Multigrid Methods
236790

Irad Yavneh
Department of Computer Science
Taub 618, Tel. 4945
Office Hour: By Appointment (email)
irad@cs.technion.ac.il
Pre-requisites: Linear Algebra, Programming in MATLAB, Numerical Algorithms

Determination of grade:

1. Participation 10%

2. Exercise grades + mini-project 90%
Some Relevant Books:


Exercises and Grade

There will be about 10 exercises + mini-project. Exercises must be submitted by e-mail to irad@cs.technion.ac.il by the due date.

Programming exercises must be implemented in MATLAB and presented in such a way (+ input and running instructions) that allows me to easily run the tests.

No joint work.

All results must be explained well.

Grade: 10% is automatically given to anyone who showed up regularly and submitted the exercises. The remaining 90% is given according to the exercise grades + mini-project.
What’s it about?

A framework of efficient iterative methods for solving problems with many variables and many scales.
• **Framework**: common concept, different methods.
• **Efficient**: usually $O(N)$ or $O(N \log N)$ operations.
• **Iterative**: most nontrivial problems in our field cannot be solved directly efficiently.
• **Solving**: approximately, subject to appropriate convergence criteria, constraints, etc.
• **Many variables**: the larger the number of variables, the greater the gain of efficient methods.
• **Many scales**: typical spatial and/or temporal sizes.

A framework of efficient iterative methods for solving problems with many variables and many scales.
Basic Concepts: Local vs. Global processing.

Imagine a large number of soldiers who need to be arranged in a straight line and at equal distances from each other.

The two soldiers at the ends of the line are fixed. Suppose we number the soldiers 0 to $N$, and that the length of the entire line is $L$. 
Initial Position
Final Position
Global processing. Let soldier number $j$ stand on the line connecting soldier $0$ to soldier $N$ at a distance $jL/N$ from soldier number $0$. 
This method solves the problem directly, but it requires a high degree of sophistication: recognition of the extreme soldiers and some pretty fancy arithmetic.
Local processing (iterative method). Suppose that the inner soldiers’ initial position is \( \mathbf{x}^{(0)} = (x_1, x_2, \ldots, x_{N-1})^{(0)} \). Then repeat for \( i=1,2,\ldots \): Let each soldier \( j, j=1,\ldots,N-1 \) at iteration \( i \) move to the point midway between the locations of soldier \( j-1 \) and soldier \( j+1 \) at iteration \( i-1 \):

\[
x_j^{(i)} = \frac{1}{2} \left( x_{j-1}^{(i-1)} + x_{j+1}^{(i-1)} \right)
\]

This is an iterative process. Each iteration brings us closer to the solution(?). The arithmetic is trivial.
A step in the right direction
Slow convergence
Fast convergence
Slow convergence
Local solution: damping
Local solution: damping
Local solution: damping
Local solution: damping
The multiscale idea: Employ the local processing with simple arithmetic. But do this on all the different scales.
Large scale
Large scale
Intermediate scale
Intermediate scale
Small scale
HOW MUCH DO WE SAVE?

Analysis of the Jacobi iterative process

Matrix representation:

\[
\mathbf{x}^{(i)} = \mathbf{S}\mathbf{x}^{(i-1)}
\]

where

\[
\mathbf{S} = \frac{1}{2}
\begin{bmatrix}
0 & 1 & & & \\
1 & 0 & 1 & & \\
1 & 0 & 1 & & \\
& & & \ddots & \ddots & \ddots \\
& & & 1 & 0 & 1 \\
& & & 1 & 0 & 1 \\
& & & & & 1 & 0 \\
\end{bmatrix}
\]
This matrix $S$ has $N - 1$ linearly independent eigenvectors, $v^k$, and corresponding real eigenvalues, $\lambda_k$

$$S v^k = \lambda_k v^k.$$ 

Since $v^k$ span the space $\mathbb{R}^{N-1}$, any initial configuration of the soldiers can be written as a linear combination:

$$x^{(0)} = \sum_{k=1}^{N-1} c_k v^k$$

with some coefficients, $c_k$. 
Hence, we obtain after $m$ iterations:

$$x^{(m)} = S x^{(m-1)} = S^2 x^{(m-2)} = \ldots = S^m x^{(0)} = S^m \sum_k c_k v^k = \sum_k c_k \lambda_k^m v^k$$

**Conclusion:**

$$\lim_{m \to \infty} x^{(m)} \to 0 \quad \text{if} \quad |\lambda_k| < 1, \quad k = 1, \ldots, N - 1$$

The iteration converges if the spectral radius, $\rho$, of the iteration matrix, $S$, is smaller than 1.
**Observation:** the eigenvectors and eigenvalues of the matrix $S$ are given by

$$
\mathbf{v}^k = \{ \mathbf{v}_j^k \} = \left\{ \sin \left( \frac{j k \pi}{N} \right) \right\}, \quad j = 1, \ldots, N - 1,
$$

$$
\lambda_k = \cos \left( \frac{k \pi}{N} \right),
$$

with $k = 1, \ldots, N - 1$.

**Proof:** Using the trigonometric identity,

$$
\frac{1}{2} \left[ \sin \left( \frac{(j-1) k \pi}{N} \right) + \sin \left( \frac{(j+1) k \pi}{N} \right) \right] = \cos \frac{k \pi}{N} \sin \frac{j k \pi}{N},
$$

and the fact that $\sin 0 = \sin \pi = 0$, we obtain by substitution, $S \mathbf{v}^k = \lambda_k \mathbf{v}^k$. 
**Note:** since $|\lambda_k| < 1$, the method converges. But, for some eigenvectors, $|\lambda_k|$ is close to 1, so convergence is slow. In particular, for $k\pi/N \ll 1$, we have,

$$\lambda_k = \cos\left(\frac{k\pi}{N}\right) \approx 1 - \frac{1}{2}\left(\frac{k\pi}{N}\right)^2.$$  

For $k = 1$ we obtain

$$\lambda_1^m \approx \left[1 - \frac{1}{2}\left(\frac{\pi}{N}\right)^2\right]^m \approx e^{-\frac{1}{2}m\left(\frac{\pi}{N}\right)^2}.$$  

**Conclusion:** $O(N^2)$ iterations are required to reduce such an error by an order of magnitude.
How much work do we save?

Jacobi’s method requires about $N^2$ iterations and $N^2 \times N = N^3$ operations to improve the accuracy by an order of magnitude.

The multiscale approach solves the problem in about $\log_2(N)$ iterations (whistle blows) and only about $N$ operations.

Example: for $N = 1000$ we require about:

10 iterations and 1000 operations

instead of about

1,000,000 iterations and 1,000,000,000 operations
How important is computational efficiency?

Suppose that we have three different algorithms for a given problem, with different computational complexities for input size $N$:

Algorithm 1: $10^6 N$ operations

Algorithm 2: $10^3 N^2$ operations

Algorithm 3: $N^3$ operations

Suppose that the problem size, $N$, is such that Algorithm 1 requires one second.

How long do the others require?
<table>
<thead>
<tr>
<th>Computer Speed (ops/sec)</th>
<th>$N$</th>
<th>Algorithm 1 $O(N)$</th>
<th>Algorithm 2 $O(N^2)$</th>
<th>Algorithm 3 $O(N^3)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1M (~$10^6$) (1980’s)</td>
<td>1</td>
<td>1 sec</td>
<td>0.001 sec</td>
<td>0.000001 sec</td>
</tr>
<tr>
<td>1G (~$10^9$) (1990’s)</td>
<td>1K</td>
<td>1 sec</td>
<td>1 sec</td>
<td>1 sec</td>
</tr>
<tr>
<td>1T (~$10^{12}$) (2000’s)</td>
<td>1M</td>
<td>1 sec</td>
<td>17 min</td>
<td>12 days</td>
</tr>
<tr>
<td>1P (~$10^{15}$) (2010’s)</td>
<td>1G</td>
<td>1 sec</td>
<td>12 days</td>
<td>31,710 years</td>
</tr>
</tbody>
</table>

**Stronger Computers**  \(\implies\)**Greater Advantage of Efficient Algorithms!**
The catch: in less trivial problems, we cannot construct appropriate equations on the large scales without first propagating information from the small scales.

Skill in developing efficient multilevel algorithms is required for:

1. Choosing a good local iteration.
2. Choosing appropriate coarse-scale variables.
3. Choosing inter-scale transfer operators.
4. Constructing coarse-scale approximations to the fine-scale problem.
Damping

Recall: the eigenvectors and eigenvalues of the iteration matrix $S$ are given by

$$\mathbf{v}^k = \{v_j^k\} = \left\{ \sin\left(\frac{jk\pi}{N}\right) \right\}, \quad j = 1, \ldots, N-1,$$

$$\lambda_k = \cos\left(\frac{k\pi}{N}\right),$$

with $k = 1, \ldots, N-1$.

Note that convergence is also slow for $k / N \approx 1$, because then $\lambda_k \approx -1$. 
This slow convergence can be overcome by damping:

\[ x_j^{(i)} = (1 - \omega)x_j^{(i-1)} + \omega \frac{1}{2} \left( x_{j-1}^{(i-1)} + x_{j+1}^{(i-1)} \right), \]

where \( \omega \) is a parameter.

Then, \( x^{(i)} = S_\omega x^{(i-1)} \), where

\[ S_\omega = (1 - \omega)I + \omega S. \]

Note: \( v^k \) are eigenvectors of \( S_\omega \). The corresponding eigenvalues are now \( \lambda_k(\omega) = 1 - \omega + \omega \lambda_k = 1 - \omega (1 - \lambda_k) \).

For \( 0 < \omega \leq 1 \), we have convergence, \( |\lambda_k(\omega)| < 1 \).
**Definition:**

Eigenvectors $\mathbf{v}_k$ with $1 \leq k < N/2$ are called **smooth** (low-frequency).

Those with $N/2 \leq k < N$ are called **rough** or oscillatory (high-frequency).

Recall that $\lambda_k = \cos\left(\frac{k\pi}{N}\right)$, so for rough eigenvectors,

$$\lambda_k \leq 0.$$
Exercise 0: Find $0 < \omega < 1$ which yields optimal convergence for the set of rough modes for arbitrary $N$: 

$$\omega : \sup_{N} \max_{\frac{N}{2} \leq k < N} \left| \lambda^{(\omega)}_{k} \right| = \min!, $$

i.e.,

$$\omega : \sup_{\lambda \in (-1, 0]} \left| 1 - \omega + \omega \lambda \right| = \min!, $$

What is then the bound on the convergence factor, $\left| \lambda^{(\omega)}_{k} \right|$, maximized over the rough modes? (Clues in my introductory paper, “Why multigrid methods are so efficient”.)
1D Model Problem

Find $u$ which satisfies:

\[ Lu = u''(x) = f(x), \quad x \in (0, 1), \quad (1) \]

\[ u(0) = u_0, \]
\[ u(1) = u_1. \]

In the particular case where $f = 0$, the solution is a straight line that connects $u_0$ with $u_1$. 
Discrete approximation: Since closed-form solutions exist only for a small number of differential equations, we solve such equations approximately by a discrete approximation.

Define a grid: divide the domain \((0,1)\) into \(N\) intervals. Assume for simplicity a uniform grid of mesh-size \(h=1/N\).
Finite-difference discretization; examples:

Forward differences:

\[ u' = \frac{u(x + h) - u(x)}{h} + O(h). \]

Backward differences:

\[ u' = \frac{u(x) - u(x - h)}{h} + O(h). \]

Central differences:

\[ u' = \frac{u(x + h) - 2u(x) + u(x - h)}{2h} + O(h^2). \]

Second derivative:

\[ u''(x) = \frac{u(x - h) - 2u(x) + u(x + h)}{h^2} + O(h^2). \] (2)

Derivation: by the Taylor theorem
We can thus approximate the differential equation by a set of algebraic difference equations:

\[
(L^h u^h)_i = \frac{u^h_{i+1} - 2u^h_i + u^h_{i-1}}{h^2} = f^h_i,
\]

\[i = 1, \ldots, N - 1,\]

\[u^h_0 = u_0,\]

\[u^h_N = u_1.\]
In matrix form:

\[
\frac{1}{h^2} \begin{bmatrix}
-2 & 1 & & & \\
1 & -2 & 1 & & \\
& \ddots & \ddots & \ddots & \\
& & 1 & -2 & 1 \\
& & & 1 & -2
\end{bmatrix} \begin{bmatrix}
u_1^h \\
u_2^h \\
\vdots \\
u_{N-2}^h \\
u_{N-1}^h
\end{bmatrix} = \begin{bmatrix}
f_1^h - u_0^h / h^2 \\
f_2^h \\
\vdots \\
f_{N-2}^h \\
f_{N-1}^h - u_1^h / h^2
\end{bmatrix}
\]

This is a tridiagonal system of equations which can be solved directly or iteratively.
Cyclic Reduction:

We can generalize the ideas of the soldier-line problem for nonsingular tridiagonal systems.

We eliminate the odd-indexed variables, obtaining a new tridiagonal system for even-indexed variables.

This is repeated recursively until we reduce to a single equation for single variable. This equation is solved trivially. Now we compute all the remaining variables recursively.
Exercise 1:

Write a MATLAB program that solves by Cyclic Reduction a tri-diagonal system of size \( N - 1 \) by \( N - 1 \), where \( N \) is a power of 2. Test the program against known solutions.

Note: you may need to normalize the equations as the algorithm progresses to avoid overflow.

Use example with dominant diagonal, e.g., for \( n = N-1 \):

```matlab
function [A] = triPosDef(n)
    %The function generates a random tridiagonal symmetric
    %positive definite n by n matrix;
    b = randn(1,n-1);
    a = [abs(b), 0] +[0, abs(b)] +abs(randn(1,n));
    A=diag(b,-1)+diag(b,1)+diag(a);
end
```
Motivation for 2D problem:

Shape from Photometric Stereo.

Several images are taken with a fixed camera location and different lighting conditions. The objective is to recover the shape.

Given three images $I_1, I_2, I_3$, of the same object taken with three different lighting directions $l_1, l_2, l_3$, the following relation holds for a Lambertian reflectance model:
Given three images, $I_1, I_2, I_3$, of the same object taken with three different lighting directions, $l_1, l_2, l_3$, the following relation holds for a Lambertian reflectance model:

$$I_i = \rho \langle l_i, N \rangle, \quad i \in \{1, 2, 3\}$$

where $\rho(x, y)$ is the albedo, and $N$ is the normal to the surface, $z(x, y)$, given by

$$N = \frac{(-z_x, -z_y, 1)}{\sqrt{1 + z_x^2 + z_y^2}}$$

The approximate surface gradient, $(p, q)^T \approx \nabla_z$, can be extracted easily from the images.
Input images with the same camera position and head object but three different lighting directions.
Given $p \approx z_x$, $q \approx z_y$, we wish to compute $z$. We therefore search for the surface $z$ which minimizes the functional,

$$\iint_{\Omega} \left( (p - z_x)^2 + (q - z_y)^2 \right) \, dx \, dy$$

Where $\Omega$ is the image domain.

The corresponding Euler-Lagrange equation is

$$z_{xx} + z_{yy} = p_x + q_y, \quad \text{in } \Omega,$$

$$\nabla z \cdot n = (p, q)^T \cdot n, \quad \text{on } \partial \Omega.$$ 

This is the Poisson problem (with Neumann boundary conditions; if $z$ is given on the boundary instead, we have Dirichlet boundary conditions).
Some Results: Side and perspective views of the reconstructed surface, with the frontal textured mapped onto it. **Left:** unconstrained; **Right:** eight constrained points.
2D Model Problem

Find $u$ which satisfies:

$$Lu = u_{xx} + u_{yy} = f(x, y), \quad (x, y) \in \Omega,$$

$$u = g(x, y), \quad (x, y) \in \partial \Omega. \quad (4)$$

This is the 2D Poisson equation, with Dirichlet boundary conditions. It is an elliptic partial differential equation which appears in many models.
$\Omega^h$
**Discrete approximation**

Define a grid: \( \Omega^h \subset \Omega \) (assumed to be uniform for simplicity, with mesh interval \( h \)).

Let \( u^h \), \( g^h \) and \( f^h \) denote discrete approximations to \( u \), \( g \) and \( f \) defined at the nodes of the grid.

Plug (2) for \( u_{xx} \), and the analogous approximation for \( u_{yy} \) into (4), obtaining:
\[ L^h u^h_{i,j} = \]
\[ \frac{u^h_{i-1,j} - 2u^h_{i,j} + u^h_{i+1,j}}{h^2} + \frac{u^h_{i,j-1} - 2u^h_{i,j} + u^h_{i,j+1}}{h^2} = f^h_{i,j} \text{ in } \Omega^h \]

\[ u^h = g^h \text{ on } \partial^h \Omega^h \]

This yields a nonsingular linear system of equations for \( u^h_{i,j} \) (the discrete operator satisfies a maximum principle.)

We consider solving this system by the classical approach of Gauss-Seidel relaxation.
Gauss-Seidel (GS) Relaxation:

1. Choose initial guess, $\tilde{u}^h$.

2. Repeat until some convergence criterion is satisfied

\{

Scan all variables in some prescribed order, and change each variable $\tilde{u}_{i,j}^h$ in turn so as to satisfy the $(i,j)$th equation.

\}
**Observation:** GS is a local process, because only near neighbors appear in each equation. Hence, it may be efficient for eliminating errors which can be detected locally. But large-scale (“smooth”) errors are eliminated very slowly.

