Seminar: The Interplay between Mathematical Modelling and Numerical Simulation

Introduction to the Multigrid Method

Bogojeska Jasmina

JASS, 2005

Abstract

The methods for solving linear systems of equations can be divided into two categories: *direct* and *iterative methods*. The first ones can determine the exact solutions, but are rather slow and are restricted to a certain small set of problems for which they show good performance. The iterative methods can be applied to a broader range of problems, but cannot damp the smooth components of the error and because of that in some cases show a very slow convergence.

The *multigrid methods* have developed from the main idea that the amount of computational work should be proportional to the amount of real physical changes in the computed system. In fully developped multigrid processes the amount of computations should be determined only by the amount of real physical information

1. Model Problems

The boundary value problems give a simple testing ground for providing a basic introduction to the *multigrid methods.* Although most of these problems can be handled analytically, the numerical methods will be presented and they will serve as model problems in order to present the multigrid method in a natural way.

The one-dimensional boundary value problem describing the steady-state temperature distribution in a long uniform rod is given by:

$$-u''(x)+\sigma\cdot u(x)=f(x) \qquad 0<x<1$$
$$u(0)=u(1)=0 \qquad \sigma\geq 0$$

With the grid points $x_j = j\cdot h$, $j=0,1,...,n$ where $h=\frac{1}{n}$, the domain of the problem is divided into *n* subintervals. The grid for this problem shown on *Figure 1* will be denoted with Ω^n.

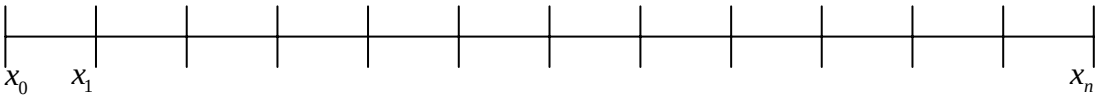

Figure 1

According to the finite difference method in the interior grid points the original differential equations can be replaced by a second-order finite difference approximation:

$$\frac{-v_{j-1}+2\cdot v_j-v_{j+1}}{h^2}+\sigma\cdot v_j=f_j \quad j=1,2,...,n-1$$
$$v_0=v_n=0$$

where v_j is an approximation to the exact solution $u(x_j)$ and f_j is $f(x_j)$ for $j=1,2,\text{K},n-1$. Defining $\mathbf{f}=\left(f(x_1),\text{K},f(x_{n-1})\right)$ and $\mathbf{v}=(v_1,\text{K},v_{n-1})$ the matrix form $\mathbf{A}\cdot\mathbf{v}=\mathbf{f}$ of the system above is:

$$\frac{1}{h^2}\begin{bmatrix} 2+\sigma\cdot h^2 & -1 & & & & \\ -1 & 2+\sigma\cdot h^2 & -1 & & & \\ & \ddots & \ddots & \ddots & & \\ & & \ddots & \ddots & -1 & \\ & & & -1 & 2+\sigma\cdot h^2 & \end{bmatrix}\begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ \vdots \\ v_{n-1} \end{bmatrix}=\begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ \vdots \\ f_{n-1} \end{bmatrix}$$

where $\mathbf{A}$ is a $(n-1)\times(n-1)$ symmetric, positive, definite matrix.

The two-dimensional boundary value problem has the form:

$$-u_{xx}-u_{yy}+\sigma\cdot u=f(x,y), \quad 0<x<1,\ 0<y<1,\ \sigma>0$$

where $u=0$ on the boundary of the unit square. The grid showed on *Figure 2*, is defined with the points $(x_i,y_j)=(i\cdot h_x, j\cdot h_y)$, where $i=1,...,m-1,\ j=1,...,m-1,\ h_x=\frac{1}{m}$ and $h_y=\frac{1}{n}$.

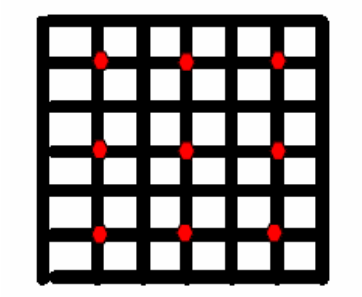

Figure 2

In the same way as for the one-dimensional boundary value problem, replacing the derivatives by the second-order finite differences leads to the system of linear equations:

$$\frac{-v_{i-1,j}+2\cdot v_{ij}-v_{i+1,j}}{h_x^{\,2}}+\frac{-v_{i,j-1}+2\cdot v_{ij}-v_{i,j+1}}{h_y^{\,2}}+\sigma\cdot v_{ij}=f_{ij}$$

$$v_{i0}=v_{in}=0,\quad i=1,2,...,m-1,\quad j=1,2,...,n-1$$

where v_{ij} is an approximation of the exact solution $u\left(x_i,y_j\right)$ and $f_{ij}=f(x_i,y_j)$. By using lexicographical ordering by lines one can define $v_i=(v_{i1},...,v_{i,n-1})^{\mathrm{T}}$ and $f_i=(f_{i1},...,f_{i,n-1})^{\mathrm{T}}$ for $i=1,...,m-1$. According to this notation the block-tridiagonal matrix form of the system is $\mathbf{A}\cdot\mathbf{v}=\mathbf{f}$ i.e.:

$$\begin{bmatrix} A_1 & -a\cdot I & & & & \\ -a\cdot I & A_2 & -a\cdot I & & & \\ & \cdot & \cdot & \cdot & & \\ & & \cdot & \cdot & \cdot & \\ & & & -a\cdot I & A_{n-2} & -a\cdot I \\ & & & & -a\cdot I & A_{n-1} \end{bmatrix}\cdot\begin{bmatrix} v_1 \\ v_2 \\ \cdot \\ \cdot \\ v_{n-2} \\ v_{n-1} \end{bmatrix}=\begin{bmatrix} f_1 \\ f_2 \\ \cdot \\ \cdot \\ f_{n-2} \\ f_{n-1} \end{bmatrix}$$

where $a=\frac{1}{h_y^{\,2}}$, I is an $(n-1)\times(n-1)$ identity matrix and A_i is an $(n-1)\times(n-1)$ tri-diagonal matrix given with:

$$A_i=\begin{bmatrix} \frac{1}{h_x^2+h_y^2}+\sigma & -\frac{1}{h_x^2} & & & \\ -\frac{1}{h_x^2} & \frac{1}{h_x^2+h_y^2}+\sigma & -\frac{1}{h_x^2} & & \\ & \cdot & \cdot & \cdot & \\ & \cdot & \cdot & \cdot & \\ & & & -\frac{1}{h_x^2} & \frac{1}{h_x^2+h_y^2}+\sigma \end{bmatrix}$$

