Introduction to Multigrid Methods for Differential Problems

Ira Livshits
Department of Mathematical Sciences
Ball State University
ilivshits@bsu.edu

Fourth International Workshop
on Numerical Analysis and Lattice QCD
Yale, May 3 2007

Presentation Plan

- Iterative methods: convergence and smoothing rates;
- Motivation to use many grids
- Geometric Multigrid method
  - Main components
  - V-cycle
  - CS scheme vs FAS scheme, FMG
- Algebraic Multigrid method
- Application to eigen problems

Problem in question

$$Lu(x) = f(x), \qquad x \in \Omega \subset \mathbf{R}^d$$

plus some boundary conditions on $\partial\Omega$.

Discretized problem (on a regular grid with mesh-size h):

$$L^h u^h = f^h$$

plus b.c.

For now assume that L is a linear operator and $L^h \equiv A$ is simply a matrix.

Iterative solvers: relaxation

Goal is to solve: $Au = f$

Current approximation : u^n

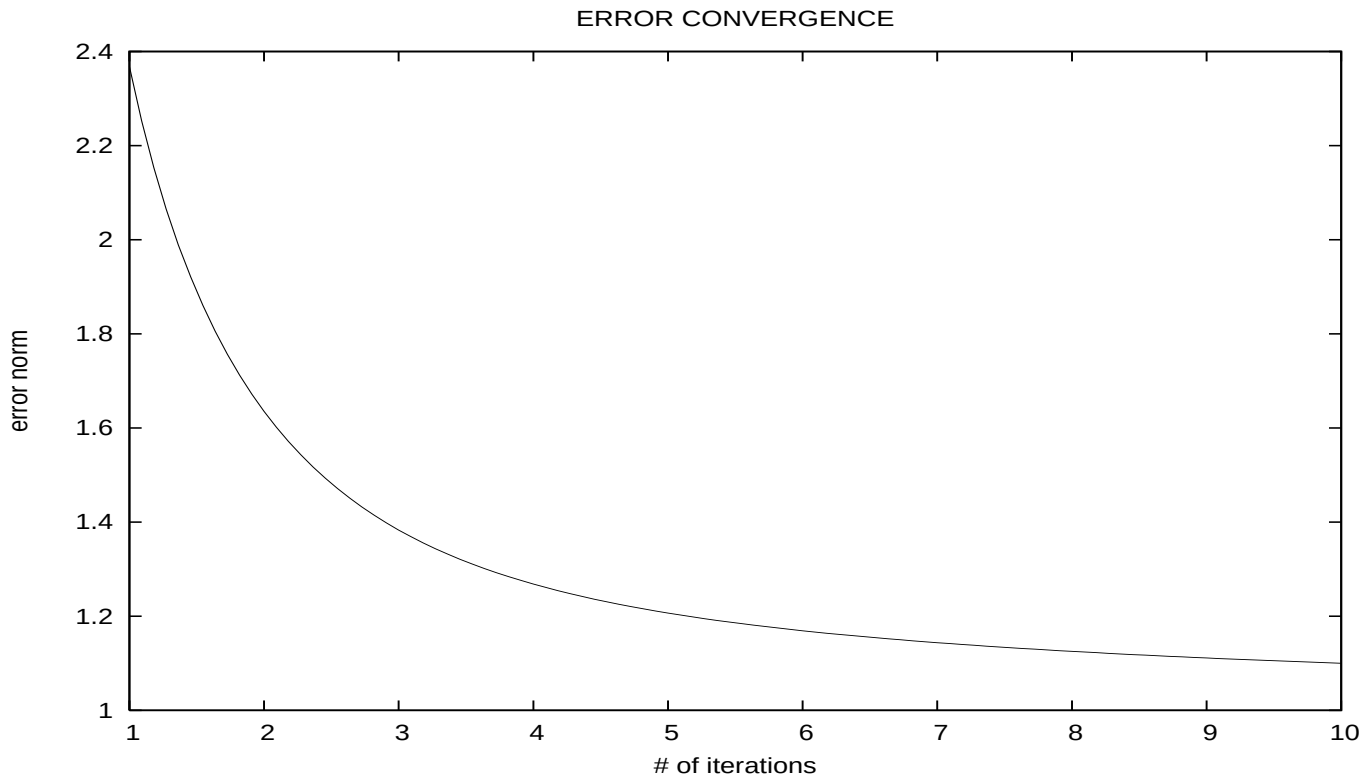
Residual: $r^n = f - Au^n$

Error: $e^n = u - u^n$

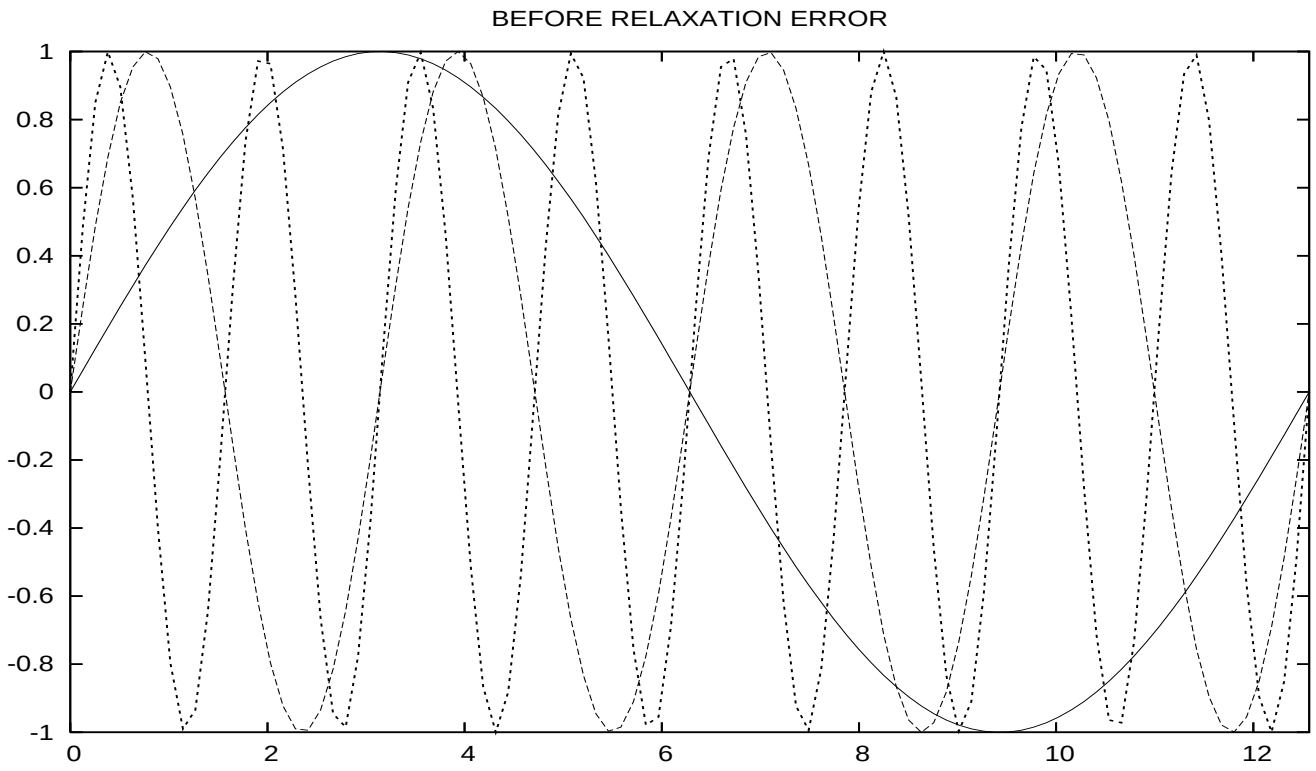
Residual equation: $Ae^n = r^n$

Examples of relaxation schemes: Jacobi, Gauss-Seidel, line relaxation, distributive relaxation (i.e., Kaczmarz).

