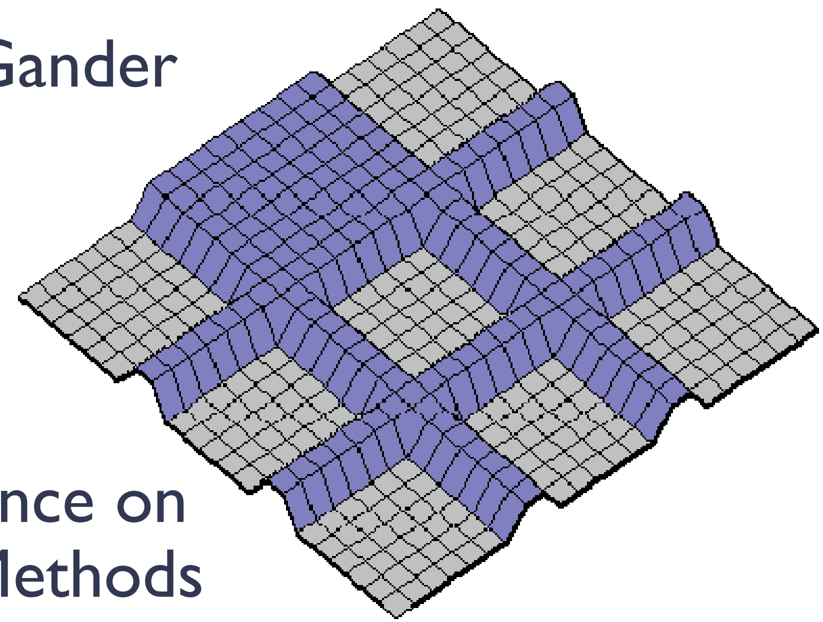
Optimal Interface Conditions for an Arbitrary Decomposition into Subdomains

Felix Kwok and Martin J. Gander

Section de mathématiques
Université de Genève



19th International Conference on Domain Decomposition Methods

Zhangjiajie, China, August 18th, 2009

Classical Schwarz Method

- Schwarz (1869), Lions (1988):

$$-\Delta u_j^{k+1} = f \qquad \text{on } \Omega_j$$

$$u_j^{k+1} = g \qquad \text{on } \partial\Omega \cap \overline{\Omega}_j$$

$$u_j^{k+1} = \quad u_i^k \quad \text{on } \Gamma_{ij}$$

Optimal Schwarz Methods

- Change boundary conditions:

$$-\Delta u_j^{k+1} = f \qquad \text{on } \Omega_j$$

$$u_j^{k+1} = g \qquad \text{on } \partial\Omega \cap \overline{\Omega}_j$$

$$B_{ij} u_j^{k+1} = B_{ij} u_i^k \quad \text{on } \Gamma_{ij}$$

- B_{ij} = linear operators *acting on u* along Γ_{ij}
- B_{ij} can be:
  - *Local:* differential operators (compact stencil), e.g. Dirichlet, Neumann, Robin, etc.
  - *Nonlocal:* convolutions (dense matrix blocks), e.g. Steklov-Poincaré, Dirichlet-to-Neumann, etc.

Optimal Schwarz Methods

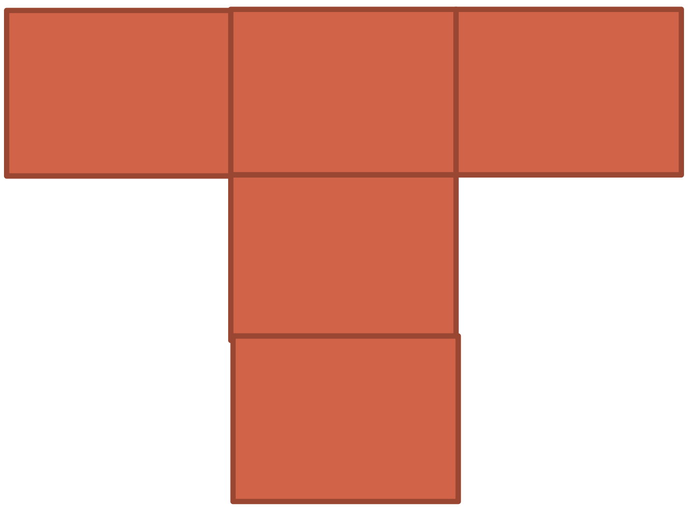
- Optimal operator for convergence is generally nonlocal:
  - Optimal means $\rho = 0$, or convergence in a finite number of iterations
  - Decomposition into strips : Use

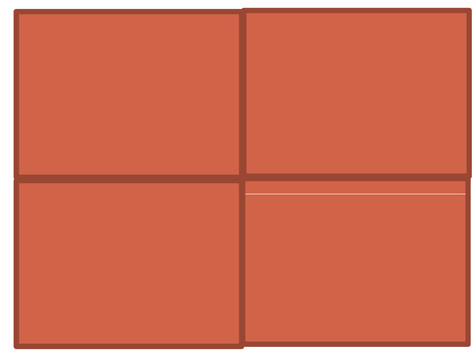
    $$\frac{\partial}{\partial \vec{n}_i} - \Lambda_i$$

    where Λ_i is the Dirichlet-to-Neumann operator (Nataf et al., 1994)
  - Corresponds to Schur complements in the discrete case

Optimal Schwarz Method

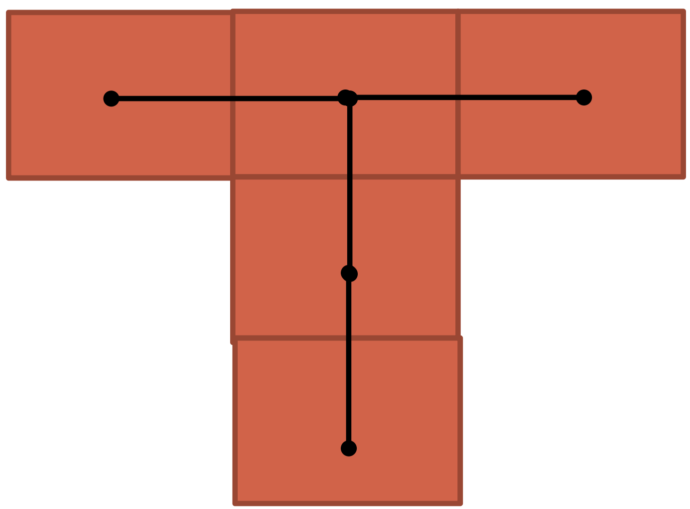
- Optimal Schwarz methods exist when the decomposition has no cycles (Nier 1995)



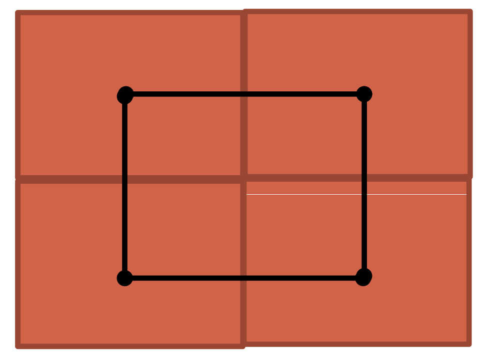


Optimal Schwarz Method

- Optimal Schwarz methods exist when the decomposition has no cycles (Nier 1995)



No Cycle



Cycle

Optimal Schwarz Method

- Conjecture : no optimal method when cycles are present, but
- Does there exist an iteration by subdomains that converges in a finite number of iterations if we are allowed to communicate more than just boundary data?

Schur complement

- For any subdomain Ω_j, we can rewrite the linear system (after permutation) as

$$\begin{bmatrix} A_j & B_j \\ C_j & D_j \end{bmatrix} \begin{pmatrix} u_j \\ u^0 \end{pmatrix} = \begin{pmatrix} f_j \\ f_j^0 \end{pmatrix}$$

Inside Ω_j

Outside Ω_j

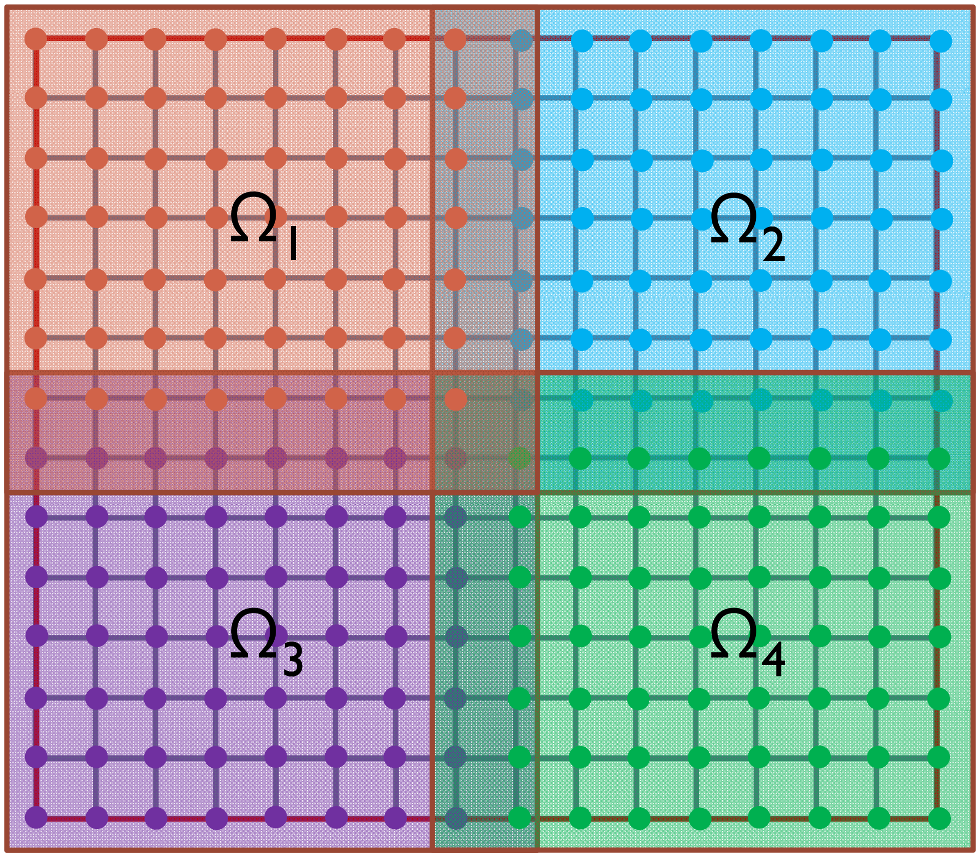
which is equivalent to

$$\left(A_j - B_j D_j^{-1} C_j\right) u_j = f_j - B_j D_j^{-1} f_j^0$$

which can be solved in parallel for each *j*.

