Chapter 6

Weighted Residual Methods

Weighted residual methods (WRM) assume that a solution can be approximated analytically or piecewise analytically. In general a solution to a PDE can be expressed as a superposition of a base set of functions

$$T(x,t) = \sum_{j=1}^{N} a_j(t)\phi_j(x)$$

where the coefficients a_j are determined by a chosen method. The method attempts to minimize the error, for instance, finite differences try to minimize the error specifically at the chosen grid points. WRM's represent a particular group of methods where an integral error is minimized in a certain way and thereby defining the specific method. Depending on the maximization WRM generate

- the finite volume method,
- finite element methods,
- spectral methods, and also
- finite difference methods.

6.1 General Formulation

The starting point for WRM's is an expansion in a set of base or trial functions. Often these are analytical in which case the numerical solution will be analytical

$$T(x,y,z,t) = T_0(x,y,z,t) + \sum_{j=1}^{N} a_j(t)\phi_j(x,y,z) \tag{6.1}$$

with the trial functions $\phi_j(x, y, z)$. $T_0(x, y, z, t)$ is chosen to satisfy initial or boundary conditions and the coefficients $a_j(t)$ have to be determined. possible trial functions are

$\phi_j(x) = x^{j-1}$ or $\phi_j(x) = \sin(j\pi x)$.

The expansion is chosen to satisfy a differential equation $L(\overline{T}) = 0$ (where $\overline{T}$is the exact solution), e.g.,

$$L(\overline{T}) = \frac{\partial \overline{T}}{\partial t} - \alpha \frac{\partial^2 \overline{T}}{\partial x^2} = 0$$

However, the numerical solution is an approximate solution, i.e., $T \neq \overline{T}$ such that the operator L applied to T produces a residual

$$L(T) = R$$

The goal of WRM's is to choose the coefficients a_j such that the residual R becomes small (in fact 0) over a chosen domain. In integral form this can be achieved with the condition

$$\int \int \int W_m(x, y, z) R dx dy dz = 0 \tag{6.2}$$

where W_m is a set of weight functions ($m = 1, ...M$) which are used to evaluate (6.2). The exact solution always satisfies (6.2) if the weight functions are analytic. This is in particular true also for any given subdomain of the domain for which a solution is sought.

There are four main categories of weight or test functions which are applied in WRM's.

i) Subdomain method: Here the domain is divided in M subdomains D_m where

$$W_m = \begin{cases} 1 & in\, D_m \\ 0 & outside \end{cases} \tag{6.3}$$

such that this method minimizes the residual error in each of the chosen subdomains. Note that the choice of the subdomains is free. In many cases an equal division of the total domain is likely the best choice. However, if higher resolution (and a corresponding smaller error) in a particular area is desired, a non-uniform choice may be more appropriate.

ii) Collocation method: In this method the weight functions are chosen to be Dirac delta functions

$$W_m(x) = \delta(x - x_m) \tag{6.4}$$

such that the error is zero at the chosen nodes x_m.

iii) Least squares method: This method uses derivatives of the residual itself as weight functions in the form

$$W_m(x) = \frac{\partial R}{\partial a_m}. \tag{6.5}$$

The motivation for this choice is to minimize $\int\int\int R^2 dxdydz$ of the computational domain. Note that this choice of the weight function implies

$$\frac{\partial}{\partial a_m}\int\int\int R^2 dxdydz = 0$$

for all values of a_m.

iv) Galerkin method: In this method the weight functions are chosen to be identical to the base functions.

$$W_m(x) = \phi_m(x)$$

In particular if the base function set is orthogonal ($\int \phi_m(x)\phi_n(x) = 0$ if $m \neq n$), this choice of weight functions implies that the residual R is rendered orthogonal with the condition (6.2) for all base functions.

Note that M weight functions yield M conditions (or equations) from which to determine the N coefficients a_j. To determine these N coefficients uniquely we need N independent condition (equations).

Example: Consider the ordinary differential equation

$$\frac{dy}{dx} - y = 0 \tag{6.6}$$

for $0 \leq x \leq 1$ with $y(0) = 1$. Let us assume an approximate solution in the form of polynomials

$$y = 1 + \sum_{j=1}^{N} a_j x^j \tag{6.7}$$

where the constant 1 satisfies the boundary condition. Substituting this expression into the differential equation (6.6) gives the residual

$$R = -1 + \sum_{j=1}^{N} a_j \left(jx^{j-1} - x^j\right) \tag{6.8}$$

We can now compute the coefficients a_j with the various methods.

Galerkin method:

Here we use the weight functions $W_m = x^{m-1}$. The maximization implies with (6.2)

$$\int_0^1 x^{m-1}\left[-1+\sum_{j=1}^{N} a_j\left(jx^{j-1}-x^j\right)\right]dx = 0$$

or

$$-\int_0^1 x^{m-1}dx + \sum_{j=1}^{N}\left[a_j\left(j\int_0^1 x^{m-1}x^{j-1}dx - \int_0^1 x^{m-1}x^j dx\right)\right] = 0$$

which yields after integration

$$-\frac{1}{m} + \sum_{j=1}^{N} a_j\left[\frac{j}{m+j-1} - \frac{1}{m+j}\right] = 0 \tag{6.9}$$

With the matrix $\underline{\underline{S}}$ (with elements $s_{mj} = \frac{j}{m+j-1} - \frac{1}{m+j}$) and the vector $\mathbf{D}$ (with the elements $d_m = -\frac{1}{m}$) we can rewrite (6.9) as

$$\underline{\underline{\mathbf{S}}}\mathbf{A} = \mathbf{D} \tag{6.10}$$

The solution $\mathbf{A}$ of this set of equations requires to invert $\underline{\underline{S}}$. For $N = 3$ the system becomes

$$\begin{bmatrix} 1/2 & 2/3 & 3/4 \\ 1/6 & 5/12 & 11/20 \\ 1/12 & 3/10 & 13/30 \end{bmatrix} \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 1/2 \\ 1/3 \end{pmatrix}$$

For this set the approximate solution becomes

$$y = 1 + 1.0141x + 0.4225x^2 + 0.2817x^3$$

Least squares method:

Here the weight functions are

$$W_m(x) = \frac{\partial R}{\partial a_m} = mx^{m-1} - x^m$$

This gives the residual condition

$$\begin{aligned} \int_0^1 \left(mx^{m-1}-x^m\right)\left[-1+\sum_{j=1}^{N} a_j\left(jx^{j-1}-x^j\right)\right]dx &= \\ -1 + \frac{1}{m+1} + \sum_{j=1}^{N}\left[a_j\int_0^1 \left(mx^{m-1}-x^m\right)\left(jx^{j-1}-x^j\right)dx\right] &= 0 \end{aligned}$$

or

$$-1 + \frac{1}{m+1} + \sum_{j=1}^{N} a_j \left[\frac{jm}{m+j-1} - \frac{j+m}{m+j} + \frac{1}{m+j+1} \right] = 0$$

Subdomain method:

Weight functions:

$$W_m = \begin{cases} 1 & x \in (\frac{m-1}{N}, \frac{m}{N}] \\ 0 & outside \end{cases}$$

Condition for the residual error:

$$\int_{\frac{m-1}{N}}^{\frac{m}{N}} \left[-1 + \sum_{j=1}^{N} a_j \left(j x^{j-1} - x^j \right) \right] dx = 0$$

Collocation method:

Collocation point (for the Dirac delta function): $x_m = \frac{m-1}{N}$.

Condition for the residual error:

$$\int_0^1 \delta(x - x_m) \left[-1 + \sum_{j=1}^{N} a_j \left(j x^{j-1} - x^j \right) \right] dx = 0$$

6.2 Finite Volume Method

This section illustrate the use of the finite volume method first for a PDE involving first order derivatives and later for second order derivatives. The program FIVOL will be introduced which can be used to solve Laplace's equation or Poisson's equation, i.e, equations of the elliptic type.

6.2.1 First order derivatives

Let us consider the equation

$$\frac{\partial q}{\partial t} + \frac{\partial F}{\partial x} + \frac{\partial G}{\partial y} = 0 \tag{6.11}$$

which is to be solved with the subdomain method on a domain given by coordinate line (j, k) describing a curved coordinate system. This equation is a typical continuity equation like the equations for conservation of mass, momentum etc.

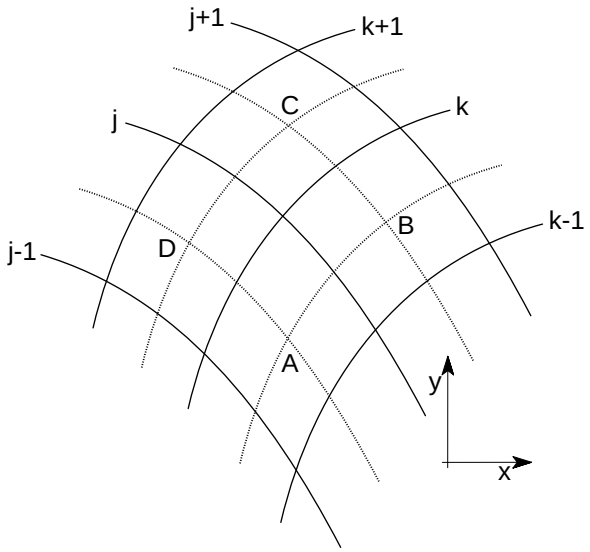


Figure 6.1: Illustration of the coordinate system to solve equation (6.11) with the finite volume method.

Equation (6.11) is to be solved as an integral over any area $\overline{ABCD}$as illustrated in Figure 6.1 =>

$$\int_{\overline{ABCD}} \left[\frac{\partial q}{\partial t} + \frac{\partial F}{\partial x} + \frac{\partial G}{\partial y} \right] dxdy = 0.$$

With $\mathbf{H} = (F, G)$ such that $\partial F/\partial x + \partial G/\partial y = \nabla \cdot \mathbf{H}$ this equation becomes (using Gauss's theorem)

$$\frac{d}{dt} \int_{\overline{ABCD}} q dV + \oint_{\partial ABCD} \mathbf{H} \cdot \mathbf{n} ds = 0$$

In Cartesian coordinates the surface element vector $d\mathbf{s} = (dx, dy)$ such that the normal vector $\mathbf{n}ds = (dy, -dx)$ and $\mathbf{H} \cdot \mathbf{n}ds = Fdy - Gdx$. Thus the integral of (6.11) For the area $\overline{ABCD}$ becomes

$$\frac{d}{dt} \left(A_r q_{jk} \right) + \sum_{ABCD} \left(F\Delta y - G\Delta x \right) = 0$$

with A_r =area of $\overline{ABCD}$. Using the notations $\Delta y_{AB} = y_B - y_A$, $\Delta x_{AB} = x_B - x_A$, and the averages $F_{AB} = 0.5 \left(F_{j,k-1} + F_{j,k} \right)$, $G_{AB} = 0.5 \left(G_{j,k-1} + G_{j,k} \right)$ and applying these to all sections of $\overline{ABCD}$we obtain

$$\begin{array}{lllll} A_r \frac{dq_{jk}}{dt} & +0.5 \left(F_{j,k-1} + F_{j,k} \right) \Delta y_{AB} & -0.5 \left(G_{j,k-1} + G_{j,k} \right) \Delta x_{AB} & & \\ & +0.5 \left(F_{j,k} + F_{j+1,k} \right) \Delta y_{BC} & -0.5 \left(G_{j,k} + G_{j+1,k} \right) \Delta x_{BC} & & \\ & +0.5 \left(F_{j,k} + F_{j,k+1} \right) \Delta y_{CD} & -0.5 \left(G_{j,k} + G_{j,k+1} \right) \Delta x_{CD} & & \\ & +0.5 \left(F_{j-1,k} + F_{j,k} \right) \Delta y_{DA} & -0.5 \left(G_{j-1,k} + G_{j,k} \right) \Delta x_{DA} & = & 0 \end{array} \tag{6.12}$$

In case of a uniform grid parallel to the x, and the y axes the area is $A_r = \Delta x \Delta y$ and equation (6.12) reduces to

$$\frac{dq_{jk}}{dt} + \frac{F_{j+1,k} - F_{j-1,k}}{2\Delta x} + \frac{G_{j+1,k} - G_{j-1,k}}{2\Delta y} = 0$$

where the spatial derivative is equal to that for the centered space finite difference approximation.