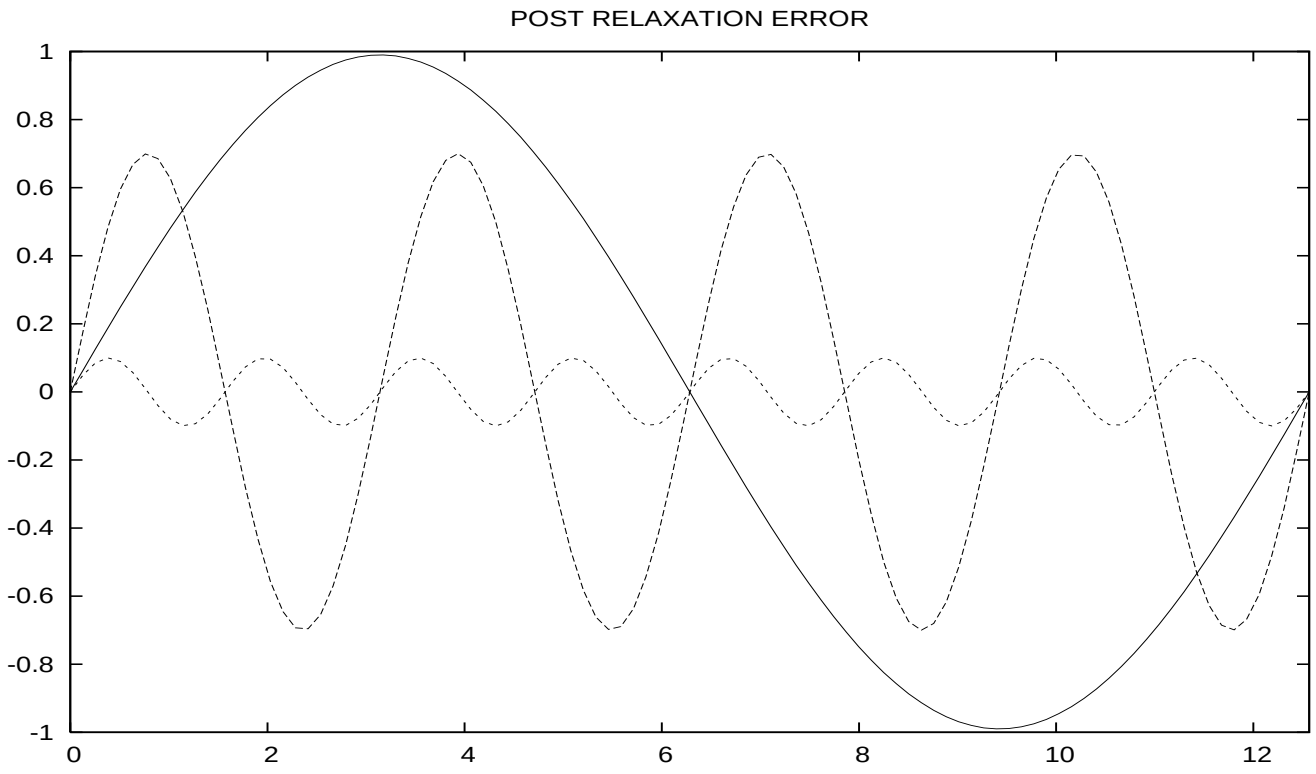
Convergence rate



Different error types: Before Relaxation



Different error types: After Relaxation



Error Fourier Analysis

Consider

$$Lu = a\frac{\partial^2 u}{\partial x^2} + b\frac{\partial^2 u}{\partial y^2} = f$$

Then in the discrete form:

$$a\frac{u_{j-1,k} + u_{j+1,k} - 2u_{j,k}}{h^2} + b\frac{u_{j,k-1} + u_{j,k+1} - 2u_{j,k}}{h^2} = f_{j,k}$$

Solving using the lexicographical Gauss-Seidel relaxation:

$$a\frac{u^{n+1}_{j-1,k} + u^{n}_{j+1,k} - 2u^{n+1}_{j,k}}{h^2} + b\frac{u^{n+1}_{j,k-1} + u^{n}_{j,k+1} - 2u^{n+1}_{j,k}}{h^2} = f_{j,k}$$

Error Fourier Analysis

For the errors e^n and e^{n+1}, true

$$a\frac{e^{n+1}_{j-1,k} + e^{n}_{j+1,k} - 2e^{n+1}_{j,k}}{h^2} + b\frac{e^{n+1}_{j,k-1} + e^{n}_{j,k+1} - 2e^{n+1}_{j,k}}{h^2} = 0.$$

Consider each error in the Fourier form:

$$e^n = \sum C^n_{\underline{\theta}} e^{i(\theta_1 j + \theta_2 k)}, \qquad e^{n+1} = \sum C^{n+1}_{\underline{\theta}} e^{i(\theta_1 j + \theta_2 k)},$$

where $\underline{\theta} = (\theta_1, \theta_2)$, and $|\theta_1|, |\theta_2| < \pi$.

(Remark: for $e^{i(\omega_1 x + \omega_2 y)}$, $\theta_j = \omega_j h$)

Amplification factor

Amplification parameter (the rate of change in amplitude):

$$\mu(\underline{\theta}) = \frac{|C_{\underline{\theta}}^{n+1}|}{|C_{\underline{\theta}}^{n}|} = \frac{|ae^{i\theta_1} + be^{i\theta_2}|}{|2a + 2b - ae^{-i\theta_1} - be^{-i\theta_2}|}$$

For which (θ_1, θ_2), μ is small ?

If $a = b$, for instance, for $\pi/2 \leq |\underline{\theta}| \leq \pi$ true that $\mu \leq .5$

Meanwhile $\mu(\underline{\theta}) \to 1$ as $(\theta_1, \theta_2) \to (0, 0)$.

Does not always work...

For instance, if $a \ll b$, for GS $\mu(\pi/2, 0) \to 1$ as $a \to 0$.

Use line relaxation instead:

$$\mu(\underline{\theta}) = \frac{|a|}{|2(a + b - b\cos\,\theta_2) - ae^{-i\theta_1}|}$$

And then for $\pi/2 \leq |\underline{\theta}| \leq \pi$ again $\mu \leq \max\{5^{-1/2}, \frac{a}{a+2b}\}$

Summary for the Laplace operator

• Important reduction: $\pi/2 \leq |\theta| \leq \pi$. These components cannot be approximated on the coarse grid $(2h)$.
Introducing *smoothing factor* $\overline{\mu}$:

$$\overline{\mu} = \min_{\pi/2 \leq |\theta| \leq \pi} \mu(\theta)$$

• Slow(er) convergence $0 \leq |\theta| \leq \pi/2$. These components can be represented on the coarser grid. For instance, there

$$[\pi/4, \pi/2] \rightarrow [\pi/2, \pi]$$

i.e., will be reduced by $\overline{\mu}$

From now on, the role of relaxation is to *"smooth"*, i.e., reduce error components that cannot be approximated and reduced on the coarser grid(s).

Multigrid Methods

Instead of annihilating error at grid h just smooth it (inexpensively) and then let the coarse grid(s) do the rest.

Every Fourier error component is oscillatory on some grid H and can be reduced there efficiently

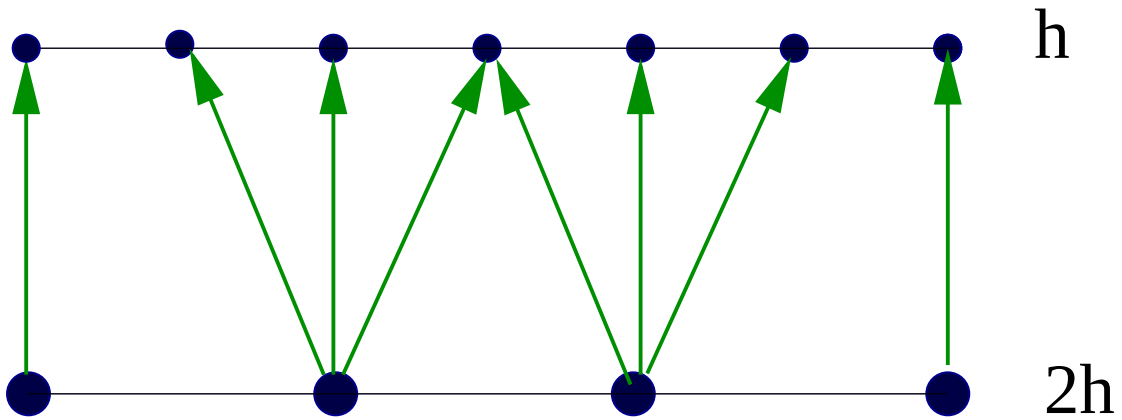
OR

it is so smooth that can be represented on the coarsest grid with few unknowns where direct solver is OK.

