Numerical Solutions for Partial Differential Equations

S. DELAHAIES

April 3, 2012

6 Weighted residuals – Galerkin

6.1 Weighted residuals method

To present concepts and methods of weighted residuals we restrict ourselves to a closed interval $[a, b] \in \mathbb{R}$, most of what will be presented in this section extends to higher dimensions. Let us consider the following problem: find u such that

$$\mathcal{L}(u(x)) = f(x), \quad x \in (a, b), \quad u(a) = \alpha, \quad u(b) = \beta, \tag{1}$$

where $\mathcal{L}$ is a differential operator and f is a given smooth function on $[a, b]$. Let V be the vector space of smooth functions from $[a, b] \in \mathbb{R}$ to $\mathbb{R}$, such that

$$v(a) = \alpha \ , \ v(b) = \beta.$$

We equiped V with the inner product $(\ , \ )$ given by

$$(u, v) = \int_a^b u(x)v(x)\,\mathrm{d}x, \tag{2}$$

for all $u, v \in V$. The weak formulation of (1) consists in finding $u \in V$ such that

$$(\mathcal{L}(u) - f, v) = 0, \tag{3}$$

for all $v \in V$. Let $\{\phi_j\}_{j\in\mathbb{N}}$ be a basis for V, then there exist a unique set $\{c_j\}_{j\in\mathbb{N}}$, $c_j \in \mathbb{R}$, $\forall j$, such that u can be expressed as

$$u(x) = \sum_{j\in\mathbb{N}} c_j \phi_j(x). \tag{4}$$

Note that V is a infinite dimensional vector space but computers can only deal with finite dimensional problems then we consider a finite set of basis functions $\{\phi_i\}_{i=0,\dots,N}$ (note that $\{\phi_i\}_{i=0,\dots,N}$ is NOT a basis of V), and we seek for an approximate weak solution U in $V_h = \mathrm{span}(\{\phi_i\})$ of the form

$$U(x) = \sum_{i=0}^{N} c_i \phi_i(x). \tag{5}$$

Inserting $U(x)$ in equation (1) gives a residual $R(x)$ given by

$$R(x) = \mathcal{L}\left(\sum_{i=0}^{N} c_i\phi_i(x)\right) - f(x).$$

The weighted residuals method consists in requiring that the inner product of the residual R by any weight function v to be zero, that is

$$(R, v) = 0, \tag{6}$$

for all $v \in V$. Choosing $N+1$ linearly independent weight functions $v_j, j = 0..N$ we obtain a system of N+1 equations

$$(R, v_j) = 0, \quad j = 0..N, \tag{7}$$

for the $N+1$ unknowns c_i, $i = 0..N$. The larger N is the more accurate the approximate solution is hoped to be. The various weighted residuals methods (eg subdomain, collocation, least square, Galerkin, moments, etc) differ from one another by the choice of the weighting functions v_i.

6.2 Galerkin method

The Galerkin method is probably the most used approximate finite element method for engeniering problems. The weighting functions are chosen to be the basis function, that is

$$v_i = \phi_i, \quad i = 0, \ldots, N.$$

Since the ϕ_i are extracted from a basis of the infinite vector space V, as $N \to \infty$, the approximate solution is capable of representing the exact solution.

We illustrate the Galerkin method with the following problem: find an approximate solution U to the second order elliptic equation

$$\frac{\partial^2 u}{\partial x^2} = -f(x), \quad x \in (0,1) \tag{8}$$

with boundary condition $u(0) = u(1) = 0$, where f is a given function. Let V denote the vector space of smooth function from $[0,1]$ to $\mathbb{R}$ such that for all $v \in V$, $v(0) = v(1) = 0$. The weak formulation of the above problem is given by

$$(\mathcal{L}(u) + f, v) = 0, \quad \forall v \in V,$$

that is

$$\int_0^1 \frac{\partial^2 u}{\partial x^2} v \,\mathrm{d}x + \int_0^1 f v \,\mathrm{d}x = 0, \quad \forall v \in V.$$