(The difference between GS and Jacobi is that old neighboring values are used in Jacobi, while the most updated values are used in GS.)
Key Observation re-worded: Relaxation cannot be generally efficient for reducing the error (i.e., the difference vector $\tilde{u}^h - u^h$). But relaxation may be extremely efficient for smoothing the error relative to the grid.

Practical conclusion:

1. A smooth error can be approximated well on a coarser grid.

2. A coarser grid implies less variables, hence less computation.

3. On the coarser grid the error is no longer as smooth relative to the grid, so relaxation may once again be efficient.
Grid-refinement algorithm

Define a sequence of progressively finer grids all covering the full domain. Then,

1. Define and solve the problem on the coarsest grid, say by relaxation.

2. Interpolate the solution to the next-finer grid. Apply several iterations of relaxation.

3. Interpolate the solution to the next-finer grid and continue in the same manner...

Does this method converge fast?
Exercise 2

Consider the Poisson problem on the unit square, with Dirichlet boundary conditions, discretized on a square grid of 64 by 64 intervals. Write a MATLAB program that solves this problem using the grid-refinement algorithm with Red-Black relaxation as follows.

First, fix a random right-hand side, $f^h$, on the finest grid, and 0 initial guess for the solution (including boundary values, which remain fixed).

**Note:** $f^h$ should be distributed uniformly with mean 0.

Next...
Restrict $f^h$ by injection to all the coarser grids, down to a grid of just one variable. Solve the problem on the coarsest grid.

Then, repeat: Interpolate the solution to the next finer grid using bi-linear interpolation, and perform 10 relaxation sweeps. When the finest grid is reached, perform 100 relaxations.

On the finest grid, plot (semilogy) the Root Mean Square (RMS) of the residual as a function of the number of iterations.

Repeat the entire exercise using full-weighting (local averaging) instead of injection for restricting $f^h$ to the coarser grids.
1D Model Problem Revisited

Fine-grid \((h)\) difference equation:

\[
L^h u^h = \frac{u_{i+1}^h - 2u_i^h + u_{i-1}^h}{h^2} = f_i^h,
\]

\(i = 1, \ldots N - 1,\)

\[
u_0^h = u_0,\]

\[
u_N^h = u_1.\]

The eigenvectors of \(L^h\) (like those of the Jacobi relaxation operation) are Sine-function “waves”:

\[
v^k = (\sin k\pi / N, \ldots \sin jk\pi / N, \ldots \sin(N - 1)k\pi / N)^T\]
Aliasing
Smooth waves—with \( k \ll N \)—have wavelengths large compared to \( h \). Hence they can be approximated well on the coarse grids. But non-smooth eigenvectors alias with smooth components on the coarse grids.

Since the right-hand side, \( f^h \), will generally have some non-smooth components, these will be “interpreted” as smooth components by the coarse grids, resulting in a smooth error.

Hence, when we interpolate a coarse-grid solution to the fine grid, we still have smooth errors in this solution. These cannot be corrected efficiently by relaxation.
Errors:

There is an important distinction here between the discretization error:

\[ I^h u - u^h, \]

and the algebraic error:

\[ u^h - \tilde{u}^h, \]

Where \( \tilde{u}^h \) is our current approximation to \( u^h \).
Note: Neither the solution, $u^h$, nor the discretization error are smoothed by relaxation, only the algebraic error. Hence, we formulate our problem in terms of this error.

Denote

$$\nu^h = u^h - \tilde{u}^h.$$  

Recall

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

Subtract $L^h \tilde{u}^h$ from both sides, and use the linearity of $L^h$ to obtain:

$$L^h \nu^h = f^h - L^h \tilde{u}^h \equiv r^h$$  \hspace{1cm} (8)
As we have seen, we need to smooth the error $v^h$ on the fine grid first, and only then solve the coarse-grid problem. Hence, we need two types of integrid transfer operations:

1. A Restriction (fine-to-coarse) operator: $I^H_h$.
2. A Prolongation (coarse-to-fine) operator: $I^h_H$.

For restriction we can often use simple injection, but full-weighted transfers are preferable.

For prolongation linear interpolation (bi-linear in 2D) is simple and usually effective.
Two-grid Algorithm

- Relax several times on grid $h$, obtaining $\tilde{u}^h$ with a smooth corresponding error.
- Calculate the residual: $r^h = f^h - L^h \tilde{u}^h$.
- Solve approximate error-equation on the coarse grid:
  \[ L^H v^H = f^H \equiv I^H_h r^h. \]
- Interpolate and add correction:
  \[ \tilde{u}^h \leftarrow \tilde{u}^h + I^h_H v^H. \]
- Relax again on grid $h$.

Multi-grid is obtained by recursion.
**Multi-grid Cycle**

Let $u^{2h}$ approximate $v^{2h}$, $u^{4h}$ approximate the error on grid $2h$, etc.

Relax on $L^h u^h = f^h$ $v_1$ times

Set $f^{2h} = I^{2h}_h (f^h - L^h u^h)$, $u^{2h} = 0$

Relax on $L^{2h} u^{2h} = f^{2h}$ $v_1$ times

Set $f^{4h} = I^{4h}_{2h} (f^{2h} - L^{2h} u^{2h})$, $u^{4h} = 0$

Relax on $L^{4h} u^{4h} = f^{4h}$ $v_1$ times

Set $f^{8h} = I^{8h}_{4h} (f^{4h} - L^{4h} u^{4h})$, $u^{8h} = 0$

... Solve $L^{Mh} - u^{Mh} = f^{Mh}$

...

Correct $u^{4h} \leftarrow u^{4h} + I^{4h}_{8h} u^{8h}$

Relax on $L^{4h} u^{4h} = f^{4h}$ $v_2$ times

Correct $u^{2h} \leftarrow u^{2h} + I^{2h}_{4h} u^{4h}$

Relax on $L^{2h} u^{2h} = f^{2h}$ $v_2$ times

Correct $u^{h} \leftarrow u^{h} + I^{h}_{2h} u^{2h}$

Relax on $L^{h} u^{h} = f^{h}$ $v_2$ times
**V cycle**

- **Finest grid**
- **Coarsest grid**

- **RELAXATION**
- **RESTRICTION**
- **PROLONGATION**
Multigrid vs. Relaxation

Residual convergence histories, 128 by 128 grid

- V(1,1) Cycles
- Red-Black Relaxation
Remarks:

1. Simple recursion yields a V cycle. More generally, we can choose a cycle index $\gamma$, and define a $\gamma$-cycle recursively as follows: Relax; transfer to next coarser grid; perform $\gamma \gamma$-cycles; interpolate and correct; Relax. (On the coarsest grid define the $\gamma$-cycle as an exact solution).

2. The best number of pre-relaxation + post-relaxation sweeps is normally 2 or 3.

3. The boundary conditions for all coarse-grid problems is zero (because the coarse-grid variable is the error). The initial guess for the coarse-grid solution must be zero.
Suggestions and Debugging Hints

• Build your program gradually, beginning with the relaxation routine. Test your functions as you progress.

• Make your program modular (relaxation, application of the operator, restriction, interpolation, etc.)

• Two arrays are required on each grid: one for the solution and one for the right-hand side.

• For debugging purposes, print out the residual norm after every relaxation sweep on every grid.
• Test relaxation to make sure the $L_2$ residual norm is reduced by every relaxation sweep. Also, the right-hand side and the boundary values should not be altered by the relaxation.

• Perform the test of multiplying the residuals by zero before transferring to the coarser grid. The residuals should then remain zero on all the coarse grids, and the multigrid solver should perform like pure relaxation.

• Check that the interpolation is accurate everywhere for a very smooth function.

• Solve a problem for which the solution to the differential equation is known. Compare the discrete solution to the differential solution on different grids to test the discretization.
• Run many cycles with zero right-hand sides and boundary conditions, to avoid the effect of round-off errors. Compare the asymptotic convergence rates to the predictions of local mode analysis.

• Check for regular behavior of the residual norms on different grids. Scale properly to allow such a comparison.

• Check whether residuals are transferred properly to the coarser grid near boundaries and corners.

• Compare different boundary conditions, including periodic. Extra processing near boundaries is sometimes crucial for optimal behavior.
Troubleshooting: IR/ICG Analysis

Diskin et al., “On Quantitative Analysis Methods for Multigrid Solutions”, SISC.

What if the code is ok, and yet convergence is still not satisfactory?

The main question is whether the problem lies in the smoothing process or in the coarse-grid correction.

To pinpoint the deficiency, we set the right-hand side and boundary conditions to zero, and perform an IR/ICG analysis (Idealized Relaxation / Idealized Coarse-Grid correction).
IR/ICG Analysis

This is a practical two-level test, carried out within the code. First, we replace the actual relaxation by an Idealized Relaxation operator (IR) - a local averaging operator, e.g.,

\[
\text{IR} = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}.
\]

The IR operator can be applied more than once. Then we test the two-level convergence factor, starting with a random guess and using, say, two IR “sweeps”. Note that the error is equal to the approximate solution, because the exact solution is zero.

Note: verify that the coarse-grid problem is solved very well.
**IR/ICG Analysis**

Next, we use the actual relaxation, but replace the actual coarse-grid correction by an Idealized Coarse-Grid Correction operator, ICG:

\[
\text{ICG} = I^h - I^h H I^H_h.
\]

That is, instead of applying the actual coarse-grid correction, we restrict the approximate solution to the coarse grid, interpolate it back to the fine grid, and subtract what we get from the current fine-grid approximation. Again we monitor the convergence factor. This is the ICG analysis.
IR/ICG Analysis

Finally, we also compute the convergence factor of the Reference Cycle, obtained by using both IR and ICG instead of the actual operators.

We compare the four convergence factors. That of the actual cycle is presumably poor. That of the reference cycle must be good.

If the IR cycle also yields good convergence, we conclude that the problem is with the actual relaxation. On the other hand, if the ICG cycle yields good convergence, we conclude that the problem is with the actual coarse-grid correction.
Exercise 3

An alternative, more general, choice for the IR operator in the IR/ICG analysis is given by

$$\text{IR} = \left(I_H^h I_h^H\right).$$

Show that if the restriction is the transpose of the prolongation, properly scaled such that

$$\rho\left(I_H^h I_h^H\right) = 1,$$

where $\rho$ denotes the spectral radius, then...
Exercise 3

The asymptotic convergence factor of the reference cycle with $\nu$ IR sweeps, defined by

$$
\rho \left[ \left( \mathbf{I}^h - \mathbf{I}^h_H \mathbf{I}^H_h \right) \left( \mathbf{I}^h_H \mathbf{I}^H_h \right)^\nu \right],
$$

is bounded by

$$
\frac{\nu^\nu}{(\nu + 1)^{\nu+1}},
$$

e.g., 0.25, 0.148, 0.105, for $\nu = 1, 2, 3$, resp.
Exercise 4

\[-\varepsilon u_{xx} - u_{yy} = f(x, y) \quad , \quad u \in \Omega\]

\[u = g(x, y) \quad , \quad u \in \partial \Omega,\]

where \(\Omega = (0,1) \times (0,1)\).

Discretize the problem as usual on a square grid of \((2^n + 1) \times (2^n + 1)\) points (including boundaries).

Define \(g\) and \(f\) such that the solution to the differential equation is

\[u = 2 \sin(\pi x) + \sin(2\pi y)\]
Implement $V(2,1)$ cycles using $n$ levels, with a random initial guess. Use full-weighted restriction, bi-linear interpolation, and the following relaxation methods:

1. Damped Jacobi, $0 \leq \omega \leq 1$;
2. Gauss-Seidel in Red-Black order.
Perform many 30 $V$-cycles, until convergence typically slows down and stops due to round-off errors. Compute the normalized $L_2$ norm (root-mean-square) of the residual after each cycle. Plot the residual reduction rate per cycle:

$$\left\| r^{h(\ell)} \right\|_2 / \left\| r^{h(\ell-1)} \right\|_2$$

where $\ell$ is the iteration number. (Each $V$-cycle is one iteration).

Use your code to answer the following questions. Explain all the results. Use at least $n = 7$ (except where stated otherwise).
a) For $\epsilon = 1$, which relaxation method is best? What is the optimal $\omega$ for Jacobi relaxation found in practice, and how does it compare to the theoretical optimum? How do the results compare with the predictions of smoothing analysis?

b) What is the influence of different (but only positive) $\epsilon$? Compare $\epsilon >> 1, \epsilon \approx 1, \epsilon << 1$.

c) For $\epsilon = 1$, compare between the convergence behavior of the residual norm and that of the “total error”, $\| u - \tilde{u}^n \|_2$, for $n = 4, 5, 6, 7$. How are these behaviors influenced by the resolution? Give quantitative answers and explain if they match our expectations.
d) For $\varepsilon = 1$, discretize the problem using the skew Laplacian:

$$\frac{1}{2h^2} \begin{bmatrix} -1 & 0 & -1 \\ 0 & 4 & 0 \\ -1 & 0 & -1 \end{bmatrix}$$

Test the residual convergence. Use the IR/ICG two-grid analysis to determine the cause of the problem. Explain.
Local Mode Analysis (LMA)

We would like to obtain a quantitative prediction of the convergence behavior of the multigrid (or at least two-grid) cycle.

This is important for debugging, choosing parameters, etc.

We first derive the iteration matrix of the two-grid cycle. That is, the matrix $S^h$, 

$$
\mathbf{v}^h_{after} = S^h \mathbf{v}^h_{before},
$$

where $\mathbf{v}^h_{before}$ and $\mathbf{v}^h_{after}$ are the algebraic errors before and after the two-grid cycle.
**Notation**

- $R^h$ – Relaxation matrix
- $L^h$ – Fine-grid matrix, $L^H$ – Coarse-grid matrix
- $I_H^h$ – Prolongation matrix, $I^H_h$ – Restriction matrix
- $I^h$ – Fine-grid identity matrix
- $\nu_1$ – # Relaxations before CGC
- $\nu_2$ – # Relaxations after CGC

**Two-grid matrix**

\[ S^h = \left( R^h \right)^{\nu_2} \left( I^h - I_H^h \left( L^H \right)^{-1} I^H_h L^h \right) \left( R^h \right)^{\nu_1} \]
For a given problem, we can compute the norm of $S^h$ and determine the convergence behavior of the two-grid algorithm, which often provides a relevant approximation of the multigrid performance.

However, this requires use of a computer, and it is only moderately useful for algorithm development.

We can in fact obtain a useful quantitative approximate prediction by means of a local (Fourier) analysis.
Fourier Analysis

Consider for simplicity the 1D problem.

Instead of fixed boundary conditions we assume periodicity. Also, we assume our operator, $L^h$, to have constant coefficients. Hence, every element of the corresponding matrix, denoted by $A$, satisfies:

$$A_{i,j} = A_{i-1,j-1} \pmod{N}.$$ 

That is, every row of $A$ is identical to the previous row, modulo $N$, shifted one place forward.

We next compute the eigenvectors and eigenvalues of matrices of this type.
Observation: any matrix $A$ representing a constant-coefficient discretization + periodicity can be written as a polynomial in the cyclic forward shift matrix,

$$
\omega = \begin{pmatrix}
0 & 1 & 0 & \cdots & 0 & 0 \\
0 & 0 & 1 & 0 & \cdots & 0 \\
0 & 0 & 0 & 1 & \cdots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & 0 & \cdots & 0 & 1 \\
1 & 0 & 0 & \cdots & 0 & 0 \\
\end{pmatrix}.
$$

Here,

$$
\omega_{i,j} = \delta_{i+1,j} \pmod{N}.
$$
Proof.

By induction, for any integer \( p \),

\[
\left( \omega^p \right)_{i,j} = \delta_{i+p,j \mod N}.
\]

That is, \( \omega^p \) is a cyclic forward shift by \( p \) places. Hence we have

\[
A = \sum_{j=0}^{N-1} A_{0,j} \omega^j.
\]
**Observation:** If $P$ is a polynomial and $B$ is a square matrix with eigenvector $v$ and corresponding eigenvalue $\lambda$, then $v$ is also an eigenvector of $P(B)$ with corresponding eigenvalue $P(\lambda)$.

**Proof:**

Suppose $P(B) = \sum_{i=0}^{m} c_i B^i$.

Observe that, by induction, $B^i v = \lambda^i v$.

Hence,

$$P(B)v = \sum_{i=0}^{m} c_i \lambda^i v = P(\lambda)v$$
Conclusion: Since any matrix $A$ of the type we are considering is a polynomial in $\omega$, we only need to compute the eigenvalues and eigenvectors of $\omega$. The corresponding eigenvalues of $A$ will then be easy to compute.

