FINITE ELEMENT METHOD PROJECT

FINITE ELEMENT METHOD FOR THE ORDINARY SECOND ORDER DIFFERENTIAL EQUATION, DIRICHLET BOUNDARY CONDITION, PARABOLIC ELEMENTS

AUTHOR:
JESUS DAVID NUÑEZ CAMPO

PROFESSOR:
IWONA WRÓBEL

CONTENTS INDEX

1. PRELIMINARIES

Prior to the development of the concepts associated to the finite element method, it is worth to introduce the following definitions and notations:

Definition 1.1 - Functional: $\phi : V \to \Re$. Where, for the case which concern us:

V - A Hilbert Space. A Euclidean space with a norm $\|.\|_v$ associated.
V^* . A dual space, a space of all linear and continuous functionals on V

Definition 1.2 - Linear operator: An operator T is linear if $\forall\, x, y \in V$ and $\forall\, \alpha, \beta \in \Re$:

$$T(\alpha x + \beta y) = \alpha T(x) + \beta T(y) \quad .$$

Definition 1.3 – Space of twice-differentiable functions $\Theta = C^2(a,b)$ functions whose second derivative is continuous.

Definition 1.4 - Space of square-integrable functions $\Phi = L^2(a,b)$, A Hilbert Space unitary and complete (*i.e.* every sequence converge)

Definition 1.5 - L^2 Inner product**:** $(v,u) := \int_a^b v\,u\,du$ and $\forall\, v,u : v,u \in L^2(a,b)$ are said to be square integrable on the interval (a,b)

***Definition 1.6 – Norm*:** $\|u\| = \sqrt{(u,u)} = \left(\int_a^b u^2(x)\,dx \right)^{1/2}$

2. FINITE ELEMET METHOD DEFINITION

A Finite Element is a Tuple ($K, P(K), \Sigma(K)$) with the following properties:

1. $K \subseteq \Re^n$ is bounded, closed and non-empty and has a piecewise smooth boundary
2. $P(K)$ is a final dimensional space of functions on K (*Space of shape functions*)
3. $\Sigma(K) = \{\phi_1, \phi_2, \ldots \phi_M\}, M = dim(P(K))$ is a set of real-valued linear functionals on $P(K)$, which are independent (*The set of nodal variables*)

3. THE SPACE OF FINITE ELEMENTS

Let $\Omega \subset \Re^n$ be open and bounded. $T(\Omega)$, cover of $\bar{\Omega}$ consisting of closed sets of non-empty and disjoint interiors.

$$\bar{\Omega} = \bigcup_{K \in T(\Omega)} K$$

$T(\Omega) = \{K_1, K_2, \ldots, K_s\}$ for every $K_i \subset T(\Omega)$, $i = 1, \ldots, s$ we choose $P(K_i)$ and $\Sigma(K_i)$, in such a way that $(K_i, P(K_i), \Sigma(K_i))$ is a finite element.

All functionals in $\Sigma(K_i)$ are linear combinations of values of a function and/or its derivatives up to certain order $Q (Q \geqslant 0)$ at certain points.

> ***Definition 4.1 – The Space of Finite-Elements*** $V(\Omega)$ **:** The space of function $v : \bar{\Omega} \to \Re$

Conditions:

1. $v|_{K_j} \in P(K_j) \quad \forall j = 1, \ldots, N$
2. $\phi \in \sum(K_i) \cap \sum(K_j) = \phi(v|_{K_j}) = \phi(v|_{K_j}), \quad \forall i, j = 1, \ldots, N$

The latter condition guaranties certain regularity of functions in $V(\Omega)$, *i.e* continuity and continuity of their derivatives.

3.1 BASIS OF $V(\Omega)$

$$\sum(\Omega)=(\psi_{1,}\psi_{2,},\ldots,\psi_{L})$$

$\sum_i$, $i=1,\ldots,L$. The set of all functionals from $\sum(\Omega), K\in T(\Omega)$ which can be identified with ψ_i.

$$T_i=\left\{K\in T(\Omega);\exists\phi\in\sum(\Omega)\cap\sum_i\right\} \quad (3.1.1)$$

Let $K\in T_i\wedge\phi\in\sum(\Omega)\cap\sum_i$. Define $P_{K,i}\in P(K)$, such that:

1. $\phi(P_{K,i})=1$
2. $\tilde{\phi}(P_{K,i})=0 \quad \forall\tilde{\phi}\neq\phi,\tilde{\phi}\in\sum(K)$

Funtions $P_{K,i}$ form a basis for $P(K)$. Let us define functions v_i, $i=1,\ldots,L$ on $\bar{\Omega}$

$$v_i(x)=\begin{cases}P_{K,i} & \forall x\in K\in T_i\\ 0 & \forall x\in K\notin T_i\end{cases}$$

Theorem: The functions $v_1,\ldots,v_L$ form a basis for $V(\Omega)$.

