OPTIMIZATION TECHNIQUES AND META-MODELS IN ENGINEERING

M. J. Colaço

Department of Mechanical and Materials Engineering
Military Institute of Engineering
colaco@ime.eb.br

9/15/2007

Optimization Problems

- **Introduction**
  - Basic Concepts
- **Deterministic Methods**
  - Steepest Descent Method
  - Conjugate Gradient Method
  - Newton-Raphson Method
  - Quasi-Newton Methods
- **Evolutionary and Stochastic Methods**
  - Genetic Algorithms
  - Differential Evolution
  - Particle Swarm
  - Simulated Annealing
- **Hybrid Methods**
  - Examples
- **Meta-models**

Introduction

Inverse Problems → - Tries to find an unknown parameter or function

- Ill-posed

Minimization Problems → - Tries to find the best configuration of a problem

Introduction

Objective Function ➡ $U(\mathbf{x});\ \mathbf{x}=\{x_1,x_2,\ldots,x_N\}$

Equality Constraints ➡ $G(\mathbf{x})=C_1$

Inequality Constraints ➡ $H(\mathbf{x}) \leq C_2$

Deterministic Methods

- ➔ Steepest-Descent Method:

Iterative process:

$$\mathbf{x}^{k+1} = \mathbf{x}^{k} + \alpha^{k}\mathbf{d}^{k}$$

$$\mathbf{d}^{k} = -\nabla U\left(\mathbf{x}^{k}\right)$$

where:

x is the vector of parameters to be optimized

α is the search-step size

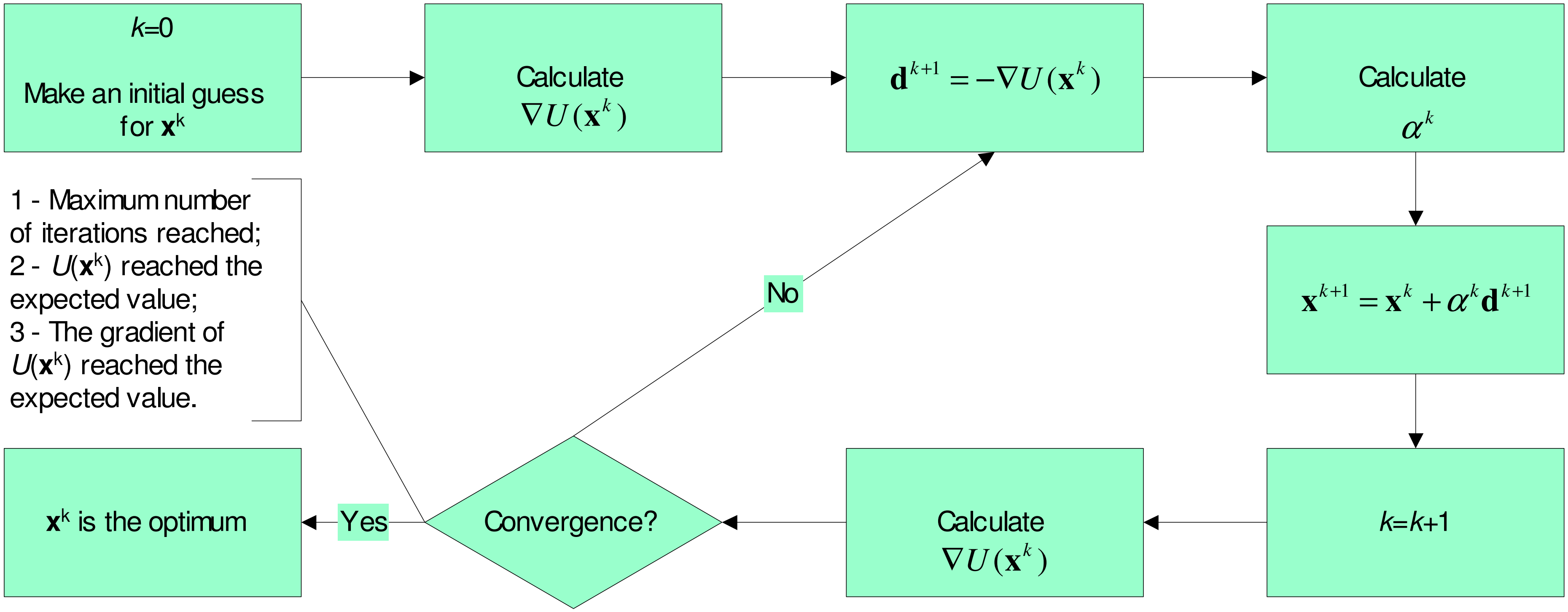
d is the direction of descent

U is the objective function

k is the iteration number

Deterministic Methods

- Steepest-Descent Method:



k=0

Make an initial guess for $\mathbf{x}^k$

Calculate $\nabla U(\mathbf{x}^k)$

$\mathbf{d}^{k+1} = -\nabla U(\mathbf{x}^k)$

Calculate α^k

$\mathbf{x}^{k+1} = \mathbf{x}^k + \alpha^k \mathbf{d}^{k+1}$

k=k+1

Calculate $\nabla U(\mathbf{x}^k)$

Convergence?

No

Yes

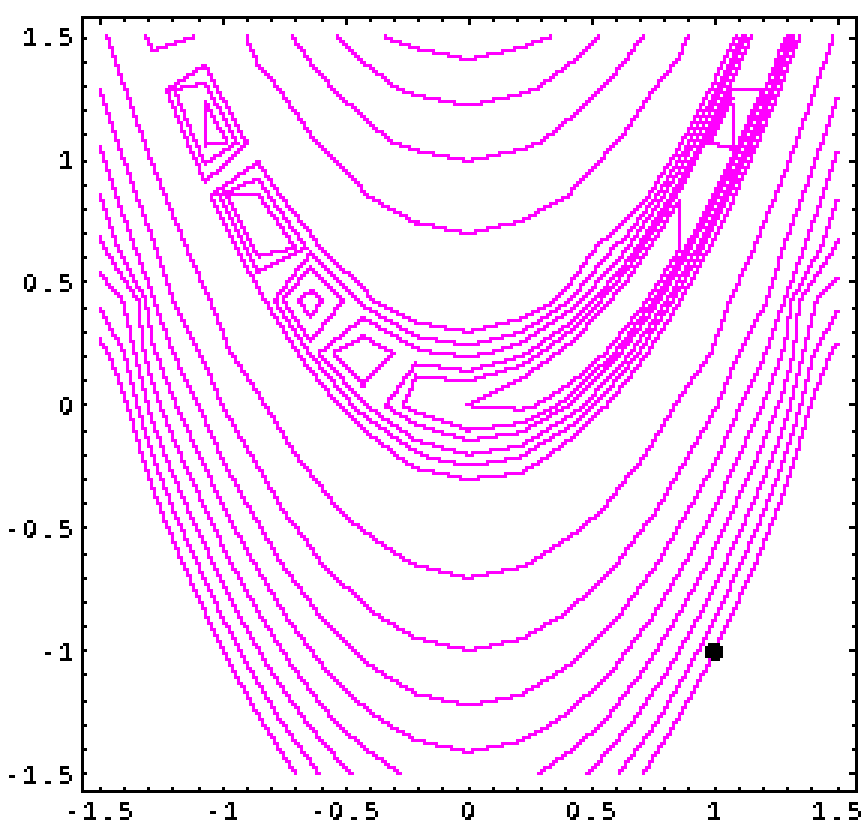
$\mathbf{x}^k$ is the optimum

1 - Maximum number of iterations reached;
2 - $U(\mathbf{x}^k)$ reached the expected value;
3 - The gradient of $U(\mathbf{x}^k)$ reached the expected value.

Deterministic Methods

- Steepest-Descent Method:

SLOW!!!!!!



Deterministic Methods

- Conjugate Gradient Method:

Iterative process:

$$\mathbf{x}^{k+1} = \mathbf{x}^{k} + \alpha^{k}\mathbf{d}^{k}$$

$$\mathbf{d}^{k} = -\nabla U\left(\mathbf{x}^{k}\right) + \gamma^{k}\mathbf{d}^{k-1}$$

where:

$\mathbf{x}$ is the vector of parameters to be optimized

α is the search-step size

$\mathbf{d}$ is the direction of descent

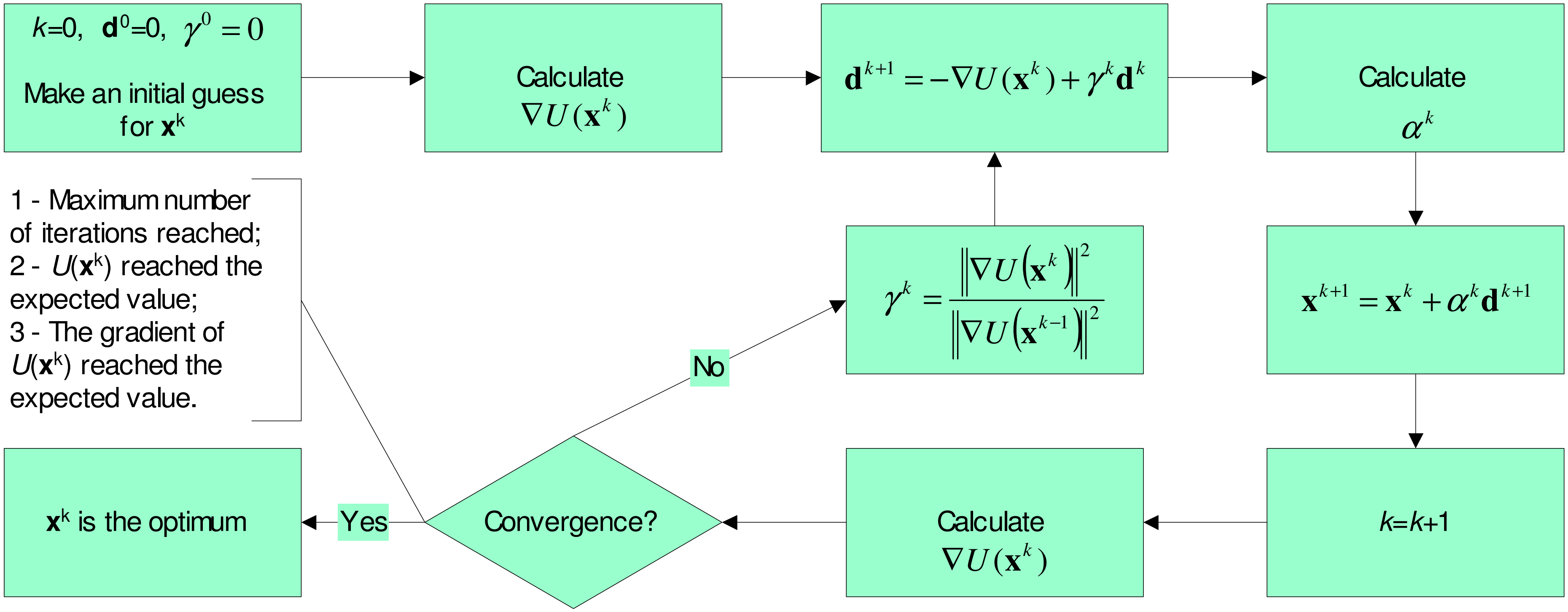
U is the objective function

k is the iteration number

γ is the conjugation coefficient

Deterministic Methods

- ➔ Conjugate Gradient Method:



k=0, $\mathbf{d}^0$=0, $\gamma^0 = 0$

Make an initial guess for $\mathbf{x}^k$

Calculate $\nabla U(\mathbf{x}^k)$

$\mathbf{d}^{k+1} = -\nabla U(\mathbf{x}^k) + \gamma^k \mathbf{d}^k$

Calculate α^k

$\mathbf{x}^{k+1} = \mathbf{x}^k + \alpha^k \mathbf{d}^{k+1}$

k=k+1

Calculate $\nabla U(\mathbf{x}^k)$

Convergence?

No

$$\gamma^k = \frac{\left\|\nabla U\left(\mathbf{x}^k\right)\right\|^2}{\left\|\nabla U\left(\mathbf{x}^{k-1}\right)\right\|^2}$$

Yes

$\mathbf{x}^k$ is the optimum

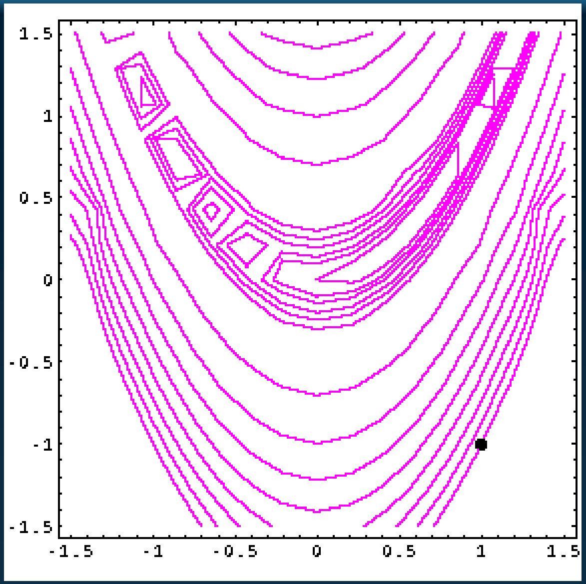
1 - Maximum number of iterations reached;
2 - $U(\mathbf{x}^k)$ reached the expected value;
3 - The gradient of $U(\mathbf{x}^k)$ reached the expected value.

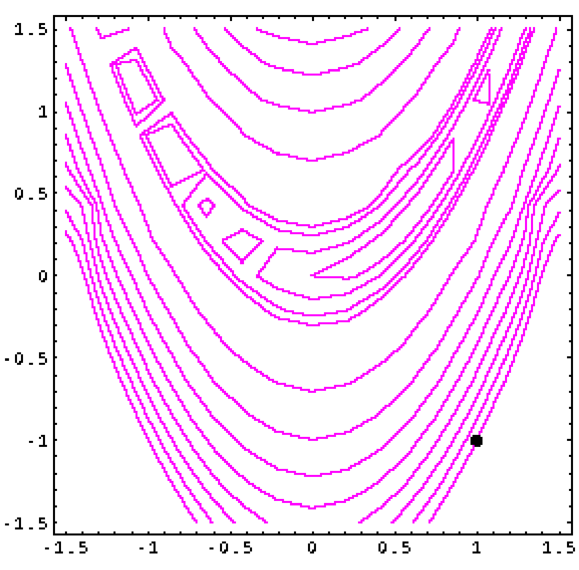
Deterministic Methods

- ➔ Conjugate Gradient Method:

Faster than the

Steepest-Descent!





Deterministic Methods

- ➔ Newton Method:

Iterative process:

$$\mathbf{x}^{k+1} = \mathbf{x}^{k} + \alpha^{k}\mathbf{d}^{k}$$

$$\mathbf{d}^{k} = -\mathbf{H}^{k}\nabla U\left(\mathbf{x}^{k}\right)$$

where:

x is the vector of parameters to be optimized

α is the search-step size

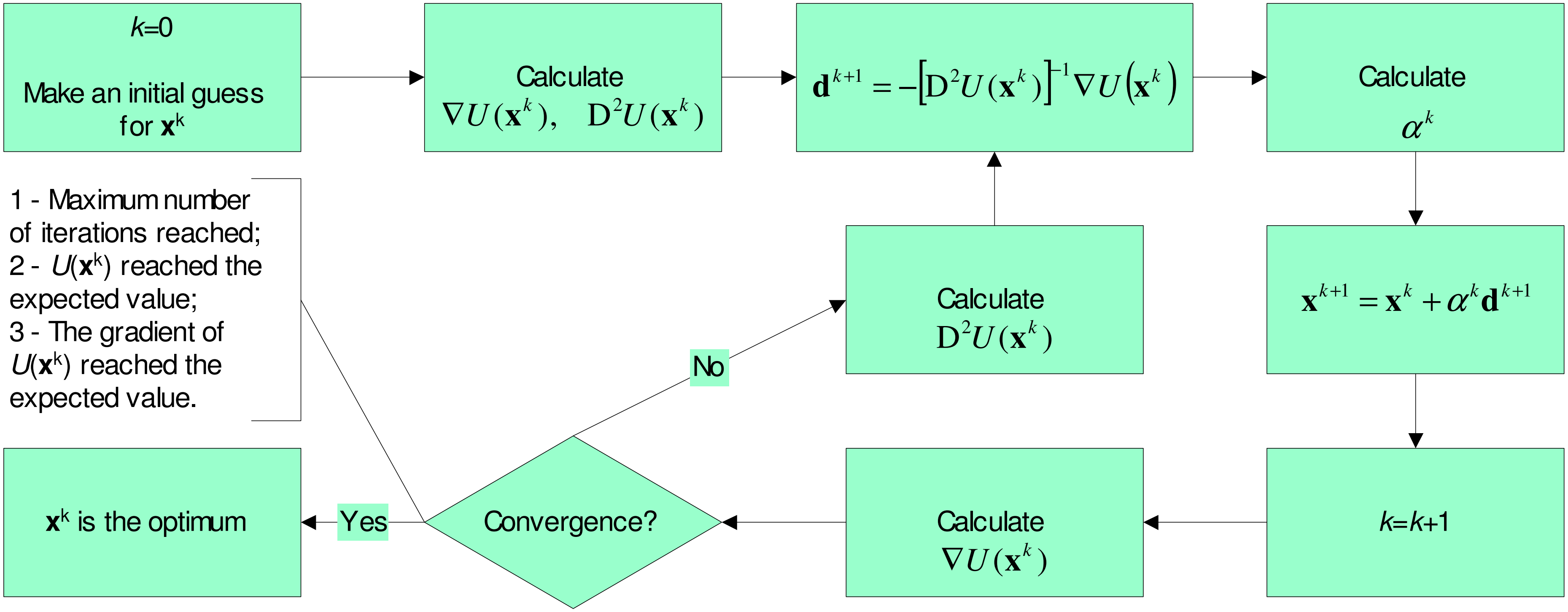
H is the matrix of 2nd order derivatives – Expensive in terms of computational cost!

U is the objective function

k is the iteration number

Deterministic Methods

- Newton Method:



k=0

Make an initial guess for $\mathbf{x}^k$

Calculate $\nabla U(\mathbf{x}^k), \quad \mathrm{D}^2 U(\mathbf{x}^k)$

$$\mathbf{d}^{k+1} = -\left[\mathrm{D}^2 U(\mathbf{x}^k)\right]^{-1} \nabla U\left(\mathbf{x}^k\right)$$

Calculate α^k

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \alpha^k \mathbf{d}^{k+1}$$

k=k+1

Calculate $\nabla U(\mathbf{x}^k)$

Convergence?

Yes

$\mathbf{x}^k$ is the optimum

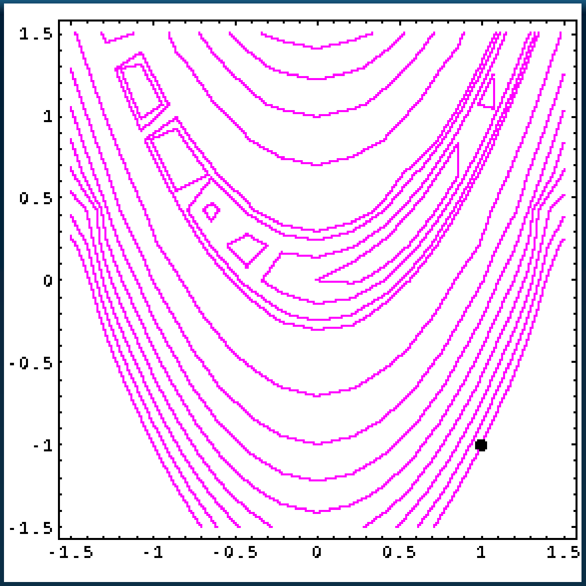
No

Calculate $\mathrm{D}^2 U(\mathbf{x}^k)$

1 - Maximum number of iterations reached;
2 - $U(\mathbf{x}^k)$ reached the expected value;
3 - The gradient of $U(\mathbf{x}^k)$ reached the expected value.

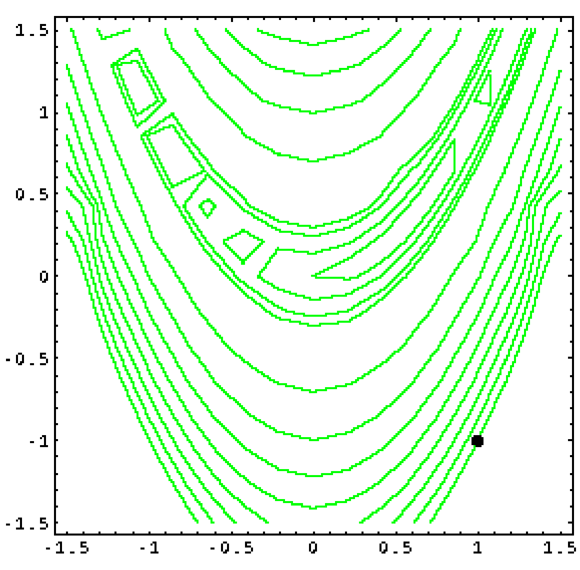
Deterministic Methods

- ➔ Newton Method:



Faster than the

Conjugate Gradient!



Deterministic Methods

- BFGS (Broyden-Fletcher-Goldfarb-Shanno) Method:
  - Quasi-Newton method, similar to the DFP method, but less dependent on the search-step size choice.
  - Uses an iterative approximation for the Hessian

$$\mathbf{H}^{k} = \mathbf{H}^{k-1} + \mathbf{M}^{k-1} + \mathbf{N}^{k-1}$$

where

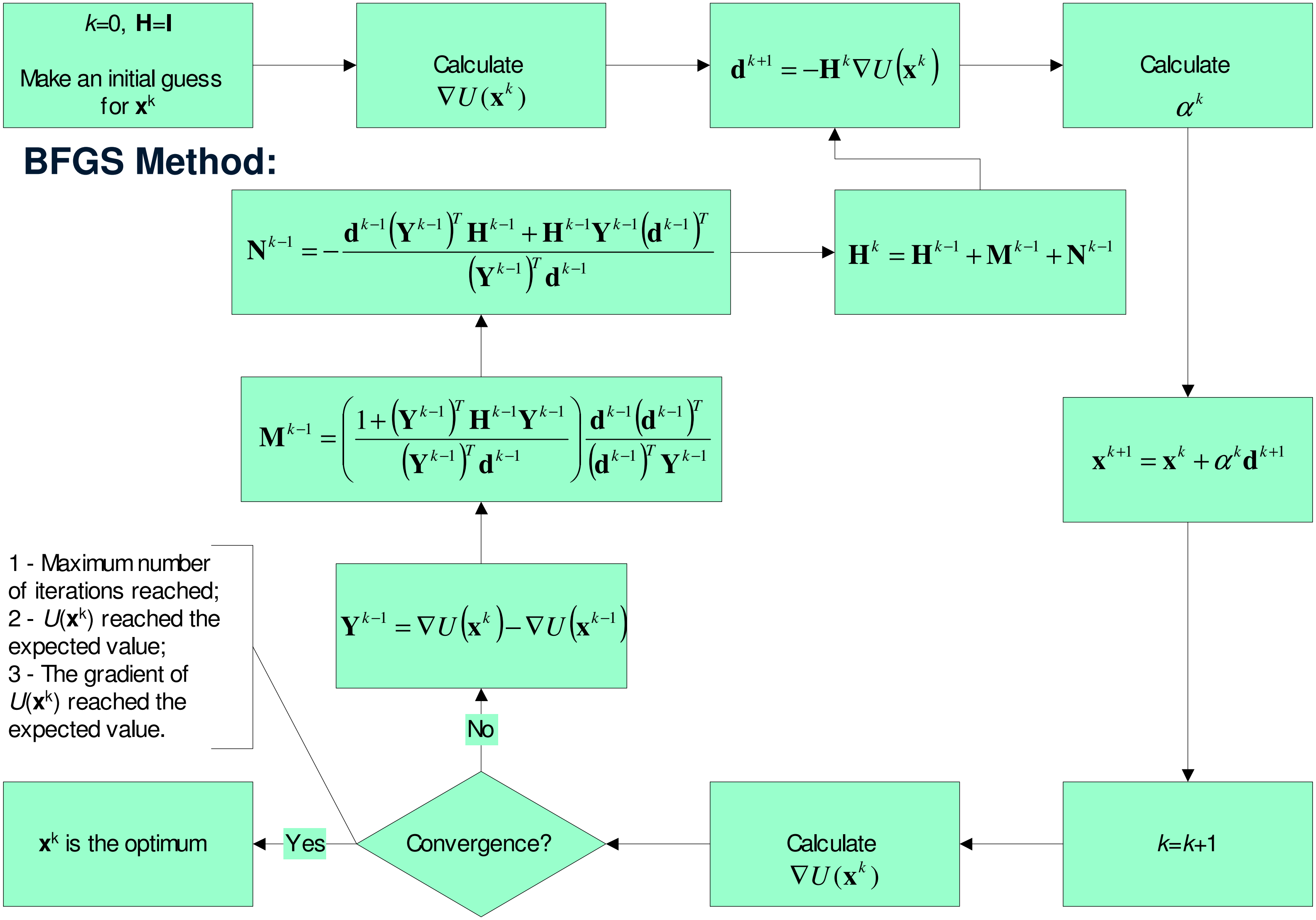
$$\mathbf{M}^{k-1} = \left(\frac{1 + \left(\mathbf{Y}^{k-1}\right)^{T} \mathbf{H}^{k-1} \mathbf{Y}^{k-1}}{\left(\mathbf{Y}^{k-1}\right)^{T} \mathbf{d}^{k-1}} \right) \frac{\mathbf{d}^{k-1} \left(\mathbf{d}^{k-1}\right)^{T}}{\left(\mathbf{d}^{k-1}\right)^{T} \mathbf{Y}^{k-1}}$$

$$\mathbf{N}^{k-1} = -\frac{\mathbf{d}^{k-1} \left(\mathbf{Y}^{k-1}\right)^{T} \mathbf{H}^{k-1} + \mathbf{H}^{k-1} \mathbf{Y}^{k-1} \left(\mathbf{d}^{k-1}\right)^{T}}{\left(\mathbf{Y}^{k-1}\right)^{T} \mathbf{d}^{k-1}}$$

$$\mathbf{Y}^{k-1} = \nabla U\left(\mathbf{x}^{k}\right) - \nabla U\left(\mathbf{x}^{k-1}\right)$$

Deterministic Methods

➔ **BFGS Method:**



k=0, **H**=**I**

Make an initial guess for $\mathbf{x}^k$

Calculate $\nabla U(\mathbf{x}^k)$

$$\mathbf{d}^{k+1} = -\mathbf{H}^k \nabla U\left(\mathbf{x}^k\right)$$

Calculate α^k

$$\mathbf{N}^{k-1} = -\frac{\mathbf{d}^{k-1}\left(\mathbf{Y}^{k-1}\right)^T \mathbf{H}^{k-1} + \mathbf{H}^{k-1}\mathbf{Y}^{k-1}\left(\mathbf{d}^{k-1}\right)^T}{\left(\mathbf{Y}^{k-1}\right)^T \mathbf{d}^{k-1}}$$

$$\mathbf{H}^k = \mathbf{H}^{k-1} + \mathbf{M}^{k-1} + \mathbf{N}^{k-1}$$

$$\mathbf{M}^{k-1} = \left(\frac{1 + \left(\mathbf{Y}^{k-1}\right)^T \mathbf{H}^{k-1}\mathbf{Y}^{k-1}}{\left(\mathbf{Y}^{k-1}\right)^T \mathbf{d}^{k-1}}\right)\frac{\mathbf{d}^{k-1}\left(\mathbf{d}^{k-1}\right)^T}{\left(\mathbf{d}^{k-1}\right)^T \mathbf{Y}^{k-1}}$$

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \alpha^k \mathbf{d}^{k+1}$$

1 - Maximum number of iterations reached;
2 - *U*($\mathbf{x}^k$) reached the expected value;
3 - The gradient of *U*($\mathbf{x}^k$) reached the expected value.

$$\mathbf{Y}^{k-1} = \nabla U\left(\mathbf{x}^k\right) - \nabla U\left(\mathbf{x}^{k-1}\right)$$

No

$\mathbf{x}^k$ is the optimum

Yes

Convergence?

Calculate $\nabla U(\mathbf{x}^k)$

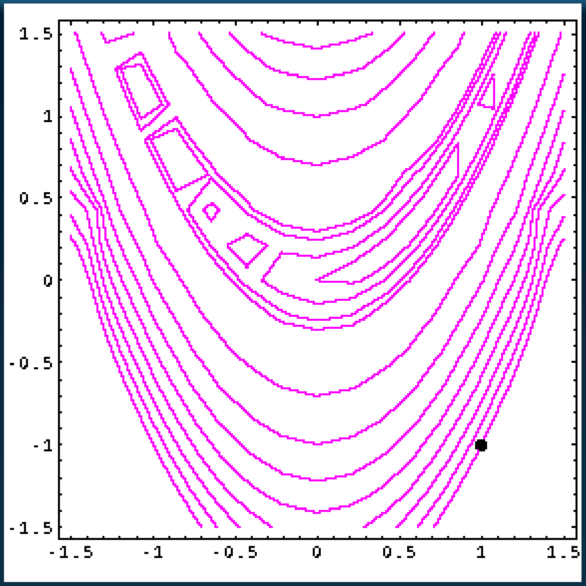
k=*k*+1

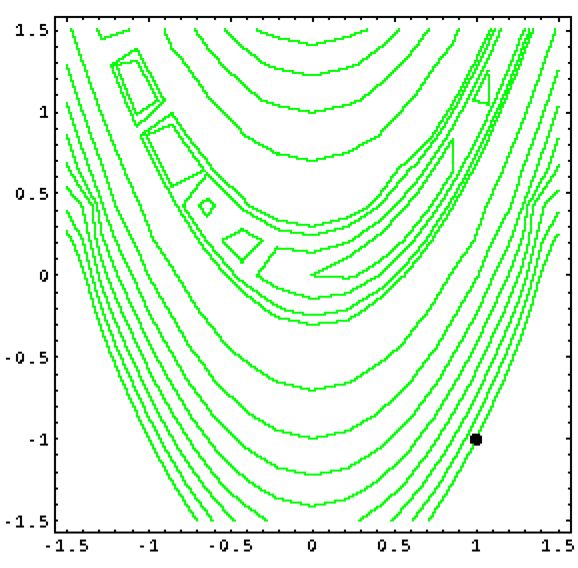
Deterministic Methods

- ➔ BFGS Method:

Faster than the

Conjugate Gradient!





Evolutionary and Stochastic Methods

- ➔ GA (Genetic Algorithm) and DE (Differential Evolution) Methods:
- ➔ Based on Darwin’s model for natural selection of species.
  - Members of a certain population matches and have children. Those children are a combination of the parents’ chromosomes.
  - The strongest members of the population will have more chances to survive under a certain environment.
  - The combination of the chromosomes is called crossover.
  - Mutations can occur. They can be good or bad mutations.

Evolutionary and Stochastic Methods

- DE Method:
  - Alternative to the Genetic Algorithm method.
  - Proposed in 1995 by Kenneth Price and Rainer Storn from Berkeley.
- The method initializes with a random generated random matrix **P** which contains N vector parameters **x**
- From the initial population matrix, generations are created until the best generation (optimum) is found.

Evolutionary and Stochastic Methods

- DE Method:
- The next generation is created as:

$$\mathbf{x}_i^{k+1} = \delta_1 \mathbf{x}_i^k + \delta_2 \left[\boldsymbol{\alpha} + F \left(\boldsymbol{\beta} - \boldsymbol{\gamma} \right) \right]$$

1st parent

2nd parent
Mutation included

where

α, β and γ are three randomly chosen members of the population matrix **P**.

F is a weighting function which defines the mutation ($0.5 < F < 1$).

k is the generation counter.

δ_1 and δ_2 are delta Dirac functions that defines the crossover.

If $U(\mathbf{x}^{k+1}) < U(\mathbf{x}^k)$ → $\mathbf{x}^{k+1}$ replaces $\mathbf{x}^k$ in the population matrix **P**

If $U(\mathbf{x}^{k+1}) > U(\mathbf{x}^k)$ → $\mathbf{x}^k$ is kept in the population matrix **P** and $\mathbf{x}^{k+1}$ is discarded

Evolutionary and Stochastic Methods

- DE Method:
- The crossover is obtained as:

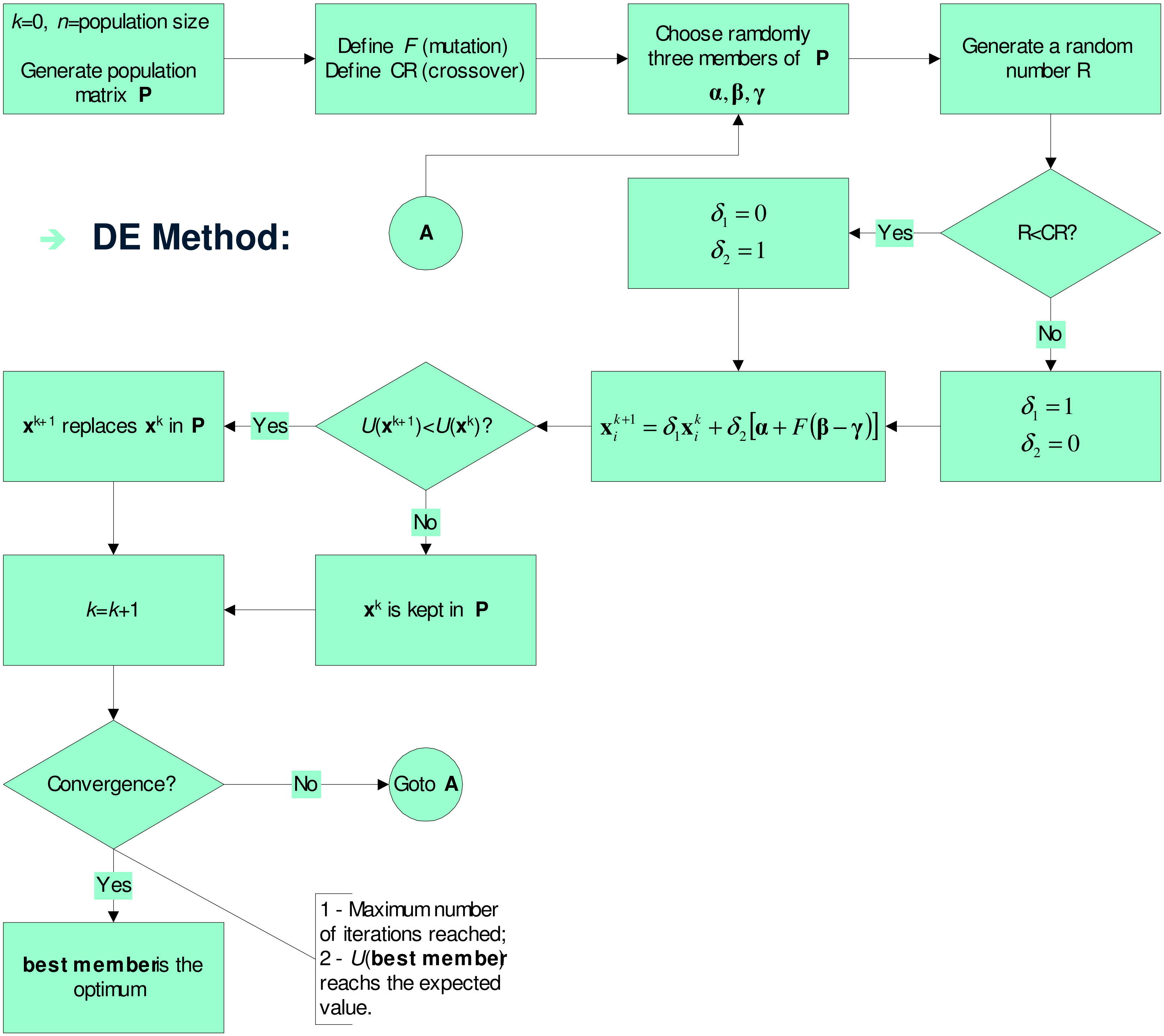
$$\mathbf{x}_i^{k+1} = \delta_1 \mathbf{x}_i^{k} + \delta_2 \left[\boldsymbol{\alpha} + F\left(\boldsymbol{\beta} - \boldsymbol{\gamma} \right) \right]$$

$\delta_1 =$ ➡
0, if R < CR
1, if R > CR

$\delta_2 =$ ➡
1, if R < CR
0, if R > CR

- R is a random number with uniform distribution between 0 and 1
- CR is the crossover factor (0.5 < CR < 1)

DE Method:



k=0, *n*=population size

Generate population matrix **P**

Define *F* (mutation)
Define CR (crossover)

Choose ramdomly three members of **P**

$\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}$

Generate a random number R

R<CR?

Yes

$\delta_1 = 0$
$\delta_2 = 1$

No

$\delta_1 = 1$
$\delta_2 = 0$

A

$\mathbf{x}_i^{k+1} = \delta_1 \mathbf{x}_i^k + \delta_2 [\boldsymbol{\alpha} + F(\boldsymbol{\beta} - \boldsymbol{\gamma})]$

$U(\mathbf{x}^{k+1}) < U(\mathbf{x}^k)$?

Yes

$\mathbf{x}^{k+1}$ replaces $\mathbf{x}^k$ in **P**

No

$\mathbf{x}^k$ is kept in **P**

k=*k*+1

Convergence?

No

Goto **A**

Yes

best member is the optimum

1 - Maximum number of iterations reached;
2 - *U*(**best member**) reaches the expected value.

Evolutionary and Stochastic Methods

- ➔ SA (Simulated Annealing) Method:
- ➔ Based on thermodynamics and solidification process of liquids and metals.

Slow cooling ➔ Pure crystal is formed with minimum energy state.

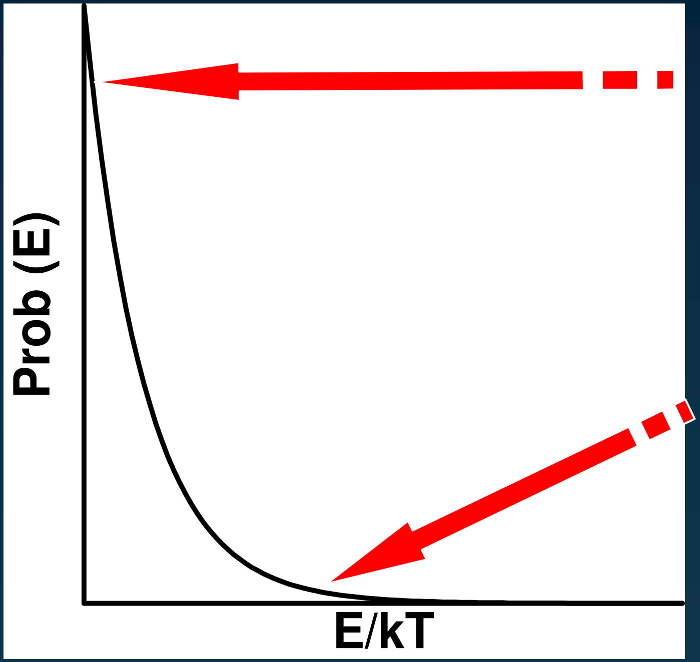
Fast cooling "quenched" ➔ Polycrystalline or amorphous state is formed with higher energy.

- ➔ Gradient methods ➔ "Fast cooling". They can lead to a local minima.

Evolutionary and Stochastic Methods

- SA Method:
- Boltzmann probability distribution:

$$\mathrm{Prob}(E) \approx \exp\left(-\frac{E}{kT}\right)$$



High temperature

High Probability of high energy state

Low temperature

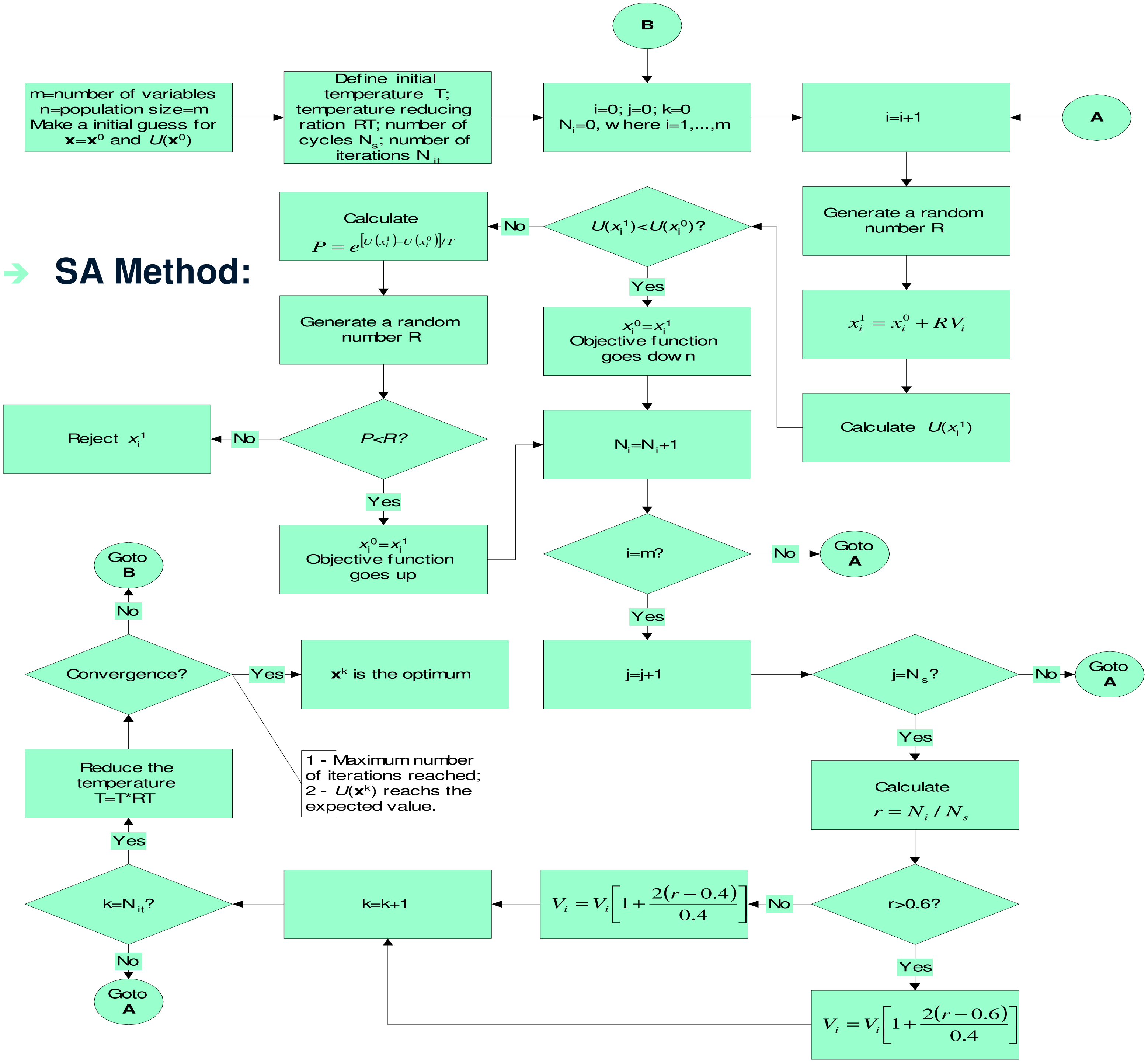
Small Probability of high energy state

- The method can move uphill as well as downhill depending on the probability of high energy states.

Evolutionary and Stochastic Methods

- ➔ SA Method:
- ➔ Iterative process is presented in the following papers:
  - Corana, A., Marchesi, M., Martini, C. e Ridella, S., “Minimizing Multimodal Functions of Continuous Variables with the ‘Simulated Annealing Algorithm’”, *ACM Transactions on Mathematical Software*, vol. 13, pp. 262-280, 1987.
  - Goffe, W. L., Ferrier, G. D. e Rogers, J., “Global Optimization of Statistical Functions with Simulated Annealing”, *Journal of Econometrics*, vol. 60, pp. 65-99, 1994.
- ➔ Excessive number of objective function evaluations!!!

➔ SA Method:



m=number of variables
n=population size=m
Make a initial guess for
$\mathbf{x}=\mathbf{x}^0$ and $U(\mathbf{x}^0)$

Define initial temperature T;
temperature reducing ration RT; number of cycles N_s; number of iterations N_{it}

B

i=0; j=0; k=0
$N_i=0$, where i=1,...,m

A

i=i+1

Generate a random number R

$x_i^1 = x_i^0 + R V_i$

Calculate $U(x_i^1)$

$U(x_i^1)<U(x_i^0)$?

Yes

$x_i^0=x_i^1$
Objective function goes down

No

Calculate
$P = e^{[U(x_i^1)-U(x_i^0)]/T}$

Generate a random number R

$P<R$?

No

Reject x_i^1

Yes

$x_i^0=x_i^1$
Objective function goes up

$N_i=N_i+1$

i=m?

No

Goto A

Yes

j=j+1

$j=N_s$?

No

Goto A

Yes

Calculate
$r = N_i / N_s$

r>0.6?

Yes

$V_i = V_i\left[1+\frac{2(r-0.6)}{0.4}\right]$

No

$V_i = V_i\left[1+\frac{2(r-0.4)}{0.4}\right]$

k=k+1

$k=N_{it}$?

No

Goto A

Yes

Reduce the temperature
T=T*RT

Convergence?

No

Goto B

Yes

$\mathbf{x}^k$ is the optimum

1 - Maximum number of iterations reached;
2 - $U(\mathbf{x}^k)$ reaches the expected value.

Evolutionary and Stochastic Methods

➔ PS (Particle Swarm) method:

- Created in 1995 by an Electric Engineer (Russel Eberhart) and a Social-Psychologist (James Kennedy) as an alternative to Genetic Algorithm.
- Based on the social behavior of various species (including humans).
- Balances the individuality and sociability of individuals in order to find a optimum.

⇧ Individuality ⇧ Chances to find alternatives places

⇩ Convergence

⇧ Sociability ⇧ Learning process among the individuals

⇩ Chances to find alternatives places. Individuals can find a local minima

Evolutionary and Stochastic Methods

- PS method:
- Update process

$$\mathbf{x}_i^{k+1} = \mathbf{x}_i^{k} + \mathbf{v}_i^{k+1}$$

$$\mathbf{v}_i^{k+1} = \alpha \mathbf{v}_i^{k} + \underbrace{\beta \mathbf{r}_{1i}\left(\mathbf{p}_i - \mathbf{x}_i^{k}\right)}_{\text{Individuality}} + \underbrace{\beta \mathbf{r}_{2i}\left(\mathbf{p}_g - \mathbf{x}_i^{k}\right)}_{\text{Sociability}}$$

where

$\mathbf{x}_i$ is i-th individual of the vector of parameters

$\mathbf{r}_{1i}$ and $\mathbf{r}_{2i}$ are are random numbers with uniform distribution between 0 and 1

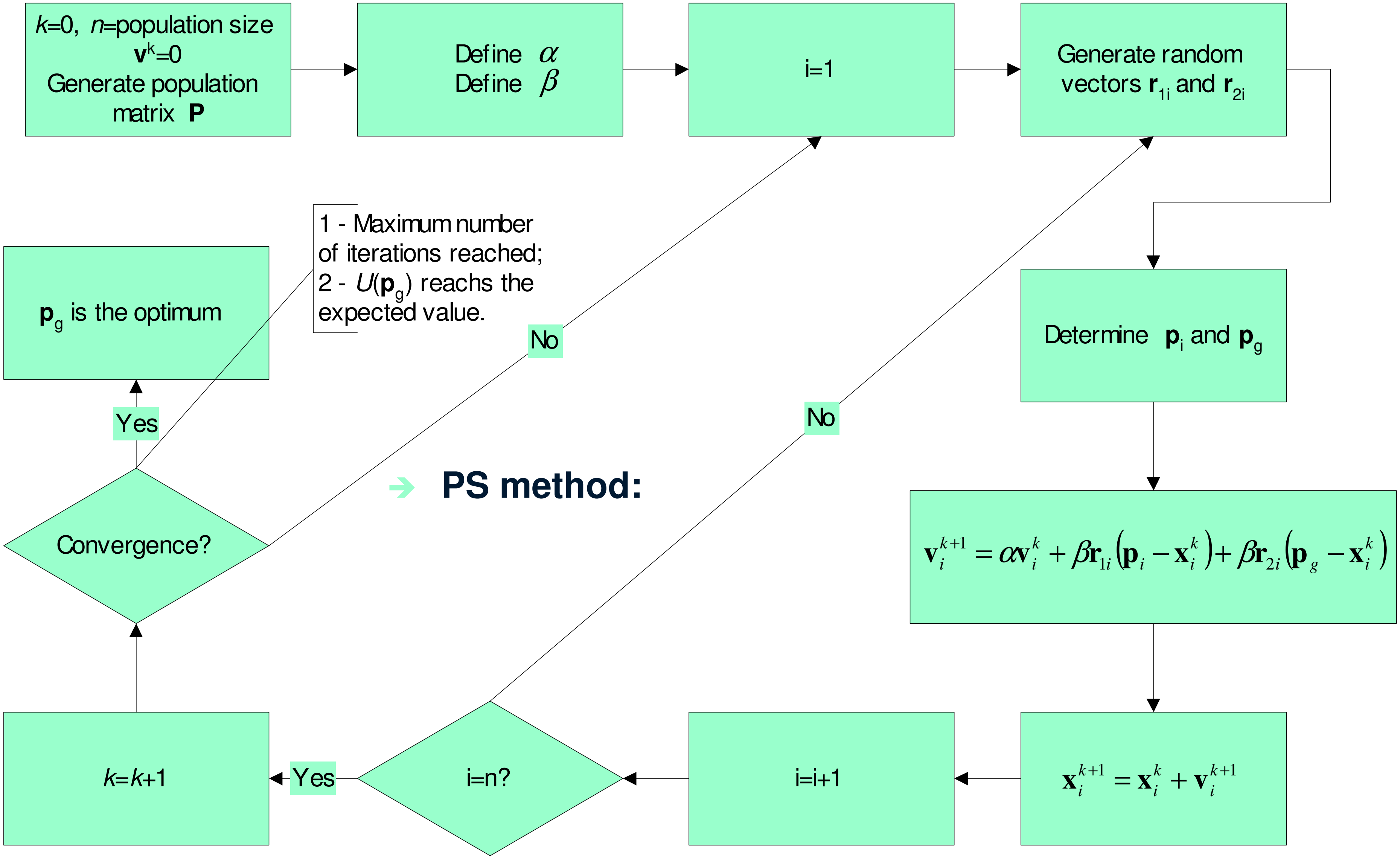
$\mathbf{p}_i$ is the best value found for the vector $\mathbf{x}_i$

$\mathbf{p}_g$ is the vest value found for the entire population

$0 < \alpha < 1; \; 1 < \beta < 2$

Evolutionary and Stochastic Methods

➔ **PS method:**



k=0, *n*=population size
$\mathbf{v}^k=0$
Generate population matrix **P**

Define α
Define β

i=1

Generate random vectors $\mathbf{r}_{1i}$ and $\mathbf{r}_{2i}$

Determine $\mathbf{p}_i$ and $\mathbf{p}_g$

$$\mathbf{v}_i^{k+1} = \alpha \mathbf{v}_i^k + \beta \mathbf{r}_{1i}\left(\mathbf{p}_i - \mathbf{x}_i^k\right) + \beta \mathbf{r}_{2i}\left(\mathbf{p}_g - \mathbf{x}_i^k\right)$$

$$\mathbf{x}_i^{k+1} = \mathbf{x}_i^k + \mathbf{v}_i^{k+1}$$

i=i+1

i=n?