This is of course with proper communication between grids.

Prolongation Operator

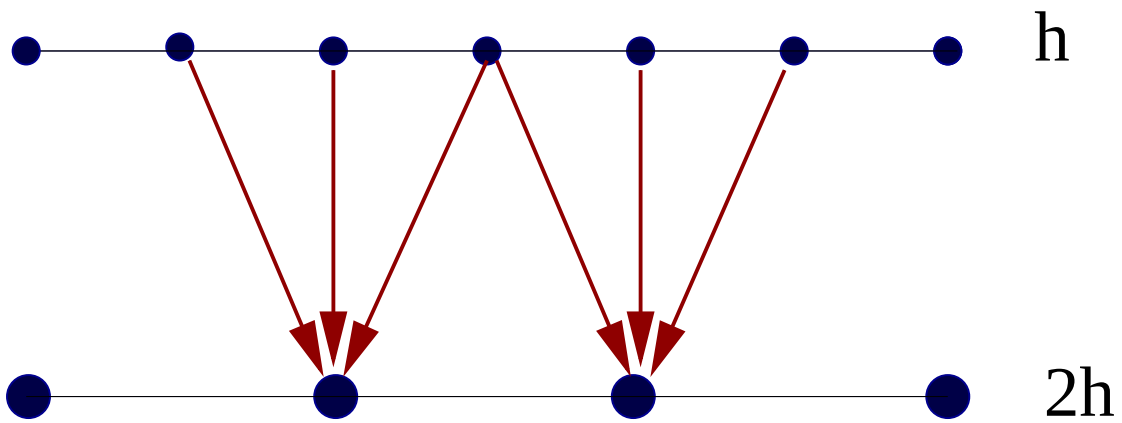
Typical P : polynomial interpolation, i.e., linear or cubic.



P = (1/2 1 1/2)

Restriction Operator

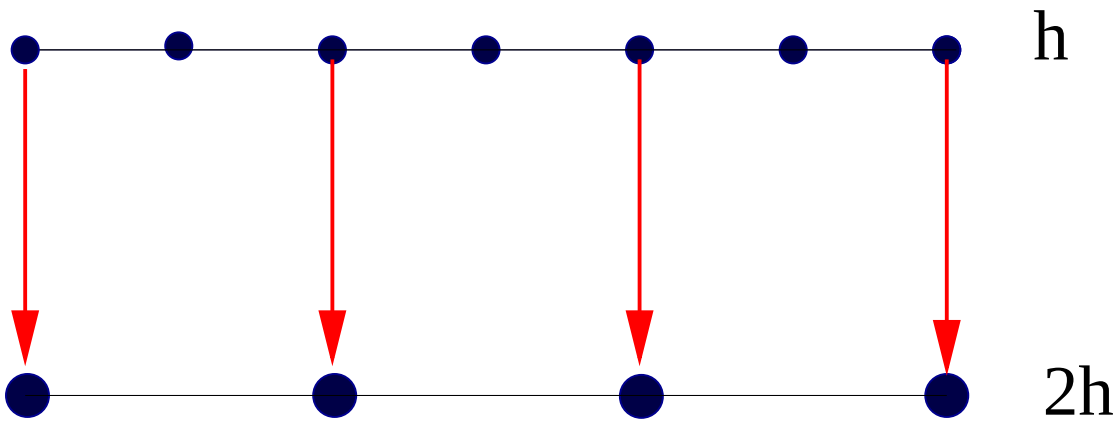
Typical R : full weighting.



R = (1/4 1/2 1/4)

Injection Operator

Injection J:



Two grid cycle
Correction Scheme (Brandt, 1971)

On the fine grid, h:

Relax ν_1 times $L^h u^h = f^h$

Calculate residual: $r^h = f^h - L^h u^h$

Restrict residual to the next grid $r^{2h} = R r^h$

On the coarse grid, $2h$:

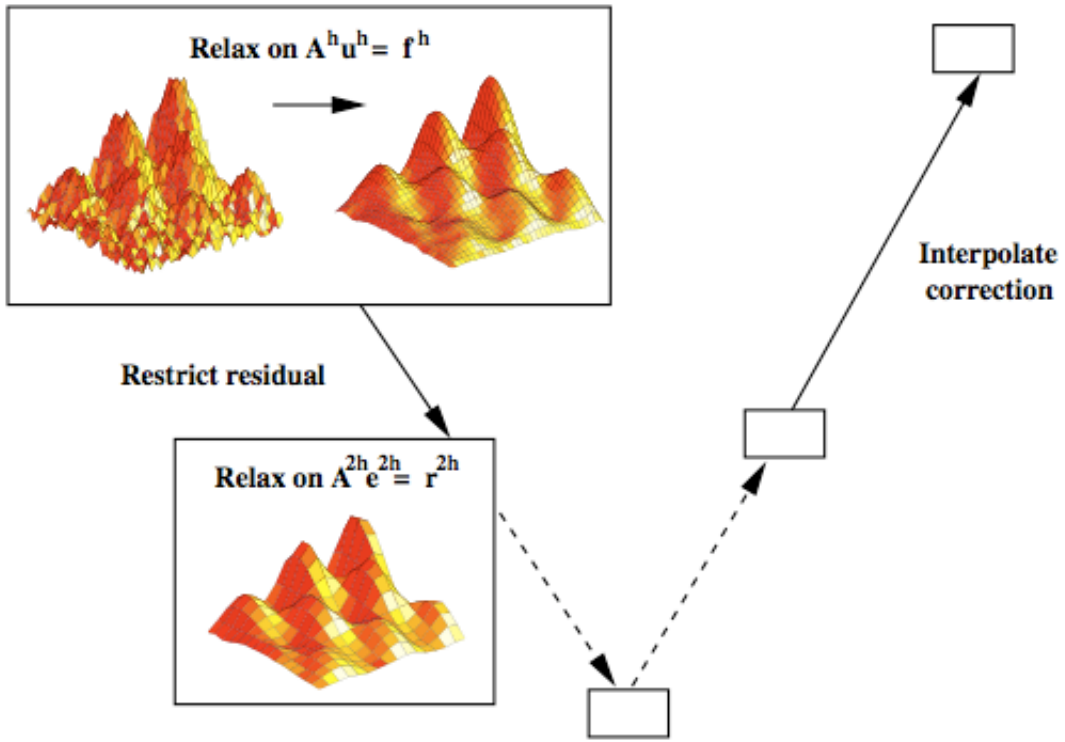
Solve $L^{2h} e^{2h} = r^{2h}$

Prolongate correction to the fine grid: $u^h \leftarrow u^h + P e^{2h}$

On the fine grid , h:

Relax ν_2 times $L^h u^h = f^h$

Geometric Multigrid (MG)



CS vs FAS
Full Approximation Scheme (1975)

CS works only for linear L:

$$L^h(u^h + e^h) = L^h u^h + L^h e^h$$

If L is not linear then instead of a correction, the *entire solution* should be represented on each grid with the residual equation being

$$L^h(u^h + e^h) - L^h u^h = r^h.$$

Warning: The coarse grid treatment should not affect part of the solution that is not represented there.

τ Correction

To guarantee the accuracy of the coarse-grid operators instead of solving

$$L^{2h}u^{2h} = f^{2h} = Rf^h$$

add τ correction to the right-hand-side:

$$L^{2h}u^{2h} = f^{2h} + \tau_h^{2h}$$

where

$$\tau_h^{2h} = L^{2h}(Ju^h) - R(L^h u^h)$$

τ is called *fine-to-coarse defect correction*.