Fact: The support of every v_i (*i.e* the set of points for which v_i is non-zero) is contained in T_i, yielding to sparsity of matrices, fact that will become clearer as we develop the topic.

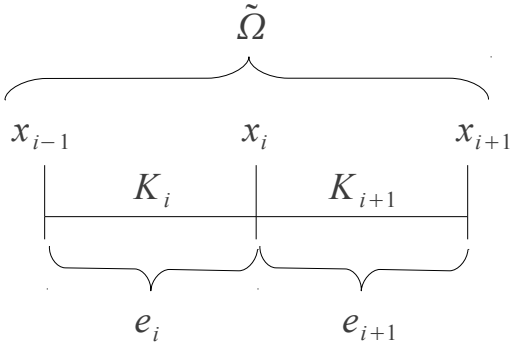


Figure 4.1: a cover of $\tilde{\Omega}\subset\Omega$ *in a 1-Dimensional grid*

e_i *and* e_{i+1} *are finite elements*

For the sake of exemplify, let us consider the figure (4.1):

$$\sum(K_i)=\{v(x_{i-1}),v(x_i)\} \quad , \quad \sum(K_{i+1})=\{v(x_i),v(x_{i+1})\}$$

$$T_1=\{K_{i-1}\} \quad T_2=\{K_i\}$$
$$T_3=\{K_{i+1}\}$$

4. SHAPE FUNCTIONS AND FUNCTIONALS

In this work we will focus on parabolic elements. The space associated to this elements has the following characteristics:

1. $P(K)=P_2(K)$ The set of all polynomials of degree $\leqslant 2$
2. $dim(P(K))=3$
3. There is a basis (p_1, p_2, p_3) of $P(K)$, such that :

$$\phi_i(P_j)=\delta_{ij}=\begin{cases}1 & i=j\\ 0 & i\neq j\end{cases} \qquad (3.1)$$

Such a system is called a *Nodal basis of* $P(K)$. Choosing bases with small support leads to a sparse, well-conditioned linear algebraic system for the solution.

The basis is created form the parabolic elements:

$$P_1^j(x)=1+3\left(\frac{x-x_j}{h_j}\right)+2\left(\frac{x-x_j}{h_j}\right)^2$$
$$P_2^{j-1}(x)=1-3\left(\frac{x-x_{j-1}}{h_j}\right)+2\left(\frac{x-x_{j-1}}{h_j}\right)^2$$
$$P_3^{j-1/2}(x)=1+4\frac{(x-x_{j-1/2})}{h_j}$$

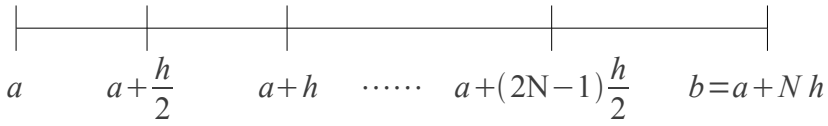
where $h_j=x_j-x_{j-1}$. The basis is then:

$$v_j(x)=\begin{cases} P_1^j & x\in K_j \\ P_2^{j-1} & x\in K_{j-1} \\ 0 & x\notin K_j\cup K_{j-1} \end{cases} \qquad (4.1)$$

$$v_{j-1/2}=\begin{cases} P_3^{j-1/2} & x\in K_j \\ 0 & x\notin K_j \end{cases}$$

where $K_j=[x_{j-1},x_j],\quad j=1,\ldots,N$. For the domain $\Omega=(a,b)$:

$\sum(\Omega)=\{\psi_0,\psi_1,\ldots,\psi_{2N}\}$:



$$h=\frac{b-a}{N}$$

$$\psi_i(v_{1/2})=v_{j/2}(a+hj/2)\Big|_{j=0}^{2N} \qquad \forall\, v_{1/2}\in P_2(K) \quad (4.2)$$

For the chosen functionals and for a single element, we have:

$\sum(K_j)=\{\phi_1^j,\phi_2^j,\phi_3^j\}$:

$$\phi_1^j(v_j)=v_j(x_{j-1}) \quad \phi_2^j(v_j)=v_j(x_j) \quad \phi_3^j(v_{j-1/2})=v_{j-1/2}(x_{j-1/2})$$

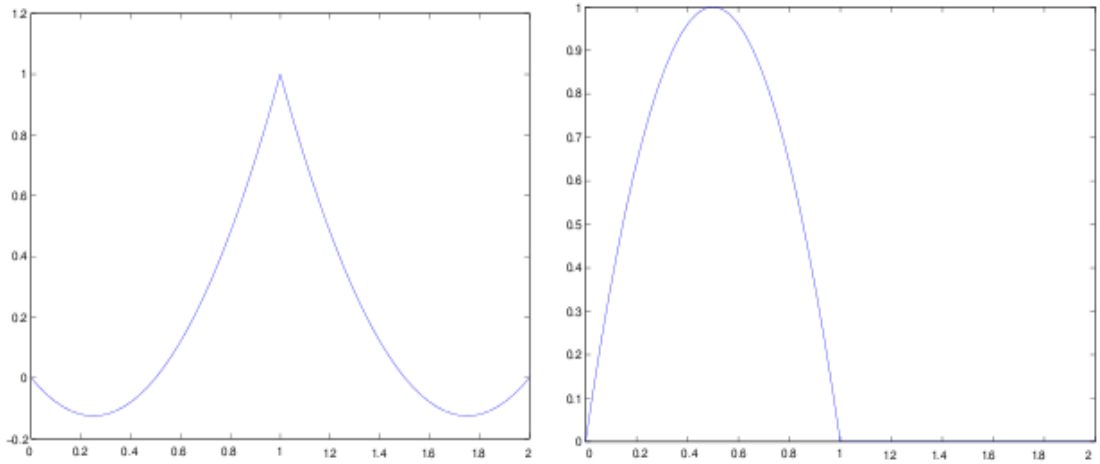


Figure 4.1: Piecewise-parabolic basis functions for a node at x = 1 (left) and an element midpoint at x = 0.5 (right)

5. MEAN-SQUARE APROXIMATION

Statement of the problem:

Let $\Phi_L \subset \Phi$ and $dim(\Phi_L)=L$. Let Φ_L be a space of finite elements defined on $\Omega=(a,b)$ and suppose we have a bases $(v_1, v_2, \ldots, v_L)$ of $V(\Omega)=V_L$. The problem of the approximation is to find:

$$f^* \in V_L \text{ , such that}$$
$$\|f^* - f\| = \min_{\tilde{f} \in V_L} \|\tilde{f} - f\|$$

This problem has unique solution f^* and it satisfies:

$$(f^* - f, v)=0, \quad \forall v \in V_L$$

with the inner product as defined in the definition *1.4*. If is equivalent to:

$$(f^* - f, v_j)=0, \quad \forall v \in V_L \quad j=(1,\ldots,L)$$

An rewriting f^* as a linear combination of the basis of V_L:

$$f^* = \sum_{i=1}^{L} \alpha_i v_i$$
$$\left(\sum_{i=1}^{L} \alpha_i v_i - f, v_j \right) = 0, \quad j=(1,\ldots,L)$$

$$\sum_{i=1}^{L} \alpha_i (v_i, v_j) = (f, v_j), \quad j=(1,\ldots,L) \qquad (4.2)$$

The equation (4.2) is called the *set of normal equations*.

$$(v_i, v_j) = \int_a^b v_i(x) v_j(x)\, dx$$

$$(f, v_j) = \int_a^b f(x) v_j(x)\, dx$$

$G\alpha = F$ *where:*

$$G = \begin{bmatrix} (v_1, v_1) & \cdots & (v_1, v_L) \\ \vdots & \ddots & \vdots \\ (v_L, v_L) & \cdots & (v_L, v_L) \end{bmatrix}$$

$$F = \begin{bmatrix} (v_1, v_L) \\ \vdots \\ (v_L, f) \end{bmatrix}$$

G is symmetric, positive-definite:

$$X^T G X < 0 \forall\ x \in \Re^L$$

G is sparse (many products $(v_i, v_j) = 0$) and banded.

