Multigrid and Domain Decomposition: Similarities and Differences

A Personal Story

Martin J. Gander
martin.gander@unige.ch

University of Geneva

CIRM, September 19th, 2019

Similarities

- Both are iterative methods for PDEs

Similarities

- Both are iterative methods for PDEs
- Both have dedicated conferences (19th MG Copper Mountain, EMG/IMG 14 Kunming, DD26 Hong Kong)

Similarities

- Both are iterative methods for PDEs
- Both have dedicated conferences (19th MG Copper Mountain, EMG/IMG 14 Kunming, DD26 Hong Kong)
- Both are 'optimal'

Definition 1.2 (Optimality). *An iterative method for the solution of a linear system is said to be optimal, if its rate of convergence to the exact solution is independent of the size of the system.*

(Toselli and Widlund 2004)

Similarities

- Both are iterative methods for PDEs
- Both have dedicated conferences (19th MG Copper Mountain, EMG/IMG 14 Kunming, DD26 Hong Kong)
- Both are 'optimal'

  Definition 1.2 (Optimality). *An iterative method for the solution of a linear system is said to be optimal, if its rate of convergence to the exact solution is independent of the size of the system.*

  (Toselli and Widlund 2004)

- For both the generalization to non-symmetric problems is not trivial, and the generalization to time harmonic wave propagation like Helmholtz is hard.

Similarities

- Both are iterative methods for PDEs
- Both have dedicated conferences (19th MG Copper Mountain, EMG/IMG 14 Kunming, DD26 Hong Kong)
- Both are 'optimal'

  Definition 1.2 (Optimality). *An iterative method for the solution of a linear system is said to be optimal, if its rate of convergence to the exact solution is independent of the size of the system.*

  (Toselli and Widlund 2004)
- For both the generalization to non-symmetric problems is not trivial, and the generalization to time harmonic wave propagation like Helmholtz is hard.
- Direct relation: J. Xu, Iterative Methods by Space Decomposition and Subspace Correction, SIREV, 1992

Multigrid

Multigrid is essentially a discrete method.

To solve a linear system

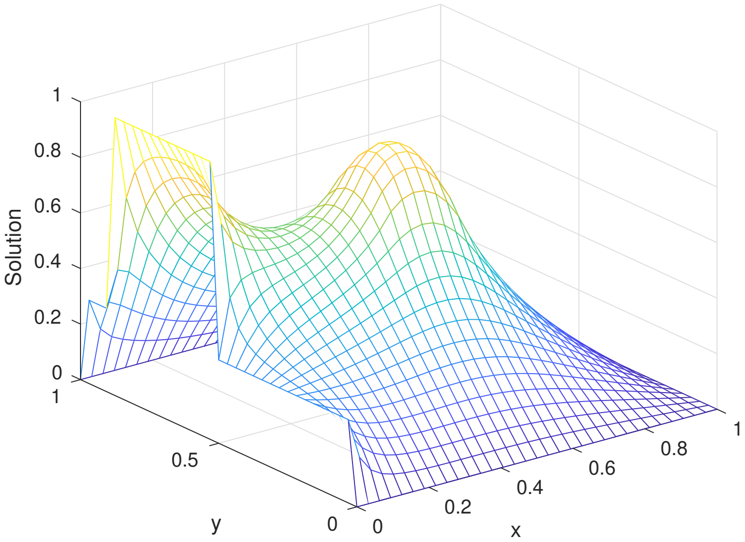
$$A\boldsymbol{u} = \boldsymbol{f}$$

which represents a discretized PDE, starting with an initial guess $\boldsymbol{u}^0$ one computes

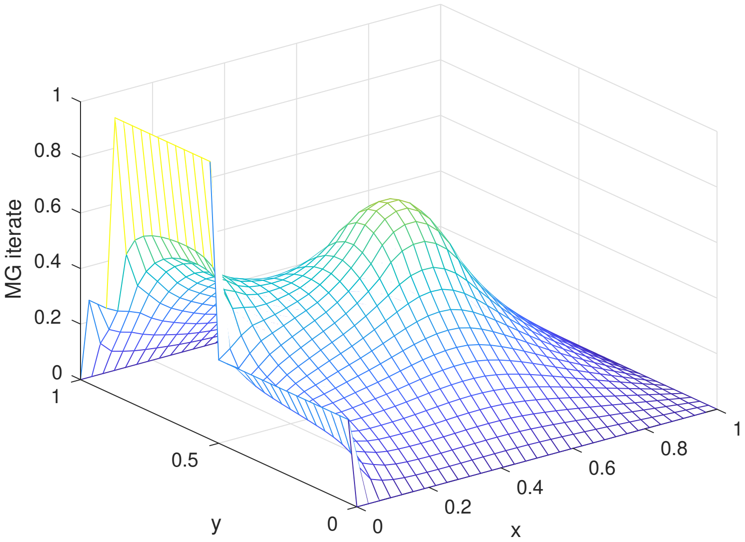
$$\begin{aligned}
\boldsymbol{u}^n &= S(A, \boldsymbol{u}^n, \boldsymbol{f}, \nu_1)\,; \\
\boldsymbol{u}^n &= \boldsymbol{u}^n + PA_c^{-1}R(\boldsymbol{f} - A\boldsymbol{u}^n)\,; \\
\boldsymbol{u}^{n+1} &= S(A, \boldsymbol{u}^n, \boldsymbol{f}, \nu_2)\,;
\end{aligned}$$

- Prolongation P (interpolation)
- Restriction $R = P^T$
- Coarse matrix $A_c = RAP$ (Galerkin)

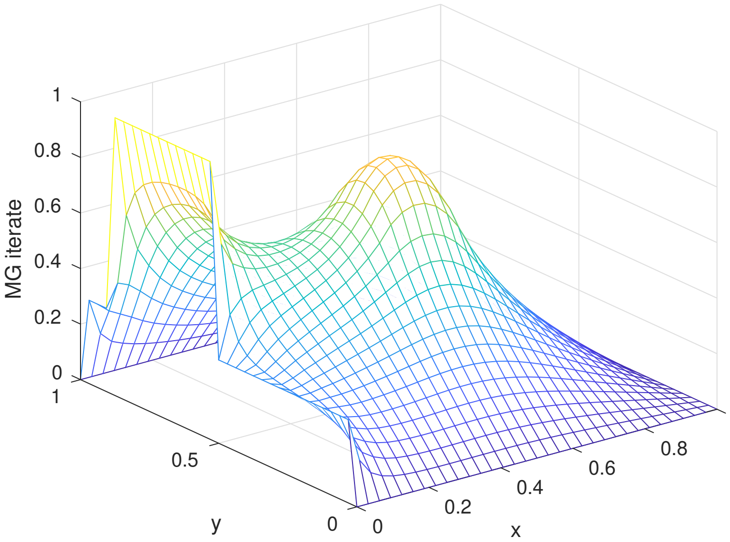
Example: temperature in a room



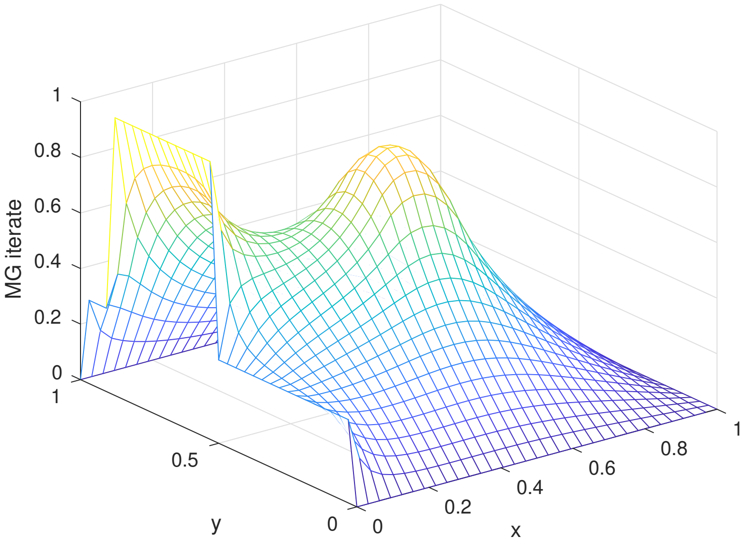
MG iteration 1



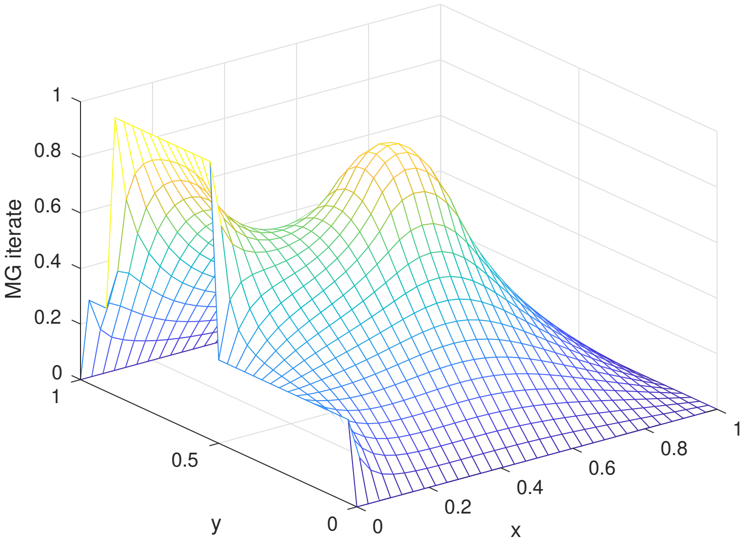
MG iteration 2



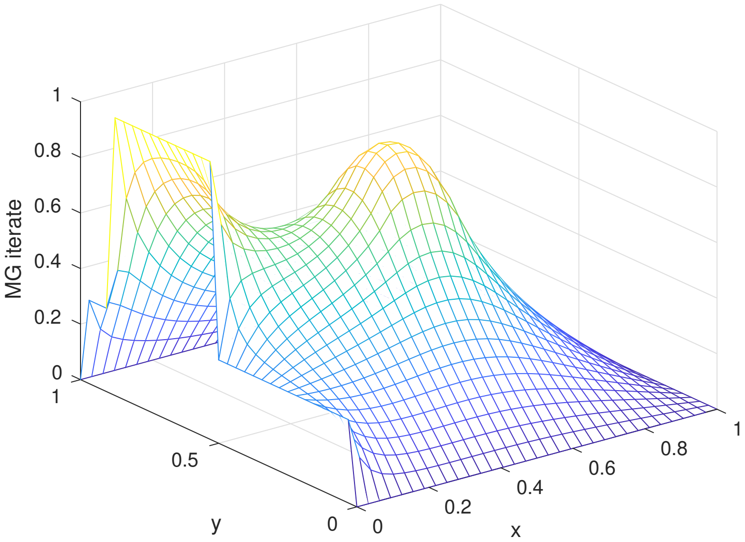
MG iteration 3



MG iteration 4



MG iteration 5



Convergence Results for Multigrid

Federenko (1964): Fourier analysis

Hackbusch (1979): Smoothing and approximation properties for stationary MG iteration

$$\boldsymbol{u}^{n+1} = G\boldsymbol{u}^n + \tilde{\boldsymbol{f}}$$

with the iteration matrix given by

$$G = (A^{-1} - PA_c^{-1}R)AS^{\nu}.$$

Theorem (Hackbusch 1979)

Contraction Estimate: *for all* $\kappa \in (0,1)$ *there exists* ν^* *s.t.*

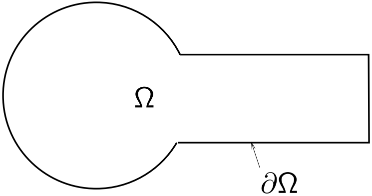
$$||G|| \leq \kappa < 1.$$

Griebel (PhD thesis 1990): 23 floating point operations per grid point and V-cycle!

Domain Decomposition

Domain decomposition methods are essentially continuous methods.

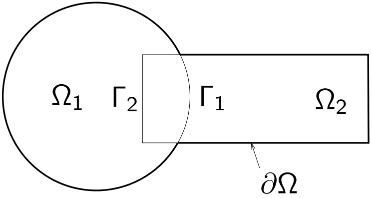
Alternating Schwarz: for $\Delta u = 0$ in Ω, $u = g$ on $\partial\Omega$



Domain Decomposition

Domain decomposition methods are essentially continuous methods.

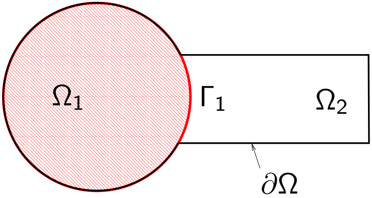
Alternating Schwarz: for $\Delta u = 0$ in Ω, $u = g$ on $\partial\Omega$



Domain Decomposition

Domain decomposition methods are essentially continuous methods.

Alternating Schwarz: for $\Delta u = 0$ in Ω, $u = g$ on $\partial\Omega$



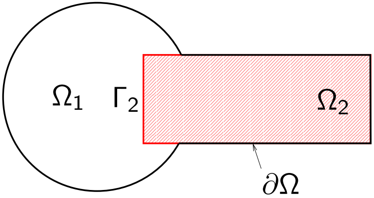
$$\begin{aligned} \Delta u_1^1 &= 0 && \text{in } \Omega_1 \\ u_1^1 &= g && \text{on } \partial\Omega \cap \overline{\Omega}_1 \\ u_1^1 &= u_2^0 && \text{on } \Gamma_1 \end{aligned}$$

solve on the disk $u_2^0 = 0$

Domain Decomposition

Domain decomposition methods are essentially continuous methods.

Alternating Schwarz: for $\Delta u = 0$ in Ω, $u = g$ on $\partial\Omega$



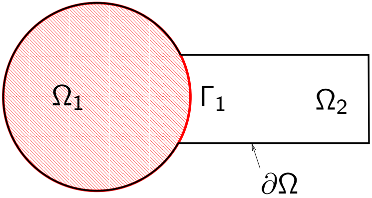
$$\begin{aligned} \Delta u_2^1 &= 0 && \text{in } \Omega_2 \\ u_2^1 &= g && \text{on } \partial\Omega \cap \overline{\Omega}_2 \\ u_2^1 &= u_1^1 && \text{on } \Gamma_2 \end{aligned}$$

solve on the rectangle

Domain Decomposition

Domain decomposition methods are essentially continuous methods.

Alternating Schwarz: for $\Delta u = 0$ in Ω, $u = g$ on $\partial\Omega$



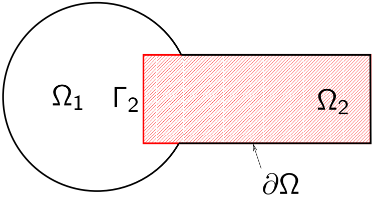
$$\begin{aligned} \Delta u_1^2 &= 0 && \text{in } \Omega_1 \\ u_1^2 &= g && \text{on } \partial\Omega \cap \overline{\Omega}_1 \\ u_1^2 &= u_2^1 && \text{on } \Gamma_1 \end{aligned}$$

solve on the disk

Domain Decomposition

Domain decomposition methods are essentially continuous methods.

Alternating Schwarz: for $\Delta u = 0$ in Ω, $u = g$ on $\partial\Omega$



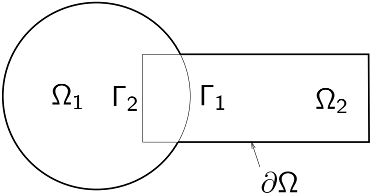
$$\begin{aligned} \Delta u_2^2 &= 0 \quad \text{in } \Omega_2 \\ u_2^2 &= g \quad \text{on } \partial\Omega \cap \overline{\Omega}_2 \\ u_2^2 &= u_1^2 \text{ on } \Gamma_2 \end{aligned}$$

solve on the rectangle

Domain Decomposition

Domain decomposition methods are essentially continuous methods.

Alternating Schwarz: for $\Delta u = 0$ in Ω, $u = g$ on $\partial\Omega$



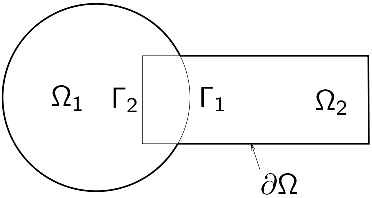
$$\begin{array}{rll} \Delta u_1^n = 0 & \text{in } \Omega_1 \\ u_1^n = g & \text{on } \partial\Omega \cap \overline{\Omega}_1 \\ u_1^n = u_2^{n-1} & \text{on } \Gamma_1 \end{array} \qquad \begin{array}{rll} \Delta u_2^n = 0 & \text{in } \Omega_2 \\ u_2^n = g & \text{on } \partial\Omega \cap \overline{\Omega}_2 \\ u_2^n = u_1^n & \text{on } \Gamma_2 \end{array}$$

solve on the disk　　　　solve on the rectangle

Domain Decomposition

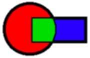
Domain decomposition methods are essentially continuous methods.

Alternating Schwarz: for $\Delta u = 0$ in Ω, $u = g$ on $\partial\Omega$



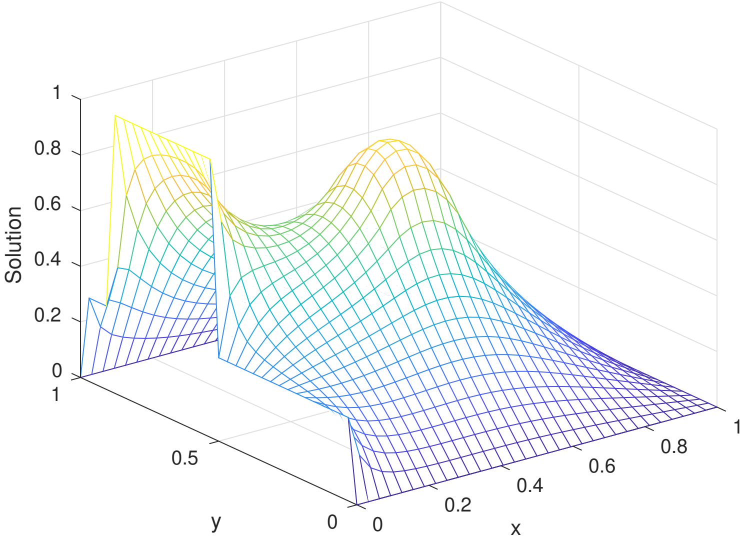
$$\begin{aligned} \Delta u_1^n &= 0 && \text{in } \Omega_1 \\ u_1^n &= g && \text{on } \partial\Omega \cap \overline{\Omega}_1 \\ u_1^n &= u_2^{n-1} && \text{on } \Gamma_1 \end{aligned} \qquad \begin{aligned} \Delta u_2^n &= 0 && \text{in } \Omega_2 \\ u_2^n &= g && \text{on } \partial\Omega \cap \overline{\Omega}_2 \\ u_2^n &= u_1^{\mathbf{n-1}} && \text{on } \Gamma_2 \end{aligned}$$

