Course 18.327 and 1.130
Wavelets and Filter Banks

Numerical solution of PDEs: Galerkin approximation; wavelet integrals (projection coefficients, moments and connection coefficients); convergence

Numerical Solution of Differential Equations

Main idea: look for an approximate solution that lies in V_j. Approximate solution should converge to true solution as $j \to \infty$.

Consider the Poisson equation

$$\frac{\partial^2 \mu}{\partial x^2} = f(x) \quad \text{------------------}\square \quad \left(\begin{array}{l}\text{leave boundary} \\ \text{conditions till later}\end{array}\right)$$

Approximate solution:

$$u_{approx}(x) = \sum_k c[k] \underbrace{2^{j/2} \phi(2^j x - k)}_{\substack{\phi_{j,k}(x) \\ \text{trial functions}}} \quad \text{------------}\square$$

Method of weighted residuals: Choose a set of test functions, $g_n(x)$, and form a system of equations (one for each n).

$$\int \frac{\partial^2 u_{approx}}{\partial x^2} g_n(x)dx = \int f(x)g_n(x)\,dx$$

One possibility: choose test functions to be Dirac delta functions. This is the collocation method.

$$g_n(x) = \delta(x - n/2^j) \qquad \text{n integer}$$

$$\Rightarrow \boxed{\sum_k c[k]\phi''_{j,k}(n/2^j) = f(n/2^j)} \quad \text{-------------------------------}\square$$

Second possibility: choose test functions to be scaling functions.

- Galerkin method if synthesis functions are used (test functions = trial functions)
- Petrov-Galerkin method if analysis functions are used

e.g. Petrov-Galerkin

$$g_n(x) = \tilde{\phi}_{j,n}(x) \quad \in \quad \tilde{V}_j$$

$$\Rightarrow \boxed{\sum_k c[k] \int_{-\infty}^{\infty} \frac{\partial^2}{\partial x^2} \phi_{j,k}(x) \,.\, \tilde{\phi}_{j,n}(x)\, dx = \int_{-\infty}^{\infty} f(x)\tilde{\phi}_{j,n}(x)\, dx}$$

Note: Petrov-Galerkin $\equiv$ Galerkin in orthogonal case

Two types of integrals are needed:

(a) Connection Coefficients

$$\int_{-\infty}^{\infty} \frac{\partial^2}{\partial x^2}\phi_{j,k}(x) \,.\, \tilde{\phi}_{j,n}(x)dx = 2^{2j}\int_{-\infty}^{\infty} 2^{j/2}\phi''(2^jx - k)2^{j/2}\tilde{\phi}(2^jx - n)dx$$

$$= 2^{2j}\int_{-\infty}^{\infty} \phi''(\tau)\tilde{\phi}(\tau + k - n)\, d\tau$$

$$= 2^{2j}h_{\partial^2/\partial x^2}[n - k]$$

where $h_{\partial^2/\partial x^2}[n]$ is defined by

$$\boxed{h_{\partial^2/\partial x^2}[n] = \int_{-\infty}^{\infty} \phi''(t)\tilde{\phi}(t - n)dt} \text{ ------------------ } \square$$

$\uparrow$

connection coefficients

(b) Expansion coefficients

The integrals $\int_{-\infty}^{\infty} f(x)\tilde{\phi}_{j,n}(x)dx$ are the coefficents for the expansion of f(x) in V_j.

$$f_j(x) \;=\; \sum_k r_j[k]\,\phi_{j,k}(x) \qquad \text{-------------------------------}\square$$

with

$$\boxed{r_j[k] \;=\; \int_{-\infty}^{\infty} f(x)\,\tilde{\phi}_{j,k}(x)\,dx} \qquad \text{-------------------------------}\square$$

So we can write the system of Galerkin equations as a convolution:

$$\boxed{2^{2j}\sum_k c[k]h_{\partial^2/\partial x^2}[n-k] \;=\; r_j[n]} \quad \text{-------------------}\square$$

$\Rightarrow$Solve a deconvolution problem to find c[k] and then find u_{approx} using equation □.

Note: we must allow for the fact that the solution may be non-unique, i.e. $H_{\partial^2/\partial x^2}(\omega)$ may have zeros.

Familiar example: 3-point finite difference operator

$$h_{\partial^2/\partial x^2}[n] \ = \ \{1, -2, 1\}$$

$$H_{\partial^2/\partial x^2}(z) \ = \ 1 - 2z^{-1} + z^{-2} \ = \ (1 - z^{-1})^2$$

$\Rightarrow$ $H_{\partial^2/\partial x^2}(\omega)$ has a 2nd order zero at $\omega = 0$.

Suppose $u_0(x)$ is a solution. Then $u_0(x) + Ax + B$ is also a solution. Need boundary conditions to fix $u_{approx}(x)$.

Determination of Connection Coefficients

$$h_{\partial^2/\partial x^2}[n] = \int_{-\infty}^{\infty} \phi''(t)\,\tilde{\phi}(t-n)\,dt$$

Simple numerical quadrature will not converge if $\phi''(t)$ behaves badly.

Instead, use the refinement equation to formulate an eigenvalue problem.