2. Basic Iterative Schemes

The next step is to consider how the model problems that are defined in the previous section can be solved using some basic *iterative* or *relaxation* schemes. The problems will be given in their matrix form $\mathbf{A}\cdot\mathbf{u}=\mathbf{f}$, where $\mathbf{u}$ is the exact solution and $\mathbf{v}$ is the corresponding approximation. The vector norms will be used as a measure for the error that is defined with $\mathbf{e}=\mathbf{u}-\mathbf{v}$. The *residual equation* is $\mathbf{A}\cdot\mathbf{e}=\mathbf{r}$, where the *residual* is defined as $\mathbf{r}=\mathbf{f}-\mathbf{A}\cdot\mathbf{v}$. The equation $\mathbf{u}=\mathbf{v}+\mathbf{e}$ is the *residual correction.* When using the equation $\mathbf{u}-\mathbf{v}=\mathbf{A}^{-1}\cdot\mathbf{r}$ and $\mathbf{v}\to\mathbf{v}^{(old)}$, $\mathbf{u}\to\mathbf{v}^{(new)}$, an *iteration* $\mathbf{v}^{(new)}=\mathbf{v}^{(old)}+\mathbf{B}\cdot\mathbf{r}^{(old)}$ can be formed where $\mathbf{B}$ is an approximation to $\mathbf{A}^{-1}$. The equation for the iteration can take slightly different form $\mathbf{v}^{(new)}=\mathbf{R}\cdot\mathbf{v}^{(old)}+\mathbf{B}\cdot\mathbf{f}=\mathbf{R}\cdot\mathbf{v}^{(old)}+g$, where $\mathbf{R}=\mathbf{I}-\mathbf{B}\cdot\mathbf{A}$. Using this form the exact solution will be fixed point i.e. $\mathbf{u}=\mathbf{R}\cdot\mathbf{u}+g$. The error will be given by $\mathbf{e}^{(new)}=\mathbf{R}\cdot\mathbf{e}^{(old)}$, or $\mathbf{e}^{(m)}=\mathbf{R}^m\cdot\mathbf{e}^{(0)}$ if m iterations are performed and it can be bounded with $\left\|\mathbf{e}^{(m)}\right\|_\infty\le\left\|\mathbf{R}\right\|^m\cdot\left\|\mathbf{e}^{(0)}\right\|$, where some proper vector and matrix norms are used. From this inequality it follows that the error will tend to zero in the relaxation process if $\left\|\mathbf{R}\right\|<1$.

Definition 1 *Assymptotic convergence factor is the spectral radius defined as* $\rho(\mathbf{R})=\max\left\{|\lambda_1|,...,|\lambda_n|\right\}$.

Lemma $\mathbf{R}^m\to\mathbf{0}$ *as* $\mathrm{m}\to\infty$ *if and only if* $\rho(\mathbf{R})<1$.

Using the lemma defined above and taking into consideration that for any initial vector $\mathbf{v}^{(0)}$, $\mathbf{e}^{(m)}\to 0$ as $m\to\infty$ if and only if $\rho(\mathbf{R})<1$, it can be concluded that the convergence of the iteration is given by the condition $\rho(\mathbf{R})<1$.

3.1 Jacobi Relaxation

One of the basic relaxation schemes is the *Jacobi Relaxation Scheme.* For simplicity the one-dimensional boundary value problem will be considered with σ=0 i.e.:

$$-u_{j-1}+2\cdot u_j-u_{j+1}=h^2\cdot f_j,\ \ j=1,2,...,n-1,\ \ u_0=u_1=0$$

The Jacobi relaxation for this problem is given by the following system of equations:

$$v_j^{(new)}=\frac{1}{2}\left(v_{j-1}^{(old)}+v_{j+1}^{(old)}+h^2\cdot f_j\right),\ j=1,...,n-1$$

The corresponding matrix form is $\mathbf{v}^{(new)}=\mathbf{R_J}\cdot\mathbf{v}^{(old)}+\mathbf{D}^{-1}\cdot\mathbf{f}$, where $\mathbf{R_J}=\mathbf{D}^{-1}\cdot(\mathbf{L}+\mathbf{U})$ and $\mathbf{A}=\mathbf{D}-\mathbf{L}-\mathbf{U}$.

The weighted or damped Jacobi relaxation is defined with:

$$v_j^{(new)}=(1-\omega)\cdot v_j^{(old)}+\frac{\omega}{2}\left(v_{j-1}^{(old)}+v_{j+1}^{(old)}+h^2\cdot f_j\right),\ j=1,...,n-1$$

or with the equivalent matrix form $\mathbf{v}^{(new)}=\mathbf{R_\omega}\cdot\mathbf{v}^{(old)}+\omega\cdot h^2\cdot\mathbf{D}^{-1}\cdot\mathbf{f}$, where $\mathbf{R_\omega}=(1-\omega)\mathbf{I}+\omega\mathbf{R_J}$ and $\omega\in$ ¡ is a *weighting factor* that is properly chosen.

3.2 Gauss-Seidel Relaxation

The *Gauss-Seidel relaxation* is similar to the Jacobi relaxation and for the simplified one-dimensional model problem with σ=0 is defined as follows:

$$v_j^{(new)} = \frac{1}{2}\left(v_{j-1}^{(new)} + v_{j+1}^{(old)} + h^2 \cdot f_j\right),\ j = 1,\ldots,n-1$$

or using a matrix form $\mathbf{v}^{(new)} = \mathbf{R_G} \cdot \mathbf{v}^{(old)} + \left(\mathbf{D}-\mathbf{L}\right)^{-1} \cdot \mathbf{f}$, where $\mathbf{R_G} = \left(\mathbf{D}-\mathbf{L}\right)^{-1} \cdot \mathbf{U}$ and $\mathbf{A} = \mathbf{D}-\mathbf{L}-\mathbf{U}$. The difference from the Jacobi relaxation is that Gauss-Seidel uses the components of the new approximation as soon as they are calculated, which reduces the storage requirements for the approximation vector $\mathbf{v}$ to n locations, because there is no need for keeping the values of this vector for the old and the new iteration.

3.3 Fourier Modes

For simplicity we will consider the homogeneous linear system $\mathbf{A} \cdot \mathbf{u} = \mathbf{0}$. We immediately can see that the exact solution to this system is $\mathbf{u} = \mathbf{0}$ and the error is $\mathbf{e} = \mathbf{u} - \mathbf{v} = -\mathbf{v}$.

Definition 2 The vectors $v_j = \sin\left(\frac{j \cdot k \cdot p}{n}\right)$, $0 \le j \le n$, $1 \le k \le n-1$, where k is frequency or wavenumber indicating the number of half-sine waves that constitute $\mathbf{v}$ on the domain are called *Fourier modes (Figure 3).*

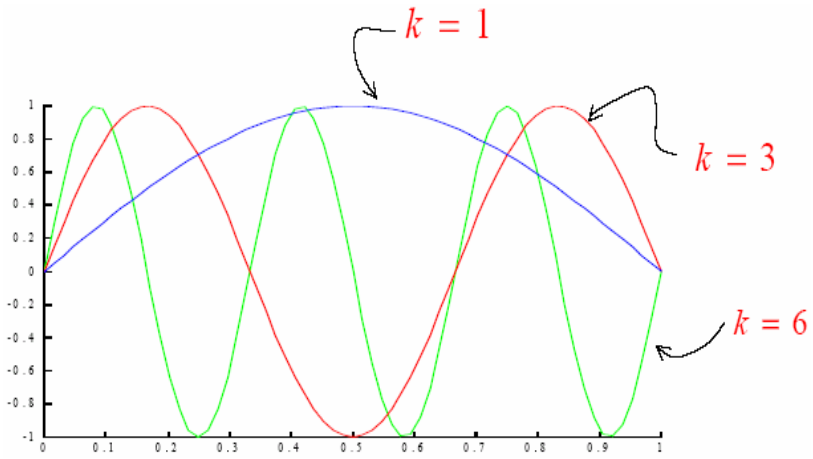

Figure 3

Definition 3 The wavenumbers in range $1 \le k \le \frac{n}{2}$ are called *smooth* or *low-frequency modes,* and those in range $\frac{n}{2} \le k \le n-1$ are called *oscillatory* or *high-frequency modes.*

If we take the Fourier modes given in Figure 3 as initial iteration and we perform 100 sweeps of the weighted Jacobi iteration, we will get the results for the error shown in Figure 4. As we can see on the figure the error decreases with each iteration, but the higher wave numbers show much larger rate of decrease.