- *Linear Elements – G is tridiagonal*
- *Parabolic Elements – G is tridiagonal*
- *Cubic Elements – G is pentadiagonal*

6. FINITE ELEMENT FOR A SECOND ORDER DIFFERENTIAL EQUATION. DIRICHLET BOUNDARY CONDITIONS.

The problem is to find u for the following second order ODE, under the given boundary conditions.

$$\Gamma[u] := -\frac{d}{dx}\left(p(x)\frac{du}{dx}\right) + q(x)u = f(x), \quad u = [a,b] \tag{5.1a}$$

$$u(a) = u_a, u(b) = u_b \tag{5.1b}$$

with $p(x), q(x), f(x)$ being smooth functions on $[a,b]$. Problems like (5.1) arise in many situations including the longitudinal deformation of an elastic rod, steady heat conduction, and the transverse section of a supported cable. In the latter case, for example, represents the lateral section at position x of a cable having (scaled) unit length that is subjected to a tensile force p, loaded by x a transverse force per unit length $f(x)$ and supported by a series of springs with elastic modulus q (Figure 5.1). The situation resembles the cable of a suspension bridge. The tensile force p is independent of x for the assumed small deformations of this model, but the applied loading and spring moduli could vary with position.

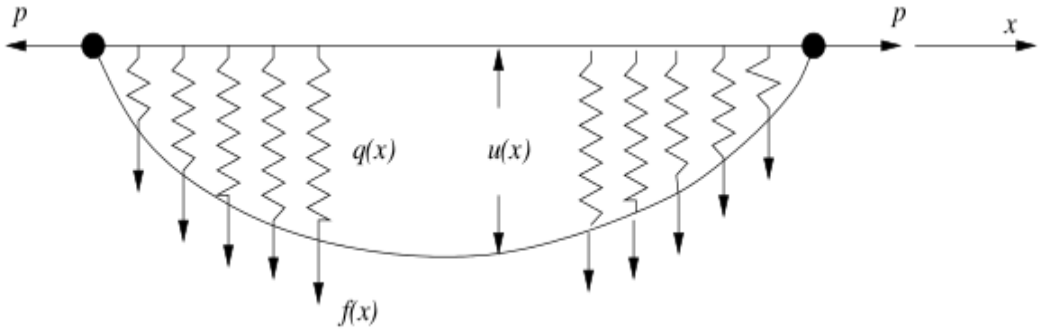


Figure 6.1 Deflection u of a cable under tension p, loaded by a force f per unit length, and supported by springs having elastic modulus q.

6.1 THE METHOD OF WEIGHTED RESIDUALS (MWR)

With finite difference techniques, derivatives in (5.1) are approximated by finite differences with respect to a mesh. With the finite element method, the method of weighted residuals (MWR) is used to construct an integral formulation of (5.1) called

a variational problem. To this end, let us multiply (5.1) by a test or weight function v and integrate over (a,b) to obtain:

$$(v, \boldsymbol{\Gamma}[u]-f)=0, \quad \forall v \in \Phi \quad (5.2)$$

The solution of (5.1a) is also a solution of (5.2) for all functions v for which the inner product exists. We'll express this requirement by writing $v \in \Phi$. All functions of class Φ are "square integrable" on (a,b); thus, (v,v) exists.

Using the method of weighted residuals, we construct approximate solutions by replacing u and v by simpler functions U and V and solving (5.2) relative to these choices. Specifically, we'll consider approximations of the form :

$$u(x) \approx U(x)=\sum_{j=1}^{N} c_j \xi_j(x)$$
$$v(x) \approx W(x)=\sum_{j=1}^{N} d_j \zeta_j(x)$$

The functions $\xi_j(x)$ and $\zeta_i(x)$, $j=1,2,\ldots,L$, are preselected and our goal is to determine the coeficients c_i, such that U is a good approximation of u. For example, we might select :

$$\xi_j(x)=\zeta_j(x)=\sin(j\pi x)$$

To obtain approximations in the form of discrete Fourier series. The approximation U is called a trial function and, W is called a test function. Since the diferential operator $\boldsymbol{\Gamma}[u]$ is second order, we might expect $U \in \Theta$. (Actually, u can be slightly less smooth, but C^2 will suffice for the present discussion.) Thus, it is natural to expect U to also be an element of Θ. Mathematically, we regard U as belonging to a finite-dimensional function space that is a subspace of Θ.

We express this condition by writing $U \in S^L(a,b) \subset \Theta$ (The restriction of these functions to the interval $a<x<b$ will, henceforth, be understood and we will no longer write the (a,b)). With this interpretation, we will call S^L the trial space and regard the preselected functions $\xi_j(x)$, $j=1,2,\ldots,L$, as forming a basis for S^L.

