Introduction to function approximation

- En las primeras clases nos hemos dedicado a ver la resolución de PDE's mediante diferencias finitas (FDM)?
- Ahora buscaremos abordar el mismo problema pero desde la utilización del método de los residuos ponderados.
- Ellos se basan en resolver formas integrales que representan la PDE's pero buscando una solución que la satisfaga en un sentido promediado y no puntualmente.
- Para ello necesitamos aproximar funciones numéricamente que contienen coeficientes de ajustes que se determinan buscando satisfacer el MRP.

Approximation of functions by continuous trial functions

- Continuous approximation means using function with global support
- Approximation by trial functions
  - Point fitting of functions – Lagrange interpolation
  - Fourier sine series
- Approximation by weighted residual
  - Point collocation
  - Subdomain collocation
  - Galerkin
  - Petrov-Galerkin (Least square)?

Approximation by trial functions

- Deseamos aproximar una dada funcion “phi” en un dominio “Omega” acotado por una curva cerrada “Gamma”
- Normalmente estas funciones se requieren que adopten ciertos valores en el contorno, de ahi la incorporación de ls funcion “psi”.
- La aproximacion la haremos mediante una familia de funciones “N(x)” llamadas funciones de prueba o de forma (trial or shape)
- lo que resta por determinar son los coeficientes de ajuste “a”

f a continuous function defined over Ω

$\Omega \in \Re^n$ with $\Gamma \in \Re^{n-1}$ its boundary

$$f \cong \hat{f} = y\,(x) + \sum_m a_m N_m(x) \quad \text{(completeness requirement)}$$

$$\{N_m ; m = 1,2,3,\cdots\} \text{such that } N_m\big|_\Gamma = 0$$

$$y\,\big|_\Gamma = f\big|_\Gamma$$

Approximation by trial functions - Completeness

- El uso de “psi” garantiza que “phi(Gamma) = phi_hat(Gamma)?
- La adecuada selección de la aproximacion se basa en criterios de *“completitud”,* es decir, la aproximacion mejora con M->infty
- Veamos el siguiente ejemplo:

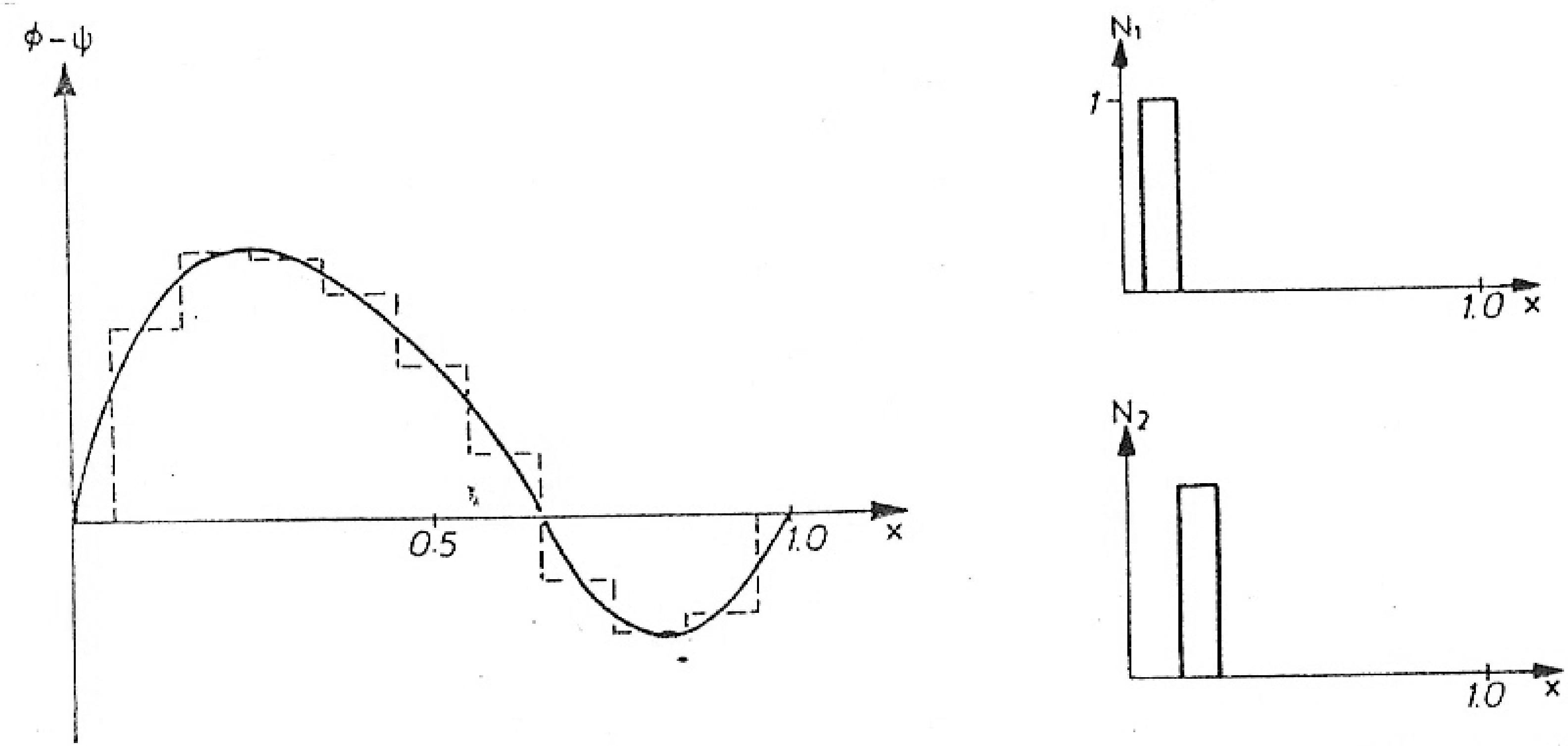


FIGURE 2.1. Discontinuous trial functions possessing the completeness property over $0 \leqslant x \leqslant 1$.

Approximation by trial functions - Completeness

- Las funciones N_m(x) son funciones de soporte local, es decir son no nulas en una pequeña porcion del domino (0<=x<=1)?
- Refinando la aproximacion (M->infty o achicando el soporte de las funciones) la aproximacion mejora gracias al requisito de completitud satisfecho por esta funciones aproximantes.

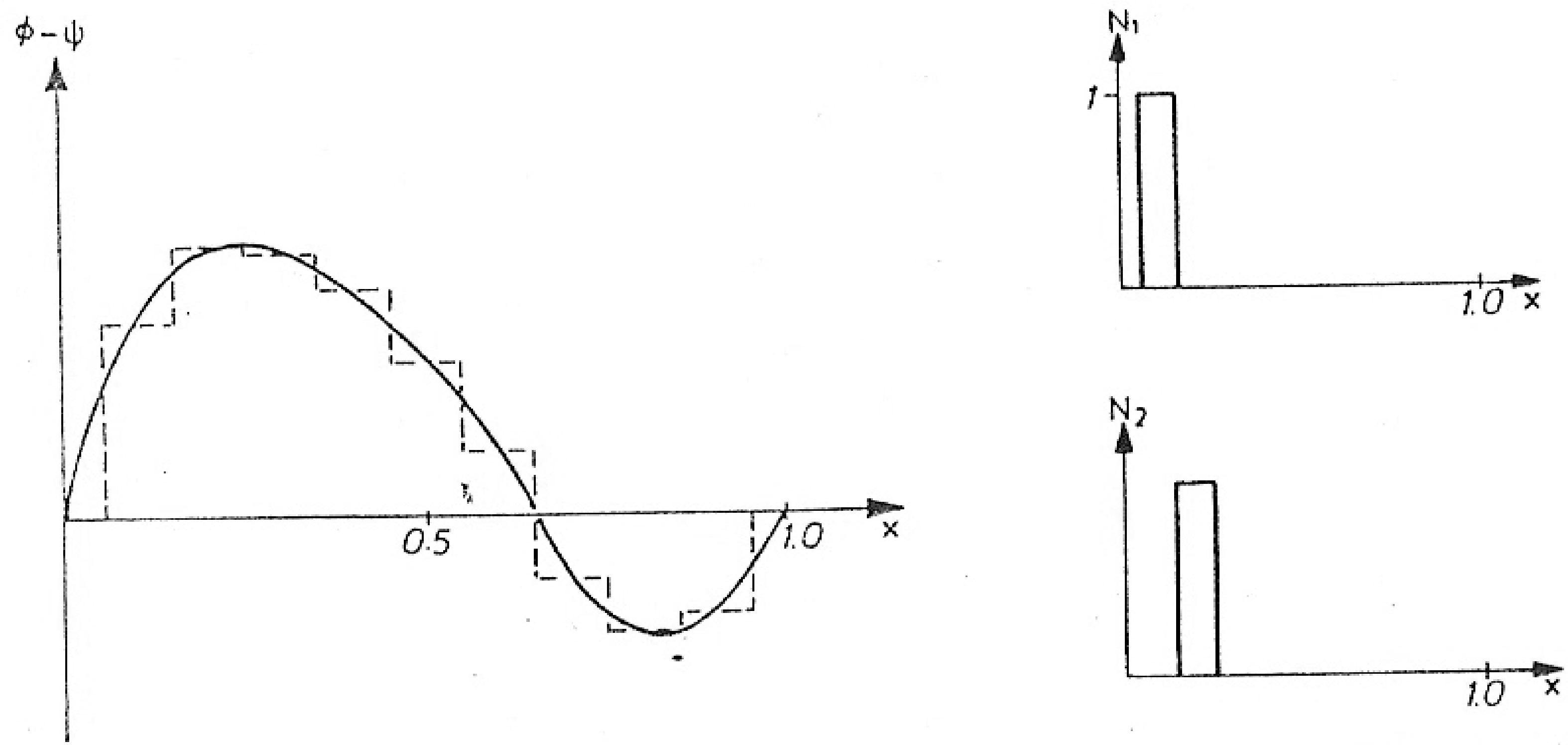


FIGURE 2.1. Discontinuous trial functions possessing the completeness property over $0 \leqslant x \leqslant 1$.

Approximation by trial functions - Completeness

- Si bien la motivación principal del curso es el uso de funciones de soporte compacto para empezar haremos una introduccion usando funciones de soporte global.
- El motivo es que este tipo de aproximación ya ha sido estudiado en cursos previos y permite al alumno entender donde esta parado
- Ademas le permite contar con una visión mas general, unificada y poderosa en temas de aproximación, tanto de funciones como de soluciones a PDE's.

Point fitting of functions – Lagrange interpolation

El objetivo es comenzar con una forma especial de aproximación
Aquella inspirada en que la aproximante “phi_hat” coincida con la función a aproximar “phi” en ciertos puntos del dominio “Omega”
Este requisito conduce a un sistema algebraico lineal de ecuaciones expresado en función de parámetros incógnitas a_m

$$\boxed{\hat{f}(x_k) = f(x_k) \quad ; \quad k = 1,2,\cdots,M}$$

$$\hat{f}(x) = y(x) + \sum_m a_m N_m(x)$$

$$\sum_m a_m N_m(x_k) = \hat{f}(x_k) - y(x_k) = f(x_k) - y(x_k)$$

$$K_{km} a_m = f_k \quad \Rightarrow a_m = \left(K^{-1} f\right)_m$$

$$K_{km} = N_m(x_k)$$

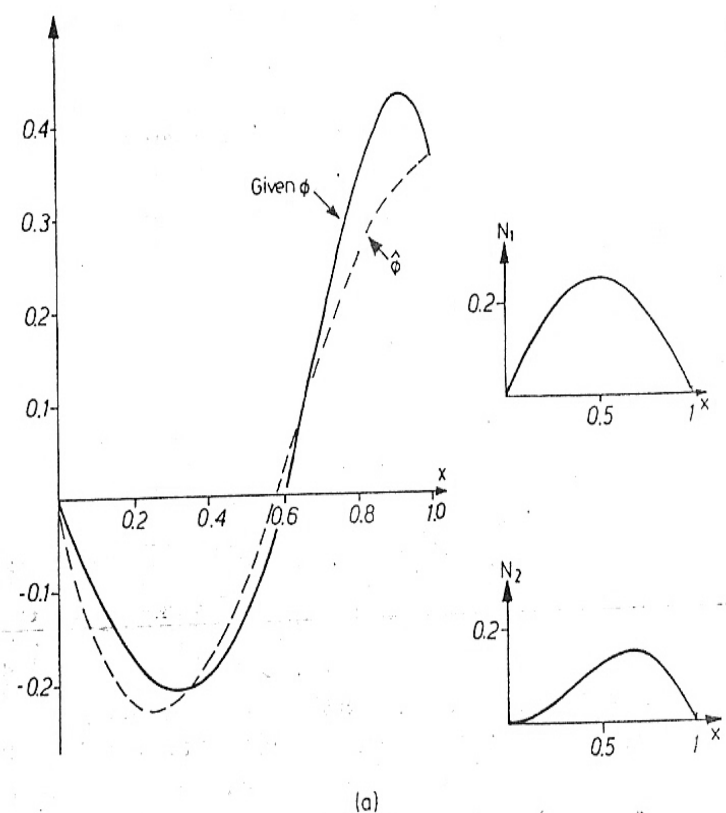
$$f_k = f(x_k) - y(x_k)$$

Point fitting of functions – Lagrange interpolation

Comenzamos aproximando mediante “point fitting” la función “phi” usando 2 diferentes funciones de interpolacion N_m(x)

una función potencial

una función trigonométrica



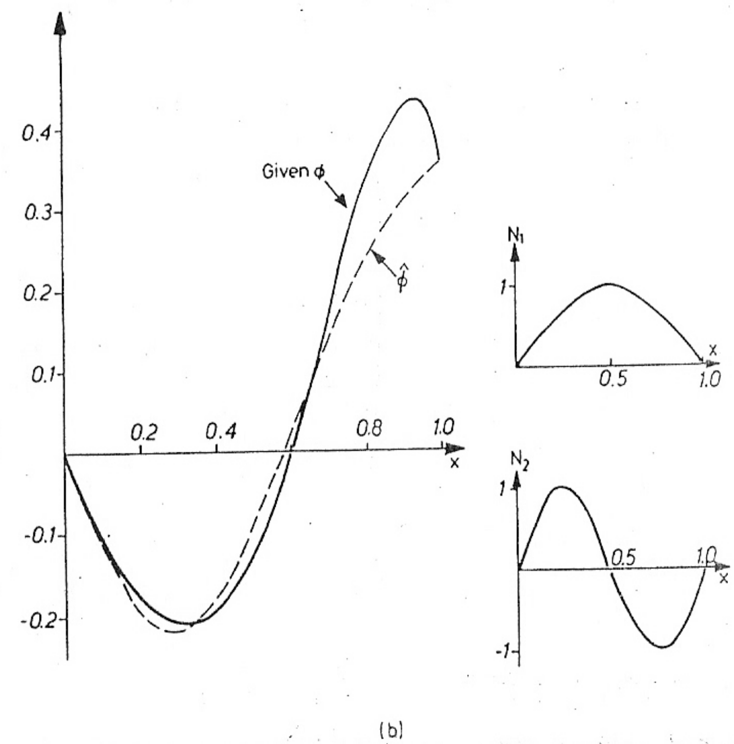
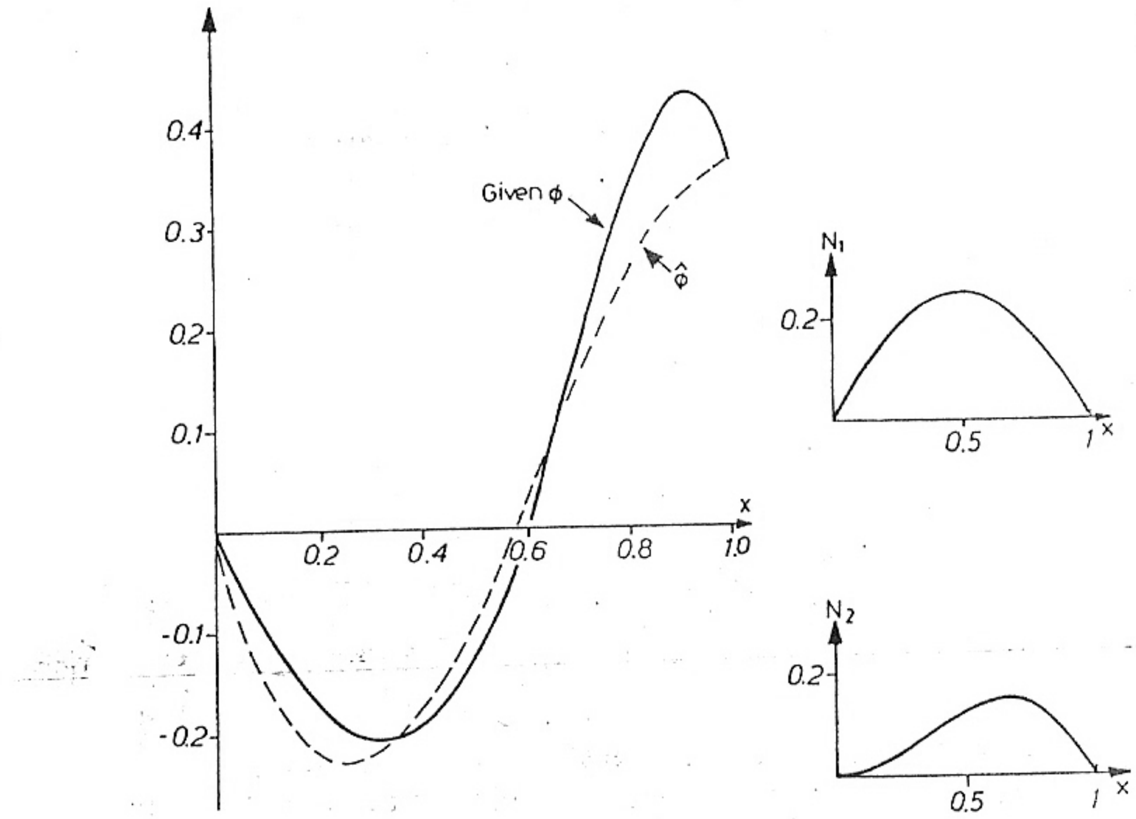


FIGURE 2.2. Trial function approximation of a given function ϕ using point fitting and the first two terms of the trial function sets (a) $\{N_m = x^m(1 - x);\ m = 1, 2, \ldots\}$ and (b) $\{N_m = \sin m\pi x;\ m = 1, 2, \ldots\}$.