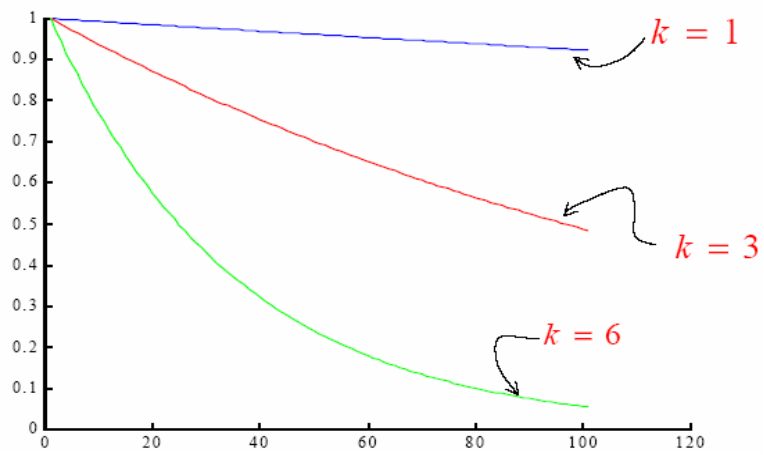

Figure 4

In order to see a more realistic case we take an initial guess that does not contain only single mode, but a combination of a low-frequency, medium-frequency and high-frequency wave, i.e.: $v_j = \frac{1}{3}\left[\sin\left(\frac{j\cdot\pi}{n}\right)+\sin\left(\frac{6\cdot j\cdot\pi}{n}\right)+\sin\left(\frac{32\cdot j\cdot\pi}{n}\right)\right]$. As we can see on *Figure 5* the error decreases very fast only in the first five iterations. After that the decrease of the error becomes very slow.

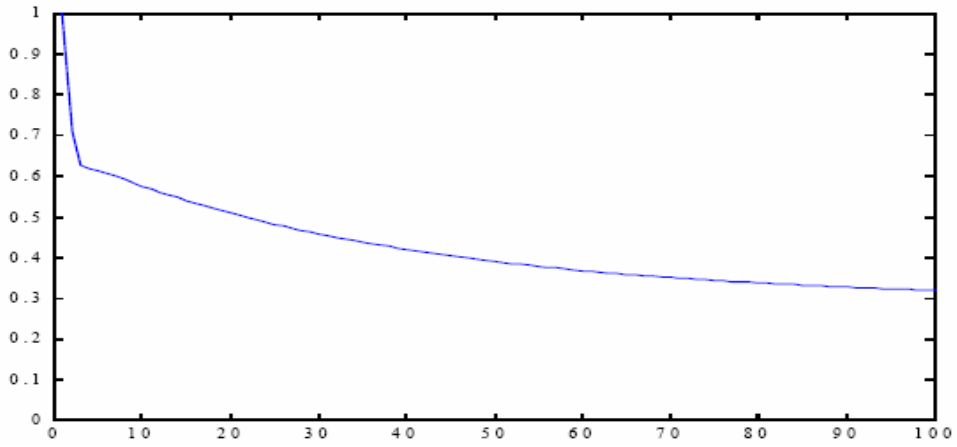

Figure 5

The quick elimination of the high-frequency modes of the error gives the fast initial decrease. The presence of the low-frequency modes results in a very slow error decrease as we continue with the iterations and significantly degrades the performance of the standard iteration methods. The iterations would converge fast only if the error contains high-frequency modes, which are damped very fast.

In order to see why this happened we must examine the problem a bit more formally. At first it should be pointed out that the weighted Jacobi method preserves modes, i.e. performing the relaxations only the amplitude of the modes is changed. Having the fact that $\mathbf{R}_\omega = \mathbf{I} - \frac{\omega}{2}\cdot\mathbf{A} \Rightarrow \lambda(\mathbf{R}_\omega) = 1 - \frac{\omega}{2}\cdot\lambda(\mathbf{A})$, we get that $\mathbf{R}_\omega$ and $\mathbf{A}$ have the same eigenvectors:

$w_{k,j} = \sin\left(\frac{j \cdot k \cdot \pi}{n}\right),\ 1 \le k \le n-1.$ The eigenvalues of **A** are given with: $\lambda_k(\mathbf{A}) = 4 \cdot \sin^2\left(\frac{k \cdot \pi}{2 \cdot n}\right),\ 1 \le k \le n-1$

and the eigenvalues of $\mathbf{R}_\omega$ are:

$$\lambda_k(\mathbf{R}_\omega) = 1 - 2 \cdot \omega \cdot \sin^2\left(\frac{k \cdot \pi}{2 \cdot n}\right),\ 1 \le k \le n-1.$$

Having the eigenvectors of **A**, we can expand the error vector $\mathbf{e}^{(0)}$ in the form: $\mathbf{e}^{(0)} = \sum_{\mathbf{k=1}}^{\mathbf{n-1}} c_k \cdot \mathbf{w_k}$. Using the formula for the error after **m** iterations and the fact that the eigenvectors of $\mathbf{R}_\omega$ and **A** are the same we get: $\mathbf{e^m} = \sum_{\mathbf{k=1}}^{\mathbf{n-1}} c_k \cdot \mathbf{R}_\omega^{\mathbf{m}} \cdot \mathbf{w_k} = \sum_{\mathbf{k=1}}^{\mathbf{n-1}} c_k \cdot \lambda_k^m(\mathbf{R}_\omega^{\mathbf{m}}) \cdot \mathbf{w_k}$. In the last formula we can clearly see that the **k**th mode of the error after **m** iterations is reduced by a factor of $\lambda_k^m(\mathbf{R}_\omega^{\mathbf{m}})$. If $0 < \omega \le 1$ then $|\lambda_k(\mathbf{R}_\omega)| < 1$ and we will have a convergent Jacobi iteration. But for all ω, $0 < \omega \le 1$ we get that:

$$\lambda_1 = 1 - 2 \cdot \omega \cdot \sin^2\left(\frac{\pi}{2 \cdot n}\right) = 1 - 2 \cdot \omega \cdot \sin^2\left(\frac{\pi \cdot h}{2}\right) \approx 1 - \frac{\omega \cdot \pi^2 \cdot h^2}{2}.$$

According to this formula the eigenvalue that corresponds to the smoothest mode will always be close to one for any choice of ω and therefore the smooth components of the error converge very slowly. If we want to improve the accuracy of the solution by taking smaller grid spacing h then λ_1 will be even more close to 1. No value of ω can reduce the smooth components of the error. We can only find the value of ω that provides us with the best damping of the oscillatory modes of the error. Solving the equation $\lambda_{n/2}(\mathbf{R}_\omega) = -\lambda_n(\mathbf{R}_\omega)$ for the weighted Jacobi method leads to $\omega = \frac{2}{3}$ and $|\lambda_k| \le \frac{1}{3}$, for $\frac{n}{2} \le k \le n-1$, which tells us that the oscillatory components of the error will be reduced at least by a factor of three in each iteration sweep. This brings us to an important characteristic of each standard relaxation scheme.

