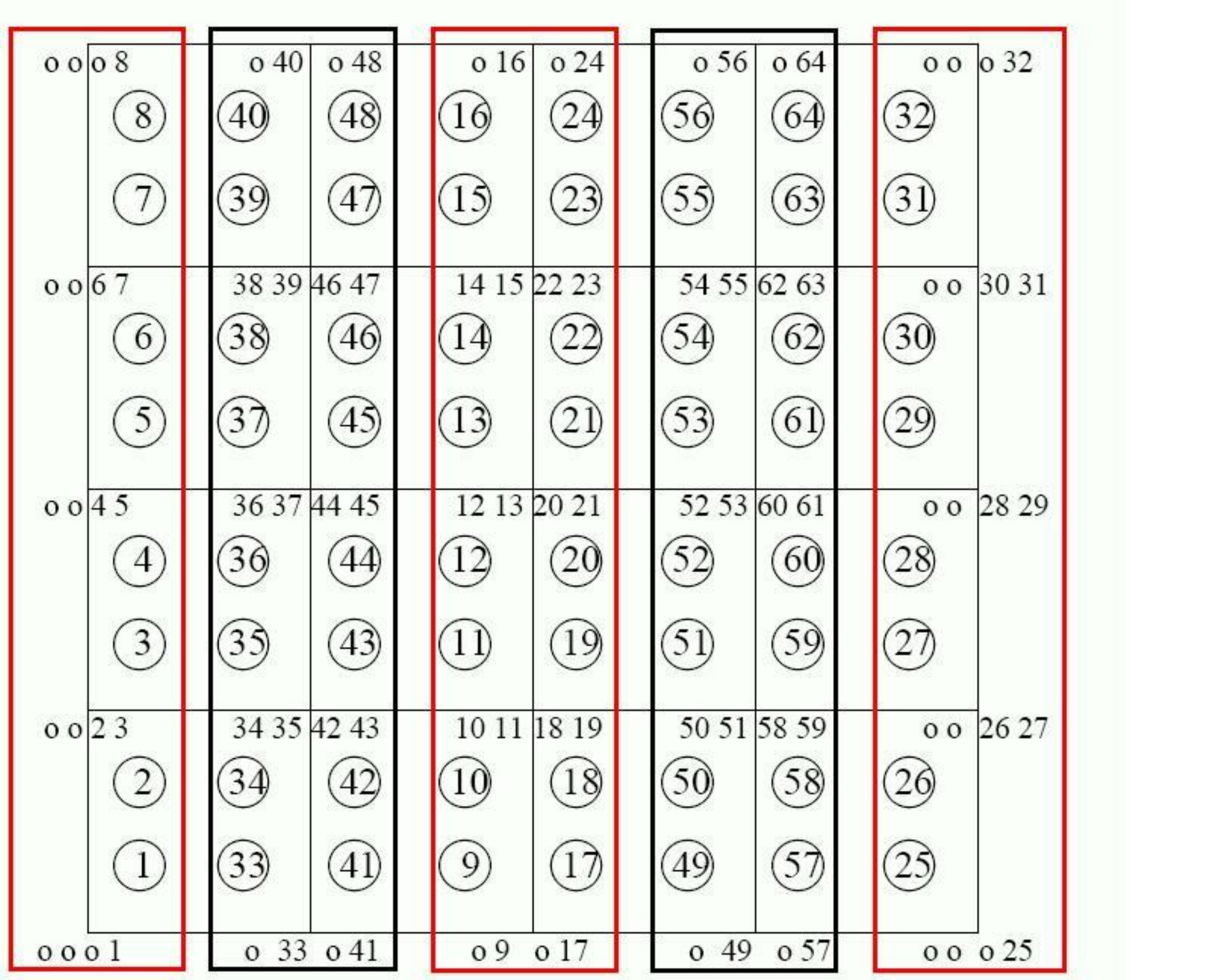
Red – Black Collocation Linear system



Red – Black Collocation Linear system



Red – Black Collocation Linear system

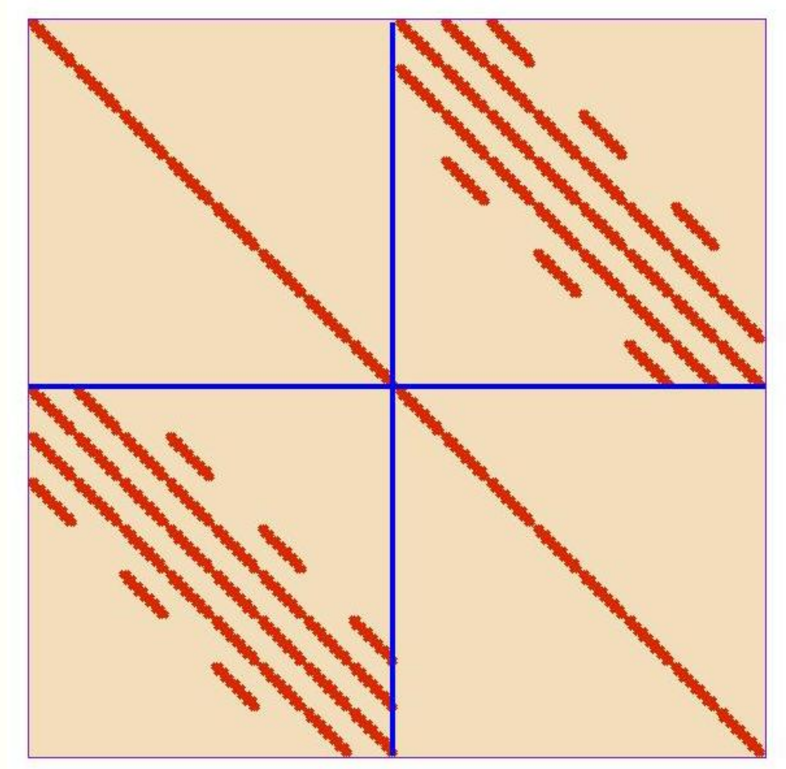


Model Problem

$$\nabla^2 u(x,y) - \lambda u(x,y) = f(x,y) \quad , \quad (x,y) \in \Omega$$

$$u(x,y) = g(x,y) \quad , \quad (x,y) \in \partial\Omega \quad \text{with} \quad \lambda \geq 0$$

$$A = \begin{bmatrix} D_R & H_B \\ H_R & D_B \end{bmatrix}$$



Helmholtz Collocation Matrix

$$A = \begin{bmatrix} D_R & H_B \\ H_R & D_B \end{bmatrix}$$

$$H_B = -\begin{bmatrix} A_3 & -A_4 & O & O & \cdots & O & O & O & O \\ A_3 & A_4 & A_3 & -A_4 & \cdots & O & O & O & O \\ -A_3 & -A_4 & A_3 & -A_4 & \cdots & O & O & O & O \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ O & O & O & O & \cdots & A_3 & A_4 & A_3 & -A_4 \\ O & O & O & O & \cdots & -A_3 & -A_4 & A_3 & -A_4 \\ O & O & O & O & \cdots & O & O & A_3 & A_4 \end{bmatrix}$$

$$D_R = 2\,\mathrm{diag}[\frac{1}{2}A_2 \; A_1 \; A_2 \; \cdots \; A_1 \; A_2 \;\; -\frac{1}{2}A_2]$$

$$D_B = 2\,\mathrm{diag}[A_1 \; A_2 \; \cdots \; A_1 \; A_2]$$

$$H_R = -\begin{bmatrix} A_4 & A_3 & -A_4 & O & O & \cdots & O & O & O & O & O \\ -A_4 & A_3 & -A_4 & O & O & \cdots & O & O & O & O & O \\ O & A_3 & A_4 & A_3 & -A_4 & \cdots & O & O & O & O & O \\ O & -A_3 & -A_4 & A_3 & -A_4 & \cdots & O & O & O & O & O \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots \\ O & O & O & O & O & \cdots & A_3 & A_4 & A_3 & -A_4 & O \\ O & O & O & O & O & \cdots & -A_3 & -A_4 & A_3 & -A_4 & O \\ O & O & O & O & O & \cdots & O & O & A_3 & A_4 & -A_4 \\ O & O & O & O & O & \cdots & O & O & -A_3 & -A_4 & -A_4 \end{bmatrix}$$

