Multigrid with Multiple Coarsening for the Helmholtz Equation

Ira Livshits

Ball State University

Recent Developments in Fast Helmholtz Solvers
TU Delft
May 18, 2015

Different goals – different approaches

- Use multigrid applied to the Helmholtz operator to precondition one of the Krylov methods;
- Use multigrid applied to the Shifted Complex Laplacian operator to precondition one of the Krylov methods;
- Use multigrid as a stand alone solver for the Helmholtz operator.

Standard multigrid does not work – needs tricks!

Different goals – different approaches

- Use multigrid applied to the Helmholtz operator to precondition one of the Krylov methods;
- Use multigrid applied to the Shifted Complex Laplacian operator to precondition one of the Krylov methods;
- Use multigrid as a stand alone solver for the Helmholtz operator.

Standard multigrid does not work – needs tricks!

Different goals – different approaches

- Use multigrid applied to the Helmholtz operator to precondition one of the Krylov methods;
- Use multigrid applied to the Shifted Complex Laplacian operator to precondition one of the Krylov methods;
- Use multigrid as a stand alone solver for the Helmholtz operator. Standard multigrid does not work – need tricks!

Different goals – different approaches

- Use multigrid applied to the Helmholtz operator to precondition one of the Krylov methods;
- Use multigrid applied to the Shifted Complex Laplacian operator to precondition one of the Krylov methods;
- Use multigrid as a stand alone solver for the Helmholtz operator.
  Standard multigrid does not work – need tricks!

Different goals – different approaches

- Use multigrid applied to the Helmholtz operator to precondition one of the Krylov methods;
- Use multigrid applied to the Shifted Complex Laplacian operator to precondition one of the Krylov methods;
- Use multigrid as a stand alone solver for the Helmholtz operator.
  Standard multigrid does not work – need tricks!

Main ingredients

- Each error component is effectively reduced on some grid where it is accurately resolved;
- The near-kernel (Fourier) components and/or the lowest eigenmodes of the finest grid discrete operator have to be accurately resolved on all grids, including the coarsest, as they are poorly reduced by residual-based relaxations.

Main ingredients

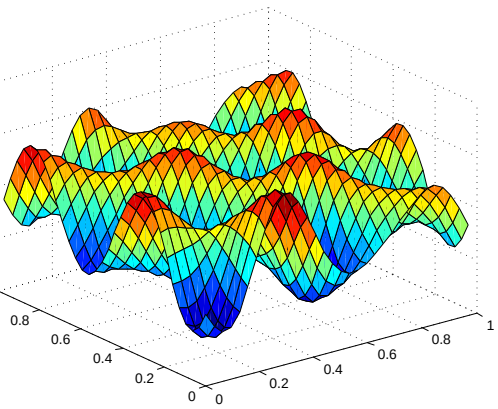
- Each error component is effectively reduced on some grid where it is accurately resolved;
- The near-kernel (Fourier) components and/or the lowest eigenmodes of the finest grid discrete operator have to be accurately resolved on all grids, including the coarsest, as they are poorly reduced by residual-based relaxations.

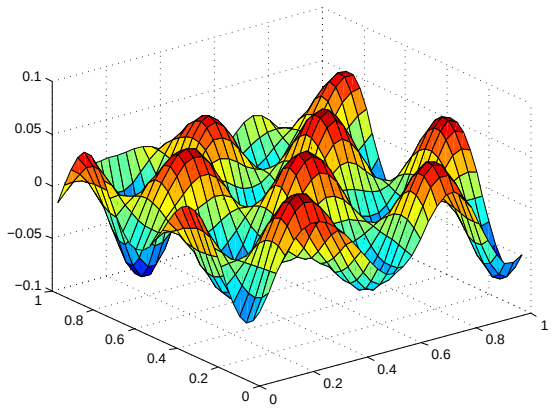
Main ingredients

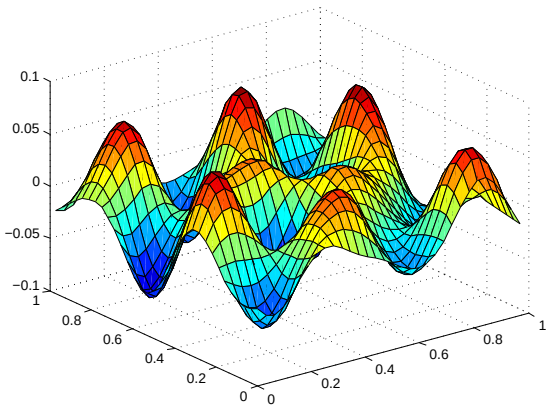
- Each error component is effectively reduced on some grid where it is accurately resolved;
- The near-kernel (Fourier) components and/or the lowest eigenmodes of the finest grid discrete operator have to be accurately resolved on all grids, including the coarsest, as they are poorly reduced by residual-based relaxations.