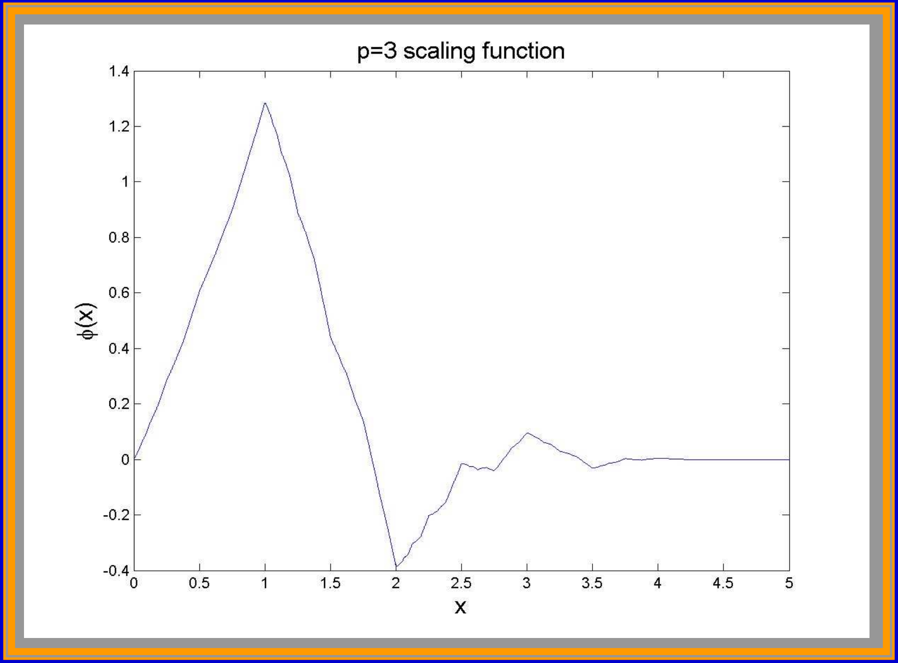
$$\phi(t) = 2\sum_k f_0[k]\phi(2t-k)$$

$$\left.\begin{aligned} \phi''(t) &= 8\sum_k f_0[k]\phi''(2t-k) \\ \tilde{\phi}(t-n) &= 2\sum_\ell h_0[\ell]\tilde{\phi}(2t-2n-\ell) \end{aligned}\right\}$$

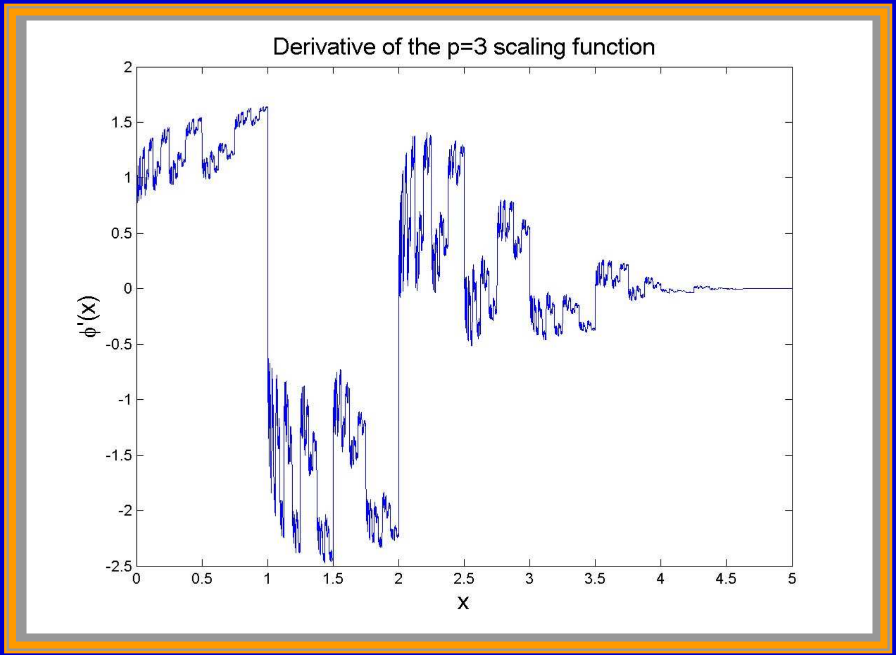
Multiply and Integrate

So

$$h_{\partial^2/\partial x^2}[n] = 8\sum_k f_0[k]\sum_\ell h_0[\ell]h_{\partial^2/\partial x^2}[2n+\ell-k]$$



Daubechies 6 scaling function



First derivative of Daubechies 6 scaling function

Reorganize as

$$h_{\partial^2/\partial x^2}[n] \; = \; 8\sum_{m} h_0[m-2n]\Big(\sum_{k} f_0[m-k]h_{\partial^2/\partial x^2}[k]\Big)$$

$$m \; = \; 2n + \ell$$

Matrix form

$$h_{\partial^2/\partial x^2} \; = \; 8\; A\; B\; h_{\partial^2/\partial x^2} \longrightarrow \text{eigenvalue problem}$$

Need a normalization condition $\longrightarrow$ use the moments of the scaling function:

If $h_0[n]$ has at least 3 zeros at π, we can write

$$\sum_{k} \mu_2[k]\phi(t-k) \; = \; t^2 \;\; ; \;\; \mu_2[k] \; = \int_{-\infty}^{\infty} t^2\tilde{\phi}(t-k)dt$$

Differentiate twice, multiply by $\tilde{\phi}(t)$ and integrate:

$$\sum_{k} \mu_2[k]h_{\partial^2/\partial x^2}[-k] \; = \; 2! \longrightarrow \text{Normalizing condition}$$

Formula for the moments of the scaling function

$$\mu_k^{\ell} \,\square\, \int_{-\infty}^{\infty} \tau^{\ell}\phi(\tau - k)d\tau$$

Recursive formula

$$\mu_0^0 = \int_{-\infty}^{\infty} \phi(\tau)d\tau = 1$$

$$\mu_0^r = \frac{1}{2^r - 1} \sum_{i=0}^{r-1} \binom{r}{i} \left(\sum_{k=0}^{N} h_0[k]k^{r-i}\right)\mu_0^i$$

$$\mu_k^{\ell} = \sum_{r=0}^{\ell} \binom{\ell}{r} k^{\ell-r} \mu_0^r$$

How to enforce boundary conditions?

One idea – extrapolate a polynomial:

$$u(x) = \sum_{k} c[k]\phi_{j,k}(x) = \sum_{\ell=0}^{p-1} a[\ell]x^{\ell}$$

Relate c[k] to a[ℓ] through moments. Extend c[k] by extending underlying polynomial.