Point fitting of functions – Lagrange interpolation

$$\begin{cases} N_m(x) = x^m(1-x) & \text{caso (a)} \\ N_m(x) = \sin(mp\ x) & \text{caso (b)} \end{cases}$$



(a)

Point fitting of functions – Lagrange interpolation

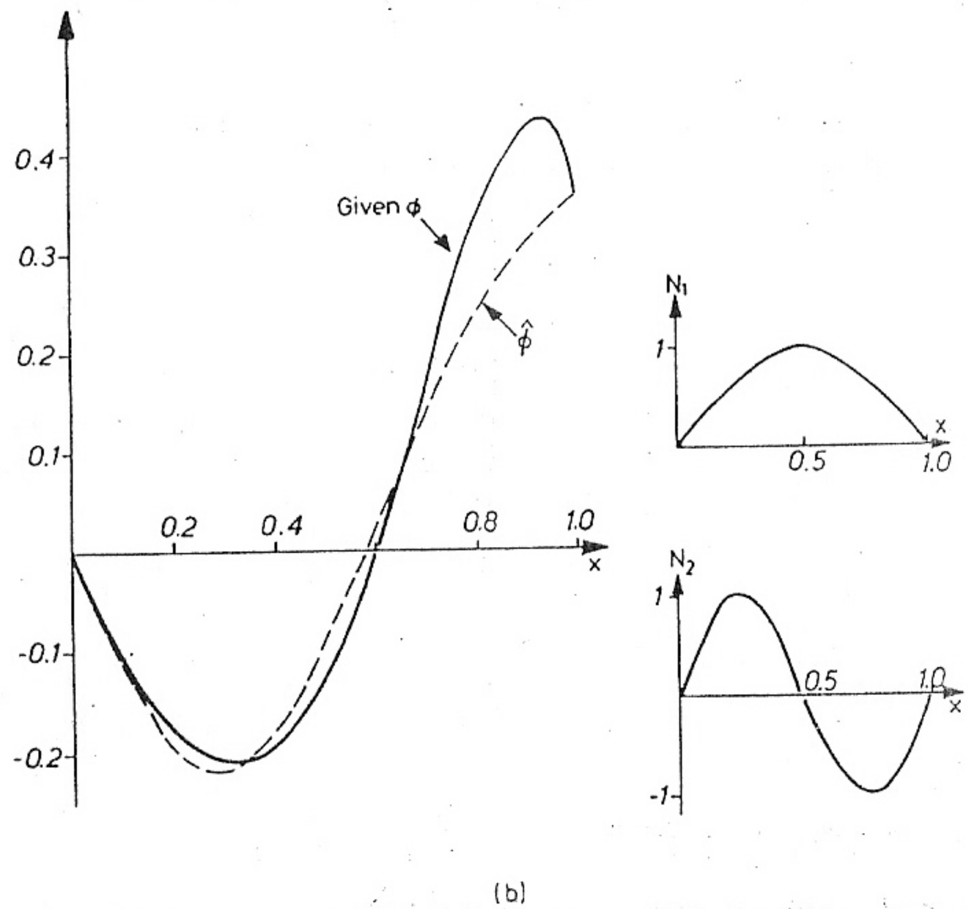


FIGURE 2.2. Trial function approximation of a given function ϕ using point fitting and the first two terms of the trial function sets (a) $\{N_m = x^m(1 - x);\ m = 1, 2, \dots\}$ and (b) $\{N_m = \sin m\pi x;\ m = 1, 2, \dots\}$.

Fourier sine series

Es posible mediante Fourier aproximar funciones “phi(x)”
sobre un rango 0<=x<=Lx
de la forma que se muestra a continuacion
La teoría exige que para que la aproximacion sea buena la función a aproximar “phi” tenga a lo sumo un conjunto finito de discontinuidades y solo un número finito de máximos y mínimos locales en el rango de interés, requisito satisfecho por la mayoría de las funciones.
Otra vez “psi” satisface las aproximación en los extremos
Los coeficientes incógnitas a_m representan las amplitudes de los distintos modos de Fourier.

$$\hat{f}(x) = y\ (x) + \sum_m a_m N_m(x)$$

$$N_m(x) = \sin\left(\frac{mp\ \ x}{L_x}\right)$$

$$a_m = \frac{2}{L_x}\int_0^{L_x} \big(f(x) - y\ (x)\big)\sin\left(\frac{mp\ \ x}{L_x}\right)dx$$

Fourier sine series

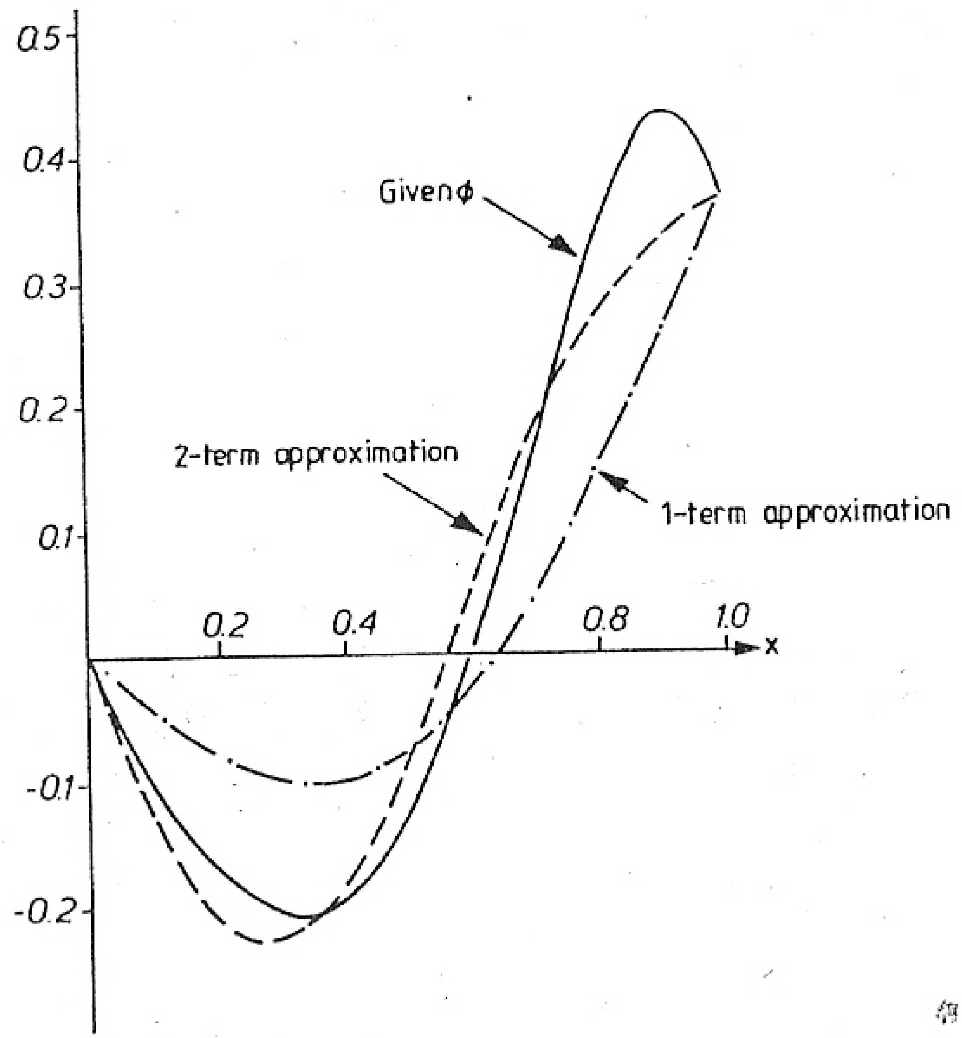


FIGURE 2.3. Truncated Fourier series used to approximate a given function.

Approximation by weighted residual

$$\int_{x} W_l R_\Omega dx = \int_{x} W_l \left(f - \hat{f} \right) dx$$

- Es un método que permite determinar los coeficientes a_m
- Veremos que los metodos de “point fitting” y “Fourier” son casos particulares de esta métodología mucho más general
- El método requiere definir el concepto de ***“residuo”*** como diferencia entre la solución aproximante (phi_hat) y la aproximada (phi)?
- Este residuo es una función de la posición en Omega
- El objetivo del método es reducir el error pero ahora en su sentido integral o global sobre todo el dominio y no en ciertos puntos.
- Para ello se recurre a escribir una condición de este tipo para diferentes funciones de peso que ponderan diferencialmente o bien zonas del dominio o ciertas características de las funciones
- W_l son un conjunto de funciones peso independientes con l=1,...M
- Convergencia requiere que phi_hat --> phi cuando M --> infty y esto equivale a que R_Omega --> 0 para todo x en Omega

Approximation by weighted residual

$$\int_{x} W_l R_\Omega dx = \int_{x} W_l \left(f - \hat{f}\right) dx$$

- Reemplazando phi_hat = psi + sum_m a_m N_m
- E igualando a cero la integral
- Conduce a un sistema lineal de ecuaciones algebraicas del tipo **K*a=f**
- Donde **a =** (a_1,a_2,a_3 , ... , a_M)?
- K_{lm} **=** int_Omega W_l N_m dOmega 1<=l,m<=M
- f_l = int_Omega W_l (phi-psi) dOmega 1<=l,m<=M
- **a** se determina especificando **psi(x)** y **N_m(x)** y **W_l(x)?**
- La selección del peso **W_l(x)** determina el método de aproximación

Point collocation

Weight function $W_l = d(x - x_l)$

$$\int_x W_l R_\Omega dx = \int_x d(x - x_l)\left(f - \hat{f}\right) = \longrightarrow \boxed{\textbf{equivalent to point fitting}}$$

$$= \int_x d(x - x_l)\left(f(x) - \left(y\ (x) + \sum_m a_m N_m(x) \right)\right) dx$$

$$\int_x d(x - x_l)\left(f(x) - y\ (x)\right) dx - \int_x d(x - x_l)\sum_m a_m N_m(x) dx = 0$$

$$\int_x d(x - x_l)\sum_m a_m N_m(x) dx = \int_x d(x - x_l)\left(f(x) - y\ (x)\right) dx$$

$$K_{lm}\ a_m\ =\ f_l$$

$$K_{lm} = \int_x d(x - x_l) N_m(x) dx = N_m(x_l)$$

$$f_l\ = \int_x d(x - x_l)\left(f(x) - y\ (x)\right) dx = f(x_l) - y\ (x_l)$$

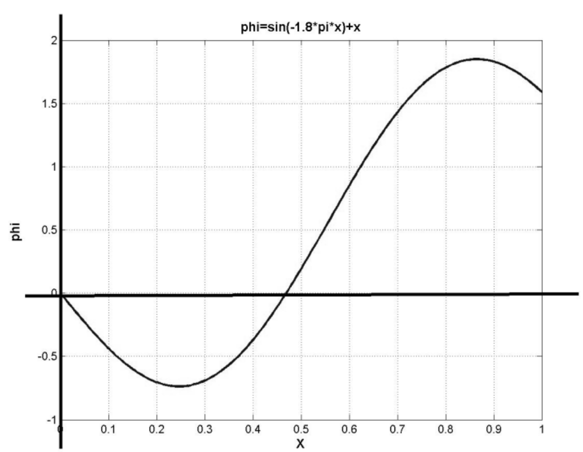
Point collocation

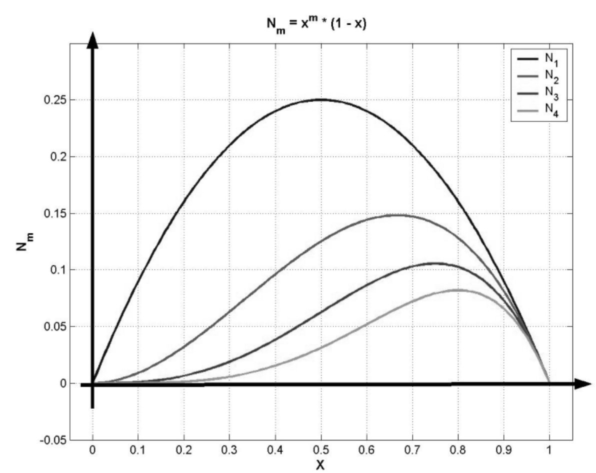
Function to be approximated $f = \sin(-1.8x) + x$

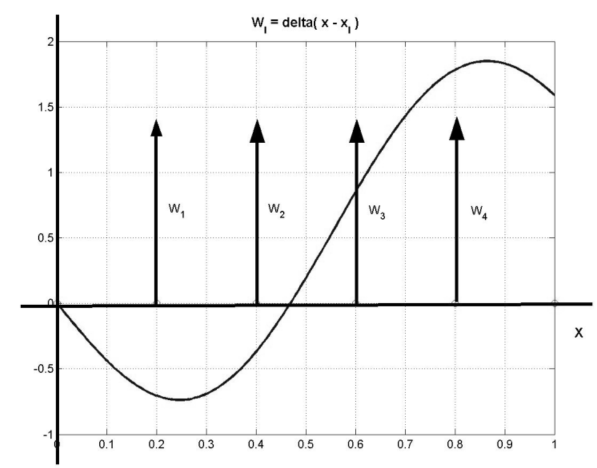
Shape function to be used $N_m = x^m(1-x)$

Weight function $W_l = d(x - x_l)$

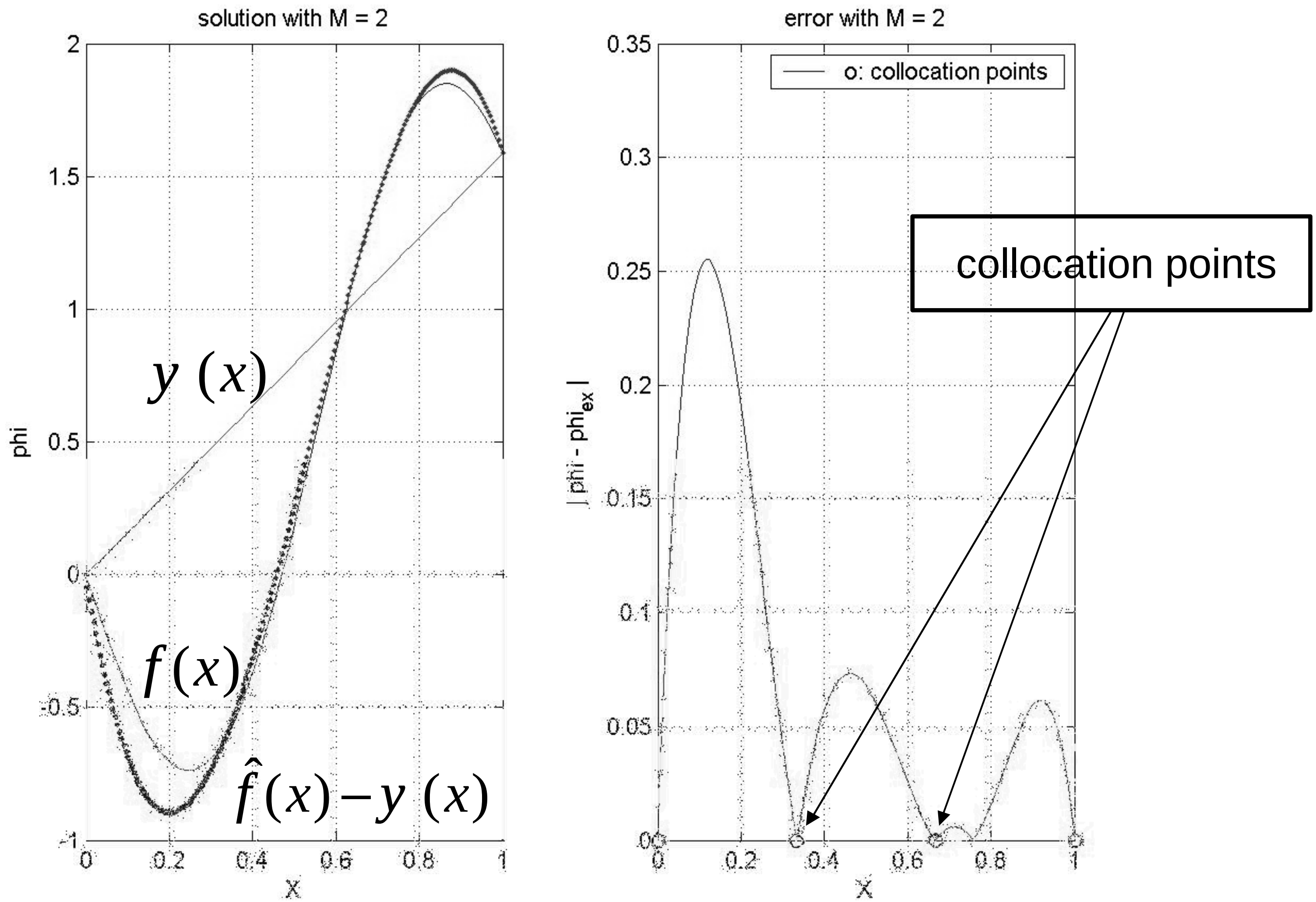
$$\int_x W_l R_\Omega dx = \int_{x=0}^{x=1} d(x-x_l)\left(f-\hat{f}\right) = \int_{x=0}^{x=1} d(x-x_l)\left(f(x) - \left(y(x) + \sum_m a_m N_m(x)\right)\right)dx$$



