Numerical Linear algebra

Multigrid methods

Martin van Gijzen

December 10, 2008

Program december 10

Multigrid

- Basic preconditioners
- Coarse grid correction
- What we didn’t talk about

Introduction

Multigrid is a solution technique for linear systems from discretised elliptic equations.

Multigrid methods are quite powerful for classes of problems, in particular for Poisson problems the complexity is of optimal order n.

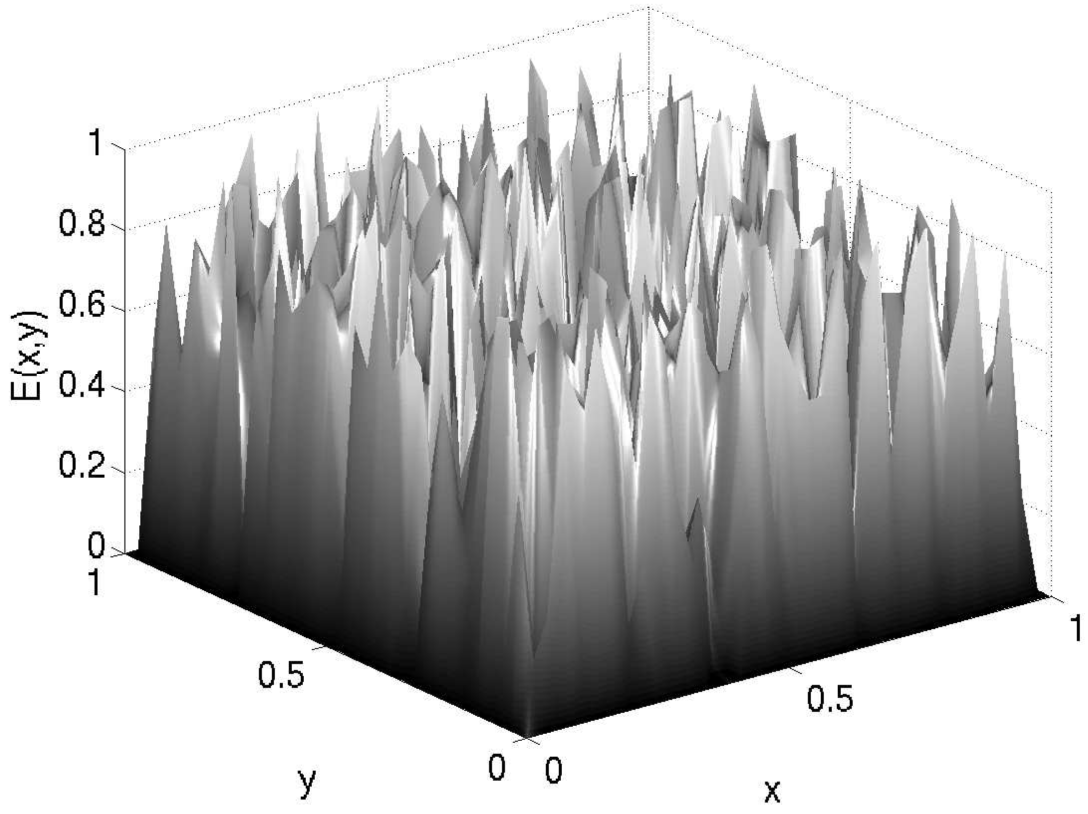
The theory on multigrid methods is vast. Today we only give a short introduction and the main ideas.

Basic preconditioners

In lesson 10 we discussed some basic preconditioners: Jacobi, Gauss-Seidel, SSOR, and ILU. Although the preconditioners discussed before can considerably reduce the number of iterations, they do normally not reduce the mesh-dependency of the number of iterations.

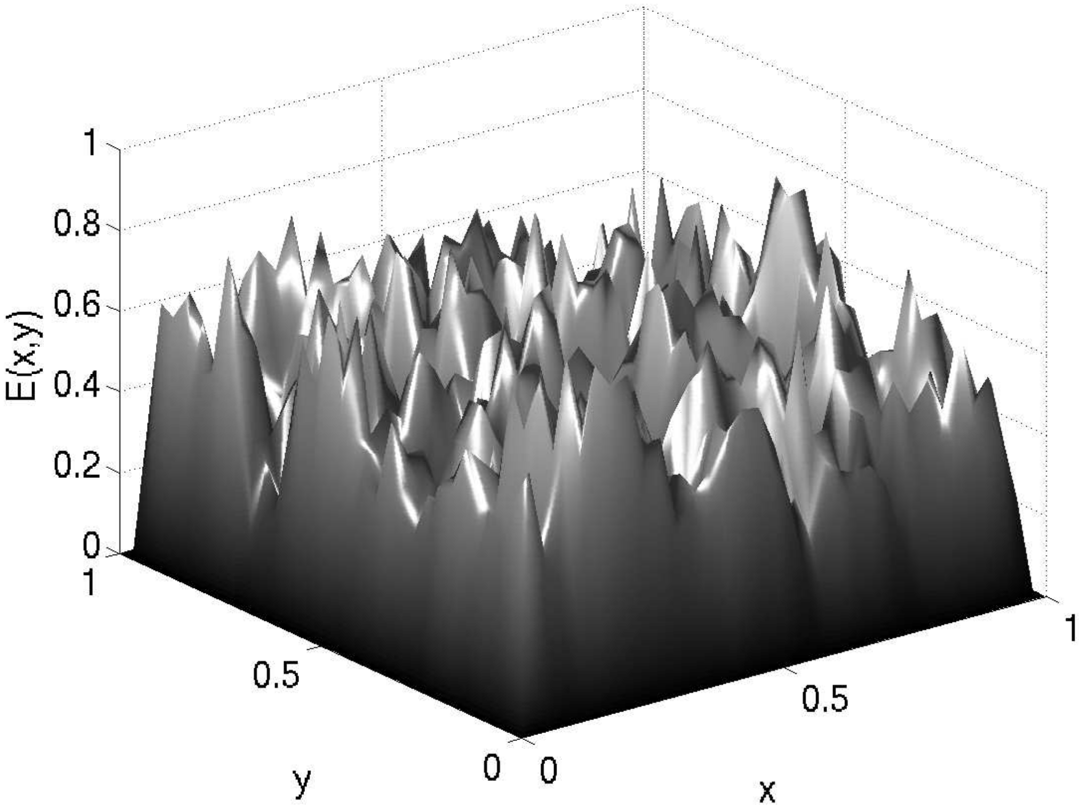
In the next slides we take a closer look at how basic iterative methods reduce the error. From the observations we make, we will develop the idea that is at the basis of one of the fastest techniques: multigrid.