Extrapolated polynomial should satisfy boundary constraints:

Dirichlet:

$$u(x_0) = \alpha \Rightarrow \sum_{\ell=0}^{p-1} a[\ell]x_0^{\ell} = \alpha$$

Neumann:

$$u'(x_0) = \beta \Rightarrow \sum_{\ell=0}^{p-1} a[\ell]\ell x_0^{\ell-1} = \beta$$

Constraint on a[ℓ]

Convergence

Synthesis scaling function:

$$\phi(x) = 2 \sum_k f_0[k]\phi(2x - k)$$

We used the shifted and scaled versions, $\phi_{j,k}(x)$, to synthesize the solution. If $F_0(\omega)$ has p zeros at π, then we can exactly represent solutions which are degree p – 1 polynomials.

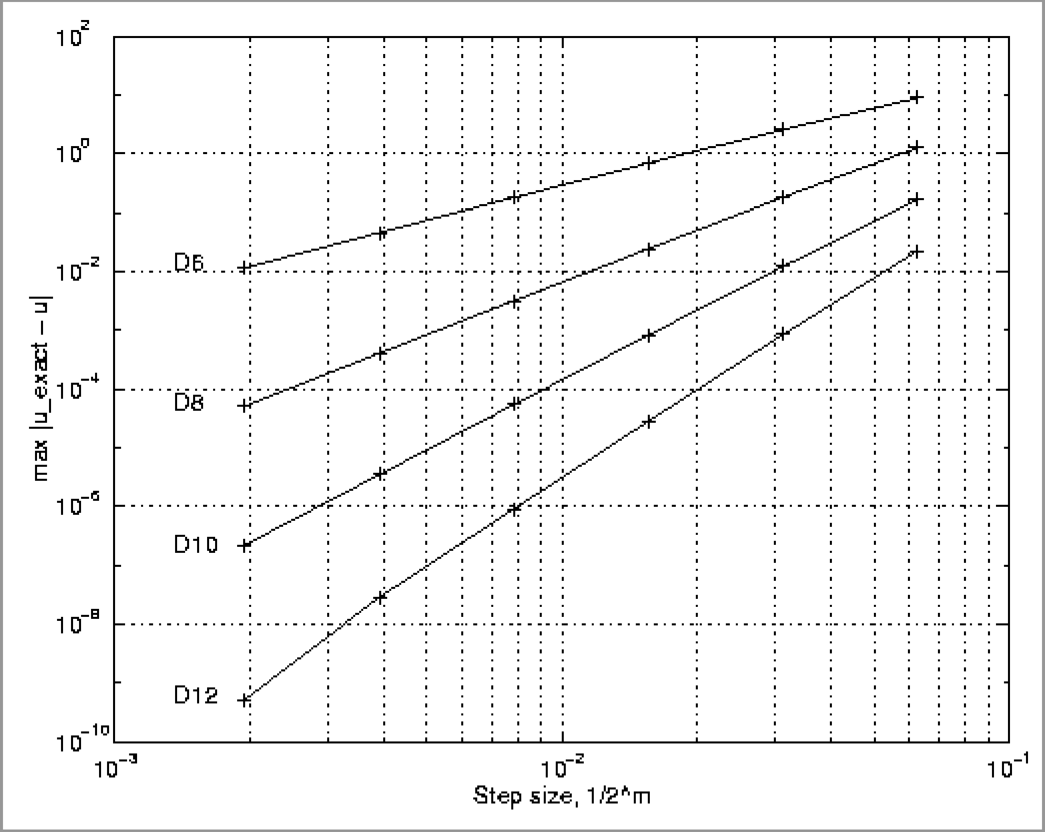
In general, we hope to achieve an approximate solution that behaves like

$$u(x) = \sum_k c[k]\phi_{j,k}(x) + O(h^p)$$

where

$$h = \frac{1}{2^j} = \text{spacing of scaling functions}$$

Reduction in error as a function of h



Multiscale Representation

e.g. $\partial^2 u/\partial x^2 = f$

Expand as

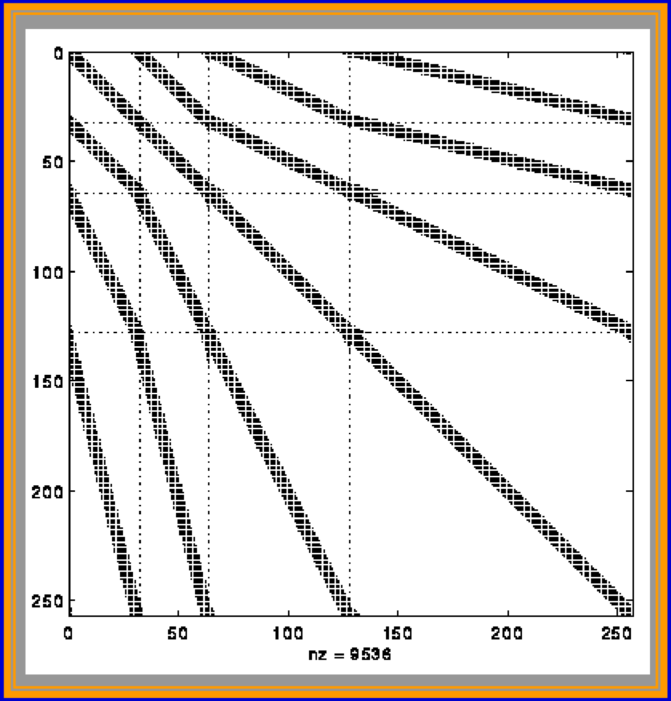
$$u = \sum_k c_k \phi(x - k) + \sum_{j=0}^{J} \sum_k d_{j,k} w(2^j x - k)$$