No

Yes

k=*k*+1

Convergence?

No

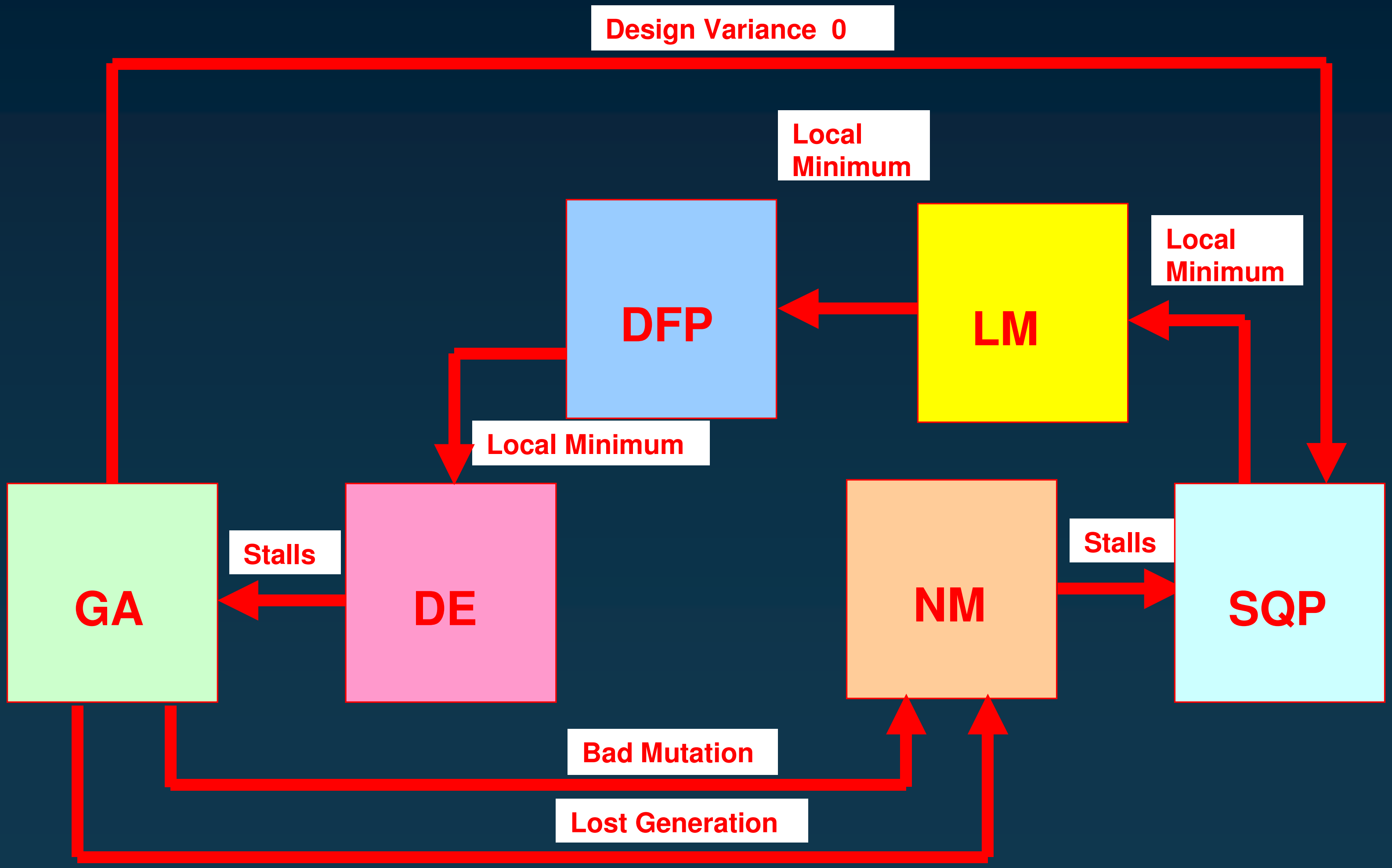
Yes

$\mathbf{p}_g$ is the optimum

1 - Maximum number of iterations reached;
2 - $U(\mathbf{p}_g)$ reachs the expected value.

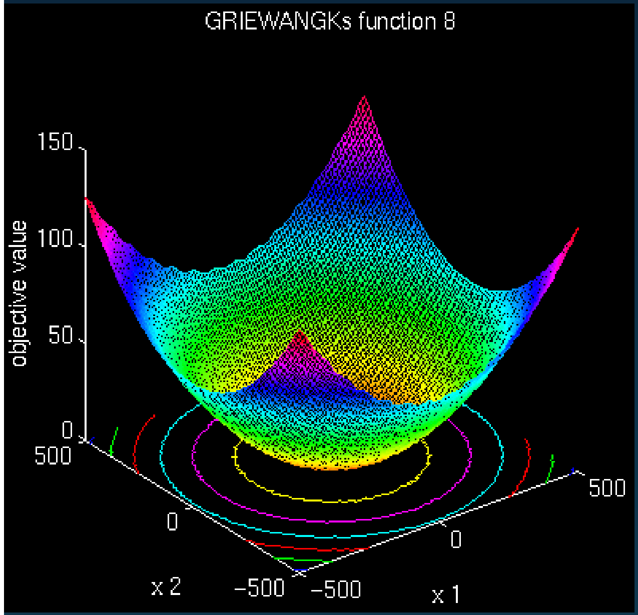
Hybrid Methods

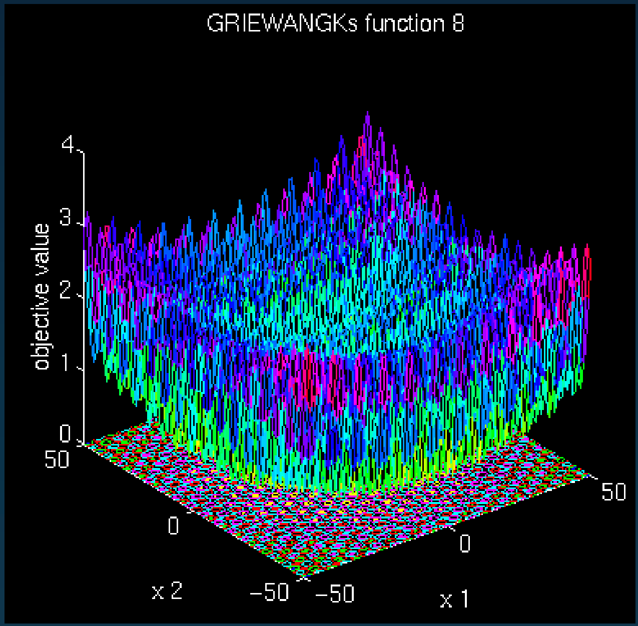
- Hybrid optimizer – version 1 (Martin, Colaço and Dulikravich)

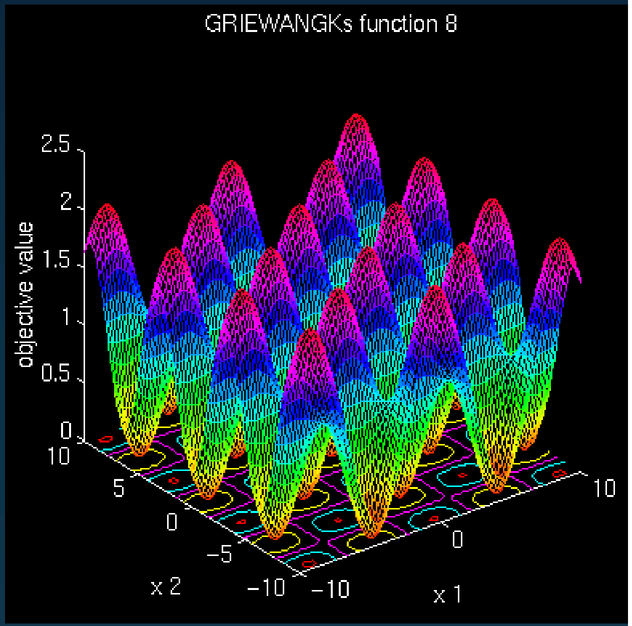


Example 1 - Griewangk's function

- ➔ Multiple local minima

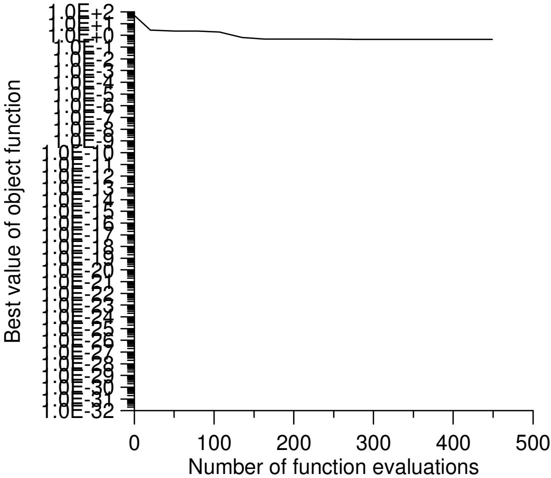


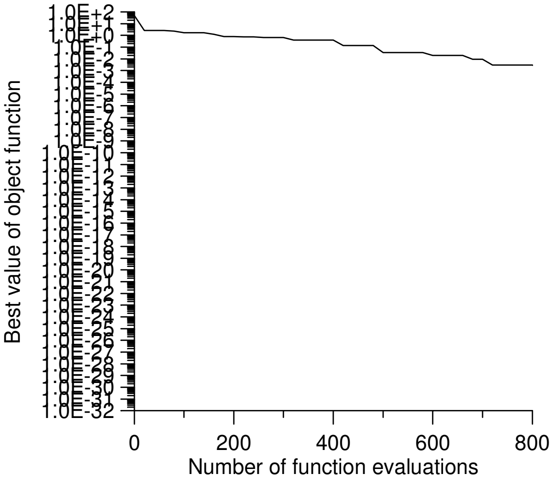


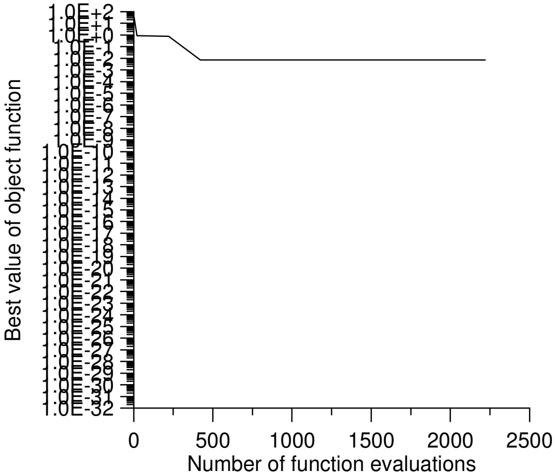


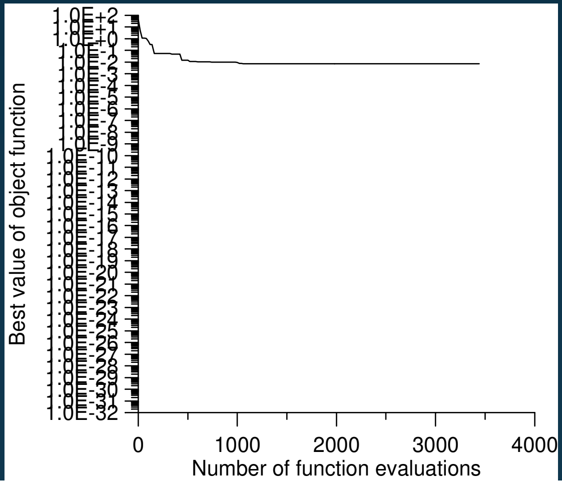
Example 1 - Griewangk's function

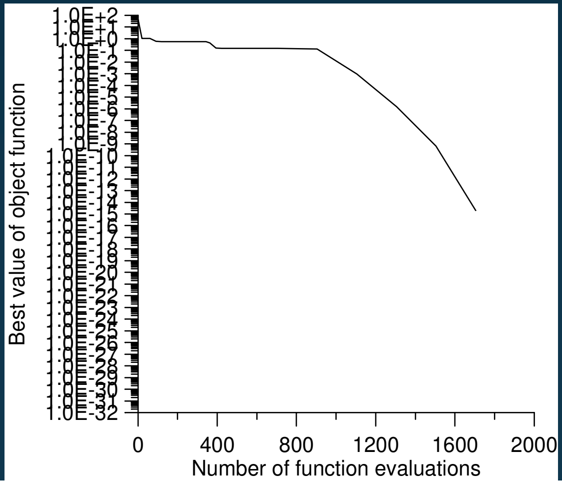
- ➔ Comparison: BFGS, DE, SA, PS, Hybrid





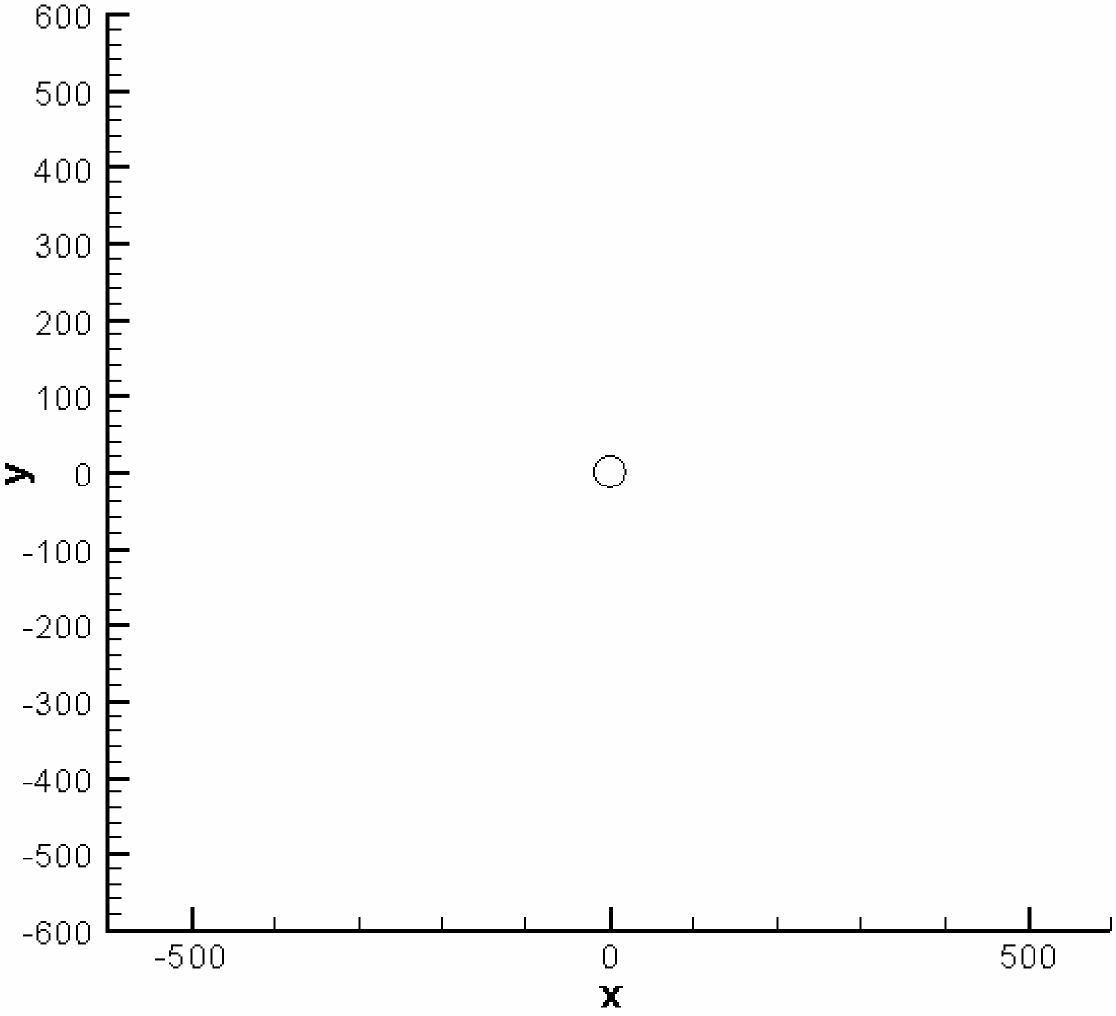






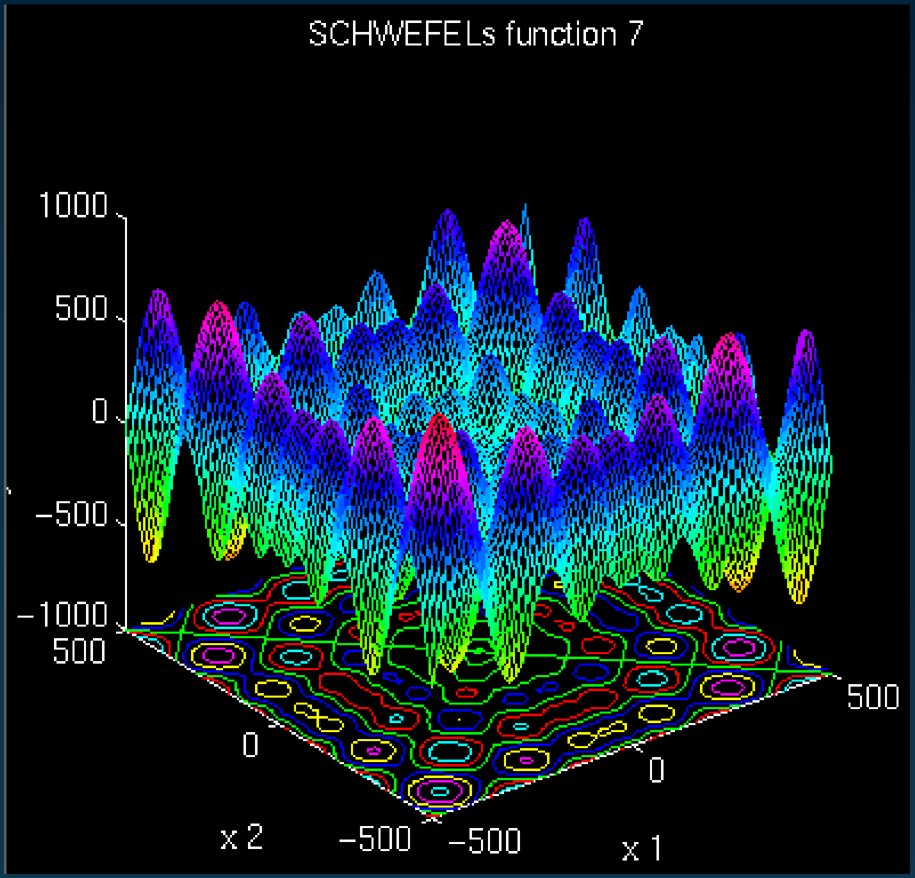
Example 1 - Griewangk's function

- Particles' history



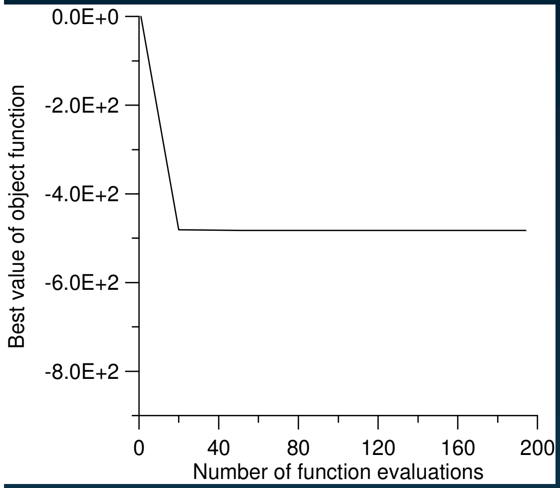
Example 2 - Schwefel’s function

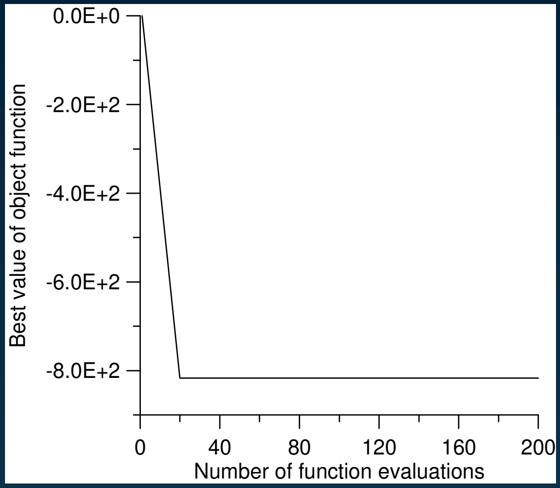
- Multiple local minima

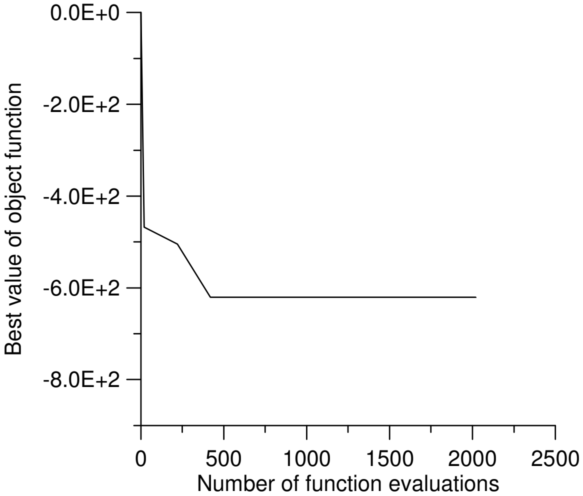


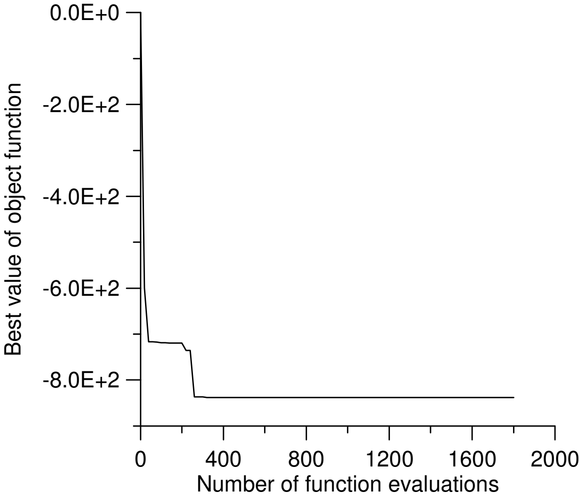
Example 2 - Schwefel’s function

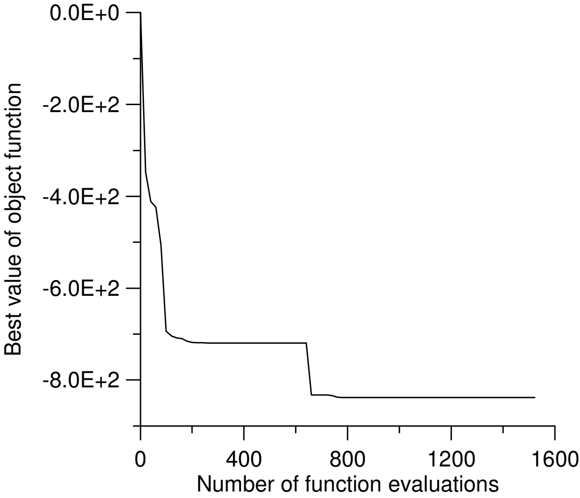
- Comparison: BFGS, DE, SA, PS, Hybrid











Direct Problem - MHD

- Conservation equations in the Cartesian coordinate system (x,y)

$$\frac{\partial Q}{\partial t}+\frac{\partial E}{\partial x}+\frac{\partial F}{\partial y}=S$$

where

$$Q=\lambda\phi$$

$$E=\lambda u\phi^{*}-\Gamma\frac{\partial\phi^{***}}{\partial x}$$

$$F=\lambda v\phi^{**}-\Gamma\frac{\partial\phi^{***}}{\partial y}$$

Conservation of	λ	ϕ	ϕ*	ϕ**	ϕ***	Γ	**S**
Mass	ρ	1	1	1	1	0	0
Species	ρ	C	C	C	C	D	$\nabla\cdot[f_s\rho_s D_s\nabla(C_s-C)]+\nabla\cdot[f_l\rho_l D_l\nabla(C_l-C)]$
x-momentum	ρ	u	u	u	u	μ	$-\frac{\partial P}{\partial x}-\frac{B_y}{\mu_m}\left[\frac{\partial B_y}{\partial x}-\frac{\partial B_x}{\partial y}\right]$
y-momentum	ρ	v	v	v	v	μ	$-\frac{\partial p}{\partial y}-\rho g[1-\beta(T-T_0)-\beta_S(C-C_0)]+\frac{B_y}{\mu_m}\left[\frac{\partial B_y}{\partial x}-\frac{\partial B_x}{\partial y}\right]$
Energy	ρ	h	h	h	T	k	$\frac{C_P}{\sigma\mu_m^2}\left[\frac{\partial B_y}{\partial x}-\frac{\partial B_x}{\partial y}\right]^2$
Magnetic x-flux	1	B_x	0	B_x	B_x	$\frac{1}{\mu_m\sigma}$	$\frac{\partial(uB_y)}{\partial y}$
Magnetic y-flux	1	B_y	B_y	0	B_y	$\frac{1}{\mu_m\sigma}$	$\frac{\partial(vB_x)}{\partial x}$

Direct Problem - MHD

- Conservation of mass, momentum, species and energy in the "computational" coordinate system (ξ,η)

$$\frac{\partial(J\rho\phi)}{\partial t}+\frac{\partial\left(\tilde{U}\rho\phi\right)}{\partial\xi}+\frac{\partial\left(\tilde{V}\rho\phi\right)}{\partial\eta}=$$
$$\frac{\partial}{\partial\xi}\left\{J\Gamma^{\phi}\left[a\frac{\partial\phi^{***}}{\partial\xi}+d\frac{\partial\phi^{***}}{\partial\eta}\right]\right\}+\frac{\partial}{\partial\eta}\left\{J\Gamma^{\phi}\left[d\frac{\partial\phi^{***}}{\partial\xi}+b\frac{\partial\phi^{***}}{\partial\eta}\right]\right\}+JS$$

$$\xi_x=\frac{y_\eta}{J}$$
$$\xi_y=-\frac{x_\eta}{J}$$
$$\eta_x=-\frac{y_\xi}{J}$$
$$\eta_y=\frac{x_\xi}{J}$$
$$J=x_\xi y_\eta-x_\eta y_\xi$$

where

$$\tilde{U}=J\left(u\xi_x+v\xi_y\right)$$
$$\tilde{V}=J\left(u\eta_x+v\eta_y\right)$$

$$a=\xi_x^2+\xi_y^2$$
$$b=\eta_x^2+\eta_y^2$$
$$d=\xi_x\eta_x+\xi_y\eta_y$$

Conservation of	λ	ϕ	ϕ^*	ϕ^{**}	ϕ^{***}	Γ	S
Mass	ρ	1	1	1	1	0	0
x-momentum	ρ	u	u	u	u	μ	$-\frac{\partial P}{\partial x}-\frac{B_y}{\mu_m}\left[\frac{\partial B_y}{\partial x}-\frac{\partial B_x}{\partial y}\right]$
y-momentum	ρ	v	v	v	v	μ	$-\frac{\partial P}{\partial y}-\rho g\left[1-\beta\left(T-T_0\right)\right]+\frac{B_y}{\mu_m}\left[\frac{\partial B_y}{\partial x}-\frac{\partial B_x}{\partial y}\right]$
Energy	ρ	h	h	h	T	K	$\frac{C_P}{\sigma\mu_m^2}\left[\frac{\partial B_y}{\partial x}-\frac{\partial B_x}{\partial y}\right]^2$

Direct Problem - MHD

- Conservation of magnetic flux in x-direction in the "computational" coordinate system (ξ,η)

$$\frac{\partial(J\rho\phi)}{\partial t}+\frac{\partial\left(\tilde{U}_{Bx}\rho\phi\right)}{\partial\xi}+\frac{\partial\left(\tilde{V}_{Bx}\rho\phi\right)}{\partial\eta}=$$

$$\frac{\partial}{\partial\xi}\left\{J\Gamma^{\phi}\left[a\frac{\partial\phi^{***}}{\partial\xi}+d\frac{\partial\phi^{***}}{\partial\eta}\right]\right\}+\frac{\partial}{\partial\eta}\left\{J\Gamma^{\phi}\left[d\frac{\partial\phi^{***}}{\partial\xi}+b\frac{\partial\phi^{***}}{\partial\eta}\right]\right\}+JS$$

where

$$\tilde{U}_{By}=J\left(0\xi_x+v\xi_y\right)$$

$$\tilde{V}_{By}=J\left(0\eta_x+v\eta_y\right)$$

$$a=\xi_x^2+\xi_y^2$$

$$b=\eta_x^2+\eta_y^2$$

$$d=\xi_x\eta_x+\xi_y\eta_y$$

$$\xi_x=\frac{y_\eta}{J}$$

$$\xi_y=-\frac{x_\eta}{J}$$

$$\eta_x=-\frac{y_\xi}{J}$$

$$\eta_y=\frac{x_\xi}{J}$$

$$J=x_\xi y_\eta-x_\eta y_\xi$$

Conservation of	λ	ϕ	ϕ^*	ϕ^{**}	ϕ^{***}	Γ	S
Magnetic flux in x-direction	1	B_x	0	B_x	B_x	$\frac{1}{\mu_m\sigma}$	$\frac{\partial(uB_y)}{\partial y}$

Direct Problem - MHD

- Conservation of magnetic flux in y-direction in the “computational” coordinate system (ξ,η)

$$\frac{\partial(J\rho\phi)}{\partial t}+\frac{\partial\left(\tilde{U}_{By}\rho\phi\right)}{\partial\xi}+\frac{\partial\left(\tilde{V}_{By}\rho\phi\right)}{\partial\eta}=$$

$$\frac{\partial}{\partial\xi}\left\{J\Gamma^{\phi}\left[a\frac{\partial\phi^{***}}{\partial\xi}+d\frac{\partial\phi^{***}}{\partial\eta}\right]\right\}+\frac{\partial}{\partial\eta}\left\{J\Gamma^{\phi}\left[d\frac{\partial\phi^{***}}{\partial\xi}+b\frac{\partial\phi^{***}}{\partial\eta}\right]\right\}+JS$$

where

$$\tilde{U}_{By}=J\left(u\xi_x+0\xi_y\right)$$
$$\tilde{V}_{By}=J\left(u\eta_x+0\eta_y\right)$$

$$a=\xi_x^2+\xi_y^2$$
$$b=\eta_x^2+\eta_y^2$$
$$d=\xi_x\eta_x+\xi_y\eta_y$$

$$\xi_x=\frac{y_\eta}{J}$$
$$\xi_y=-\frac{x_\eta}{J}$$
$$\eta_x=-\frac{y_\xi}{J}$$
$$\eta_y=\frac{x_\xi}{J}$$
$$J=x_\xi y_\eta-x_\eta y_\xi$$

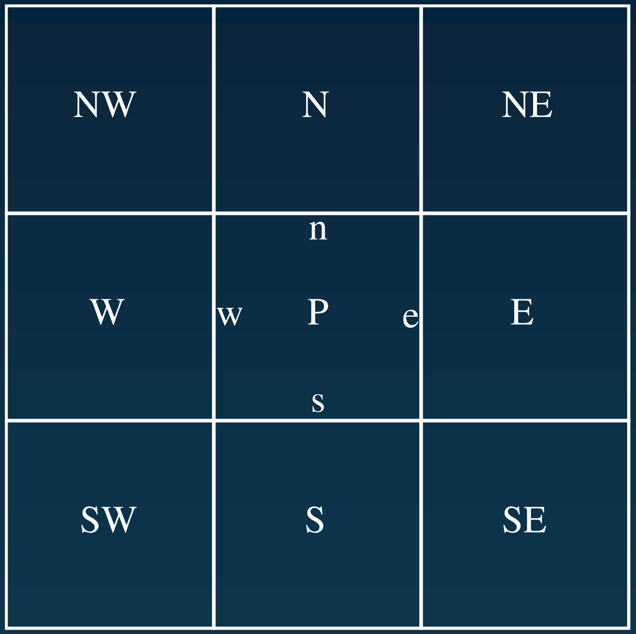
Conservation of	λ	ϕ	ϕ^*	ϕ^{**}	ϕ^{***}	Γ	S
Magnetic flux in y-direction	1	B_y	B_y	0	B_y	$\frac{1}{\mu_m\sigma}$	$\frac{\partial(vB_x)}{\partial x}$

Direct Problem - MHD

- Numerical method → Finite Volumes (Implicit)
- Linear system solver → Bi-Conjugate gradient method (GMRES)
- Interpolation function for the convective terms → WUDS (Raithby and Torrance, 1974)
- Pressure-velocity coupling → SIMPLEC (Van Doormal and Raithby, 1984)
- Phase-change model → Enthalpy method (Voller et al, 1989)

Direct Problem - MHD

- Finite Volume Method
  - Control volume where the conservation equations are integrated



Direct Problem - MHD

- ➔ Finite Volume Method – Integrated conservation equations

$$\frac{(M_P\phi_P - M_P^0\phi_P^0)}{\Delta t} + \dot{M}_e\phi_e - \dot{M}_w\phi_w + \dot{M}_n\phi_n - \dot{M}_s\phi_s =$$

$$= \left[D_{11}\frac{\partial\phi}{\partial\xi} + D_{12}\frac{\partial\phi}{\partial\eta}\right]_e - \left[D_{11}\frac{\partial\phi}{\partial\xi} + D_{12}\frac{\partial\phi}{\partial\eta}\right]_w +$$

$$+ \left[D_{21}\frac{\partial\phi}{\partial\xi} + D_{22}\frac{\partial\phi}{\partial\eta}\right]_n - \left[D_{21}\frac{\partial\phi}{\partial\xi} + D_{22}\frac{\partial\phi}{\partial\eta}\right]_s +$$

$$+ L[S^\phi J]_P\Delta V - L[P^\phi J]_P\Delta V$$

where

$$M_P = \rho_P \Delta V J_P$$

$$M_P^0 = \rho_P^0 \Delta V J_P$$

$$\dot{M}_e = \rho\widetilde{U}|_e\Delta\eta$$

$$\dot{M}_w = \rho\widetilde{U}|_w\Delta\eta$$

$$\dot{M}_n = \rho\widetilde{V}|_n\Delta\xi$$

$$\dot{M}_s = \rho\widetilde{V}|_s\Delta\xi$$

$$D_{11} = \frac{\alpha_{11}}{J}\Gamma^\phi\Delta\eta$$

$$D_{12} = \frac{\alpha_{12}}{J}\Gamma^\phi\Delta\eta$$

$$D_{21} = \frac{\alpha_{21}}{J}\Gamma^\phi\Delta\xi$$

$$D_{22} = \frac{\alpha_{22}}{J}\Gamma^\phi\Delta\xi$$

$$\alpha_{11} = aJ^2$$

$$\alpha_{12} = \alpha_{21} = dJ^2$$

$$\alpha_{22} = bJ^2$$

Direct Problem - MHD

- WUDS interpolation scheme

$$\phi_e = \left(\frac{1}{2} + \overline{\alpha}_e\right)\phi_P + \left(\frac{1}{2} - \overline{\alpha}_e\right)\phi_E$$

$$\left.\frac{\partial\phi}{\partial\xi}\right|_e = \overline{\beta}_e\left(\frac{\phi_E - \phi_P}{\Delta\xi}\right)$$

where

$$\overline{\alpha} = \frac{r^2}{10 + 2r^2} sign(r)$$

$$\overline{\beta} = \frac{1 + 0.005r^2}{1 + 0.05r^2}$$

$$r = \frac{\dot{M}}{D}$$

Note that:

$\alpha=0; \beta=1 \Rightarrow$ Central differencing scheme

$\alpha=0.5; \beta=0 \Rightarrow$ Upwind scheme for $u > 0$

$\alpha= - 0.5; \beta=0 \Rightarrow$ Upwind scheme for $u < 0$

Direct Problem - MHD

- WUDS interpolation scheme

$$A_e = -\dot{M}_e\left(\frac{1}{2} - \overline{\alpha}_e\right) + \frac{D_{11e}\overline{\beta}_e}{\Delta\xi} + \frac{D_{21n} - D_{21s}}{4\Delta\xi}$$

$$A_w = \dot{M}_w\left(\frac{1}{2} + \overline{\alpha}_w\right) + \frac{D_{11w}\overline{\beta}_w}{\Delta\xi} + \frac{D_{21s} - D_{21n}}{4\Delta\xi}$$

$$A_n = -\dot{M}_n\left(\frac{1}{2} - \overline{\alpha}_n\right) + \frac{D_{22n}\overline{\beta}_n}{\Delta\eta} + \frac{D_{12e} - D_{12w}}{4\Delta\eta}$$

$$A_s = \dot{M}_s\left(\frac{1}{2} + \overline{\alpha}_s\right) + \frac{D_{22s}\overline{\beta}_s}{\Delta\eta} + \frac{D_{12w} - D_{12e}}{4\Delta\eta}$$

$$A_{ne} = \frac{D_{12e}}{4\Delta\eta} + \frac{D_{21n}}{4\Delta\xi}$$

$$A_{se} = -\frac{D_{12e}}{4\Delta\eta} - \frac{D_{21s}}{4\Delta\xi}$$

$$A_{nw} = -\frac{D_{12w}}{4\Delta\eta} - \frac{D_{21n}}{4\Delta\xi}$$

$$A_{sw} = \frac{D_{12w}}{4\Delta\eta} + \frac{D_{21s}}{4\Delta\xi}$$

$$\begin{aligned} A_P = {} & \frac{M_P}{\Delta t} + \\ & + \dot{M}_e\left(\frac{1}{2} + \overline{\alpha}_e\right) - \dot{M}_w\left(\frac{1}{2} - \overline{\alpha}_w\right) + \\ & + \dot{M}_n\left(\frac{1}{2} + \overline{\alpha}_n\right) - \dot{M}_s\left(\frac{1}{2} - \overline{\alpha}_s\right) + \\ & + \frac{D_{11e}\overline{\beta}_e}{\Delta\xi} + \frac{D_{11w}\overline{\beta}_w}{\Delta\xi} + \frac{D_{22n}\overline{\beta}_n}{\Delta\eta} + \frac{D_{22s}\overline{\beta}_s}{\Delta\eta} \end{aligned}$$

$$B_P = \frac{M_P^0\phi_P^0}{\Delta t} + L[S^\phi J]_P\Delta V - L[P^\phi J]_P\Delta V$$

$$\begin{aligned} A_P\phi_P = {} & A_e\phi_E + A_w\phi_W + A_n\phi_N + A_s\phi_S + \\ & + A_{ne}\phi_{NE} + A_{nw}\phi_{NW} + A_{se}\phi_{SE} + A_{sw}\phi_{SW} + B_P \end{aligned}$$

Solved by the GMRES method

Direct Problem - MHD

➔ SIMPLEC

- The mass conservation equation and the momentum conservation equations are coupled through the pressure.
- SIMPLEC is one of the methodologies to deal with this problem which is known as pressure-velocity coupling.
- Others methods are SIMPLE, SIMPLER, PRIME, CELLS, etc
- SIMPLE means *Semi Implicit Linked Equations*
- SIMPLEC means *SIMPLE Consistent*

Direct Problem - MHD

- Phase-Change Model – Energy equation:

$$\frac{\partial(\rho h)}{\partial t}+\frac{\partial(uh)}{\partial x}+\frac{\partial(vh)}{\partial y}=\frac{\partial}{\partial x}\left[K\frac{\partial T}{\partial x}\right]+\frac{\partial}{\partial y}\left[K\frac{\partial T}{\partial y}\right]+S$$

- Enthalpy-Temperature relationships:

If $h < h_{solid}$:

$$T=\frac{h}{C_{Ps}}$$

If $h > h_{liquid}$:

$$T=\frac{h+T_s\left(C_{Pl}-C_{Ps}\right)-L}{C_{Pl}}$$

If $h_{solid} < h < h_{liquid}$:

$$T=\frac{h+\left[T_s\left(C_{Pl}-C_{Ps}\right)-L\right]\left(1-f\right)}{C_{Pl}+f\left(C_{Ps}-C_{Pl}\right)}$$

- Solid fraction (Scheil's model):

$$f=1-\left(\frac{T_s-T}{T_s-T_l}\right)^{1/(k-1)}$$

***f* and *T* equations are coupled**

Direct Problem - MHD

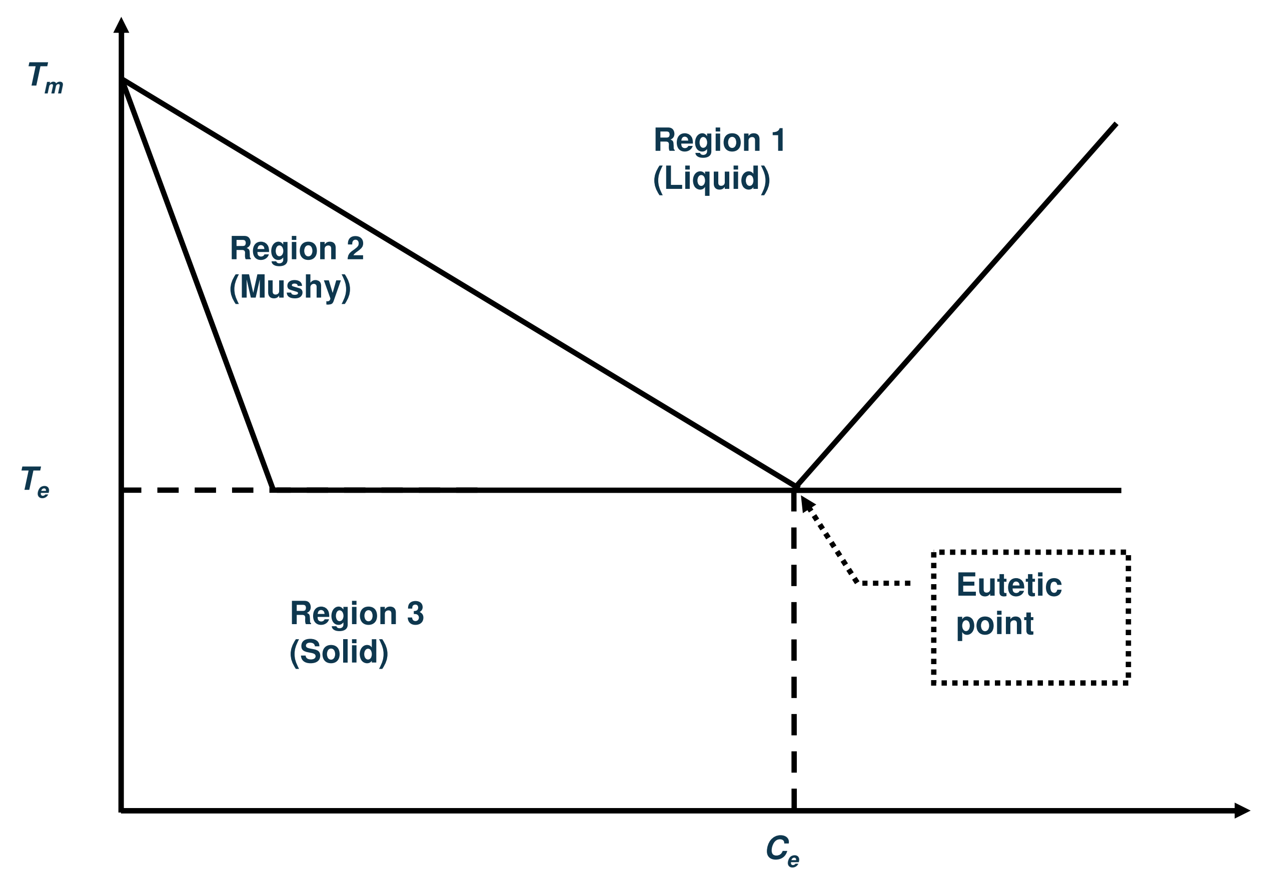
- Phase-Change Model
  - Mixture Enthalpy-Temperature relationships

$$\frac{T - h + \left[T_s\left(C_{Pl} - C_{Ps}\right) - L\right]\left[\left(\frac{T_s - T}{T_s - T_l}\right)^{1/(k-1)}\right]}{C_{Pl} + \left[1 - \left(\frac{T_s - T}{T_s - T_l}\right)^{1/(k-1)}\right]\left(C_{Ps} - C_{Pl}\right)} = 0$$

Solved by the secant method

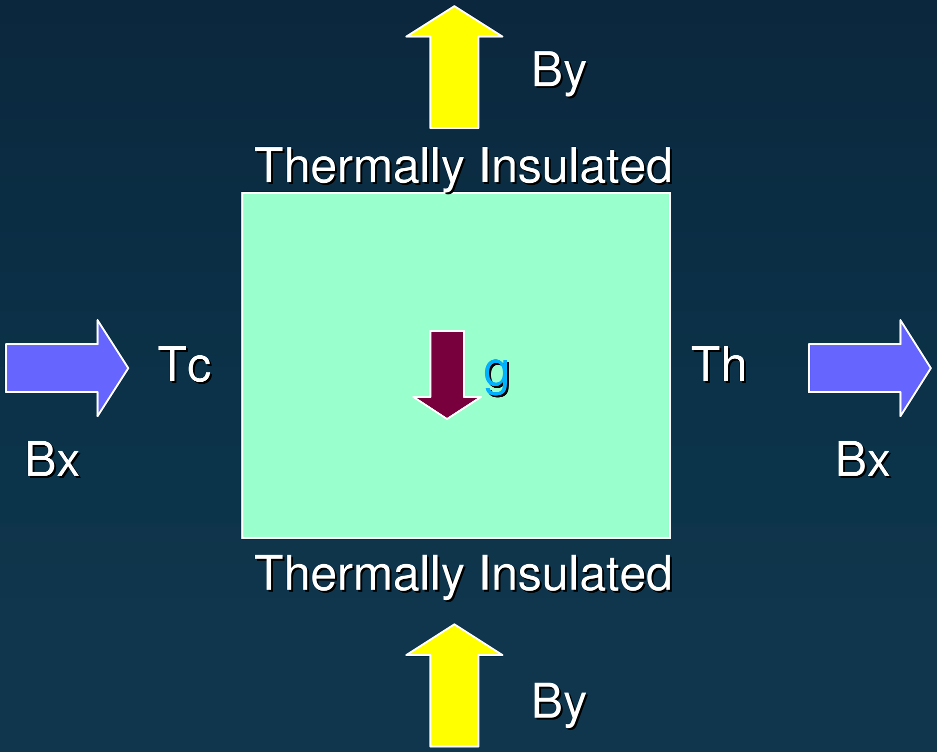
Direct Problem - MHD

- Binary Diagram



MHD – Example 1

- ➔ Optimization of the magnetic boundary conditions in order to minimize the natural convection effects
- ➔ Test problem:



MHD – Example 1

- Inverse Problem
- The boundary conditions are parameterized as:

$$B(x_k) = \sum_{i=1}^{M} P_i \, C_i \, (x_k)$$

where:

$$C_i(x_k) = \cos\left[(i-1)\frac{\pi}{2} x_k \right] \quad \text{for i} = 1,3,5,...$$

$$C_i(x_k) = \cos\left[i\frac{\pi}{2} x_k \right] \quad \text{for i} = 2,4,6,...$$

MHD – Example 1

- Inverse Problem
- Two test cases: the physical parameters are:

$\rho_{\ell} = 2550$ kg / m^3	$\rho_s = 2330$ kg / m^3
$c_{p_{\ell}} = 1059$ J / kg K	$c_{p_s} = 1038$ J / kg K
$\kappa_{\ell} = 64$ W / m K	$\kappa_s = 22$ W / m K
$T_{\ell} = 1685$ K	$T_s = 1681$ K
$\sigma_{\ell} = 12.3 \times 10^5$ 1 / Ω m	$\sigma_s = 4.3 \times 10^4$ 1 / Ω m
$\mu_{v\ell} = 7.018 \times 10^{-4}$ kg / m s	$\mu = 7.022 \times 10^{-4}$ kg m / A^2 s^2
$L = 1.803 \times 10^6$ J / kg	$\alpha = 1.4 \times 10^{-4}$ 1 / K

Case 1 (small container)	Case 2 (large container)
$l_r = 0.01$ m	$l_r = 0.02$ m
Re = 1000	Re = 1000
$B_r = 0.1$ T	$B_r = 0.1$ T
$v_r = 2.7522 \times 10^{-2}$ m/s	$v_r = 1.3761 \times 10^{-2}$ m/s
Pr = 1.1613×10^{-2}	Pr = 1.1613×10^{-2}
Gr = 1.8132×10^5	Gr = 1.4506×10^6
Ra = 2.1056×10^3	Ra = 1.6845×10^4
Fr = 8.7870×10^{-2}	Fr = 3.1067×10^{-2}
Ec = 7.1524×10^{-8}	Ec = 1.7881×10^{-8}
Ht = 4.1864×10^1	Ht = 8.3729×10^1

