Introduction to Multigrid

Basic Iterative Methods

System of nonlinear equations: $f(x) = 0$
Rewrite as $x = F(x)$, and iterate $x_{i+1} = F(x_i)$ (fixed-point iteration)
Converges if $\rho(\nabla F^T) < 1$ and ∇F^T Lip. cont. in neighborhood of solution

Linear system: $Ax = b$
Matrix splitting: $[P + (A - P)]x = b \Leftrightarrow Px = (P - A)x + b \Leftrightarrow$
$x = (I - P^{-1}A)x + P^{-1}b$
Iterate: $x_{i+1} = (I - P^{-1}A)x_i + P^{-1}b$
Converges if $\rho(I - P^{-1}A) < 1$

Methods: Jacobi iteration, Gauss-Seidel, (S)SOR, ...

Fixed-point: $x = (I - P^{-1}A)x + P^{-1}b \Leftrightarrow P^{-1}Ax = P^{-1}b$
Fixed-point is solution of the preconditioned system: $P^{-1}Ax = P^{-1}b$

Basic Iterative Methods

$$x_{i+1} = (I - P^{-1}A)x_i + P^{-1}b = x_i + P^{-1}b - P^{-1}Ax_i$$

Linear System: $Ax = b$ Prec. system: $P^{-1}Ax = P^{-1}b$

Residual: $r_i = b - Ax_i$ Prec. residual: $\tilde{r}_i = P^{-1}b - P^{-1}Ax_i$

$$x_{i+1} = x_i + \tilde{r}_i \quad \Rightarrow \quad x_{i+1} = x_0 + \tilde{r}_0 + \tilde{r}_1 + \cdots + \tilde{r}_i$$

Update $x_{i+1} - x_0 = \tilde{r}_0 + \tilde{r}_1 + \cdots + \tilde{r}_i$

$$\tilde{r}_{i+1} = P^{-1}b - P^{-1}Ax_{i+1} = P^{-1}b - P^{-1}Ax_i - P^{-1}A\tilde{r}_i = \tilde{r}_i - P^{-1}A\tilde{r}_i$$
$$\tilde{r}_{i+1} = (I - P^{-1}A)\tilde{r}_i \quad = \quad (I - P^{-1}A)^{i+1}\tilde{r}_0$$

$$\tilde{r}_i \in \text{span}\{\tilde{r}_0, P^{-1}A\tilde{r}_0, \ldots, (P^{-1}A)^i\tilde{r}_0\} \equiv K^{i+1}(P^{-1}A, \tilde{r}_0)$$ Krylov subspace

$$x_i - x_0 \in \text{span}\{\tilde{r}_0, \tilde{r}_1, \ldots, \tilde{r}_{i-1}\} = K^i(P^{-1}A, \tilde{r}_0)$$

Basic Iterative Methods

Solution to $Ax = b$: $\hat{x}$ Error: $e_i = \hat{x} - x_i$

Residual and error: $r_i = b - Ax_i = A\hat{x} - Ax_i = Ae_i$ $(\tilde{r}_i = P^{-1}Ae_i)$

Theorem: $\hat{x}$ is a fixed point of $x_{i+1} = (I - P^{-1}A)x_i + P^{-1}b$ iff
$\hat{x}$ is solution of $P^{-1}Ax = P^{-1}b$ $(\Leftrightarrow Ax = b)$

Proof: $x = (I - P^{-1}A)x + P^{-1}b = x - P^{-1}Ax + P^{-1}b \Leftrightarrow$
$P^{-1}Ax = P^{-1}b$

$$e_{i+1} = \hat{x} - x_{i+1} = (I - P^{-1}A)\hat{x} + P^{-1}b - (I - P^{-1}A)x_i - P^{-1}b$$
$$= (I - P^{-1}A)e_i$$

$e_{i+1} = (I - P^{-1}A)e_i = (I - P^{-1}A)^{i+1}e_0$ and $\tilde{r}_{i+1} = (I - P^{-1}A)^{i+1}\tilde{r}_0$

$$e_{i+1} \in \text{span}\left\{e_0, P^{-1}Ae_0, (P^{-1}A)^2e_0, \ldots, (P^{-1}A)^{i+1}e_0\right\}$$
$$e_{i+1} \in \text{span}\left\{e_0, \tilde{r}_0, P^{-1}A\tilde{r}_0, \ldots, (P^{-1}A)^i\tilde{r}_0\right\}$$

Review Basic Iterative Methods

Solving $Au = f$

Write $A = D - L - U$, where $-U$ is the strict upper triangular part, $-L$ is the strict lower triangular part, and D is the diagonal.