Galerkin gives a system

$$Ku = f$$

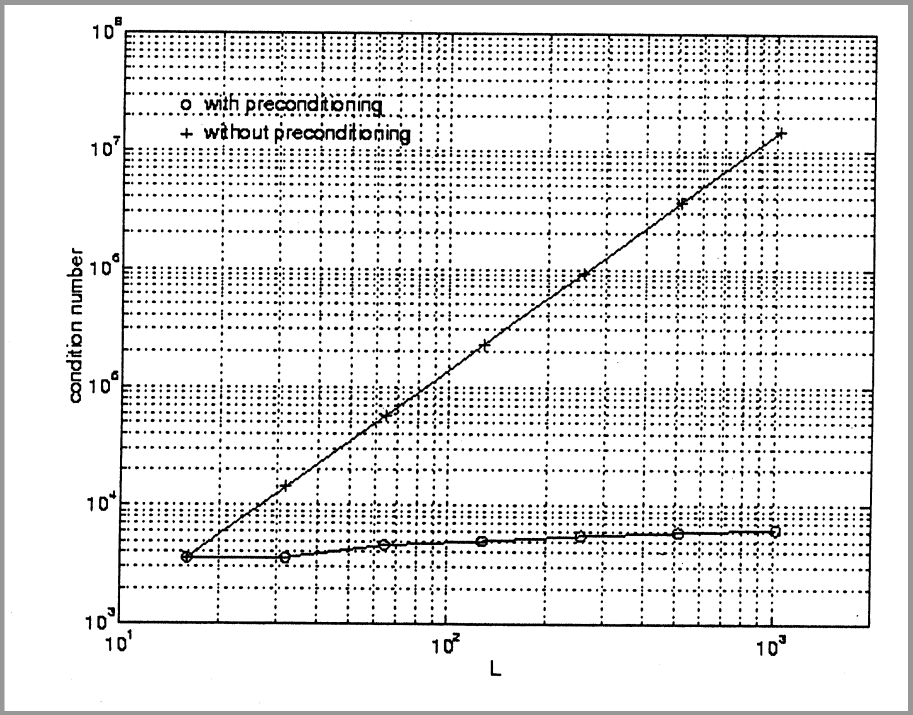
with typical entries

$$K_{m,n} = 2^{2j} \int_{-\infty}^{\infty} \frac{\partial^2}{\partial x^2} w(x - n) w(x - m)\, dx$$



Effect of Preconditioner

- Multiscale equations: $(WKW^T)(Wu) = Wf$
- Preconditioned matrix: $K_{prec} = DWKW^TD$



Simple diagonal preconditioner

$$D = \begin{bmatrix} 1 &&&&&&& \\ & 1 &&&&&& \\ && \tfrac{1}{2} &&&&& \\ &&& \tfrac{1}{2} &&&& \\ &&&& \tfrac{1}{4} &&& \\ &&&&& \tfrac{1}{4} && \\ &&&&&& \tfrac{1}{4} \end{bmatrix}$$