6.2.2 Second order derivatives

To introduce the finite volume second derivatives let us consider Laplace's equation

$$\frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial x^2} = 0 \tag{6.13}$$

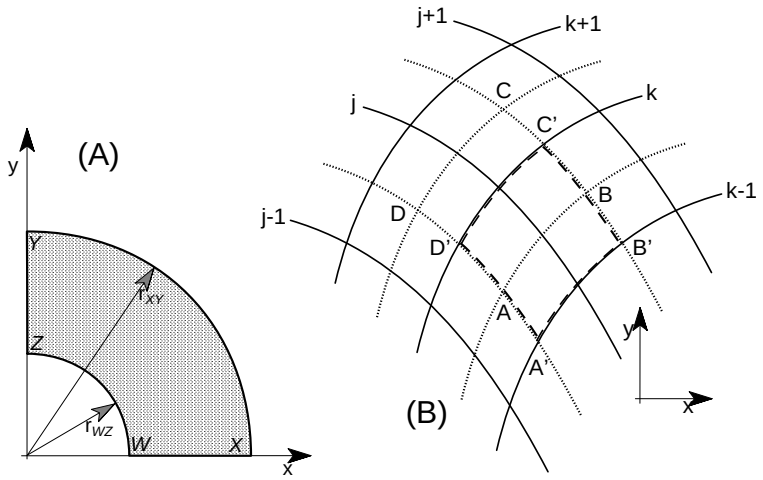


Figure 6.2: Illustrations of the domain for the solution of Laplace's equation (A), and of the grid geometry to evaluate the second derivatives for the finite volume method.

We are seeking a solution for this equation in a domain as illustrated by the shaded area in Figure~6.2. The appropriate coordinate system for this domain is a polar coordinate system with the variable r and θ. The boundary conditions are

$$\phi = 0 \text{ at boundary } \overline{WX},$$

$$\phi = \sin\theta / r_{xy} \text{ at boundary } \overline{XY},$$

$$\phi = 1 / r_{yz} \text{ at boundary } \overline{YZ},$$

$$\phi = \sin\theta / r_{wz} \text{ at boundary } \overline{WZ}.$$

With this choice of boundary conditions Laplace's equation (6.13) has the exact solution

$$\phi = \frac{\sin\theta}{r}$$

To apply the finite volume method we determine the integral of the residual in the area given by $\overline{ABCD}$ in Figure 6.2

$$\int\int_{\overline{ABCD}} \left(\frac{\partial^2\phi}{\partial x^2} + \frac{\partial^2\phi}{\partial y^2} \right) dxdy = \oint_{\partial ABCD} \mathbf{H}\cdot\mathbf{n}ds = 0 \tag{6.14}$$

with $\mathbf{H}\cdot\mathbf{n}ds = \frac{\partial\phi}{\partial x}dy - \frac{\partial\phi}{\partial y}dx$. Note that $\mathbf{H} = (\partial\phi/\partial x, \partial\phi/\partial y)$ and that the direction normal to $d\mathbf{s} = (dx, dy)$ is $\mathbf{n}ds = (dy, -dx)$. Using the geometry shown in Figure 6.2 equation (6.14) is approximated with

$$\begin{aligned}
&\left.\frac{\partial\phi}{\partial x}\right|_{j,k-1/2} \Delta y_{AB} - \left.\frac{\partial\phi}{\partial y}\right|_{j,k-1/2} \Delta x_{AB} \\
&\quad + \left.\frac{\partial\phi}{\partial x}\right|_{j+1,k} \Delta y_{BC} - \left.\frac{\partial\phi}{\partial y}\right|_{j+1,k} \Delta x_{BC} \\
&+ \left.\frac{\partial\phi}{\partial x}\right|_{j,k+1/2} \Delta y_{CD} - \left.\frac{\partial\phi}{\partial y}\right|_{j,k+1/2} \Delta x_{CD} \\
&\quad + \left.\frac{\partial\phi}{\partial x}\right|_{j-1,k} \Delta y_{DA} - \left.\frac{\partial\phi}{\partial y}\right|_{j-1,k} \Delta x_{DA} \;=\; 0
\end{aligned} \tag{6.15}$$

The derivatives $\partial\phi/\partial x$ and $\partial\phi/\partial y$ at the midpoints of each section $\overline{AB}$, $\overline{BC}$, etc are determined through averages over the appropriate section. For instance along $\overline{AB}$ the derivatives are

$$\begin{aligned}
\left.\frac{\partial\phi}{\partial x}\right|_{j,k-1/2} &= \frac{1}{S_{AB}} \int\int_{\overline{A'B'C'D'}} \frac{\partial\phi}{\partial x} dxdy \;=\; \frac{1}{S_{AB}} \oint_{\partial A'B'C'D'} \phi dy \\
\left.\frac{\partial\phi}{\partial y}\right|_{j,k-1/2} &= \frac{1}{S_{AB}} \int\int_{\overline{A'B'C'D'}} \frac{\partial\phi}{\partial y} dxdy \;=\; -\frac{1}{S_{AB}} \oint_{\partial A'B'C'D'} \phi dx
\end{aligned}$$

with S_{AB} as the area of $\overline{A'B'C'D'}$ and

$$\begin{aligned}
\oint_{\partial A'B'C'D'} \phi dy &= \phi_{j,k-1}\Delta y_{A'B'} + \phi_B \Delta y_{B'C'} + \phi_{j,k}\Delta y_{C'D'} + \phi_A \Delta y_{D'A'} \\
\oint_{\partial A'B'C'D'} \phi dx &= \phi_{j,k-1}\Delta x_{A'B'} + \phi_B \Delta x_{B'C'} + \phi_{j,k}\Delta x_{C'D'} + \phi_A \Delta x_{D'A'}
\end{aligned}$$

If the mesh is not too distorted we can approximate

$$\begin{aligned}
\Delta y_{A'B'} &\approx -\Delta y_{C'D'} \approx \Delta y_{AB} \\
\Delta y_{B'C'} &\approx -\Delta y_{D'A'} \approx \Delta y_{k-1,k}
\end{aligned}$$

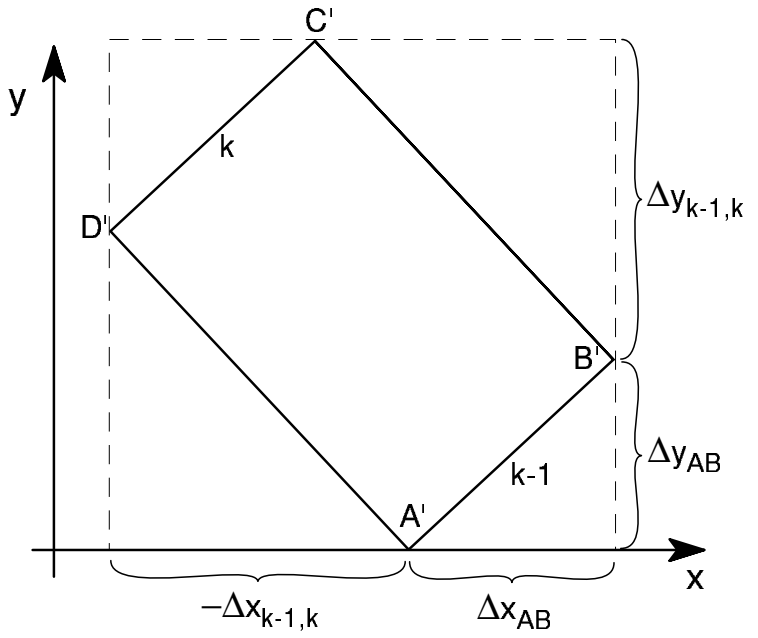


Figure 6.3: Illustration of the geometry of the elements $\overline{ABCD}$.

and similar for Δx. Figure 6.3 shows that the area is approximated by

$$\begin{aligned} S_{AB} = S_{A'B'C'D'} &= (\Delta x_{AB} - \Delta x_{k-1,k})(\Delta y_{AB} + \Delta y_{k-1,k}) \\ &\quad -\Delta x_{AB}\Delta y_{AB} + \Delta x_{k-1,k}\Delta y_{k-1,k} \\ &= \Delta x_{AB}\Delta y_{k-1,k} - \Delta y_{AB}\Delta x_{k-1,k} \end{aligned}$$

With this we obtain:

$$\begin{aligned} \left.\frac{\partial\phi}{\partial x}\right|_{j,k-1/2} &= \frac{\Delta y_{AB}\left(\phi_{j,k-1} - \phi_{j,k}\right) + \Delta y_{k-1,k}\left(\phi_B - \phi_A\right)}{S_{AB}} \\ \left.\frac{\partial\phi}{\partial y}\right|_{j,k-1/2} &= -\frac{\Delta x_{AB}\left(\phi_{j,k-1} - \phi_{j,k}\right) + \Delta x_{k-1,k}\left(\phi_B - \phi_A\right)}{S_{AB}} \end{aligned}$$

Similarly we obtain

$$\begin{aligned}
\left.\frac{\partial\phi}{\partial x}\right|_{j+1/2,k} &= \frac{\Delta y_{BC}\left(\phi_{j+1,k}-\phi_{j,k}\right)+\Delta y_{j+1,j}\left(\phi_C-\phi_B\right)}{S_{BC}} \\
\left.\frac{\partial\phi}{\partial y}\right|_{j+1/2,k} &= -\frac{\Delta x_{BC}\left(\phi_{j+1,k}-\phi_{j,k}\right)+\Delta x_{j+1,j}\left(\phi_C-\phi_B\right)}{S_{BC}} \\
\left.\frac{\partial\phi}{\partial x}\right|_{j,k+1/2} &= \frac{\Delta y_{CD}\left(\phi_{j,k+1}-\phi_{j,k}\right)+\Delta y_{k+1,k}\left(\phi_D-\phi_C\right)}{S_{CD}} \\
\left.\frac{\partial\phi}{\partial y}\right|_{j,k+1/2} &= -\frac{\Delta x_{CD}\left(\phi_{j,k+1}-\phi_{j,k}\right)+\Delta x_{k+1,k}\left(\phi_D-\phi_C\right)}{S_{CD}} \\
\left.\frac{\partial\phi}{\partial x}\right|_{j-1/2,k} &= \frac{\Delta y_{DA}\left(\phi_{j-1,k}-\phi_{j,k}\right)+\Delta y_{j-1,j}\left(\phi_A-\phi_D\right)}{S_{DA}} \\
\left.\frac{\partial\phi}{\partial y}\right|_{j-1/2,k} &= -\frac{\Delta x_{DA}\left(\phi_{j-1,k}-\phi_{j,k}\right)+\Delta x_{j-1,j}\left(\phi_A-\phi_D\right)}{S_{DA}}
\end{aligned}$$

