Classical iterative methods and multigrid methods to solve large sparse systems

1. Classical methods

2. Multigrid methods

3. Multilevel methods

4. Application

Motivation

We want to solve a large sparse system of linear equations

$$Ax = b\,.$$

We assume that A is obtained from discretization of a partial differential equation.

- Direct methods work always (slow),
- Iterative methods (may) work fast.
- Multigrid is potentially a very fast method.

The problems arise from discretization of partial differential equations, like they are used to describe

- single phase flow,
- transport,
- multi phase flow,
- ...

Complexity of linear solvers

The complexity of solvers, i.e. the number of operations necessary to solve the linear equations, varies strongly with the method used. Let

$h = \sqrt[d]{N}, N$ be the number of unknowns,

A has a constant number of entries per row (sparsity)

A reduction of the residual $r := b - Ax$
by a factor of ϵ takes

Dimension	$d = 2$	$d = 3$
Gaussian elimination	$O(N^3)$	$O(N^3)$
Richardson, GS, Jacobi	$O(N^2)$	$O(N^{1.67})$
Conjugate Gradients (CG)	$O(N^{1.5})$	$O(N^{1.33})$
Multigrid method	$O(N)$	$O(N)$

The number of operations for one step of the linear iteration methods (J, GS, R) is $O(N)$.

Classical methods

- Problem: solve $A\,x = b$ for large A

$\to$ Linear iterative solution method

- Iteration scheme reads:

$$x_{i+1} = x_i + M^{-1}(b - A\,x_i)$$

with splitting: $\quad A = L + D + R$

Possible methods are:

- Richardson: $M = I$
- Jacobi: $M = D$
- Gauss-Seidel: $M = L + D$
- Symmetric Gauss-Seidel, ILU...

Including a scalar damping factor ω the final method reads

$$x_{i+1} = x_i + \omega\,M^{-1}(b - A\,x_i)$$

J. Eberhard, *Classical iterative methods and multigrid methods*

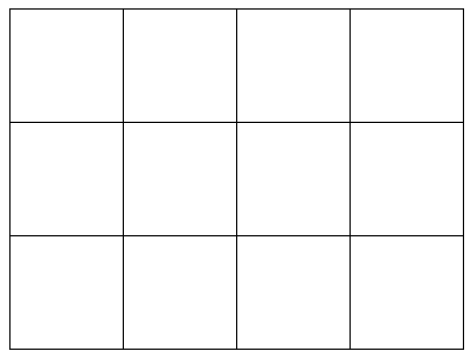
- Example: Stencil for Δ from finite difference scheme

$$A_h = \frac{1}{h^2}\begin{bmatrix} & -1 & \\ -1 & 4 & -1 \\ & -1 & \end{bmatrix}$$

$$\Rightarrow \quad M^{-1} = diag\left(\tfrac{h^2}{4}\right)$$

For $b = 0$:

$$\Rightarrow \quad x_{i+1}^{(C)} = \tfrac{1}{4}\left(x_i^{(E)} + x_i^{(W)} + x_i^{(N)} + x_i^{(S)}\right)$$



Convergence

The iterative scheme is said to converge if

$$\lim_{i \to \infty} x_i = x \quad \forall x_0 \text{ (initial guess).}$$

A linear iterative method with iteration matrix $S = I - M^{-1}A$ converges if

$$\rho(S) < 1 \quad \rho(S) \text{ the spectral radius of } S.$$

- If A is symmetric, positive definite the Richardson and the Jacobi method converge for suitable ω.
- The Gauss-Seidel method converges for all symmetric, positive definite matrices.

Disadvantage of classical iteration methods:

Convergence rate $\to 1$
if the number of unknowns $N \to \infty$.

Example for the convergence rates of Gauss-Seidel:

$$-\Delta u = f, \text{ in } [0,1]^2$$

Step	N=49	N=225	N=961	N=15876
1	0.5604	0.7893	0.8975	0.9749
2	0.6174	0.8562	0.9386	0.9861
3	0.6196	0.8708	0.9494	0.9891
10	0.6189	0.8797	0.9654	0.9941

Reason:
Convergence $\Leftrightarrow$ spectral radius $\rho(S) < 1$
For large N one finds:

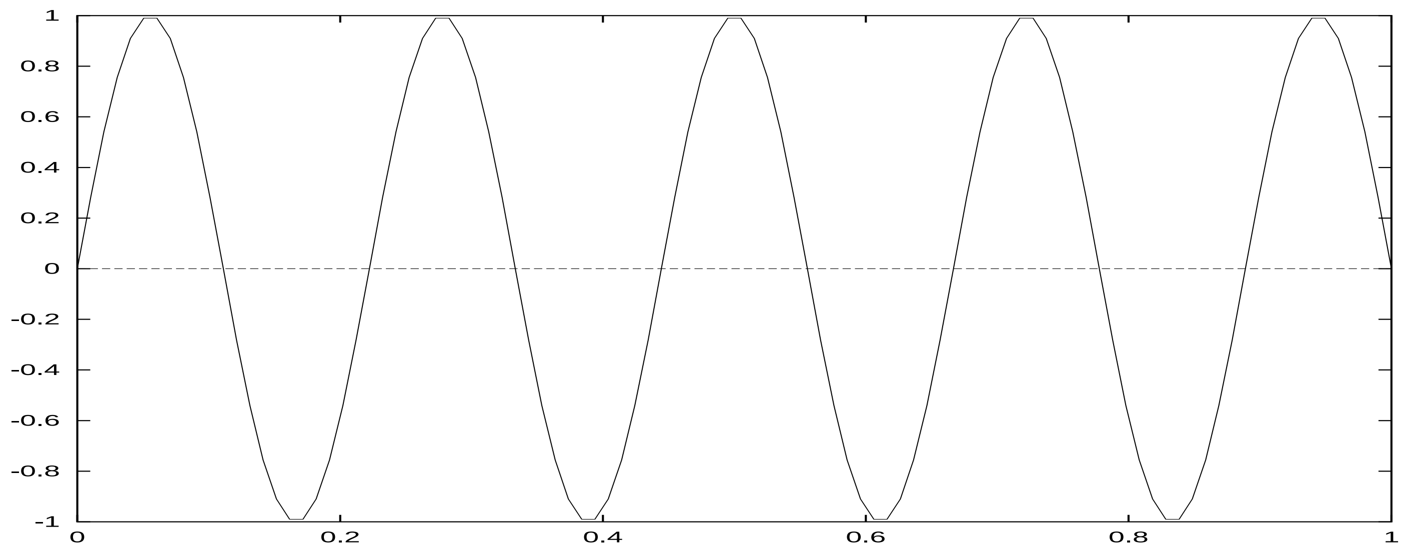
$$\rho(S) = 1 - O\left(\tfrac{1}{N^d}\right) \qquad \text{(Jacobi)}$$

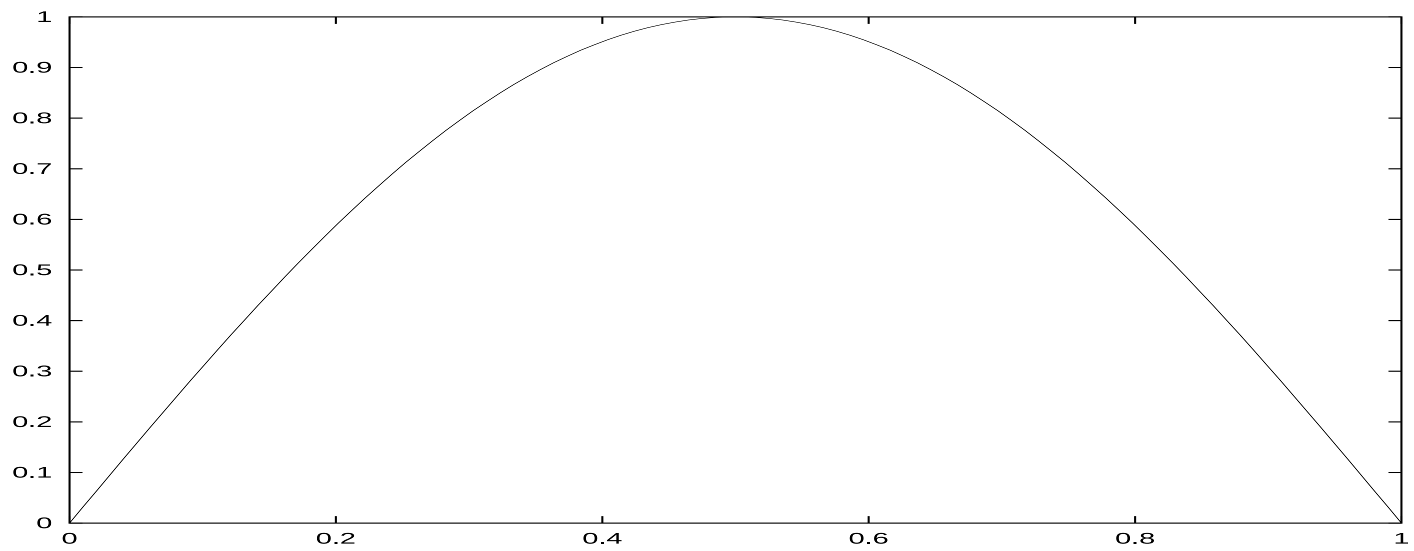
Multigrid methods

Observation: Classical iteration schemes act like “smoothers”

- Error components corresponding to large eigenvalues are damped efficiently.
- Error components corresponding to small eigenvalues are damped slowly.