Red – Black Collocation Linear system

$$A_i = \begin{bmatrix}
a_2 & a_3 & -a_4 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 \\
a_4 & a_1 & -a_2 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 \\
0 & a_1 & a_2 & a_3 & -a_4 & \cdots & 0 & 0 & 0 & 0 & 0 \\
0 & a_3 & a_4 & a_1 & -a_2 & \cdots & 0 & 0 & 0 & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & 0 & 0 & 0 & 0 & \cdots & a_1 & a_2 & a_3 & -a_4 & 0 \\
0 & 0 & 0 & 0 & 0 & \cdots & a_3 & a_4 & a_1 & -a_2 & 0 \\
0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & a_1 & a_2 & -a_4 \\
0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & a_3 & a_4 & -a_2
\end{bmatrix}$$

	a_1	a_2	a_3	a_4
A_1	r^+	s^+	q	t^+
A_2	s^+	u^+	t^-	ϵ
A_3	q	t^-	r^-	s^-
A_4	t^+	ϵ	s^-	u^-

with

$$\epsilon = -\frac{\lambda}{24n_s^2} \;, \quad q = 24 + 22\epsilon \;,$$

$$r^{\pm} = 86\epsilon - 24 \pm (48\epsilon - 18)\sqrt{3} \;,$$

$$s^{\pm} = 13\epsilon - 12 \pm (7\epsilon - 8)\sqrt{3} \;,$$

$$t^{\pm} = 5\epsilon + 3 \pm (\epsilon + 1)\sqrt{3} \;,$$

$$u^{\pm} = 2\epsilon - 3 \pm (\epsilon - 2)\sqrt{3}.$$

Iterative
+
Parallel

The collocation matrix is large, sparse and enjoys no pleasant properties (e.g. symmetric, definite)

Iterative Solution

$$A = \begin{bmatrix} D_R & H_B \\ H_R & D_B \end{bmatrix}$$

$$A = D_A - L_A - U_A$$

with

$$D_A = \begin{bmatrix} D_R & O \\ O & D_B \end{bmatrix} , \; L_A = \begin{bmatrix} O & O \\ -H_R & O \end{bmatrix} , \; U_A = \begin{bmatrix} O & -H_B \\ O & O \end{bmatrix}$$

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{x}_R \\ \boldsymbol{x}_B \end{bmatrix} \text{ and } \boldsymbol{b} = \begin{bmatrix} \boldsymbol{b}_R \\ \boldsymbol{b}_B \end{bmatrix} .$$

Iterative Solution

$$M_1^{-1}\, A\, M_2^{-1}\, M_2 \boldsymbol{x} \;=\; M_1^{-1} \boldsymbol{b}$$

where $M_1 \;=\; D_A - L_A \;=\; D_A(I - D_A^{-1} L_A)$

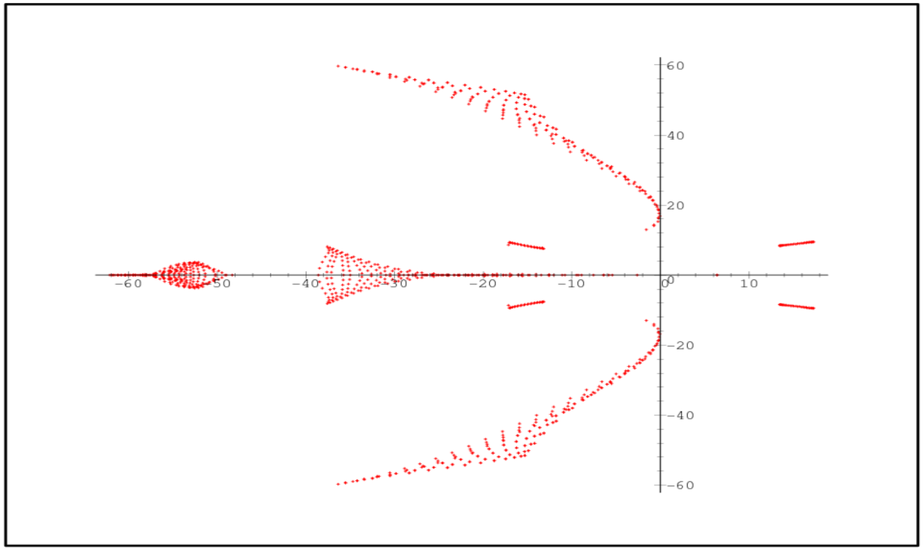
and $M_2 \;=\; I - D_A^{-1} U_A$

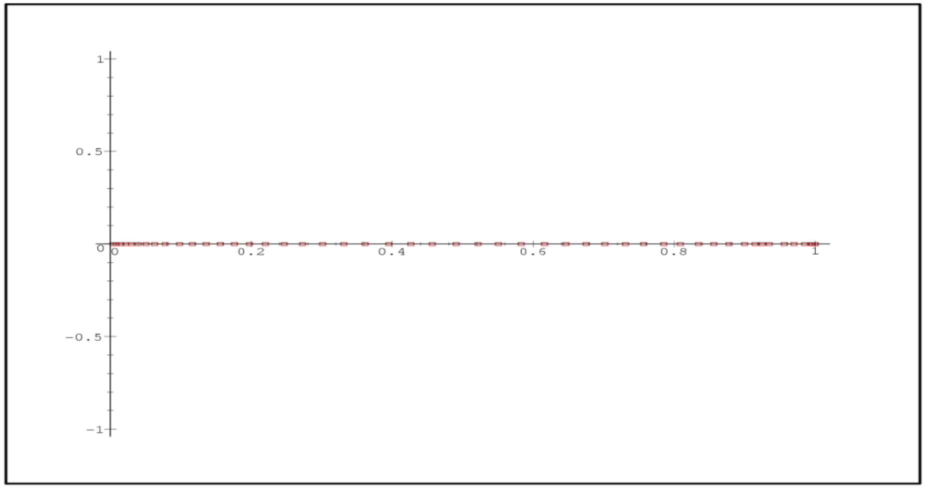
$$\begin{bmatrix} I & O \\ O & S \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_R + D_R^{-1} H_R \boldsymbol{x}_B \\ \boldsymbol{x}_B \end{bmatrix} = \begin{bmatrix} \hat{\boldsymbol{b}}_R \\ \hat{\boldsymbol{b}}_B \end{bmatrix}$$

where $S = D_B - H_R D_R^{-1} H_B$

$$\hat{\boldsymbol{b}}_R \;=\; D_R^{-1} \boldsymbol{b}_R \quad \text{and} \quad \hat{\boldsymbol{b}}_B \;=\; \boldsymbol{b}_B - H_R \hat{\boldsymbol{b}}_R$$

Eigenvalues of Collocation matrix





TECHNICAL UNIVERSITY OF CRETE
APPLIED MATHEMATICS AND COMPUTERS LABORATORY

Schur Complement Iterative Solution

S1: <u>*Solve*</u> $D_R \hat{\boldsymbol{b}}_R = \boldsymbol{b}_R$

S2: <u>*Evaluate*</u> $\hat{\boldsymbol{b}}_B = \boldsymbol{b}_B - H_R \hat{\boldsymbol{b}}_R$