Let $\lambda_k$ denote an eigenvalue of $\omega$, and let $v^k$ denote the corresponding eigenvector. We have:
Hence,
\[ \mathbf{v}_1^k = \lambda_k \mathbf{v}_0^k, \]
\[ \mathbf{v}_2^k = \lambda_k \mathbf{v}_1^k = \lambda_k^2 \mathbf{v}_0^k, \]
\[ \ldots \]
\[ \mathbf{v}_j^k = \lambda_k^j \mathbf{v}_0^k, \]
\[ \ldots \]
\[ \mathbf{v}_0^k = \lambda_k \mathbf{v}_{N-1}^k = \lambda_k^N \mathbf{v}_0^k. \]

Thus, the \( N \) eigenvalues are the \( N \) th roots of 1:

\[ \lambda_k = 1^{1/N} = e^{i2\pi k/N}. \]
The $N$ eigenvectors are

$$v^k = \begin{pmatrix} 1 \\ \lambda_k \\ \lambda_k^2 \\ \vdots \\ \lambda_k^j \\ \vdots \\ \lambda_k^{N-1} \end{pmatrix} = \begin{pmatrix} 1 \\ e^{ik2\pi/N} \\ e^{2ik2\pi/N} \\ \vdots \\ e^{jik2\pi/N} \\ \vdots \\ e^{(N-1)ik2\pi/N} \end{pmatrix}. $$

Here, $k$ is any integer. But note that if $k_1 = k_2 \pmod{N}$ then the eigenvalues corresponding to $k_1$ and $k_2$ alias. Thus, $k$ runs over any $N$ consecutive integers.
Consider now a grid of \( N \) equally spaced points over a domain of size \( 2\pi \). The grid-points are located at

\[
x_j = hj = \frac{2\pi j}{N}, \quad j = 0, \ldots, N - 1,
\]

where \( h = 2\pi / N \) is the mesh-size.

The eigenvectors can be considered as grid-based functions of the form:

\[
\phi_k(x_j) = e^{ikx_j}.
\]

Here, \( k \) can be any integer, but note that \( \phi_k \) aliases with \( \phi_{k+N} \). We therefore restrict:

\[
k \in \left[ -\frac{N}{2} + 1, \frac{N}{2} \right].
\]
The functions $\varphi_k$ are called **Fourier components**. They possess some useful properties. Amongst these:

Define an inner product,

$$ (f, g) = \frac{1}{N} \sum_{j=0}^{N-1} f(x_j) \overline{g(x_j)}, $$

where “bar” denotes complex conjugate. Then the Fourier components are **orthogonal**:

$$ (\varphi_k, \varphi_l) = \delta_{k,l} \, (\text{mod} \, N). $$
The Fourier expansion is unique:

\[
f(x) = \sum_{k=0}^{N-1} c_k \varphi_k(x),
\]

where the coefficients \(c_k\) are uniquely determined by

\[
c_k = (f, \varphi_k).
\]

Furthermore,

\[
\sum |c_k|^2 = (f, f).
\]

(Parseval’s theorem).
Additional Remarks.

1. Domain length $2L$: replace $x$ by $\pi x/L$.

2. Nonperiodic, with function vanishing at endpoints: Antisymmetric continuation (sine series).

3. The wavelength is $l=2\pi/k$.

4. For multigrid analysis we define

$$\theta = hk = \frac{2\pi k}{N}, \quad -\pi < \theta \leq \pi.$$ 

Then,

$$\varphi_\theta(x_j) = e^{i\theta x_j / h} = e^{i\theta j}.$$ 

The wavelength is $2\pi h/\theta$. 
Fourier Analysis

Local mode (Fourier) analysis is the main tool used for practical analysis of multigrid solvers. Though it is rigorously justified only for rather special situations, it is useful for quantitative predictions in a wide set of circumstances.

The underlying assumption is that small subsets comprised of one or a few Fourier components of the form (in $d$ dimensions)

$$\varphi = e^{i\theta \cdot j},$$

$$\theta = (\theta_1, \theta_2, \ldots, \theta_d),$$

$$j = \begin{pmatrix} x_1/h_1, & x_2/h_2, & \ldots, & x_d/h_d \end{pmatrix},$$

are invariant under operations of the common multigrid components.
The Symbol

The symbol is a generalization of the eigenvalue. The symbol of an operator $L$ is denoted by $\hat{L}$. When the Fourier mode is an eigenfunction, it is defined by:

$$Le^{i\theta \cdot j} = \hat{L}(\theta)e^{i\theta \cdot j}.$$ 

Examples (1D):

$$Lu = u_{xx},$$

$$Le^{i\theta x/h} = -\frac{\theta^2}{h^2}e^{i\theta x/h},$$

$$\Rightarrow \hat{L} = -\frac{\theta^2}{h^2}$$
Suppose we discretize $L$ by

$$L^h u^h = \frac{u_{j+1}^h - 2u_j^h + u_{j-1}^h}{h^2}.$$  

Then,

$$L^h e^{i\theta x/h} = \frac{e^{i\theta} - 2 + e^{-i\theta}}{h^2} e^{i\theta x/h}.$$  

So,

$$\hat{L}^h(\theta) = \frac{2}{h^2}(\cos(\theta) - 1) = -\frac{4}{h^2} \sin^2\left(\frac{\theta}{2}\right).$$  

Truncation error:

$$\hat{L}(\theta) - \hat{L}^h(\theta) = -\frac{\theta^2}{h^2} + \frac{4}{h^2} \sin^2\left(\frac{\theta}{2}\right) = O\left(\frac{\theta^4}{h^2}\right).$$
The symbol of relaxation

Consider the discretized equation

\[ L^h u^h = f^h. \]

A pointwise relaxation can often be written as

\[ L^{h+} u^h_{after} + L^{h-} u^h_{before} = f^h, \]

where \( u^h_{before} \) denotes the old approximation to \( u^h \)
(before the relaxation step), and \( u^h_{after} \) denotes
the new approximation.

Thus, the relaxation is characterized by the splitting:

\[ L^h = L^{h+} + L^{h-}. \]
Examples:

For 1D Poisson,

\[ L^h u^h = \frac{u^h_{j+1} - 2u^h_j + u^h_{j-1}}{h^2}, \]

we obtain for Jacobi relaxation,

\[ (L^+ h u^h_{after})_j = \frac{-2(u^h_{after})_j}{h^2}, \quad (L^- h u^h_{before})_j = \frac{(u^h_{before})_{j+1} + (u^h_{before})_{j-1}}{h^2}, \]

and for Gauss-Seidel relaxation,

\[ (L^+ h u^h_{after})_j = \frac{(u^h_{after})_{j-1} - 2(u^h_{after})_j}{h^2}, \quad (L^- h u^h_{before})_j = \frac{(u^h_{before})_{j+1}}{h^2}, \]
Let $v^h_{\text{before}} = u^h_{\text{before}} - u^h$ denote the algebraic error before the relaxation, and let $v^h_{\text{after}} = u^h_{\text{after}} - u^h$ denote the new error. Then,

$$L^{h+} v^h_{\text{after}} + L^{h-} v^h_{\text{before}} = 0.$$ 

Now, consider an error that is a single Fourier component (assumed to be an eigenfunction of $L^{h\pm}$):

$$v^h_{\text{before}} = A_\theta e^{i\theta x / h}, \quad v^h_{\text{after}} = \overline{A}_\theta e^{i\theta x / h}.$$
The relaxation operator, $R^h$, is defined by

$$v^h_{\text{after}} = R^h v^h_{\text{before}}.$$ 

We obtain that the symbol of $R^h$ is:

$$\hat{R}^h(\theta) = \frac{\bar{A}_\theta}{A_\theta} = -\frac{\hat{L}^{h-}(\theta)}{\hat{L}^{h+}(\theta)}.$$ 

**Examples:** for Jacobi relaxation we have

$$\hat{L}^{h+}(\theta) = \frac{-2}{h^2}, \quad \hat{L}^{h-}(\theta) = \frac{e^{i\theta} + e^{-i\theta}}{h^2},$$

So the symbol of Jacobi relaxation is

$$\hat{R}^h_{\text{Jac}}(\theta) = \frac{e^{i\theta} + e^{-i\theta}}{2} = \cos(\theta).$$
For damped Jacobi relaxation we have

\[
\overline{A}_\theta = \left[ 1 - \omega + \omega \hat{R}_{\omega, Jac}^h (\theta) \right] A_\theta
\]

\[
= \left[ 1 - \omega + \omega \cos(\theta) \right] A_\theta.
\]

So the symbol of damped Jacobi relaxation is

\[
\hat{R}_{\omega, Jac}^h (\theta) = 1 - \omega + \omega \cos(\theta).
\]
For Gauss-Seidel relaxation we have
\[
\left( v_{\text{before}}^h \right)_{j+1} - 2 \left( v_{\text{after}}^h \right)_j + \left( v_{\text{after}}^h \right)_{j-1} = 0.
\]
Substituting the Fourier function yields
\[
A_\theta e^{i\theta(j+1)} - 2 \bar{A}_\theta e^{i\theta j} + \bar{A}_\theta e^{i\theta(j-1)} = \left[ A_\theta e^{i\theta} - 2 \bar{A}_\theta + \bar{A}_\theta e^{-i\theta} \right] e^{i\theta j} = 0.
\]

Hence,
\[
\bar{A}_\theta = \frac{e^{i\theta}}{2-e^{-i\theta}} A_\theta.
\]
The symbol of Gauss-Seidel relaxation is therefore
\[
\hat{R}_{GS}^h (\theta) = \frac{e^{i\theta}}{2-e^{-i\theta}}.
\]
Aliasing Revisited

Note that

\[ e^{i(\theta \pm 2\pi) x_j / h} = e^{i(\theta \pm 2\pi) j} = \]

\[ e^{\pm i 2\pi j} e^{i\theta j} = e^{i\theta j} = e^{i\theta x_j / h}. \]

That is, Fourier modes \( e^{i\theta x / h} \) and \( e^{i(\theta \pm 2\pi) x / h} \) alias with each other on grid \( h \).

On grid \( 2h \) the component \( e^{i\theta x / h} \) becomes \( e^{i2\theta x / 2h} \).

That is, its frequency relative to the grid is doubled.
**Aliasing Revisited**

Fourier modes $e^{i\theta x/h}$ and $e^{i(\theta \pm 2\pi)x/h}$ alias with each other on grid $h$.

On grid $2h$ the component $e^{i\theta x/h}$ becomes $e^{i2\theta x/2h}$. That is, its frequency relative to the grid is doubled.

Thus, the fine-grid components $e^{i\theta x/h}$ and $e^{i(\theta \pm \pi)x/h}$ alias when sampled on grid $2h$.

**Conclusion:** the coarse grid $2h$ resolves only about $\frac{1}{2}$ of the fine-grid frequencies – those in the range $|\theta| \leq \pi / 2$. 
**Smoothing analysis**

We simplify the analysis of the two-grid algorithm by making the following approximation:

1. The coarse-grid correction eliminates all smooth fine-grid components, those with $|\theta| \leq \pi / 2$.

2. The coarse-grid correction has no effect on the rough fine-grid error components, with $|\theta| > \pi / 2$.

With these simplifications we can predict approximately the convergence rate per fine-grid relaxation sweep of the two-grid cycle by computing the smoothing factor defined below.
The smoothing factor

Let $\hat{R}^h$ denote the symbol of a relaxation operator whose eigenvectors are Fourier components. Then the smoothing factor is defined by

$$\mu = \max_{\pi/2 \leq |\theta| \leq \pi} \left| \hat{R}^h (\theta) \right|.$$
Example:

The smoothing factor of Gauss-Seidel relaxation is

\[
\mu = \max_{\pi/2 \leq \theta \leq \pi} \left| \hat{R}_{GS}^h (\theta) \right|
\]

\[
= \max_{\pi/2 \leq \theta \leq \pi} \left| \frac{e^{i\theta}}{2 - e^{-i\theta}} \right|
\]

\[
= \max_{\pi/2 \leq \theta \leq \pi} \left| \frac{1}{2 - \cos(\theta) + i \sin(\theta)} \right|
\]

\[
= \max_{\pi/2 \leq \theta \leq \pi} \left| \frac{1}{\sqrt{(2 - \cos(\theta))^2 + \sin^2(\theta)}} \right| = \frac{1}{\sqrt{5}}.
\]

The maximum is obtained at \( \theta = \pm \pi / 2 \).
The smoothing factor of Jacobi relaxation with no damping is 1, because $|\cos(\pm \pi)| = 1$. Jacobi relaxation does not smooth errors at the highest frequencies, only changes their signs.

What is the smoothing factor of damped Jacobi? What is the optimal damping? That is, what is the $\omega$ which minimizes $\mu$ of damped Jacobi? What is the corresponding $\mu$?
Higher dimensions

The Fourier analysis can be generalized to $d$ dimensions.

$$\mathbf{x}_j = \left( x_{j_1}^{(1)}, x_{j_2}^{(2)}, \ldots, x_{j_d}^{(d)} \right), \quad \theta = (\theta_1, \theta_2, \ldots, \theta_d),$$

where

$$x_{j_k}^{(k)} = j_k \ h_k, \quad k = 1, \ldots, d.$$
Higher dimensions

The Fourier components are

$$\varphi_{\theta}(x_j) = e^{i \left( \sum_{k=1}^{d} \theta_k x_{jk}^{(k)} / h_k \right)} = e^{i \left( \sum_{k=1}^{d} \theta_k j_k \right)},$$

Where $h_k$ is the mesh-size in the $k$th coordinate.
The smoothing factor (for standard coarsening) is now defined as in the 1D case, with

\[ |\theta| = \max (|\theta_1|, \ldots, |\theta_d|). \]

**Example (2D)**

The symbol of Gauss-Seidel Relaxation for the Poisson problem is

\[ \hat{R}(\theta) = \frac{e^{i\theta_1} + e^{i\theta_2}}{4 - e^{-i\theta_1} - e^{-i\theta_2}}. \]

The smoothing factor \( \mu \) is the maximum of \( |\hat{R}(\theta)| \) over all \( \theta \) for which the absolute value of at least one component is at least \( \pi/2 \).
The shaded area marks the part of the $\theta$ domain that is ignored when computing the smoothing factor.
The smoothing factor is most easily computed approximately by a small computer program. For Gauss-Seidel relaxation of the 5-point Laplacian, the smoothing factor is found to be $\frac{1}{2}$.

**Conclusion:** a $V(2,1)$ cycle is expected to reduce the error approximately by a factor 8 per cycle.
Exercise 5

Write a MATLAB function that computes approximately the smoothing factor, $\mu$. The input should be two small matrices representing the stencils of $L^{h+}$ and $L^{h-}$, and also a damping parameter, $\omega$. The function should compute the symbol of the relaxation, and maximize its absolute value over a discrete subset spanning the high frequencies,

$$
(\theta_1, \theta_2) \subset [-\pi, \pi] \times [-\pi, \pi] \\setminus \left[ \left[-\frac{\pi}{2}, \frac{\pi}{2}\right] \times \left[ -\frac{\pi}{2}, \frac{\pi}{2}\right] \right].
$$

The resolution of the $\theta$’s should also be a parameter (run with 65 by 65). Use the program to verify the analytical results of the smoothing analysis for Jacobi relaxation (presented later). Verify also that the smoothing factor of point Gauss-Seidel for the 5-point Laplacian is 0.5.
For example, for Gauss-Seidel the input is

\[
L^{h+} = \begin{bmatrix} 0 & 0 & 0 \\ 1 & -4 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \quad L^{h-} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix},
\]

whereas for Jacobi it is

\[
L^{h+} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & -4 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad L^{h-} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}.
\]

The symbol of the relaxation is then given by

\[
R^h = 1 - \omega + \omega \left( \frac{\hat{L}^{-}}{\hat{L}^{+}} \right).
\]
Ellipticity and $h$-Ellipticity

Partial differential operators (PDO) are classified according to type. The type of operator is determined by the symbol of the principal part (highest-derivative terms in each coordinate).

Let $L_p$ be the principal part of a PDO $L$. The symbol of $L_p$ is defined by

$$L_p e^{i\mathbf{k} \cdot \mathbf{x}} = \hat{L}_p (\mathbf{k}) e^{i\mathbf{k} \cdot \mathbf{x}},$$

where

$$\mathbf{k} = (k_1, \ldots, k_d), \quad \mathbf{x} = (x_1, \ldots, x_d).$$
Definition:

If the equation

\[ \hat{L}_p (k) = 0 \]

has no real roots except \( k = 0 \), then \( L \) is said to be elliptic.

Example:

The Laplacian.

\[ L = L_p = \partial_{xx} + \partial_{yy} , \]
\[ \hat{L}_p = - |k|^2 . \]

Thus, \( L \) is clearly elliptic.
More generally, if

\[ L = A \partial_{xx} + 2B \partial_{xy} + C \partial_{yy} + D \partial_x + E \partial_y + F, \]

where \( A, B, C, D, E, F, \) are functions of \( x \) and \( y, \) then \( L \) is elliptic at a point \((x,y)\) if at this point

\[ AC - B^2 > 0. \]

If \( L \) is elliptic at every point in the domain then \( L \) is called elliptic.

Solutions of elliptic boundary-value problems usually satisfy certain smoothness properties, whereas nonelliptic problems often exhibit discontinuous solutions.
Multigrid methods are particularly useful and often straightforward for **elliptic problems**. But the “on-off” definition of ellipticity is inadequate for numerical purposes, and a quantitative measure of “ellipticity” of the discrete operator is important. This is given by the \( h \)-ellipticity measure, \( E^h \), defined by

\[
E^h \left( L^h \right) = \frac{\min_{\pi/2 \leq |\theta| \leq \pi} \left| \hat{L}^h (\theta) \right|}{\max_{\pi/2 \leq |\theta| \leq \pi} \left| \hat{L}^h (\theta) \right|},
\]

where \( \hat{L}^h \) is the symbol of \( L^h \).
We say that a discrete operator, \( L^h \), is \( h \)-elliptic if \( E^h \) is bounded away from zero. Generally, for ordinary (i.e., local) relaxation methods, larger \( E^h \) corresponds to better “error-smoothability” by local processing.