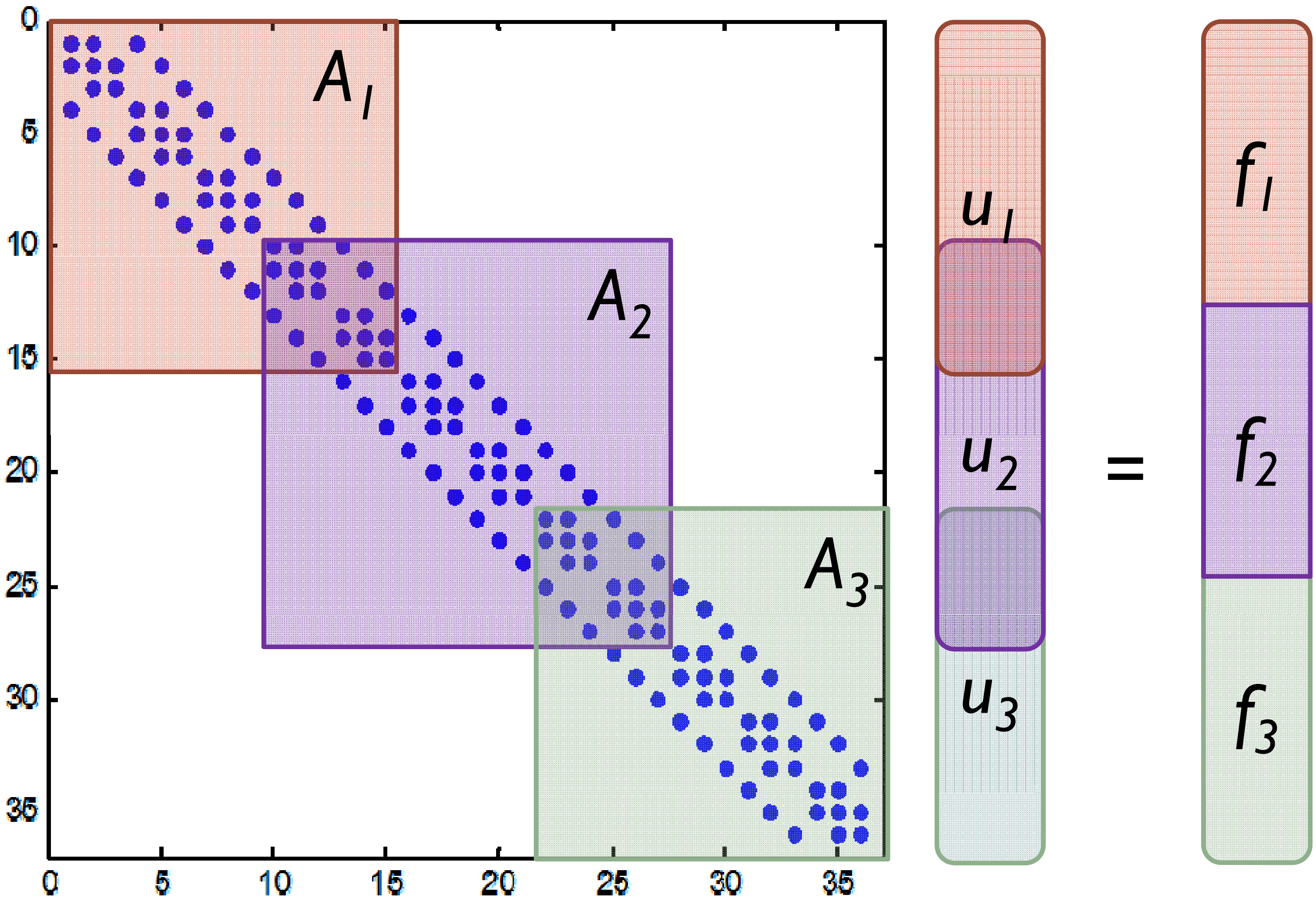
- *How to reconstruct f_j^o (RHS outside Ω_j) using solutions from other subdomains?*

Sufficient Overlap

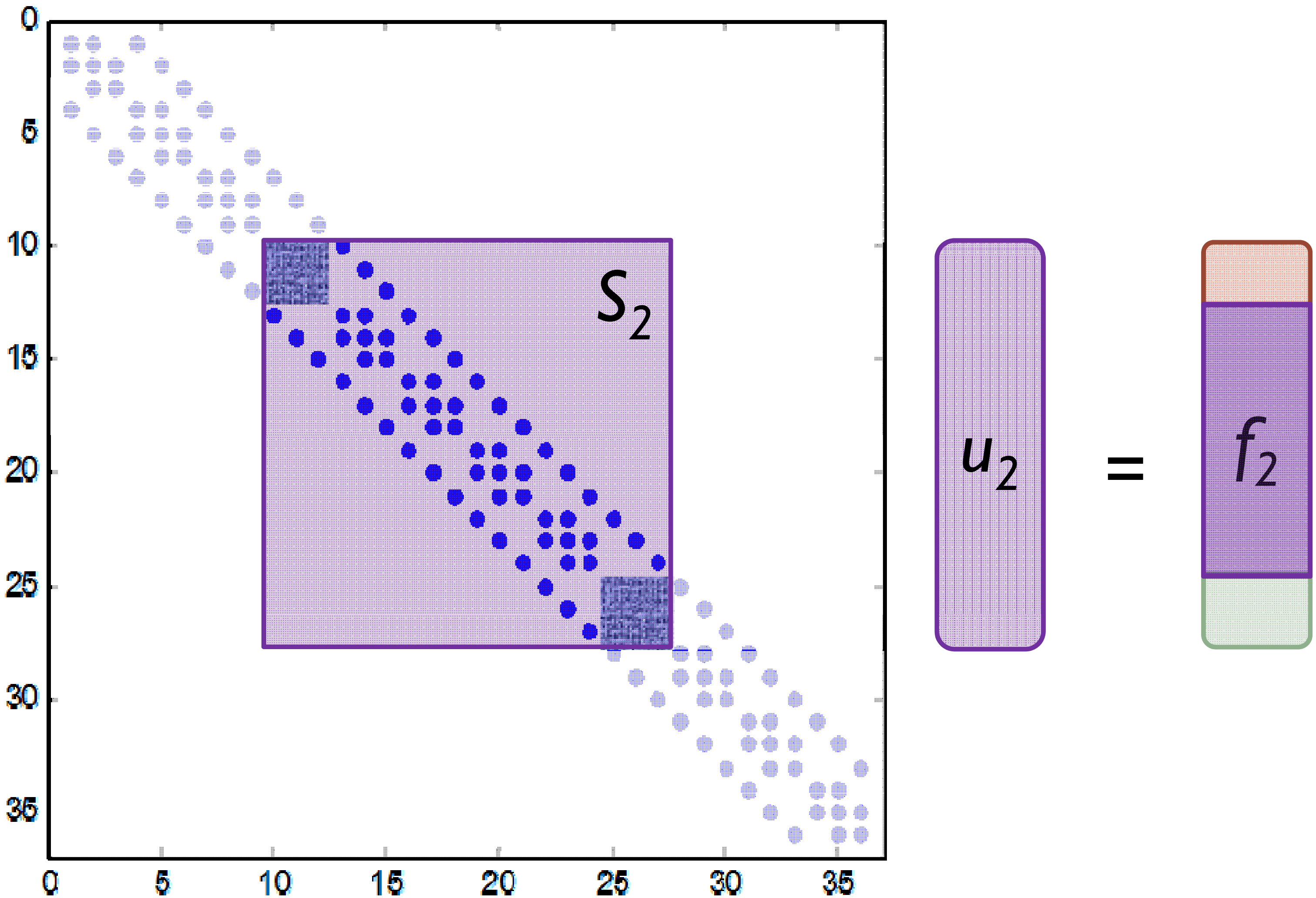


- Assume each grid point lies in the *interior* of at least one subdomain (away from interface)