S3: <u>*Solve with BiCGSTAB*</u> $S\, \boldsymbol{x}_B = \hat{\boldsymbol{b}}_B$

S4: <u>*Evaluate*</u> $\hat{\boldsymbol{x}}_B = H_B \boldsymbol{x}_B$

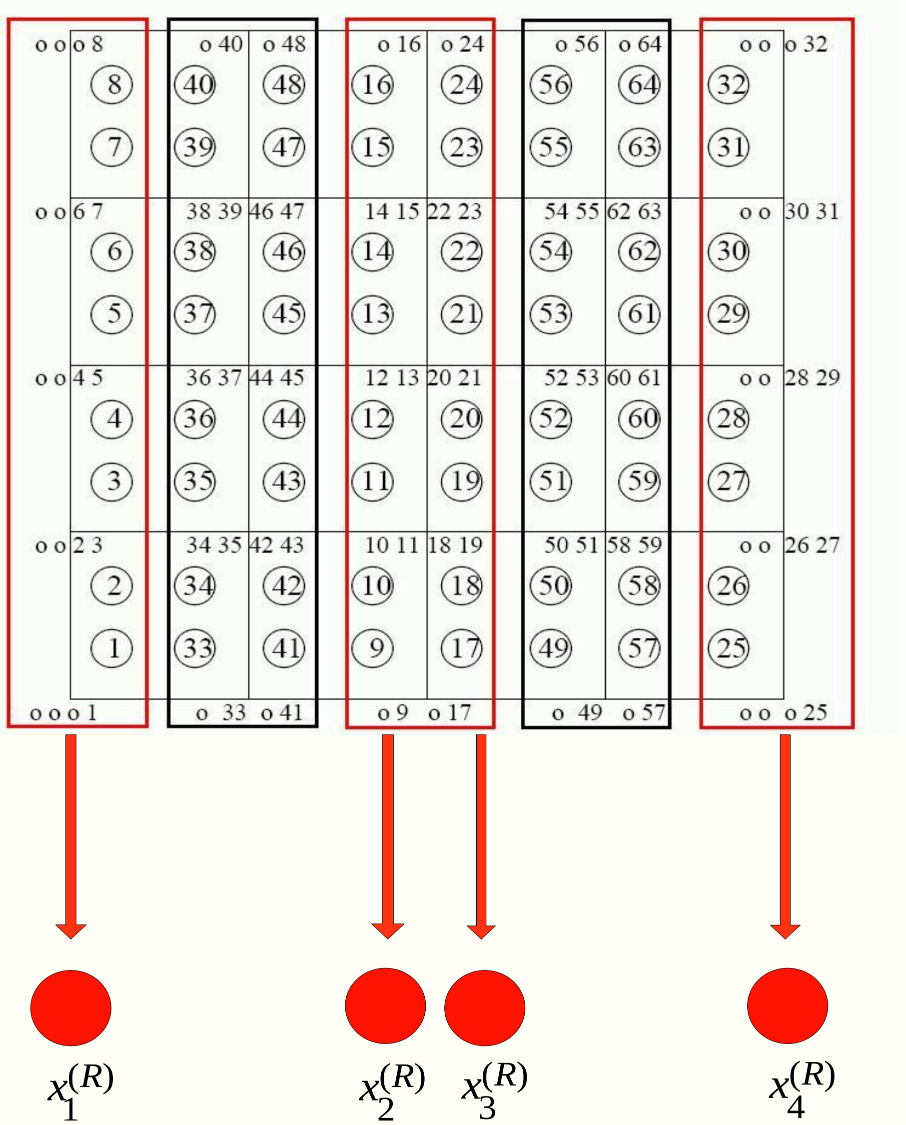
S5: <u>*Solve*</u> $D_R \hat{\boldsymbol{x}}_R = \hat{\boldsymbol{x}}_B$

S6: <u>*Evaluate*</u> $\boldsymbol{x}_R = \hat{\boldsymbol{b}}_R - \hat{\boldsymbol{x}}_R$

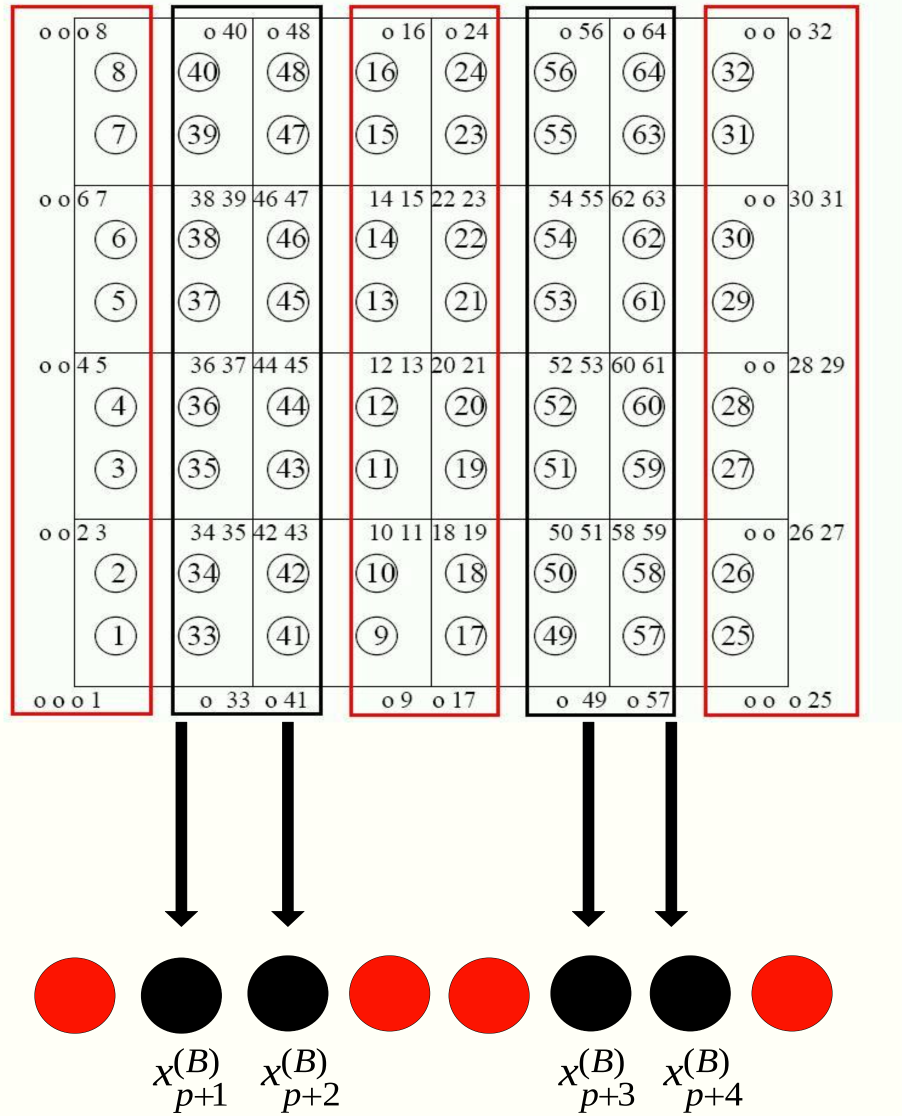
Parallel Iterative Solution of Collocation Linear system on Shared Memory Architectures

- Uniform Load Balancing between core threads
- Minimal Idle Cycles of core threads
- Minimal Communication