Definition 4 The largest absolute value among the eigenvalues in the upper half of the spectrum (the oscillatory modes) of the iteration matrix is called *smoothing factor.*

3. The Multigrid Method

4.1 Coarse Grids

Providing a good initial guess can improve the performance of a relaxation scheme in the initial iteration sweeps. A good way for getting a better initial guess is taking a coarse grid and performing a certain number of iterations. On *Figure 6* a smooth wave (wavenumber 4) is shown on a grid with 12 points and on a coarse grid with 6 points. We see that the smooth wave on the fine grid looks more oscillatory when pojected on the coarse grid i.e. the smoothing property when using coarse grids becomes an advantage. Moreover, the relaxation on a coarse grid is less expensive because there are less points that should be kept in memory and the coarse grid has a marginally improved convergence rate – the convergence factor is $1 - O(h^2)$.

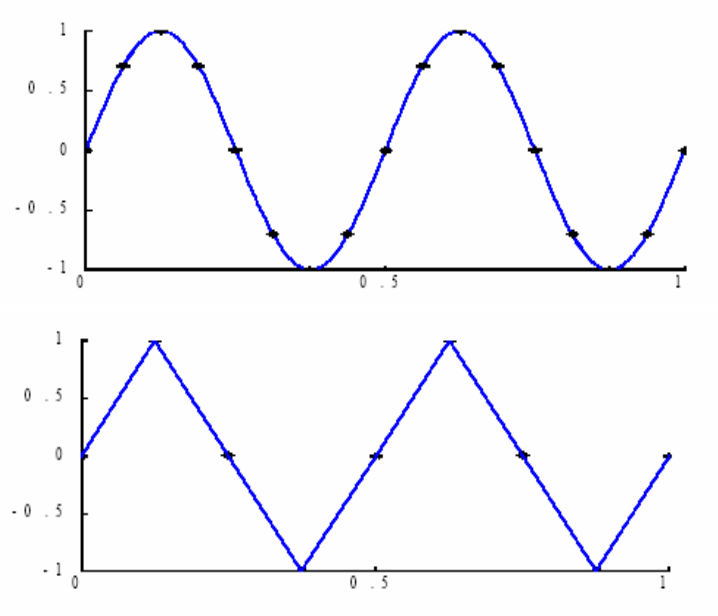

Figure 6

Let us see the projection of the smooth wave on the coarse grid into more detail. The kth mode on Ω^h becomes kth mode on $\Omega^{2\cdot h}$ for $1 \le k < \frac{n}{2}$ i.e. :

$w^h_{k,2\cdot j} = \sin\left(\frac{2\cdot j\cdot k\cdot \pi}{n}\right) = \sin\left(\frac{j\cdot k\cdot \pi}{n/2}\right) = w^{2\cdot h}_{k,2\cdot j}$. Because of aliasing, for $k > \frac{n}{2}$ the kth mode on Ω^h becomes $(n-k)$th mode on $\Omega^{2\cdot h}$ and the oscillatory modes will be misinterpreted as relatively smooth:

$$w^h_{k,2\cdot j} = \sin\left(\frac{2\cdot j\cdot k\cdot \pi}{n}\right) = -\sin\left(\frac{2\cdot j\cdot \pi\cdot (n-k)}{n}\right) = -\sin\left(\frac{j\cdot (n-k)\cdot \pi}{n/2}\right) = -w^{2\cdot h}_{n-k,2\cdot j}\,.$$

The concept of coarse grid and its main property of making smooth modes to look more oscillatory, gives the idea to move to coarser grid when the relaxation begins to stall because the relaxation will be more effective in damping the oscillatory components of the error.

4.2 Nested Iteration

The nested iteration is based on the idea of performing a certain number of preliminary iterations in order to get a better initial guess for the fine-grid iteration. It can be described as follows:

- Relax on $\mathbf{A}\cdot\mathbf{u} = \mathbf{f}$ on a very coarse grid to obtain an initial guess for the next finer grid

.....

- Relax on $\mathbf{A}\cdot\mathbf{u} = \mathbf{f}$ on $\Omega^{4\cdot h}$ to obtain an initial guess for $\Omega^{2\cdot h}$
- Relax on $\mathbf{A}\cdot\mathbf{u} = \mathbf{f}$ on $\Omega^{2\cdot h}$ to obtain an initial guess for Ω^{h}
- Relax on $\mathbf{A}\cdot\mathbf{u} = \mathbf{f}$ on Ω^{h} to obtain a final approximation to the solution

4.3 Correction Scheme

The correction scheme uses the idea that we can relax directly on the error by using the residual equation: $\mathbf{A}\cdot\mathbf{e}=\mathbf{r}=\mathbf{f}-\mathbf{A}\cdot\mathbf{v}$ with initial guess $\mathbf{e}=\mathbf{0}$. Additionally this previously described relaxation is equivalent to a relaxation on the equation $\mathbf{A}\cdot\mathbf{u}=\mathbf{f}$ with an arbitrary initial guess $\mathbf{v}$. The correction scheme can be described with:

- Relax on $\mathbf{A}\cdot\mathbf{u}=\mathbf{f}$ on Ω^h to obtain an approximation $\mathbf{v}^h$
- Compute the residual $\mathbf{r}=\mathbf{f}-\mathbf{A}\cdot\mathbf{v}^h$
- Relax on the residual equation $\mathbf{A}\cdot\mathbf{e}=\mathbf{r}$ on $\Omega^{2\cdot h}$ to obtain an approximation to the error $\mathbf{e}^{2\cdot h}$
- Correct the approximation obtained on Ω^h with the error estimate obtained on $\Omega^{2\cdot h}$: $\mathbf{v}^h \leftarrow \mathbf{v}^h+\mathbf{e}^{2\cdot h}$

The main idea here is that at first we relax on the fine grid. When the convergence becomes slow we relax on the residual equation on a coarser grid where we obtain an approximation to the error. Then we return back to the fine grid using the obtained approximation to the error.

4.4 Interpolation Operator

In the previous two subsections we gave two schemes that can potentially improve the performance of the relaxation methods. But some of the steps, like how do we transfer a vector from the coarse grid to the fine grid and vice versa, still need to be specified into more details. We should also point out that we will consider only the case where the coarse grid has twice as less points compared to the preceding fine grid. This is done because of simplicity and also because we will get the same conclusions using different grid spacings.

The interpolation operator is based on a common procedure in numerical analysis called *interpolation* or *prolongation* and provides us with the necessary tecnique for transferring the error approximation e^{2h} from the coarse grid Ω^{2h} to the fine grid Ω^h. Practise has shown that for most multigrid implementations the linear interpolation gives very good results, so we will also use it here.

