Today’s Topics

- Variational formulation for PDEs
- Projection into basis
  - Analytical basis
  - Finite-element basis
- If time permits
  - A posteriori error analysis & adaptive methods
  - Numerical integration

Modern PDE theory briefly

- Usually the equation $-\Delta u = f$ is cast into weak form
- Multiply by a fucntion *v* and integrate over domain $\Rightarrow$ $-\int_V \Delta u v \, d\mathbf{r} = \int_V f v \, d\mathbf{r}$
- Use Gauss-Green formula on $\Delta u = \nabla \cdot \nabla u$

$$\Rightarrow \quad -\int_V \Delta u v \, d\mathbf{r} = \int_V \nabla u \cdot \nabla v \, d\mathbf{r} - \int_{\partial V} \frac{\partial u}{\partial n} v \, dS$$

$\int_{\partial V_N} g v \, dS$ on Neumann boundary where $\frac{\partial u}{\partial n} = g$

0 on Dirichlet boundary where $v = u = 0$

$$\Rightarrow \quad \boxed{\int_V \nabla u \cdot \nabla v \, d\mathbf{r} = \int_V f v \, d\mathbf{r} + \int_{\partial V_N} g v \, dS}$$

Weak or variational formulation

Goodies:

- Only gradient needed
- Symmetric in *u* and *v*
- Boundary conditions included

Expansion in a basis

- FD is just fine in many cases
- However, many of the ”Cons” can be removed by using a basis expansion
- Write: $u(x) \approx u_h(x) = \sum_{i=1}^{M} \alpha_i \phi_i(x)$
- Plug in: $-u_h''(x) = -\sum_{i=1}^{M} \alpha_i \phi_i''(x) = f(x)$
- Need to solve for α_i – how?

1. Require at points x_j: $-\sum_{i=1}^{M} \alpha_i \phi_i''(x_j) = f(x_j)$ (collocation)
2. Multiply by ϕ_j and integrate (variational formulation):

$$-\sum_{i=1}^{M} \alpha_i \int_0^L \phi_i''(x)\phi_j(x)\,dx = \int_0^L f(x)\phi_j(x)\,dx$$

Expansion in a basis: Variational formulation

- Integration by parts is invoked: $-\int_0^L \phi_i''(x)\phi_j(x)\,dx = \int_0^L \phi_i'(x)\phi_j'(x)\,dx$

Assume homogeneous boundary conditions

$$\left.\begin{aligned} -\sum_{i=1}^{M} \alpha_i \int_0^L \phi_i''(x)\phi_j(x)\,dx &= \int_0^L f(x)\phi_j(x)\,dx \\ \Downarrow \\ \sum_{i=1}^{M} \left(\int_0^L \phi_i'(x)\phi_j'(x)\,dx \right) \alpha_i &= \int_0^L f(x)\phi_j(x)\,dx \end{aligned}\right\}$$

$$A\alpha = b$$

where

$$\left\{\begin{aligned} &A = (a_{ji})_{i,j=1}^{M}, \quad a_{ji} = \int_0^L \phi_i'(x)\phi_j'(x)\,dx \\ &b = (b_j)_{j=1}^{M}, \quad b_j = \int_0^L f(x)\phi_j(x)\,dx \\ &\alpha = (\alpha_i)_{i=1}^{M} \end{aligned}\right.$$

Discuss: What kind of basis functions could / should be used?

Expansion in a basis: Variational formulation

- Need to solve: $A\alpha = b$ where
- Let's explore basis options:
  1. Spectral basis:

$$A = (a_{ji})_{i,j=1}^{M}, \quad a_{ji} = \int_0^L \phi_i'(x)\phi_j'(x)\,dx$$

$$b = (b_j)_{j=1}^{M}, \quad b_j = \int_0^L f(x)\phi_j(x)\,dx$$

$$\alpha = (\alpha_i)_{i=1}^{M}$$

$$\phi_j(x) = \sqrt{\frac{2}{L}}\sin\left(\frac{j\pi}{L}x\right)$$

$$\Downarrow$$

A is diagonal with $a_{jj} = j^2\pi^2$

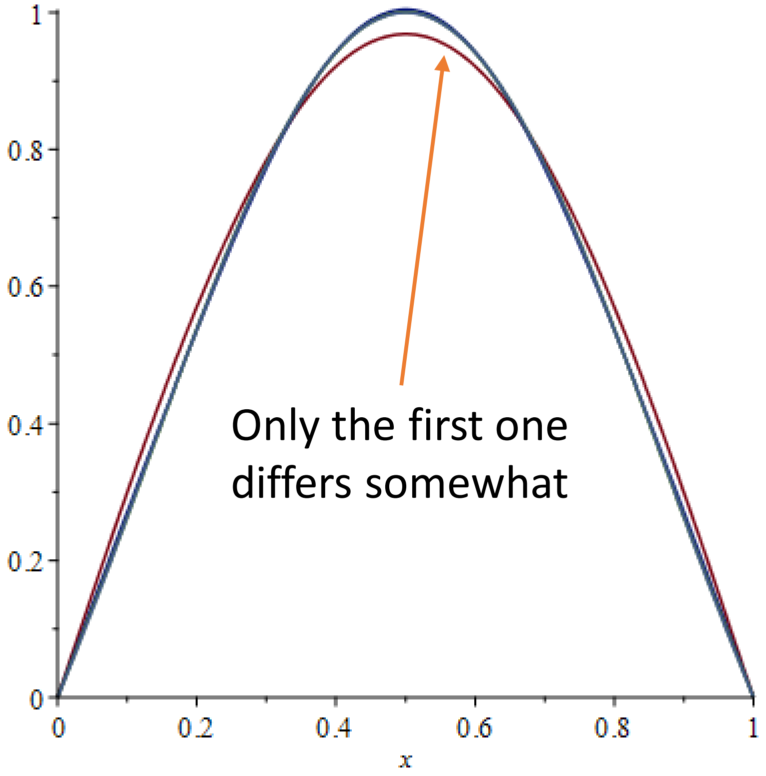
b is essentially Fourier sine-series coefficients of *f*: $b_j = \sqrt{\frac{2}{L}}\int_0^L f(x)\sin\left(\frac{j\pi}{L}x\right)dx$

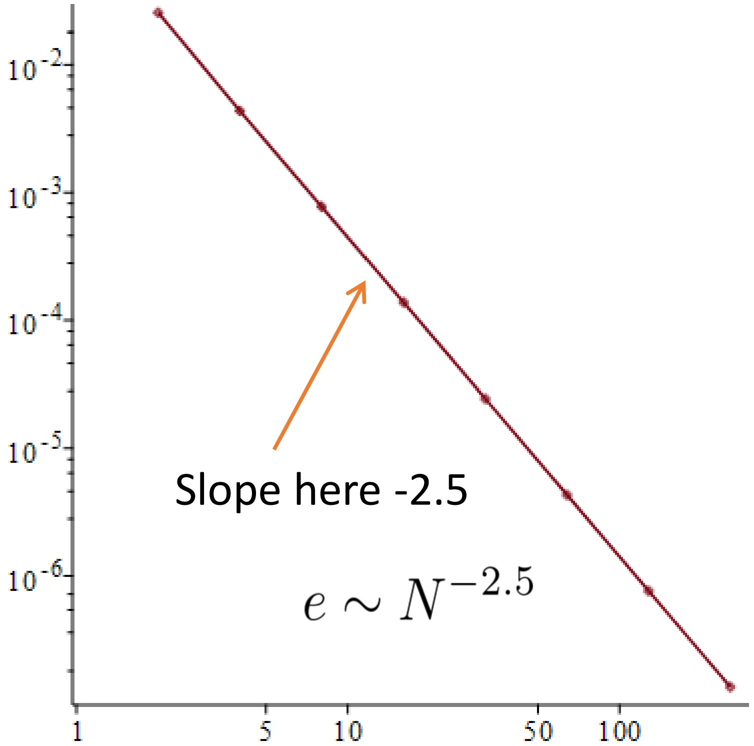
Convergence in Spectral Basis

Easy case: $u(x) = \sin\left(\frac{\pi x}{L}\right) \exp\left(-(x - L/2)^2\right)$

$N = 2, 4, 8, \ldots, 256$

Error as: $e = \sqrt{\int_0^1 \left(u(x) - u_h(x)\right)^2 \, dx}$



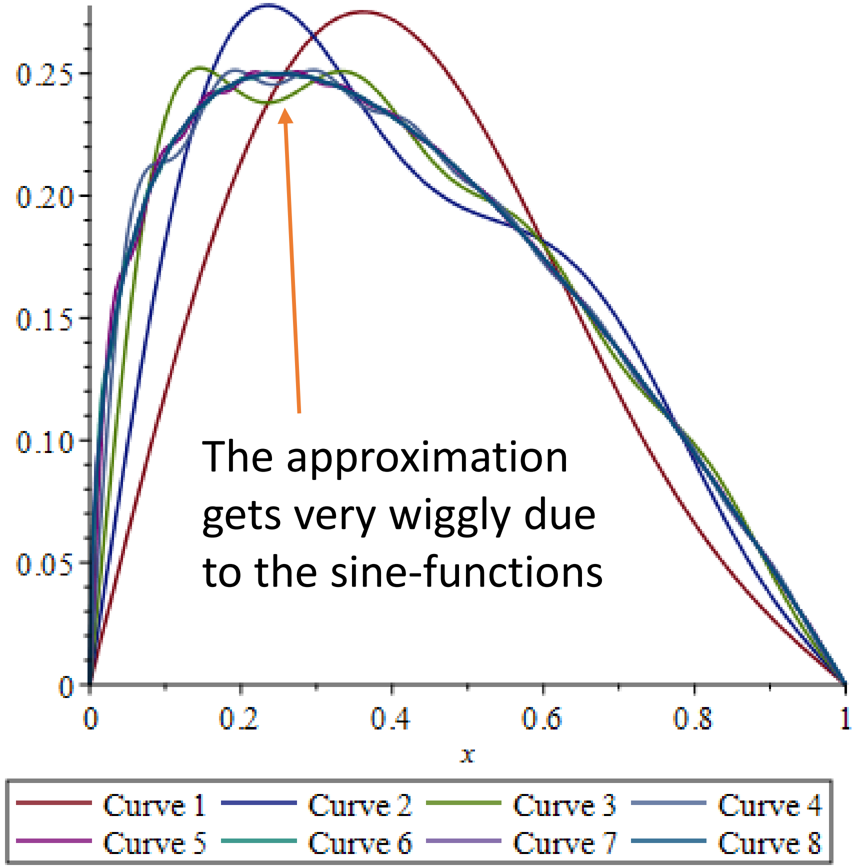


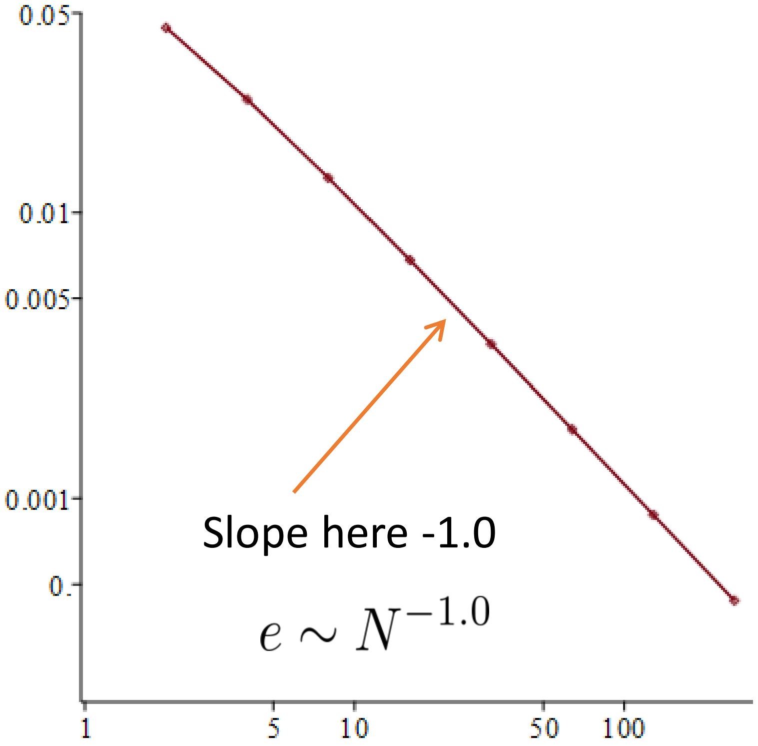
Convergence in spectral basis

Tougher case: $u(x) = \sqrt{x} - x$

$$N = 2, 4, 8, \ldots, 256$$

Error as: $e = \sqrt{\int_0^1 \left(u(x) - u_h(x)\right)^2 \, dx}$





Expansion in a basis: Variational formulation

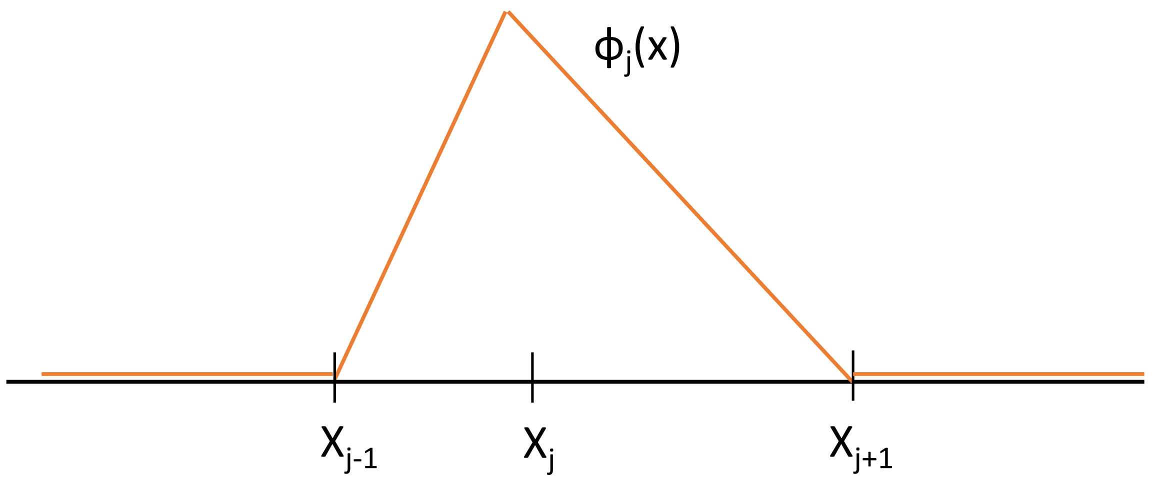
- Need to solve: $A\alpha = b$ where
- Let's explore basis options:
  1. Spectral basis:

$$\phi_j(x) = \sqrt{\frac{2}{L}} \sin\left(\frac{j\pi}{L} x\right)$$

$$\begin{cases} A = (a_{ji})_{i,j=1}^{M}, \quad a_{ji} = \int_0^L \phi_i'(x)\phi_j'(x)\,dx \\ b = (b_j)_{j=1}^{M}, \quad b_j = \int_0^L f(x)\phi_j(x)\,dx \\ \alpha = (\alpha_i)_{i=1}^{M} \end{cases}$$

  2. Piecewise polynomial basis:
     - Polynomial: easy to integrate & flexible
     - Piecewise: sparse matrices

$$\phi_j(x) = \begin{cases} \frac{x - x_{j-1}}{x_j - x_{j-1}}, & x_{j-1} < x < x_j \\ \frac{x_{j+1} - x}{x_{j+1} - x_j}, & x_j < x < x_{j+1} \\ 0, & \text{otherwise} \end{cases} \quad \Longrightarrow \quad \phi_j(x_i) = \delta_{ij}$$