Basically, a large \( E^h \) implies that all high-frequency errors generate relatively large residuals that are roughly of the same size. If \( E^h \) is small, then there are some error components whose residuals are relatively small. Such components cannot be detected locally, and hence they cannot be reduced efficiently by a local relaxation.

**Remark:** an elliptic PDO may have a discretization that is *not* \( h \)-elliptic, while a nonelliptic PDO might have one that *is*. 
Examples:

Consider three different stencils for the Laplacian.

\[
L^h = \frac{1}{h^2} \begin{bmatrix}
  . & 1 & . \\
  1 & -4 & 1 \\
  . & 1 & . 
\end{bmatrix}.
\]

(1)

Here,

\[
\hat{L}^h (\theta) = \frac{1}{h^2} (2 \cos \theta_1 + 2 \cos \theta_2 - 4) = -\frac{4}{h^2} \left( \sin^2 \left( \frac{\theta_1}{2} \right) + \sin^2 \left( \frac{\theta_2}{2} \right) \right).
\]

Hence, \( E^h = 1/4 = O(1) \), and we say that \( L^h \) is \( h \)-elliptic.
Next, choose

\[ L^h = \frac{1}{2h^2} \begin{bmatrix} 1 & \cdot & 1 \\ \cdot & -4 & \cdot \\ 1 & \cdot & 1 \end{bmatrix}. \] (2)

Here,

\[ \hat{L}^h (\theta) = \frac{2}{h^2} (\cos \theta_1 \cos \theta_2 - 1), \]

and

\[ E^h = 0. \]

**Conclusion:** if this discretization is used, the error cannot be smoothed efficiently by local relaxation.
Finally, choose

\[
L^h = \frac{1}{4h^2} \begin{bmatrix}
  
  & 1 & 
  & &  
  & & 
  & & 1 
  
  & & 
  & & 1
  1 & -4 & 1 
  & & 
  & & 1 
  & & 
  & & 1 
\end{bmatrix}.
\]

(3)

Here,

\[
\hat{L}^h(\theta) = \frac{1}{4h^2} \left( 2 \cos 2\theta_1 + 2 \cos 2\theta_2 - 4 \right) = -\frac{4}{h^2} \left( \sin^2 \theta_1 + \sin^2 \theta_2 \right).
\]

Again \( E^h = 0 \). Note that in the last two examples, the Red-Black mode could not be smoothed.
Smoothing and $h$-ellipticity

$h$-ellipticity is a necessary and sufficient condition for the existence of pointwise smoothers based on a local splitting,

$$L^h = L^{h^+} + L^{h^-},$$

where $L^{h^+}$ is comprised of the coefficients of variables already relaxed, while $L^{h^-}$ is comprised of the coefficients of unrelaxed variables.

Obviously, $E^h (L^h) = 0$ implies $\hat{L}^h (\theta^*) = 0$ for some high frequency $\theta^*$, and therefore

$$L^{h^+} (\theta^*) = -L^{h^-} (\theta^*).$$
For \( \hat{L}^{h+}(\theta^*) \neq 0 \), we get

\[
\left| \hat{R}(\theta^*) \right| = \left| \frac{\hat{L}^{h-}(\theta^*)}{\hat{L}^{h+}(\theta^*)} \right| = 1.
\]

Thus the smoothing factor is at best 1.
Exercise 6

Show that optimally damped Jacobi relaxation yields for symmetric constant-coefficient operators, a smoothing factor

$$\mu = \frac{1 - E^h}{1 + E^h},$$

where $E^h$ is the $h$-ellipticity measure. (Note that the symmetry of the operators implies that the symbols are real).

**Hint:** without loss of generality, assume that $L^h$ is normalized such that its diagonal is the identity matrix. Then, write the damped Jacobi relaxation matrix in terms of $L^h$. 
For non-symmetric stencils we can obtain the bound

$$\mu \leq \frac{1 - (E^h)^2}{1 + (E^h)^2} < 1.$$ 

By means of a distributive relaxation. However, in practice we can usually find far simpler and more effective smoothers.
Anisotropic Diffusion

Let

$$Lu = u_{ss} + \varepsilon u_{tt}.$$  

Let $\phi$ be the angle between $(s,t)$ and the grid-aligned coordinate system, $(x,y)$. Hence,

$$Lu = \left(C^2 + \varepsilon S^2\right)u_{xx} + 2(1-\varepsilon)CSu_{xy} + \left(\varepsilon C^2 + S^2\right)u_{yy},$$

with

$$C = \cos(\phi), \quad S = \sin \phi.$$
Anisotropic Diffusion

We discretize $L$ using the stencil

$$L^h = \frac{1}{h^2} \begin{bmatrix} -\frac{1}{2} (1 - \varepsilon) CS & \varepsilon C^2 + S^2 & \frac{1}{2} (1 - \varepsilon) CS \\ C^2 + \varepsilon S^2 & -2(1 + \varepsilon) & C^2 + \varepsilon S^2 \\ \frac{1}{2} (1 - \varepsilon) CS & \varepsilon C^2 + S^2 & -\frac{1}{2} (1 - \varepsilon) CS \end{bmatrix}.$$ 

Exercise 7

Set $\varepsilon = 10^{-5}$ and perform the IR/ICG analysis with (a) $\phi = 0$, (b) $\phi = \pi/4$. 
Motivation: Selective Smoothing of Images.

Given an image, $I$, one method of denoising is by integrating the parabolic equation,

$$I_t = \Delta_g I$$

where $\Delta_g$ is the Beltrami operator, which is essentially a scaled second derivative in the direction normal to the gradient. Thus, the image is smoothed along edges, but the edges remain sharp.
\( \phi = 0 \) (aligned)
\[ \phi = \frac{\pi}{4} \text{ (rotated)} \]
**Anisotropic Diffusion: The Aligned case**

In the previous examples, of the Laplacian operator, we were able to find a discretization that would give us a good $h$-ellipticity measure. But suppose that the differential operator is

$$L = \varepsilon \partial_{xx} + \partial_{yy}$$
A 2\textsuperscript{nd} order discretization for the aligned linear diffusion operator is again obtained by the standard five-point stencil:

\[
L^h = \frac{1}{h^2} \begin{bmatrix}
\varepsilon & 1 & \\
-2(1+\varepsilon) & \varepsilon & \\
1 & 1 & 
\end{bmatrix}.
\]

Here,

\[
\hat{L}^h = \frac{1}{h^2} \left[ 2(\varepsilon \cos(\theta_1) + \cos(\theta_2) - 2(1+\varepsilon)) \right]
= -\frac{4}{h^2} \left[ \varepsilon \sin^2 \left( \frac{\theta_1}{2} \right) + \sin^2 \left( \frac{\theta_2}{2} \right) \right].
\]
Setting \((\theta_1, \theta_2) = (\pi/2, 0)\) vs. \((\pi, \pi)\), for example, we obtain \(E^h = O(\varepsilon), \ \varepsilon \to 0\), hence small \(h\)-ellipticity. Indeed, all error components which have high-frequency oscillations only in the \(x\) direction generate relatively small residuals. Such errors cannot be reduced efficiently by local relaxation.

For example, the symbol of the Gauss-Seidel relaxation for this operator is

\[
\hat{R}(\theta) = \frac{\varepsilon e^{i\theta_1} + e^{i\theta_2}}{2(1 + \varepsilon) - \varepsilon e^{-i\theta_1} - e^{-i\theta_2}}
\]  

(9)

setting \(\theta = (\pi, 0)\), for example we get

\[
|\hat{R}(\pi, 0)| = \frac{1-\varepsilon}{1+3\varepsilon} = 1 - 4\varepsilon + O(\varepsilon^2), \ \varepsilon \to 0
\]

(10)

Thus the smoothing factor is very poor for small \(\varepsilon\) as expected.
Treating Aligned Anisotropy

There are two general approaches for handling anisotropic operators. One method is to employ line-relaxation in the direction of the strong coupling (i.e., for which the coefficient is relatively large - the $y$ direction in this example.) This means that we relax simultaneously a full line of variables for each gridpoint index $i$ in the $x$ direction.

In our Gauss-Seidel example, the resulting relaxation symbol is

$$
\hat{R}(\theta) = \frac{\varepsilon e^{i\theta_1}}{2(1 + \varepsilon) - \varepsilon e^{-i\theta_1} - e^{i\theta_2} - e^{-i\theta_2}}
$$
Now, $|R(\theta)|$ is maximized over the high frequencies for $
abla = (\pi/2,0)$, yielding $|R(\pi/2,0)| = 1/\sqrt{5}$, for $\varepsilon \to 0$, which implies very good smoothing.

The drawback of this approach is that for each $y$-line we must solve a tri-diagonal system of equations. We can do this by the usual Gaussian elimination or by a 1D multigrid cycle.
An alternative approach for anisotropic operators is partial coarsening (or semi-coarsening). In this approach we use the usual relaxation, but we only coarsen in the direction in which the error is smoothed efficiently ($y$ in our example).

Thus, the relaxation symbol remains as usual, but the coarse grid resolves more components, and the definition of the smoothing factor changes accordingly.
In our example, the **smoothing factor** is given by

\[ \mu = \max_{\pi/2 \leq \theta \leq \pi} |\hat{R}(\theta)|, \]

where \( \hat{R}(\theta) \) is the usual Gauss-Seidel symbol (9). Given the range of \( \theta \), the maximum (for small \( \varepsilon \)) is now obtained at \( \theta = (0, \pi/2) \), yielding,

\[ |\hat{R}(\theta)| = \left| \frac{i + \varepsilon}{2(1 + \varepsilon) - \varepsilon - i} \right| \xrightarrow{\varepsilon \to 0} \frac{1}{\sqrt{5}}, \]

again implying good smoothing properties.
The General Rule of Block Relaxation

The general rule is that local (point) relaxation only smooths the error efficiently in the direction of the strongest coupling. If we relax the strongly-coupled variables simultaneously (block relaxation), then the relaxation will smooth well also in the direction of the second-strongest coupling. Thus, we can use block-relaxation (line, plane, etc.) to regain full multigrid efficiency.

Alternatively, we can refrain from coarsening in the directions along which the error is not smoothed. The ultimate form of this is Algebraic Multigrid (AMG).
Both techniques can be used simultaneously. For example, if we do not know a priori the direction of strong coupling, then we can use line relaxation in one direction while coarsening only in the other. This can be generalized to higher dimensions.

Alternatively, we can use line relaxation in each of the directions (alternating), but then the generalization to higher dimensions is more cumbersome.
Skilled use of relaxation parameters

As we have seen, Jacobi relaxation for the five-point Laplacian operator requires damping in order to be a useful smoother.

While this is not always true, optimal or near-optimal relaxation parameters can be crucial for maintaining good smoothing properties.

How should we choose the damping parameter?

Fourier smoothing analysis can often provide a good answer.
Skilled use of relaxation parameters

Consider a homogeneous second-order constant-coefficient PDO:

\[ L = c_{11} \frac{\partial}{\partial x} - 2c_{12} \frac{\partial}{\partial y} + c_{22} \frac{\partial}{\partial y}, \]

satisfying the ellipticity condition,

\[ c_{11}c_{22} > (c_{12})^2. \]

A generic second-order finite-difference discretization with 9-point stencil can be written as

\[ L^h = A\frac{\partial^h}{\partial x} - 2B\frac{\partial^h}{\partial y} + C\frac{\partial^h}{\partial y} + D\Delta_s, \]
Skilled use of relaxation parameters
where the stencils of the discrete operators are given by

\[ \partial^h_{xx} = h^{-2} \begin{bmatrix} 1 & -2 & 1 \end{bmatrix}, \quad \partial^h_{xy} = 0.25h^{-2} \begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix}, \]

\[ \partial^h_{yy} = h^{-2} \begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix}, \quad \Delta_s = 0.5h^{-2} \begin{bmatrix} 1 & 1 \\ -4 & 1 \end{bmatrix}, \]
Skilled use of relaxation parameters

and $A, B, C,$ and $D,$ satisfy

$$D + A = c_{11}, \quad B = c_{12}, \quad D + C = c_{22},$$

by consistency, and therefore,

$$D + A > 0, \quad D + C > 0,$$

$$(D + A)(D + C) > B^2,$$

by ellipticity. Also, without loss of generality, we normalize the coefficients to satisfy

$$A + C + D = 1.$$
Skilled use of relaxation parameters

The symbol of unweighted Jacobi relaxation is now given by

\[ s_1(\theta) = A \cos(\theta_1) + C \cos(\theta_2) + B \sin(\theta_1) \sin(\theta_2) + D \cos(\theta_1) \cos(\theta_2). \]

The symbol of \( \omega \)-weighted Jacobi is given by

\[ s_\omega = 1 - \omega + \omega s_1. \]

Recall that the smoothing factor is defined by

\[ \mu = \max_{\pi/2 \leq |\theta| \leq \pi} |s_\omega|. \]
Skilled use of relaxation parameters

Now, since $s_\omega$ depends bi-linearly on $s_1$, $\omega$, it follows that the optimal value of $\omega$ (i.e., that which minimizes the smoothing factor) depends only on the maximal and minimal values of $s_1$ in the high-frequency zone:

The optimal $\omega$ is that which satisfies:

$$
|1 - \omega + \omega s_{\text{min}}| = |1 - \omega + \omega s_{\text{max}}|.
$$
Skilled use of relaxation parameters

We thus obtain for the optimal relaxation parameter and the corresponding smoothing factor:

\[
\omega_{\text{opt}} = \frac{2}{2 - (s_{\text{max}} + s_{\text{min}})},
\]

\[
\mu_{\text{opt}} = \frac{(s_{\text{max}} - s_{\text{min}})}{2 - (s_{\text{max}} + s_{\text{min}})}.
\]

All we need to do, therefore, is to find expressions for \(s_{\text{min}}\) and \(s_{\text{max}}\).
Skilled use of relaxation parameters

Denote

\[ z_1 = \cos(\theta_1), \quad z_2 = \cos(\theta_2). \]

Plugging this into the symbol of undamped Jacobi, we find that we need to find the minimum and maximum of

\[ s_1(z_1, z_2) = Az_1 + Cz_2 + B\sqrt{(1-z_1^2)(1-z_2^2)} + Dz_1z_2, \]

over the high-frequency domain,

\[ Z_{hf} = [-1,1] \times [-1,1] \setminus (0,1] \times (0,1]. \]
Skilled use of relaxation parameters

Suppose first that $B = 0$. Then $s_1$ is bilinear in $z_1, z_2$. Therefore the extremal values can only occur at corners of the domain $Z_{hf}$.

Assuming without loss of generality $C \leq A$,

(else $A$ and $C$ are interchanged) we find

$$s_{\min} = -A - |D - C|,$$

$$s_{\max} = \max\left( A, |D - C| - A \right).$$

For the Laplacian operator, in particular, we get to choose some combination of $D$ and $A = C$. 
Skilled use of relaxation parameters

Examples:

For $D = 0, A = C = 1/2$, we get the usual five-point Laplacian. The optimal damping and corresponding smoothing factor are:

Undamped Jacobi yields $\mu = 1$.

For $D = 1, A = C = 0$, we get $\mu = 1$ for any damping (as also seen above).

However, for $A = C = D = 1/3$, undamped Jacobi is excellent, yielding a smoothing factor of $1/3$. 

$\omega_{\text{opt}} = 0.8, \quad \mu_{\text{opt}} = 0.6$. 
Skilled use of relaxation parameters

Examples:

For the bilinear-element Laplacian with square elements we have $D = 1/2, A = C = 1/4$, we get:

$$\omega_{\text{opt}} = 8/9, \quad \mu_{\text{opt}} = 1/3.$$  

Finally, the fourth-order compact Mehrstellenverfahren Laplacian is obtained for $D = 1/5, A = C = 2/5$, yielding:

$$\omega_{\text{opt}} = 10/11, \quad \mu_{\text{opt}} = 5/11.$$
Skilled use of relaxation parameters

Next, consider the case \( D = 0 \). Equating the derivatives of \( s_1 \) with respect to \( \theta_1 \) and \( \theta_2 \) to zero yields

\[
A \sin(\theta_1) = B \sin(\theta_2) \cos(\theta_1),
\]
\[
C \sin(\theta_2) = B \sin(\theta_1) \cos(\theta_2).
\]

Hence, for \( \sin(\theta_1)\sin(\theta_2) \neq 0 \), we obtain

\[
AC = B^2 \cos(\theta_1) \cos(\theta_2) \leq B^2,
\]

which contradicts the ellipticity assumption. Thus, the extrema must lie on the boundary of \( Z_{hf} \), and are given by:

\[
s_{\text{min}} = -1, \quad s_{\text{max}} = \sqrt{A^2 + B^2}.
\]
Skilled use of relaxation parameters

If we set $A = C = 1/2$, we find that damped Jacobi remains a useful smoother throughout the elliptic regime. However, it loses its efficiency in the anisotropic limit, $C \ll A$.


An alternative to tuning the damping parameter by Fourier analysis is suggested in Broeker et al., SISC 23 (4), 1395-1416 (2001): use SPAI for choosing the damping.

EXERCISE 4 (GOTO Slide 111)
Multigrid as Optimization

We examine the multigrid algorithm in the framework of convex optimization to gain insight into how to choose components of the multigrid cycle.

We first consider linear problems and then continue to nonlinear problems and introduce the nonlinear multigrid algorithm, know as FAS (Full Approximation Scheme algorithm).
The Linear Problem

Solve the differential equation:

\[ L(u) = f(x), \quad x \in \Omega, \]

where \( L(u) \) is some Symmetric Positive Definite (SPD) linear (partial) differential operator, plus appropriate boundary conditions.