Substitution back into equation (6.15) gives

$$\begin{aligned}
Q_{AB}\left(\phi_{j,k-1}-\phi_{j,k}\right)+Q_{BC}\left(\phi_{j+1,k}-\phi_{j,k}\right)+Q_{CD}\left(\phi_{j,k+1}-\phi_{j,k}\right)+Q_{DA}\left(\phi_{j-1,k}-\phi_{j,k}\right) & \\
+P_{AB}\left(\phi_B-\phi_A\right)+P_{BC}\left(\phi_C-\phi_B\right)+P_{CD}\left(\phi_D-\phi_C\right)+P_{DA}\left(\phi_A-\phi_D\right) &= 0
\end{aligned}$$

with

$$\begin{aligned}
Q_{AB} &= \left(\Delta x_{AB}^2+\Delta y_{AB}^2\right)/S_{AB} \quad , \quad P_{AB} = \left(\Delta x_{AB}\Delta x_{k-1,k}+\Delta y_{AB}\Delta y_{k-1,k}\right)/S_{AB} \\
Q_{BC} &= \left(\Delta x_{BC}^2+\Delta y_{BC}^2\right)/S_{BC} \quad , \quad P_{BC} = \left(\Delta x_{BC}\Delta x_{j+1,j}+\Delta y_{BC}\Delta y_{j+1,j}\right)/S_{BC} \\
Q_{CD} &= \left(\Delta x_{CD}^2+\Delta y_{CD}^2\right)/S_{CD} \quad , \quad P_{CD} = \left(\Delta x_{CD}\Delta x_{k+1,k}+\Delta y_{CD}\Delta y_{k+1,k}\right)/S_{CD} \\
Q_{DA} &= \left(\Delta x_{DA}^2+\Delta y_{DA}^2\right)/S_{DA} \quad , \quad P_{DA} = \left(\Delta x_{DA}\Delta x_{j-1,j}+\Delta y_{DA}\Delta y_{j-1,j}\right)/S_{DA}
\end{aligned}$$

Finally we evaluate ϕ_A, x_A, and y_A as the average over the surrounding nodes, e.g.,

$$\phi_A = 0.25(\phi_{j,k}+\phi_{j-1,k}+\phi_{j-1,k-1}+\phi_{j,k-1})$$

Substitution into our main equation then yields

$$\begin{aligned}
0.25\left(P_{CD}-P_{DA}\right)\phi_{j-1,k+1}+0.25\left(P_{BC}-P_{CD}\right)\phi_{j+1,k+1} & \\
+0.25\left(P_{AB}-P_{BC}\right)\phi_{j+1,k-1}+0.25\left(P_{DA}-P_{AB}\right)\phi_{j-1,k-1} & \\
+\left[Q_{CD}+0.25\left(P_{BC}-P_{DA}\right)\right]\phi_{j,k+1}+\left[Q_{DA}+0.25\left(P_{CD}-P_{AB}\right)\right]\phi_{j-1,k} & \\
+\left[Q_{AB}+0.25\left(P_{DA}-P_{BC}\right)\right]\phi_{j,k-1}+\left[Q_{BC}+0.25\left(P_{AB}-P_{CD}\right)\right]\phi_{j+1,k} & \\
-\left(Q_{AB}+Q_{BC}+Q_{CD}+Q_{DA}\right)\phi_{j,k} &= 0 \qquad (6.16)
\end{aligned}$$

Here the coefficients can be determined initially and then used during the computation. Equation (6.16) is solved in the program FIVOL using successive over-relaxation (SOR). The estimate for ϕ is determined from (6.16)

$$\begin{aligned}\phi^{\star}_{j,k} &= \{0.25\,(P_{CD}-P_{DA})\,\phi_{j-1,k+1}+0.25\,(P_{BC}-P_{CD})\,\phi_{j+1,k+1} \\ &\quad +0.25\,(P_{AB}-P_{BC})\,\phi_{j+1,k-1}+0.25\,(P_{DA}-P_{AB})\,\phi_{j-1,k-1} \\ &\quad +[Q_{CD}+0.25\,(P_{BC}-P_{DA})]\,\phi_{j,k+1}+[Q_{DA}+0.25\,(P_{CD}-P_{AB})]\,\phi_{j-1,k} \\ &\quad +[Q_{AB}+0.25\,(P_{DA}-P_{BC})]\,\phi_{j,k-1}+[Q_{BC}+0.25\,(P_{AB}-P_{CD})]\,\phi_{j+1,k}\}^{n} \\ &\quad /\,(Q_{AB}+Q_{BC}+Q_{CD}+Q_{DA})\end{aligned} \tag{6.17}$$

The iteration step is completed with

$$\phi^{n+1}_{j,k}=\phi^{n}_{j,k}+\omega(\phi^{\star}_{j,k}-\phi^{n}_{j,k}) \tag{6.18}$$

Note that the discretized equation (6.16) reduces to centered finite differences on a uniform rectangular grid

$$\frac{\phi_{j-1,k}-2\phi_{j,k}+\phi_{j+1,k}}{\Delta x^2}+\frac{\phi_{j,k-1}-2\phi_{j,k}+\phi_{j,k+1}}{\Delta y^2}=0 \tag{6.19}$$

6.2.3 Program Fivol

The finite volume method as described above is implemented in the program Fivol. The program parameter used are summarized in Table (6.1).

Table 6.1: Program parameter for the program Fivol.

Parameter	Description
nr,ntheta	Number of grid points in r and θ directions
niter	Maximum number of iterations
eps	Tolerance for the iterated solution
om	Relaxation parameter ω
rms	RMS error
rw, rx, ry, rz	radial distance to points w, x, y, and z
theb, then	Min and max values in the θ direction
x,y	x and y coordinates of the grid
r,theta	r and θ coordinates
dr,dtheta	increments in the r and θ directions
qab, pab, qbc, pbc, qcd, pcd, qda, pda	Weights for the iterations
phi	Iterated solution
phix	Exact solution

The program reads the input parameters niter, rw, rx, ry, rz, theb, then, eps, om from the data file fivol.dat. Program parameters are defined in the include file fivin. The program then writes parameters to a data file fivol.out, generates the grid, initializes the iteration through an initial state (for phi), boundary conditions, and calculates the matrix coefficients. Subsequently a subroutine

SOR is called until the iterated solution tolerance is reached. The result is written to a binary file which can serve as input for the plotting routine plofivol.

The solution error for the finite volume method is listed in Table (6.2) for different number of grid points in the same domain. The table illustrates that the solution error is second order. This is to be expected from the centered differences to which the finite volume method reduces on a uniform rectangular grid. However, for strongly distorted grids the solution error is larger than second order.

Table 6.2: Solution errors for the finite volume method in program Fivol2.f

Grid	$\lvert\phi - \phi_{exact}\rvert_{rms}$	No of iterations
6×6	0.1326	15
11×11	0.0471	19
21×21	0.0138	51

The finite volume method is well suited for for somewhat irregular grid domains and does not require an orthogonal grid. The number of iterations for convergence depends on the domain size.

6.3 Finite Element Method

Finite element methods are used mostly in engineering. For many problems the finite element method can be interpreted as a maximization of the potential energy of a system. In most applications the finite element method is used with the Galerkin formulation for the weighted residuals. The approximating functions are simple polynomials defined in local domains.

$$T = \sum_{j=1}^{N} T_j \phi_j(x, y, z) \tag{6.20}$$

Where for a suitable set of functions the local domains can be of any shape. Since the finite element method is used with local coordinates the domains can be subdivided (into same shape domains) to increase the resolution where it is desired. The interpolating functions are called trial or shape functions.

6.3.1 Basic formulation

a) Linear Interpolation

The linear interpolation uses linear functions which are 1 at the nodal point, assume 0 at the neighboring points, and are identical to 0 outside the domains (x_{j-1}, x_{j+1}) as illustrated in Figure 6.4.

The shape function for the nodal point j is

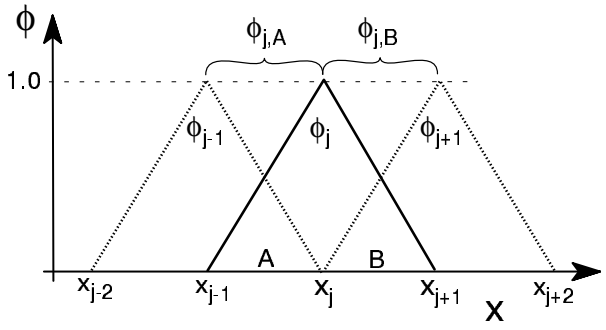


Figure 6.4: Illustration of one-dimensional linear finite elements.

$$\phi_j = \begin{cases} \frac{x-x_{j-1}}{x_j-x_{j-1}} & for \quad x_{j-1} \le x \le x_j \ (element\, A) \\ \frac{x_{j+1}-x}{x_{j+1}-x_j} & for \quad x_j \le x \le x_{j+1} \ (element\, B) \end{cases} \tag{6.21}$$

and $\phi_j = 0$ outside of elements A and B. The shape function ϕ_j overlaps only with its direct neighbors and the superposition of elements between nodal points yields a linear function in these regions

$$in\, A: \qquad T = T_{j-1}\phi_{j-1} + T_j\phi_j \tag{6.22}$$

$$in\, B: \qquad T = T_j\phi_j + T_{j+1}\phi_{j+1} \tag{6.23}$$

In any given element only two shape functions overlap. In element A ϕ_j is given element by (6.21) and ϕ_{j-1} is given by

$$\phi_{j-1} = \frac{x_j - x}{x_j - x_{j-1}}$$

Similar for any element B ϕ_j is given element by (6.21) and ϕ_{j+1} is given by

$$\phi_{j+1} = \frac{x - x_j}{x_{j+1} - x_j}.$$

The particular form of the trial functions makes it straightforward to use them to approximate any given function $f(x)$. Since the trial functions are 1 at the nodal points and all except for one trial functions are 0 at any nodal point Thus the expansion of a function $f(x)$ in terms of the shape functions is given by

$$f(x) = \sum_{j=1}^{N} f_j\phi_j(x) \tag{6.24}$$

with the coefficients

$$f_k = f(x_k) \tag{6.25}$$

where the x_k are the nodal points. Formally this is seen by $f(x_k) = \sum_{j=1}^{N} f_j \phi_j(x_k) = f_k$
The method is illustrated using the function

$$f(x) = 1 + \cos(\pi x/2) + \sin(\pi x) \tag{6.26}$$

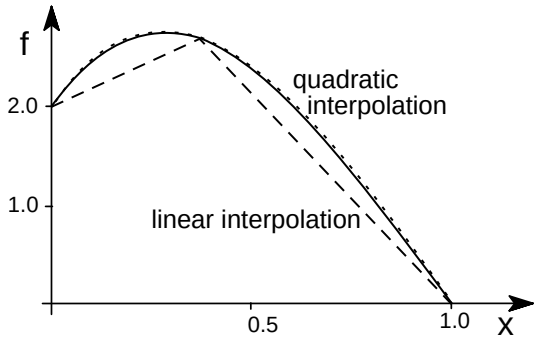


Figure 6.5: Linear (dashed)and quadratic (dotted) finite element approximation of function (6.26).

in the range $[0, 1]$. With two elements and nodal points at $(0, 0.5, 1.0)$ the expansion coefficients are $f_1 = 2$, $f_2 = 2.7071$, and $f_3 = 0$. Figure (6.5) shows the function $f(x)$ and the finite element interpolation for linear and quadratic elements.

b) Quadratic interpolation

Quadratic interpolation requires simple quadratic polynomials for trial function. Again a trial function should be 1 only at the corresponding nodal point and 0 at all other nodes.