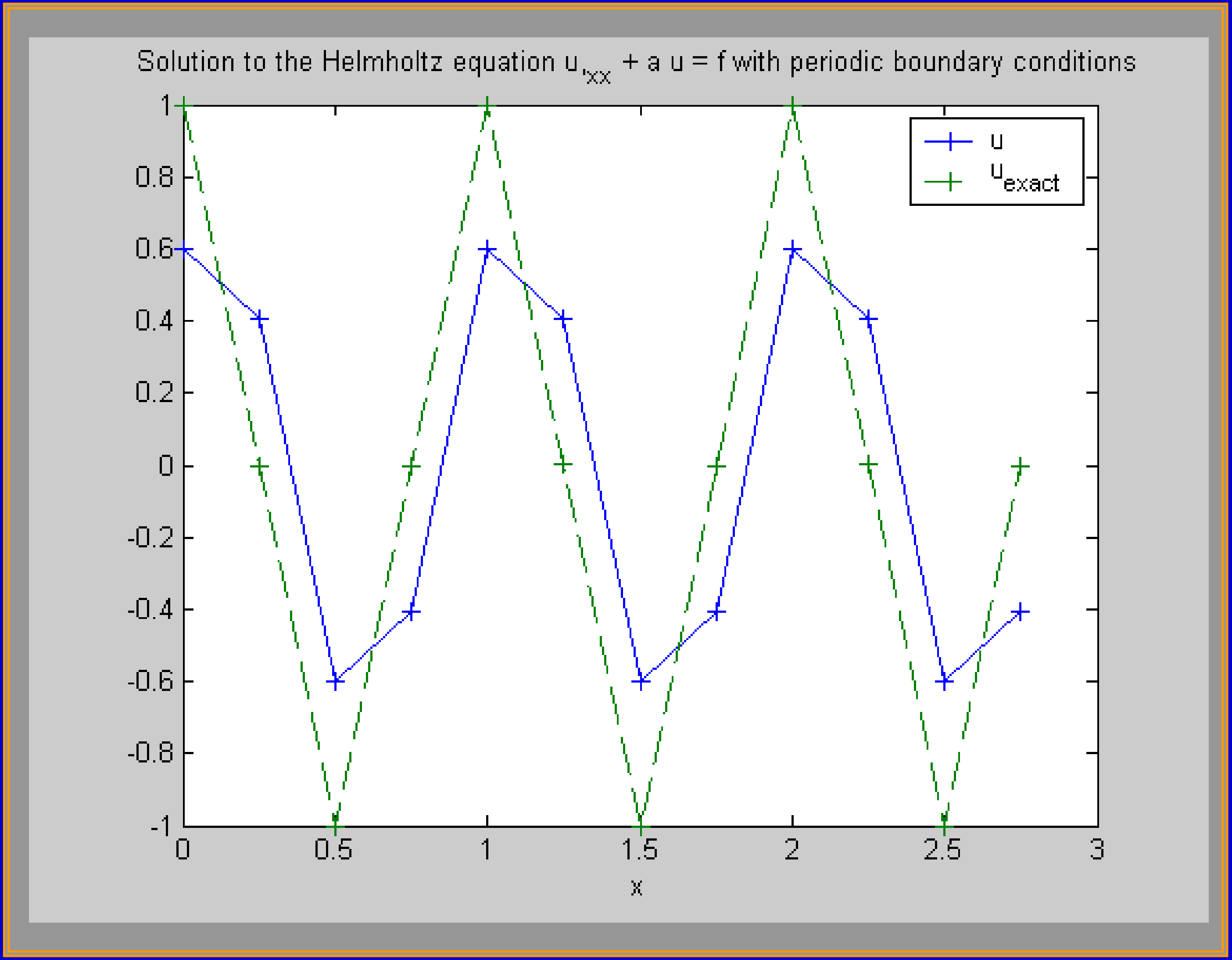
Matlab Example

Numerical solution of Partial Differential Equations

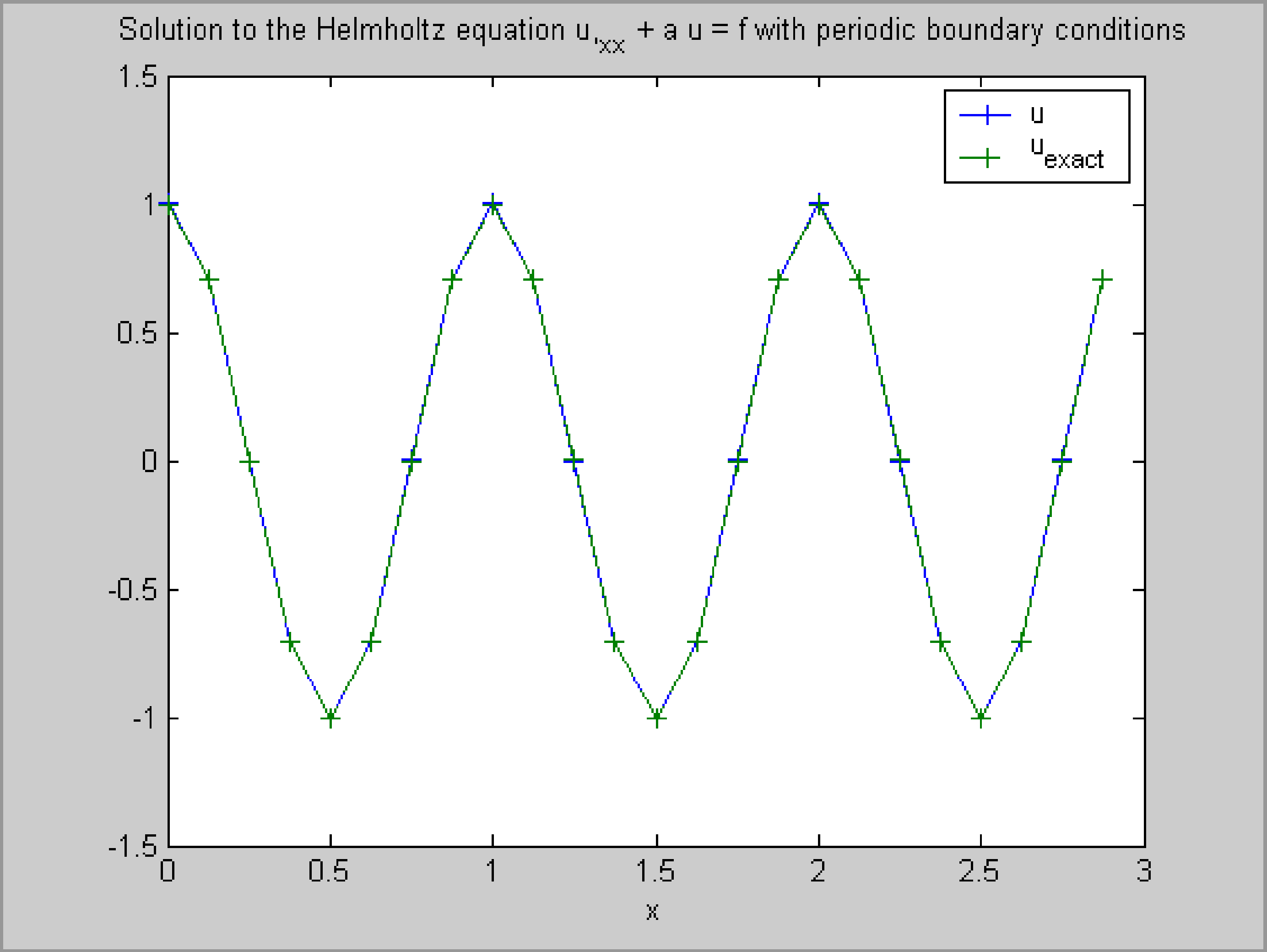
The Problem

1. Helmholtz equation: $u_{xx} + a\, u = f$
   - *p=6;* % Order of wavelet scheme (p_{min}=*3*)
   - *a = 0*
   - *L = 3*; % Period.
   - *nmin = 2*; % Minimum resolution
   - *nmax = 7*; % Maximum resolution