solve on the disk   solve on the rectangle

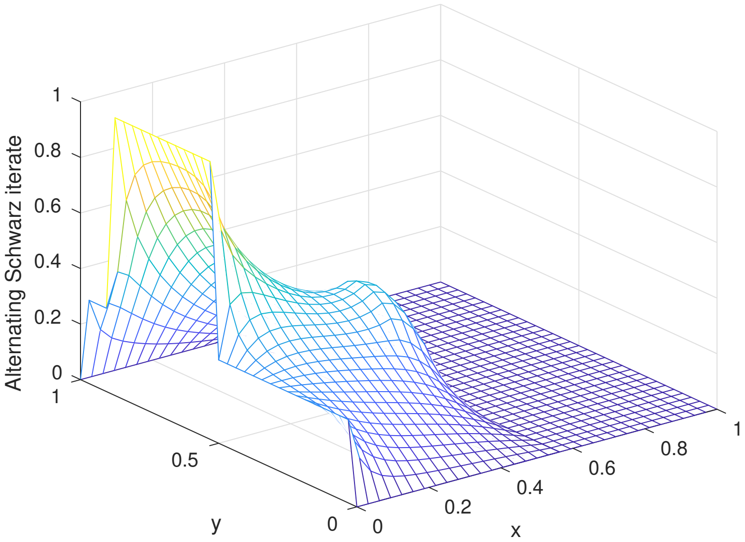


Lions introduced in 1988 the now called **Parallel Schwarz Method**

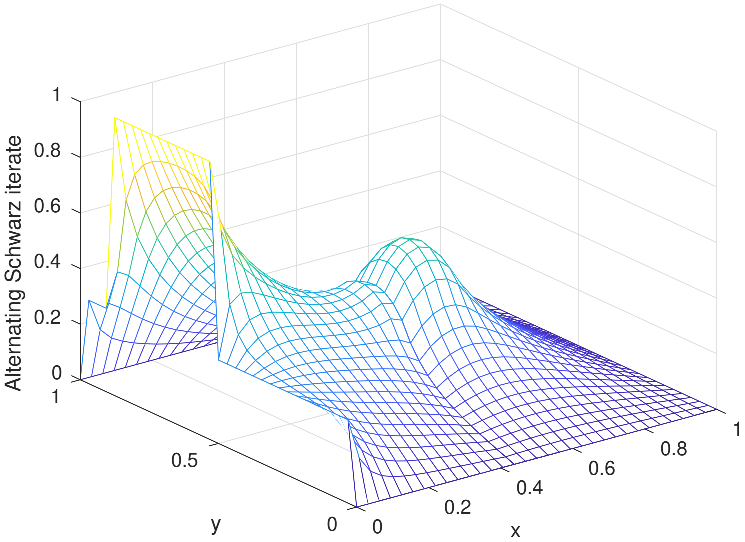
Example: temperature in a room



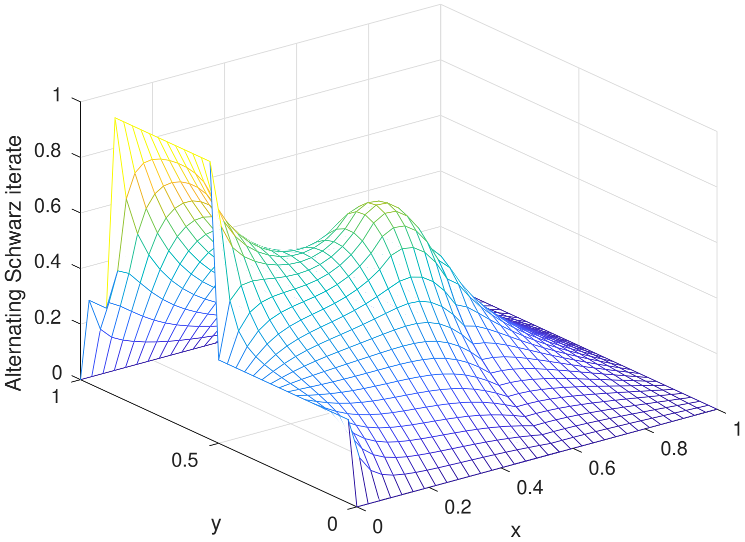
Alternating Schwarz iteration 1 on Ω_1



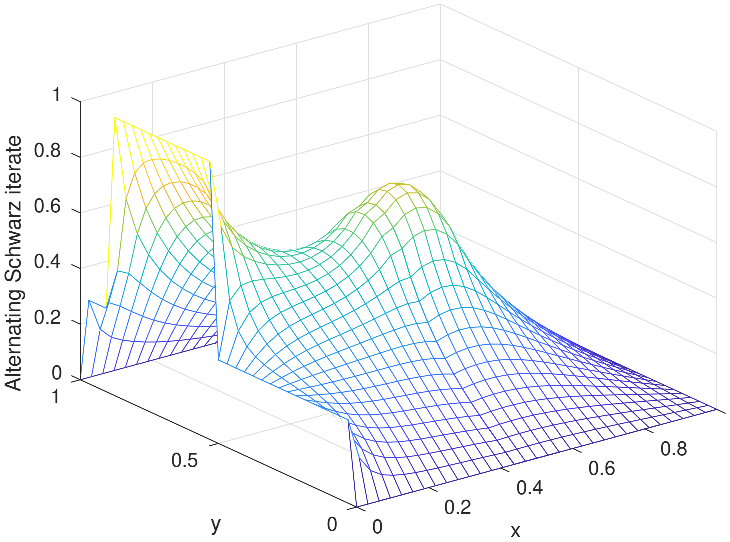
Alternating Schwarz iteration 1 on Ω_2



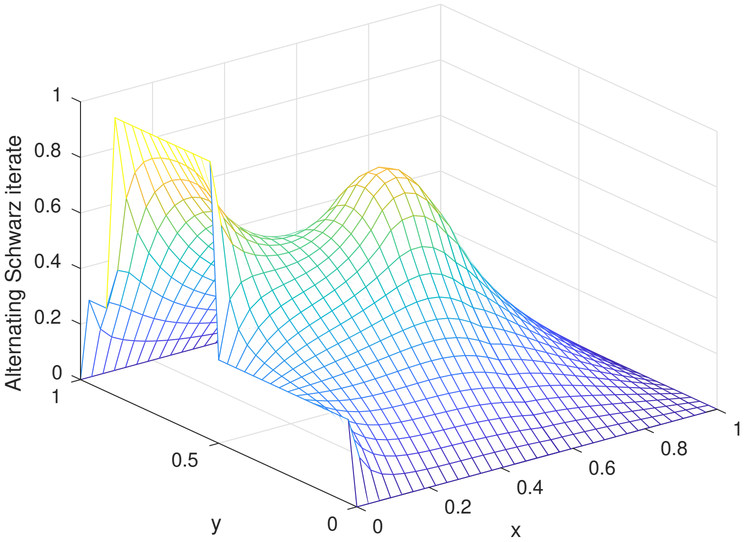
Alternating Schwarz iteration 2 on Ω_1



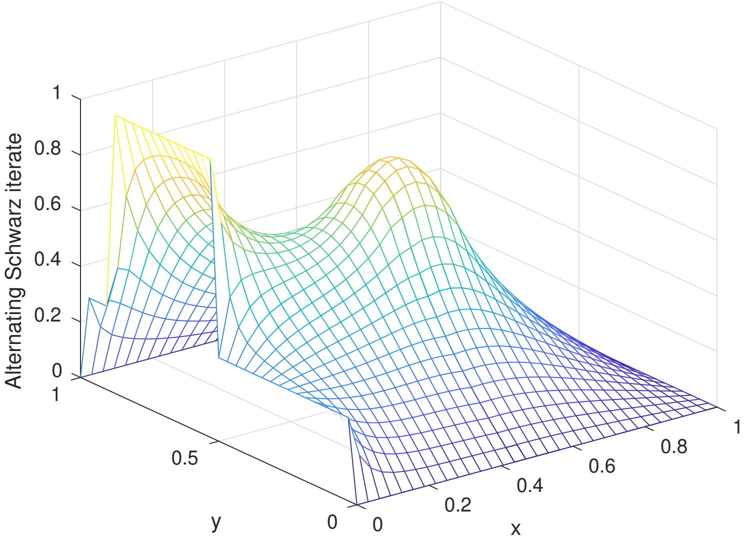
Alternating Schwarz iteration 2 on Ω_2



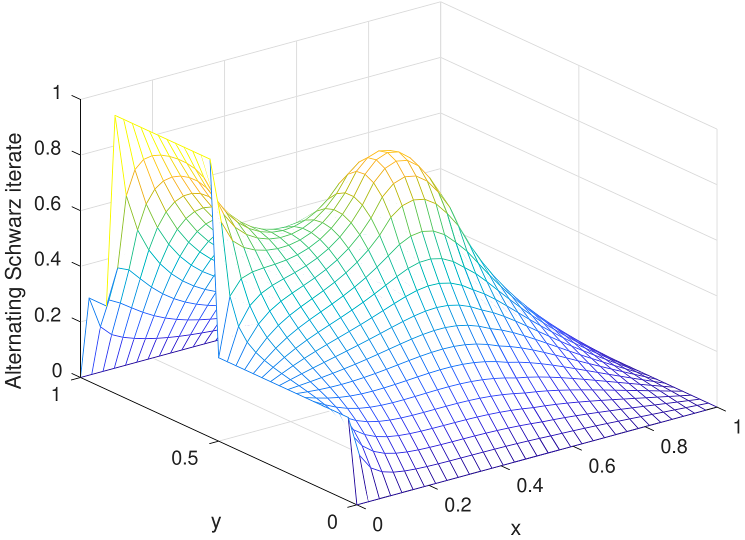
Alternating Schwarz iteration 3 on Ω_1



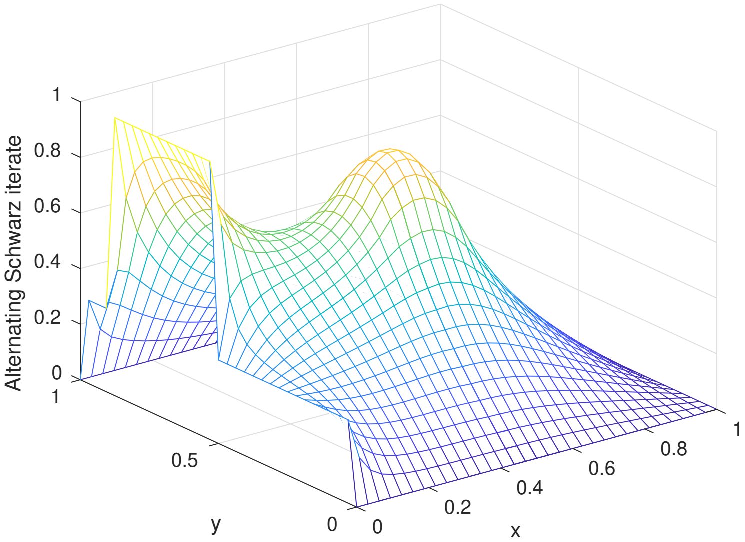
Alternating Schwarz iteration 3 on Ω_2



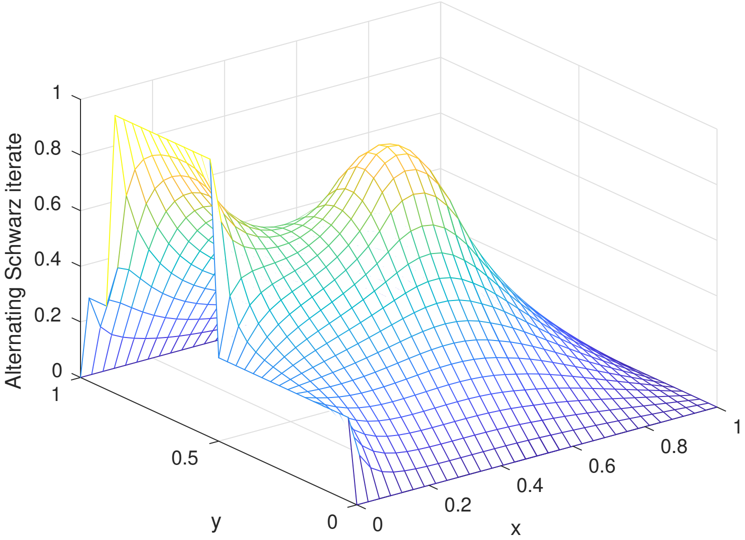
Alternating Schwarz iteration 4 on Ω_1



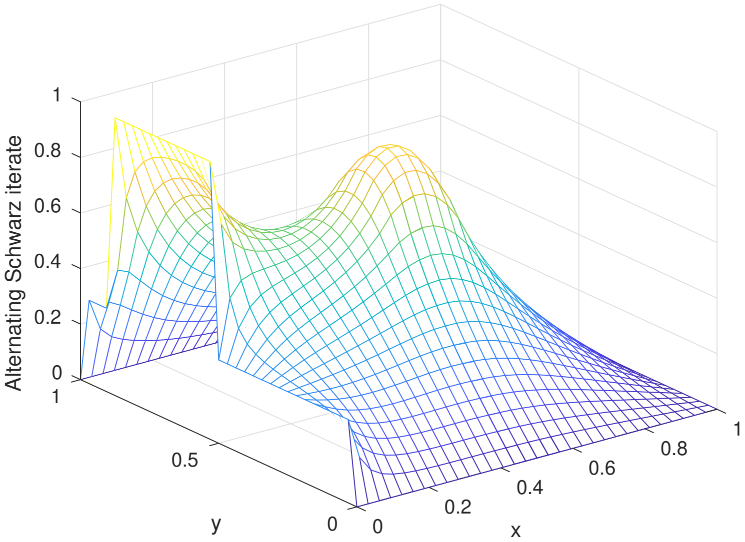
Alternating Schwarz iteration 4 on Ω_2



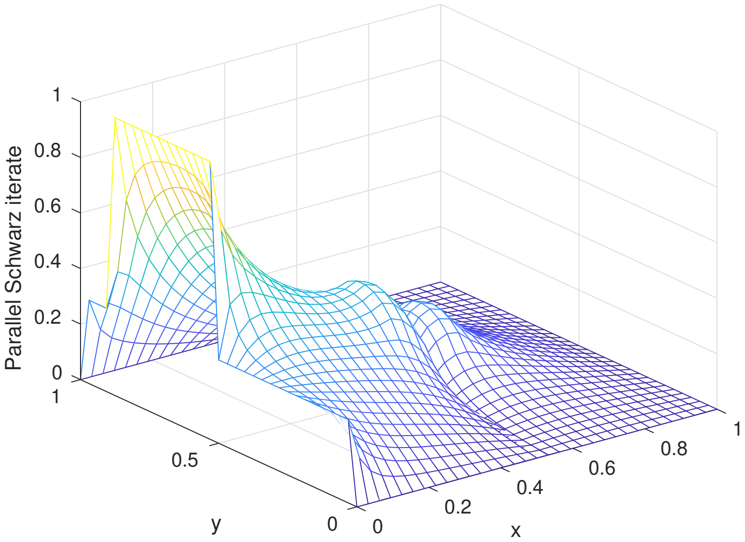
Alternating Schwarz iteration 5 on Ω_1



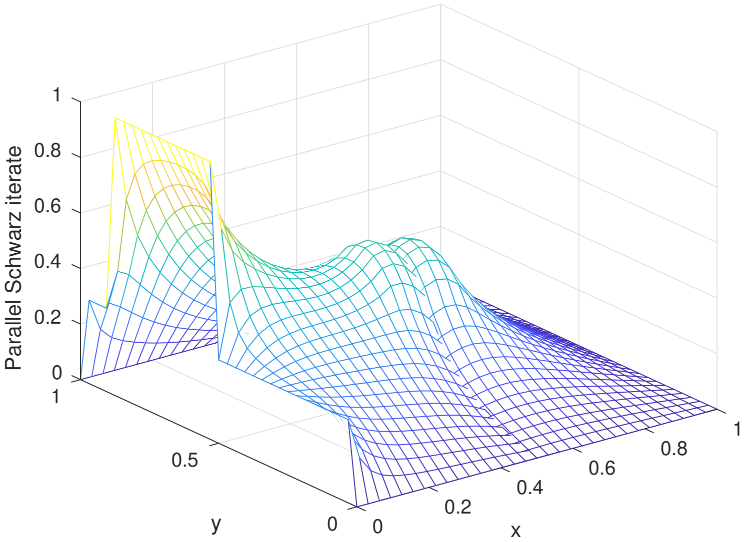
Alternating Schwarz iteration 5 on Ω_2



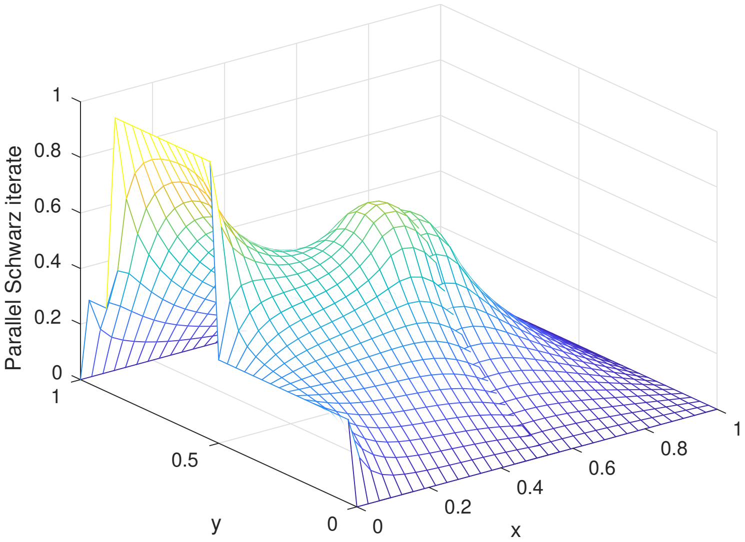
Parallel Schwarz iteration 1



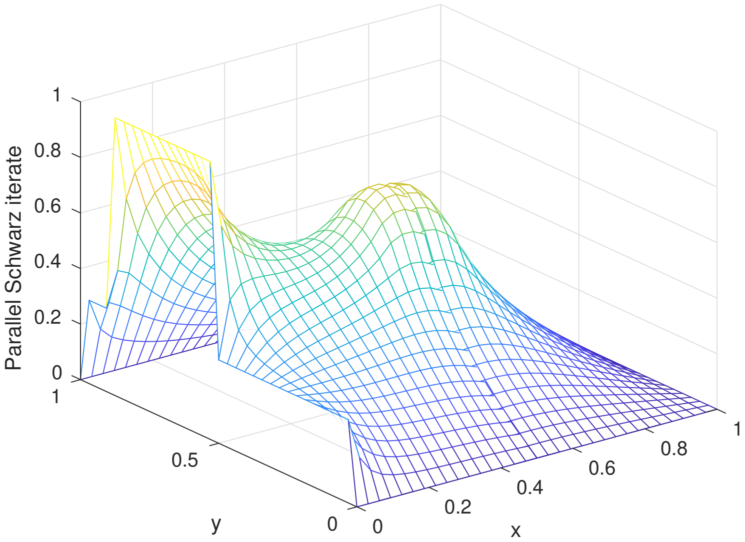
Parallel Schwarz iteration 2



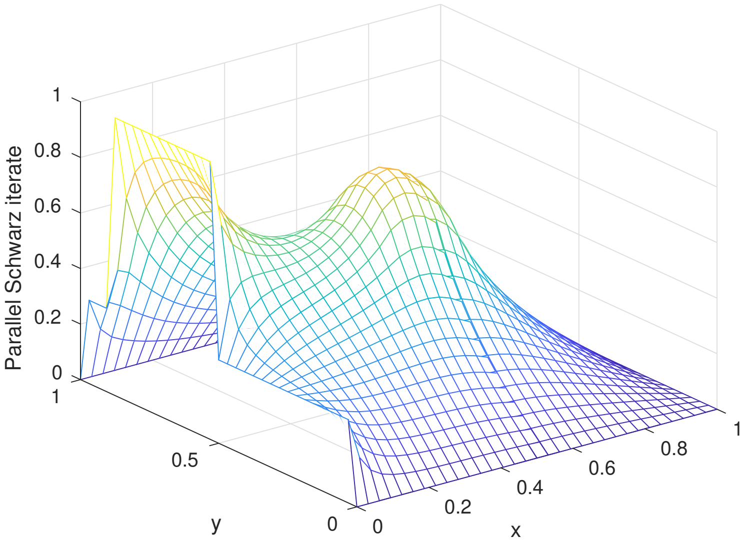
Parallel Schwarz iteration 3



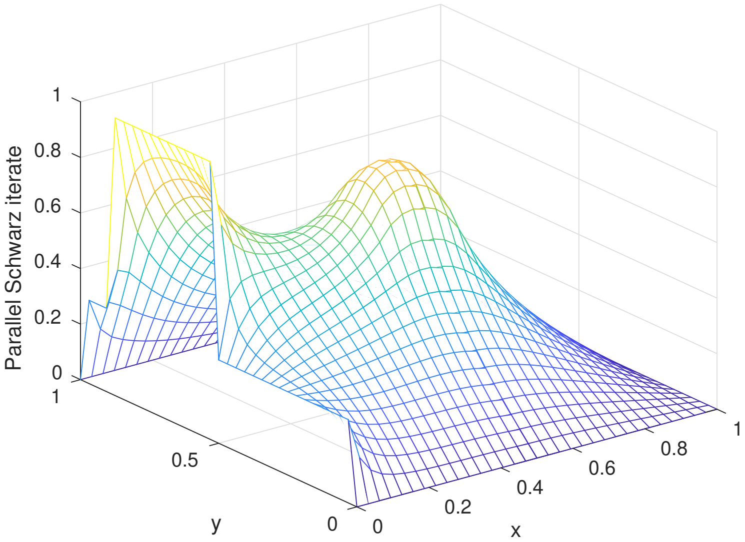
Parallel Schwarz iteration 4



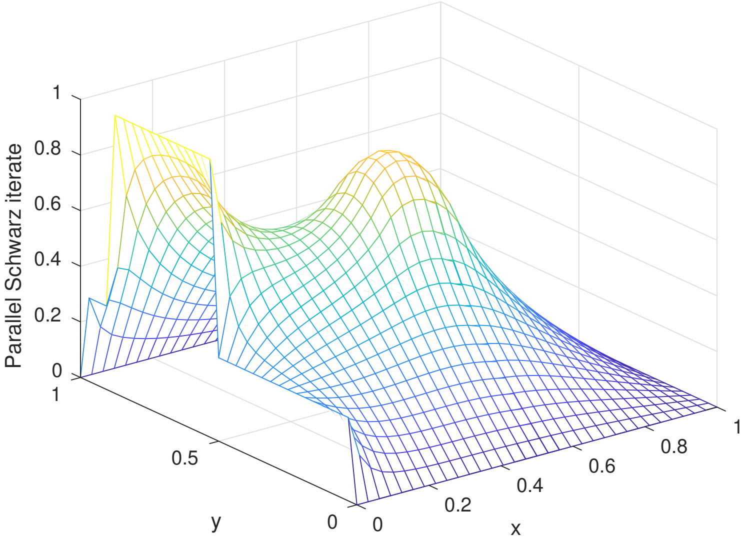
Parallel Schwarz iteration 5



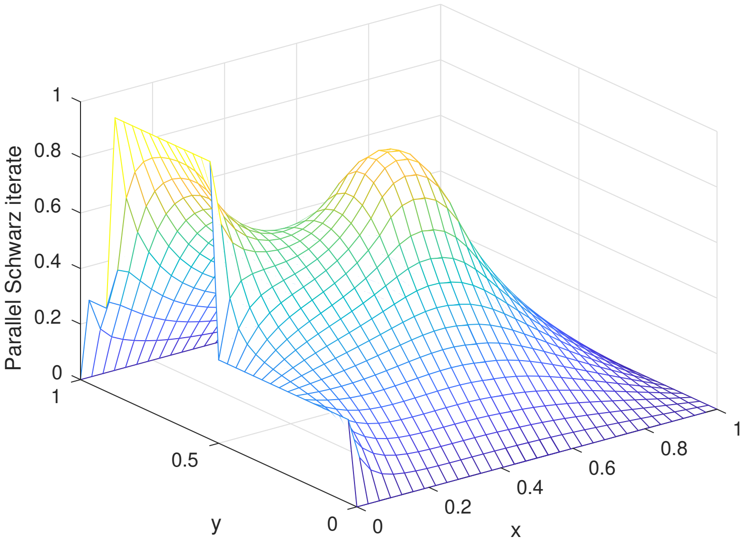
Parallel Schwarz iteration 6



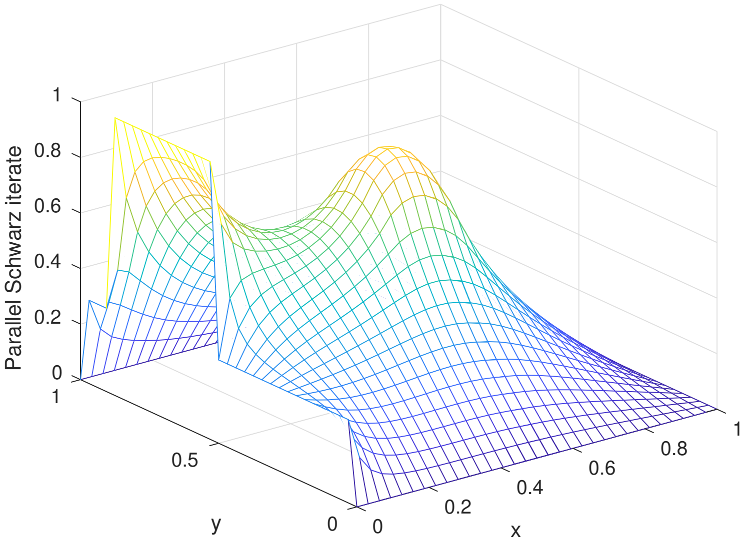
Parallel Schwarz iteration 7



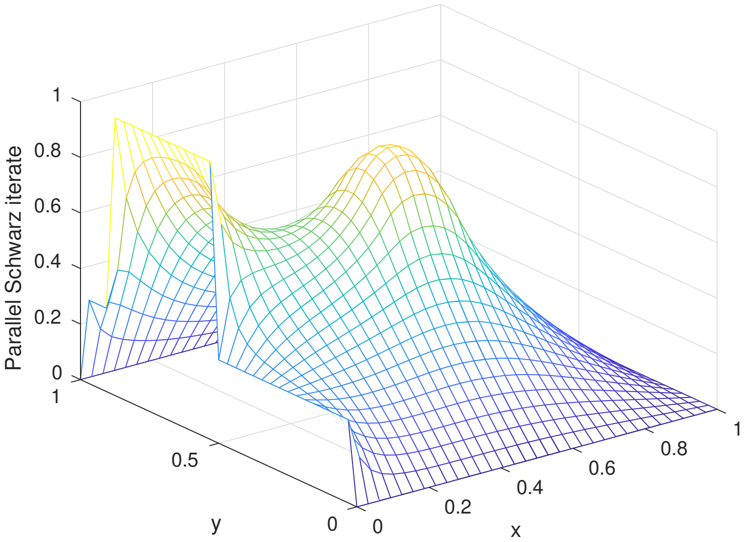
Parallel Schwarz iteration 8



Parallel Schwarz iteration 9



Parallel Schwarz iteration 10



Convergence Results for Schwarz

Schwarz (1869): Maximum principle

Lions (1988): Variational analysis based on projections

Chan and Resasco (1988): Fourier analysis

Lions (1989): Maximum principle

Bramble, Pasciak, Wang, Xu (1991): Continuous and discrete orthogonal projection estimates

All these results give **contraction estimates** for Schwarz methods at the continuous level!

Convergence Results for Schwarz

Schwarz (1869): Maximum principle

Lions (1988): Variational analysis based on projections

Chan and Resasco (1988): Fourier analysis

Lions (1989): Maximum principle

Bramble, Pasciak, Wang, Xu (1991): Continuous and discrete orthogonal projection estimates

All these results give **contraction estimates** for Schwarz methods at the continuous level!

Dryja and Widlund (1989): Discrete two level condition number estimate for the **Additive Schwarz Method**

$$\kappa(M_{AS}^c A) \leq C\left(1+\frac{H}{\delta}\right),$$

where H is the coarse mesh size and $\delta \sim h$ the overlap.

For Additive Schwarz there is a **condition number estimate**, not a **contraction estimate**!

Convergence Results for Schwarz

Schwarz (1869): Maximum principle

Lions (1988): Variational analysis based on projections

Chan and Resasco (1988): Fourier analysis

Lions (1989): Maximum principle

Bramble, Pasciak, Wang, Xu (1991): Continuous and discrete orthogonal projection estimates

All these results give **contraction estimates** for Schwarz methods at the continuous level!

Dryja and Widlund (1989): Discrete two level condition number estimate for the **Additive Schwarz Method**

$$\kappa(M_{AS}^c A) \leq C\left(1 + \frac{H}{\delta}\right),$$

where H is the coarse mesh size and $\delta \sim h$ the overlap.

For Additive Schwarz there is a **condition number estimate**, not a **contraction estimate**!

Result holds at the continuous level (G, Halpern, Santugini 2014)

Neumann-Neumann and FETI

Bourgat, Glowinski, LeTallec, Vidrascu (1989): For $\Delta u = f$ in Ω, two non-overlapping subdomains Ω_1 and Ω_2, interface Γ: starting with an interface guess h^0, compute for $n = 1, 2, \ldots$

$$\begin{array}{rcl} \Delta u_i^n & = & f \quad \text{in } \Omega_i \\ u_i^n(\Gamma) & = & h^{n-1} \end{array}$$

followed by the correction computation

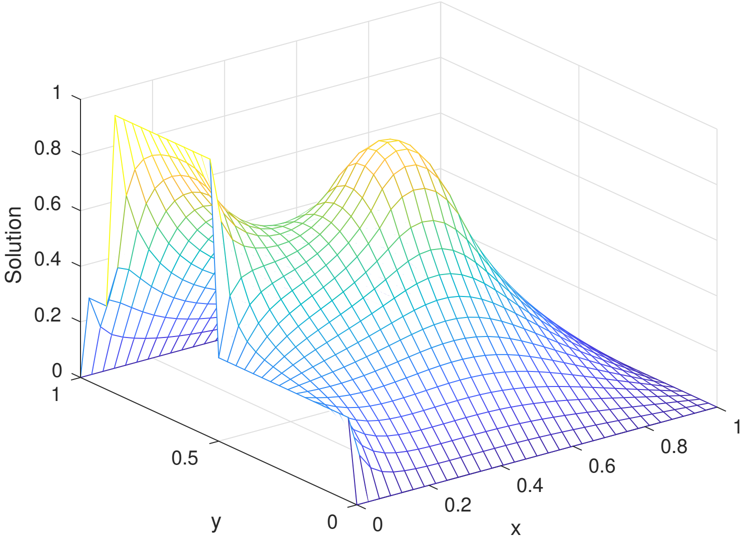
$$\begin{array}{rcl} \Delta \psi_i^n & = & 0 \quad \text{in } \Omega_i \\ \partial_{n_i} \psi_i^n(\Gamma) & = & \partial_{n_1} u_1^n(\Gamma) + \partial_{n_2} u_2^n(\Gamma) \end{array}$$

and then obtain a new interface trace

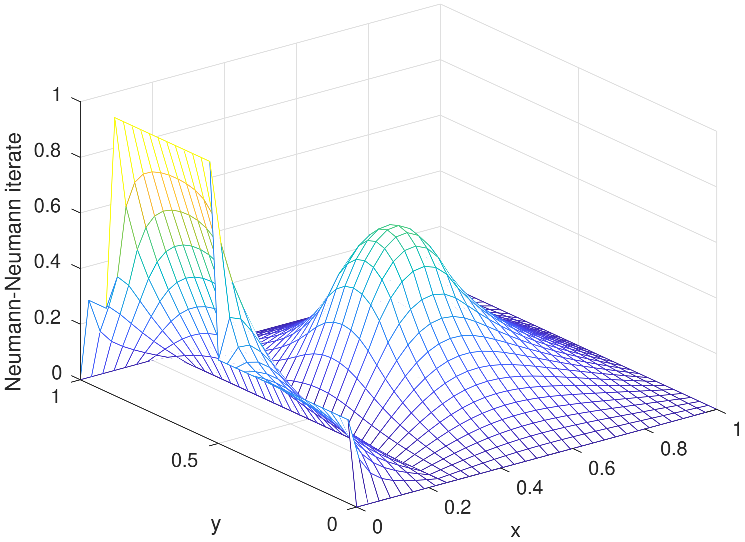
$$h^n = h^{n-1} - \theta(\psi_1(\Gamma) + \psi_2(\Gamma)).$$

Farhat and Roux (1991): Finite Element Tearing and Interconnect (FETI)

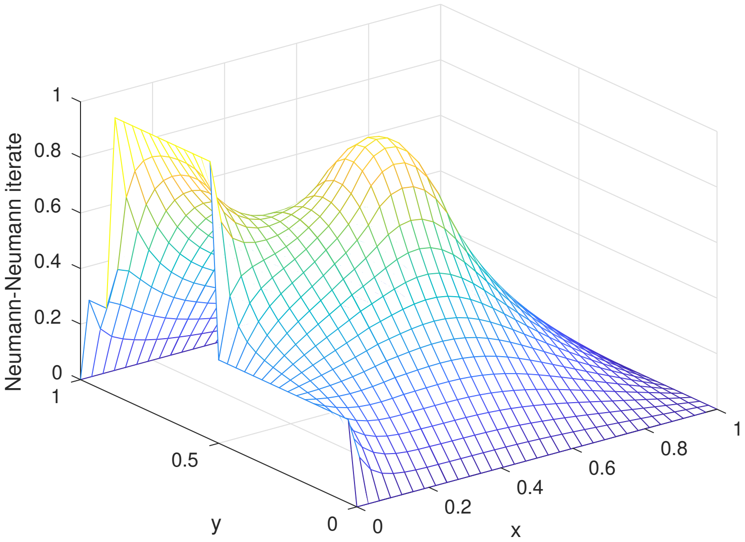
Example: temperature in a room



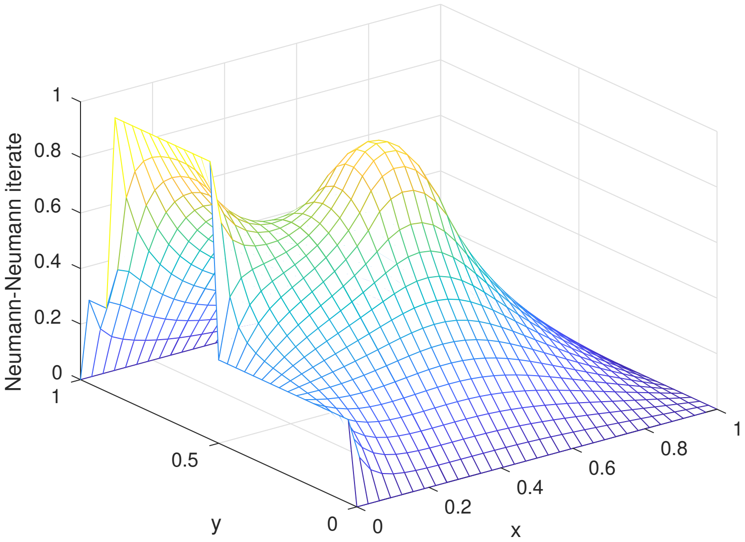
Neumann-Neumann iteration 1



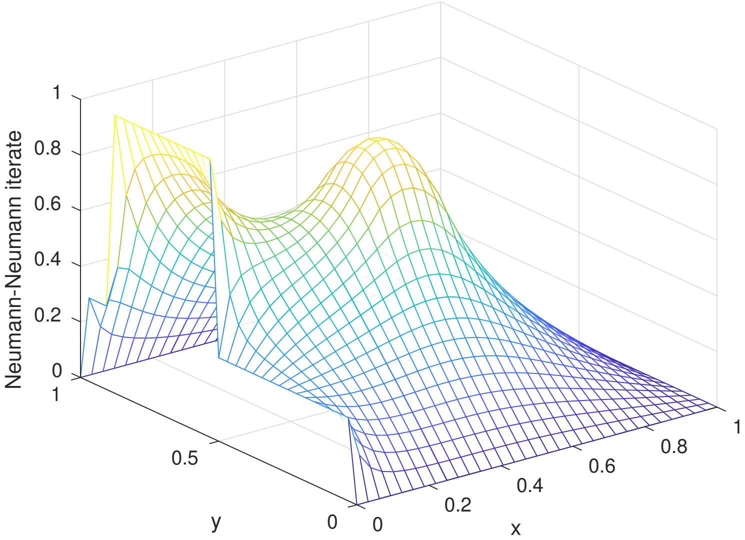
Neumann-Neumann iteration 2



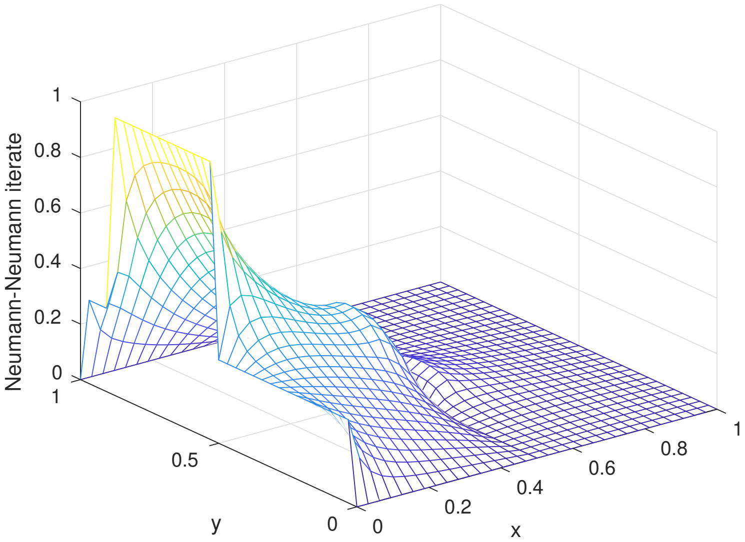
Neumann-Neumann iteration 3



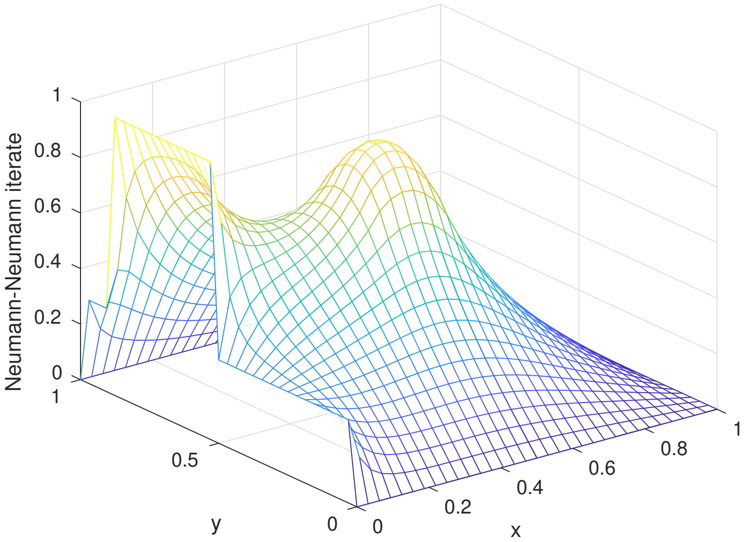
Neumann-Neumann iteration 4



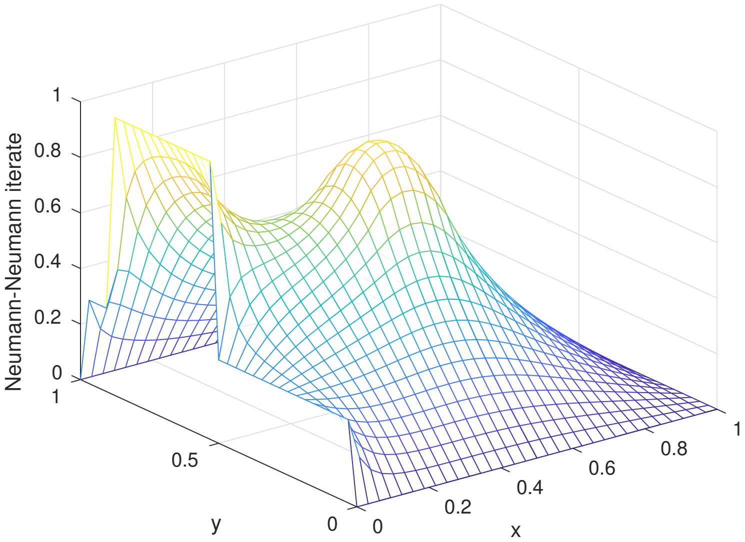
Neumann-Neumann iteration 1, symmetric Ω_i



Neumann-Neumann iteration 2, symmetric Ω_i



Neumann-Neumann iteration 3, symmetric Ω_i



Results for Neumann-Neumann

De Roeck and LeTallec (1991): continuous analysis without cross points, discrete with cross points

Mandel and Brezina (1993): Balancing Neumann-Neumann, **condition number estimate**

$$\kappa(M,S) \leq C\left(1+\log^2\frac{H}{h}\right),$$

where h is the fine and H is the coarse mesh size.

Result for FETI

Mandel and Tezaur (1996): condition number estimate

$$\kappa(M,S) \leq C\left(1+\log\frac{H}{h}\right)^{\gamma},$$

where h is the fine and H is the coarse mesh size, and $\gamma = 3$, or $\gamma = 2$ in special cases.

$\Longrightarrow$ **Klawonn et al. (2001, 2002, 2006, 2007,. . .)**

Two Differences so Far

1. MG is defined at the discrete level,
   DDs are defined at the continuous level.

Two Differences so Far

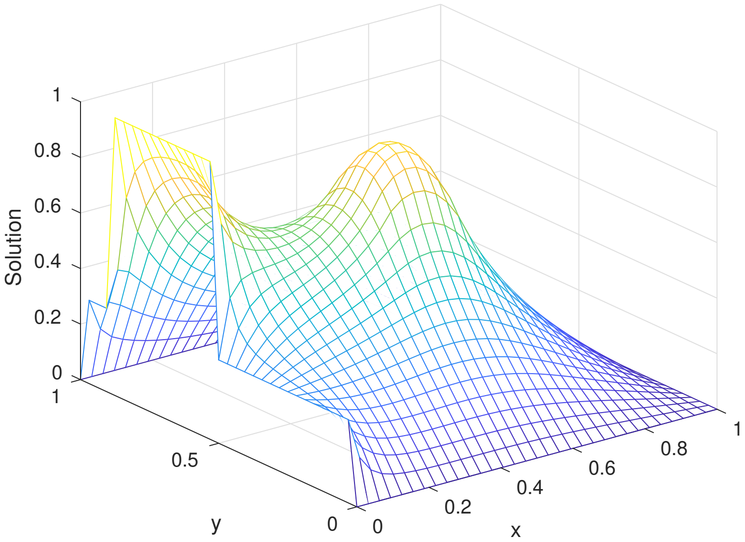
1. MG is defined at the discrete level,
   DDs are defined at the continuous level.

2. **Contraction estimates** for MG,
   Contraction estimates for some DD.

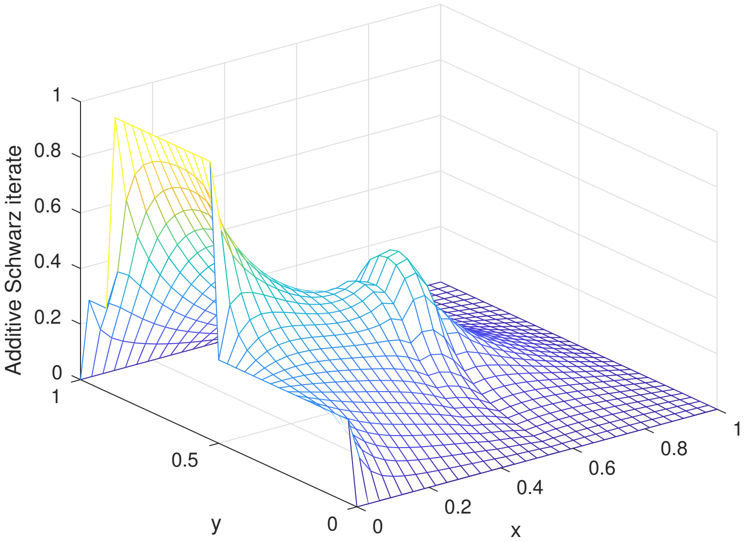
   Condition number estimates for Additive Schwarz, Neumann-Neumann with cross points, and FETI with cross points.

Why ?