Jacobi iteration: $Du_{k+1} = (L + U)u_k + f$

Gauss-Seidel iteration: $(D - L)u_{k+1} = Uu_k + f$

Block versions possible.

Basic iteration: $u_{k+1} = P^{-1}(P - A)u_k + P^{-1}f = R_x u_k + P^{-1}f$
Convergence iff $\rho(R_x) < 1$ and asymptotic convergence rate: $\rho(R_x)$.

$R_J = D^{-1}(L + U)$
$R_{GS} = (D - L)^{-1}U$

Weighted relaxations

We can use a relaxation parameter to improve convergence:

Jacobi iteration: $\tilde{u}_k = D^{-1}(L + U)u_k + D^{-1}f$
$u_{k+1} = (1 - \omega)u_k + \omega\tilde{u}_k$
Alternatively: $u_{k+1} = u_k + \omega\tilde{r}_k$ with $\tilde{r}_k = D^{-1}(f - Au_k)$

$R_{J,\omega} = (1 - \omega)I + \omega R_J$

Gauss-Seidel iteration: $\tilde{u}_k = (D - L)^{-1}Uu_k + (D - L)^{-1}f$
$u_{k+1} = (1 - \omega)u_k + \omega\tilde{u}_k$
Alternatively: $u_{k+1} = u_k + \omega\tilde{r}_k$ with $\tilde{r}_k = (D - L)^{-1}(f - Au_k)$

$R_{GS,\omega} = (1 - \omega)I + \omega R_{GS}$

Convergence for Model Problem

Let's consider pointwise algorithm for model problem

$-u_{xx} + \sigma u = f$, for $0 < x < 1$, and $u(0) = u(1) = 0$

Equidistant grid points: $x_i = ih$ and $u_i = u(x_i)$.

Centered finite differences:
$-u_{i-1} + 2u_i - u_{i+1} = h^2 f_i$ where $i = 1 \ldots n-1$.

Jacobi iteration: $u^{(k+1)} = D^{-1}(L+U)u^{(k)} + D^{-1}f$

Update at each point: $u_i^{(k+1)} = \frac{1}{2}\left(h^2 f_i + u_{i-1}^{(k)} + u_{i+1}^{(k)}\right) = u_i^{(k)} + \frac{1}{2} r_i^{(k)}$

Update can be done for all points at once (no dependencies).

With relaxation parameter ω: $u_i^{(k+1)} = u_i^{(k)} + \frac{1}{2}\omega r_i^{(k)}$

Convergence for Model Problem

Let's consider pointwise algorithm for model problem

$-u_{xx} + \sigma u = f$, for $0 < x < 1$, and $u(0) = u(1) = 0$

Equidistant grid points: $x_i = ih$ and $u_i = u(x_i)$.

Centered finite differences:
$-u_{i-1} + 2u_i - u_{i+1} = h^2 f_i$ where $i = 1 \ldots n-1$.

Gauss-Seidel iteration: $u^{(k+1)} = (D-L)^{-1}Uu^{(k)} + (D-L)^{-1}f$

Update at each point: $u_i^{(k+1)} = \frac{1}{2}\left(h^2 f_i + u_{i-1}^{(k+1)} + u_{i+1}^{(k)}\right) = u_i^{(k)} + \frac{1}{2} r_i^{(k)}$

where $r_i^{(k)} \equiv \left(h^2 f_i + u_{i-1}^{(k+1)} + u_{i+1}^{(k)} - 2u_i^{(k)}\right)$:
the residual evaluated after updating $u_{i-1} \rightarrow u_{i-1}^{(k+1)}$