Multigrid V-cycle $V(\nu_1, \nu_2)$
Full Approximation Scheme Scheme

From the finest to the coarsest grid

`for` $m = M, \ldots, 2$

Relax ν_1 times $L^m u^m = f^m$

Initial coarse grid approximation: $u^{m-1} = J u^m$

Restrict residual $f^{m-1} = R(f^m - L^m u^m) + L^{m-1} u^{m-1}$

`end for`

Coarsest grid $m = 1$

Solve $L^m u^m = f^m$

From the coarsest to the finest grid

`for` $m = 2, \ldots M$

Coarse grid correction $u^m = u^m + P(u^{m-1} - J u^m)$

Relax ν_2 times $L^m u^m = f^m$

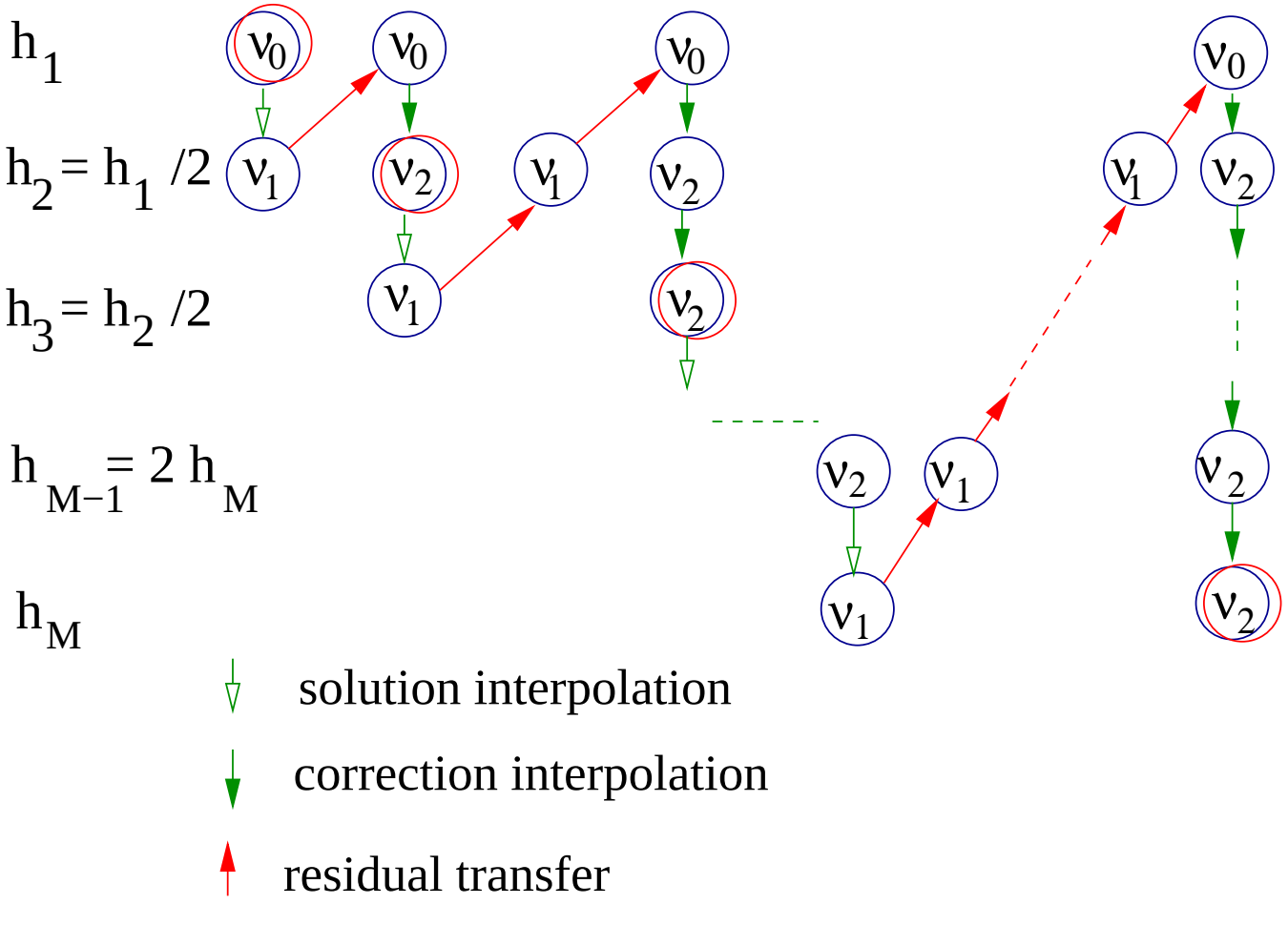
`end for`

Work

Work per cycle is is $O(n)$ for both CS and FAS

n is the size of the finest grid.

Full Multigrid (FMG)



FMG
Brandt (1981)

- Each sub-cycle solves the problem upto the truncation error and then accurately interpolates solution approximation (very good quality) to the new finest grid.
- Truncation error there is only four times smaller, one V-cycle easily makes such reduction.
- After the FMG is done, the approximate solution is accurate upto the truncation error on the finest grid.

Differential eigenvalue problems and FMG
Brandt, McCormick, Ruge (1983)
Hackbush (1980)

Approach:

- Treat $Ax = \lambda x$ as a nonlinear problem (FAS)

- Heavily use smoothness of eigenvalues and global nature of λ: a lot of good comes from very coarse grids (FMG).

- Work: $O(n)$ for the first eigenvalue
$O(q^2 n + q^3 \log n)$ for $q > 1$ eigenvalues. The increase is due to orthogonalization and Ritz projection.

More on convergence: Relation of convergence rate with eigenvalues of A

Assume $\{u_1, \ldots, u_n\}$ are eigenfunctions of$A$ with eigenvalues $\lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n = \lambda_{max}$

Consider error $e^k = \sum e_i^k u_i$ (the residual is then $r^k = \sum \lambda_i e_i^k u_i$). Thus, the after one relaxation sweep,

$$e_i^{k+1} \leftarrow (1 - O(|\lambda_i|/|\lambda_{max}|)e_i^k$$

Remark on the side: Helmholtz operator
Brandt, Livshits (1997)

$$Lu = \Delta u + k^2 u = f(x)$$

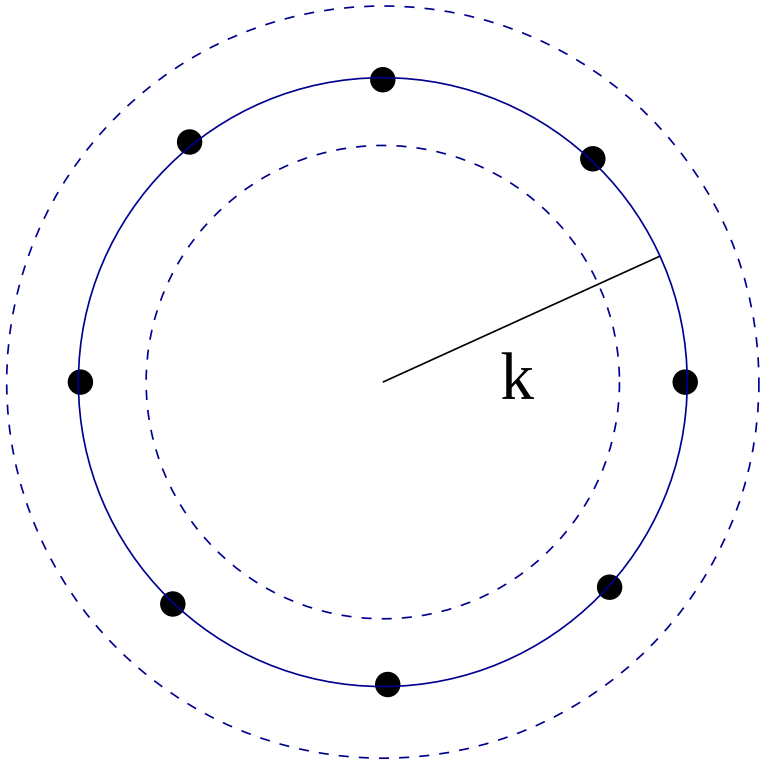
Null space: $exp(i(k_1 x + k_2 y)),\ k_1^2 + k_2^2 = k^2$

Slow to converge error components $Le \approx 0$:

$$exp(i(\omega_1 x + \omega_2 y)), \quad \omega_1^2 + \omega_2^2 \approx k^2$$

They are not smooth anymore: Cannot send them to the coarsest grids and deal with them there just yet!