case of $n_s = 2p$

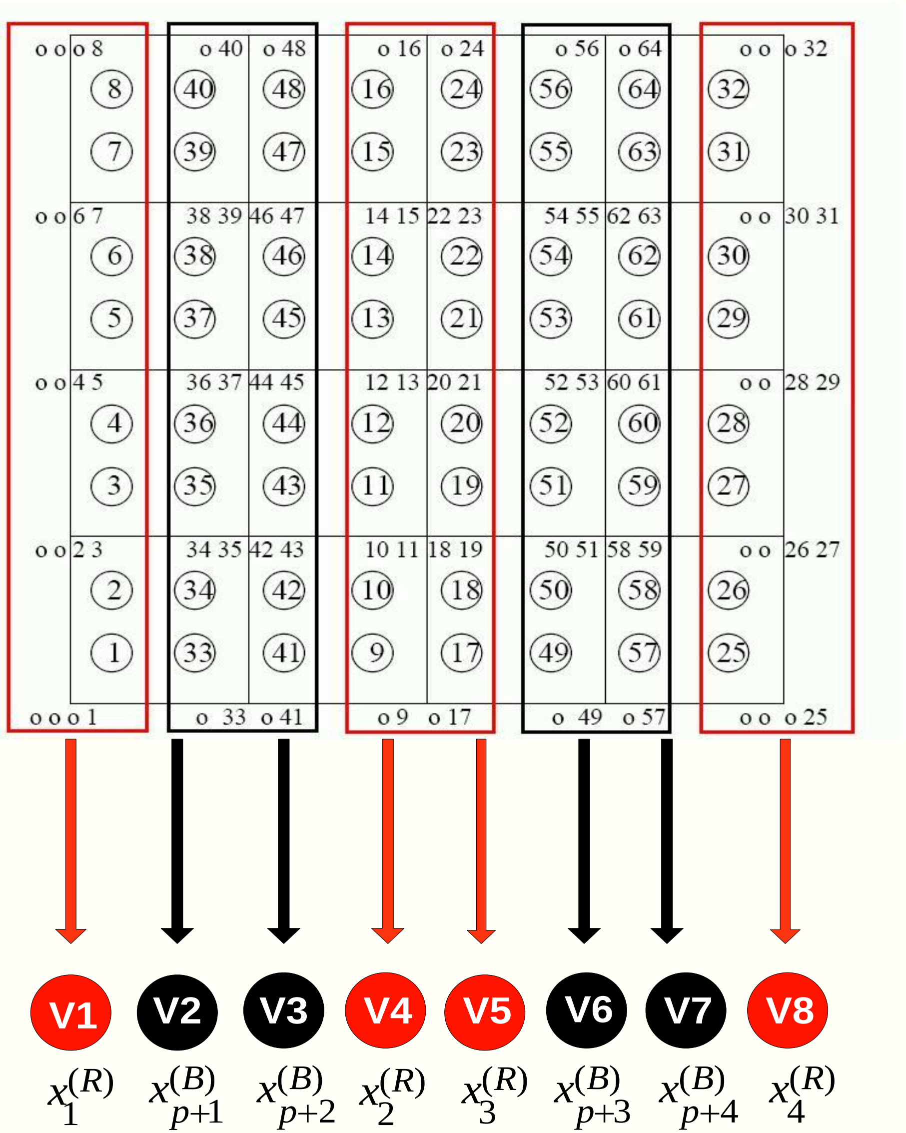


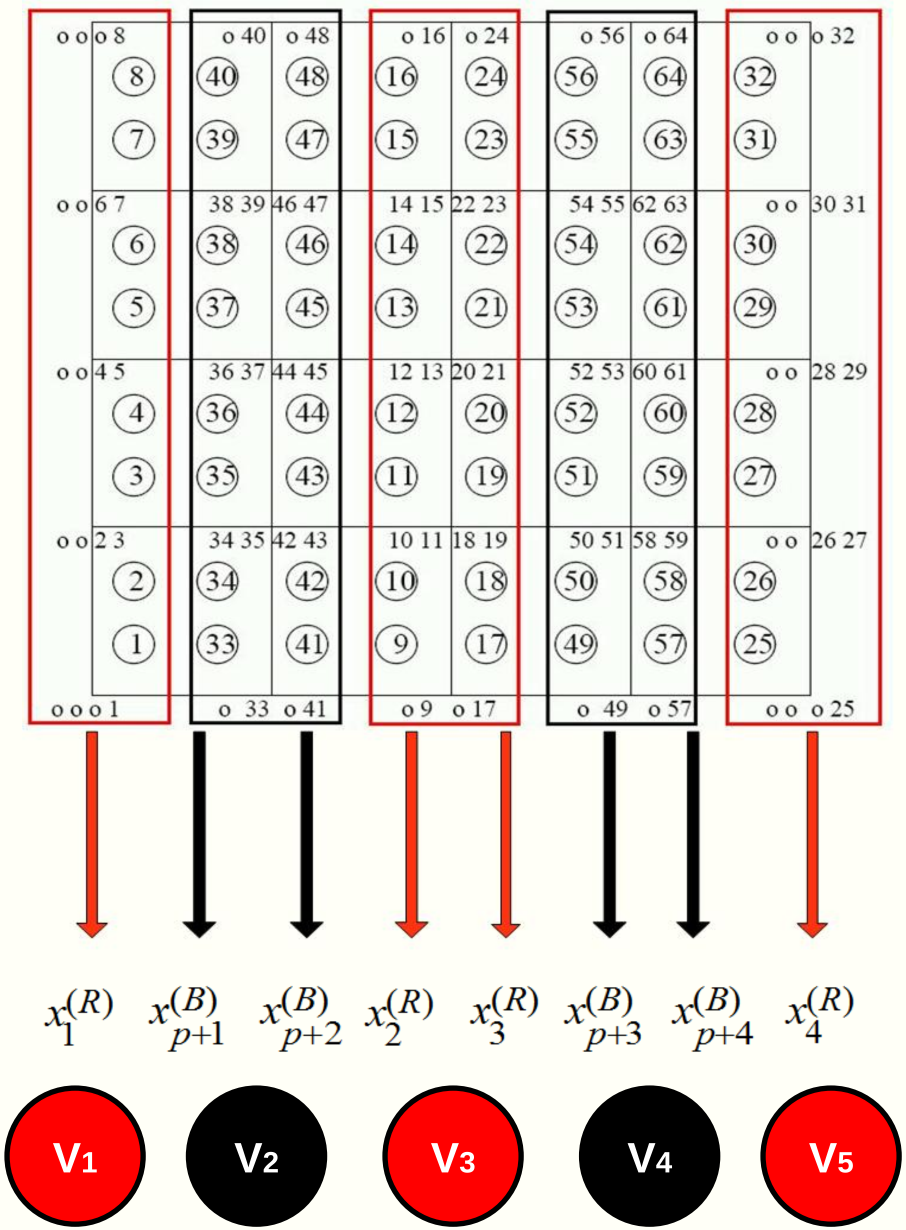
case of $n_s = 2p$



o o o 8 | o 40 o 48 | o 16 o 24 | o 56 o 64 | o o o 32
8 40 48 16 24 56 64 32
7 39 47 15 23 55 63 31
o o 6 7 | 38 39 46 47 | 14 15 22 23 | 54 55 62 63 | o o 30 31
6 38 46 14 22 54 62 30
5 37 45 13 21 53 61 29
o o 4 5 | 36 37 44 45 | 12 13 20 21 | 52 53 60 61 | o o 28 29
4 36 44 12 20 52 60 28
3 35 43 11 19 51 59 27
o o 2 3 | 34 35 42 43 | 10 11 18 19 | 50 51 58 59 | o o 26 27
2 34 42 10 18 50 58 26
1 33 41 9 17 49 57 25
o o o 1 | o 33 o 41 | o 9 o 17 | o 49 o 57 | o o o 25
$x^{(B)}_{p+1}$
$x^{(B)}_{p+2}$
$x^{(B)}_{p+3}$
$x^{(B)}_{p+4}$

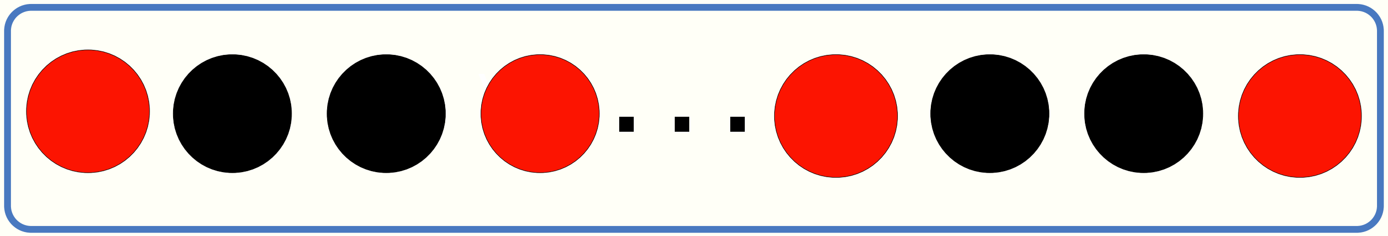
case of $n_s = 2p$





$x_1^{(R)}$
$x_{p+1}^{(B)}$
$x_{p+2}^{(B)}$
$x_2^{(R)}$
$x_3^{(R)}$
$x_{p+3}^{(B)}$
$x_{p+4}^{(B)}$
$x_4^{(R)}$
V_1
V_2
V_3
V_4
V_5

Mapping into a fixed size Architecture of N Cores

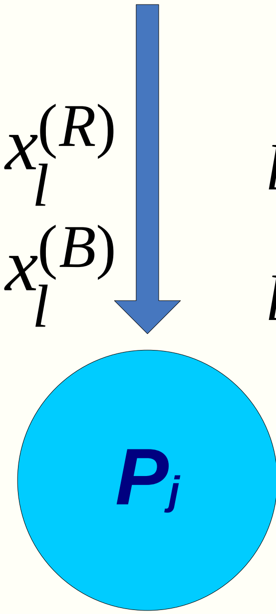


case of $k = 2p/N$ even

$2k$ virtual threads

$x_l^{(R)}$ $\quad l=(j-1)k+1,\ldots,jk$

$x_l^{(B)}$ $\quad l=2p+(j-1)k+1,\ldots,2p+jk$



$j=1,\ldots,N$

Parallel Schur Complement Iterative Solution

S1: *Solve in parallel on host* $D_R\hat{\mathbf{b}}_R = \mathbf{b}_R$

S2: *Send matrices* A_3 *and* A_4 *to GPU*

S3: *Evaluate in parallel on GPU* $\hat{\mathbf{b}}_B = \mathbf{b}_B - H_R\hat{\mathbf{b}}_R$