The interpolation operator $\mathrm{I}_{2\cdot h}^{h}$ is a linear operator from ¡ $^{\frac{n}{2}-1}$ to ¡ $^{n-1}$, with a full rank and a trivial null-space. It can be seen as a mapping $\mathrm{I}_{2\cdot h}^{h}:\Omega^{2\cdot h}\to\Omega^h$, that transforms coarse-grid vectors into fine-grid vectors using the formula $\mathrm{I}_{2\cdot h}^{h}\cdot\mathbf{v}^{2\cdot h}=\mathbf{v}^h$, where $v_{2\cdot j}^{h}=v_j^{2\cdot h}$, $v_{2\cdot j+1}^{h}=\frac{1}{2}\left(v_j^{2\cdot h}+v_{j+1}^{2\cdot h}\right)$, $0\le j\le\frac{n}{2}-1$. This procedure is illustrated on *Figure 7.*

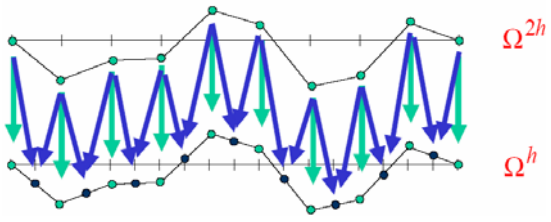

Figure 7

It is important to see how this operator works when we have smooth and when we have oscillatory vector on the fine grid. The interpolation process when the real vector is smooth is illustrated on *Figure 8.*

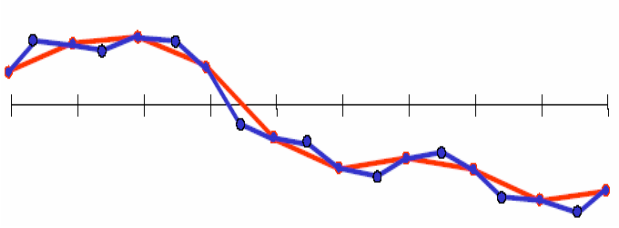

Figure 8

From the picture above we can see that if the error on Ω^h is smooth the interpolant will also be smooth, i.e. an interpolant of the coarse-grid error gives a good interpretation of the real error.

When the real error is oscillatory, *Figure 9* shows that the interpolant is smooth, i.e. in this case an interpolant of the coarse-grid error may give a poor interpretation of the real error.

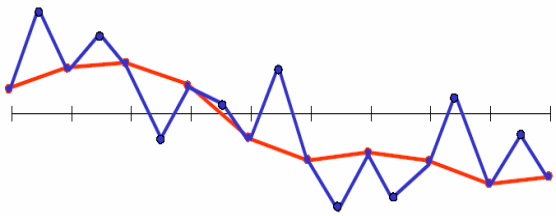

Figure 9

Being efficient when the error is smooth, the interpolation operator provides a complement to the relaxation process. The interpolation process is a part of the nested iteration and correction scheme, so they also show best performance for smooth errors.

4.5 Restriction Operator

The restriction operators are used for transferring vectors from a fine grid to a coarse grid. They are linear operators from ¡ $^{n-1}$ to ¡ $^{\frac{n}{2}-1}$ denoted as I_h^{2h}, are with a full rank and have a nullspace of dimension $\frac{n}{2}$. The restriction operators can be seen as mappings $\mathrm{I}_h^{2\cdot h}:\Omega^h \to \Omega^{2\cdot h}$ that using the formula $\mathrm{I}_h^{2\cdot h}\cdot \mathbf{v}^h = \mathbf{v}^{2\cdot h}$ take fine-grid vectors and produce coarse-grid vectors. The simplest one is *injection*, defined with $v_j^{2\cdot h} = v_{2\cdot j}^h$, where the corresponding value of the fine-grid point is simply taken as a value of the coarse-grid point. Another restriction operator is *full weighting*, defined with $v_j^{2\cdot h} = \frac{1}{4}\cdot\left(v_{2\cdot j-1}^h + 2\cdot v_{2\cdot j}^h + v_{2\cdot j+1}^h\right), 1 \le j \le \frac{n}{2}\text{-}1$, where we take weighted averages of values at neighbouring fine-grid points in order to get the values of the coarse-grid points. This process is illustrated in *Figure 10*.

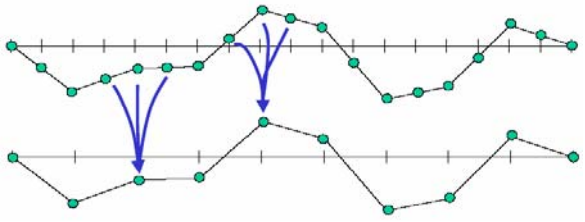

Figure 10

4.6 Two-Grid Correction Scheme

Having the detailed definitions of the interpolation and the restriction operator we can now give the procedure that describes the *two-grid correction scheme*.

$$\mathbf{v}^h \leftarrow MG(\mathbf{v}^h, \mathbf{f}^h)$$

- Relax ν_1 times on $\mathbf{A}^h \cdot \mathbf{u}^h = \mathbf{f}^h$ on Ω^h with initial guess $\mathbf{v}^h$
- Compute the fine-grid residual $\mathbf{r}^h = \mathbf{f}^h - \mathbf{A}^h \cdot \mathbf{v}^h$ and restrict

it to the coarse grid by $\mathbf{r}^{2h} = \mathrm{I}_h^{2h} \cdot \mathbf{r}^h$

- Solve $\mathbf{A}^{2h} \cdot \mathbf{e}^{2h} = \mathbf{r}^{2h}$ on Ω^{2h}
- Interpolate the coarse-grid error to the fine-grid by

$\mathbf{e}^{2h} = \mathrm{I}_{2h}^h \cdot \mathbf{e}^{2h}$ and correct the fine-grid approxiamtion by

$\mathbf{v}^h \leftarrow \mathbf{v}^h + \mathbf{e}^h$

- Relax ν_2 times on $\mathbf{A}^h \cdot \mathbf{u}^h = \mathbf{f}^h$ on Ω^h with initial guess $\mathbf{v}^h$

A nice illustration is given in *Figure 11*.

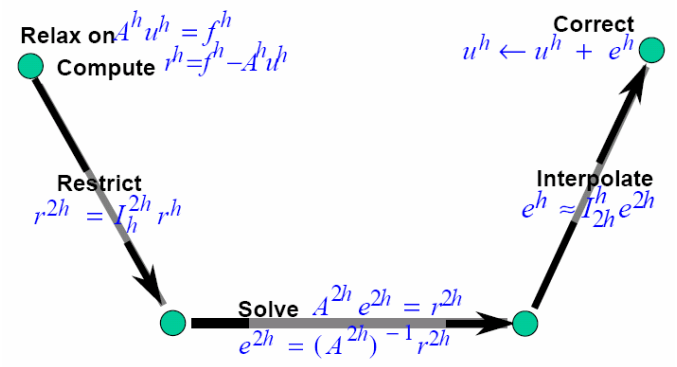

Figure 11

As we can see on the picture above, at the beginning we relax usually 1 to 3 times on the fine-grid. After we calculate the residual of the approximation that we got we transfer it by the restriction operator to the coarse grid. Then the residual equation is solved (or approximate solution is found) on the coarse grid. The last step is transferring the error (or the approximated error) with the interpolation operator back to the fine-grid and correcting the fine-grid approximation. This is also followed by a few iteration sweeps.

The important thing that should be noted here is that with the relaxation we eliminate the oscillatory components of the error, and assuming that we can get an accurate solution to the residual equation, the interpolation operator will get a relatively smooth error. As we know from before the interpolation operator is most effective on smooth errors, so we are supposed to get a good correction of the fine grid approximation.