Likewise, since $v \in \Phi$, we will regard W as belonging to another finite-dimensional function space $\hat{S}^L$ called the test space. Thus, $W \in \hat{S}^L \subset \Phi$ and $\zeta_i(x)$ $j = 1, 2, \ldots, L$, provide a basis for $\hat{S}^L$.

Now, replacing v and u in (5.2) by their approximations W and U , we have :

$$(W, \boldsymbol{\Gamma}[U] - f) = 0 \quad , \quad \forall\, W \in \hat{S}^L \qquad (5.3a)$$

The residual :

$$r(x) := \boldsymbol{\Gamma}[U] - f(x) \qquad (5.3b)$$

is apparent and clarifies the name "method of weighted residuals." The vanishing of the inner product (5.3a) implies that the residual is orthogonal in L^2 to all functions $W \in \hat{S}^L$.

6.2 VARIATIONAL FORMULATION (GALERKIN'S METHOD)

There are many reasons to prefer a more symmetric variational form of (5.1a) than (5.2), e.g., problem (5.1a) is symmetric (self-adjoint) and the variational form should reflect this. Additionally, we might want to choose the same trial and test spaces, but ask for less continuity on the trial space S^L. This is typically the case. We can construct the symmetric variational form that we need by integrating the second derivative terms in (5.2) by parts thus, using (5.1a) :

$$\int_a^b v[-(pu')' + qu - f]\,dx = \int_a^b (v' pu' + vqu - vf)\,dx - vpu'\Big|_a^b \;=\; 0 \quad (6.2.1)$$

where $()' = d()/dx$. The treatment of the last *(boundary)* term will need greater attention. But for what to this work concerns, will consider v satisfying the same trivial boundary conditions (5.1b) as u . In this case, the boundary term vanishes and (6.2.1) becomes

$$\boldsymbol{a}(v, u) - (v, f) = 0 \quad (6.2.2a)$$

where:

$a : VxV \to \Re$. A bilinear form (i.e linear with respect to every coordinate).

$$a(v,u) = \int_a^b (v' pu' + vqu)\, dx \quad (6.2.2b)$$

The bilinear form $a(v,u)$ is called the strain energy. In mechanical systems It frequently corresponds to the stored or internal energy in the system.

Definition 6.2.1: A form a is continuous if there exist a constant

$$\exists M > 0 | a(v,u) | \geq M \|u\|_v \|v\|_v, \quad \forall u, v \in V$$

Definition 6.2.2 – A form is V-Elliptic (or coercive) if

$$\exists \alpha > 0 | a(u,u) | \geq \alpha \|u\|_v^{2,} \quad \forall u \in V$$

Definition 6.2.2 – A form is symmetric if

$$a(v,u) = a(u,v), \quad \forall u, v \in V$$

The integration by parts has eliminated second derivative terms from the formulation. Thus, solutions of (6.2.2a) might have less continuity than those satisfying either (5.1a) or (5.2). For this reason, they are called weak solutions in contrast to the strong solutions of (5.1a) or (5.2). Weak solutions may lack the continuity to be strong solutions, but strong solutions are always weak solutions. In situations where weak and strong solutions differ, the weak solution is often the one of physical interest.

Since we have added a derivative to v by the integration by parts, v must be restricted to a space where functions have more continuity than those in Φ . Having symmetry in mind, we will select functions u and v that produce bounded values of

$$\boldsymbol{a}(u,u)=\int_a^b (p(u')^2+qu^2)\,dx$$

Actually, since p and q are smooth functions, it suffices for u and v to have bounded values of

$$\int_a^b ((u')^2+u^2)\,dx \quad (6.2.3)$$

Functions where (6.2.3) exist are said to be elements of the Sobolev Space H^1. The Sovolev space is a Hilbert space, fact that will be useful for satisfying the requeriments of the Galerkin approximation problem. By using the Friendrich inequallity, the concepts of norm and semi-norm in the Sobolev space, we have a mean to verify that the form built form functions in such space meets the requeriments provided by definitions (6.2.1), (6.2.2), (6.2.3). We will not provide the complete proof here. We have also required that u and v satisfy the boundary conditions (5.1b). We identify those functions in H^1 that also satisfy (5.1b) as being elements of H_0^1. Thus in summary the variational problem consist of determining u in H_0^1 such that