Convergence for Model Problem

Gauss-Seidel depends on ordering.
Often-used special ordering: Red-Black ordering, where red unknowns reference only black unknowns and vice versa.
In this case, order *even grid points* first and then *odd grid points*

Red/Even update: $u_{2i}^{(k+1)} = \frac{1}{2}\left(h^2 f_{2i} + u_{2i-1}^{(k)} + u_{2i+1}^{(k)}\right) = u_{2i}^{(k)} + \frac{1}{2} r_{2i}^{(k)}$

where $r_{2i}^{(k)} \equiv \left(h^2 f_{2i} + u_{2i-1}^{(k)} + u_{2i+1}^{(k)} - 2u_{2i}^{(k)}\right)$

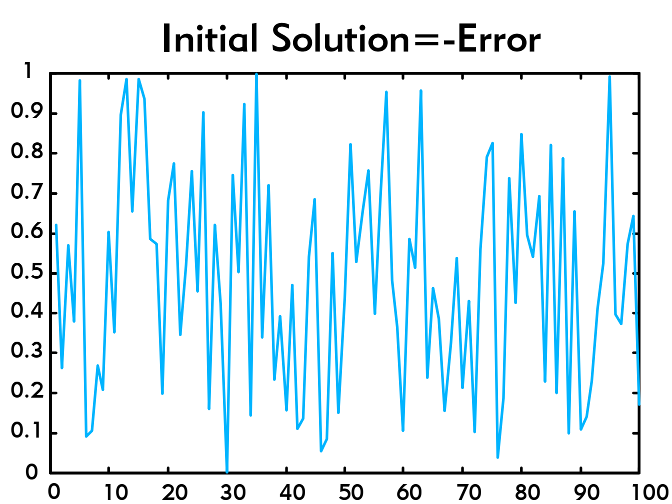
With relaxation parameter ω: $u_{2i}^{(k+1)} = u_{2i}^{(k)} + \frac{1}{2}\omega r_{2i}^{(k)}$

Black/Odd update: $u_{2i-1}^{(k+1)} = \frac{1}{2}\left(h^2 f_{2i-1} + u_{2i-2}^{(k+1)} + u_{2i+2}^{(k+1)}\right) = u_{2i-1}^{(k)} + \frac{1}{2} r_{2i-1}^{(k+1)}$

where $r_{2i-1}^{(k+1)} \equiv \left(h^2 f_{2i-1} + u_{2i-2}^{(k+1)} + u_{2i+2}^{(k+1)} - 2u_{2i-1}^{(k)}\right)$

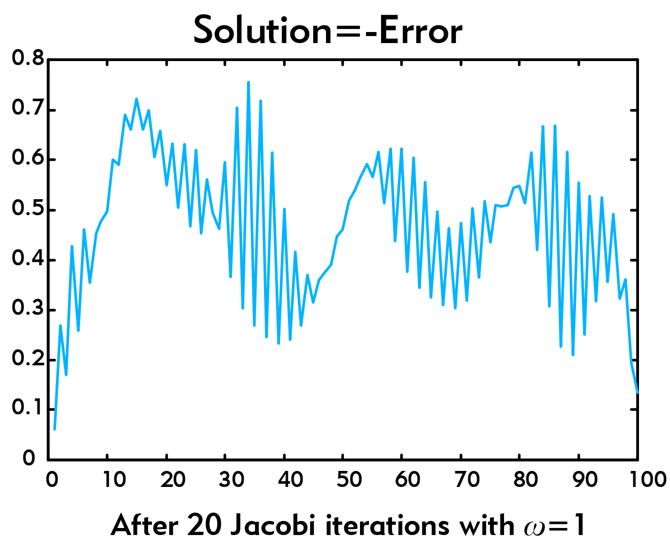
With relaxation parameter ω: $u_{2i-1}^{(k+1)} = u_{2i-1}^{(k)} + \frac{1}{2}\omega r_{2i-1}^{(k+1)}$

