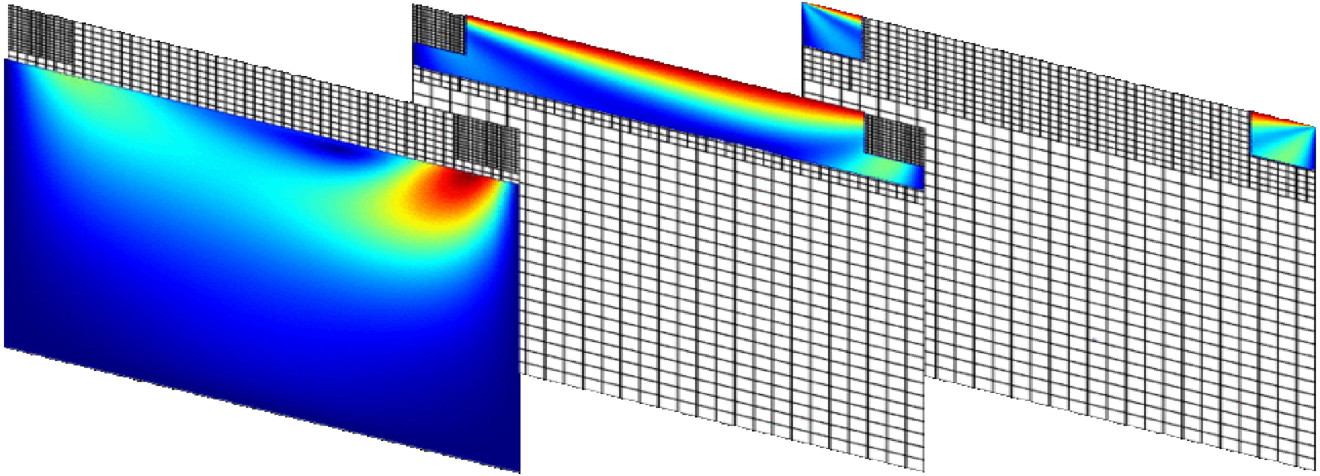


Iterative Method for Linear Equation

$$Ax = b$$

Outline

- Some History
- Basic Iteration Method
- Multigrid Method
- Algebraic Multigrid Method
- KSP Method
- Preconditioner
- PETSc and HYPRE
- Reference

Some History

- Prehistory: Gauss to 1940: no computer
- First period(Stationary iterative method): SOR was proposed by David Young: $x^{k+1} = Tx^k + c$ (hard to estimate, too slow)
- Second period(Krylov subspace method): CG was discovered by Lanczos, Stiefel and Hestenes (only for symmetric, definite), MINRES, SYMMLQ for indefinite. GMRES, Bi-CGSTAB for nonsymmetric.
- Multigrid method: was given by Brandt and Hackbusch targeted at solving 2nd order elliptic equation