Smoothing Property



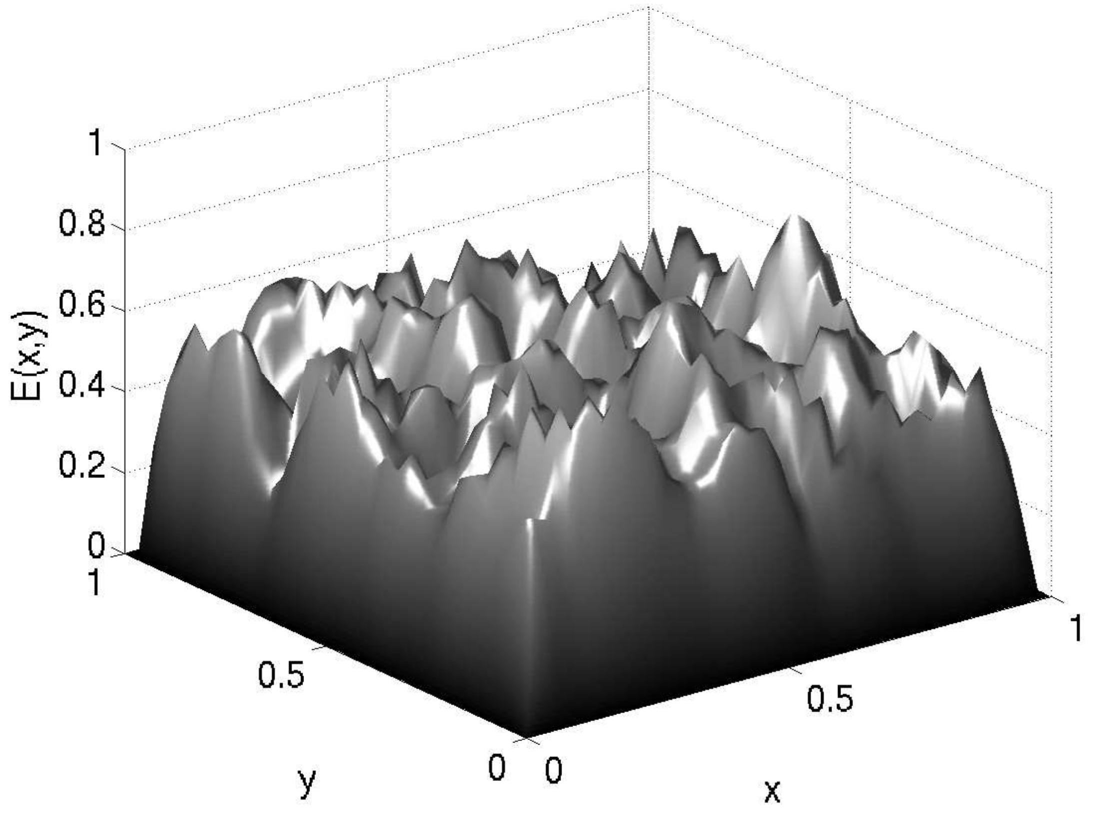
Random initial error

Smoothing Property



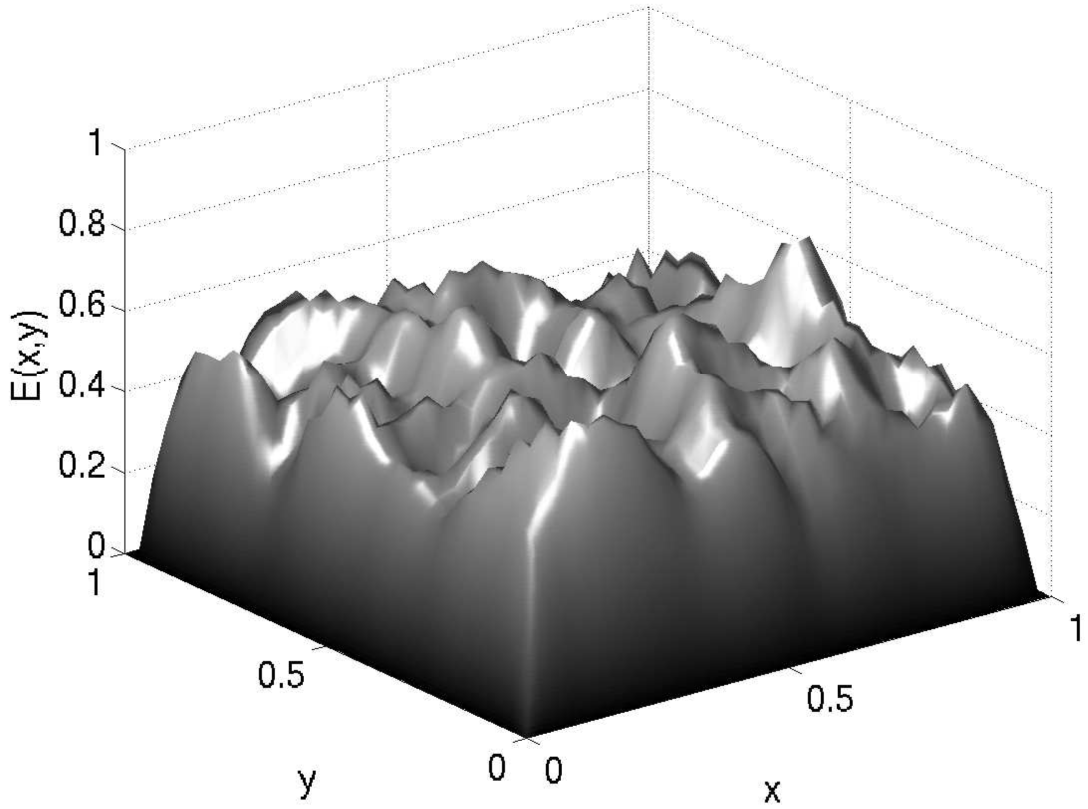
Error after 1 Jacobi iterations

Smoothing Property



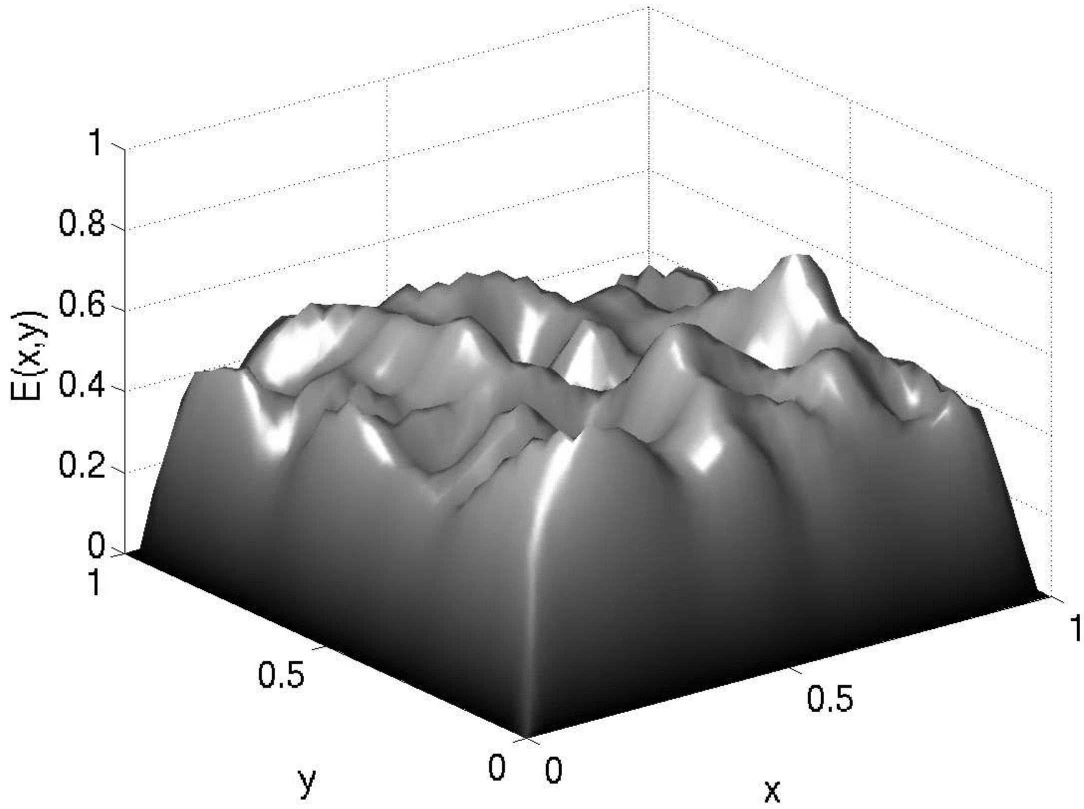
Error after 2 Jacobi iterations

Smoothing Property



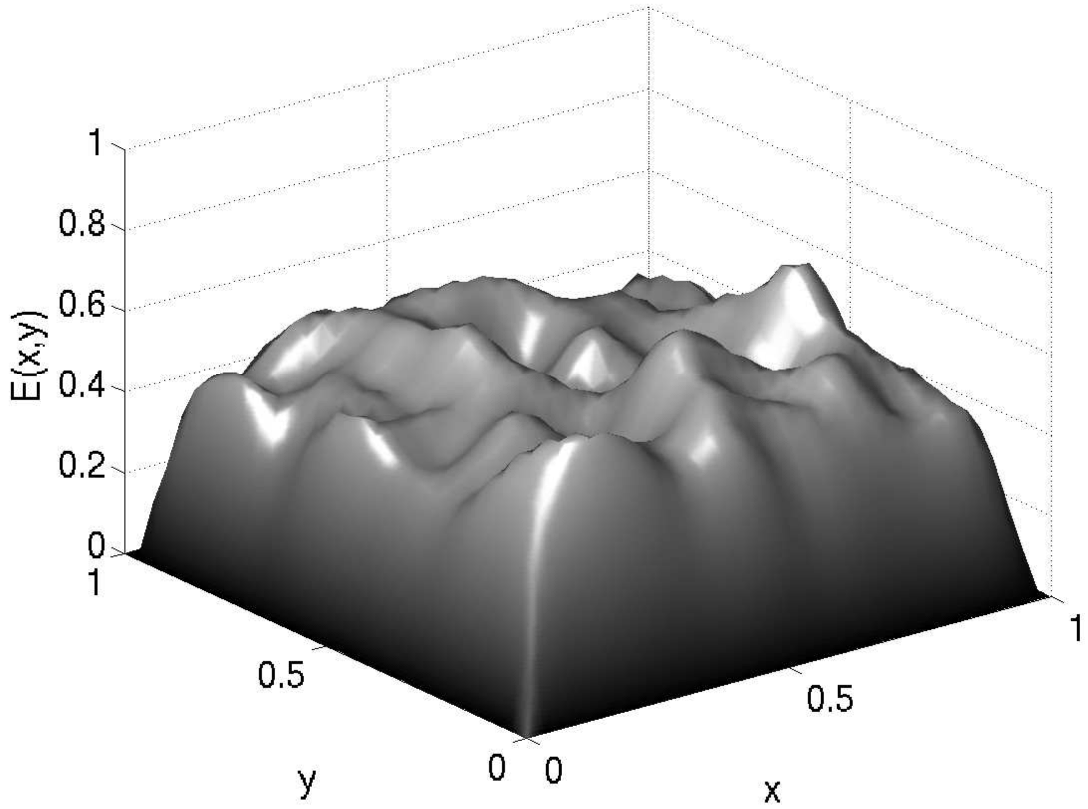
Error after 3 Jacobi iterations

Smoothing Property



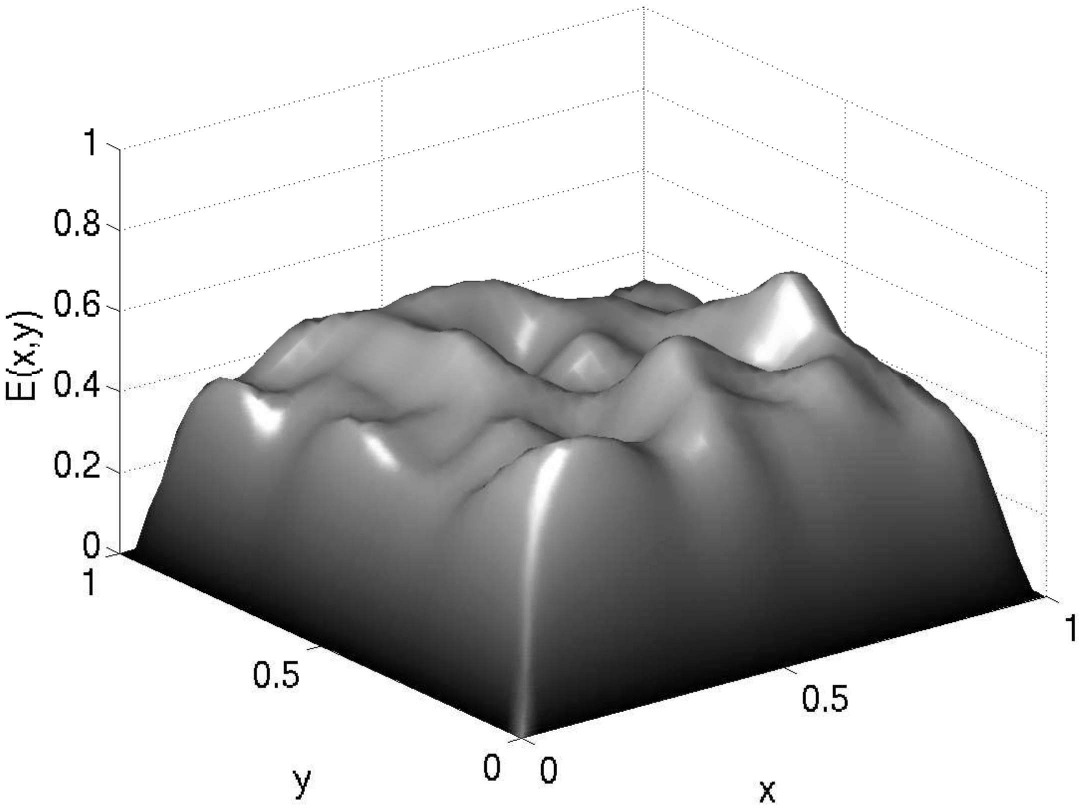
Error after 4 Jacobi iterations

Smoothing Property



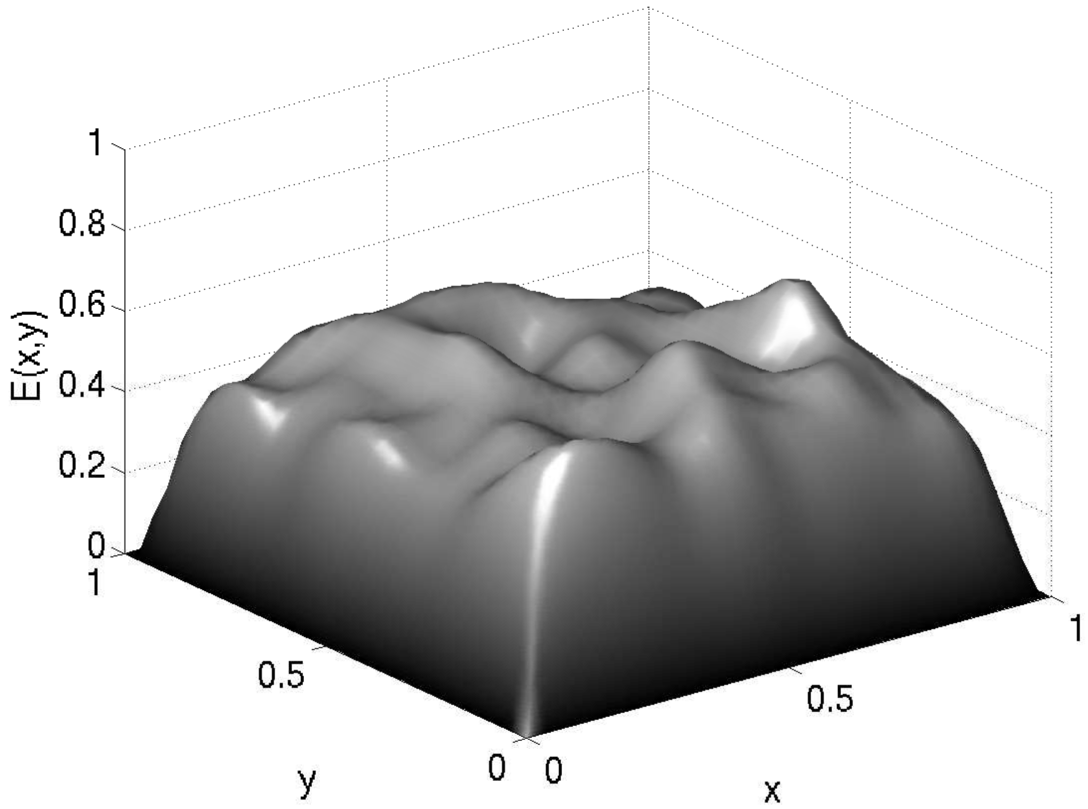
Error after 5 Jacobi iterations

Smoothing Property



Error after 6 Jacobi iterations