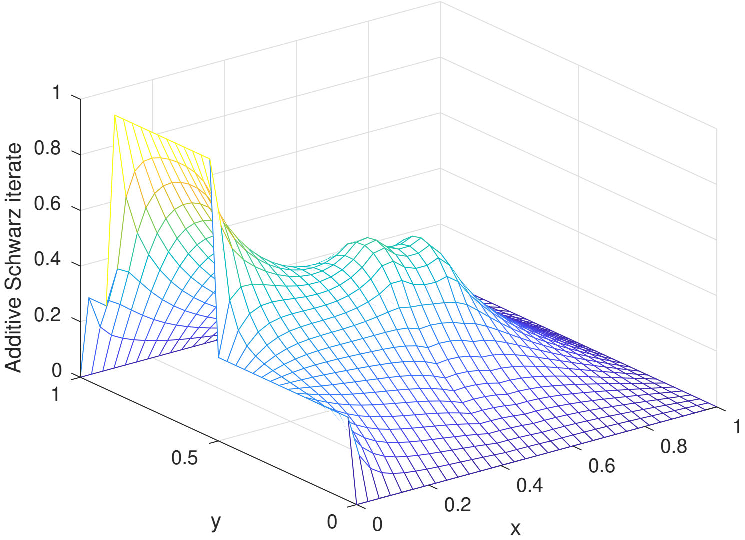
AS Example: temperature in a room



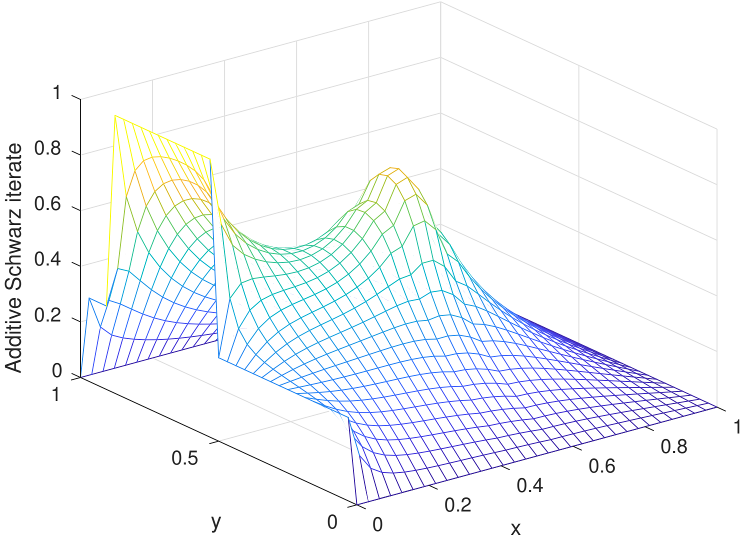
Additive Schwarz iteration 1



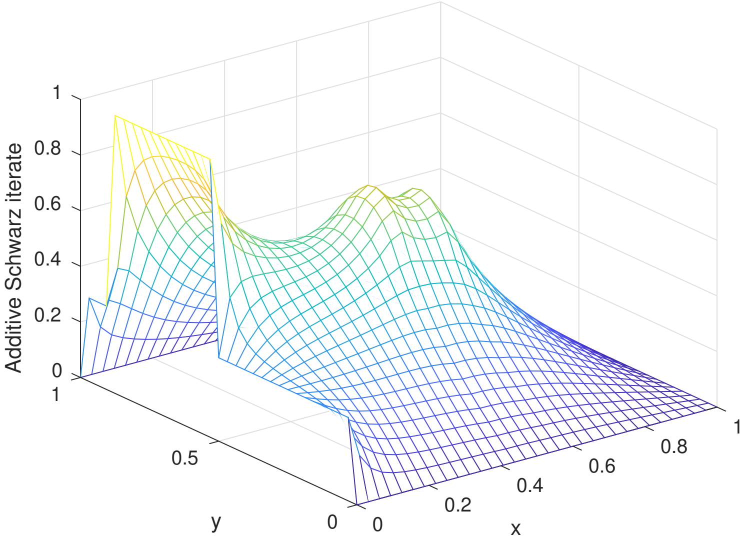
Additive Schwarz iteration 2



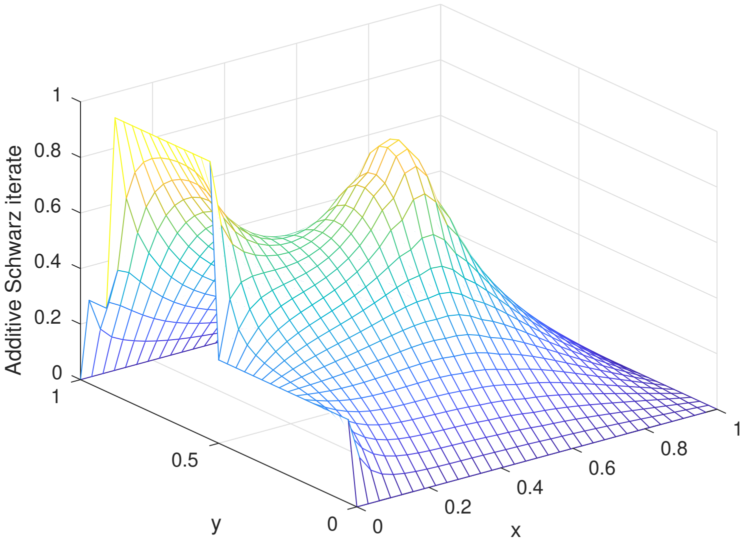
Additive Schwarz iteration 3



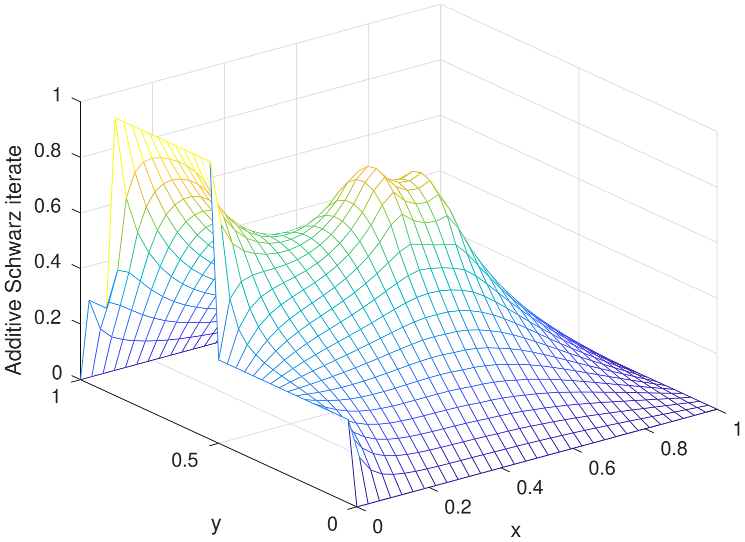
Additive Schwarz iteration 4



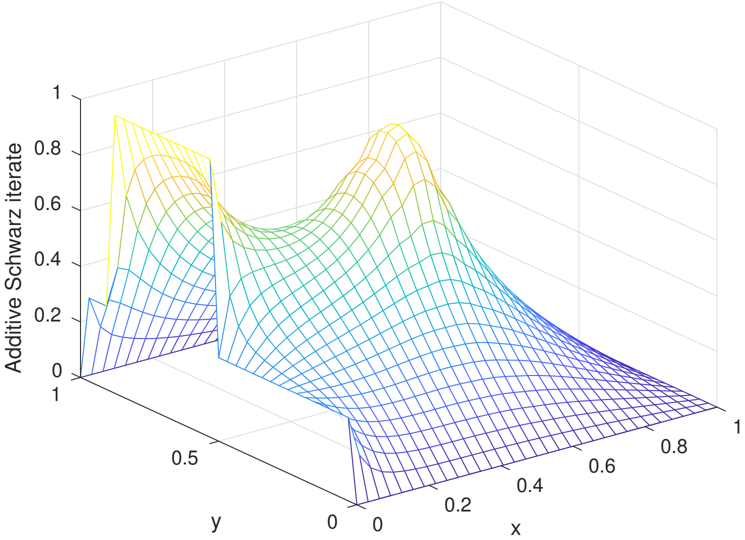
Additive Schwarz iteration 5



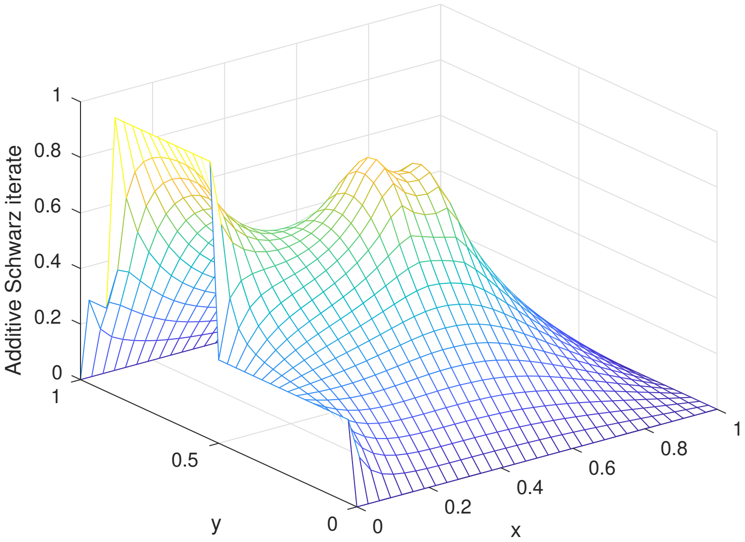
Additive Schwarz iteration 6



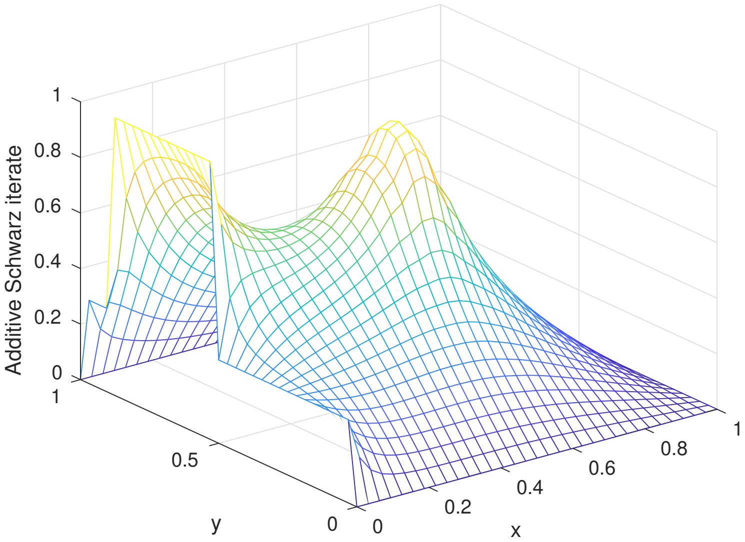
Additive Schwarz iteration 7



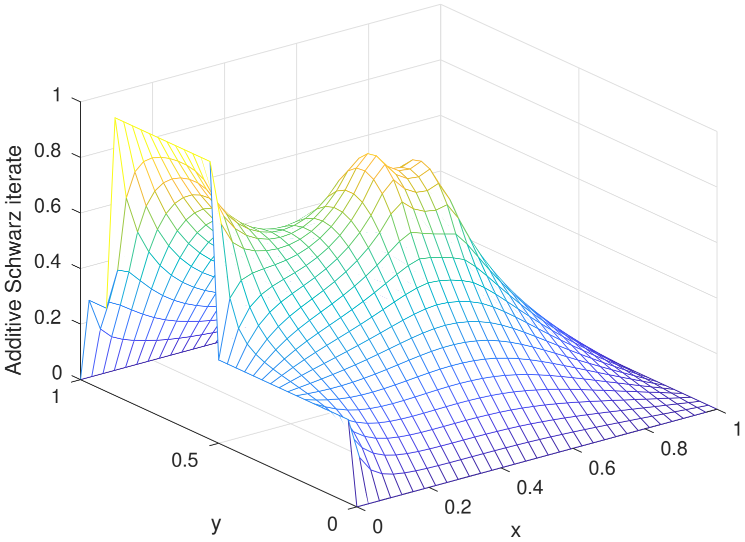
Additive Schwarz iteration 8



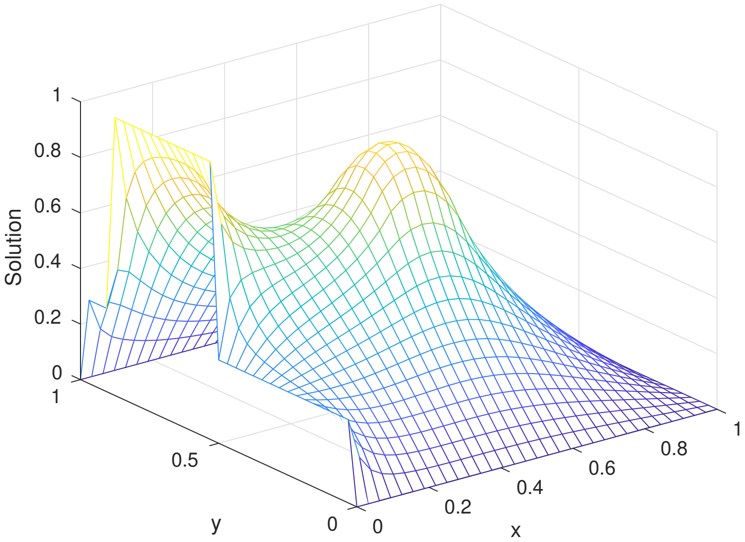
Additive Schwarz iteration 9



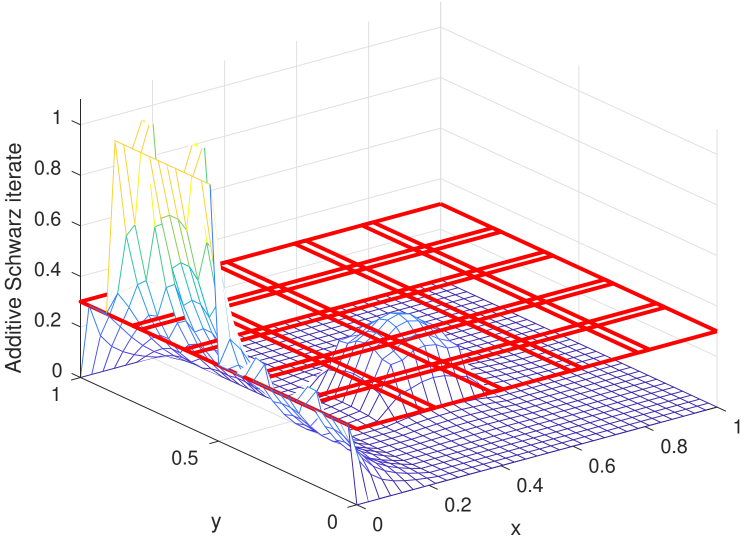
Additive Schwarz iteration 10



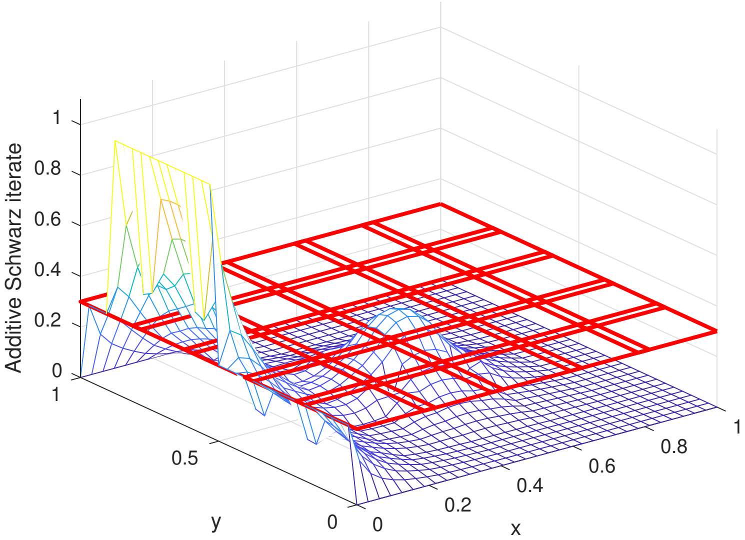
And with 25 subdomains



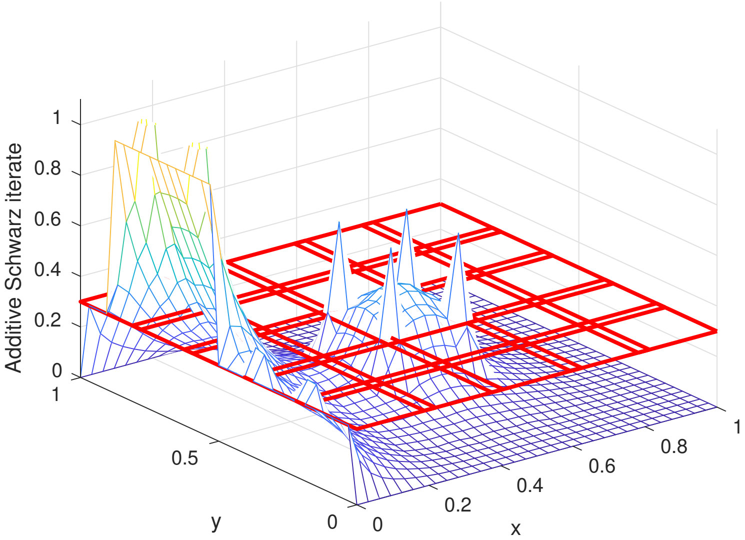
Additive Schwarz iteration 1



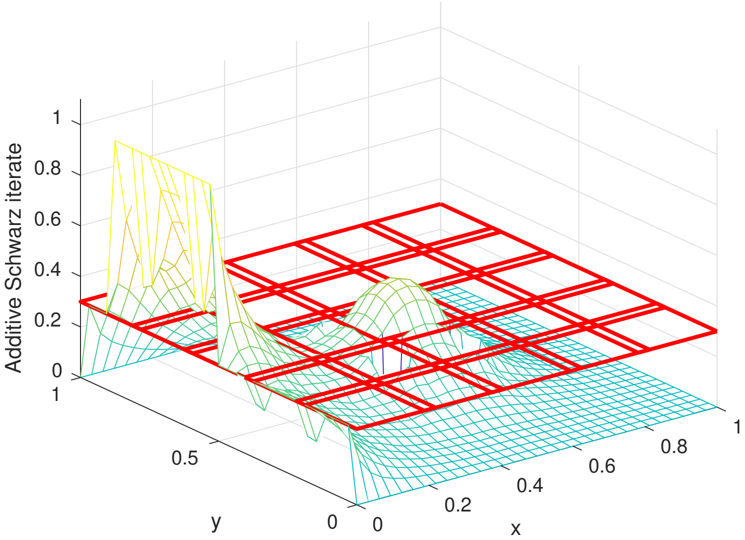
Additive Schwarz iteration 2



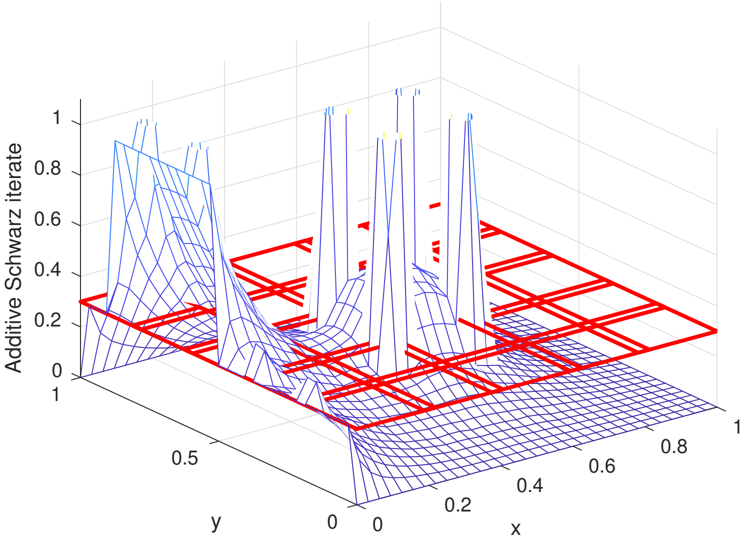
Additive Schwarz iteration 3



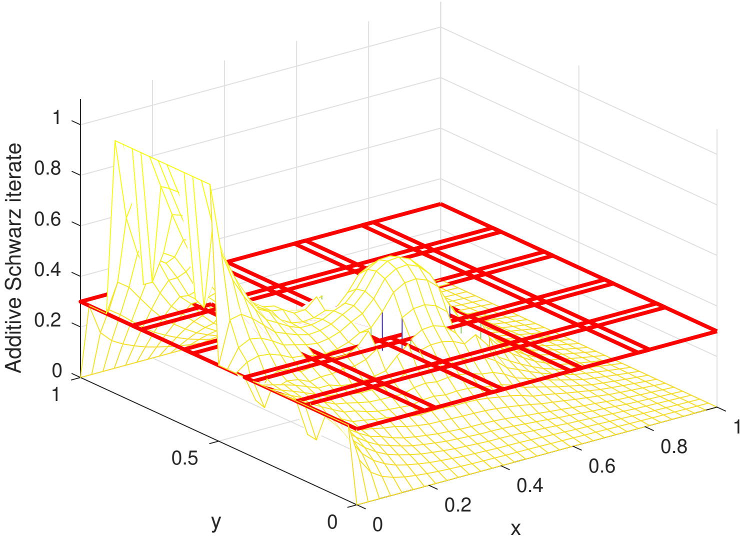
Additive Schwarz iteration 4



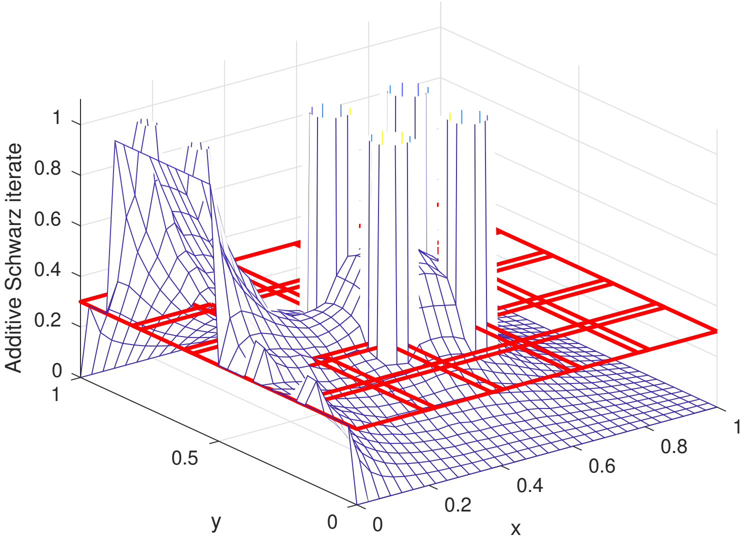
Additive Schwarz iteration 5



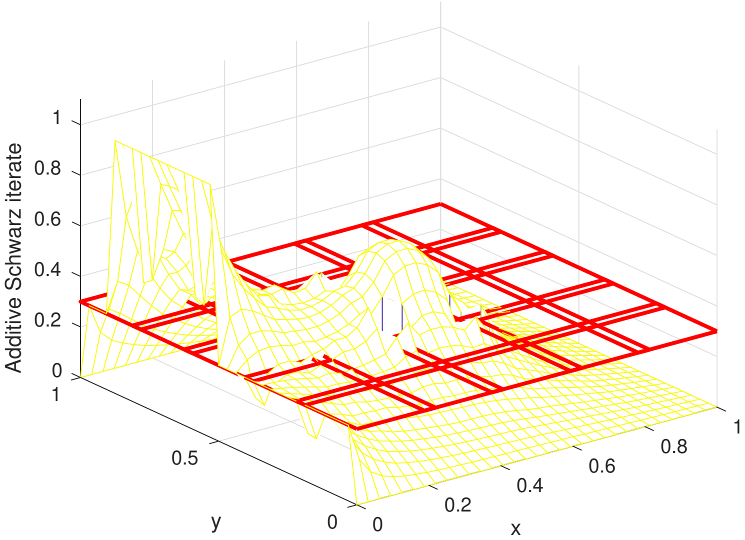
Additive Schwarz iteration 6



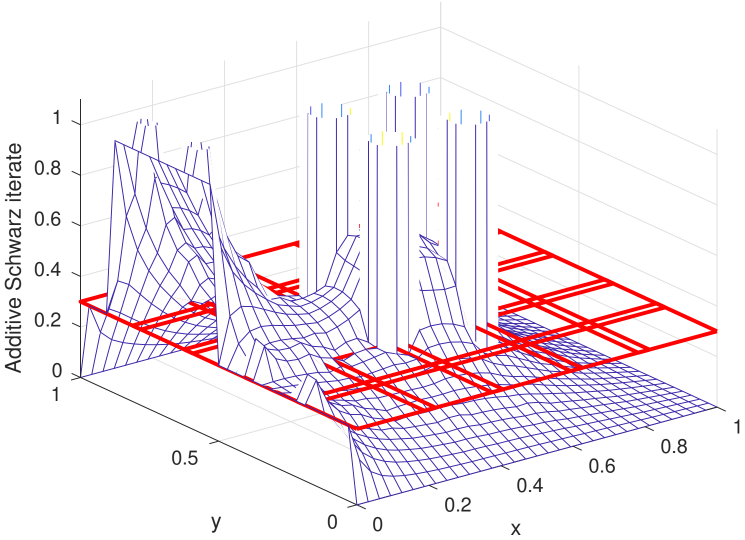
Additive Schwarz iteration 7



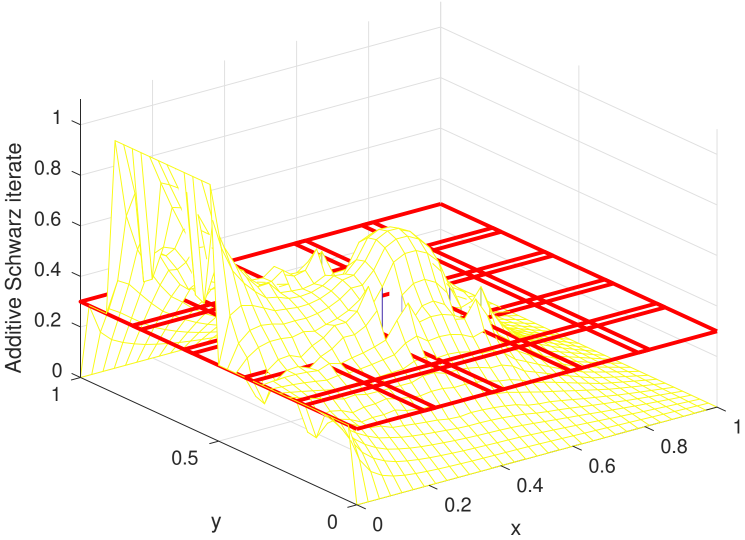
Additive Schwarz iteration 8



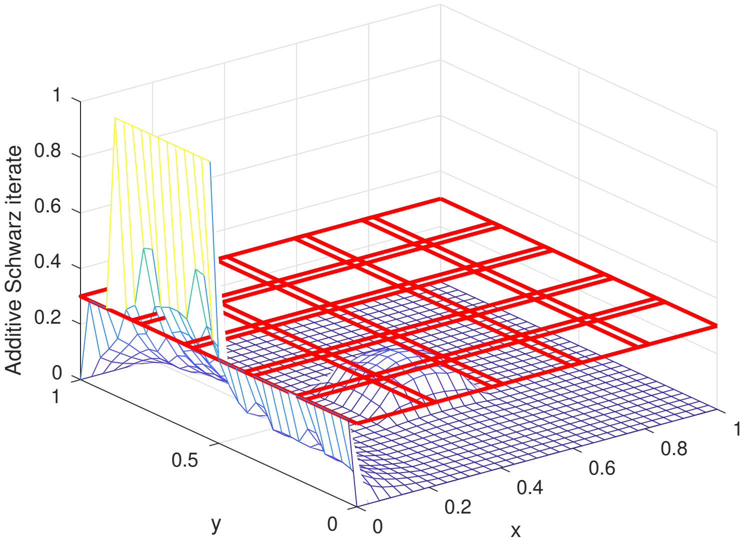
Additive Schwarz iteration 9



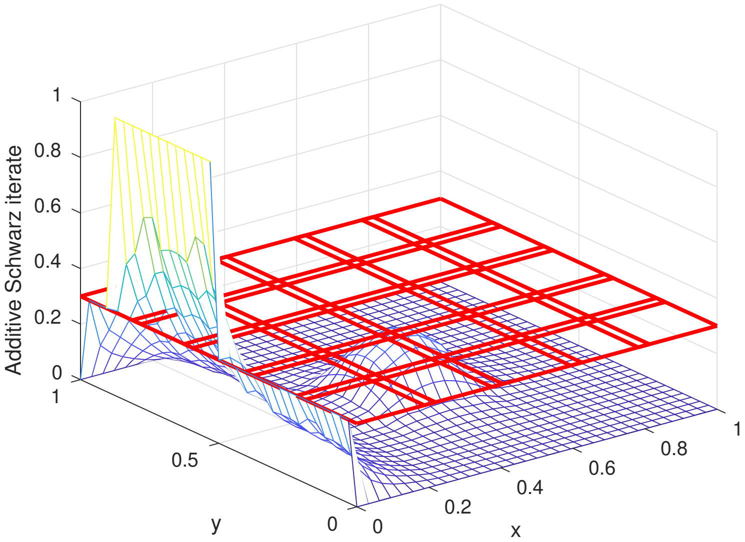
Additive Schwarz iteration 10



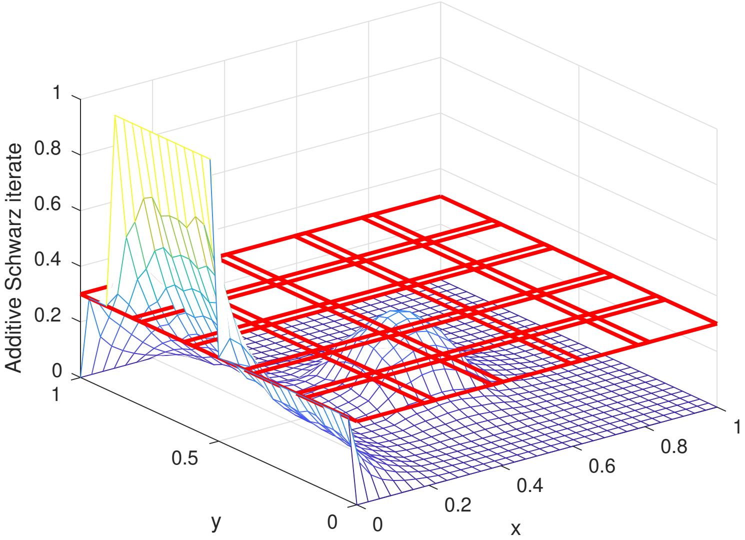
Damped Additive Schwarz iteration 1



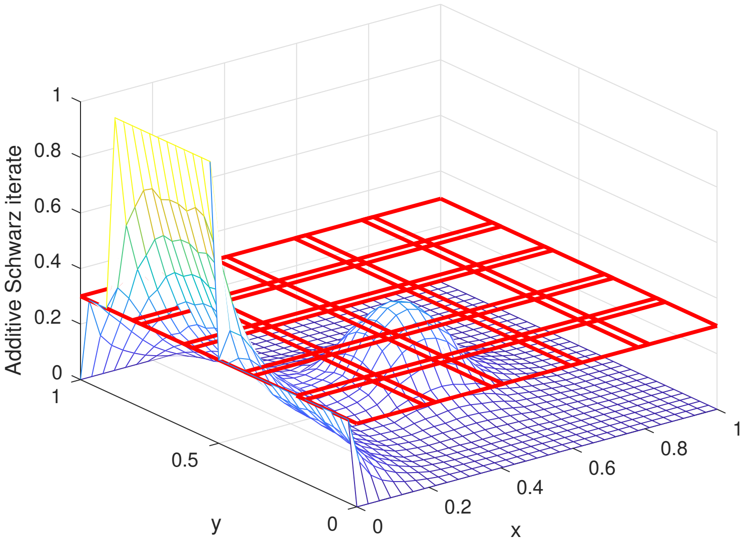
Damped Additive Schwarz iteration 2



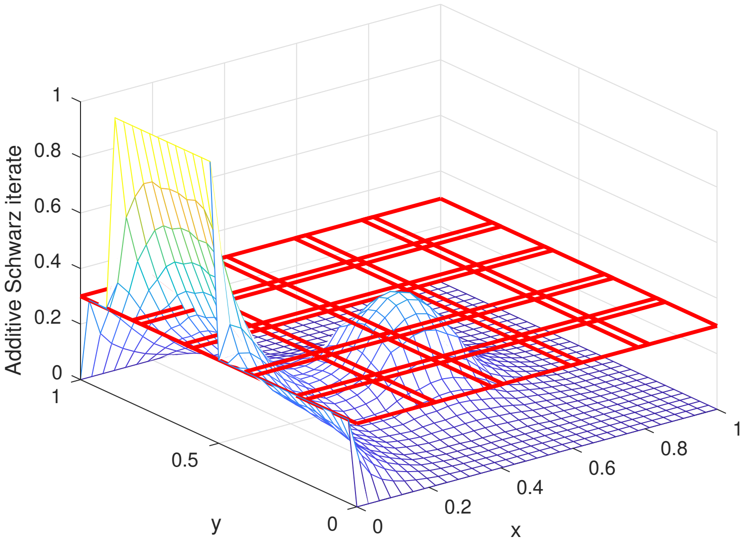
Damped Additive Schwarz iteration 3



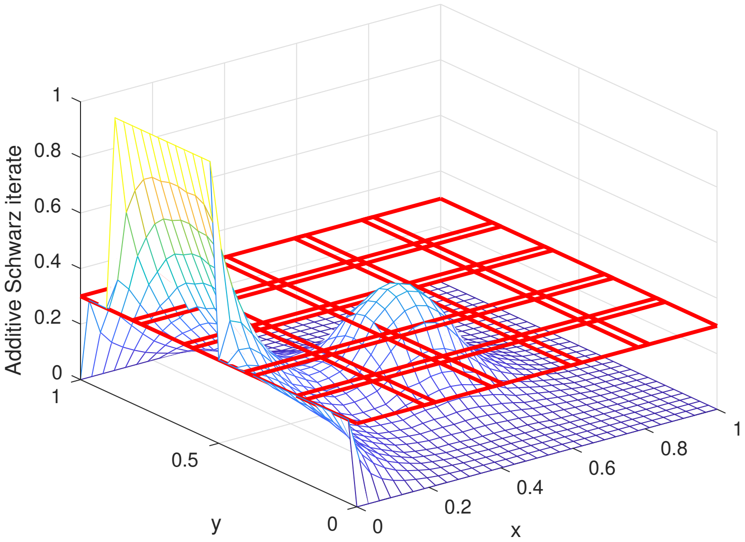
Damped Additive Schwarz iteration 4



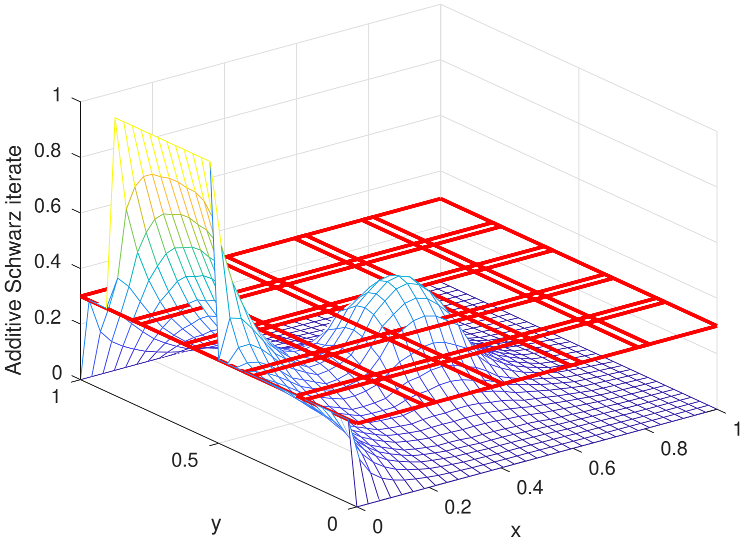
Damped Additive Schwarz iteration 5



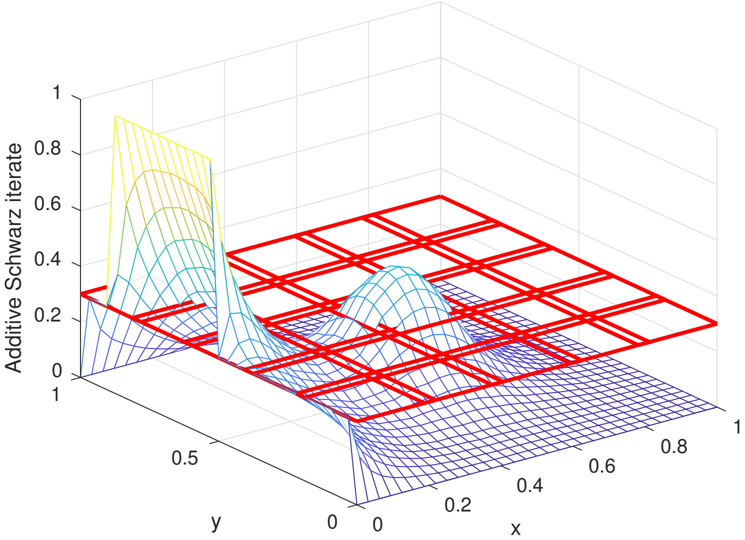
Damped Additive Schwarz iteration 6



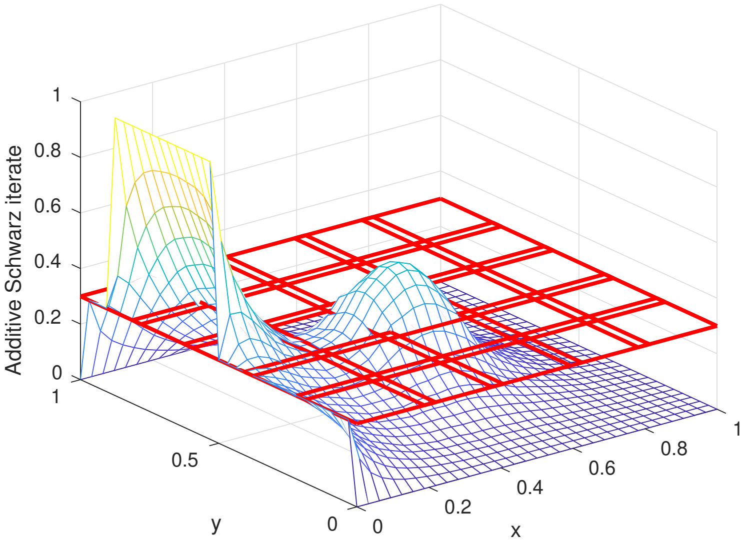
Damped Additive Schwarz iteration 7



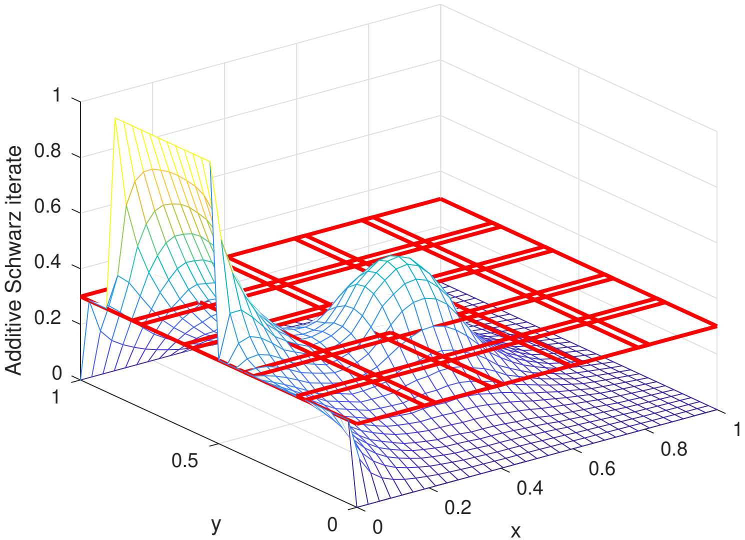
Damped Additive Schwarz iteration 8



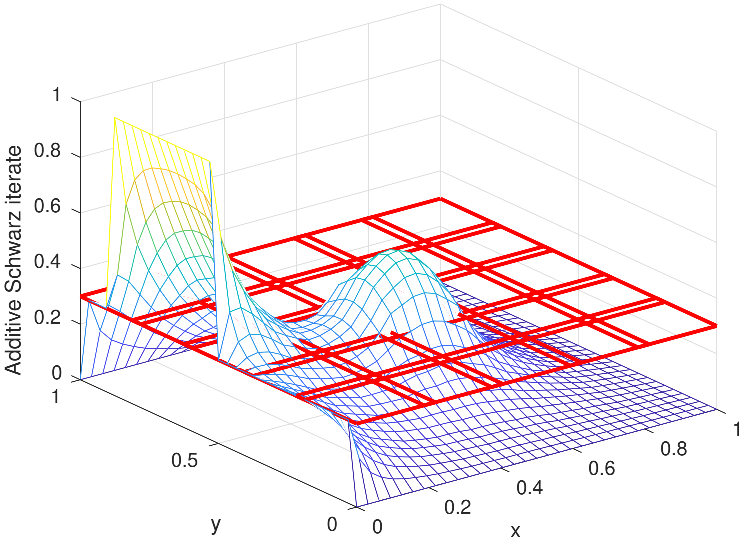
Damped Additive Schwarz iteration 9



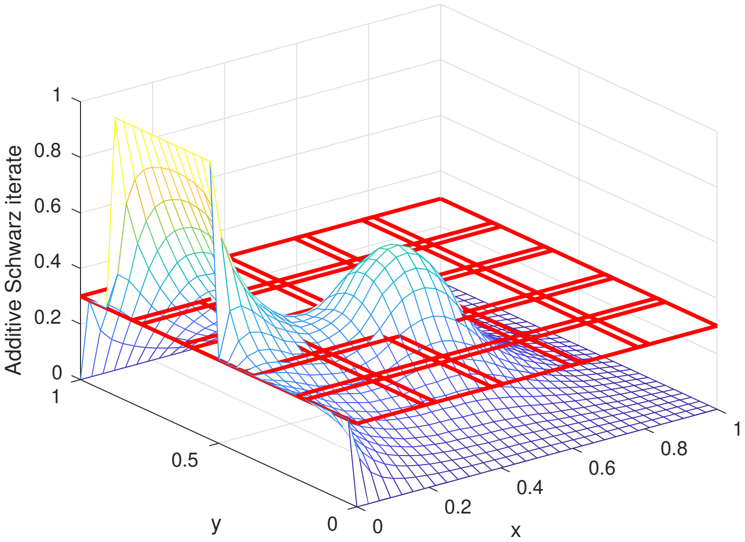
Damped Additive Schwarz iteration 10



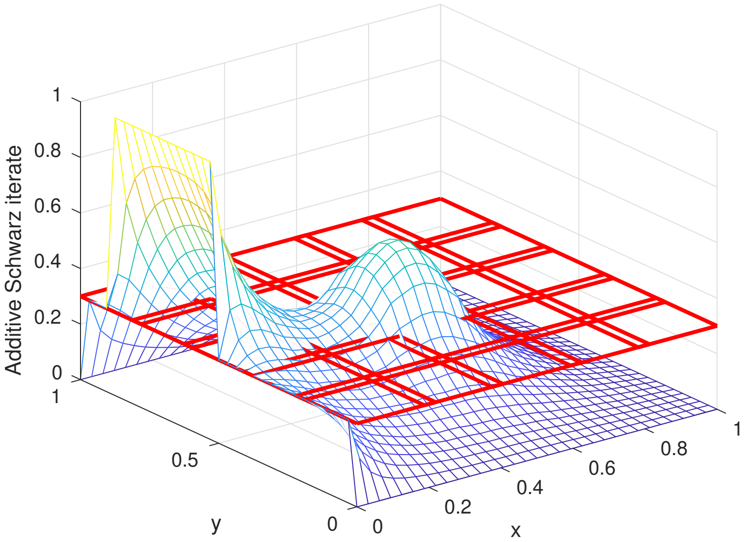
Damped Additive Schwarz iteration 11



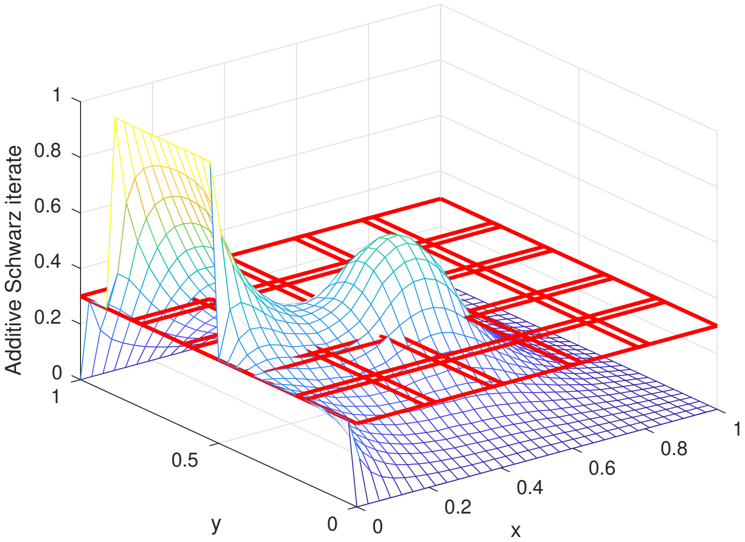
Damped Additive Schwarz iteration 12



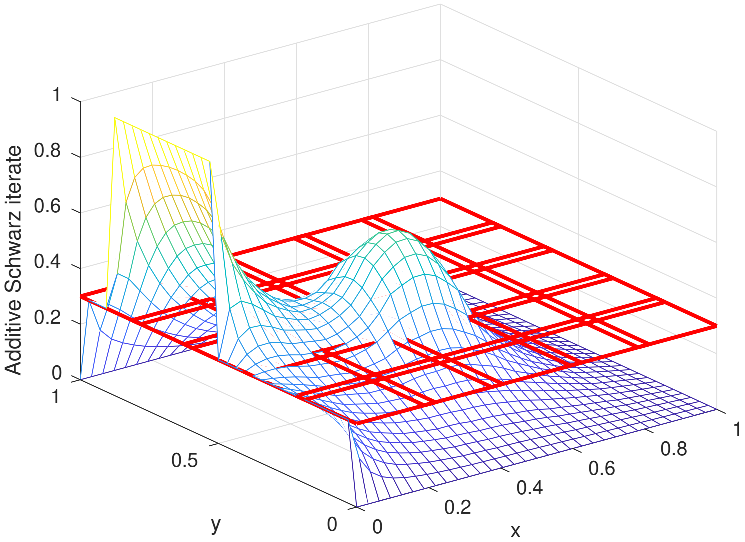
Damped Additive Schwarz iteration 13



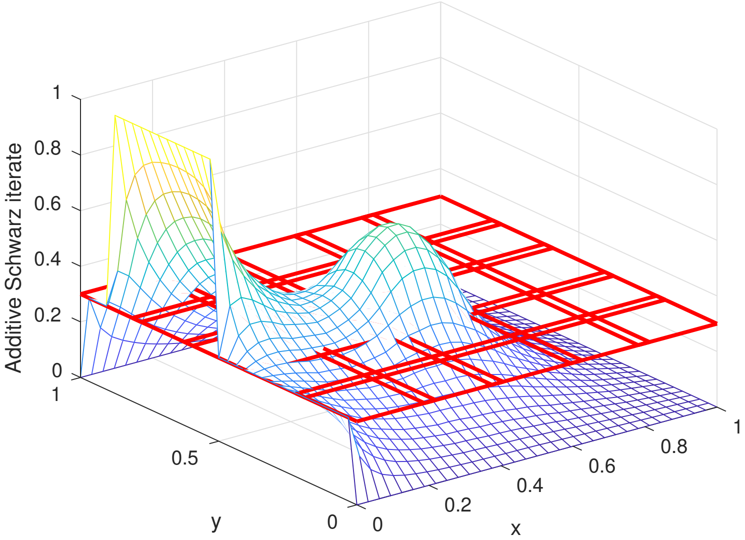
Damped Additive Schwarz iteration 14



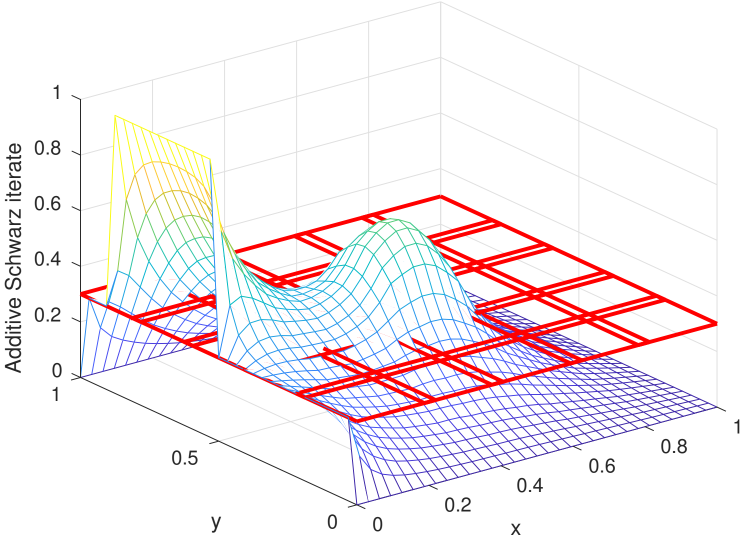
Damped Additive Schwarz iteration 15



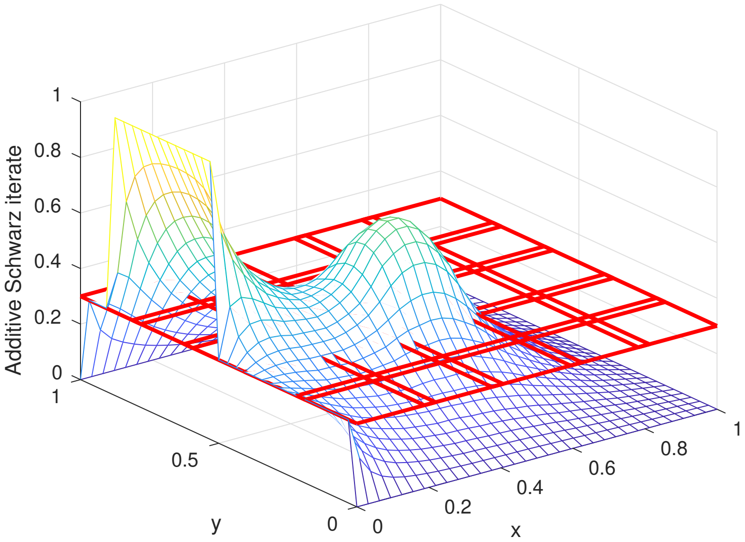
Damped Additive Schwarz iteration 16



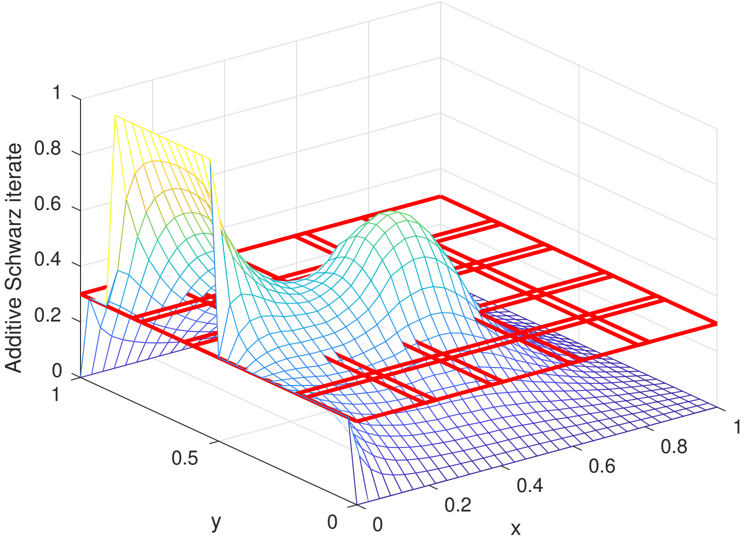
Damped Additive Schwarz iteration 17



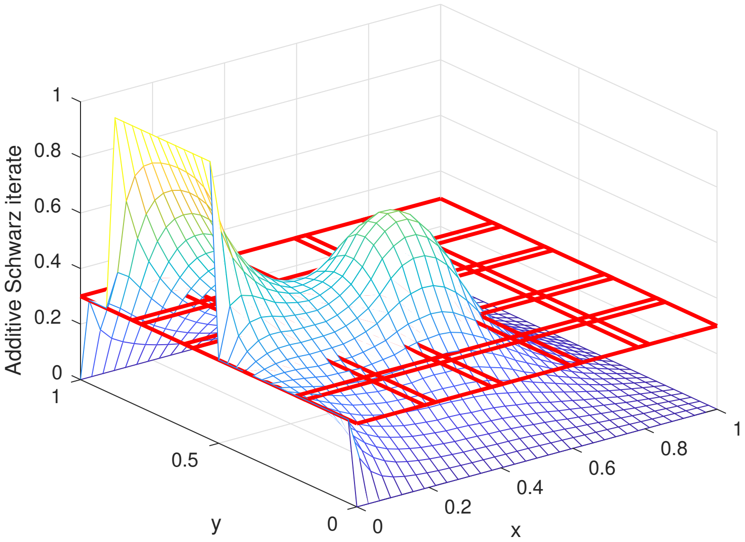
Damped Additive Schwarz iteration 18



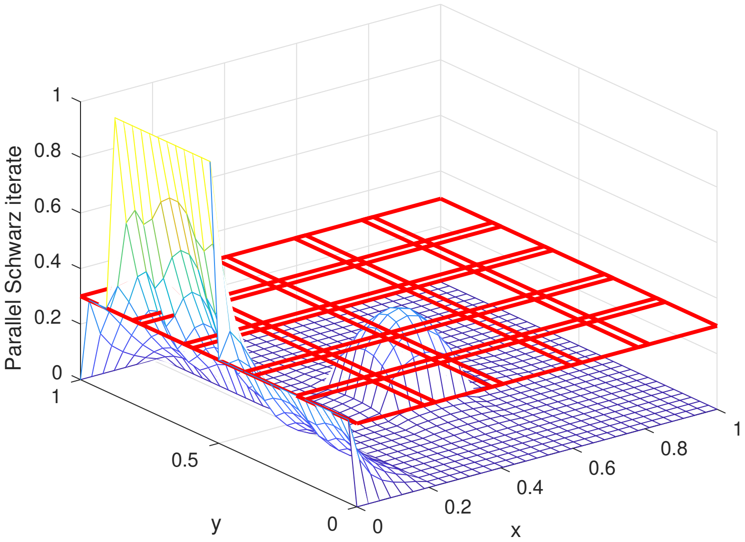
Damped Additive Schwarz iteration 19



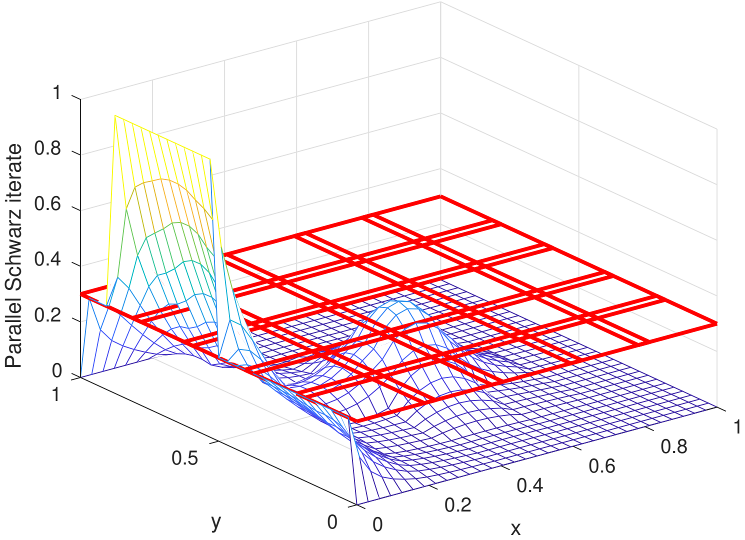
Damped Additive Schwarz iteration 20



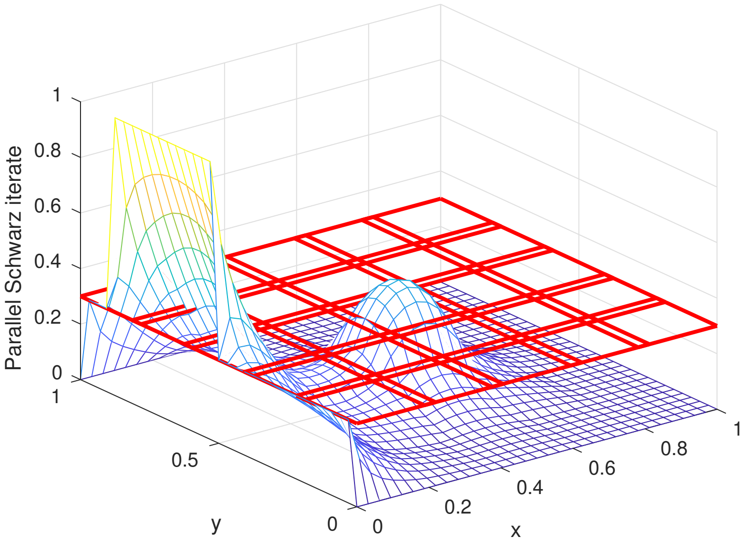
Parallel Schwarz iteration 1