4.7 V-Cycle Scheme

There is one problem in the procedure described in the previous subsection and that is the solution of the residual equation $\mathbf{A}^{2h} \cdot \mathbf{e}^{2h} = \mathbf{r}^{2h}$ on the coarse grid. If we can notice that this

problem is not much different than the original problem we can solve it recursively. Namely we can apply the two-grid correction procedure to the residual equation on Ω^{2h} and then move to a coarser grid i.e. Ω^{4h} in order to obtain the correction. We repeat the process until we reach a grid where we can find an exact solution of the residual equation (we can even reach to grids with one point if it is necessary). After that we go up to the finer grids using the corresponding interpolation operators. A notation modification is needed in order to be able to describe this recursive procedure algorithmically. The right hand-side of the residual equation will be denoted as $\mathbf{f^{2h}}$, $\mathbf{u^{2h}}$ will replace the solution of the residual equation $\mathbf{e^{2h}}$ and finally $\mathbf{v^{2h}}$ will denote the approximations to $\mathbf{u^{2h}}$. These changes are appropriate because solving the residual equation is handled the same way as the original equations and we get simplified notation for describing the whole procedure. It should also be pointed out that as a initial guess for the first visit to Ω^{2h} we will choose $\mathbf{v^{2h}} = 0$, because there no information available for the solution $\mathbf{u^{2h}}$. The process described above is shown in *Figure 12*.

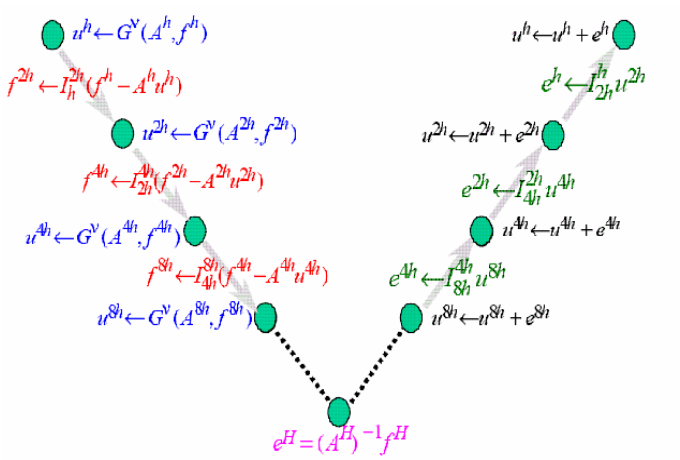

Figure 12

Taking into account that $\mathbf{v}^h$ and $\mathbf{f}^h$ must be stored at each level and that for d dimensions the coarser grid has 2^{-d} the number of points as the finer grid, for the storage costs of the V-Cycle we have $2n^d\left(1+2^{-d}+2^{-2d}+\ldots+2^{-Md}\right) < \frac{2n^d}{1-2^{-d}}$. The computational costs of a V-Cycle with one pre-Coarse-Grid correction relaxation sweep and one post-Coarse-Grid relaxation sweep are given with $2\left(1+2^{-d}+2^{-2d}+2^{-3d}+\ldots+2^{-Md}\right) < \frac{2}{1-2^{-d}}$, where the cost of one relaxation sweep on the fine grid is one working unit (WU).

4.8 Full Multigrid V-Cycle

The *full multigrid V-Cycle* combines the nested iteration and the V-Cycle. The basic idea here is that a better initial guess for the first fine-grid iteration of the V-Cycle can improve its performance. In the context of multigriding a good candidate is the nested iteration, which suggests performing preliminary iterations on the coarse grid Ω^{2h}. Now we also need an initial guess for the Ω^{2h} problem. The nested iteration uses recursion for solving this problem. Again we move the problem to the coarser grid Ω^{4h}, and we continue this process until we reach the coarsest grid where we can solve the problem explicitly. After that we move up to the finer grids using the interpolation operator.

The full multigrid V-Cycle, where the coarse-grid right-sides are initialized by transferring $\mathbf{f^h}$ from the fine grid, can be described with the following procedure:

$$\mathbf{v}^h \leftarrow FMG^h\left(\mathbf{f}^h\right)$$

g Initialize $\mathbf{f}^{2h} \leftarrow \mathrm{I}_h^{2h}\,\mathbf{f}^h,\ \mathbf{f}^{4h} \leftarrow \mathrm{I}_{2h}^{4h}\,\mathbf{f}^{2h},\ \ldots.$

g Solve or relax on coarsest grid

. . . . . .

g Interpolate initial guess $\mathbf{v}^{2h} \leftarrow \mathrm{I}_{4h}^{2h}\,\mathbf{v}^{4h}$

g Perform V-cycle $\mathbf{v}^{2h} \leftarrow MV^{2h}\left(\mathbf{v}^{2h}, \mathbf{f}^{2h}\right)$ ν_0 times

g Interpolate initial guess $\mathbf{v}^h \leftarrow \mathrm{I}_{2h}^h\,\mathbf{v}^{2h}$

g Perform V-cycle $\mathbf{v}^{2h} \leftarrow MV^h\left(\mathbf{v}^h, \mathbf{f}^h\right)$ ν_0 times

The parameter that specifies the number of V-Cycles performed at each level ν_0 is determined experimentally and usually has the value one. As we can see in the described procedure each V-Cycle is preceded by a V-Cycle performed on a coarser grid in order to provide a good initial guess.

Figure 13 bellow gives a nice schematic representation.

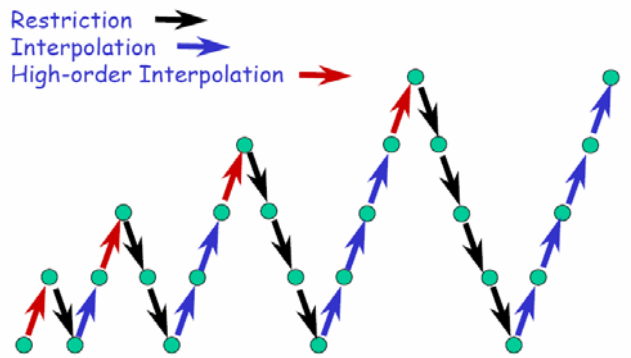

Figure 13

The recursive procedure for the full multigrid is as follows:

$$\mathbf{v}^h \leftarrow FMG^h\left(\mathbf{f}^h\right)$$

1. Initialize $\mathbf{f}^{2h} \leftarrow \mathrm{I}_h^{2h}\,\mathbf{f}^h,\ \mathbf{f}^{4h} \leftarrow \mathrm{I}_{2h}^{4h}\,\mathbf{f}^{2h},\ \ldots.$
2. If $\Omega^h \rightarrow$ coarsest grid, set $\mathbf{v}^h \leftarrow 0$, then go to 3.