Helmholtz - the black sheep of multigrid

$$Lu(x) = -\Delta u(x) - k^2(x)u(x) = 0$$





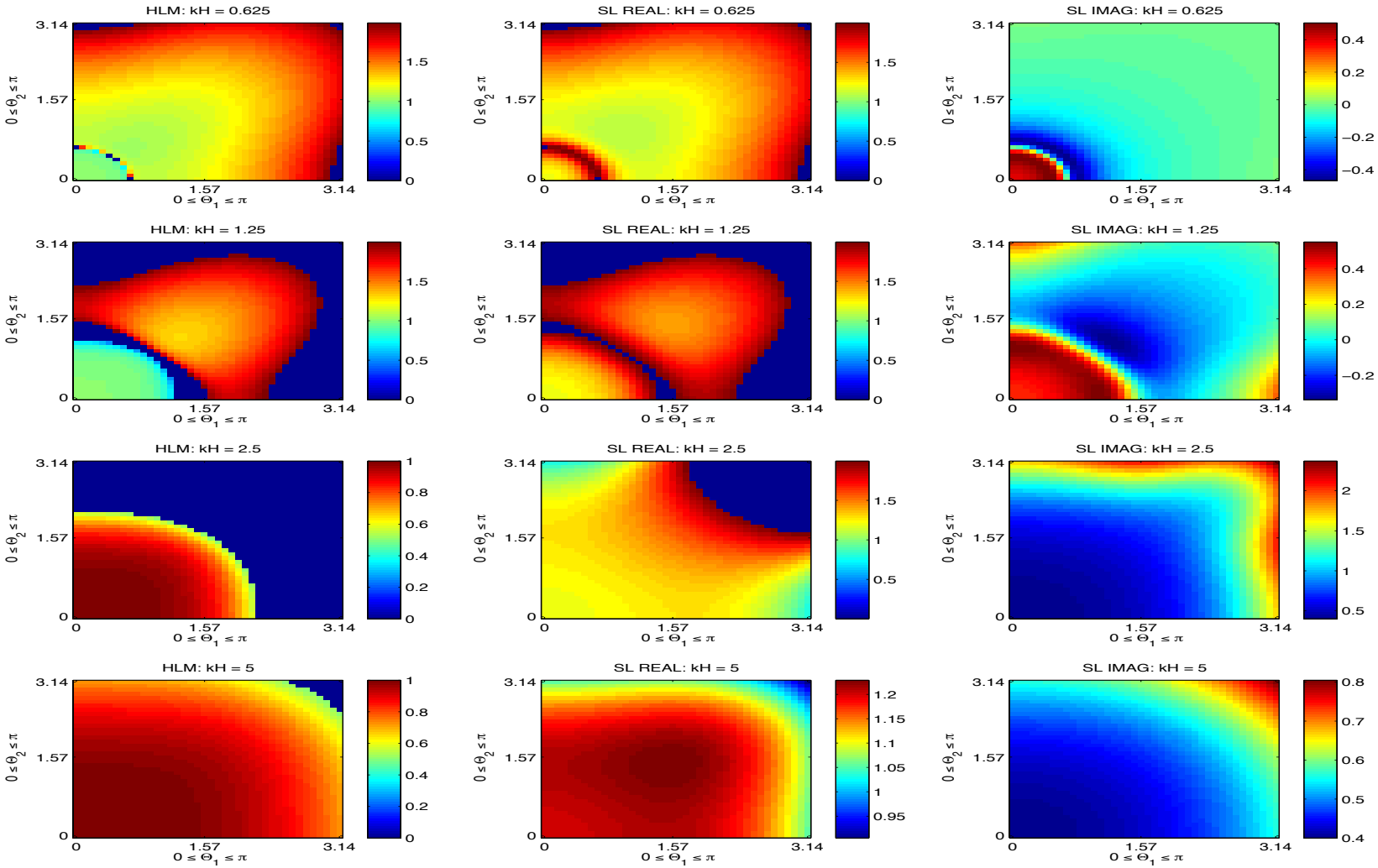


Results of applying three multigrid V-cycles to the homogeneous Helmholtz equation using random initial approximations;