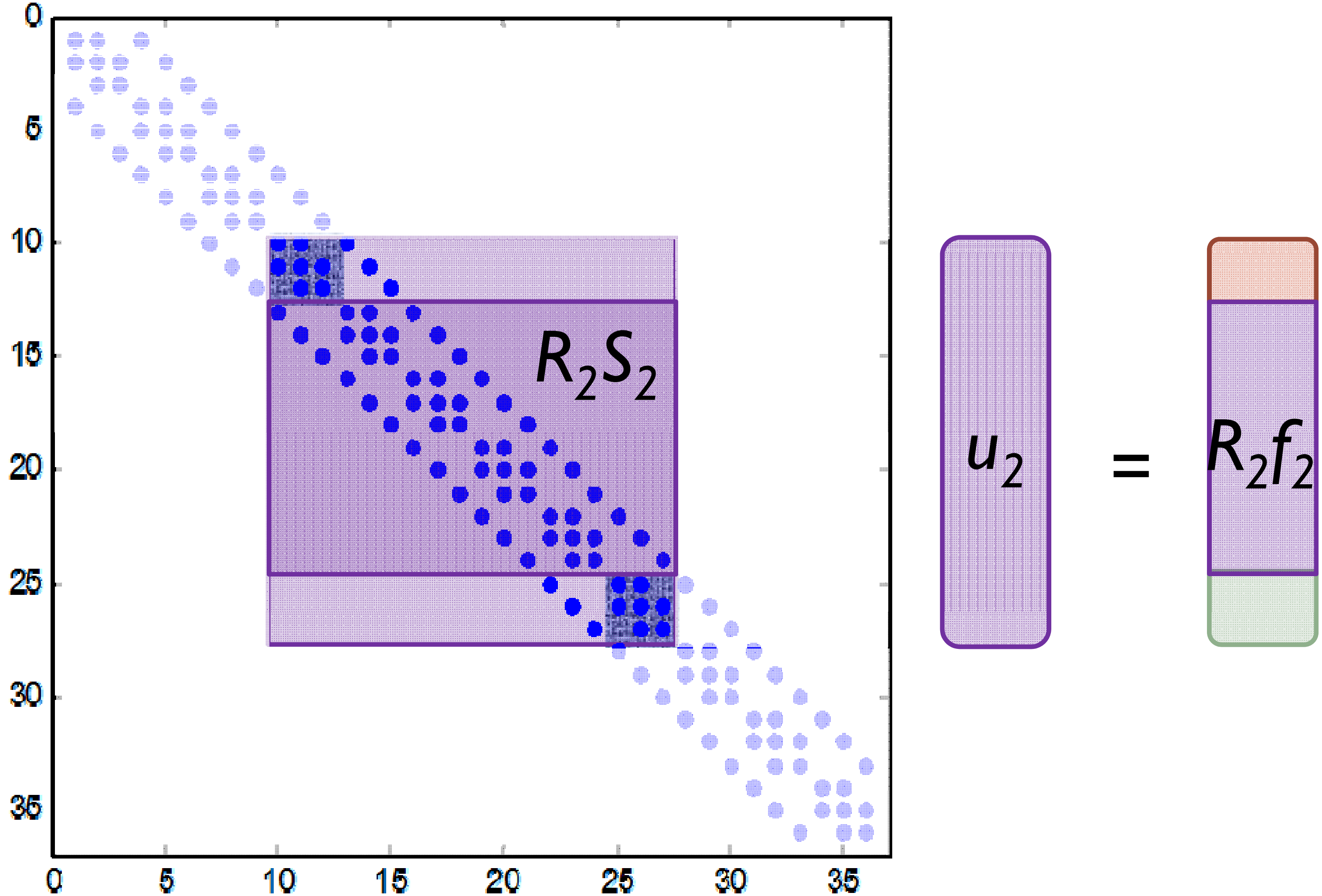
Extracting f_j^o



Extracting f_j^o



Extracting f_j^o



Parallel Algorithm – Version I

$$\left(A_j - B_j D_j^{-1} C_j\right) u_j = f_j - B_j D_j^{-1} f_j^0$$

$$\Downarrow$$

$$\left(A_j - B_j D_j^{-1} C_j\right) u_j^{k+1} = f_j - \sum_{i \neq j} B_j D_j^{-1} \begin{bmatrix} 0 \\ R_i A_i \\ 0 \end{bmatrix} u_i^k$$

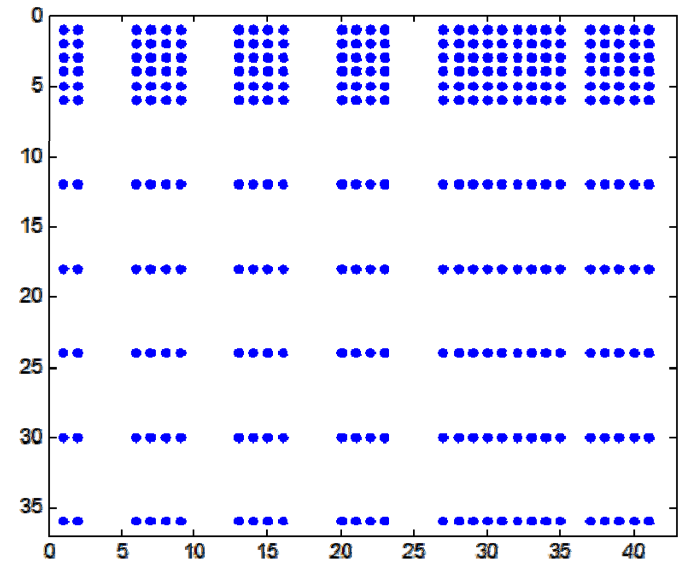
- u_j^{k+1} will yield the exact solution as long as each u_i^k satisfies $R_i A_i u_i = R_i f_i$ $(i \neq j)$
- *Algorithm converges in two steps!*

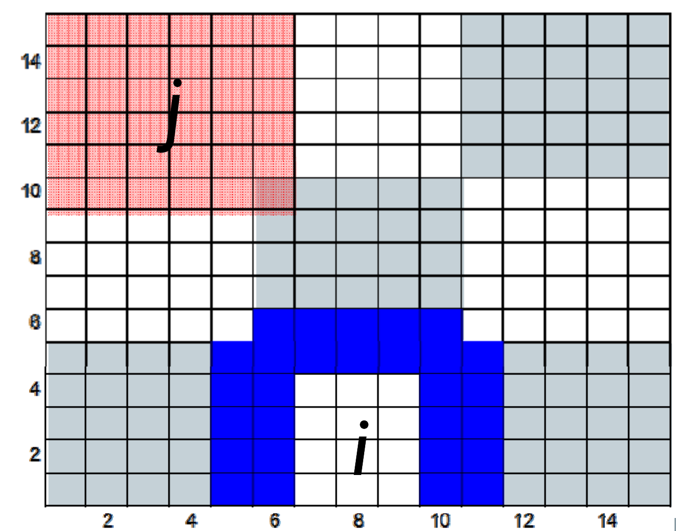
Reducing Communication

- Observation:

$$B_j D_j^{-1} \begin{bmatrix} 0 \\ R_i A_i \\ 0 \end{bmatrix}$$

  has a very specific sparsity pattern
- Column is nonzero only at interfaces between subdomains
- *Values of interior nodes not needed!*





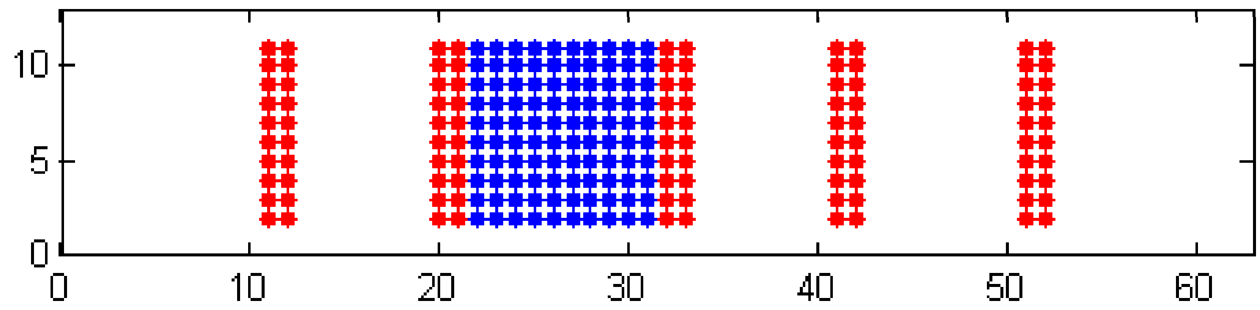
Parallel Algorithm – Version II

$$\left(A_j - B_j D_j^{-1} C_j\right) u_j^{k+1} = f_j - \sum_{i \neq j} B_j D_j^{-1} \begin{bmatrix} 0 \\ R_i A_i P_{ji}^T \\ 0 \end{bmatrix} P_{ji} u_i^k$$