S4: *Solve in parallel with BiCGSTAB* $S\,\boldsymbol{x}_B = \hat{\mathbf{b}}_B$

S5: *Evaluate in parallel on GPU* $\hat{\boldsymbol{x}}_B = H_B\boldsymbol{x}_B$

S6: *Solve in parallel on host* $D_R\hat{\boldsymbol{x}}_R = \hat{\boldsymbol{x}}_B$

S7: *Evaluate in parallel on host* $\boldsymbol{x}_R = \hat{\mathbf{b}}_R - \hat{\boldsymbol{x}}_R$

Parallel BiCGSTAB

Choose initial approximation $x^{(0)}$ of the solution x_B

$r^{(0)} = b_B - Sx^{(0)}$

Choose $\hat{r}$ (usually $\hat{r} = r^{(0)}$)

for $i = 1, 2, \ldots$

$\rho_{i-1} = \hat{r}^T \, r^{(i-1)}$

if $\rho_{i-1} = 0$ method fails

if $i = 1$

$p^{(1)} = r^{(0)}$

else

$\beta_{i-1} = \frac{\rho_{i-1}}{\rho_{i-2}} \frac{\alpha_{i-1}}{\omega_{i-1}}$

$p^{(i)} = r^{(i-1)} + \beta_{i-1} \left(p^{(i-1)} - \omega_{i-1} \, v^{(i-1)}\right)$

endif

$v^{(i)} = S \, p^{(i)}$

$\alpha_i = \frac{\rho_{i-1}}{\hat{r}^T v^{(i)}}$

$s = r^{(i-1)} - \alpha_i \, v^{(i)}$

if $\| s \|$ is small enough then

$x_B^{(i)} = x_B^{(i-1)} + \alpha_i \, p^{(i)}$ stop

$t = S \, s$

$\omega_i = \frac{s^T t}{t^T t}$

$x_B^{(i)} = x_B^{(i-1)} + \alpha_i \, p^{(i)} + \omega_i \, s$

Check for Convergence

if $\omega_i = 0$ stop

$r^{(i)} = s - \omega_i \, t$

end

Parallel BiCGSTAB

Evaluation of $\boldsymbol{t} = S\boldsymbol{p}$

S1: *Send* $\mathbf{p}$ *from host to GPU*

S2: *Evaluate in parallel on GPU* $\boldsymbol{t} = H_B\,\boldsymbol{p}$

S3: *Send* $\boldsymbol{t}$ *from GPU to host*

S4: *Solve in parallel on host* $D_R\boldsymbol{s} = \boldsymbol{t}$

S5: *Send* $\boldsymbol{s}$ *from host to GPU*

S6: *Evaluate in parallel on GPU* $\boldsymbol{q} = H_R\,\boldsymbol{s}$

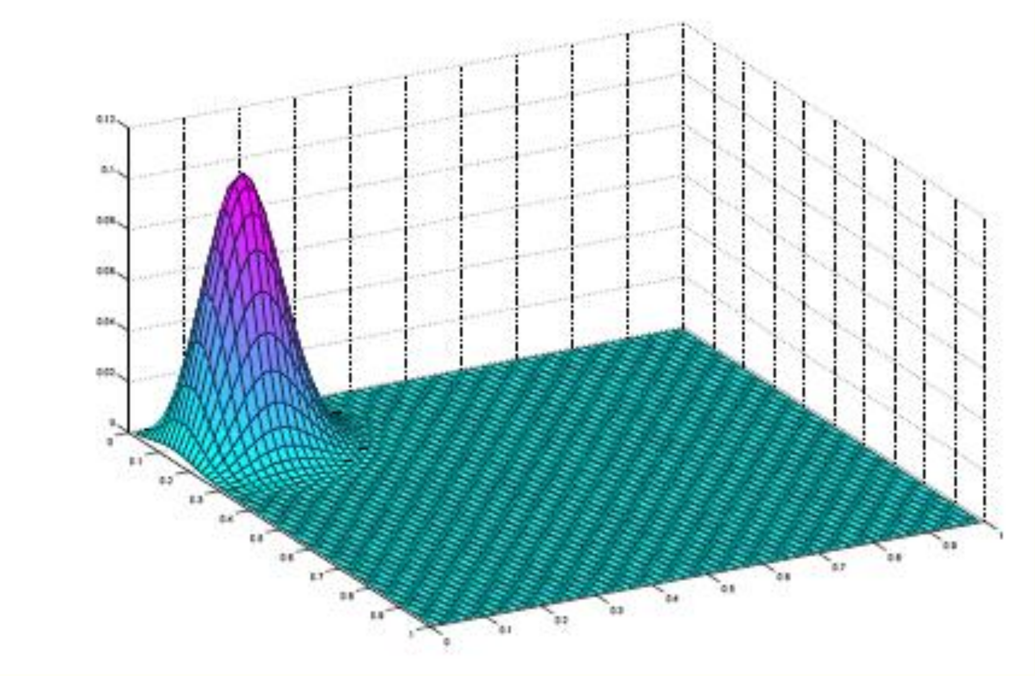
S7: *Send* $\boldsymbol{q}$ *from GPU to host*

S8: *Evaluate in parallel on host* $\boldsymbol{t} = D_B\,\boldsymbol{p} - \boldsymbol{q}$

The Dirichlet Helmholtz Problem

$$u(x,y)=10\varphi(x)\varphi(y) \quad , \quad (x,y)\in[0,1]\times[0,1]$$

with $\varphi(x)=(x^2-x)e^{-100(x-0.1)^2}$



HP SL390s - Tesla M2070 GPUs

HP SL390s

+ 2 x

6 core Xeon@2.8GHz
24GB memory
Oracle Linux 6.3 x64
PGI 13.5 Fortran
PCI-e gen2 x16

TECHNICAL SPECIFICATIONS

	Tesla M2070 / M2075
Peak double precision floating point performance	515 Gigaflops
Peak single precision floating point performance	1030 Gigaflops
CUDA cores	448
Memory size (GDDR5)	6 GigaBytes
Memory bandwidth (ECC off)	150 GBytes/sec