Obvious bottlenecks for multigrid

- Dominant phase discretization error: $O((dk)(k^2h^2))$;
- Poor multigrid relaxation - standard schemes diverge due to many *negative* modes.
- Quality of standard coarse-grid and prolongation operators deteriorates on coarse grids;

Coarse grid approximation (2014)



More bottlenecks for multigrid

- Oscillatory near-zero modes;
- There are many of them;
- They are different from each other;
- On sufficiently coarse grids, there is no a single prolongation or a coarse grid operator of a reasonable sparsity that works for all of them.

Ideas

- Choose a *basis* for the set of near-kernel (low-energy) components (with smooth rather than constant coordinates);
- Build *coarse* and *smooth* representation for each of the basis functions and its *neighbors*
- Hope: together, multiple coarse representations will effectively reduce all near-kernel error components at reasonable costs.

Ideas

- Choose a *basis* for the set of near-kernel (low-energy) components (with smooth rather than constant coordinates);
- Build *coarse* and *smooth* representation for each of the basis functions and its *neighbors*
- Hope: together, multiple coarse representations will effectively reduce all near-kernel error components at reasonable costs.

Ideas

- Choose a *basis* for the set of near-kernel (low-energy) components (with smooth rather than constant coordinates);
- Build *coarse* and *smooth* representation for each of the basis functions and its *neighbors*
- Hope: together, multiple coarse representations will effectively reduce all near-kernel error components at reasonable costs.

Geometric multigrid: multiple coarse representation

1997, 2006, 2014

Helmholtz near-kernel Fourier components are $e_\omega = e^{i(\omega_1 x + \omega_2 y)}, \quad |\omega| \approx k$.
Indeed,

$$Le_\omega \approx (k^2 - |w|^2)e_\omega.$$

The error unreduced by standard multigrid

$$e(x,y) = \sum_{|\omega| \approx k} \alpha_\omega e_\omega,$$

$$(1-\gamma_1)k \le |\omega| \le (1+\gamma_2)k,$$

where typical are values $\gamma_j \approx 0.3$

Multiple representations

Consider basic functions $u_\kappa = e^{i(k_1^\kappa x + k_2^\kappa y)}$, $(k_1^\kappa)^2 + (k_2^\kappa)^2 = k^2$, $\kappa = 1, \ldots, K$).

$$(k_1^\kappa, k_2^\kappa) = k(\cos((\kappa - 1)\frac{\pi}{4}, \sin((\kappa - 1)\frac{\pi}{4})$$

Then unreduced error can be represented as

$$e(x, y) = \sum_{\kappa=1}^{K} e_\kappa(x, y) u_\kappa(x, y)$$

where *weight functions* e_κ are smooth compared to u_κ.
Idea: Approximate smooth e_κ instead of the oscillatory $e(x, y)$.

Components of multiple representation: Geometric multigrid, constant k.

- Differential operators L_κ for e_κ is obtained from

$$Le(x,y) = L\left(\sum_{\kappa=1}^{K} e_\kappa(x,y)u_\kappa(x,y)\right) = \sum_{\kappa=1}^{K} u_\kappa(x,y)L_\kappa e_\kappa(x,y)$$

- Residuals:

$$r(x,y) = Le(x,y) = \sum_{\kappa=1}^{K} r_\kappa(x,y)u_\kappa(x,y)$$

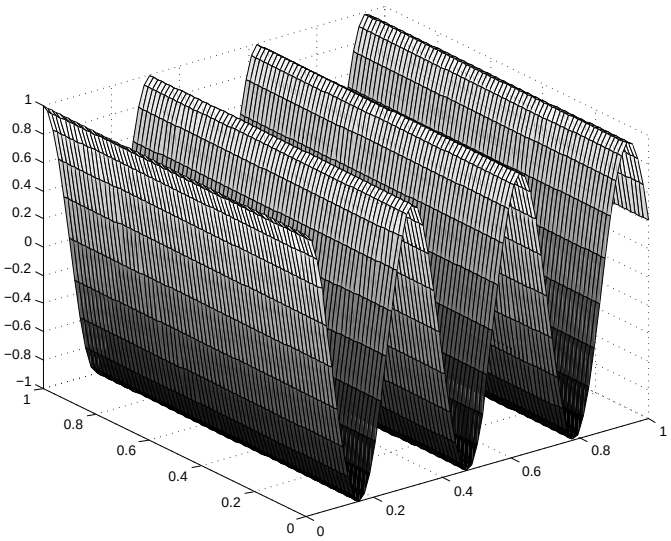
  where r_κ are separated using averaging to the target coarse grid;
- Discrete $L_\kappa e_\kappa = r_\kappa$ are solved (using line relaxation) on scale with $kH = O(1)$.