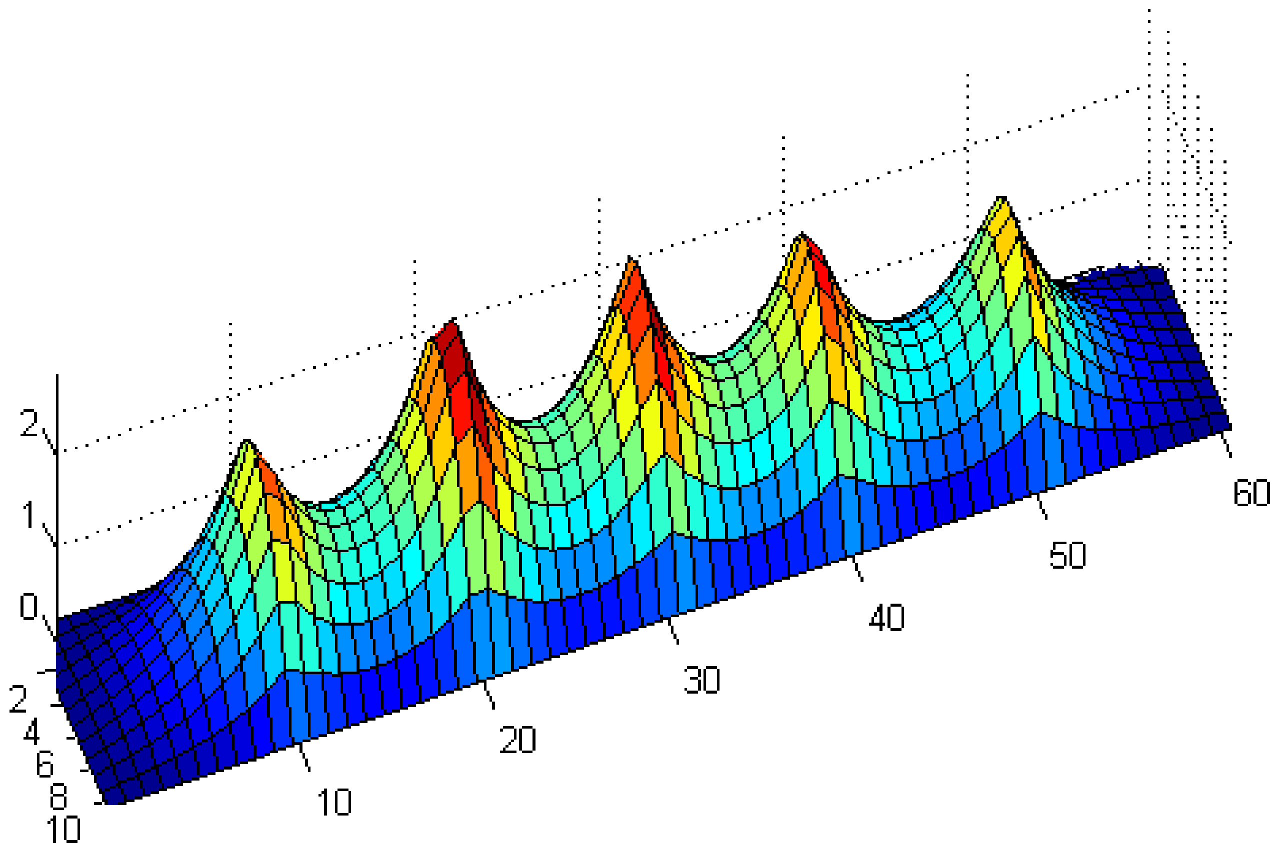
(P_{ji} restricts u_i^k to the "boundary")

- Identical iterates for the two versions
- Convergence in two steps, *even though f_j^0 is no longer reconstructed faithfully*
- Communication reduced by a factor of *H/h* !