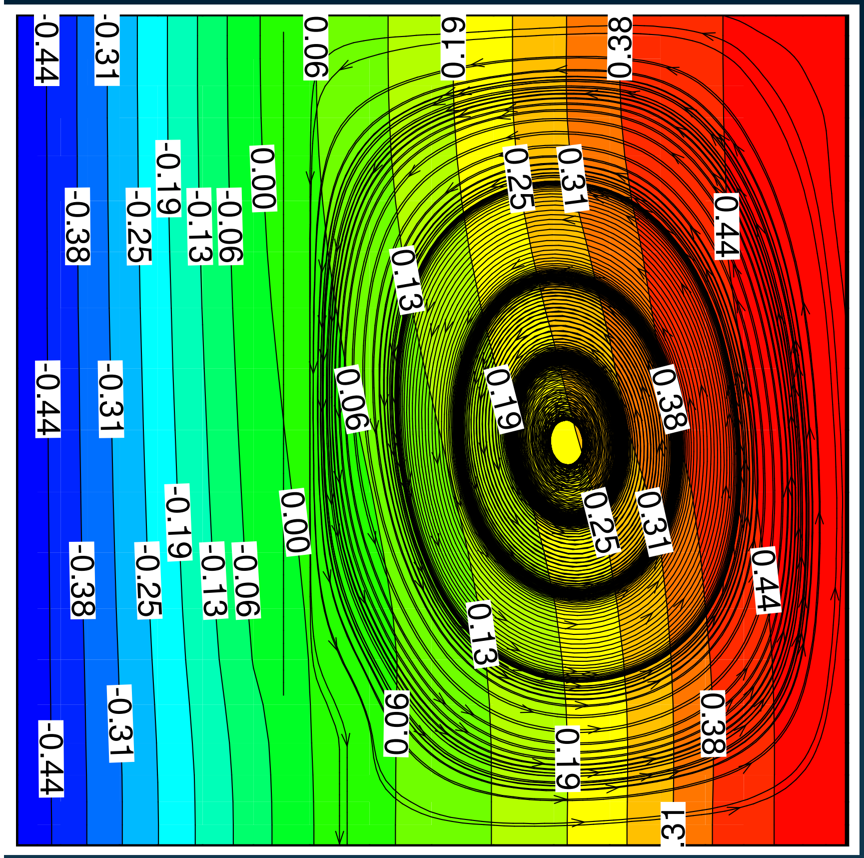
MHD – Example 1

- Inverse Problem:
- Objective function

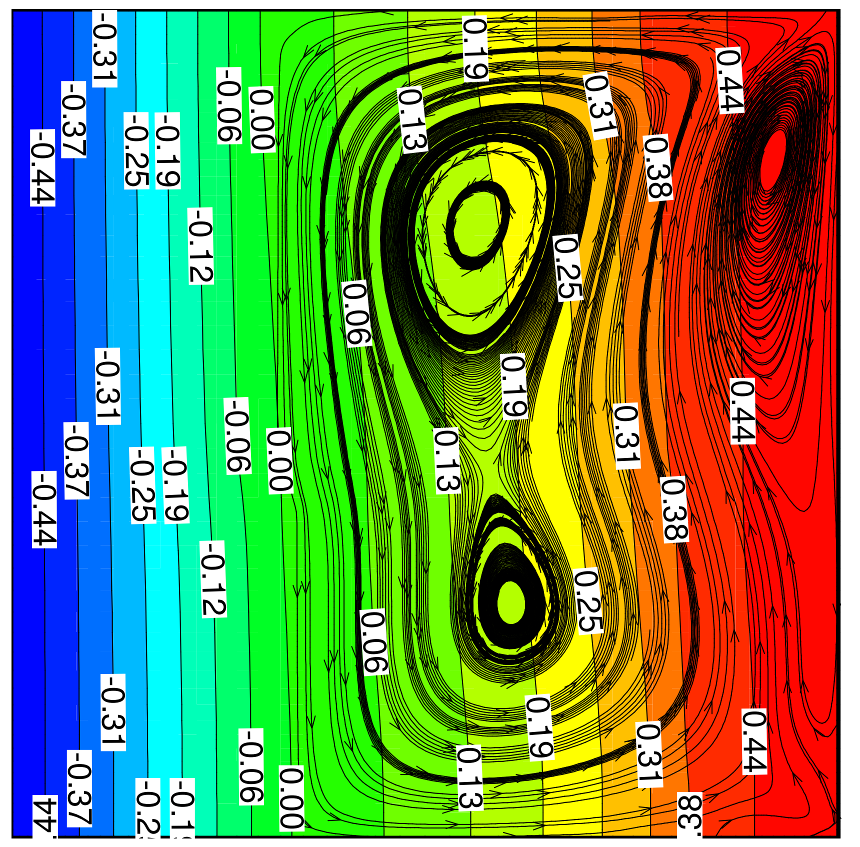
$$F = \left[\frac{1}{\#\text{liquid cells}} \sum_{i=1}^{\#\text{liquid cells}} \left(\frac{\partial \mathrm{T}}{\partial \mathrm{y}} \right)^2 \right]^{1/2}$$

MHD – Example 1

- Inverse Problem: Test case 1 (small container), 3 parameters per boundary (Ra=2.1056x10^3)



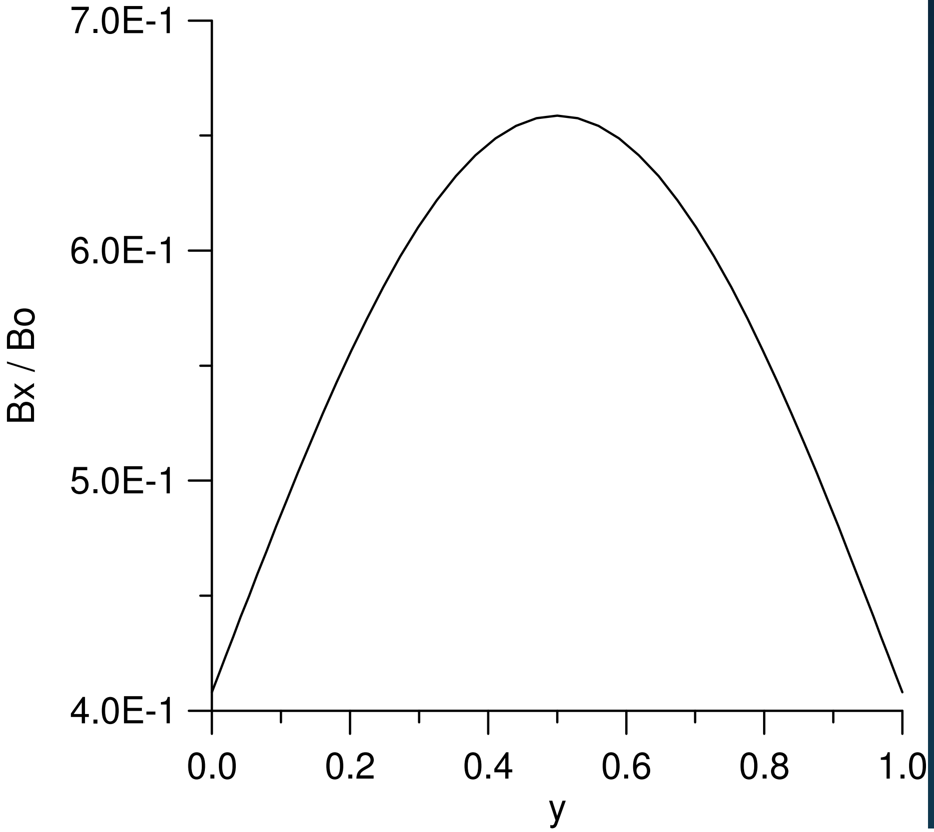
Without magnetic fields

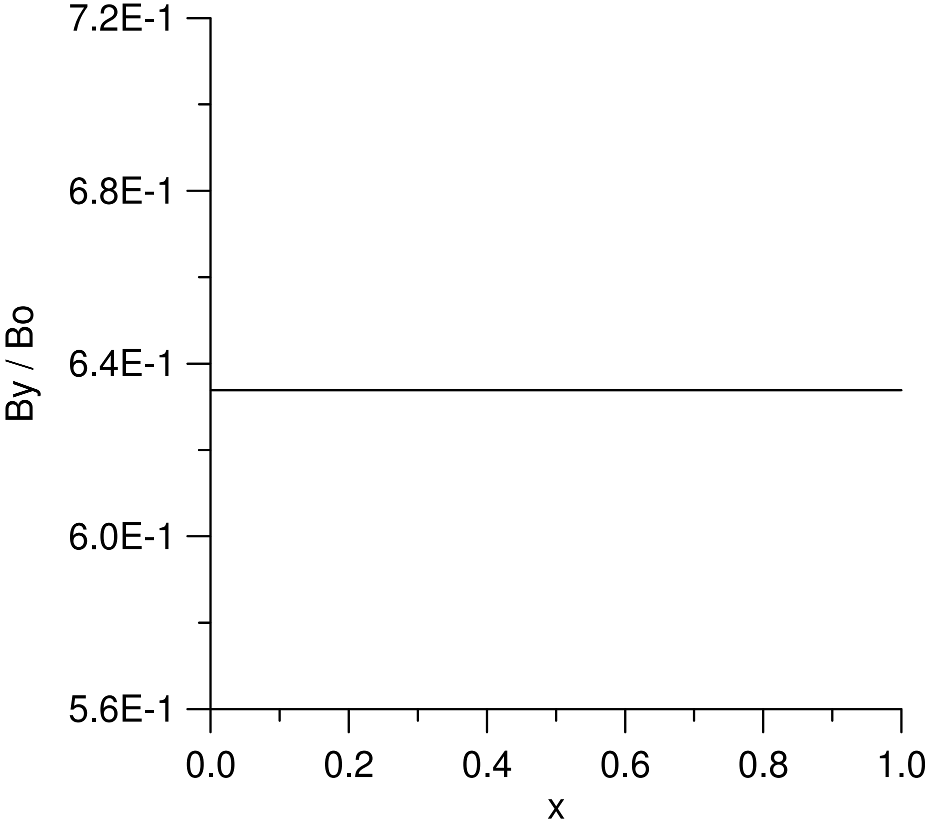


With optimized magnetic fields

MHD – Example 1

- Inverse Problem: Test case 1 (small container), 3 parameters per boundary
- Optimized boundary conditions

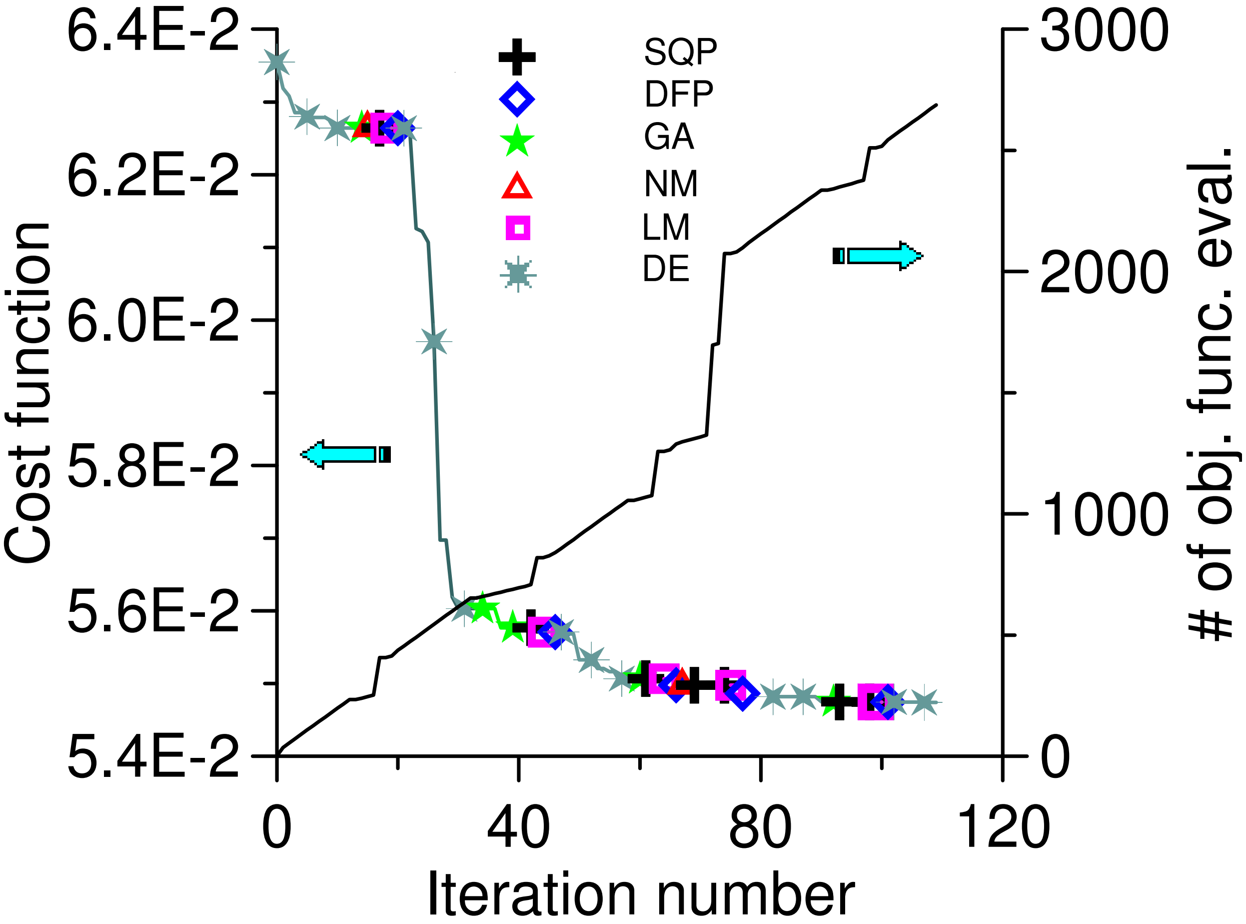




MHD – Example 1

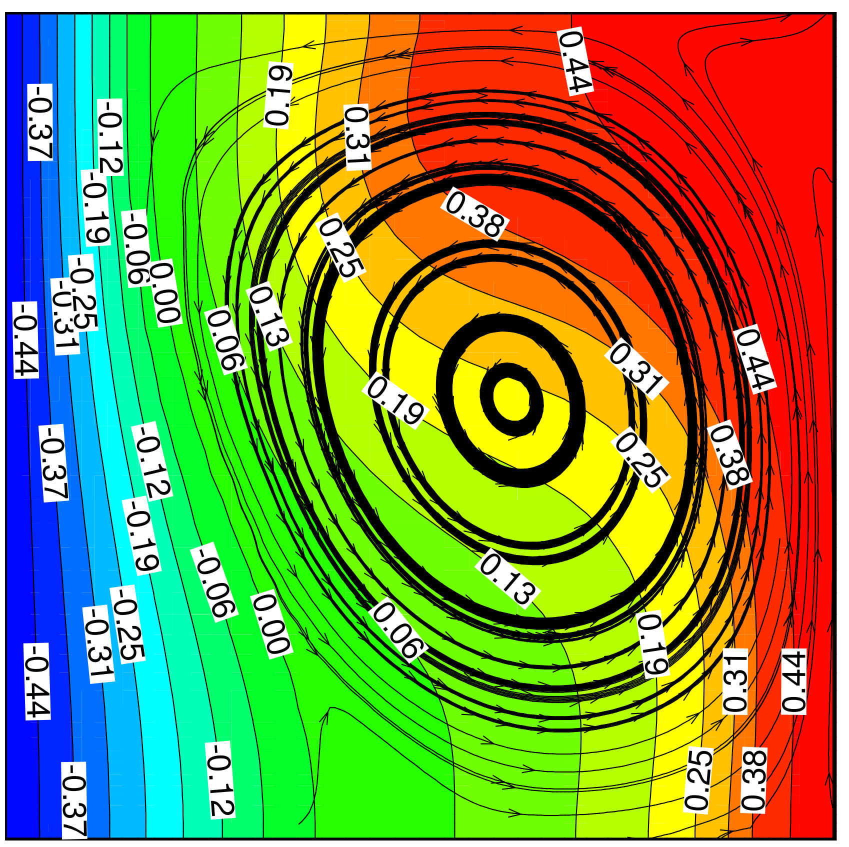
- Inverse Problem: Test case 1 (small container), 3 parameters per boundary
- Convergence history

DE did almost all the job

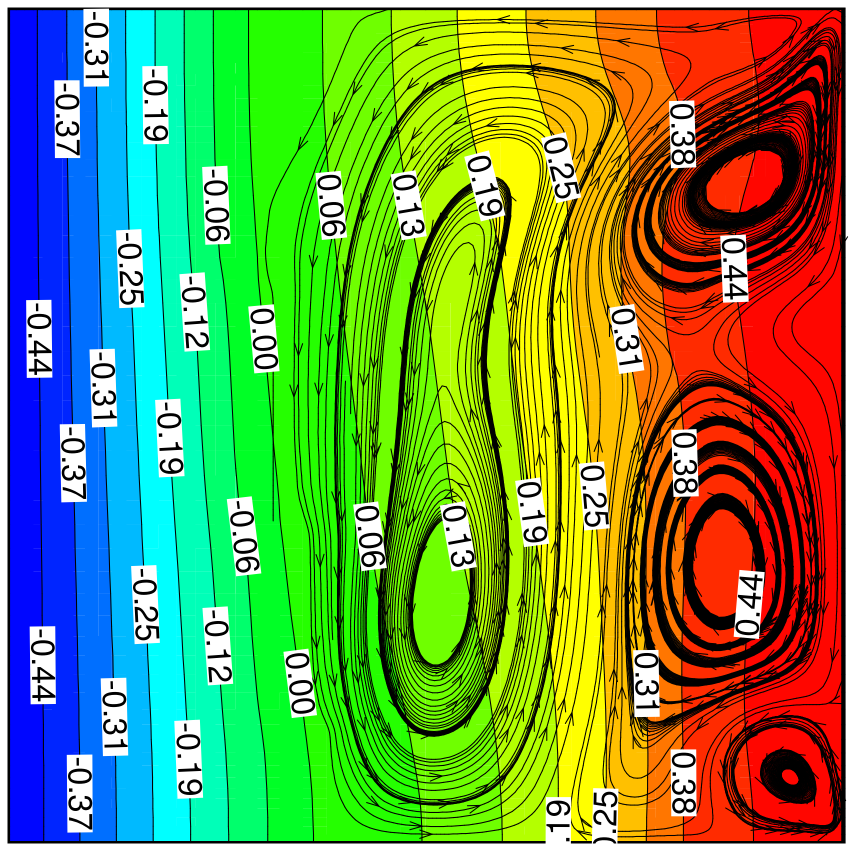


MHD – Example 1

- ➔ Inverse Problem: Test case 2 (large container), 3 parameters per boundary ($Ra=1.6845x10^4$)



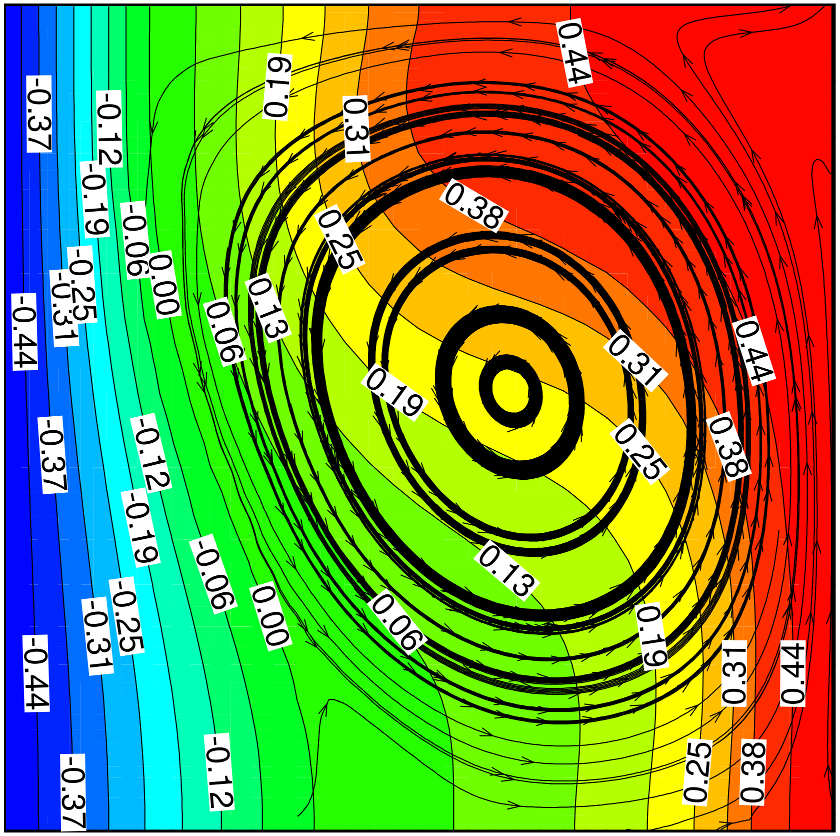
Without magnetic fields



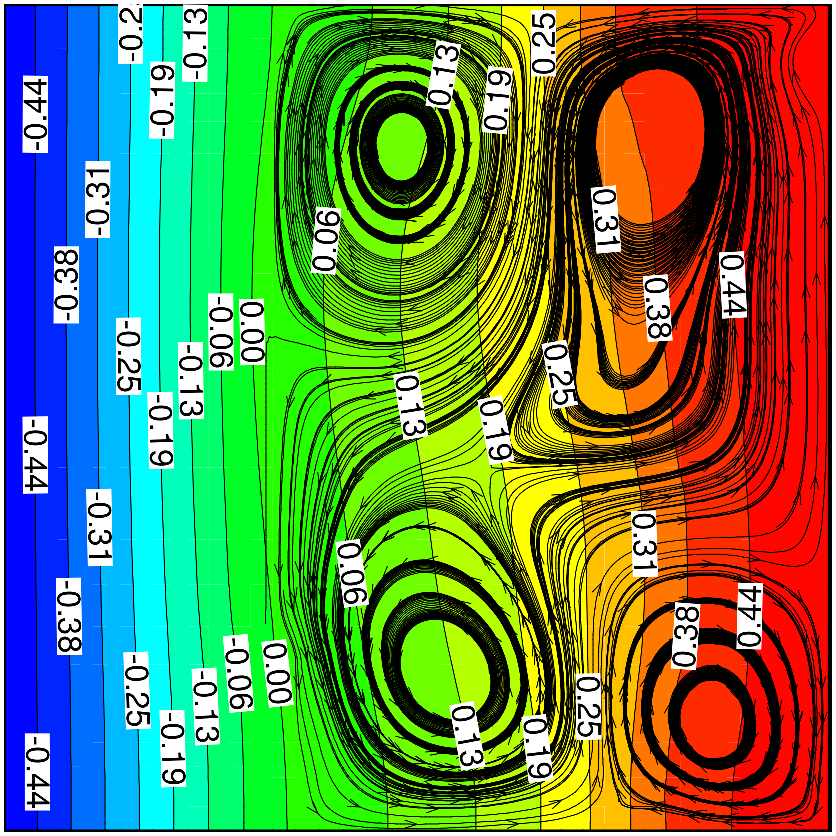
With optimized magnetic fields

MHD – Example 1

- ➔ Inverse Problem: Test case 2 (large container), 6 parameters per boundary (Ra=1.6845x10^4)



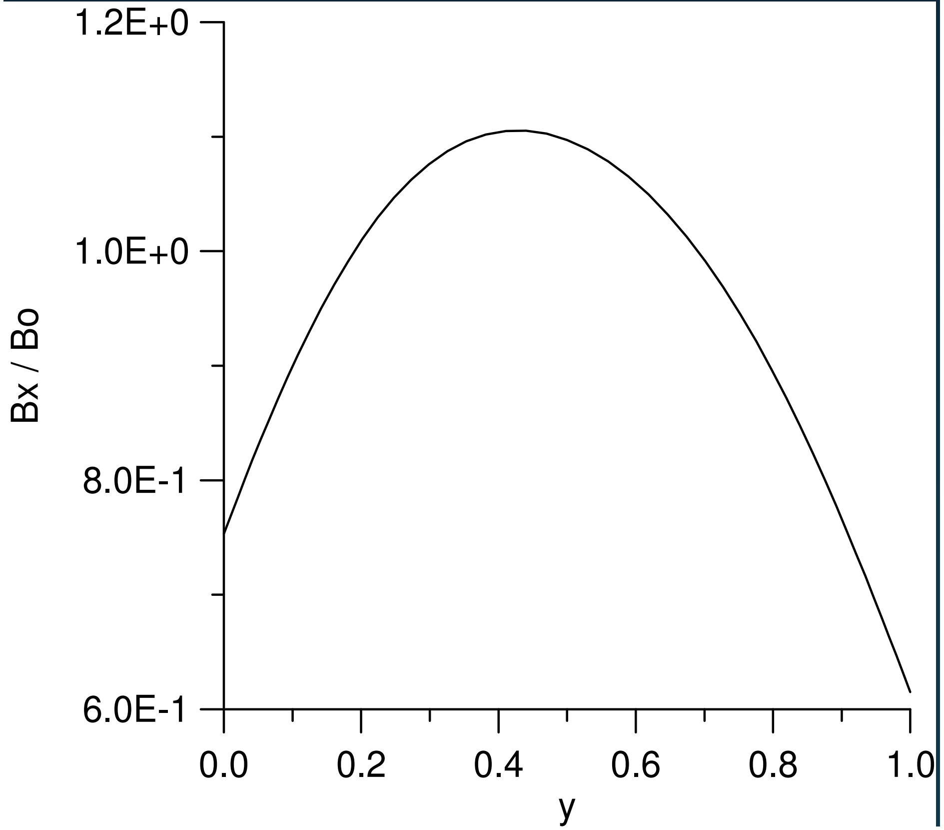
Without magnetic fields

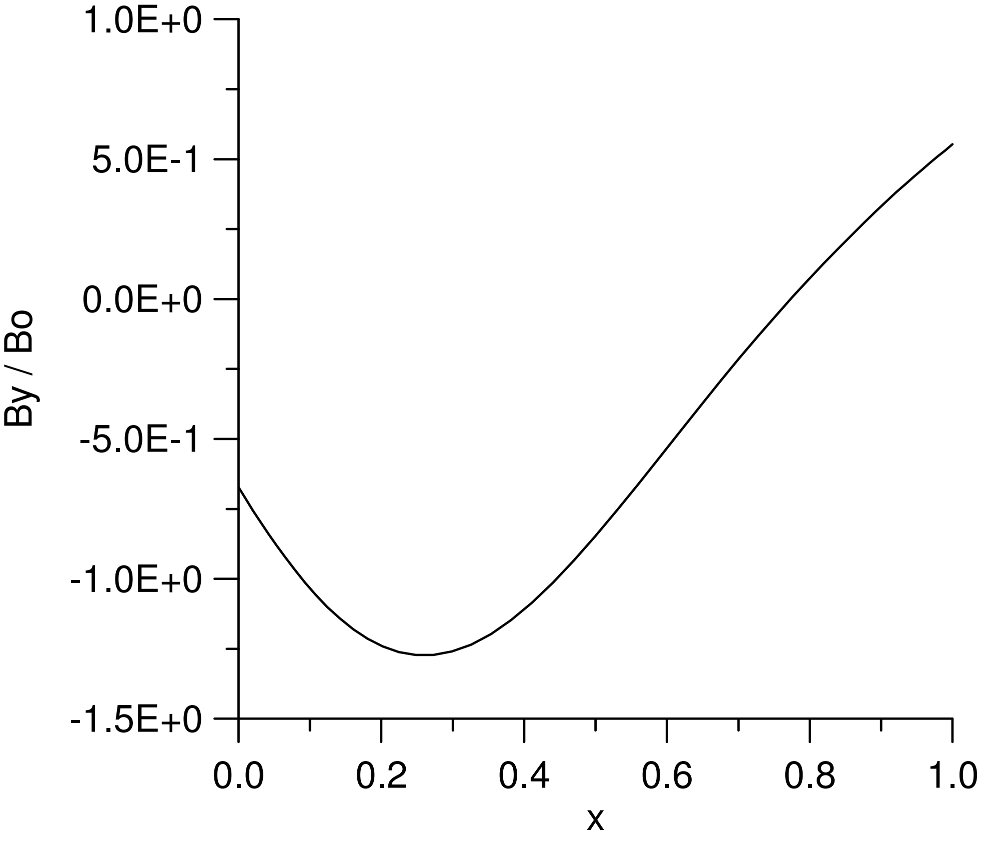


With optimized magnetic fields

MHD – Example 1

- ➔ Inverse Problem: Test case 2 (large container), 6 parameters per boundary
- ➔ Optimized boundary conditions

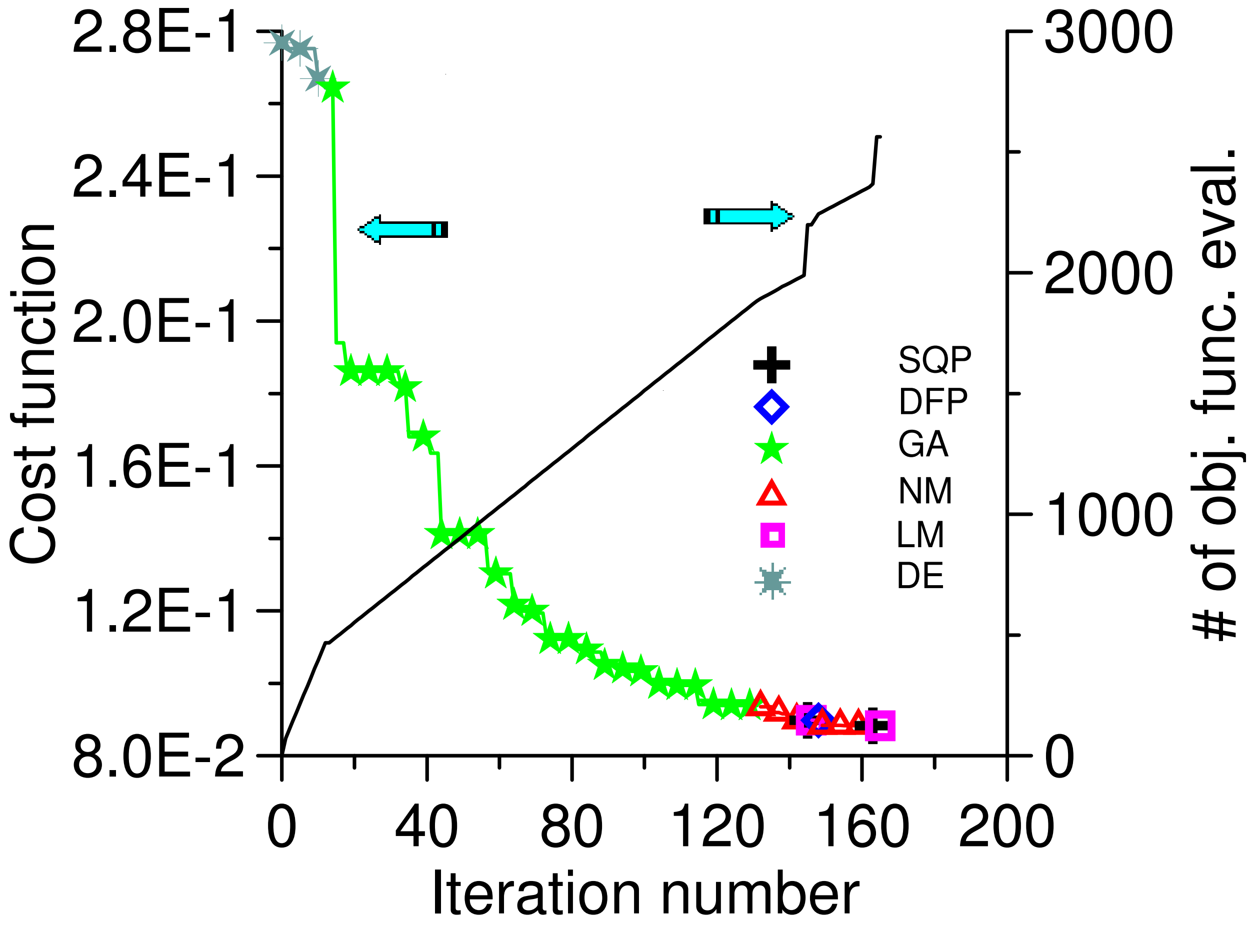




MHD – Example 1

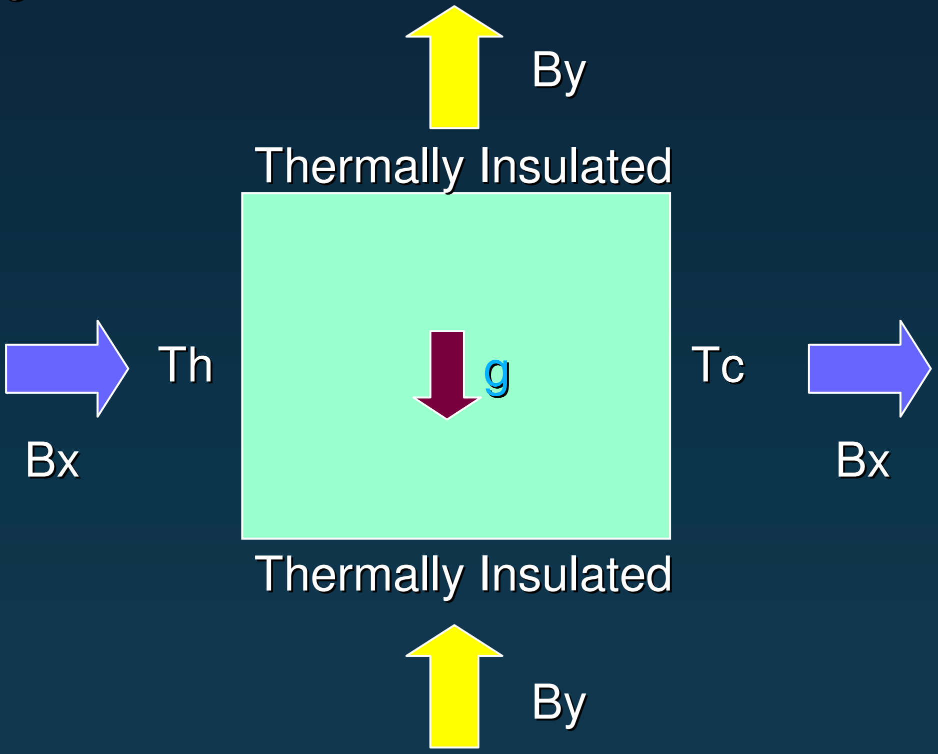
- ➔ Inverse Problem: Test case 2 (large container), 6 parameters per boundary
- ➔ Convergence history

GA did almost all the job



MHD – Example 2

- Optimization of the magnetic boundary conditions in order to minimize the natural convection effects (transient optimization)
- Final time = 300 s
- Test problem:



MHD – Example 2

- Optimization Problem
- The boundary conditions are parameterized as B-Splines.
- NETLIB's subroutine GCVSPL, based on the cross-validation smoothing procedure, was used for the interpolation.
- Objective function:

$$F = \sqrt{\frac{1}{\#cells}\sum_{i=1}^{\#cells}\left(\frac{\partial T_i}{\partial y_i}\right)^2}$$

MHD – Example 2

- Optimization Problem
- Physical properties for silicon:

$\rho_l = 2550$ kg m^{-3}

$k_l = 64$ W m^{-1} K^{-1}

$C_{Pl} = 1059$ J kg^{-1} K^{-1}

$\mu_l = 0.0032634$ kg m^{-1} s^{-1}

$\sigma_l = 12.3 \times 10^{5}$ 1/m Ω

$\beta = 8.3 \times 10^{-4}$ K^{-1}

$\mu_m = 1.2566 \times 10^{-5}$ T m A^{-1}

$\rho_l = 2330$ kg m^{-3}

$k_s = 22$ W m^{-1} K^{-1}

$C_{Ps} = 1038$ J kg^{-1} K^{-1}

$\mu_s = 1.0 \times 10^{3}$ kg m^{-1} s^{-1}

$\sigma_s = 4.3 \times 10^{4}$ 1/m Ω

$g = 9.81$ m s^{-2}

$L = 1.803 \times 10^{6}$ J kg^{-1}

$\sigma_s = 4.3 \times 10^{4}$ 1/m Ω

- Test case without solidification → H=0.15 m, ΔT=0.654351 K, Ra=10^5
- Test case with solidification → H=0.069624 m, ΔT=6.54351 K, Ra=10^5

MHD – Example 2

- Optimization Problem:Test case 1(without solidification),4 parameters per boundary

Without magnetic fields

With optimized magnetic fields

MHD – Example 2

- Optimization Problem:Test case 1(without solidification),6 parameters per boundary

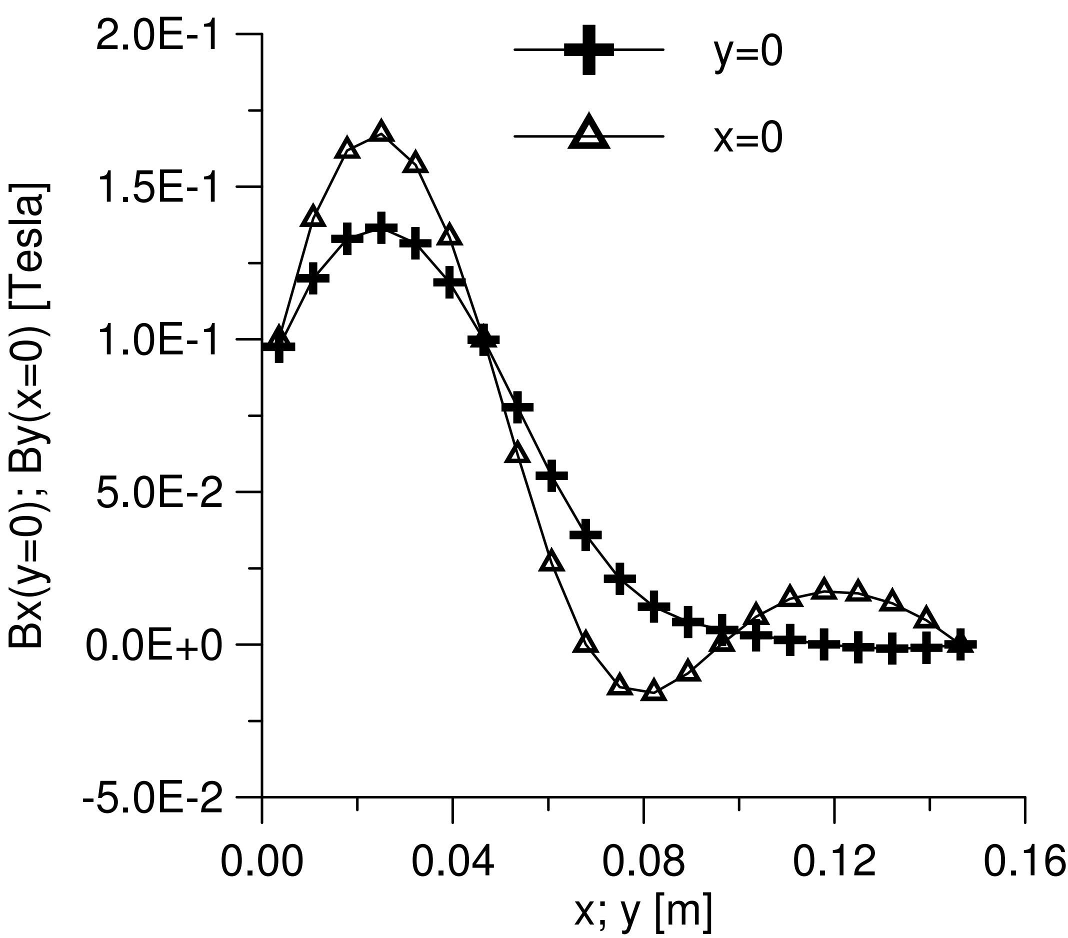
Without magnetic fields

With optimized magnetic fields

MHD – Example 2

- Optimization Problem:Test case 1(without solidification),6 parameters per boundary

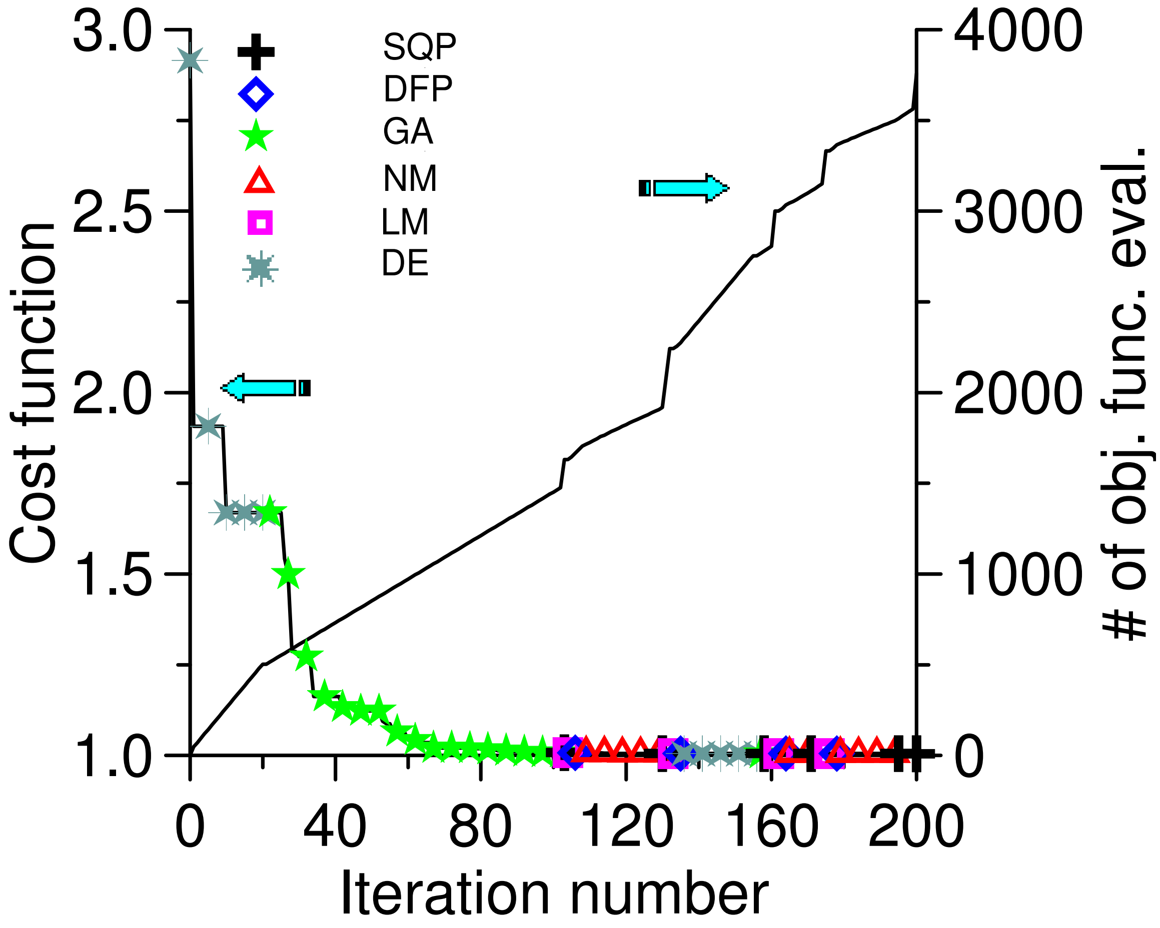
Optimized boundary conditions



MHD – Example 2

- ➔ Optimization Problem:Test case 1(without solidification),6 parameters per boundary
- ➔ Convergence history

DE and GA did almost all the job



MHD – Example 2

- ➔ Optimization Problem: Test case 2 (with solidification), 6 parameters per boundary

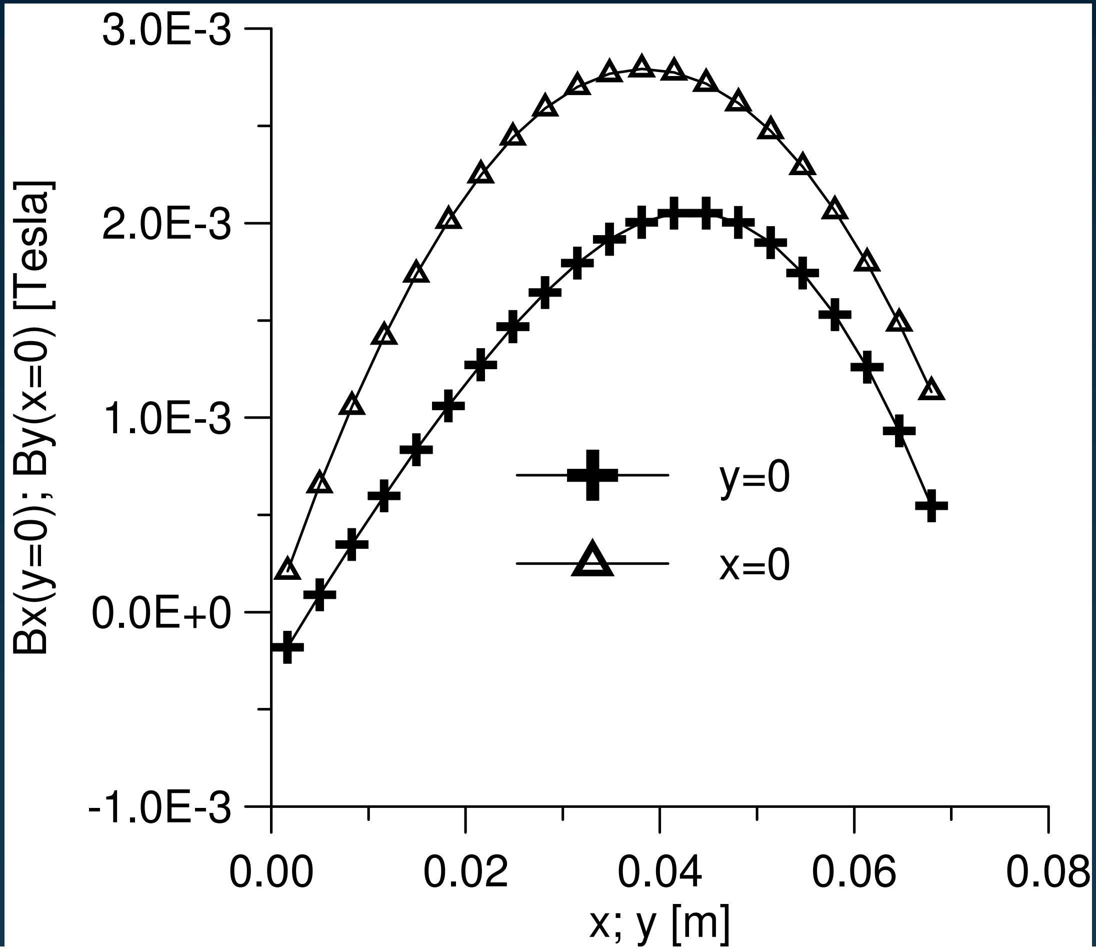
Without magnetic fields

With optimized magnetic fields

MHD – Example 2

- Optimization Problem: Test case 2 (with solidification), 6 parameters per boundary