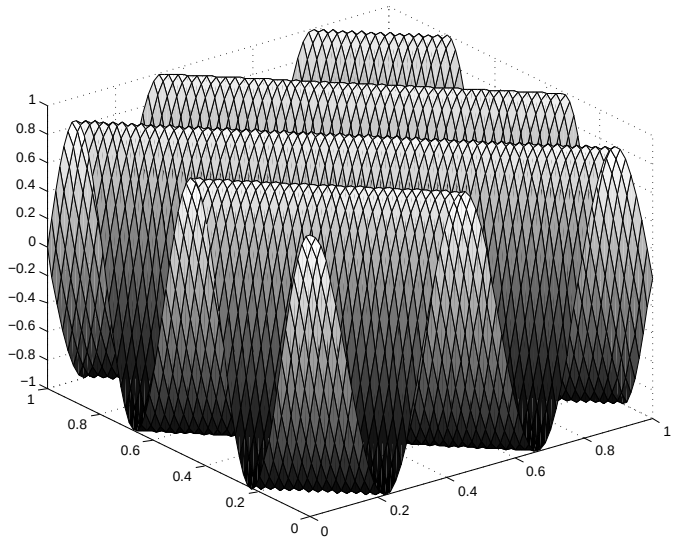
Geometric Multigrid: constant k

kh		k			
		40	80	160	320
0.625	Cycles	13	14	14	14
0.3125	Cycles	12	12	13	14
0.1562	Cycles	15	15	16	16

Table: The number of geometric cycles needed to satisfy $||r||/||f|| \leq 10^{-6}$; The results are computed for different choices of k and the finest h; the number of levels varies from $L = 5$ ($k = 40, kh = 0.625$) to $L = 10$ ($k = 320, kh = 0.15625$).

Algebraic multigrid: constant k (2015)





Multiple corrections in AMG

- There is a set of prolongation operators $\{P_\kappa\}$, $\kappa = 1, \ldots, K$, $K = O(1)$,

where each P_κ is $n \times n_c$ sparse operator with $n_c \ll n$, such that e belongs to its collective range:

$$e \in \bigcup_{\kappa=1}^{K} \mathcal{R}(P_\kappa). \tag{1}$$

Then e can be approximately represented as

$$e = \sum_{\kappa=1}^{K} P_\kappa e_\kappa^c, \tag{2}$$

where $\{e_\kappa^c\}_{\kappa=1}^K$ is a set of coarse functions. The fine scale residual equation is then approximated by

$$\sum_{\kappa=1}^{K} A P_\kappa e_\kappa^c = r. \tag{3}$$

A coarse (block) system

$$\mathbb{A}^c \mathbf{e}^c = \mathbf{r}^c, \tag{4}$$

where each block is given by

$$A_{p,q}^c = P_p^* A P_q, \quad p, q = 1, \ldots, K. \tag{5}$$

Prolongation operators

Each operators P_κ are built to exactly interpolate a given basis vector u_κ from a constant coarse function:

$$u_\kappa = P_\kappa 1^{n_c},$$

where 1^{n_c} is a coarse scale unit vector. More precisely, P_κ is defined as

$$P_\kappa = B_\kappa I_{n_c}^n, \tag{6}$$

where $I_{n_c}^n : \mathbb{R}^n \to \mathbb{R}^{n_c}$ is a bilinear interpolation operator and B is a $n \times n$ diagonal matrix with u_κ on the main diagonal.
Clearly,

$$P_\kappa 1^{n_c} = B_\kappa I_{n_c}^n 1^{n_c} = B_\kappa 1^n = u_\kappa.$$

With out choice of prolongations $P_q^t P_r \approx \mathbf{0}^{n_c}$ and $\mathbf{A}^c$ is block diagonally dominant!