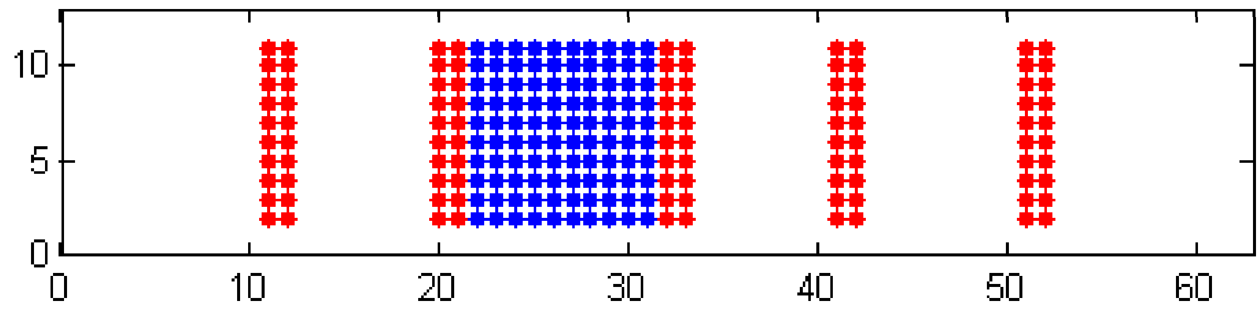
6 × 1 Decomposition



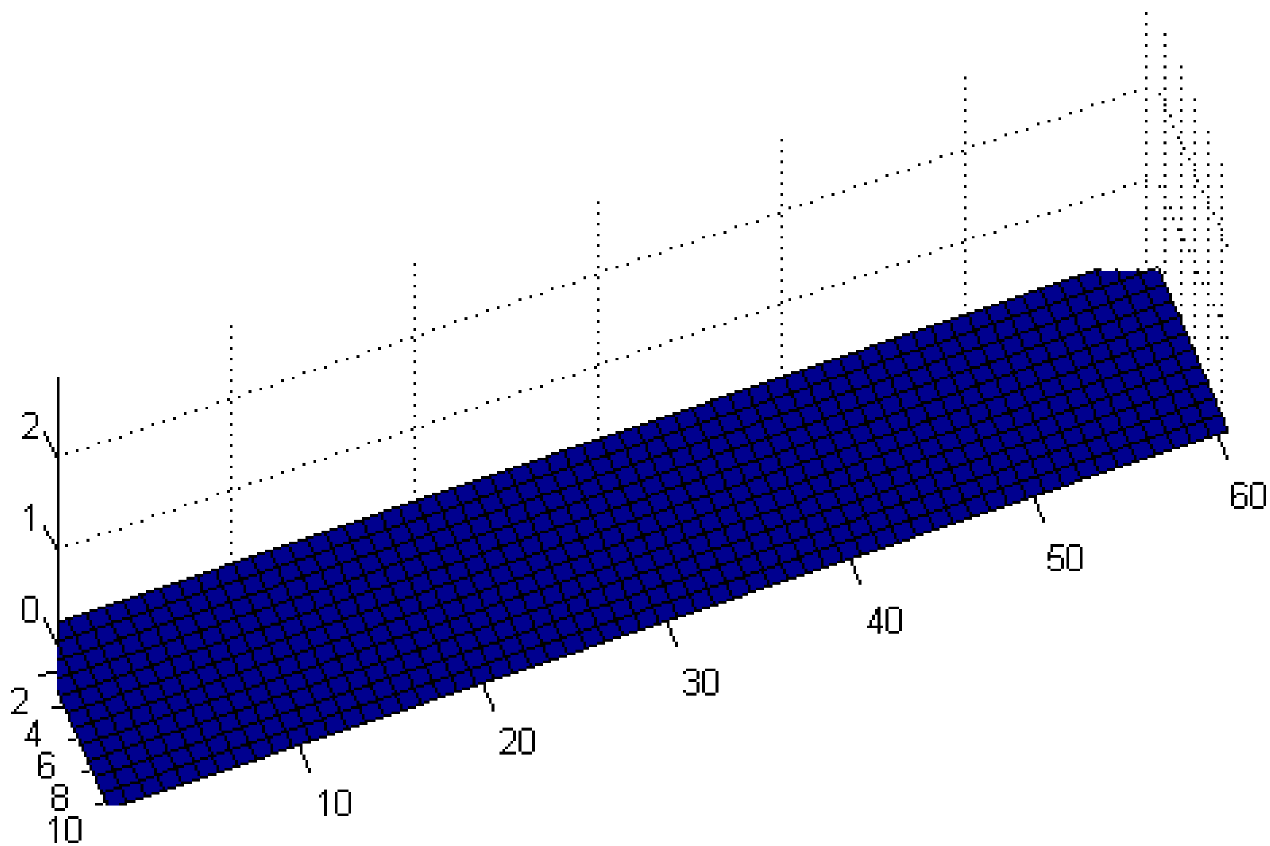
k = 1, Max. Error = 2.86e+000



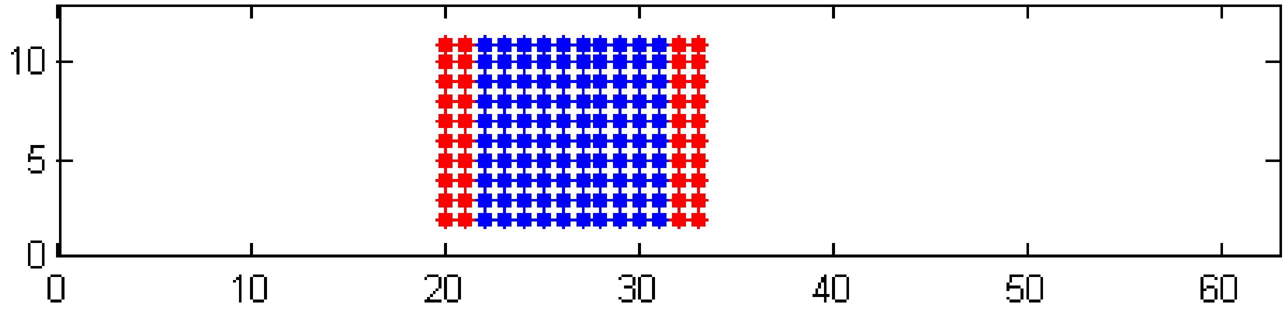
6 × 1 Decomposition

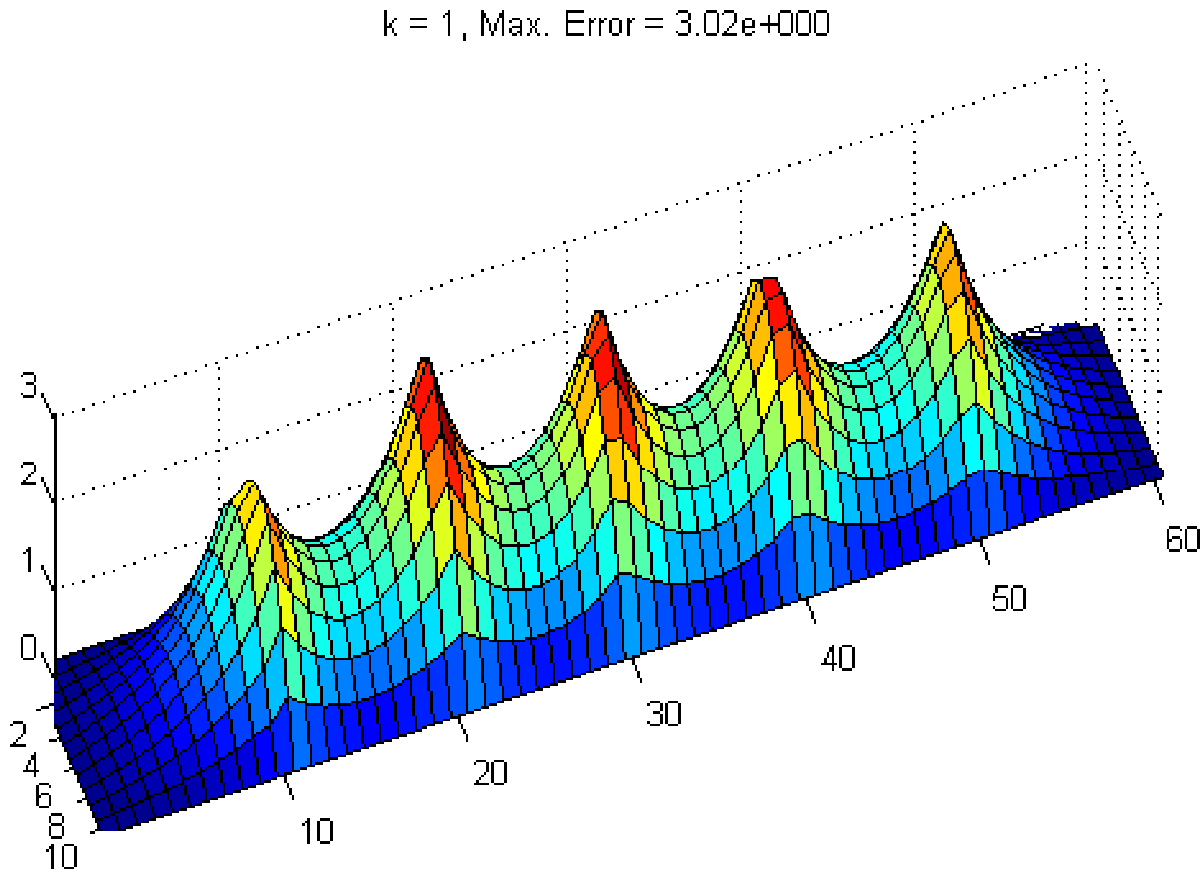


k = 2, Max. Error = 9.77e-015

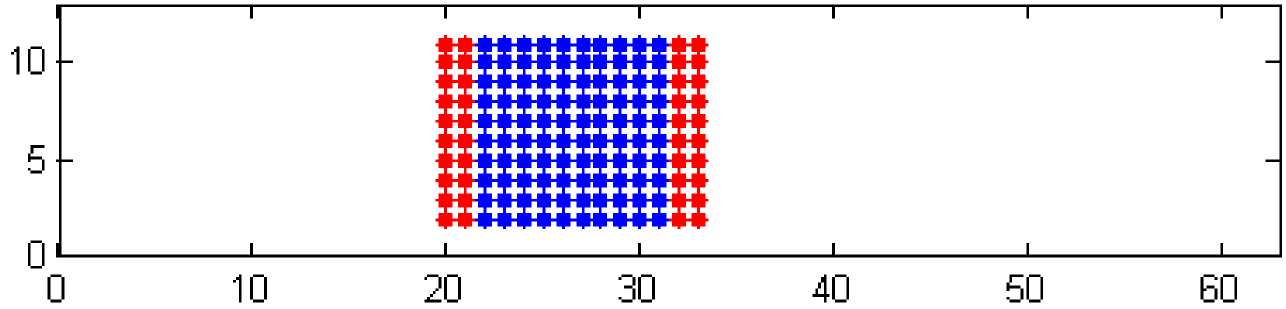


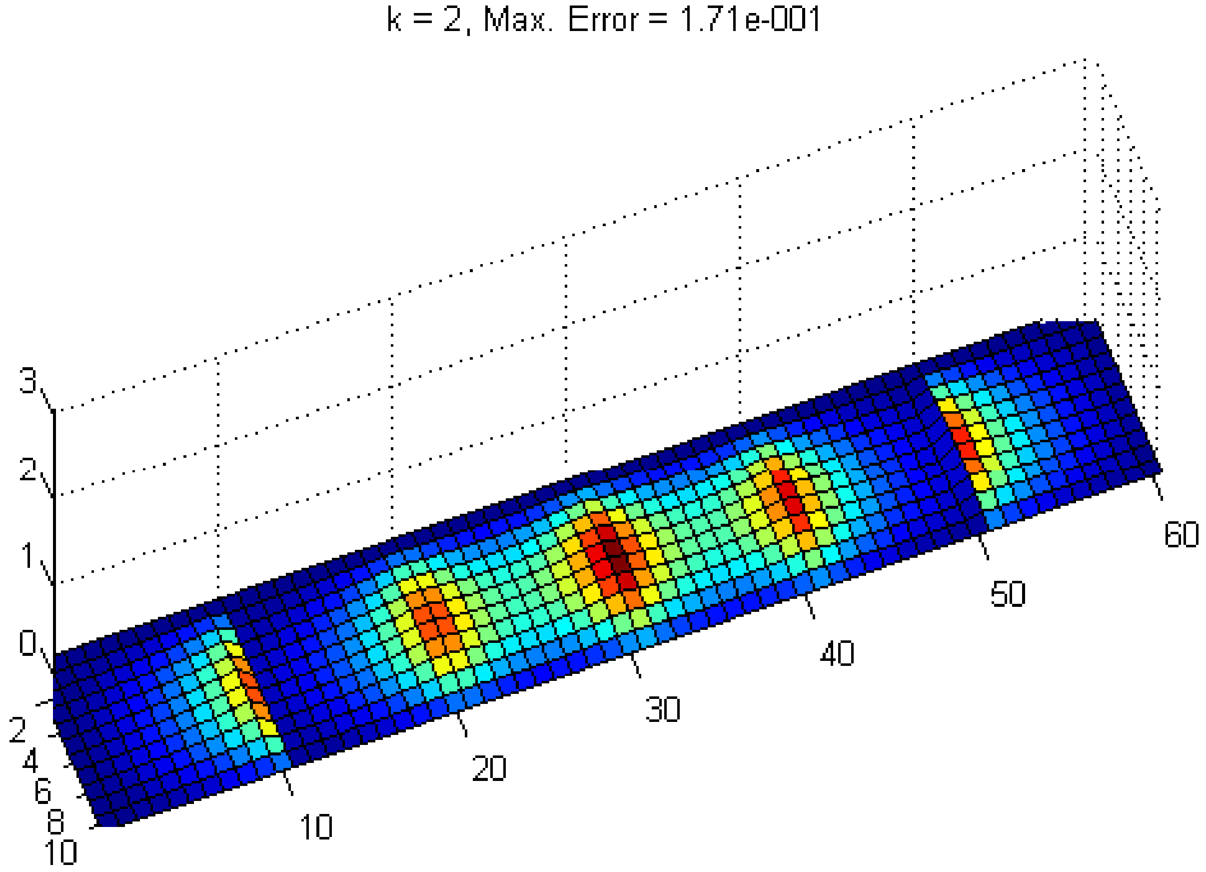
6 × 1 Decomposition



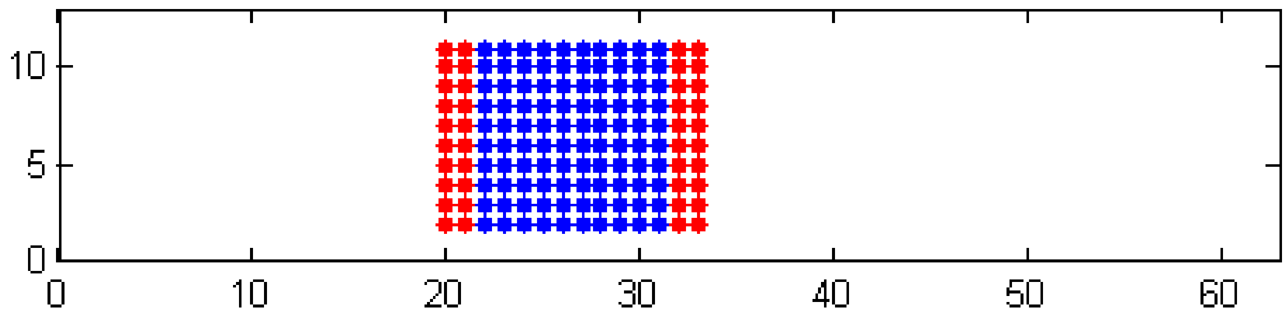