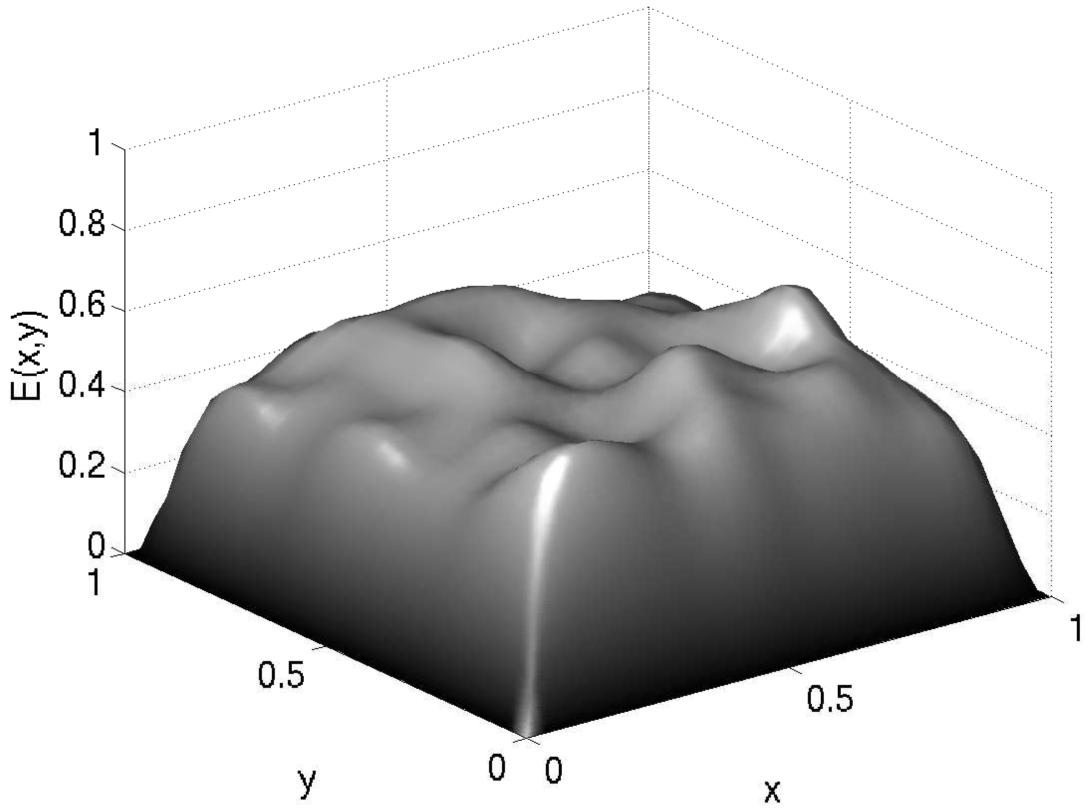
Smoothing Property



Error after 7 Jacobi iterations

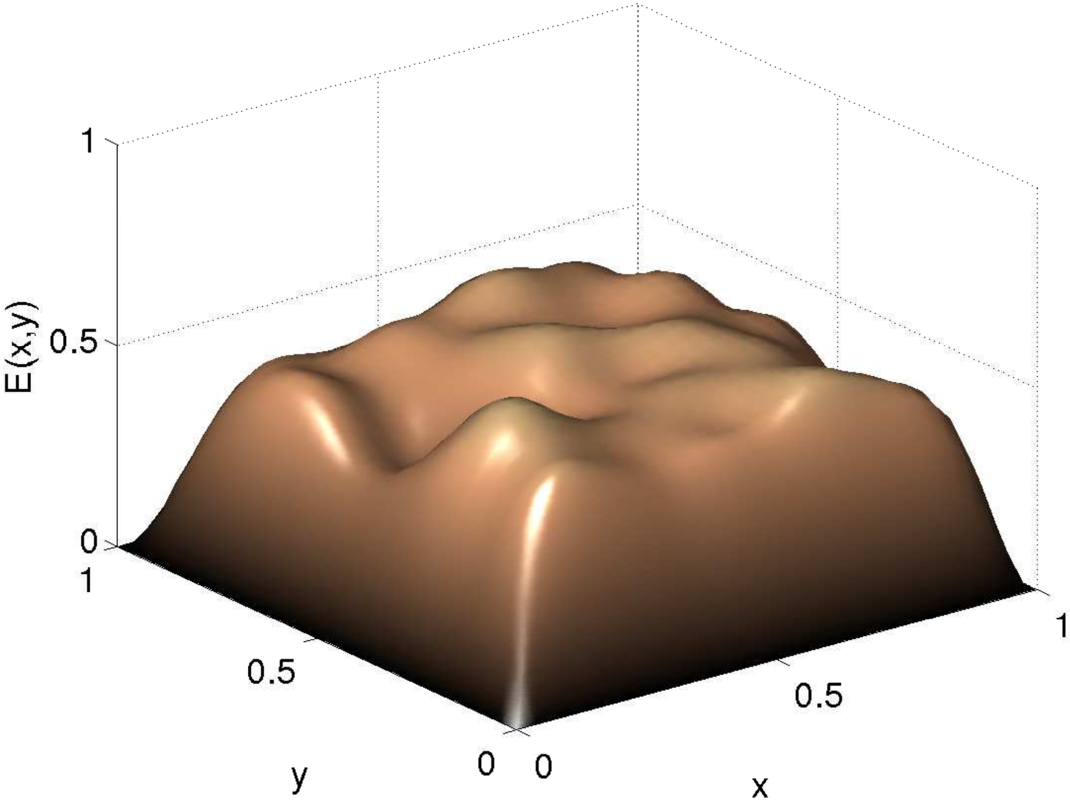
TU Delft

Smoothing Property



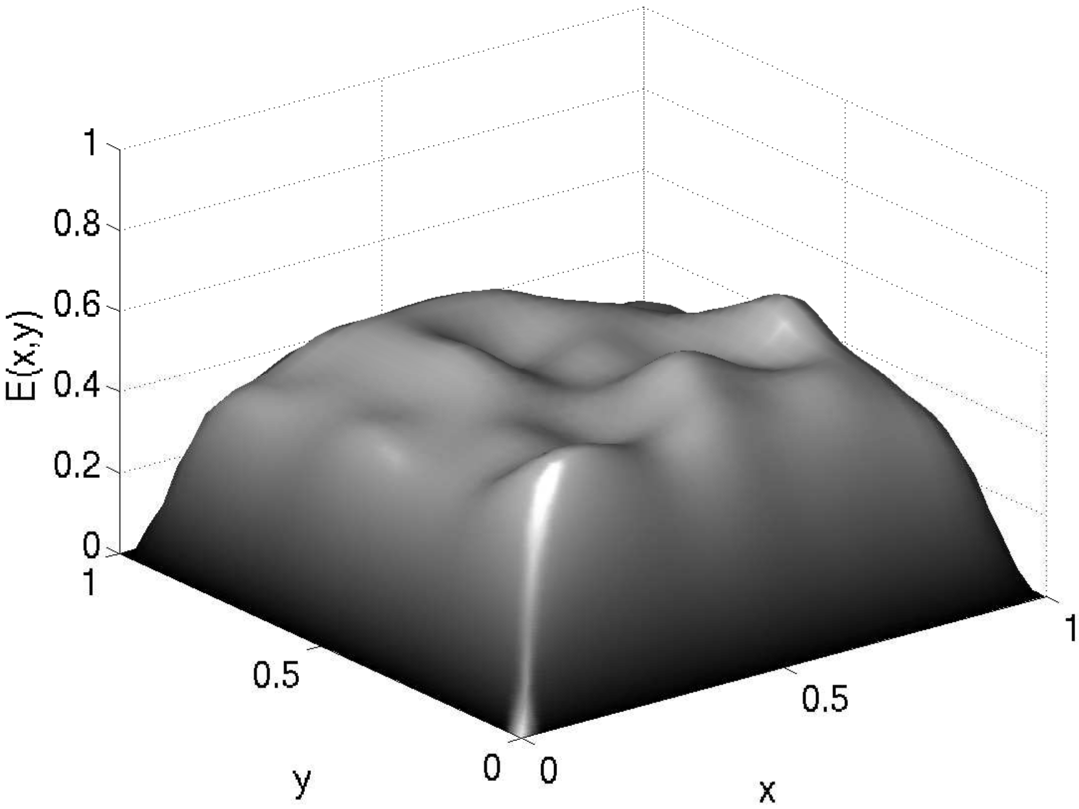
Error after 8 Jacobi iterations

Smoothing Property



Error after 9 Jacobi iterations

Smoothing Property



Error after 10 Jacobi iterations

TUDelft

Complementarity

- Error after a few Jacobi iterations has structure, this is the same for the other basic iterative methods.
- Instead of discarding the method, look to complement its failings

How can we best correct errors that are slowly reduced by basic iterative method?