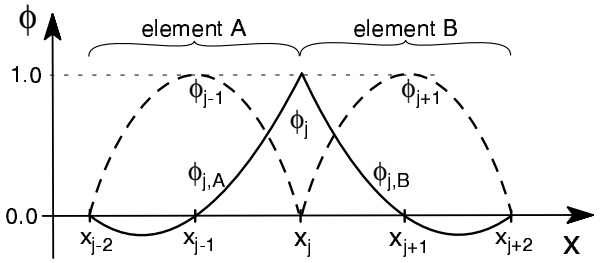


Figure 6.6: Illustration of one-dimensional linear finite elements.

Following the illustration 6.6 the trial functions are defined by

$$\phi_j = \begin{cases} 0 & for \quad x \le x_{j-2},\, x \ge x_{j+2} \\ \frac{x-x_{j-2}}{x_j-x_{j-2}} \frac{x-x_{j-1}}{x_j-x_{j-1}} & for \quad x_{j-2} \le x \le x_j, \quad element\, A \\ \frac{x-x_{j+2}}{x_j-x_{j+2}} \frac{x-x_{j+1}}{x_j-x_{j+1}} & for \quad x_j \le x \le x_{j+2}, \quad element\, B \end{cases} \tag{6.27}$$

With this form $\phi_j(x_j) = 1$ and $\phi_j(x_{j-2}) = \phi_j(x_{j-1}) = \phi_j(x_{j+1}) = \phi_j(x_{j+1}) = 0$. The trial functions of this form extend over 5 nodes. such that in a given interval 4 trial functions overlap

if they were all chosen of the same form. This is improved by choosing different trial functions at x_{j-1} and at x_{j+1} of the form:

$$\phi_j = \frac{x - x_{j-1}}{x_j - x_{j-1}} \frac{x - x_{j+1}}{x_j - x_{j+1}} \; for \; x_{j-2} \leq x \leq x_j \tag{6.28}$$

and 0 otherwise. Note that this creates a structure where all odd nodes have elements of the form of (6.27) and all even elements are of the form (6.28) where we start with an index of 1 for the first node (x_{min} boundary).

Similar to the linear elements it is straightforward to expand any given function in term of these shape functions

$$f(x) = \sum_{j=1}^{N} f_j \phi_j(x)$$

with the coefficients $f_k = f(x_k)$ as in the case of the linear elements.

with $T_j = f(x_j)$. With these functions the interpolation in elements A and B take the following form

$$T = \begin{cases} T_{j-2}\phi_{j-2} + T_{j-1}\phi_{j-1} + T_j\phi_j & in\, element\, A \\ T_j\phi_j + T_{j+1}\phi_{j+1} + T_{j+2}\phi_{j+2} & in\, element\, B \end{cases}$$

The specific form of the shape functions which are nonzero in any element A is

$$\begin{aligned} \phi_{j-2} &= \left(\frac{x - x_j}{x_{j-2} - x_j}\right)\left(\frac{x - x_{j-1}}{x_{j-2} - x_{j-1}}\right) \\ \phi_{j-1} &= \left(\frac{x - x_{j-2}}{x_{j-1} - x_{j-2}}\right)\left(\frac{x - x_j}{x_{j-1} - x_j}\right) \\ \phi_j &= \left(\frac{x - x_{j-2}}{x_j - x_{j-2}}\right)\left(\frac{x - x_{j-1}}{x_j - x_{j-1}}\right) \end{aligned}$$

For the special function (6.26) with only three nodes we have only one element A and no element B. As before the expansion coefficients are $f_1 = 2$, $f_2 = 2.7071$, and $f_3 = 0$.

The errors for the linear and quadratic finite element approximation of function (6.26) is listed in Table 6.3. While the linear approximation scales quadratic in the error the quadratic interpolation scales cubic in the resolution (inverse number of nodes).

c) Two-dimensional interpolation

Linear elements: The particular strength of finite elements is there flexibility in two and three dimensions. Based on the previous introduction we want to illustrate the use of linear and quadratic

Table 6.3: Error for linear and quadratic finite element interpolation for the function in (6.26).

Linear Interpolation		Quadratic interpolation	
No of elements	RMS error	No of elements	RMS error
2	0.18662	1	0.04028
4	0.04786	2	0.01599
6	0.02138	3	0.00484
8	0.01204	4	0.00206

finite elements in two and three dimensions. In two dimensions a trial function centered at (x_j, y_j) spans four elements A, B, C, and D. The approximate solution in this region can be conveniently written with local element based coordinates (ξ, η) as

$$T = \sum_{l=1}^{4} T_l \phi_l(\xi, \eta) \tag{6.29}$$

with $-1 \leq \xi \leq 1$ and $-1 \leq \eta \leq 1$. The approximating functions $\phi_l(\xi, \eta)$ in each element are of the form

$$\phi_l(\xi, \eta) = 0.25(1 + \xi_l \xi)(1 + \eta_l \eta) \tag{6.30}$$

with $\xi_l = \pm 1$ and $\eta_l = \pm 1$ or explicit

$$\begin{aligned} \phi_1 &= 0.25(1-\xi)(1-\eta) \\ \phi_2 &= 0.25(1+\xi)(1-\eta) \\ \phi_3 &= 0.25(1+\xi)(1+\eta) \\ \phi_4 &= 0.25(1-\xi)(1+\eta) \end{aligned}$$

The nodal geometry and local coordinate systems are illustrated in Figure 6.7.

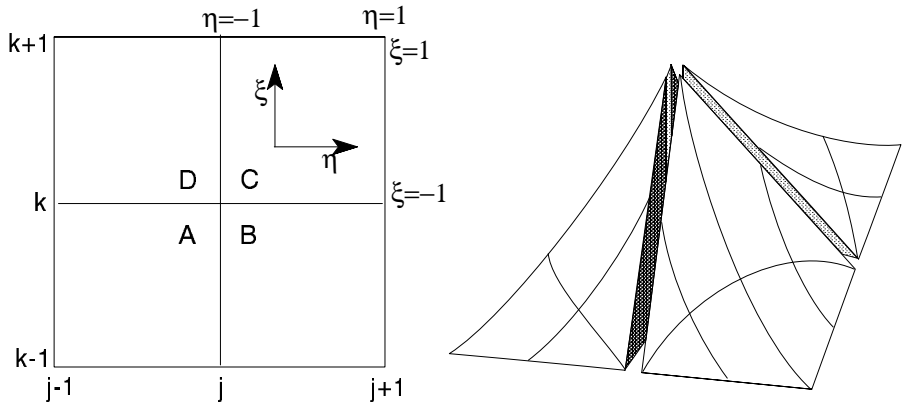


Figure 6.7: Illustration of the nodal geometry and sketch of the two-dimensional linear trial functions.

A solution is constructed separate in each element A, B, C, and D where continuity is provided by the overlapping shape functions. For instance a solution in element A implies that shape functions with values of 1 along the boundary to element B overlap into the region of element B and similar for all other boundaries of element A.

Bi-quadratic elements:

Similar to the bilinear elements the approximate solution for bi-quadratic elements in this region can be conveniently written with local element based coordinates (ξ, η) as

$$T = \sum_{l=1}^{9} T_l \phi_l(\xi, \eta) \tag{6.31}$$

with $-1 \leq \xi \leq 1$ and $-1 \leq \eta \leq 1$. The approximating functions $\phi_l(\xi\,\eta)$ in each element depend on the location.

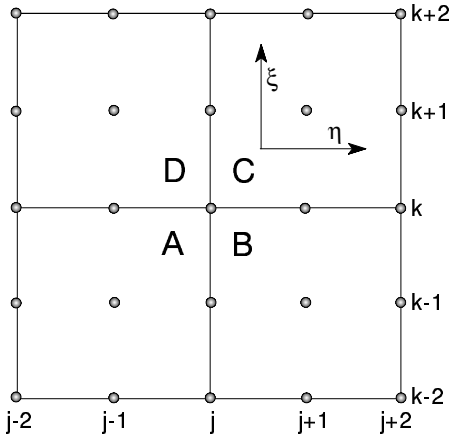


Figure 6.8: Illustration of the nodal geometry for bi-quadratic elements.

Specifically the form of the shape function depends on how they are centered, i.e., where they assume a value of 1.

$$\phi_l(\xi, \eta) = 0.25\xi_l\xi(1 + \xi_l\xi)\eta_l\eta(1 + \eta_l\eta) \quad \text{corner nodes} \tag{6.32}$$

$$\phi_l(\xi, \eta) = 0.5(1 - \xi^2)\eta_l\eta(1 + \eta_l\eta) \quad \text{midside nodes } (\xi_l = 0) \tag{6.33}$$

$$\phi_l(\xi, \eta) = 0.5(1 - \eta^2)\xi_l\xi(1 + \xi_l\xi) \quad \text{midside nodes } (\eta_l = 0) \tag{6.34}$$

$$\phi_l(\xi, \eta) = 0.5(1 - \eta^2)(1 - \xi^2) \quad \text{internal nodes} \tag{6.35}$$

Again solutions are constructed in each element A, B, C, and D where the shape functions on the side and corner nodes overlap into the adjacent region. Note that quadratic elements require at 3 nodes in each direction and the total number of nodes has to be odd to accommodate quadratic elements.