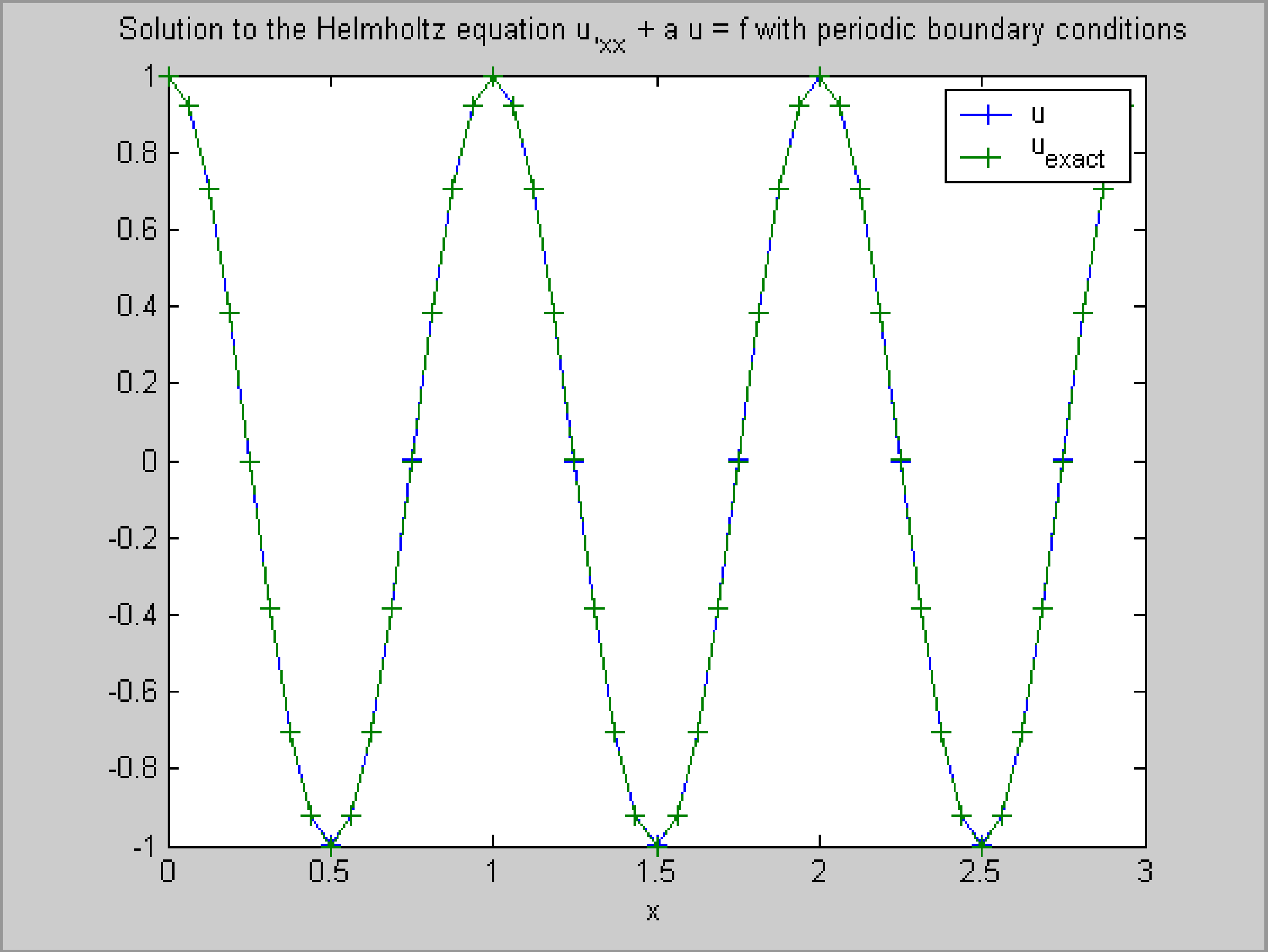
Solution at Resolution 2



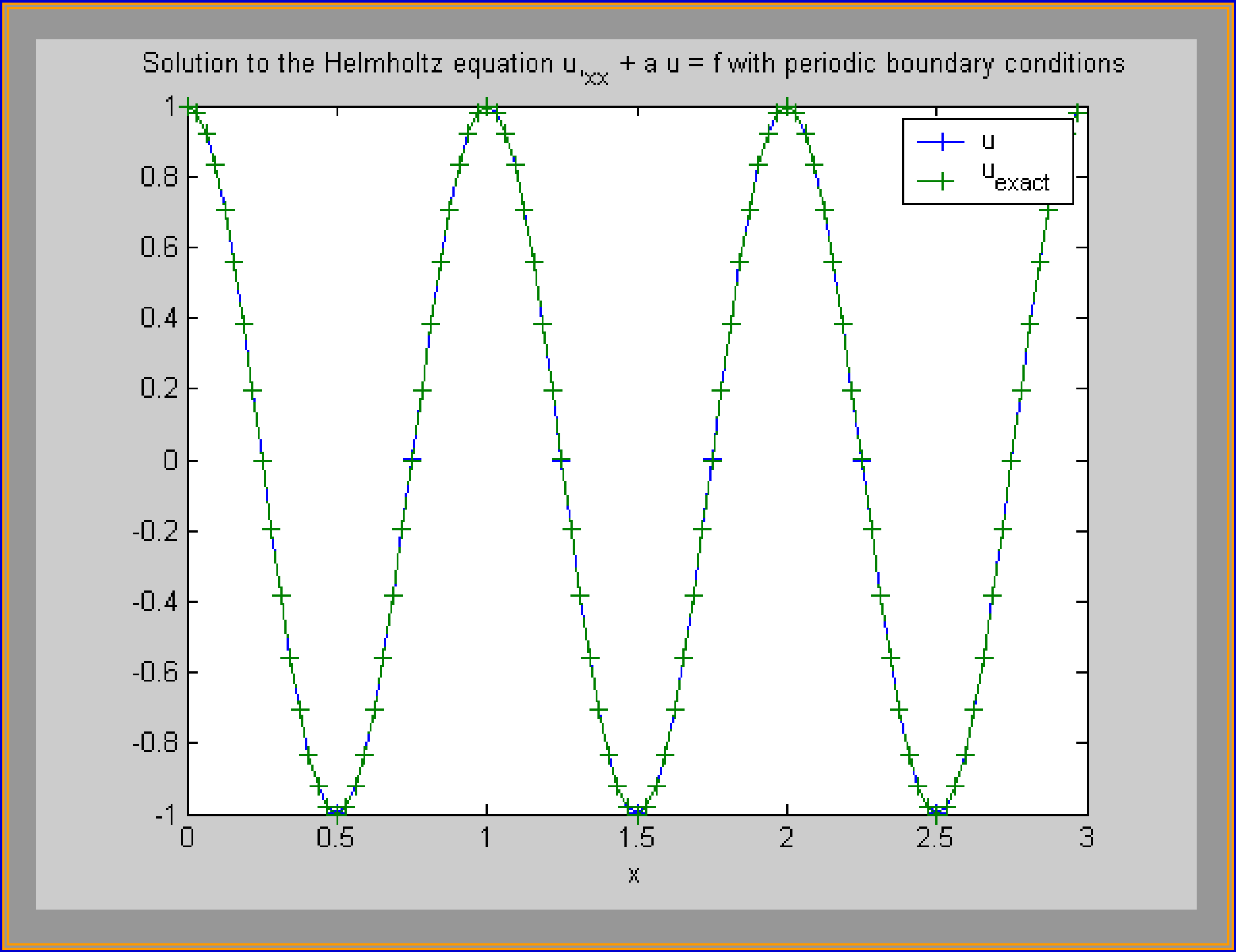
Solution at Resolution 3