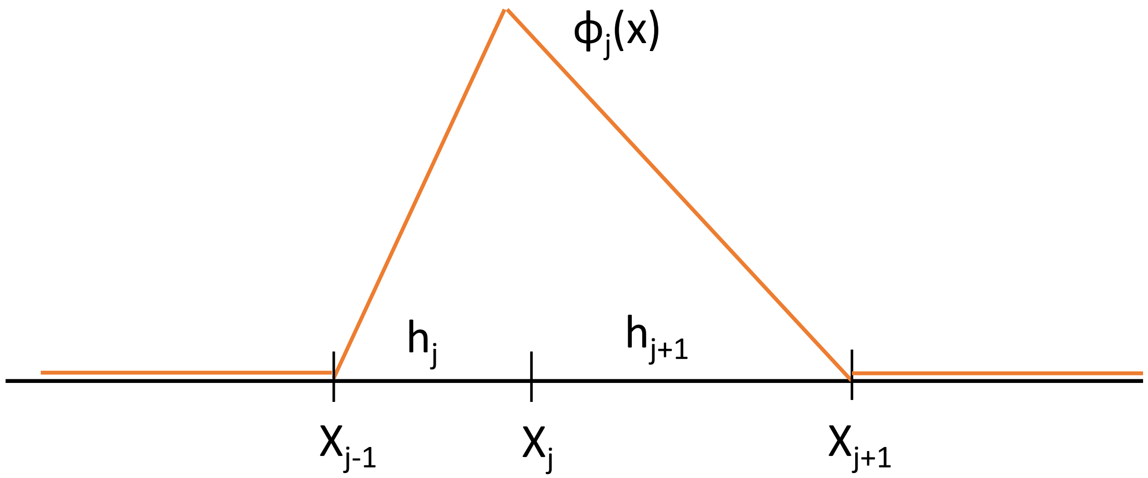


Expansion in a Piecewise Polynomial Basis

$$\phi_j(x) = \begin{cases} \frac{x - x_{j-1}}{x_j - x_{j-1}}, & x_{j-1} < x < x_j \\ \frac{x_{j+1} - x}{x_{j+1} - x_j}, & x_j < x < x_{j+1} \\ 0, & \text{otherwise} \end{cases}$$

$$A = (a_{ji})_{i,j=1}^{M}, \quad a_{ji} = \int_0^L \phi_i'(x)\phi_j'(x)\,dx$$

$$\left\{ \begin{array}{l} a_{jj} = \dfrac{1}{h_j} + \dfrac{1}{h_{j+1}} \\ a_{j,j+1} = -\dfrac{1}{h_{j+1}} \quad a_{j-1,j} = -\dfrac{1}{h_j} \end{array} \right.$$



For future refecence, let’s calculate also

$$\boxed{s_{ji} = \int_0^L \phi_i(x)\phi_j(x)\,dx}$$

$$\left\{ \begin{array}{l} s_{jj} = \dfrac{1}{3}(h_j + h_{j+1}) \\ s_{j-1,j} = \dfrac{1}{6}h_j \\ s_{j,j+1} = \dfrac{1}{6}h_{j+1} \end{array} \right.$$

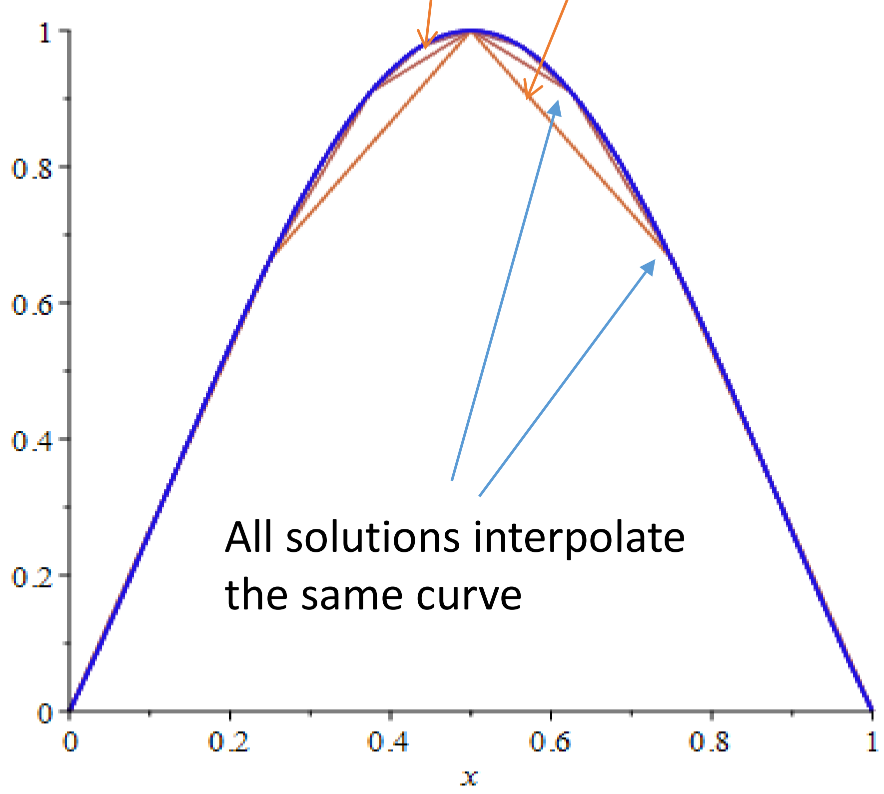
Solving the Poisson equation with FEM: 1D

Easy case: $u(x) = \sin\left(\frac{\pi x}{L}\right) \exp\left(-(x - L/2)^2\right)$

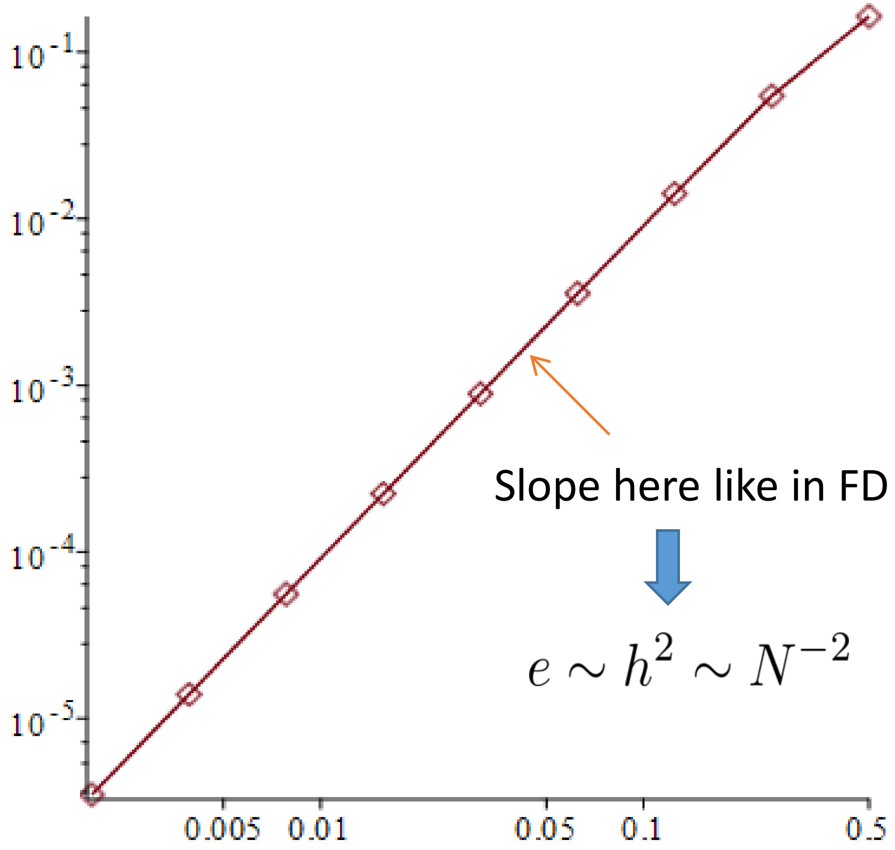
$N = 2^k, \quad k = 2 \ldots 9$

Error as: $e = \sqrt{\int_0^1 (u(x) - u_h(x))^2 \, dx}$

Only the sparsest grids differs noticeably



All solutions interpolate the same curve



Slope here like in FD = 2

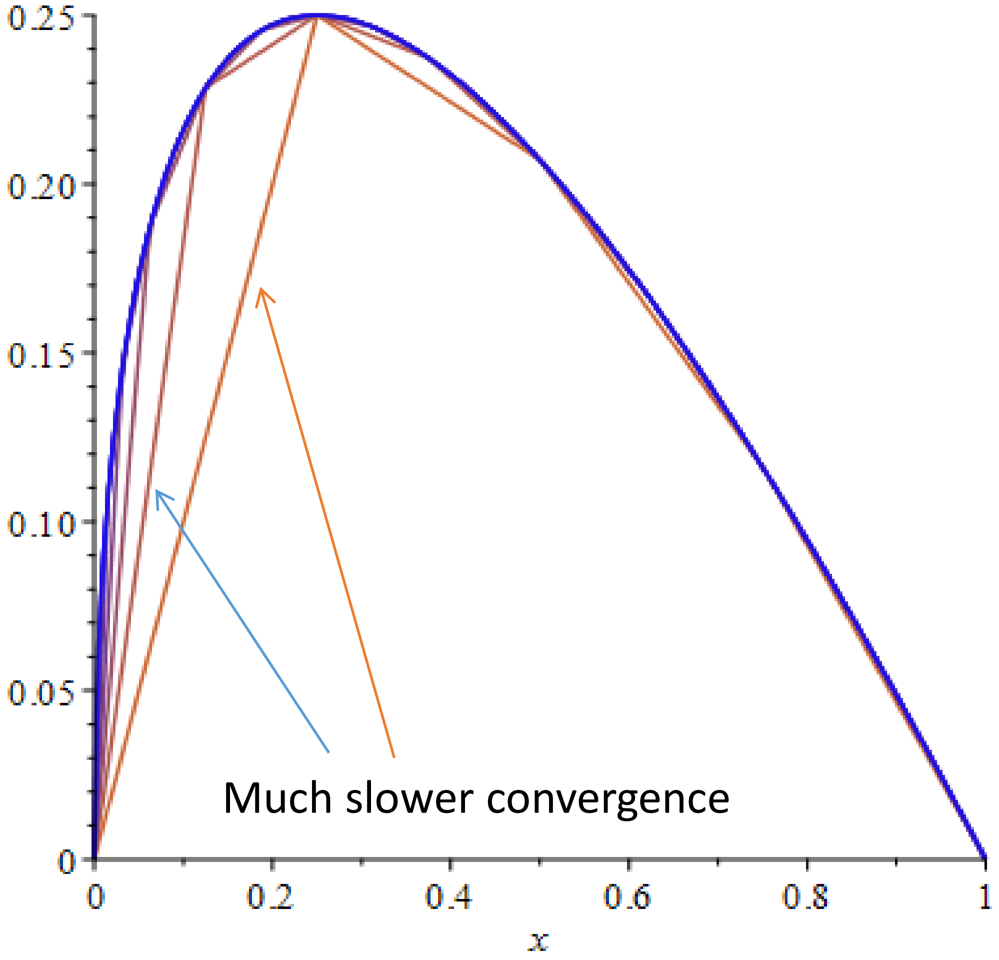
$e \sim h^2 \sim N^{-2}$

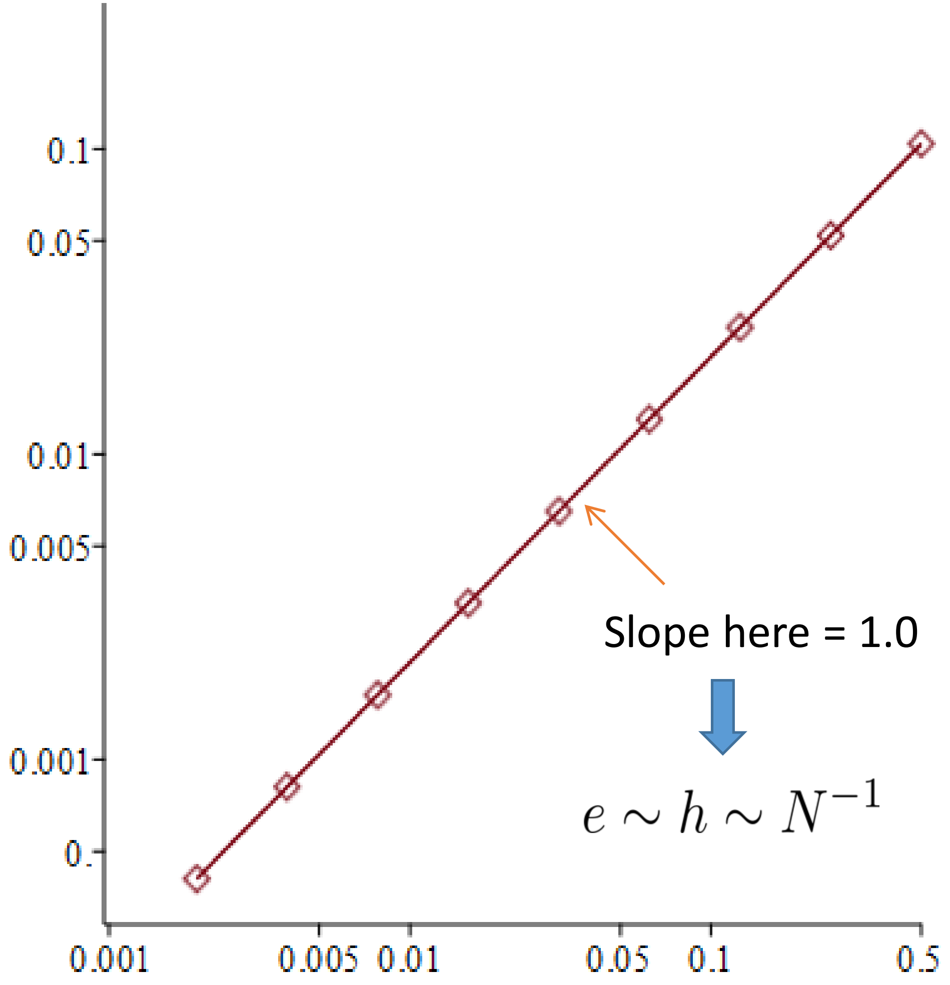
Solving the Poisson equation: 1D

Tougher case: $u(x) = \sqrt{x} - x$

$$N = 2^k, \quad k = 2 \ldots 9$$

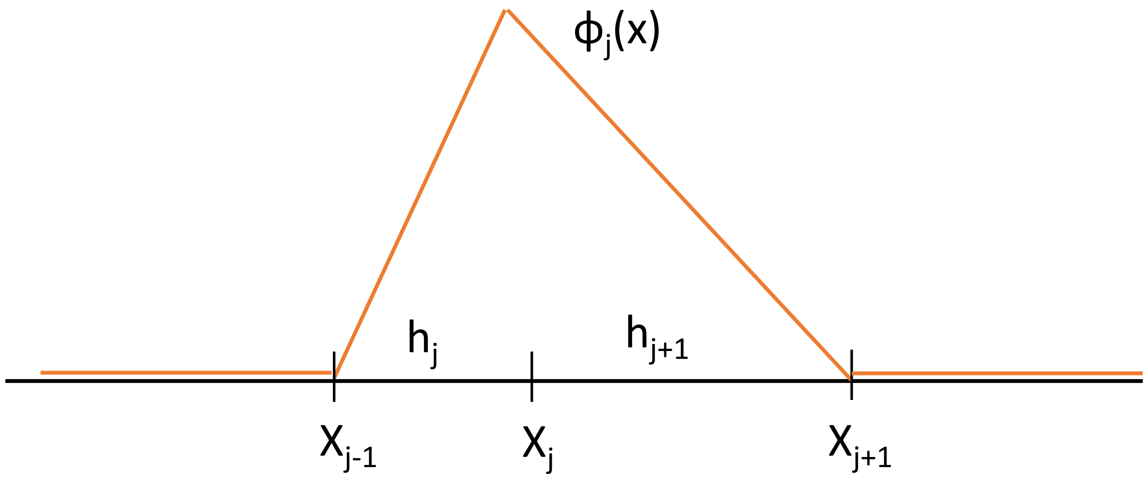
Error as: $e = \sqrt{\int_0^1 (u(x) - u_h(x))^2 \, dx}$





Basis Function Expansion

$$\phi_j(x) = \sqrt{\frac{2}{L}} \sin\left(\frac{j\pi}{L} x\right)$$



Discuss: First impressions of the basis function expansion

- Can you implement such a scheme?
- What would you do in 2D? Or 3D?