AMG: constant k – multigrid solver

	k				
kh	20	40	80	160	320
0.625	8	12	21	25	40
0.3125	8	9	16	27	38
0.15625	8	9	14	16	38

AMG: constant k – multigrid preconditioner

	k				
kh	20	40	80	160	320
0.625	5	7	9	11	24
0.3125	6	7	8	11	20
0.15625	6	7	8	11	20

Discontinuous $k(x)$

$$k(x,y) = \begin{cases} \gamma_1 k_0, & \text{if } y \le \ell_1(x), \\ k_0, & \text{if } \ell_1(x) < y < \ell_2(x), \\ \gamma_2 k_0, & \text{if } y \ge \ell_2(x), \end{cases} \tag{7}$$

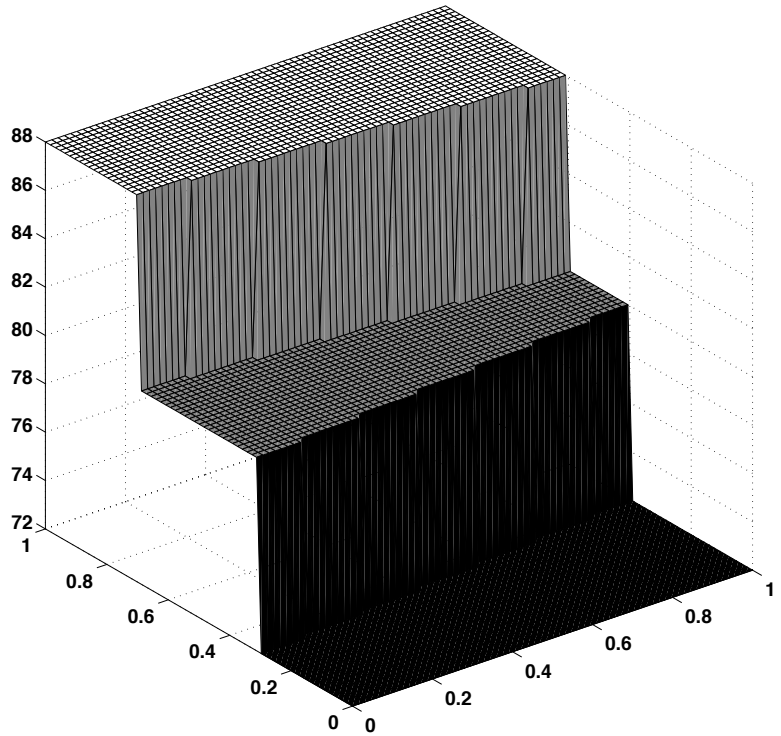
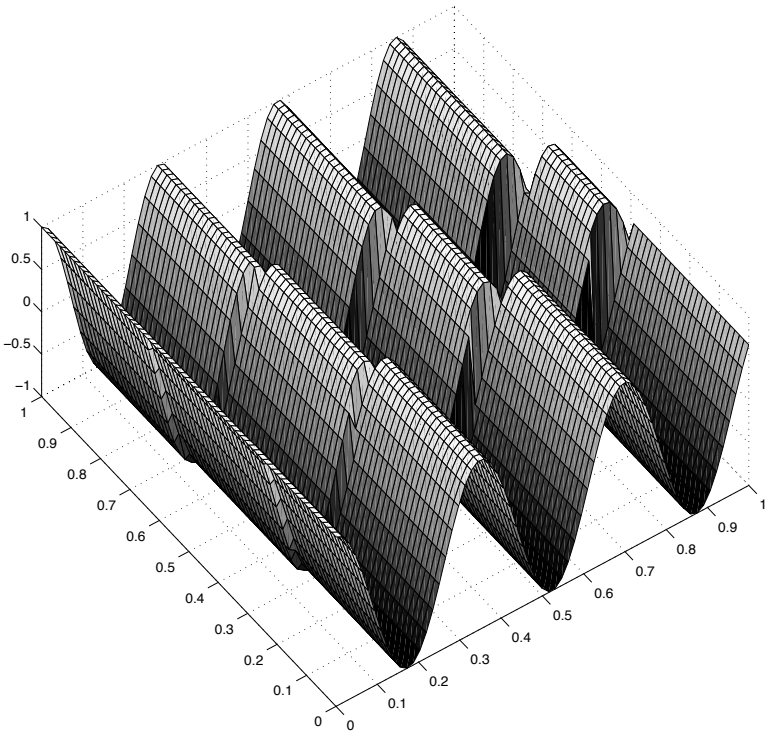