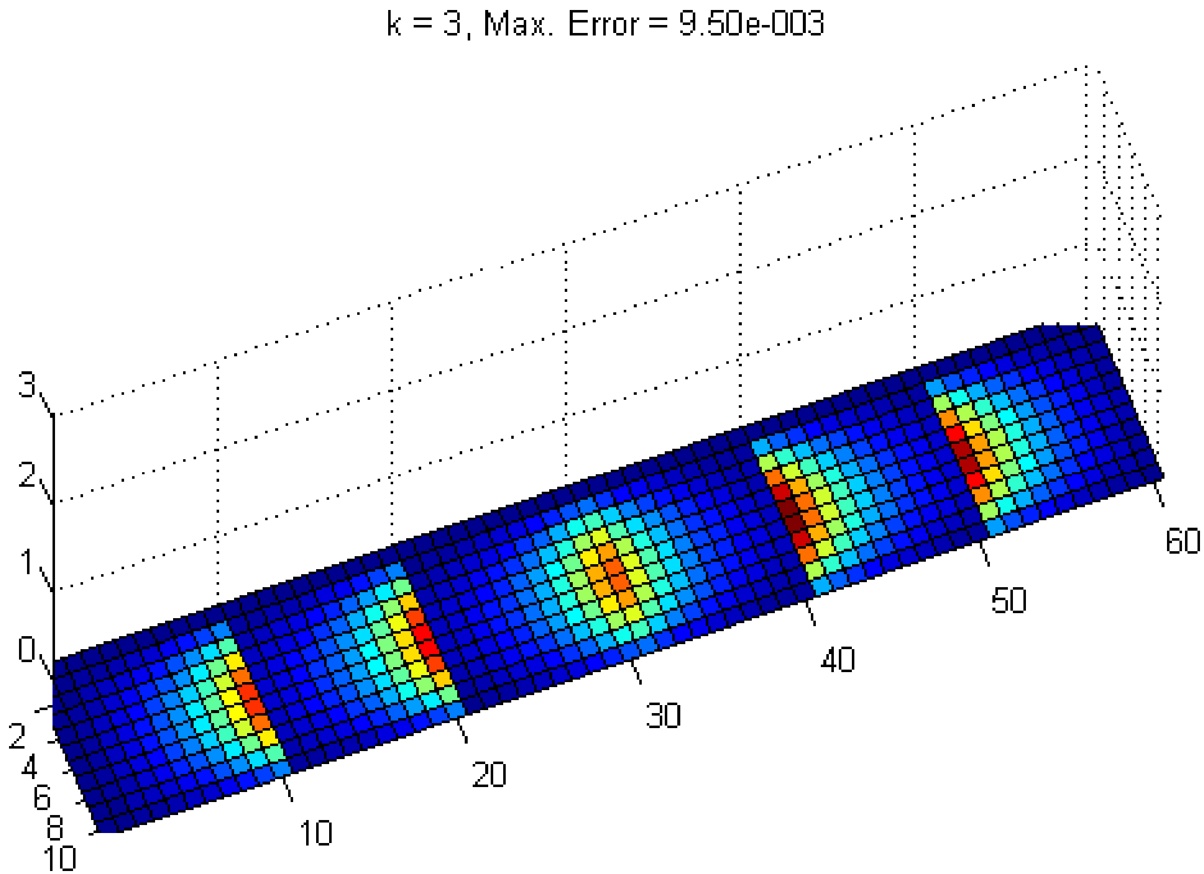
6 × 1 Decomposition



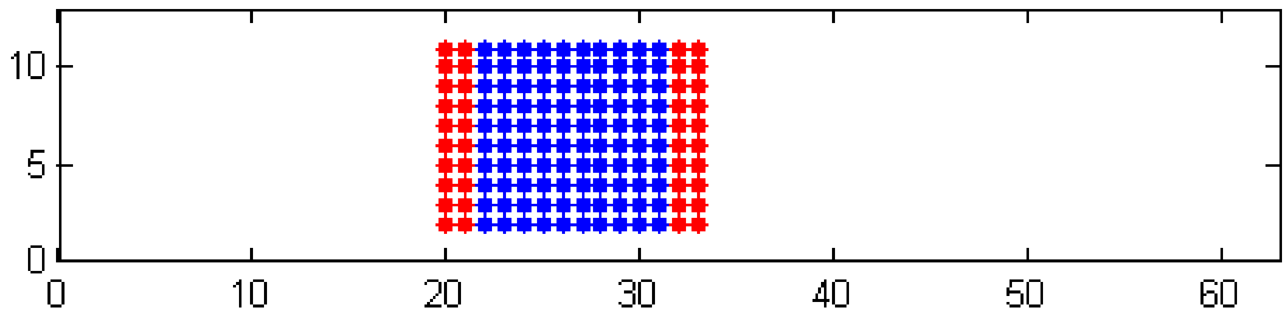


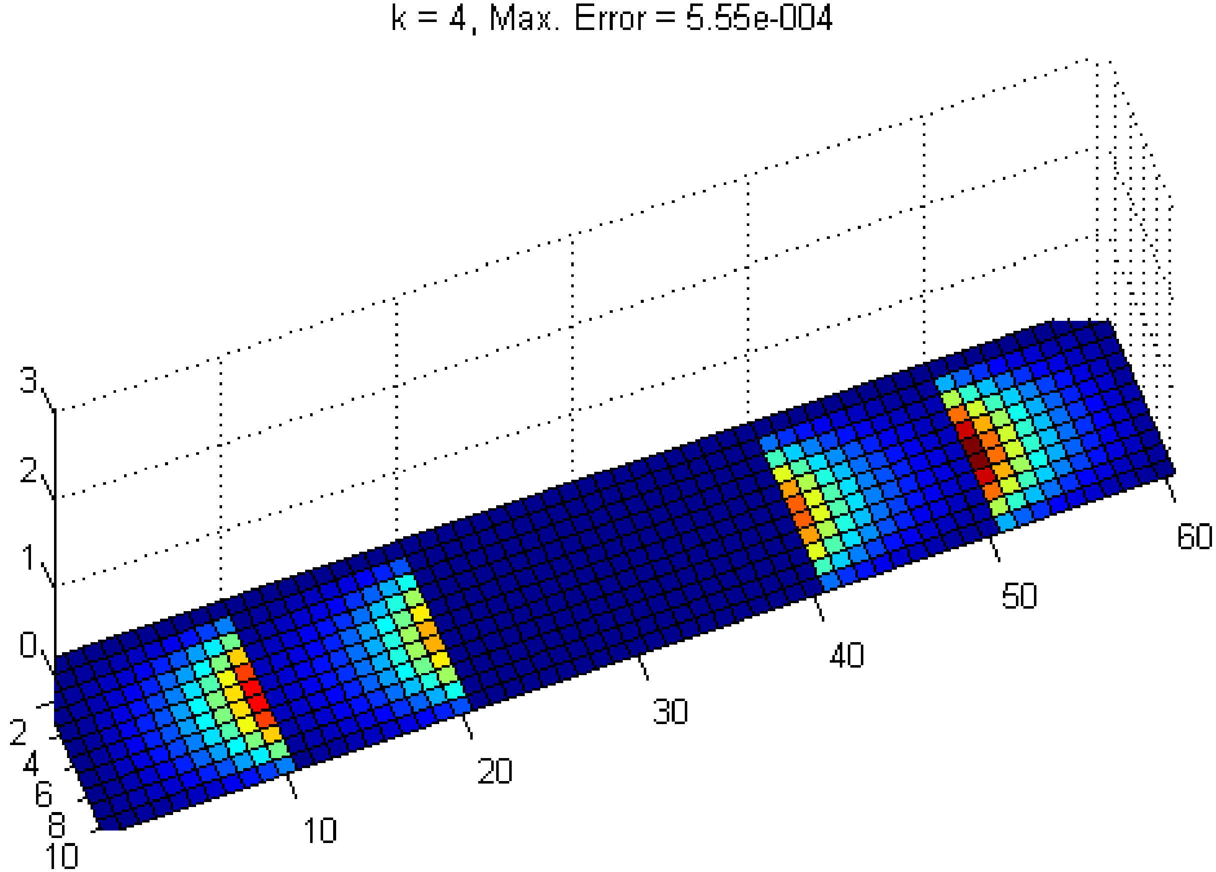
6 × 1 Decomposition



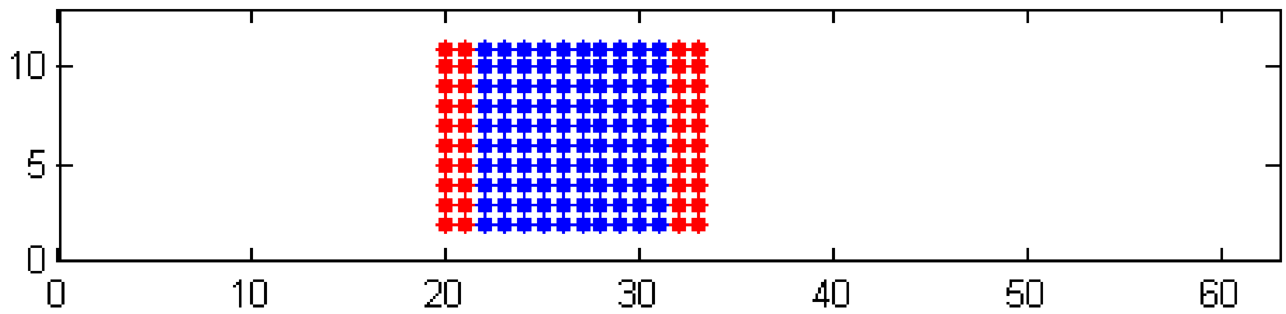


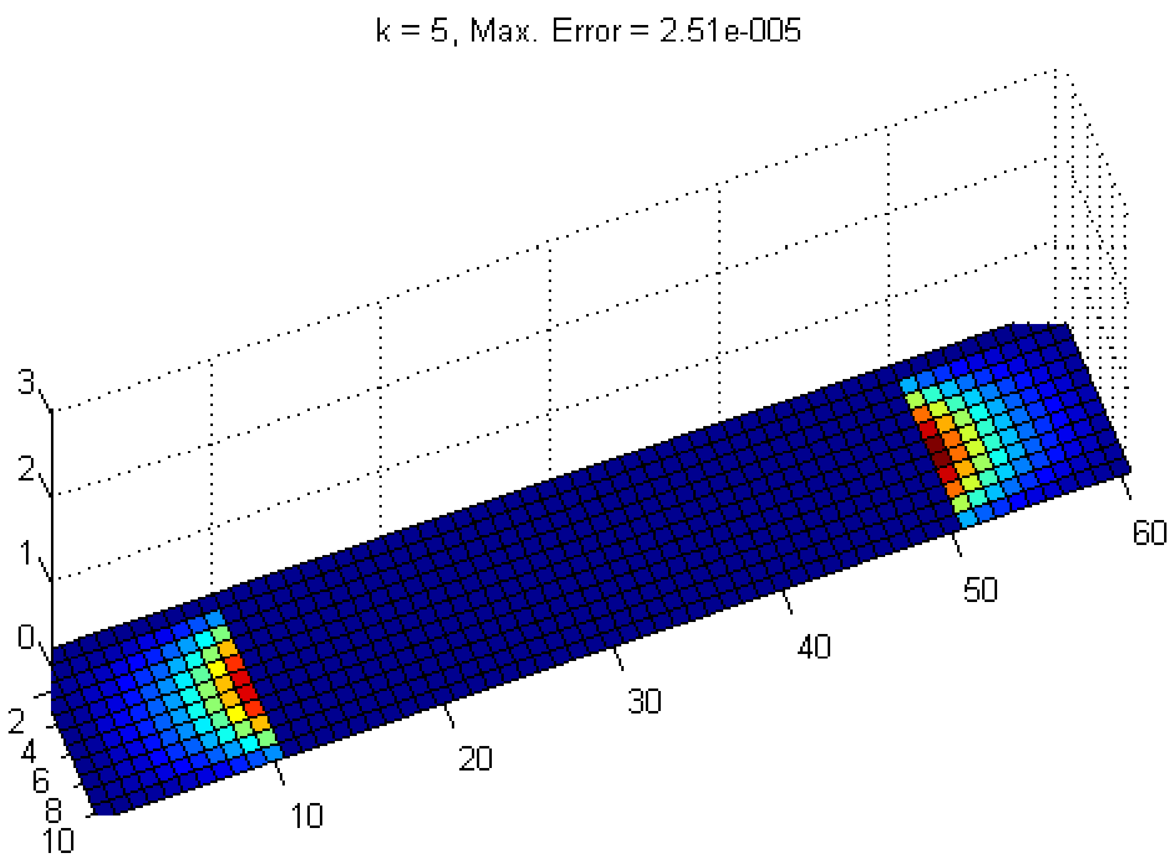
6 × 1 Decomposition



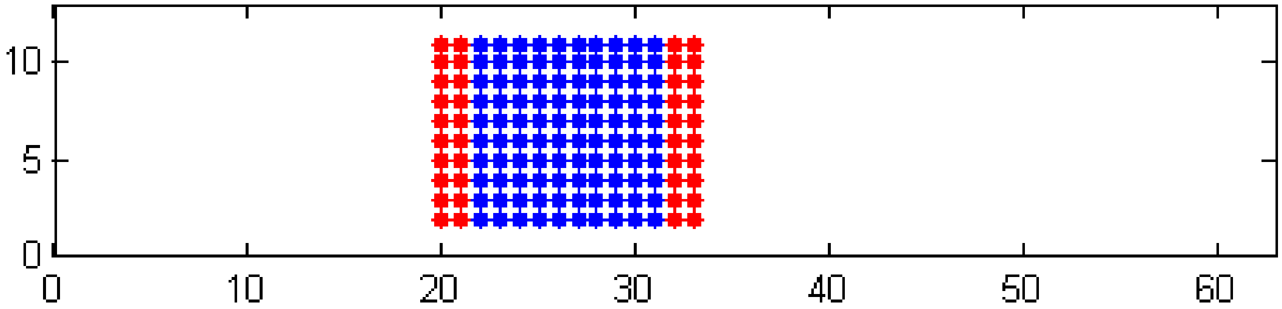


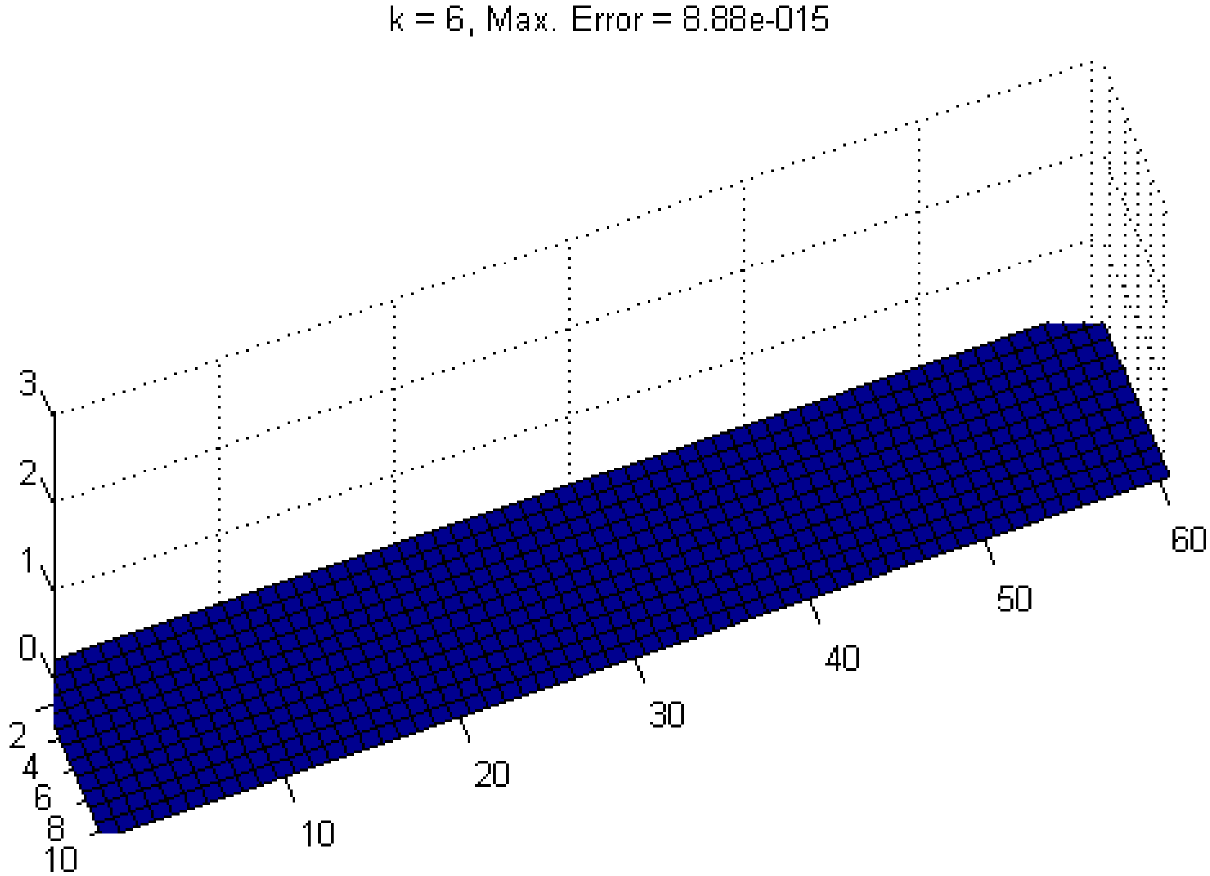
6 × 1 Decomposition



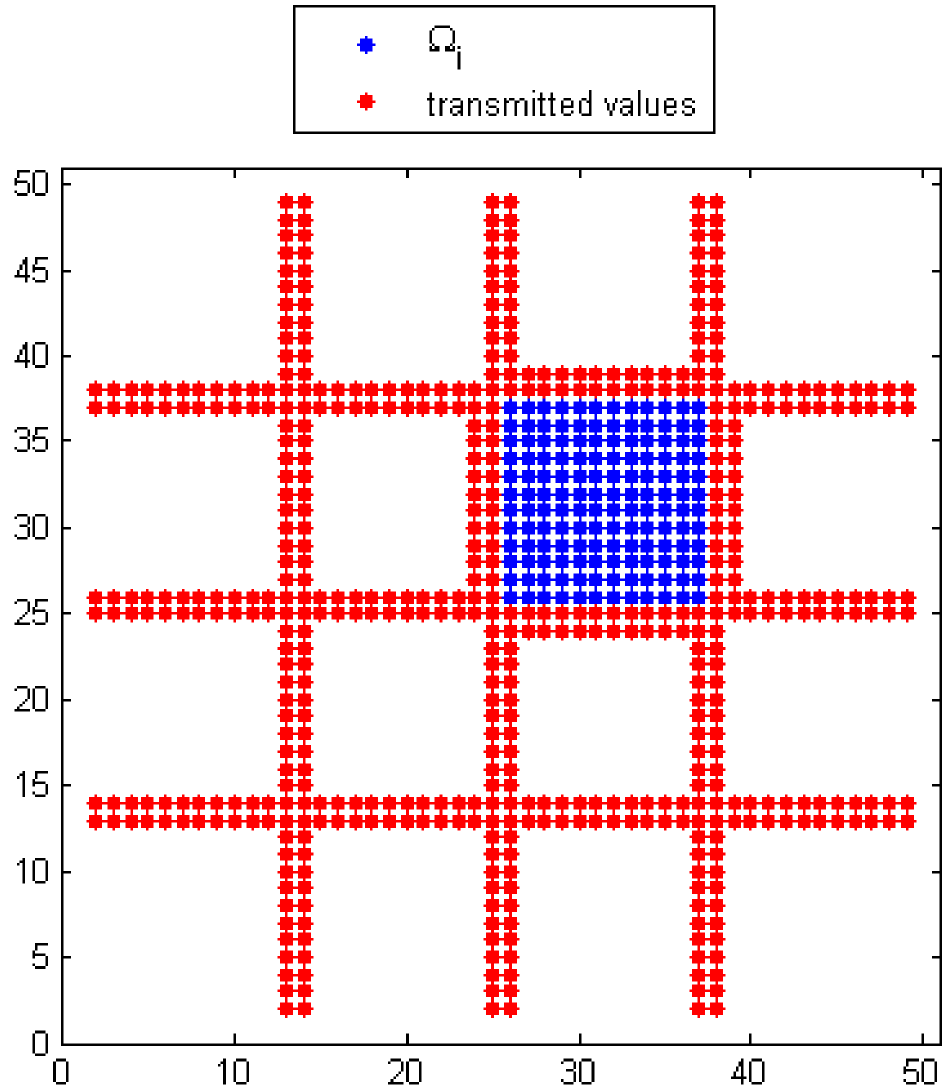