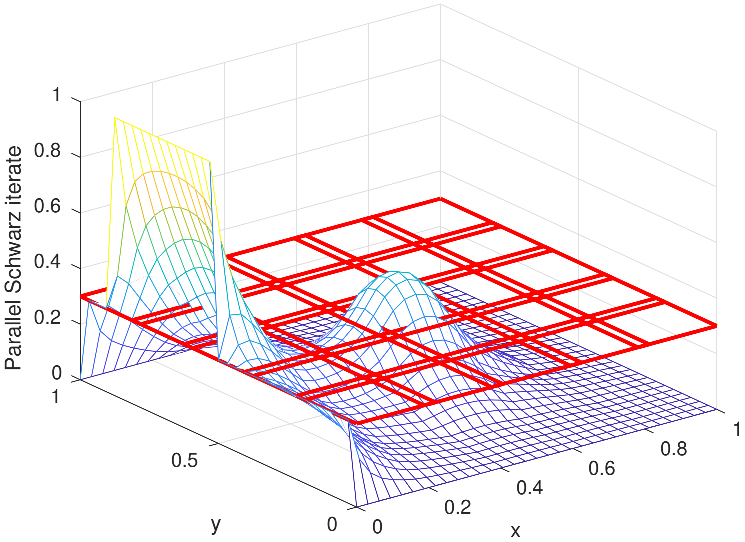
Parallel Schwarz iteration 2



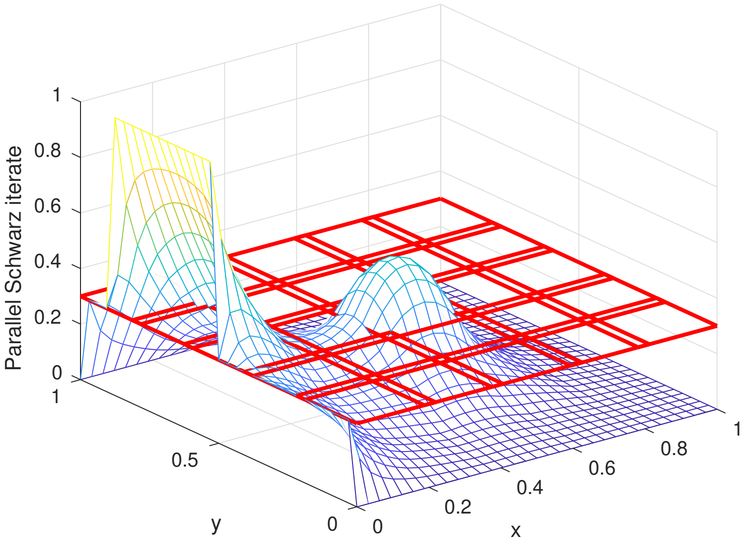
Parallel Schwarz iteration 3



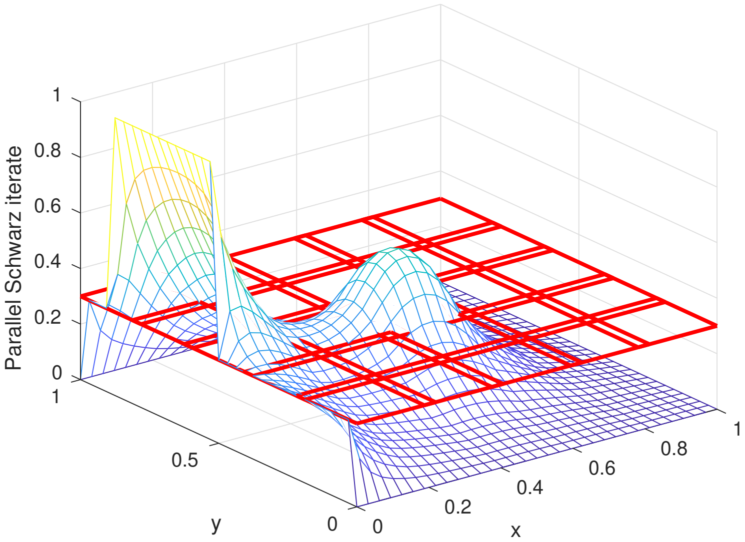
Parallel Schwarz iteration 4



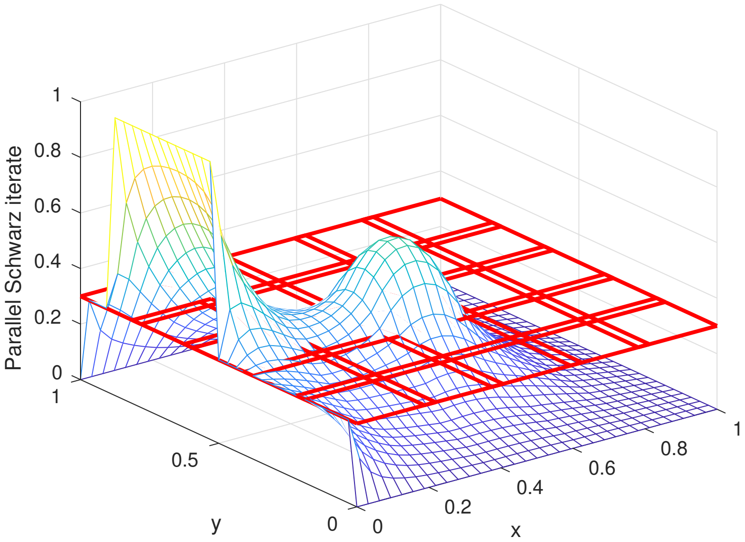
Parallel Schwarz iteration 5



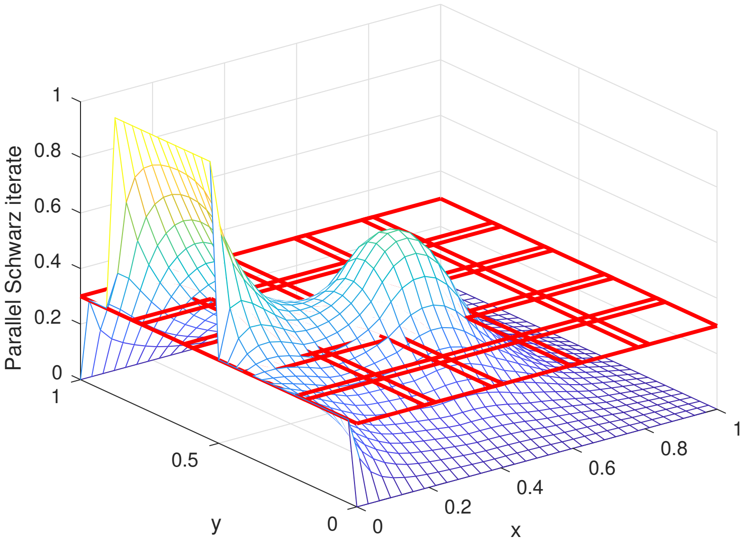
Parallel Schwarz iteration 6



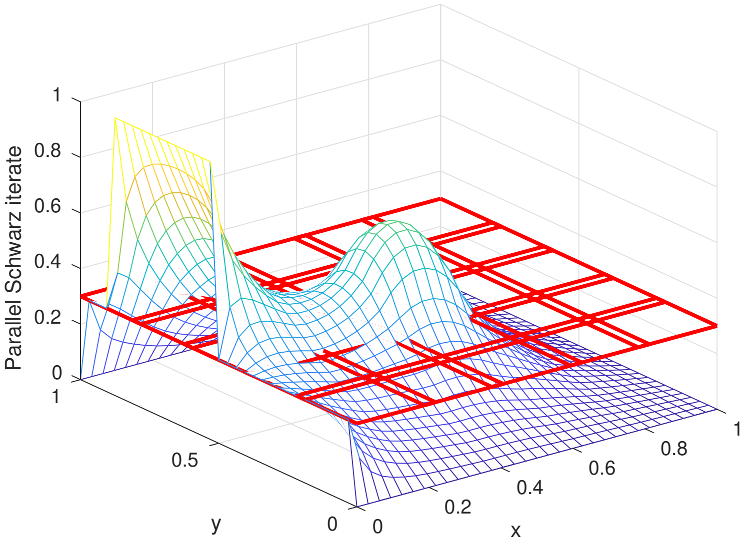
Parallel Schwarz iteration 7



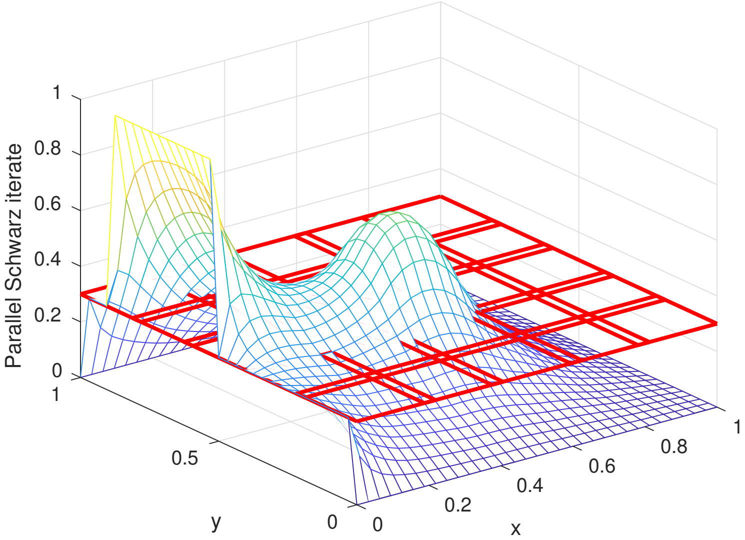
Parallel Schwarz iteration 8



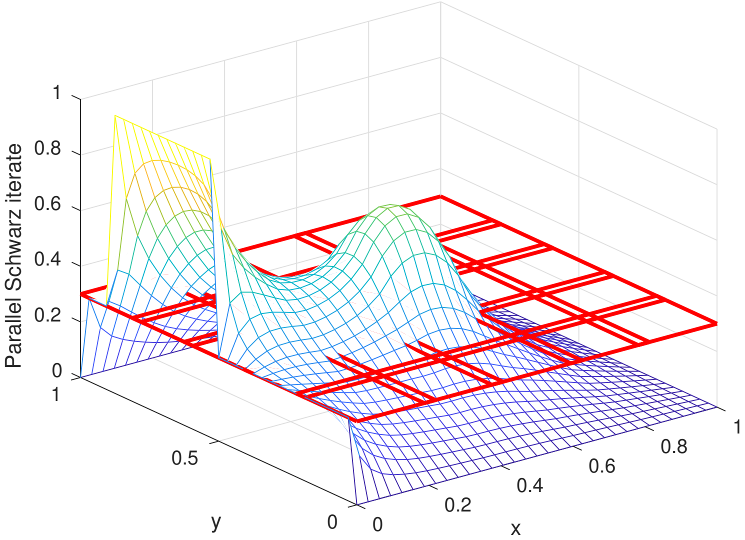
Parallel Schwarz iteration 9



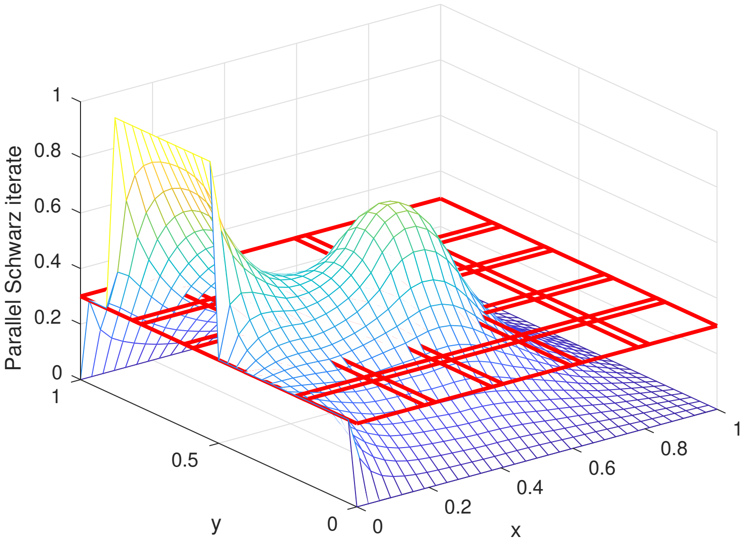
Parallel Schwarz iteration 10



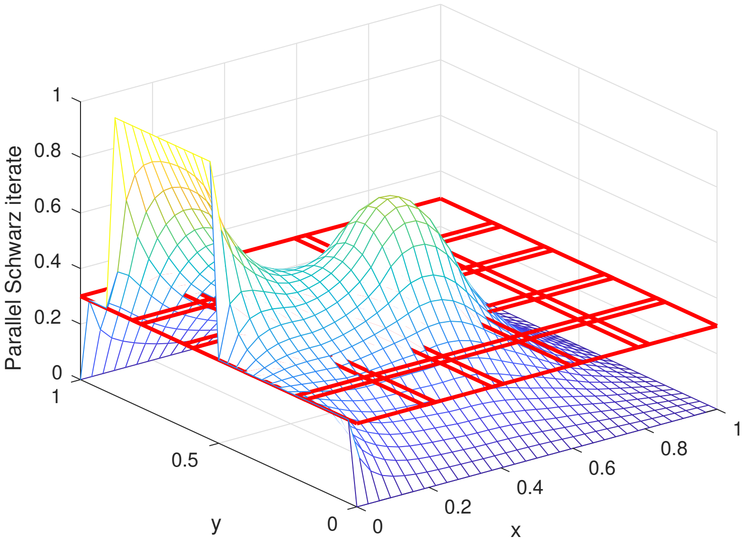
Parallel Schwarz iteration 11



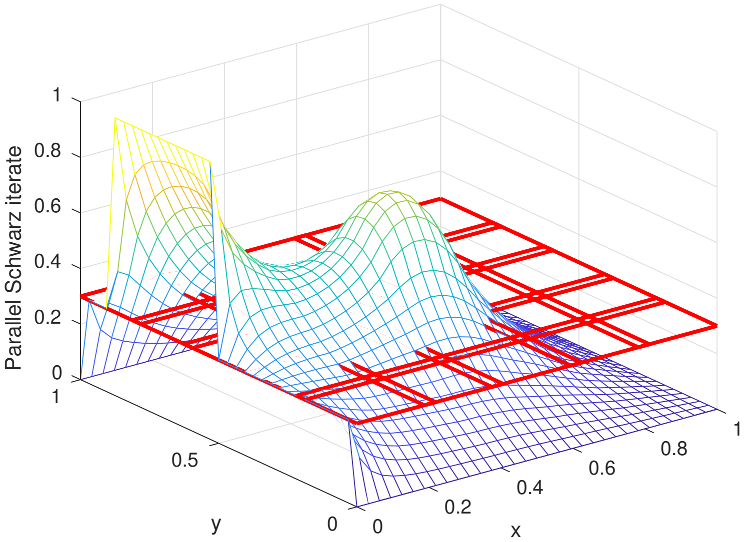
Parallel Schwarz iteration 12



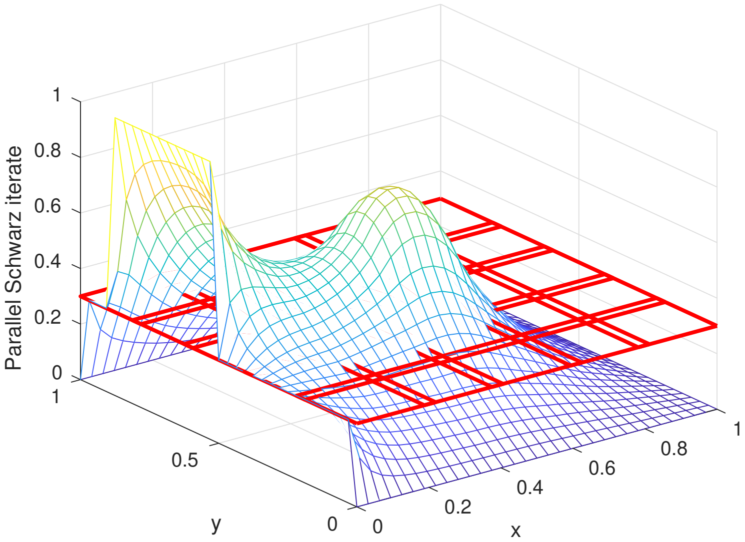
Parallel Schwarz iteration 13



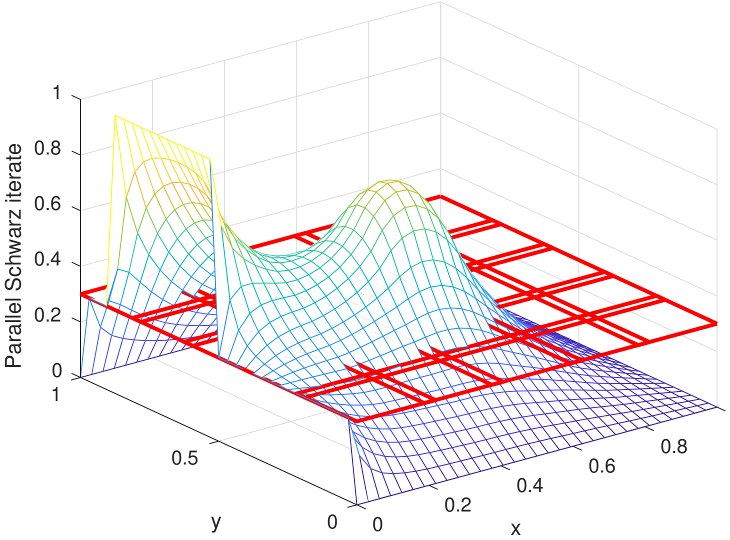
Parallel Schwarz iteration 14



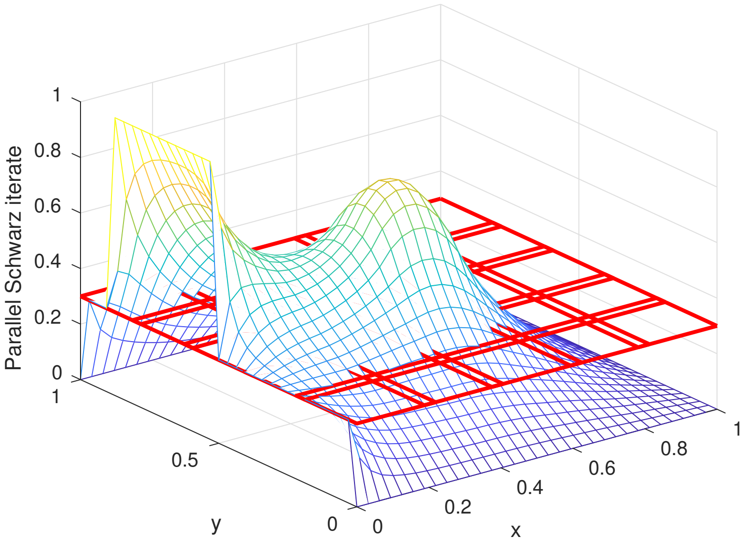
Parallel Schwarz iteration 15



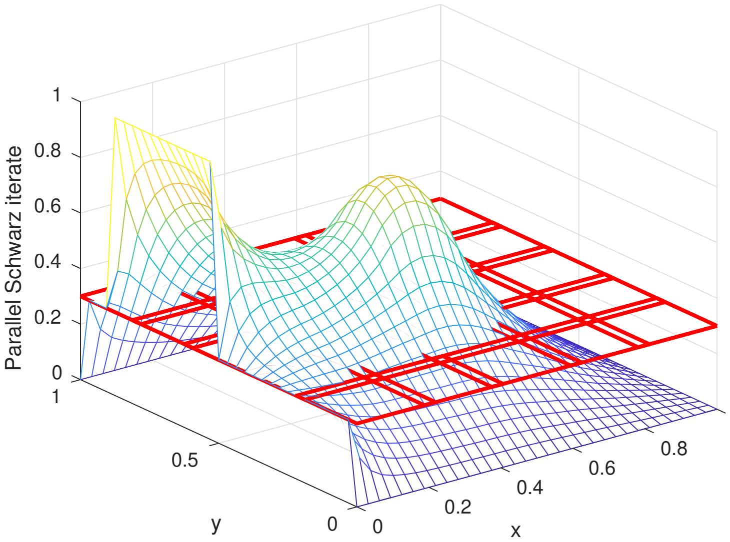
Parallel Schwarz iteration 16



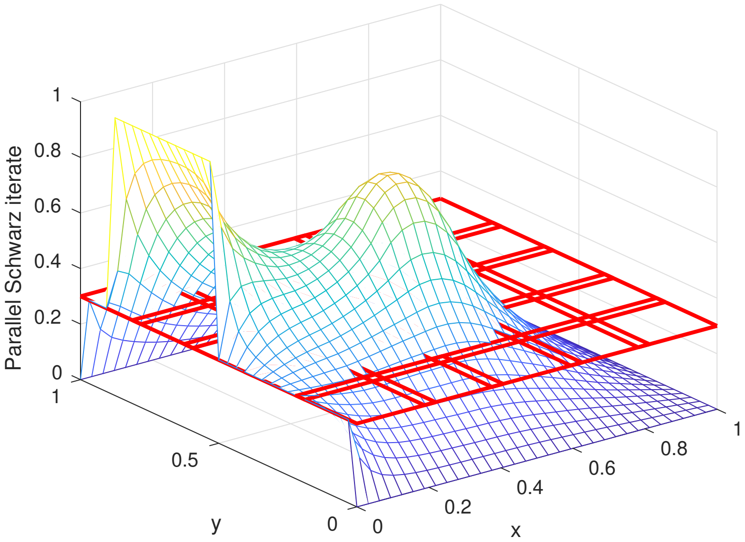
Parallel Schwarz iteration 17



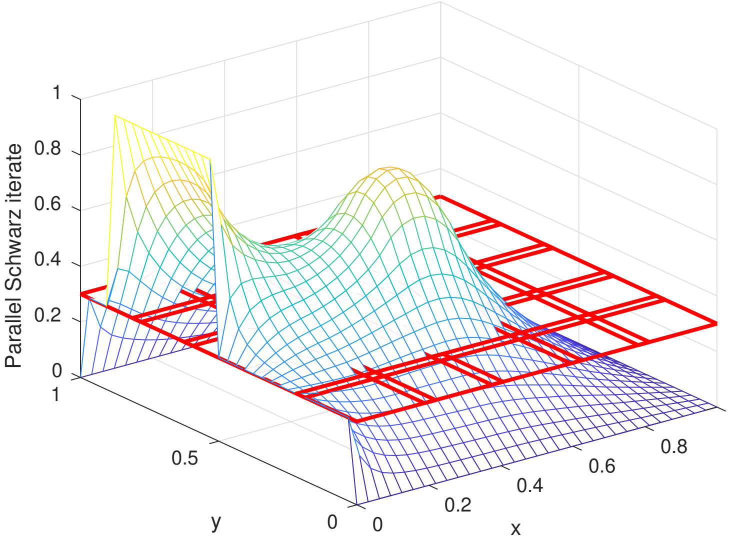
Parallel Schwarz iteration 18



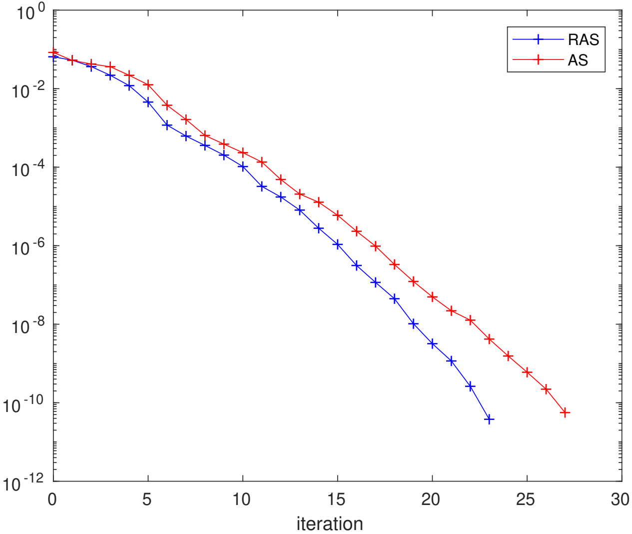
Parallel Schwarz iteration 19



Parallel Schwarz iteration 20

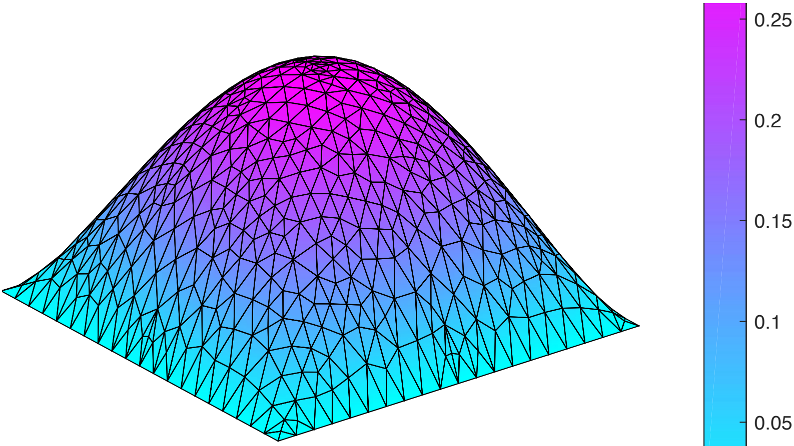


Comparison with Krylov Acceleration



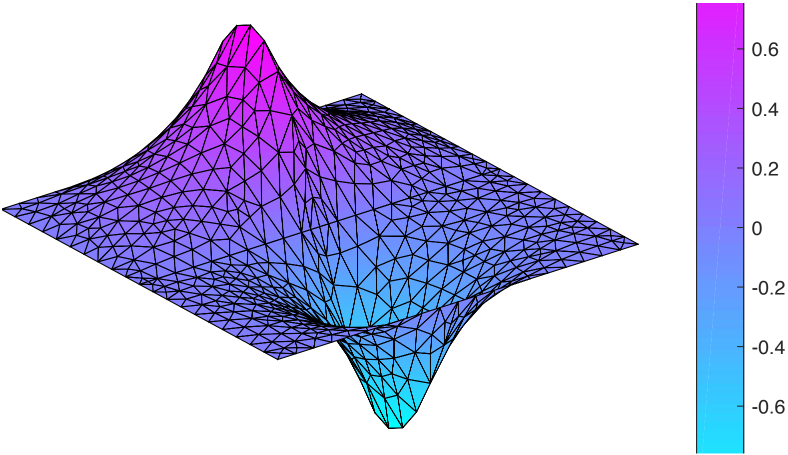
$\Longrightarrow$ Additive Schwarz is a preconditioner, parallel Schwarz is a standalone solver and a preconditioner (like multigrid)! (see Efstathou, G (2003), G (2008))

Neumann-Neumann Example



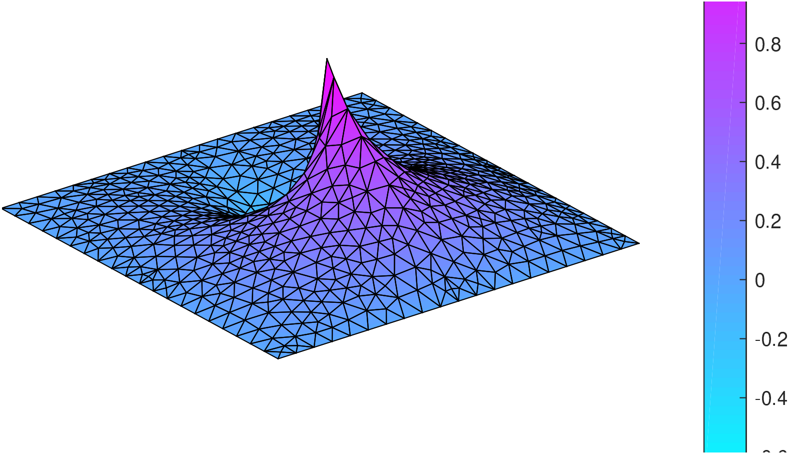
Solution to be computed by Neumann-Neumann

Neumann-Neumann initial guess



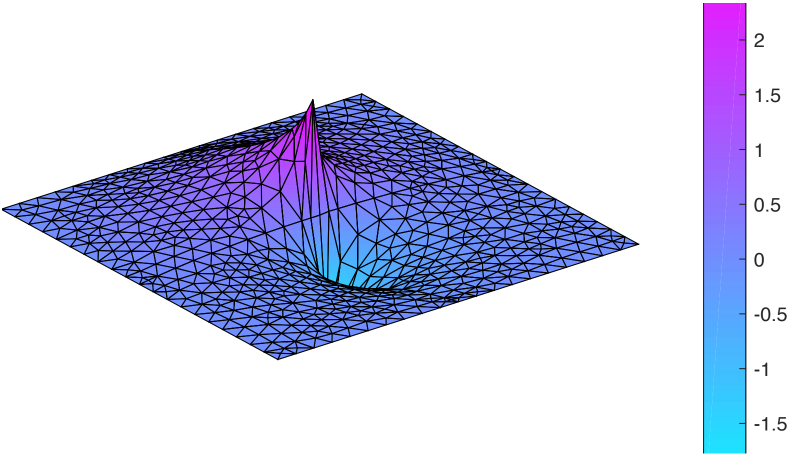
Computations done by Faycal Chaouqui

Neumann-Neumann example iteration 1



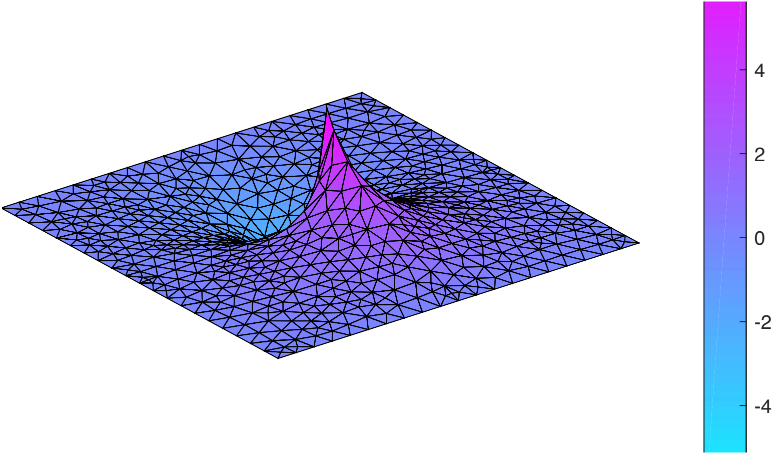
Computations done by Faycal Chaouqui

Neumann-Neumann example iteration 2



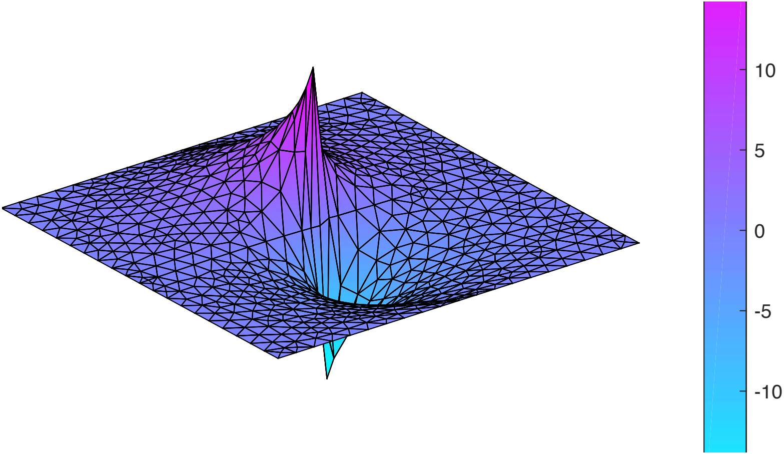
Computations done by Faycal Chaouqui

Neumann-Neumann example iteration 3



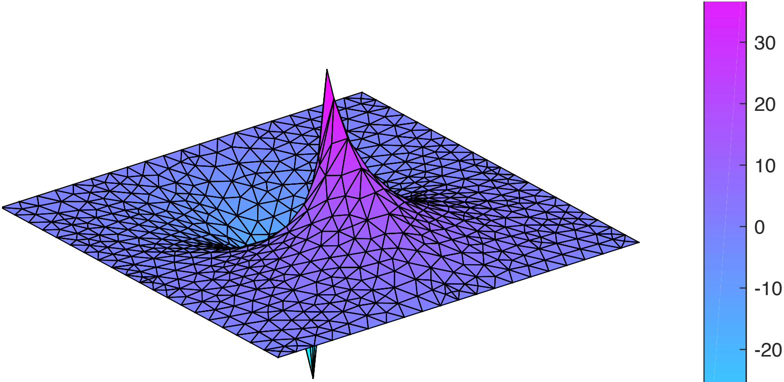
Computations done by Faycal Chaouqui

Neumann-Neumann example iteration 4



Computations done by Faycal Chaouqui

Neumann-Neumann example iteration 5



Computations done by Faycal Chaouqui

Well Posedness of Neumann-Neumann/FETI

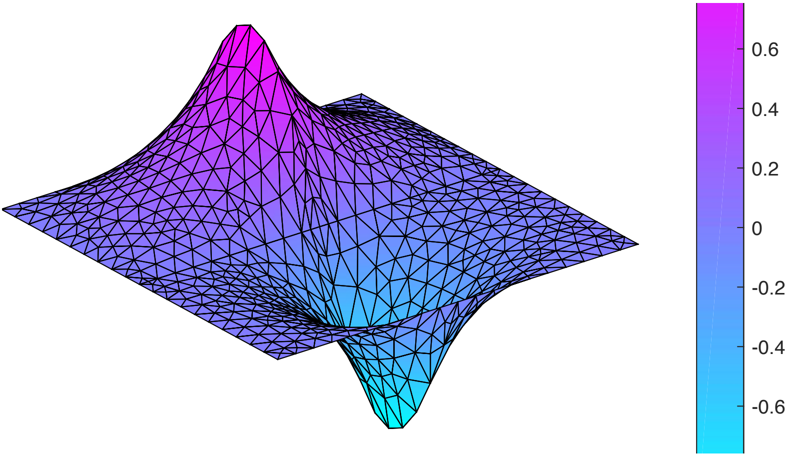
If cross points are present in the decomposition:

- Neumann-Neumann (and FETI) are not well posed domain decomposition methods at the continuous level !
- Neumann-Neumann (and FETI) can only be used as preconditioners, not as standalone solvers.

With a local coarse space correction, one can obtain a well posed Neumann-Neumann (and FETI) method at the continuous level (in H^2) even with cross points!

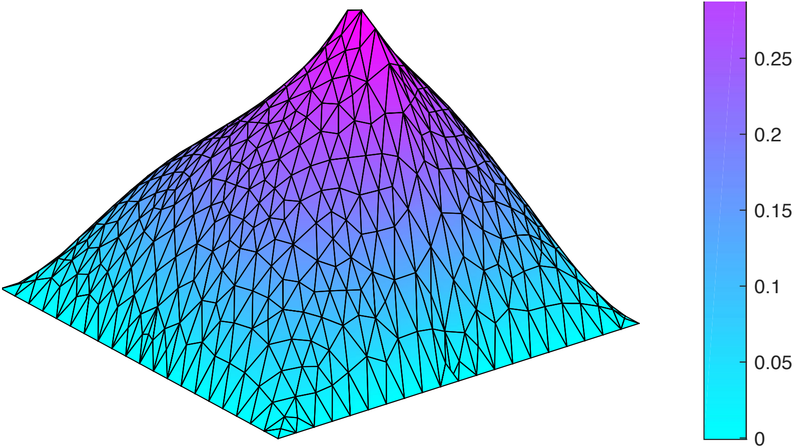
(Chaouqui, G, Santugini 2019)

Well Posed Neumann-Neumann initial guess



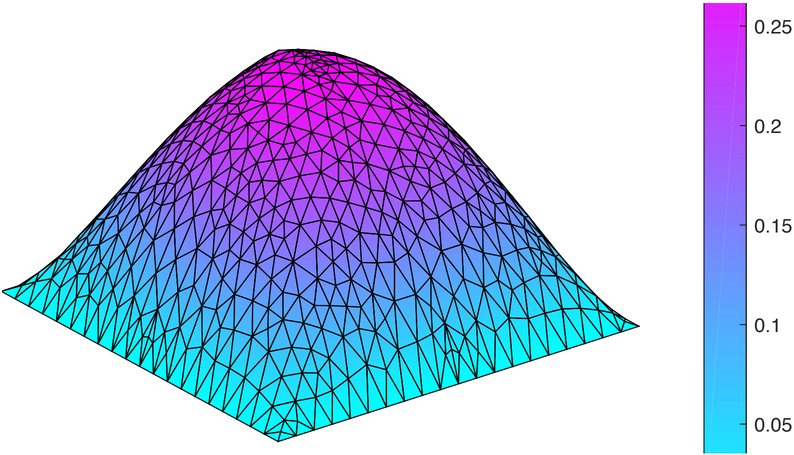
Well posed using a local coarse correction!

Well Posed Neumann-Neumann iteration 1



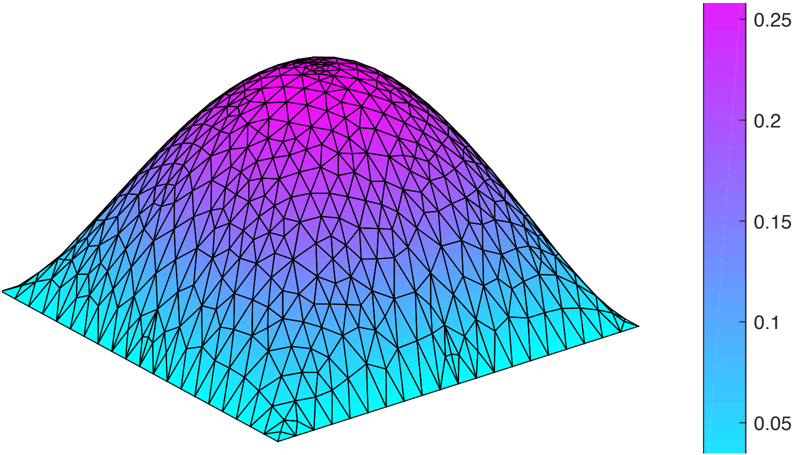
Well posed using a local coarse correction!

Well Posed Neumann-Neumann iteration 2



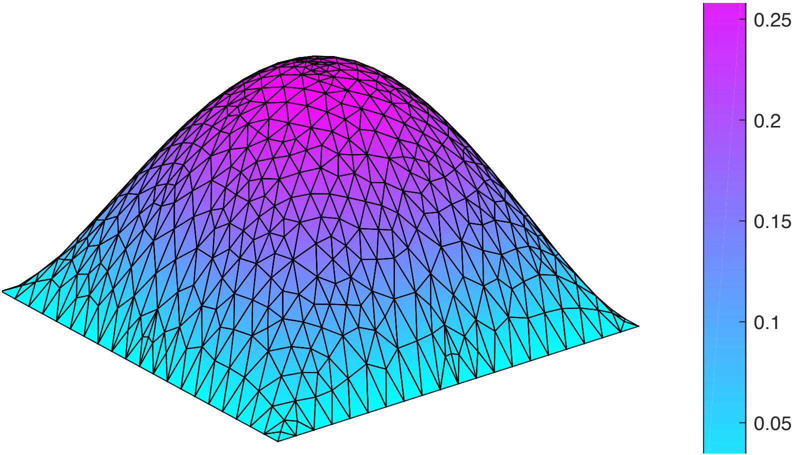
Well posed using a local coarse correction!