$$\boldsymbol{a}(v,u)-(v,f)=0, \qquad \forall v\in H_0^1 \quad (6.2.4)$$

6.3 Galerkin approximation problem

Let us take a roundabout to the problem, starting from the most general case, with the Hilbert space V and finally extending the actual facts to the Sovolev Space H_0^1. By this end, a more general version of (6.2.4) can be achieved by regarding the term (v,f) as a functional $\boldsymbol{F}\in V^*$ with the following characteristics:

$$\boldsymbol{F}(\alpha_1 v+\alpha_2 u)=\alpha_1\boldsymbol{F}(v)+\alpha_2\boldsymbol{F}(u), \quad \forall v,u\in V, \forall \alpha_{1,}\alpha_2\in\Re$$

$$\exists M>0\,|\boldsymbol{F}(v)|\leqslant M\|v\|_v, \quad \forall v\in V$$

Taking into account the definitions (6.2.1), (6.2.2), (6.2.3) the problem of finding a solution for (6.2.4) can be stated as follows:

Given $\boldsymbol{a}: VxV\to\Re$ and $\boldsymbol{F}: V\to\Re$ find $u\in V$ for which,

$$\boldsymbol{a}(v,u)=\boldsymbol{F}(v), \qquad \forall v\in V \quad (6.2.5)$$

As we are searching in the whole space V, the procedure is not efficient; we need to restrict ourselves to a subspace.

Statement of the problem:

Given a finite-dimensional sub-space $V_h \subset V$, and $\boldsymbol{F} \subset V^*$ find $u_h \in V_h$ such that:

$$\boldsymbol{a}(u_h, v) = F(v), \quad \forall v \in V_h \quad (6.2.6)$$

Theorem: (Lax-Miloran)

Given a Hilbert space V, a continuous, coercive and bilinear form $\boldsymbol{a}$, and a continuous linear functional $\boldsymbol{F} \in V^*$ there exist a unique $u \in V$ as solution for (6.2.5)

We are going to use piecewise polynomial approximations of u and v

V_h is a subspace of V, hence it is also a Hilbert space. Also $\boldsymbol{F}|_{v_h} \in V_h^*$ by the Lax-Miloran theorem, the Galerkin approximation problem has unique solution.

What is the error of aproximation of u with u_h?

Theorem: Under the assumption of the Lax-Miloran theorem, the solution u_h to the Galerkin problem satisfices

$$\|u - u_h\|_v, \leqslant \frac{M}{\alpha} \inf_{v \in V_h} \|u - v\|_v \quad \forall v \in V$$

Where *M* is the continuiy constant and α the coercivity constant

Let $(v_1, v_2, \ldots, v_L)$ be a basis of $V_h \subset V$ then,

$$u_h = \sum_{i=1}^{L} \alpha_i v_i$$

A form $\boldsymbol{a}$ is bilinear hence $\boldsymbol{a}(u_h, v_j) = \boldsymbol{a}(\sum_{i=1}^{L} \alpha_i v_i, v_j)$.

As the form is bilinear we can write:

$$\sum_{i=1}^{L} \alpha_i \boldsymbol{a}(v_i, v_j) = \boldsymbol{F}(v_j), \quad j = 1, \ldots, L$$

$$A\bar{\alpha}=\bar{F}$$

when $\bar{\alpha}$ is the vector of all α_i

$$\bar{\alpha}=(\alpha_{1,}\ldots,\alpha_L)^T$$

$$\bar{F}=(\boldsymbol{F}(v_1),\ldots,\boldsymbol{F}(v_L))^T$$

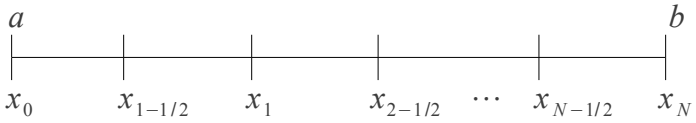
$$A=\{\boldsymbol{a}(v_i,v_j)\}_{i,j=1}^{L}$$

If a is symmetric and coercive, then the matrix A is symmetric and possitive definite.

Definition 6.3.1: A matrix $A\in\Re^{nxn}$ is possitive definite if

$$X^T A X>0,\quad \forall x\in\Re^n, x\neq 0$$

Finally we have that the test space V_h is the Sovolev space H_0^1, which meets all requeriments of the Galerkin approximation problem so the basis $(v_{1,}v_{2,}\ldots,v_L)$ is that of H_0^1, and the form given by (6.2.2b) is enough for our pourposes.