PGI Accelerator Compilers

The Portland Group

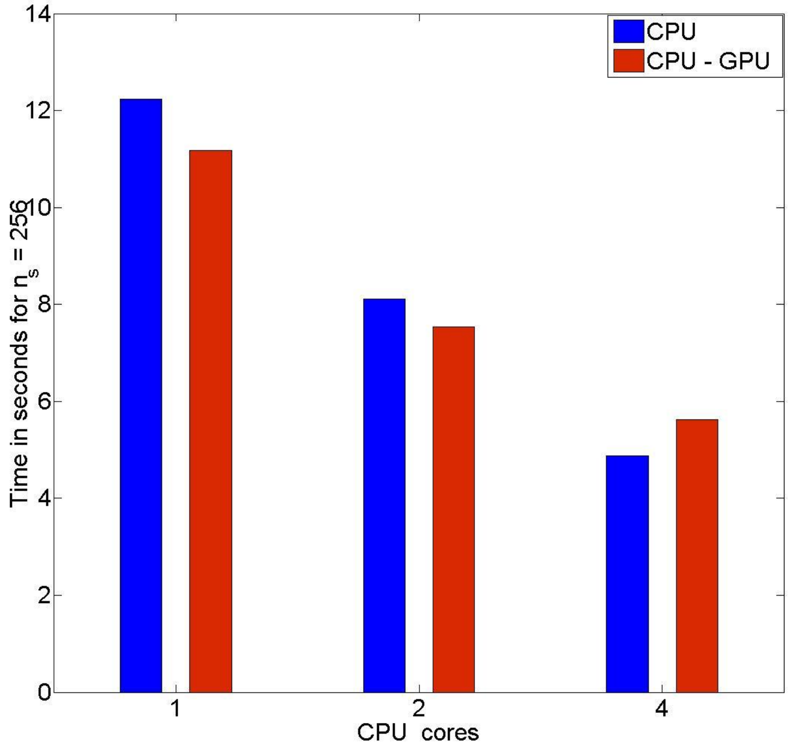
OpenACC
DIRECTIVES FOR ACCELERATORS

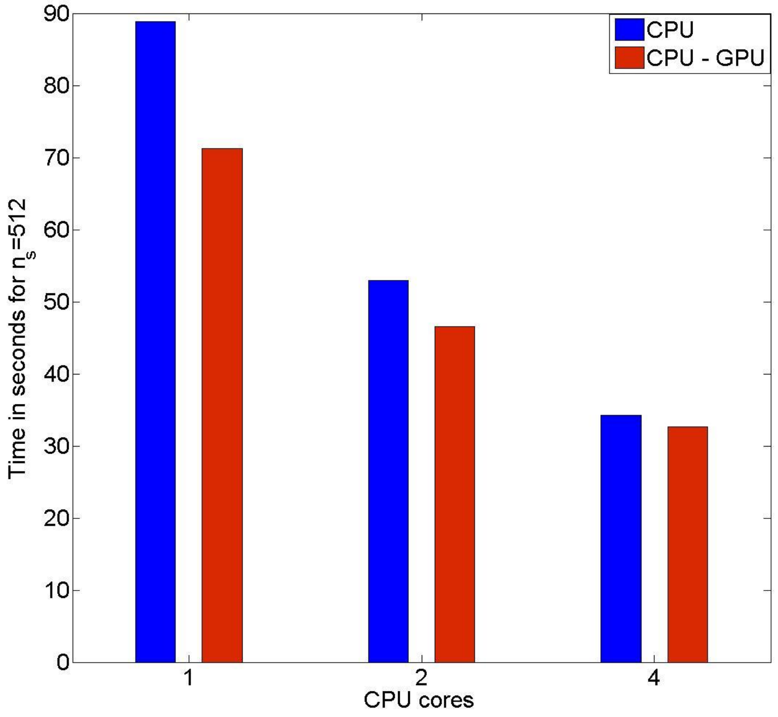
OpenMP

Realization on HP SL390s Tesla GPU machine
Iterations / Error measurements

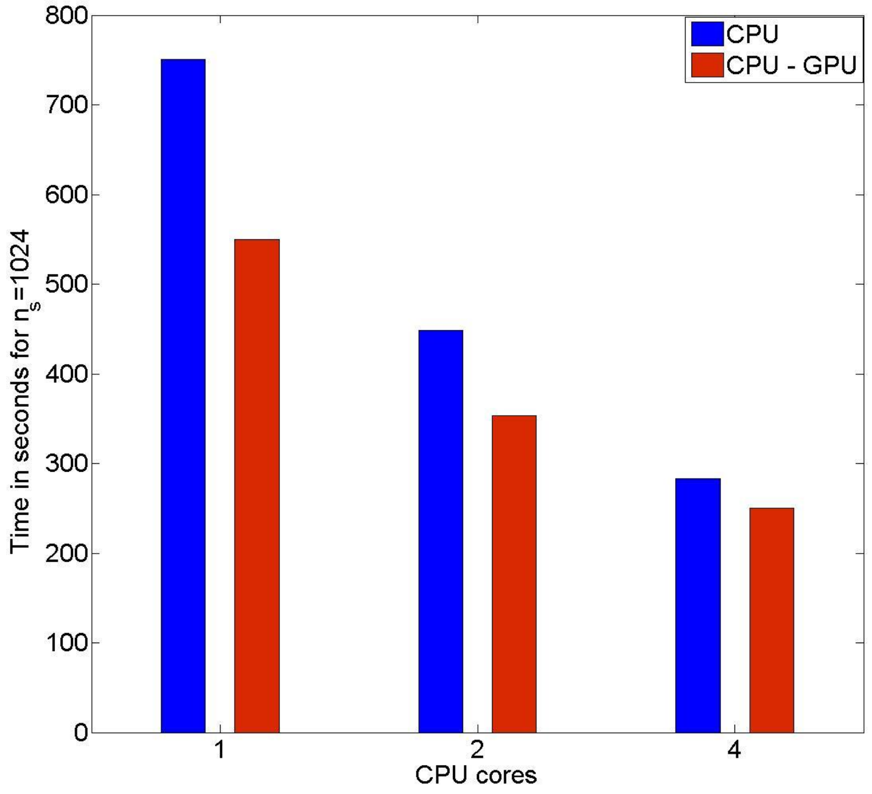
n_s	BiCGSTAB Iterations	$\|\| \mathbf{b} - A\mathbf{x}_n \|\|_2$
256	294	6.06e-11
512	589	2.85e-11
1024	1161	1.39e-11
2048	3726	9.59e-12

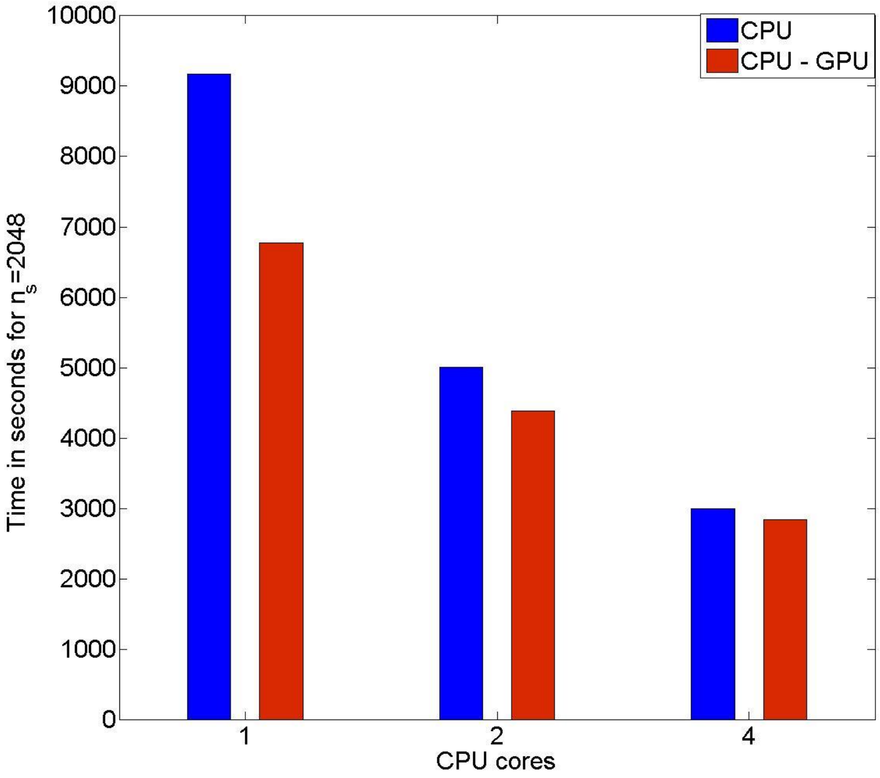
Realization on HP SL390s Tesla GPU machine
Time measurements