Well Posed Neumann-Neumann iteration 3



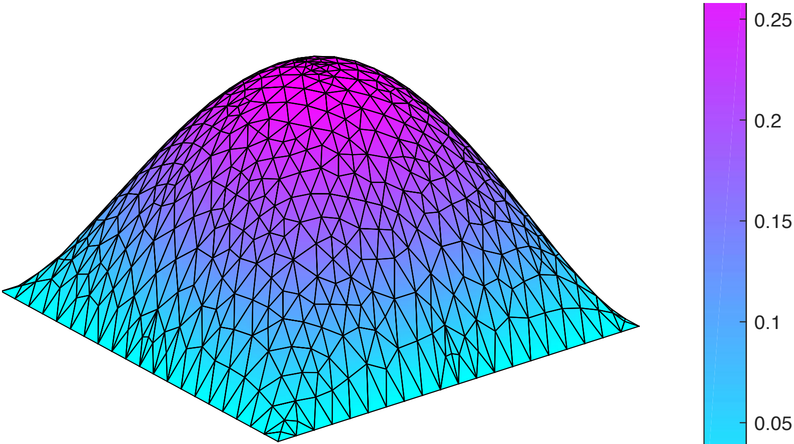
Well posed using a local coarse correction!

Well Posed Neumann-Neumann iteration 4



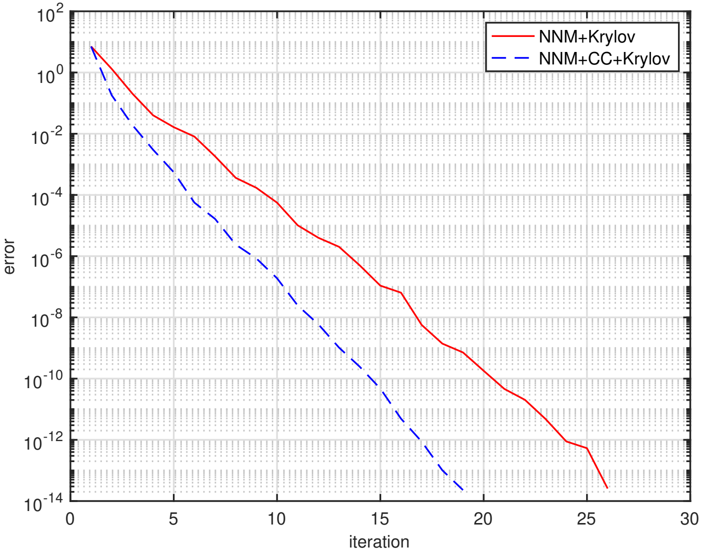
Well posed using a local coarse correction!

Well Posed Neumann-Neumann iteration 5



Well posed using a local coarse correction!

Comparison with Krylov acceleration: 16×16



With (blue) and without (red) local coarse correction!

Optimized Schwarz Methods

Replace Dirichlet conditions in parallel Schwarz,

$$\begin{aligned} u_1^n(\Gamma_1) &= u_2^{n-1}(\Gamma_1) \\ u_2^n(\Gamma_2) &= u_1^{\mathbf{n-1}}(\Gamma_2) \end{aligned}$$

by combination of Dirichlet and Neumann traces,

$$\begin{aligned} (\partial_{n_1} + p)u_1^n(\Gamma_1) &= (\partial_{n_1} + p)u_2^{n-1}(\Gamma_1) \\ (\partial_{n_2} + p)u_2^n(\Gamma_2) &= (\partial_{n_2} + p)u_1^{\mathbf{n-1}}(\Gamma_2) \end{aligned}$$

(already proposed by Lions in 1989)

Optimized Schwarz Methods

Replace Dirichlet conditions in parallel Schwarz,

$$\begin{aligned} u_1^n(\Gamma_1) &= u_2^{n-1}(\Gamma_1) \\ u_2^n(\Gamma_2) &= u_1^{\mathbf{n-1}}(\Gamma_2) \end{aligned}$$

by combination of Dirichlet and Neumann traces,

$$\begin{aligned} (\partial_{n_1} + p)u_1^n(\Gamma_1) &= (\partial_{n_1} + p)u_2^{n-1}(\Gamma_1) \\ (\partial_{n_2} + p)u_2^n(\Gamma_2) &= (\partial_{n_2} + p)u_1^{\mathbf{n-1}}(\Gamma_2) \end{aligned}$$

(already proposed by Lions in 1989)

Theorem (Nataf et al (1994))

If $p = DtN$, the method converges in 2 iterations, and for J subdomains of a strip decomposition in J iterations.

Optimized Schwarz Methods

Replace Dirichlet conditions in parallel Schwarz,

$$\begin{array}{rcl} u_1^n(\Gamma_1) & = & u_2^{n-1}(\Gamma_1) \\ u_2^n(\Gamma_2) & = & u_1^{\mathbf{n-1}}(\Gamma_2) \end{array}$$

by combination of Dirichlet and Neumann traces,

$$\begin{array}{rcl} (\partial_{n_1} + p)u_1^n(\Gamma_1) & = & (\partial_{n_1} + p)u_2^{n-1}(\Gamma_1) \\ (\partial_{n_2} + p)u_2^n(\Gamma_2) & = & (\partial_{n_2} + p)u_1^{\mathbf{n-1}}(\Gamma_2) \end{array}$$

(already proposed by Lions in 1989)

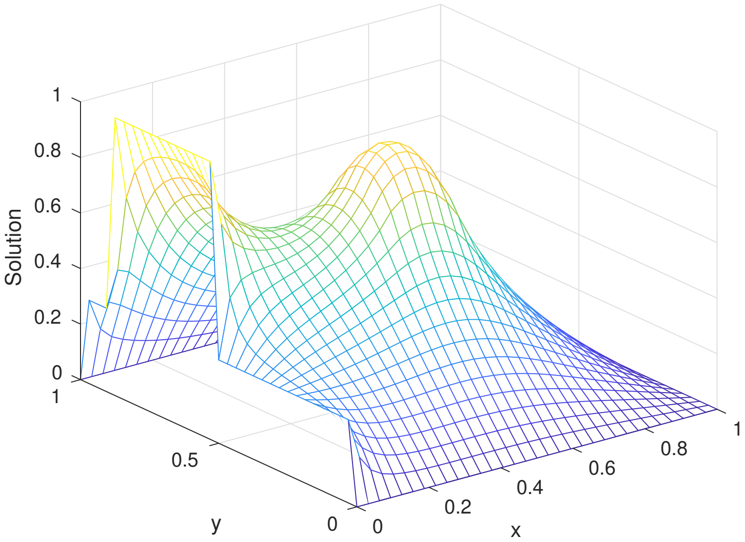
Theorem (Nataf et al (1994))

If $p = DtN$, the method converges in 2 iterations, and for J subdomains of a strip decomposition in J iterations.

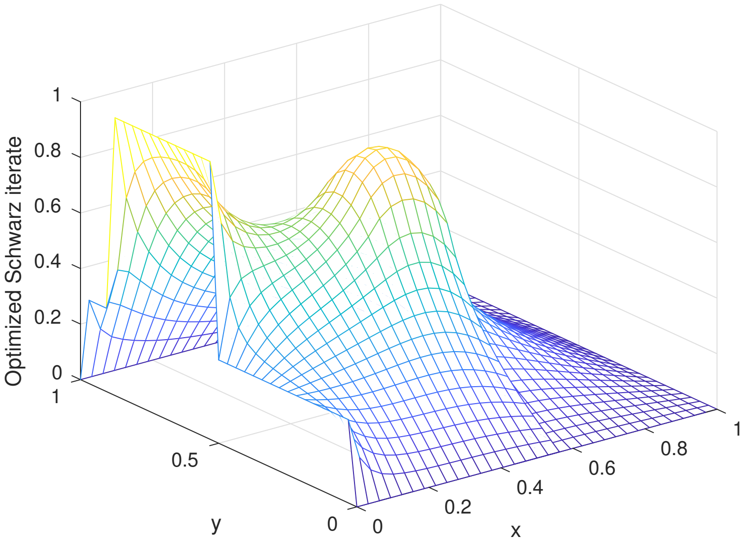
Theorem (G. and Zhang (2017))

If $p = DtN$, the alternating method corresponds to a block LU decomposition and converges in 1 iteration for any strip decomposition (no cross points).

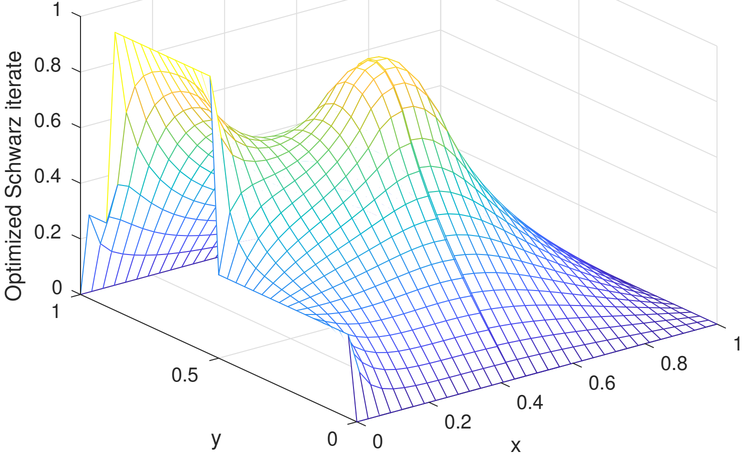
Example: optimized $p \in \mathbb{R}$, parallel



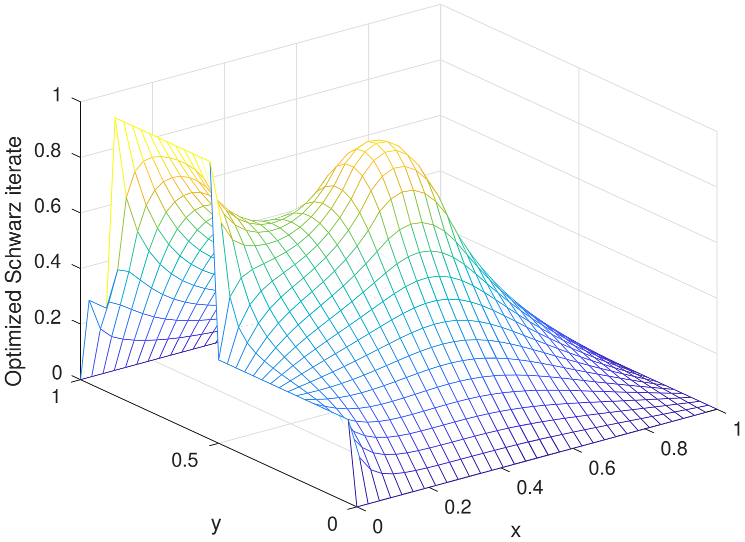
Optimized Schwarz iteration 1



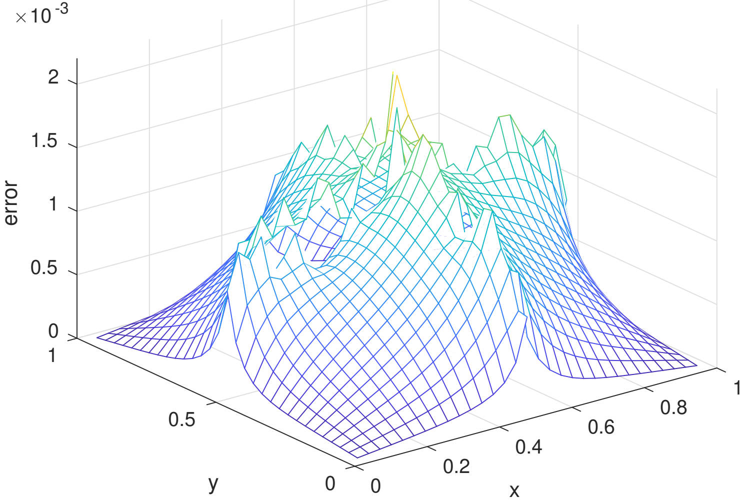
Optimized Schwarz iteration 2



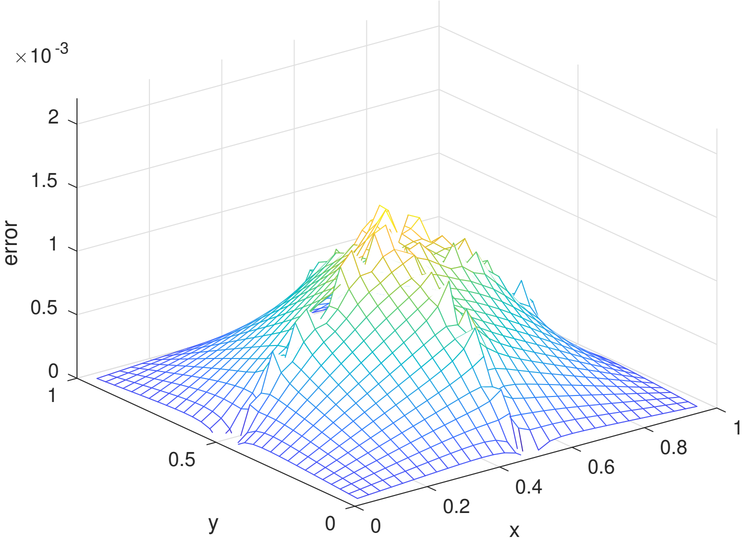
Optimized Schwarz iteration 3



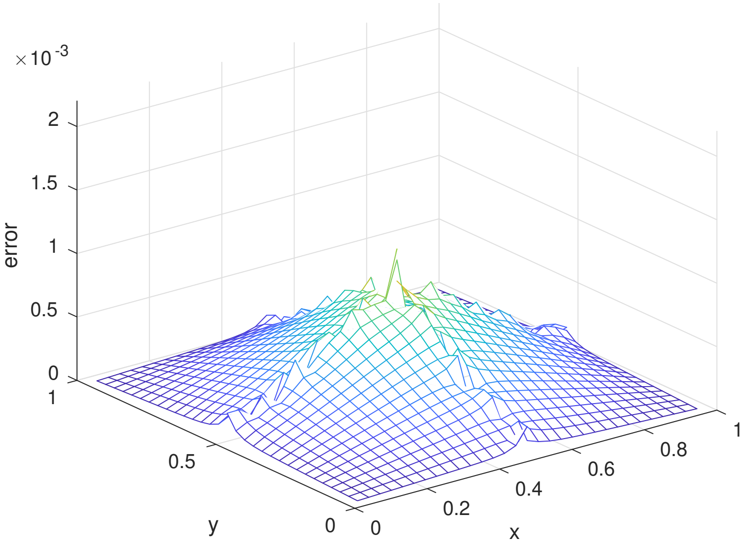
Optimized Schwarz error with cross-point



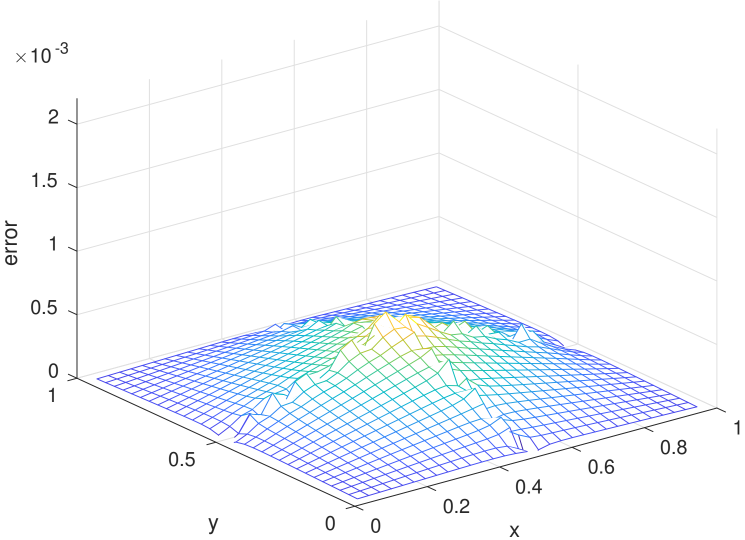
Optimized Schwarz error with cross-point



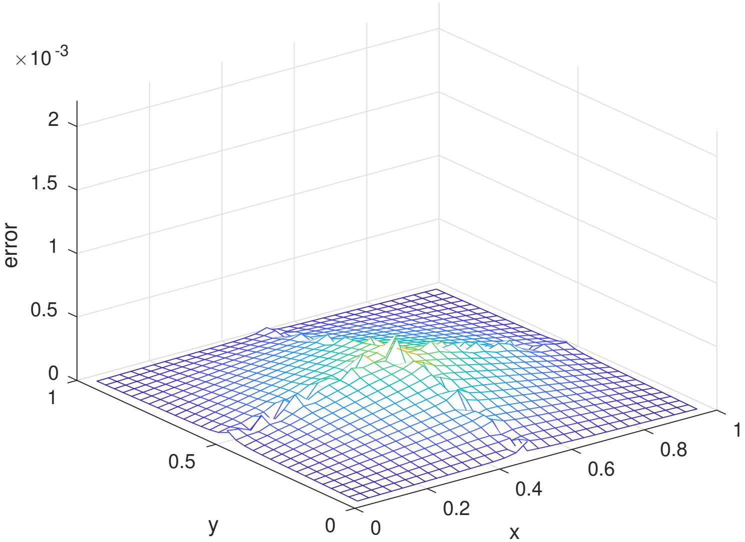
Optimized Schwarz error with cross-point



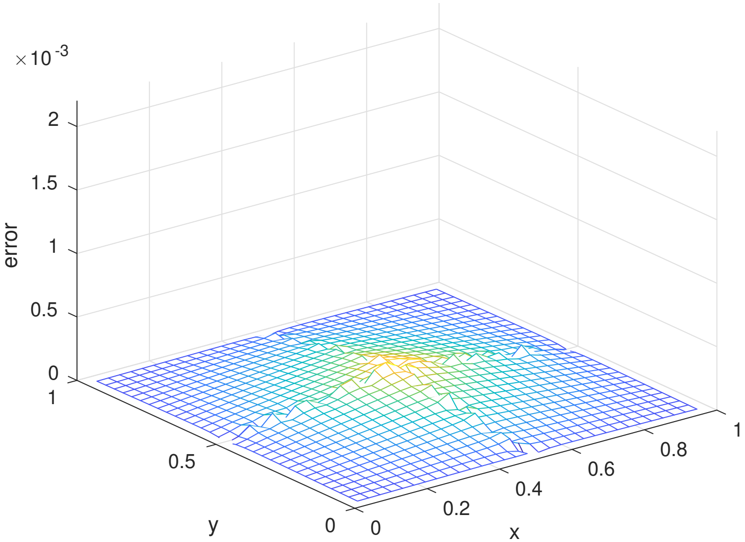
Optimized Schwarz error with cross-point



Optimized Schwarz error with cross-point



Optimized Schwarz error with cross-point



Methods Using Such Techniques

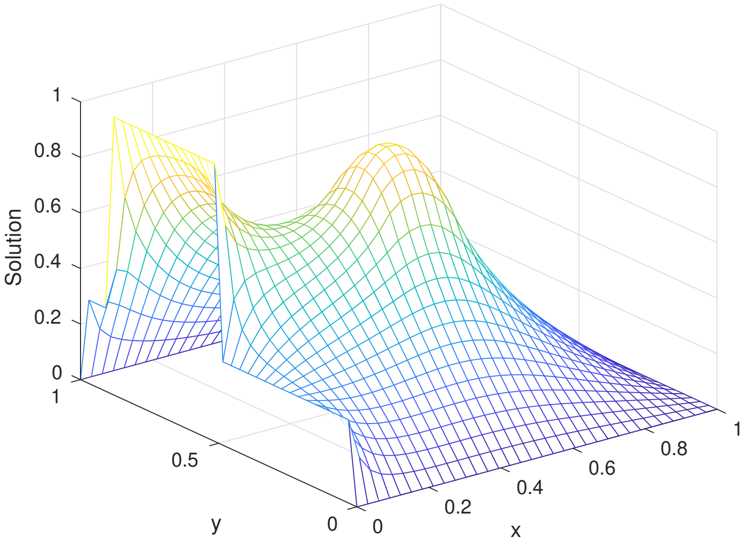
- Bank and Jimack (2001), Bank and Vassilevski (2008)
- AILU (G, Nataf (2001, 2005), Achdou, Nataf (2007))
- Sweeping Preconditioner (Engquist and Ying 2011)
- Single Layer Potential Method (Stolk 2013)
- Source Transfer (Chen and Xiang 2013)
- Method of Polarized Traces (Zepeda-Núñez and Demanet 2015)

Reviews on These Techniques

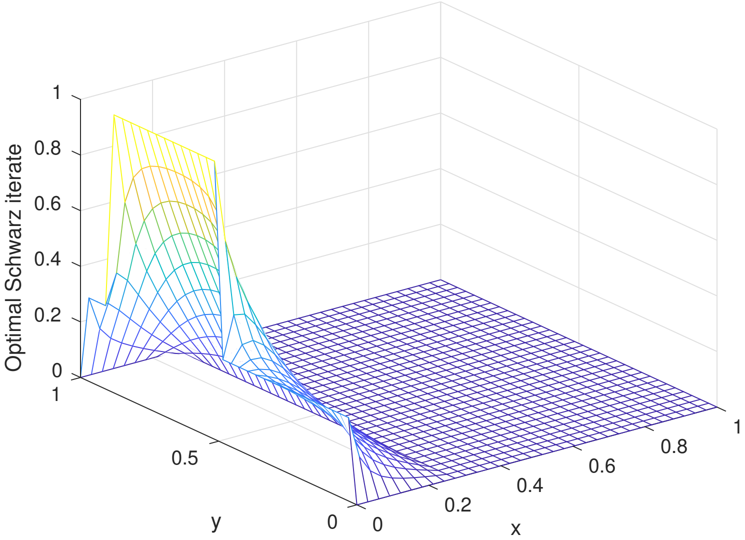
G. (2006): Optimized Schwarz Methods

G. and Zhang (2019): A Class of Iterative Solvers for the Helmholtz Equation: Factorizations, Sweeping Preconditioners, Source Transfer, Single Layer Potentials, Polarized Traces, and Optimized Schwarz Methods