Discretizing on some grid denoted by its typical mesh-size \( h \), we obtain a linear system:
The Discrete Linear Problem

Solve:

\[ L_h u_h = f_h, \]

where the \((\text{Sparse, Large and Ill-Conditioned - SLIC})\) SPD matrix \(L_h \in \mathbb{R}^{n \times n}\) is the discrete approximation to \(L\).

The vectors \(u_h \in \mathbb{R}^n, f_h \in \mathbb{R}^n\) are discrete approximations to the solution \(u\) and the given right-hand side \(f\).
Linear System Solver

For sufficiently SLIC problems, iterative methods are attractive.

Our personal favorite is of course multigrid methods - a recursive extension of the following two-grid approach.
Linear System Solver

\[ L_h u_h = f_h \]

Given some initial approximation to \( u_h \), apply first some simple relaxation (Gauss-Seidel, Jacobi...), obtaining a new approximation, denoted \( \tilde{u}_h \), with error denoted by

\[ e_h = u_h - \tilde{u}_h. \]
Linear System Solver

\[ e_h = u_h - \tilde{u}_h . \]

The error satisfies the linear system

\[ L_h e_h = L_h (u_h - \tilde{u}_h) = f_h - L_h \tilde{u}_h = r_h . \]

After suitable relaxation, the error is smooth relative to the grid, and this system can be approximated very well on a coarser grid, with mesh-size \( H > h \), as follows.
Coarse-Grid Correction

Define full-rank prolongation, restriction, and coarse-grid operator:

\[
P \in \mathbb{R}^{n \times N}, \quad R \in \mathbb{R}^{N \times n}, \quad L_H \in \mathbb{R}^{N \times N}, \quad N < n,
\]

and solve the coarse-grid problem:

\[
L_H e_H = R r_h.
\]

Apply coarse-grid correction:

\[
\tilde{u}_h \leftarrow \tilde{u}_h + P e_H.
\]

Multi-grid obtained by recursion.
Choosing the Operators

How should we choose $P$, $R$, and $L_H$?

Let us recast our problem as a convex functional minimization task:

$$u_h = \arg \min_{v_h \in \mathbb{R}^n} \left( \frac{1}{2} v_h^T L_h v_h - v_h^T f_h \right)$$

$$\iff$$

$$L_h u_h - f_h = 0.$$  

The gradient of the functional must vanish.
Choosing the Operators

Recall the coarse-grid correction step:

\[
\tilde{u}_h \leftarrow \tilde{u}_h + Pe_H.
\]

Given \( \tilde{u}_h \), our current approximation, we wish to add a correction \( Pe_H \) that will reduce the fine-grid functional as much as possible.

Note that the set of possible corrections is the space spanned by the columns of \( P \), called the range of \( P \).
Choosing the Operators

Plugging into the functional yields:

\[
e_H = \arg \min_{c_H \in \mathbb{R}^N} \left( \frac{1}{2} \left( \tilde{u}_h + Pc_H \right)^T L_h \left( \tilde{u}_h + Pc_H \right) - \left( \tilde{u}_h + Pc_H \right)^T f_h \right)
\]

⇔ The gradient with respect to \(c_H\) must vanish

\[
P^T L_h Pe_H + P^T \left( L_h \tilde{u}_h - f_h \right) = 0.
\]

\([L_H e_H = R r_h]\)
Choosing the Operators

Conclusions:

1. We must define $L_H = P^T L_h P$ (Galerkin coarsening) for the coarse-grid problem,

   \[ L_H e_H = P^T r_h. \]

   Note that $L_H$ is SPD.

2. We should choose $P$ such that the unknown error, $e_h$, is “approximately in its range”.

   This depends on the relaxation, hence on $L_h$. 

---

202
Choosing the Operators

Related well-known conclusions:

1. If $e_h$ is in the range of $P$ prior to the coarse-grid correction, then it is eliminated exactly.

2. Amongst all possible coarse-grid corrections with the given $P$, the Galerkin correction minimizes the resulting fine-grid error in the energy norm:

$$\left\| e_h \right\|_{L_h}^2 = \left\langle e_h^T, L_h e_h \right\rangle = \left\langle e_h^T, r_h \right\rangle.$$

3. $P$ should be such that error not effectively reduced by the relaxation is in its range.
Back to Multigrid as a Solver of Equations

There are many problems for which multigrid is suitable in principle but cannot be applied in a straightforward way. For example,

1. Unstructured grids and complex geometries
2. Non-PDE applications

Such situations require algebraic multigrid methods.

The multigrid components can be expressed as matrices. Consider, for example, the 1D model problem using linear interpolation and full-weighted residual transfers.
\[ L^h = \frac{1}{h^2} \begin{bmatrix}
-2 & 1 & & & \\
1 & -2 & 1 & & \\
1 & -2 & 1 & & \\
& & & \ddots & \\
& & & & 1 & -2 & 1 \\
& & & & 1 & -2 & 1 \\
& & & & 1 & -2 & \\
\end{bmatrix} \]

\[ I^h_H = 2(I^H_h)^T = \frac{1}{2} \begin{bmatrix}
1 & 1 & & & \\
2 & & & & \\
1 & 1 & & & \\
& & & \ddots & \\
& & & & 1 & 1 \\
& & & & 2 \\
& & & & 1 & 1 \\
\end{bmatrix} \]
Given the fine-grid matrix, $L^h$, and the prolongation and restriction matrices, $I_H^h$ and $I_h^H$, how should we define the coarse-grid matrix, $L^H$?

The coarse grid should be able to correct smooth errors. We use the following (algebraic) definition of smoothness: An error $v_{\text{before}}^h$ is smooth if it is in the range of interpolation. That is, if there exists some coarse-grid function, $w^H$, such that

$$v_{\text{before}}^h = I_H^h w^H.$$
The error after the coarse-grid correction is given by

\[ \nu^h_{\text{after}} = C^h \nu^h_{\text{before}} \]

where

\[ C^h = I^h - I^h_H \left( L^H \right)^{-1} I^H_h L^h. \]

Plugging in our smooth error we obtain:
\[ \nu^h_{after} = C^h \nu^h_{before} \]

\[ = \left[ I^h - I^h_H \left( L^H \right)^{-1} I^H_I L^h \right] I^h_H w^H \]

\[ = \left[ I^h_H - I^h_H \left( L^H \right)^{-1} I^H_H L^h I^h_H \right] w^H \]

\[ = I^h_H \left[ I^H - \left( L^H \right)^{-1} I^H_H L^h I^h_H \right] w^H. \]

In order to annihilate the error we must choose the Petrov-Galerkin coarse-grid operator:

\[ L^H = I^H_I L^h I^h_H. \]
For symmetric problems especially, the preferred choice for the restriction is the transpose of the prolongation. Along with the Galerkin coarse-grid operator this yields so-called variational coarsening, which arises naturally in finite-element formulations and convex minimization.

It remains only to define the prolongation (and, implicitly, the set of variables which defines the coarse grid). The prolongation operator should produce good approximate fine-grid values from given coarse-grid values. Therefore, $I_H^h$ needs to be determined using $L^h$. When used with appropriate coarse grids, such methods yield fast and robust algebraic solvers.
For tridiagonal matrices in $1D$ the different algebraic methods become the same: an exact multigrid solver that is equivalent to cyclic reduction.

If the fine-grid equations are

$$a_i u_{i-1} + b_i u_i + c_i u_{i+1} = f_i,$$

$I = 1, \ldots, n-1$, with $a_1 = c_{n-1} \equiv 0$, we choose the prolongation matrix to be
\[ I^h_H = \begin{bmatrix}
  -\frac{c_1}{b_1} & 1 & \cdots & \\
  -\frac{a_3}{b_3} & -\frac{c_3}{b_3} & 1 & \\
  & -\frac{a_5}{b_5} & \cdots & \\
  & & \frac{c_{n-5}}{b_{n-5}} & 1 & \cdots & \\
  & & & \frac{a_{n-3}}{b_{n-3}} & -\frac{c_{n-3}}{b_{n-3}} & 1 & \\
  & & & & \cdots & -\frac{a_{n-1}}{b_{n-1}} & \\
\end{bmatrix} \]
Furthermore, we let \( I_h^H = (I_h^H)^T \) and employ Galerkin coarsening. For smoothing we use half-Red-Black relaxation. That is, before restricting residuals we relax only on odd-indexed gridpoints, and after the coarse-grid correction only on even-indexed points.

Theorem: the two-level cycle is an exact solver. Furthermore, the coarse-grid equations are tridiagonal. Hence, the multigrid cycle is an exact solver.

We next prove this in a more general setting.
Algebraic Multigrid (AMG)

Introduced by Brandt et al. (1983) and developed by Ruge and Stueben.

AMG takes the algebrization of multigrid to the limit. Here, a relaxation method is chosen (usually, point Gauss-Seidel), and then the coarse-grid variables are chosen by a heuristic graph algorithm such that each fine-grid variable depends strongly on one or more coarse-grid variable (i.e., with relatively large coefficient).

AMG enables us to handle unstructured and non-PDE problems.
An Abstract View of AMG

Consider the linear system

\[ Au = f. \]

Suppose we partition the variables, \( u_i \), into \( F \) variables and \( C \) variables, and permute the equations and variables to produce the following partitioning of the system:

\[
Au = \begin{pmatrix}
A_{FF} & A_{FC} \\
A_{CF} & A_{CC}
\end{pmatrix}
\begin{pmatrix}
u_F \\
u_C
\end{pmatrix} = \begin{pmatrix}
f_F \\
f_C
\end{pmatrix}.
\]
An Abstract View of AMG

Given an approximate solution, \( \tilde{u} \), define the error as

\[ v = u - \tilde{u}. \]

Then, the partitioned equation for the error is

\[
Av = \begin{pmatrix} A_{FF} & A_{FC} \\ A_{CF} & A_{CC} \end{pmatrix} \begin{pmatrix} v_F \\ v_C \end{pmatrix} = \begin{pmatrix} r_F \\ r_C \end{pmatrix},
\]

where

\[
\begin{align*}
r_F &= f_F - A_{FF} \tilde{u}_F - A_{FC} \tilde{u}_C, \\
r_C &= f_C - A_{CF} \tilde{u}_F - A_{CC} \tilde{u}_C.
\end{align*}
\]
An Abstract View of AMG

The upper block yields

\[ A_{FF} v_F = r_F - A_{FC} v_C, \]

\[ \Rightarrow v_F = A_{FF}^{-1} (r_F - A_{FC} v_C). \]

Plugging this into the lower block yields

\[ A_{CF} A_{FF}^{-1} (r_F - A_{FC} v_C) + A_{CC} v_C = r_C, \]

\[ \Rightarrow \left( A_{CC} - A_{CF} A_{FF}^{-1} A_{FC} \right) v_C = r_C - A_{CF} A_{FF}^{-1} r_F. \]
An Abstract View of AMG

Conclusion: the “ideal” prolongation and restriction are

\[ P = \begin{pmatrix} -A_{FF}^{-1}A_{FC} \\ I \end{pmatrix}, \quad R = \begin{pmatrix} -A_{CF}A_{FF}^{-1} \\ I \end{pmatrix}, \]

with the coarse-grid operator given by

\[ A_C = RAP = A_{CC} - A_{CF}A_{FF}^{-1}A_{FC}. \]
An Abstract View of AMG

In particular, it is straightforward to verify that the two-level solution is exact in this case, provided that either a pre-relaxation or a post-relaxation eliminates $r_F$.

(If this is done by post-relaxation, only $u_F$ should be relaxed.)

The problem: $A_{FF}^{-1}$ is not sparse, and therefore, neither are $P$ and $R$. Therefore, we generally look for good sparse approximations.

One exception is tri-diagonal systems, where $A_{FF}$ is diagonal. In this case the multigrid $V$-cycle with the appropriate prolongation and restriction, and with relaxation only on $u_F$ is an exact solver, as stated above.
Multigrid for Nonlinear Problems

(Back to the optimization framework)

Thus far we have considered linear problems, so the functional was quadratic. A general (assumed convex) minimization problem can be written as

\[ u_h = \arg \min_{v_h \in \mathbb{R}^n} \left( F_h(v_h) - v_h^T f_h \right) \]

\[ \iff \]

\[ G_h(u_h) = f_h, \]

With \( G_h \triangleq \nabla F_h \).

The gradient of the functional must vanish.
Nonlinear Example

The one-dimensional $p$-Laplacian

$$F(v) = \int |\nabla v|^p \, dx,$$

$$G(u) = p \nabla \cdot \left( |\nabla u|^{p-2} \nabla u \right).$$
Nonlinear Example

Discretization:

\[ F_h (v_h) = h^{1-p} \sum_i \left| v_{h,i} - v_{h,i-1} \right|^p, \]

\[ G_h (u_h)_i = h^{1-p} p \left| u_{h,i} - u_{h,i+1} \right|^{p-2} (u_{h,i} - u_{h,i+1}) + h^{1-p} p \left| u_{h,i} - u_{h,i-1} \right|^{p-2} (u_{h,i} - u_{h,i-1}) \]
Nonlinear Multigrid

Following the same approach as in the linear case, the optimal coarse-grid correction is the solution to the minimization problem:

$$e_H = \arg \min_{c_H \in \mathbb{R}^N}$$

$$
\left( F_h (\tilde{u}_h + Pc_H) - (\tilde{u}_h + Pc_H)^T f_h \right)
\Longleftrightarrow
\nabla_{e_H} F_h (\tilde{u}_h + Pe_H) = P^T f_h. $$
Nonlinear Multigrid

In principle, we can solve the coarse-grid problem for the optimal coarse-grid correction.

However, the coarse functional is typically more complicated than $F_h$.

This requires special treatment, especially because further (recursive) coarsening complicates the operator still further...
Nonlinear Problems

First step: change coarse variables

\[ \nu_H = R\tilde{u}_h + c_H, \]

where \( R \in \mathbb{R}^{N \times n} \) is yet another restriction operator, e.g., \( R = P^T \) or injection.

The optimal coarse functional is now:

\[
F_H (\nu_H) \equiv F_h (\tilde{u}_h + P c_H)
= F_h (\tilde{u}_h + P (\nu_H - R\tilde{u}_h))
= F_h ((I - PR)\tilde{u}_h + P\nu_H).
\]
Nonlinear Multigrid

Vanishing gradient now reads:

$$\nabla F_H(u_H) = \nabla_{u_H} F_h\left((I-PR)\tilde{u}_h + Pu_H\right) = P^T f_h.$$ 

Remark: *I-PR* is a “high-pass filter”. The argument of the functional is thus split into a given oscillatory part and a smooth part that is to be prolonged from the coarse grid.

(In so-called adaptive multigrid methods, *P* is chosen such that $(I-PR)\tilde{u}_h = 0$.)
Nonlinear Multigrid

We have seen that our “optimal” coarse grid problem can be written as:

\[ \nabla F_H(u_H) = \nabla_{u_H} F_h \left( (I - PR)\tilde{u}_h + Pu_H \right) = P^T f_h. \]

Now, if this \( F_H \) is “similar to” \( F_h \), then we can recurse as in the linear case. If not, then we typically must resort to employing an approximation of \( F_H \) in the following way.
Back to the Example

\[ F_h (v_h) = h^{1-p} \sum_{i=1,n} |v_{h,i} - v_{h,i-1}|^p, \]

Using linear interpolation for \( P \) we get

\[ F_H (v_H) = 2^{-p} h^{1-p} \sum_{I=1,N} \left( \begin{array}{c} |v_{H,I} - v_{H,I-1} + c_{2I-1}|^p \\ + |v_{H,I} - v_{H,I-1} - c_{2I-1}|^p \end{array} \right), \]

where \( c_{2I-1} \) is a perturbation that depends on \( \tilde{u}_{h,i} \) and its neighbors. It is small so long as \( \tilde{u}_h \) is smooth.
Approximating $F_H$

Let $\hat{F}_H$ denote a simple convex approximation to $F_H$, e.g., a discretization of $F$ on grid $H$.

In our example this means neglecting $c_{2I-1}$.

Rewriting the optimal coarse functional as:

$$F_H (v_H) = \hat{F}_H (v_H) + \left[ F_H (v_H) - \hat{F}_H (v_H) \right]$$

$$= \hat{F}_H (v_H) + \left[ F_h ( ((I - PR) \tilde{u}_h + P v_H ) - \hat{F}_H (v_H) \right] ,$$

we proceed by expanding the bracketed term in a first-order truncated Taylor series around $R \tilde{u}_h$. 
First-Order Approximation

Expanding to first order in $c_H$ yields the convex approximate functional

\[
F_H (\nu_H) \approx F_H^1 (\nu_H) = \hat{F}_H (\nu_H) + \left[ F_h (\tilde{u}_h) - \hat{F}_H (R\tilde{u}_h) \right] + c_H^T \left[ P^T G_h (\tilde{u}_h) - \hat{G}_H (R\tilde{u}_h) \right],
\]

where $G$ again denotes gradients.