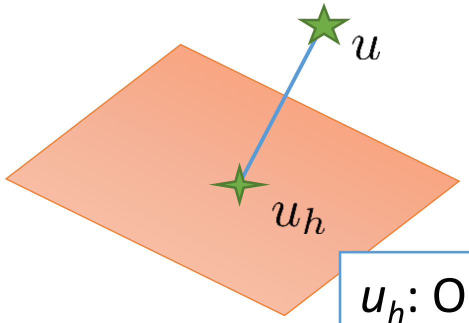
Convergence Theory in One Slide

- Define $a(u,v) = \int_0^L u'(x)v'(x)\,dx$ and $(f,v) = \int_0^L f(x)v(x)\,dx$

- The exact solution satisfies $a(u,v) = (f,v) \quad \forall v$
- The variational solution satisfies $a(u_h,v_h) = (f,v_h) \quad \forall v_h = \sum_j \beta_j \phi_j(x)$

$\Rightarrow$ $a(u-u_h,v_h) = 0 \quad \forall v_h$

$\Rightarrow$ $a(u-u_h,u-u_h) = a(u-u_h,u-u_h) + a(u-u_h,u_h-v_h) = a(u-u_h,u-v_h)$



$$||u-u_h||_a^2 \leq ||u-u_h||_a ||u-v_h||_a \Rightarrow ||u-u_h||_a \leq \min_{v_h} ||u-v_h||_a$$

u_h: Orthogonal projection of u to the space spanned by the basis functions ϕ_j

Analytic basis for 1D Schrödinger

(Typically orthonormal, complete basis, here general)

Expanding ψ using basis $\{\phi_i(\mathbf{r})\}_{i=1}^{M}$ as $\psi = \sum_{i=1}^{M} \alpha_i \phi_i$.

Plug to $\mathcal{H}\psi = E\psi$, and get

$$\mathcal{H}\sum_{i=1}^{M} \alpha_i \phi_i = E \sum_{i=1}^{M} \alpha_i \phi_i$$

which is

$$\sum_{i=1}^{M} \alpha_i \mathcal{H} \phi_i = E \sum_{i=1}^{M} \alpha_i \phi_i$$

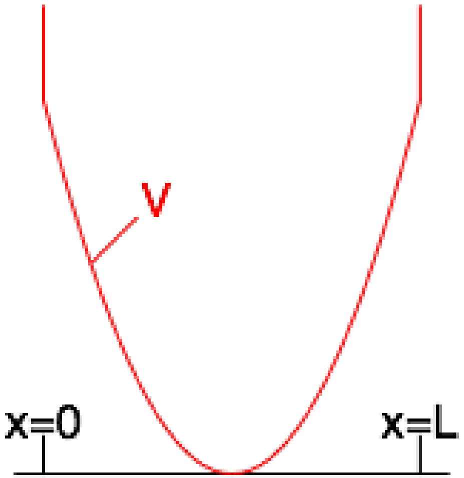
multiply by ϕ_j and integrate:

$$\sum_{i=1}^{M} \alpha_i \langle \phi_j | \mathcal{H} | \phi_i \rangle = E \sum_{i=1}^{M} \alpha_i \langle \phi_j | \phi_i \rangle \qquad \forall\, j$$

or in matrix form (H and S are Hamiltonian and overlap matrices): $H\boldsymbol{\alpha} = ES\boldsymbol{\alpha}$

Matrix elements: $h_{ji} = \langle \phi_j | \mathcal{H} | \phi_i \rangle$ and $s_{ji} = \langle \phi_j | \phi_i \rangle$ are d-dimensional integrals.

Analytic basis for 1D Schrödinger



Particle in a box part:

$$\mathcal{H}_0\,\phi_i = E_i\,\phi_i$$

$$\phi_i = \sin(i\tfrac{\pi x}{L})\sqrt{\tfrac{2}{L}} \qquad E_i = \frac{\pi^2 i^2}{2L^2}$$

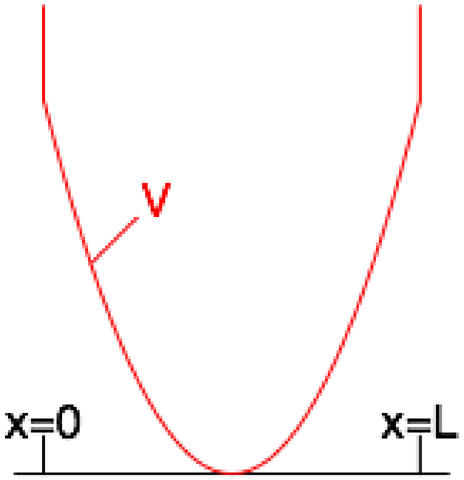
Take $\mathcal{H}_I = \frac{1}{2}(x - L/2)^2$ as additional term in $\mathcal{H} = \mathcal{H}_0 + \omega^2\mathcal{H}_I$

$$s_{ij} = \delta_{i,j}.$$

$$h_{ij} = \delta_{i,j}E_i + \omega^2 h^I_{ij}$$

$$h^I_{ij} = \frac{2}{L}\int \sin\left(i\frac{\pi x}{L}\right)\mathcal{H}_I \sin\left(j\frac{\pi x}{L}\right)$$ doable

Analytic basis for 1D Schrödinger



Particle in a box part: $\phi_i = \sin(i\frac{\pi x}{L})\sqrt{\frac{2}{L}}$

$$\mathcal{H}_0\,\phi_i = E_i\,\phi_i \qquad E_i = \frac{\pi^2 i^2}{2L^2}$$

```
psi[i_, x_] := Sin[i Pi x / L] Sqrt[2 / L]

S[l_, m_] := Integrate[psi[l, x] psi[m, x], {x, 0, L}]

Simplify[S[l, l], Assumptions → {l ∈ Integers}]
1
```

OK

```
Simplify[S[l, m], Assumptions → {{l, m} ∈ Integers}]
0
```

?, we have to be careful (l can still be m)

```
V[l_, m_] := Integrate[psi[l, x] 1 / 2 (x - L / 2) ^2 psi[m, x],
  {x, 0, L}, Assumptions → {{l, m} ∈ Integers}]

FullSimplify[Limit[V[l, l], L → Pi],
 Assumptions → {{l, m} ∈ Integers}]
```

$$\frac{1}{24}\left(-\frac{6}{l^2}+\pi^2\right)$$

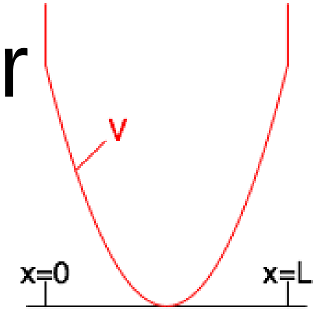
```
FullSimplify[Limit[V[l, m], L → Pi],
 Assumptions → {{l, m} ∈ Integers}]
```

$$\frac{2\left(1+(-1)^{l+m}\right) l\, m}{\left(l^2-m^2\right)^2}$$

$$h^I_{ij} = \frac{2}{L}\int \sin\left(i\frac{\pi x}{L}\right)\mathcal{H}_I \sin\left(j\frac{\pi x}{L}\right)$$

$$\mathcal{H}_I = \frac{1}{2}(x - L/2)^2$$

Analytic basis for 1D Schrödinger



$$h_{ij} = \delta_{i,j} E_i + \omega^2 h^I_{ij}$$

Mathematica

Diagonal and off-diagonal defined:

```
H[omega_, m_, l_] :=
 N[Piecewise[{{omega^2 * 1/24 (-6/l^2 + π^2) + l^2/2, l == m}}, omega^2 * (2 (1 + (-1)^(l+m)) l m)/(l^2 - m^2)^2]]
```

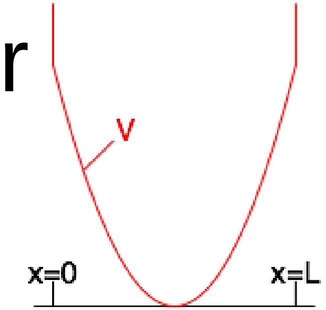
Matlab

```
function [ H ] = HO_BOX_Ham_matrix( omega, N )
% Harmonic potential inside box.
% Omega is the confinement strength and
% N is the number of basis functions.
% This function return the Hamiltonian matrix.

H=zeros(N,N);
for l=1:N,
   H(l,l)=l^2/2 + omega^2*1/24*(-(6/l^2) + pi^2);
  for m=l+1:N,
     H(l,m)=omega^2*(2*(1 + (-1)^(l + m))*l*m)/(l^2 - m^2)^2;
     H(m,l)=H(l,m);
  end
end
end
```

The matrix turns out to be small in this case, let's use
[V, D]=eig(H);
D(i,i) should be same as V(:,i)'*H*V(:,i)./(V(:,i)'*V(:,i))

Analytic basis for 1D Schrödinger



$$h_{ij} = \delta_{i,j} E_i + \omega^2 h^I_{ij}$$

diagonalization for ω=1, converges with a small basis:

Basis 1:	0.6612335167				
Basis 2:	0.6612335167	2.3487335167			
Basis 3:	0.6529233923	2.3487335167	4.8917658633		
Basis 4:	0.6529233923	2.3405778658	4.8917658633	8.4037641676	
Basis 5:	0.6528750885	2.3405778658	4.8848369028	8.4037641676	12.9082107811
Basis 6:	0.6528750885	2.3404877248	4.8848369028	8.3979330006	12.9082107811
Basis 7:	0.6528714914	2.3404877248	4.8847320051	8.3979330006	12.9032293222
Basis 8:	0.6528714914	2.3404800868	4.8847320051	8.3978278411	12.9032293222
Basis 9:	0.6528709903	2.3404800868	4.8847221866	8.3978278411	12.9031290968
Basis 10:	0.6528709903	2.3404788906	4.8847221866	8.3978173000	12.9031290968
Basis 20:	0.6528708466	2.3404785049	4.8847199326	8.3978146862	12.9031158108
Basis 30:	0.6528708454	2.3404785010	4.8847199181	8.3978146661	12.9031157605
Basis 40:	0.6528708454	2.3404785007	4.8847199173	8.3978146650	12.9031157580
Basis 50:	0.6528708453	2.3404785007	4.8847199172	8.3978146648	12.9031157578
Basis 60:	0.6528708453	2.3404785007	4.8847199172	8.3978146648	12.9031157577
Basis 70:	0.6528708453	2.3404785007	4.8847199172	8.3978146648	12.9031157577
Basis 80:	0.6528708453	2.3404785007	4.8847199172	8.3978146648	12.9031157577
Basis 90:	0.6528708453	2.3404785007	4.8847199172	8.3978146648	12.9031157577
Basis 100:	0.6528708453	2.3404785007	4.8847199172	8.3978146648	12.9031157577
Basis 500:	0.6528708453	2.3404785007	4.8847199172	8.3978146648	12.9031157577
Basis 1000:	0.6528708453	2.3404785007	4.8847199172	8.3978146647	12.9031157577
vs 1000:	0.6528708453	2.3404785007	4.8847199172	8.3978146648	12.9031157577

Last line from `V(:,i)'*H*V(:,i)./(V(:,i)'*V(:,i))`

Compare with finite difference with 1000 points:

Analytic basis

0.6528708453	2.3404785007	4.8847199172	8.3978146648	12.9031157577
0.6528703697	2.3404717988	4.8846865399	8.3977094753	12.9028591328

Finite difference

Analytic basis much more accurate even with a small basis size!

Analytic basis for 1D Schrödinger

$$\mathcal{H} = \mathcal{H}_0 + \omega^2 \frac{1}{2} \left(x - L/2\right)^2$$

Diagonalization for ω=10

```
Basis    1: 16.623351671
Basis    2: 16.623351671 36.8733516712
Basis    3: 6.8551329283 36.8733516712 52.6137926363
Basis    4: 6.8551329283 19.3614060013 52.6137926363 65.0727973411
Basis    5: 5.2241379901 19.3614060013 31.4173098668 65.0727973411 75.4508293789
Basis    6: 5.2241379901 15.6076040208 31.4173098668 42.6678363893 75.4508293789
Basis    7: 5.0133343304 15.6076040208 25.9790900916 42.6678363893 53.1400271119
Basis    8: 5.0133343304 15.0393896812 25.9790900916 36.2225504138 53.1400271119
Basis    9: 5.0003378098 15.0393896812 25.0654194770 36.2225504138 46.3018057280
Basis   10: 5.0003378098 15.0009936360 25.0654194770 35.0800066988 46.3018057280
Basis   20: 5.0000000011 15.0000000498 25.0000011241 35.0000160941 45.0001638412
Basis   30: 5.0000000011 15.0000000498 25.0000011239 35.0000160923 45.0001638174
Basis   40: 5.0000000011 15.0000000498 25.0000011239 35.0000160921 45.0001638146
Basis   50: 5.0000000011 15.0000000498 25.0000011239 35.0000160920 45.0001638141
Basis  100: 5.0000000011 15.0000000498 25.0000011239 35.0000160920 45.0001638140
Basis  500: 5.0000000011 15.0000000499 25.0000011239 35.0000160920 45.0001638140
Basis 1000: 5.0000000011 15.0000000499 25.0000011239 35.0000160921 45.0001638140
 vs   1000: 5.0000000011 15.0000000498 25.0000011239 35.0000160920 45.0001638140
```