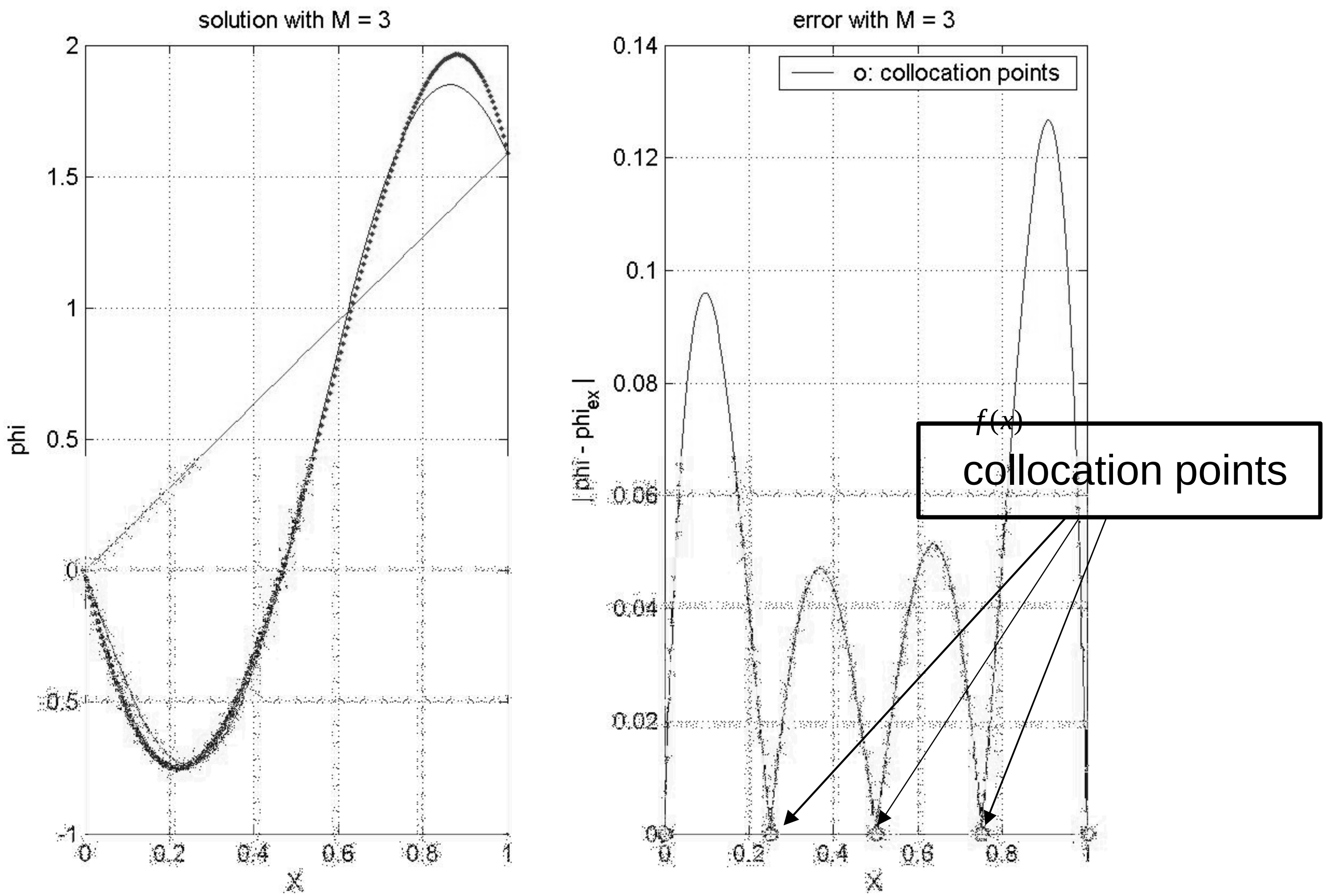




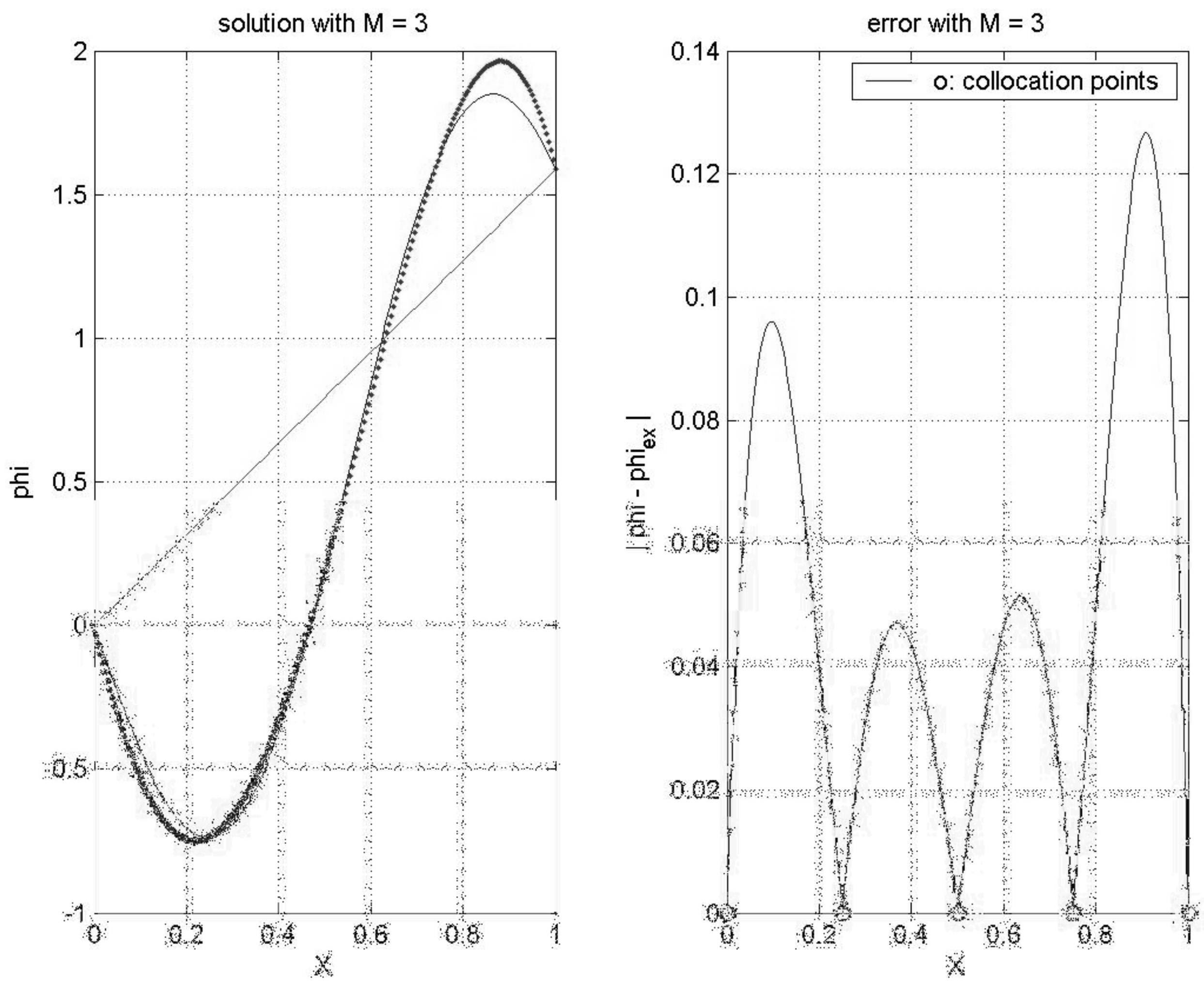
Point collocation



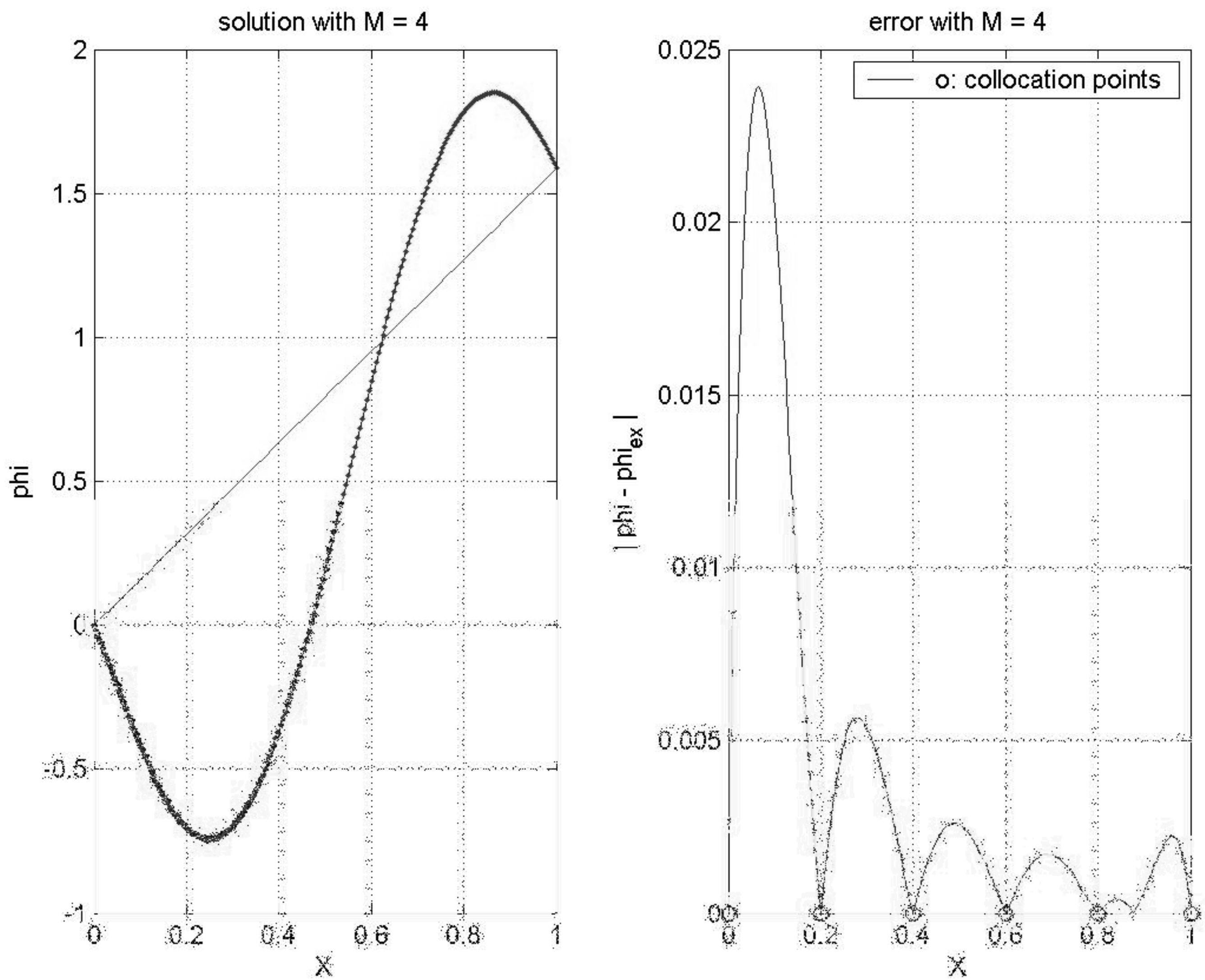
Approximation function by point collocation



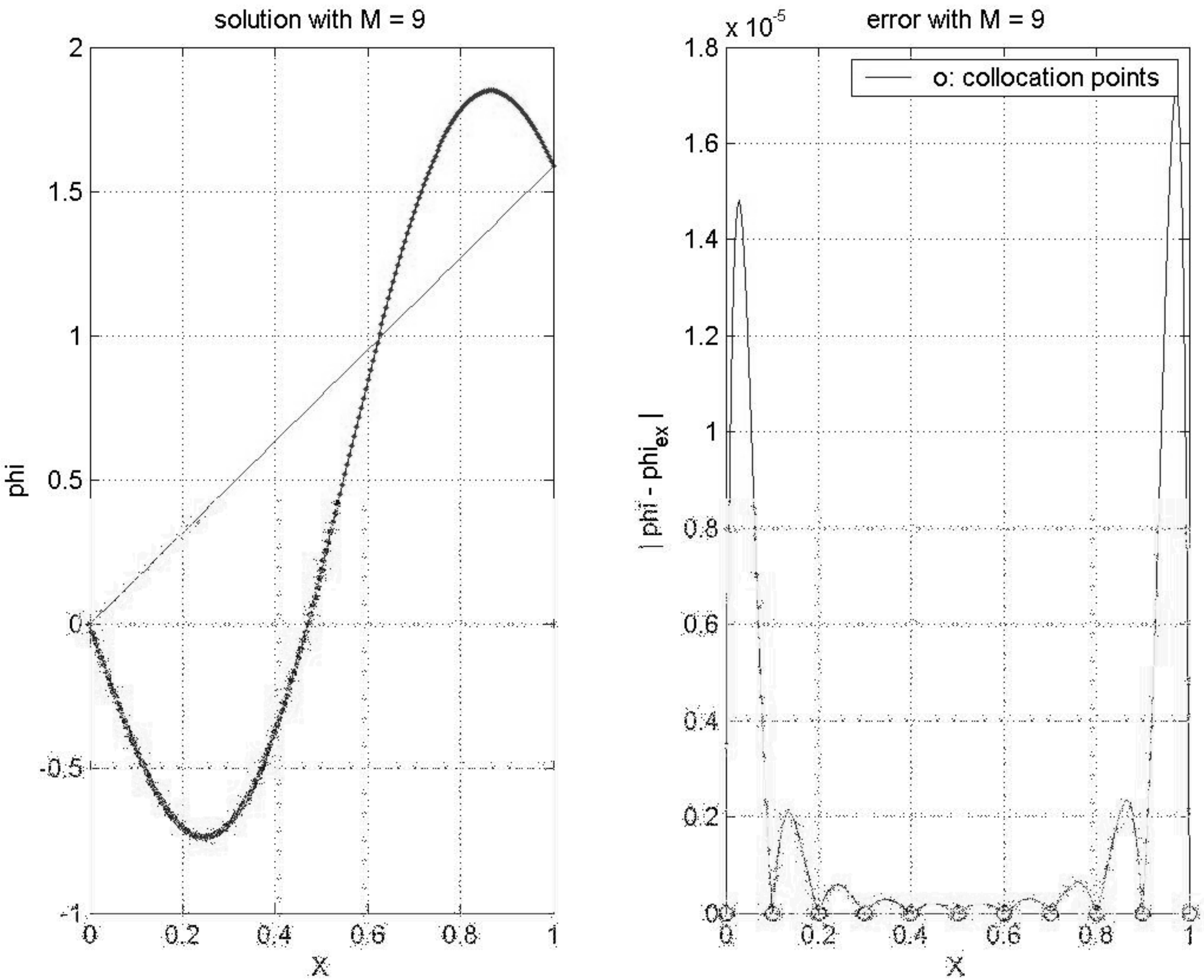
Approximation function by point collocation