6.3.2 Finite element method applied to the Sturm-Liouville equation

Here we will use the Galerkin finite element method to discretize and solve the Sturm-Liouville equation

$$\frac{d^2y}{dx^2} + y = F(x) \tag{6.36}$$

subject to boundary conditions $y(0) = 0$ and $dy/dx(1) = 0$ and

$$F(x) = -\sum_{l=1}^{L} a_l \sin\left((l - 0.5)\pi x\right). \tag{6.37}$$

The exact solution to this problem for $F(x)$ is given by

$$y(x) = -\sum_{l=1}^{L} \frac{a_l}{1 - \left((l - 0.5)\pi\right)^2} \sin\left((l - 0.5)\pi x\right). \tag{6.38}$$

The trial solution is as before

$$\widetilde{y} = \sum_{j=1}^{N} y_j \phi_j.$$

Linear Interpolation:

In an element based coordinate system we use local coordinates with

$$\phi_j = 0.5(1 + \xi) \; and \quad \xi = \frac{2\left(x - \frac{x_{j-1}+x_j}{2}\right)}{\Delta x_j} \qquad for\, element\, A$$
$$\phi_j = 0.5(1 - \xi) \; and \quad \xi = \frac{2\left(x - \frac{x_j+x_{j+1}}{2}\right)}{\Delta x_{j+1}} \qquad for\, element\, B$$

with $\Delta x_j = x_j - x_{j-1}$, and $\Delta x_{j+1} = x_{j+1} - x_j$. The residual is

$$R = \frac{d^2\widetilde{y}}{dx^2} + \widetilde{y} - F(x)$$

and the weight function for the weighted residual is $w_m = \phi_m$ which yields the equation

$$\int_0^1 \phi_m(x) \left(\frac{d^2\widetilde{y}}{dx^2} + \widetilde{y} - F(x)\right) dx = 0$$

One can integrate the first term brackets by parts

$$\int_0^1 \phi_m \frac{d^2\widetilde{y}}{dx^2} dx = \left[\phi_m \frac{d\widetilde{y}}{dx}\right]_0^1 - \int_0^1 \frac{d\phi_m}{dx}\frac{d\widetilde{y}}{dx} dx$$

because the boundary conditions imply $\phi_1 = 0$ and $dy/dx(1) = 0$ such that $\left[\phi_m \frac{d\tilde{y}}{dx}\right]_0^1 = 0$ such that the residual equation becomes

$$\sum_{j=1}^{N} \left[y_j \int_0^1 \left(-\frac{d\phi_m}{dx}\frac{d\phi_j}{dx} + \phi_m \phi_j \right) dx \right] = \int_0^1 \phi_m F(x) dx$$

or in matrix form

$$\underline{\underline{\mathbf{B}}}\mathbf{Y} = \mathbf{G}$$

with the elements

$$\begin{aligned} b_{mj} &= \int_0^1 \left(-\frac{d\phi_m}{dx}\frac{d\phi_j}{dx} + \phi_m \phi_j \right) dx \\ g_m &= \int_0^1 \phi_m F(x) dx. \end{aligned}$$

In the computation of these elements it is convenient to make use of the local coordinates. It should also be noted that the residual equation only has contributions for $j = m-1$, $j = m$, and $m = j+1$. With the transformation to local coordinates we have in element A

$$\begin{aligned} \frac{d}{dx} &= \frac{d\xi}{dx}\frac{d}{d\xi} = \frac{2}{\Delta x_j}\frac{d}{d\xi} \\ dx &= \frac{dx}{d\xi} d\xi = \frac{\Delta x_j}{2} d\xi \end{aligned}$$

and in element B

$$\begin{aligned} \frac{d}{dx} &= \frac{d\xi}{dx}\frac{d}{d\xi} = \frac{2}{\Delta x_{j+1}}\frac{d}{d\xi} \\ dx &= \frac{dx}{d\xi} d\xi = \frac{\Delta x_{j+1}}{2} d\xi \end{aligned}$$

with $\Delta x_j = x_j - x_{j-1}$ and $\Delta x_{j+1} = x_{j+1} - x_j$.

Element $b_{m,m-1}$: A contribution exists only in region A for node m (corresponding to a region B for node $m-1$). Thus

$$\begin{aligned} b_{m,m-1} &= \int_0^1 \left(-\frac{d\phi_m}{dx}\frac{d\phi_{m-1}}{dx} + \phi_m \phi_{m-1} \right) dx \\ &= \int_{-1}^1 \left(-\frac{d\xi_A}{dx}\frac{d\phi_A}{d\xi}\frac{d\phi_B}{d\xi} + \frac{dx}{d\xi_A}\phi_A \phi_B \right) d\xi \\ &= \frac{1}{2\Delta x_m}\int_{-1}^1 d\xi + \frac{\Delta x_m}{8}\int_{-1}^1 (1-\xi^2) d\xi \\ &= \frac{1}{\Delta x_m} + \frac{\Delta x_m}{6} \end{aligned}$$

Similar the element $b_{m,m+1}$ involves only overlap in region B of element ϕ_m with region A of element ϕ_{m+1}. The expression is the same only that it regards the interval Δx_{m+1}. Thus

$$b_{m,m+1} = \frac{1}{\Delta x_{m+1}} + \frac{\Delta x_{m+1}}{6}$$

Element $b_{m,m}$: Here we have overlap of ϕ_m with itself in regions A and B:

$$\begin{aligned}
b_{m,m} &= \int_0^1 \left(-\frac{d\phi_m}{dx}\frac{d\phi_m}{dx} + \phi_m\phi_m \right) dx \\
&= \int_{-1}^1 \left(-\frac{d\xi}{dx}\left(\frac{d\phi_A}{d\xi}\right)^2 + \frac{dx}{d\xi}\phi_A^2 \right)_A d\xi + \int_{-1}^1 \left(-\frac{d\xi}{dx}\left(\frac{d\phi_B}{d\xi}\right)^2 + \frac{dx}{d\xi}\phi_B^2 \right)_B d\xi \\
&= \int_{-1}^1 \left(-\frac{2}{\Delta x_m}\left(\frac{1}{2}\right)^2 + \frac{\Delta x_m}{2}\left(\frac{1+\xi}{2}\right)^2 \right)_A d\xi \\
&\quad + \int_{-1}^1 \left(-\frac{2}{\Delta x_{m+1}}\left(\frac{1}{2}\right)^2 + \frac{\Delta x_{m+1}}{2}\left(\frac{1-\xi}{2}\right)^2 \right)_A d\xi \\
&= -\frac{1}{\Delta x_m} - \frac{1}{\Delta x_{m+1}} + \frac{\Delta x_m}{8}\int_{-1}^1 (1+\xi)^2 d\xi + \frac{\Delta x_{m+1}}{8}\int_{-1}^1 (1-\xi)^2 d\xi \\
&= -\frac{1}{\Delta x_m} - \frac{1}{\Delta x_{m+1}} + \frac{\Delta x_m + \Delta x_{m+1}}{3}
\end{aligned}$$

In summary the nonzero elements b_{mj} for $2 \le m \le N-1$ are

$$\begin{aligned}
b_{m,m-1} &= \frac{1}{\Delta x_m} + \frac{\Delta x_m}{6} \\
b_{m,m} &= -\left[\frac{1}{\Delta x_m} + \frac{1}{\Delta x_{m+1}} \right] + \frac{\Delta x_m + \Delta x_{m+1}}{3} \\
b_{m,m+1} &= \frac{1}{\Delta x_{m+1}} + \frac{\Delta x_{m+1}}{6}
\end{aligned}$$

for $m = N$

$$\begin{aligned}
b_{N,N-1} &= \frac{1}{\Delta x_N} + \frac{\Delta x_N}{6} \\
b_{N,N} &= -\frac{1}{\Delta x_N} + \frac{\Delta x_N}{6} \\
b_{N,N+1} &= 0.
\end{aligned}$$

The inhomogeneity $F(x)$ is known analytically such that one can evaluate $\int_0^1 \phi_m F(x)dx$ directly. However, in more complex situations it is more convenient to interpolate $F(x)$ through the trial functions

$$F(x) = \sum_{j=1}^{N} F_j \phi_j.$$

such that

$$g_m = \sum_{j=1}^{N} F_j \int_0^1 \phi_m \phi_j dx$$

For the linear interpolation this yields

$$g_m = \frac{\Delta x_m}{6} F_{m-1} + \frac{\Delta x_m + \Delta x_{m+1}}{3} F_m + \frac{\Delta x_{m+1}}{6} F_{m+1}$$

Finally consider the special case of a uniform grid with $\Delta x_m = \Delta x_{m+1} = \Delta x$. In this case the equation for the coefficients becomes

$$\frac{y_{m-1} - 2y_m + y_{m+1}}{\Delta x^2} + \left(\frac{1}{6} y_{m-1} + \frac{2}{3} y_m + \frac{1}{6} y_{m+1}\right) = \left(\frac{1}{6} F_{m-1} + \frac{2}{3} F_m + \frac{1}{6} F_{m+1}\right)$$

Exercise: Determine the elements at the min and max boundary, i.e., $b_{1,1}$, $b_{1,2}$, and $b_{2,1}$.

Elements with $m = 1$ ($y_1 = 0$): There is no equation for y_1 needed because $y_1 = 0$ such that the indices for the array dimensions decrease by 1.

Elements with $m = 2$

$$y_2 \int_0^1 \left(-\frac{d\phi_2}{dx}\frac{d\phi_2}{dx} + \phi_2\phi_2\right) dx + y_3 \int_0^1 \left(-\frac{d\phi_2}{dx}\frac{d\phi_3}{dx} + \phi_2\phi_3\right) dx = \int_0^1 \phi_2 F(x) dx$$

Elements with $m = N$

$$y_{N-1} \int_0^1 \left(-\frac{d\phi_N}{dx}\frac{d\phi_{N-1}}{dx} + \phi_N\phi_{N-1}\right) dx + y_N \int_0^1 \left(-\frac{d\phi_N}{dx}\frac{d\phi_N}{dx} + \phi_N\phi_N\right) dx = \int_0^1 \phi_N F(x) dx$$

Quadratic interpolation:

The program offers the possibility to use a linear or quadratic finite element interpolation. For the second case the equations for the b_{mj} are

$$
\begin{aligned}
b_{m,m-2} &= -\frac{1}{6\Delta x_m} - \frac{\Delta x_m}{15} \\
b_{m,m-1} &= \frac{4}{3\Delta x_m} + \frac{2\Delta x_m}{15} \\
b_{m,m} &= -\frac{7}{6}\left[\frac{1}{\Delta x_m} + \frac{1}{\Delta x_{m+1}}\right] + \frac{4}{15}\left(\Delta x_m + \Delta x_{m+1}\right) \\
b_{m,m+1} &= \frac{4}{3\Delta x_{m+1}} + \frac{2\Delta x_{m+1}}{15} \\
b_{m,m+1} &= -\frac{1}{6\Delta x_{m+1}} + \frac{\Delta x_{m+1}}{15}
\end{aligned}
$$

and

$$
\begin{aligned}
g_m &= -\frac{\Delta x_m}{15}F_{m-2} + \frac{2\Delta x_m}{15}F_{m-1} + \frac{4}{15}\left(\Delta x_m + \Delta x_{m+1}\right)F_m \\
&\quad + \frac{2\Delta x_{m+1}}{15}F_{m+1} - \frac{\Delta x_m}{15}F_{m+2}
\end{aligned}
$$

A the boundary $m = N$ one obtains

$$
\begin{aligned}
b_{N,N-2} &= -\frac{1}{6\Delta x_N} - \frac{\Delta x_N}{15} \\
b_{N,N-1} &= \frac{4}{3\Delta x_N} + \frac{2\Delta x_N}{15} \\
b_{N,N} &= -\frac{7}{6}\frac{1}{\Delta x_N} + \frac{4}{15}\Delta x_N \\
g_N &= -\frac{\Delta x_N}{15}F_{N-2} + \frac{2\Delta x_N}{15}F_{N-1} + \frac{4}{15}\Delta x_N F_N
\end{aligned}
$$

For the equations at midside nodes one obtains

$$
\begin{aligned}
b_{m,m-1} &= \frac{4}{3\Delta x_m} + \frac{2\Delta x_m}{15} \\
b_{m,m} &= -\frac{8}{3\Delta x_m} + \frac{16\Delta x_m}{15} \\
b_{m,m+1} &= \frac{4}{3\Delta x_{m+1}} + \frac{2\Delta x_{m+1}}{15} \\
g_m &= \frac{2\Delta x_m}{15}F_{m-1} + \frac{16\Delta x_m}{15}F_m + \frac{2\Delta x_m}{15}F_{m+1}
\end{aligned}
$$

The program Sturm solves the Sturm-Liouville equation (6.36) for the boundary condition and the inhomogeneity defined at the beginning of this section. The parameter int determines linear or quadratic finite element interpolation. The matrix inversion for$\underline{\underline{\mathbf{B}}}\mathbf{Y} = \mathbf{G}$ is solved with the Thomas algorithm (explained in Chapter 7) which is a special case of Gauss elimination for the case of tridiagonal banded matrices. The subroutines Bianca and ban sol factorize and solve banded matrices (tridiagonal or pentadiagonal;. The program can be downloaded from the website. For the program the following values for a_i were chosen:

$$a_1 = 1.0,\ a_2 = -1.3,\ a_3 = 0.8,\ a_4 = -0.2,\ a_5 = 1.6$$

The RMS error is defined as

$$||y - y_{exact}|| = \left[\frac{1}{N-1}\sum_{i=1}^{N-1}\left(y_i - y_{exact,i}\right)^2\right]^{1/2}$$

Table summarizes the solution error for the Sturm-Liouville equation.