6 × 1 Decomposition

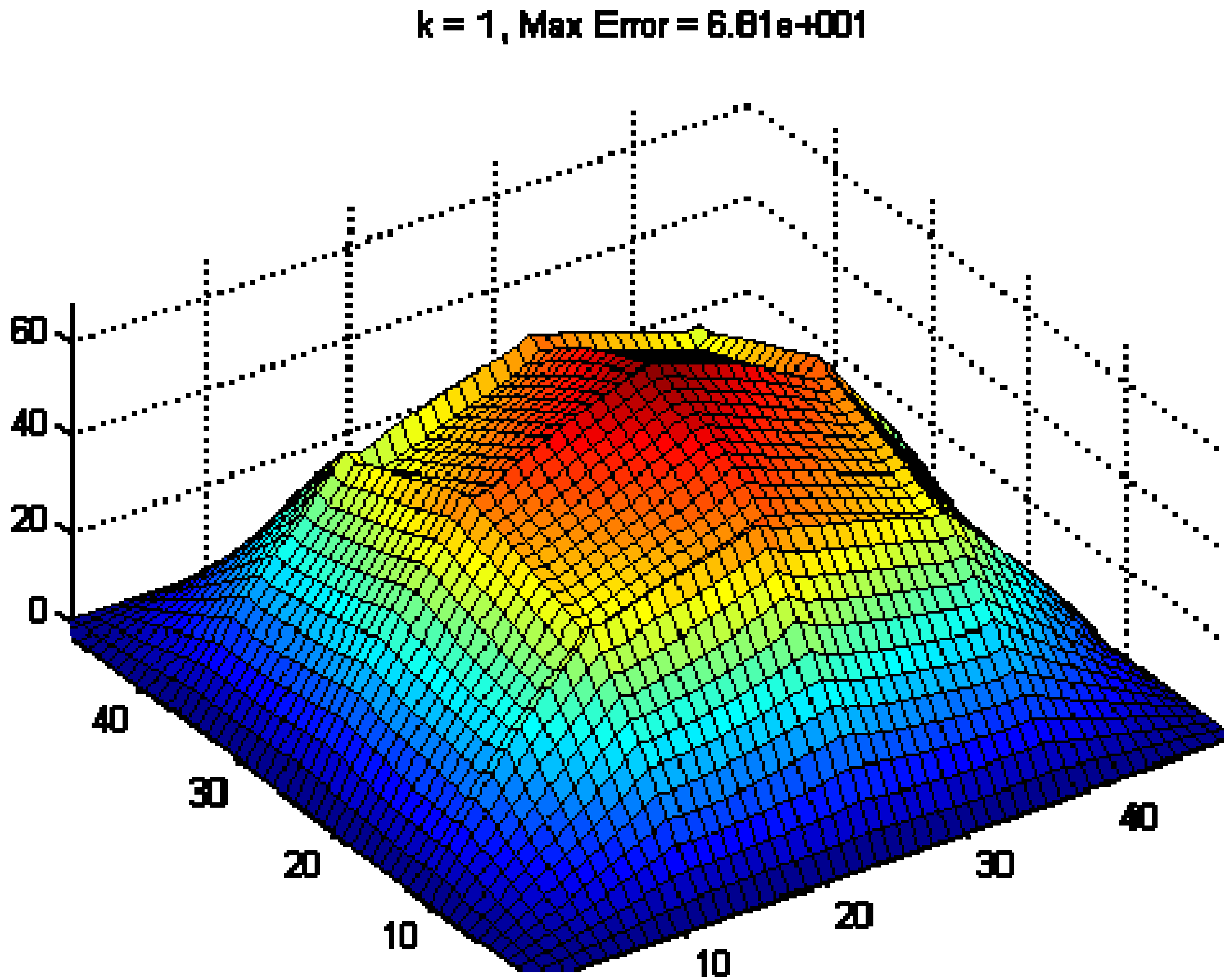




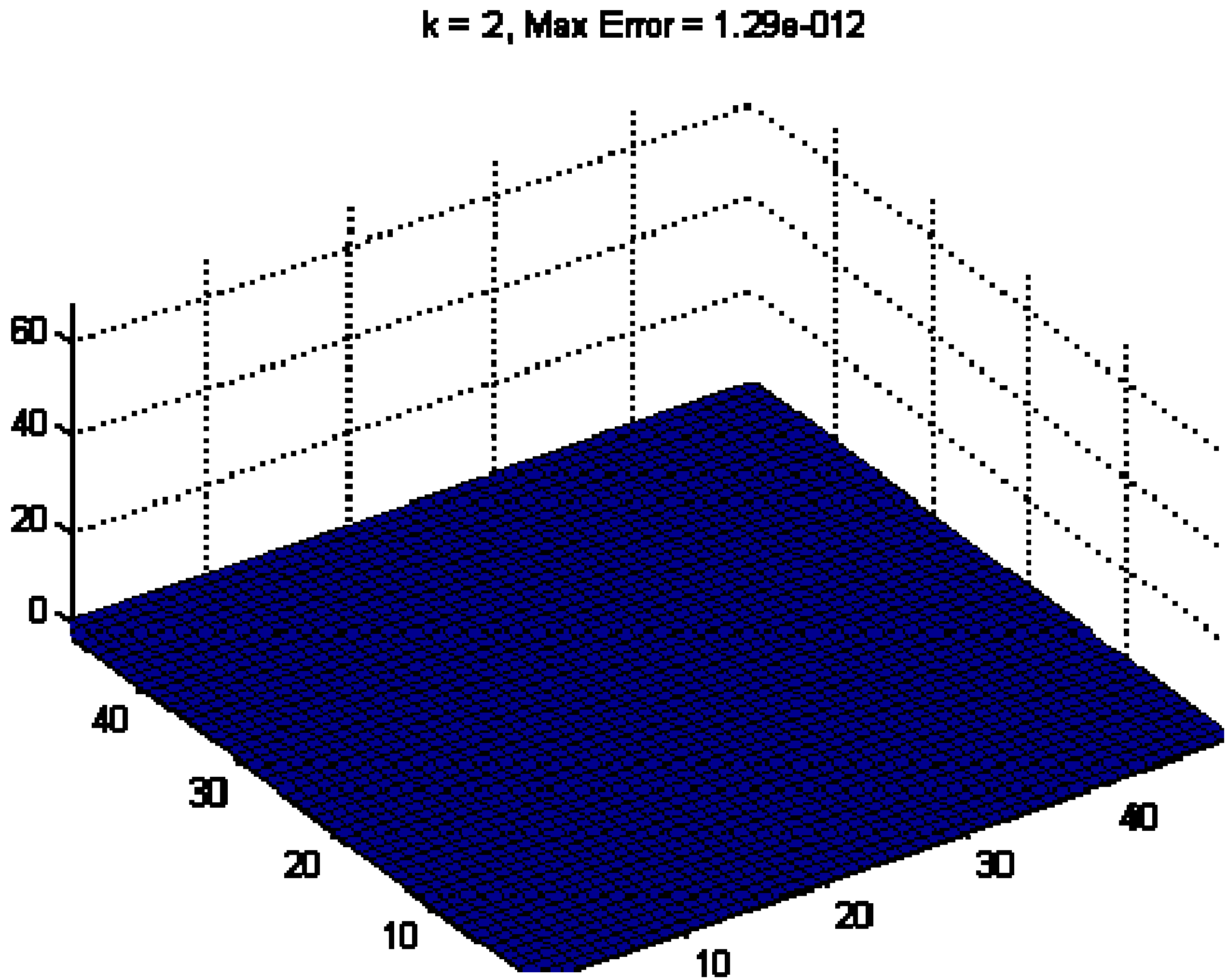
4 × 4 Decomposition



4 × 4 Decomposition



4 × 4 Decomposition



Parallel Algorithm – Version II

$$\left(A_j - B_j D_j^{-1} C_j\right) u_j^{k+1} = f_j - \sum_{i \neq j} B_j D_j^{-1} \begin{bmatrix} 0 \\ R_i A_i P_{ji}^T \\ 0 \end{bmatrix} P_{ji} u_i^k$$

- LHS is the Schur complement (same as tree case), but
- RHS is a special linear combination of data gathered from each of the other subdomains

Conclusions

- New algebraic method based on Schur complements
- Convergence in two iterations possible if one also uses boundary data from non-neighbours
- Works for arbitrary decompositions into subdomains
- Ongoing work:
  - Derive associated *optimized* methods with local approximations of the Schur complements