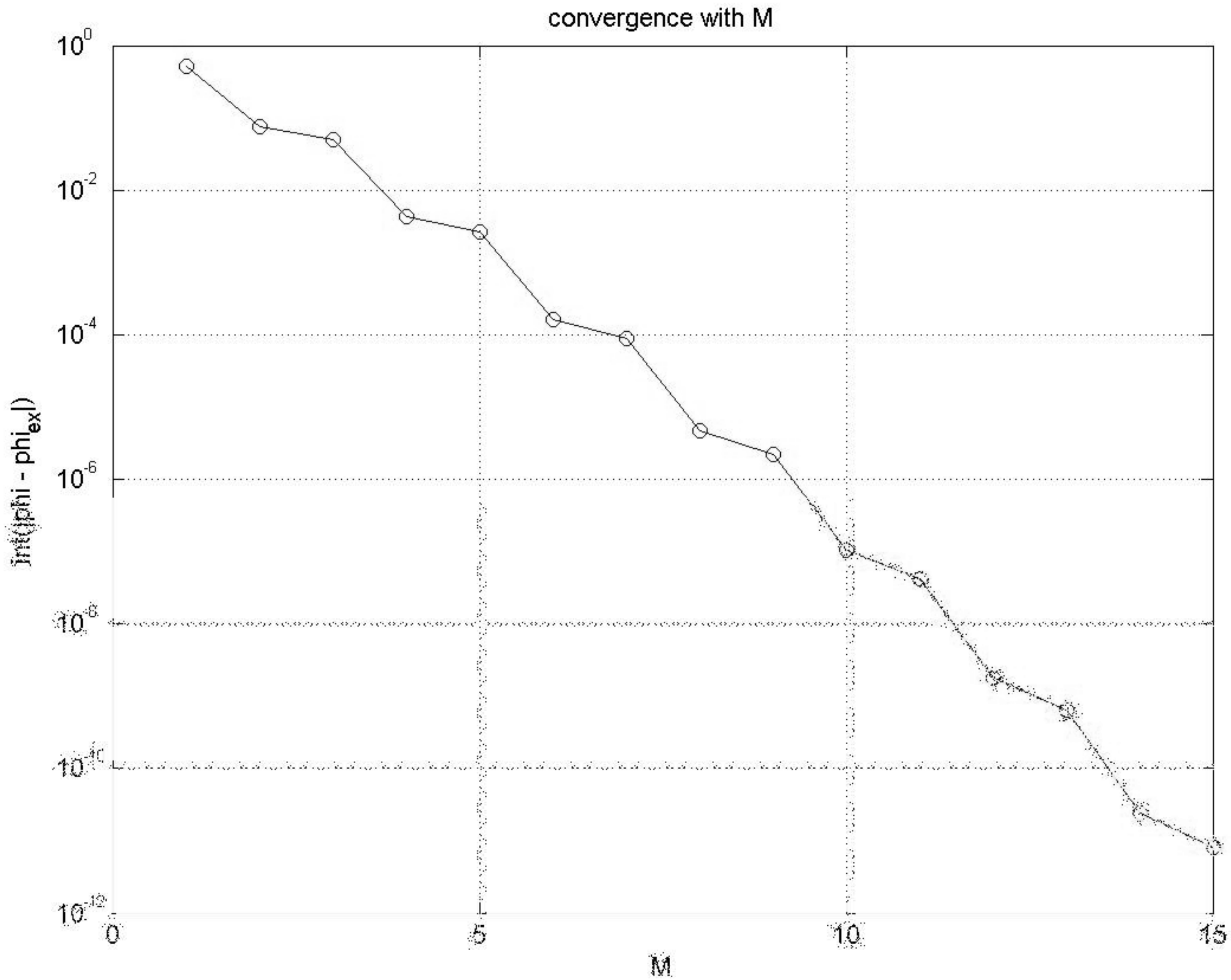
Approximation function by point collocation



Approximation function by point collocation



Approximation function
by point collocation



Subdomain collocation

Weight function $$W_l = c_{x_l}^{x_{l+1}} = \begin{cases} 1 & x_l < x < x_{l+1} \\ 0 & x < x_l \; , \; x > x_{l+1} \end{cases}$$

$$\int_x W_l R_\Omega dx = \int_x c_{x_l}^{x_{l+1}} \left(f - \hat{f} \right) =$$

$$= \int_x c_{x_l}^{x_{l+1}} \left(f(x) - \left(y\;(x) + \sum_m a_m N_m(x) \right) \right) dx$$

$$\int_x c_{x_l}^{x_{l+1}} \left(f(x) - y\;(x) \right) dx - \int_x c_{x_l}^{x_{l+1}} \sum_m a_m N_m(x) dx = 0$$

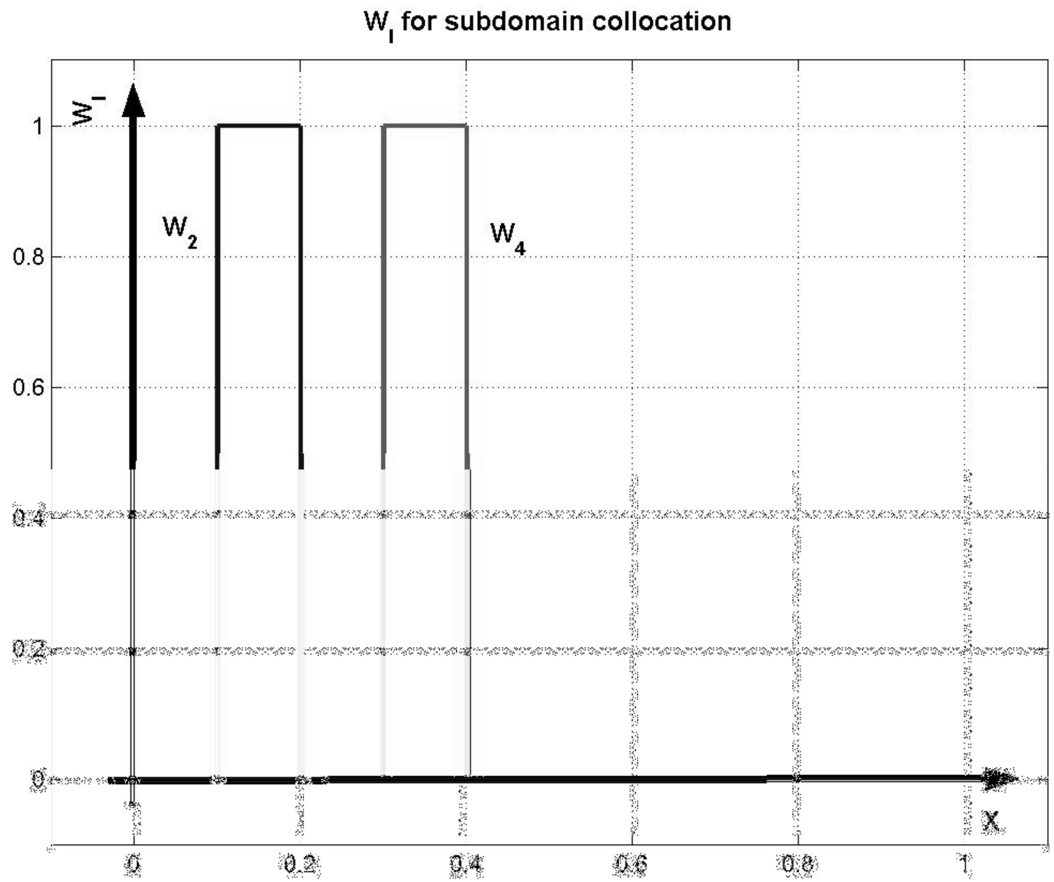
$$\int_{x_l}^{x_{l+1}} \sum_m a_m N_m(x) dx = \int_{x_l}^{x_{l+1}} \left(f(x) - y\;(x) \right) dx$$

$$K_{lm}\, a_m \;=\; f_l$$

$$K_{lm} = \int_{x_l}^{x_{l+1}} N_m(x) dx \qquad , \qquad f_l \;=\; \int_{x_l}^{x_{l+1}} \left(f(x) - y\;(x) \right) dx$$

Subdomain collocation

Weight function $$W_l = c_{x_l}^{x_{l+1}} = \begin{cases} 1 & x_l < x < x_{l+1} \\ 0 & x < x_l \ , x > x_{l+1} \end{cases}$$



Galerkin

Weight function $$W_l = N_l$$

$$\int_x W_l R_\Omega dx = \int_x W_l \left(f - \hat{f}\right) =$$

$$= \int_x W_l \left(f(x) - \left(y(x) + \sum_m a_m N_m(x) \right) \right) dx$$

$$\int_x N_l \left(f(x) - y(x)\right) dx - \int_x N_l \sum_m a_m N_m(x) dx = 0$$

$$\int_x N_l \sum_m a_m N_m(x) dx = \int_x N_l \left(f(x) - y(x)\right) dx$$

$$K_{lm} \, a_m = f_l$$

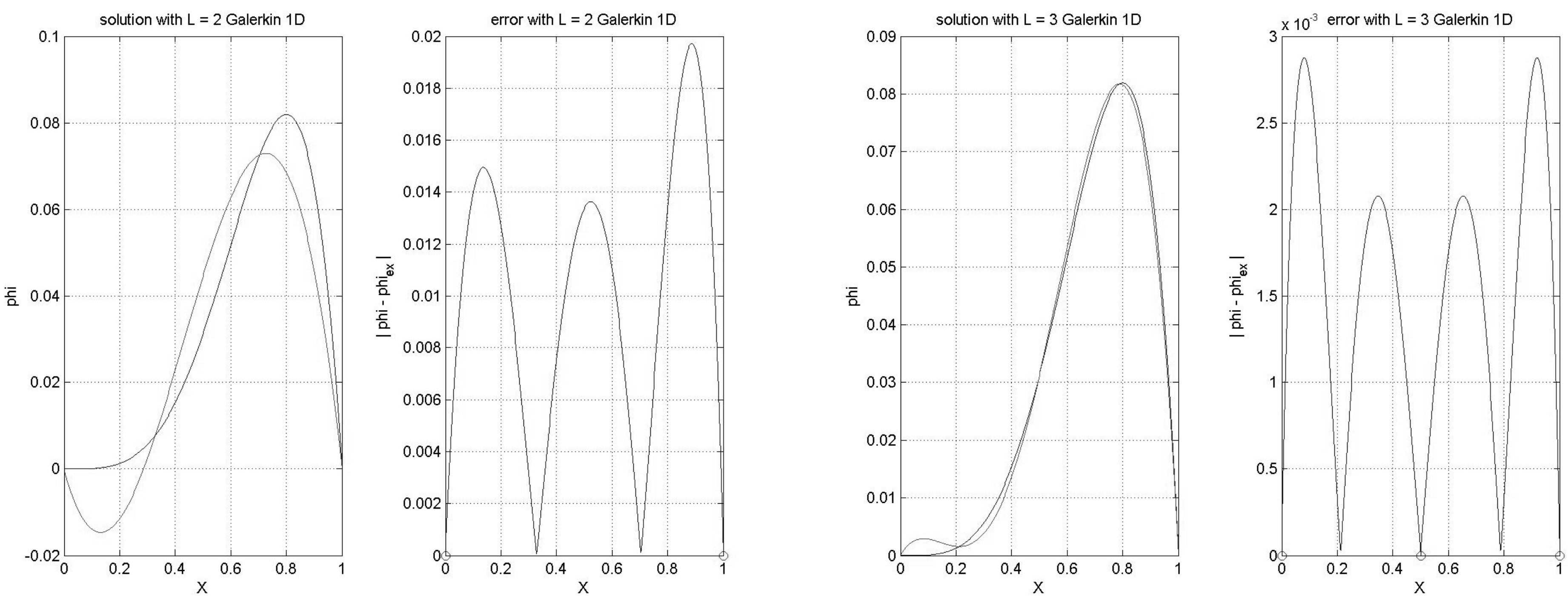
$$K_{lm} = \int_x N_l(x) \, N_m(x) dx \qquad , \qquad f_l = \int_x N_l(x) \left(f(x) - y(x)\right) dx$$

Galerkin in 1D approximations

Function to be approximated $f = x^4(1-x)$

Shape function to be used $N_m = x^m(1-x)$

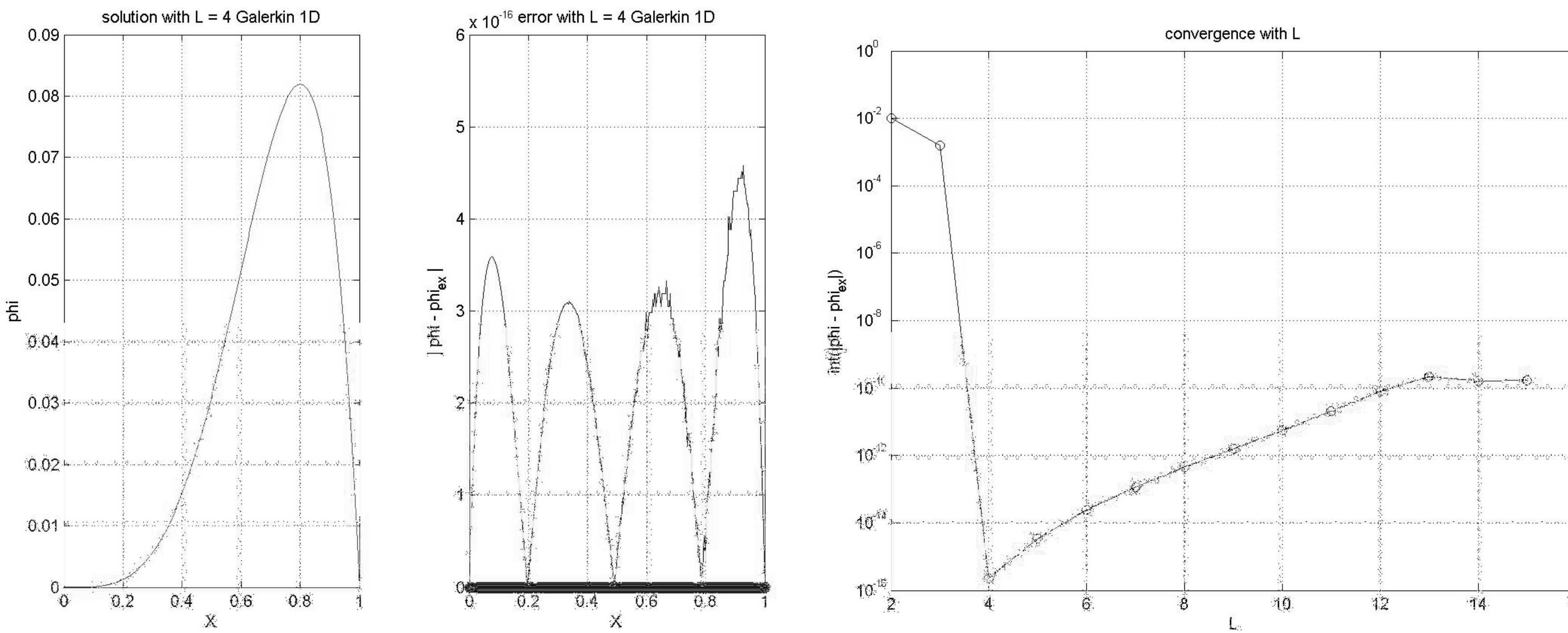
Weight function $W_l = N_l$



Galerkin in 1D approximations

Function to be approximated $$f = x^4(1-x)$$

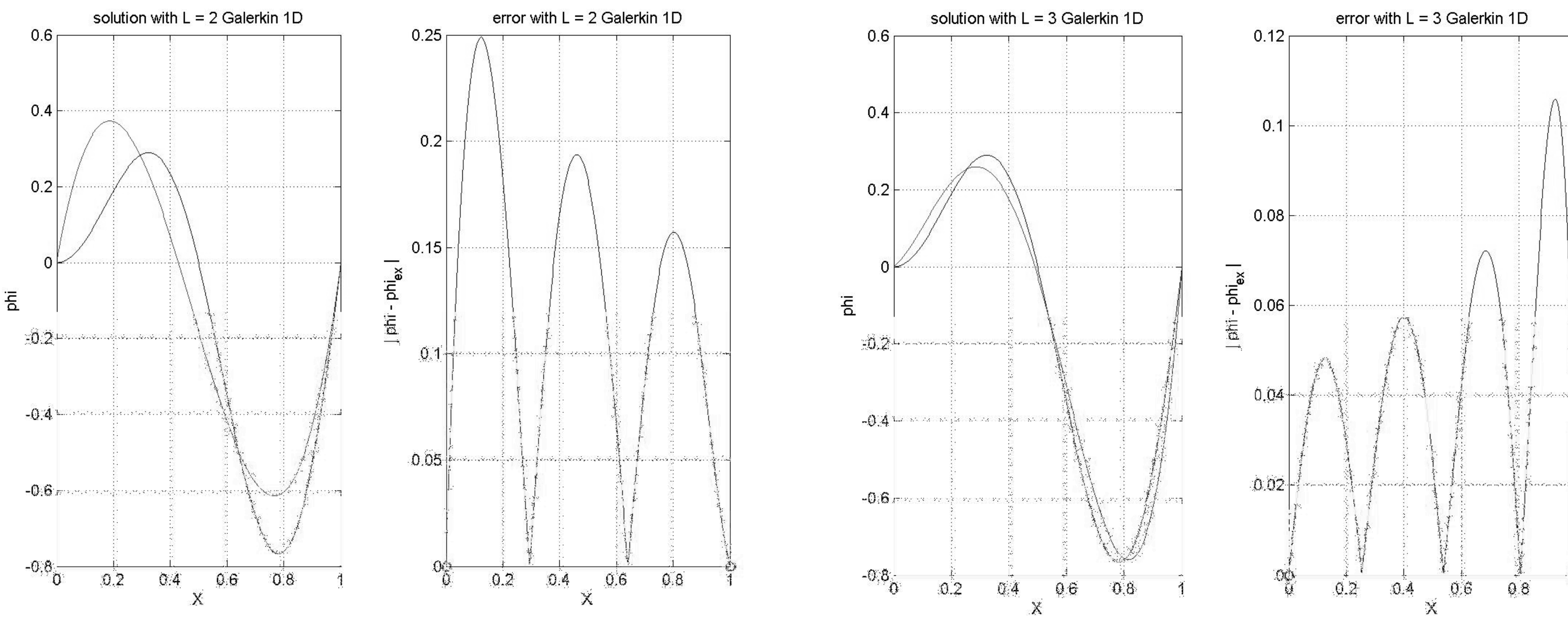
Shape function to be used $$N_m = x^m(1-x)$$