Table 6.4: RMS solution error for the Sturm Liouville equation.

Grid Δx	Linear interpolation	Quadratic interpolation
1/4	0.014	0.30
1/8	0.0039	0.0017
1/16	0.00093	0.000072
1/24	0.00040	0.000022

Notes:

- Higher order interpolation on a coarse grid is not much better or can be worse than linear interpolation.
- The solution error for linear interpolation decreases approximately with Δx^2 and for quadratic interpolation it decreases with $\sim \Delta x^3$.
- Smaller grid spacing than listed in the table requires higher machine accuracy (i. e., variables need to be defined as double precision.

```
Sturm-Liouville problem, fem:  quadratic interpolation
nx= 9 a= 0.10E+01 -0.13E+01 0.80E+00 -0.20E+00 0.16E+01
i= 2 x=0.12500 y=0.11654 yex+0.11611 dy=0.00044
i= 3 x=0.25000 y=0.21361 yex+0.21262 dy=0.00099
i= 4 x=0.37500 y=0.31541 yex+0.31575 dy=-.00034
i= 5 x=0.50000 y=0.43428 yex+0.43608 dy=-.00180
i= 6 x=0.62500 y=0.54579 yex+0.54528 dy=0.00051
i= 7 x=0.75000 y=0.64174 yex+0.63904 dy=0.00270
i= 8 x=0.87500 y=0.72503 yex+0.72543 dy=-.00040
i= 9 x=1.00000 y=0.76235 yex+0.76568 dy=-.00333
rms= 0.171E-02 nx= 9
```

Table 6.5: Parameters used in program Sturm.f

Parameter	Description
nx	number of grid points
nterm	number of terms in the inhomogeneity
x	grid for x
y, yex	numerical and exact solution
b	coefficients matrix
g	coefficients for the inhomogeneity
f	function F
a	coefficients a
fd	only used in ban sol

6.3.3 Further applications of the finite element method

Diffusion equation

Consider the diffusion equation

$$\frac{\partial T}{\partial t} - \alpha \frac{\partial^2 T}{\partial x^2} = 0$$

and linear finite element interpolation on a uniform grid. In this case the second derivative is treated as the second derivative in the Sturm-Liouville equation. The time derivative can be taken out of the integrals for the shape functions such that it is treated as the second term or the inhomogeneity in the Sturm-Liouville equation. Thus the linear finite element method leads to the equation

$$\frac{1}{6}\left.\frac{dT}{dt}\right|_{i-1} + \frac{2}{3}\left.\frac{dT}{dt}\right|_{i} + \frac{1}{6}\left.\frac{dT}{dt}\right|_{i} - \frac{\alpha}{\Delta x^2}\left(T_{i-1} - 2T_i + T_{i+1}\right) = 0$$

Here the form of the time derivative has not yet been specified. Also this form allows the freedom to evaluate the second derivative at a suitable time level. Defining $\Delta T^{n+1} = T^{n+1} - T^n$ and time derivatives as $dT/dt = \Delta T^{n+1}/\Delta t$ and using a parameter β to control the time level at which the second derivative is evaluated we obtain

$$\begin{aligned} & \frac{1}{6}\frac{\Delta T_{i-1}^{n+1}}{\Delta t} + \frac{2}{3}\frac{\Delta T_{i}^{n+1}}{\Delta t} + \frac{1}{6}\frac{\Delta T_{i+1}^{n+1}}{\Delta t} \\ -\alpha\left((1-\beta)\frac{T_{i-1}^{n} - 2T_i^n + T_{i+1}^n}{\Delta x^2} + \beta\frac{T_{i-1}^{n+1} - 2T_i^{n+1} + T_{i+1}^{n+1}}{\Delta x^2}\right) &= 0 \end{aligned} \qquad (6.39)$$

Defining operators

$$\begin{aligned} M_x &= \left(\frac{1}{6}, \frac{2}{3}, \frac{1}{6}\right) \\ L_{xx} &= \left(\frac{1}{\Delta x^2}, \frac{-2}{\Delta x^2}, \frac{1}{\Delta x^2}\right) \end{aligned}$$

we can write (6.39) in a more compact form

$$\frac{M_x \Delta T_i^{n+1}}{\Delta t} - \alpha \left[(1-\beta) L_{xx} T_i^n + \beta L_{xx} T_i^{n+1}\right] = 0 \tag{6.40}$$

Comparison with the finite difference method suggests that the main difference to the finite element method is the distribution of the time derivative over adjacent nodes. The above equation is reminiscent of the general two-level scheme introduced earlier. This scheme is recovered by using the finite difference mass operator

$$M_x^{fd} = (0,\, 1,\, 0)$$

Finally (6.40) can be cast into the equation

$$\left(M_x - \Delta t \alpha \beta L_{xx}\right) T_i^{n+1} = \left[M_x - \Delta t \alpha \left(1-\beta\right) L_{xx}\right] T_i^n \tag{6.41}$$

which is an implicit equation for T^{n+1}. Note

- for $\beta = 0$ the finite difference method generates an explicit method while the finite element method yields an implicit algorithm;
- the matrix defined by $M_x - \Delta t \alpha \beta L_{xx}$ is tridiagonal and can be solved by the Thomas algorithm (as in the case of Sturm.f);
- the symmetry of M_x and L_{xx} allows to construct an explicit scheme for $\Delta t = \Delta x^2 / \left(6\alpha\beta\right)$;
- equation (6.41) is consistent with the diffusion equation and unconditionally stable for $\beta \geq 0.5$.

Viscous flow

Stationary viscous flow through a rectangular cross section (in the x, y plane, Figure) can be described by the the z component of the momentum equation

$$\frac{\partial p}{\partial z} = \mu \left(\frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2}\right) \tag{6.42}$$

where the first term is the pressure gradient which drives the flow and w is the velocity profile of the flow. The problem is actual three-dimensional but is the pressure gradient is known or can be

prescribed one can use (6.42) to determine the flow profile in the cross section of the duct. In many problems it is an advantage to normalize the basic equation. Since $x \in [-a, a]$ and $y \in [-b, b]$ we use as normalization

$$\widetilde{x} = \frac{x}{a}, \qquad \widetilde{y} = \frac{y}{b}, \quad \widetilde{w} = \frac{w}{w_0}$$

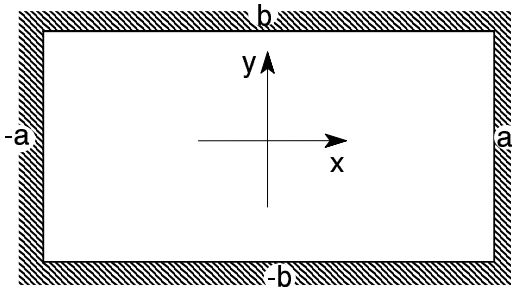
substitution in (6.42) yields

$$\frac{\partial p}{\partial z} = \frac{\mu w_0}{b^2}\left(\left(\frac{b}{a}\right)^2 \frac{\partial^2 \widetilde{w}}{\partial \widetilde{x}^2} + \frac{\partial^2 \widetilde{w}}{\partial \widetilde{y}^2}\right)$$

such that the choice $w_0 = -\mu/\left(b^2 \partial p/\partial z\right)$ yields

$$\left(\frac{b}{a}\right)^2 \frac{\partial^2 \widetilde{w}}{\partial \widetilde{x}^2} + \frac{\partial^2 \widetilde{w}}{\partial \widetilde{y}^2} + 1 = 0 \tag{6.43}$$

with the boundary conditions $\widetilde{w} = 0$ at $x = \pm 1$ and $y = \pm 1$.



Similar to the procedure for the Sturm-Liouville equation and the diffusion equation we introduce the solution

$$\widetilde{w} = \sum_{i=1}^{N} w_i \phi_i(x, y)$$

This yields the residual $R = \sum_{i=1}^{N} w_i \left[(b/a)^2 \partial^2 \phi_i/\partial x^2 + \partial^2 \phi_i/\partial y^2\right] + 1$ and using the Galerkin method generates the equation

$$\sum_{i=1}^{N} w_i \int_{-1}^{1}\int_{-1}^{1}\left(\left(\frac{b}{a}\right)^2 \frac{\partial^2 \phi_i}{\partial x^2} + \frac{\partial^2 \phi_i}{\partial y^2}\right)\phi_m dxdy = -\int_{-1}^{1}\int_{-1}^{1}\phi_m dxdy$$

or

$$\sum_{i=1}^{N} w_i \left[\int_{-1}^{1}\left(\frac{b}{a}\right)^2 \frac{\partial \phi_i}{\partial x}\phi_m\Big|_{x=-1}^{x=1} dy + \int_{-1}^{1}\frac{\partial \phi_i}{\partial y}\phi_m\Big|_{y=-1}^{y=1} dx\right]$$
$$-\sum_{i=1}^{N} w_i \int_{-1}^{1}\int_{-1}^{1}\left(\left(\frac{b}{a}\right)^2 \frac{\partial \phi_i}{\partial x}\frac{\partial \phi_m}{\partial x} + \frac{\partial \phi_i}{\partial y}\frac{\partial \phi_m}{\partial y}\right)dxdy = -\int_{-1}^{1}\int_{-1}^{1}\phi_m dxdy.$$