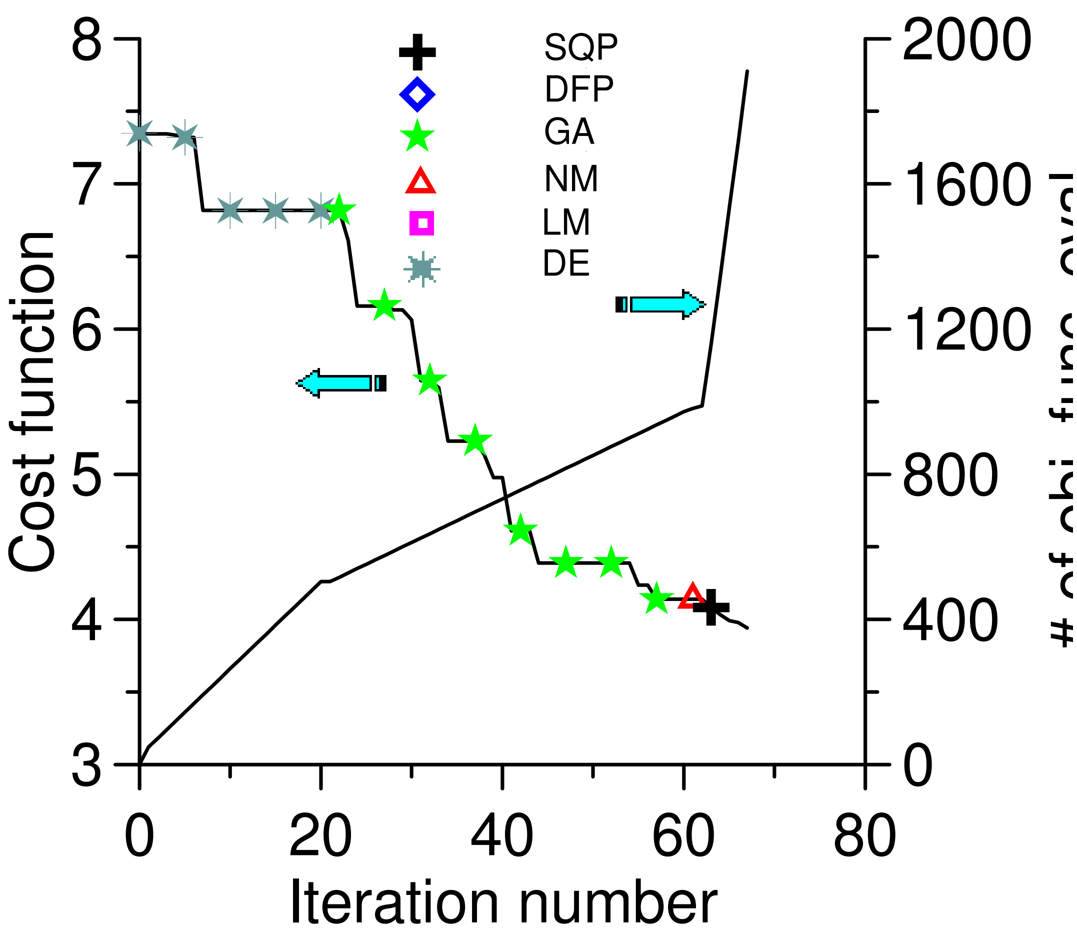
  Optimized boundary conditions



MHD – Example 2

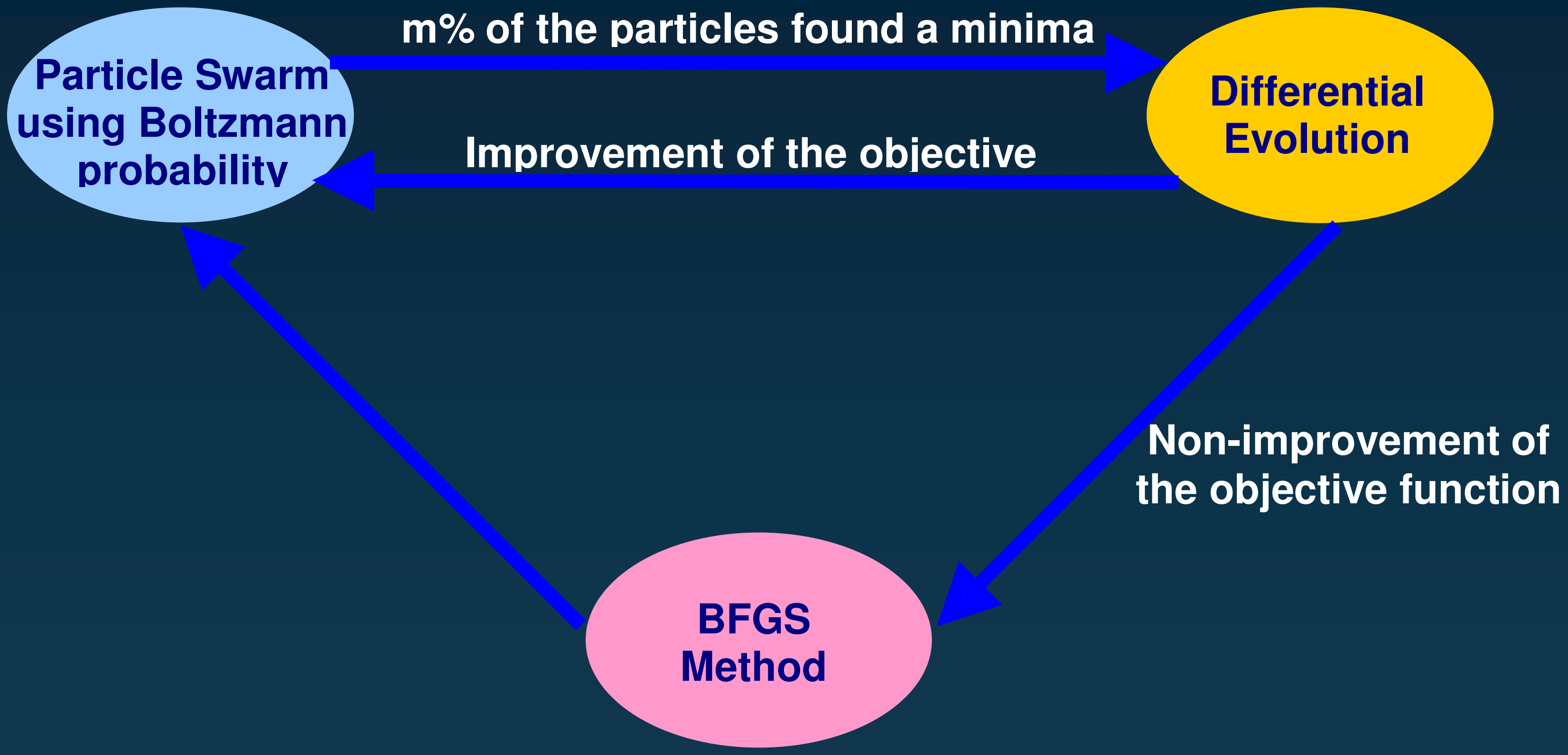
- ➔ Optimization Problem: Test case 2 (with solidification), 6 parameters per boundary
- ➔ Convergence history

DE and GA did almost all the job



Hybrid Methods

- Hybrid optimizer – version 2 (Colaço and Dulikravich)



Particle Swarm using Boltzmann probability

m% of the particles found a minima

Differential Evolution

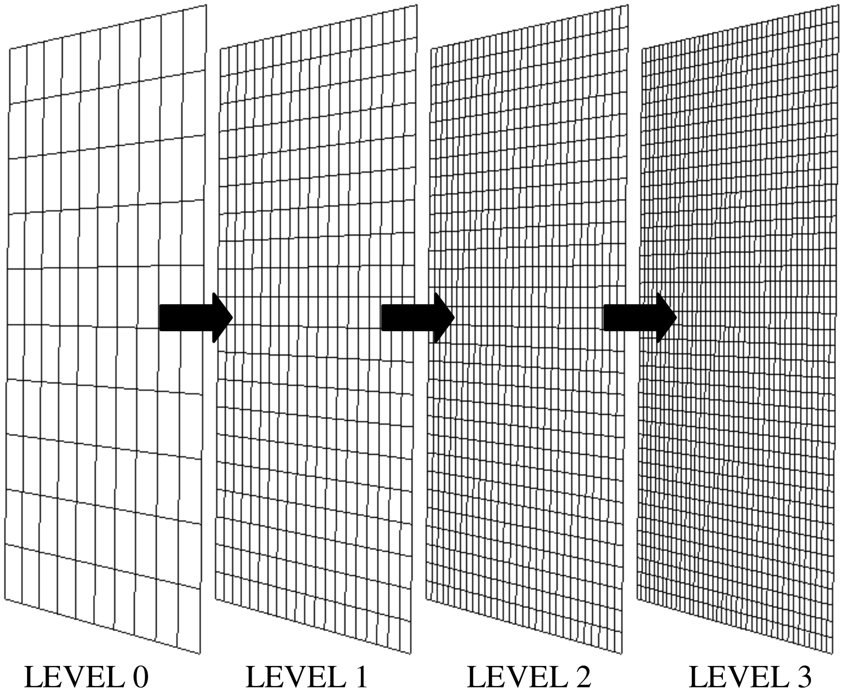
Improvement of the objective

Non-improvement of the objective function

BFGS Method

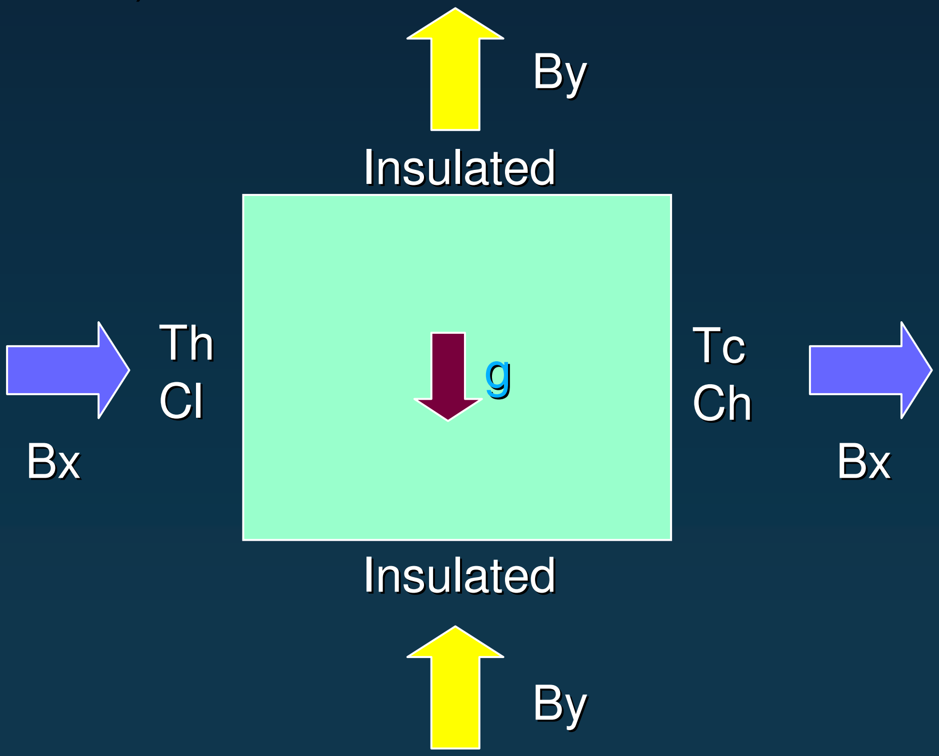
Hybrid Methods

- ➔ Multilevel strategy
  - Uses different grid sizes in order to speed-up the optimization task.



MHD – Example 3

- ➔ Optimization of the magnetic boundary conditions in order to minimize the natural convection effects in a fluid flow with a dispersed solute without solidification (multilevel optimization)
- ➔ Test problem:



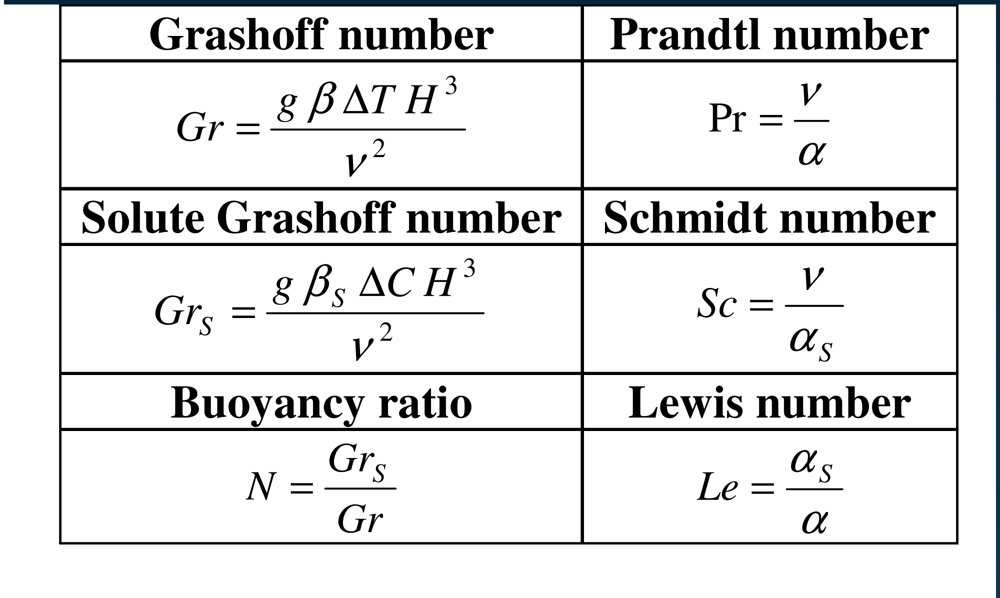
MHD – Example 3

- Optimization Problem
- The boundary conditions are parameterized as B-Splines.
- NETLIB's subroutine GCVSPL, based on the cross-validation smoothing procedure, was used for the interpolation.
- Objective function:

$$F = \sqrt{\frac{1}{\#cells} \sum_{i=1}^{\#cells} \left(\frac{\partial C_i}{\partial y_i} \right)^2}$$

MHD – Example 3

- Optimization Problem
- Parameters for silicon:



Grashoff number	Prandtl number
$Gr = \frac{g\,\beta\,\Delta T\,H^3}{\nu^2}$	$\Pr = \frac{\nu}{\alpha}$
Solute Grashoff number	**Schmidt number**
$Gr_S = \frac{g\,\beta_S\,\Delta C\,H^3}{\nu^2}$	$Sc = \frac{\nu}{\alpha_S}$
Buoyancy ratio	**Lewis number**
$N = \frac{Gr_S}{Gr}$	$Le = \frac{\alpha_S}{\alpha}$

$Gr = 10^5$

$Pr=0.054$

$Le = 2$

$Grs = 5 \times 10^5$

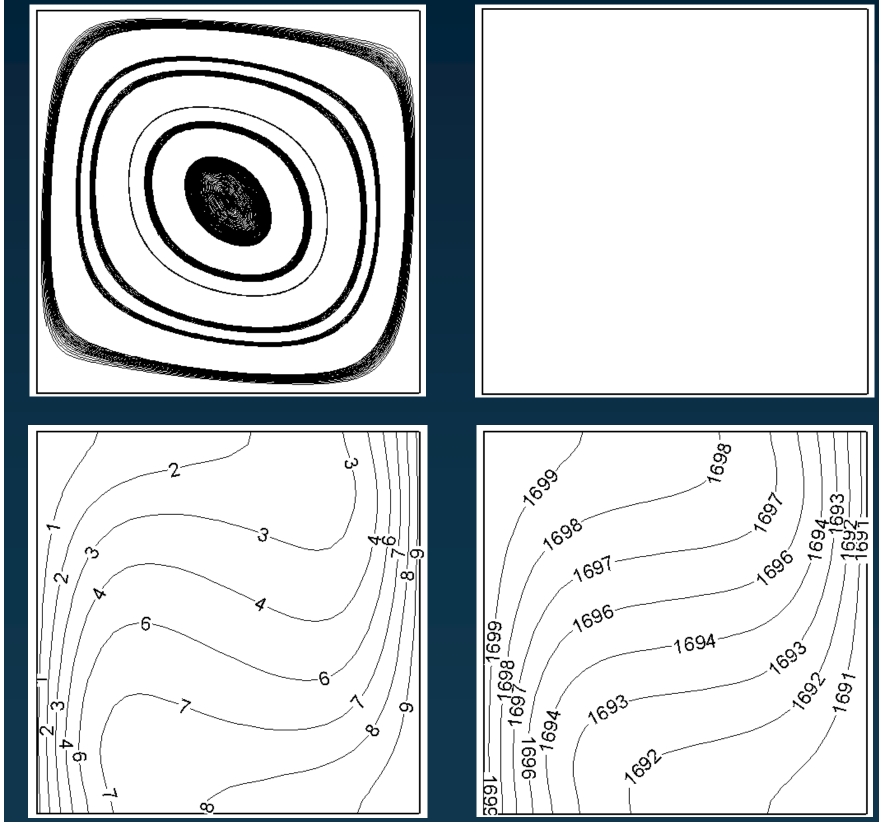
$N = 5.0$

$Sc = 0.108$

- Test case without solidification → H=23 m, ΔT=10 K, ΔT=10 kg / m^3

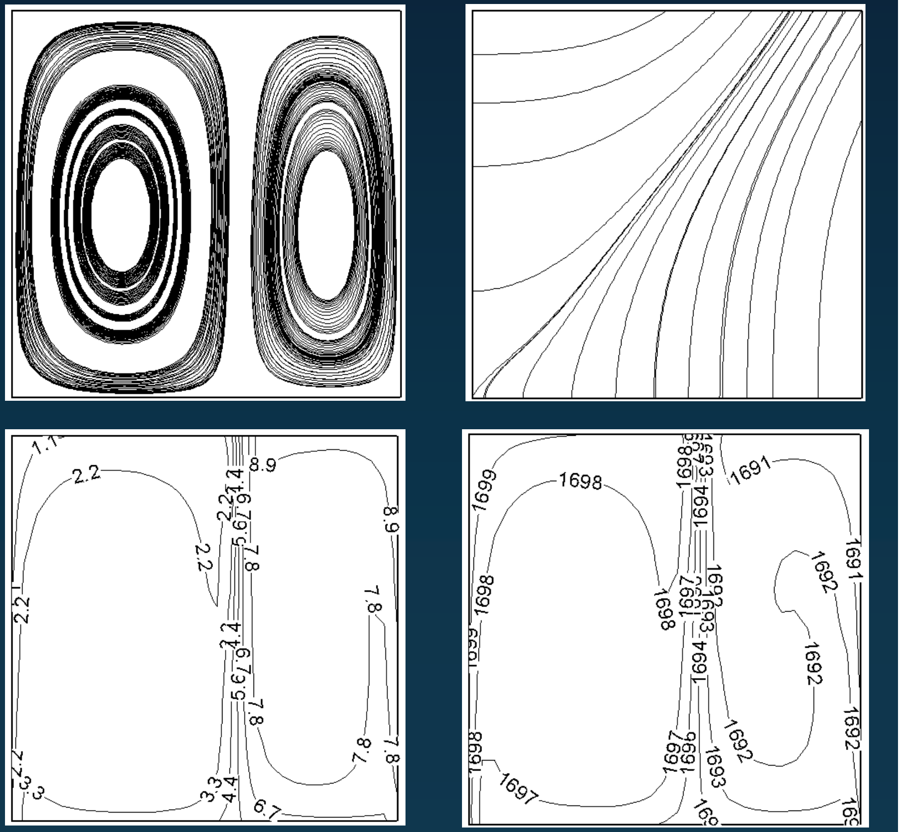
MHD – Example 3

- Optimization Problem:Test case 1(singlelevel optimization),6 parameters/boundary



With optimized magnetic fields

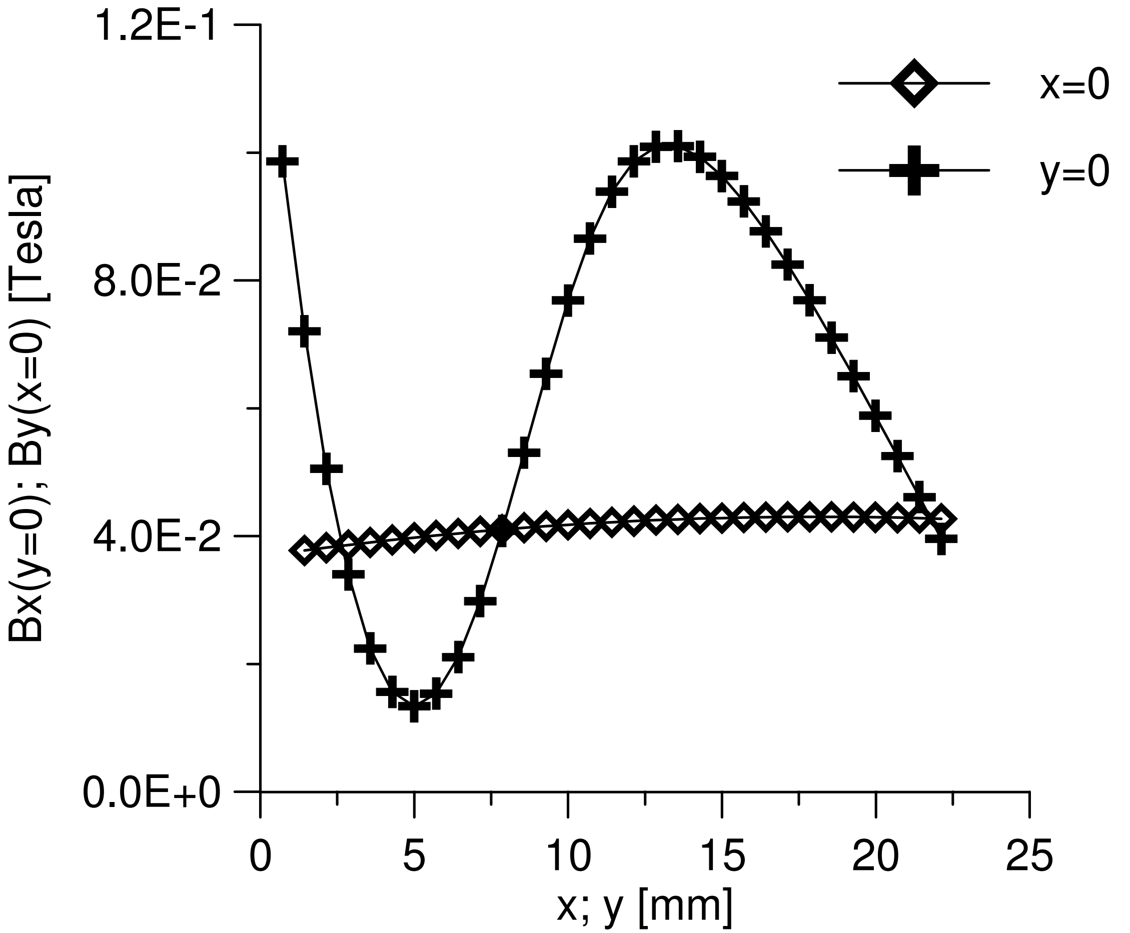
Without magnetic fields



MHD – Example 3

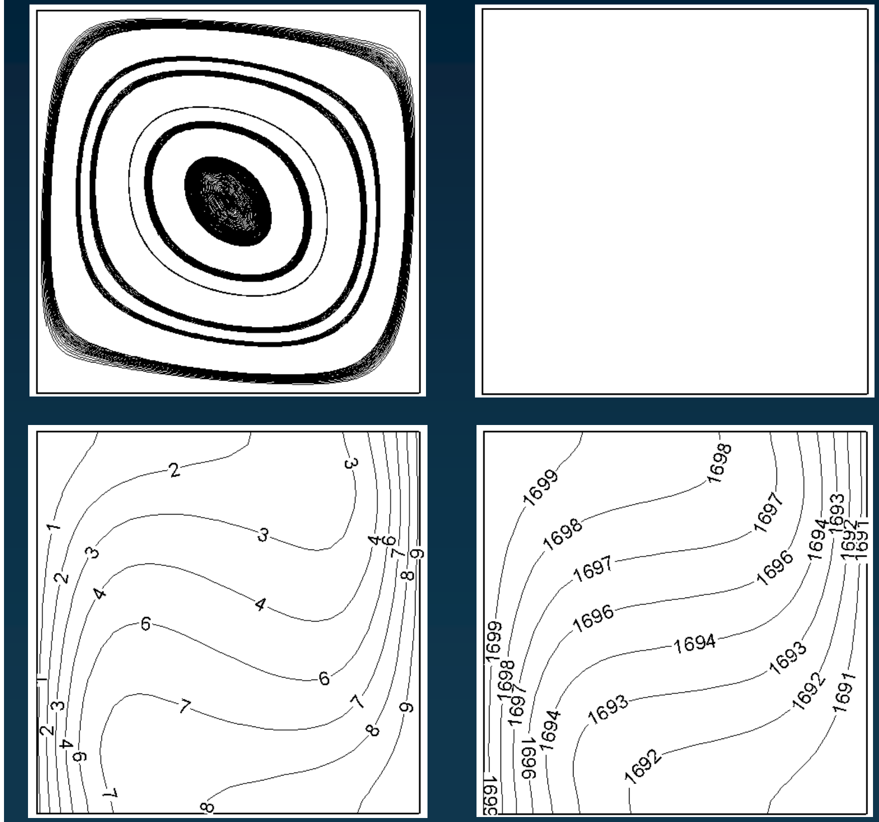
- Optimization Problem:Test case 1(singlelevel optimization),6 parameters/boundary

  Optimized boundary conditions

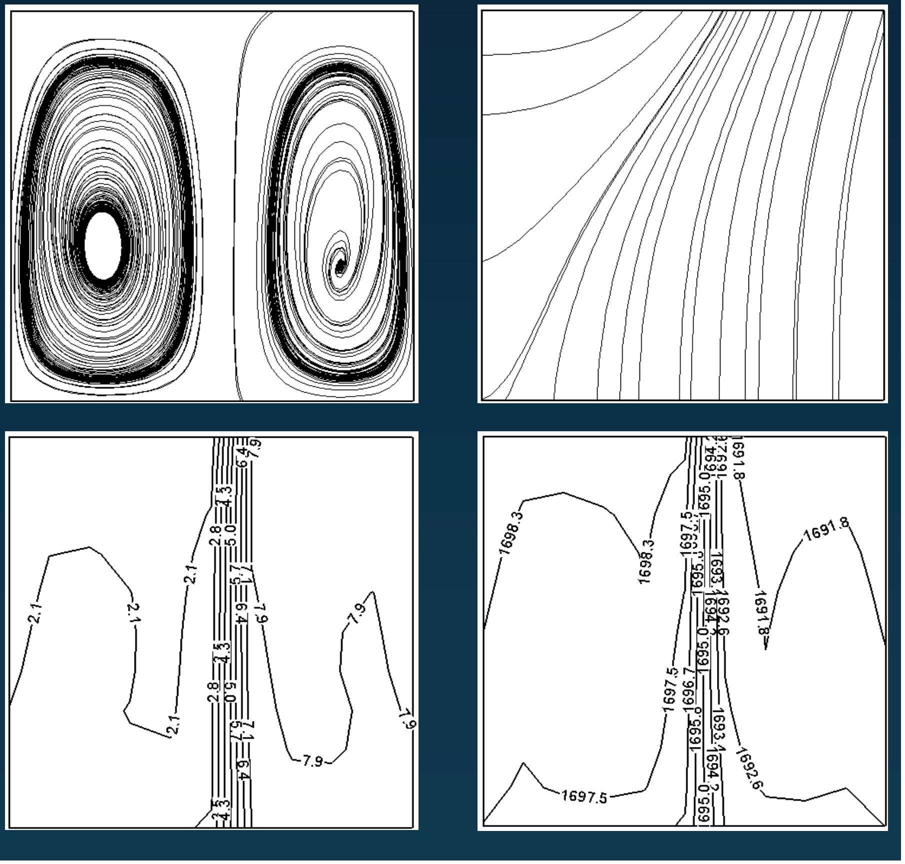


MHD – Example 3

- ➔ Optimization Problem:Test case 2(multilevel optimization),6 parameters/boundary



Without magnetic fields

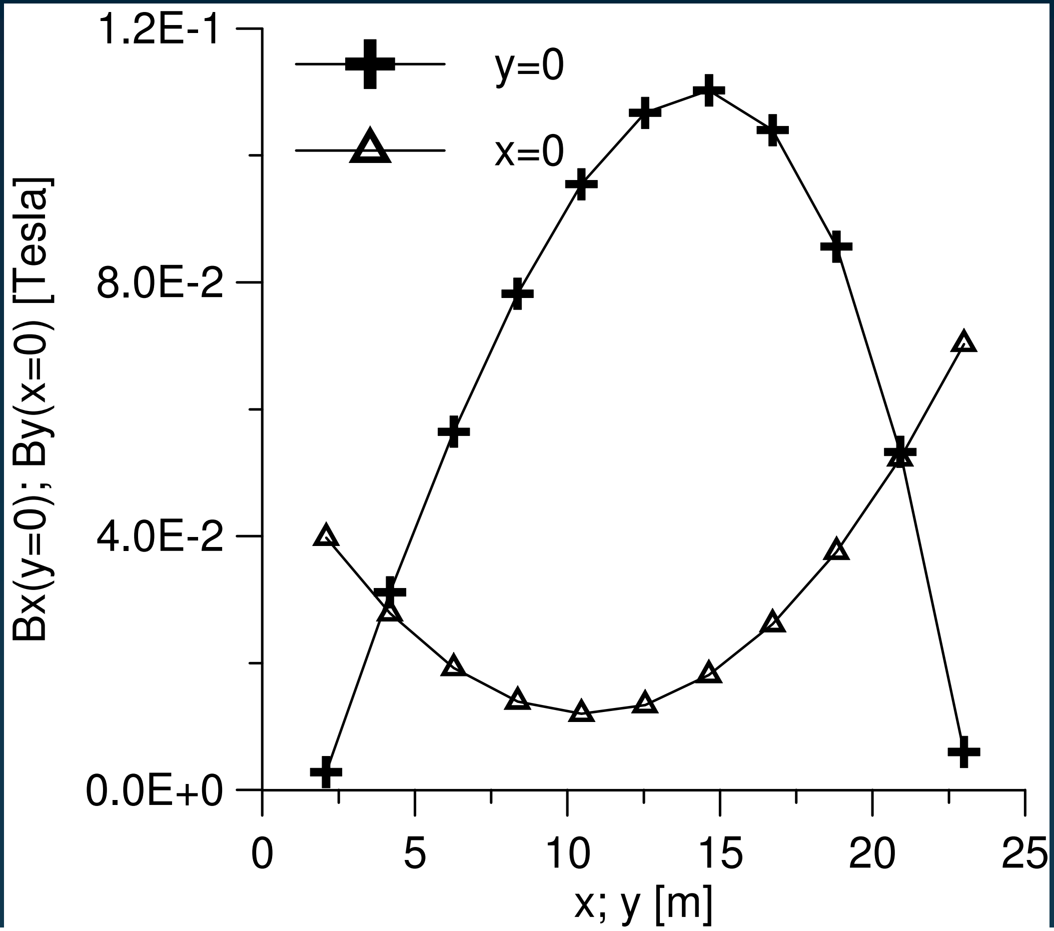


With optimized magnetic fields

MHD – Example 3

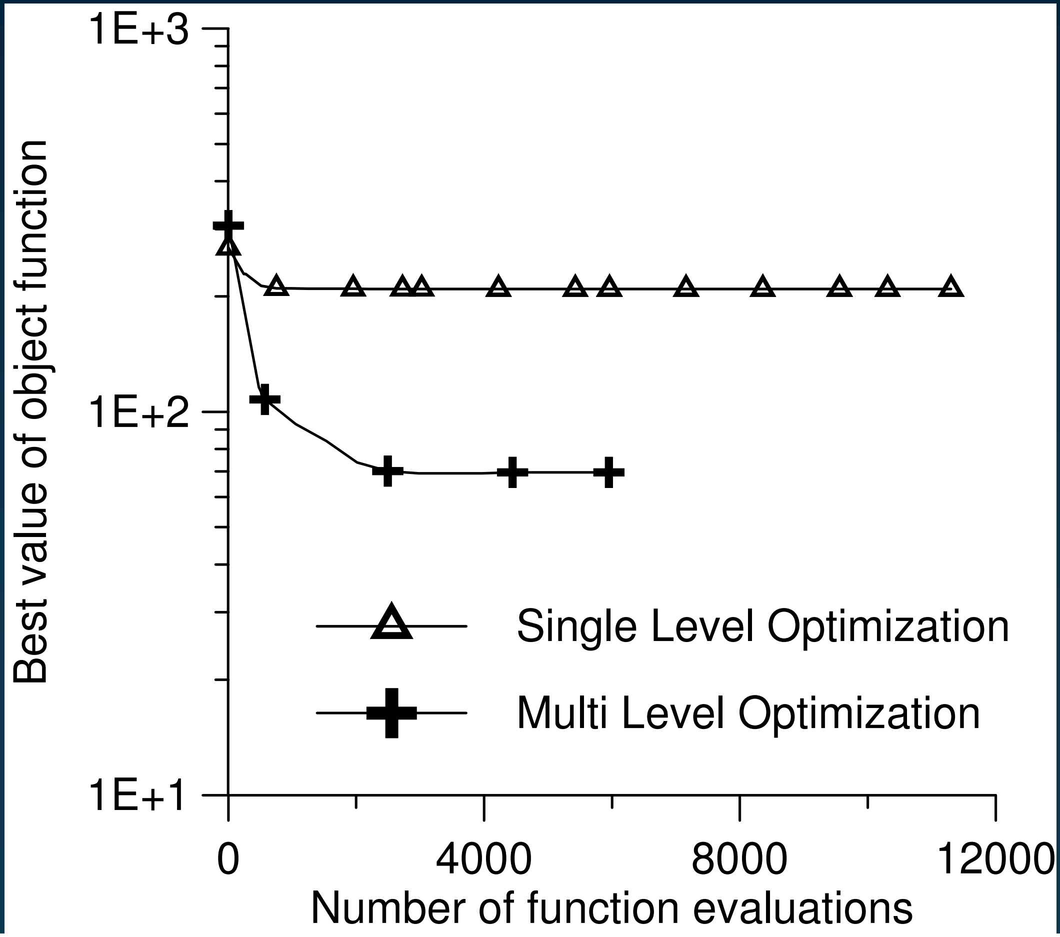
- ➔ Optimization Problem:Test case 2(multilevel optimization),6 parameters/boundary

  Optimized boundary conditions



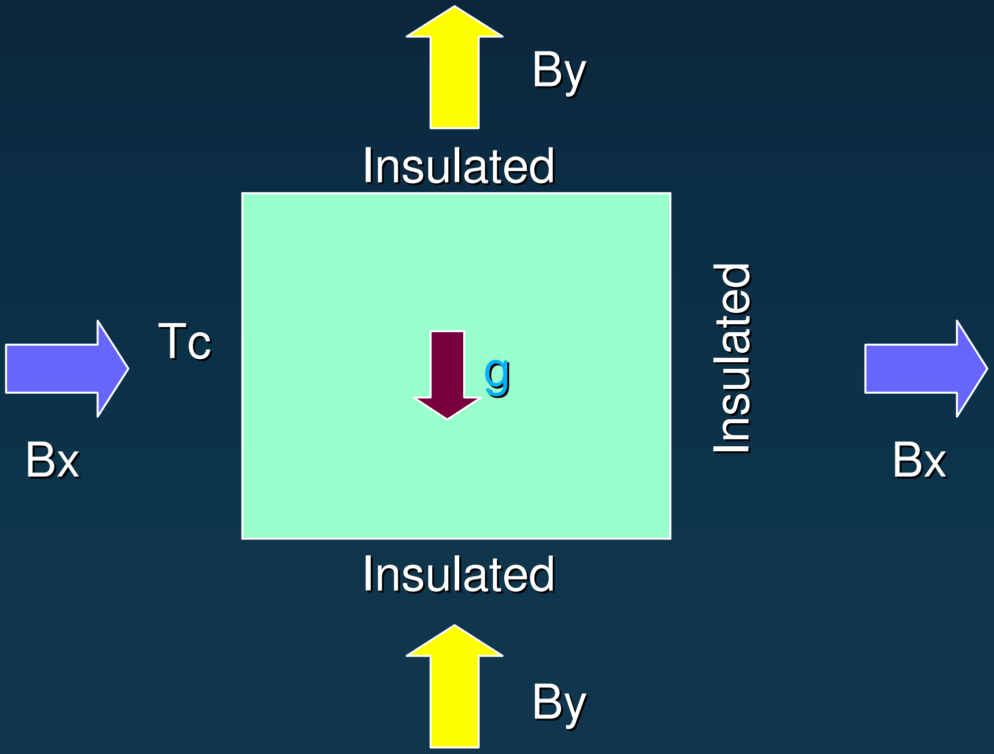
MHD – Example 3

- ➔ Optimization Problem: Singlelevel versus Multilevel optimization
- ➔ Convergence history



MHD – Example 4

- Optimization of the magnetic boundary conditions in order to minimize the natural convection effects in a fluid flow with a dispersed solute (transient with phase change)
- Test problem:



MHD – Example 4

- Optimization Problem
- The boundary conditions are parameterized as B-Splines.
- NETLIB's subroutine GCVSPL, based on the cross-validation smoothing procedure, was used for the interpolation.
- Objective function:

$$F = \sqrt{\frac{1}{\#cells} \sum_{i=1}^{\#cells} \left(\frac{\partial C_i}{\partial y_i} \right)^2}$$

MHD – Example 4

- Optimization Problem
- Parameters for silicon:

$\rho_l = 2550$ kg m^{-3}
$C_{\mathrm{Pl}} = 1059$ J kg^{-1} K^{-1}
$\sigma_l = 12.3 \times 10^5$ 1/m Ω
$D_l = 6.043 \times 10^{-9}$ kg m^{-1} s^{-1}
L=1.8 x 10^6 J/kg

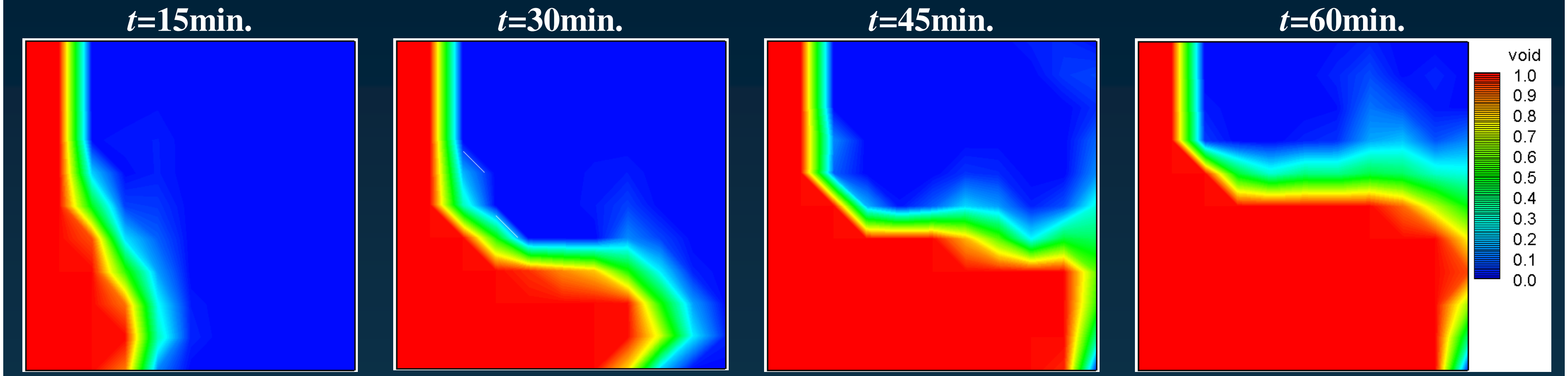
$\rho_s = 2550$ kg m^{-3}
$C_{\mathrm{Ps}} = 1059$ J kg^{-1} K^{-1}
$\sigma_s = 4.3 \times 10^4$ 1/m Ω
$D_s = 0$ kg m^{-1} s^{-1}
To=1685.04 K

$k_l = 64$ W m^{-1} K^{-1}
$\mu_l = 0.0032634$ kg m^{-1} s^{-1}
$\beta = 1.4 \times 10^{-4}$ K^{-1}
$g = 9.81$ m s^{-2}
Tc=1624.96 K

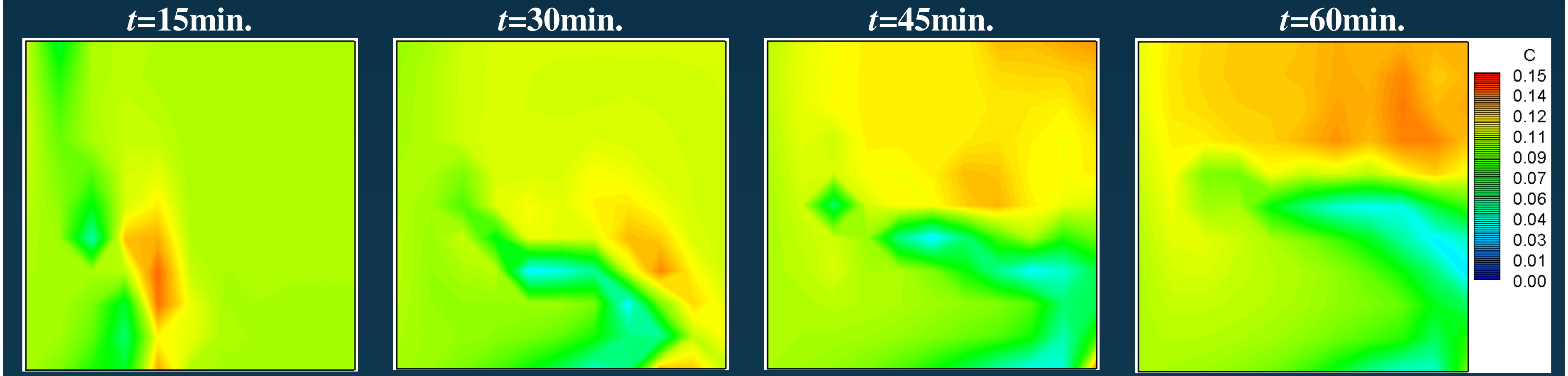
$k_s = 64$ W m^{-1} K^{-1}
$\mu_s = 326.34$ kg m^{-1} s^{-1}
$\beta_s = 0.0875$
$\mu_{\mathrm{m}} = 1.2566 \times 10^{-5}$ T m A^{-1}
Co=0.1 kg m^{-3}

MHD – Example 4

- ➔ Without Magnetic Fields



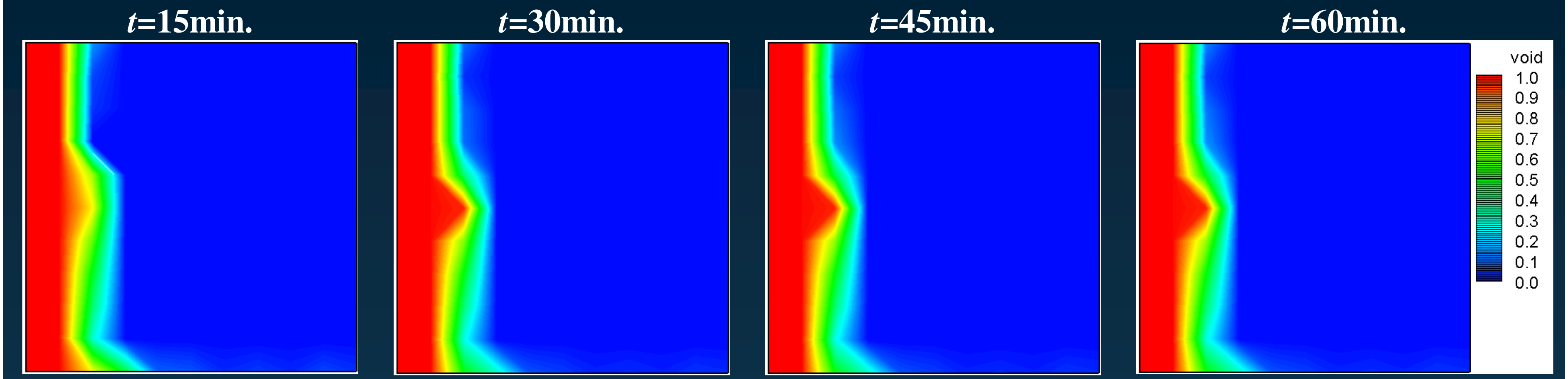
(a) iso-void fraction lines



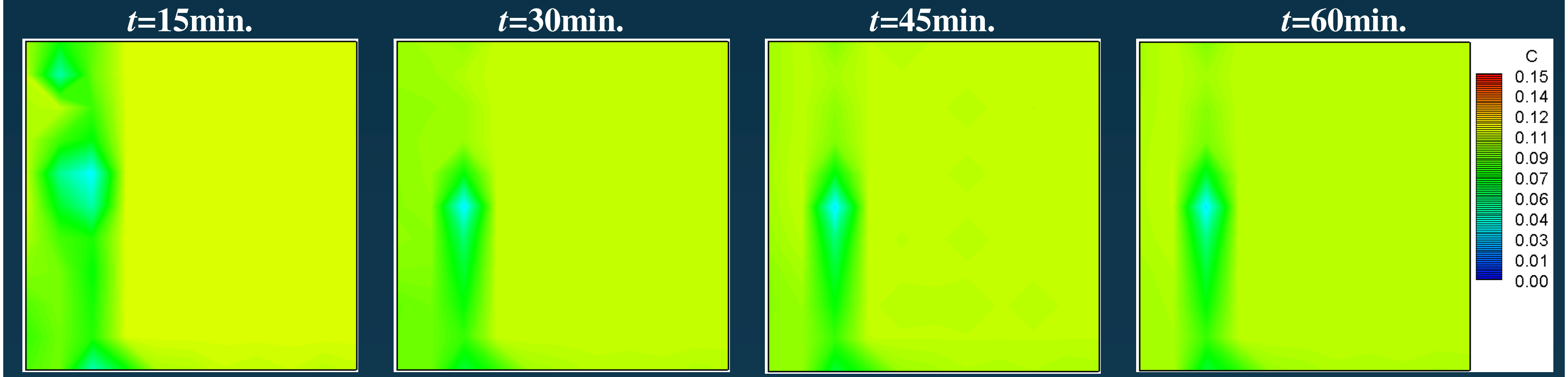
(b) iso-concentration lines

MHD – Example 4

- ➔ With Optimized Magnetic Fields


t=15min. *t*=30min. *t*=45min. *t*=60min.

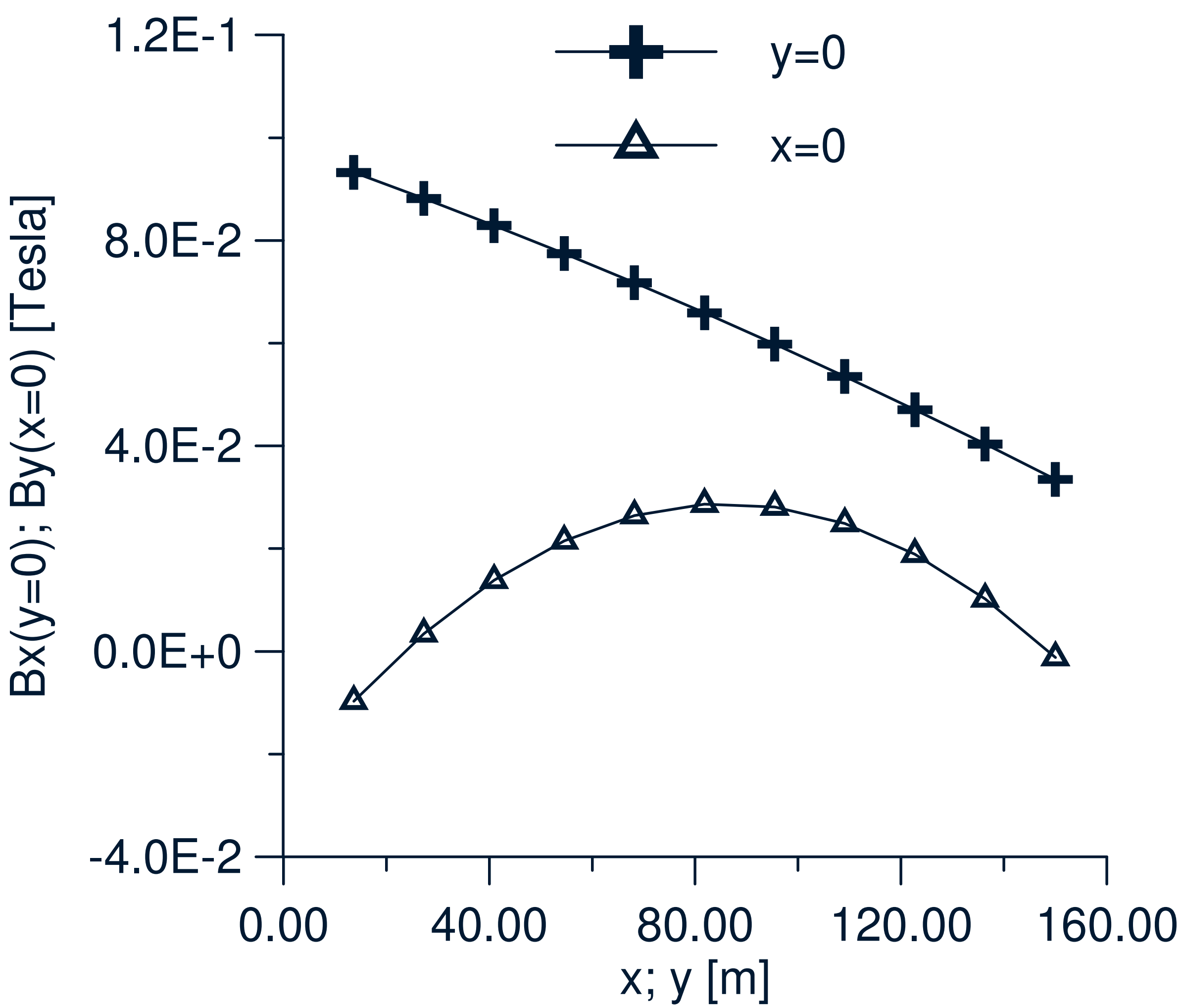
(a) iso-void fraction lines


t=15min. *t*=30min. *t*=45min. *t*=60min.