The gradient of $F_H^1$ reads:

\[
G_H^1 (u_H) = \hat{G}_H (u_H) + \left[ P^T G_h (\tilde{u}_h) - \hat{G}_H (R\tilde{u}_h) \right].
\]
First-Order Approximation

The resulting coarse problem reads

\[ G_H^1 (u_H) = P^T f_h, \]

i.e.,

\[ \hat{G}_H (u_H) = P^T \left[ f_h - G_h (\tilde{u}_h) \right] + \hat{G}_H (R\tilde{u}_h). \]

As we shall show below, this coincides with the classical FAS algorithm of Brandt (1975).
Nonlinear Problems: Back to Multigrid as a Solver for Equations

There are two common approaches for handling nonlinear problems by multigrid methods. One is the classical approach of employing Newton linearization (and solving the resulting linear problems by the usual multigrid methods).

A second approach is to employ nonlinear multigrid methods, most commonly the “Full Approximation Scheme” (FAS).
Recall that, in the usual multigrid approach, we use the coarse grid to approximate the correction to the fine-grid error. That is, we approximate the fine-grid equation

\[ L^h v^h = r^h, \]

by the coarse-grid equation

\[ L^H v^H = I_h^H r^h. \]

We can rewrite the fine-grid equation as

\[ L^h u^h - L^h \tilde{u}^h = r^h. \]
We approximate this equation on the coarse grid by

\[ L^H u^H - L^H \tilde{u}^H = r^H, \]

with

\[ \tilde{u}^H = I_h^H \tilde{u}^h. \]

The difference is that now the variable, \( \tilde{u}^H \), approximates the full solution rather than just the correction. Hence, this approach can be applied to nonlinear problems. After we solve the coarse-grid problem, we interpolate and add the correction:

\[ \tilde{u}^h \leftarrow \tilde{u}^h + I_H^h \left( u^H - \tilde{u}^H \right). \]
Two-grid FAS Algorithm

• Relax several times on grid $h$, obtaining $\tilde{u}^h$ with a smooth corresponding error.

• Calculate the residual: $r^h = f^h - L^h \tilde{u}^h$.

• Solve approximate equation for the full solution on the coarse grid:
  \[ L^H u^H = f^H \equiv I_h^H r^h + L^H \hat{I}_h^H \tilde{u}^h. \]

• Interpolate and add correction:
  \[ \tilde{\tilde{u}}^h \leftarrow \tilde{u}^h + I_h^H \left( u^H - \hat{I}_h^H \tilde{u}^h \right). \]

• Relax again on grid $h$.

Multi-grid is obtained by recursion.
Exercise 10

Building on your linear multigrid code, construct a nonlinear (FAS) solver, and use it to solve the problem:

\[ Lu = u_{xx} + u_{yy} - u^3 = f. \]

Use \( V(2,1) \) cycles with Red-Black relaxation. (For the local nonlinear problems, use one Newton-Raphson iteration).

Show residual convergence results for \( f = 1 \), and \( u = 0 \) on the boundary of the square domain.
FAS: Dual point of view

We can rewrite the FAS coarse-grid problem in the form

\[ L^H u^H = I^h_h f^h + \tau^H_h, \]

where

\[ \tau^H_h = L^H \hat{I}^H_h \tilde{u}^h - I^H_h L^h \tilde{u}^h. \]

Observe that \( \tau^H_h \) is the fine-to-coarse defect correction - an increment to the right-hand side designed to make its solution coincide with the fine-grid solution.

This reverses our point of view: the fine-grid "visits" can be seen as a means of obtaining a better estimate of the \( \tau \)-correction.
$\tau^h$ extrapolation

The truncation error on grid $h$ is

$$\tau^h = L^h I^h u - I^h Lu.$$  

That is, $\tau^h$ is what we need to add to the right hand side of the grid-$h$ equation to make the solution coincide with the differential solution. Similarly, $\tau_{2h}^h$ is the correction that would make the coarse-grid solution coincide with the fine-grid solution. Hence, upon convergence we have

$$\tau_{2h}^h \approx I_{2h}^h \tau^h + \tau_{h}^{2h}.$$  

We can use this relation to raise the order of approximation inexpensively.
If the local approximation order at point $x$ is $p$, i.e.,

$$\tau^h(x) \approx c(x)h^p,$$

where $c(x)$ is independent of $h$, then

$$\tau^{2h}(x) \approx 2^p c(x)h^p.$$

Hence,

$$\tau_h^{2h} \approx (2^p - 1)c(x)h^p$$

and therefore

$$\tau^{2h}(x) \approx \frac{2^p}{2^p - 1} \tau_h^{2h}(x)$$
To raise the order of approximation all we have to do is multiply the $\tau_h^{2h}$ term of the coarse-grid equations by the fixed factor $2^p/(2^p - 1)$:

$$L^{2h} u^{2h} = I_h^{2h} f^h + \frac{2^p}{2^p - 1} \tau_h^{2h}$$

This operation is called $\tau$-extrapolation.
Double-Discretization

An alternative method for obtaining higher-order accuracy is the method of double-discretization.

The residuals are computed using a discretization that is different (normally higher-order) from that used in the relaxation process.

The discretization of the residuals generally determines the accuracy of the method, while that used for relaxation determines the stability.

This method is related to the $\tau$-extrapolation technique.
Fourier Analysis of Red-Black Relaxation

Smoothing analysis of multi-colored relaxation is more complicated, because Fourier modes are no longer eigenfunctions of the relaxation operator. However, under suitable conditions, small subspaces comprised of just a few Fourier modes are invariant under the relaxation.

Red-Black relaxation is defined as a Jacobi sweep over “Red” points – grid-points whose index vector sum is even – followed by a Jacobi sweep over “Black” points – the complement. For this relaxation, subspaces comprised of pairs of Fourier modes of frequencies $\Theta$ and $\tilde{\Theta} = \Theta + \{\pm \pi\}^d$ are invariant. Here, $d$ is the dimension.
To see this, observe that

\[ \varphi_{\tilde{\theta}} = e^{i \left( \sum_{k=1}^{d} (\theta_k \pm \pi) j_k \right)} = (-1)^{\sum_{k=1}^{d} j_k} \varphi_{\theta}. \]

That is, \( \varphi_{\tilde{\theta}} \) is equal to \( \varphi_{\theta} \) on Red grid points, but equal to \( -\varphi_{\theta} \) on Black ones.

Thus, if we begin with some linear combination of such a pair of Fourier modes, and perform, say, a Jacobi sweep on Red points only, then the sum of the mode amplitudes will change accordingly, but the difference will remain unchanged (and vice versa if we sweep over Black points).
Fourier Analysis of Red-Black Relaxation

Armed with this understanding, we can compute the symbol of Red-Black relaxation as follows:

\[ \hat{R}_{RB} = A^{-1}OEA, \]

where

\[ A = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \]

is the transformation to the Red-Black basis. Now, for \(2d+1\) point operators,
Fourier Analysis of Red-Black Relaxation

\[ E = \begin{pmatrix} 0 & s(\theta) \\ 0 & 1 \end{pmatrix} \]

is the symbol of Jacobi relaxation (whose symbol is denoted by \( s \)) on Red points, and

\[ O = \begin{pmatrix} 1 & 0 \\ s(\theta) & 0 \end{pmatrix} \]

is the symbol of Jacobi relaxation on Black points. This yields

\[ \hat{R}_{RB} = \frac{1}{2} \begin{pmatrix} s(s+1) & -s(s+1) \\ -s(s-1) & s(s-1) \end{pmatrix} \]
Fourier Analysis of Red-Black Relaxation

Observe that the relaxation symbol is now a small matrix rather than a scalar function of $\theta$. We therefore need to generalize our notion of the smoothing factor.

The frequency cutoff used to define the smoothing factor is an idealized coarse-grid operator that annihilates the smooth modes. Presently, we must define a similar operator by a 2 by 2 matrix:

$$Q = \begin{pmatrix} q(\theta) & 0 \\ 0 & q(\tilde{\theta}) \end{pmatrix},$$

where $q = 1$ for a high-frequency argument and $q = 0$ for a low-frequency argument.
Fourier Analysis of Red-Black Relaxation

The smoothing factor for \( \nu \) relaxation sweeps is now defined by

\[
\mu = \max_{\theta \in [-\pi, \pi]^d} \rho \left[ Q(\theta, \tilde{\theta}) \hat{R}_{RB}(\theta)^\nu \right]^{1/\nu},
\]

where \( \rho \) denotes the spectral radius.

Consider now the usual \( 2d+1 \) star discretization of the elliptic operator,

\[
L = \sum_{i=1}^{d} c_i \partial_{x_i x_i}, \quad \sum_{i=1}^{d} c_i = 1,
\]

where the \( c_i \)'s are constant positive coefficients. Denote

\[
c_{\min} = \min_{i=1, \ldots, d} c_i.
\]
Fourier Analysis of Red-Black Relaxation

For a single sweep, $\nu = 1$, the smoothing factor is given by:

$$\mu = \max \left[ (1 - c_{\text{min}})^2, 0.125 \right].$$

**Proof:** Observe first that either $\theta$ or $\tilde{\theta}$ (or both) are necessarily in the high-frequency zone. We will assume by convention that $\tilde{\theta}$ is high-frequency, hence $q(\tilde{\theta}) = 1$. Now, since $QS$ is singular, one of its eigenvalues is zero. $\mu$ is therefore the maximal absolute value of the second eigenvalue:

$$\mu = \max_{\theta} \frac{1}{2} \left| [q(\theta) + 1] s^2 + [q(\theta) - 1] s \right|.$$
Fourier Analysis of Red-Black Relaxation

Thus,

\[ \mu = \max_{\theta} \begin{cases} 
  s^2 & \text{if } \theta \text{ is high-frequency} \\
  \frac{1}{2} |s - s^2| & \text{if } \theta \text{ is low-frequency}
\end{cases} \]

If \( \theta \) is high-frequency, then one of its components is high-frequency. But then at least one other must be low-frequency, otherwise \( \tilde{\theta} \) would be low-frequency, contradicting our convention. Now, since

\[ s(\theta) = \sum_{i=1}^{d} c_i \cos(\theta_i), \]

we get

\[ \max_{\pi/2 \leq |\theta| \leq \pi} s^2 = \left(1 - c_{\min}\right)^2. \]
Fourier Analysis of Red-Black Relaxation

Next, for low-frequency $\theta$, $s$ can take on any value between 0 and 1. The maximum of $0.5|s - s^2|$ over this range is 0.125, obtained for $s = \frac{1}{2}$. This completes the proof.

For general $\nu$ we obtain:

$$
\mu = \max \left[ (1 - c_{\min})^2, F(\nu) \right],
$$

$$
F(\nu) = \left( \frac{2\nu - 1}{2\nu} \right)^2 \left[ \frac{1}{2(2\nu - 1)} \right]^{1/\nu}.
$$
Fourier Analysis of Red-Black Relaxation

This result and analysis, originally given by K. Stueben and U. Trottenberg in


Fourier Analysis of Red-Black Relaxation

Can relaxation parameters help here too?

Notwithstanding the contrariwise claims of several textbooks, much can be gained from introducing over-relaxation parameters to Red-Black relaxation.

To generalize our analysis to this case, we need to change the two by two matrices associated with the Red and Black relaxation sweeps. For relaxation parameter $\omega$, we get:
Fourier Analysis of Red-Black Relaxation

\[ E = \begin{pmatrix} 1 - \omega & \omega s(\theta) \\ 0 & 1 \end{pmatrix}, \]

\[ O = \begin{pmatrix} 1 & 0 \\ \omega s(\theta) & 1 - \omega \end{pmatrix}, \]

\[ \hat{R}_{\text{SOR}} = A^{-1}OEA = \]

\[ \frac{1}{2} \begin{pmatrix} 2 + \omega(s-1)(\omega s + 2) & -\omega^2 s(s + 1) \\ -\omega^2 s(s - 1) & 2 + \omega(s + 1)(\omega s - 2) \end{pmatrix}. \]
Fourier Analysis of Red-Black Relaxation

The analysis continues along the same lines, though it is more technically complicated. The main result is:

\[
\omega_{\text{opt}} \approx \frac{2}{1 + \sqrt{1 - (1 - c_{\text{min}})^2}},
\]

\[
\mu_{\text{opt}} \approx \frac{1 + (1 - c_{\text{min}})^2}{2 \left( 1 + \sqrt{1 - (1 - c_{\text{min}})^2} \right)^2}.
\]
Fourier Analysis of Red-Black Relaxation

The relaxation parameter becomes relatively more advantageous as \( c_{\text{min}} \) is decreased, due to anisotropy or high dimension or both.

The table compares two-level convergence factors with one relaxation sweep, with and without the over-relaxation indicated above.

<table>
<thead>
<tr>
<th>( c_{\text{min}} )</th>
<th>( \omega = \omega_{\text{opt}} )</th>
<th>( \omega = 1 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/3</td>
<td>0.236</td>
<td>0.444</td>
</tr>
<tr>
<td>1/5</td>
<td>0.313</td>
<td>0.640</td>
</tr>
<tr>
<td>1/10</td>
<td>0.426</td>
<td>0.810</td>
</tr>
</tbody>
</table>
Fourier Analysis of Red-Black Relaxation

As we can see, using relaxation parameters increases the range for which the cycle is efficient quite significantly.

Inter-grid Transfers and Non-ellipticity

The problem of choosing prolongation and restriction operators spans several aspects. Here we study the matter of the orders of inter-grid transfers.

The question of how to treat efficiently non-elliptic and singular-perturbation operators in multigrid algorithms is also quite involved. Here we consider it from the point of view of obtaining accurate coarse-grid approximations.

The two subjects are related, hence discussed together. We follow:

Inter-grid Transfers and Non-ellipticity

Definitions and Notation

$L$ – A homogeneous linear constant-coefficient partial differential operator of order $M$, defined on a $(-\pi,\pi)^d$ periodic domain. For simplicity, we set $d = 2$.

$L^h$ – A $p_h$-order accurate finite-difference discretization on a uniform square grid $h$.

$L^H$ – Coarse-grid approximation on grid $H$.

A function $u^h$ defined on grid $h$ can be expanded in a Fourier series:

$$u^h = \sum_{\frac{-\pi}{h} < \omega_1, \omega_2 \leq \frac{\pi}{h}} A_\omega e^{i(\omega_1 x + \omega_2 y)}.$$
Inter-grid Transfers and Non-ellipticity

Definitions and Notation

Denote $\omega = (\omega_1, \omega_2)$, and define

$$|\omega| = \max(|\omega_1|, |\omega_2|).$$

For a constant coefficient operator, $Q$, denote its symbol by:

$$Q(e^{i(\omega_1 x + \omega_2 y)}) = \hat{Q}(\omega) e^{i(\omega_1 x + \omega_2 y)}.$$

We assume throughout that

$$\hat{L}^h(\omega), \hat{L}^H(\omega) = 0 \iff |\omega| = 0,$$

and the analysis excludes the case $|\omega| = 0$. 


Inter-grid Transfers and Non-ellipticity

Definitions and Notation

The absolute and relative orders of approximation of $L^h$ by $L^H$, respectively, are denoted by $p_{h,H}$ and $q_{h,H}$, defined as the largest integers satisfying:

$$
|\hat{L}^H (\omega) - \hat{L}^h (\omega)| = O\left(h^{p_{h,H}}\right),
$$

$$
\frac{|\hat{L}^H (\omega) - \hat{L}^h (\omega)|}{|\hat{L}^H (\omega)|} = O\left(h^{q_{h,H}}\right),
$$

$h \to 0$. 

Inter-grid Transfers and Non-ellipticity

Example

Let $L^h$ and $L^H$ be the usual five-point star discretization of the Laplacian. Then,

$$\hat{L}^h = -\frac{4}{h^2} \left[ \sin^2 \left( \frac{h\omega_1}{2} \right) + \sin^2 \left( \frac{h\omega_2}{2} \right) \right],$$

and, for $H = 2h$,

$$\hat{L}^H = -\frac{1}{h^2} \left[ \sin^2 (h\omega_1) + \sin^2 (h\omega_2) \right].$$

A Taylor series expansion now yields

$$\left| \hat{L}^H - \hat{L}^h \right| = 0.25h^2 \left[ \omega_1^4 + \omega_2^4 + O(h^2) \right],$$

$h \to 0.
Inter-grid Transfers and Non-ellipticity

Example

Since, clearly,

$$\left| \hat{L}^H \right|^{-1} = O(1),$$

we obtain

$$p_{h,H} = q_{h,H} = 2.$$  

This is the behavior we generally expect for elliptic operators.
Inter-grid Transfers and Non-ellipticity

Suppose, however, that $L$ is non-elliptic. Then, by definition, there is at least one characteristic direction (i.e., some ratio of $\omega_1$ and $\omega_2$ for which $\hat{L} = 0$).

Consider now a Taylor expansion of $\hat{L}^H$ around $H = 0$. By consistency, the first term is equal to $\hat{L}$. Thus, for so-called smooth characteristic Fourier components, satisfying $\hat{L}(\omega) = 0$, and $|\omega| = O(1)$, what remains is the truncation error, yielding,

$$\left| \hat{L}^H(\omega) \right| = O\left( h^{p_H} \right),$$

where $p_H$ denotes the order of accuracy of the coarse-grid discretization with respect to the differential operator.
Inter-grid Transfers and Non-ellipticity

Thus, we obtain that for smooth characteristic components the relative order of approximation is reduced as follows:

\[ q_{h,H} = p_{h,H} - p_H. \]

In particular, if (as usual) we use the same discretization on the fine and coarse grids, then, for such components,

\[ p_{h,H} = p_H, \quad \Rightarrow \quad q_{h,H} = p_{h,H} - p_H = 0. \]
Inter-grid Transfers and Non-ellipticity

Example

Let $L^h$ and $L^H$ be the standard first-order accurate upstream discretizations of the convection operator,

$$L = a \partial_x + b \partial_y,$$

where $a$ and $b$ are positive constants. Assume standard coarsening, $H = 2h$.