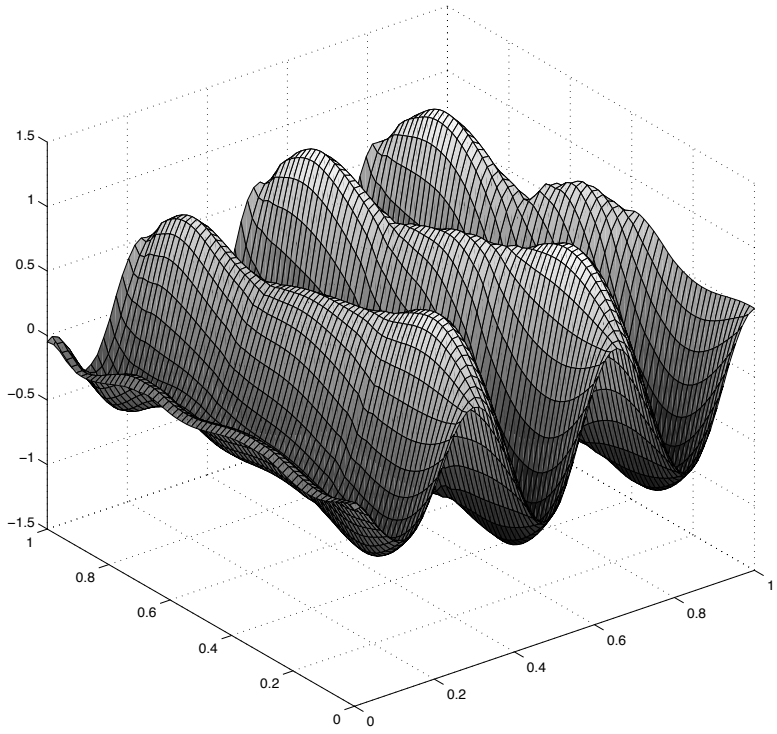
Figure: Wave number function $k(x,y)$, defined for for $k_0 = 80$, $\gamma_1 = 0.9$ and $\gamma_2 = 1.1$.

Basis functions: with and without multigrid preprocessing

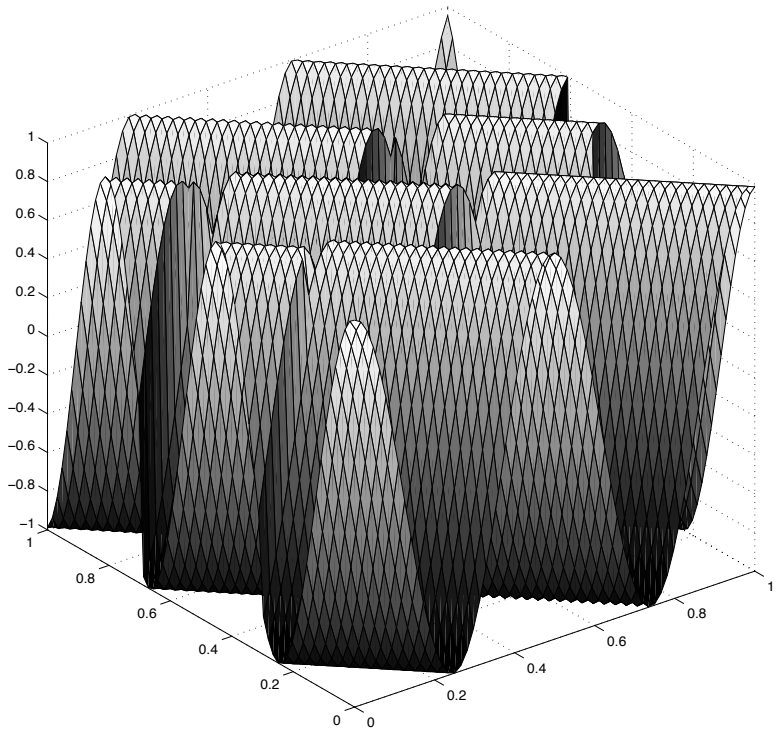
Basis functions u_κ, $\kappa = 1, 2$ for $k_0 = 20$, $\gamma_1 = 1.1$ and $\gamma_2 = 0.9$ before and after application of one standard V-cycle to $Ab_\kappa = 0$.