Small deviation from parabolic potential

Analytic basis for 1D Schrödinger

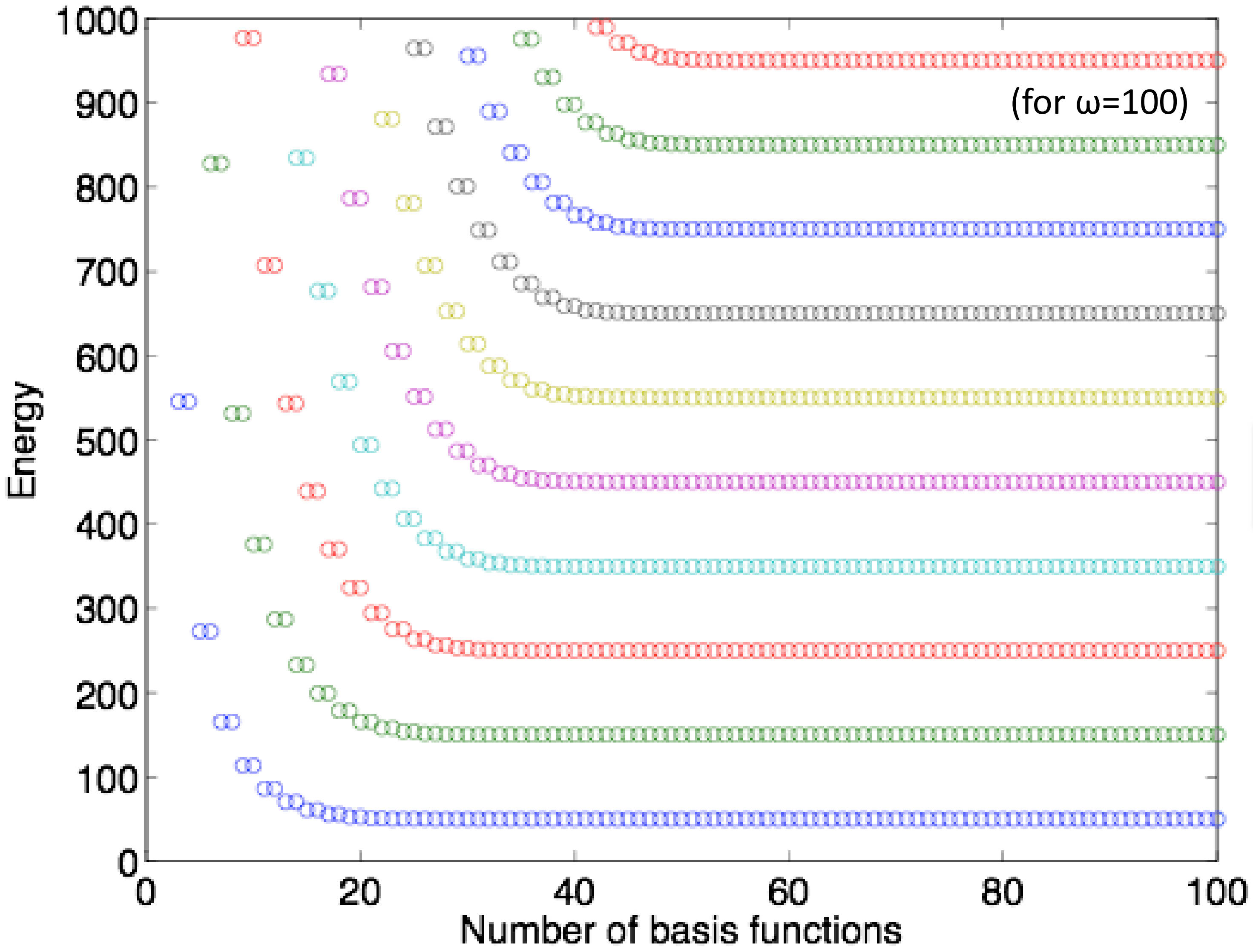
$$\mathcal{H} = \mathcal{H}_0 + \omega^2 \frac{1}{2} \left(x - L/2\right)^2$$

Diagonalization for ω=100

```
Basis    1: 1612.83516712
Basis    2: 1612.83516712 3489.335167121
Basis    3: 545.432268720 3489.335167121 4906.460287744
Basis    4: 545.432268720 1491.845854559 4906.460287744 5961.574479683
Basis    5: 272.846669406 1491.845854559 2437.749780998 5961.574479683 6766.131273180
Basis    6: 272.846669406 827.9383734759 2437.749780998 3289.957372255 6766.131273180
Basis    7: 165.725626383 827.9383734759 1459.116875949 3289.957372255 4035.375770036
Basis    8: 165.725626383 531.4414647108 1459.116875949 2088.259456524 4035.375770036
Basis    9: 114.020783491 531.4414647107 977.1212760878 2088.259456524 2684.174530378
Basis   10: 114.020783491 376.2131869272 977.1212760878 1449.817435511 2684.174530378
Basis   20: 52.8383732874 165.5702774235 324.8097098466 494.1124867823 786.7267172313
Basis   30: 50.0253517681 150.2496119637 253.0023577851 358.8577688673 486.7774845544
Basis   40: 50.0000270879 150.0003977966 250.0120946317 350.0570108938 450.6426996155
Basis   50: 50.0000000036 150.0000000681 250.0000042282 350.0000264935 450.0006836060
Basis   60: 50.0000000000 150.0000000000 250.0000000002 350.0000000012 450.0000000576
Basis   70: 50.0000000000 150.0000000000 250.0000000000 350.0000000000 450.0000000000
Basis   80: 50.0000000000 150.0000000000 250.0000000000 350.0000000000 450.0000000000
Basis   90: 50.0000000000 150.0000000000 250.0000000000 350.0000000000 450.0000000000
Basis  100: 50.0000000000 150.0000000000 250.0000000000 350.0000000000 450.0000000000
Basis  500: 50.0000000001 150.0000000000 250.0000000000 350.0000000001 449.9999999999
Basis 1000: 50.0000000000 150.0000000001 250.0000000002 350.0000000002 449.9999999999
 vs   1000: 50.0000000000 150.0000000000 250.0000000000 350.0000000000 450.0000000000
```

Analytic basis for 1D Schrödinger

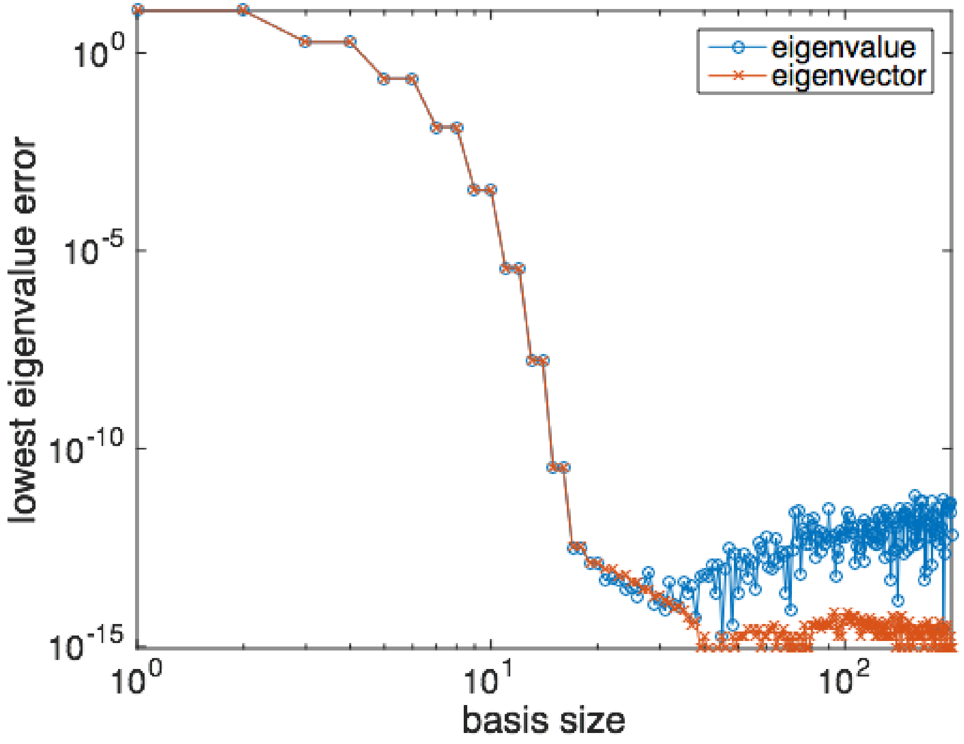
Convergence visually:



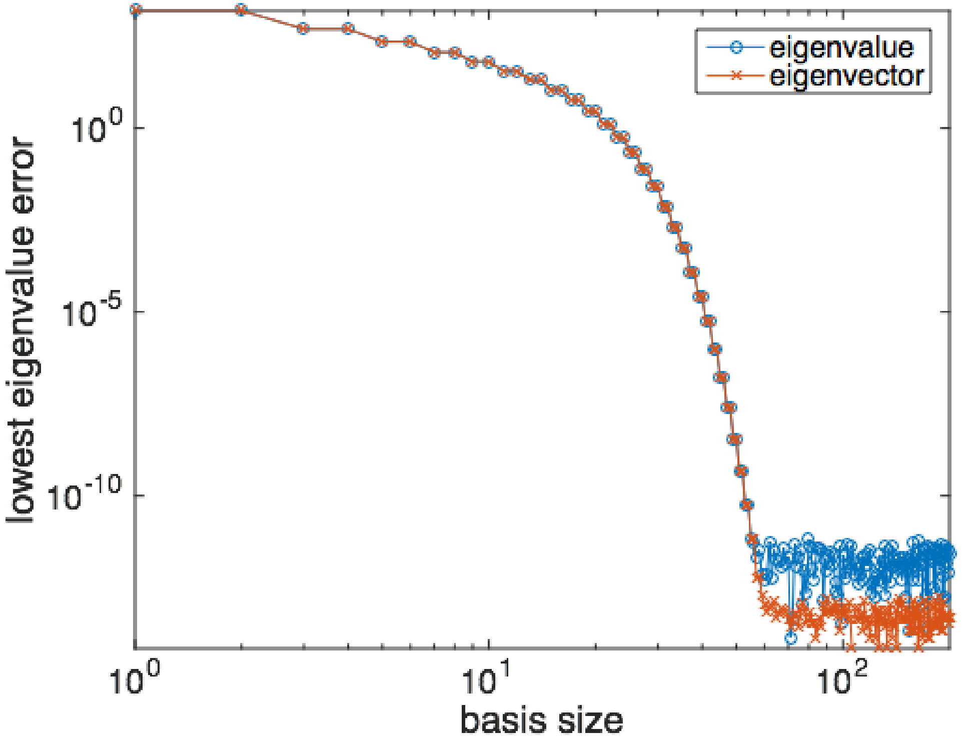
Smaller eigenvalues converge faster than larger.

Analytic basis for 1D Schrödinger

Lowest energy convergence, ω=10



and for ω=100.



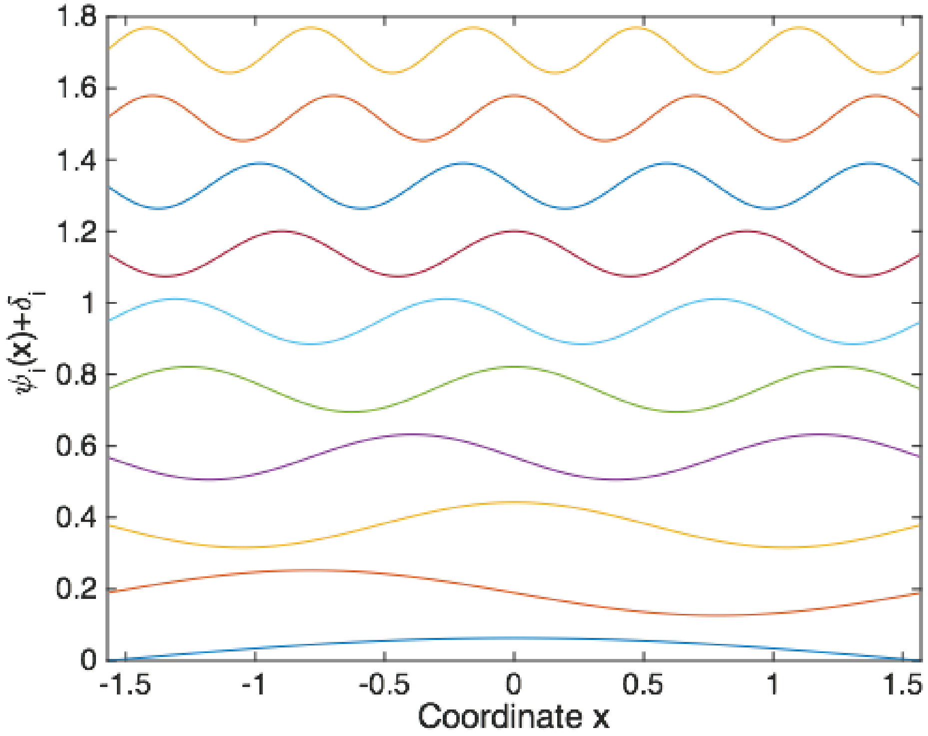
Eigenvector uses (v'*H*v)./(v'*v) instead of the returned eigenvalue.

It seems like we can reach the numerical accuracy with a basis of the order of 100!

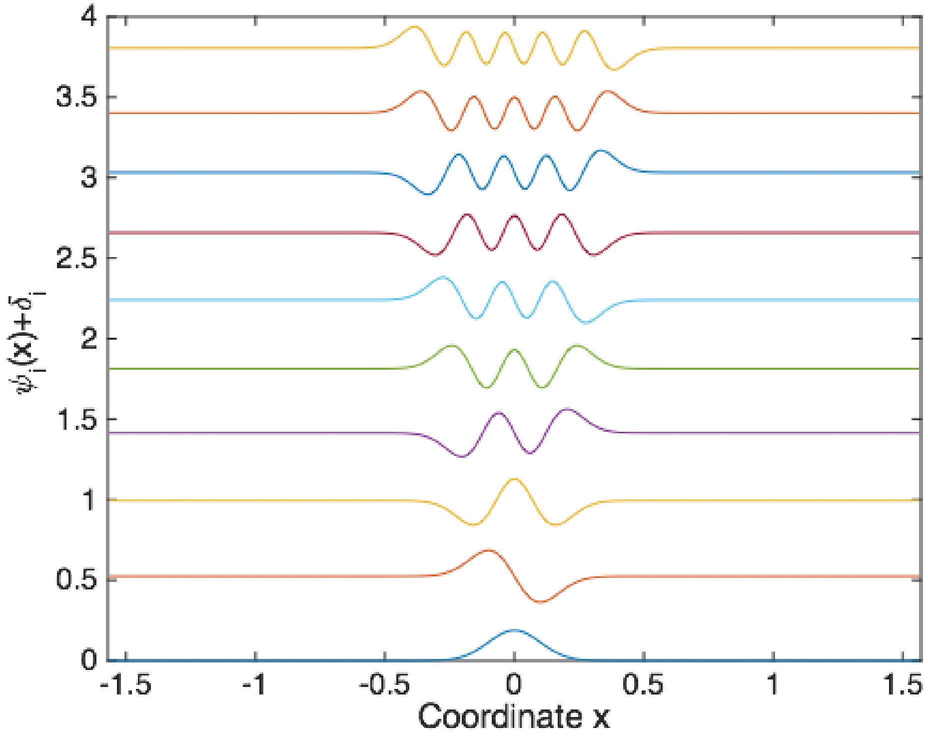
Much more efficient than the finite difference scheme, but not as flexible, as now we would need to calculate the analytic matrix elements for the unknown part of the Hamiltonian.

Analytic basis for 1D Schrödinger

Non-trivial that states like these:



can be used to expand states for ω=100.