(b) iso-concentration lines

MHD – Example 4

- ➔ Optimized boundary conditions



Electrohydrodynamic

Direct Problem - EHD

- Conservation equations in the Cartesian coordinate system (x,y)

$$\frac{\partial Q}{\partial t}+\frac{\partial E}{\partial x}+\frac{\partial F}{\partial y}=S$$

where

$$Q=\lambda\phi$$

$$E=(\lambda u+\zeta E_x)\phi-\Gamma\frac{\partial\phi^*}{\partial x}$$

$$F=(\lambda v+\zeta E_y)\phi-\Gamma\frac{\partial\phi^*}{\partial y}$$

$$\mathbf{E}=-\nabla\varphi$$

Conservation of	λ	ζ	ϕ	ϕ^*	Γ	S
Mass	ρ	0	1	1	0	0
x-momentum	ρ	0	u	u	μ	$-\frac{\partial P}{\partial x}+q_e E_x$
y-momentum	ρ	0	v	v	μ	$-\frac{\partial P}{\partial y}-\rho g[1-\beta(T-T_0)]+q_e E_y$
Energy	ρ	0	h	T	K	$C_P\left[b\left(E_x^2+E_y^2\right)+uE_x+vE_y-D_e\left(E_x\frac{\partial q_e}{\partial x}+E_y\frac{\partial q_e}{\partial y}\right)\right]$
Electric potential	0	0	0	φ	1	$-\frac{q_e}{\varepsilon_0}$
Electric charged particles distribution	1	b	q_e	q_e	D_e	0

Direct Problem - EHD

- Conservation of mass, momentum and energy in the “computational” coordinate system (ξ,η)

$$\frac{\partial(J\rho\phi)}{\partial t}+\frac{\partial(\tilde{U}\rho\phi)}{\partial\xi}+\frac{\partial(\tilde{V}\rho\phi)}{\partial\eta}=$$

$$\frac{\partial}{\partial\xi}\left\{J\Gamma^{\phi}\left[a\frac{\partial\phi *}{\partial\xi}+d\frac{\partial\phi *}{\partial\eta}\right]\right\}+\frac{\partial}{\partial\eta}\left\{J\Gamma^{\phi}\left[d\frac{\partial\phi *}{\partial\xi}+b\frac{\partial\phi *}{\partial\eta}\right]\right\}+JS$$

where

$$\begin{aligned} a &= \xi_x^2+\xi_y^2 \\ b &= \eta_x^2+\eta_y^2 \\ d &= \xi_x\eta_x+\xi_y\eta_y \end{aligned}$$

$$\begin{aligned} \tilde{U} &= J(u\xi_x+v\xi_y) \\ \tilde{V} &= J(u\eta_x+v\eta_y) \end{aligned}$$

$$\begin{aligned} \xi_x &= \frac{y_\eta}{J} \\ \xi_y &= -\frac{x_\eta}{J} \\ \eta_x &= -\frac{y_\xi}{J} \\ \eta_y &= \frac{x_\xi}{J} \\ J &= x_\xi y_\eta - x_\eta y_\xi \end{aligned}$$

Direct Problem - EHD

- Conservation of the electric potential in the “computational” coordinate system (ξ,η)

$$\frac{\partial}{\partial \xi}\left\{J\Gamma^{\phi}\left[a\frac{\partial \phi^*}{\partial \xi}+d\frac{\partial \phi^*}{\partial \eta}\right]\right\}+\frac{\partial}{\partial \eta}\left\{J\Gamma^{\phi}\left[d\frac{\partial \phi^*}{\partial \xi}+b\frac{\partial \phi^*}{\partial \eta}\right]\right\}+JS=0$$

where

$$a=\xi_x^2+\xi_y^2$$
$$b=\eta_x^2+\eta_y^2$$
$$d=\xi_x\eta_x+\xi_y\eta_y$$

$$\xi_x=\frac{y_\eta}{J}$$
$$\xi_y=-\frac{x_\eta}{J}$$
$$\eta_x=-\frac{y_\xi}{J}$$
$$\eta_y=\frac{x_\xi}{J}$$
$$J=x_\xi y_\eta-x_\eta y_\xi$$

Direct Problem - EHD

- Conservation of the electric charged particles distribution in the “computational” coordinate system (ξ,η)

$$\frac{\partial(J\rho\phi)}{\partial t}+\frac{\partial\left(\tilde{U}\rho\phi\right)}{\partial\xi}+\frac{\partial\left(\tilde{V}\rho\phi\right)}{\partial\eta}=$$

$$\frac{\partial}{\partial\xi}\left\{J\Gamma^{\phi}\left[a\frac{\partial\phi^{*}}{\partial\xi}+d\frac{\partial\phi^{*}}{\partial\eta}\right]\right\}+\frac{\partial}{\partial\eta}\left\{J\Gamma^{\phi}\left[d\frac{\partial\phi^{*}}{\partial\xi}+b\frac{\partial\phi^{*}}{\partial\eta}\right]\right\}+JS$$

where

$$a=\xi_x^2+\xi_y^2$$

$$b=\eta_x^2+\eta_y^2$$

$$d=\xi_x\eta_x+\xi_y\eta_y$$

$$\tilde{U}=J\left[\left(u+bE_x\right)\xi_x+\left(v+bE_y\right)\xi_y\right]$$

$$\tilde{V}=J\left[\left(u+bE_x\right)\eta_x+\left(v+bE_y\right)\eta_y\right]$$

$$\xi_x=\frac{y_\eta}{J}$$

$$\xi_y=-\frac{x_\eta}{J}$$

$$\eta_x=-\frac{y_\xi}{J}$$

$$\eta_y=\frac{x_\xi}{J}$$

$$J=x_\xi y_\eta-x_\eta y_\xi$$

Direct Problem - EHD

- Test-cases analyzed: Gallium Arsenide under natural convection

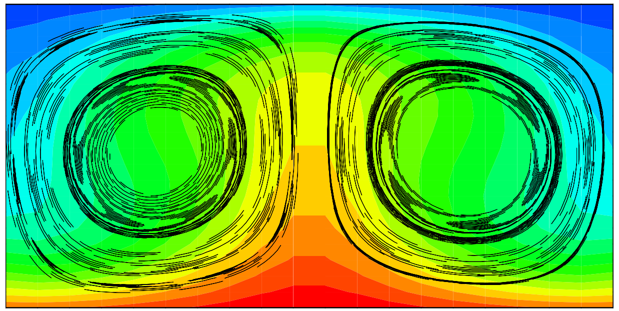
Property	Value
ρ_l	5710 kg/m^3
ρ_s	5196 kg/m^3
C_{Pl}	434 J/kg.K
C_{Ps}	416 J/kg.K
K_l	17.8 W/m.K
K_s	7 W/m.K
b_l	1 x 10^{-8} m^2/V
b_s	1 x 10^{-14} m^2/V
D_{el}	2.5 x 10^{-10} m^2/s
D_{es}	2.5 x 10^{-16} m^2/s

β_l	1.87 x 10^{-4} 1/K
β_s	1.87 x 10^{-4} 1/K
σ_l	8 x 10^5 1/W.m
σ_s	3 x 10^4 1/W.m
ε_0	8.854 x 10^{-12} kg.m/s^{2}V^2
L	726,000 J/kg
μ_l	2.79 x 10^{-3} kg/m.s
μ_s	2.79 x 10^2 kg/m.s
T_l	1511.005 K
T_s	1511 K

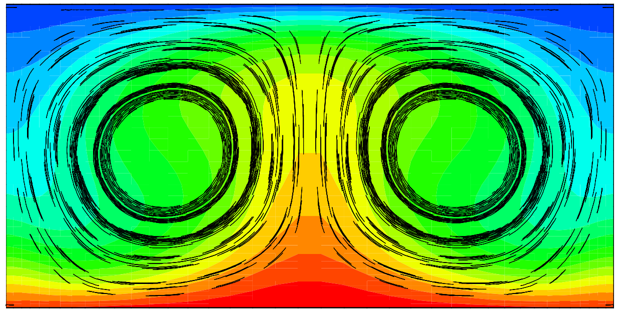
Direct Problem - EHD

- Natural Convection without solidification (Ra=1.9 x 10^4)

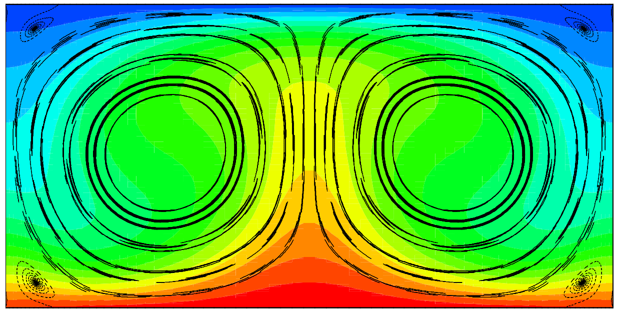
Grid 20x20 cells (9 min.)



Grid 40x40 cells (47 min.)



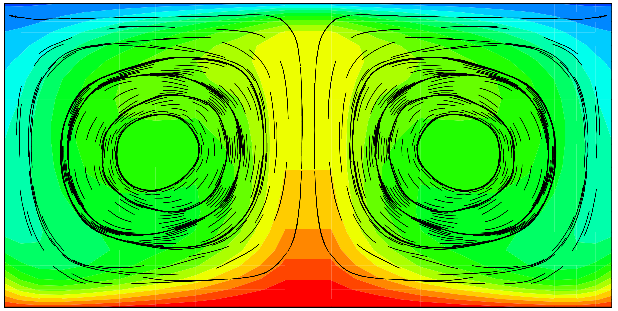
Grid 80x80 (892 min.)



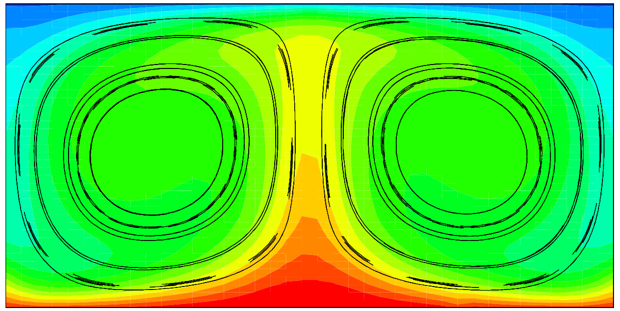
Direct Problem - EHD

- Natural Convection without solidification (Ra=1.9 x 10^5)

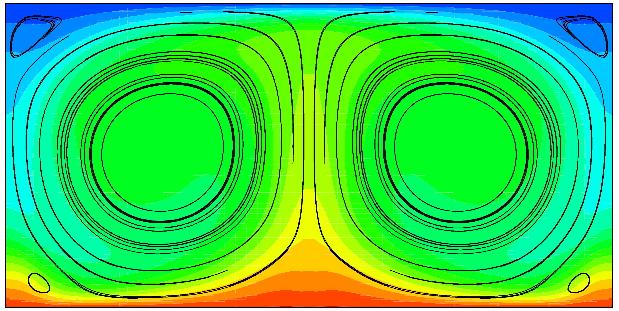
Grid 20x20 cells (8 min.)



Grid 40x40 cells (43 min.)

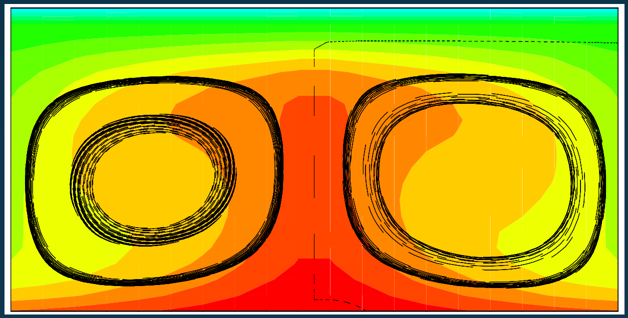


Grid 80x80 (694 min.)

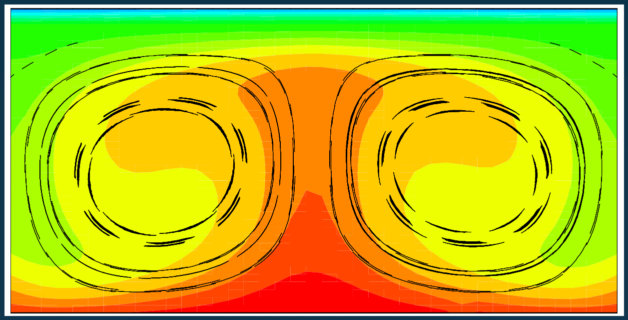


- Natural Convection with solidification (Ra=1.9 x 10^5)

Grid 20x20 cells (15 min.)



Grid 40x40 cells (116 min.)



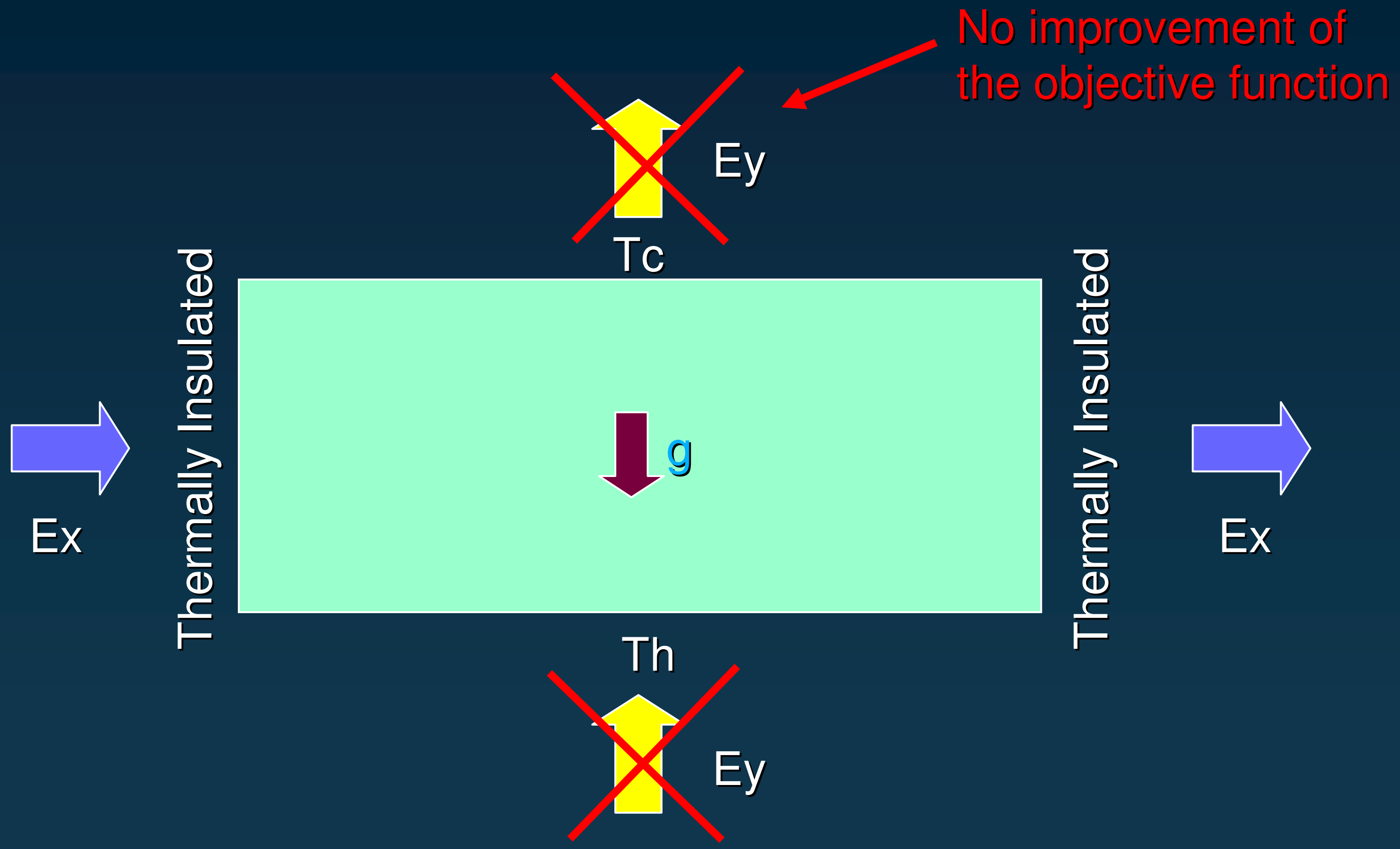
Optimization Problem - EHD

- Motivation
  - Control the natural convection effects by the introduction of an externally applied electric field.
  - Reduce the internal thermal stresses within the solid material
- Cost function to be minimized
  - Reduce the temperature gradients along the *x* coordinate

$$F = \sqrt{\frac{1}{\#cells} \sum_{i=1}^{\#cells} \left(\frac{\partial T_i}{\partial x_i} \right)^2}$$

Optimization Problem - EHD

- Reduce the natural convection effects by applying an externally electric potential

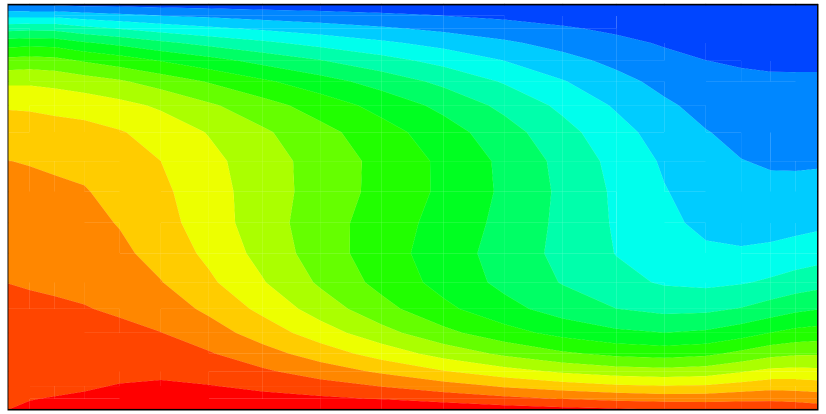


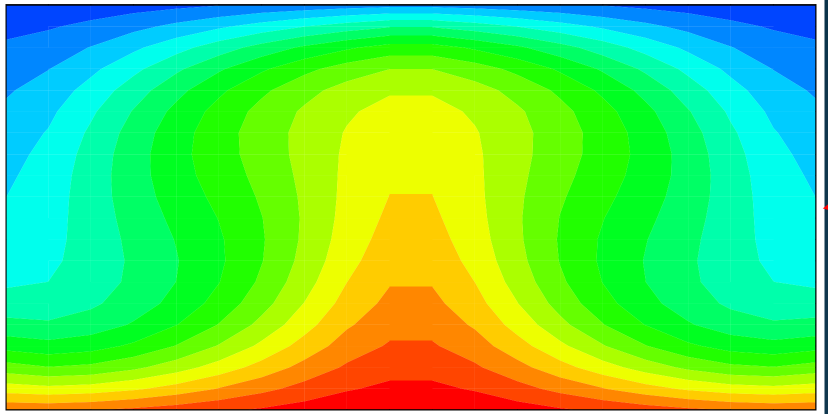
Optimization Problem - EHD

➔ Results for the natural convection case without solidification

- 6 parameters per boundary for electric potential discretization
- Ra=1.9x10^4

With optimized electric potential

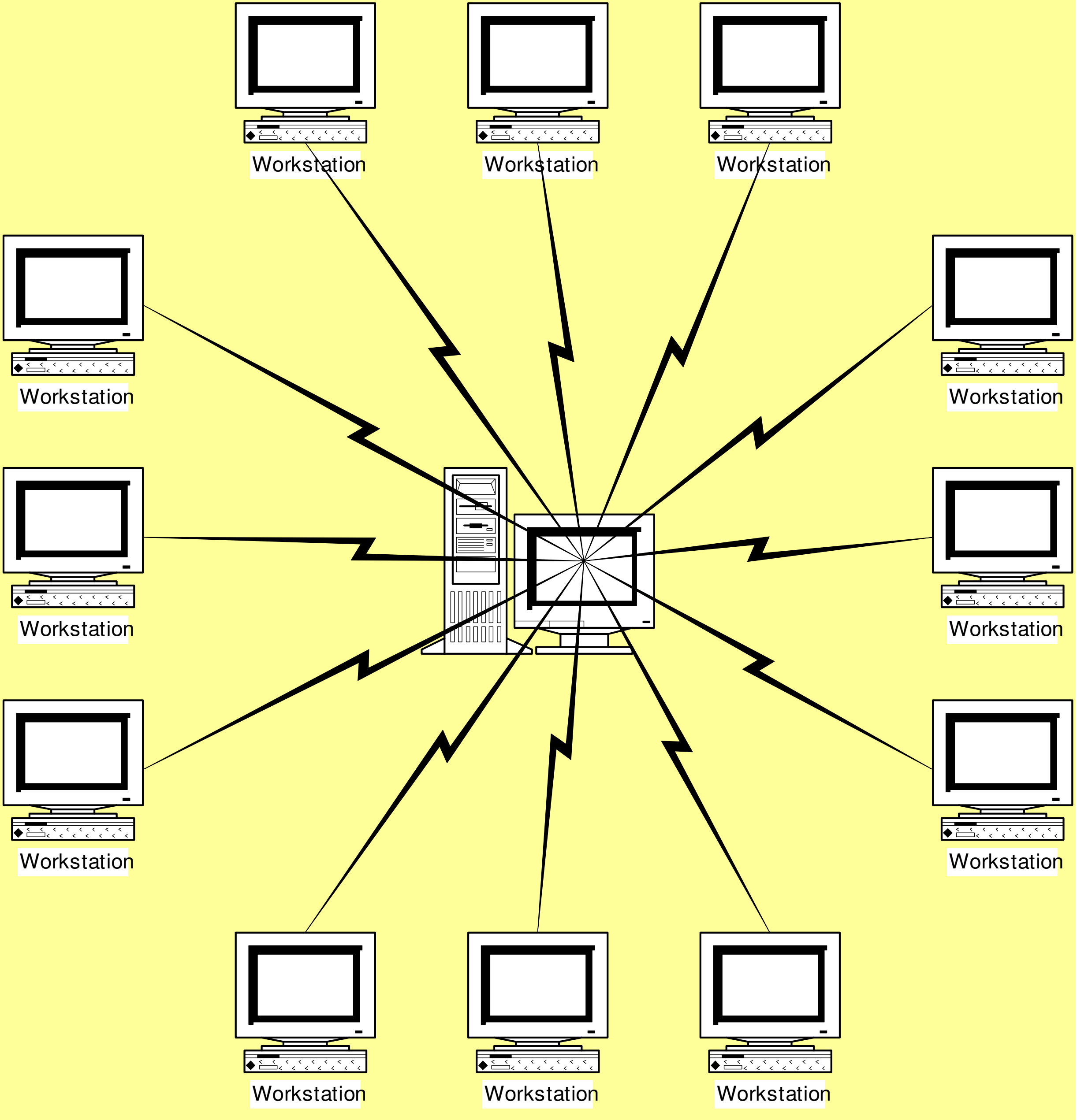




Without electric potential

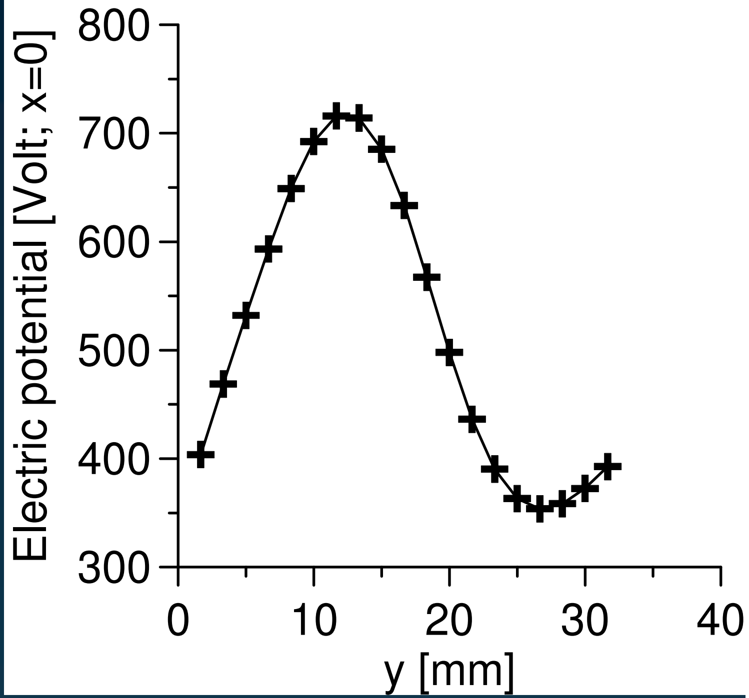
Hybrid Methods

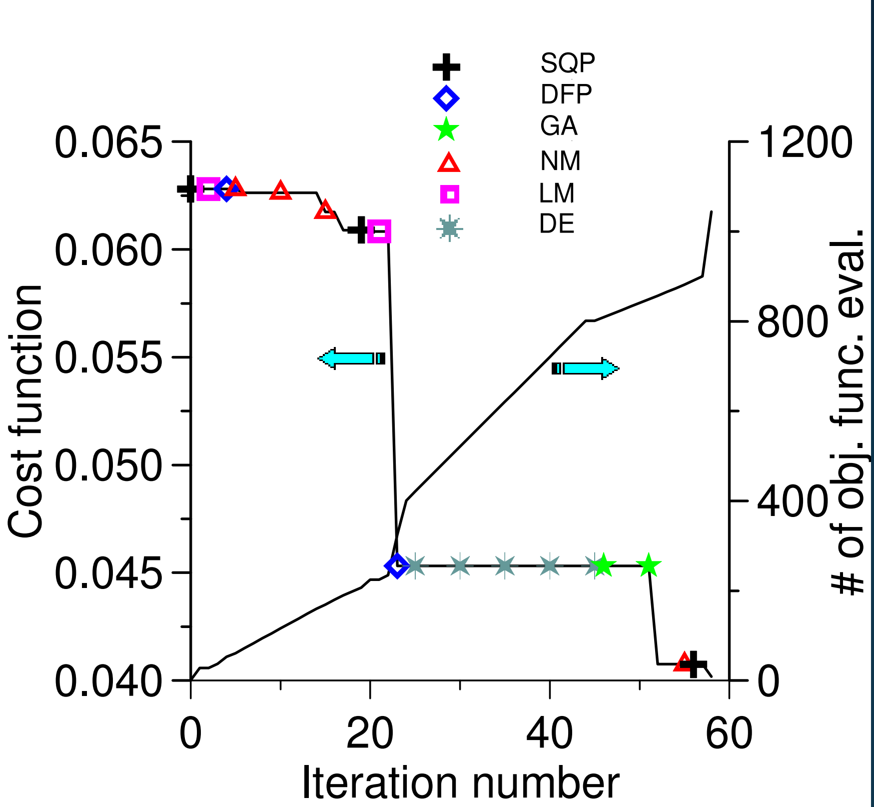
- ➔ Parallel Computing



Optimization Problem - EHD

- Results for the natural convection case without solidification
  - 6 parameters per boundary for electric potential discretization
  - Ra=1.9x10^4





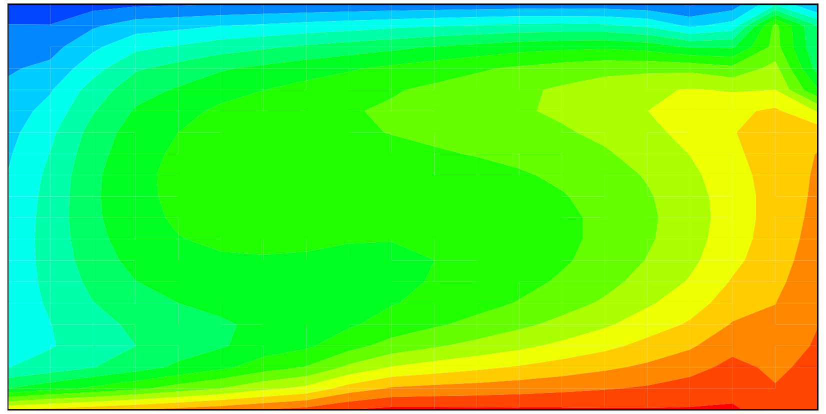
How long it would take, on a single machine, using:
20x20 grid cells = 7 days
40x40 grid cells = 34 days
80x80 grid cells = 647 days

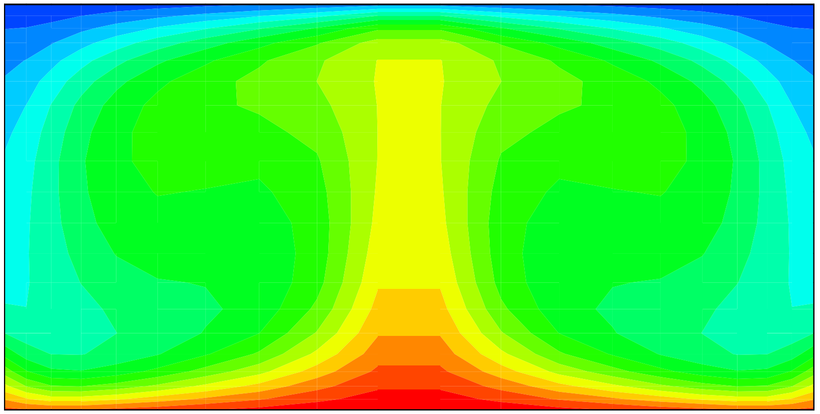
Optimization Problem - EHD

➔ Results for the natural convection case without solidification

- 6 parameters per boundary for electric potential discretization
- $Ra=1.9x10^5$

With optimized electric potential

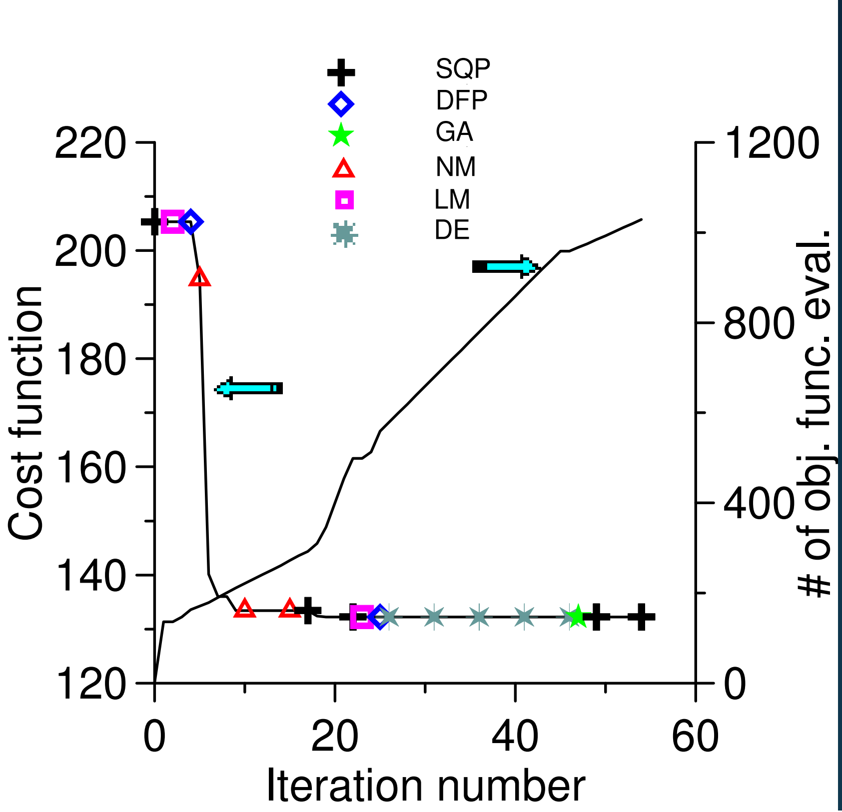


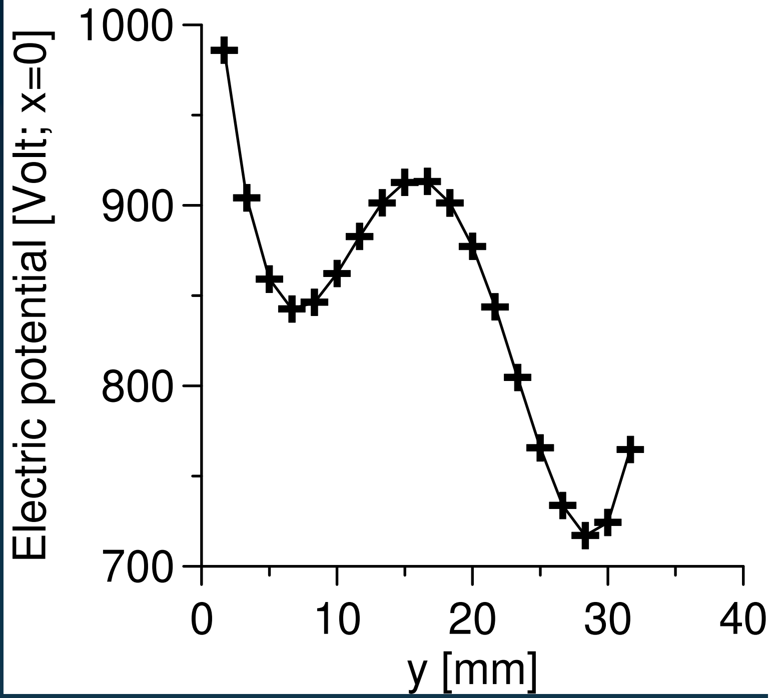


Without electric potential

Optimization Problem - EHD

- ➔ Results for the natural convection case without solidification
  - 6 parameters per boundary for electric potential discretization
  - Ra=1.9x10^5

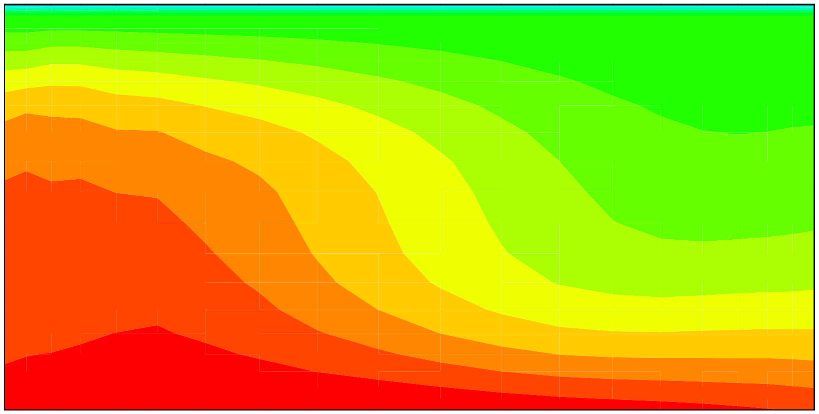




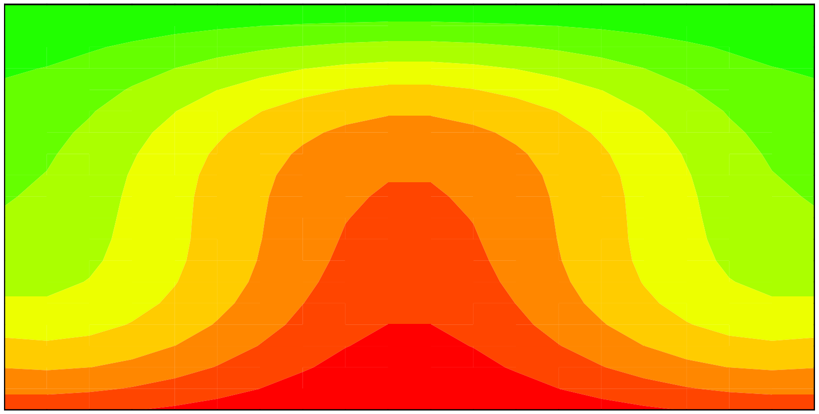
How long it would take, on a single machine, using:
20x20 grid cells = 6 days
40x40 grid cells = 31 days
80x80 grid cells = 496 days

Optimization Problem - EHD

- Results for the natural convection case with solidification
  - 6 parameters per boundary for electric potential discretization
  - $Ra=1.9x10^4$



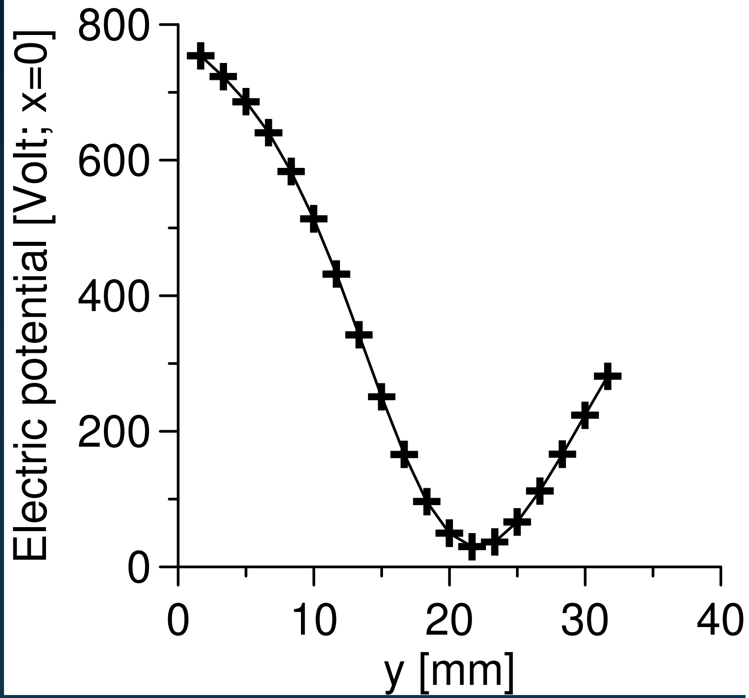
With optimized electric potential

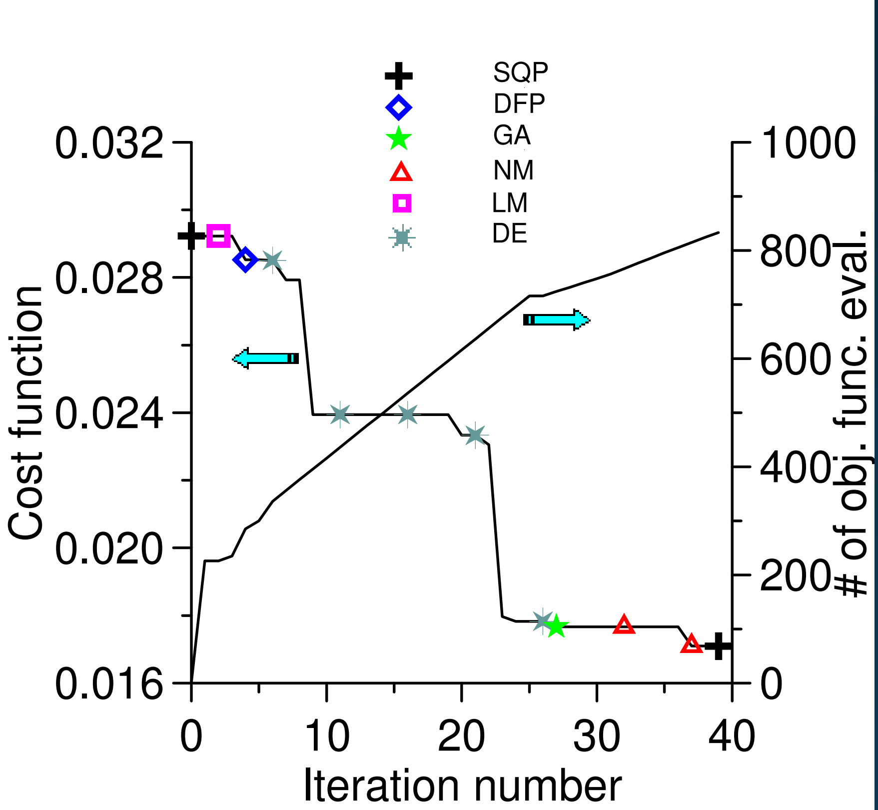


Without electric potential

Optimization Problem - EHD

- Results for the natural convection case with solidification
  - 6 parameters per boundary for electric potential discretization
  - Ra=1.9x10^4





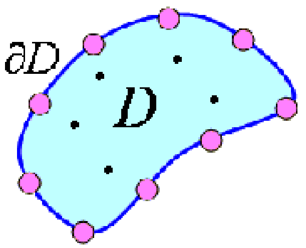
How long it would take, on a single machine, using:
20x20 grid cells = 9 days
40x40 grid cells = 67 days

Radial Basis Function

- Can be classified as a “meshless method”.
- Can be used to generate an interpolation function or solve Partial Differential Equations.
- Original references:
  - Hardy, R.L., 1971, “Multiquadric Equations of Topography and Other Irregular Surfaces”, Journal of Geophysics Res., Vol. 176, pp. 1905-1915.
  - Kansa, E.J., 1990, “Multiquadrics – A Scattered Data Approximation Scheme with Applications to Computational Fluid Dynamics – II: Solutions to Parabolic, Hyperbolic and Elliptic Partial Differential Equations”, Comput. Math. Applic., Vol. 19, pp. 149-161.

Basic Theory

- Kansa's method (or asymmetric collocation) starts by building an approximation to the field of interest from the superposition of radial basis functions conveniently placed at points in the domain (and/or at the boundary).



- The unknowns (which are the coefficients of each RBF) are obtained from the (approximate) enforcement of the boundary conditions as well as the governing equations by means of collocation. Usually, this approximation only considers regular radial basis functions.

Basic Theory

- ➔ The general equation for the RBF has the following form

$$s(x_i) = f(x_i) = \sum_{j=1}^{N} \alpha_j \phi\left(\left|x_i - x_j\right|\right) + \sum_{k=1}^{M} \beta_k p_k(x_i) + \beta_0$$

- ➔ where $f(x_i)$ is known for a series of points x_i and $p_k(x_i)$ is one of the M terms of a given basis of polynomials. This approximation is solved for the α_j unknowns from the system of N linear equations, subject to the conditions (for the sake of uniqueness)

$$\sum_{j=1}^{N} \alpha_j p_k(x_j) = 0$$

$$\sum_{j=1}^{N} \alpha_j = 0$$

Basic Theory

➔ Choices for the basis function

- Globally supported
  - Multiquadrics

$$\phi\left(\left|x-x_j\right|\right)=\sqrt{\left(x-x_j\right)^2+c_j^2}$$

  - Gaussian

$$\phi\left(\left|x-x_j\right|\right)=e^{-c\left(x-x_j\right)^2}$$

  - Polyharmonic splines

$$\phi\left(\left|x-x_j\right|\right)=\begin{cases}\left(x-x_j\right)^{2n}\log\left(x-x_j\right), & n\geq 1, \quad \text{in 2D},\\ \left(x-x_j\right)^{2n-1}, \ n\geq 1, & \quad \text{in 3D}.\end{cases}$$

  - etc

Basic Theory

- Choices for the basis function
  - Locally supported
    - Wendland forms

$$\text{W41:}\quad \phi_i(x,y)=\left(\sqrt{(x-x_i)^2+(y-y_i)^2}-1\right)_+^4\left(1+4\sqrt{(x-x_i)^2+(y-y_i)^2}\right)$$

$$\text{W42:}\quad \phi_i(x,y)=\left(\sqrt{(x-x_i)^2+(y-y_i)^2}-1\right)_+^6\left\{3+18\sqrt{(x-x_i)^2+(y-y_i)^2}+35\left[(x-x_i)^2+(y-y_i)^2\right]\right\}$$

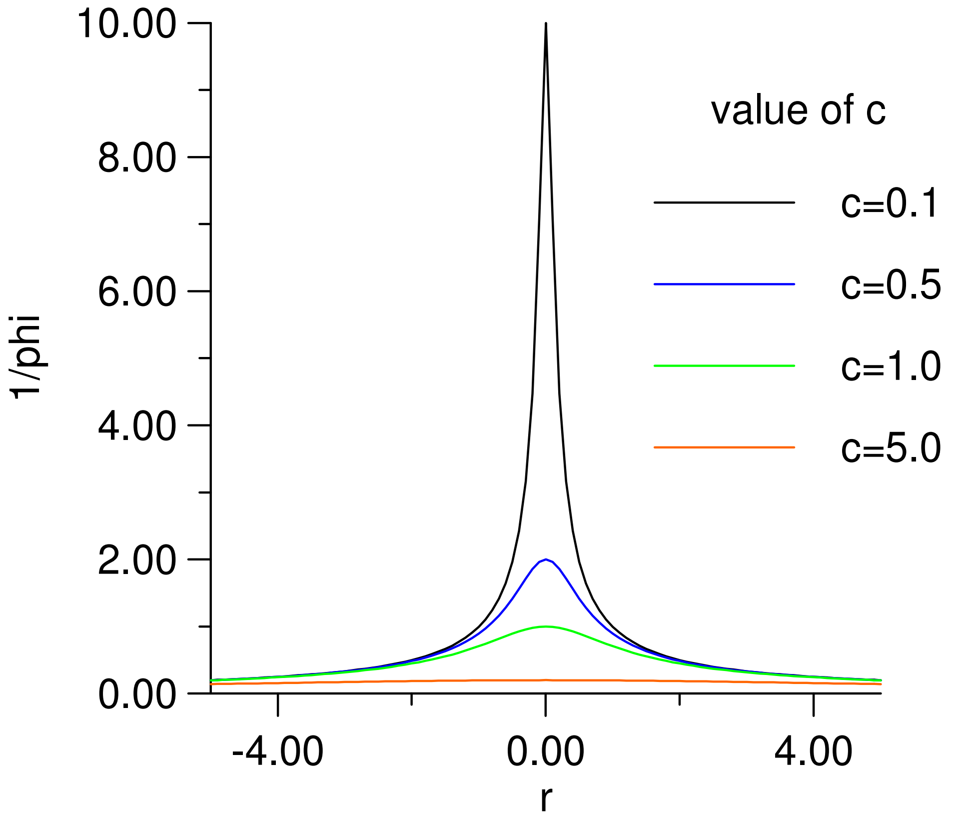
$$\begin{aligned}\text{W43:}\quad \phi_i(x,y)=&\left(\sqrt{(x-x_i)^2+(y-y_i)^2}-1\right)_+^8\\ &\left\{1+8\sqrt{(x-x_i)^2+(y-y_i)^2}+25\left[(x-x_i)^2+(y-y_i)^2\right]+32\left[(x-x_i)^2+(y-y_i)^2\right]^{3/2}\right\}\end{aligned}$$

Basic Theory

- Multiquadrics

$$\phi\left(\left|x-x_j\right|\right)=\sqrt{\left(x-x_j\right)^2+c_j^2}$$

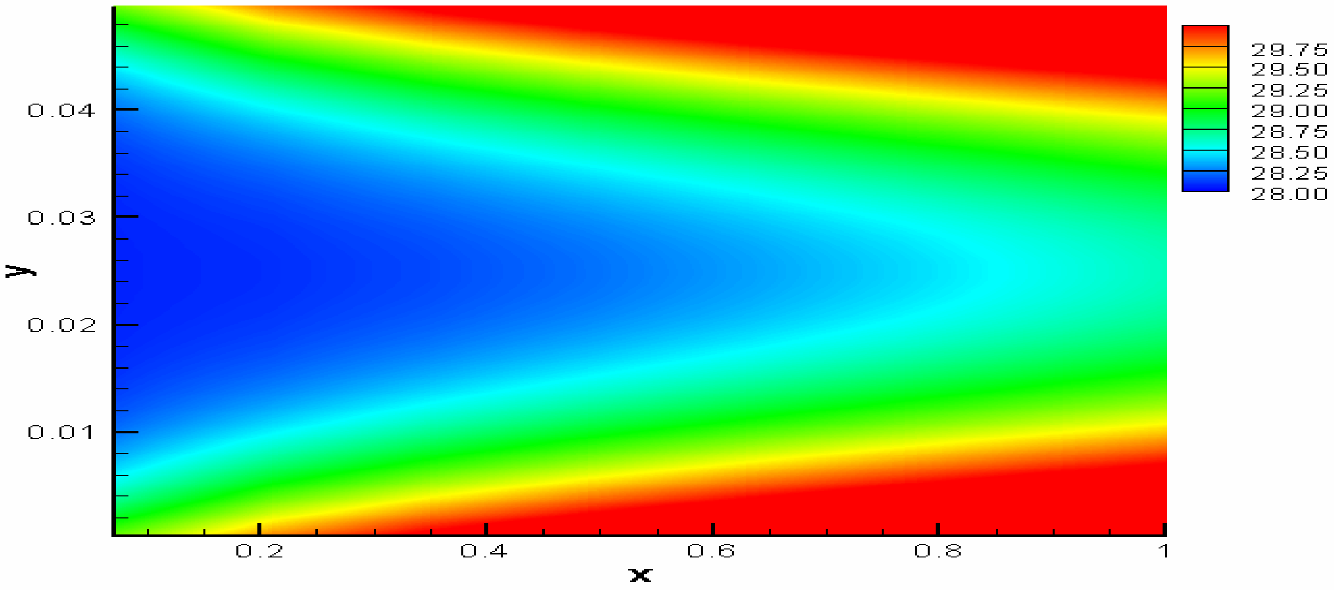
  - Depending on the choice of *c*, the function can be more or less smooth.



$$r=\left(x-x_j\right)$$

Solving PDEs with RBFs

- Linear Problems
  - Example: Thermally developing flow
    - Velocity profile is known
    - Walls subjected to a constant heat flux



Solving PDEs with RBFs

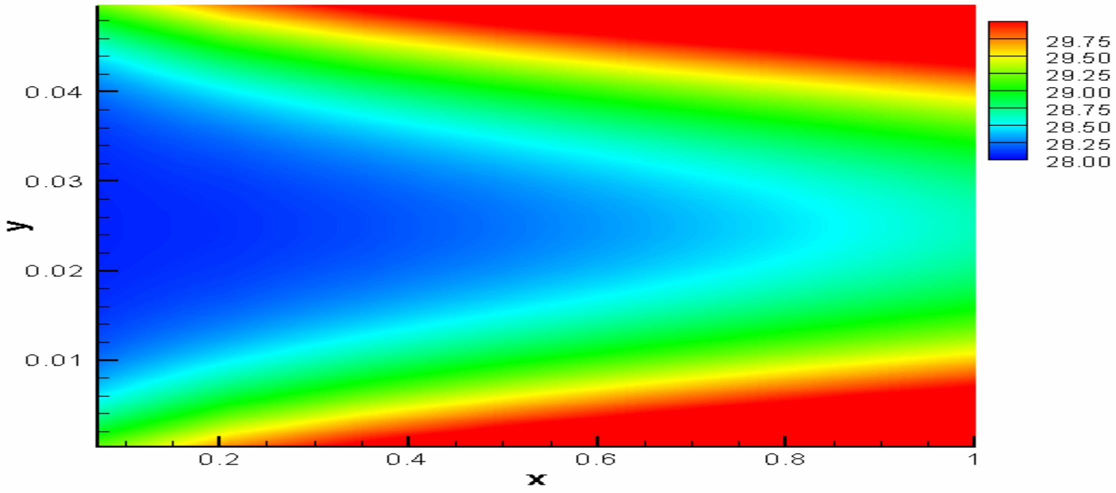
- Linear Problems
  - Example: Thermally developing flow

$$u\frac{\partial T}{\partial x}+v\frac{\partial T}{\partial y}=\frac{k}{\rho\, c_p}\left(\frac{\partial^2 T}{\partial x^2}+\frac{\partial^2 T}{\partial y^2}\right) \quad \text{in } 0<x<L;\ 0<y<h$$

$$T=T_0 \qquad \text{at } x=0;\ 0<y<h$$

$$k\frac{\partial T}{\partial y}=q \qquad \text{at } y=0;\ 0<x<L$$

$$-k\frac{\partial T}{\partial y}=q \qquad \text{at } y=h;\ 0<x<L$$



Solving PDEs with RBFs

➔ Linear Problems

- Example: Thermally developing flow
  - RBF Approximation $T(x,y)=\sum_{i=1}^{N}\phi_i\psi(\mathbf{r}_i)$ $\psi_i(x,y)=\left[(x-x_i)^2+(y-y_i)^2+c_i^2\right]^{\beta}$
  - Substituting in the original PDE, we obtain

$$u\sum_{i=1}^{N}\phi_i\frac{\partial\psi(\mathbf{r}_i)}{\partial x}+v\sum_{i=1}^{N}\phi_i\frac{\partial\psi(\mathbf{r}_i)}{\partial y}=\frac{k}{\rho\,c_p}\left[\sum_{i=1}^{N}\phi_i\frac{\partial^2\psi(\mathbf{r}_i)}{\partial x^2}+\sum_{i=1}^{N}\phi_i\frac{\partial^2\psi(\mathbf{r}_i)}{\partial y^2}\right]\quad \text{in } 0<x<L;\ 0<y<h$$

$$\sum_{i=1}^{N}\phi_i\psi(\mathbf{r}_i)=T_0\qquad \text{at } x=0;\ 0<y<h$$

$$k\sum_{i=1}^{N}\phi_i\frac{\partial\psi(\mathbf{r}_i)}{\partial y}=q\qquad \text{at } y=0;\ 0<x<L$$

$$-k\sum_{i=1}^{N}\phi_i\frac{\partial\psi(\mathbf{r}_i)}{\partial y}=q\qquad \text{at } y=h;\ 0<x<L$$

$$\mathbf{\Psi\Phi=\beta}$$

Solving PDEs with RBFs

- Linear Problems
  - Example: Thermally developing flow
    - Comparison with benchmark solution