   Else $\mathbf{f}^{2h} \leftarrow \mathrm{I}_h^{2h}\left(\mathbf{f}^h\right)$

   $\mathbf{v}^{2h} \leftarrow FMG^{2h}\left(\mathbf{f}^{2h}\right)$
3. Correct $\mathbf{v}^h \leftarrow \mathrm{I}_{2h}^h\,\mathbf{v}^{2h}$
4. $\mathbf{v}^h \leftarrow MV^h\left(\mathbf{v}^h, \mathbf{f}^h\right)$ ν_0 times

Taking into account that the size of the working unit for the coarse grid j is 2^{-jd} times of the size of the working unit on the fine grid, the costs of full multigrid for $\nu_0 = \nu_1 = ... = 1$ are less than $\left(\frac{2}{1-2^{-d}}\right)\left(1+2^{-d}+2^{-2d}+....\right) = \frac{2}{\left(1-2^{-d}\right)^2}$.

4.9 Building $\mathbf{A^{2h}}$

At the beginning for simplicity we assume that for the error holds $\mathbf{e^h} = \mathbf{u^h} - \mathbf{v^h} \in \Re\left(I_{2h}^h\right)$, i.e. the error lies in the range of the interpolation. From this it follows that there exists a vector $\mathbf{u^{2h}} \in \Omega^{2h}$, such that $\mathbf{e^h} = I_{2h}^h \cdot \mathbf{u^{2h}}$ and for the residual equation we get $\mathbf{A}^h \mathrm{I}_{2h}^h \mathbf{u}^{2h} = \mathbf{r}^h$. *Figure 14* shows how $\mathbf{A}^h$ acts on $Range\left(\mathrm{I}_{2h}^h\right)$.

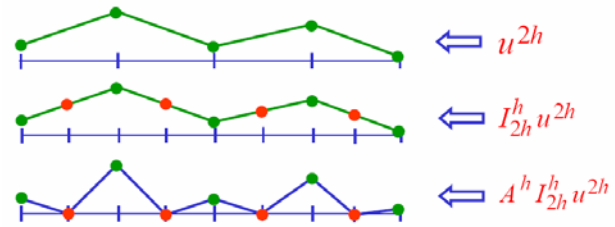

Figure 14

The values $\mathbf{A}^h \mathrm{I}_{2h}^h \mathbf{u}^{2h}$ are zero at the odd grid points of Ω^h, so the odd rows of $\mathbf{A}^h \mathrm{I}_{2h}^h$ are zero and the even rows are actually the coarse-grid points of Ω^{2h}. According to this if we leave out the odd rows in the residual equation we get its coarse-grid form. This can be done by applying the restriction operator I_h^{2h} and we get $\mathrm{I}_h^{2h} \mathbf{A}^h \mathrm{I}_{2h}^h \mathbf{u}^{2h} = \mathrm{I}_h^{2h} \mathbf{r}^h$. From here we can define the coarse-grid operator as $\mathbf{A}^{2h} = \mathrm{I}_h^{2h} \mathbf{A}^h \mathrm{I}_{2h}^h$. The same result can be obtained when using the second-order finite differences when the original problem is discretized on Ω^{2h}.

The argument that $e^h \in Range\left(\mathrm{I}_{2h}^h\right)$ does not hold in the general case, because if it holds we can immediately solve exactly the residual equation on Ω^{2h}. However, it gives an understandable definition of $\mathbf{A^{2h}}$ and the two very important *variational properties*:

- Galerkin Condition: $\mathbf{A}^{2h} = \mathrm{I}_h^{2h} \mathbf{A}^h \mathrm{I}_{2h}^h$
- $\mathrm{I}_h^{2h} = c\left(\mathrm{I}_{2h}^h\right)^T, \quad c \in ¡$

that is enough for basic understanding and an introduction to the multigrid method.

4.10 Spectral Analysis

The spectral analysis of the restriction and interpolation operator answers the question of how these two operators act on the modes of $\mathbf{A^h}$. The modes of $\mathbf{A^h}$ for the one-dimensional model problem, as we defined them in one of the previous sections, are given with $\mathrm{w}_{\mathrm{k,j}} = \sin\left(\frac{j \cdot k \cdot \pi}{n}\right),\ 1 \le k \le n-1, 0 \le j \le n$.

When the restriction operator acts upon the modes of $\mathbf{A^h}$ we get:

g smooth modes: $\mathrm{I}_h^{2h} w_k^h = \cos^2\left(\frac{k\pi}{2n}\right) w_k^{2h}, \quad 1 \le k \le \frac{n}{2}$

g oscillatory modes: $\mathrm{I}_h^{2h} w_{(n-k)}^h = -\sin^2\left(\frac{k\pi}{2n}\right) w_k^{2h}, \quad 1 \le k < \frac{n}{2}$

According to this we can conclude that using the restriction operator the oscillatory modes on Ω^h cannot be represented on Ω^{2h}. This operator transforms this modes into relatively smooth modes on Ω^{2h}. The **k**th and **(n-k)**th modes on Ω^h, both represent the **k**th mode on Ω^{2h}.

Definition 5 The pair of fine grid modes $\left\{w_k^h, w_{n-k}^h\right\}$ is called *complementary modes*. It also holds that $w_{n-k,j}^h = (-1)^{j+1} \cdot w_{k,j}^h$.

Denoting $W_k^h = span\left\{w_k^h, w_{n-k}^h\right\}$, it can be stated that $\mathrm{I}_h^{2h}: \; W_k^h \to span\left\{w_k^{2h}\right\}$.

The same analysis, but now performed on the interpolation operator I_{2h}^h gives: $\mathrm{I}_{2h}^h w_k^{2h} = \cos^2\left(\frac{k\pi}{2n}\right) w_k^h - \sin^2\left(\frac{k\pi}{2n}\right) w_{n-k}^h, \quad 1 \le k < \frac{n}{2}$, so we can conclude that interpolation of smooth modes on Ω^{2h} creates oscillatory modes on Ω^h.

For the two-grid correction scheme (TG) we have:

$$\mathbf{v}^h \leftarrow \mathbf{R}^\nu \mathbf{v}^h + C(\mathbf{f}) + \mathrm{I}_{2h}^h \left(\mathbf{A}^{2h}\right)^{-1} \mathrm{I}_h^{2h} \left(\mathbf{f}^h - \mathbf{A}^h \left(\mathbf{R}^\nu \mathbf{v}^h + C(\mathbf{f})\right)\right)$$

$$\Rightarrow \mathbf{e}^h \leftarrow \left[\mathbf{I} - \mathrm{I}_{2h}^h \left(\mathbf{A}^{2h}\right)^{-1} \mathrm{I}_h^{2h} \mathbf{A}^h\right] \mathbf{R}^\nu \mathbf{e}^h \equiv TG\mathbf{e}^h$$

As we know from before, the error can be expressed as a linear combination of the modes of $\mathbf{A^h}$. We are now interested in finding out how TG acts on the modes of $\mathbf{A^h}$. If we consider TG with no iterations ($\nu = 0$) and the made spectral analysis of the operators, we have:

$$TGw_k = s_k w_k + s_k w_{n-k}$$

$$TGw_{n-k} = c_k w_k + c_k w_{n-k}, \quad 1 \le k \le \frac{n}{2} \quad .$$

From the equations above it can be seen that TG without relaxations eliminates the smooth modes of the error and leaves the oscillatory modes undamped. If we include the relaxations, we have:

$$TGw_k = \lambda_k^\nu s_k w_k + \lambda_k^\nu s_k w_{n-k}$$

$$TGw_{n-k} = \lambda_{n-k}^\nu c_k w_k + \lambda_{n-k}^\nu c_k w_{n-k}, \quad 1 \le k \le \frac{n}{2}$$

where λ_k is the eigenvalue of the relaxation method corresponding to w_k. Observing the equations above, the fact that the relaxations show the best efficiency on the oscillatory modes and TG alone acts on the smooth modes, we can conclude that this combination will eliminate both the oscillatory and the smooth modes of the error.