Error Smoothing

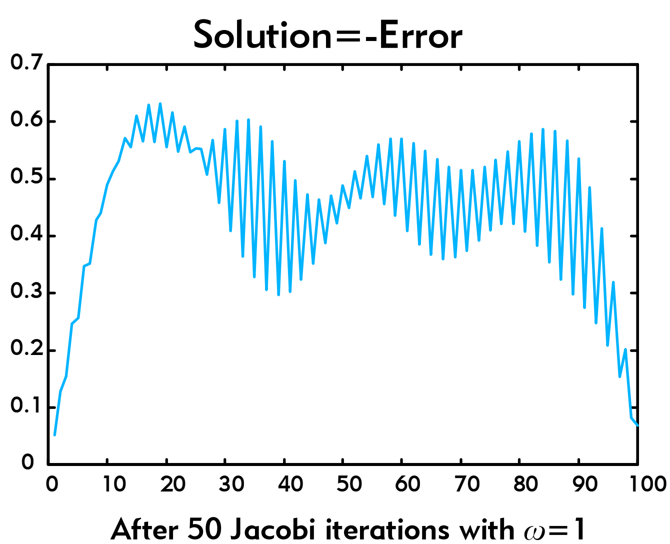


DCT: [4.9, 0.27, 0.16, 0.061, -2.4, ...], O(0.1) or O(0.2)

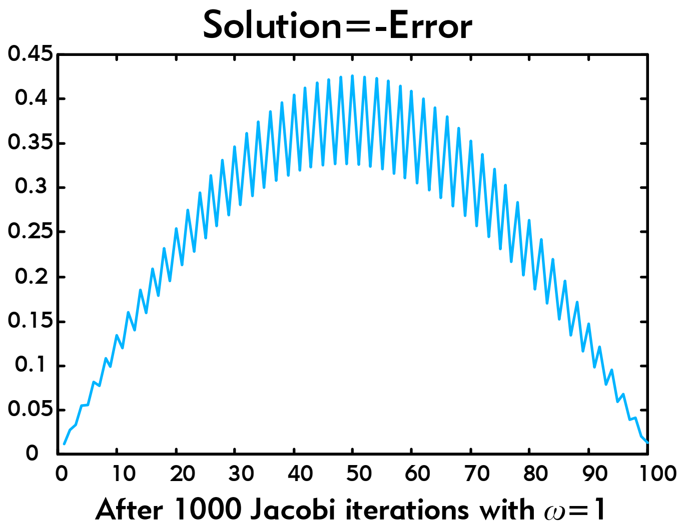
Error Smoothing



Error Smoothing

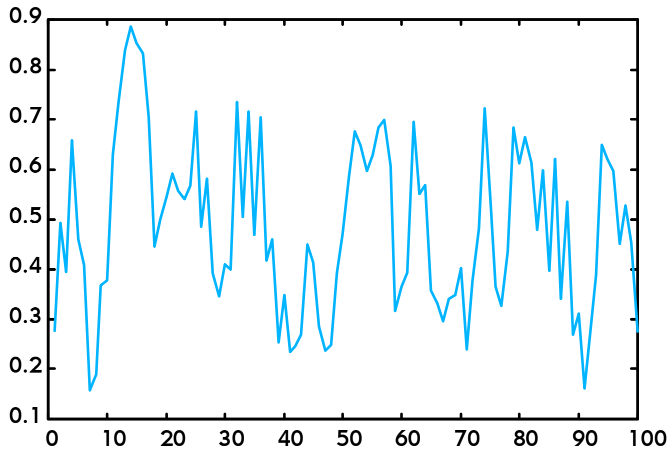


Error Smoothing

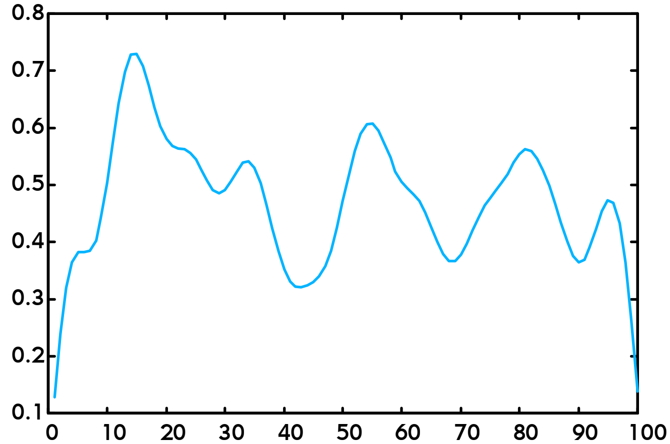


DCT: [2.4, ε, -1.1, ε, ... , ε, 0.35], ε=O(0.01), mainly O(1e-3)