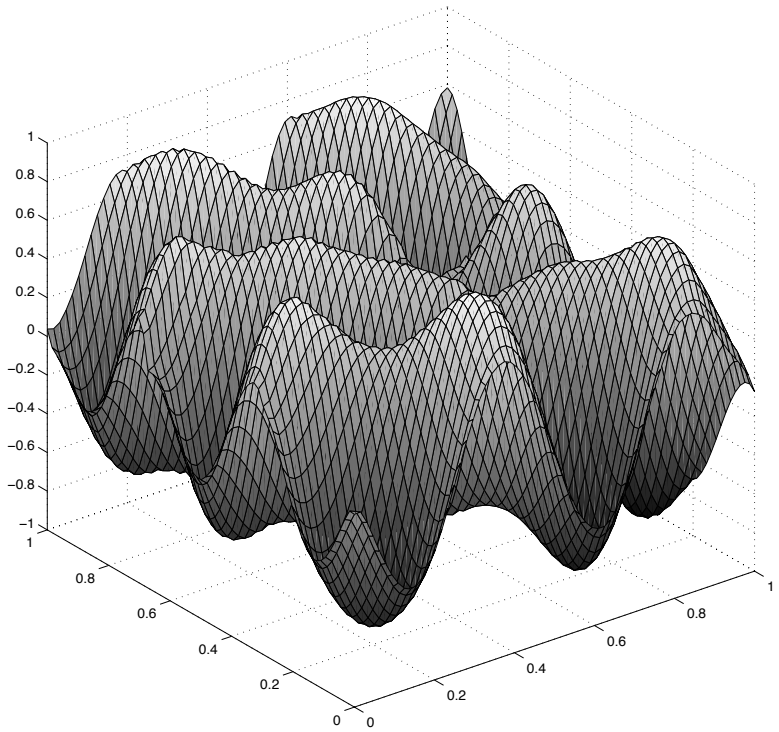
(a) Real part of u_1 before preprocessing



(b) Real part of u_1 after preprocessing



(c) Real part of u_2 before preprocessing



(d) Real part of u_2 after preprocessing

Mutigrid as preconditioner without/with preprocessing

	k_0				
$\gamma = \gamma_2/\gamma_1$	40	60	80	120	240
$1.2 = 1.1/0.9$	18	22	30	35	42
$1.5 = 1.2/0.8$	22	23	38	35	45
$2 = 1.4/0.7$	27	27	40	45	80
$3 = 1.8/0.6$	52	33	64	63	102

	k_0				
$\gamma = \gamma_2/\gamma_1$	40	60	80	120	240
$1.2 = 1.1/0.9$	15	13	17	16	36
$1.5 = 1.2/0.8$	14	13	16	24	32
$2 = 1.4/0.7$	22	18	36	23	61
$3 = 1.8/0.6$	48	22	52	39	73

Conclusions

- Solvers are a combination of standard multigrid and multiple corrections;
- Basis functions are exponent-based, providing local independence and orthogonality;
- Preprocessing of basis functions is beneficial for non-constant wave numbers;
- The complexity is similar to the one for Laplace, it is $O(n)$;

New directions include

- Choice of basis functions based on the laws of geometric optics (tested in 1D);
- Use of different scales for multiple representations, depending on local values of k (tested in 1D);
- Numerical discovery of basis functions using standard multigrid and local orthogonalization (tested in 1D).

New directions include

- Choice of basis functions based on the laws of geometric optics (tested in 1D);
- Use of different scales for multiple representations, depending on local values of k (tested in 1D);
- Numerical discovery of basis functions using standard multigrid and local orthogonalization (tested in 1D).

New directions include

- Choice of basis functions based on the laws of geometric optics (tested in 1D);
- Use of different scales for multiple representations, depending on local values of k (tested in 1D);
- Numerical discovery of basis functions using standard multigrid and local orthogonalization (tested in 1D).

New directions include

- Choice of basis functions based on the laws of geometric optics (tested in 1D);
- Use of different scales for multiple representations, depending on local values of k (tested in 1D);
- Numerical discovery of basis functions using standard multigrid and local orthogonalization (tested in 1D).