A piecewise parabolic polynomial w.r.t a mesh $x_0=a, x_N=b$. The basis is created from the parabolic elements from section 2.

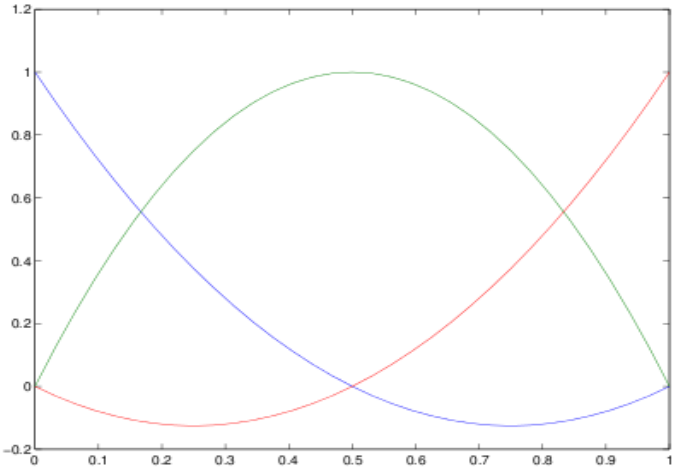


Figure 6.2: The three parabolic elements on the element $K_j=[x_{j-1},x_j], j=1$

If the functionals meet the requirement given in the section 2 the approximation is of the form:

$$u_h(x)=\sum_{j=0}^{2N}(\alpha_{j/2}v_{j/2}(x))$$

For a given point x_k in the domain we have

$$u_h(x_k)=\sum_{j=0}^{2N}(\alpha_{j/2}v_{j/2}(x_k))=\alpha_{1/2}$$

Thus the values at the interior nodes along the grid, are the values of u on those points.

6.4 THE STIFFNESS MATRIX ON THE INTERVAL (LOCAL SYSTEM)

Note that when the bilinear form is symmetric, the matrix A is symmetric and positive-definite, a fact of considerable practical importance for the numerical solution of the linear system. By contrast, this is not usually the case for the standard finite difference methods, except for domains with special geometries, such as rectangular domains.

Yet another noticeable practical feature of the matrix A is its *sparsity*, fact that allows the matrices being stored in efficient manners, yielding fast matrix operations from a computational standpoint.

For a single element and for the chosen space of shape functions, the local system has the form,

$$A_j=\int_{x_j}^{x_{j-1}} p(x)\begin{bmatrix} v'_{j-1} \\ v'_{j-1/2} \\ v'_j \end{bmatrix}\begin{bmatrix} v'_{j-1} & v'_{j-1/2} & v'_j \end{bmatrix}dx \; + \int_{x_j}^{x_{j-1}} q(x)\begin{bmatrix} v_{j-1} \\ v_{j-1/2} \\ v_j \end{bmatrix}\begin{bmatrix} v_{j-1} & v_{j-1/2} & v_j \end{bmatrix}dx$$

With j ranging from 1 to $2N+1$, where N is the number of elements in the grid. By contructing local systems (stiffness matrix for every element) for every $K\in T(\Omega)$ we create a local system that we build in the global system $A\bar{\alpha}=\bar{F}$.

The stiffness matrix is stored in COO format, based on coordinates and values, is compact and is suitable for the final assembly since is possible to sum up repeated values of coordinates as needed for the values of the Matrix on th right nodes of each preceding element and the next leftmost node *(i.e* elements meet at those points as defined for the functionals from the relation (3.1.1).

7. NUMERICAL INTEGRATION

In this work we have chosen the *Clenshaw–Curtis quadrature* for the evaluation of the integrals on the matrix A. This method of numerical integration, or "quadrature", is based on an expansion of the integrand in terms of Chebyshev polynomials.

Equivalently, they employ a change of variables $x = cos\theta$ and use a discrete cosine transform (DCT) approximation for the cosine series.

Briefly, the function *f*(*x*) to be integrated is evaluated at the *N* extrema or roots of a Chebyshev polynomial and these values are used to construct a polynomial approximation for the function. This polynomial is then integrated exactly.

In practice, the integration weights for the value of the function at each node are precomputed, and this computation can be performed in *O*(*N*log*N*) time by means of Fast-Fourier-Transform-related algorithms for the DCT.