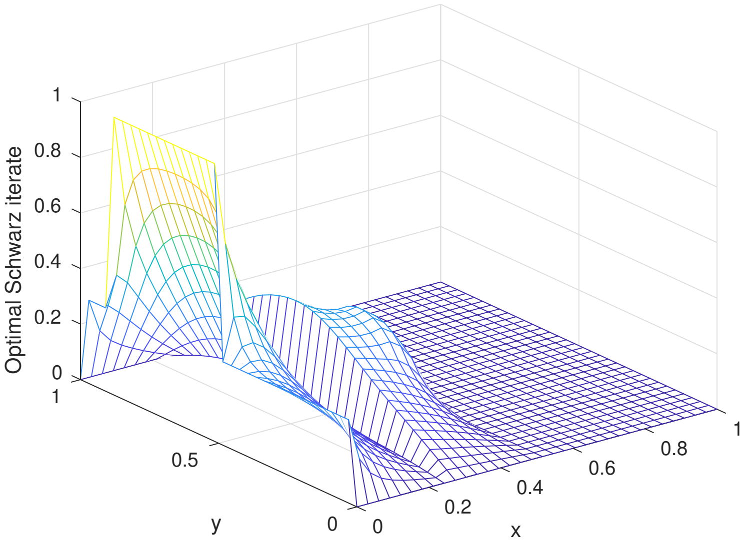
Example: temperature in a room



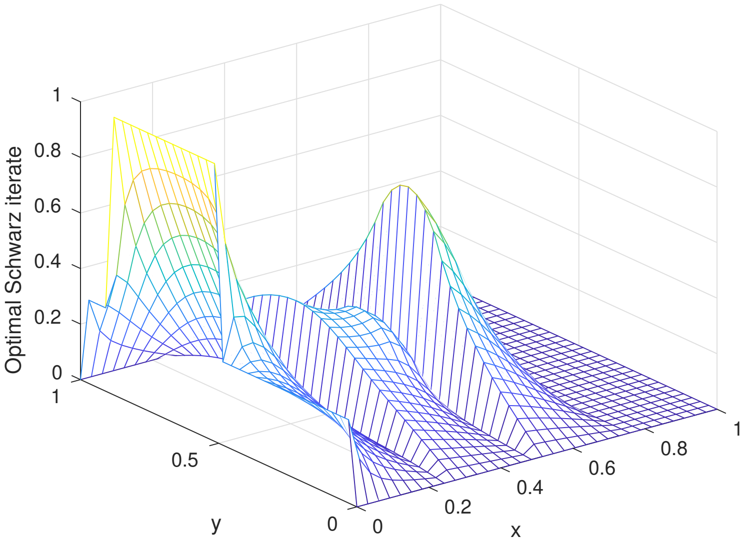
Optimal Sweeping Schwarz Iteration 1



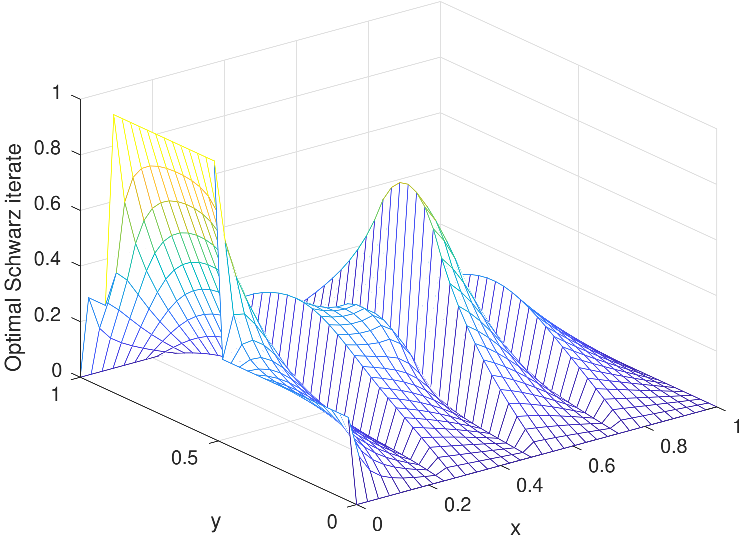
Optimal Sweeping Schwarz Iteration 1



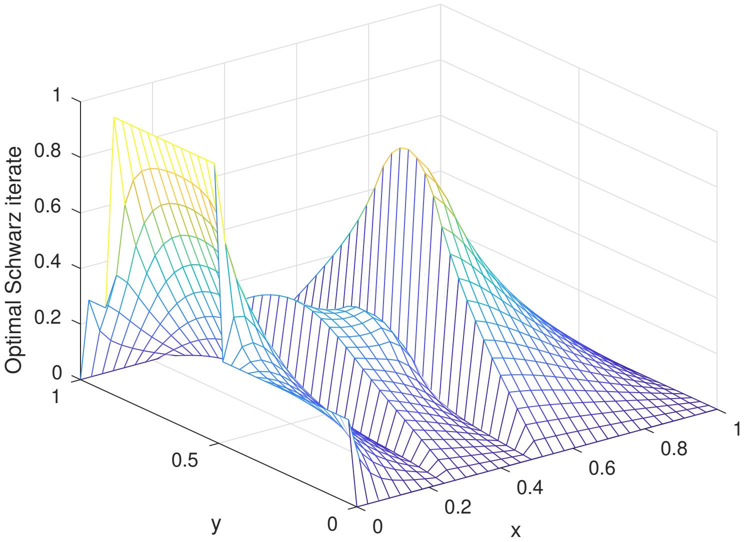
Optimal Sweeping Schwarz Iteration 1



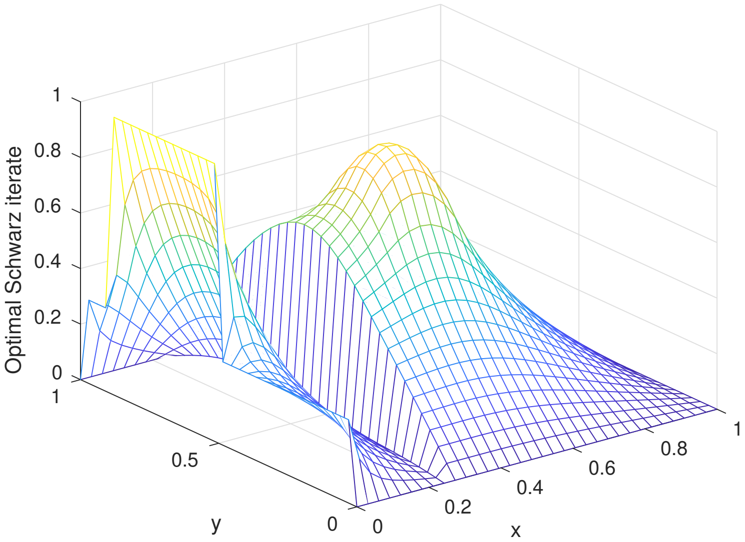
Optimal Sweeping Schwarz Iteration 1



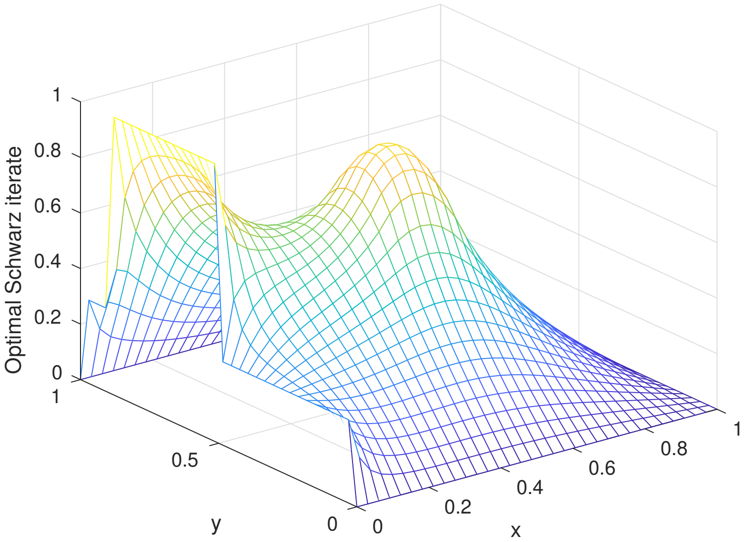
Optimal Sweeping Schwarz Iteration 1



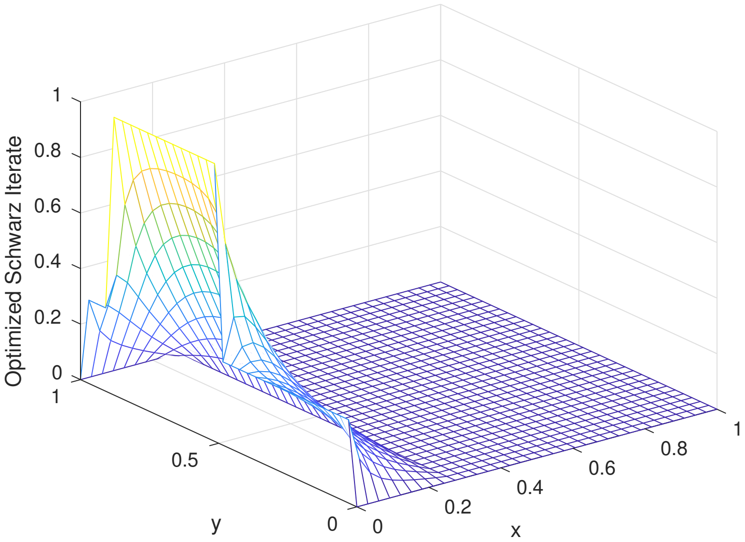
Optimal Sweeping Schwarz Iteration 1



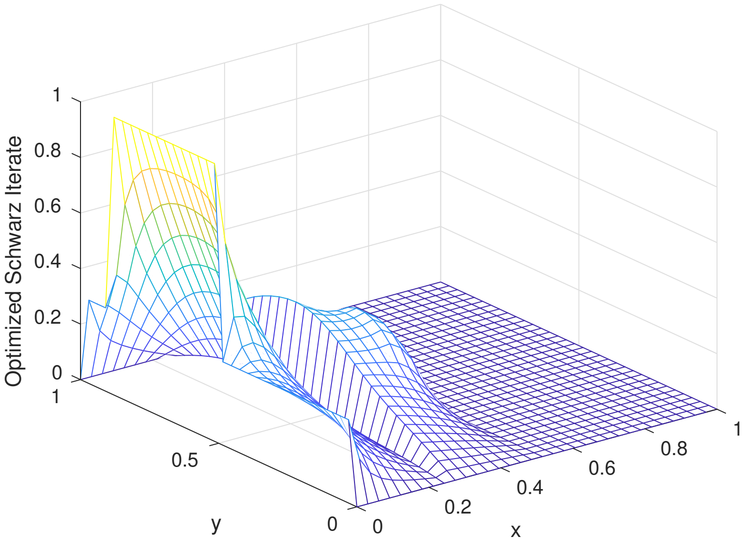
Optimal Sweeping Schwarz Iteration 1



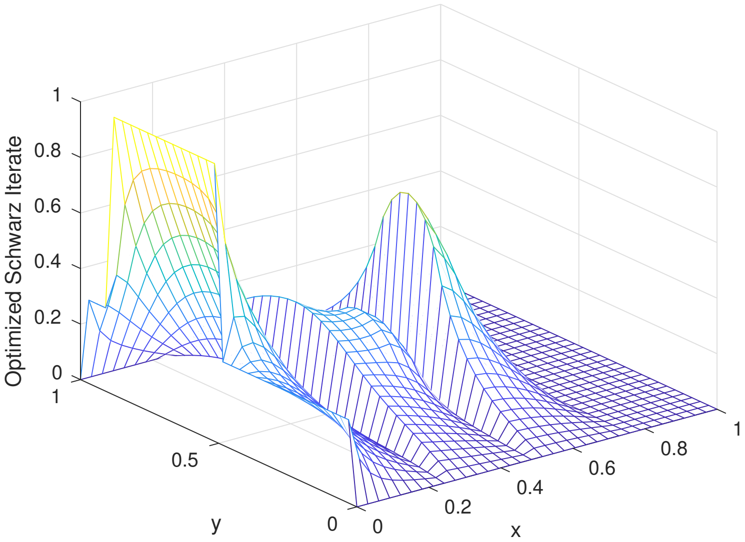
Optimized Sweeping Schwarz Iteration 1



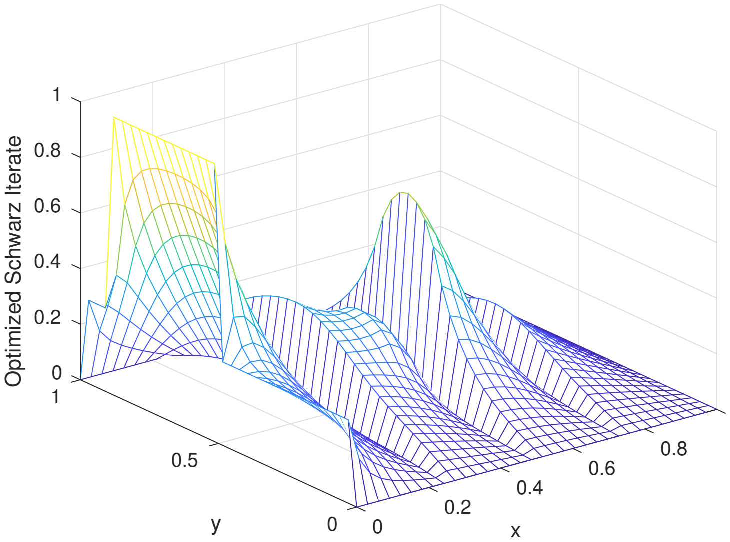
Optimized Sweeping Schwarz Iteration 1



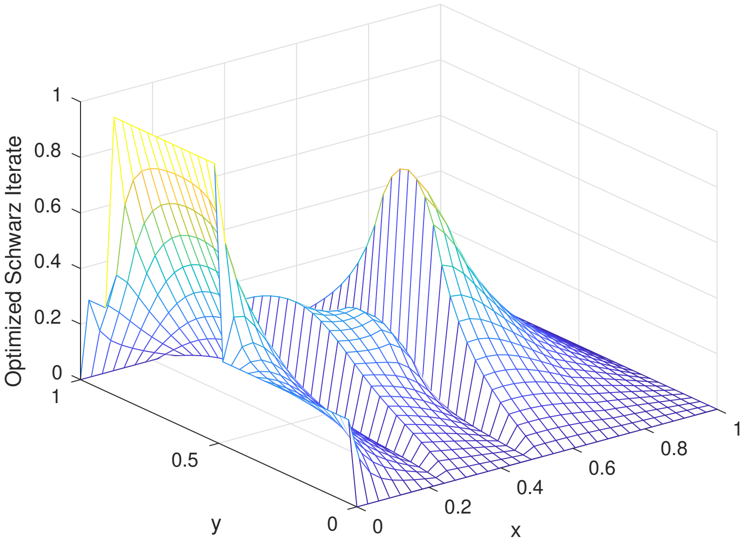
Optimized Sweeping Schwarz Iteration 1



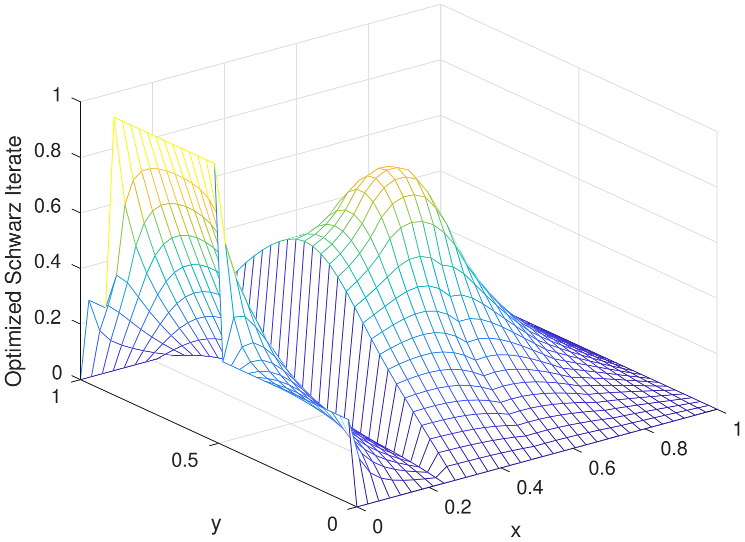
Optimized Sweeping Schwarz Iteration 1



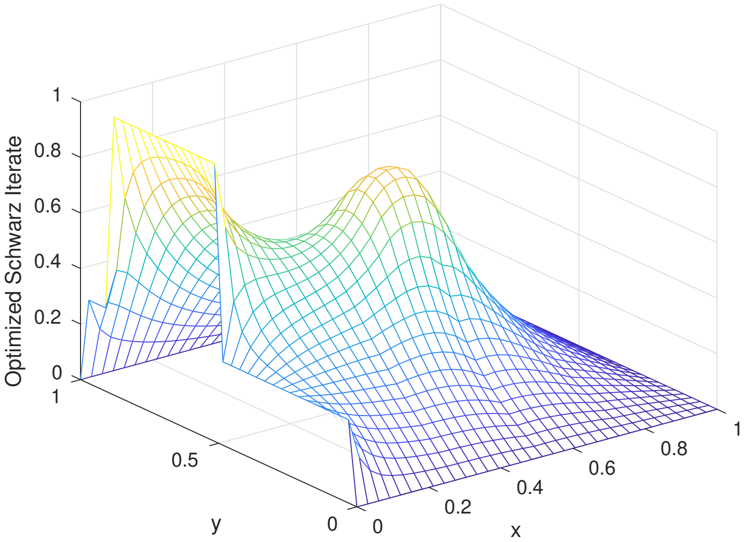
Optimized Sweeping Schwarz Iteration 1



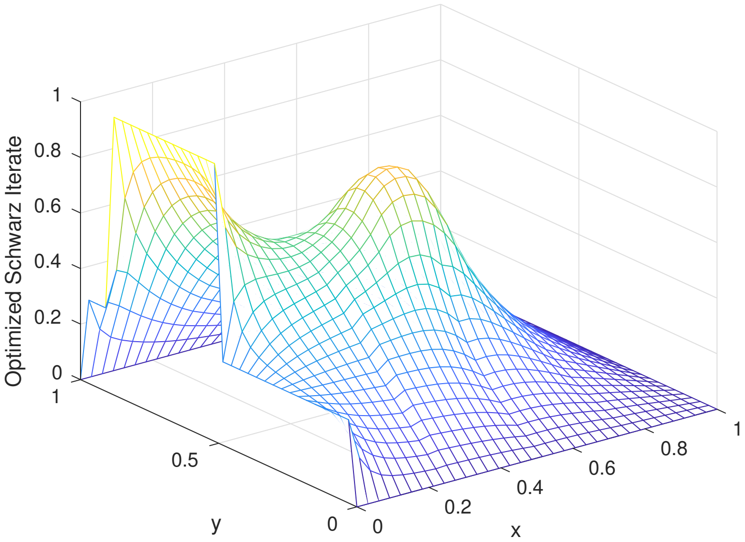
Optimized Sweeping Schwarz Iteration 1



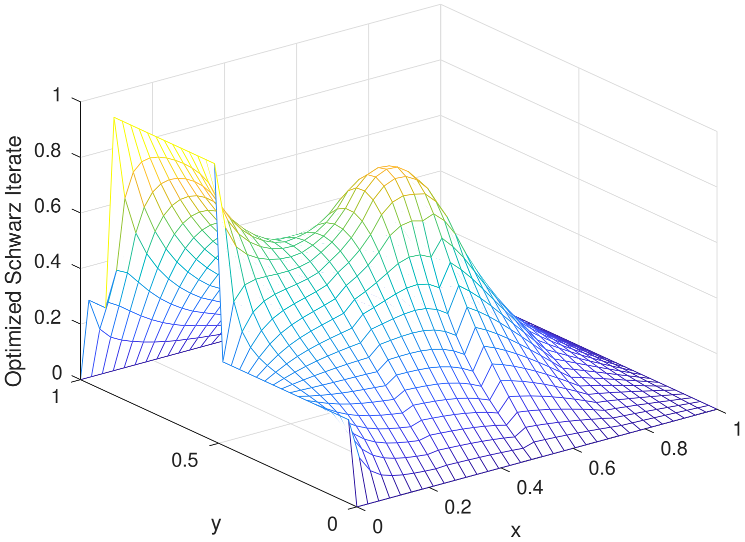
Optimized Sweeping Schwarz Iteration 1



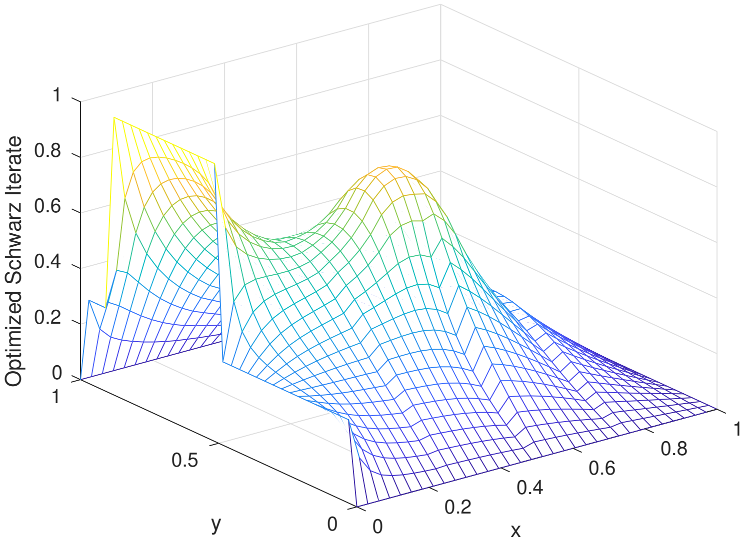
Optimized Sweeping Schwarz Iteration 2



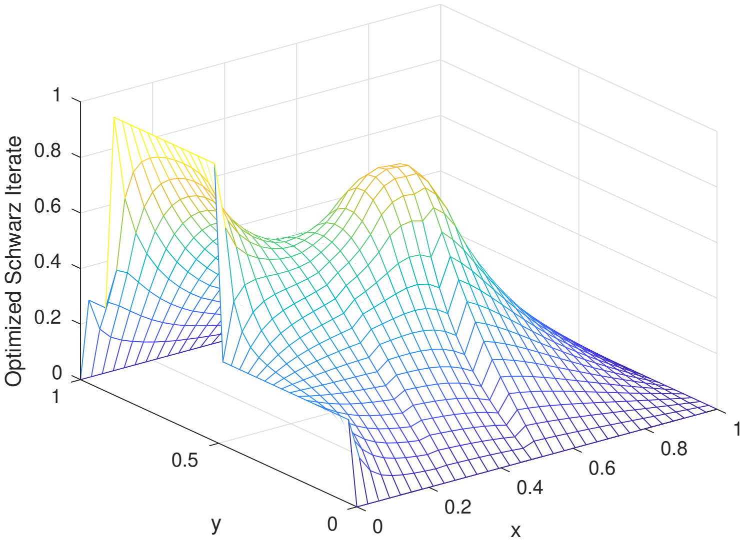
Optimized Sweeping Schwarz Iteration 2



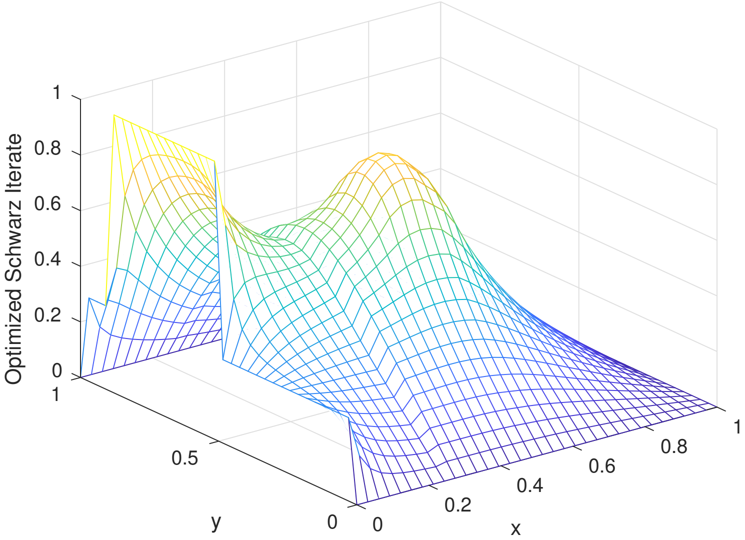
Optimized Sweeping Schwarz Iteration 2



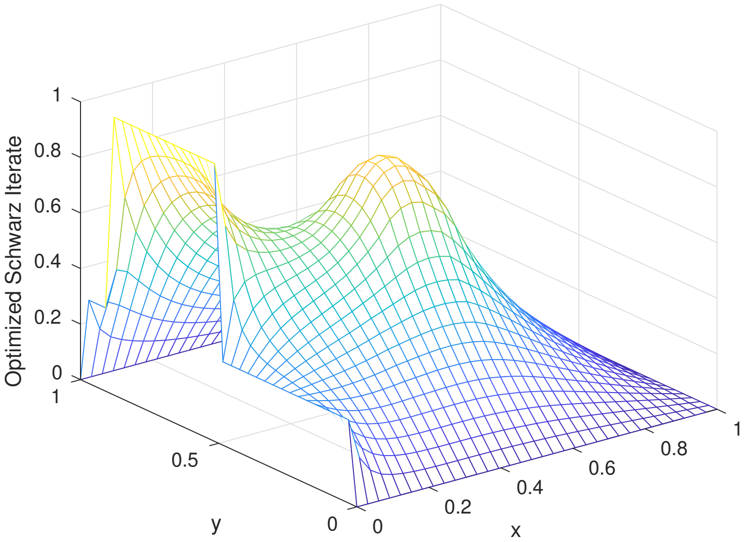
Optimized Sweeping Schwarz Iteration 2



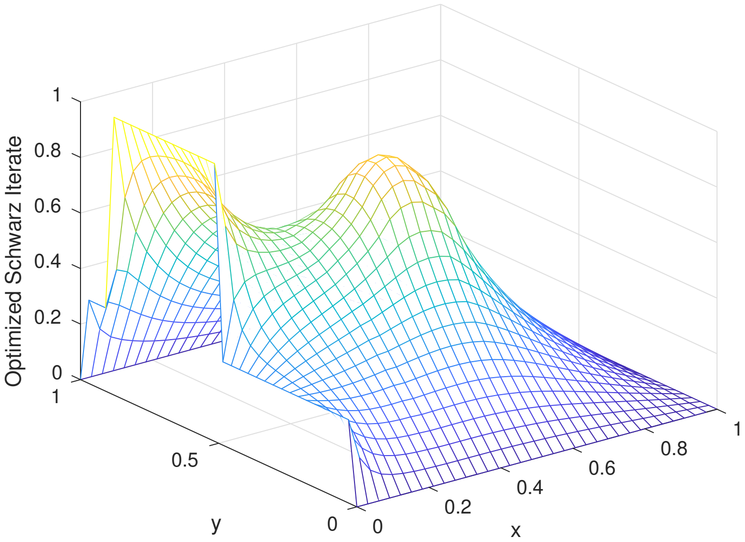
Optimized Sweeping Schwarz Iteration 2



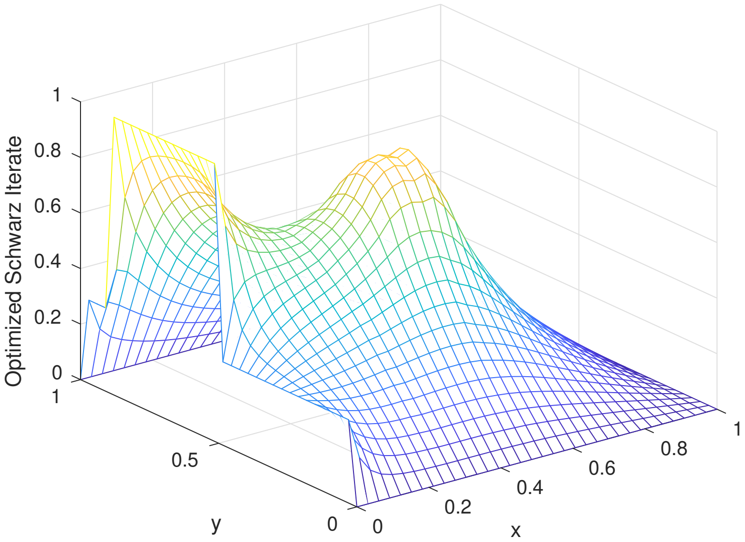
Optimized Sweeping Schwarz Iteration 2



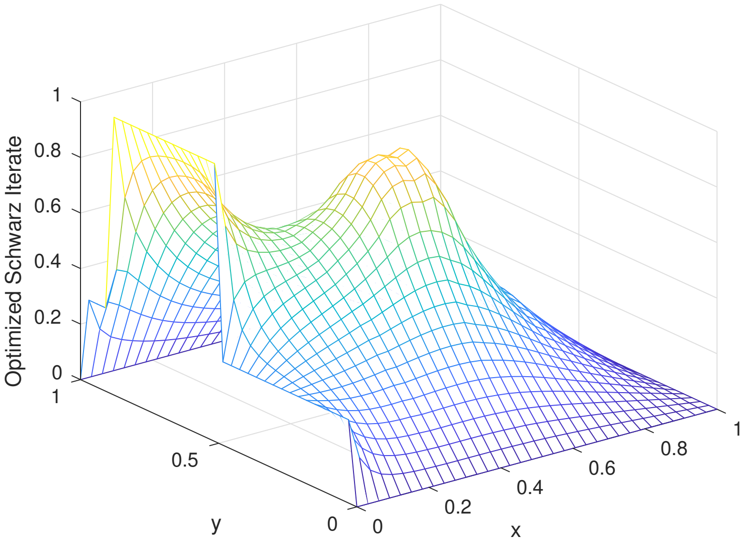
Optimized Sweeping Schwarz Iteration 3



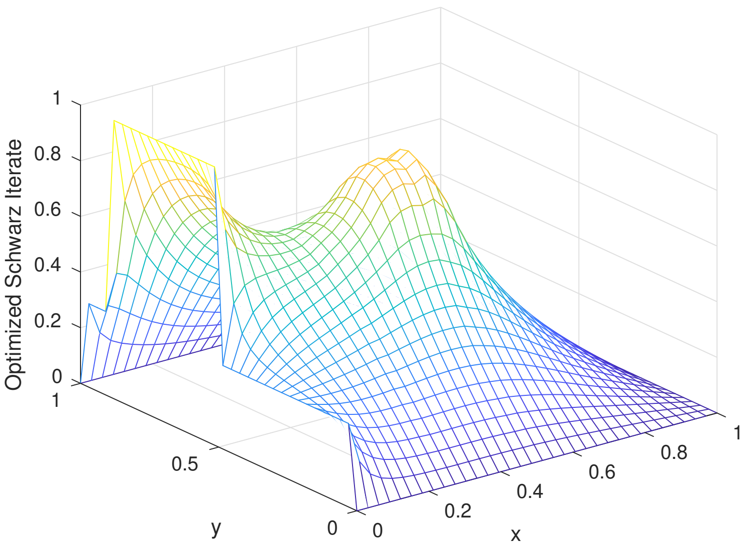
Optimized Sweeping Schwarz Iteration 3



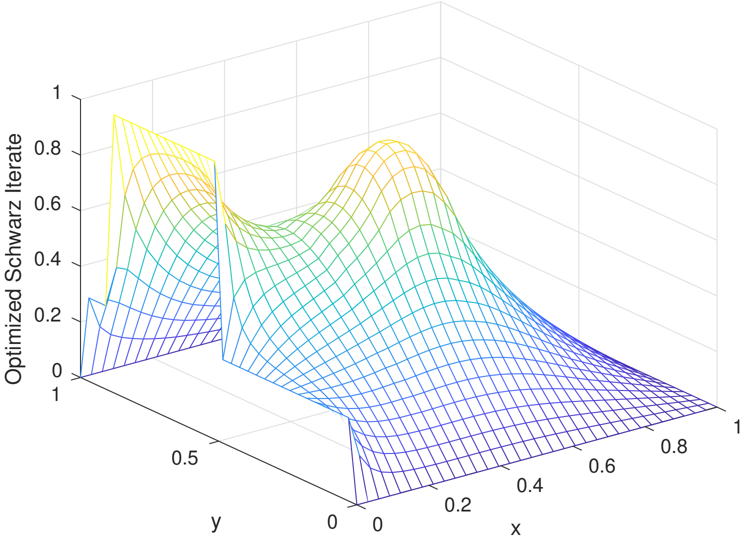
Optimized Sweeping Schwarz Iteration 3



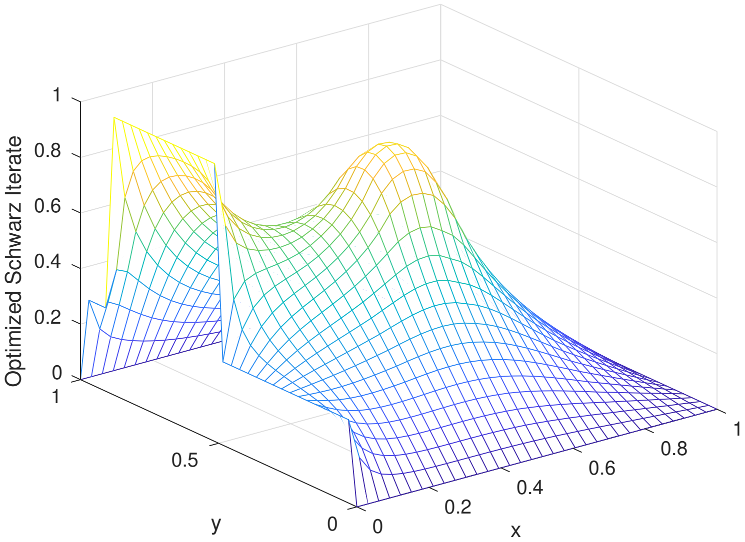
Optimized Sweeping Schwarz Iteration 3



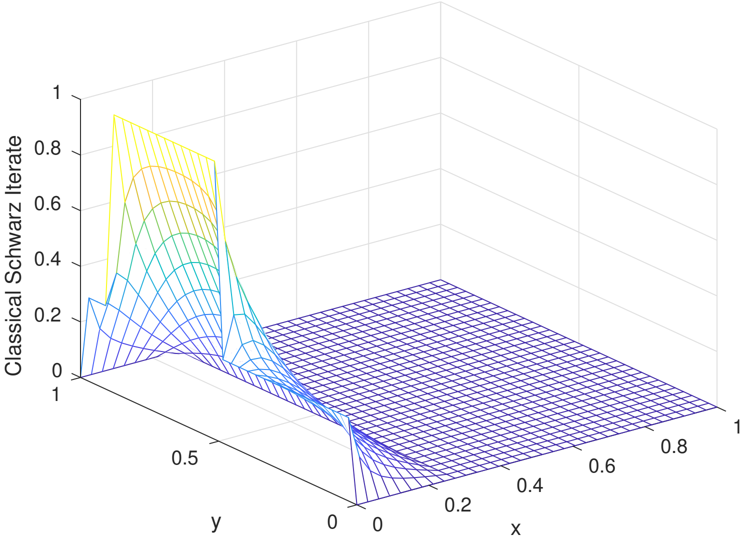
Optimized Sweeping Schwarz Iteration 3



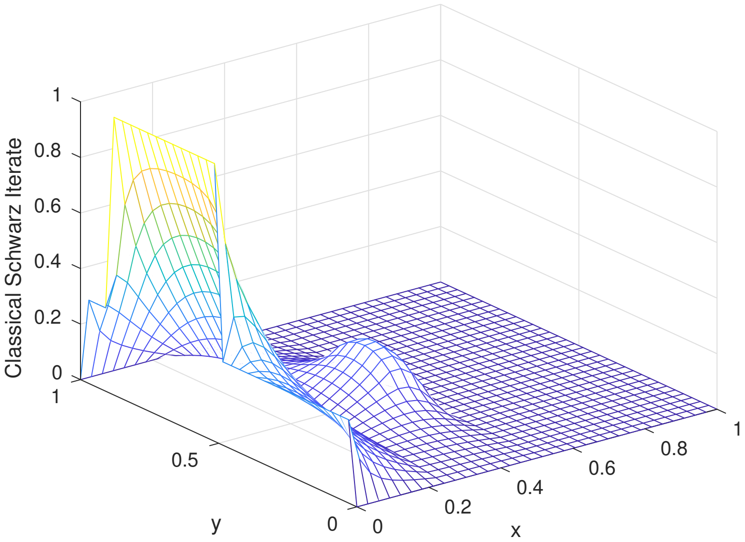
Optimized Sweeping Schwarz Iteration 3



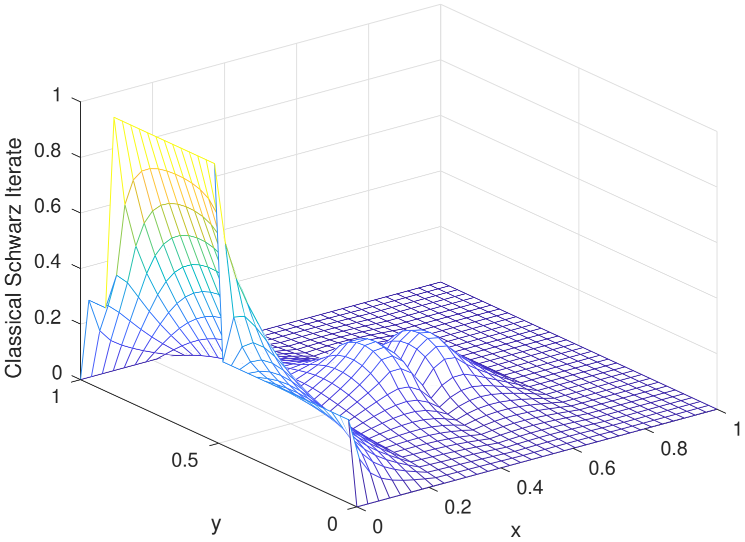
Classical Sweeping Schwarz Iteration 1



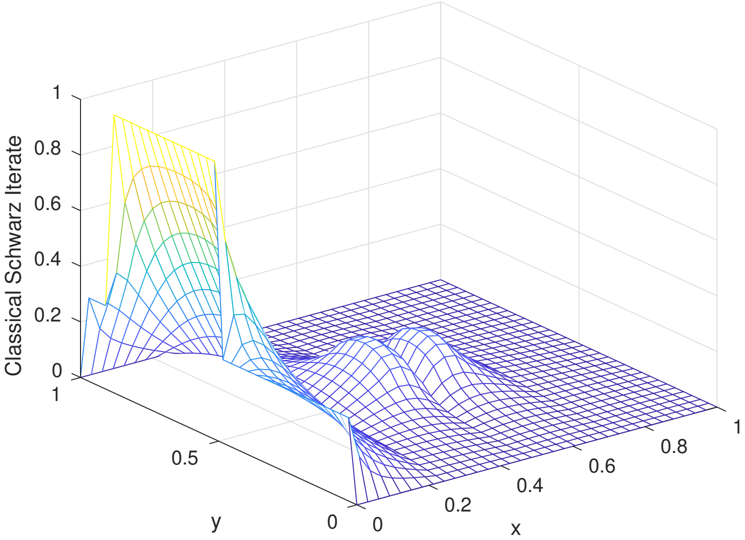
Classical Sweeping Schwarz Iteration 1



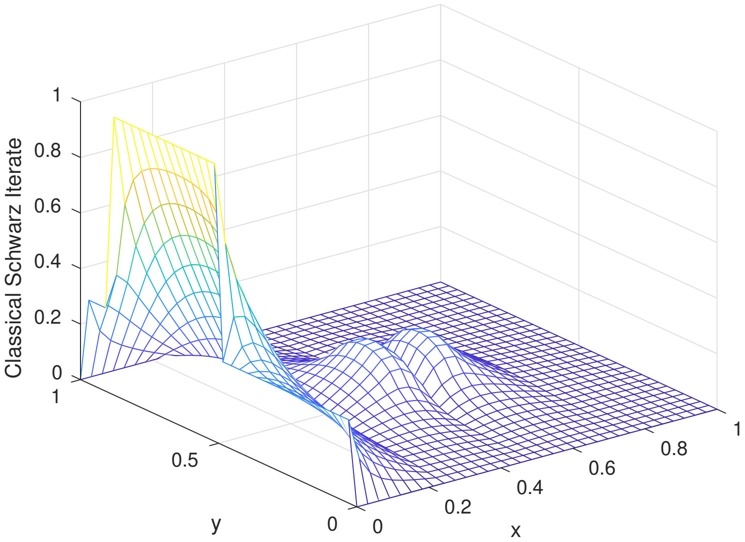
Classical Sweeping Schwarz Iteration 1



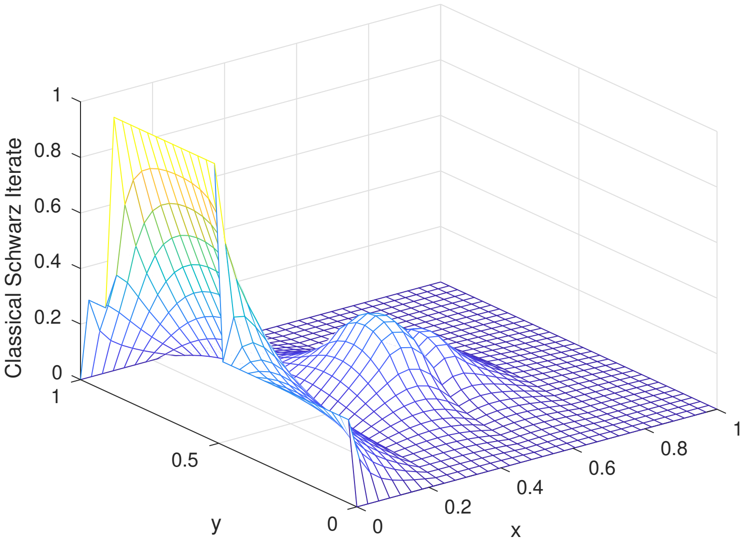
Classical Sweeping Schwarz Iteration 1



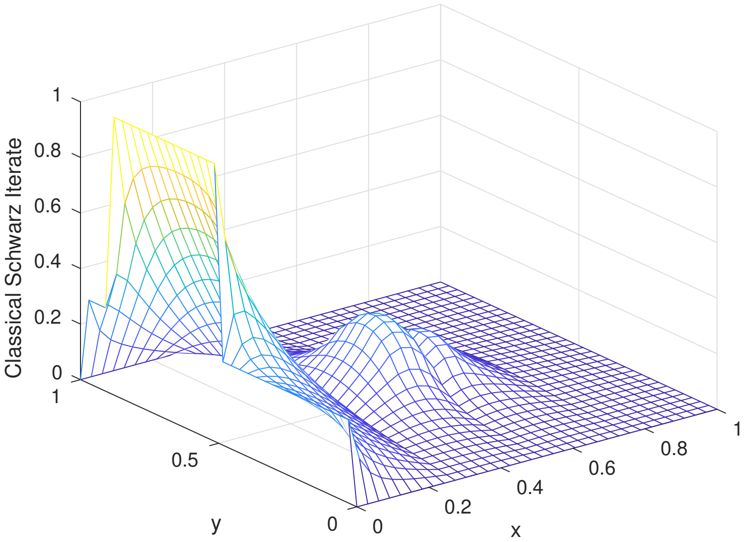
Classical Sweeping Schwarz Iteration 1



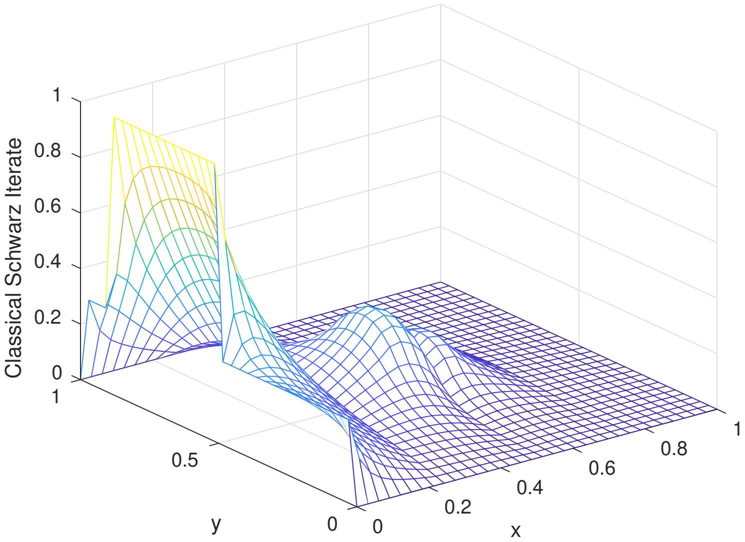
Classical Sweeping Schwarz Iteration 1



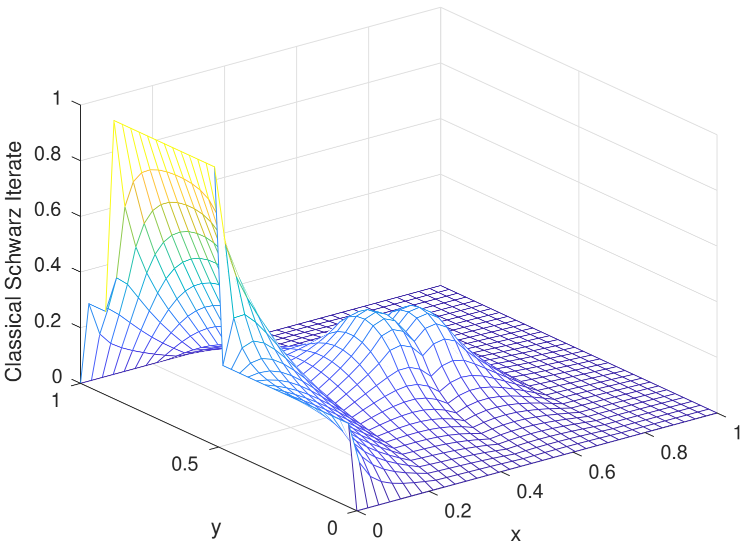
Classical Sweeping Schwarz Iteration 1



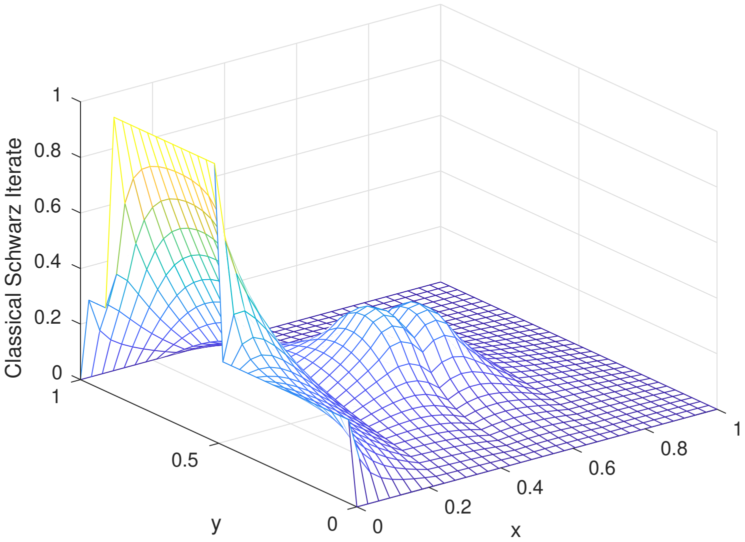
Classical Sweeping Schwarz Iteration 2



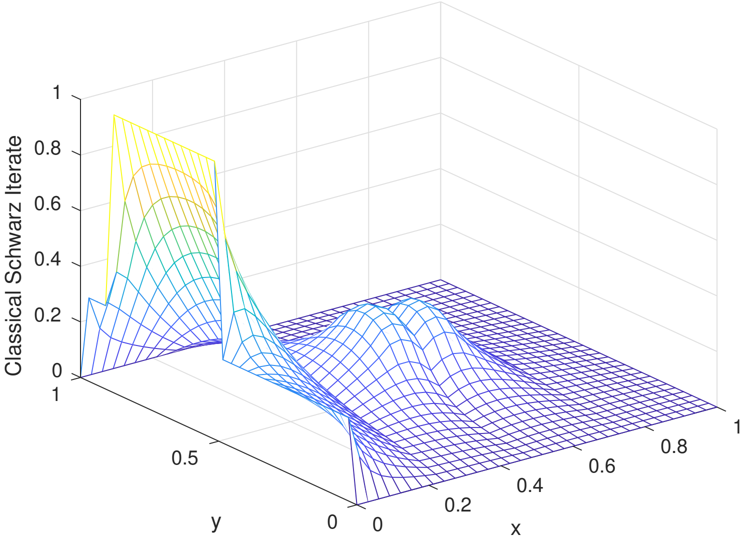
Classical Sweeping Schwarz Iteration 2



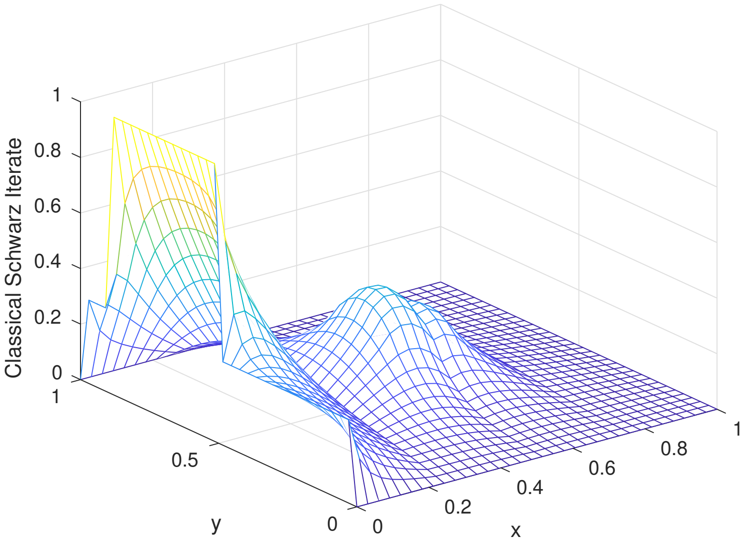
Classical Sweeping Schwarz Iteration 2



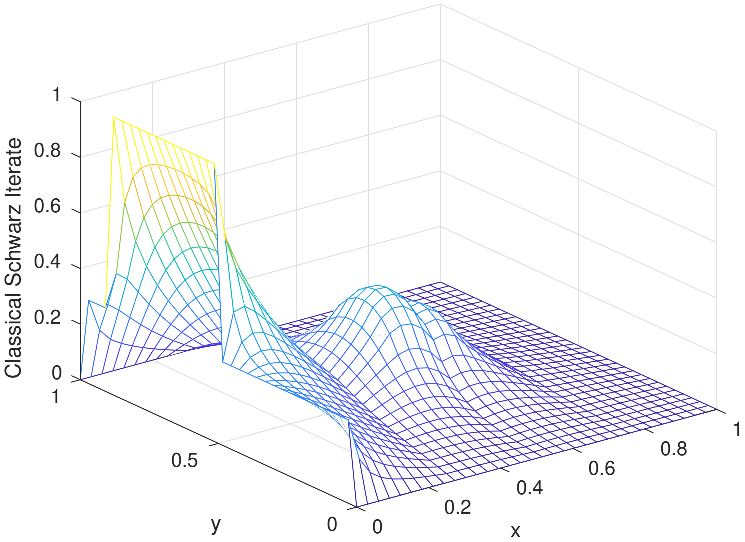
Classical Sweeping Schwarz Iteration 2



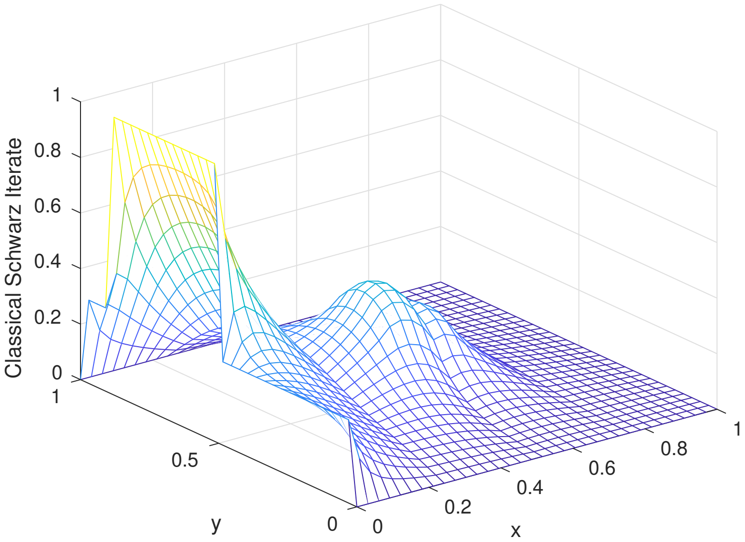
Classical Sweeping Schwarz Iteration 2



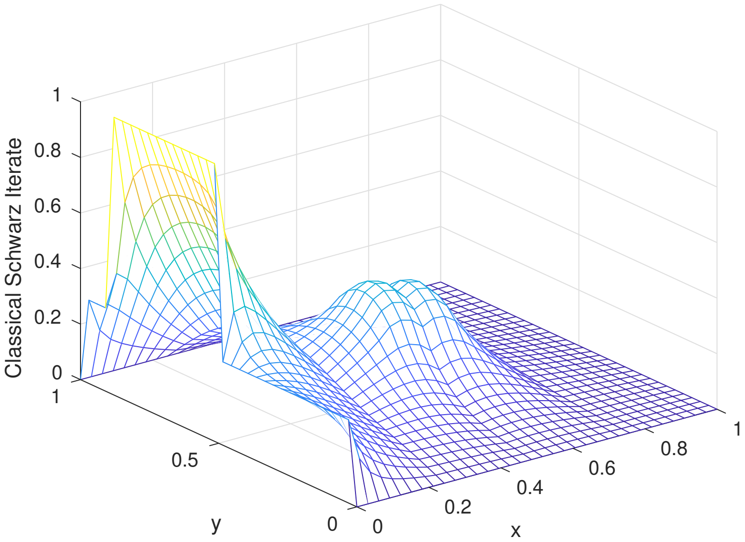
Classical Sweeping Schwarz Iteration 2



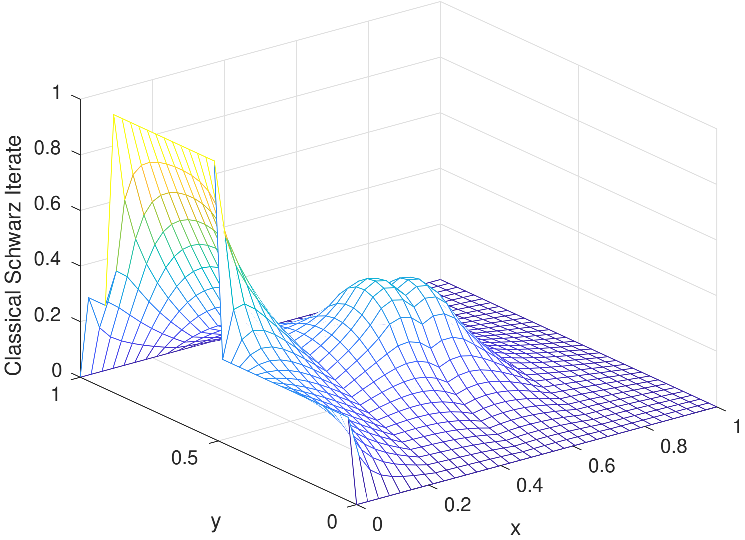
Classical Sweeping Schwarz Iteration 3



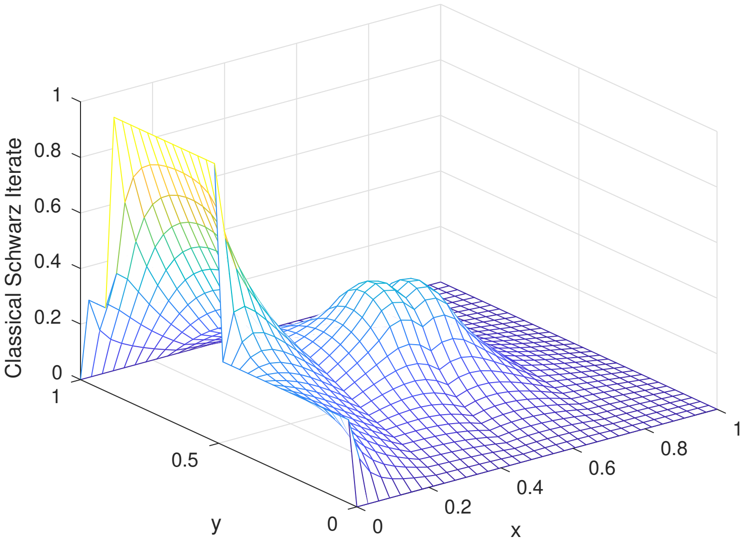
Classical Sweeping Schwarz Iteration 3



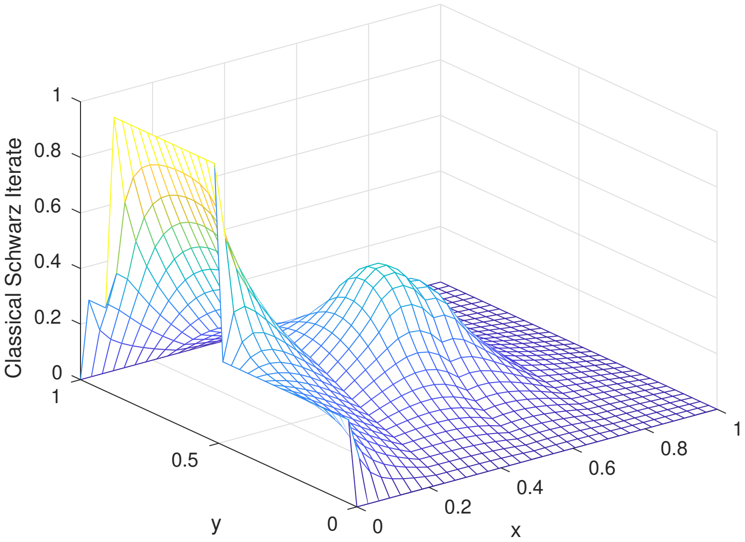
Classical Sweeping Schwarz Iteration 3