Now,

$$\hat{L}^h = \frac{1}{h} \left[ a \left(1 - e^{-i2\omega_1} \right) + b \left(1 - e^{-i2\omega_2} \right) \right],$$

$$\hat{L}^H = \frac{1}{2h} \left[ a \left(1 - e^{-i2\omega_1} \right) + b \left(1 - e^{-i2\omega_2} \right) \right],$$
Inter-grid Transfers and Non-ellipticity

Example

and a Taylor expansion yields

\[ |\hat{L}^H - \hat{L}^h| = 0.5h \left[ a\omega_1^2 + b\omega_2^2 + O(h) \right] , \]
\[ \hat{L}^H = ia\omega_1 + ib\omega_2 + a\omega_1^2 + bh\omega_2^2 + O(h^2) , \]
\[ h \to 0 . \]

Therefore, for those \( O(1) \) \( \omega \)'s that satisfy

\[ a\omega_1 = -b\omega_2 , \]

we get

\[ |\hat{L}^H| = O(h) . \]
Inter-grid Transfers and Non-ellipticity

Example

For such smooth characteristic components,

\[
\left| \frac{\hat{L}^H - \hat{L}^h}{\hat{L}^H} \right| \approx 0.5,
\]

so, indeed, \( q_{h,H} = 0 \).

An exception occurs in the case of alignment of the characteristic direction with the grid, in which case either \( a \) or \( b \) vanish. In this case, the truncation error for smooth characteristic components vanishes.
Exercise 8

Consider the $\pi/4$-rotated diffusion operator in the anisotropic limit:

$$Lu = u_{xx} + 2u_{xy} + u_{yy},$$

discretized by

$$L^h = \frac{1}{h^2} \begin{bmatrix} -0.5 & 1 & 0.5 \\ 1 & -4 & 1 \\ 0.5 & 1 & -0.5 \end{bmatrix},$$

whose symbol is given by

$$\hat{L}^h = \frac{2}{h^2} \left[ -2 + \cos(h\omega_1) + \cos(h\omega_2) - \sin(h\omega_1) \sin(h\omega_2) \right].$$

Show that for $\omega_1 = -\omega_2$ we get

$$\frac{|\hat{L}^{2h} - \hat{L}^h|}{|\hat{L}^{2h}|} \approx 0.75, \quad h \to 0.$$
Inter-grid Transfers and Non-ellipticity

What are the minimal requirements from the relative order of approximation and from the inter-grid operators?

The two-grid correction matrix can be written as

\[ T^h = C^h \left(S^h\right)^\nu , \]
\[ C^h = I^h - P^h \left(L^H\right)^{-1} R^h L^h , \]

where \( C^h \) is the coarse-grid correction, \( S^h \) is the relaxation matrix, and \( P^h \) and \( R^h \) are the prolongation and restriction matrices, respectively. We concentrate here on \( C^h \).

The coarse-grid correction operator couples subspaces of Fourier components of size four:
Inter-grid Transfers and Non-ellipticity

\[ \{(\omega_1, \omega_2), (\tilde{\omega}_1, \omega_2), (\omega_1, \tilde{\omega}_2), (\tilde{\omega}_1, \tilde{\omega}_2)\}, \]

with

\[ \tilde{\omega}_{1,2} = \omega_{1,2} \pm \frac{\pi}{h}, \]

where \( \omega = (\omega_1, \omega_2) \) is smooth, in particular, representable on the coarse grid, i.e.,

\[ |\omega| \leq \frac{\pi}{H}. \]

Thus, the symbol of \( C^h \) is given by a four by four matrix of the form
Inter-grid Transfers and Non-ellipticity

\[ \hat{C}^h = \hat{I}^h - \hat{P}^h \left( \hat{L}^H \right)^{-1} \hat{R}^h \hat{L}^h, \]

where

\[ \hat{L}^h = \text{Diag}\left[ \hat{L}^h (\omega_1, \omega_2), \hat{L}^h (\tilde{\omega}_1, \omega_2), \hat{L}^h (\omega_1, \tilde{\omega}_2), \hat{L}^h (\tilde{\omega}_1, \tilde{\omega}_2) \right], \]

\[ \hat{P}^h = \left[ \hat{P}^h (\omega_1, \omega_2), \hat{P}^h (\tilde{\omega}_1, \omega_2), \hat{P}^h (\omega_1, \tilde{\omega}_2), \hat{P}^h (\tilde{\omega}_1, \tilde{\omega}_2) \right]^T, \]

\[ \hat{R}^h = \left[ \hat{R}^h (\omega_1, \omega_2), \hat{R}^h (\tilde{\omega}_1, \omega_2), \hat{R}^h (\omega_1, \tilde{\omega}_2), \hat{R}^h (\tilde{\omega}_1, \tilde{\omega}_2) \right], \]

\[ \hat{L}^H = \hat{L}^H (\omega_1, \omega_2). \]
Inter-grid Transfers and Non-ellipticity

Orders of inter-grid transfers

Definition: $R^h$ has a low-frequency (LF) order $m_R$, and a high-frequency (HF) order, $n_R$, if $m_R$ and $n_R$ are the largest numbers satisfying

\[
\hat{R}^h = \left[ 1 + O\left(h^{m_R}\right), O\left(h^{n_{R_1}}\right), O\left(h^{n_{R_2}}\right), O\left(h^{n_{R_3}}\right) \right],
\]

\[h \to 0,\]

with

\[n_R = \min_{i=1,2,3} n_{R_i}.
\]
Inter-grid Transfers and Non-ellipticity

Orders of inter-grid transfers

A large $m_R$ means that LF fine-grid components are transferred to the coarse grid with very small error. A large $n_R$ means that HF fine-grid components (which would alias with LF components on the coarse grid) are filtered out well by the restriction operator.

The orders of the prolongation, $m_P$ (LF order) and $n_P$ (HF order) are defined analogously. A large $m_P$ means that the amplitude of a Fourier component that has been interpolated to the fine grid is very close to what it was on the coarse grid. A large $n_P$ means that the amplitudes of HF modes generated by this interpolation are small.
Inter-grid Transfers and Non-ellipticity

Examples (in 1D)

\[ R^h = \frac{1}{4} \begin{bmatrix} 1 & 2 & 1 \end{bmatrix}, \]

\[ \hat{R}^h(\omega) = \frac{1}{2} \left[ 1 + \cos(\omega h) \right] = 1 + O(h^2), \]

\[ \hat{R}^h(\tilde{\omega}) = \frac{1}{2} \left[ 1 - \cos(\omega h) \right] = O(h^2), \]

\[ R^h = \frac{1}{16} \begin{bmatrix} -1 & 4 & 10 & 4 & -1 \end{bmatrix}, \]

\[ \hat{R}^h(\omega) = \frac{1}{8} \left[ 5 + 4\cos(\omega h) - \cos(2\omega h) \right] = 1 + O(h^4), \]

\[ \hat{R}^h(\tilde{\omega}) = \frac{1}{8} \left[ 5 - 4\cos(\omega h) - \cos(2\omega h) \right] = O(h^2), \]

\[ R^h = \frac{1}{16} \begin{bmatrix} 1 & 4 & 6 & 4 & 1 \end{bmatrix}, \]

\[ \hat{R}^h(\omega) = \frac{1}{8} \left[ 3 + 4\cos(\omega h) + \cos(2\omega h) \right] = 1 + O(h^2), \]

\[ \hat{R}^h(\tilde{\omega}) = \frac{1}{8} \left[ 3 - 4\cos(\omega h) + \cos(2\omega h) \right] = O(h^4). \]
Inter-grid Transfers and Non-ellipticity

Denote by $m_H$ the smallest number satisfying

$$
\left| \hat{L}^H (\omega_1, \omega_2) \right|^{-1} = O(h^{-m_H}), \quad h \to 0,
$$

and observe that

$$
\hat{L}^H = O(h^{-M})
$$

For HF modes.

Substituting all these asymptotic relations into $\hat{C}^h$, and assuming henceforth that all the orders are positive, we obtain:
Inter-grid Transfers and Non-ellipticity

\[ \hat{C}^h = \{ c_{i,j}^h \}, \]

with

\[ c_{1,1}^h = 1 - \left[ 1 - O\left( h^{\min(m_R,m_P)} \right) \right] \frac{\hat{L}^h(\omega_1,\omega_2)}{\hat{L}^H(\omega_1,\omega_2)}, \]

\[ c_{i,i}^h = 1 - O\left( h^{n_R+n_P-m_H-M} \right), \quad i = 2,3,4, \]

\[ c_{1,i}^h = O\left( h^{n_R-m_H-M} \right), \quad i = 2,3,4, \]

\[ c_{i,1}^h = O\left( h^{n_P} \right) \frac{\hat{L}^h(\omega_1,\omega_2)}{\hat{L}^H(\omega_1,\omega_2)}, \quad i = 2,3,4, \]

\[ c_{i,j}^h = O\left( h^{n_R+n_P-m_H-M} \right), \quad \text{otherwise}. \]
Inter-grid Transfers and Non-ellipticity

From this we find that the condition that the eigenvalues of $\hat{C}^h$ remain bounded as $h$ tends to zero is:

$$n_R + n_P \geq m_H + M.$$


For non-elliptic operators, as we have seen, there exist smooth characteristic components for which $m_H = p_H$, so the condition becomes more restrictive.

Next, we wish to study the behavior of the two-level cycle.
Inter-grid Transfers and Non-ellipticity

Suppose, for simplicity, that the relaxation is such for which Fourier components are eigenfunctions, e.g., damped Jacobi. Thus, the 4 by 4 symbol of the relaxation is a diagonal matrix,

\[ \hat{S}^h = \text{Diag} \left[ \hat{S}^h (\omega_1, \omega_2), \hat{S}^h (\tilde{\omega}_1, \omega_2), \hat{S}^h (\omega_1, \tilde{\omega}_2), \hat{S}^h (\tilde{\omega}_1, \tilde{\omega}_2) \right]. \]

Suppose also that the restriction satisfies

\[ n_R > m_H + M, \]

so that all the off-diagonal terms of \( \hat{C}^h \) tend to zero with \( h \). Then, in the limit of \( h \) tending to zero, the symbol of the two-level iteration, \( \hat{T}^h = \hat{C}^h \left( \hat{S}^h \right)^\nu \), tends to a diagonal matrix:

\[ \hat{T}^h = \text{Diag} \left[ \left( 1 - \frac{\hat{L}^h (\omega_1, \omega_2)}{\hat{L}^H (\omega_1, \omega_2)} \right) \hat{S}^h (\omega_1, \omega_2)^\nu, \hat{S}^h (\tilde{\omega}_1, \omega_2)^\nu, \hat{S}^h (\omega_1, \tilde{\omega}_2)^\nu, \hat{S}^h (\tilde{\omega}_1, \tilde{\omega}_2)^\nu \right]. \]
Inter-grid Transfers and Non-ellipticity

Each element of this diagonal matrix corresponds to an eigenvalue of the two-level matrix. The last three are all bounded by $\mu^\nu$, where $\mu$ is the smoothing factor. For the first one we have for $|\omega| = O(1)$,

$$\lambda_1 \rightarrow \left(1 - \frac{\hat{L}^h(\omega_1, \omega_2)}{\hat{L}^H(\omega_1, \omega_2)}\right), \quad h \rightarrow 0.$$  

We find that $\lambda_1$ tends to zero when $q_{h,H}$ is positive (as in elliptic problems), indicating that such components are corrected very well by the coarse grid, but it is $O(1)$ when $q_{h,H} = 0$. This happens, as we have seen, when we use a naïve coarse-grid approximation in non-elliptic problems. For example, in the case of first-order upstream discretization of the convection operator, we get $\lambda_1 \rightarrow 0.5, \quad h \rightarrow 0$.  

278
Inter-grid Transfers and Non-ellipticity

Remedies

We have seen that using standard coarsening yields poor coarse-grid correction of smooth characteristic error components. Following are means of reducing this problem.

1. Use a global relaxation method to eliminate such errors efficiently (e.g., down-stream marching for upstream-discretized convection).

2. Use W cycles to reduce the accumulating effect of poor coarse-grid correction.

3. Combine W cycles with acceleration by overweighting of residuals (or coarse-grid corrections).
Inter-grid Transfers and Non-ellipticity

Remedies

4. Improve $q_{h,H}$ via (Petrov-) Galerkin coarse-grid discretizations with appropriate inter-grid transfers, or by explicit correction terms in the coarse-grid discretization.

To demonstrate the acceleration approach, suppose that $\lambda_1$ is real and satisfies (for $h$ tending to zero)

$$0 \leq \lambda_1 = \left(1 - \frac{\hat{L}^h (\omega_1, \omega_2)}{\hat{L}^H (\omega_1, \omega_2)}\right) \leq C < 1.$$ 

If we over-weight the residual by a factor $\beta > 1$, then we obtain instead
Inter-grid Transfers and Non-ellipticity

Residual overweighting

$$1 - \beta \leq \lambda_{1,\beta} = \left(1 - \frac{\beta \hat{L}^h (\omega_1, \omega_2)}{\hat{L}^H (\omega_1, \omega_2)}\right) \leq 1 - \beta + \beta C.$$ 

The optimal $\beta$ is obtained by equating the absolute values of the lower and upper bounds, obtaining

$$|1 - \beta| = |1 - \beta + \beta C|,$$

Hence,

$$\beta_{\text{opt}} = \frac{2}{2 - C},$$

and

$$\left|\lambda_{1,\beta_{\text{opt}}}\right| \leq \frac{C}{2 - C}.$$
Inter-grid Transfers and Non-ellipticity

Residual overweighting

For the upstream convection operator mentioned above, $\lambda_1$ is thus reduced from $1/2$ to $1/3$.

This technique, in conjunction with an improved coarse-grid operator introduced via a defect-correction process (which essentially doubles the convergence rate), is analyzed and tested in

Exercise 9

Let

\[ Lu = u_{ss} + \varepsilon u_{tt}. \]

Let \( \phi \) be the angle between \((s,t)\) and the grid-aligned coordinate system, \((x,y)\). Hence,

\[ Lu = \left(C^2 + \varepsilon S^2\right)u_{xx} + 2(1-\varepsilon)CSu_{xy} + \left(\varepsilon C^2 + S^2\right)u_{yy}, \]

with

\[ C = \cos(\phi), \quad S = \sin(\phi). \]

Discretize \( L \) using the stencil

\[ L^h = \frac{1}{h^2} \begin{bmatrix}
-\frac{1}{2}(1-\varepsilon)CS & \varepsilon C^2 + S^2 & \frac{1}{2}(1-\varepsilon)CS \\
C^2 + \varepsilon S^2 & -2(1+\varepsilon) & C^2 + \varepsilon S^2 \\
\frac{1}{2}(1-\varepsilon)CS & \varepsilon C^2 + S^2 & -\frac{1}{2}(1-\varepsilon)CS
\end{bmatrix}, \]
Set $f = 0$, and also $u = 0$ on the boundary. Select a random initial guess for the solution and run 20 cycles with a 64 by 64 finest grid for each of the following questions.

a) For $\phi = 0$, test the convergence factor per cycle for $V(1,1)$ cycles, with $\varepsilon = 1, 1/3, 1/10, 1/100$. Use Jacobi relaxation with the theoretically optimal damping, and Red-Black, with and without the theoretical over-relaxation parameter. Check whether the performance is roughly as predicted by the analysis.

b) Set $\varepsilon = 10^{-5}, \phi = \pi/4$. Using Red-Black relaxation, compare the convergence factors of $V(2,1)$ cycles and $W(2,1)$ cycles. For the $W(2,1)$ cycles find approximately the optimal residual-overweighting parameter, and compare the performance.
Inter-grid Transfers and Non-ellipticity

Petrov-Galerkin Coarsening

A widely used method for constructing the coarse-grid operator is Petrov-Galerkin:

\[ L^H = R^h L^h P^h. \]

what is the relative order of approximation, \( q_{h,H} \)?

Theorem:

\[ q_{h,H} = \max (m, 0), \]

with

\[ m = \min (m_R, m_p, n_R + n_p - \tilde{M}), \]

where, usually, \( M = \tilde{M}, \) but for smooth characteristic components, \( M = \tilde{M} + p_h. \)
Inter-grid Transfers and Non-ellipticity

Petrov-Galerkin Coarsening

Proof (in $d$ dimensions, with $H = 2h$): fix $\omega$ and denote

$$J = \left\{ \begin{array}{c} \{ 0, \frac{\pi}{h} \} \\
\{0^d\} \end{array} \right\}.$$

Then, since

$$\hat{L}^h (\omega \pm j) = O \left( h^{-M} \right), \quad j \in J,$$

we obtain

$$\hat{L}^{H} (\omega) = \hat{L}^{h} (\omega) \hat{P}^{h} (\omega) \hat{R}^{h} (\omega) +$$

$$\sum_{j \in J} \hat{L}^{h} (\omega \pm j) \hat{P}^{h} (\omega \pm j) \hat{R}^{h} (\omega \pm j)$$

$$= \hat{L}^{h} (\omega) \left[ 1 + O \left( h^{{\min(m_{P},m_{R})}} \right) \right] + O \left( h^{n_{P}+n_{R}-M} \right).$$
Inter-grid Transfers and Non-ellipticity

Petrov-Galerkin Coarsening

Hence,

$$\frac{\hat{L}^h(\omega)}{\hat{L}^H(\omega)} = \frac{1}{1 + O(h^{\min(m_p,m_R)}) + O(h^{n_p+n_R-M})}.$$ 

Observe that $\tilde{M}$ rather than $M$ appears in the denominator, because for smooth characteristic components,

$$\hat{L}^h(\omega)^{-1} = O(h^{-p_h}).$$

Thus, we get

$$\frac{\hat{L}^h(\omega)}{\hat{L}^H(\omega)} = \begin{cases} 
1 + O(h^m), & m > 0 \\
\rightarrow 0, & m < 0, \\
\approx \text{const.} \neq 1, & m = 0.
\end{cases}$$
Inter-grid Transfers and Non-ellipticity

Petrov-Galerkin Coarsening

We find that we require \( m > 0 \), equivalently, \( q_{h,H} > 0 \), to obtain the usual excellent correction of asymptotically smooth error components. Otherwise, we will see once again poor performance and, especially, strong deterioration in the case of \( V \) cycles.