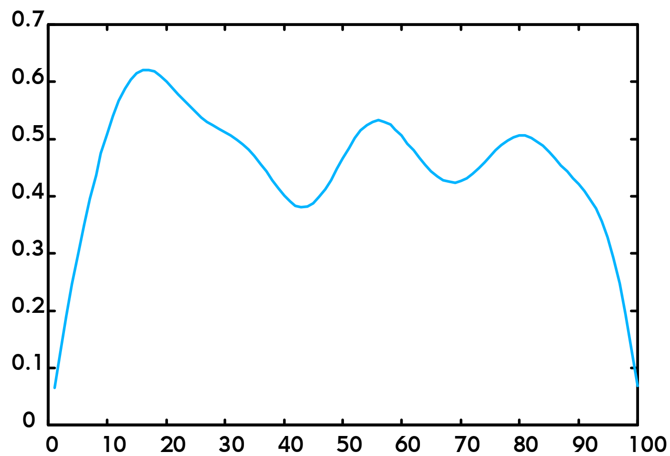
Error Smoothing



Error Smoothing

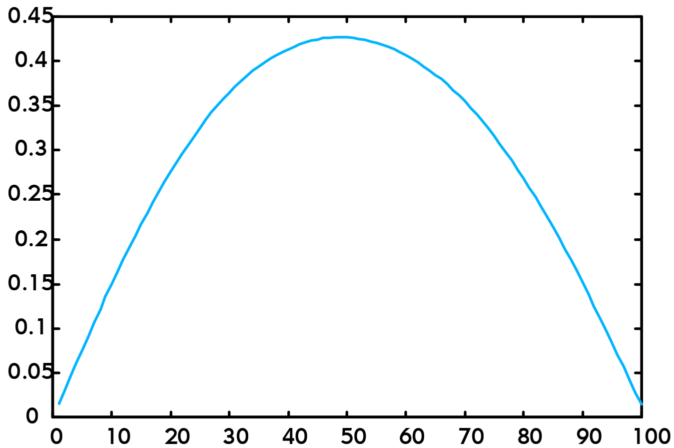


Error Smoothing



©2001 Eric de Sturler

Error Smoothing



Convergence

Convergence: $e^{(k)} = G^k e^{(0)}$, where G is iteration matrix

$G = (I - P^{-1}A)$.

$u^{(k+1)} = u^{(k)} + \omega P^{-1} r^{(k)} = u^{(k)} + \omega P^{-1}(f - Au^{(k)}) =$

$(I - \omega P^{-1}A)u^{(k)} + \omega P^{-1} f$

Jacobi iteration matrix: $(1-\omega)I + \omega R_J = (1-\omega)I + \omega D^{-1}(D - A)$

This gives: $I - \frac{\omega}{2}A$

Eigenvalues $I - \frac{\omega}{2}A$: $\lambda(I - \frac{\omega}{2}A) = 1 - \frac{\omega}{2}\lambda(A)$

Eigenvalues of A

Convergence

Eigenvalues of A:
Assume eigenvector close to physical eigenvector: $v_k = \sin\frac{\pi kj}{n}$

Apply pointwise rule for Jacobi:
$Av_{j,k} = -\sin\frac{\pi k(j-1)}{n} + 2\sin\frac{\pi kj}{n} - \sin\frac{\pi k(j+1)}{n} = 2\sin\frac{\pi kj}{n} - 2\sin\frac{\pi kj}{n}\cos\frac{\pi k}{n} =$
$2\sin\frac{\pi kj}{n}(1-\cos\frac{\pi k}{n})$ (so eigenvector indeed)

$$1-\cos\frac{\pi k}{n} = 1-\cos\frac{2\pi k}{2n} = 1-(1-2\sin^2(\frac{\pi k}{2n})) = 2\sin^2(\frac{\pi k}{2n})$$

$$2\sin\frac{\pi kj}{n}(1-\cos\frac{\pi k}{n}) = \sin\frac{\pi kj}{n} \bullet 4\sin^2(\frac{\pi k}{2n})$$