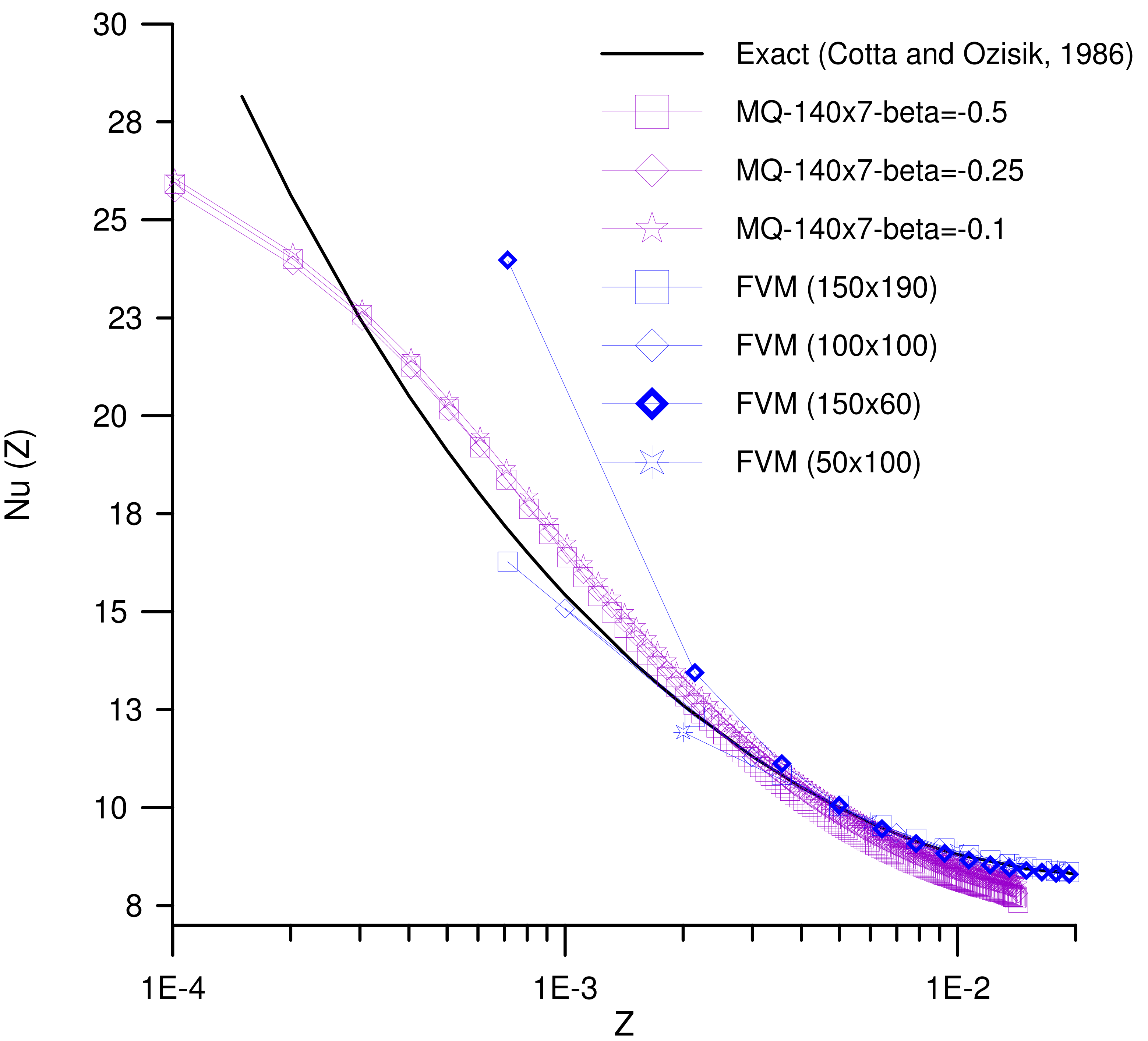
$$Nu(Z) = \frac{2hq}{k\left[T(x, y = h) - T_{av}(x)\right]}$$

$$T_{av} = \frac{\int_0^h u(x, y)T(x, y)dy}{u_{av}h}$$

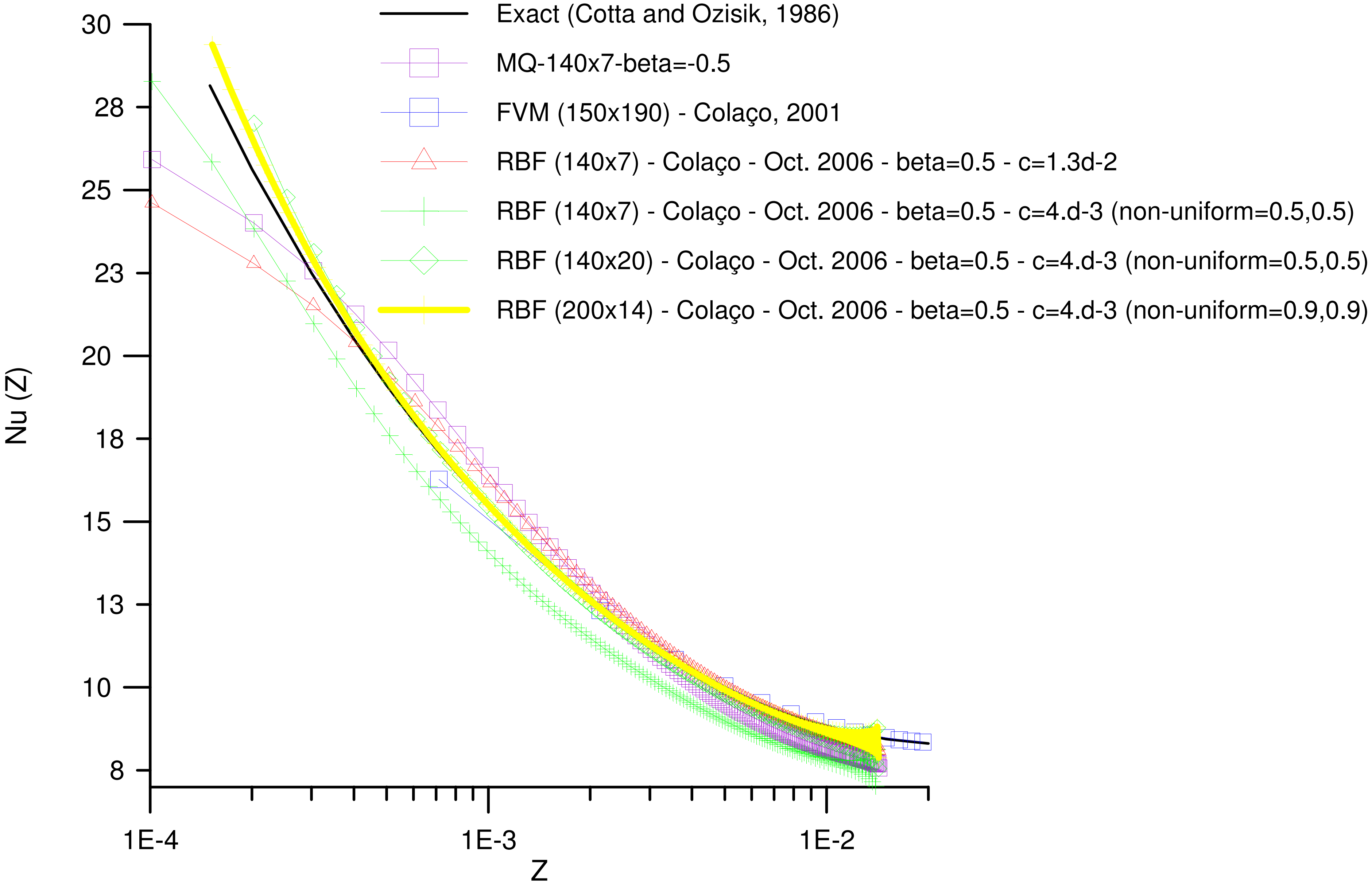
$$Z = \frac{(k / \rho c_p)x}{4u_{av}h^2}$$

Solving PDEs with RBFs

Points in the expansion	Shape parameter - c	Expoent - β	Residual - ε	CPU time (seconds)
80x4	0.003	0.5	7495	0.08
120x6	0.043	-0.5	1295	0.81
	0.011	1.5	41049381	0.79
	0.043	-0.25	1185	0.88
	0.015	0.25	8564	0.78
	0.004	0.5	2311	0.83
100x5	0.006		9790	0.33
140x7	0.024	-0.5	9614	1.96
	0.021	-0.25	2504	1.98
	0.001	0.5	44736	1.96
	0.020	-0.1	3084	1.98
140x8	0.023	-0.5	5158	2.6
160x7	0.038		6575	2.9


30
28
25
23
20
18
15
13
10
8
Nu (Z)
1E-4
1E-3
1E-2
Z
Exact (Cotta and Ozisik, 1986)
MQ-140x7-beta=-0.5
MQ-140x7-beta=-0.25
MQ-140x7-beta=-0.1
FVM (150x190)
FVM (100x100)
FVM (150x60)
FVM (50x100)

Solving PDEs with RBFs



Solving PDEs with RBFs

- Non-Linear Problems
  - Example: Hydrodynamic Developing Flow
    - Velocity profile is unknown
    - Temperature is constant
    - Pouiseulle Flow

$$\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}=0 \quad \text{in 0<x<L; 0<y<h}$$

$$u\frac{\partial u}{\partial x}+v\frac{\partial u}{\partial y}=\frac{\mu}{\rho}\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}\right)-\frac{1}{\rho}\frac{\partial P}{\partial x} \quad \text{in 0<x<L; 0<y<h}$$

$$u\frac{\partial v}{\partial x}+v\frac{\partial v}{\partial y}=\frac{\mu}{\rho}\left(\frac{\partial^2 v}{\partial x^2}+\frac{\partial^2 v}{\partial y^2}\right)-\frac{1}{\rho}\frac{\partial P}{\partial y} \quad \text{in 0<x<L; 0<y<h}$$

$$u=u_0 \qquad \text{at x=0; 0<y<h}$$

$$v=0 \qquad \text{at x=0; 0<y<h}$$

$$u=v=0 \qquad \text{at y=0 and y=h; 0<x<L}$$

Solving PDEs with RBFs

Non-Linear Problems

- Example: Hydrodynamic Developing Flow
  - RBF Approximation

$$u(x, y) = \sum_{i=1}^{N} \phi_i \psi(\mathbf{r}_i)$$

$$v(x, y) = \sum_{j=1}^{N} \phi_j \psi(\mathbf{r}_j)$$

$$P(x, y) = \sum_{k=1}^{N / \partial\Omega} \phi_k \psi(\mathbf{r}_k)$$

$$\psi_i(x, y) = \left[(x - x_i)^2 + (y - y_i)^2 + c_i^2 \right]^{\beta}$$

Solving PDEs with RBFs

- Non-Linear Problems
  - Example: Hydrodynamic Developing Flow
    - Substituting in the original PDE, we obtain

$$\sum_{i=1}^{N}\phi_i\frac{\partial\psi_i}{\partial x}+\sum_{j=1}^{N}\phi_j\frac{\partial\psi_j}{\partial x}=0 \quad \text{in 0<x<L; 0<y<h}$$

$$\left(\sum_{i=1}^{N}\phi_i\psi_i\right)\left(\sum_{i=1}^{N}\phi_i\frac{\partial\psi_i}{\partial x}\right)+\left(\sum_{j=1}^{N}\phi_j\psi_j\right)\left(\sum_{i=1}^{N}\phi_i\frac{\partial\psi_i}{\partial y}\right)=\frac{\mu}{\rho}\left(\sum_{i=1}^{N}\phi_i\frac{\partial^2\psi_i}{\partial x^2}+\sum_{i=1}^{N}\phi_i\frac{\partial^2\psi_i}{\partial y^2}\right)-\frac{1}{\rho}\sum_{k=1}^{N/\partial\Omega}\phi_k\frac{\partial\psi_k}{\partial x} \quad \text{in 0<x<L; 0<y<h}$$

$$\left(\sum_{i=1}^{N}\phi_i\psi_i\right)\left(\sum_{j=1}^{N}\phi_j\frac{\partial\psi_j}{\partial x}\right)+\left(\sum_{j=1}^{N}\phi_j\psi_j\right)\left(\sum_{j=1}^{N}\phi_j\frac{\partial\psi_j}{\partial y}\right)=\frac{\mu}{\rho}\left(\sum_{j=1}^{N}\phi_j\frac{\partial^2\psi_j}{\partial x^2}+\sum_{j=1}^{N}\phi_j\frac{\partial^2\psi_j}{\partial y^2}\right)-\frac{1}{\rho}\sum_{k=1}^{N/\partial\Omega}\phi_k\frac{\partial\psi_k}{\partial x} \quad \text{in 0<x<L; 0<y<h}$$

$$\sum_{i=1}^{N}\phi_i\psi_i=u_0 \qquad \text{at x=0; 0<y<h}$$

$$\sum_{i=1}^{N}\phi_j\psi_j=0 \qquad \text{at x=0; 0<y<h}$$

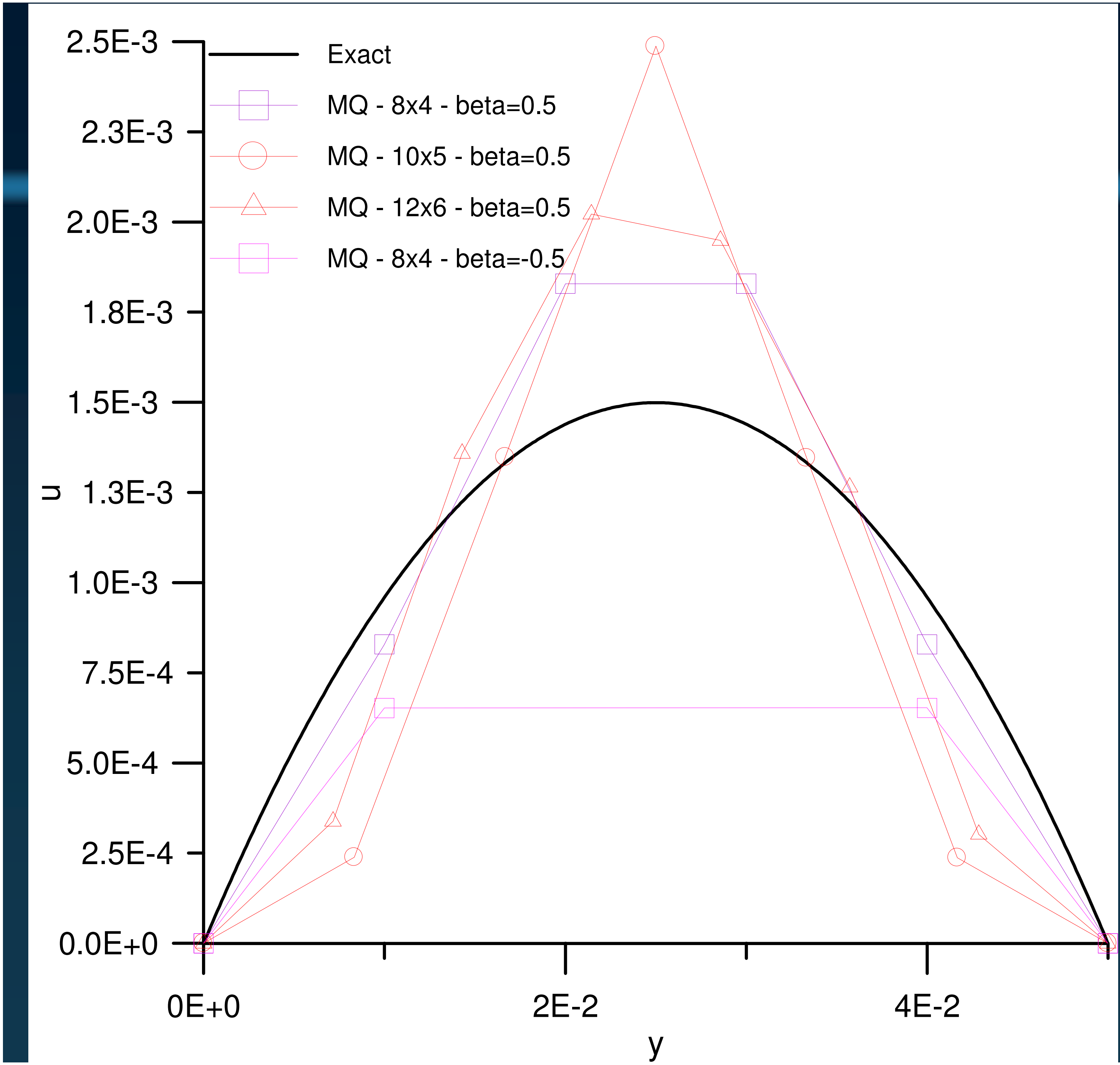
$$\sum_{i=1}^{N}\phi_i\psi_i=\sum_{i=1}^{N}\phi_j\psi_j=0 \qquad \text{at y=0 and y=h; 0<x<L}$$

NON-LINEAR SYSTEM

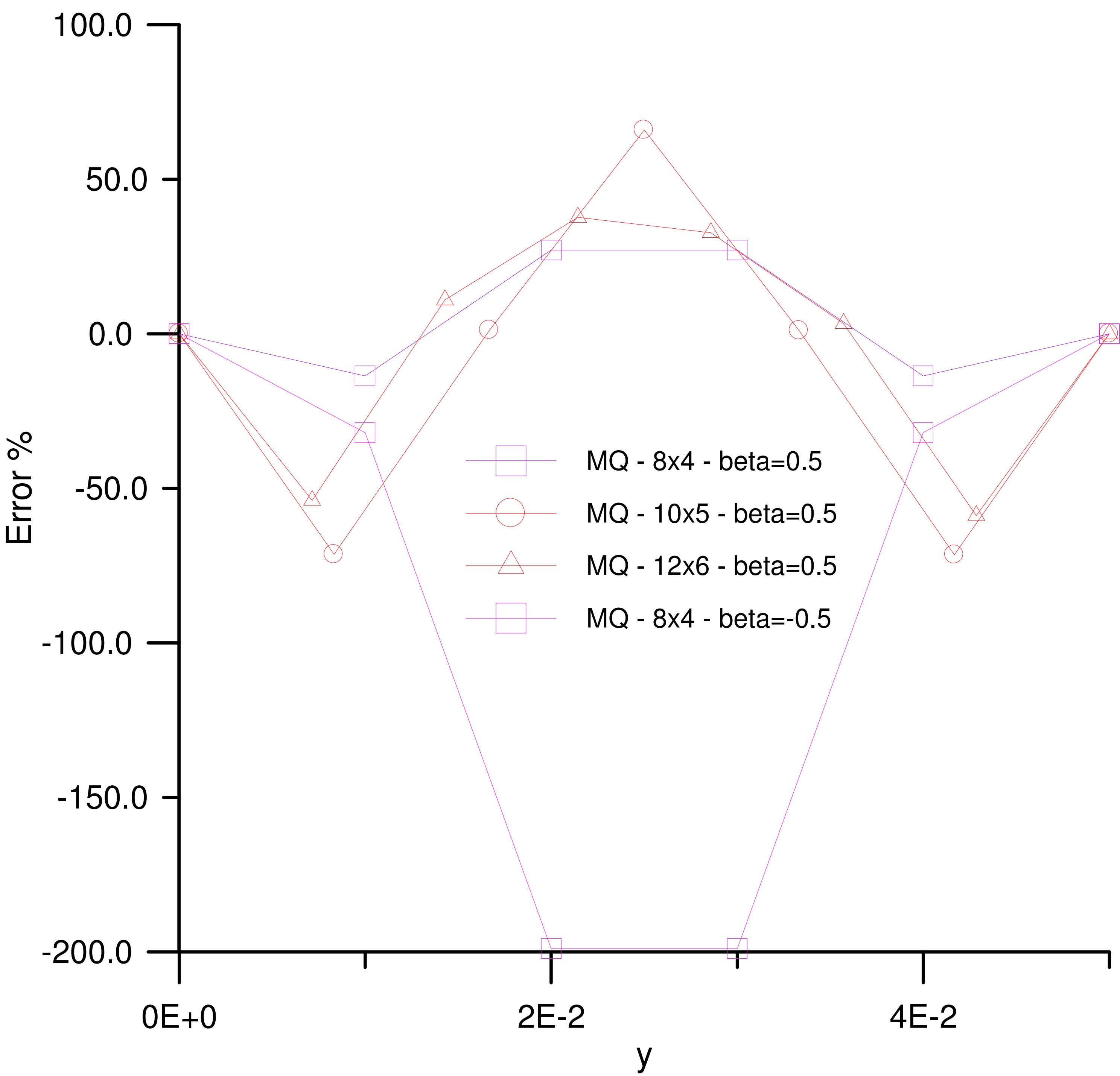
Solving PDEs with RBFs

- Non-Linear Problems
  - Example: Hydrodynamic Developing Flow
    - Comparison with benchmark solution

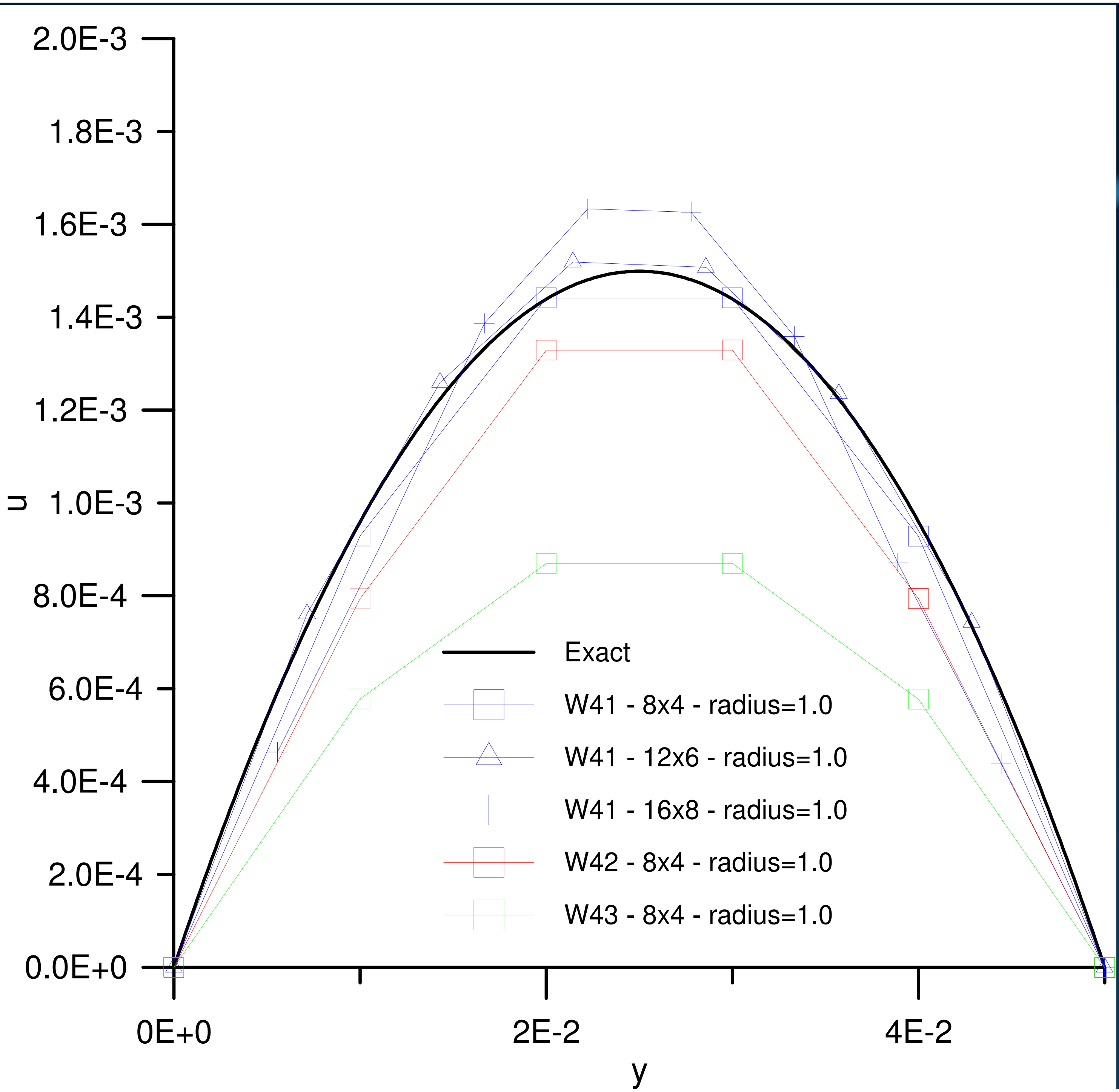
$$u(y) = -\frac{6u_{av}}{h^2}\left(y^2 - hy\right)$$



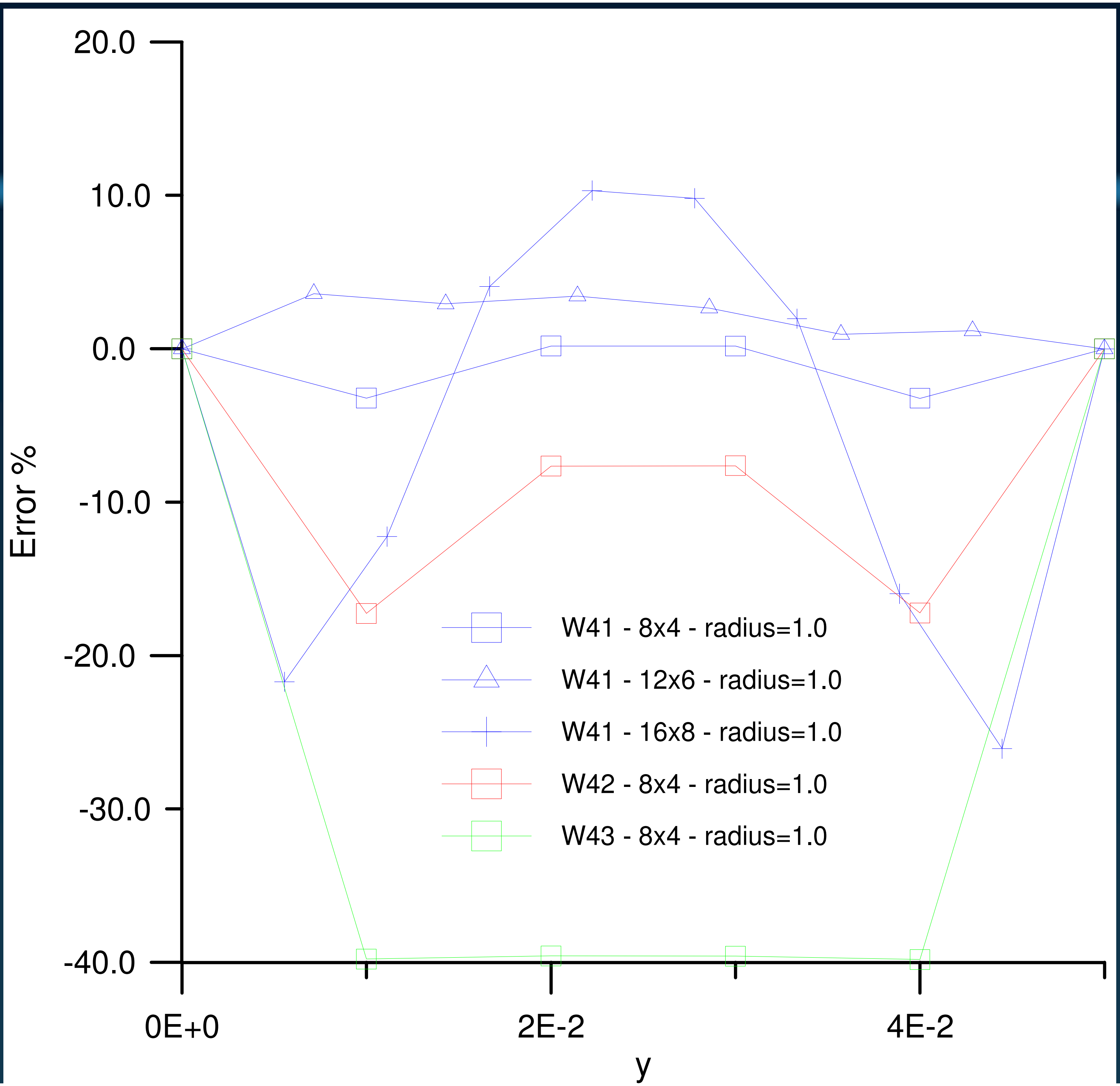
- ➔ Multiquadrics



- ➔ Multiquadrics


2.0E-3
1.8E-3
1.6E-3
1.4E-3
1.2E-3
1.0E-3
8.0E-4
6.0E-4
4.0E-4
2.0E-4
0.0E+0
u
0E+0
2E-2
4E-2
y
Exact
W41 - 8x4 - radius=1.0
W41 - 12x6 - radius=1.0
W41 - 16x8 - radius=1.0
W42 - 8x4 - radius=1.0
W43 - 8x4 - radius=1.0

➔ Wendland


20.0
10.0
0.0
-10.0
-20.0
-30.0
-40.0
Error %
0E+0
2E-2
4E-2
y
W41 - 8x4 - radius=1.0
W41 - 12x6 - radius=1.0
W41 - 16x8 - radius=1.0
W42 - 8x4 - radius=1.0
W43 - 8x4 - radius=1.0

- Wendland

Solving PDEs with RBFs

Type of function	Points in the expansion	Expoent - β	Shape factor - c	Residual - ε	CPU time (seconds)
MQ	8x4	0.5	0.001	8.35 E-5	1.40
		-0.5	0.001	1.76 E-5	1.45
	10x5	0.5	0.0002	1.68 E-5	4.32
	12x6	0.5	0.0005	5.31 E-5	11.11
W41	8x4				0.84
	10x5				7.38
	12x6				6.06
	16x8				61.93
W42	8x4				1.78
W43					2.91

Generating Response Surfaces with RBFs

- General Formulation

$$s(x_i) = f(x_i) = \sum_{j=1}^{N} \alpha_j \phi\left(\left|x_i - x_j\right|\right) + \sum_{k=1}^{M} \beta_k p_k(x_i) + \beta_0 \qquad \sum_{j=1}^{N} \alpha_j p_k(x_j) = 0 \qquad \sum_{j=1}^{N} \alpha_j = 0$$

- The polynomial part of the expansion can be taken, for examples, as

$$p_k(x_i) = x_i^k$$

- And the Multiquadrics can be used as basis functions

$$\phi\left(\left|x_i - x_j\right|\right) = \sqrt{\left(x_i - x_j\right)^2 + c_j^2}$$

$\hat{y}_i$

Generating Response Surfaces with RBFs

- Performance parameters
  - R square

$$R^2 = 1 - \frac{\sum_{i=1}^{n} \left(y_i - \hat{y}_i \right)^2}{\sum_{i=1}^{n} \left(y_i - \overline{y} \right)^2} = 1 - \frac{MSE}{\text{variance}}$$

– While *MSE* (Mean Square Error) represents the departure of the metamodel from the real simulation model, the variance captures how irregular the problem is. *The larger the value of R Square, the more accurate the metamodel.*

$\hat{y}_i$

Generating Response Surfaces with RBFs

- Performance parameters
  - RAAE

$$RAAE = \frac{\sum_{i=1}^{n} \left| y_i - \hat{y}_i \right|}{n * STD}$$

– where STD stands for standard deviation. The smaller the value of RAAE, the more accurate the metamodel.

$\hat{y}_i$

Generating Response Surfaces with RBFs

- Performance parameters
  - RMAE

$$RMAE = \frac{\max\left(\left|y_1 - \hat{y}_1\right|, \left|y_2 - \hat{y}_2\right|, \ldots, \left|y_n - \hat{y}_n\right|\right)}{STD}$$

– Large RMAE indicates large error in one region of the design space even though the overall accuracy indicated by R Square and RAAE can be very good. Therefore, a small RMAE is preferred; however, since this metric cannot show the overall performance in the design space, it is not as important as R Square and RAAE.

Generating Response Surfaces with RBFs

- Some sample problems

Test function	Training points	Testing points	Minimum value of f	Maximum value of f	Standard deviation of f	Average value of f	Non-linearity
1	198	1000	-21.01	29.25	8.92	8.60	High
2	198		-1717.43	-539.36	187.40	-1146.82	Low
3	198		-1327241.16	-1.14	224561.73	-213124.84	High
4	198		258.78	4779.01	1002.27	2185.25	Low
5	459		7136.99	195608.81	27294.72	69060.73	Low
6	100		98.94	187.53	17.75	127.73	High
7	100		-0.26	0.26	0.15	0.00	High
8	100		-1.51	8.91	2.08	1.66	Low
9	125		2.15	146369.38	21527.31	10842.11	High
10	125		3.42	32.84	7.70	29.13	High
11	125		-40829.97	-40749.42	21.60	-40790.41	Low
12	100		-12.72	13.75	5.74	0.52	Low
13	100		-54.77	77.80	30.94	12.60	Low

$\hat{y}_i$

Generating Response Surfaces with RBFs

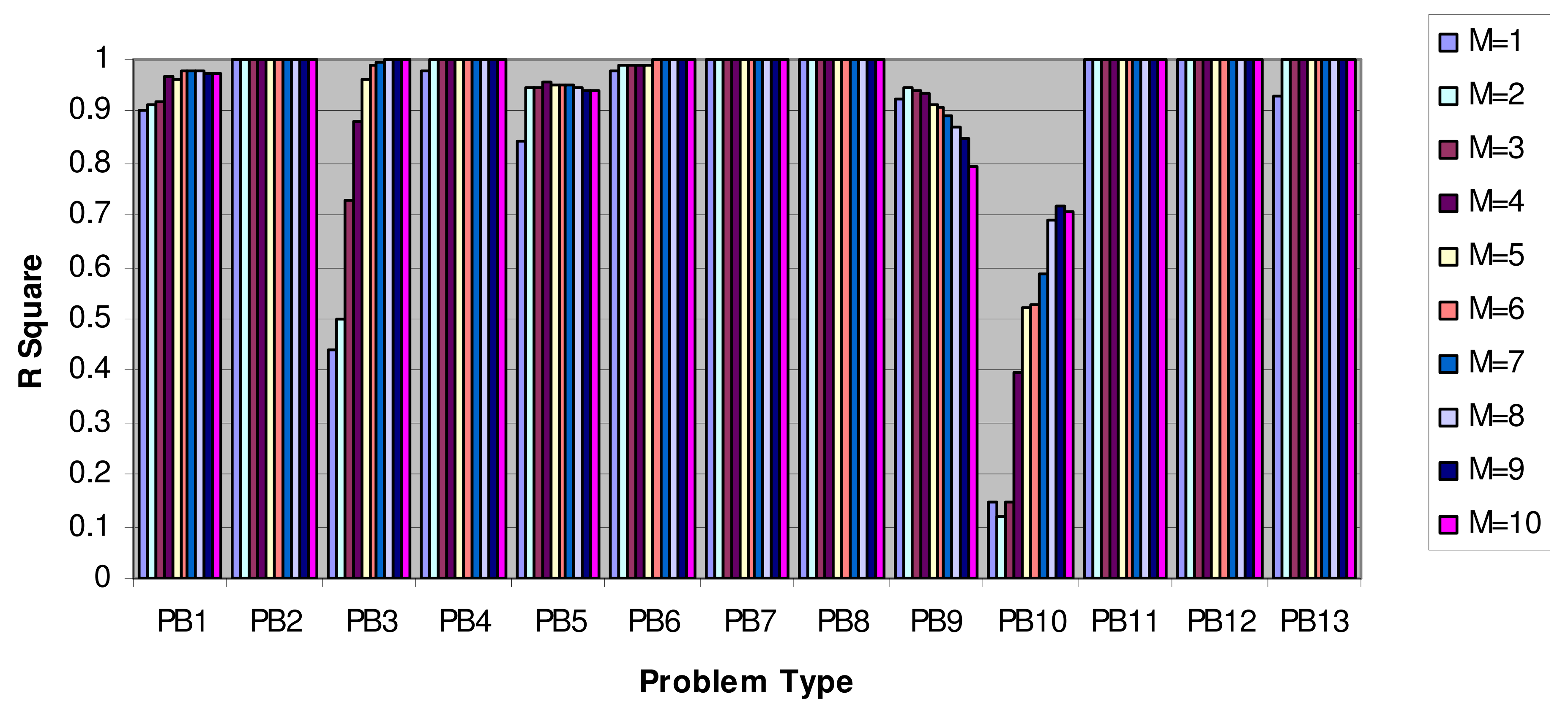
Number of training and testing points						Non Linearity
		Training			Testing	
PB #	# Vars	Scarce	Small	Large		
PB1	10	30	100	198	1000	High
PB2	10	30	100	198		Low
PB3	10	30	100	198		High
PB4	10	30	100	198		Low
PB5	16	48	160	459		Low
PB6	2	N/A	9	100		High
PB7	2		9	100		High
PB8	2		9	100		Low
PB9	3		27	125		High
PB10	3		27	125		High
PB11	3		27	125		Low
PB12	2		9	100		Low
PB13	2		9	100		Low

Generating Response Surfaces with RBFs

- ➔ Large set of training points

$$s(x_i) = f(x_i) = \sum_{j=1}^{N} \alpha_j \phi\left(\left|x_i - x_j\right|\right) + \sum_{k=1}^{M} \beta_k p_k(x_i) + \beta_0$$

$$p_k(x_i) = x_i^k$$

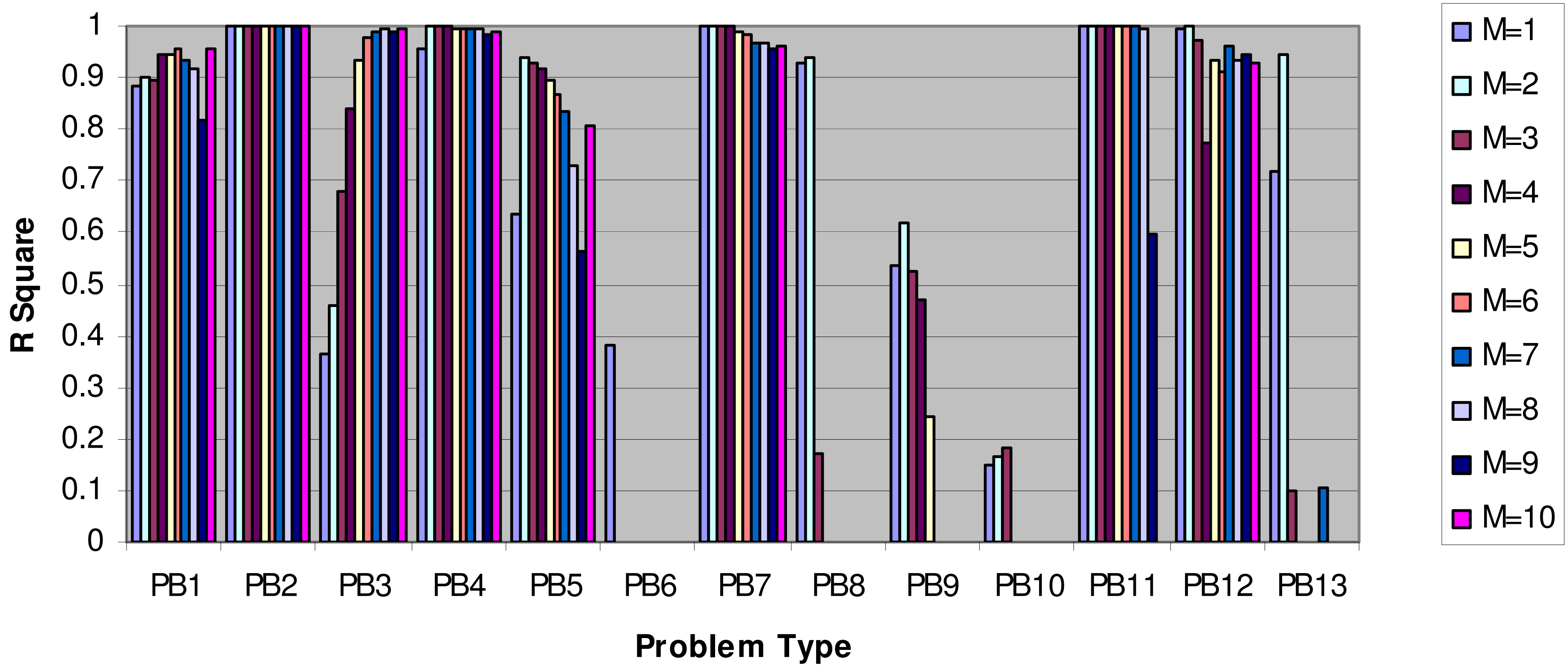


Generating Response Surfaces with RBFs

- ➔ Small set of training points

$$s(x_i) = f(x_i) = \sum_{j=1}^{N} \alpha_j \phi\left(\left|x_i - x_j\right|\right) + \sum_{k=1}^{M} \beta_k p_k(x_i) + \beta_0$$

$$p_k(x_i) = x_i^k$$

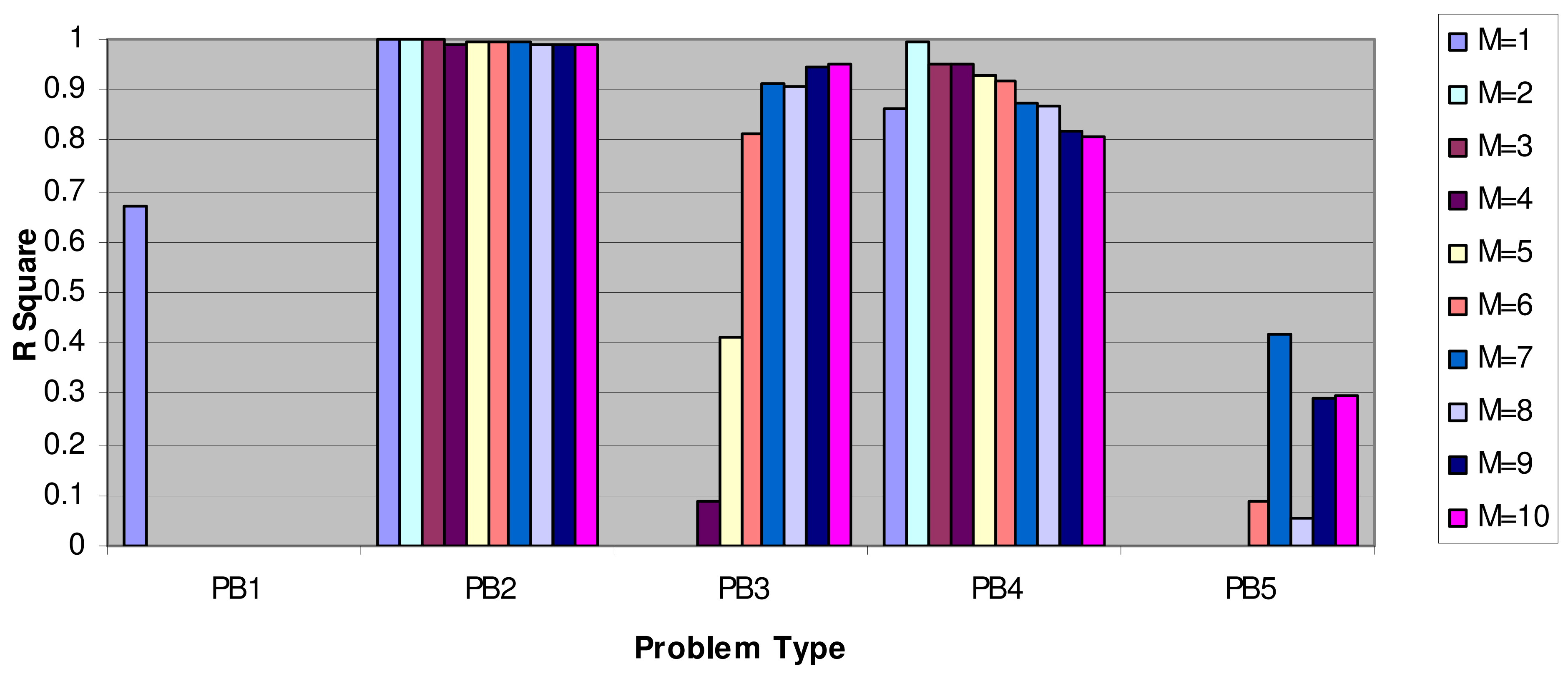


Generating Response Surfaces with RBFs

- ➔ Scarce set of training points

$$s(x_i) = f(x_i) = \sum_{j=1}^{N} \alpha_j \phi\left(\left|x_i - x_j\right|\right) + \sum_{k=1}^{M} \beta_k p_k(x_i) + \beta_0$$

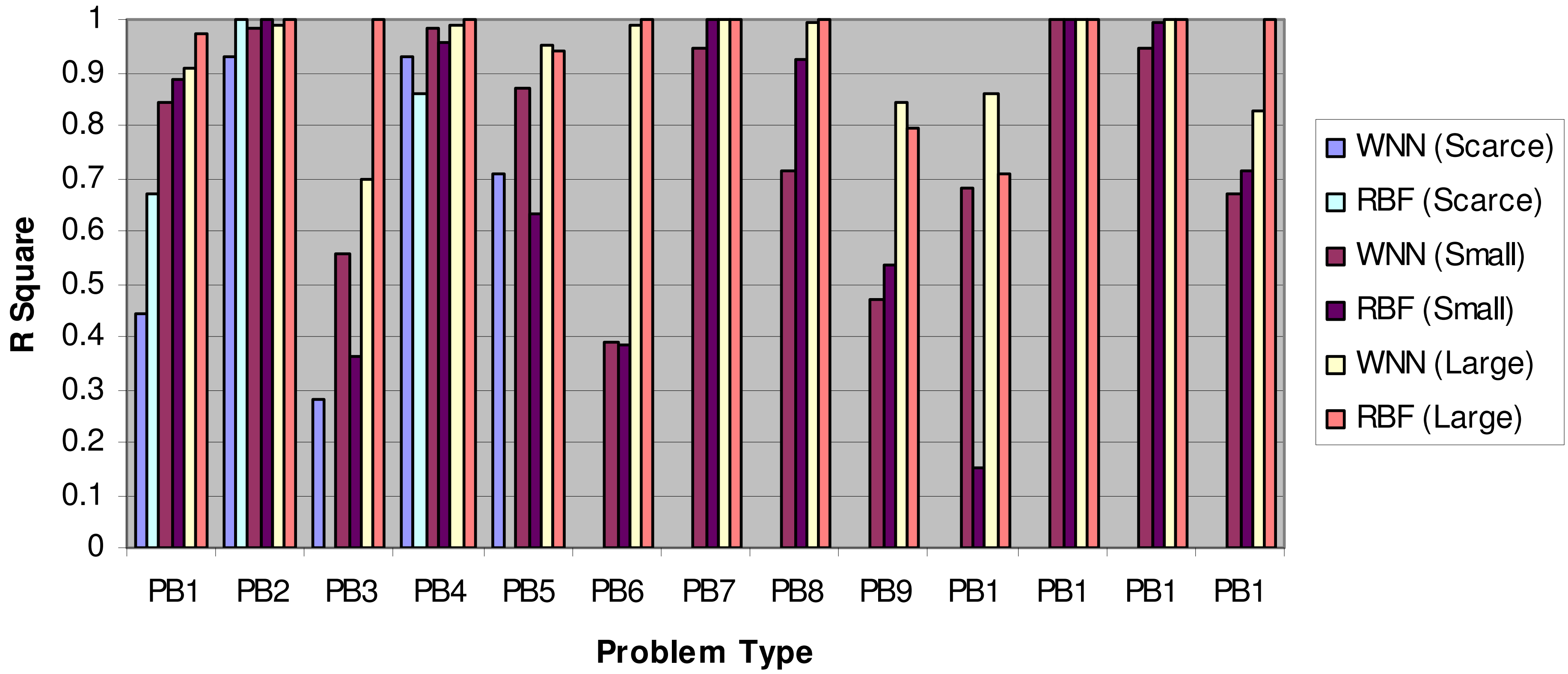
$$p_k(x_i) = x_i^k$$



Generating Response Surfaces with RBFs

➔ Wavelets *versus* RBFs (R Square)

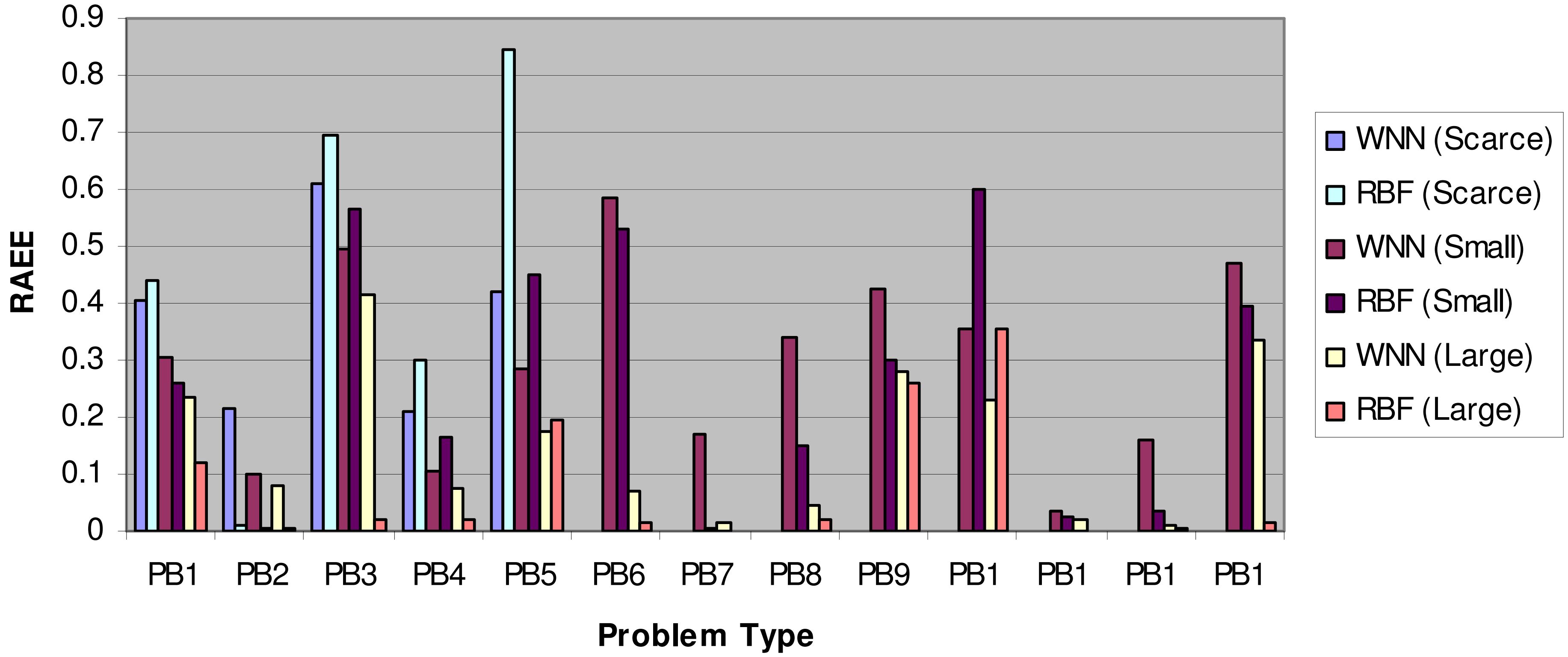
$$R^2 = 1 - \frac{\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2}{\sum_{i=1}^{n}\left(y_i - \bar{y}\right)^2} = 1 - \frac{MSE}{\text{variance}}$$