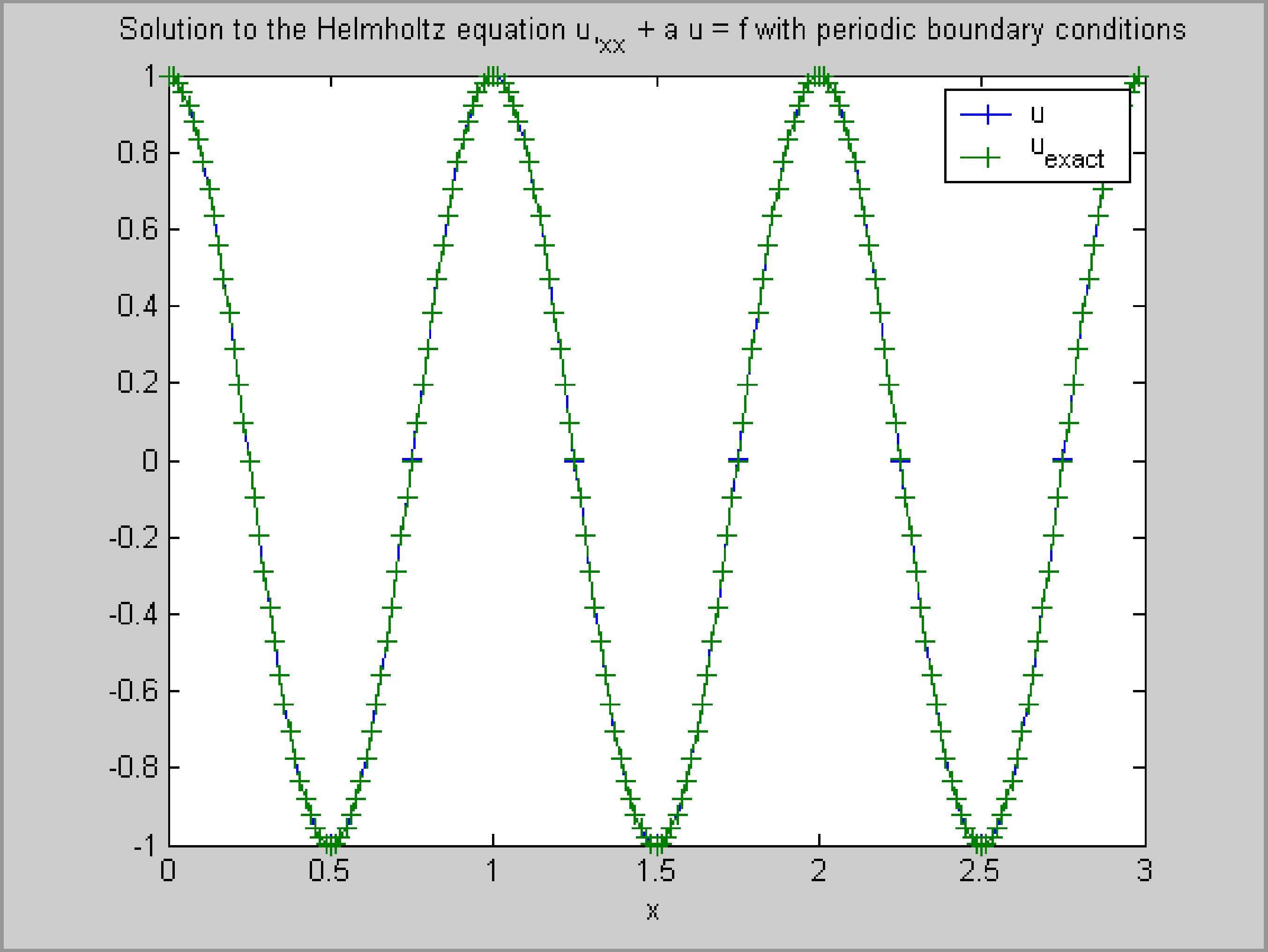
Solution at Resolution 4



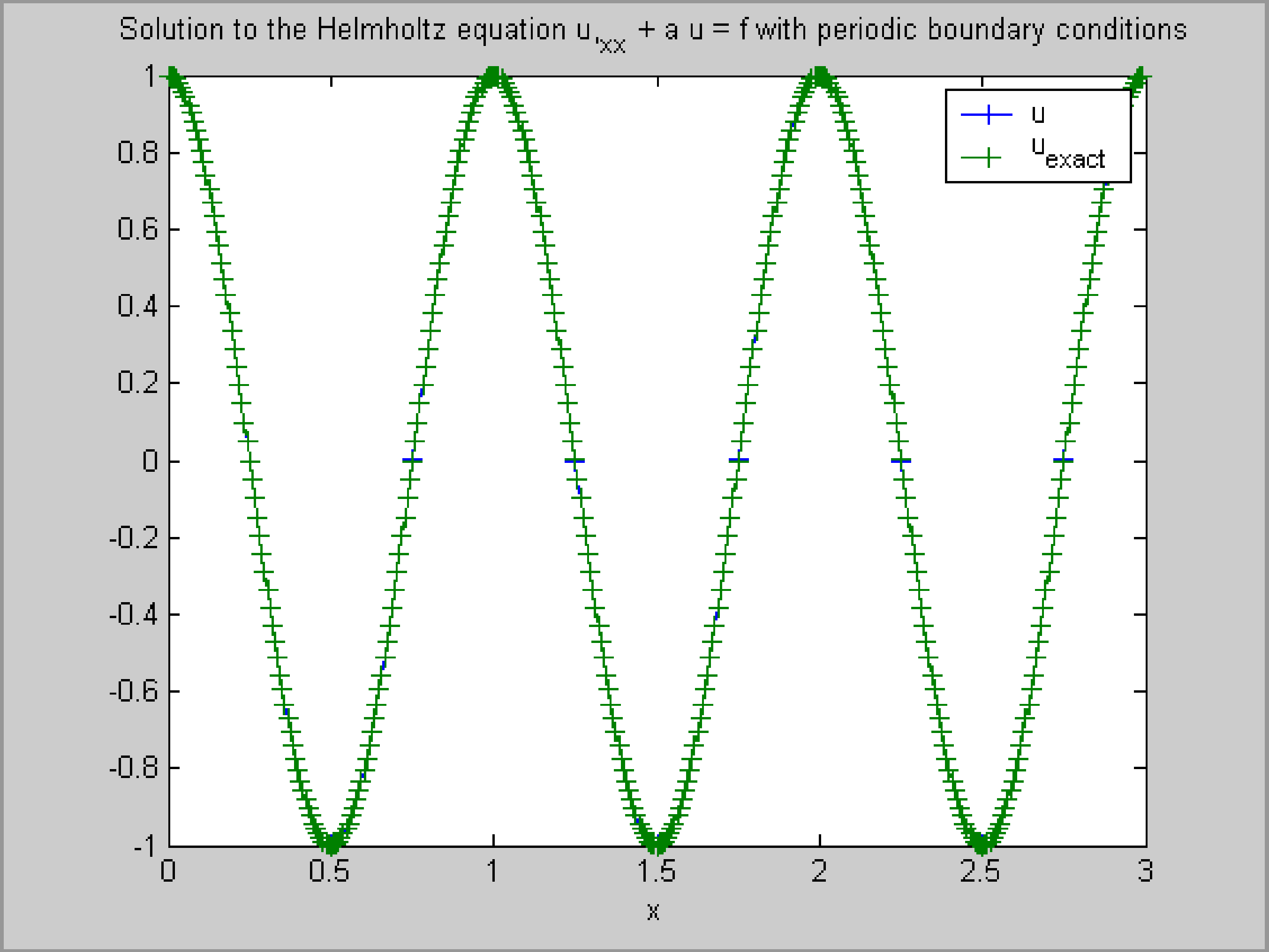
Solution at Resolution 5