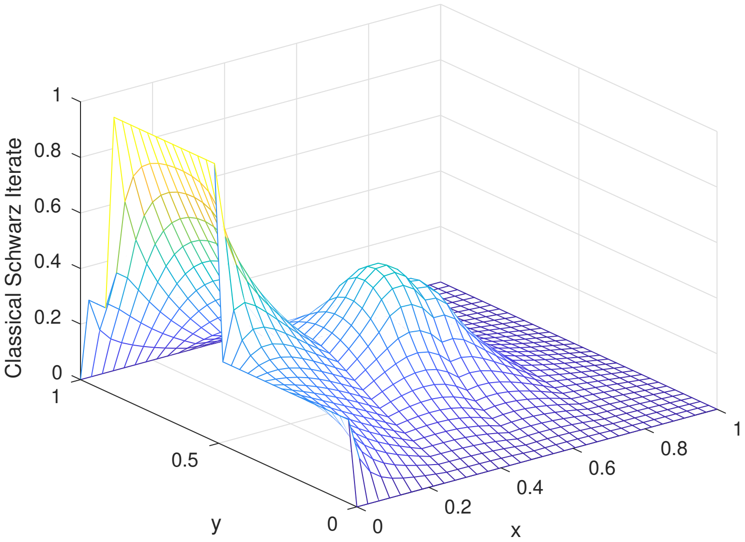
Classical Sweeping Schwarz Iteration 3



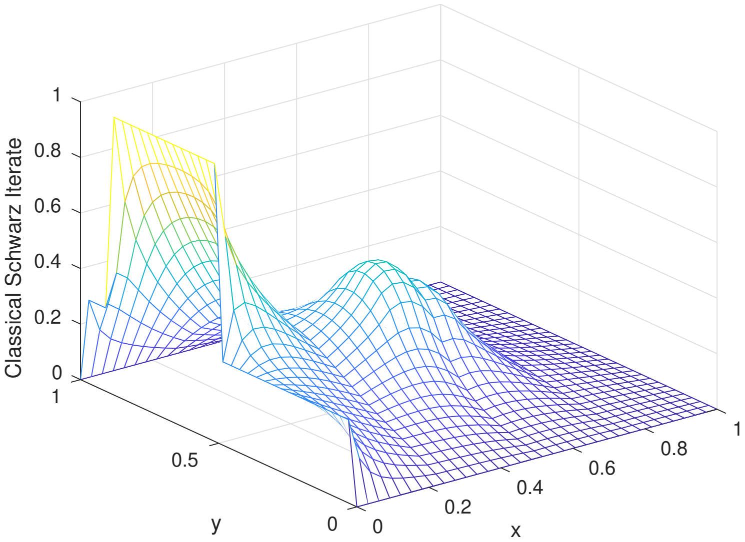
Classical Sweeping Schwarz Iteration 3



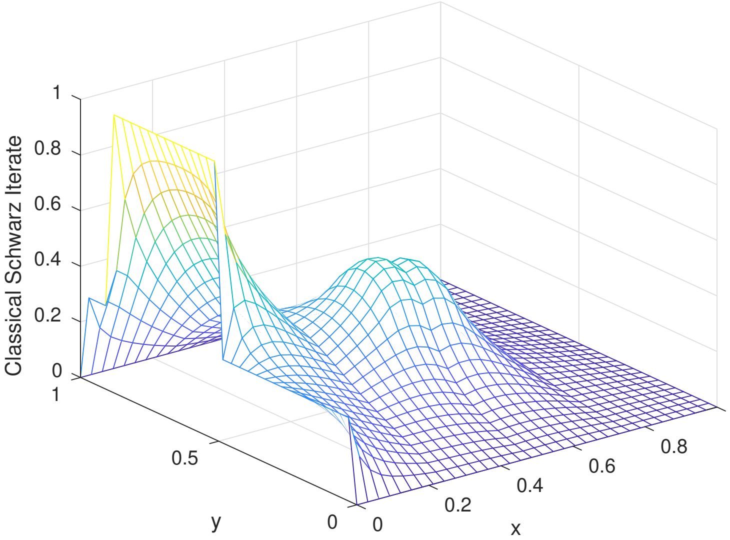
Classical Sweeping Schwarz Iteration 3



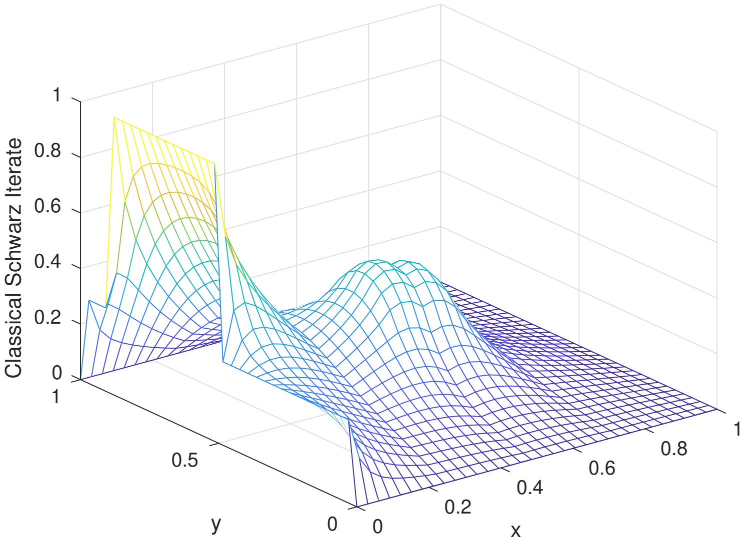
Classical Sweeping Schwarz Iteration 4



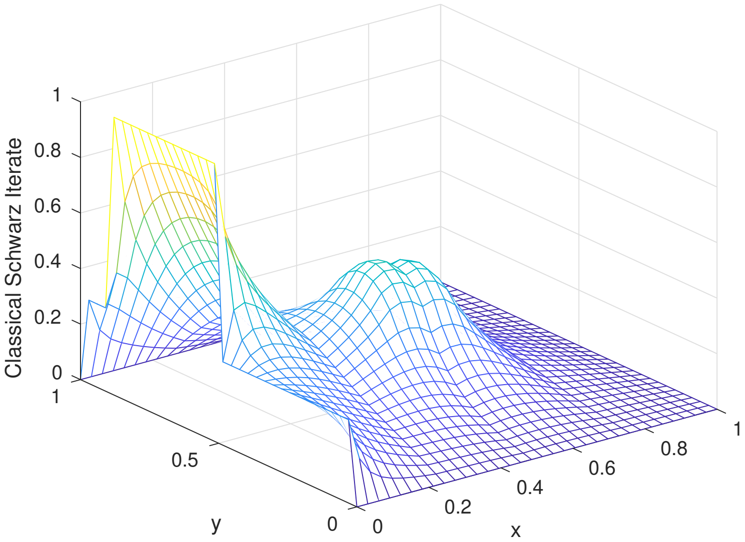
Classical Sweeping Schwarz Iteration 4



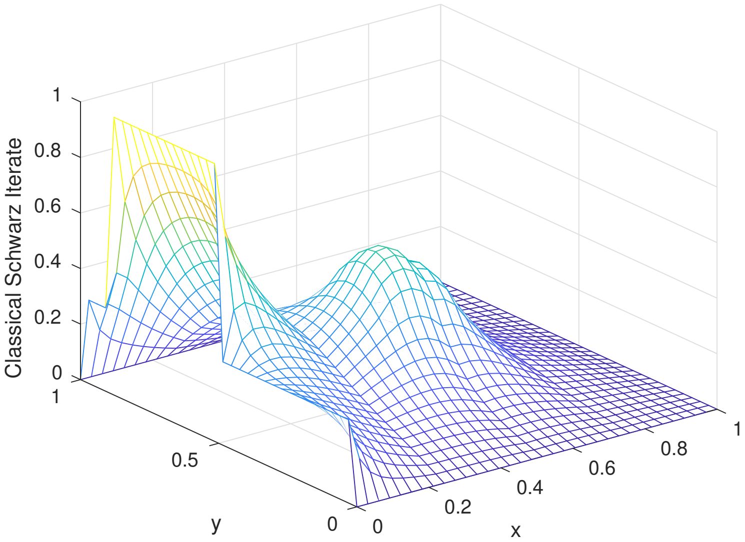
Classical Sweeping Schwarz Iteration 4



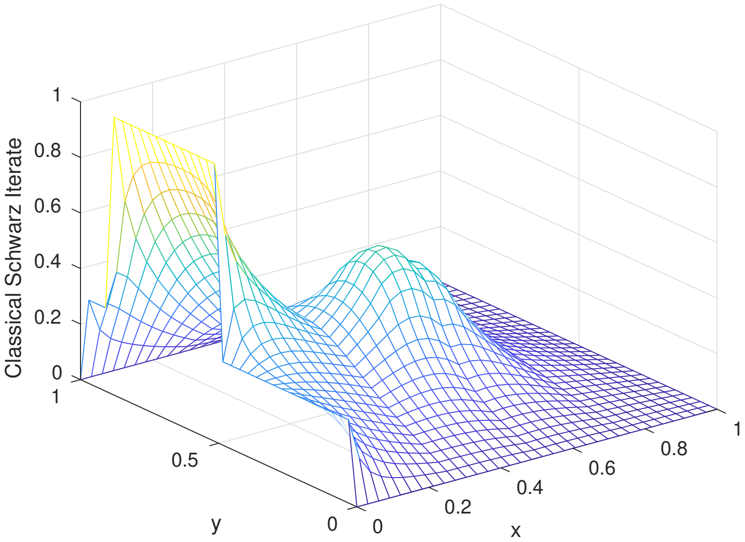
Classical Sweeping Schwarz Iteration 4



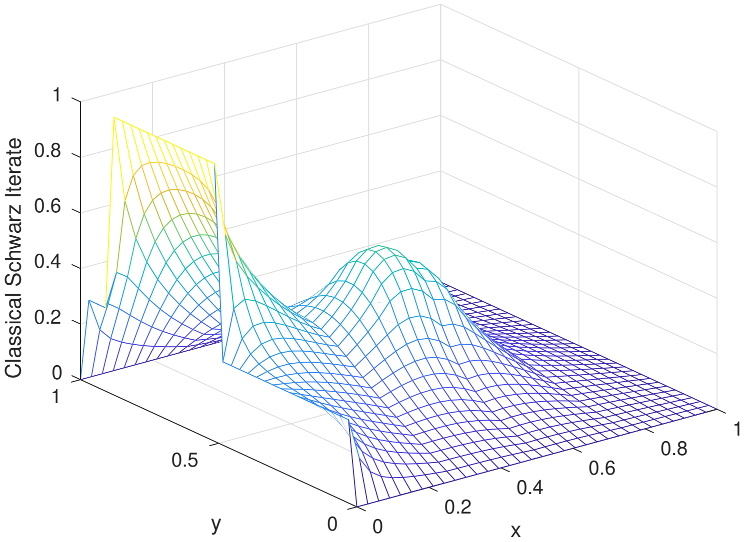
Classical Sweeping Schwarz Iteration 4



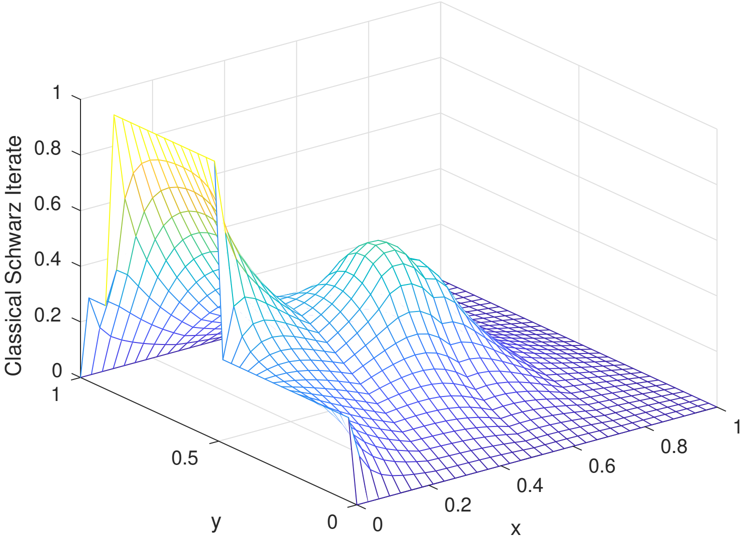
Classical Sweeping Schwarz Iteration 4



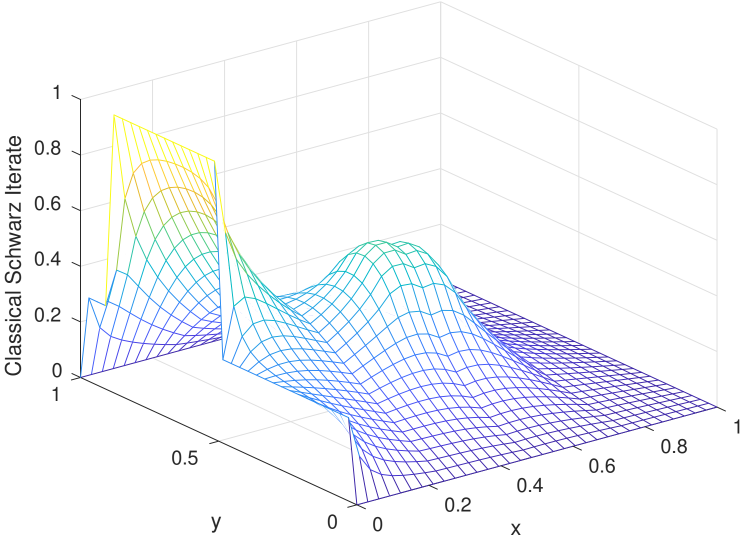
Classical Sweeping Schwarz Iteration 5



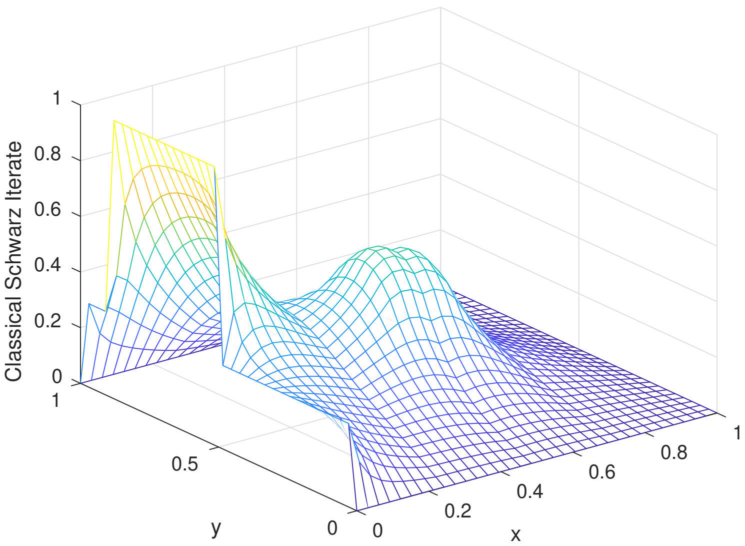
Classical Sweeping Schwarz Iteration 5



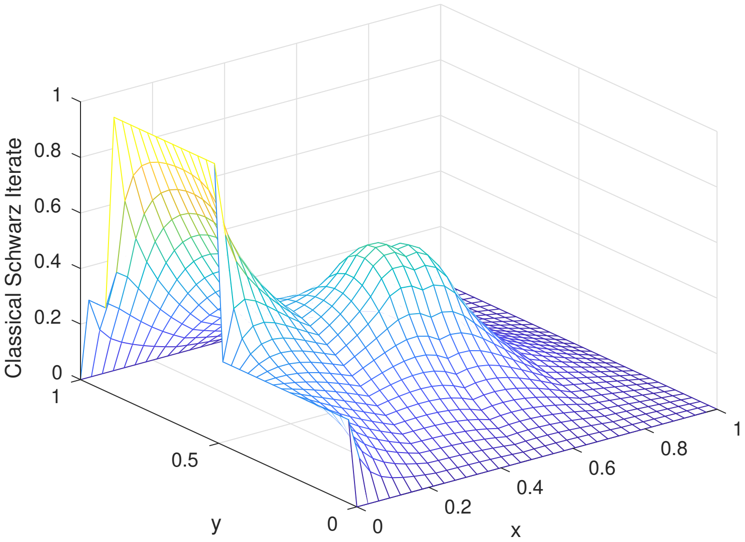
Classical Sweeping Schwarz Iteration 5



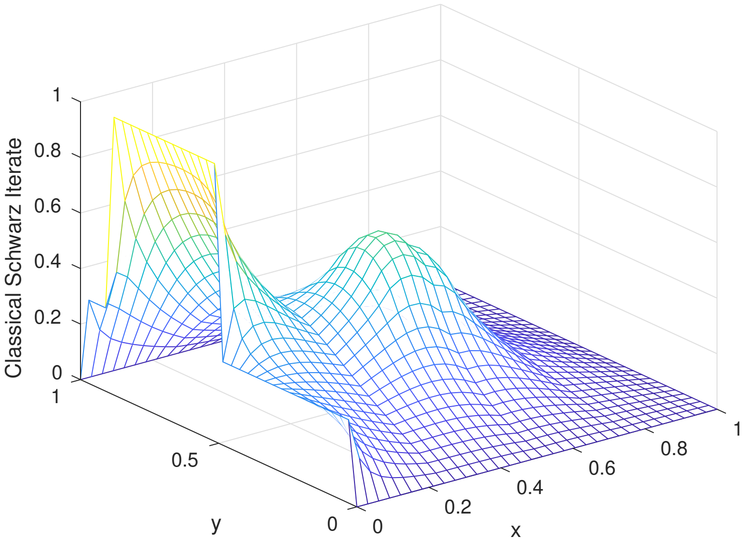
Classical Sweeping Schwarz Iteration 5



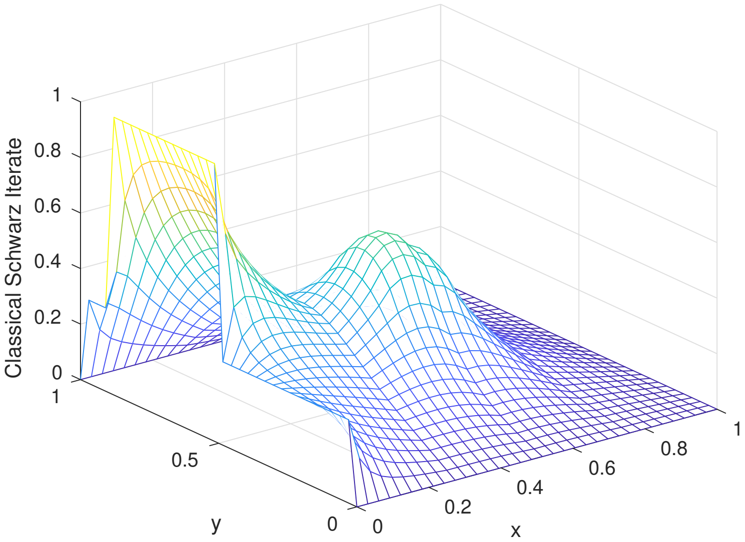
Classical Sweeping Schwarz Iteration 5



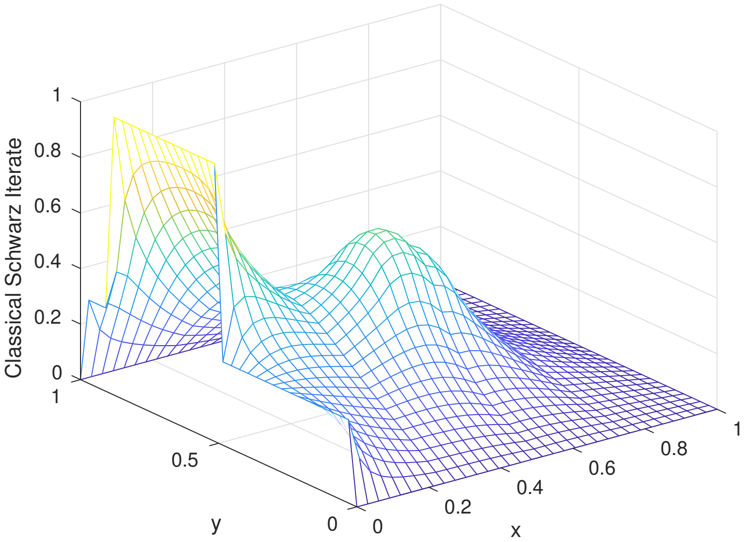
Classical Sweeping Schwarz Iteration 5



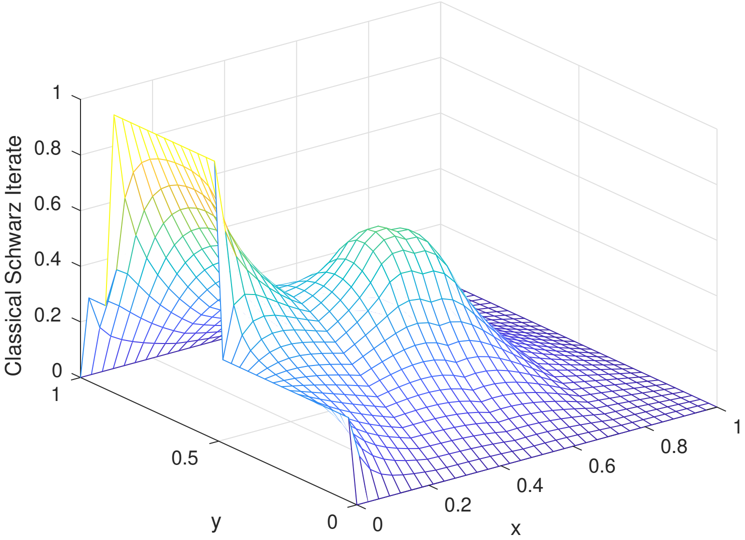
Classical Sweeping Schwarz Iteration 5



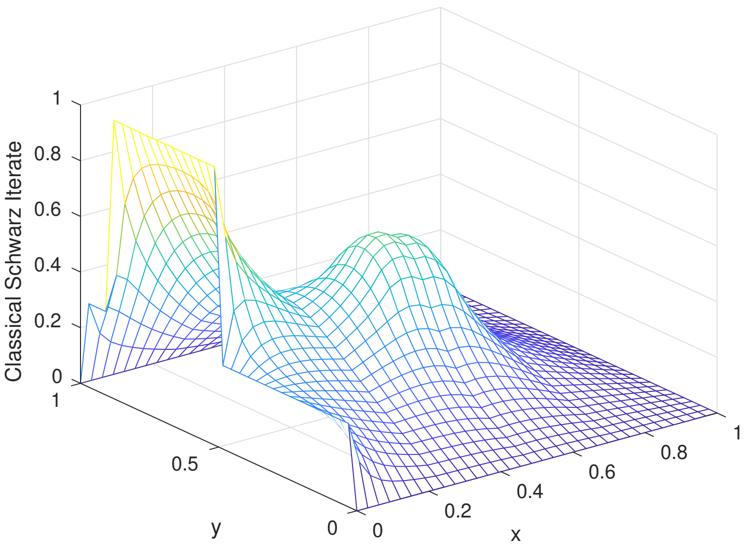
Classical Sweeping Schwarz Iteration 6



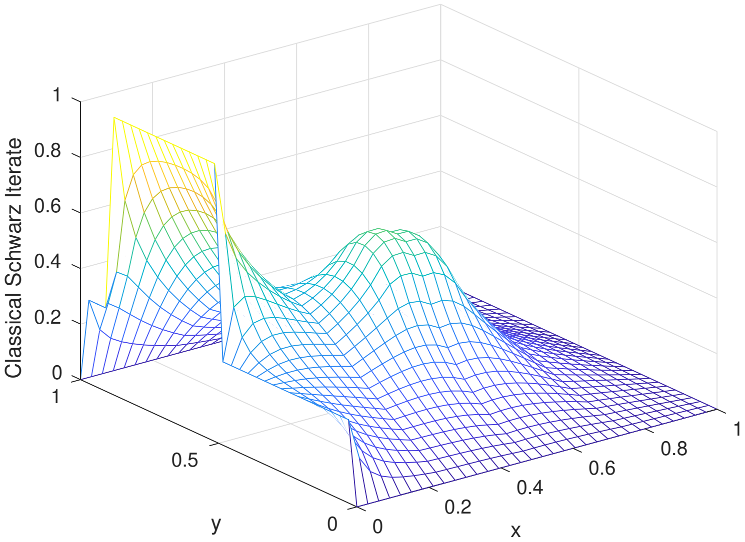
Classical Sweeping Schwarz Iteration 6



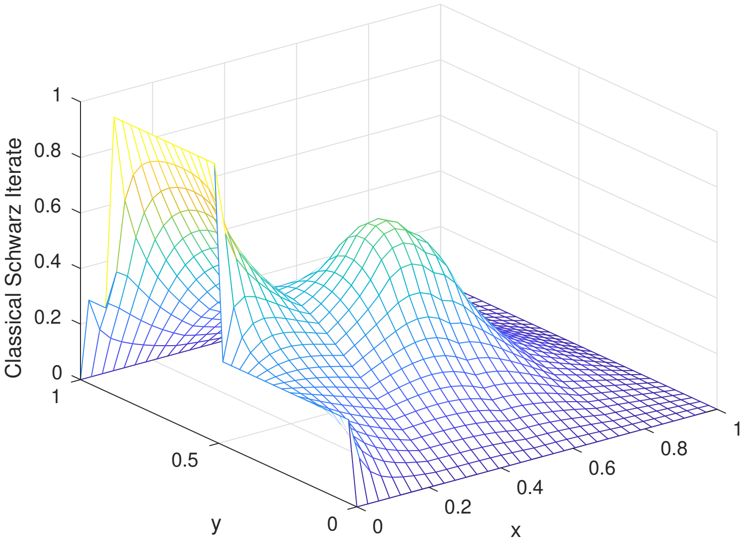
Classical Sweeping Schwarz Iteration 6



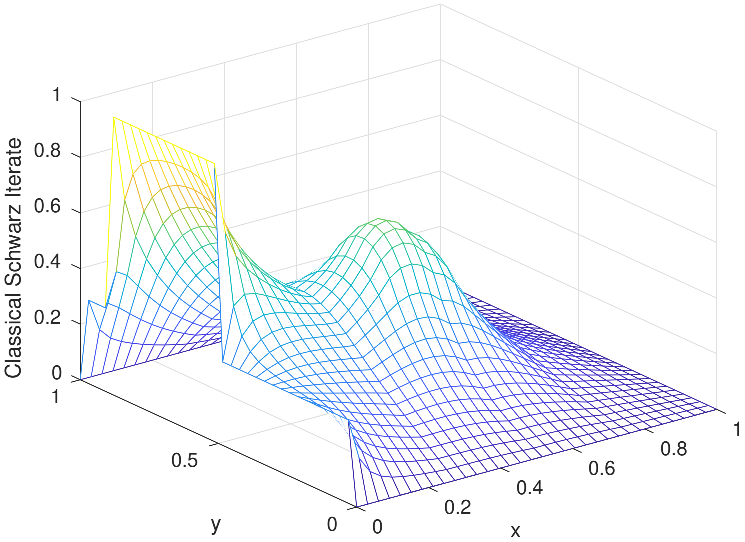
Classical Sweeping Schwarz Iteration 6



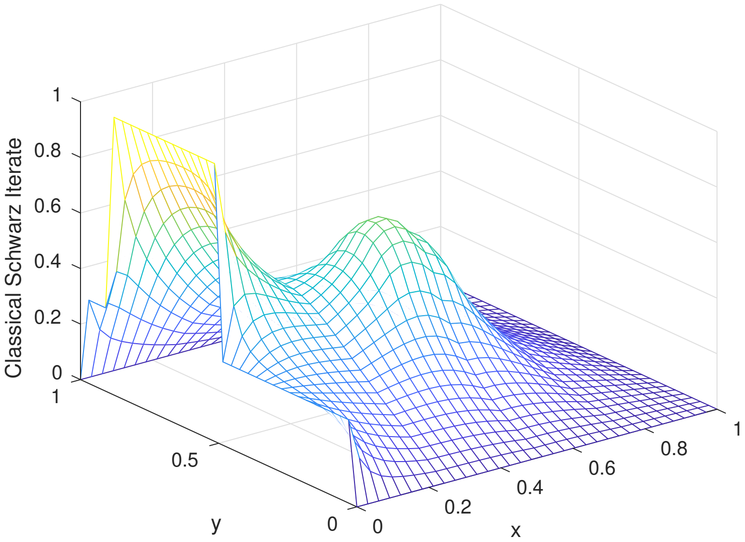
Classical Sweeping Schwarz Iteration 6



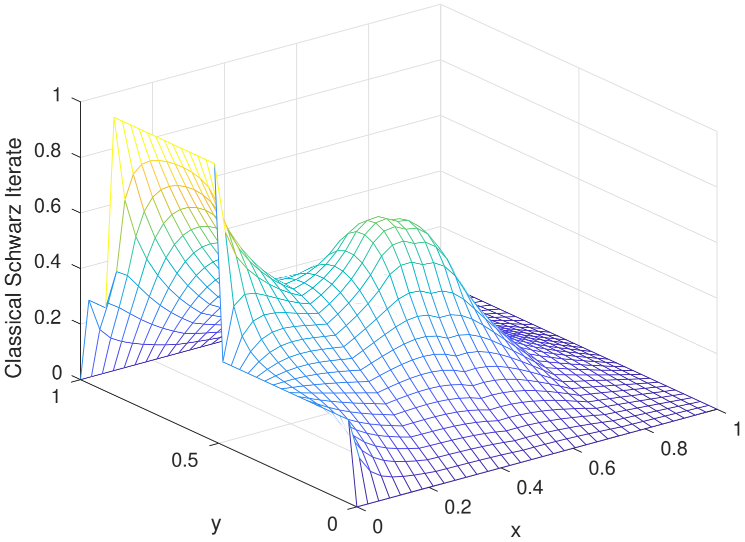
Classical Sweeping Schwarz Iteration 6



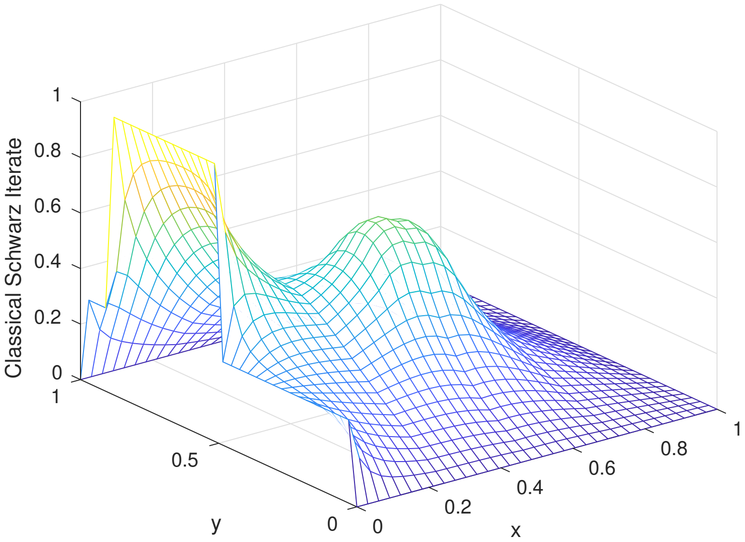
Classical Sweeping Schwarz Iteration 7



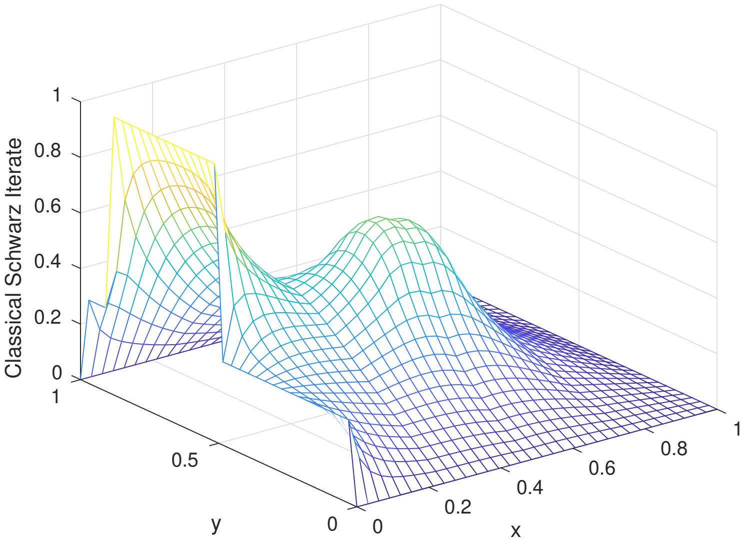
Classical Sweeping Schwarz Iteration 7



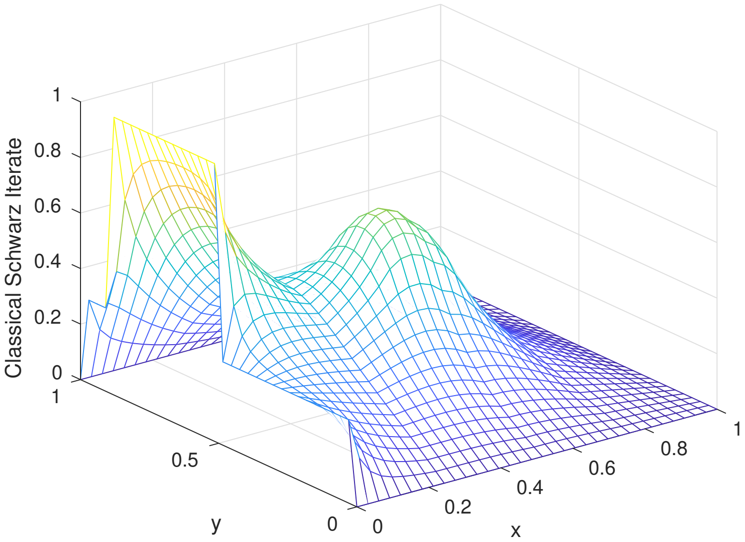
Classical Sweeping Schwarz Iteration 7



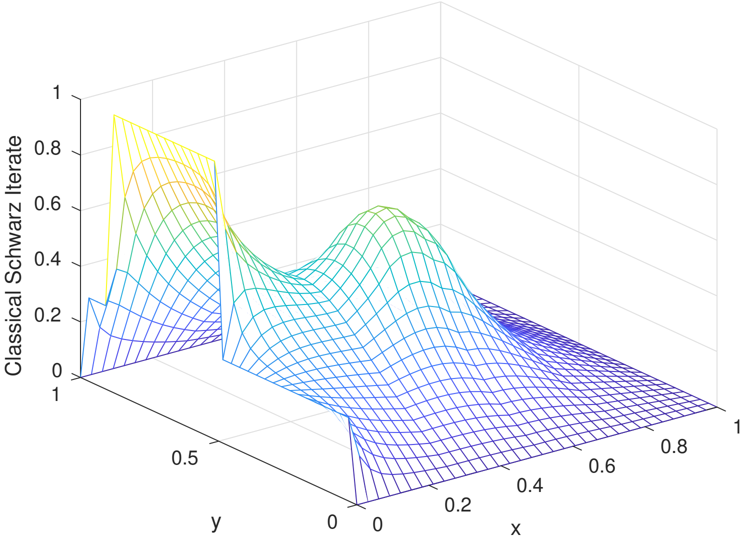
Classical Sweeping Schwarz Iteration 7



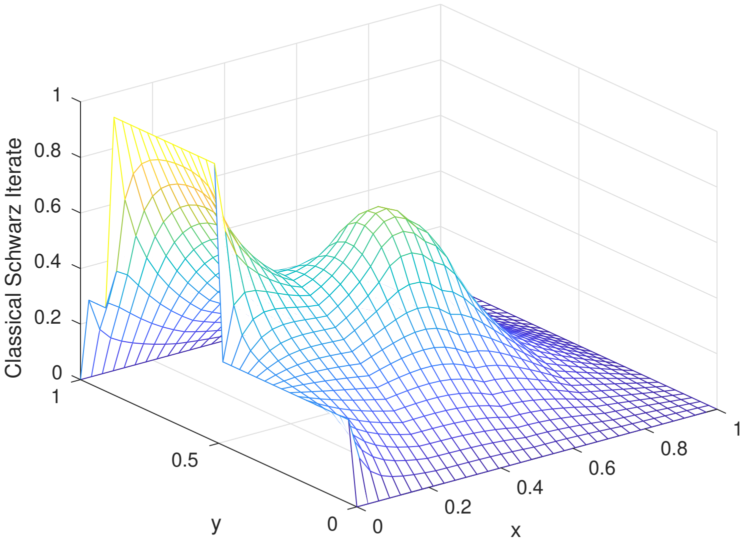
Classical Sweeping Schwarz Iteration 7



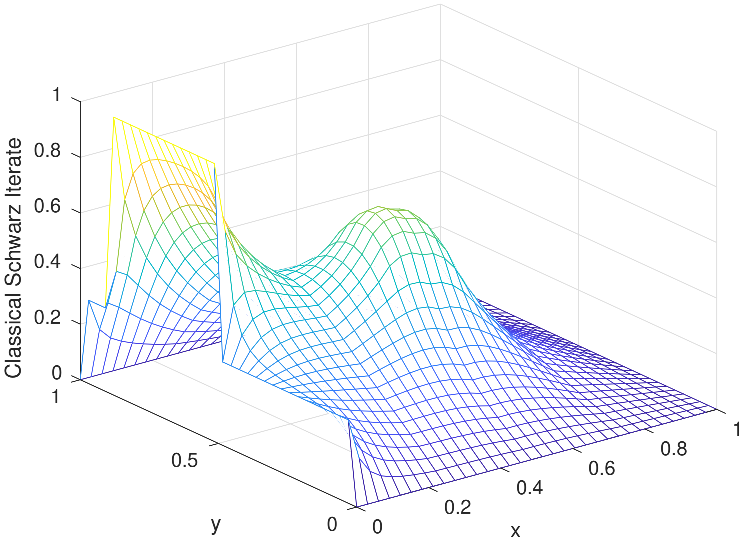
Classical Sweeping Schwarz Iteration 7



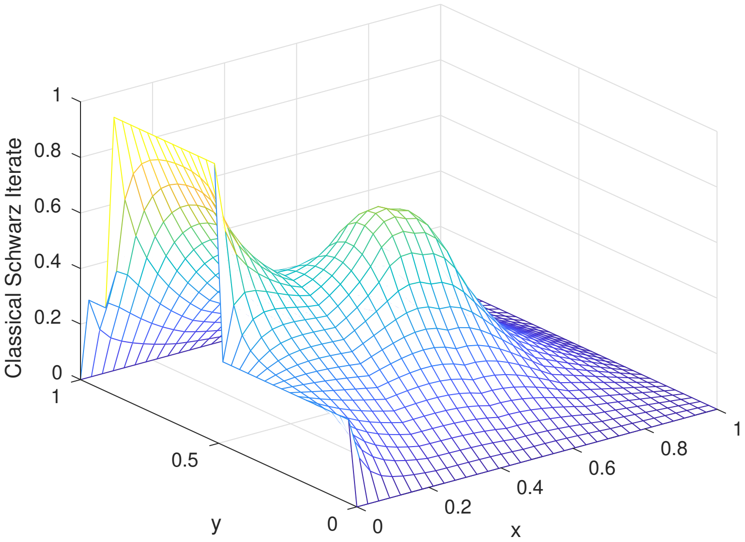
Classical Sweeping Schwarz Iteration 8



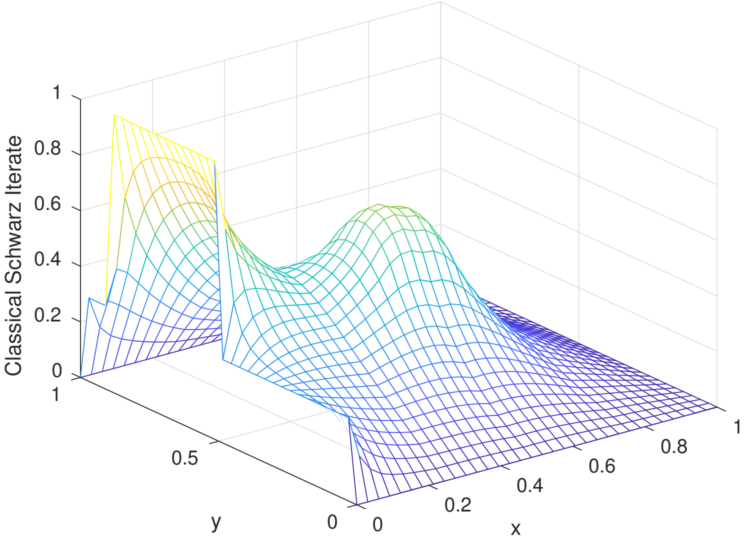
Classical Sweeping Schwarz Iteration 8



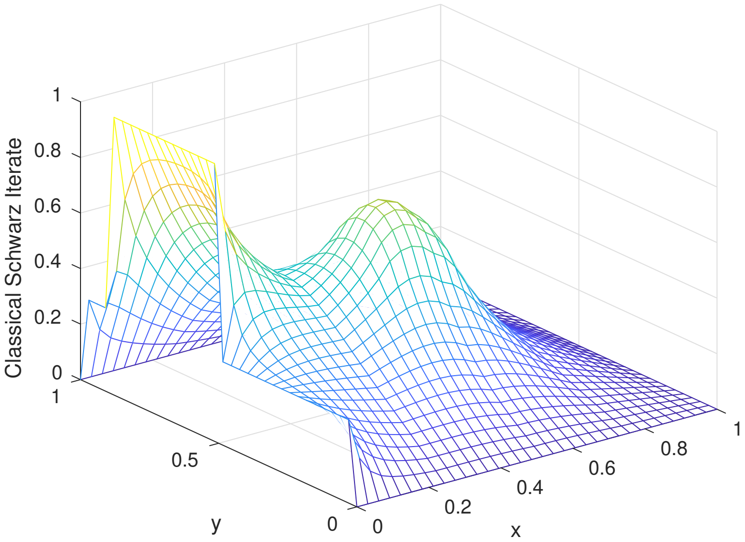
Classical Sweeping Schwarz Iteration 8



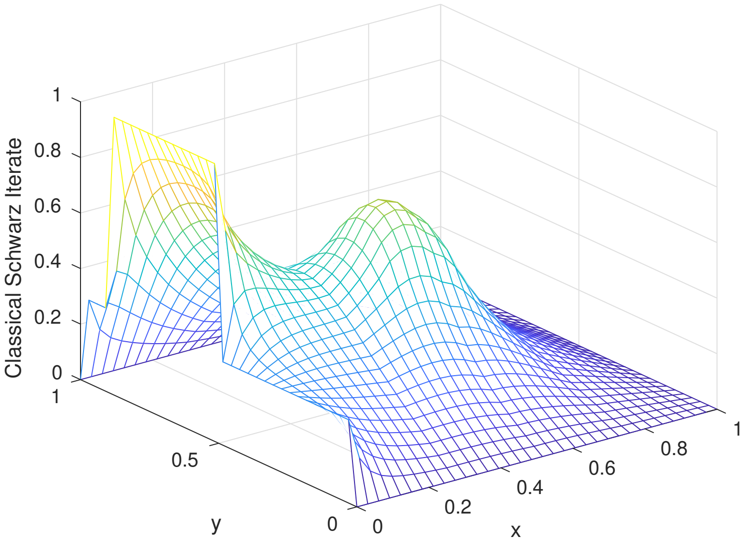
Classical Sweeping Schwarz Iteration 8



Classical Sweeping Schwarz Iteration 8



Classical Sweeping Schwarz Iteration 8



A Third Major Difference

In addition to the two differences

1. discrete MG versus continuous DD
2. Contraction estimate versus condition number estimate

there is a third major difference:

3. Coarse spaces

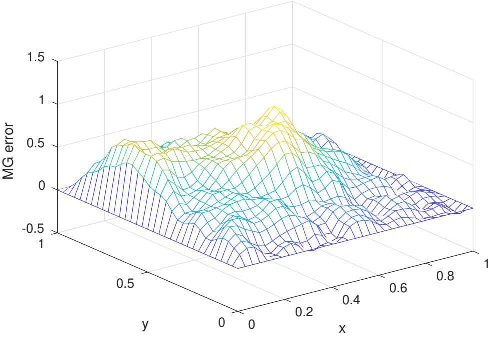
Coarse Correction Needed For MG

One uses the error equation: for an approximation $\boldsymbol{u}^n$

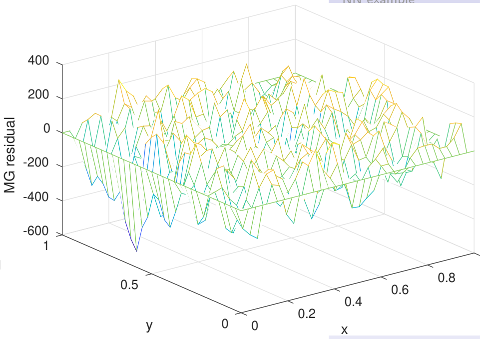
$$A\boldsymbol{u} = A(\boldsymbol{u} - \boldsymbol{u}^n) + A\boldsymbol{u}^n = A\boldsymbol{e}^n + A\boldsymbol{u}^n = \boldsymbol{f}$$

$$\Longrightarrow A\boldsymbol{e}^n = \boldsymbol{f} - A\boldsymbol{u}^n = \boldsymbol{r}^n$$

With random $\boldsymbol{u}^0$ after 3 Jacobi smoothing steps with $\omega = 2/3$



error



residual

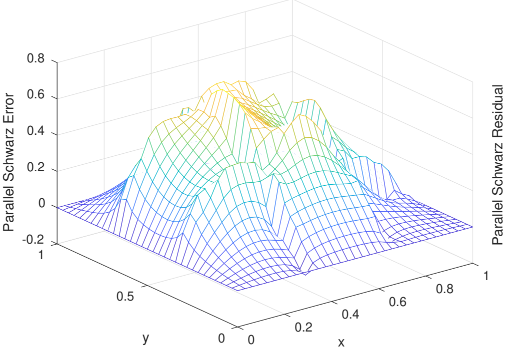
Coarse Correction Needed For DD ?