Galerkin in 1D approximations

Function to be approximated $$f = x\sin(2p\ x)$$

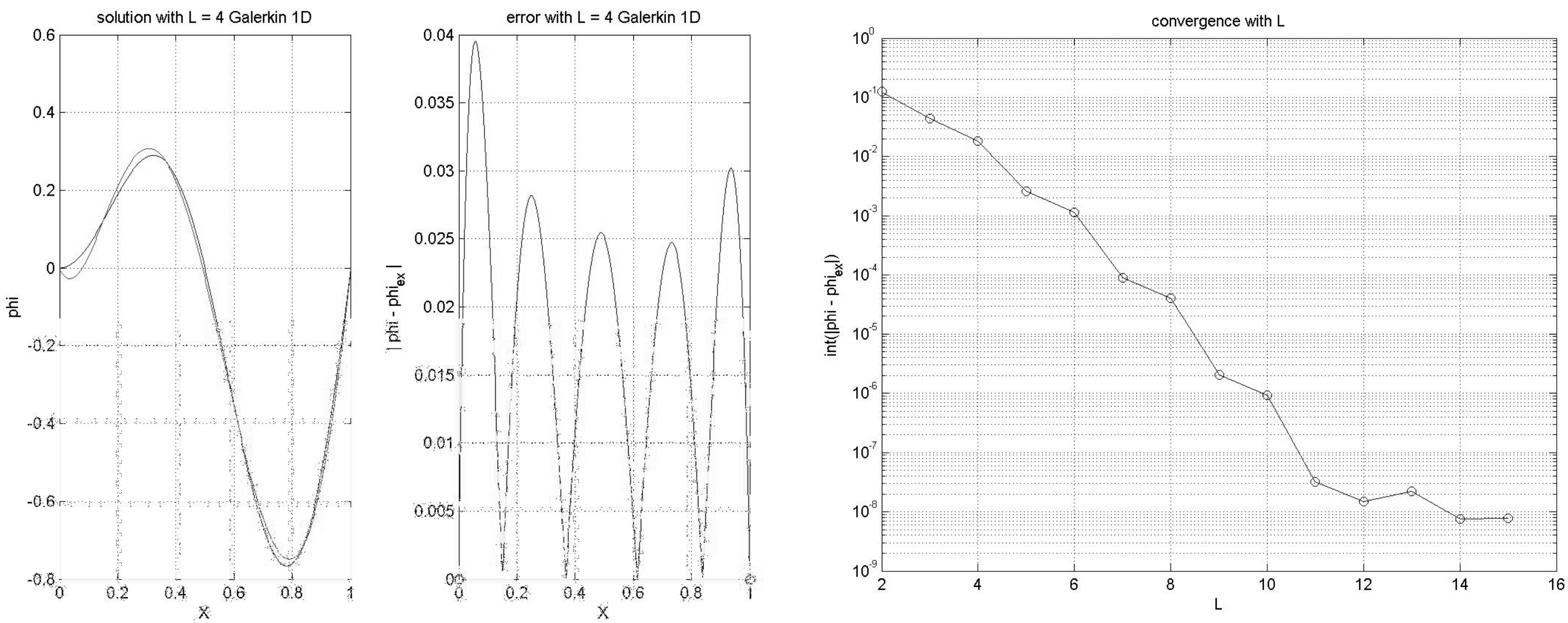
Shape function to be used $$N_m = x^m(1-x)$$



Galerkin in 1D approximations

Function to be approximated $$f = x\sin(2p\ x)$$

Shape function to be used $$N_m = x^m(1-x)$$

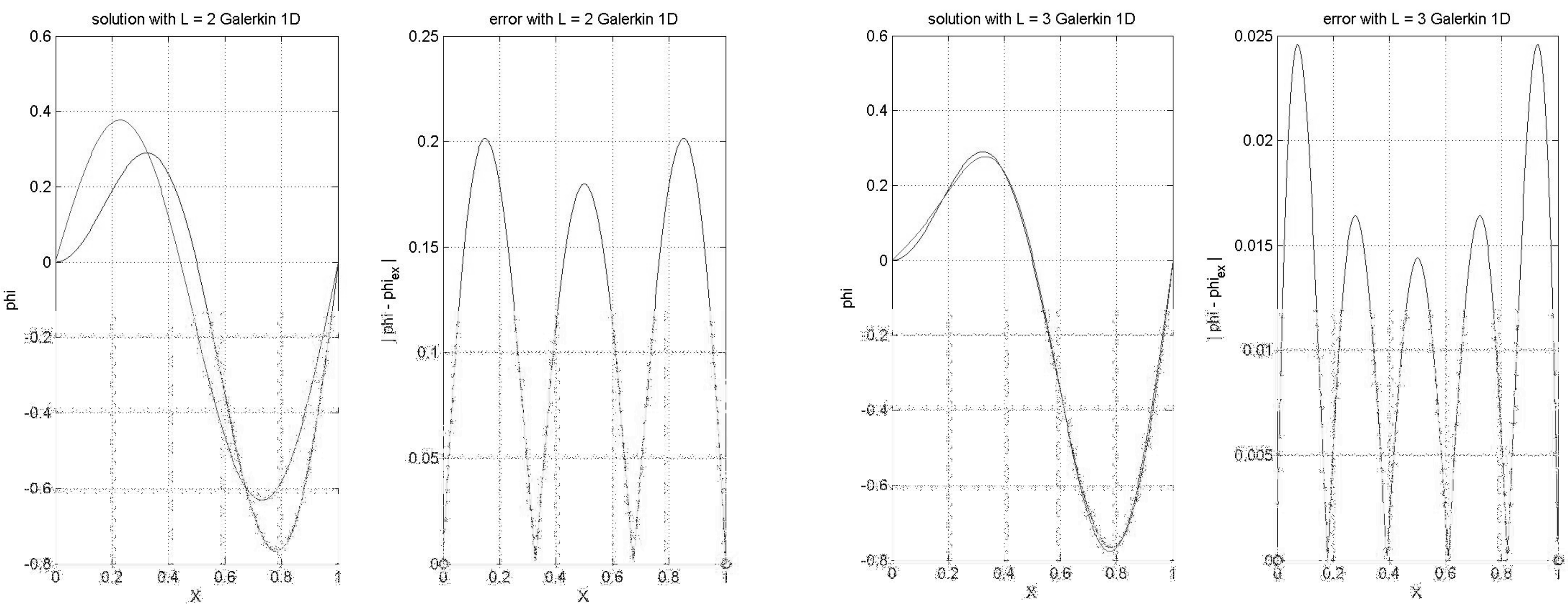


Galerkin in 1D approximations

Function to be approximated $$f = x\sin(2p\ \ x)$$

Shape function to be used $$N_m = \sin(mp\ \ x)$$

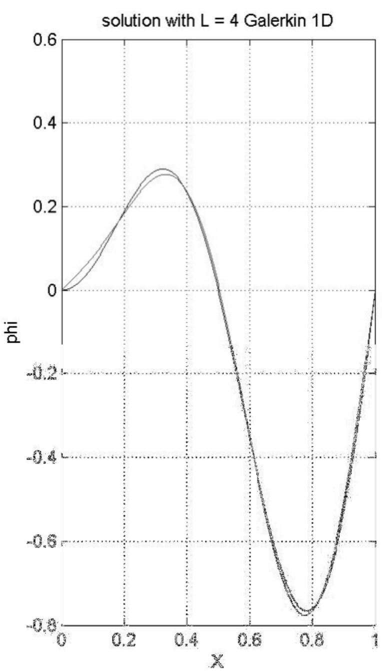
equivalent to Fourier series

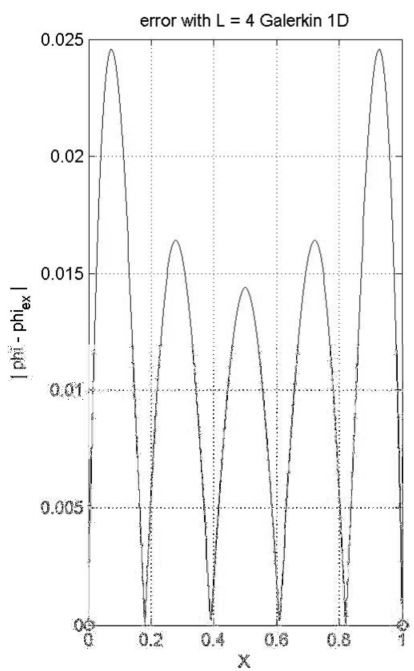


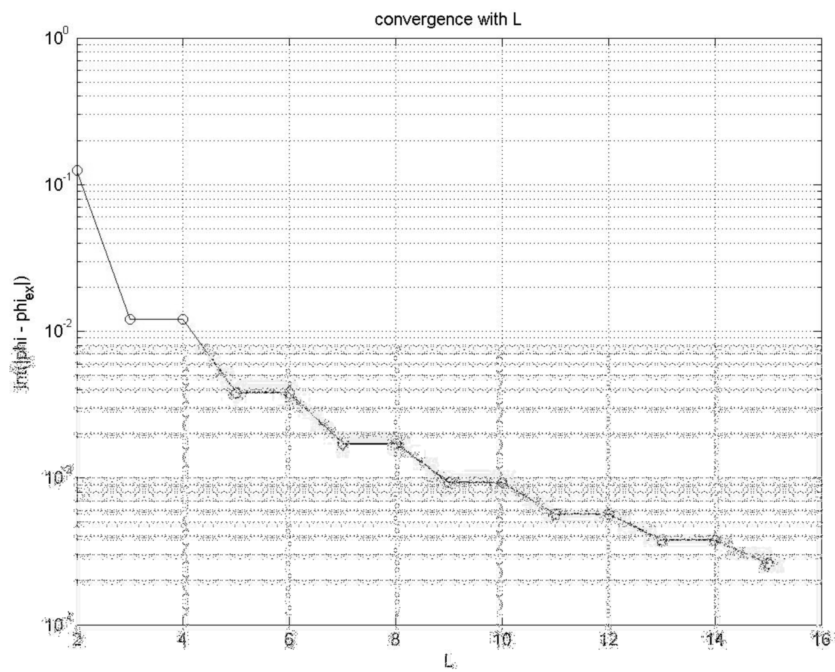
Galerkin in 1D approximations

Function to be approximated $f = x \sin(2 p\ x)$

Shape function to be used $N_m = \sin(m p\ x)$



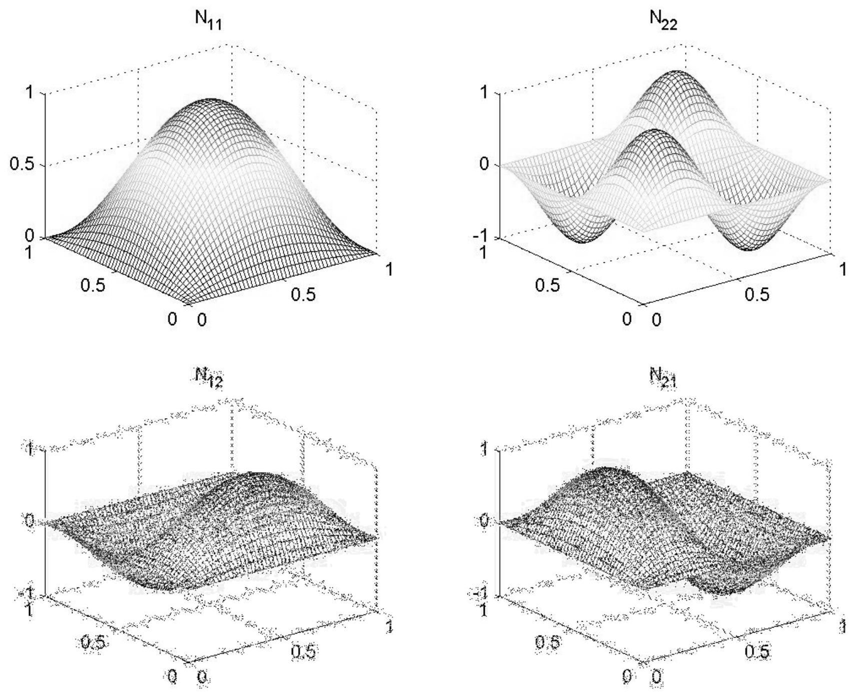


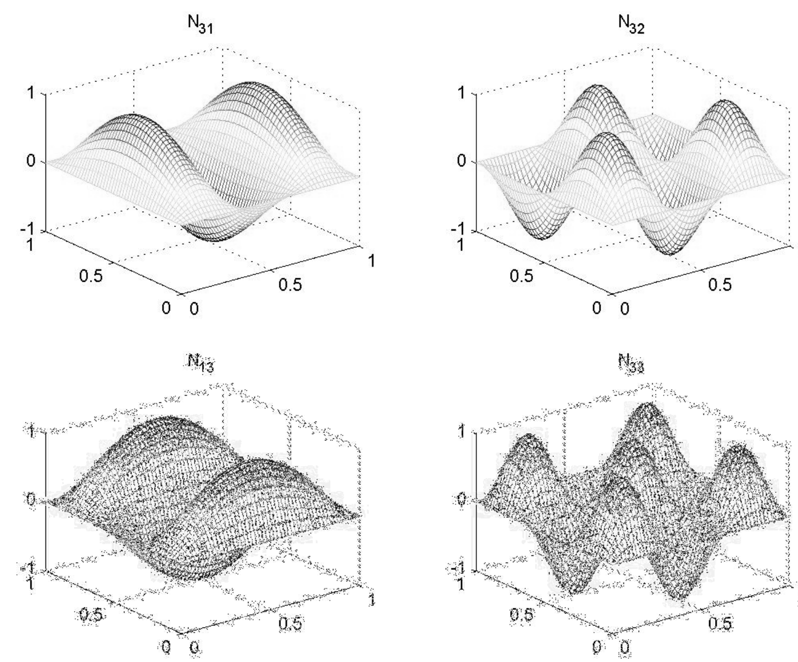


Galerkin in 2D approximations

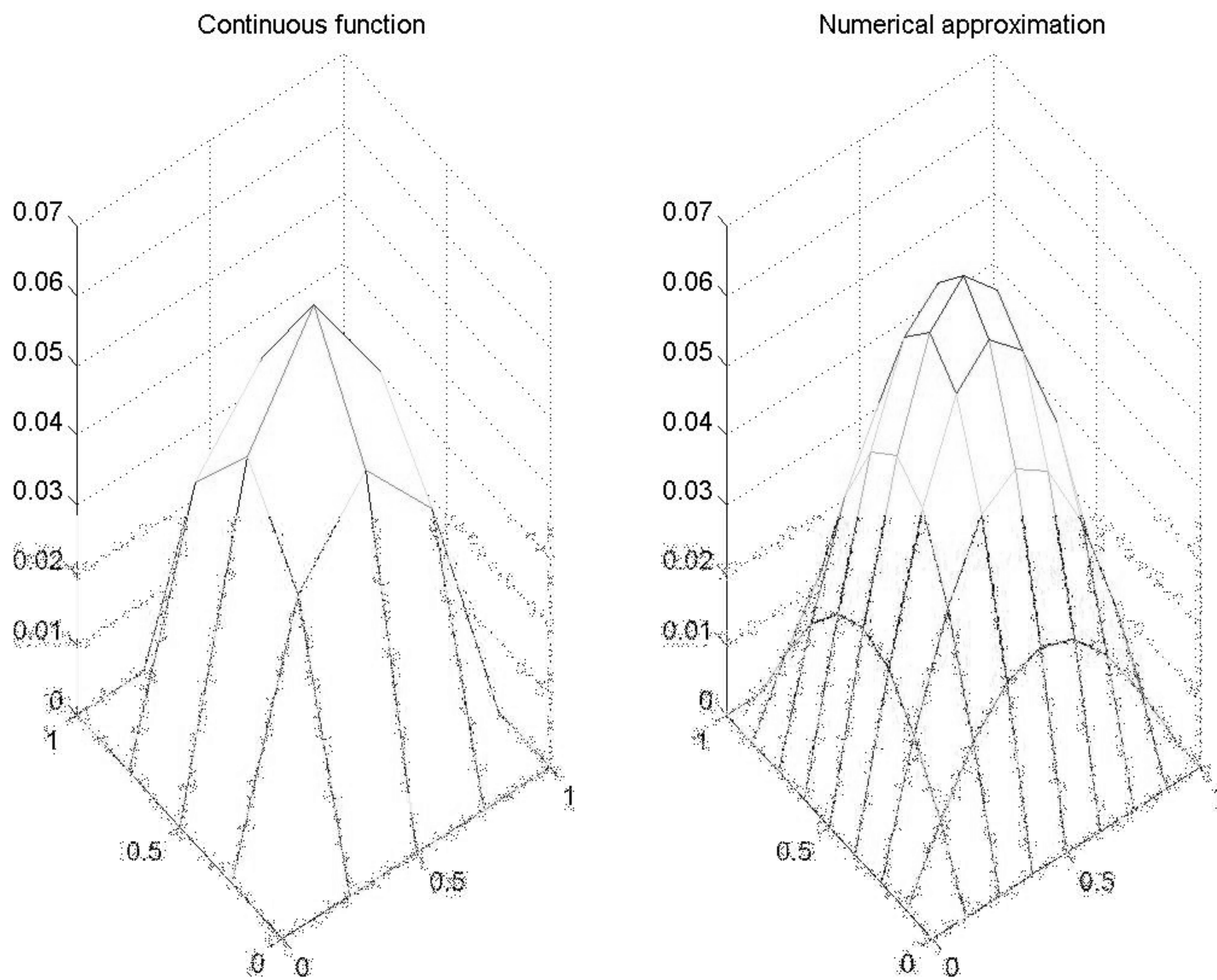
Function to be approximated $$f = x\,y\,(1-x)(1-y)$$