Complementarity

- Error after a few Jacobi iterations has structure, this is the same for the other basic iterative methods.
- Instead of discarding the method, look to complement its failings

How can we best correct errors that are slowly reduced by basic iterative method?

- Slow-to-converge errors are smooth
- Smooth vectors can be accurately represented using fewer degrees of freedom

Coarse-Grid Correction

- Smooth vectors can be accurately represented using fewer degrees of freedom
- Idea: transfer job of resolving smooth components to a coarser grid version of the problem
- Need:
  - Complementary process for resolving smooth components of the error on the coarse grid
  - Way to combine the results of the two processes

Multigrid

- Relaxation is the name for applying one or a few basic iteration steps.
- Idea is to correct the approximation after relaxation, $x^{(1)}$, from a coarse-grid version of the problem
- Need interpolation map, P, from coarse grid to fine grid
- Corrected approximation will be $x^{(2)} = x^{(1)} + Px_c$
- x_c is the solution of the coarse-grid problem and satisfies $(P^TAP)x_c = P^TA(x - x^{(1)}) = P^Tr^{(1)}$

Two-grid cycle

Two-grid cycle

Multigrid Components

- Relaxation

Relax: $x^{(1)} = x^{(0)} + M^{-1} r^{(0)}$

- Use a smoothing process (such as Jacobi or Gauss-Seidel) to eliminate oscillatory errors
- Remaining error satisfies $Ae^{(1)} = r^{(1)} = b - Ax^{(1)}$

Two-grid cycle

Multigrid Components

- Relaxation
- Restriction

Relax: $x^{(1)} = x^{(0)} + M^{-1} r^{(0)}$

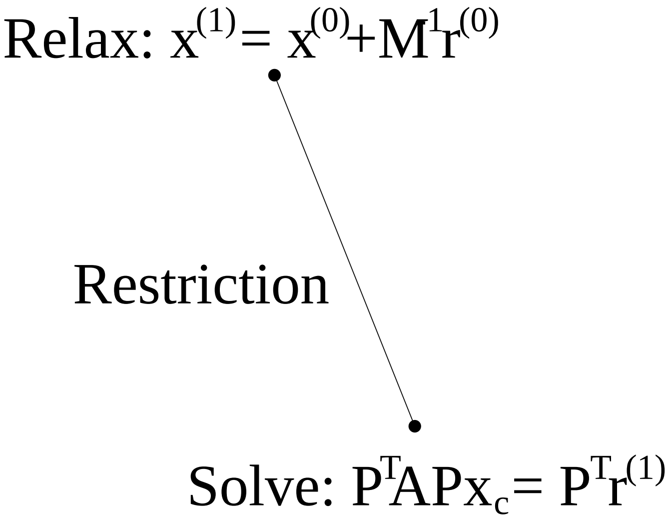
Restriction

- Transfer residual to coarse grid
- Compute $P^T r^{(1)}$

Two-grid cycle

Multigrid Components

- Relaxation
- Restriction
- Coarse-Grid Correction



- Use coarse-grid correction to eliminate smooth errors
- Best correction x_c satisfies