Classical approach is to use a coarse mesh like in multigrid:

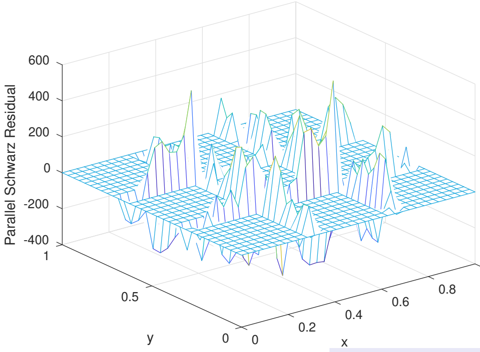
Toselli and Widlund (2005):

> *"The subspace V_0 is usually related to a coarse problem, often built on a coarse mesh"*

With random $\boldsymbol{u}^0$ after one parallel Schwarz iteration



error



residual

Error equation for DD

For any DD method and problem $\mathcal{L}u = f$ in Ω, the subdomain solutions u_j^n satisfy

$$\mathcal{L}u_j^n = f, \quad \text{in } \Omega_j.$$

Hence the error $e_j^n(x) := u(x) - u_j^n(x)$ in Ω_j satisfies

$$\mathcal{L}e_j^n = 0 \quad \text{in } \Omega_j.$$

Optimal coarse space for DD (G, Halpern, Santugini 2018): piece-wise harmonic functions on Ω_j.
Jumps in the Dirichlet and Neumann traces between subdomain solutions:

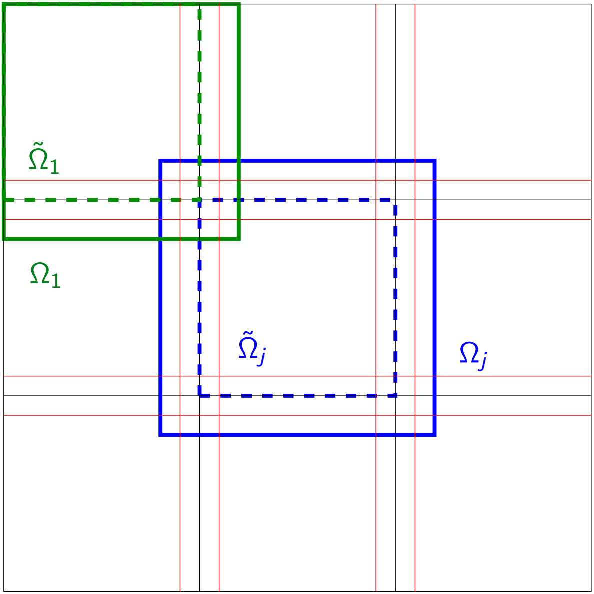
$$g_{ji}^n := u_j^n - u_i^n, \quad h_{ji}^n := \partial_{n_j} u_j^n + \partial_{n_i} u_i^n, \quad \text{on } \Gamma_{ji}.$$

Then the error satisfies the transmission problem

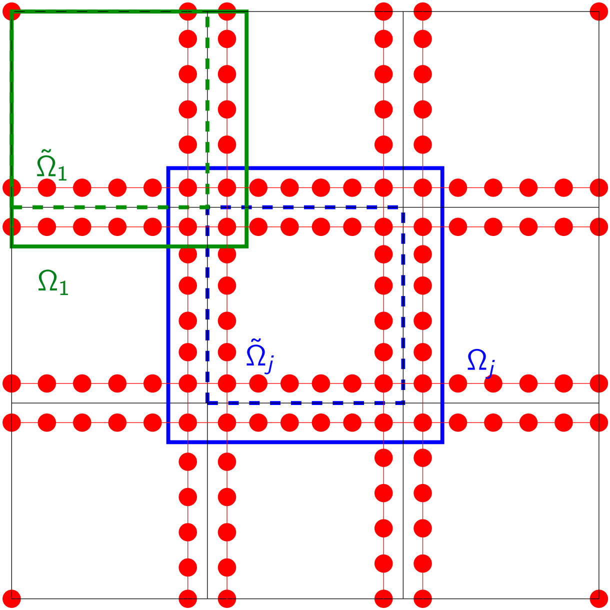
$$\begin{aligned} \mathcal{L}e_j^n &= 0 && \text{in } \tilde{\Omega}_j, \\ e_j^n(x) - e_i^n(x) &= g_{ji}^n(x) && \text{on } \Gamma_{ji}, \\ \partial_{n_j} e_j^n(x) + \partial_{n_i} e_i^n(x) &= h_{ji}^n(x) && \text{on } \Gamma_{ji}. \end{aligned}$$

$\Longrightarrow$ solving this, any DD converges after one iteration!

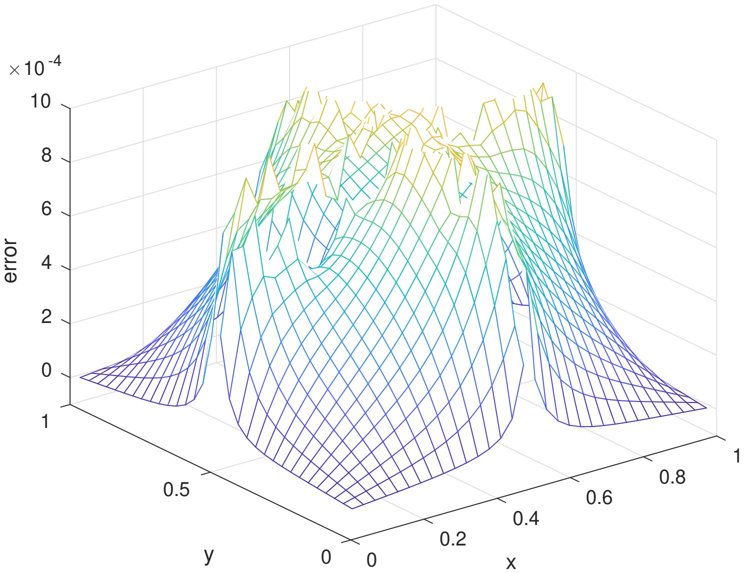
Optimal Coarse Space in Two Dimensions



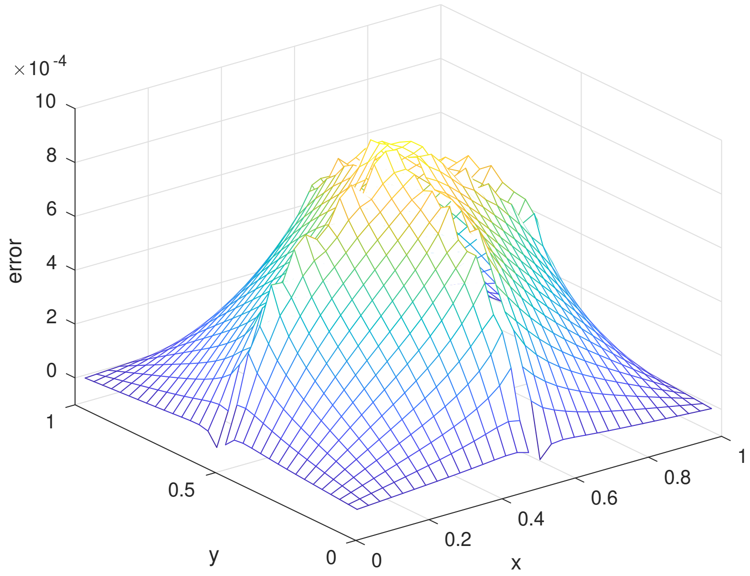
Optimal Coarse Space in Two Dimensions



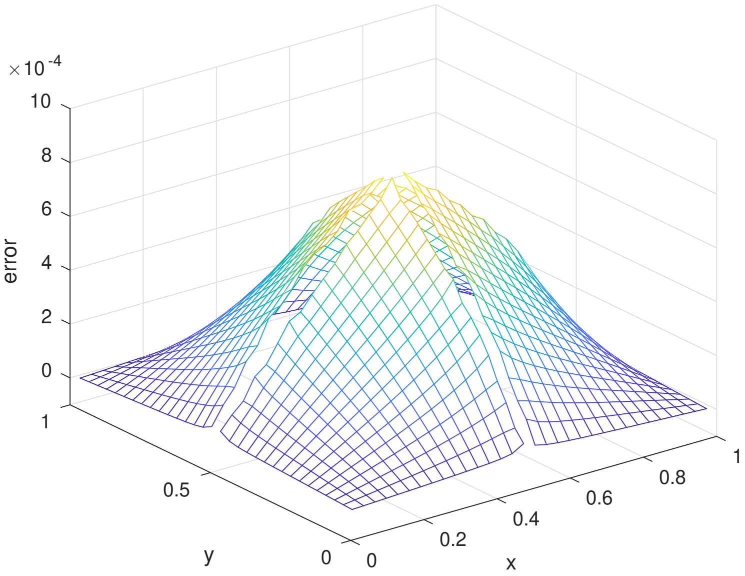
Optimized Schwarz error without coarse



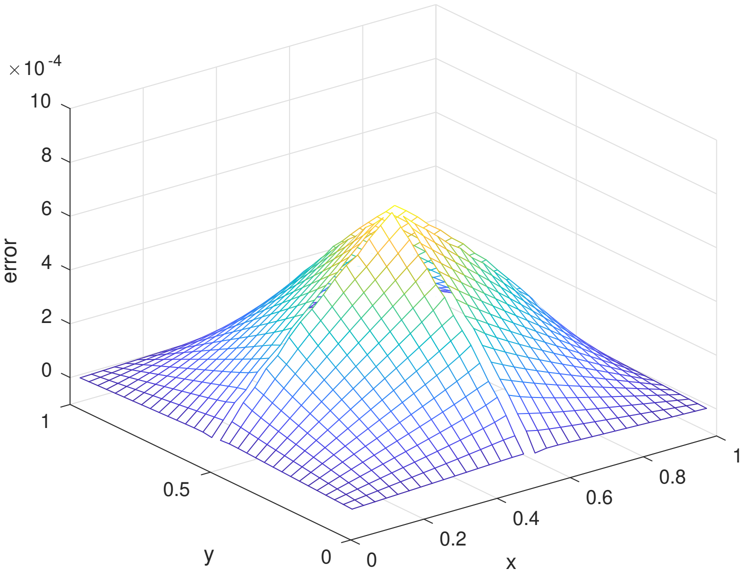
Optimized Schwarz error without coarse



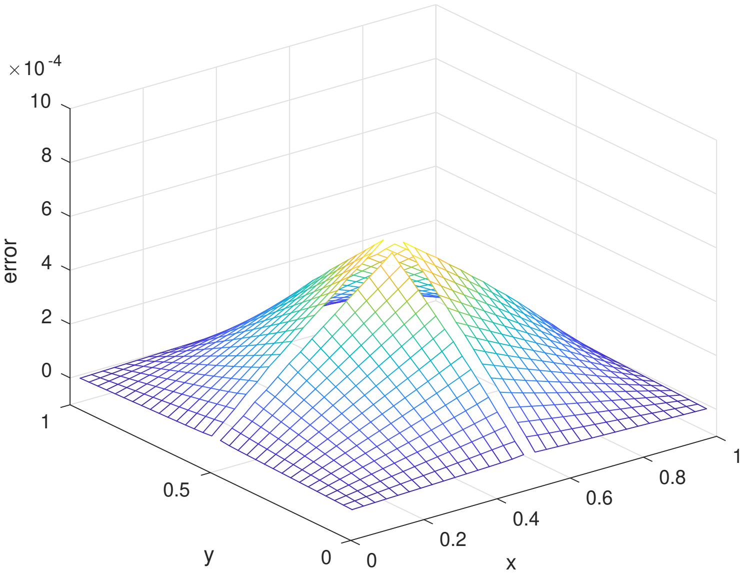
Optimized Schwarz error without coarse



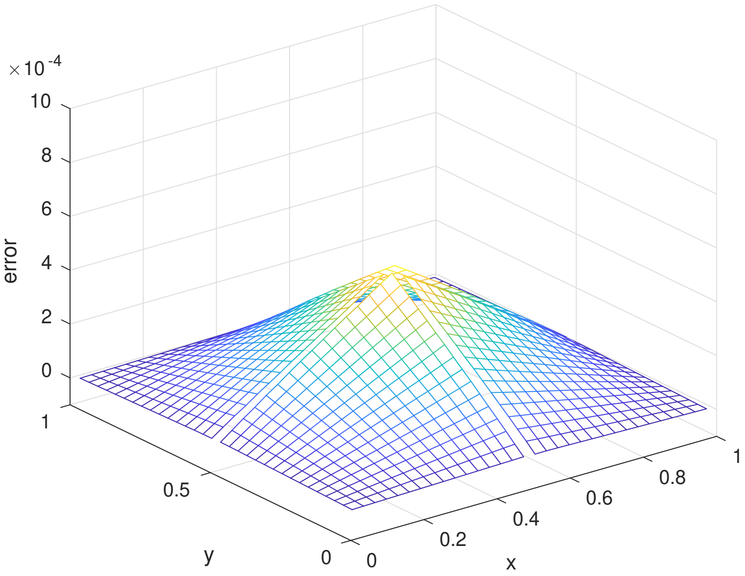
Optimized Schwarz error without coarse



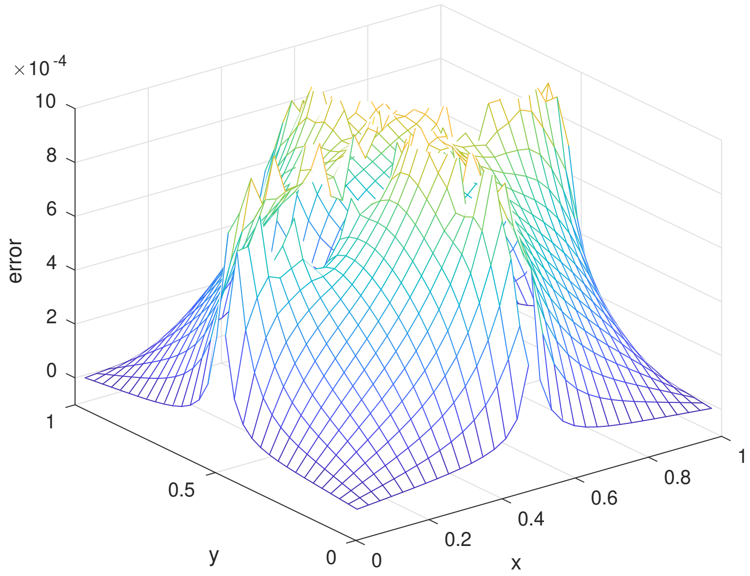
Optimized Schwarz error without coarse



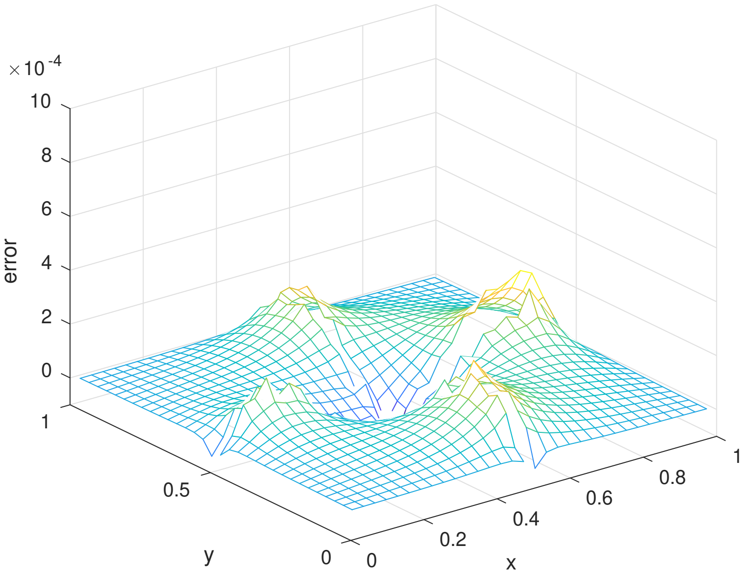
Optimized Schwarz error without coarse



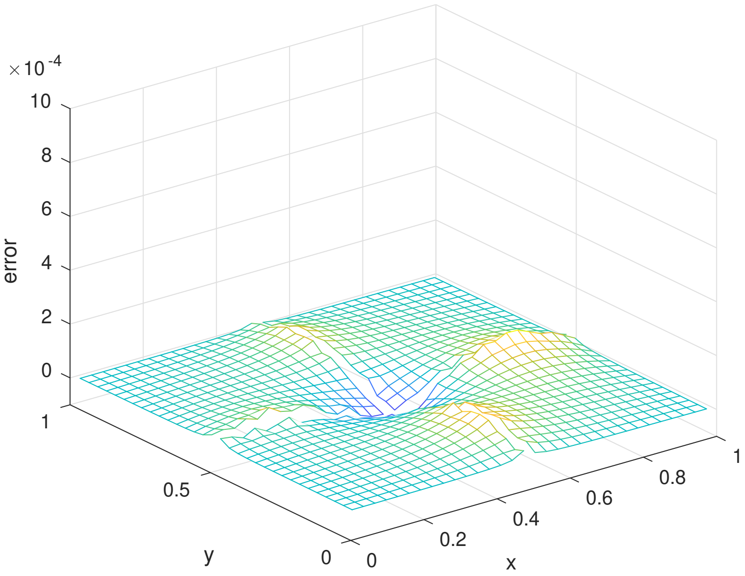
Optimized Schwarz error with optimized coarse



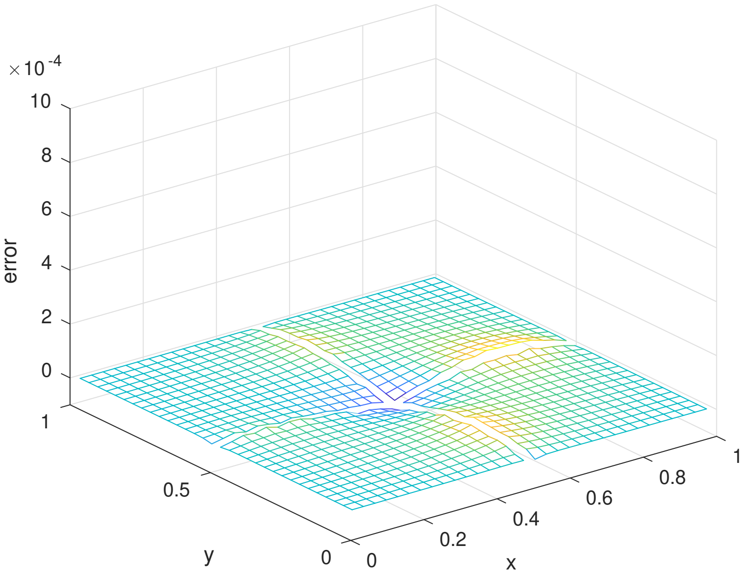
Optimized Schwarz error with optimized coarse



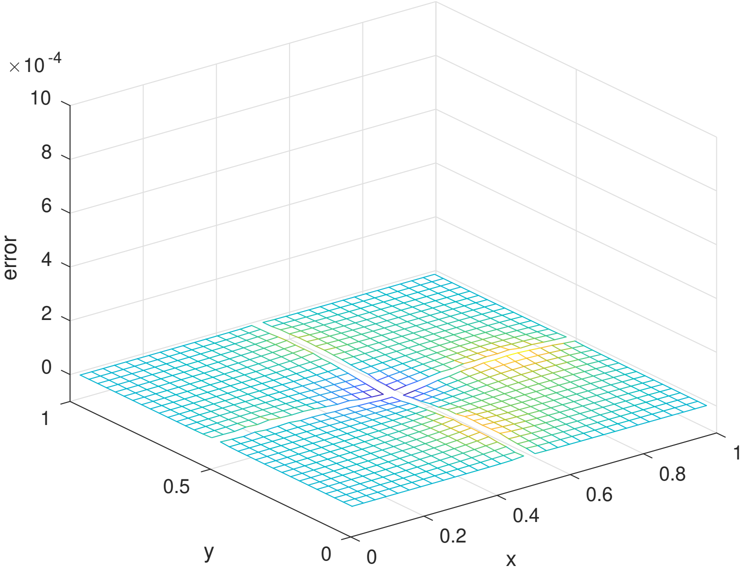
Optimized Schwarz error with optimized coarse



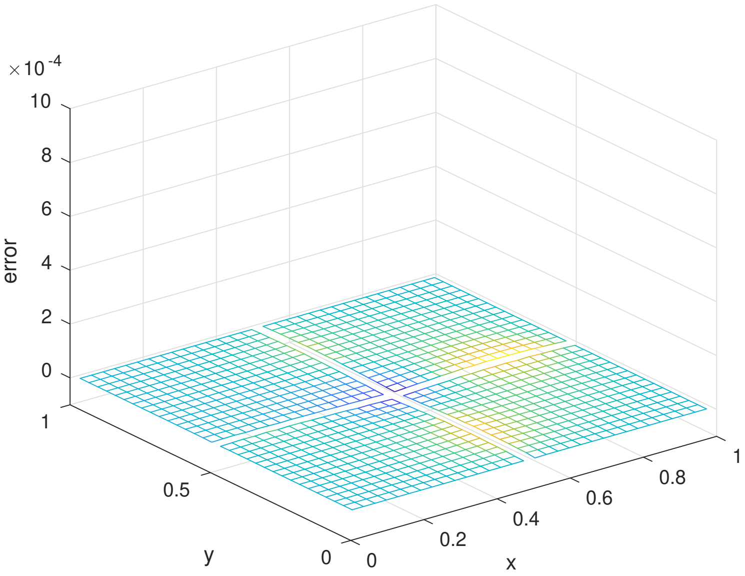
Optimized Schwarz error with optimized coarse



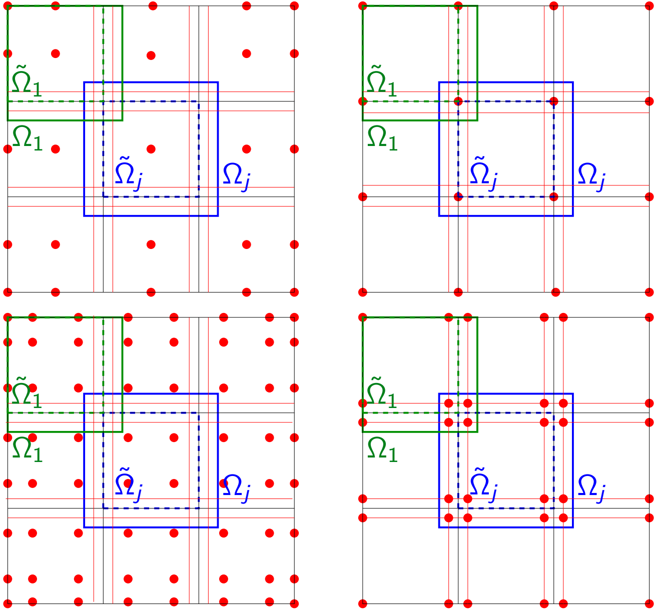
Optimized Schwarz error with optimized coarse



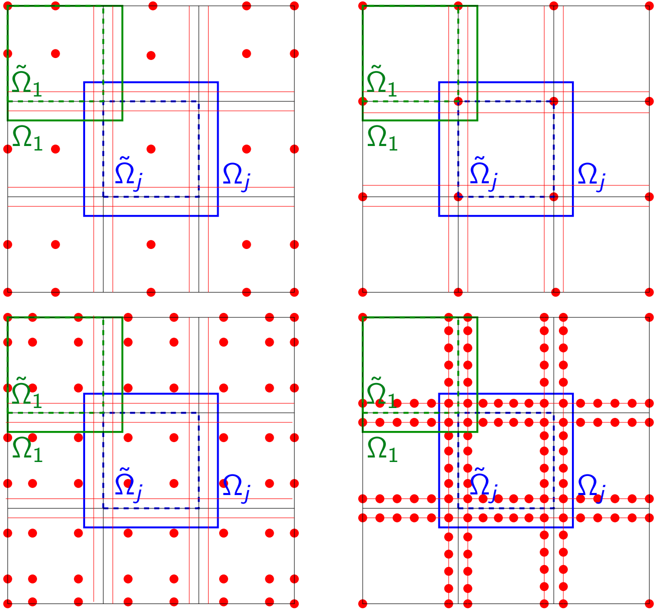
Optimized Schwarz error with optimized coarse



Comparison of Various Q1 Coarse Spaces

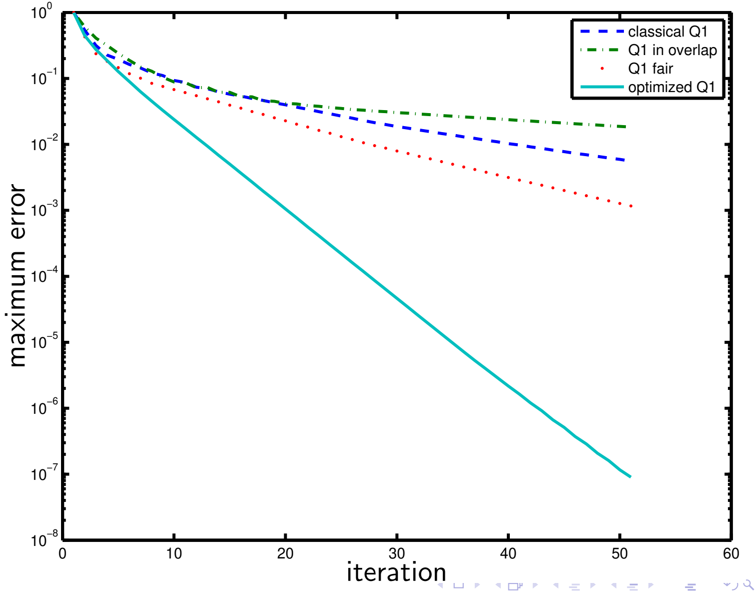


Only one Approximates the Optimal One!

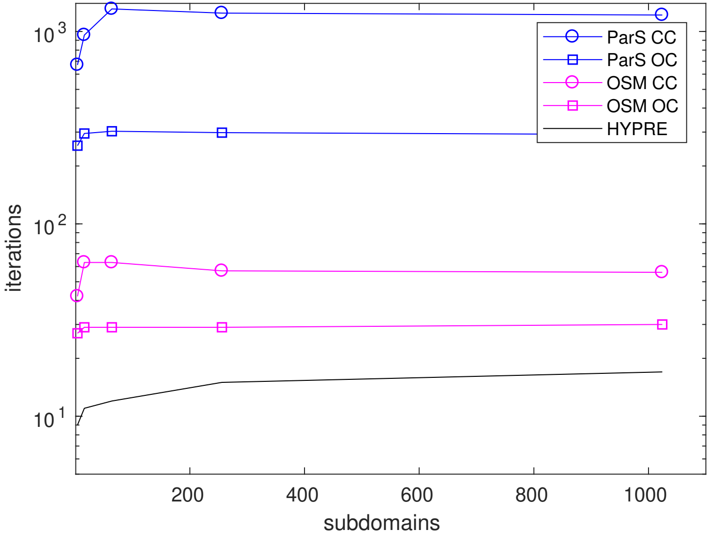


Parallel Schwarz with Various Coarse Spaces

ParS, 16×16 subdomains, 256×256 grid-points, h overlap

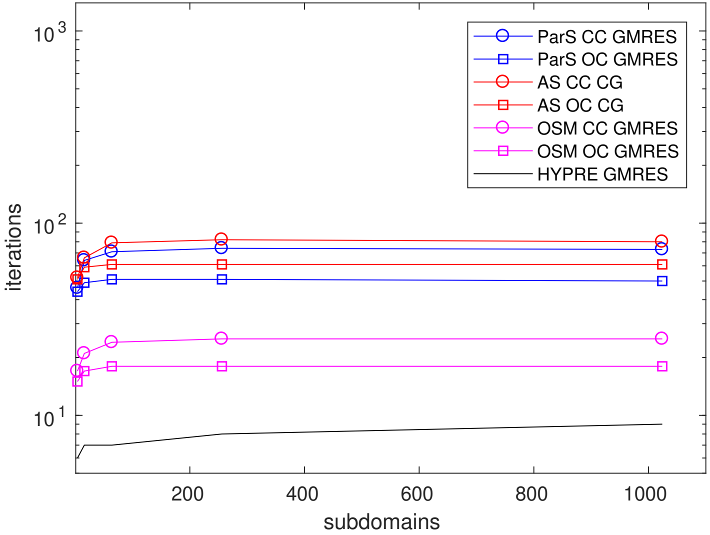


Comparison of 2-level Schwarz Methods



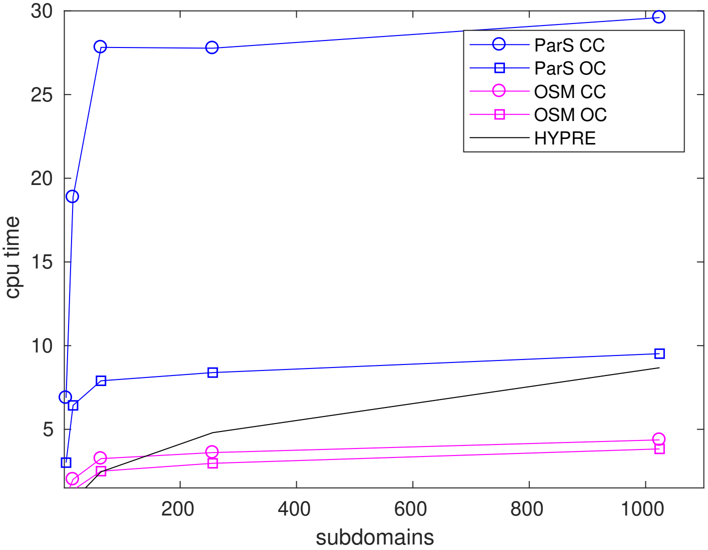
Work load $h/H = 0.004$ (256×256 local subdomain mesh)
In PETSc by Serge van Criekingen (IDRIS)

With Krylov Acceleration



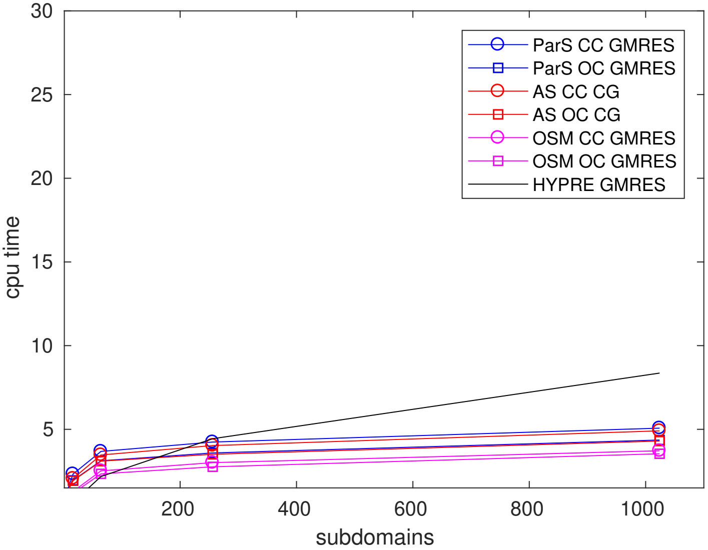
Work load $h/H = 0.004$ (256×256 local subdomain mesh)
In PETSc by Serge van Criekingen (IDRIS)

CPU time comparison of the Methods



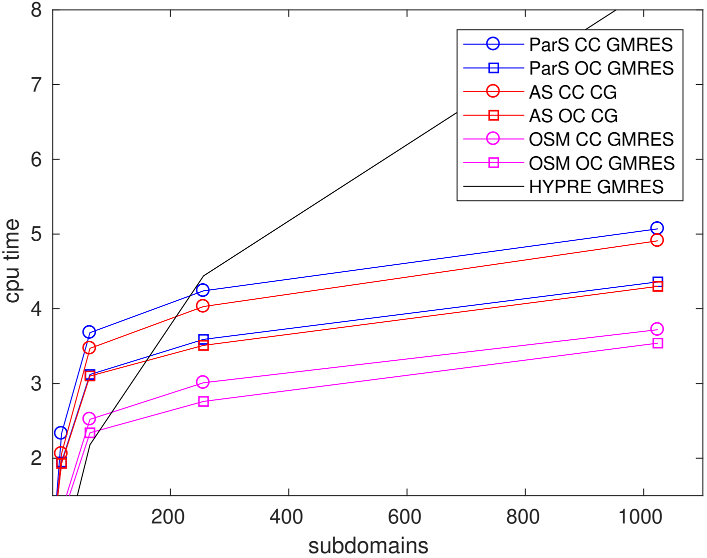
Work load $h/H = 0.004$ (256×256 local subdomain mesh)
In PETSc by Serge van Criekingen (IDRIS)

With Krylov Acceleration



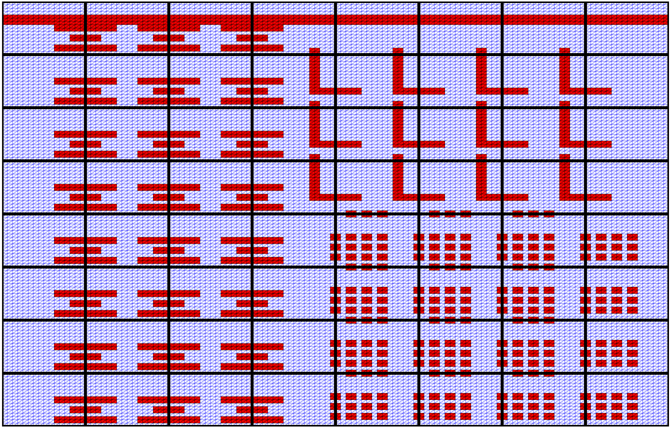
Work load $h/H = 0.004$ (256×256 local subdomain mesh)
In PETSc by Serge van Criekingen (IDRIS)

Zoom



Work load $h/H = 0.004$ (256×256 local subdomain mesh)
In PETSc by Serge van Criekingen (IDRIS)

Coarse Spaces can Fix DD Problems: SHEM



α	MS	SHEM_1	SHEM_2	SHEM_3	SHEM_4	SHEM_a
10^0	21	16	14	13	13	21
10^2	122	71	49	19	17	25
10^4	367	256	130	20	20	25
10^6	610	454	221	20	20	25
dim.	49	161	273	385	497	233

Summary of Differences

1. MG discrete, DD continuous
2. Contraction estimate versus condition number estimate
3. Coarse spaces

References and preprints can be found on my web page at

`www.unige.ch/~gander/`

Also two new books $\Longrightarrow$