Phase space



Basic functions: $u_\ell = exp(i(k_1^\ell x + k_2^\ell y))$

Separation

Unreduced error:

$$e(x) = \sum_{\ell} e_{\ell}(x) u_{\ell}(x)$$

Residual:

$$r(x) = \sum_{\ell} r_{\ell}(x) u_{\ell}(x)$$

Instead of finding oscillatory $e(x)$ approximate smooth weights $e_{\ell}(x)$

Helmholtz operator: finding $e_\ell(x)$

Substitute the error:

$$Le(x) = \sum_\ell u_\ell L_\ell e^\ell = \sum_\ell u_\ell r_\ell$$

Separate:$r(x)$ into r_ℓ and solve separately

$$L_\ell e_\ell = r_\ell$$

and then reconstruct back

$$e = \sum_\ell e_\ell$$

.

Algebraic Multigrid (AMG)
Brandt, McCormick, Ruge (1984)
Ruge, Stüben (1987)

Consider a system of linear equations:

$$Ax = b$$

Assume here that A is an M-matrix: symmetric, positive definite, $a_i i > 0$, $a_{ij} < 0, i \neq j$.

Coarse grid operator

1. Geometric MG (for PDEs) : all operators approximate the PDE.

2. Algebraic MG: each next coarse operator approximates the previous fine one

   (a) unstructured grids;

   (b) no PDE at all: matrices

   (c) discontinuous coefficients

AMG components

Coarse grid operator is no more a mere discretization of PDE!

- Coarse grid variables
- Prolongation operator: P, from coarse to fine, replaces fine grid values with the coarse grid values;
- Restriction operator: R, from fine to coarse, averages fine grid equations;
- Coarse grid operator A_c (fine grid operator A is given)

Coarse grid operator is given by Galerkin procedure

$$A_c = RAP, R = P^t$$

$R = P^T$ comes from variational considerations.

Algebraic smoothness

Which error components should be represented on the coarse grid(s) and accurately interpolated from it ?
The ones that are not reduced by relaxation on the fine grid.

Such error components are called *algebraically smooth* (vs *physically smooth*), and they satisfy

$$Ae \approx 0$$

BTW

$$a_{ii}e_i \approx -\sum_{i \neq j} a_{ij}e_j$$

gives a hint to interpolation...

Desired properties

◇The goal of the coarse grid is to accurately resolve the smooth components.

◇ The goal of the prolongation operator is to accurately transfer them back to the fine grid.

Strong dependence and strong influence

Variable u_i strongly depends on variable u_j if

$$-a_{ij} \geq \theta \max_{k \neq i} \{-a_{ik}\}, \quad 0 < \theta \leq 1$$

The variable u_j is then said to *strongly influence* u_i.

Makes sense if the u_i is strongly influenced by u_j then it can be approximated using u_j.

Interpolation

Consider all fine grid points $\{1, 2, \ldots, n\} = \mathbf{F} \bigcup \mathbf{C}$

$$\sum_{j \neq i} \frac{|a_{ij}|}{a_{ii}} \frac{(e_i - e_j)^2}{e_i^2} \ll 1, \quad 1 \leq i \leq n$$

If $\mathbf{C}$-point j strongly influences an $\mathbf{F}$-point i then $e_i \approx e_j$.

The error is smooth in the direction of strong connections and can be represented by the coarse grid.

Coarse grid variables

Consider a set of the fine grid variables $\mathbf{F} = \{1, \ldots, N\}$.

The set of coarse grid points $\mathbf{C} \subset \mathbf{F}$ should satisfy:

• For each point $i \in \mathbf{F}$, every point $j \in S_i, \quad (a_{ij} \neq 0)$, that strongly influences i should be either in C or should strongly depend on at least one point in $\mathbf{C}$.

• The set of $\mathbf{C}$ points should be a maximal subset of all points with the property that no $\mathbf{C}$-point strongly depends on another $\mathbf{C}$ point

Interpolation

$$(Pe)_i = \begin{cases} e_i, & i \in \mathbf{C} \\ \sum_{j \in C_i} \omega_{ij} e_j, & i \in \mathbf{F} \end{cases}$$

The coefficients come from the error equation:

$$a_{ii} e_i \approx - \sum_{i \neq j} a_{ij} e_j$$

$$\omega_{ij} = (a_{ij} + \sum_{m \in D_i^s} \{a_{im} a_{jm} / \sum_{k \in C_i} a_{mk}\}) / (a_{ii} + \sum_{n \in D_i^w} a_{in})$$

More of AMG

Works nicely for PDE problems with structured grids: benefits from using *smart* prolongation operators that accommodate *algebraically smooth* error components (near null space).

This is important when the near null space is physically oscillatory.

Problem in question

$$-\Delta u(x) + V(x)u(x) = \lambda u(x), \qquad x \in \Omega \subset \mathbf{R}^d, \lambda \in \Lambda$$

Both $V(x)$ and $u(x)$ are periodic in Ω.

What we want:

Find all $\lambda \in \Lambda$ and all associated eigenfunctions using MEB (Multiscale EigenBasis Structure)

More concerned with $\lambda - V(x) \gg 0$.

Motivation for MEB

- MEB: Very limited description on the finest grid AND accurate and full description of all eigenfunctions on the coarsest grid.

- The cost of just storing all eigenfunctions on the finest grid is $O(N^2)$.

- We want everything (calculation, storing, applications) in $O(N \log N)$ operations.

MEB was introduced by Livne and Brandt for solving 1d Schrödinger; employed *monodromies*.

Strategy

On each grid we need operators (including prolongation) that will describe all eigenfunctions (their representatives)!

If a regular multigrid structure is used this does not happen for the following reasons:

Discrete Operators

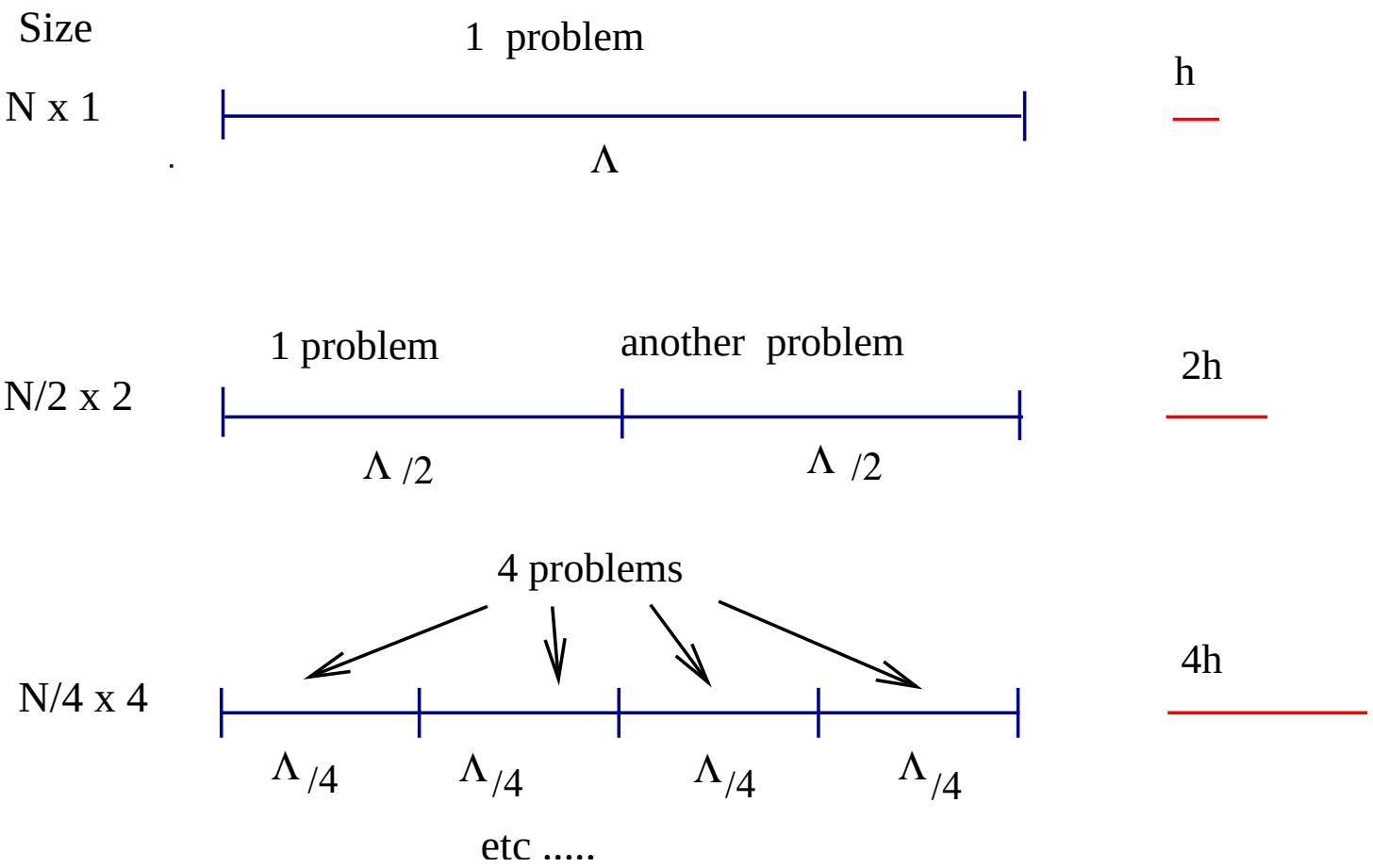
- Only PDE operator equally well describes all eigenfunctions;
- For a given discretization scale, any discrete PDE and any prolongation works well (if at all) only for a limited range of eigenfunctions;
- As the scale increases this range shrinks.

Moreover..

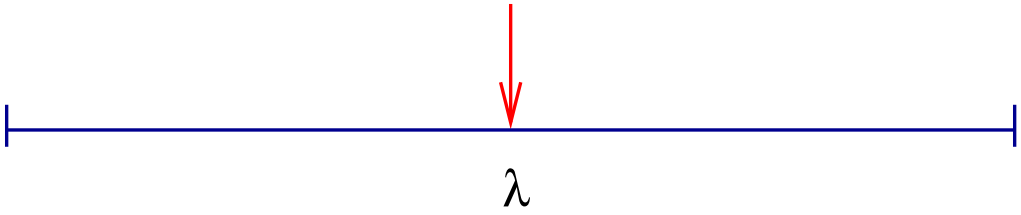
- System with N unknowns can have at most N eigenfunctions
- On the next scale ($N/2^d$ unknowns) – at most $N/2^d$ eigenfunctions..
- ...

Possible strategy: On scale with $N/2^d$ unknowns use 2^d systems of size $N/2^d$, or 2^{d-1} systems of size $2N/2^d$, or etc.

MEB

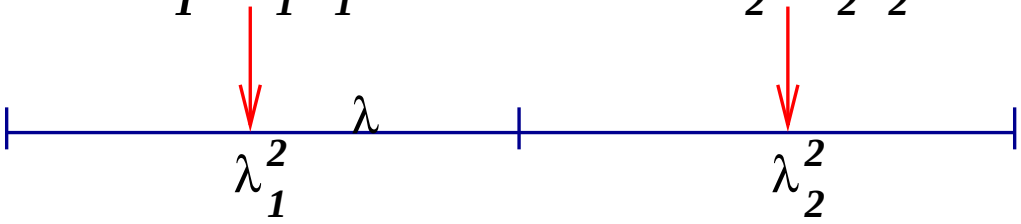

Size
1 problem
h
N x 1
Λ
1 problem
another problem
2h
N/2 x 2
Λ /2
Λ /2
4 problems
4h
N/4 x 4
Λ /4
Λ /4
Λ /4
Λ /4
etc .....

SOLVE : $A^1 u^1 = \lambda^1 u^1$



FIND A^2 **using** A^1 **and** u^1

SOLVE : $A^2 u_1^2 = \lambda_1^2 u_1^2$ $A^2 u_2^2 = \lambda_2^2 u_2^2$



FIND A_1^3 **using** A^2 **and** u_1^2 etc

Exact Interpolation Scheme

P accurately interpolates $\widehat{u}^\ell \equiv u^1$ and its vicinity on the next scale, $\ell + 1$. For any $u^{\ell+1}$

$$Pu^{\ell+1} = \widehat{u}^\ell I u^{\ell+1}$$

where $I \equiv I^\ell_{\ell+1}$ is a polynomial interpolation.

Reasoning:

- $\widehat{u}^\ell$ is represented by a constant on scale $\ell + 1$ and perfectly interpolated from there;

- if $\tilde{u}^\ell = \widehat{u}^\ell u^\ell$ for smooth u^ℓ then $\tilde{u}^\ell \approx Pu^{\ell+1}$ since smooth $u^\ell \approx Iu^{\ell+1}$. As a result $\tilde{u}^\ell$ has a smooth representative $u^{\ell+1} \equiv u^2$ and accurate prolongation.

Local Basis

Define, for each subinterval of Λ and each scale ℓ, a local basis $\{\hat{u}_1^\ell, \ldots, \hat{u}_K^\ell\}$ as follows:

- Each $\hat{u}_k^\ell$ satisfies $A^\ell \hat{u}_k^\ell \approx \lambda_0^\ell M^\ell \hat{u}_k^\ell$
- For any $\tilde{u}^\ell$ such that $A^\ell \tilde{u}^\ell = \lambda^\ell \tilde{u}^\ell$, $\lambda^\ell \approx \lambda_0^\ell$: $\tilde{u}^\ell = \sum \hat{u}_k^\ell u_k^\ell$
- $\{\hat{u}_1^\ell, \ldots, \hat{u}_K^\ell\}$ is locally orthogonal (i.e., $P_i^* P_j \approx 0$).

Knowing $\{\hat{u}_1^\ell, \ldots, \hat{u}_K^\ell\}$, $(P_1, \ldots, P_K)$ allows an accurate description of $\tilde{u}^\ell$ by the weight functions$\{u_1^\ell, \ldots, u_K^\ell\}$.

New Coarse Operators (Galerkin)

$$A^{\ell+1}u^{\ell+1} = \lambda M^{\ell+1}u^{\ell+1}$$

where

$$A_{ij}^{\ell+1} = R_i A^{\ell} P_j, \quad M_{ij}^{\ell+1} = R_i M^{\ell} P_j$$

and

$$u^{\ell+1} = [u_1^{\ell+1}, \ldots u_K^{\ell+1}]^t$$

This is where orthogonality of the local basis is important – leads to a block-diagonal system

Next step: Find a new local basis... etc

SOLVE: Local V-cycles

On scale ℓ, 2^ℓ problems are to be solved. Each is

$$A^\ell u^\ell = \lambda M^\ell u^\ell$$

for a fixed λ.

Good solver: regular AMG (Galerkin) $V(0,*)$ cycle with linear I. No relaxation on the way to the coarsest scale because looking for the smoothest possible weight functions u.

Local Basis Again (1D)

Consider

$$Au = \lambda_0 Mu$$

Find two (in 1D) approximate solutions (different and "orthogonal") u_1, u_2.

Locally good approximations are $u_{1,2} \approx \exp(\pm i\varphi(x))$,
$d(\varphi(x)) = \sqrt{\max\{0, \lambda_0 - V(x)}$

We need this to discuss our choice of restriction operator R (normally, $R = P^*$).