This gives for the weighted Jacobi method: $I - \frac{\omega}{2}A$

$\lambda_{(R_{J,\omega})} = 1-2\omega\sin^2(\frac{\pi k}{2n})$ where $0 < \omega \le 1$

Always converges. Poor convergence for which modes

Smoothing

Some observations:
We see from experiments and analysis that convergence for smooth and oscillatory modes is very different. More precisely, our analysis (given some ω) how much each mode is reduced in norm per iteration (sweep over all points).

Choosing appropriate ω we can make all oscillatory modes converge relatively fast. However, no choice for ω exists that makes the convergence for modes with small k fast.

We could analyze this problem (Laplacian and Jacobi) easily because eigenvectors A are the eigenvectors of iteration matrix $(I - D^{-1}A)$. This is not generally the case.

The Jacobi iteration does not mix modes. The image of a sine wave under the iteration matrix is same wave damped. Not general.

Smoothing

Some terminology:

Consider vectors $v_k = \sin\frac{jk\pi}{n}$ on 'normalized domain': $[0,1]$
where $1 \le k \le n-1$ and $0 \le j \le n$

number of grid points: n
wavenumber: k
wavelength: $l = \frac{2}{k}$ (since n grid points span domain of size 1)
This also shows that mode k gives $\frac{k}{2}$ full sine waves on domain.

We cannot represent waves on our grid with a wavelength less than $2h$
This corresponds to wavenumber larger than n. Such waves would actually be disguised as waves with longer wavelength: aliasing.

The wavenumber $k = n/2$ corresponds to wavelength $l = 4/n \approx 4h$.

Smoothing

We saw that it is important to distinguish oscillatory and smooth waves:

Low frequency/smooth if $1 \le k < n/2$
High frequency/oscillatory if $n/2 \le k \le n-1$

A particular iteration scheme (splitting) and omega give rise to the iteration
$u_{i+1} = (I - \omega P^{-1}A)u_i + \omega P^{-1}f$ which means for the error
$e_{i+1} = (I - \omega P^{-1}A)e_i = (I - \omega P^{-1}A)^i e_0$

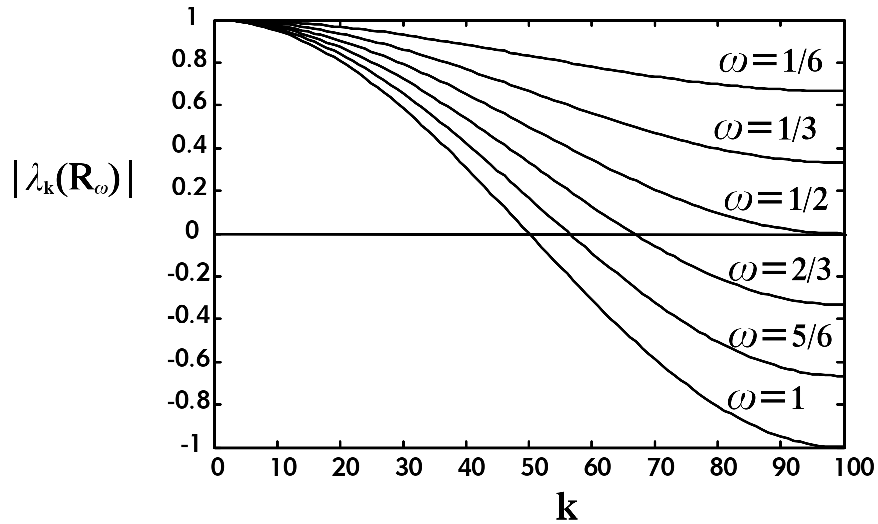
So if $(I - \omega P^{-1}A) = V\Lambda V^{-1}$ and $e_0 = V\eta_0$ then $e_i = \sum_k v_k \lambda_k^i \eta_{k,0}$

So analogous to analysis of Jacobi for Laplacian we can simplify the analysis of the convergence by considering eigenvectors separately. We can consider which ω is best for which eigenvector (we can pick only one per iteration) and if there's a best ω overall.