Basic iteration methods

- Basic iteration methods for $Ax = b$

1. Matrix split: $\boldsymbol{A = D - L - U}$

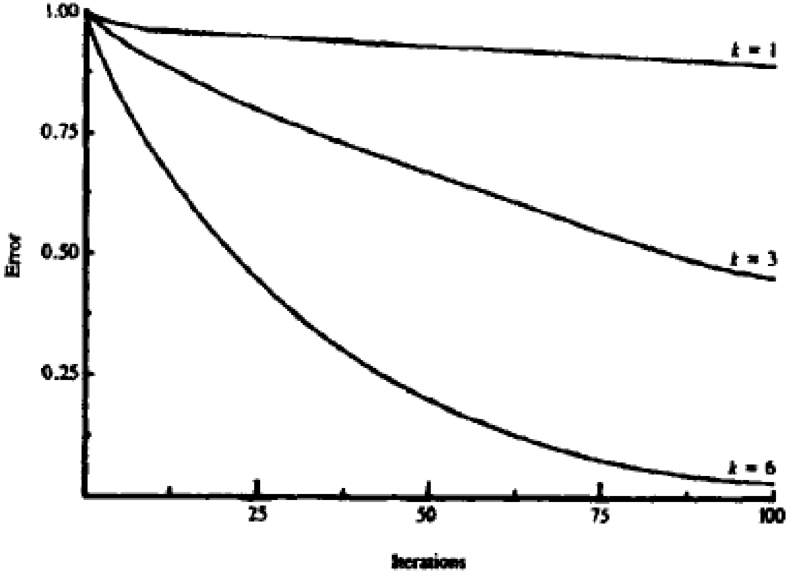
$$(D - L - U)x = b$$

2. Fixed point iteration:

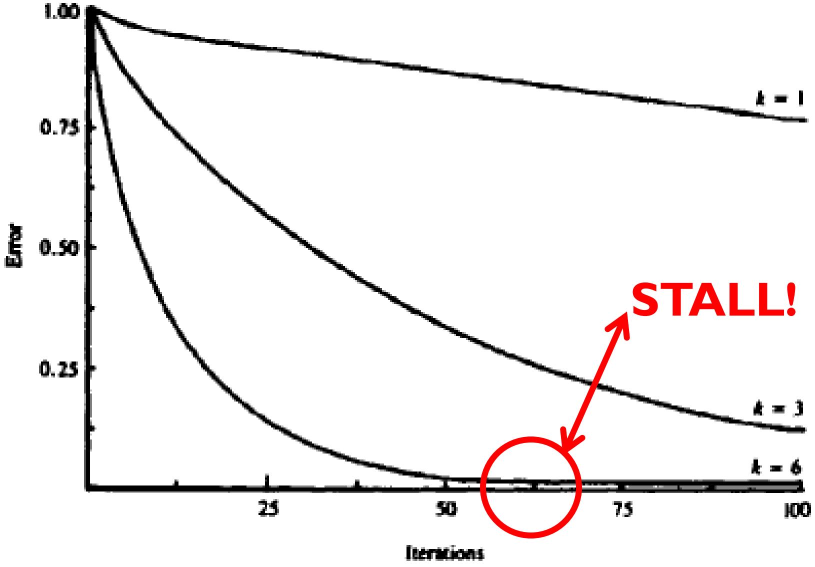
$$x_{n+1} = D^{-1}(L+U)x_n + D^{-1}b \quad \text{(Jacobi)}$$

$$x_{n+1} = (D-L)^{-1}Ux_n + (D-L)^{-1}b \quad \text{(Gauss Seidel)}$$

Weakness of basic iteration method



Error of Jacobi iteration

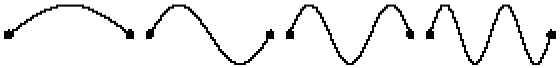


Error of Gauss-Seidel iteration

The reasons

- “*The slow elimination of the low-frequency components degrades the performance of these methods*” *[2]*