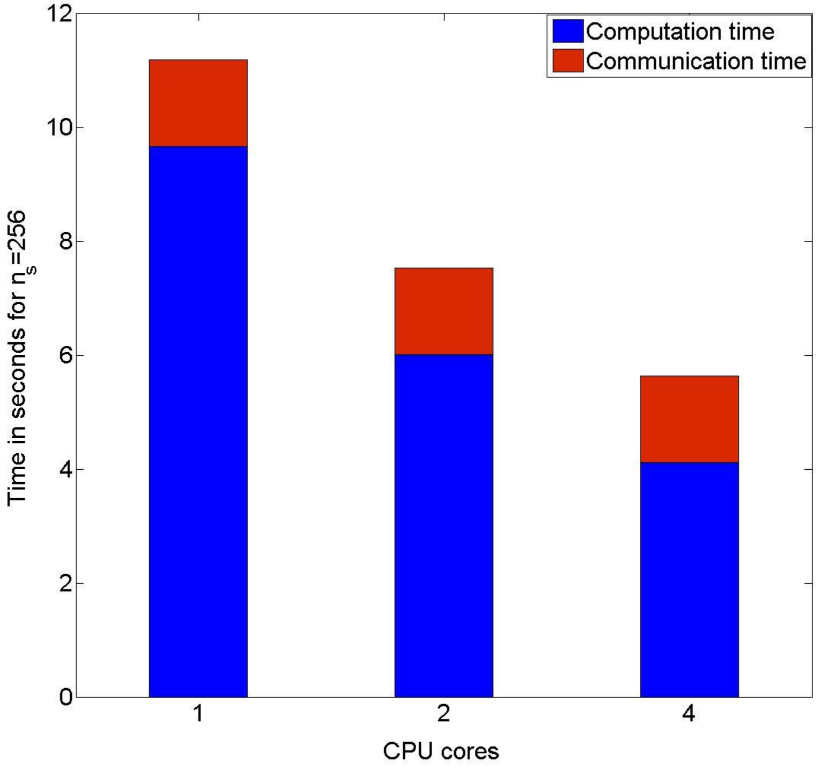


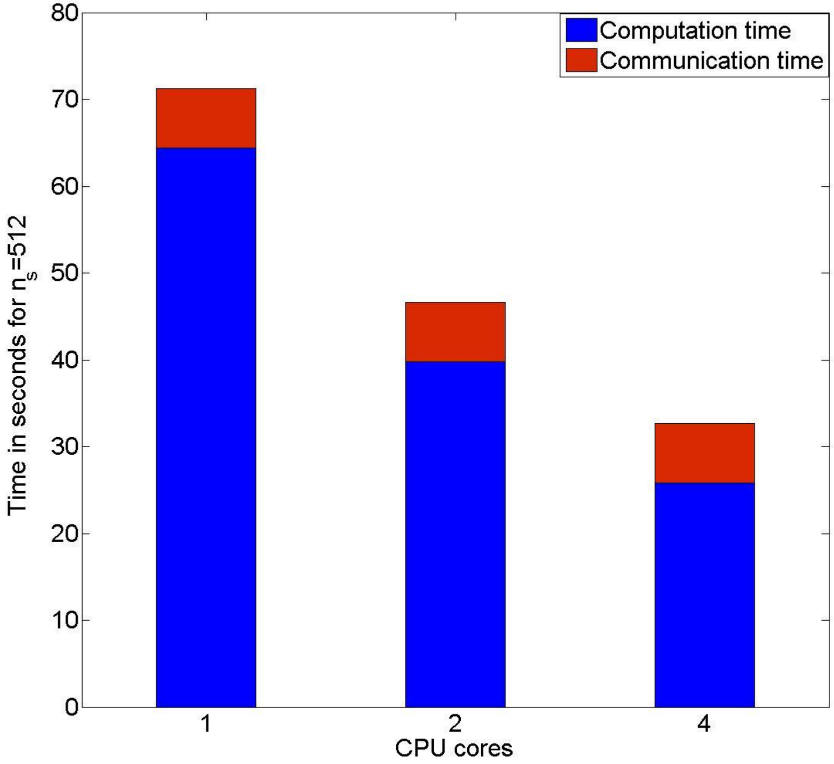
Realization on HP SL390s Tesla GPU machine
Time measurements



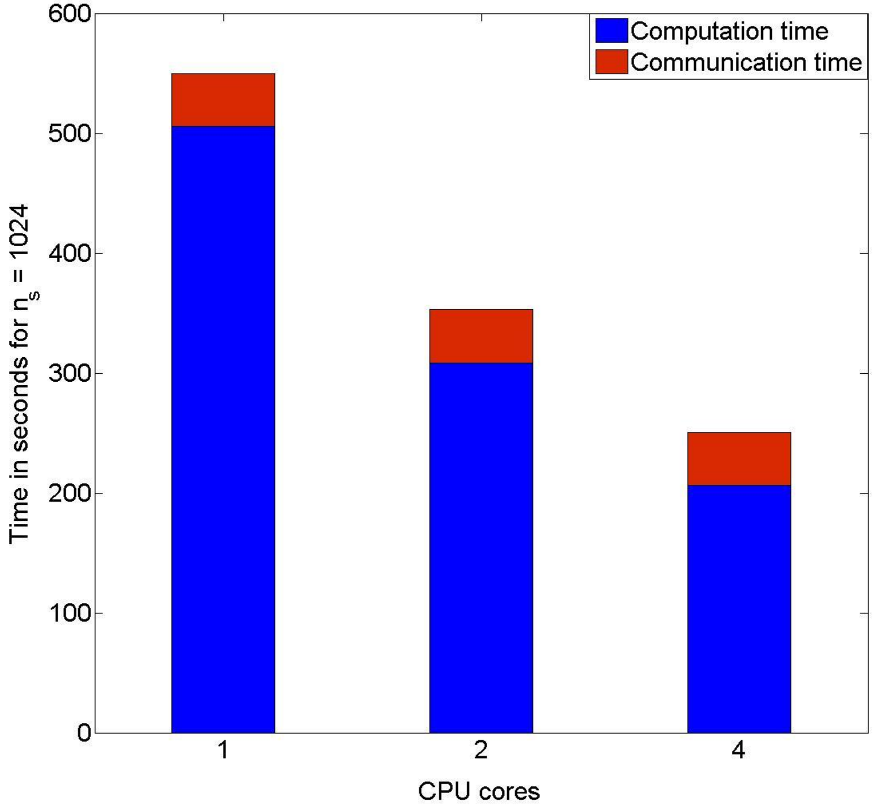


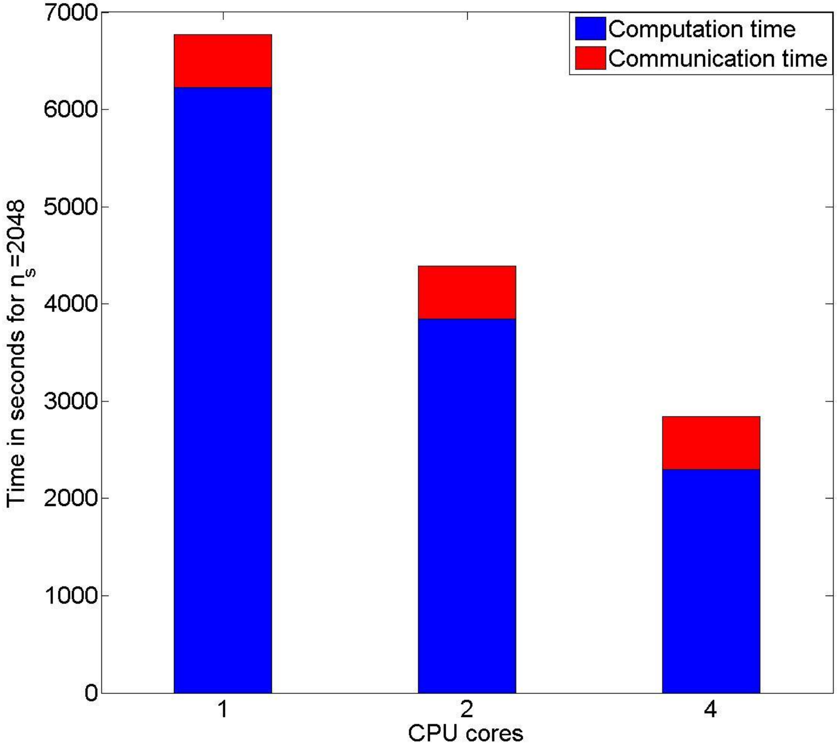
Realization on HP SL390s Tesla GPU machine
Time measurements