Shape function to be used $$N_{lm} = \sin(lp\ x)\sin(mp\ y)$$

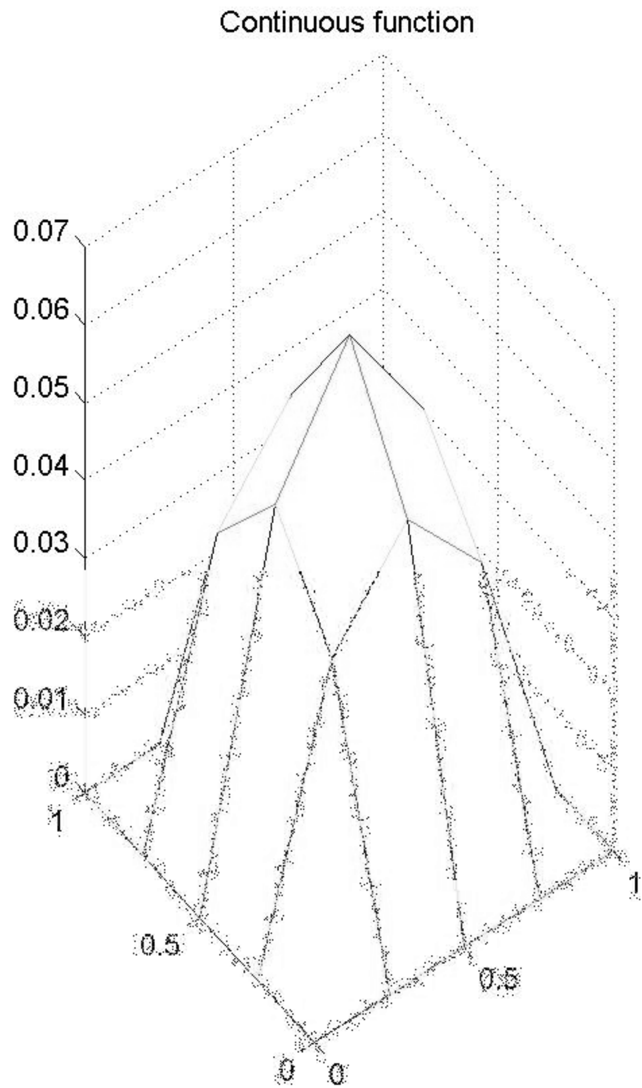


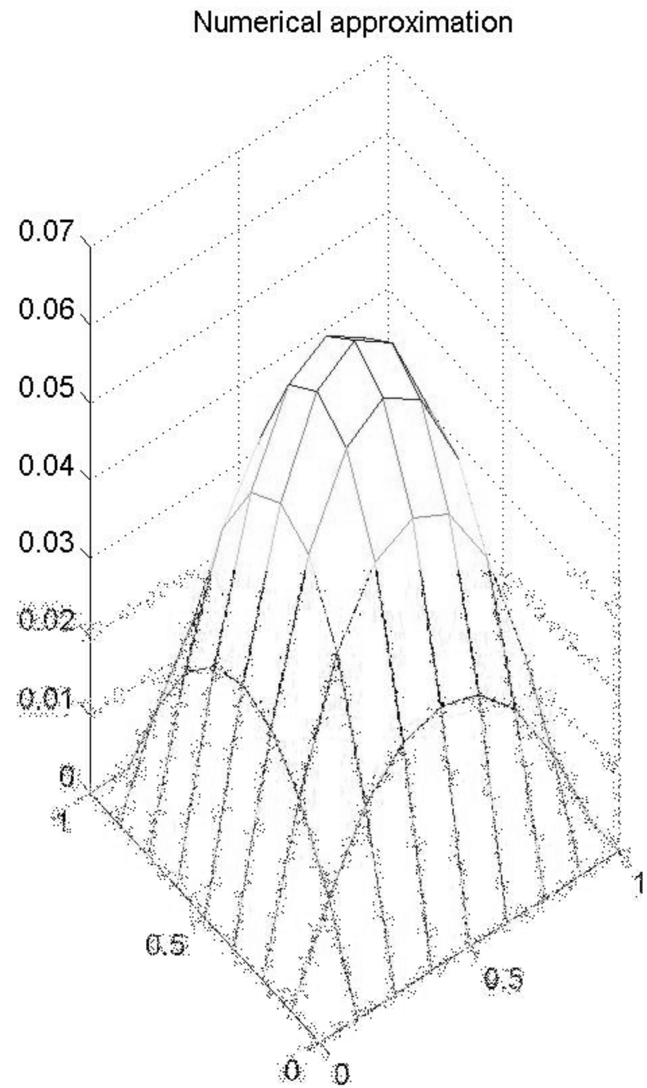


Galerkin in 2D approximations – L=M=2

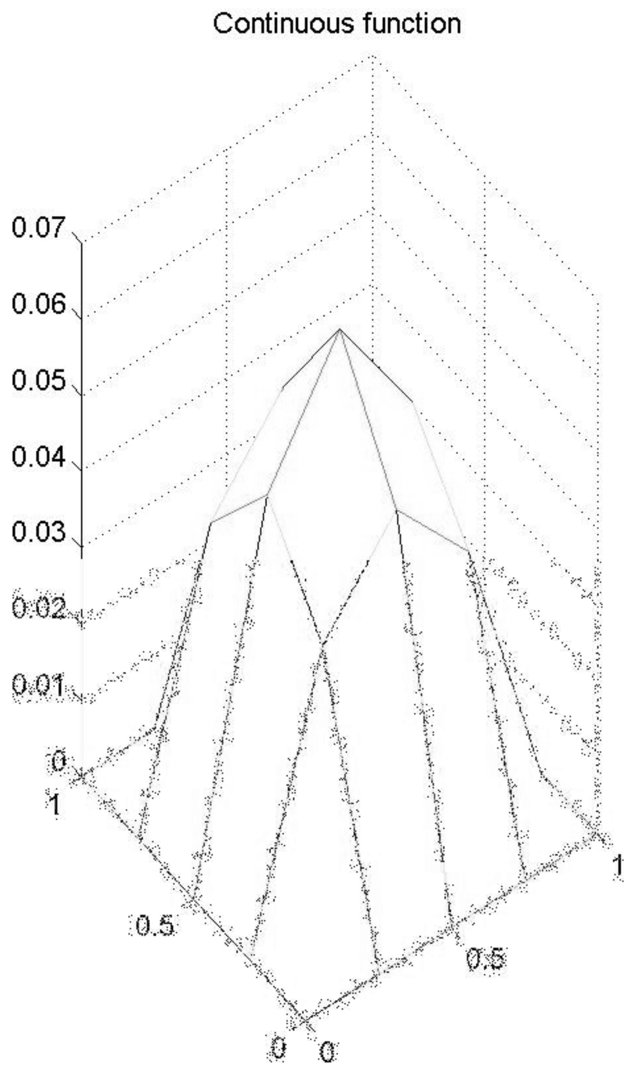


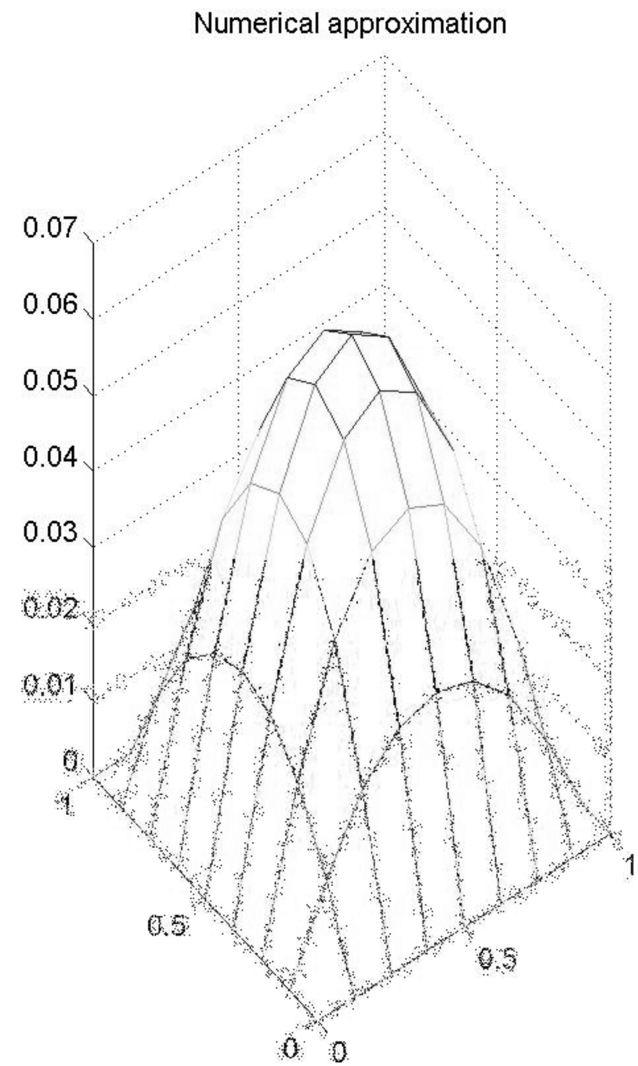
Galerkin in 2D approximations – L=M=3



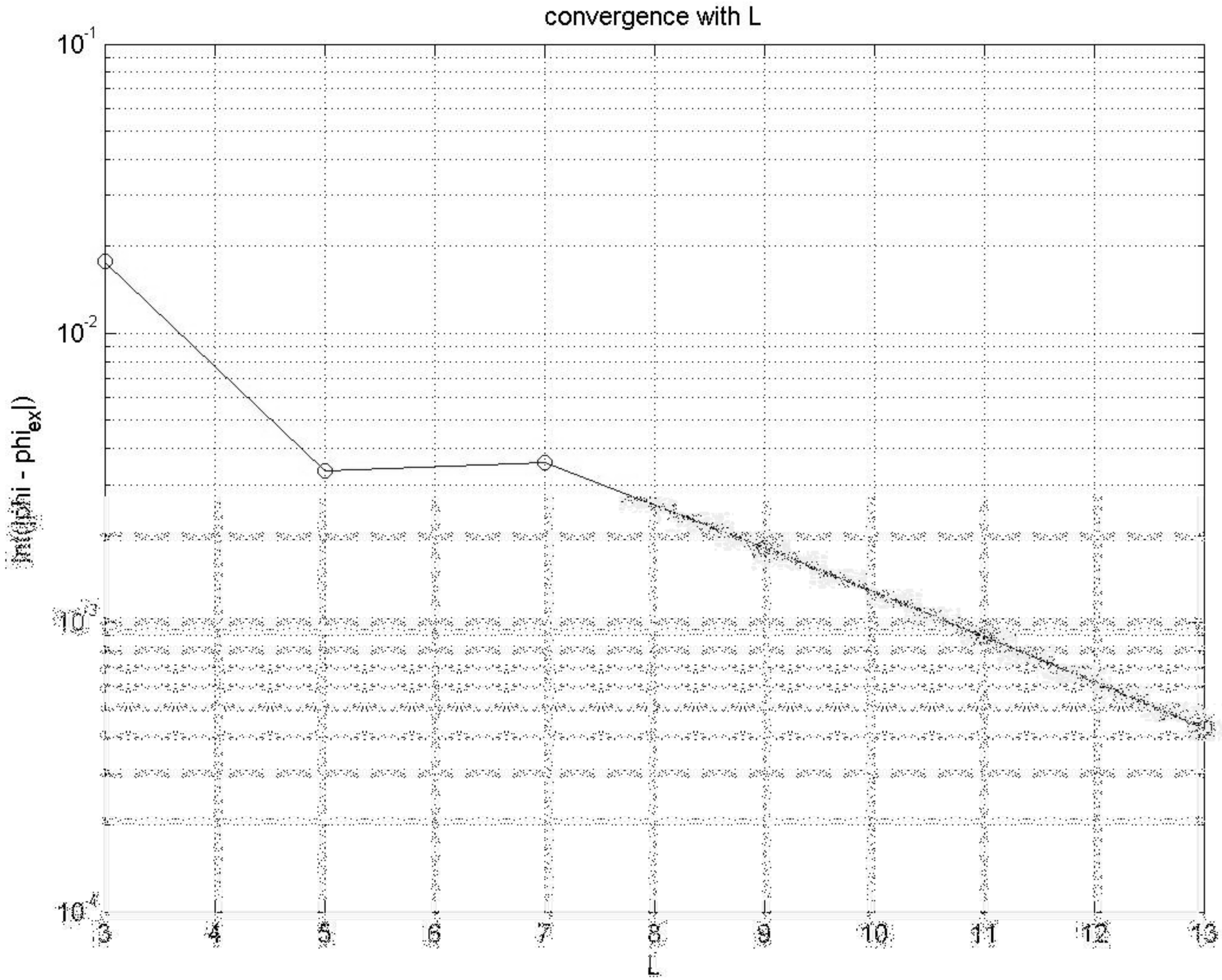


Galerkin in 2D approximations – L=M=4

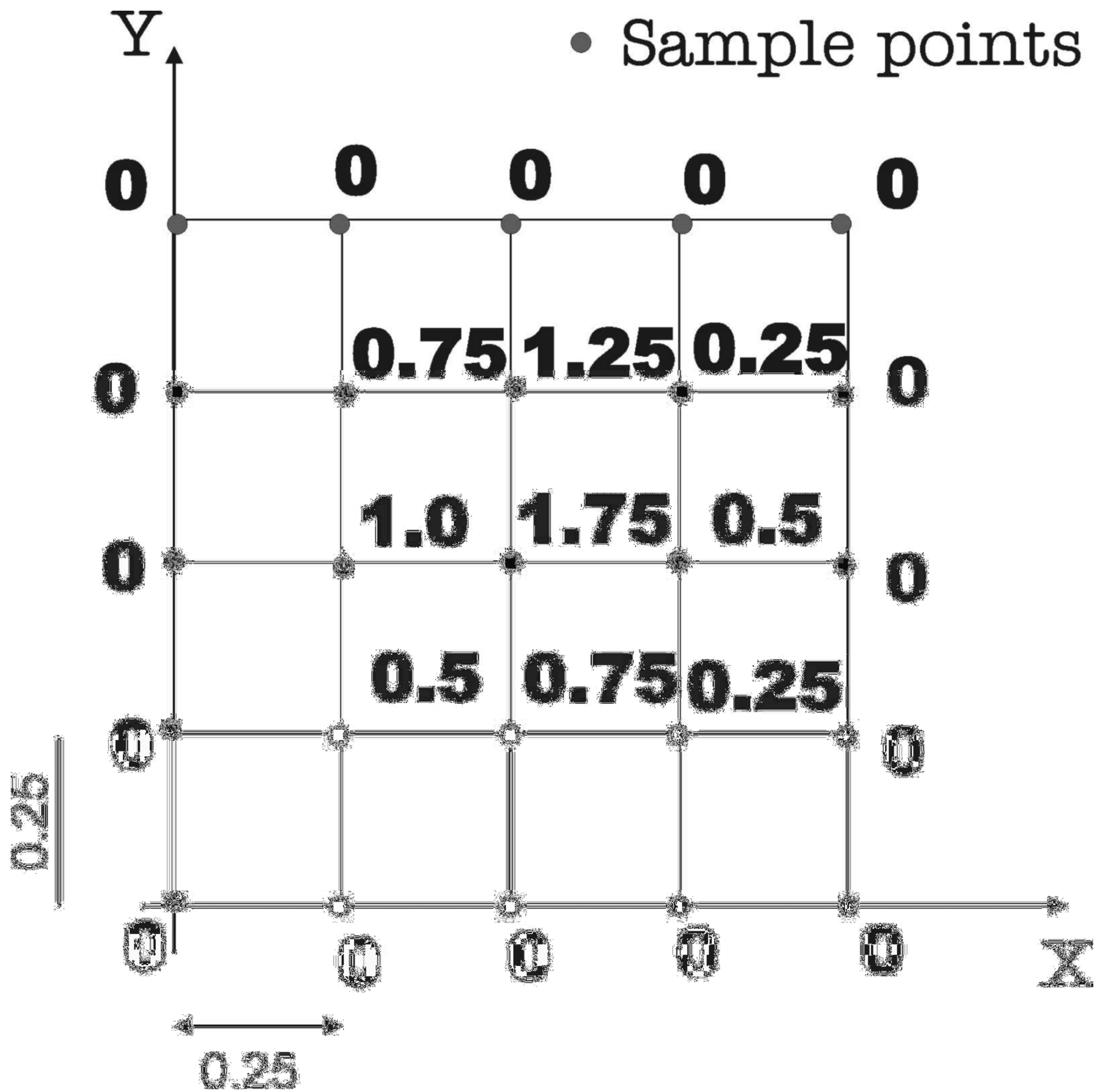




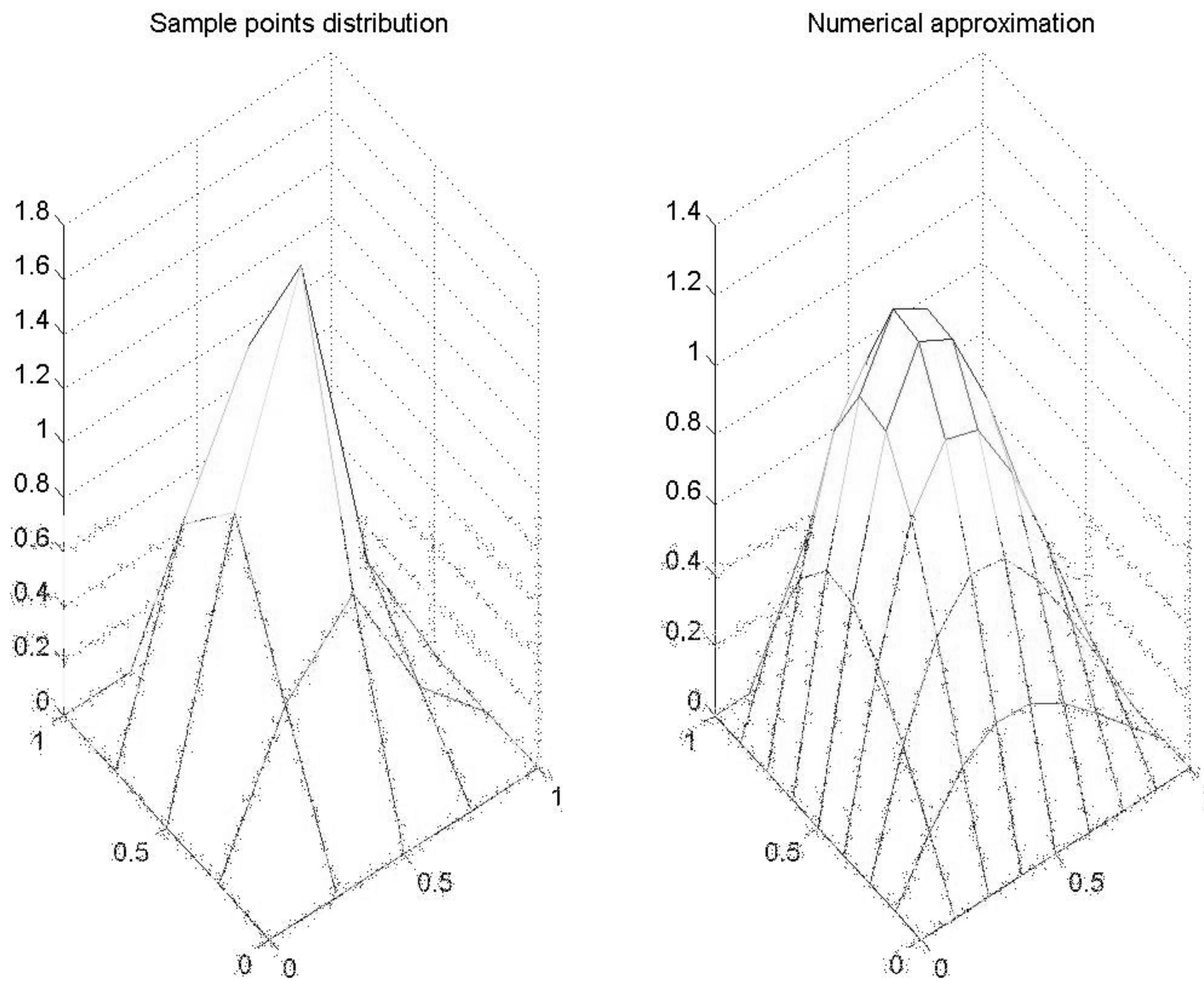
Galerkin in 2D approximations
Convergence with M & L



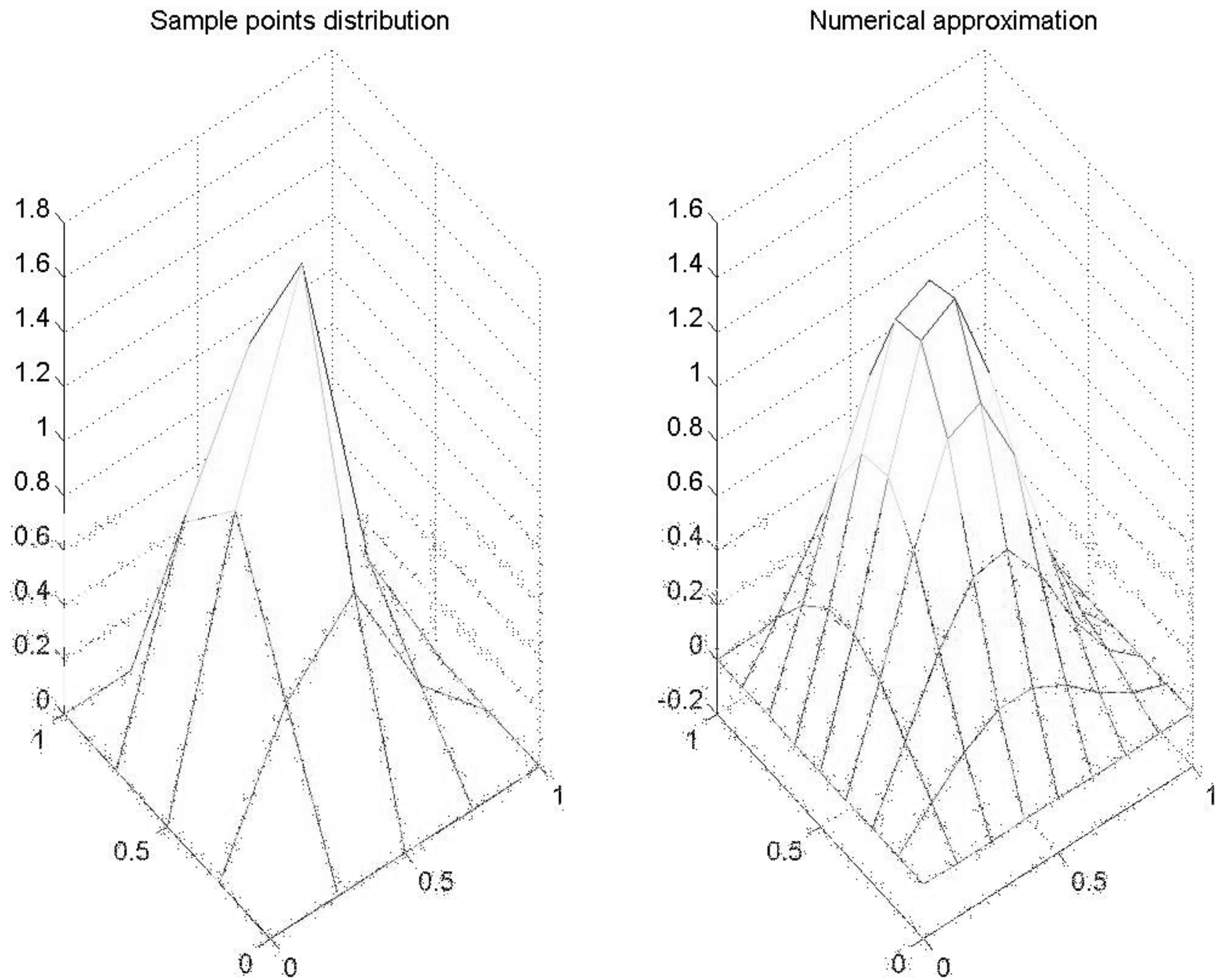
Galerkin in 2D approximations – A discrete problem



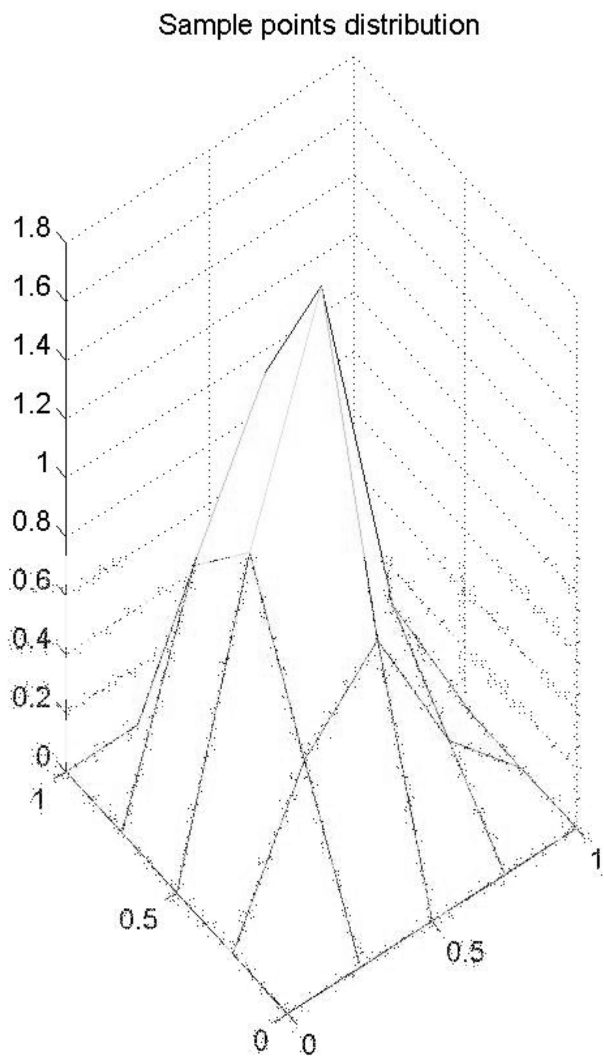
Galerkin in 2D approximations – L=M=2

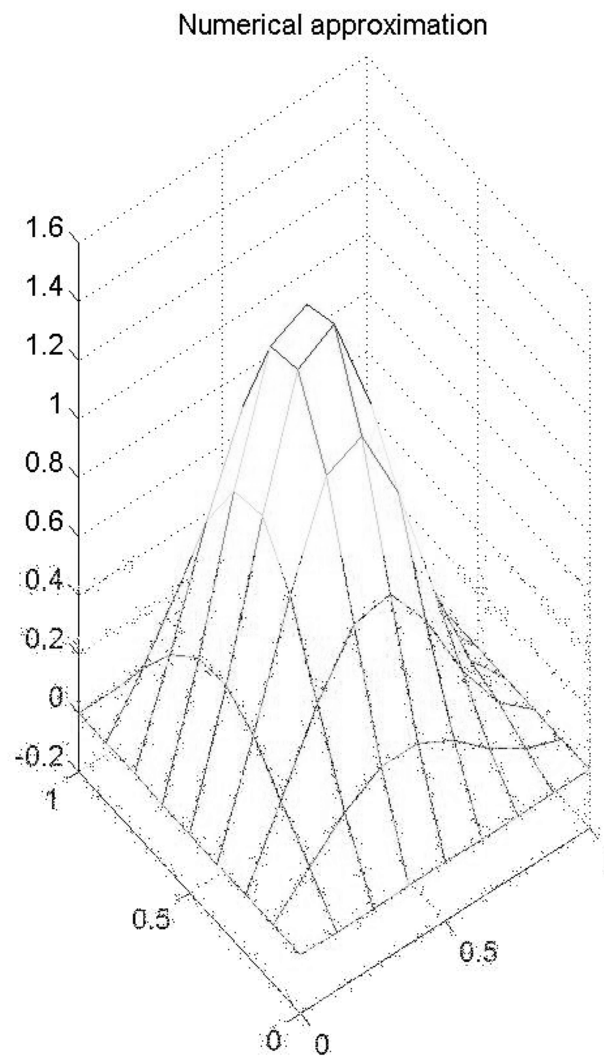