We should not think this

$$\mathcal{H} = \mathcal{H}_0 + \omega^2 \mathcal{H}_I$$

as perturbation theory on the original basis with a small ω^2

Perturbation theory for 1D Schrödinger

Perturbation on the original basis with a small ω^2

$$\mathcal{H} = \mathcal{H}_0 + \omega^2 \mathcal{H}_I \qquad E_n = E_n^0 + \omega^2 E_n^1 + \omega^4 E_n^2 + \ldots$$

ψ_n^0 n'th basis state, eigenstate of $\mathcal{H}_0$

$$E_n^1 = \langle \psi_n^0 | \mathcal{H}_I | \psi_n^0 \rangle$$

$$E_n^2 = \sum_{m \neq n} \frac{|\langle \psi_m^0 | \mathcal{H}_I | \psi_n^0 \rangle|^2}{E_n^0 - E_m^0}$$

The matrix elements are what we calculated before, if ω=1 and diagonal is modified:

```
Hpert=HO_BOX_Ham_matrix(1, N);
for l=1:N,
    Hpert(l,l)=Hpert(l,l)-l^2/2;
end
```

```
n=1; % lowest state
En0=n^2/2;
En1=Hpert(n,n); % 1st order perturbation theory
En2=0;
for m=1:N,
    if m~=n,
        En2=En2+abs(Hpert(m,n))^2./(n^2/2 - m^2/2); % 2nd order
    end
end
om=0:.1:4;
plot(om,En0+om.^2*En1,'-',om,En0+om.^2*En1+om.^4*En2,'r-');
```

Perturbation theory for 1D Schrödinger

Perturbation on the original basis with a small ω^2

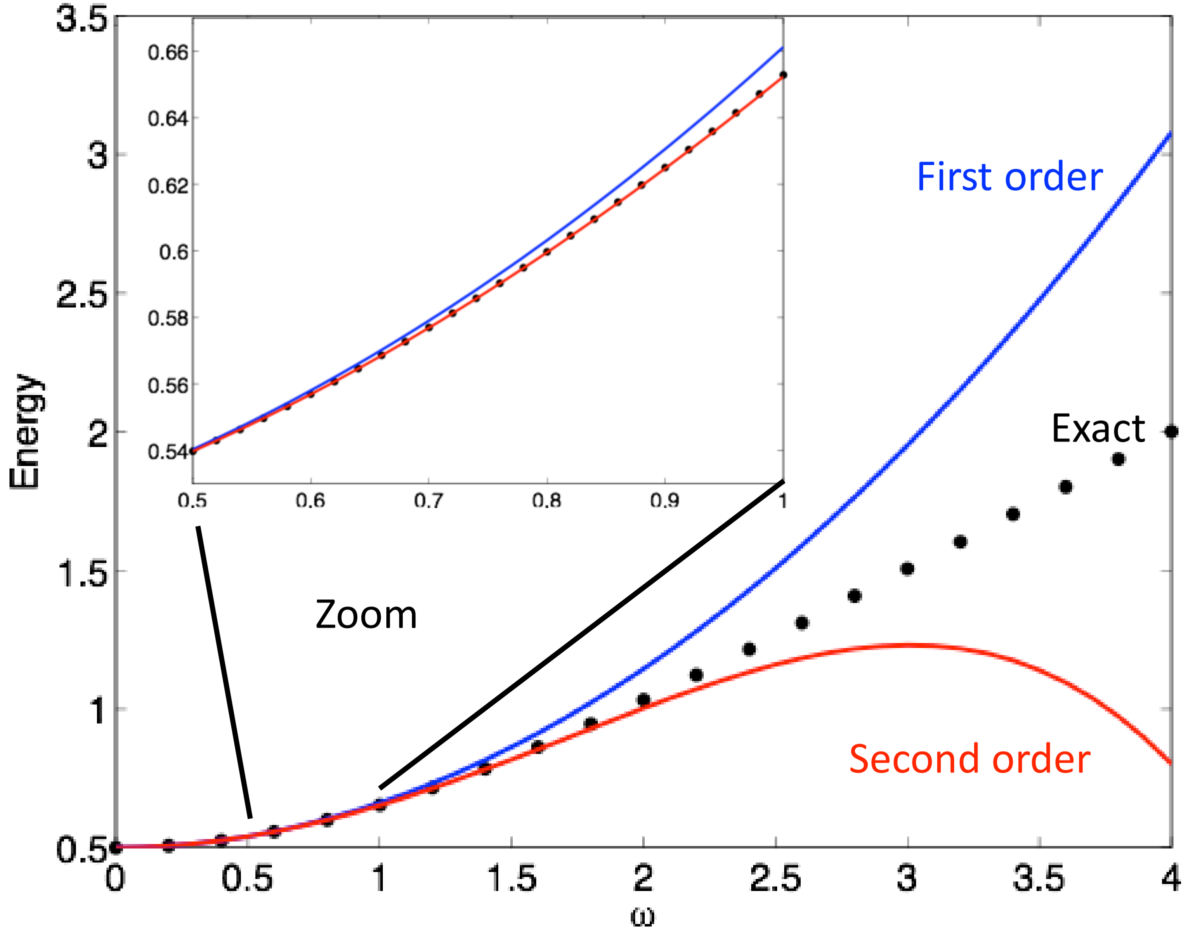
$$\mathcal{H} = \mathcal{H}_0 + \omega^2 \mathcal{H}_I$$

$$E_n = E_n^0 + \omega^2 E_n^1 + \omega^4 E_n^2 + \dots$$

$$E_n^1 = \langle \psi_n^0 | \mathcal{H}_I | \psi_n^0 \rangle$$

ψ_n^0 n'th basis state, eigenstate of $\mathcal{H}_0$

$$E_n^2 = \sum_{m \neq n} \frac{|\langle \psi_m^0 | \mathcal{H}_I | \psi_n^0 \rangle|^2}{E_n^0 - E_m^0}$$



Second order perturbation is first more accurate, but then fails seriously at large omega

Using analytic basis is different from perturbation theory since a new eigenvalue problem is solved

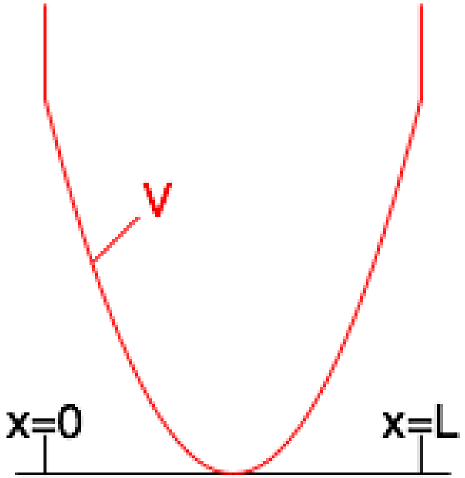
FEM Basis for Schrödinger Equation

- Now, with the potential $V(x)=½\,\omega^2\,(x\text{-}L/2)^2$ the matrix elements look like

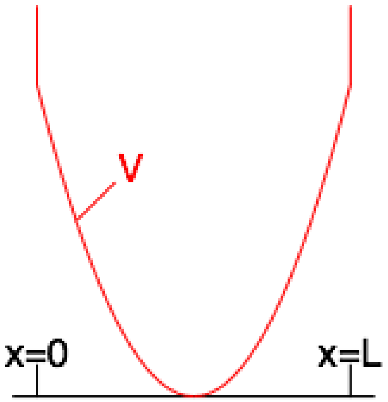
$$h_{ji} = h^0_{ji} + h^I_{ji} = \int_0^L \frac{1}{2}\phi_i'(x)\phi_j'(x) + V(x)\phi_i(x)\phi_j(x)\,dx$$

$$s_{ji} = \int_0^L \phi_i(x)\phi_j(x)\,dx$$

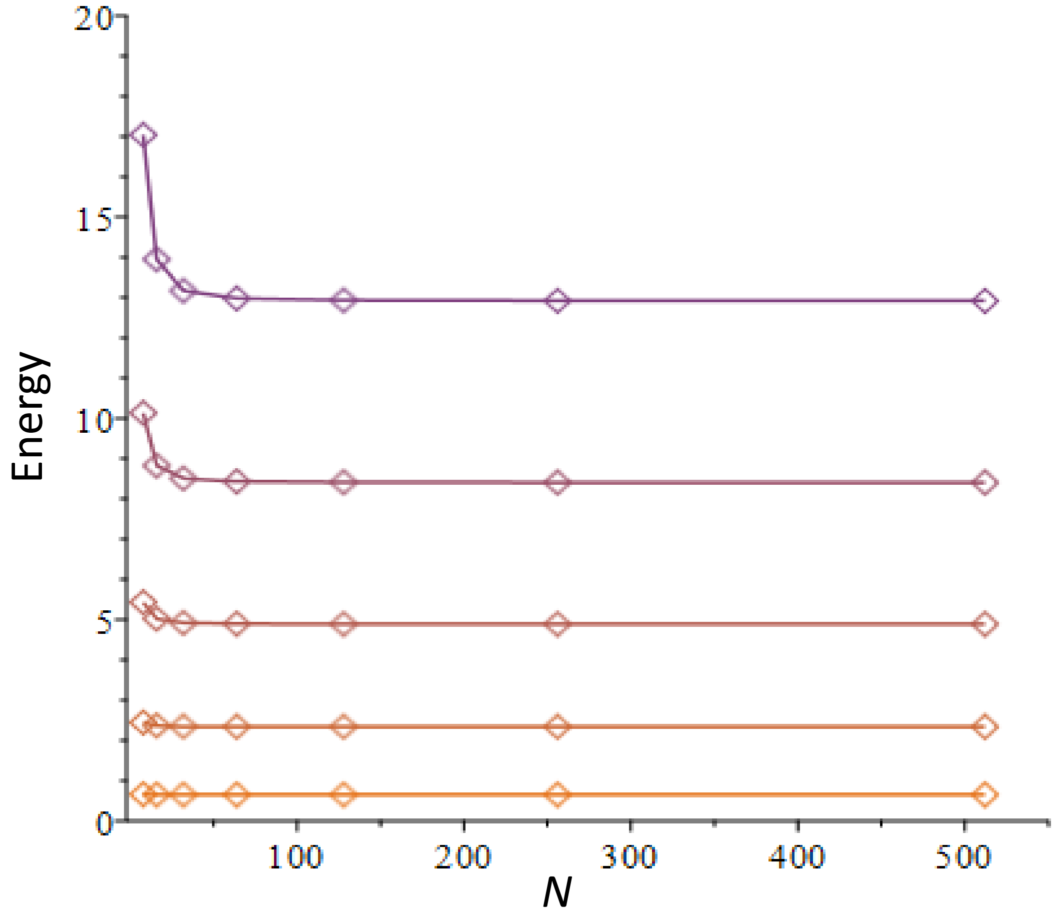
and both *H* and *S* are tridiagonal.

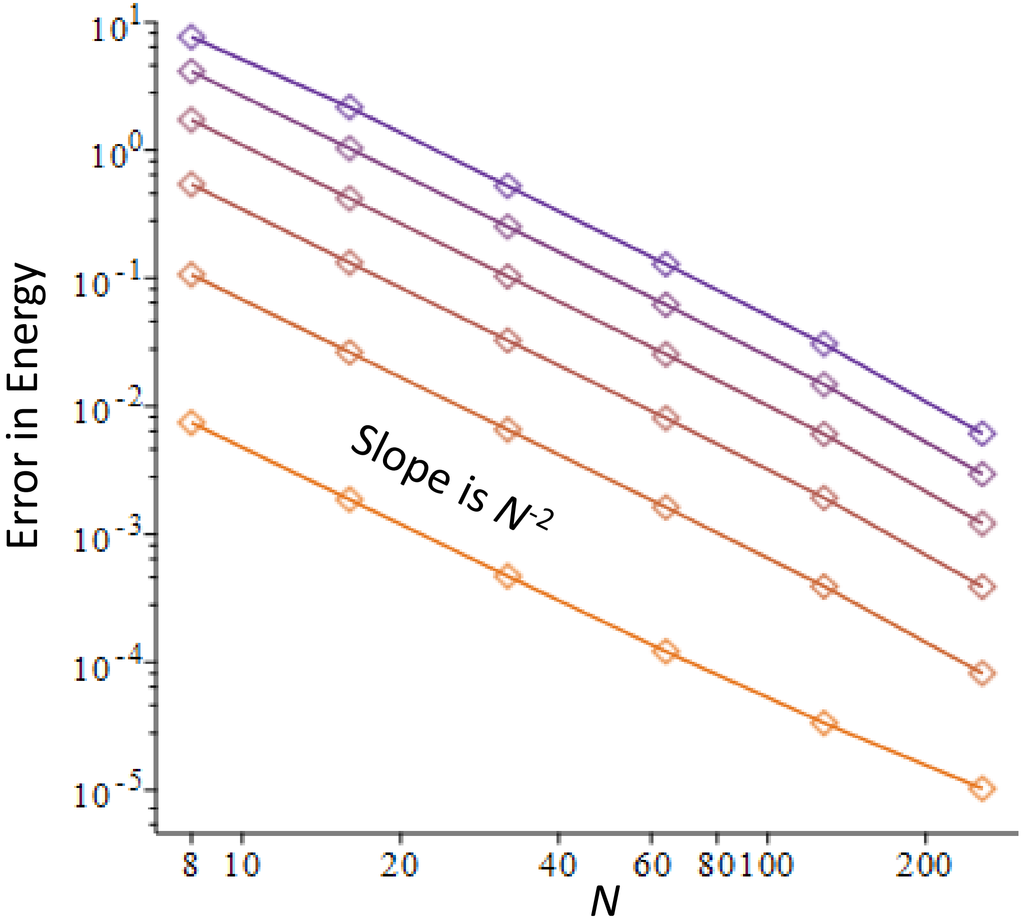


FEM Basis for Schrödinger Equation

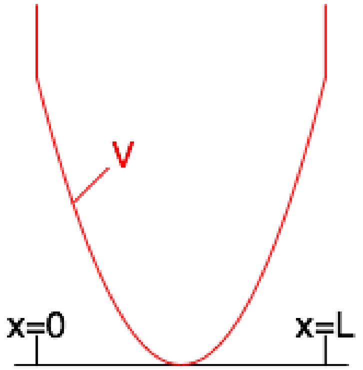


- Convergence of the lowest eigenvalues for ω=1.0 looks familiar:

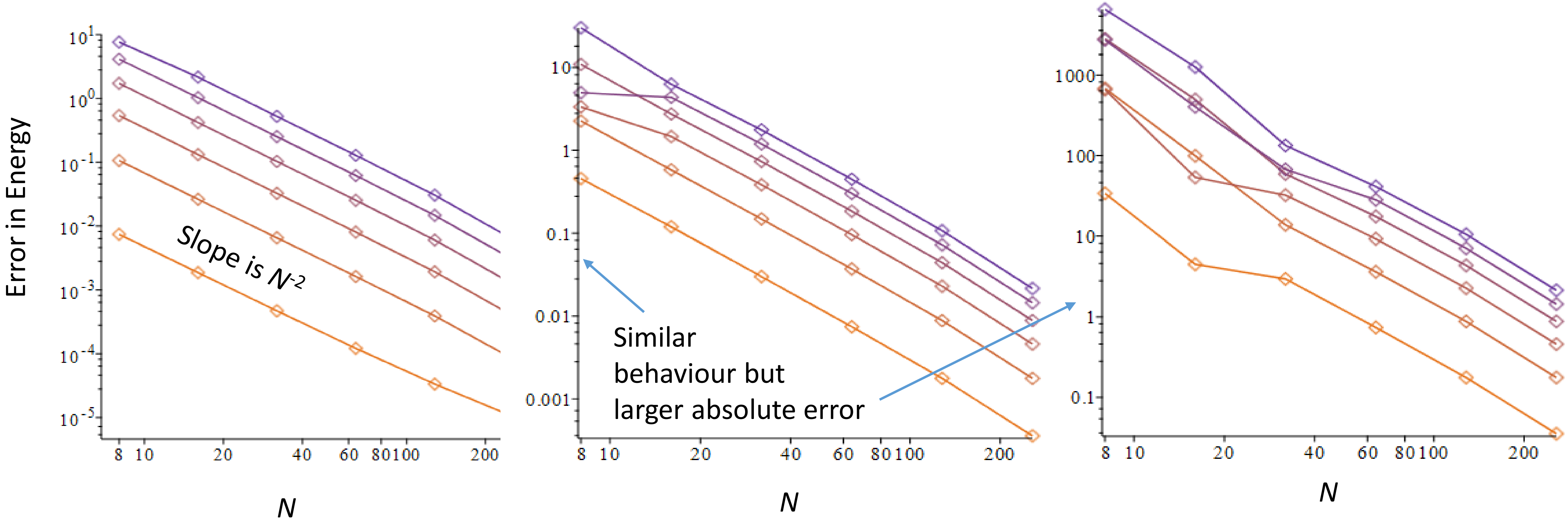




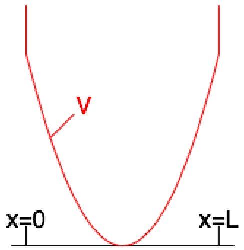
FEM Basis for Schrödinger Equation



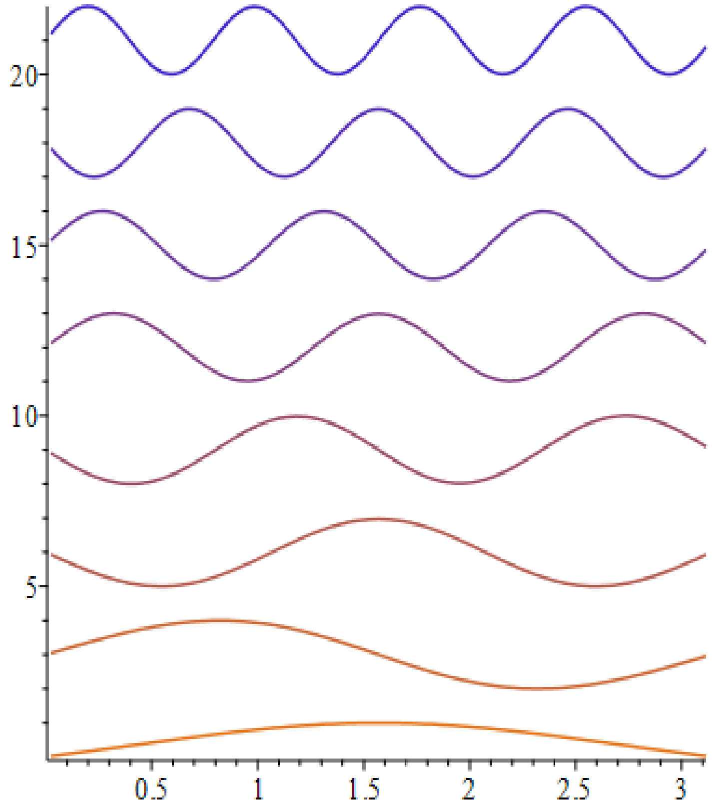
- Convergence of the lowest eigenvalues for ω=1.0, ω=10.0 and ω=100.0

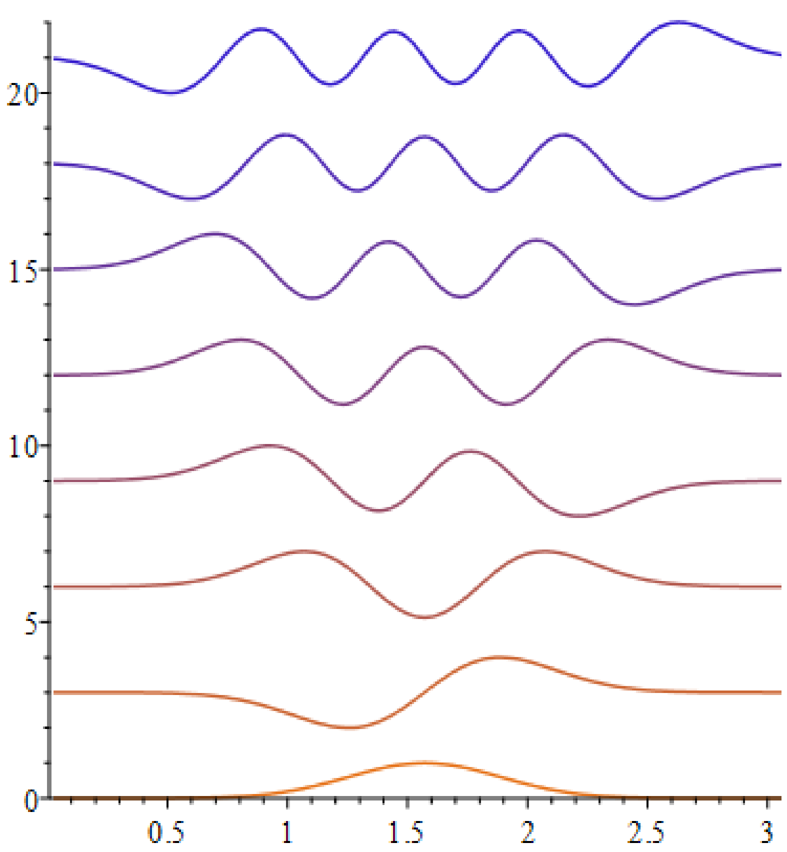


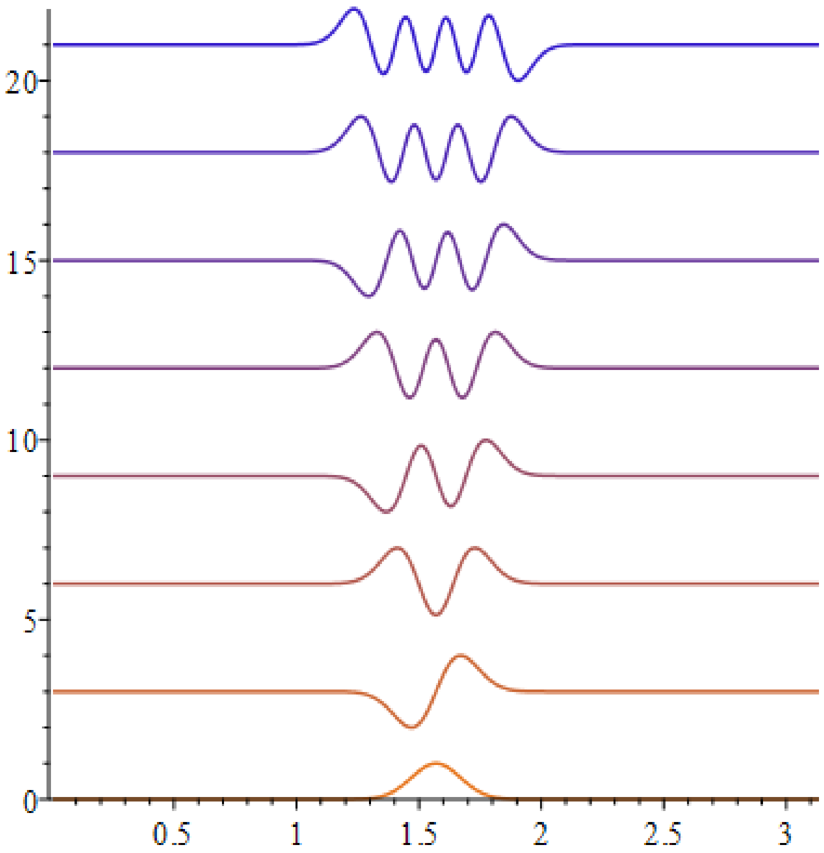
FEM Basis for Schrödinger Equation



- Eigenfunctions for ω=1.0 ω=10.0 ω=100.0







Homework 8

Finite-element discretizations of the Schrödinger equation

In this exercise we study finite-element discretization of the Schrödinger equation in one dimensional potential well. Consider the eigenvalue problem

$$\begin{cases} -\frac{1}{2}\psi''(x) + V(x)\psi(x) = \epsilon\psi(x), & 0 < x < 1 \\ \psi(0) = \psi(1) = 0 \end{cases}$$

where the potential is given by two cusps

$$V(x) = -150e^{-40(x-0.25)^2} - 50e^{-10(x-0.75)^2}$$

a) Implement a finite-element solver for the eigenvalue problem. (You can use the code skeleton provided with the homework.) How many of the states have negative energy indicating that they are bound to the cusps? Use uniform distribution of the nodes, $x_i = ih$. To calculate the entries of the potential matrix

$$V_{ij} = \int_0^1 \phi_i(x)V(x)\phi_j(x)\,dx$$

you can either use some library routine for numerical integration (see, e.g. `integral`

b) Plot some of the lowest eigenfunctions. Is the uniform distribution of the nodes optimal? Given a budget of $N = 30$ nodes devise a more effective placement in the interval $[0, 1]$ to approximate the lowest eigenpair. (2 p.)

Basis Function Expansions: Analytic & FEM

Discuss: What kind the pros and cons of analytic and FEM basis functions

A Posteriori Error Analysis

- Recall $a(u - u_h, v_h) = 0 \quad \forall v_h \quad \Rightarrow \quad a(u-u_h, v) = R(v) = R(v - v_h) = a(u - u_h, v - v_h) \quad \forall v$
- Here the *residual*: $R(v) = \int_0^L (u-u_h)' v' \, dx = \sum_i - \int_{h_i} (u-u_h)'' v \, dx + \sum_i [[u_h']]_i v$

$$= \sum_i \int_{h_i} f (v - v_h) \, dx + \sum_i [[u_h']]_i (v - v_h)$$

$$a(u-u_h, v) = \sum_i \int_{h_i} f (v - v_h) \, dx + \sum_i [[u_h']]_i (v - v_h) \quad \forall v$$

$$||u - u_h||_a ||v||_a$$

$$\left(\sqrt{\sum_i \underbrace{h_i^2 ||f||_{L^2}^2 + h_i [[u_h']]_i^2}_{\mathcal{E}_i^2}} \right) ||v||_a$$

A Posteriori Error Analysis

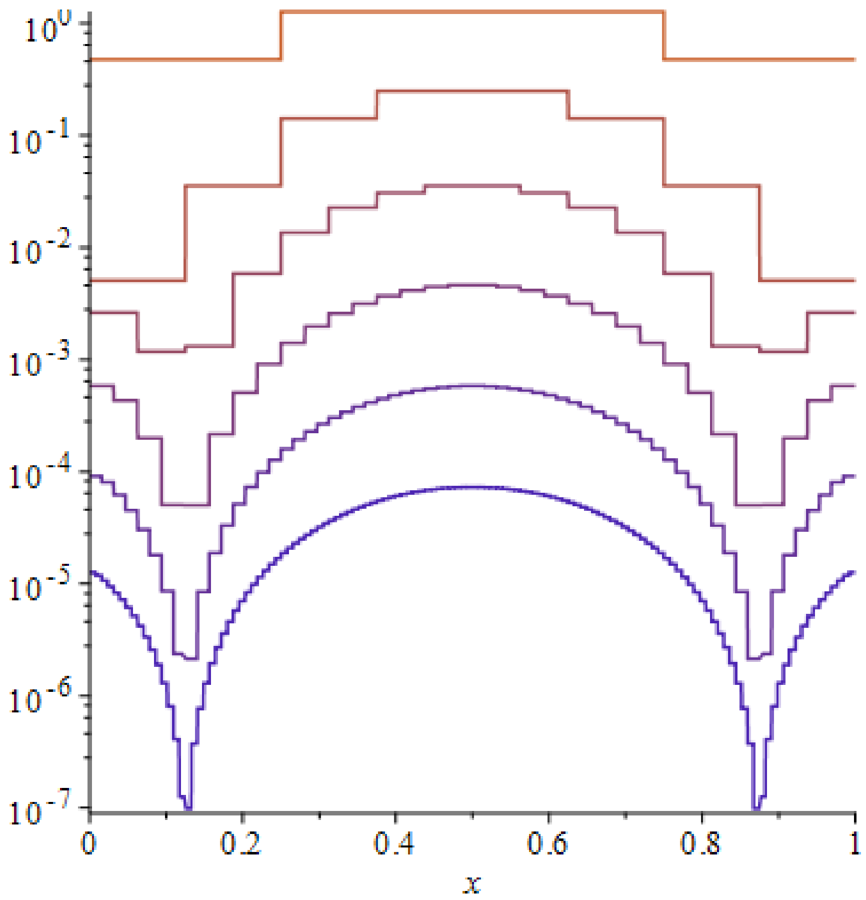
- A posteriori error estimate $$||u - u_h||_a \leq C \left(\sqrt{\sum_i h_i^2 ||f||_{L^2}^2 + h_i [[u_h']]_i^2} \right)$$

Discuss: What this kind of error estimate can do for you?

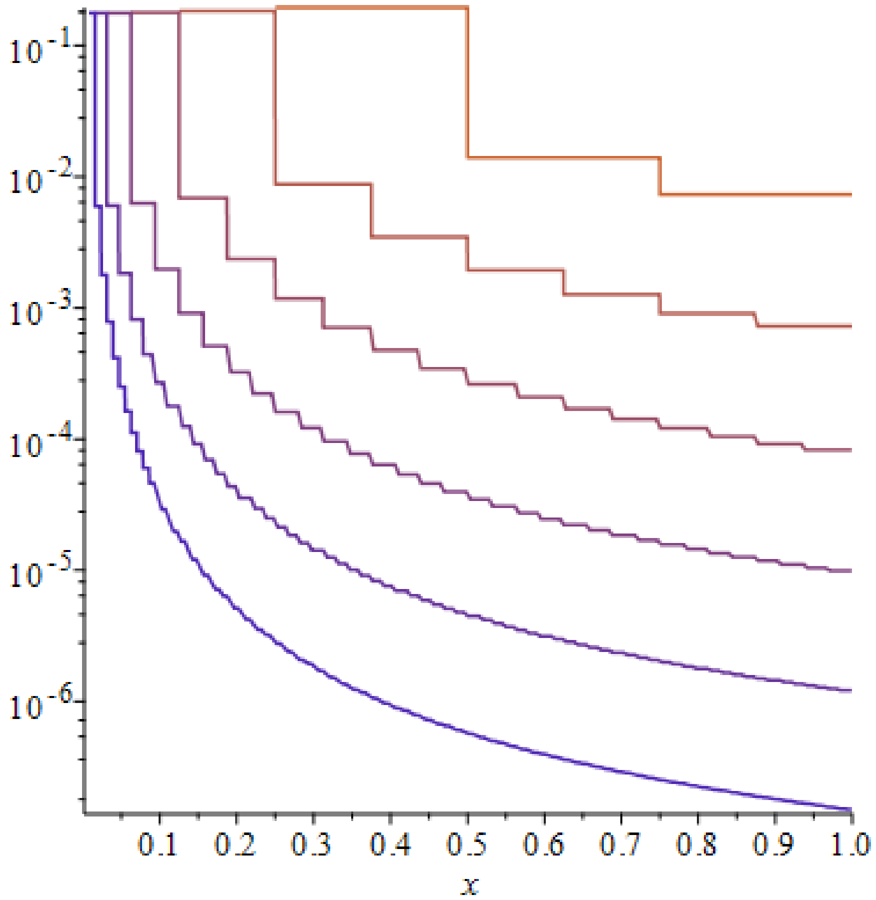
A Posteriori Error Analysis

- Plot $\mathcal{E}_i^2 = h_i^2 ||f||_{L^2}^2 + h_i [[u_h']]_i^2$

$$u(x) = \sin\left(\frac{\pi x}{L}\right) \exp\left(-(x - L/2)^2\right)$$



$$u(x) = \sqrt{x} - x$$

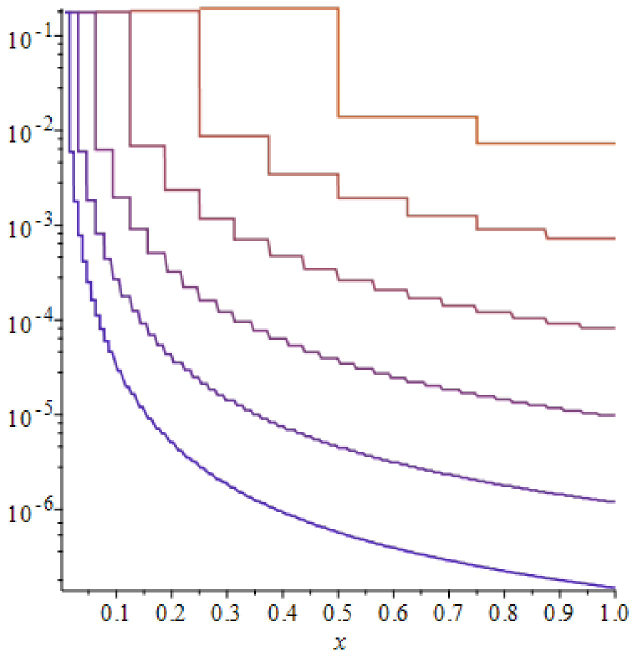


A Posteriori Error Analysis

- Estimator obviously senses trouble at x=0 but… overshoots terribly:

$$\mathcal{E}_i^2 = h_i^2 ||f||_{L^2}^2 + h_i [[u_h']]_i^2$$

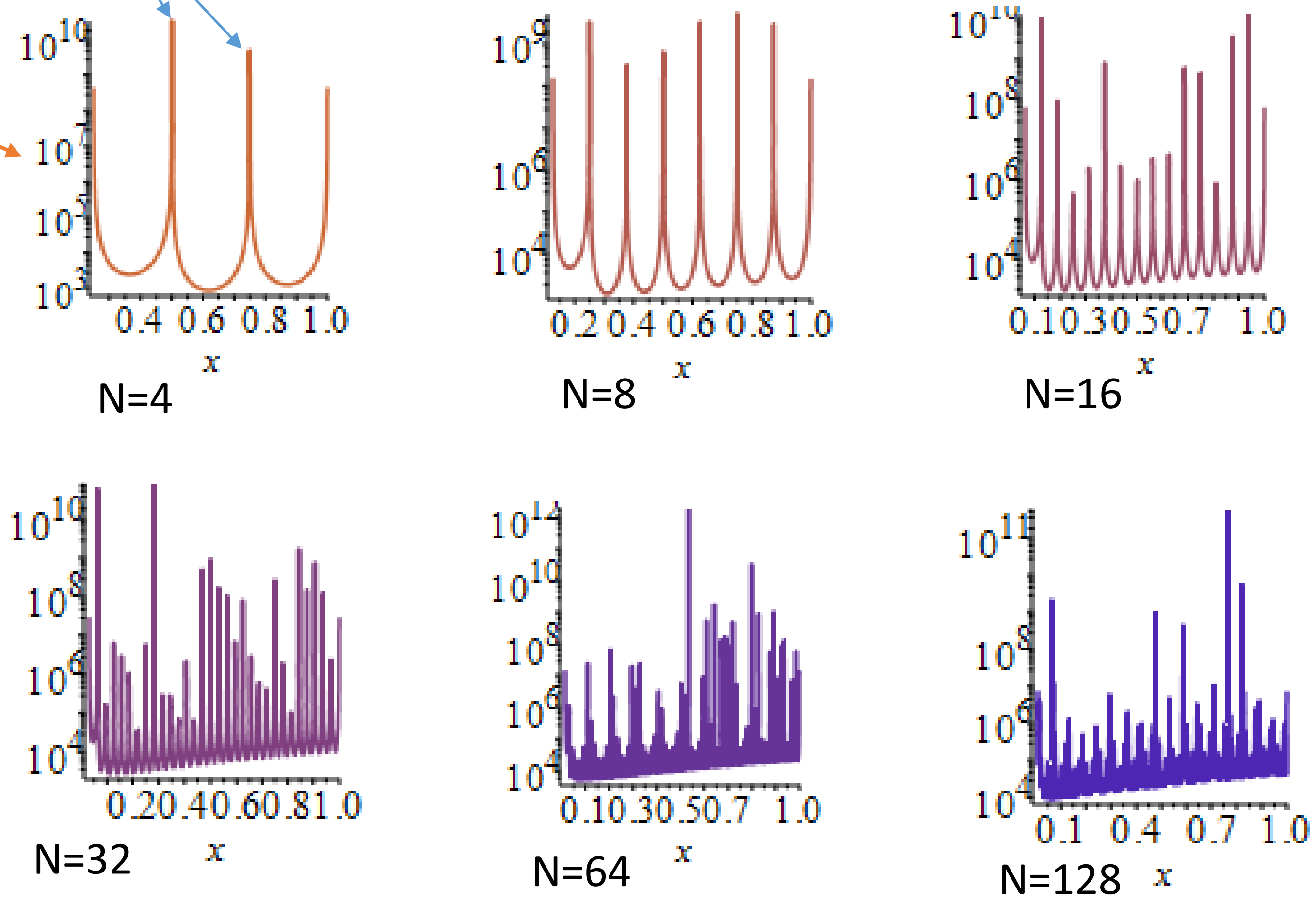
$$u(x) = \sqrt{x} - x$$



Interpolation at nodes

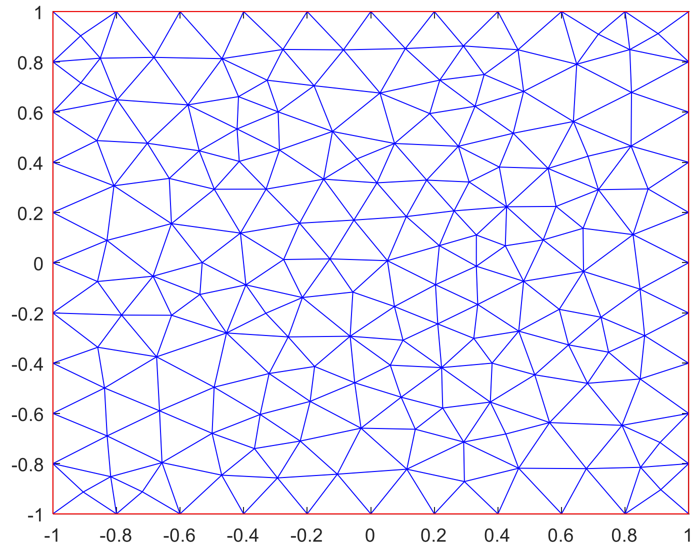
$$\mathcal{E}_i^2 / (u(x) - u_h(x))^2$$

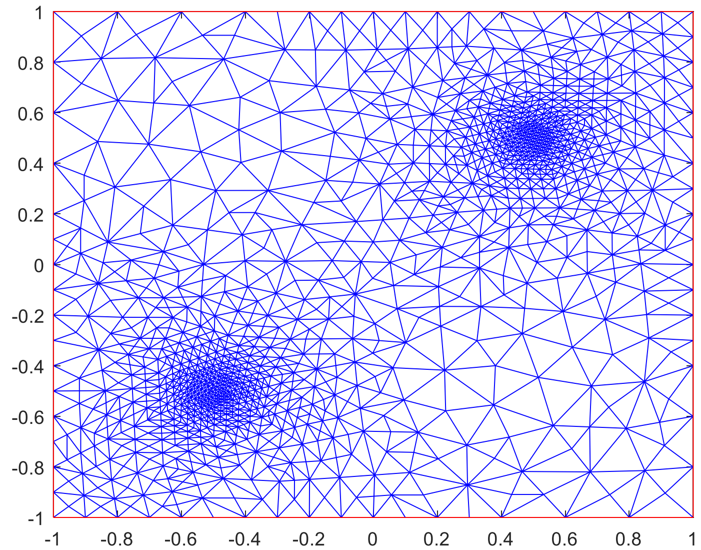
Log-scale

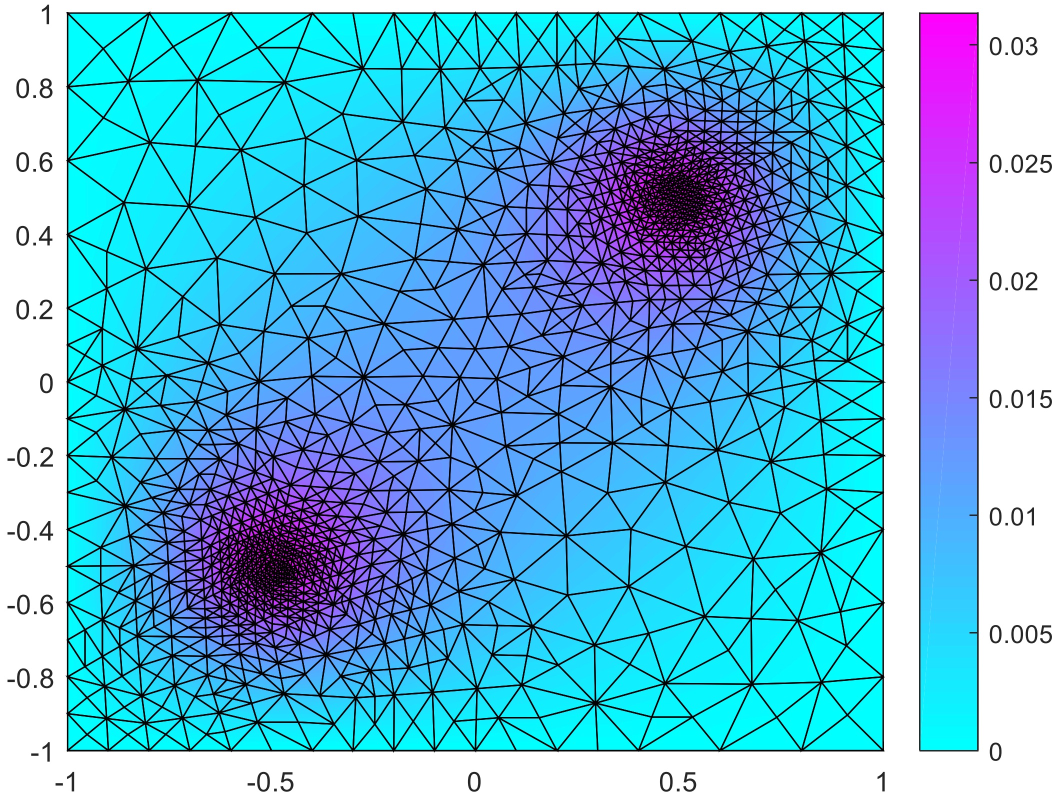


Adaptive Meshing in 2D

$$f(x,y) = \exp\left(1 - 100((x-0.5)^2 + (y-0.5)^2)\right) + \exp\left(1 - 100((x+0.5)^2 + (y+0.5)^2)\right)$$



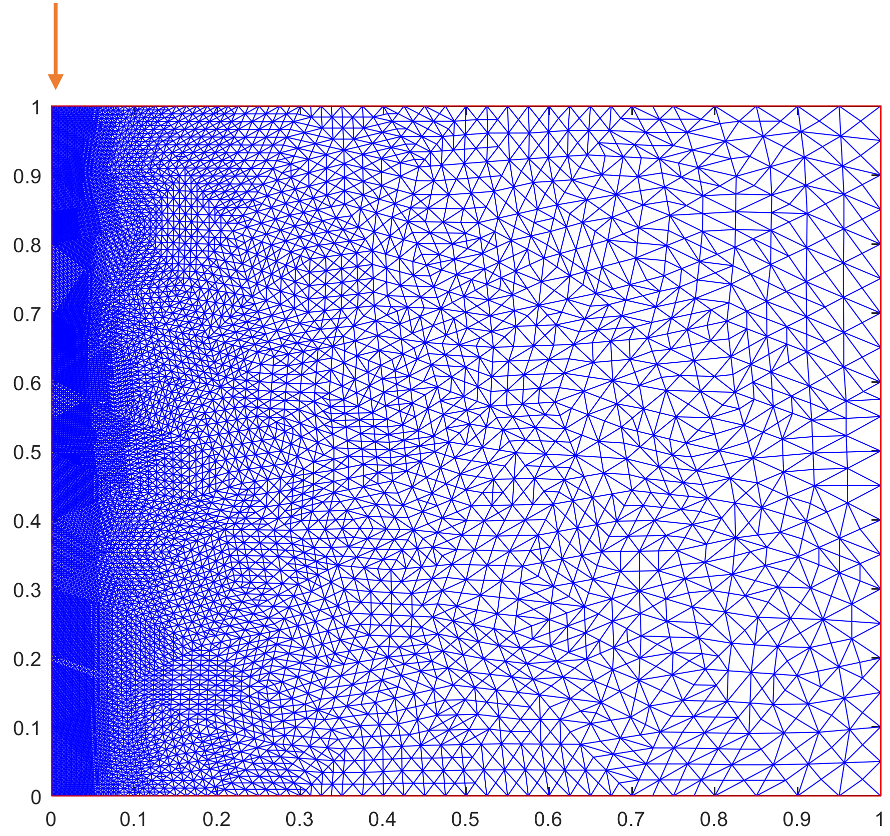


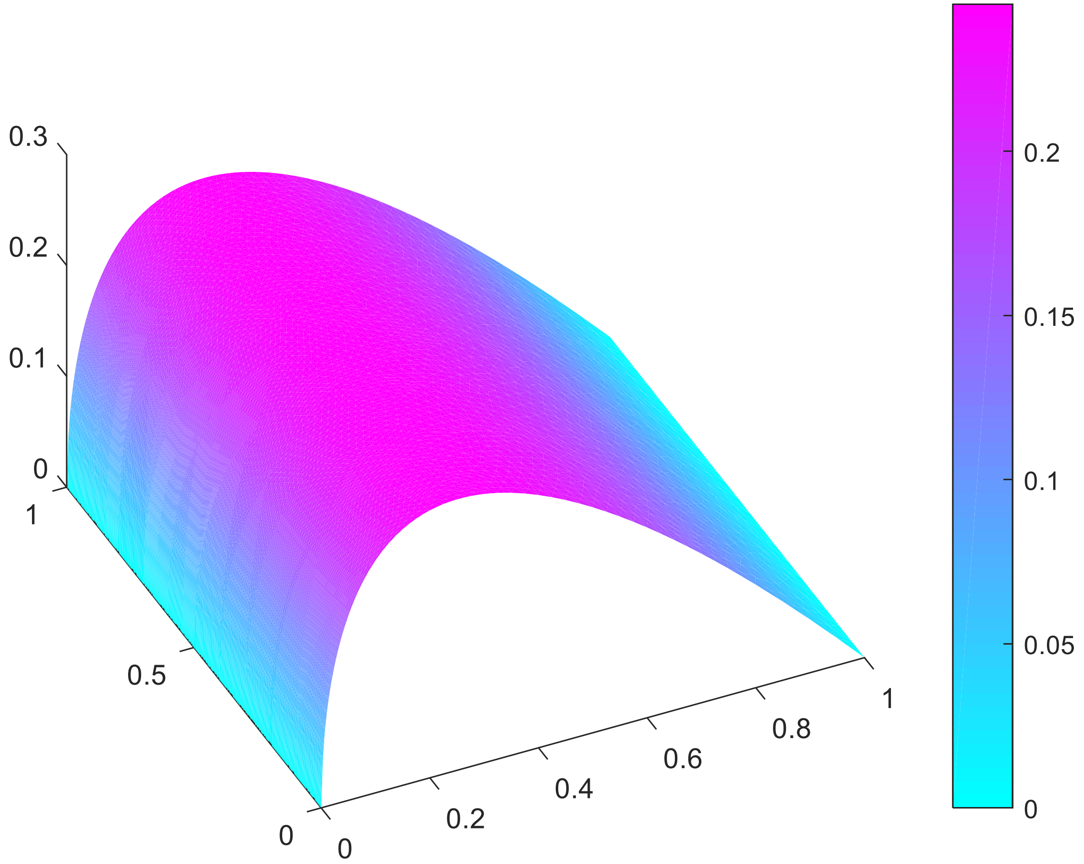


Adaptive Meshing in 2D

- Revisit the old problem with: $u(x,y) = \sqrt{x} - x$

Also the 2D adaptive algorithm captures the singularity at x=0





A Posteriori Error Estimator for Eigenproblems

- Recall $\mathcal{E}_i^2 = h_i^2 ||f||_{L^2}^2 + h_i[[u_h']]_i^2$
- For eigenproblems: $\begin{cases} a(\psi_j, v) = \epsilon_j(\psi_j, v) \\ a(\psi_{hj}, v_h) = \epsilon_{hj}(\psi_{hj}, v_h) \end{cases} \Rightarrow a(\psi_j - \psi_{hj}, v_h) = (\epsilon_j \psi_j - \epsilon_{hj}\psi_{hj}, v_h)$