Integrating by parts the first integral in the expression above gives

$$\left[\frac{\partial u}{\partial x} v\right]_0^1 - \int_0^1 \frac{\partial u}{\partial x}\frac{\partial v}{\partial x} \,\mathrm{d}x + \int_0^1 f v \,\mathrm{d}x = 0, \quad \forall v \in V.$$

Since $v \in V$, we have $v(1) = v(0) = 0$ then the first term vanishes and we obtain the weak formulation

$$-\int_0^1 \frac{\partial u}{\partial x}\frac{\partial v}{\partial x} \,\mathrm{d}x + \int_0^1 f v \,\mathrm{d}x = 0, \quad \forall v \in V, \tag{9}$$

that is

$$(u', v') = (f, v), \quad \forall v \in V.$$

We now apply Galerkin method: we choose a basis for V, $\{\phi_j\}_{j\in\mathbb{N}}$, from which we extract a finite number $N+1$ of basis functions, $\{\phi_i\}_{i=0..N}$, and we seek for a set of coefficients $\{c_i\}_{i=0..N}$ such that an approximate solution U is given by

$$U(x) = \sum_{i=0}^{N} c_i \phi_i(x),$$

and the c_is are obtained by solving the $N+1$ equations

$$\left(\left(\sum_{i=0}^{N} c_i \phi_i(x)\right)', \phi_j'(x)\right) = (f, \phi_j(x)), \quad j = 0, \ldots, N.$$

By linearity the above expression can be written as

$$\sum_{i=0}^{N} c_i \left(\phi_i(x)', \phi_j'(x)\right) = (f, \phi_j(x)), \quad j = 1, \ldots, N.$$

These equations can be written as a single matrix equation

$$\mathbf{K}\mathbf{c} = \mathbf{b}, \tag{10}$$

where $\mathbf{K}$ is the stiffness matrix given by

$$\mathbf{K} = \begin{pmatrix} (\phi_0'(x), \phi_0'(x)) & \ldots & (\phi_0'(x), \phi_N'(x)) \\ \vdots & \ddots & \vdots \\ (\phi_N'(x), \phi_0'(x)) & \ldots & (\phi_N'(x), \phi_N'(x)) \end{pmatrix}, \tag{11}$$

$\mathbf{c} = (c_i)_{i=0,\ldots,N}$ is the (column) vector of coefficients and $\mathbf{b}$ is the load matrix given by

$$\mathbf{b} = \begin{pmatrix} (f(x), \phi_0(x)) \\ \vdots \\ (f(x), \phi_N(x)) \end{pmatrix}. \tag{12}$$

Note that since the inner product is symetric then the stiffness matrix $\mathbf{K}$ is symetric. Form equation (10) we see that in the case we are studying here (linear, ...), finding an approximate solution using the Galerkin method amounts to solving a linear system.

We now have to choose the basis functions. In finite elements methods we use local functions, that is functions with compact support. As an example we consider *linear hat functions*. We divide our domain $[0, 1]$ into N regular intervals $[x_i; x_{i+1}]$, $i = 0, \ldots, N$, and we define the basis functions ϕ_i as follows

$$\phi_i(x) = \begin{cases} \dfrac{x - x_{i-1}}{x_i - x_{i-1}}, & \text{if } x_{i-1} \leq x \leq x_i, \\ \dfrac{x_{i+1} - x}{x_{i+1} - x_i}, & \text{if } x_i \leq x \leq x_{i+1}, \\ 0, \quad \text{if } x \leq x_{i-1} \text{ or } x_{i+1} \leq x, \end{cases} \tag{13}$$