Example for $d = 1$:





$\rightarrow$ High-frequency contributions of the function are reduced very well

$\rightarrow$ A smooth function can be represented on a coarser grid well

Idea: First to smooth,
then to represent the remaining error on the coarser grid
and to reduce it here (by smoothing) = *coarse grid correction*

$\rightarrow$ Two-grid method

By recursive appliance of this idea $\rightarrow$ multigrid method with

$$\text{convergence rate } < \theta < 1$$

independent of the number of unknowns.

Multigrid algorithm:

For solving $A_l\,x = b_l$ with start vector x_0

$MGM(x_i, b_l, l)$
{
  if ($l = 0$) $x_i := A_0^{-1} b_0$ (coarsest grid)
  else
  {
    smooth x_i with classical scheme on grid l
    calculate defect $d_l = A_l x_i - b_l$ ($A_l e_l = d_l$)
    transfer d_l on to grid $l-1$
    $e_{l-1} := 0$
    $MGM(e_{l-1}, d_{l-1}, l-1)$
    transfer e_{l-1} on to grid l
    $x_i := x_i + e_l$
    smooth again x_i
    $\rightarrow x_{i+1}$
  }
}

Transfer between the grids is done by prolongation and restriction operators:

$\rightarrow$ We introduce two linear mappings

$$P_l : V_{l-1} \rightarrow V_l \text{ prolongation,} \quad R_l : V_l \rightarrow V_{l-1} \text{ restriction}$$

How should P_l and R_l be chosen?

- P_l and R_l should have the correct order, i.e.

$$m_p + m_r > 2m$$

- In the finite-element case P_l is the canonical finite-element interpolation

- For the restriction one uses $R_l = P_l^T$ in the finite-element case.

Multilevel methods

Problems:

- Appropriate choose of restriction and prolongation
- How can one get A_l on grid l

Possibilities are:

- Usually: linear/bilinear interpolation
  Matrix-dependent transfer operators $\rightarrow$ algebraic multigrid methods
- New discretization
  Galerkin product

  Average methods and discretization $\rightarrow$ algebraic methods
  Upscaling methods and discretization
  Adapted choose of the grid $\rightarrow$ algebraic multilevel methods

Application

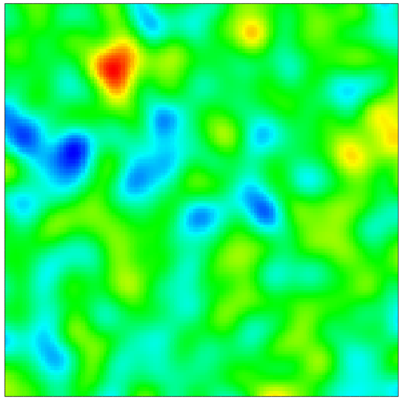
Flow equation for a porous medium:

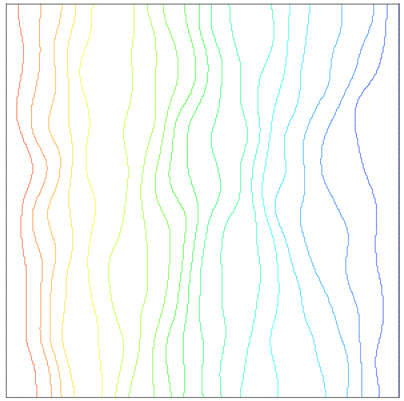
$$-\nabla k_f(x)\nabla u(x) = f(x) \text{ in } [0,1]^2$$
$$+ \text{ boundary conditions}$$

$u =$ piezometric head (pressure)
$k_f =$ permeability of the medium
$f =$ source term

Discretization $\rightarrow A\,x = b$

Results:





Convergence rate:

```
************** linear_solver.ls.mgs **************
 0    u: 1.44086e+01   ---
 1    u: 7.72044e-01   5.35819e-02          7    u: 2.27608e-05   2.97320e-01
 2    u: 7.20197e-02   9.32845e-02          8    u: 6.90513e-06   3.03377e-01
 3    u: 1.05129e-02   1.45973e-01          9    u: 2.22084e-06   3.21622e-01
 4    u: 1.65848e-03   1.57755e-01          10   u: 7.03723e-07   3.16871e-01
 5    u: 3.35678e-04   2.02401e-01          11   u: 2.24994e-07   3.19719e-01
 6    u: 7.65531e-05   2.28055e-01          12   u: 7.13847e-08   3.17273e-01

 12  average:  u: 1.44086e+01   7.13847e-08   2.031e-01
```

Summary

- Multigrid methods use a hierarchy of grids
  - Classical iteration methods are taken as smoother
  - Error correction is calculated on the coarser grids

$\rightarrow$ Efficient solver for $A\,x = b$

- Extensions $\rightarrow$ algebraic multigrid methods