Galerkin in 2D approximations – L=M=3

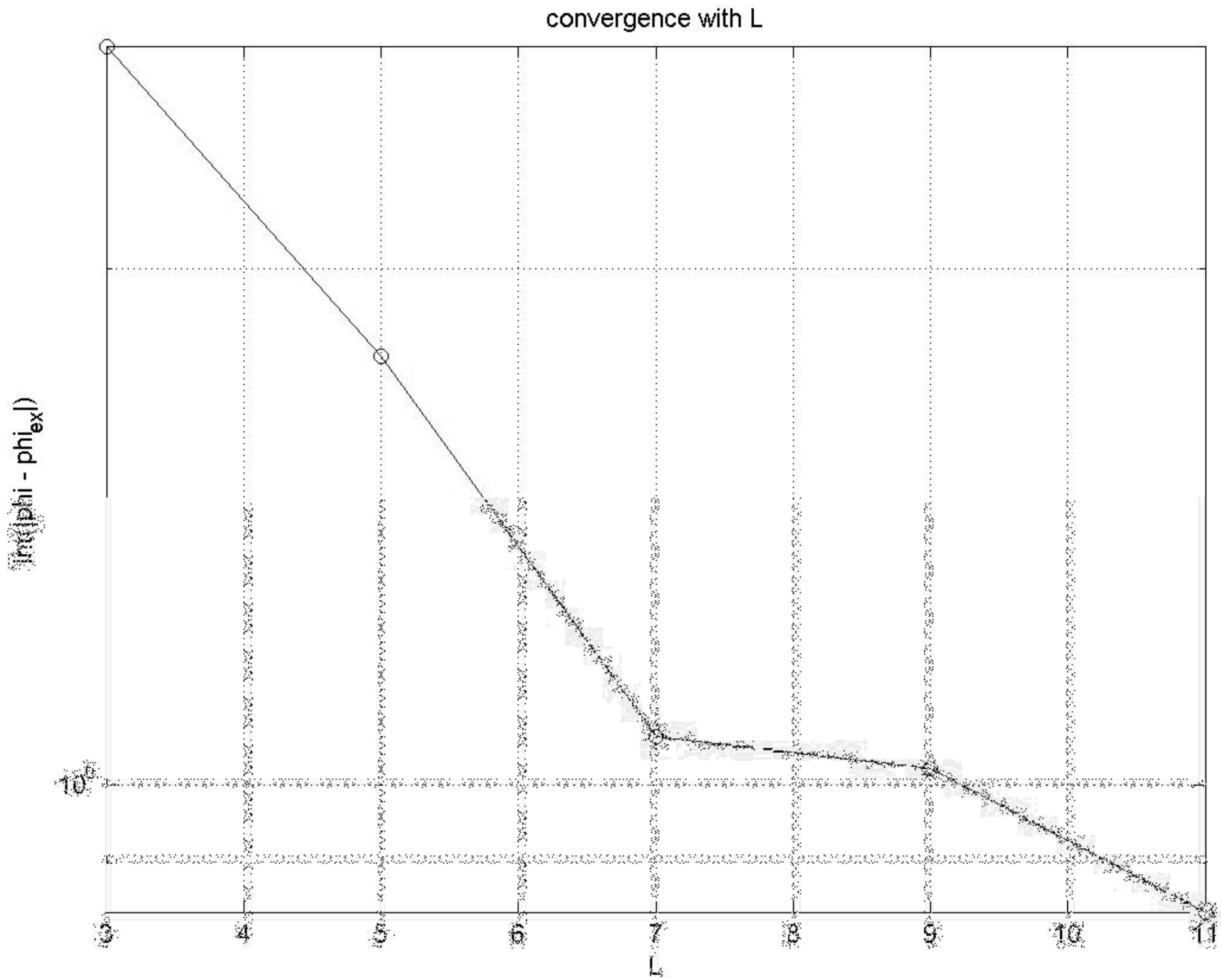


Galerkin in 2D approximations – L=M=4





Galerkin in 2D approximations
Convergence with M & L



Galerkin in 2D approximations – A code
Preprocessing – input data

```
global X Y Z ll mm ll2 mm2

Number of terms to be used (M)?
M = 3;
L = 3;

% grid of samples
[X,Y]=meshgrid(0:0.25:1,0:0.25:1);

% some refined grid
[XI,YI]=meshgrid(0:0.25/2:1,0:0.25/2:1);

% function to be approximated, defined in a discrete way
Z = zeros(size(X)); Z(2,2)=0.5; Z(2,3)=0.75; Z(2,4)=0.25;Z(3,2)=1;
Z(3,3)=1.75; Z(3,4)=0.5; Z(4,2)=0.75; Z(4,3)=1.25; Z(4,4)=0.25;

% interpolate on a refined grid
ZI = interp2(X,Y,Z,XI,YI);
```

Galerkin in 2D approximations – A code
Assembling and solving the algebraic system.