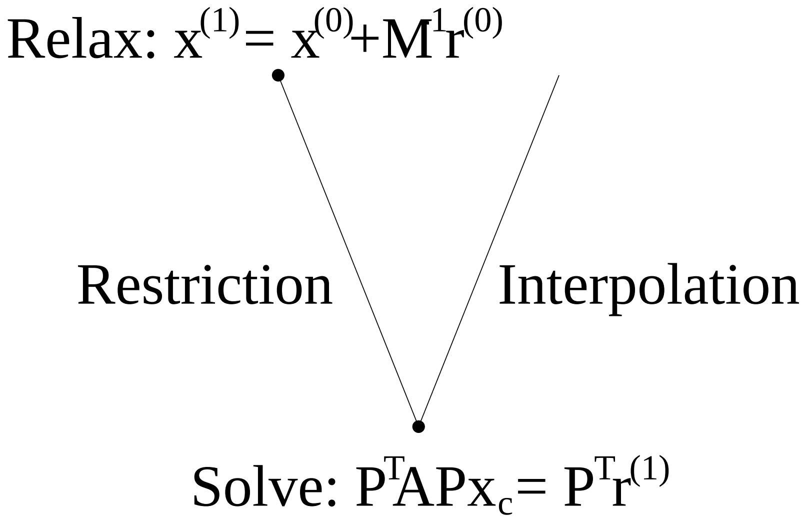
$$P^T A P x_c = P^T r^{(1)}$$

TUDelft

Two-grid cycle

Multigrid Components

- Relaxation
- Restriction
- Coarse-Grid Correction
- Interpolation

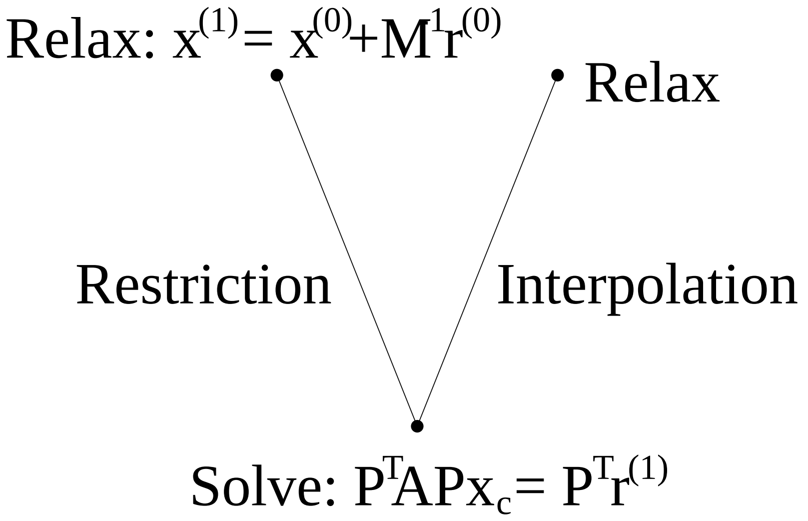


- Transfer correction to fine grid
- Compute $x^{(2)} = x^{(1)} + P x_c$

Two-grid cycle

Multigrid Components

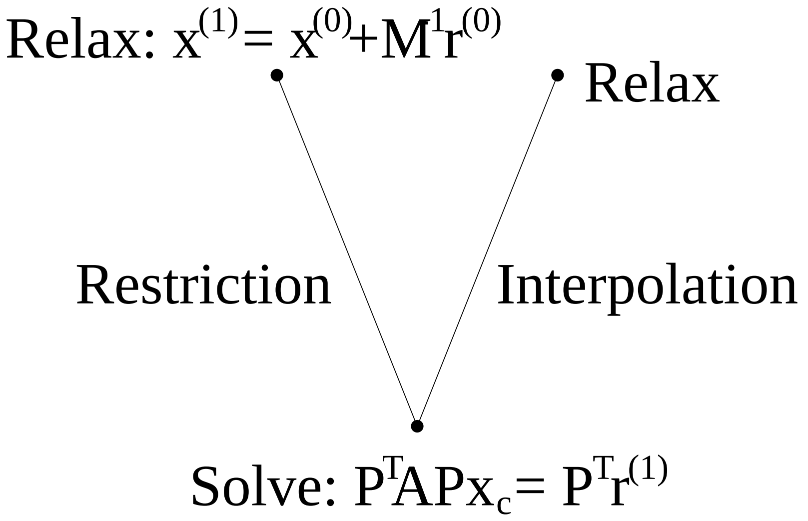
- Relaxation
- Restriction
- Coarse-Grid Correction
- Interpolation
- Relaxation
- Relax once again to remove oscillatory error introduced in coarse-grid correction



Two-grid cycle

Multigrid Components

- Relaxation
- Restriction
- Coarse-Grid Correction
- Interpolation
- Relaxation

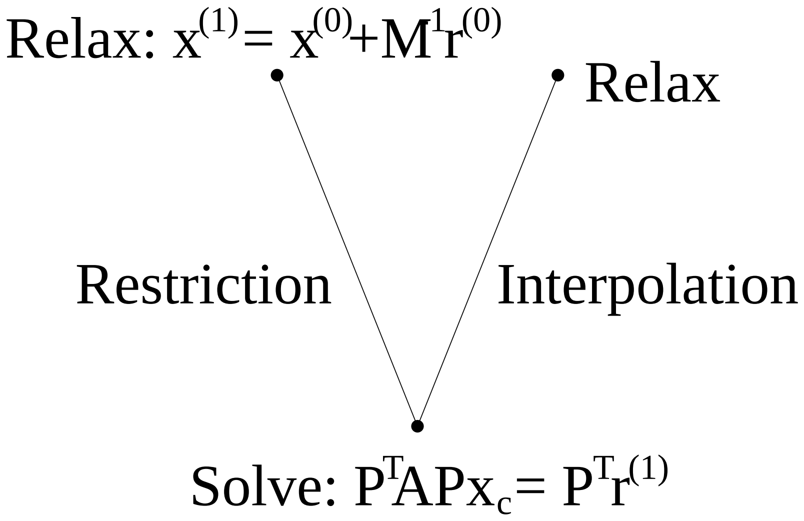


Direct solution of coarse-grid problem isn't practical

Two-grid cycle

Multigrid Components

- Relaxation
- Restriction
- Coarse-Grid Correction
- Interpolation
- Relaxation



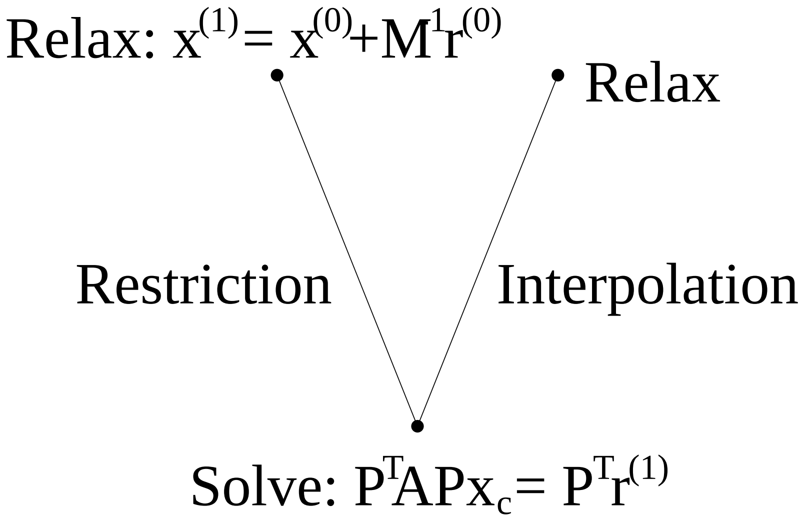
Direct solution of coarse-grid problem isn't practical

Use an iterative method!