Generating Response Surfaces with RBFs

- Wavelets *versus* RBFs (RAEE)

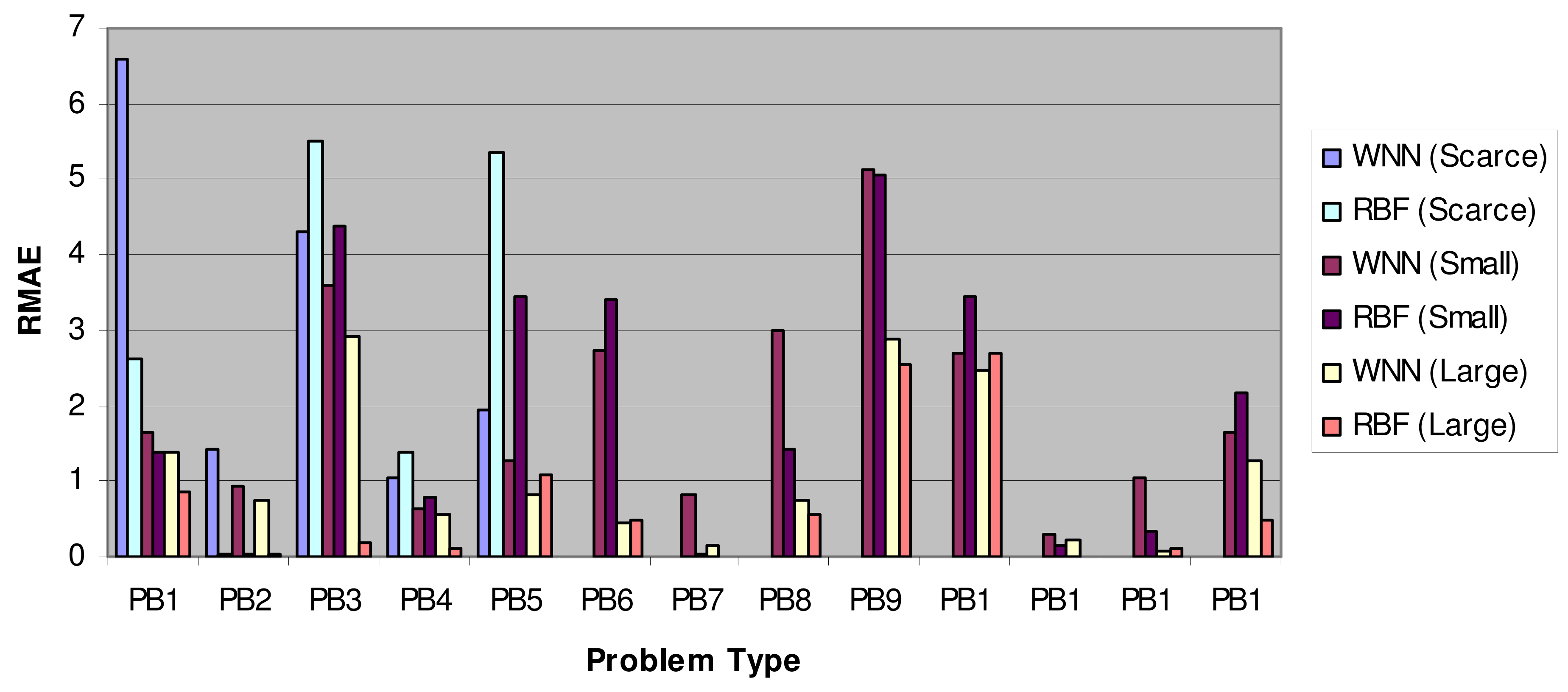
$$RAAE = \frac{\sum_{i=1}^{n} \left| y_i - \hat{y}_i \right|}{n * STD}$$



Generating Response Surfaces with RBFs

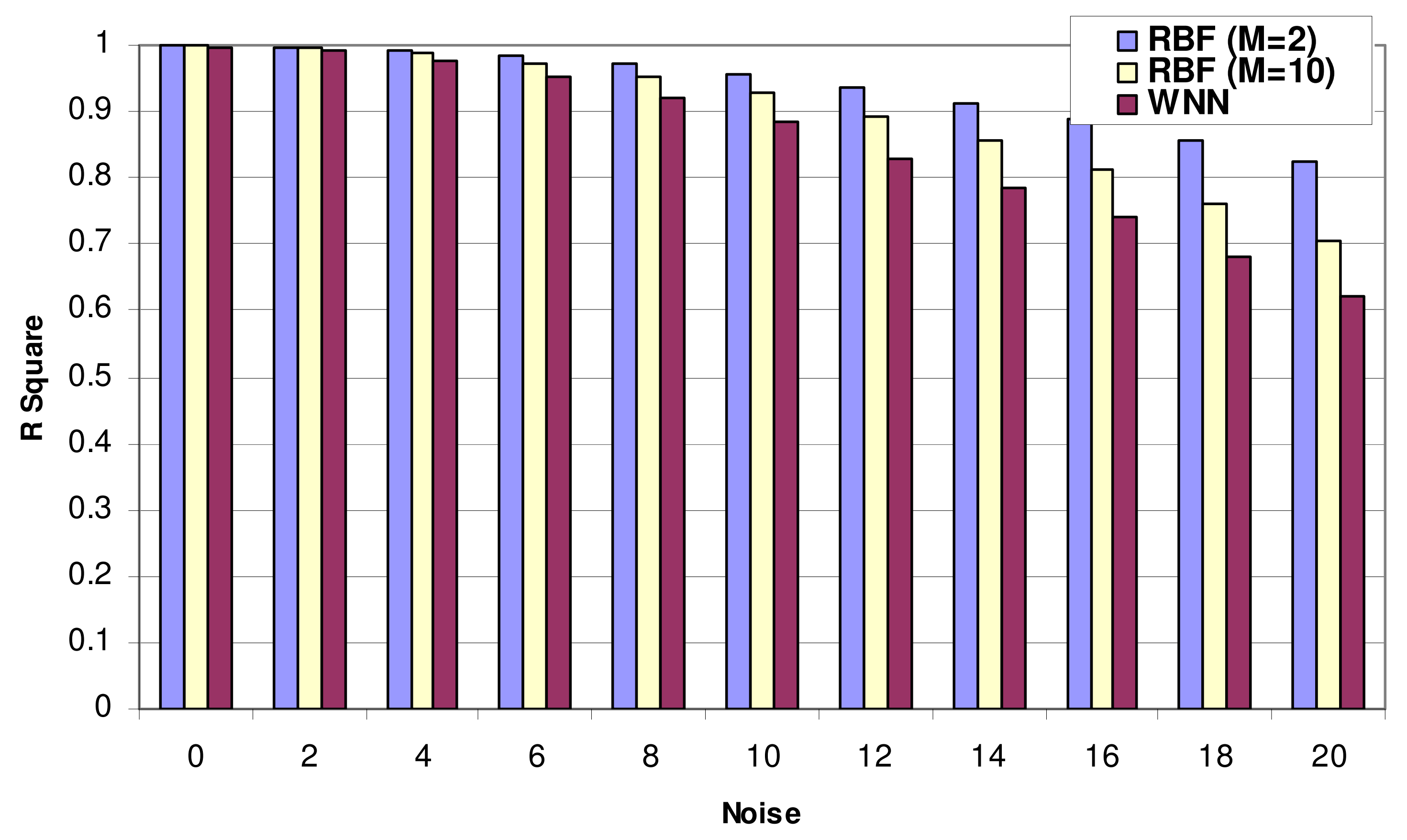
- Wavelets *versus* RBFs (RMAE)

$$RMAE = \frac{\max\left(|y_1 - \hat{y}_1|, |y_2 - \hat{y}_2|, \ldots, |y_n - \hat{y}_n|\right)}{STD}$$



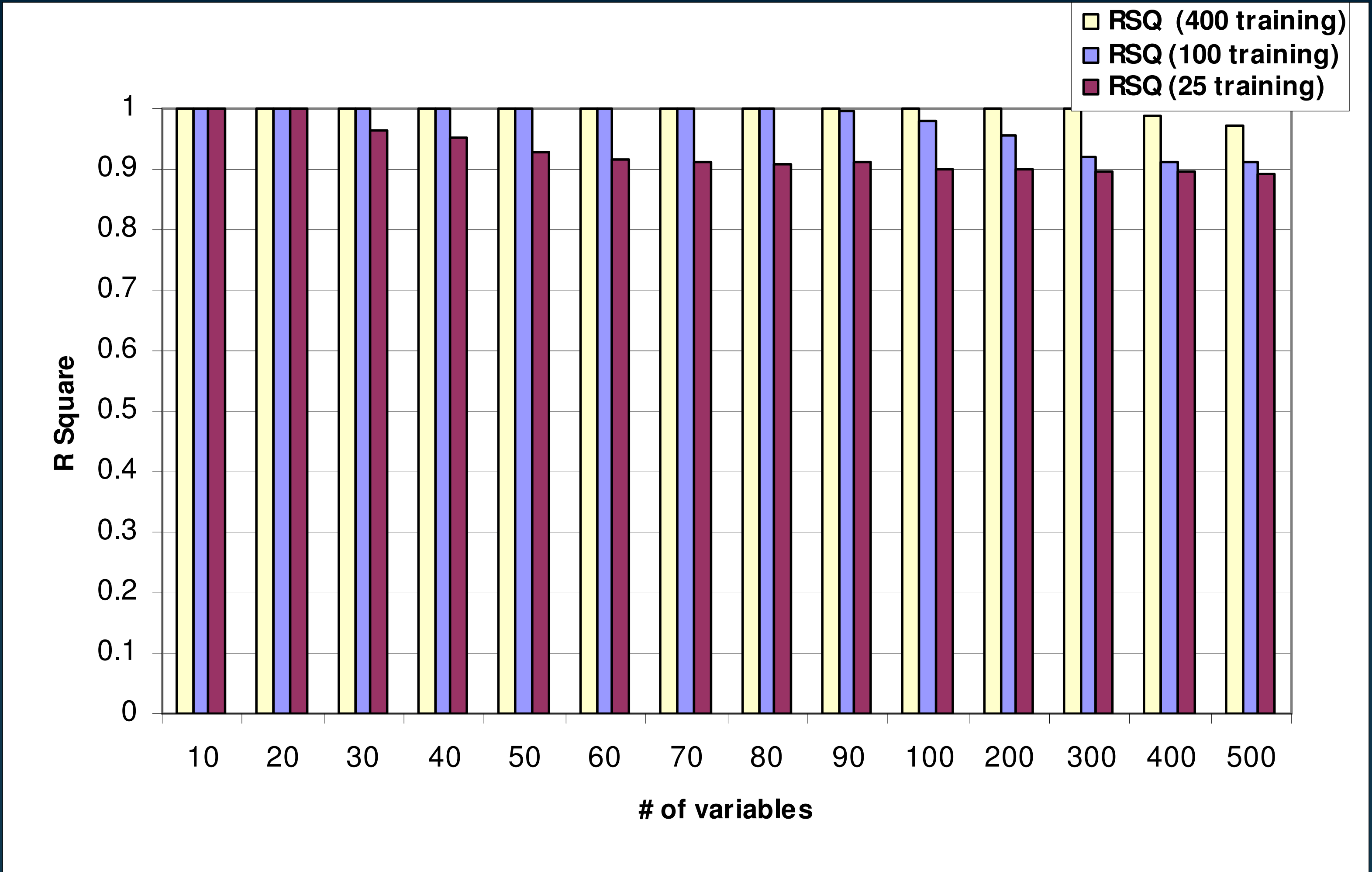
Generating Response Surfaces with RBFs

- Wavelets *versus* RBFs (RSquare – PB#13 with NOISE)



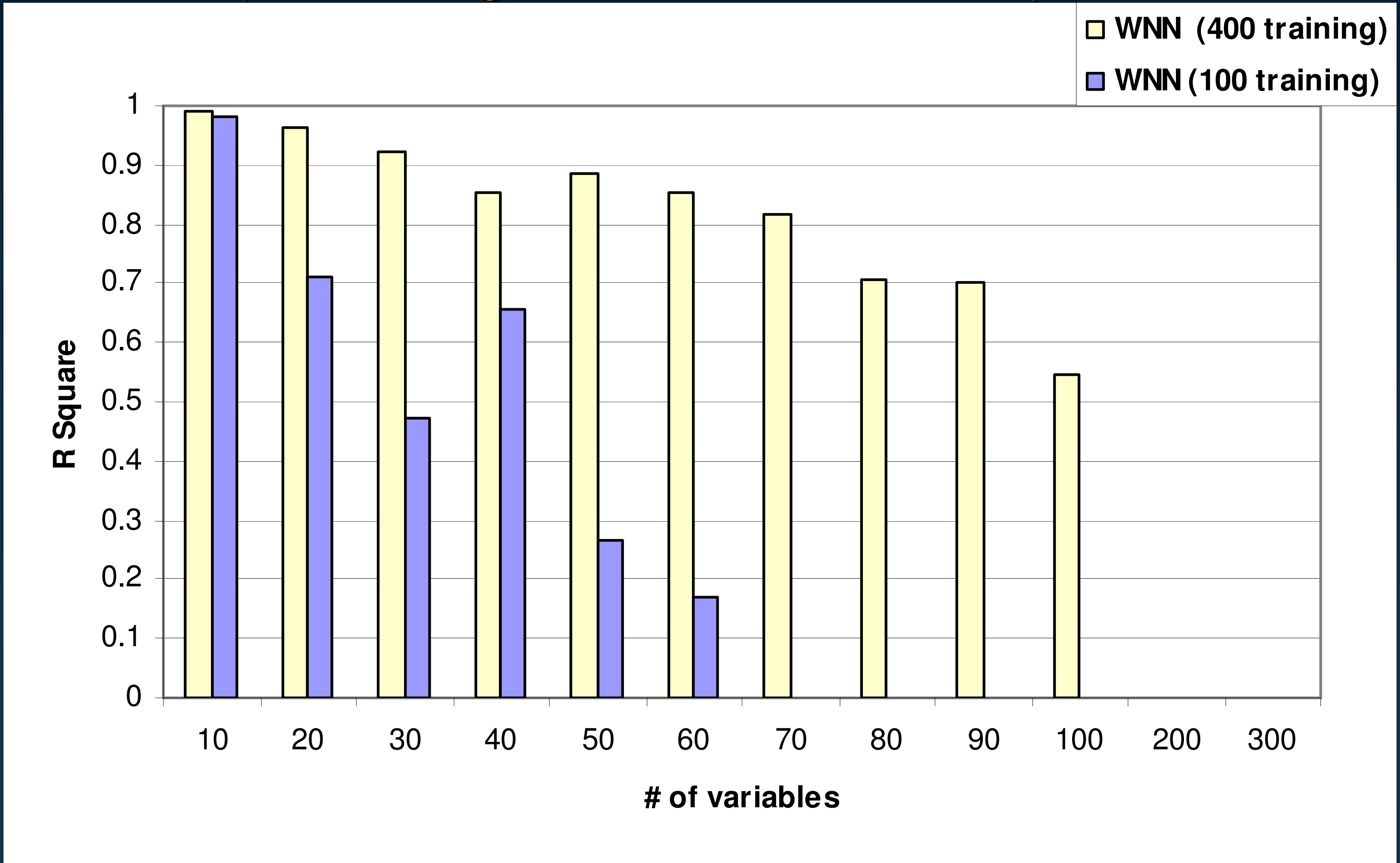
Generating Response Surfaces with RBFs

- RBFs (PB#2 Large number of variables)



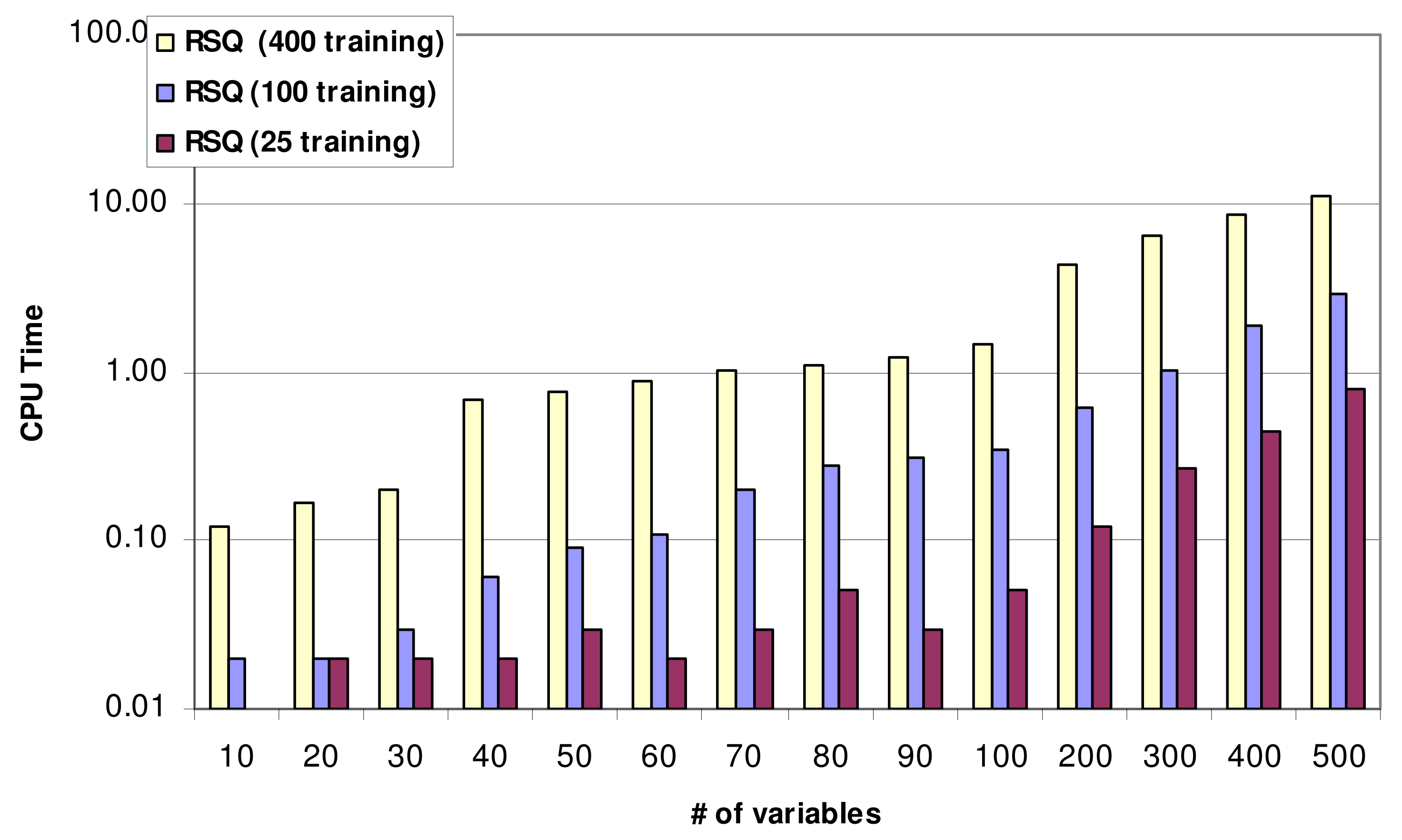
Generating Response Surfaces with RBFs

- Wavelets (PB#2 Large number of variables)



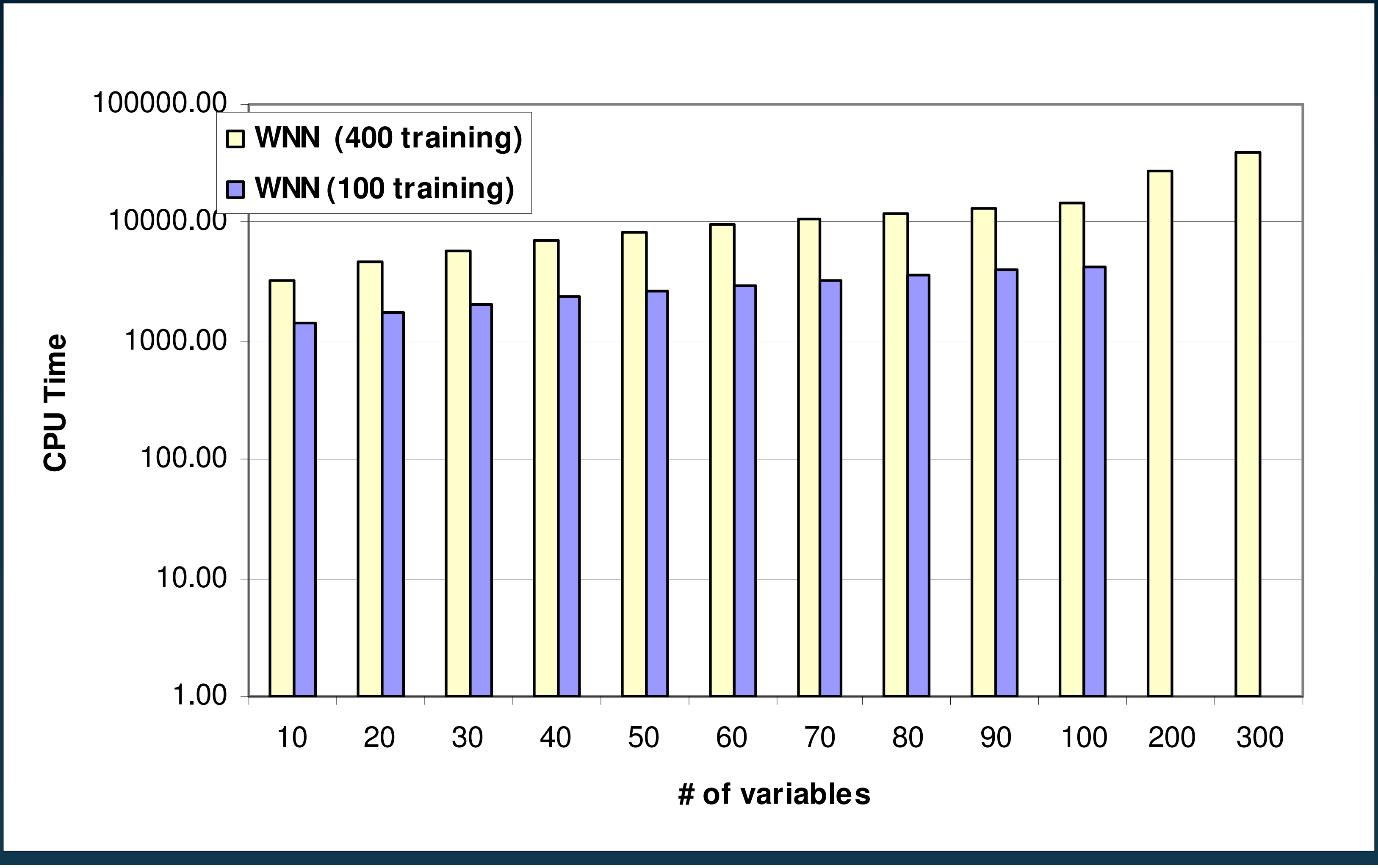
Generating Response Surfaces with RBFs

- RBFs (PB#2 Large number of variables)



Generating Response Surfaces with RBFs

- Wavelets (PB#2 Large number of variables)



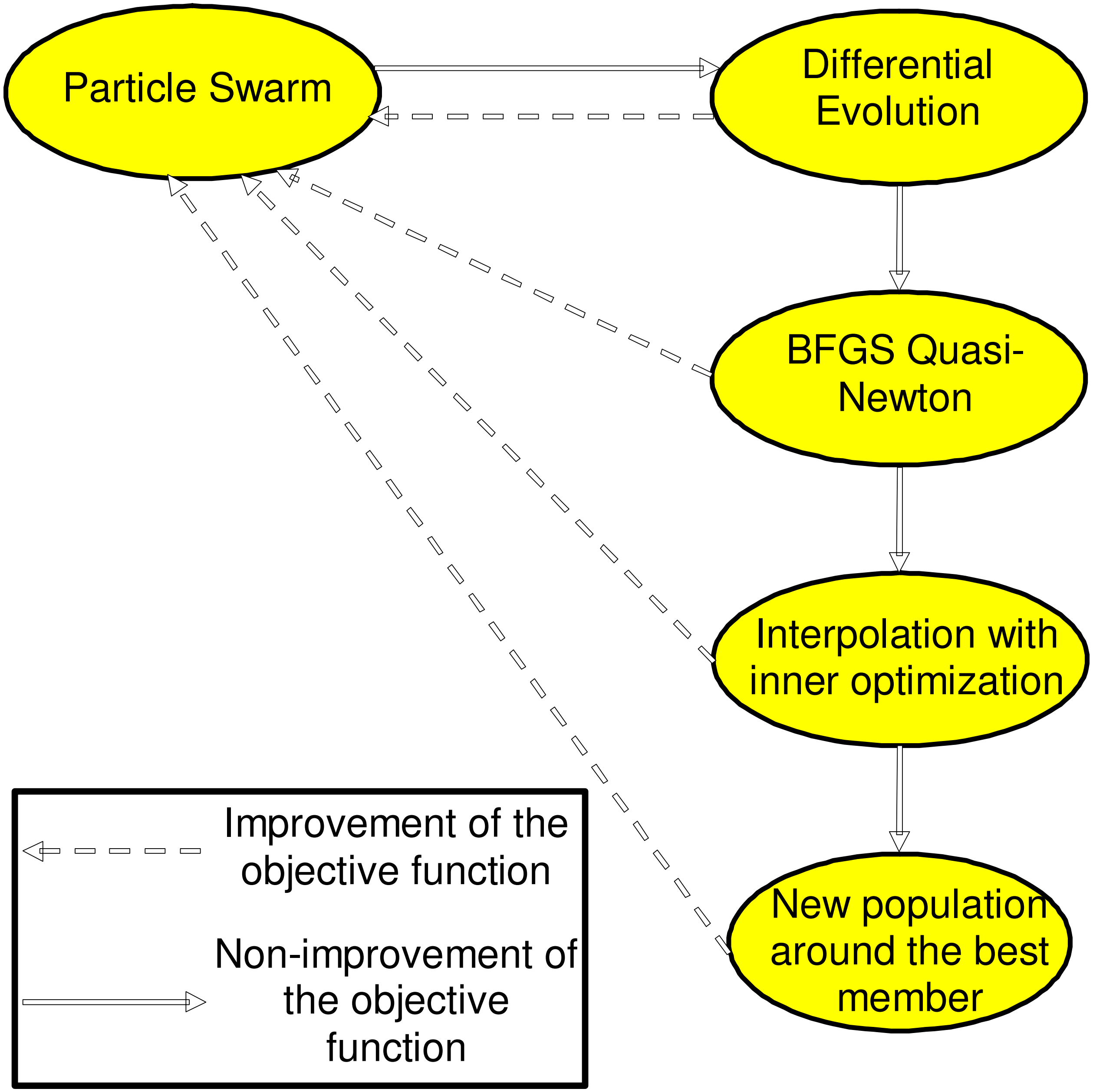
Hybrid Methods

- ➔ Metamodel strategy

Colaço and Dulikravich

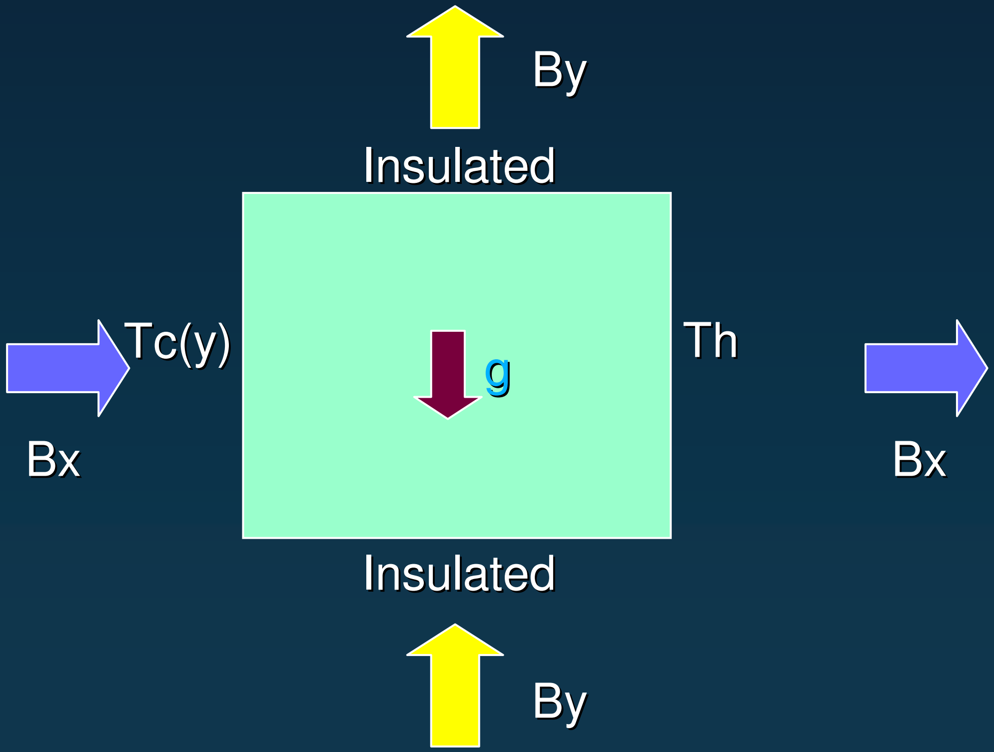
Hybrid Optimizer

version 3 – Nov. 2006



MHD – Example 5

- Optimization of the boundary conditions in order to minimize the natural convection effects (thermomagnetic optimization)
- Test problem:



MHD – Example 5

- Optimization Problem
- The boundary conditions are parameterized as B-Splines.
- NETLIB’s subroutine GCVSPL, based on the cross-validation smoothing procedure, was used for the interpolation.
- Objective function:

$$F = \sqrt{\frac{1}{\#\text{liquid cells}} \sum_{i=1}^{\#\text{liquid cells}} \left(\frac{\partial C_i}{\partial y_i} \right)^2}$$

MHD – Example 5

- Optimization Problem
- Parameters for silicon:

$\rho_l = 2550 \text{ kg m}^{-3}$ $\rho_s = 2550 \text{ kg m}^{-3}$

$C_{Pl} = 1059 \text{ J kg}^{-1} \text{ K}^{-1}$ $C_{Ps} = 1059 \text{ J kg}^{-1} \text{ K}^{-1}$

$\sigma_l = 12.3 \text{ x } 10^5 \text{ 1/m } \Omega$ $\sigma_s = 4.3 \text{ x } 10^4 \text{ 1/m } \Omega$

$D_l = 6.043 \text{ x } 10^{-9} \text{ kg m}^{-1} \text{ s}^{-1}$ $D_s = 0 \text{ kg m}^{-1} \text{ s}^{-1}$

$L=1.8 \text{ x } 106 \text{ J kg}^{-1}$ $C_0 = 0.1 \text{ kg m}^{-3}$

$n = 0.3$ $T_h=1685.04 \text{ K}$ $1620 \text{ K} < T_c < 1630 \text{ K}$

$k_l = 64 \text{ W m}^{-1} \text{ K}^{-1}$ $k_s = 64 \text{ W m}^{-1} \text{ K}^{-1}$

$\mu_l = 0.0032634 \text{ kg m}^{-1} \text{ s}^{-1}$ $\mu s = 326.34 \text{ kg m}^{-1} \text{ s}^{-1}$

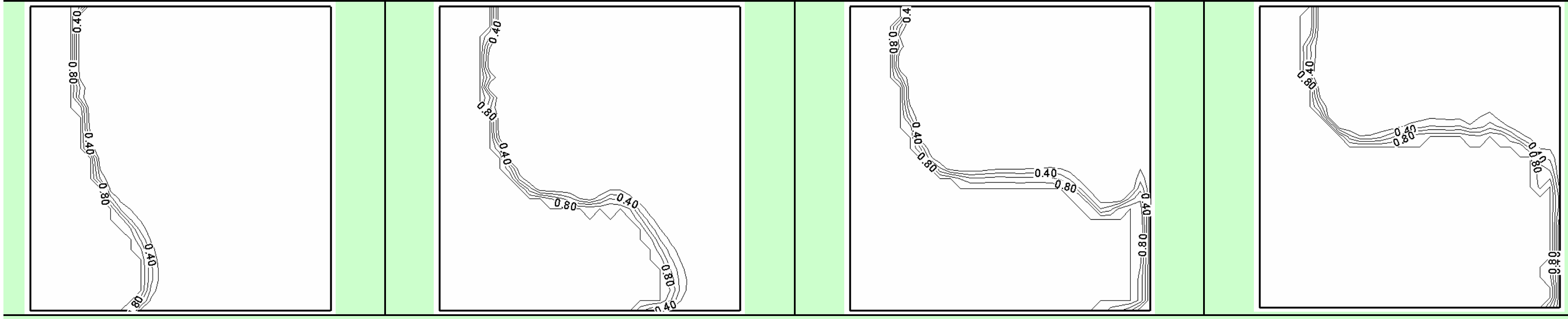
$\beta = 1.4 \text{ x } 10^{-4} \text{ K}^{-1}$ $\beta s = 0.0875$

$g = 9.81 \text{ m s}^{-2}$ $\mu_m = 1.2566 \text{ x } 10^{-5} \text{ T m A}^{-1}$

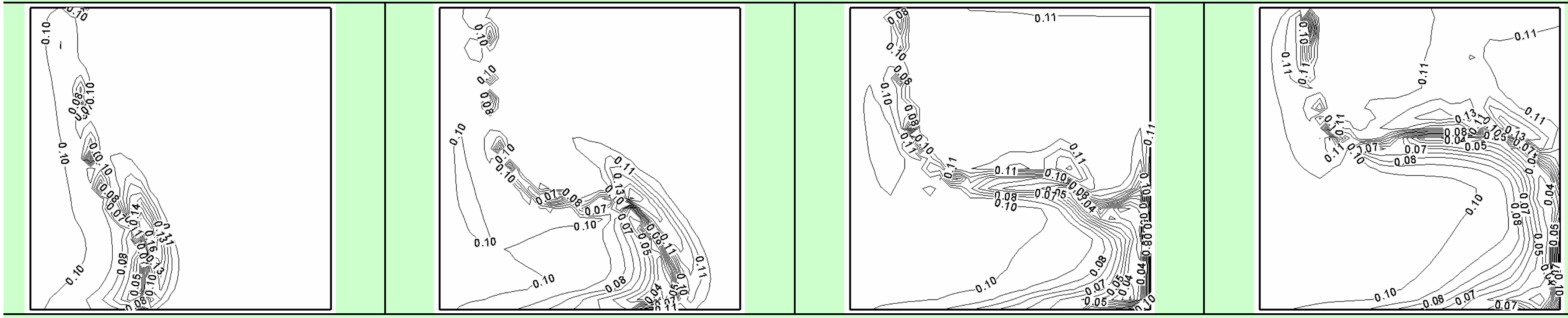
$Te = 1681 \text{ K}$ $C_e = 0.8 \text{ kg m}^{-3}$

MHD – Example 5 (no magnetic fields)

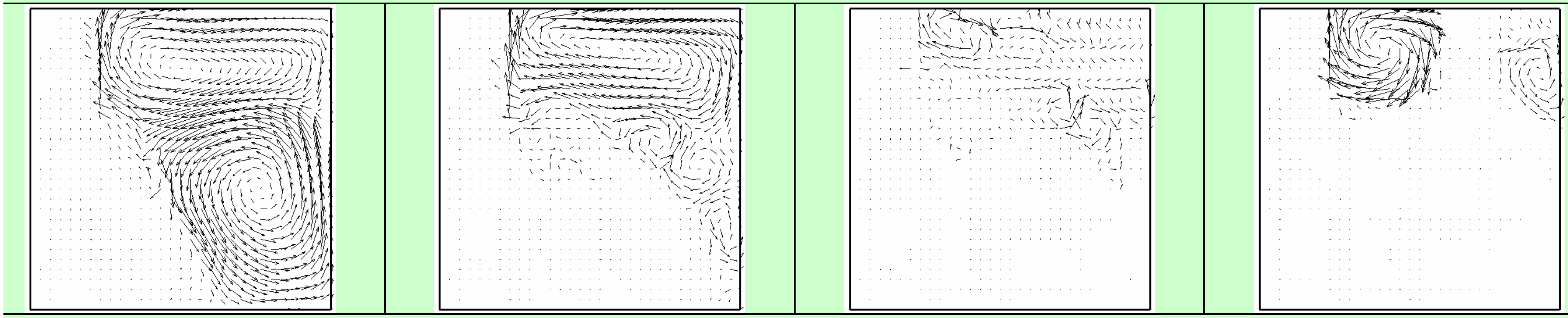
15 min | 30 min | 45 min | 60 min



(a) Void fraction



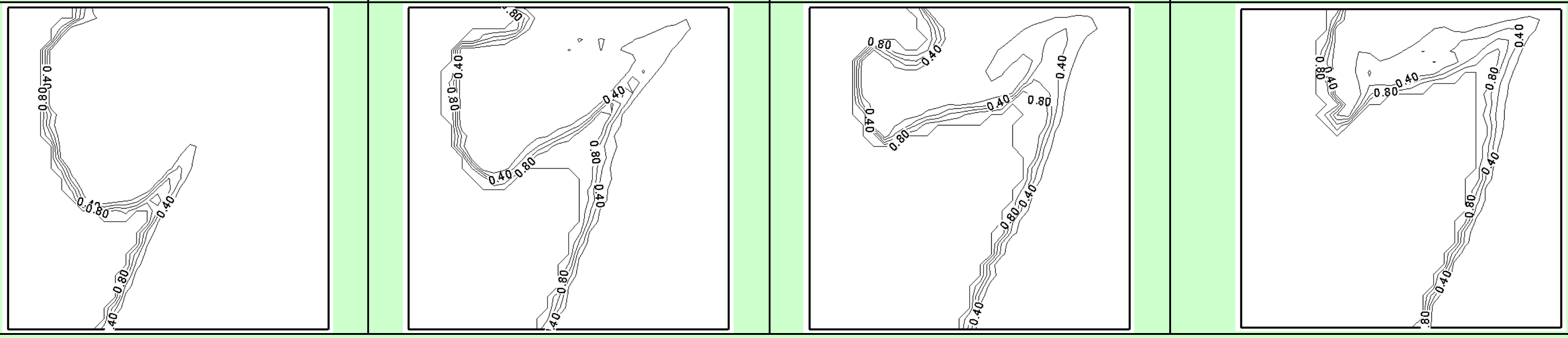
(b) Concentration



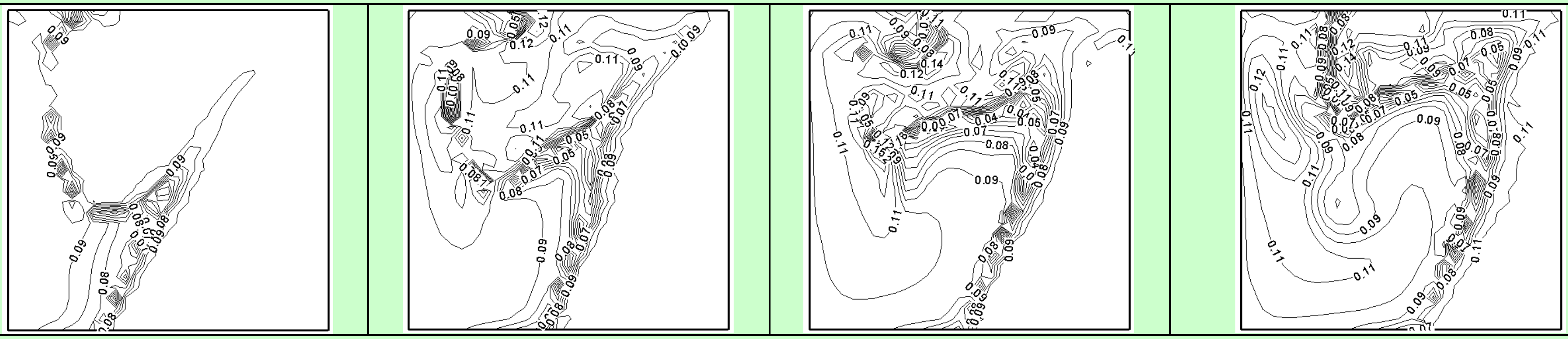
(c) Velocity vectors

MHD – Example 5 (optmized magnetic fields)

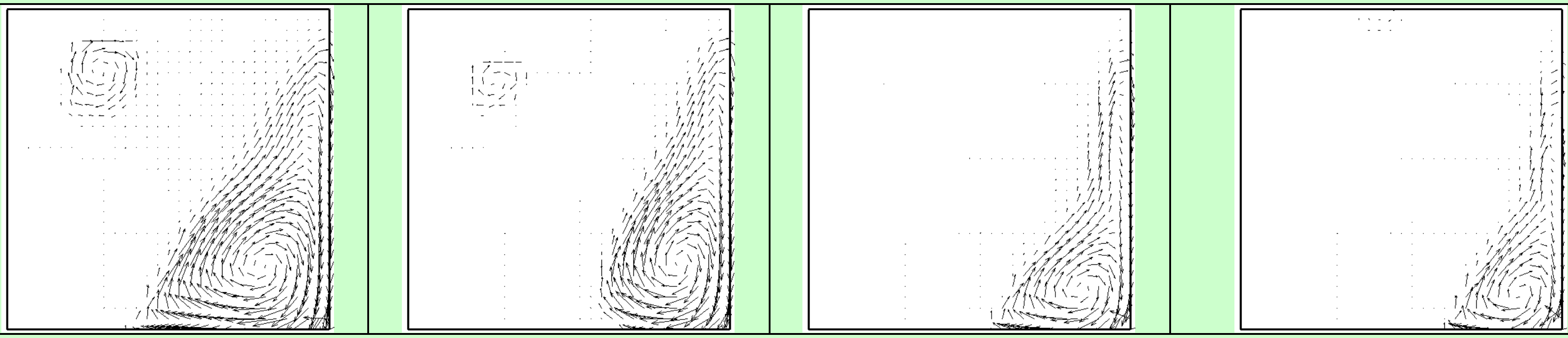
15 min **30 min** **45 min** **60 min**



(a) void fraction



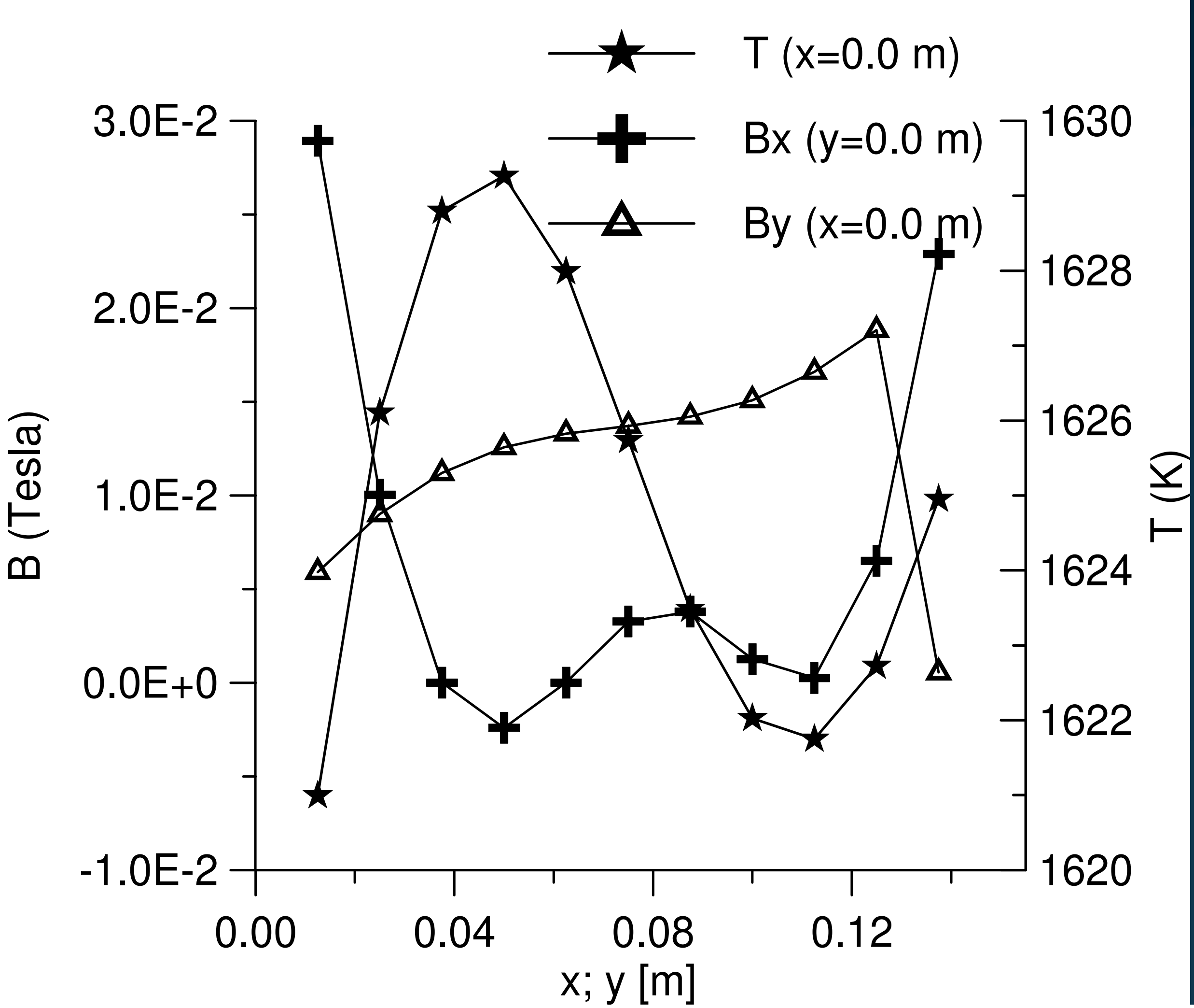
(b) Concentration



(c) Velocity vectors

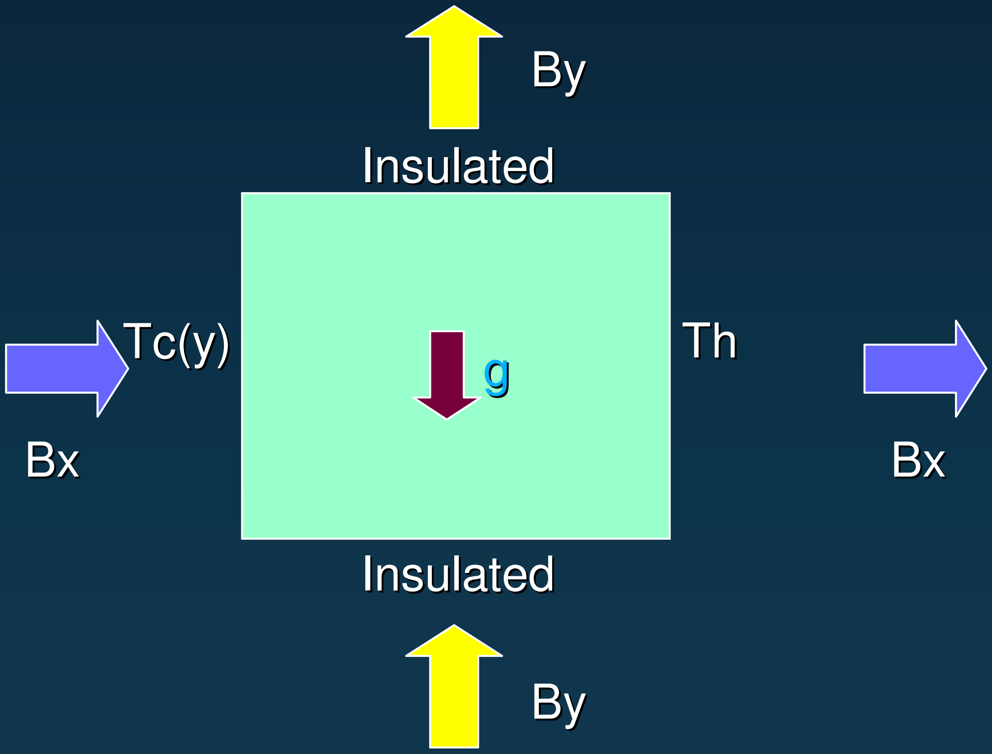
MHD – Example 5

- ➔ Optimized boundary conditions



MHD – Example 6

- Optimization of the boundary conditions in order to minimize the natural convection effects (thermomagnetic multiobjective optimization)
- Test problem:



MHD – Example 6

- Optimization Problem
- The boundary conditions are parameterized as B-Splines.
- NETLIB's subroutine GCVSPL, based on the cross-validation smoothing procedure, was used for the interpolation.
- Objective function:

$$F = \left(\text{liquid area fraction}\right)^2 + \sqrt{\frac{1}{\#\,\text{liquid cells}} \sum_{i=1}^{\#\text{liquid cells}} \left(\frac{\omega_i - \bar{\omega}_i}{\omega_{\max}} \right)^2}$$

MHD – Example 6

- Optimization Problem
- Parameters for silicon:

$\rho_l = 2550 \text{ kg m}^{-3}$ $\quad \rho_s = 2550 \text{ kg m}^{-3}$

$C_{Pl} = 1059 \text{ J kg}^{-1} \text{ K}^{-1}$ $\quad C_{Ps} = 1059 \text{ J kg}^{-1} \text{ K}^{-1}$

$\sigma_l = 12.3 \text{ x } 10^5 \text{ 1/m } \Omega$ $\quad \sigma_s = 4.3 \text{ x } 10^4 \text{ 1/m } \Omega$

$D_l = 6.043 \text{ x } 10^{-9} \text{ kg m}^{-1} \text{ s}^{-1}$ $\quad D_s = 0 \text{ kg m}^{-1} \text{ s}^{-1}$

$L = 1.8 \text{ x } 106 \text{ J kg}^{-1}$ $\quad C_0 = 0.1 \text{ kg m}^{-3}$

$n = 0.3 \quad T_h = 1685.04 \text{ K} \quad 1620 \text{ K} < T_c < 1630 \text{ K}$

$k_l = 64 \text{ W m}^{-1} \text{ K}^{-1}$ $\quad k_s = 64 \text{ W m}^{-1} \text{ K}^{-1}$

$\mu_l = 0.0032634 \text{ kg m}^{-1} \text{ s}^{-1}$ $\quad \mu s = 326.34 \text{ kg m}^{-1} \text{ s}^{-1}$

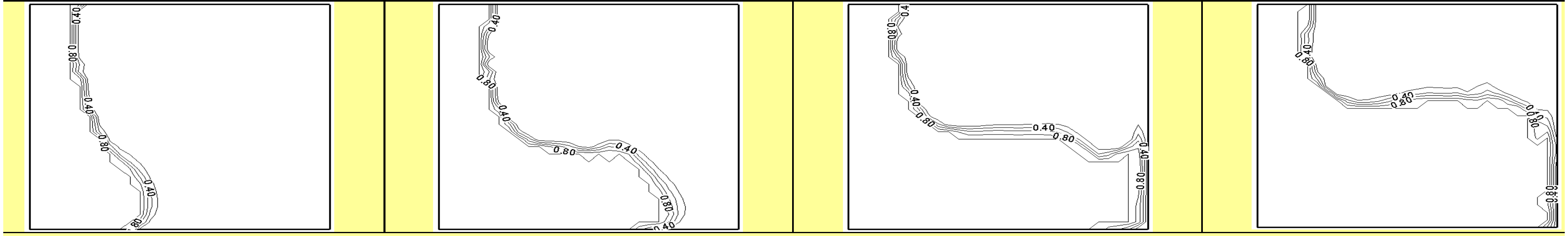
$\beta = 1.4 \text{ x } 10^{-4} \text{ K}^{-1}$ $\quad \beta s = 0.0875$

$g = 9.81 \text{ m s}^{-2}$ $\quad \mu_m = 1.2566 \text{ x } 10^{-5} \text{ T m A}^{-1}$

$Te = 1681 \text{ K}$ $\quad C_e = 0.8 \text{ kg m}^{-3}$

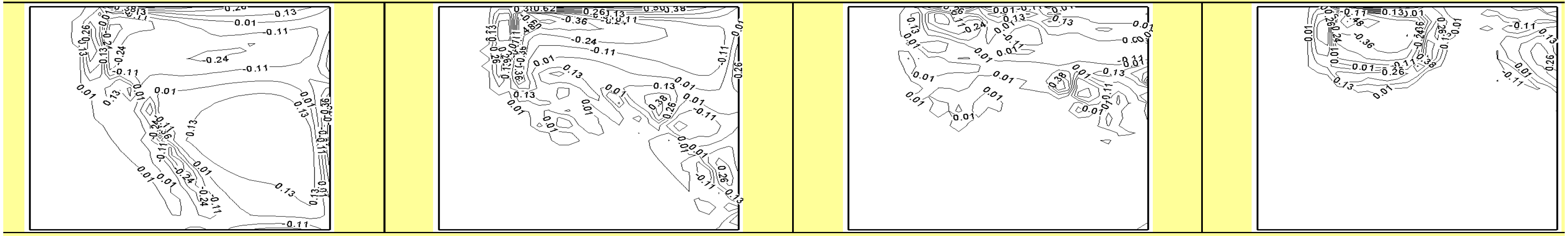
MHD – Example 6 (no magnetic fields)

15 min	30 min	45 min	60 min

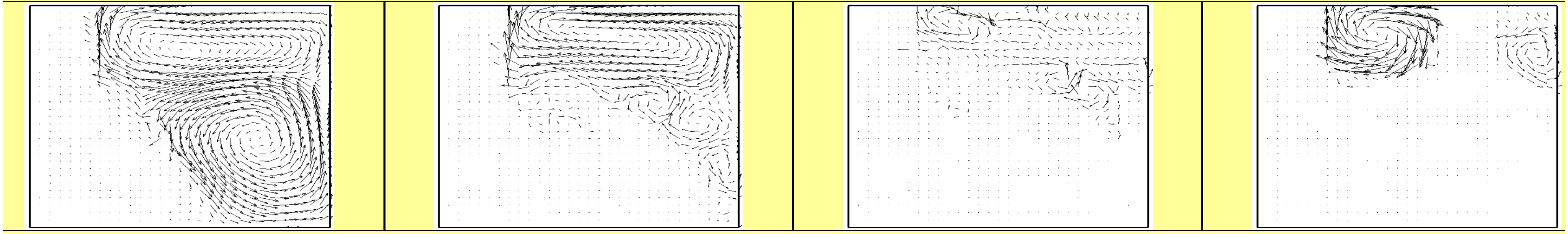


(a) Void fraction

(b) Concentration



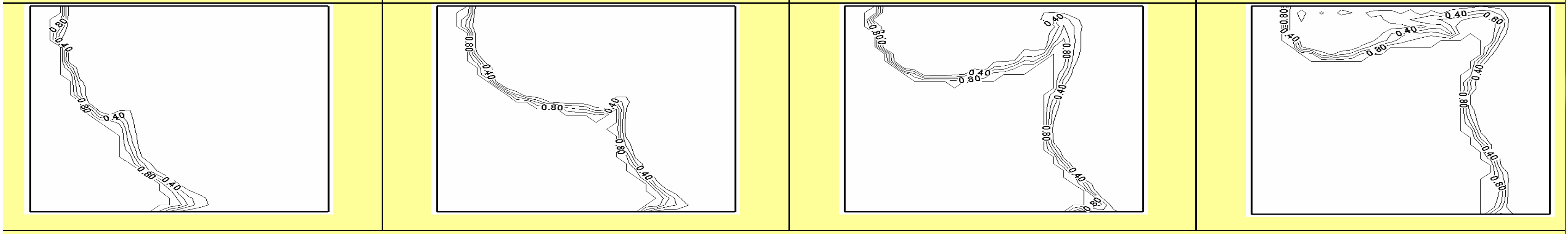
(c) Vorticity



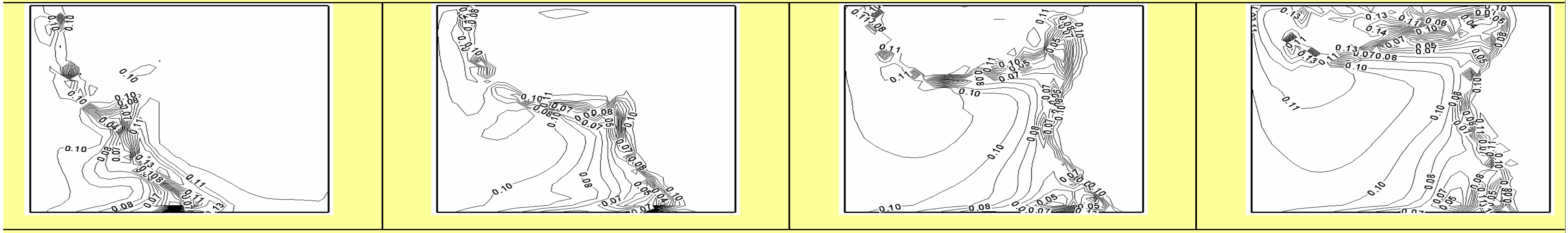
(e) Velocity vectors

MHD – Example 6 (optmized magnetic fields)

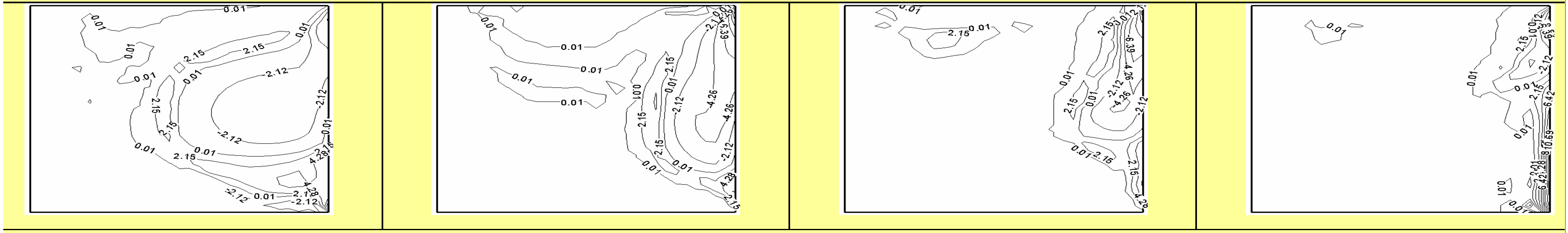
15 min **30 min** **45 min** **60 min**



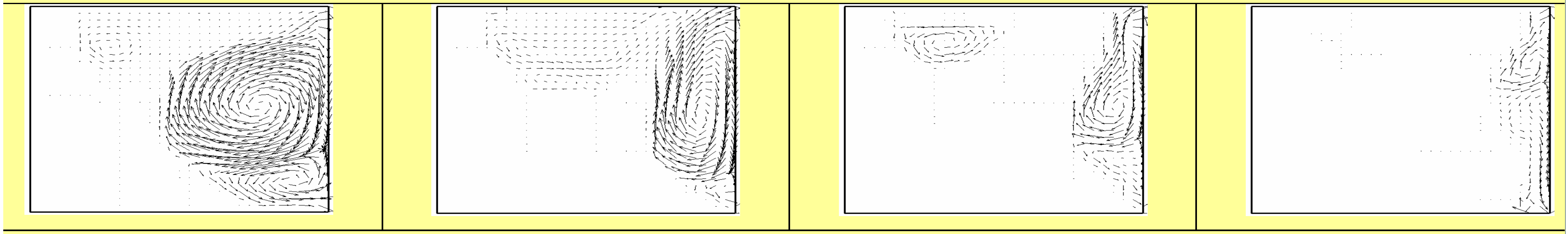
(a) Void fraction



(b) Concentration



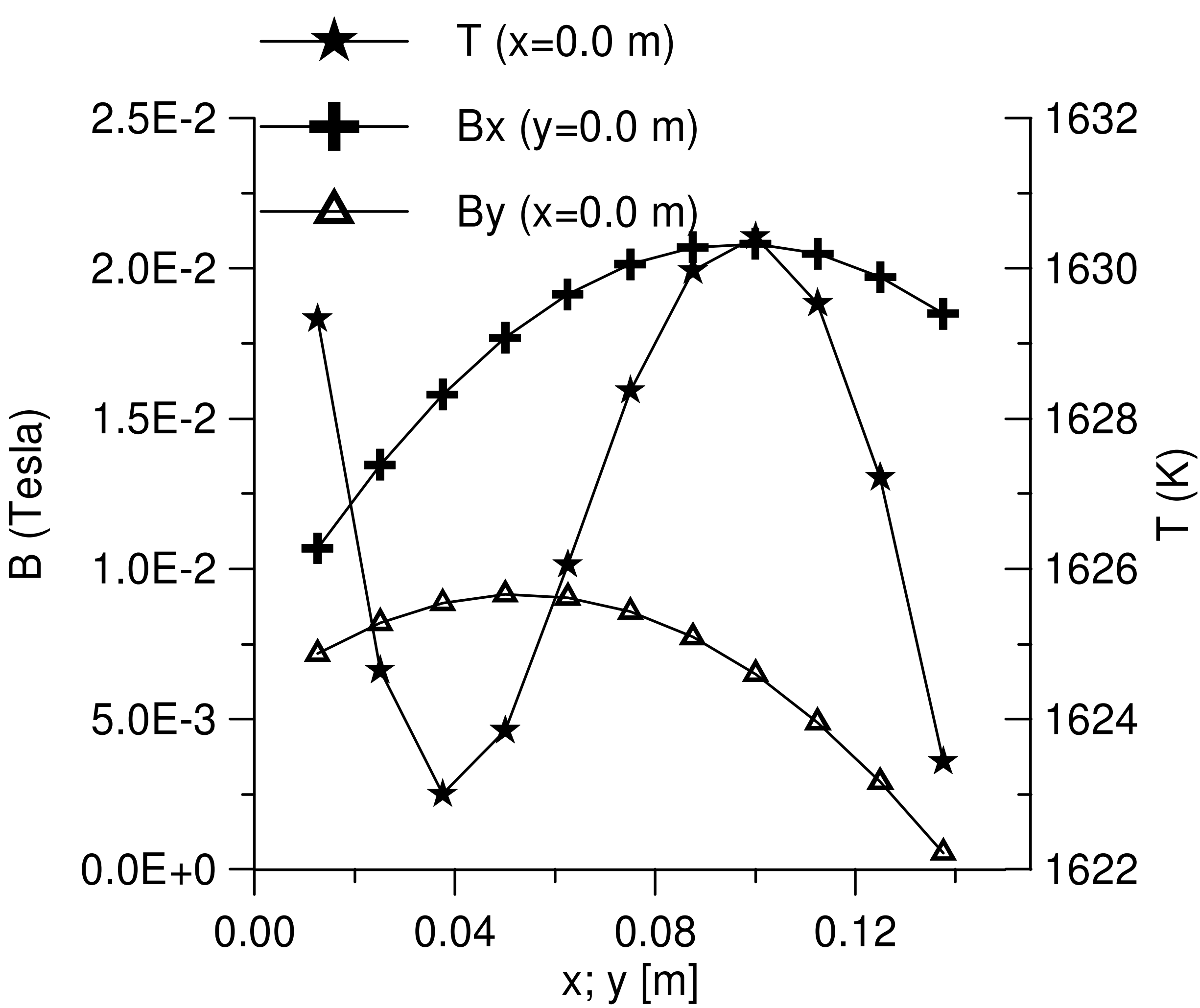
(c) Vorticity



(e) Velocity vectors

MHD – Example 6

- Optimized boundary conditions



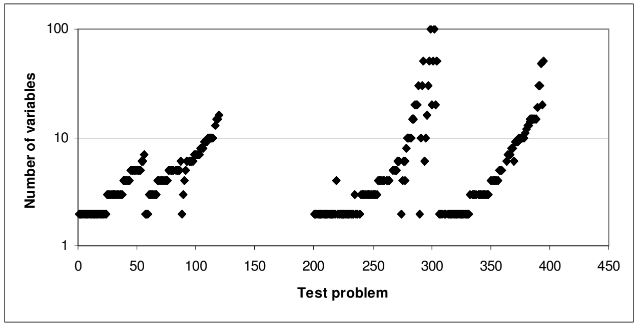
Summary of the results for the 295 test cases defined by Prof. Klaus Schittkowski

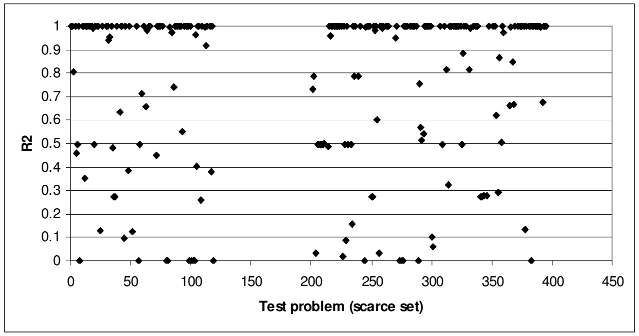
-K. Schittkowski (1987): More Test Examples for Nonlinear Programming, Lecture Notes in Economics and Mathematical Systems, Vol. 282, Springer

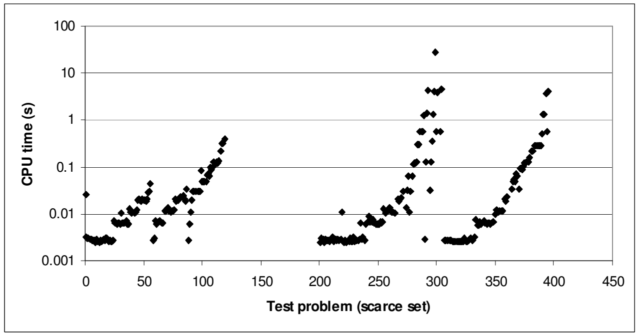
-W. Hock, K. Schittkowski (1981): Test Examples for Nonlinear Programming Codes, Lecture Notes in Economics and Mathematical Systems, Vol. 187, Springer

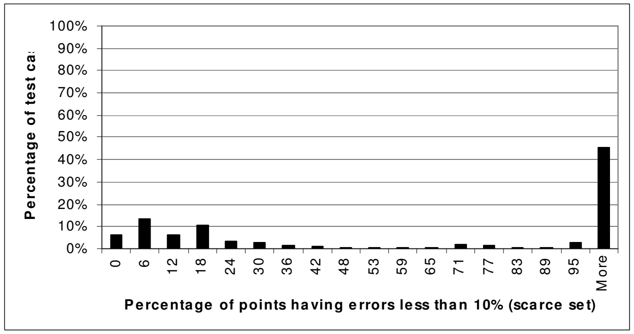
Scarce set – Number of training points = 3 times number of variables
Small set – Number of training points = 10 times number of variables
Medium set – Number of training points = 50 times number of variables
Number of testing points = 100 times number of training points

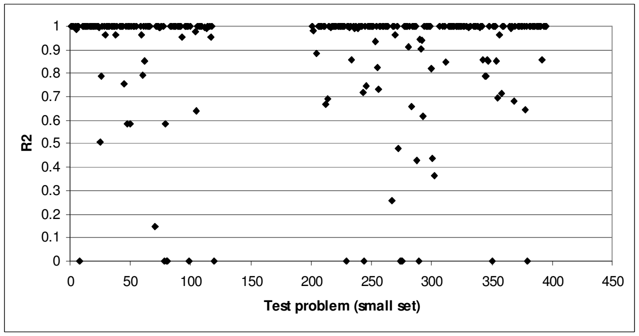
Polynomials up to order 6
Half of the training points used to validate the surface (ciclic)
Other preconditioning for the GMRES solver
1 RBFs used – Normalised and Auto scaled (SEPTEMBER, 2007)

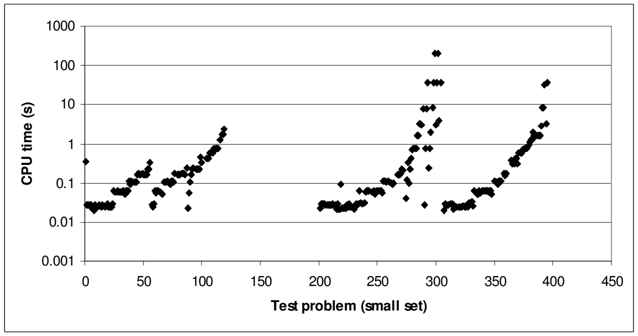


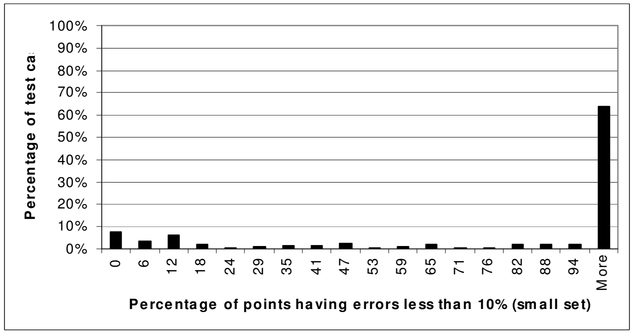


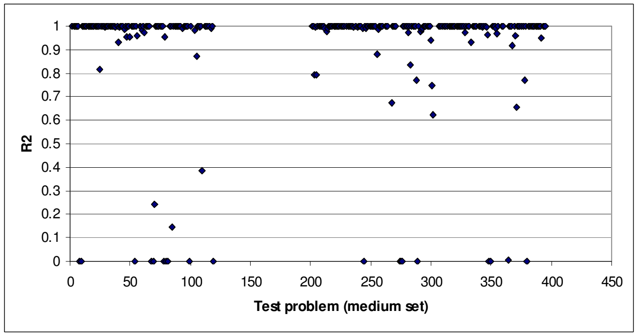


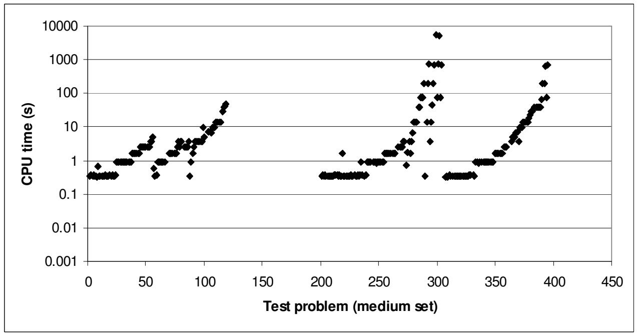


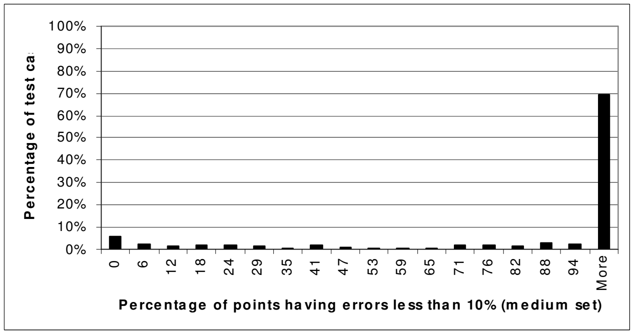










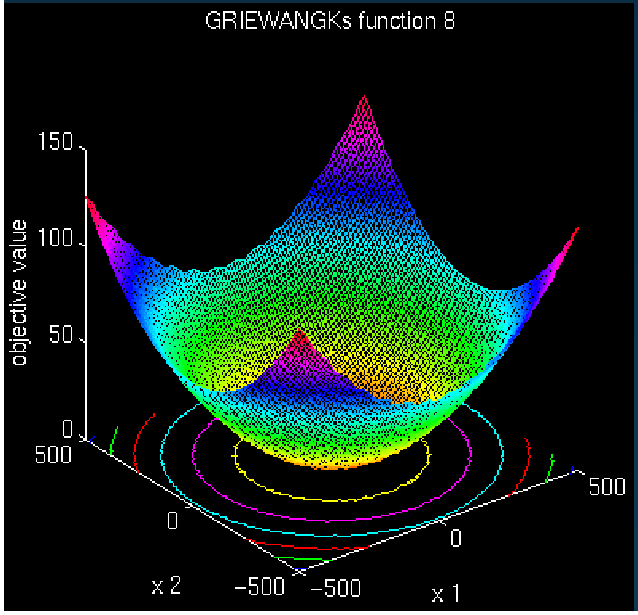


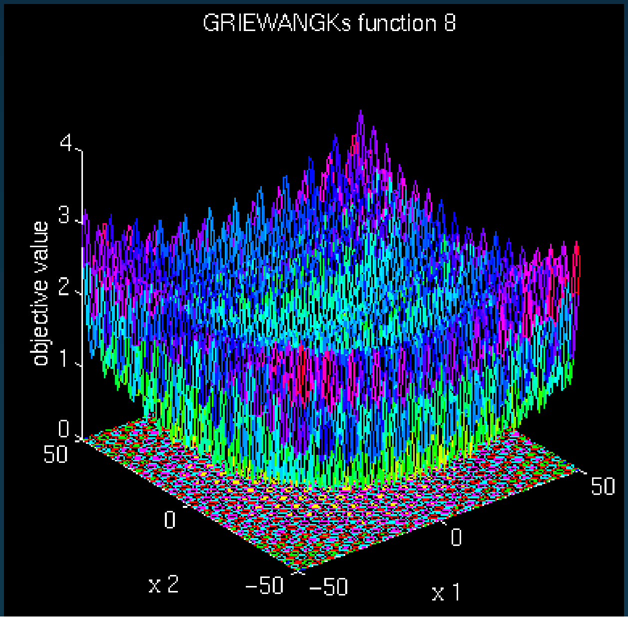
Using RBFs as a MetaModel in Optimization

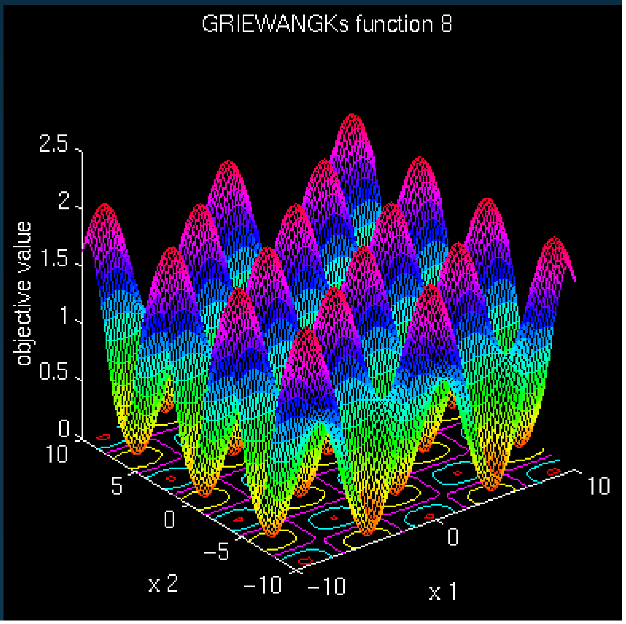
- ➔ Sample test function (Griewangk's function)

$$U = \sum_{i=1}^{n} \frac{x_i^2}{4000} - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$$

$$x \quad \in \quad]-600,600[$$



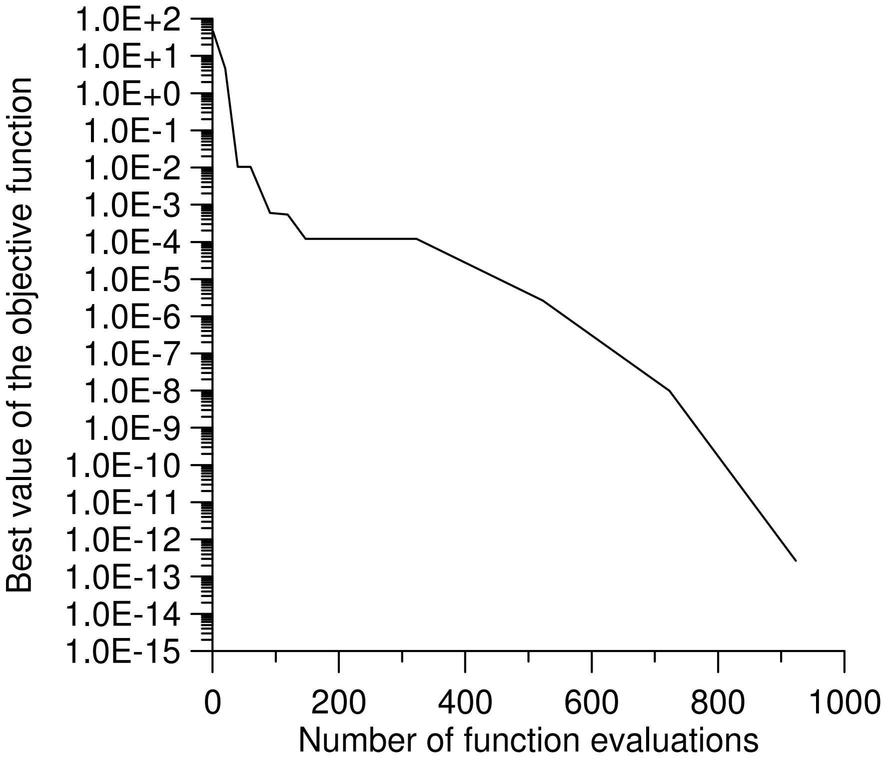




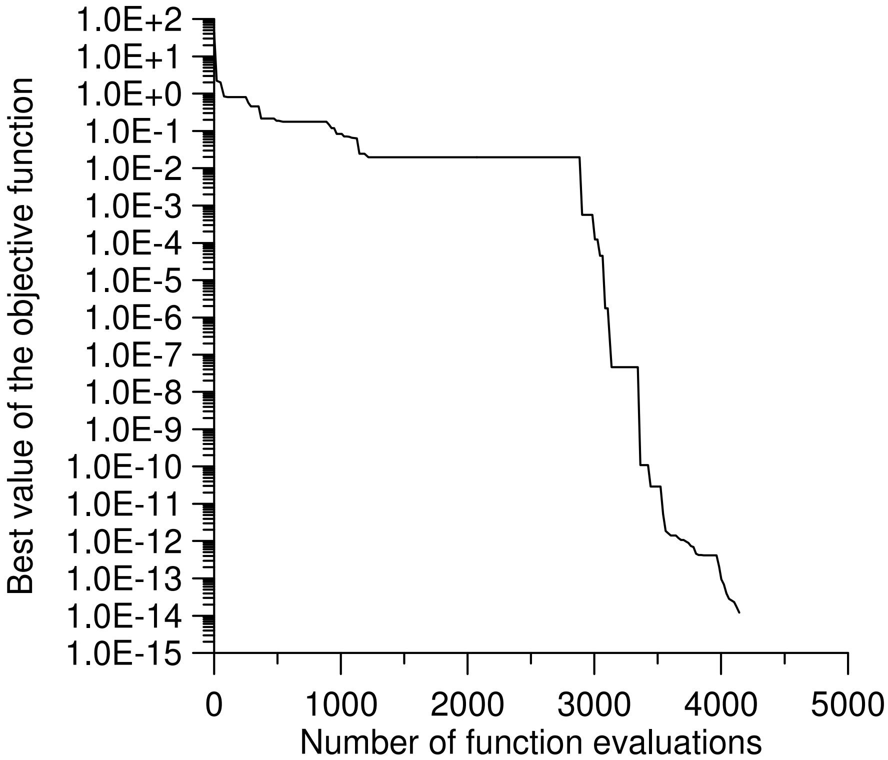
Using RBFs as a MetaModel in Optimization

- ➔ Optimization history (Hybrid Code)

with RBFs



without RBFs



Using RBFs as a MetaModel in Optimization

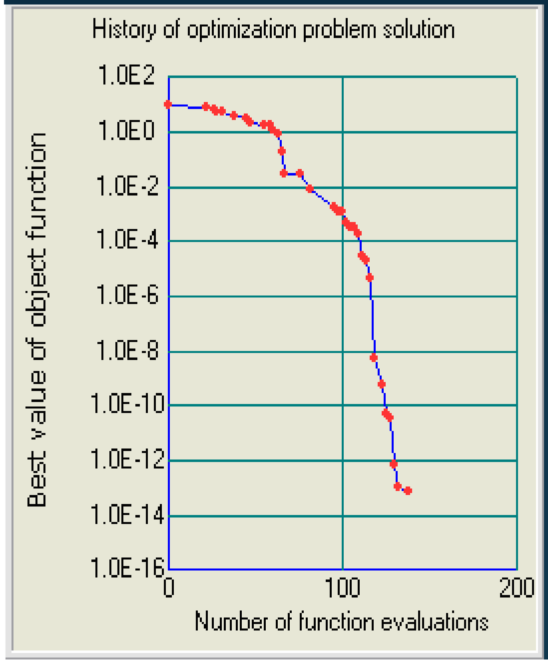
RBF model#2 with cross-validation – Will appear in IPSE Journal

- ➔ Comparison against commercial optimization codes
  - Levy#9 test function

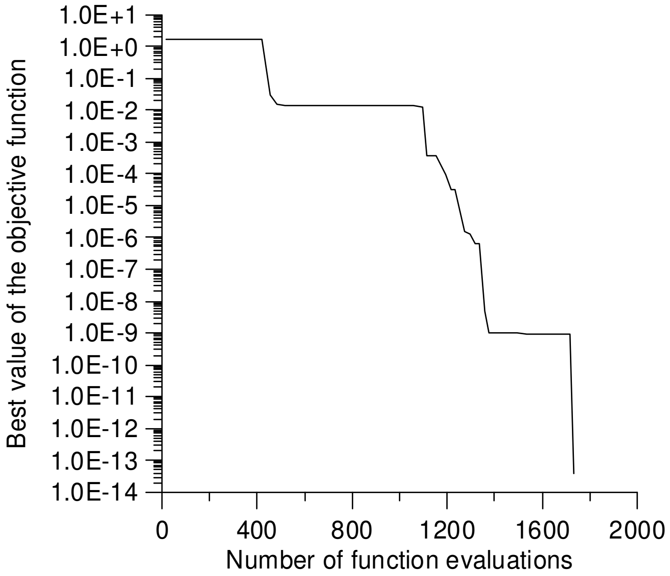
$$f(\mathbf{x}) = \sin^2(\pi - z_1) + \sum_{i=1}^{n-1}(z_i - 1)^2\left[1 + 10\sin^2(\pi z_{i+1})\right] + (z_4 - 1)^2$$

$$z_i = 1 + \frac{x_i - 1}{4}, (i = 1, 4)$$

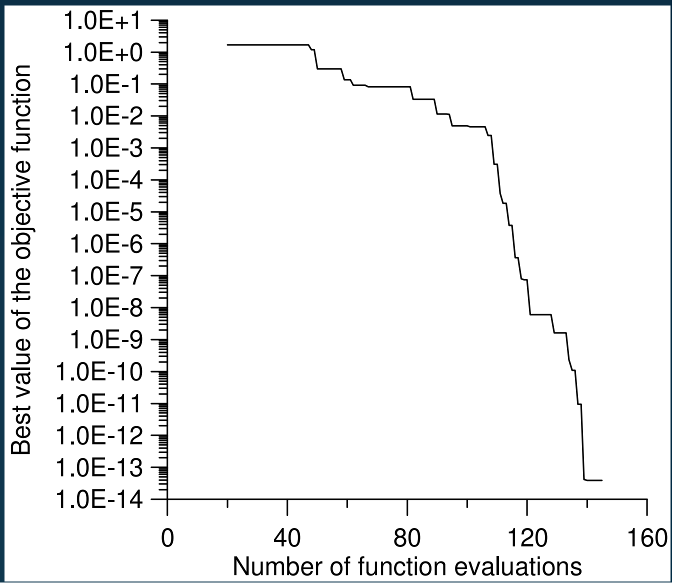
IOSO Code



Hybrid (no RBF)



Hybrid (with RBF)



Using RBFs as a MetaModel in Optimization

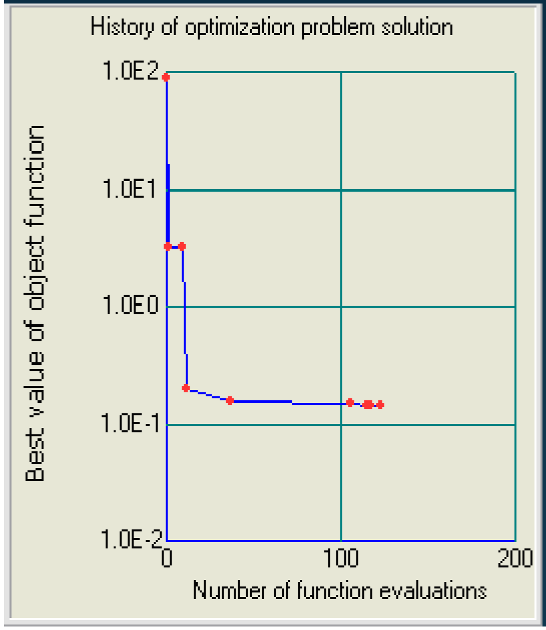
RBF model#2 with cross-validation – will appear in IPSE Journal

- Comparison against commercial codes
  - Griewangk's function

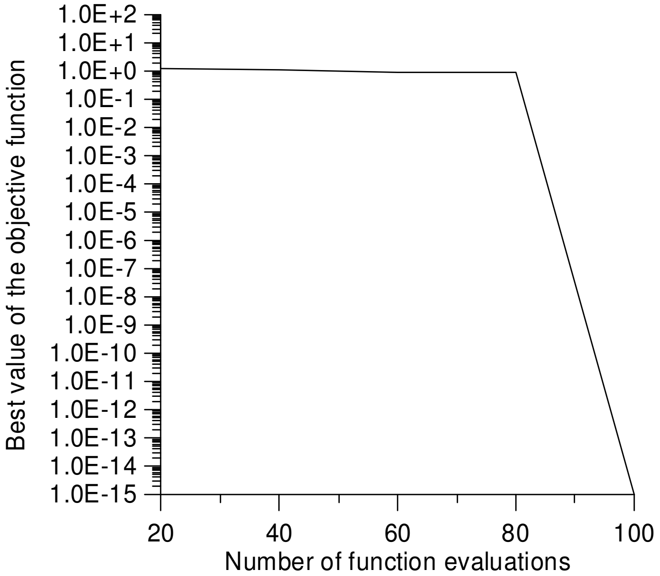
$$f(\mathbf{x}) = \sum_{i=1}^{n} \frac{x_i^2}{4000} - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$$

$$x_i \in \left] -600, 600 \right[\quad ,(i = 1,2)$$

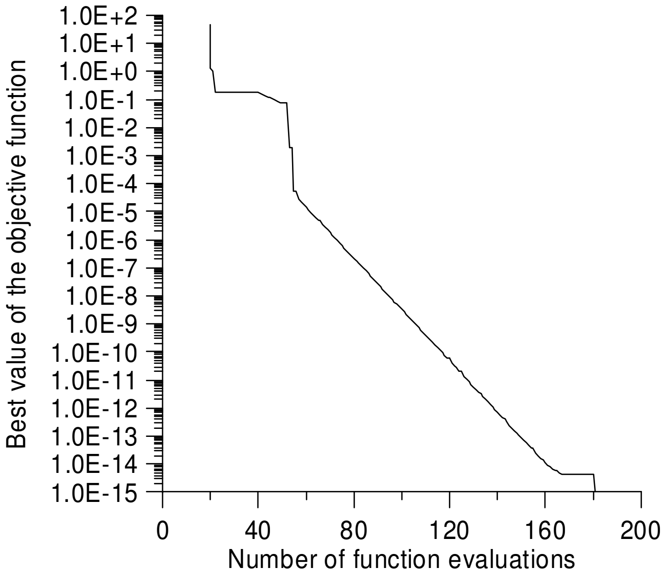
IOSO Code



Hybrid (no RBF)



Hybrid (with RBF)



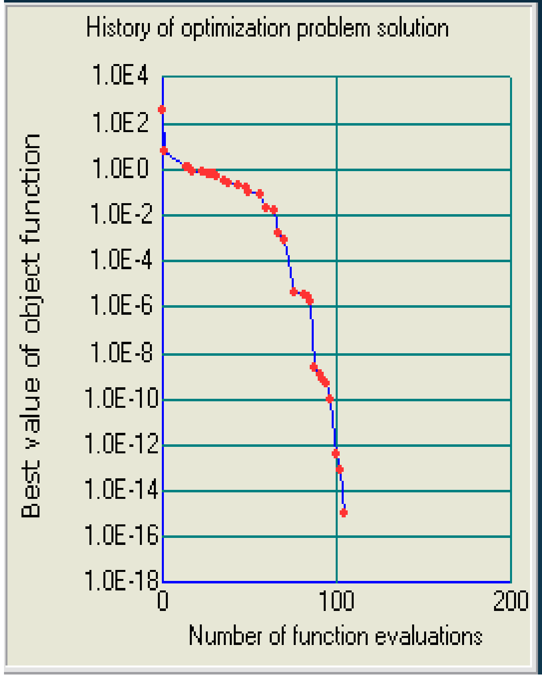
Using RBFs as a MetaModel in Optimization

RBF model#2 with cross-validation – will appear in IPSE Journal

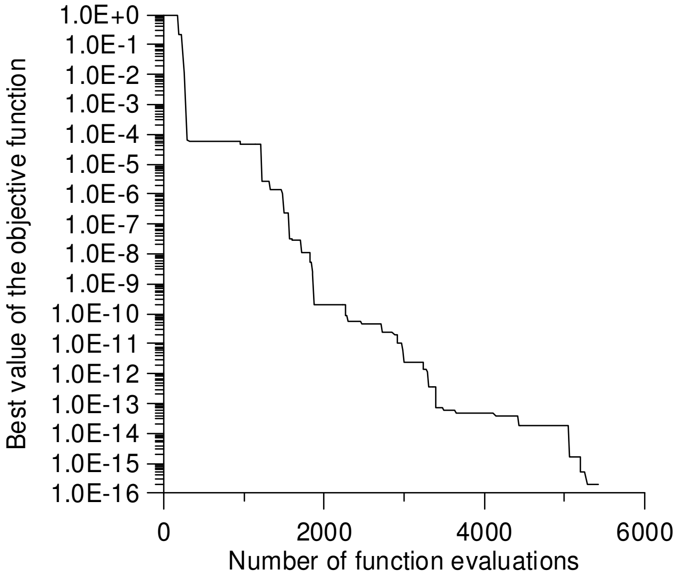
- ➔ Comparison against commercial codes
  - Rosenbrook’s test function

$$f\left(x_1,x_2\right)=100\left(x_2-x_1^2\right)^2+\left(1-x_1\right)^2$$

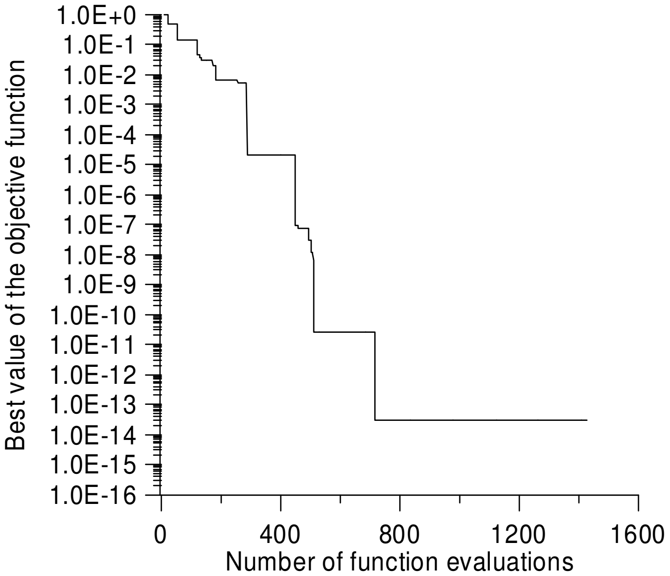
IOSO Code



Hybrid (no RBF)



Hybrid (with RBF)



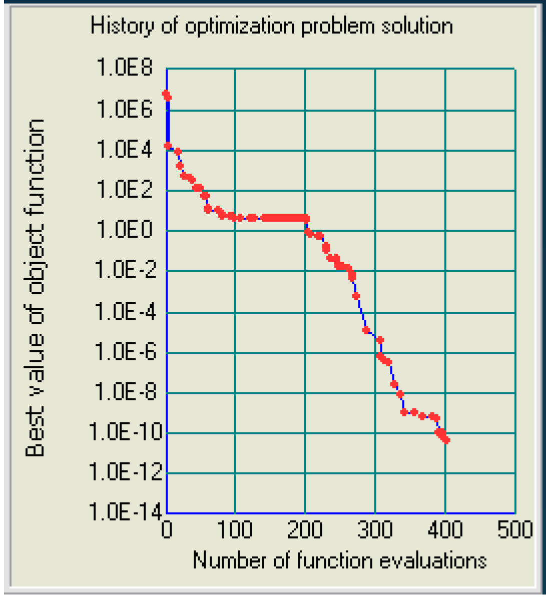
Using RBFs as a MetaModel in Optimization

RBF model#2 with cross-validation – will appear in IPSE Journal

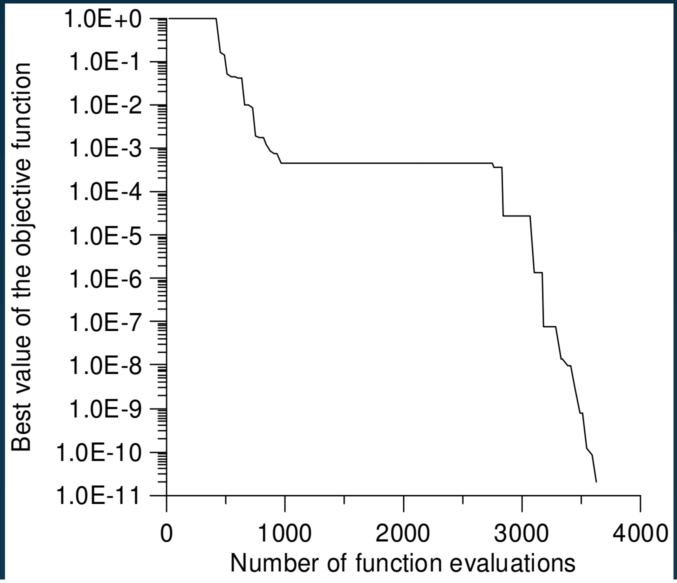
- Comparison against commercial codes
  - Mielle-Cantrel’s test function

$$f(\mathbf{x}) = \left[\exp^{(x_1-x_2)}\right]^4 + 100\left(x_2-x_3\right)^6 + \arctan^4\left(x_3-x_4\right) + x_1^2$$

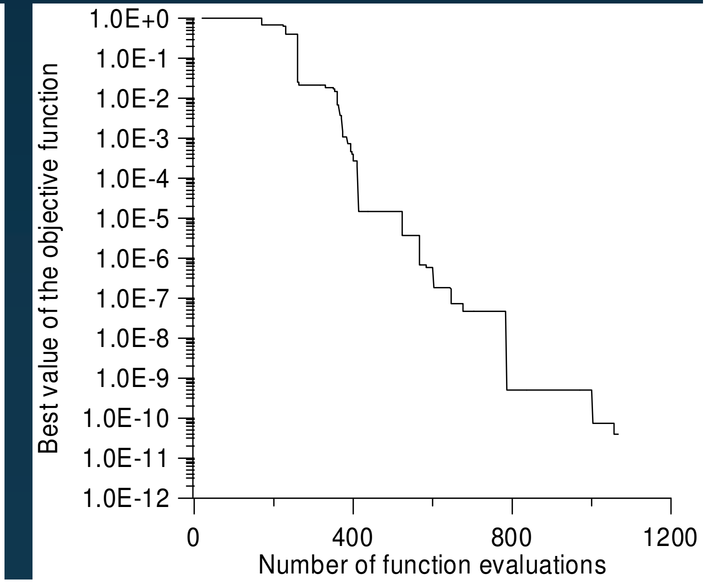
IOSO Code



Hybrid (no RBF)



Hybrid (with RBF)



Using RBFs as a MetaModel in Optimization

- ➔ RBF model used in Bayesian estimates – Helcio's presentation

- ➔ RBF model#3 with cross-validation – Not published yet (cooperation with Esteco company)

Acknowledgments

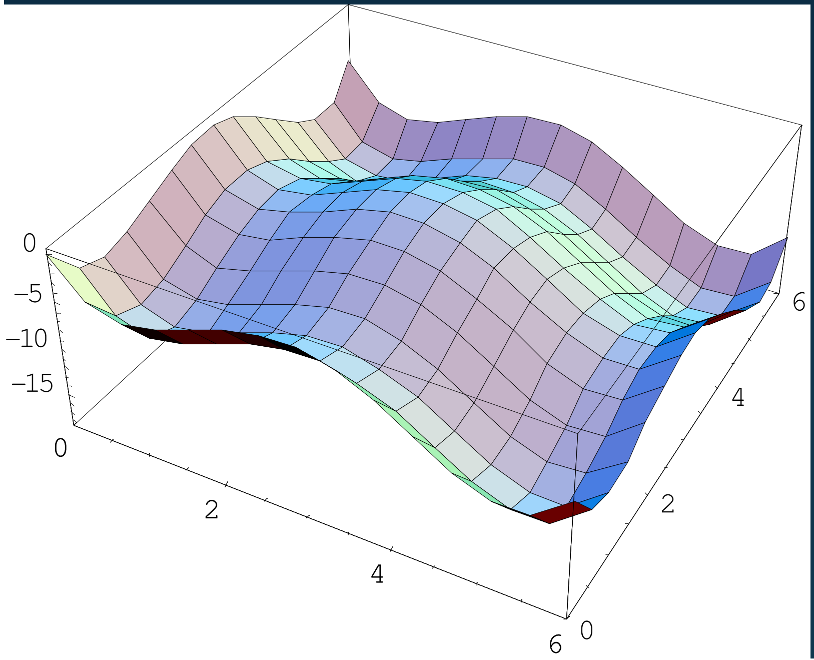
- Profª. Gloria Frontini
- CAPES and CNPq, two Brazilian agencies for scientific and technological development.
- IME, Military Institute of Engineering.
- UTA, University of Texas at Arlington.
- FIU, Florida International University.
- UFRJ, Federal University of Rio de Janeiro.

Other people that contributed in this work

- George S. Dulikravich, Florida International University
- Helcio R. B. Orlande, Federal University of Rio de Janeiro
- Brian H. Dennis, University of Texas at Arlington
- Thoman J. Martin, Pratt & Whitney Engine Group

Sample codes

- Minimization of Rosen’s function
  - Steepest Descent Method
  - Conjugate Gradient Method
  - Newton’s Method
  - Particle Swarm Method
- RBF
  - Multiquadrics in R^N



$$s(x_i) = f(x_i) = \sum_{j=1}^{N} \alpha_j \phi\left(\left|x_i - x_j\right|\right)$$

$$\phi\left(\left|x - x_j\right|\right) = \sqrt{\left(x - x_j\right)^2 + c_j^2}$$

Thank you!

Questions?