4.11 Algebraic Analysis

From the properties of the interpolation and restriction operator, and the orthogonality relationships between the subspaces of a linear operator, we get: $N\left(\mathrm{I}_h^{2h}\right) \perp Range\left(\left(\mathrm{I}_h^{2h}\right)^T\right)$.

The second variational property then gives us: $N\left(\mathrm{I}_h^{2h}\right) \perp Range\left(\mathrm{I}_{2h}^h\right)$. If also the notion of A-orthogonality is used it can be obtained that: $N\left(\mathrm{I}_h^{2h}\,\mathbf{A}^h\right) \perp_{\mathbf{A^h}} Range\left(\mathrm{I}_{2h}^h\right)$. This allows us to decompose the space Ω^h in the following way: $\Omega^h = Range\left(\mathrm{I}_{2h}^h\right) \oplus N\left(\mathrm{I}_h^{2h}\,\mathbf{A}^h\right)$. Each vector $\mathbf{e^h} \in \Omega^h$, can now be represented in the form: $\mathbf{e}^h = \mathbf{s}^h + \mathbf{t}^h$, where $\mathbf{s^h} \in Range\left(I_{2h}^h\right)$ and $\mathbf{t^h} \in N\left(I_h^{2h} \cdot \mathbf{A^h}\right)$. Taking into account this decomposition and the variational properties, it can be shown that TG is the identity when it acts on $N\left(I_h^{2h}\mathbf{A^h}\right)$, and its null space is exactly $Range\left(I_{2h}^h\right)$.

Overall, we got a spectral decomposition $\Omega^h = L \oplus H$, where L contains the low-frequency modes and H contains the high-frequency modes, and an algebraic decomposition $\Omega^h = Range\left(\mathrm{I}_{2h}^h\right) \oplus N\left(\mathrm{I}_h^{2h}\,\mathbf{A}^h\right)$ of Ω^h.

4.12 How it works?

Now we are ready to see what happens behind the curtons and how the multigrid method manages to eliminate the error in a very efficient manner. The *Figure 15* gives a very good illustration of the process.

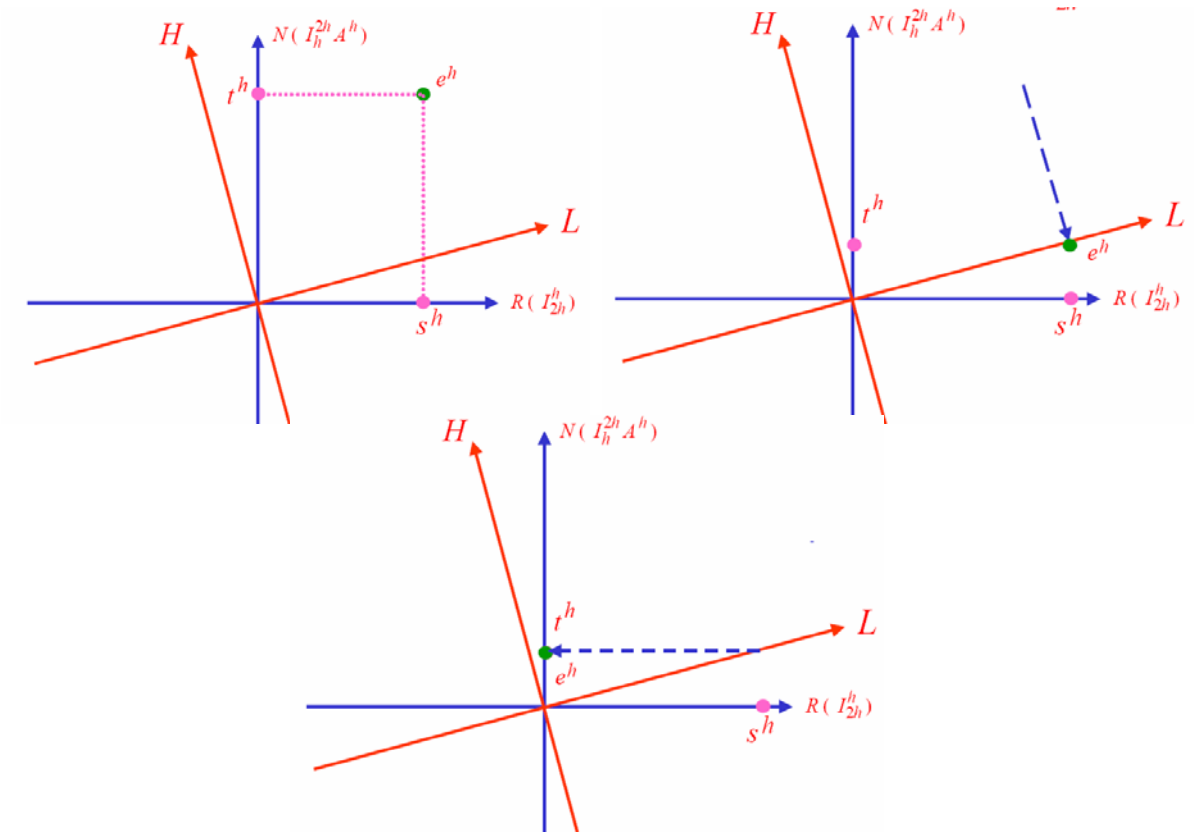

Figure 15

The axes in the figure correspond to the two previously described decompositions. The vector $\mathbf{e^h} \in \Omega^h$, can have projections on the four axes and those projections can be further projected. Analysing the pictures in *Figure 15* in the direction left to right, bottom to top, we can see how the error is efficiently damped. First the relaxation sweeps eliminate the high-frequency components of the error ($\mathbf{e^h}$ is projected onto the L-axis), then the two-grid correction scheme

eliminates the component of $\mathbf{e^h}$ along the $Range\left(I_{2h}^{h}\right)$ axis, since that is the null space of TG ($\mathbf{e^h}$ is projected onto the $N\left(I_{h}^{2h}\mathbf{A^h}\right)$-axis). The non-zero component of the error along the H-axis is because TG excites oscillatory modes. Repeating this process efficiently eliminates the error.

4. Is everything really that simple?

Although the application of the multigrid method can be very natural and gives very good results on some basic problems, that is not always the case in reality. There are many problems that introduce very difficult and not trivial problems in choosing specifying the grid, choosing the operators and so on. Some of those problems are given in the list bellow, but it is above the scope of this introduction to the multigrid method to try to solve or describe them into more details.

- Anisotropic operators and grids
- Discontinuous or anisotropic coefficients
- Nonlinear problems
- Non-scalar PDE systems
- High order discretization
- Algebraic Turbulence models
- Chemicaly reacting flows
- Shocks
- Small-scale singularities

References

William L. Briggs, Van Emden Henson, and Steve F. McCormik. A Multigrid Tutorial. SIAM, 2000

A. Brandt. Multigrid Techniques: 1984 Guide with Applications to Fluid Dynamics. The Weizmann Institute of Applied Science, Rehovot, Israel, 1984