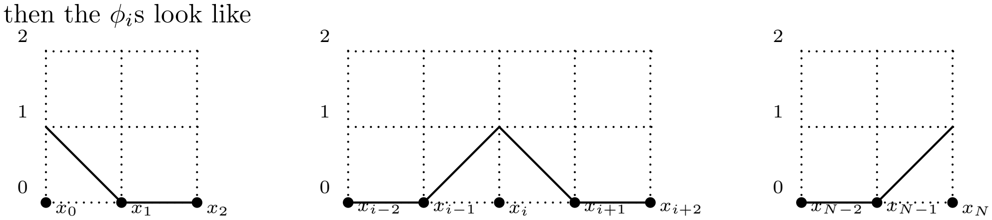
for $i = 1, \dots, N-1$. If we have zero boundary conditions (ie $u(0) = u(1) = 0$) then we do not need ϕ_0 and ϕ_N, otherwise we choose

$$\phi_0(x) = \begin{cases} \dfrac{x_1 - x}{x_1 - x_0}, & \text{if } x_0 \le x \le x_1, \\ 0, & \text{if } x_1 \le x, \end{cases} \tag{14}$$

and

$$\phi_N(x) = \begin{cases} \dfrac{x - x_{N-1}}{x_N - x_{N-1}}, & \text{if } x_{N-1} \le x \le x_N, \\ 0, & \text{if } x \le x_{N-1}, \end{cases} \tag{15}$$

then the ϕ_is look like



ϕ_0 ϕ_i ϕ_N

We see that the basis functions satisfy the property

$$\phi_i(x_j) = \delta_{ij},$$

where δ denotes the Kronecker symbol, therefore using the boundary conditions we have

$$U(0) = \sum_{j=0..N} c_j \phi_j(0) = c_j \phi_j(0) = c_0 = 0$$

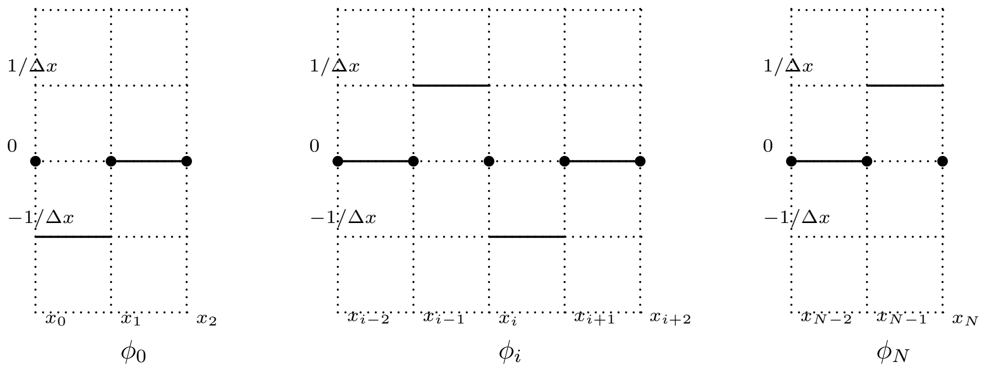
and similarly $U(N) = c_N = 0$. Thus we can consider a reduced problem (10) with i, j running from 1 to $N-1$. We need to work out the inner products (ϕ_i', ϕ_j'). Since the grid is regular we have $x_i - x_{i-1} = \Delta x$, and we can rewrite the ϕ_is as

$$\phi_i(x) = \begin{cases} \dfrac{x - x_{i-1}}{\Delta x}, & \text{if } x_{i-1} \le x \le x_i, \\ \dfrac{x_{i+1} - x}{\Delta x}, & \text{if } x_i \le x \le x_{i+1}, \\ 0, & \text{if } x \le x_{i-1} \text{ or } x_{i+1} \le x, \end{cases}$$

and we have

$$\phi_i'(x) = \begin{cases} 1/\Delta x, & \text{if } x_{i-1} \le x \le x_i, \\ -1/\Delta x, & \text{if } x_i \le x \le x_{i+1}, \\ 0, & \text{if } x \le x_{i-1} \text{ or } x_{i+1} \le x, \end{cases}$$

that is



Then we see that (ϕ_i', ϕ_j') unless $j = i-1, i, i+1$. We consider (ϕ_i', ϕ_{i-1}'), we have