For the chosen boundary conditions the first two integral are zero such that the resulting equation can be written as

$$\underline{\underline{\mathbf{B}}}\mathbf{W} = \mathbf{G} \tag{6.44}$$

with

$$b_{mi} = -\int_{-1}^{1}\int_{-1}^{1}\left(\left(\frac{b}{a}\right)^2 \frac{\partial\phi_i}{\partial x}\frac{\partial\phi_m}{\partial x} + \frac{\partial\phi_i}{\partial y}\frac{\partial\phi_m}{\partial y}\right)dxdy \tag{6.45}$$

$$g_m = \int_{-1}^{1}\int_{-1}^{1}\phi_m dxdy \tag{6.46}$$

In the following we replace the single index i with a pair k, l representing the x and y coordinates. The indices k and l range from 2 to n_{x-1} and from 2 to n_{y-1} respectively because the boundary condition imply that the coefficient of the shape function at the boundaries is zero. The x integration of the first term in (6.45) yields the operator L_{xx} and the y integration of this term yield the operator M_y with

$$L_{xx}w_{k,l} = \frac{w_{k-1,l} - 2w_{k,l} + w_{k+1,l}}{\Delta x^2} \tag{6.47}$$

$$M_y w_{k,l} = \frac{1}{6}w_{k,l-1} + \frac{2}{3}w_{k,l} + \frac{1}{6}w_{k,l+1} \tag{6.48}$$

such that equation (6.44) assumes the form

$$\left[\left(\frac{b}{a}\right)^2 M_y \otimes L_{xx} + M_x \otimes L_{yy}\right] w_{k,l} = -1 \tag{6.49}$$

with

$$M_y \otimes L_{xx}w_{k,l} = \frac{1}{6}\left[\frac{w_{k-1,l-1} - 2w_{k,l-1} + w_{k+1,l-1}}{\Delta x^2}\right] + \frac{2}{3}\left[\frac{w_{k-1,l} - 2w_{k,l} + w_{k+1,l}}{\Delta x^2}\right] + \frac{1}{6}\left[\frac{w_{k-1,l+1} - 2w_{k,l+1} + w_{k+1,l+1}}{\Delta x^2}\right] \tag{6.50}$$

$$M_x \otimes L_{yy}w_{k,l} = \frac{1}{6}\left[\frac{w_{k-1,l-1} - 2w_{k-1,l} + w_{k-1,l+1}}{\Delta y^2}\right] + \frac{2}{3}\left[\frac{w_{k,l-1} - 2w_{k,l} + w_{k,l+1}}{\Delta y^2}\right] + \frac{1}{6}\left[\frac{w_{k+1,l-1} - 2w_{k+1,l} + w_{k+1,l+1}}{\Delta y^2}\right] \tag{6.51}$$

and similar for $M_x \otimes L_{yy}w_{k,l}$. In a corresponding finite difference approach we would have $M_{x,fd} = (0, 1, 0)$.

Notes:

- The operator L_{xx} and M_y are commutative. The result does not depend on the order the integration in (6.47) is executed.
- The method provides a straightforward way to employ finite element methods similar to finite difference methods.

The result provides a simple equation which can be used with SOR to solve the coefficients for the shape functions. Solving equation (6.49) for $w_{k,l}$ yields

$$\begin{aligned} w^*_{k,l} &= \frac{3}{4c_0}\left[1. + c_1\left(w_{k-1,l-1} + w_{k+1,l-1} + w_{k-1,l+1} + w_{k+1,l+1}\right)\right. \\ &\quad \left. + c_2\left(w_{k-1,l} + w_{k+1,l}\right) + c_3\left(w_{k,l-1} + w_{k,l+1}\right)\right] \end{aligned}$$

with

$$\begin{aligned} c_0 &= \left(\frac{b}{a}\right)^2 \frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} \\ c_1 &= \frac{1}{6} c_0 \\ c_2 &= \left(\frac{b}{a}\right)^2 \frac{2}{3\Delta x^2} - \frac{1}{3\Delta y^2} \\ c_3 &= -\left(\frac{b}{a}\right)^2 \frac{1}{3\Delta x^2} + \frac{2}{3\Delta y^2} \end{aligned}$$

With the usual SOR method the solution is iterated using

$$w^{n+1}_{k,l} = w^n_{k,l} + \lambda\left(w^*_{k,l} - w^n_{k,l}\right)$$

The program duct.f can be found on the course web page. To provide a reference for the approximate solution the exact solution is given by

$$\widetilde{w} = \left(\frac{8}{\pi^2}\right)^2 \sum_{i=1,3,5..}^{N} \sum_{j=1,3,5..}^{N} \left[\frac{(-1)^{(i+j)/2-1}}{ij\left[\left(\frac{ib}{a}\right)^2 + j^2\right]} \cos\left(\frac{i\pi x}{2}\right) \cos\left(\frac{j\pi y}{2}\right)\right]$$

A second important measure for this problem is the total flow rate q, i.e., the flow integrated over the cross sectional area

$$q = 2\left(\frac{8}{\pi^2}\right)^3 \sum_{i=1,3,5..}^{N} \sum_{j=1,3,5..}^{N} \left[\frac{1}{i^2 j^2\left[\left(\frac{ib}{a}\right)^2 + j^2\right]}\right]$$

Table 6.6: Input parameters for the program duct.f

Variable	Value	Description
me	2	method: 1-linear elements, 2-3pt cent dif:
nem	25	no terms in exact solution
ipr	1	ipr=0, prints solutions to the output file duct.out
niter	800	max no iterations for SOR
bar	1.0e-0	aspect ratio a/b for channel
eps	.1e-5	tolerance par for SOR
om	1.5e0	relaxation parameter (SOR), λ

The structure is similar to that of Fivol.f. The program package uses an include file ductin which declares all variables and common blocks and it make use of a parameter file duct.dat. Variables declared in the parameter file are summarized in Table 6.6.

Additional variables used in the program are listed in Table 6.7.

Table 6.7: Other variables used in the program duct.f

Variable	Description
nx, ny	Number of nodes in the x and y directions
x, y	x and y coordinates
dx, dy	grid separation in the x and y directions
flow, flowex	iterated flow rate and exact flow rate
f, fexact	iterated and exact solutions
rms	RMS error

The program generates an ASCII output file which lists basic parameter and results including the iterated solution and the exact solution and the correspond flow rates and errors. In addition the program generates a binary file which can be used as input for graphics routines. The supplied IDL program product.pro reads this this file and plots the iterated and exact solutions.

The results of this program indicate that the error decreases with Δx^2 and Δy^2. The result for using the finite element method (fem) are very similar to those of the finite difference method (fdm). The error is very similar with slightly smaller errors for fdm while the flow rate is slightly closed to the exact flow rate for fem.

Distorted computational domains - Isoparametric mapping

The finite element method is very comparable to finite differences for Cartesian grids. However the strength or the finite element method is the ease to apply it to distorted domains and complicated geometries. It is illustrated that the introduction of local coordinates is a particular strength to evaluate the integrals involving shape functions. There are various geometries to choose basic elements such as triangular, rectangular, or tetrahedral. The various method allow a simple grid refinement technique by just dividing a basic element into several new element of the same geometry.

The advantage of the finite element method is that the coordinates themselves can be described by the shape functions. Figure shows a distorted grid with rectangular elements and the mapping into a local (ξ, η) coordinate system.

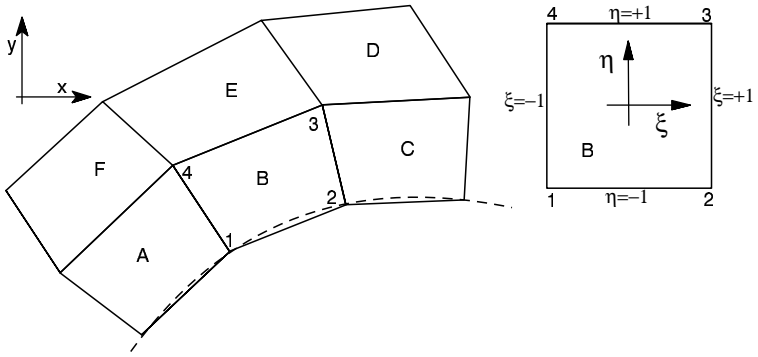


Figure 6.9: Isoparametric mapping.

The transformation between (x, y) coordinates and (ξ, η) can be defined by

$$x = \sum_{l=1}^{4} \phi_l(\xi, \eta)\, x_l \quad \text{and} \quad y = \sum_{l=1}^{4} \phi_l(\xi, \eta)\, y_l \tag{6.52}$$

with (x_l, y_l) as the x, y coordinates of the corner numbered l and $\phi_l(\xi, \eta)$ is the shape function with the value of 1 at the corner l. The transformation affects the evaluation of the weighted residual integral. Consider Laplace's equation $\Delta\Psi = 0$ as an example. The Galerkin fem produces a system of linear equations

$$\underline{\underline{\mathbf{B}}}\mathbf{W} = \mathbf{G}$$

with

$$b_{m,i} = \int_{area} \left(\frac{\partial \phi_i}{\partial x}\frac{\partial \phi_m}{\partial x} + \frac{\partial \phi_i}{\partial y}\frac{\partial \phi_m}{\partial y} \right) dxdy$$

Let us consider the first term

$$I = \int\int \frac{\partial \phi_i}{\partial x}\frac{\partial \phi_m}{\partial x} dxdy$$

The derivative $\partial/\partial x$ can be computed as

$$\frac{\partial}{\partial \xi} = \frac{\partial x}{\partial \xi}\frac{\partial}{\partial x} + \frac{\partial y}{\partial \xi}\frac{\partial}{\partial y}$$

such that $\partial\phi/\partial x$ and $\partial\phi/\partial y$ are related to $\partial\phi/\partial\xi$ and $\partial\phi/\partial\eta$ by

$$\begin{pmatrix} \partial\phi/\partial\xi \\ \partial\phi/\partial\eta \end{pmatrix} = [\mathbf{J}] \begin{pmatrix} \partial\phi/\partial x \\ \partial\phi/\partial y \end{pmatrix} \tag{6.53}$$

$$\text{with} \qquad [\mathbf{J}] = \begin{pmatrix} \partial x/\partial\xi & \partial y/\partial\xi \\ \partial x/\partial\eta & \partial y/\partial\eta \end{pmatrix} \tag{6.54}$$

where $[\mathbf{J}]$ is the Jacobian. The derivatives in the Jacobian can easily be computed from the mapping (6.52), e.g.,

$$\frac{\partial y}{\partial\xi} = \sum_{l=1}^{4} \left(\frac{\partial\phi_l}{\partial\xi}(\xi,\eta) \right) y_l$$

From (6.53) one can determine the following explicit formulation for $\partial\phi/\partial x$

$$\frac{\partial\phi_i}{\partial x} = \frac{1}{\det\mathbf{J}} \left(\frac{\partial y}{\partial\xi}\frac{\partial\phi_i}{\partial\eta} - \frac{\partial y}{\partial\eta}\frac{\partial\phi_i}{\partial\xi} \right) \tag{6.55}$$

Exercise: Derive equation (6.55).

Using $dxdy = \det\mathbf{J}\, d\xi d\eta$ one obtains

$$I = \int_{-1}^{1}\int_{-1}^{1} \frac{1}{\det\mathbf{J}} \left[\left(\frac{\partial y}{\partial\xi}\frac{\partial\phi_i}{\partial\eta} - \frac{\partial y}{\partial\eta}\frac{\partial\phi_i}{\partial\xi} \right) \left(\frac{\partial y}{\partial\xi}\frac{\partial\phi_m}{\partial\eta} - \frac{\partial y}{\partial\eta}\frac{\partial\phi_m}{\partial\xi} \right) \right] d\xi d\eta$$

All derivatives in this formulation are known because of the simple form that the ϕ_l assume. The integral can evaluated numerically or analytically.