Two-grid cycle

Multigrid Components

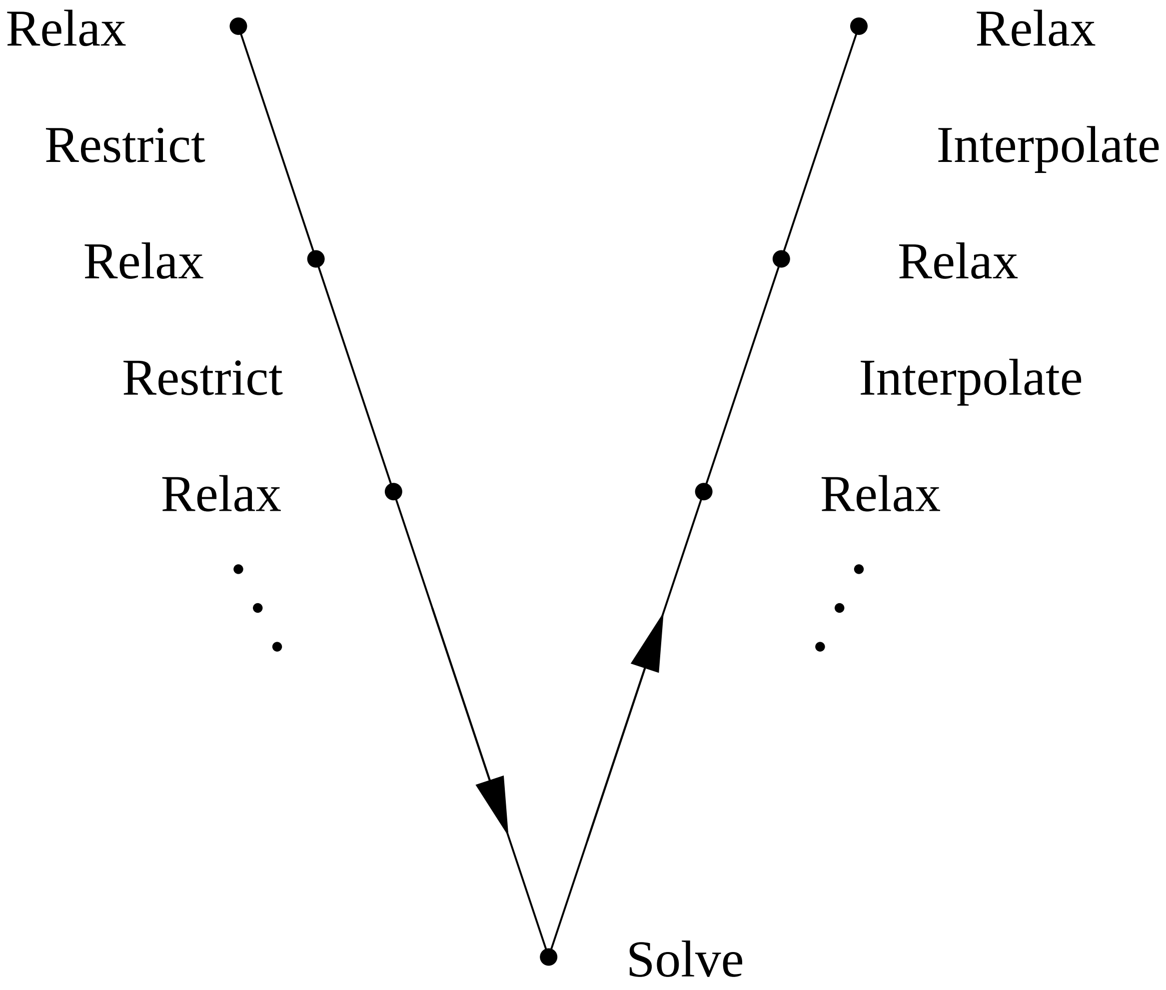
- Relaxation
- Restriction
- Coarse-Grid Correction
- Interpolation
- Relaxation



Recursion!

Apply same methodology to solve coarse-grid problem

The Multigrid V-cycle

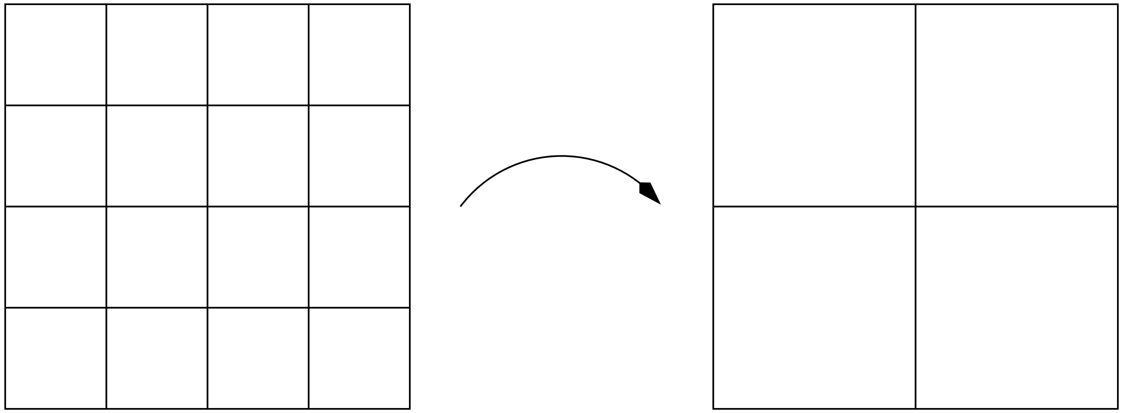


Properties of Effective Cycles

- Fast convergence
  - Effective reduction of all error components
  - On each level, coarse-grid correction must effectively reduce exactly those errors that are slow to be reduced by relaxation alone
  - Hierarchy of coarse-grid operators resolves relevant physics at each scale
- Low iteration cost
  - Simple relaxation scheme (cheap computation of $M^{-1}r$ on all levels)
  - Sparse coarse-grid operators
  - Sparse interpolation/restriction operations

Choosing Coarse Grids

- No *best* strategy to choose coarse grids
- Operator dependent, but also machine dependent
- For structured meshes, often use uniform de-refinement approach



- For unstructured meshes, various weighted independent set algorithms are often used.

What didn't we talk about?

- How do we choose P?
  - Number of columns
  - Sparsity structure
  - Non-zero values
- Choices depend closely on the properties of the relaxation method

Concluding remarks about multigrid

Multigrid works well if the problem

- is grid-based. However, matrix-based multigrid methods (Algebraic Multigrid) do exist and are often successful;
- has a smooth solution. An underlying assumption is that the error can be represented on a coarser grid. Multigrid works particularly well for Poisson-type problems. For these problems the number of operations is $O(n)$.

Multigrid can be used as a separate solver, but is often used as a preconditioner for a Krylov-type method, or for example as building block in a saddle point preconditioner.