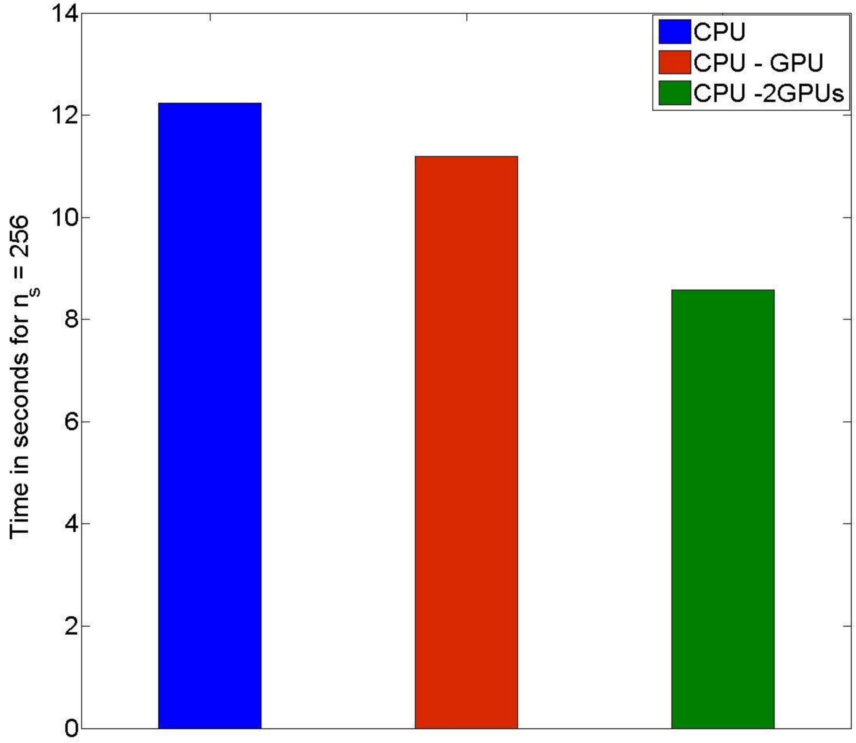


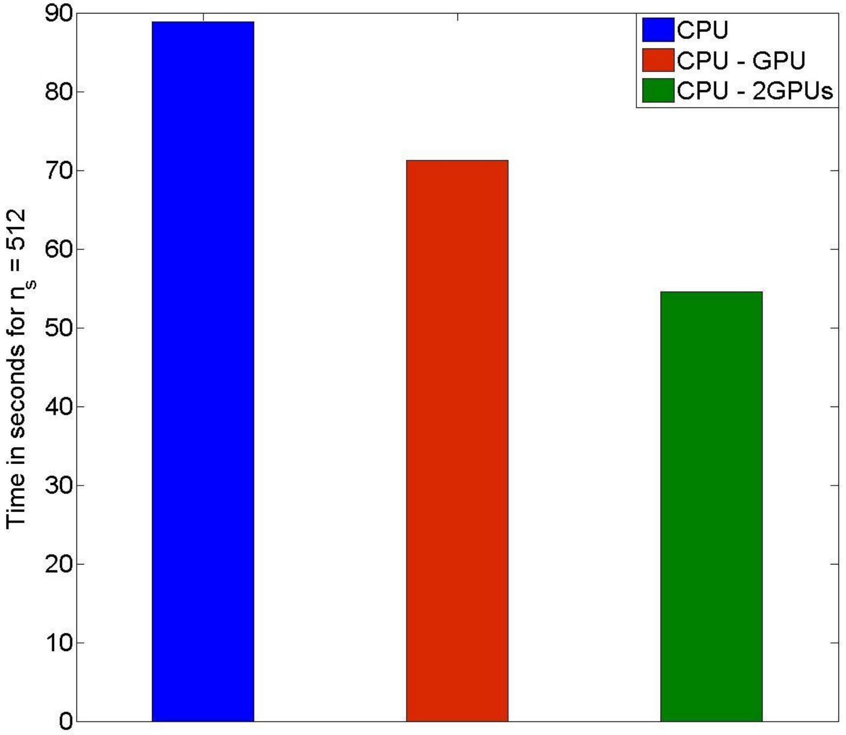
Realization on HP SL390s Tesla GPU machine
Time measurements



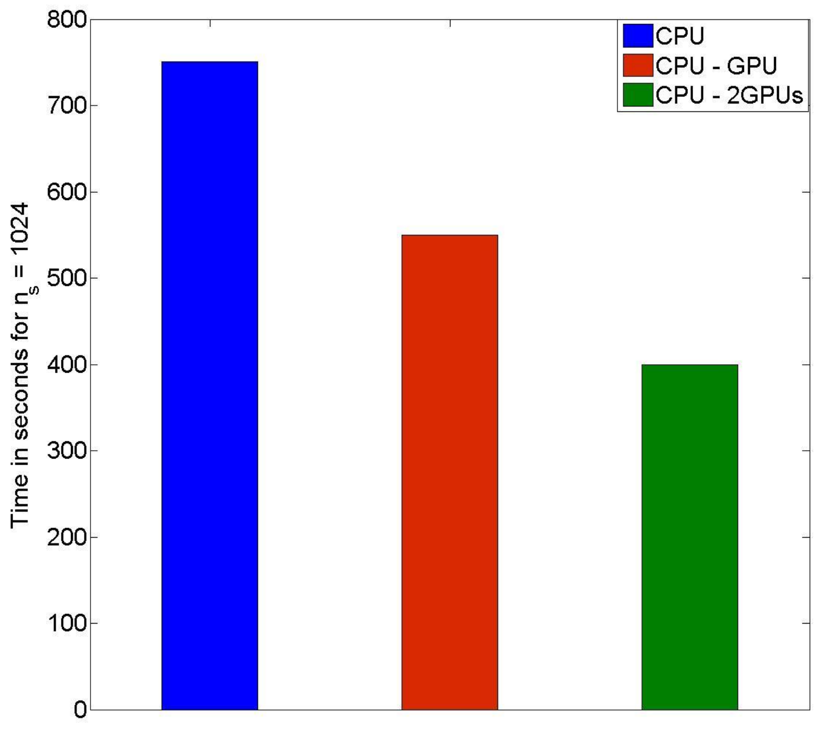


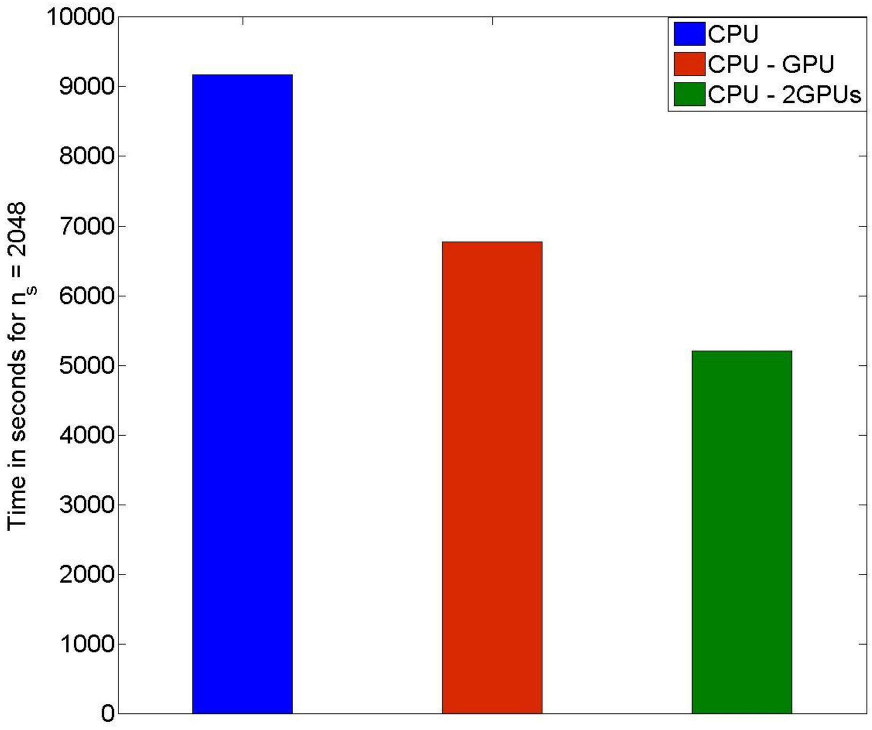
Realization on HP SL390s Tesla GPU machine
Time measurements





Realization on HP SL390s Tesla GPU machine
Time measurements





Conclusions

- A new parallel algorithm implementing the Schur complement with BiCGSTAB iterative method for Hermite Collocation equations has been developed.
- The algorithm is realized on Shared Memory multi-core machines with GPUs .
- A performance acceleration of up to 30% is observed.

Future work

- Design an efficient parallel Schur complement algorithm of the Hermite Collocation equations for Multiprocessor /Grid machines with GPUs.