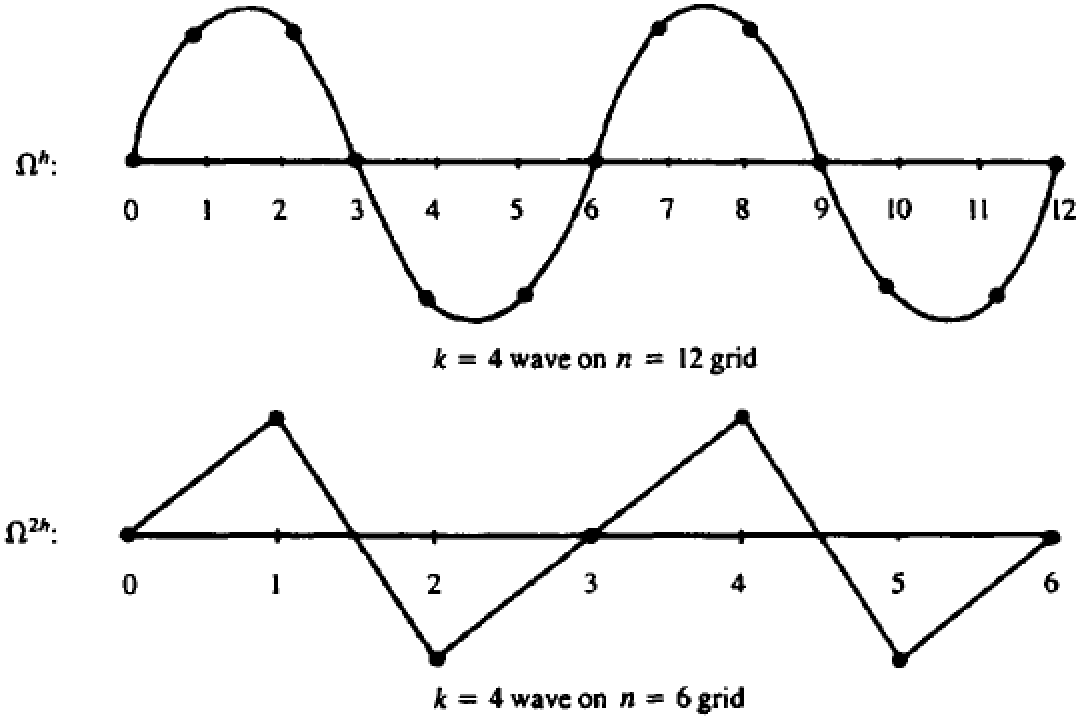
$$e^{(0)} = \sum_{k=1}^{n-1} c_k w_k$$

- w_k is the Fourier modes 
- c_k give the “amount” of each mode in the error

[2] W. Briggs, V. Henson, S. McCormick, A Multigrid Tutorial, SIAM, 2000.

Multigrid method

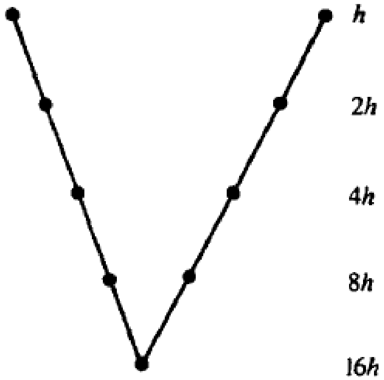
- *The coarse grid "sees" a wave that is more oscillatory on the coarse grid than on the fine grid*



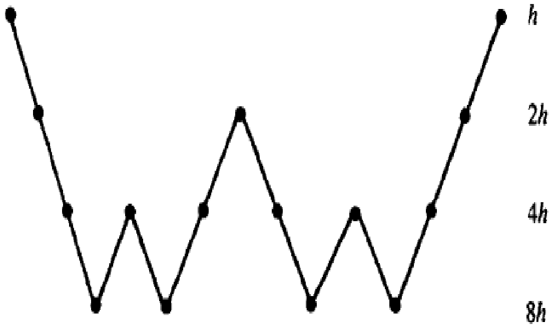
Procedure

- Iterate one step for $Ax = b$ on Ω^h, obtain x^h
- Compute residual $r^h = b - Ax^h$
- Inject r^h to Ω^{2h}, obtain r^{2h}
- Iterate one step for $Ae = r$ on Ω^{2h}, obtain e^{2h}
- Interpolate e^{2h} to Ω^h, obtain e^h
- $x^h \leftarrow x^h + e^h$

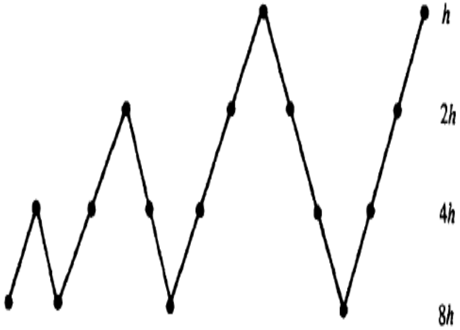
Procedure



(a)



(b)



(c)

Cheap
V-cycle

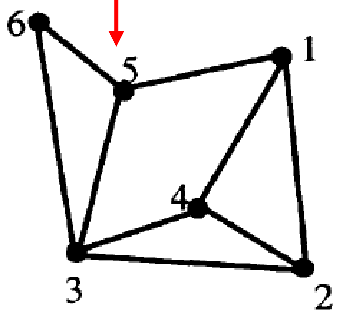
Expensive
W-cycle

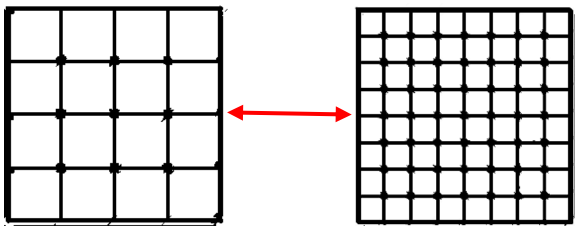
Combination
F-cycle

Algebraic multigrid method (AMG)

- What if we don't have grid?
- How to define the coarse grid?
- How to define interpolation? ($\boldsymbol{C} \rightarrow \boldsymbol{F}$)
- How to define injection? ($\boldsymbol{F} \rightarrow \boldsymbol{C}$)
- AMG solves the above problems.

$$A = \begin{bmatrix} x & x & & x & x & \\ x & x & x & x & & \\ & x & x & x & x & x \\ x & x & x & x & & \\ x & & x & & x & x \\ & & x & & x & x \end{bmatrix}$$





Krylov Subspace method

- We are looking for "good" combinations of $A^j r_0$

$$x_k - x_0 = \sum_{j=0}^{k-1} c_j A^j r_0$$

- $\kappa_k(A, r_0)$ =Span$\{r_0, Ar_0, \ldots, A^{k-1} r_0\}$ is Krylov Subspace
- Two ways to define "good":
  1. min$\|Ax_k - b\|_2$ (GMRES)
  2. $Ax_k - b \perp \kappa_k(A, r_0)$ (FOM, Bi-CG)

Preconditioner

- KSP has high performance, but still can stall
- Use AMG as a preconditioner (how?)