$$\begin{aligned}(\phi_i', \phi_{i-1}') &= \int_{S(\phi_i'(x)) \cap S(\phi_{i-1}'(x))} \phi_i'(x)\phi_{i-1}'(x)\,\mathrm{d}x \\ &= \int_{x_{i-1}}^{x_i} \frac{1}{\Delta x}\frac{-1}{\Delta x}\,\mathrm{d}x \\ &= \left[-\frac{x}{\Delta x^{\,2}}\right]_{x_{i-1}}^{x_i} \\ &= \frac{-1}{\Delta x}\end{aligned}$$

moreover $(\phi_i', \phi_{i-1}') = (\phi_i', \phi_{i+1}') = -1/\Delta x$. Finally

$$\begin{aligned}(\phi_i', \phi_i') &= \int_{x_{i-1}}^{x_i} \left(\frac{1}{\Delta x}\right)^2 \mathrm{d}x + \int_{x_i}^{x_{i+1}} \left(\frac{-1}{\Delta x}\right)^2 \mathrm{d}x \\ &= \left[\frac{x}{\Delta x^{\,2}}\right]_{x_{i-1}}^{x_i} + \left[\frac{x}{\Delta x^{\,2}}\right]_{x_i}^{x_{i+1}} \\ &= \frac{2}{\Delta x}\end{aligned}$$

then we see that the stiffness matrix reduces to

$$\mathbf{K} = \frac{1}{\Delta x}\begin{pmatrix} 2 & -1 & 0 & \dots & 0 \\ -1 & 2 & -1 & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & -1 & 2 & -1 \\ 0 & \dots & 0 & -1 & 2 \end{pmatrix}. \tag{16}$$

We now compute the load matrix $\mathbf{b}$ with the following example, finding a weak solution of

$$\frac{\partial^2 u}{\partial x^2} = -2, \quad x \in (0,1) \tag{17}$$

with boundary condition $u(0) = u(1) = 0$. By definition the load matrix is given by

$$\mathbf{b} = \begin{pmatrix} (2, \phi_1(x)) \\ \vdots \\ (2, \phi_{N-1}(x)) \end{pmatrix}. \tag{18}$$

For $i = 1, \ldots, N-1$ we have

$$
\begin{aligned}
(2, \phi_i) &= \int_{x_{i-1}}^{x_i} 2\frac{x - x_{i-1}}{\Delta x}\,\mathrm{d}x + \int_{x_i}^{x_{i+1}} 2\frac{x_{i+1} - x}{\Delta x}\,\mathrm{d}x \\
&= \left[\frac{(x - x_{i-1})^2}{\Delta x}\right]_{x_{i-1}}^{x_i} + \left[-\frac{(x_{i+1} - x)^2}{\Delta x}\right]_{x_i}^{x_{i+1}} \\
&= 2\Delta x
\end{aligned}
$$

therefore $\mathbf{b} = 2h(1\ \ldots\ 1)^T$.

To illustrate these results we consider problem (17) with boundary conditions $u(0) = u(1) = 0$, we choose $N = 4$. As we said previously, finding an approximate solution in the weak sense amounts to solving the matrix equation $\mathbf{Kc} = \mathbf{b}$ where $\mathbf{K}$ and $\mathbf{b}$ are given by equation (16) and (18) respectively, then we have

$$
4\begin{pmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{pmatrix}\begin{pmatrix} c_1 \\ c_2 \\ c_3 \end{pmatrix} = \frac{1}{2}\begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}. \tag{19}
$$

Solving this equation gives

$$
\begin{aligned}
c_1 &= 3/16, \\
c_2 &= 1/4, \\
c_3 &= 3/16
\end{aligned}
$$

therefore

$$
U(x) = \frac{3}{16}\phi_1(x) + \frac{1}{4}\phi_2(x) + \frac{3}{16}\phi_3(x) \tag{20}
$$