6.4 Spectral method

The Galerkin spectral method is similar to the traditional Galerkin method but uses a suitable set of orthogonal functions such that

$$\int \phi_k\phi_l dx \begin{cases} \neq 0 & \text{for } k \neq l \\ = 0 & \text{for } k = l \end{cases}$$

Example for such functions are Fourier series, Legendre, or Chebyshev polynomials. In this sense the spectral methods can be considered global methods rather than local as in the case of finite differences or finite elements.

6.4.1 Diffusion equation

Consider as an example the diffusion equation

$$\frac{\partial T}{\partial t} = \alpha \frac{\partial^2 T}{\partial x^2}$$

for $x \in [0, 1]$ and the boundary and initial conditions

$$T(0,t) = 0, \quad T(1,t) = 1, \text{ and } \quad T(x,0) = \sin(\pi x) + x$$

Approximate solution

$$T = \sin(\pi x) + x + \sum_{j=1}^{N} a_j(t) \sin(j\pi x)$$

where the $a_j(t)$ are the unknown coefficients which need to be determined. This yields the residual

$$R = \sum_{j=1}^{N} \left[\frac{da_j}{dt} + \alpha (j\pi)^2 a_j \right] \sin(j\pi x) + \alpha\pi^2 \sin(\pi x)$$

Evaluation of the weighted integral $\int R \sin(m\pi x)\, dx$ yields

$$\sum_{j=1}^{N} \left[\frac{da_j}{dt} + \alpha (j\pi)^2 a_j \right] \int_0^1 \sin(j\pi x) \sin(m\pi x)\, dx + \alpha\pi^2 \int_0^1 \sin(\pi x) \sin(m\pi x)\, dx = 0$$

The integral yield

$$\begin{aligned} \int_0^1 \sin(j\pi x) \sin(m\pi x)\, dx &= \begin{cases} 1/2 \text{ for } j = m \\ 0 \text{ for } j \neq m \end{cases} \\ \int_0^1 \sin(\pi x) \sin(m\pi x)\, dx &= \begin{cases} 1/2 \text{ for } m = 1 \\ 0 \text{ for } m \neq 1 \end{cases} \end{aligned}$$

which yields

$$\begin{aligned} \frac{da_m}{dt} + \alpha (j\pi)^2 a_m + r_m &= 0, \quad m = 1, ..., N \\ r_m &= \begin{cases} \alpha\pi^2 \text{ for } m = 1 \\ 0 \text{ for } m \neq 1 \end{cases} \end{aligned}$$

with the solution

$$\begin{aligned} a_1 &= \mathrm{e}^{-\alpha\pi^2 t} - 1 \\ a_m &= 0 \end{aligned}$$

The solution for T is therefore

$$T = \sin(\pi x)\,\mathrm{e}^{-\alpha\pi^2 t} + x$$

This is in fact the exact solution, however, this has been obtained for this particular set of initial and boundary conditions. For a more realistic case of

$$T(x,0) = 5x - 4x^2$$

and replacing $\sin(\pi x) + x$ with $5x - 4x^2$ in the trial solution one obtains

$$r_m = \begin{cases} \frac{16\alpha}{m} \text{ for } m = 1,3,5,... \\ 0 \text{ for } m = 2,4,6,... \end{cases}$$

With forward time differencing $da_m/dt = \left(a_m^{n+1} - a_m^n\right)/\Delta t$ one obtains

$$a_m^{n+1} = a_m^n - \Delta t\left[\alpha\left(j\pi\right)^2 a_m^n + r_m\right]$$

Table 6.8: Value of the solution at $x = 0.5$ for different times

t	N=1	N=3	n=5	Exact solution
0	1.5	1.5	1.5	1.5
0.1	0.851	0.889	0.881	0.885
0.2	0.610	0.648	0.640	0.643

Table 6.8 shows the value of T for selected times and different values of N at $x = 0.5$. Note that the error is caused by two sources one of which is the limited accuracy of the representation in terms of the base functions and the other is the error in the temporal integration.

Notes:

- The spectral method achieves relatively high accuracy with relatively few unknowns.
- For Dirichlet conditions the method reduces the problem to a set of ordinary differential equations
- Treatment of nonlinear terms is not yet clear (and can be difficult)

6.4.2 Neumann boundary conditions

Let us now consider the diffusion equation again, however, for Neumann boundary conditions. Specifically the initial and boundary conditions are

$$\begin{aligned} T(x,0) &= 3 - 2x - 2x^2 + 2x^3, \\ \frac{\partial T}{\partial x}(0,t) = -2.0 \quad &\text{and} \quad T(1,t) = 1.0 \end{aligned}$$

Different from the prior example it is not attempted to incorporate the initial and boundary conditions into the approximate solution. Rather a general trial solution is used

$$T(x,t) = b_0(t) + \sum_{j=1}^{N} \left[a_j(t) \sin(j2\pi x) + b_j(t) \cos(j2\pi x) \right]$$

The boundary conditions require the following relations for the coefficients

$$\sum_{j=1}^{N} a_j 2\pi j = -2 \quad , \quad \sum_{j=0}^{N} b_j = 1$$

which can be used to eliminate a_N and b_N from the the approximate solution:

$$a_N = -\frac{1}{\pi N} - \sum_{j=1}^{N-1} a_j \frac{j}{N} \quad , \quad b_N = 1 - \sum_{j=0}^{N-1} b_j$$

This yields for the approximate solution

$$\begin{aligned} T(x,t) &= \cos(N2\pi x) - \frac{1}{\pi N} \sin(N2\pi x) + \sum_{j=1}^{N-1} a_j(t) \left[\sin(j2\pi x) - \frac{j}{N} \sin(N2\pi x) \right] \\ &\quad + \sum_{j=0}^{N-1} b_j(t) \left[\cos(j2\pi x) + \cos(N2\pi x) \right] \end{aligned}$$

This implies that the terms in square brackets now need to be considered as the base functions or

$$\begin{aligned} w_{m_a} &= \sin(m2\pi x) - \frac{m}{N} \sin(N2\pi x), \quad 1 \le m \le N-1 \\ w_{m_b} &= \cos(m2\pi x) + \cos(N2\pi x), \quad 0 \le m \le N-1 \end{aligned}$$

Substitution into the the diffusion equation yields the residual

$$\begin{aligned} R &= \sum_{j=1}^{N-1}\left\{\left[\frac{da_j}{dt}+\alpha\left(2\pi j\right)^2 a_j\right]\sin\left(2\pi jx\right)-\left[\frac{da_j}{dt}+\alpha\left(2\pi N\right)^2 a_j\right]\frac{j}{N}\sin\left(2\pi Nx\right)\right\} \\ &+\sum_{j=0}^{N-1}\left\{\left[\frac{db_j}{dt}+\alpha\left(2\pi j\right)^2 b_j\right]\cos\left(2\pi jx\right)+\left[\frac{db_j}{dt}+\alpha\left(2\pi N\right)^2 b_j\right]\cos\left(2\pi Nx\right)\right\} \\ &+\alpha\left(2\pi N\right)^2\cos\left(2\pi Nx\right)-\alpha 4\pi N\sin\left(2\pi Nx\right) \end{aligned}$$

Evaluating the weighted residuals yields

$$\frac{da_m}{dt}+\alpha\left(2\pi m\right)^2 a_m+\frac{m}{N}\sum_{j=1}^{N-1}\frac{j}{N}\left[\frac{da_j}{dt}+\alpha\left(2\pi N\right)^2 a_j\right] = -\alpha 4\pi N \tag{6.56}$$

$$\frac{db_m}{dt}+\alpha\left(2\pi m\right)^2 b_m+\sum_{j=0}^{N-1}\left[\frac{db_j}{dt}+\alpha\left(2\pi N\right)^2 b_j\right] = \alpha\left(2\pi N\right)^2 \tag{6.57}$$

$$2\frac{db_0}{dt}+\sum_{j=0}^{N-1}\left[\frac{db_j}{dt}+\alpha\left(2\pi N\right)^2 b_j\right] = \alpha\left(2\pi N\right)^2 \tag{6.58}$$

Equations (6.56) for a_m are linearly independent from equations (6.57) and can be solved independently. To solve for the time integration it is necessary, however, to factorize the the system and carry out a matrix multiplication for each time step. Start values for a_m and b_m are easily obtained using the trial solution for the initial condition.

Note, since the problem is linear the factorization is only needed once because the subsequent time steps always use the same matrix (coefficients in equations (6.56) to (6.58) are constant).

6.4.3 Pseudospectral method

Main disadvantage of spectral method is large computational effort particularly for nonlinear problems. An alternative to Galerkin method which basically solve the diffusion equation in spectral space is to use a mixture of spectral and real space. An frequently used method in this category is the pseudospectral approach which uses the collocation method. The example is again the diffusion equation

$$\frac{\partial u}{\partial t}-\alpha\frac{\partial^2 u}{\partial x^2}=0$$

subject to the boundary and initial conditions

$$u\left(-1,t\right)=-1,\quad u\left(1,t\right)=1,\quad u\left(x,0\right)=\sin\pi x+x$$

When compared to the spectral method, the pseudospectral method does not determine the solution entirely in spectral space. Recall that the solution for the Galerkin spectral method is found by integrating the spectral coefficients. For a linear problem this is feasible but a nonlinear problem requires the inversion of a large matrix for each time step.

The pseudo spectral method often uses an expansion

$$u(x,t) = \sum_{k=1}^{N+1} a_k(t)\, T_{k-1}(x)$$

consists of three basic steps:

- Given a solution u_j^n one determines the spectral coefficients a_k. This transforms the problem from physical to spectral space. Using Chebichev Polynomials the step can be done with a fast Fourier Transform (FFT) if the collocation points are $x_j = \cos\left(\frac{\pi(j-1)}{N}\right)$ (time efficient with number of operations proportional to $N \log N$).
- The next step is to evaluate the second derivative $\partial^2 u/\partial x^2$ from the coefficients a_k which makes use of recurrence relations and is also very efficient:

$$\begin{aligned} \frac{\partial u}{\partial x} &= \sum_{k=1}^{N+1} a_k^{(1)} T_{k-1}(x) \\ \frac{\partial^2 u}{\partial x^2} &= \sum_{k=1}^{N+1} a_k^{(2)} T_{k-1}(x) \end{aligned}$$

- The final step then is to integrate in time. Different from the Galerkin spectral method this is done in real space directly for u: $u_j^{n+1} = u_j^n + \alpha \Delta t \left.\frac{\partial^2 u}{\partial x^2}\right|_j^n$ thereby avoiding to have to solve a large system of equations for the time derivatives of da_k/dt.

6.5 Summary on different weighted residual methods:

- local approximations: finite difference, finite volume, finite element
  - irregular domains
  - variable grid resolution
  - boundary conditions relatively simple
- global base function: spectral methods
  - high accuracy

- arbitrary often differentiable
- Galerkin method computationally expensive for nonlinear problems
- Pseudospectral method employs spectral space only to determine spatial derivatives (much more efficient)