- ”Orthogonality” reads $a(\psi_j - \psi_{hj}, v_h) - (\epsilon_j \psi_j - \epsilon_{hj}\psi_{hj}, v_h) = 0$
- And the error estimator becomes $\mathcal{E}_{i,j}^2 = h_i^2 \epsilon_{hj}^2 ||\psi_{hj}||_{L^2}^2 + h_i[[\psi_{hj}']]_i^2$

Larger eigenvalues more problematic

For further FEM stuff: Go to MS-E1653 Finite Element Method

Numerical Integration

weights

Compute: $I(f) = \int_a^b f(x)\, dx$ by $I_n(f) = \sum_{i=0}^{n} w_i f(x_i)$

points

Newton-Cotes formulae

interpolating Lagrange polynomial

$$I_n(f) = I(P_n) \qquad P_n(x) = \sum_{i=0}^{n} L_{n,i}(x) f(x_i) \quad \longrightarrow \quad w_i = \int_a^b L_{n,i}(x)\, dx$$

Closed formulae $x_0 = a,\ x_n = b,\ \Delta x = (b-a)/n$

$$n = 1:\ L_{1,0}(x) = \frac{b-x}{b-a},\ L_{1,1}(x) = \frac{x-a}{b-a} \quad \longrightarrow \quad I_{1,\text{closed}}(f) = \frac{b-a}{2}[f(a) + f(b)]$$

Trapezoidal rule

$$n = 2: \quad \longrightarrow \quad I_{2,\text{closed}}(f) = \frac{b-a}{6}\left[f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)\right]$$

Simpson's rule

$$n = 3: \quad \longrightarrow \quad I_{3,\text{closed}}(f) = \frac{b-a}{8}[f(a) + 3f(a+\Delta x) + 3f(a+2\Delta x) + f(b)]$$

Three-eighths rule

Numerical Integration

Newton-Cotes formulae $\quad I_n(f) = I(P_n) \qquad P_n(x) = \sum_{i=0}^{n} L_{n,i}(x) f(x_i) \quad \longrightarrow \quad w_i = \int_a^b L_{n,i}(x)\,dx$

Closed formulae $\quad x_0 = a,\ x_n = b,\ \Delta x = (b-a)/n$

$$n = 1:\ L_{1,0}(x) = \frac{b-x}{b-a},\ L_{1,1}(x) = \frac{x-a}{b-a} \longrightarrow I_{1,\text{closed}}(f) = \frac{b-a}{2}[f(a) + f(b)]$$

Trapezoidal rule

$$n = 2: \longrightarrow I_{2,\text{closed}}(f) = \frac{b-a}{6}\left[f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)\right]$$

Simpson’s rule

$$n = 3: \longrightarrow I_{3,\text{closed}}(f) = \frac{b-a}{8}[f(a) + 3f(a+\Delta x) + 3f(a+2\Delta x) + f(b)]$$

Three-eighths rule

Open formulae $\quad \Delta x = (b-a)/(n+2),\ x_i = a + (i+1)\Delta x$

$$n = 0: \longrightarrow I_{0,\text{open}}(f) = (b-a)f\left(\frac{a+b}{2}\right)$$

Midpoint rule

$$n = 1: \longrightarrow I_{1,\text{open}}(f) = \frac{b-a}{2}[f(a+\Delta x) + f(a+2\Delta x)]$$

$$n = 2: \longrightarrow I_{2,\text{open}}(f) = \frac{b-a}{3}[2f(a+\Delta x) - f(a+2\Delta x) + 2f(a+3\Delta x)]$$

Composite Newton-Cotes Quadrature

Compute $I_n(f) = I(P_n)$ for increasing n ➡ high-order interpolating polynomial ➡ stability problem

Composite rules: Subdivide the interval into pieces by $h = \dfrac{b-a}{n},\ x_j = a + jh$

Trapezoidal rule

$$I_{1,\text{closed}}(f) = \frac{b-a}{2}[f(a) + f(b)] \Rightarrow I_{1,\text{composite}} = \frac{h}{2}\left[f(a) + 2\sum_{j=1}^{n-1} f(x_j) + f(b)\right]$$

Simpson's rule for n=2m

$$I_{2,\text{closed}}(f) = \frac{b-a}{6}\left[f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)\right] \Rightarrow I_{2,\text{composite}}(f) = \frac{h}{3}\left[f(a) + 4\sum_{j=1}^{m} f(x_{2j-1}) + 2\sum_{j=1}^{m-1} f(x_{2j}) + f(b)\right]$$

1 4 1

1 4 2 4 2 4 2 4 2 4 2 4 2 4 2 4 2 4 2 4 2 4 1

Gaussian Quadrature

Instead of using uniformly distributed points it is possible to search for points and weight such that

$$I_n(f) = \sum_{i=1}^{n} w_i f(x_i)\, dx$$

is exact for the highest possible polynomial degree

Solution: Legendre polynomials for -1 < t < 1:

$$P_0(t) = 1,\ P_1(t) = t,\ P_{n+1}(t) = \frac{2n+1}{n+1} t P_n(t) - \frac{n}{n+1} P_{n-1}(t) \qquad x = \frac{b-a}{2} t + \frac{a+b}{2}$$

Take the roots t_j of $P_n(t)$ map $x_j = \frac{b-a}{2} t_j + \frac{a+b}{2}$ set $L_i(x) = \prod_{j=1, j\neq i}^{n} \frac{x - x_j}{x_i - x_j}$ $w_i = \int_a^b L_i(x)\, dx$

This is Gaussian quadrature of order *n*. It exact for polynomials up to order *2n-1*

$$n = 1:\ \ I_1(f) = (b-a) f\left(\frac{a+b}{2}\right) \qquad n = 2:\ \ t_{1,2} = \pm\sqrt{\frac{1}{3}} \qquad I_2(f) = \frac{b-a}{2}\left[f\left(\frac{a+b}{2} - \sqrt{\frac{1}{3}}\frac{b-a}{2}\right) + f\left(\frac{a+b}{2} + \sqrt{\frac{1}{3}}\frac{b-a}{2}\right)\right]$$

and so on...

Gaussian Quadrature: Why Does It Work?

Basic reason: Legendre polynomials are orthogonal: $\int_{-1}^{1} P_i(t)P_j(t)\,dt = 0,\ i \neq j \Rightarrow \int_a^b P_i(x)P_j(x)\,dx = 0,\ i \neq j$

Suppose that *g(x)* is a polynomial of degree ≤ *n-1*. Then $g(x) = \sum_{i=1}^{n} g(x_i)L_i(x)$

$$\Rightarrow \quad I(g) = \int_a^b g(x)\,dx = \sum_{i=1}^{n} g(x_i) \int_a^b L_i(x)\,dx = \sum_{i=1}^{n} w_i g(x_i) = I_n(g)$$

Next, suppose that *p(x)* is a polynomial of degree ≤ *2n-1*. Then $p(x) = q(x)P_n(x) + r(x)$ with degrees *q(x), r(x)* ≤ *n-1*

$$I(p) = I(qP_n + r) = \underbrace{\int_a^b q(x)P_n(x)\,dx}_{=0 \text{ since } q(x) = \sum_{j=0}^{n-1} c_j P_j(x)} + I(r) = I(r)$$

and by above $I(r) = I_n(r)$

But $p(x_i) = q(x_i)P_n(x_i) + r(x_i) = r(x_i) \Rightarrow I_n(r) = I_n(p) \Rightarrow I(p) = I_n(p)$

Adaptive Quadrature

1. Compute an approximation $I_0(f) \approx \int_a^b f(x)\,dx$
2. Split the interval in two pieces: [a,c] and [c,b] with c = (a+b)/2 and compute $\begin{cases} I_1(f) \approx \int_a^c f(x)\,dx \\ I_2(f) \approx \int_c^b f(x)\,dx \end{cases}$
3. Compare $I_1 + I_2$ to I_0 to estimate the error. Is the error below tolerance ε?
4. If the error is small enough we are done. If not split [a,c] and [c,b] and continue in 2 with tolerance ε/2.

Step 3:

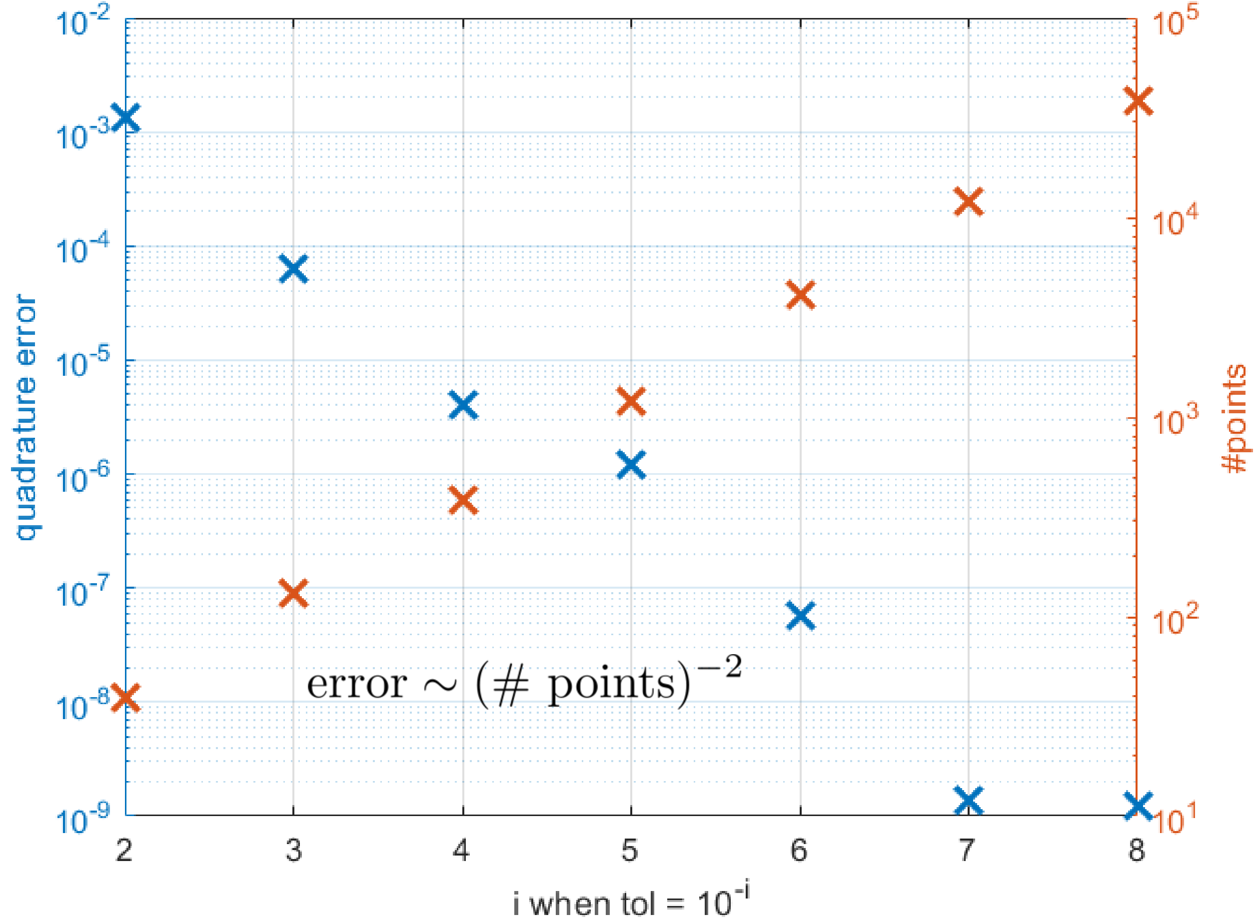
If the error has a form $\int_a^b f(x)\,dx = I_0(f) + Ch^p + O(h^{p+1}) \Rightarrow \int_a^b f(x)\,dx = I_1(f) + I_2(f) + C\left(\frac{h}{2}\right)^p + O(h^{p+1})$

Then $C\left(\frac{h}{2}\right)^p = \underbrace{\frac{1}{2^p - 1}\left[I_1(f) + I_2(f) - I_0(f)\right]}_{\text{Computable error estimator}}$

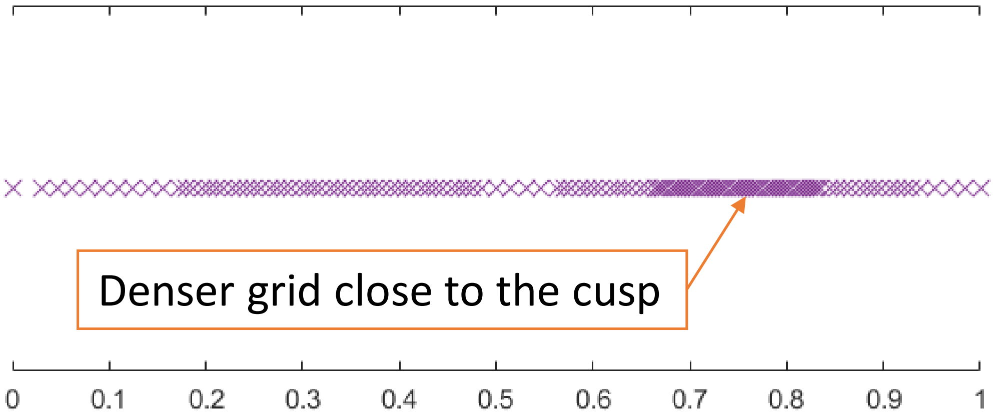
Adaptve Trapezoidal Rule

Consider $f(x) = 50e^{-10(x-0.75)^2}, \quad \int_0^1 f(x)\,dx$

reference value: $I(f) = 24.320775623813379$ tol = 10^{-i}



For tol = 10^{-3}



Denser grid close to the cusp

Uniform error distribution