To test this, we consider once again the rotated anisotropic diffusion operator with \( \varepsilon = 0 \) and several non-alignment angles, \( \phi \). We perform a two-level analysis, using the powerful alternating “zebra” relaxation, which is robust for this problem. We apply Galerkin coarsening, comparing three different inter-grid transfers.
Inter-grid Transfers and Non-ellipticity

Petrov-Galerkin Coarsening

<table>
<thead>
<tr>
<th>$f$</th>
<th>$n$</th>
<th>Standard Coarsening</th>
<th>$n_R=2$ $n_P=2$</th>
<th>$n_R=4$ $n_P=2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$0.15p$</td>
<td>1</td>
<td>0.75</td>
<td>0.54</td>
<td>0.24</td>
</tr>
<tr>
<td>$0.15p$</td>
<td>2</td>
<td>0.75</td>
<td>0.52</td>
<td>0.11</td>
</tr>
<tr>
<td>$0.15p$</td>
<td>3</td>
<td>0.75</td>
<td>0.52</td>
<td>0.09</td>
</tr>
<tr>
<td>$0.25p$</td>
<td>1</td>
<td>0.76</td>
<td>0.54</td>
<td>0.30</td>
</tr>
<tr>
<td>$0.25p$</td>
<td>2</td>
<td>0.75</td>
<td>0.52</td>
<td>0.13</td>
</tr>
<tr>
<td>$0.25p$</td>
<td>3</td>
<td>0.75</td>
<td>0.51</td>
<td>0.10</td>
</tr>
<tr>
<td>$0.40p$</td>
<td>1</td>
<td>0.76</td>
<td>0.60</td>
<td>0.47</td>
</tr>
<tr>
<td>$0.40p$</td>
<td>2</td>
<td>0.75</td>
<td>0.55</td>
<td>0.28</td>
</tr>
<tr>
<td>$0.40p$</td>
<td>3</td>
<td>0.75</td>
<td>0.53</td>
<td>0.19</td>
</tr>
</tbody>
</table>

Two-grid convergence factors
Indefinite Operators

For $h$-elliptic discretizations, we know that good smoothers are available. Nevertheless, for some types of problems, the standard multigrid algorithm may stagnate or even diverge.

Consider (for example) once again the 1D model problem, slightly modified:

\[ Lu = u''(x) - \sigma u = f(x), \quad x \in (0,1), \]

\[ u(0) = u_0, \]

\[ u(1) = u_1. \]
Recall that the standard discretization produces the finite difference stencil

\[ L^h = \frac{1}{h^2} \begin{bmatrix} 1 & -2 - \sigma h^2 & 1 \end{bmatrix}, \]

whose symbol is

\[ \hat{L}^h = -\frac{4}{h^2} \sin^2 \left( \frac{h\omega}{2} \right) - \sigma, \]

If standard coarsening is used, then the symbol of the coarse-grid operator (for the same Fourier component) is

\[ \hat{L}^{2h} = -\frac{1}{h^2} \sin^2 (h\omega) - \sigma. \]
For good coarse-grid correction, we must have for all smooth components \((\omega = O(1))\):

\[
\frac{\hat{L}^h}{\hat{L}^{2h}} = \frac{-\omega^2 + O(h^2) - \sigma}{-\omega^2 + O(h^2) - \sigma} \approx 1.
\]

This clearly holds for non-negative \(\sigma\). But if \(\sigma\) is so negative such that \(L^h\) is nearly singular, \(\sigma \approx -\omega^2\), then this is no longer true, because the difference between the coarse-grid and the fine-grid symbols might even be larger than \(\hat{L}^h\) itself. The coarse-grid correction might even be of the wrong sign.

Note: as we decrease \(\sigma\), first the smoothest component causes problems, then the second smoothest component, etc.


**Remedies:**

- If there are only a few poorly-converging modes, then they can be corrected by a process of recombination of successive approximations such that the residual is minimized in some norm (as in GMRES).

- Use a different representation on the coarse-grid (Brandt and Livshits, 1997).

- In other similar cases, the problem is not as extreme, and then W cycles, possibly with some acceleration, may well suffice.

The Full Multi-Grid (FMG) Algorithm

The multigrid \( V \) cycle is an iterative method, and hence it requires an initial guess for the solution. This initial approximation is obtained from a coarser grid, and so on recursively.

The FMG algorithm combines the grid-refinement approach with the V cycle.

For many problems, FMG with just a single \( V \) cycle per level suffices to reduce the error below truncation level. In this case, only \( O(N) \) operations are required overall.
FMG ALGORITHM

Coarsest grid

Finest grid

○ RELAXATION  

\downarrow \hspace{1cm} \text{RERESTCTION} \hspace{1cm} \uparrow \hspace{1cm} \text{PROLONGATION}
Simplified quantitative analysis of FMG

Let $u^h_{FMG}$ denote the solution obtained on grid $h$ by the FMG algorithm. We want its algebraic error to be at worst comparable to the discretization error:

$$\left\| u^h - u^h_{FMG} \right\| \leq \beta \left\| I^h u - u^h \right\|,$$

where $u^h$ is the exact solution to the discrete problem, $u$ is the solution to the differential problem, $I^h$ is a restriction to the grid, and $\beta$ is a constant. This immediately implies:

$$\left\| I^h u - u^h_{FMG} \right\| \leq (1 + \beta) \left\| I^h u - u^h \right\|.$$

We see that there is no sense in working hard to reduce below $\beta \sim 1$, because the effort is better spent on reducing the discretization error by using a finer grid.
Suppose that a $p$-order discretization is used, so that for $u$ that is smooth on the scale of the grid,

$$I^h u - u^h \approx I^h c h^p,$$

where $c$ is some (nearly) $h$-independent function.

Suppose that the FMG solution on grid $2h$ satisfies the criterion we require. When we interpolate this solution to grid $h$ we have:

$$\left\| u^h - I_{2h}^h u_{FMG}^{2h} \right\|$$

$$= \left\| u^h - I_{2h}^h u^{2h} + I_{2h}^h (u^{2h} - u_{FMG}^{2h}) \right\|$$

$$\leq \left\| u^h - I_{2h}^h u^{2h} \right\| + \left\| I_{2h}^h (u^{2h} - u_{FMG}^{2h}) \right\|$$

$$\sim (2^p - 1) I^h c h^p + \beta I^h c (2h)^p$$

$$= [(1 + \beta) 2^p - 1] I^h c h^p.$$
Hence, in order to continue satisfying the criterion as the grid is further refined, we need to reduce the error by a factor \( \beta / [(1 + \beta)2^p - 1] \) or better.

Alternatively, if we reduce the error by a factor \( \mu_V \) per \( V \) cycle, then

\[
\beta = \frac{\mu_V (2^p - 1)}{1 - 2^p \mu_V}.
\]

For example, if \( \mu_V = 0.1 \) and \( p = 2 \), then \( \beta = 0.5 \).
Systems

Multigrid is just as useful for systems of PDE’s as it is for scalar problems.

If the system is decoupled, its solution amounts to repeated use of the standard multigrid algorithm. For example, the biharmonic equation,

$$\Delta^2 u = f,$$

is often solved as a system of equations:

$$\Delta u = v,$$
$$\Delta v = f.$$ 

If periodic “boundary conditions” are imposed, for example, we have two decoupled Poisson problems, and we can simply solve the second, then the first.
Other boundary conditions will generally lead to a coupling at the boundary. For elliptic systems, it is generally possible to match (essentially) the periodic-BC efficiency at the small cost of additional processing near the boundaries and corners.

For analysis purposes, it is useful to write systems in matrix-vector form, e.g.,

\[
\begin{pmatrix}
\Delta & -1 \\
0 & \Delta
\end{pmatrix}
\begin{pmatrix}
u \\
v
\end{pmatrix}
= \begin{pmatrix}
0 \\
f
\end{pmatrix}.
\]

In this example, we see that the matrix operator is triangular, implying that (subject to appropriate treatment near the boundaries), the coupling is benign.
Consider next the 2D Stokes equations:

\[ u_x + v_y = f_0 \]
\[ -\Delta u + p_x = f_1, \quad \text{in } \Omega, \]
\[ -\Delta v + p_y = f_2, \]

or, in matrix form,

\[
L \begin{pmatrix} p \\ u \\ v \end{pmatrix} = \begin{pmatrix} 0 & \partial_x & \partial_y \\ \partial_x & -\Delta & 0 \\ \partial_y & 0 & -\Delta \end{pmatrix} \begin{pmatrix} p \\ u \\ v \end{pmatrix} = \begin{pmatrix} f_0 \\ f_1 \\ f_2 \end{pmatrix}.
\]

Now it is far less clear how to solve (or indeed, discretize) the problem. Much is learned by examining the determinant of the operator: \( \det(L) = \Delta^2. \)
Two boundary conditions are required at each point on the boundary of the domain. Usually, we are given the velocities, \( u \) and \( v \), on the boundaries:

\[
\begin{pmatrix}
  u \\
  v
\end{pmatrix} = G(x), \quad \text{on } \partial \Omega.
\]

The resulting system and boundary conditions constitute a well-posed problem provided that the compatibility condition obtained from the continuity equation and the divergence theorem is satisfied:

\[
\int_{\Omega} f_0 \, dx = \int_{\partial \Omega} G \, d\sigma,
\]

where \( d\sigma \) is the boundary element multiplying an outward normal unit vector.
**Systems: Discretization**

The properties of the discretization are best examined by considering the implied discretization of the determinant. If this discretization is not $h$-elliptic, the solution will often contain spurious oscillations (in addition to the difficulties of multigrid solution). Therefore, a **staggered-grid** discretization is often used:

```
  v     v  
  
 u   p   u  p  u  h  
  
  v     v  
  
 u   p   u  p  u  
  
  v     v  
```
Systems: Discretization

The variables, \( p, u, \) and \( v \), are designated at distinct locations. The continuity, \( x \)-momentum and \( y \)-momentum equations (hence, \( f_0, f_1, \) and \( f_2 \)) are centered at the corresponding locations. Then, the standard second-order differencing is applied for the Laplacian, and short central differences for the first derivatives.

Coarsening is now fairly straightforward, but how should we relax? It seems clear that the \( x \) and \( y \) momentum equations can be relaxed by Gauss-Seidel or Jacobi for \( u \) and \( v \), respectively. But this leaves the continuity equation, and the \( p \) variable, which does not even appear in this equation.
Systems: Relaxation

The components of the determinant are (5-point) Laplacians. This tells us that it is possible to devise a relaxation for the system that is as effective as is relaxation for this Laplacian. We obtain this efficiency by means of distributive relaxation. The motivation comes from an implicit variable transformation that decouples the system (ignoring boundary conditions for now). Let

\[
\begin{pmatrix}
  p \\
  u \\
  v
\end{pmatrix}
= 
M
\begin{pmatrix}
  \tilde{p} \\
  \tilde{u} \\
  \tilde{v}
\end{pmatrix},
\]

where \( M \) is called a distribution matrix. Our problem in the new variables can be written as
Systems: Relaxation

\[
\begin{pmatrix}
\ddot{p} \\
\ddot{u} \\
\ddot{v}
\end{pmatrix}
= M
\begin{pmatrix}
f_0 \\
f_1 \\
f_2
\end{pmatrix}.
\]

The trick now is to choose \( M \) such that the operator \( LM \) is triangular with just Laplacians on the diagonal. This is obtained by choosing

\[
M = \begin{pmatrix}
-\Delta & 0 & 0 \\
-\partial_x & 1 & 0 \\
-\partial_y & 0 & 1
\end{pmatrix},
\]

yielding

\[
LM = \begin{pmatrix}
-\Delta & \partial_x & \partial_y \\
0 & -\Delta & 0 \\
0 & 0 & -\Delta
\end{pmatrix}.
\]
**Systems: Relaxation**

In principle, on an infinite domain (or a finite one with periodicity imposed) we could transform the variables explicitly, solve the resulting decoupled system, which requires just three solutions of Poisson problems, and then reconstruct the original variables.

In practical problems this won’t work, because of the coupling at the boundaries. Instead, we apply multigrid to the original variables but employ relaxation that is equivalent to Gauss-Seidel for the transformed variables (away from boundaries). The way to do this is implied by the discrete form of $\mathbf{M}$, specifically by its columns.
Systems: Relaxation

The upshot is that, in each relaxation sweep, we first relax the momentum equations for $u$ and $v$. Then, we relax the continuity equation by introducing changes in a local neighborhood of the relaxed point as indicated by the first column of $M$, as follows:
Systems: Relaxation

Here, $\delta$ is equal to the dynamic residual of the corresponding continuity equation before the relaxation, multiplied by $h/4$. It is easy to verify that, right after we relax at a point, the residual of the continuity equation at that point vanishes, while the residuals of the momentum equations remain unchanged. Near the boundaries we must alter the relaxation to take the coupling into account. Even then, full efficiency can only be achieved at the cost of small extra work along the boundaries.

There are other types relaxation methods for Stokes (notably Vanka). However, the distributive approach is a general tool that ensures that we obtain the same smoothing factor for the system as we would get for the factors of its determinant.
Systems

Relevant references for this and related treatment of the Stokes equations


More generally on relaxing systems


Algebraic Multigrid (AMG)

Introduced by Brandt et al. (1983) and developed by Ruge and Stueben.

AMG takes the algebrization of multigrid to the limit. Here, a relaxation method is chosen (usually, point Gauss-Seidel), and then the coarse-grid variables are chosen by a heuristic graph algorithm such that each fine-grid variable depends strongly on one or more coarse-grid variable (i.e., with relatively large coefficient).

AMG enables us to handle unstructured and non-PDE problems.
An Abstract View of AMG

Consider the linear system

\[ Au = f. \]

Suppose we partition the variables, \( u_i \), into \( F \) variables and \( C \) variables, and permute the equations and variables to produce the following partitioning of the system:

\[
Au = \begin{pmatrix}
A_{FF} & A_{FC} \\
A_{CF} & A_{CC}
\end{pmatrix}\begin{pmatrix}
u_F \\
u_C
\end{pmatrix} = \begin{pmatrix}
f_F \\
f_C
\end{pmatrix}.
\]
An Abstract View of AMG

Given an approximate solution, \( \tilde{u} \), define the error as

\[
\nu = u - \tilde{u}.
\]

Then, the partitioned equation for the error is

\[
A\nu = \begin{pmatrix}
A_{FF} & A_{FC} \\
A_{CF} & A_{CC}
\end{pmatrix}
\begin{pmatrix}
\nu_F \\
\nu_C
\end{pmatrix} = \begin{pmatrix}
r_F \\
r_C
\end{pmatrix},
\]

where

\[
r_F = f_F - A_{FF}\tilde{u}_F - A_{FC}\tilde{u}_C,
\]

\[
r_C = f_C - A_{CF}\tilde{u}_F - A_{CC}\tilde{u}_C.
\]
An Abstract View of AMG

The upper block yields

\[ A_{FF} v_F = r_F - A_{FC} v_C, \]

\[ \Rightarrow v_F = A_{FF}^{-1} (r_F - A_{FC} v_C). \]

Plugging this into the lower block yields

\[ A_{CF} A_{FF}^{-1} (r_F - A_{FC} v_C) + A_{CC} v_C = r_C, \]

\[ \Rightarrow (A_{CC} - A_{CF} A_{FF}^{-1} A_{FC}) v_C = r_C - A_{CF} A_{FF}^{-1} r_F. \]
An Abstract View of AMG

Conclusion: the “ideal” prolongation and restriction are

\[ P = \begin{pmatrix} -A_{FF}^{-1}A_{FC} \\ I \end{pmatrix}, \quad R = \begin{pmatrix} -A_{CF}A_{FF}^{-1} & I \end{pmatrix}, \]

with the coarse-grid operator given by

\[ A_C = RAP = A_{CC} - A_{CF}A_{FF}^{-1}A_{FC}. \]
In particular, it is straightforward to verify that the two-level solution is exact in this case, provided that either a pre-relaxation or a post-relaxation eliminates $r_F$.

(If this is done by post-relaxation, only $u_F$ should be relaxed.)

The problem: $A_{FF}^{-1}$ is not sparse, and therefore, neither are $P$ and $R$. Therefore, we generally look for good sparse approximations.

One exception is tri-diagonal systems, where $A_{FF}$ is diagonal. In this case the multigrid $V$-cycle with the appropriate prolongation and restriction, and with relaxation only on $u_F$ is an exact solver, equivalent to total reduction.