ASSEMBLING

```
f = zeros(L,M);
K = zeros(L,M,L,M);
for ll=1:L
    for mm=1:M
        f(ll,mm) = gauss_integration('ffun_Ej_2_0_2D_rhs',0,1,0,1);
        for ll2=1:L
            for mm2=1:M
                K(ll,mm,ll2,mm2) = gauss_integration('ffun_Ej_2_0_2D_lhs',0,1,0,1);
            end
        end
    end
end
```

SOLVING

```
K = reshape(K,M*L,M*L);
f = reshape(f,M*L,1);
a = K\f;
a = reshape(a,L,M);
```

$$\int_{\Omega} I(\vec{x})\, d\Omega$$

$$\Omega : \left\{ x \in (0,1),\, y \in (0,1) \right\}$$

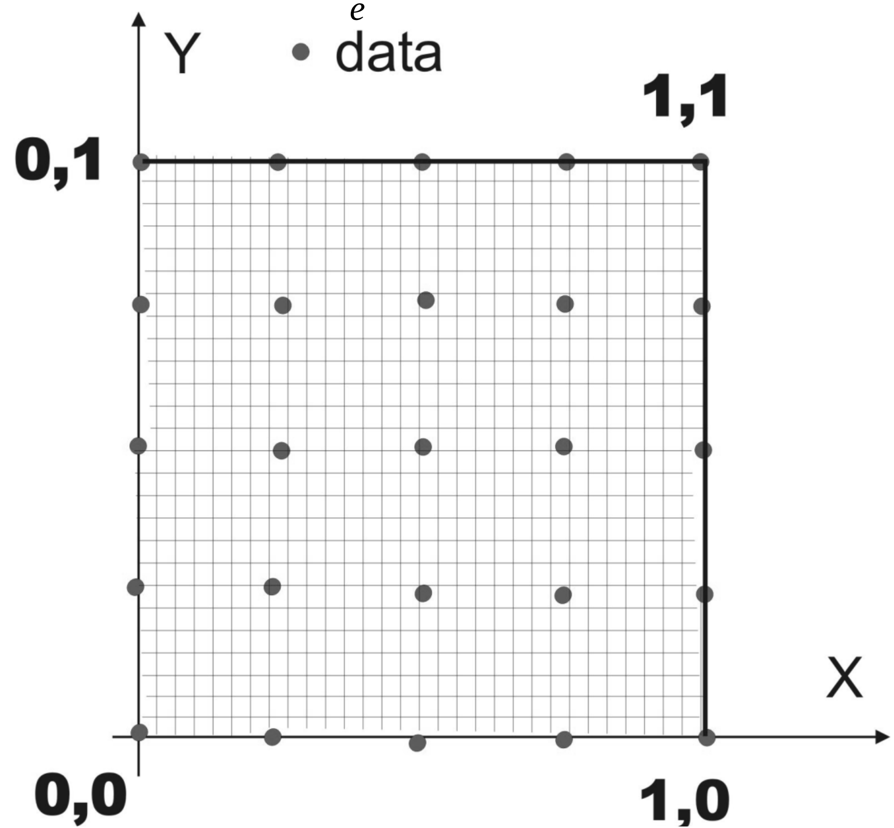
Galerkin in 2D approximations – A code
Gauss integration

```
Nx = Np; Ny = Np;

psi = (0:Nx)'/Nx;  eta = (0:Ny)'/Ny;
x1 = a+(b-a)*psi; x1 = [x1,0*x1];
x2 = c+(d-c)*eta; x2 = [0*x2,x2];
[xnod,icone] = qq3d(x1,x2);
[numel,nen] = size(icone);
psi_i = [-1,1,1,-1];  eta_i = [-1,-1,1,1];

fI = 0; xs = []; ys = [];

for k=1:nen
    xs = [xs , xnod(icone(:,k),1)];
    ys = [ys , xnod(icone(:,k),2)];
end

hx = (xs(:,2)-xs(:,1));
hy = (ys(:,4)-ys(:,1));
xsj = hx.*hy/4;
```

GENERATE A MESH FOR INTEGRATION

$$\Omega : \left\{ x \in (0,1),\, y \in (0,1) \right\}$$

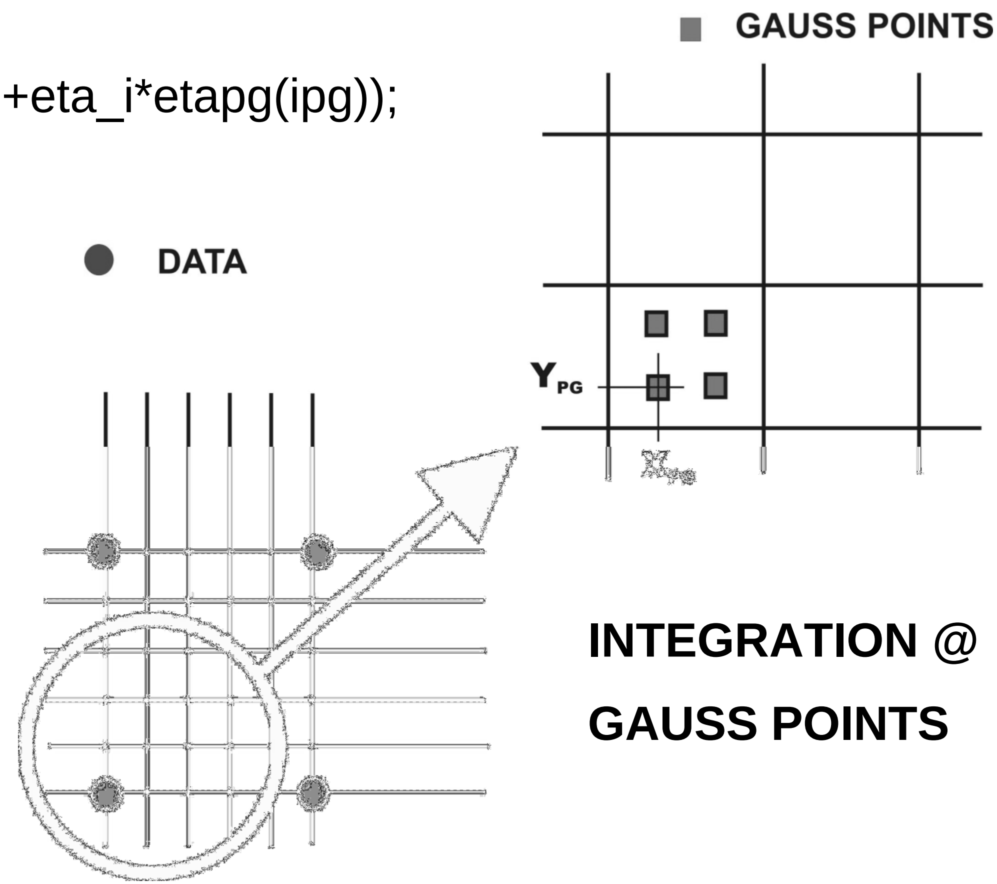
$$\Omega \cong \sum_e \Omega^e$$



Galerkin in 2D approximations – A code
Gauss integration – how to integrate

$$\boxed{\int_{\Omega} I(\vec{x})\, d\Omega \cong \sum_{e} \int_{\Omega^e} I(\vec{x})\, d\Omega^e \cong \sum_{e} \sum_{PG} I(\vec{x}^{\,e}_{PG})\, w_{PG}}$$

```
f_PG = 0;
for ipg=1:npg
   shape = 1/4*(1+psi_i*psipg(ipg)).*(1+eta_i*etapg(ipg));
   xpg = (shape*xs')';
   ypg = (shape*ys')';
   eval(['fpg = ' fun '(xpg,ypg);'])?
   f_PG = f_PG + wpg(ipg)*fpg.*xsj;
end
fI = sum(f_PG);
```

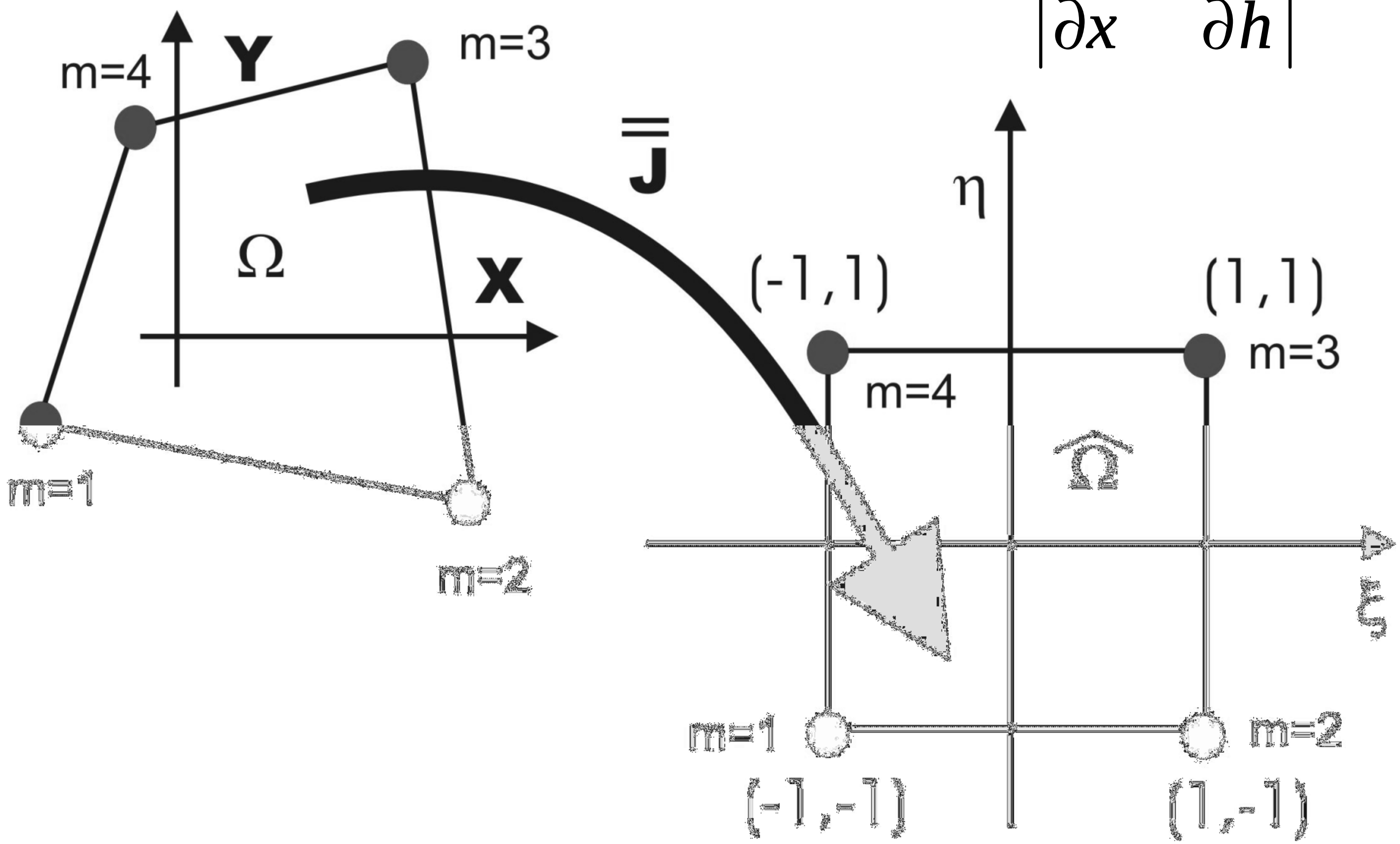


INTEGRATION @

GAUSS POINTS

Galerkin in 2D approximations – A code
Gauss integration - Mapping

$$x = x(x,h) \quad ; \quad y = y(x,h); \quad J = \begin{vmatrix} \dfrac{\partial x}{\partial x} & \dfrac{\partial x}{\partial h} \\ \dfrac{\partial y}{\partial x} & \dfrac{\partial y}{\partial h} \end{vmatrix}$$



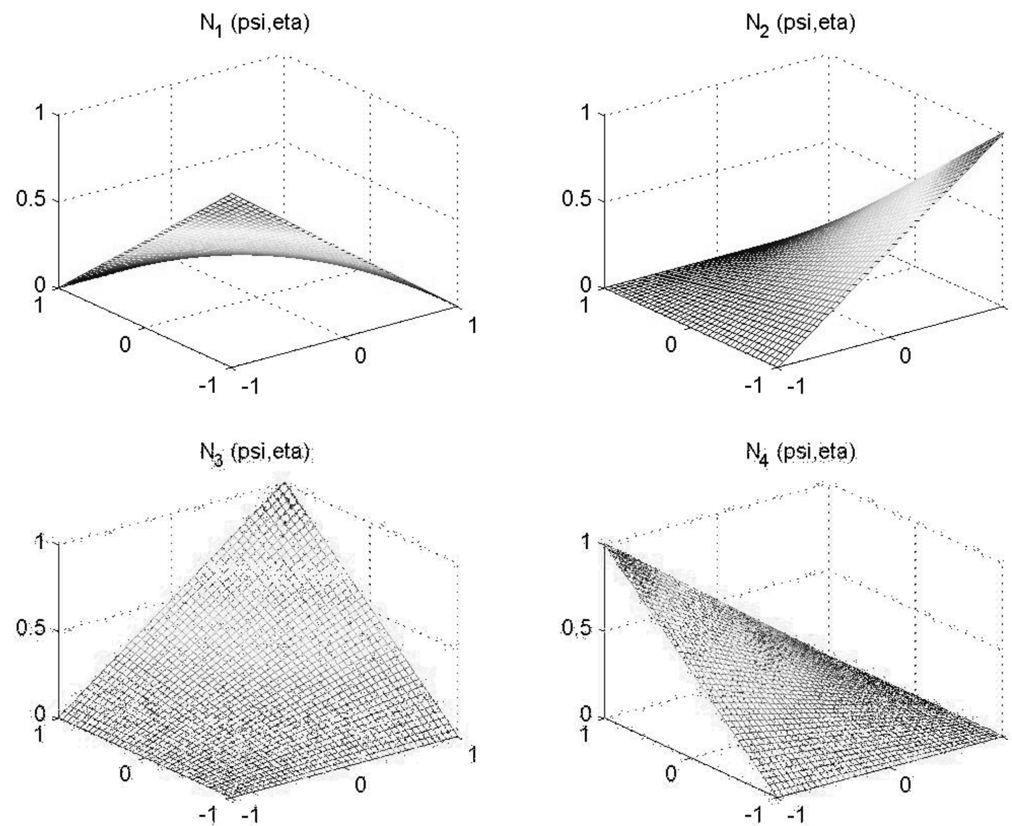
Galerkin in 2D approximations – A code
Gauss integration – how to integrate

$$\boxed{\sum_e \int_{\Omega^e} I(\vec{x})\, d\Omega^e \cong \sum_e \int_{\hat{\Omega}^e} I(\vec{x}(x,h))\, |J|\, d\hat{\Omega}^e}$$

$$f(x,h) = \sum_m f_m \; N_m(x,h)$$

$$m = 1,2,3,4$$

$$f = \begin{cases} x, y \\ f \end{cases}$$



Other weightings

$$I\left(a_1, a_2, a_3, \cdots, a_M\right) = \int_\Omega \left(f - \hat{f}\right)^2 d\Omega$$

$$\frac{\partial I}{\partial a_l} = 0 \qquad l = 1, 2, \cdots, M$$

$$\frac{\partial \int_\Omega \left(f - \hat{f}\right)^2 d\Omega}{\partial a_l} = \int_\Omega \frac{\partial}{\partial a_l}\left(f - \hat{f}\right)^2 d\Omega = \int_\Omega 2\left(f - \hat{f}\right)\frac{\partial \hat{f}}{\partial a_l} d\Omega$$

$$\hat{f} = y + \sum_l a_l N_l \quad \Rightarrow \frac{\partial \hat{f}}{\partial a_l} = N_l$$

$$\therefore \int_\Omega 2\left(f - \hat{f}\right) N_l \, d\Omega = 0 \quad \Rightarrow \quad \underbrace{\int_\Omega \left(f - \hat{f}\right) N_l \, d\Omega = 0}_{GALERKIN}$$

$$\text{LEAST SQUARE APPROXIMATION} = \text{GALERKIN}$$