Smoothing

Analyze which ω best for Laplacian and Jacobi iteration:

$\lambda_{(R_{J,\omega})} = 1 - 2\omega \sin^2(\frac{\pi k}{2n})$ **where** $0 < \omega \leq 1$



Smoothing

Standard Jacobi $\omega = 1$ works for the middle range wavenumbers, but does poorly for both low and high frequency waves.

The choice $\omega = 2/3$ does well for a fairly large range of wavenumbers

No ω does well for the lowest frequencies.

In fact $k = 1 : 1 - 2\omega \sin^2(\frac{\pi}{2n}) \approx 1 - 2\omega \sin^2(\frac{h\pi}{2}) \approx 1 - \frac{\omega h^2 \pi^2}{2}$

So for small h there will be no $\omega \leq 1$ that will make $|\lambda_1(R_\omega)|$ small

Worse, for as $h \to 0$ (solving problem more accurately) the reduction factor becomes closer and closer to 1.

Smoothing

Let's give up on the low frequencies and focus on high frequencies. (or at least postpone elimination of low frequency error)

oscillatory modes: $n/2 \le k \le n-1$
We see that several values for ω do well. To get 'the best' we require again the equi-oscillating one (remember Chebyshev polynomials):

$$\lambda_{n/2}(R_\omega) = -\lambda_n(R_\omega)$$

This gives
$$\begin{aligned}
&1 - 2\omega\sin^2\left(\tfrac{(n/2)\pi}{2n}\right) = -1 + 2\omega\sin^2\left(\tfrac{n\pi}{2n}\right) \Leftrightarrow \\
&1 - 2\omega\sin^2\left(\tfrac{\pi}{4}\right) = -1 + 2\omega\sin^2\left(\tfrac{\pi}{2}\right) \Leftrightarrow \\
&1 - \omega = -1 + 2\omega \Leftrightarrow \\
&2 = 3\omega \Leftrightarrow \\
&\omega = \tfrac{2}{3}
\end{aligned}$$

Worst convergence factor attained at $k = \frac{2}{3} : 1 - \frac{4}{3}\sin^2\left(\frac{\pi}{4}\right) = 1 - \frac{4}{6} = \frac{1}{3}$.

Smoothing

So $|\lambda_k| \le 1/3$ and oscillatory components are reduced by at least a factor 3 each iteration.

This factor is called the smoothing factor.
From its derivation we see that it is independent of h.

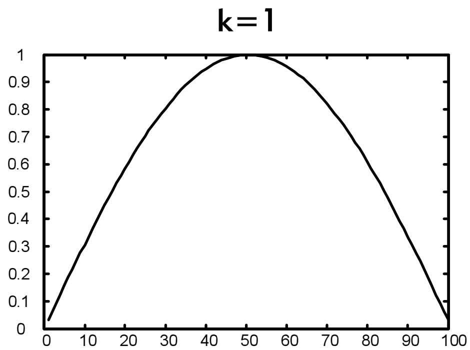
Suppose we wanted to reduce smooth modes by at least factor $1/2$.
$$1 - 2\omega\sin^2\left(\tfrac{\pi h}{2}\right) \approx 1 - 2\omega\tfrac{\pi^2 h^2}{4} = 1 - \tfrac{\omega\pi^2 h^2}{2} \to \omega\pi^2 h^2 = 1 \Rightarrow \omega = \tfrac{1}{\pi^2 h^2}$$

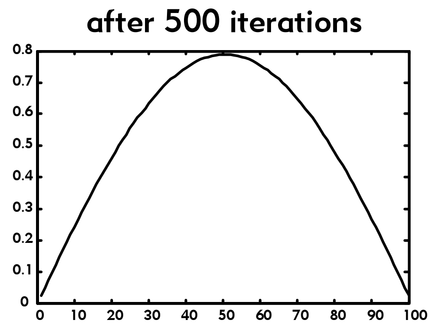
Then for $k = n/2$ we get:

$$1 - \tfrac{2}{\pi^2 h^2}\sin^2\left(\tfrac{\pi}{4}\right) = 1 - \tfrac{1}{\pi^2 h^2}$$