  ILU: $A \approx LU$

  $$L^{-1}AU^{-1}u = L^{-1}b, \quad x = U^{-1}u$$

  ILU is accomplished by AMG. [1]
- This combination gives a "black box" solver
- Unstructured mesh or mesh free application
- General purpose, purely algebraic methods

[1] M. Benzi, Preconditioning techniques for Large Linear System: A Survey, JCP, 182: 418-477, 2002.

PETSc and HYPRE

- The **Portable, Extensible Toolkit for Scientific Computation** (**PETSc**, pronounced PET-see; the S is silent): Linear solver, nonlinear solver, parallel, preconditioner
- The **High Performance Preconditioner** (**HYPRE**): Parallel multigrid method for both structured and unstructured grid problems.

PETSc in FronTier++

- Located on: FronTier++/solver/solver.cpp
- Simplify the procedure
- Only need to set up matrix
- Solving is automatic

```
#if defined __HYPRE__
void PETSc::Solve_HYPRE(void)
{
        PC pc;
        start_clock("Assemble matrix and vector");
        ierr = MatAssemblyBegin(A,MAT_FINAL_ASSEMBLY);
        ierr = MatAssemblyEnd(A,MAT_FINAL_ASSEMBLY);

        ierr = VecAssemblyBegin(x);
        ierr = VecAssemblyEnd(x);

        ierr = VecAssemblyBegin(b);
        ierr = VecAssemblyEnd(b);
        stop_clock("Assembly matrix and vector");

        KSPSetType(ksp,KSPBCGS);
        KSPSetOperators(ksp,A,A,DIFFERENT_NONZERO_PATTERN);
        KSPGetPC(ksp,&pc);
        PCSetType(pc,PCHYPRE);
        PCHYPRESetType(pc,"boomeramg");
        KSPSetFromOptions(ksp);
        KSPSetUp(ksp);

        start_clock("KSPSolve");
        KSPSolve(ksp,b,x);
        stop_clock("KSPSolve");

}
#endif // defined __HYPRE__
```

A simple example of PETSc

Create matrix A, vector b, solution x:

$$\begin{vmatrix} -2 & 1 & & & \\ 1 & -2 & 1 & & \\ & 1 & -2 & 1 & \\ & & 1 & -2 & 1 \\ & & & 1 & -2 \end{vmatrix}$$

```
for (i = 1; i < 5; i++)
{
    col[0] = i-1;  col[1] = i; col[2] = i+1;
    value[3] = {1, -2, 1};
    MatSetValues(A,1,&i,3,col,value,INSERT_VALUE);
}
```

The first row and last row need to be set separately

A simple example of PETSc

- Create linear solver

```
PC pc;
KSPSetType(ksp,KSPGMRES);      /*determine KSP solver*/
KSPSetOperators(ksp,A,A,DIFFERENT_NONZERO_PATTERN);
KSPGetPC(ksp,&pc);
PCSetType(pc,PCHYPRE);      /*determine preconditioner*/
PCHYPRESetType(pc,"boomeramg");
KSPSetFromOptions(ksp);
KSPSetup(ksp);
```

A simple example of PETSc

- Solve equation and obtain solution

```
double *values;
KSPSolve(ksp,b,x);            /*Solve equation*/
VecGetArray(x,&values);       /*Get solution*/
```

Summary

- Basic iteration may stall
- Multigrid can accelerate, but need structure grid.
- AMG is purely algebraic, no structure information needed
- A combination of AMG and KSP gives a good solver
- PETSc and HYPRE do this for us
- Enjoy!

Reference

[1] M. Benzi, Preconditioning techniques for Large Linear System: A Survey, Journal of Computational Physics, 182: 418-477, 2002.

[2] W. Briggs, V. Henson, S. McCormick, A Multigrid Tutorial, Society for Industrial and Applied Mathematics, 2000.