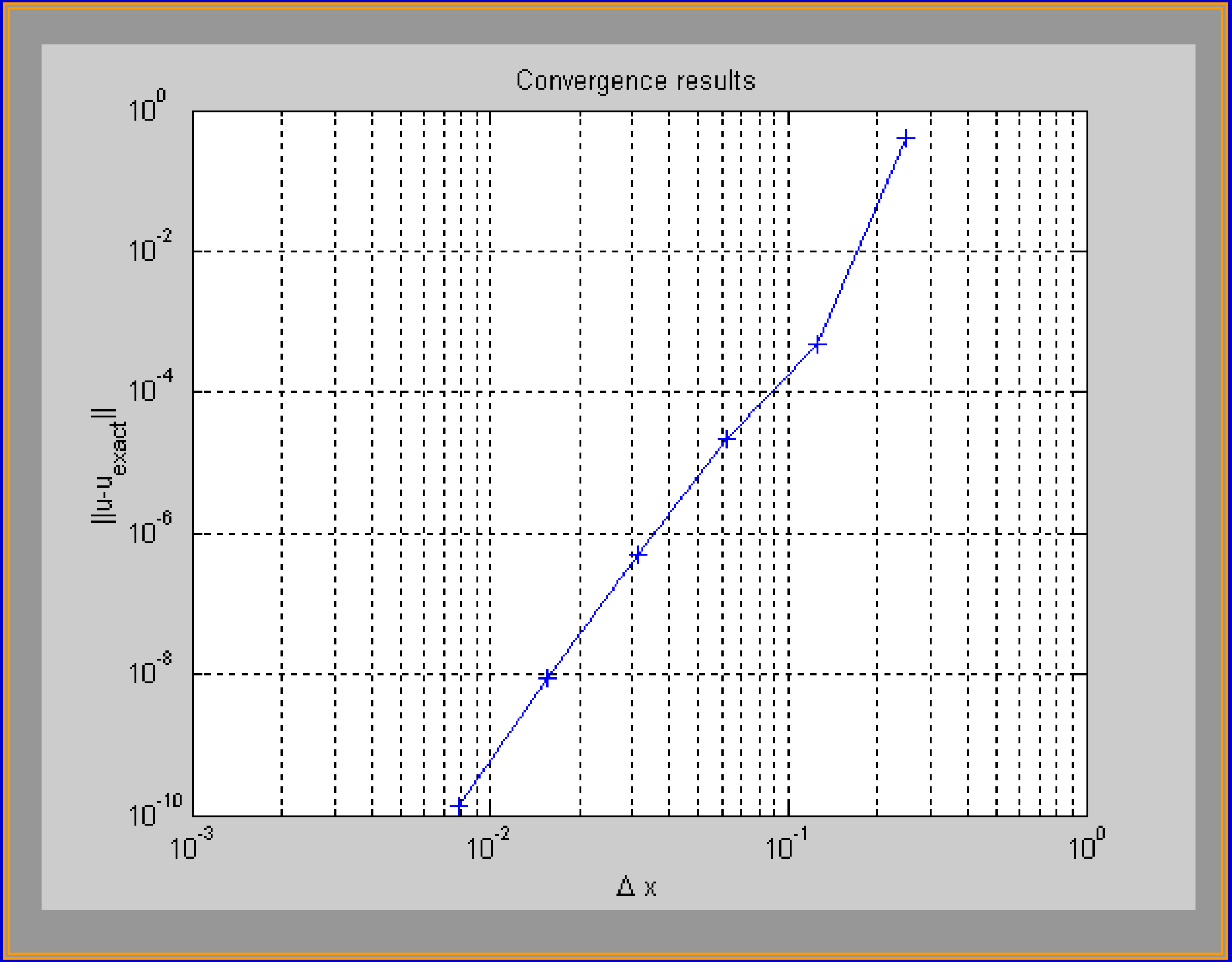
Solution at Resolution 6



Solution at Resolution 7



Convergence Results



```
>> helmholtz slope =  5.9936
```