Differential weight operators

For $V(x) - \lambda_0 < 0$:

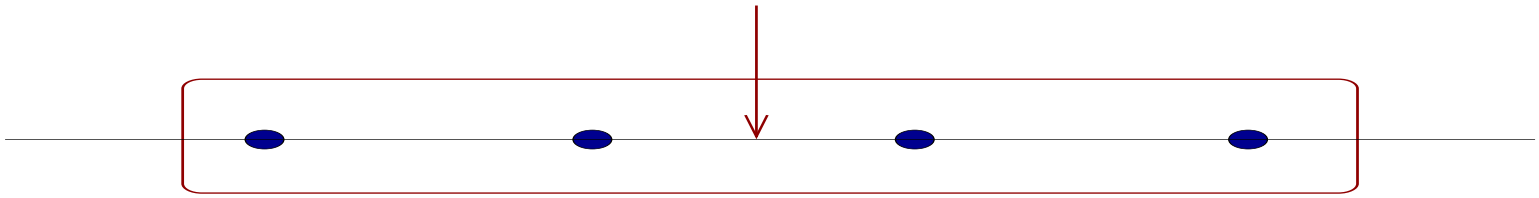
$$(L - \lambda_0)(u(x)e^{i\varphi}) = (-u'' - 2i\varphi' u' - i\varphi'' u)e^{i\varphi}$$

and For $V(x) - \lambda_0 \geq 0$:

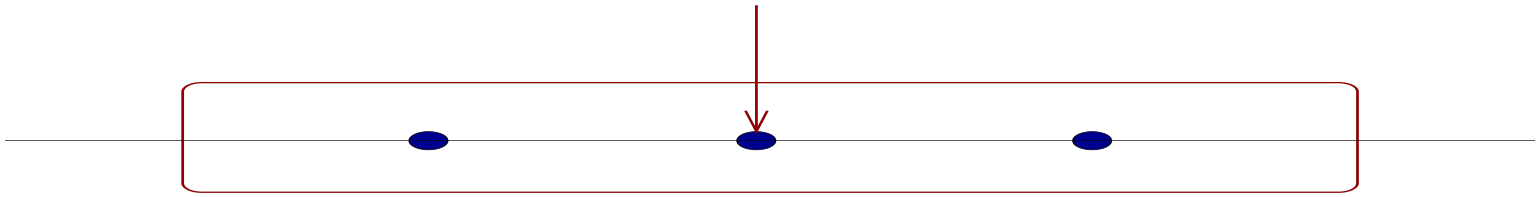
$$-u'' + (V - \lambda_0)u$$

Discretization stensils

$$V(x) - \lambda \ll 0$$

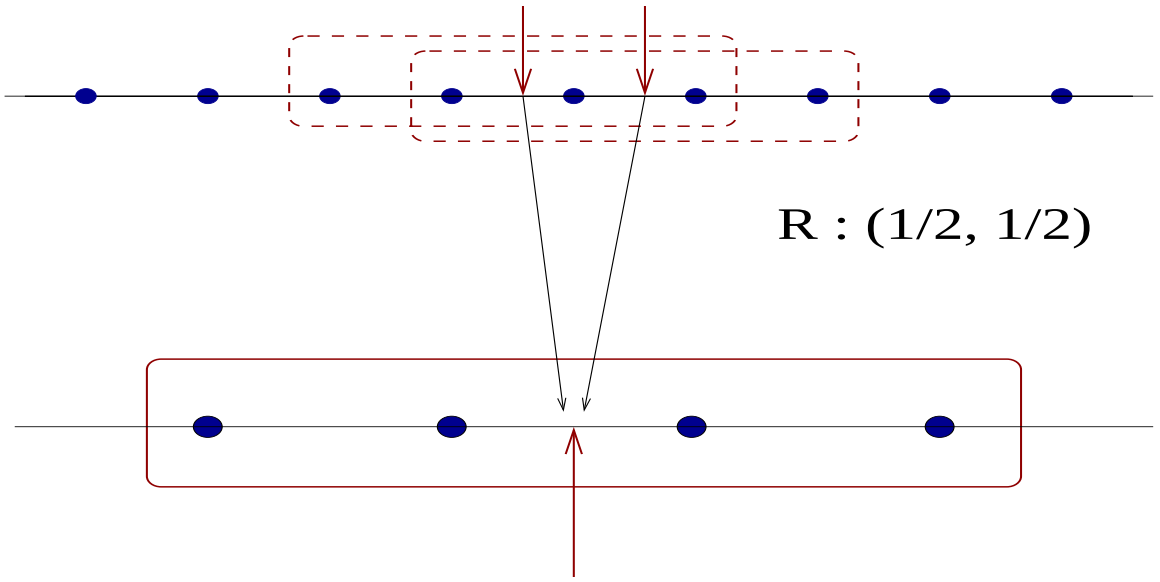


otherwise



Restriction R

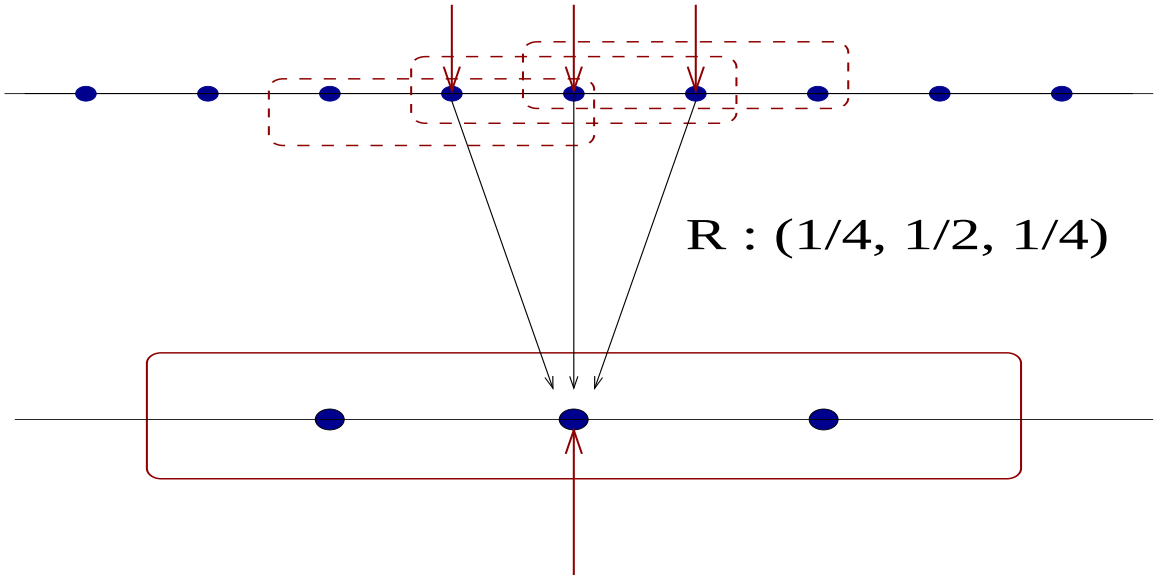
$$V(x) - \lambda^{\ell+1} << 0 \quad \&\& \quad V(x) - \lambda^{\ell} < 0$$



R is chosen to approximate the EIS P^* but with possibly different constant coefficients.