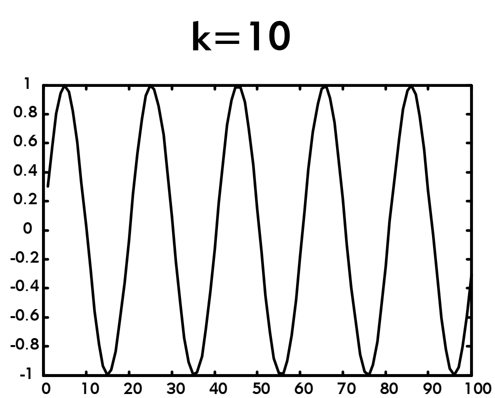
So for $h < \frac{1}{\pi\sqrt{2}}$ we will amplify the oscillating modes! Divergence.

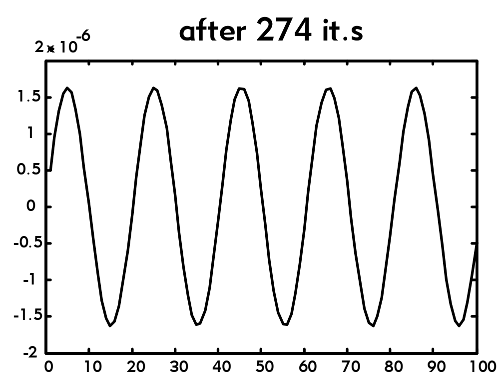
Smoothing Experiments



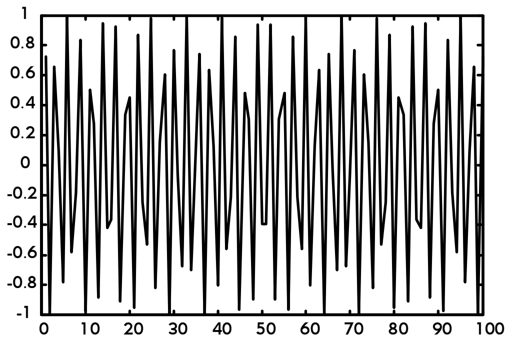


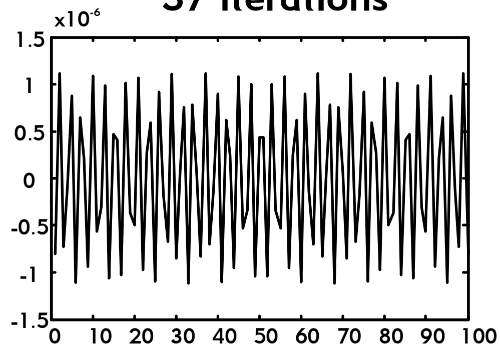
Smoothing Experiments



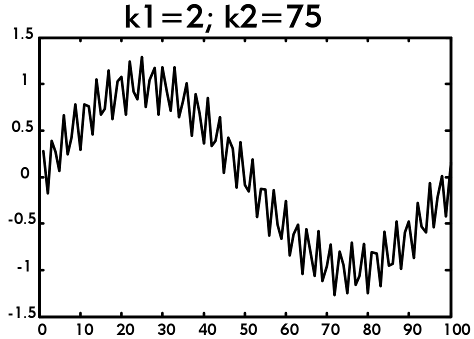


Smoothing Experiments



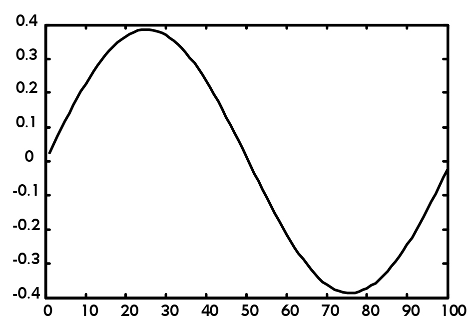


Smoothing Experiments



after 500 iterations

coarse part removed in a few iterations



Smoothing Experiments