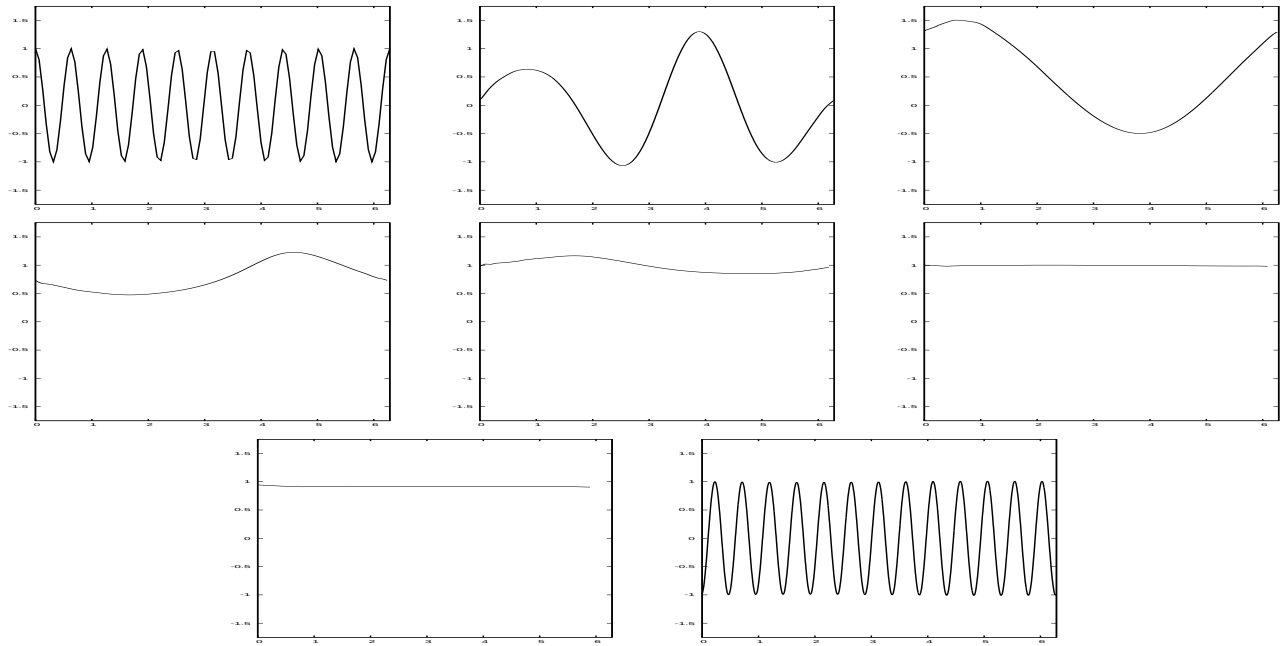
Restriction R

$$V(x) - \lambda^{\ell+1} \geq 0 \quad \&\& \quad V(x) - \lambda^{\ell} \geq 0$$



R is chosen to approximate the EIS P^* but with possibly different constant coefficients.

MEB in Action



$V(x) = -100$. The eigenfunction is associated with $\lambda = 69$. The snapshots are made on the increasingly coarser scales. The final one is the reconstructed (on the finest scale) actual eigenfunction.

Numerical Experiments

M	-36.0	-19.0	0.0	21.0	44.0
6	-36.163015	-19.013837	0.11142759	21.050244	44.148021
7	-35.922581	-18.978212	0.026406604	21.001174	43.944225
8	-35.985884	-18.989143	0.014880117	21.003859	43.987206
9	-35.994773	-18.995245	0.005703412	20.998649	43.996893
10	-35.998341	-18.997613	0.0025906716	20.999942	43.998535
11	-35.999262	-19.000032	0.00070075148	21.000313	44.000205
12	-35.999572	-18.999879	-0.00024166004	21.000453	43.999858

Weight PDEs, $p = 4$, $V = -100$

M	-36.0	-19.0	0.0	21.0	44.0
6	-36.826235	-20.305766	1.1049825	18.091001	39.905914
7	-36.189308	-19.318236	-0.48227638	20.276804	42.966252
8	-36.052675	-19.073732	-0.1101791	20.822695	43.744047
9	-36.011517	-19.016339	-0.02391381	20.953409	43.93638
10	-36.002533	-19.002871	-0.0062959449	20.988634	43.983389
11	-36.000309	-19.001344	-0.0018867208	20.997487	43.996415
12	-35.999834	-19.000207	-0.0007297946	20.999747	43.998911

Schrödinger PDE, $p = 4$, $V = -100$

M	-36.0	-19.0	0.0	21.0	44.0
6	-36.163015	-19.013837	0.11142759	21.050244	44.148021
7	-35.922581	-18.978212	0.026406604	21.001174	43.944225
8	-35.985884	-18.989143	0.014880117	21.003859	43.987206
9	-35.994773	-18.995245	0.005703412	20.998649	43.996893
10	-35.998341	-18.997613	0.0025906716	20.999942	43.998535
11	-35.999262	-19.000032	0.00070075148	21.000313	44.000205
12	-35.999572	-18.999879	-0.00024166004	21.000453	43.999858

Weight PDEs, $p = 4$, $V = -100$

M	-36.0	-19.0	0.0	21.0	44.0
6	-36.80291	-19.043965	1.0138234	21.065505	44.642709
7	-36.191769	-19.005795	0.26487493	21.022192	44.1662
8	-36.044376	-18.999708	-0.055182284	21.006638	44.040704
9	-36.010185	-18.999317	-0.011737716	21.001595	44.010552
10	-36.002994	-18.999711	-0.0030461405	21.00036	44.003102
11	-36.000712	-18.999867	-0.00099222517	21.00013	44.000855
12	-36.000029	-18.999924	-0.00021906591	21.000133	44.000216

Weight PDEs, $p = 2$, $V = -100$

M	-48.363146	-33.930701	-17.461388	1.2359528	21.987031	44.819641
7	-48.407263	-34.035775	-17.483754	1.2825186	22.165257	44.940064
8	-48.171214	-33.975234	-17.310086	1.0639833	21.87618	44.724361
9	-48.042596	-33.910144	-17.319428	0.9688164	21.615584	44.821105
10	-48.284795	-33.917122	-17.421561	1.2828984	22.028707	44.819156
11	-48.16908	-33.912838	-17.405412	1.2366406	22.11619	44.819709
12	-48.206817	-33.900991	-17.468307	1.1645153	22.132768	44.818492

Weight PDEs, $p = 4$, $V(x) = -100 + 30\cos(5x)$.

M	-48.363146	-33.930701	-17.461388	1.2359528	21.987031	44.819641
7	-48.698139	-34.265817	-17.793359	0.70434873	21.25357	43.782616
8	-48.298226	-34.032278	-17.543688	1.0896674	21.80743	44.557441
9	-48.150837	-33.980731	-17.481617	1.1861352	21.943762	44.756828
10	-48.083451	-33.958645	-17.461294	1.2085395	21.976488	44.803775
11	-48.055789	-33.954236	-17.454826	1.2131808	21.984521	44.81683
12	-48.023985	-33.950807	-17.452403	1.2144852	21.987305	44.819376

Schrödinger PDE, $p = 4$, $V(x) = -100 + 30\cos(5x)$.

M	-48.363146	-33.930701	-17.461388	1.2359528	21.987031	44.819641
7	-48.698139	-34.265817	-17.793359	0.70434873	21.25357	43.782616
8	-48.298226	-34.032278	-17.543688	1.0896674	21.80743	44.557441
9	-48.150837	-33.980731	-17.481617	1.1861352	21.943762	44.756828
10	-48.083451	-33.958645	-17.461294	1.2085395	21.976488	44.803775
11	-48.055789	-33.954236	-17.454826	1.2131808	21.984521	44.81683
12	-48.023985	-33.950807	-17.452403	1.2144852	21.987305	44.819376

Schrödinger PDE, $p = 4$, $V(x) = -100 + 30\cos(5x)$.

M	-48.363146	-33.930701	-17.461388	1.2359528	21.987031	44.819641
7	-49.261933	-34.396099	-17.827294	0.47309162	21.257366	43.840523
8	-48.609076	-34.086605	-17.554719	1.0312624	21.808048	44.574333
9	-48.280272	-33.996475	-17.483958	1.1684235	21.942334	44.759816
10	-48.115929	-33.964581	-17.466092	1.2023517	21.976503	44.805739
11	-47.992249	-33.947184	-17.460468	1.2115868	21.985157	44.817029
12	-47.887103	-33.932205	-17.458074	1.2143396	21.987333	44.819889

Schrödinger PDE, $p = 2$, $V(x) = -100 + 30\cos(5x)$

Current Developments: 2D and more (with Achi Brandt)

- $2D$ local basis is tricky: involves many directions and eigenfunctions travel between them;

- Just doubling the number of eigenintervals does not help: $N/4 \times 2 \neq N$, need to increase the size of local basis;

- Focus: Robust procedure of finding exhausting sets of prolongation operators that accommodate all fine scale eigenfunctions and use them to build coarse scale operators;

- An extra benefit – a smarter (adaptive) division of eigenintervals.

- This approach is not limited to the Schrödinger equation.