The algorithm is normally expressed for integration of a function $f(x)$ over the interval $[-1,1]$ (any other interval can be obtained by appropriate rescaling). For this integral, we can write:

$$\int_{-1}^{1} f(x)\,dx = \int_{0}^{\pi} f(\cos\theta)\sin(\theta)\,d\theta$$

That is, we have transformed the problem from integrating $f(x)$ to one of integrating $f(cos\theta)\,sin\theta$. This can be performed if we know the cosine series for $f(cos\theta)$:

$$f(cos\theta) = \frac{a_0}{2} + \sum_{k=1}^{\infty} a_k \cos(k\theta)$$

In which case the integral becomes:

$$\int_0^{\pi} f(\cos\theta)\sin(\theta)\,d\theta = a_0 + \sum_{k=1}^{\infty} \frac{2a_k}{1-(2k)^2}$$

In order to calculate the cosine series coefficients

$$a_k = \frac{2}{\pi}\int_0^{\pi} f(\cos\theta)\sin(\theta)\,d\theta$$

one must again perform a numeric integration, so at first this may not seem to have simplified the problem. Unlike computation of arbitrary integrals, however, Fourier-series integrations for periodic functions (like $f(\cos\theta)$, by construction), up to the *Nyquist* frequency $k = N$, are accurately computed by the $N + 1$ equally spaced and equally weighted points $\theta_n = n\pi/N$ for $n = 0, \ldots, N$ (except the endpoints are weighted by 1/2, to avoid double-counting, equivalent to the trapezoidal rule or the Euler – McLaurin formula. That is, we approximate the cosine-series integral by the type-I discrete cosine transform (DCT):

$$a_{2k} \approx \frac{2}{N}\left[f\frac{(1)}{2} + \frac{f(-1)}{2}(-1)^k + \sum_{n=1}^{N-1} f(\cos[n\pi/N])\cos[n\pi/N] \right]$$

for $k = 0, \ldots, N$ and then use the formula above for the integral in terms of these a_k. Because only a_{2k} is needed, the formula simplifies further into a type-I DCT of order $N/2$, assuming N is an even number:

$$a_{2k} \approx \frac{2}{N}\left[\frac{f(1)+f(-1)}{2} + f(0)(-1)^k + \sum_{n=1}^{N/2-1} \left\{ f(\cos[n\pi/N]) + f(-\cos[n\pi/N]) \right\} \cos\left(\frac{nk\pi}{N/2}\right) \right]$$

From this formula, it is clear that the Clenshaw–Curtis quadrature rule is symmetric, in that it weights $f(x)$ and $f(-x)$ equally.

Because of aliasing, one only computes the coefficients a_{2k} up to $k=N/2$, since discrete sampling of the function makes the frequency of $2k$ indistinguishable from that of $N–2k$. Equivalently, the a_{2k} are the amplitudes of the unique trigonometric interpolation polynomial with minimal mean-square slope passing through the $N+1$

points where $f(\cos\theta)$ is evaluated, and we approximate the integral by the integral of this interpolation polynomial. There is some subtlety in how one treats the a_N coefficient in the integral, however—to avoid double-counting with its alias it is included with weight 1/2 in the final approximate integral (as can also be seen by examining the interpolation polynomial):

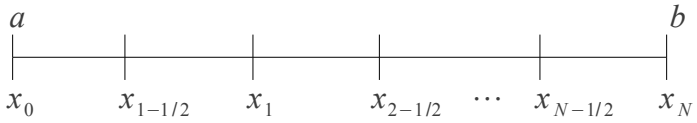
$$\int_0^{\pi} f(\cos\theta)\sin(\theta)\,d\theta \approx a_0 + \sum_{k=1}^{N/2-1} \frac{2a_k}{1-(2k)^2} + \frac{a_N}{1-N^2}$$

8. DESCRIPTION OF THE PROCEDURE

8.1 FINITE ELEMENT FOR A 1 DIMENSIONAL ODE WITH POLYMORPHIC USE OF SHAPE FUNCTIONS

Initialization of the sub-space (Grid) over which we are looking for an approximate solution to the 1-D n-order ordinary differential equation. The parameter k in the initialization is the number of element, currently the elements are numbered as follows:

For N elements:



There exist 2N+1 nodes along the grid. x_0 and x_N correspond to the values at boundaries (i.e $[a,b], x_0=a, x_N=b$)

8.1.1 Build the local stiffness matrix, and force vector

8.1.2 Assemble the global matrix

8.1.3 Apply the Boundary Dirichlet Conditions.

On this project we have decided to conserve the symmetry of the stiffness matrix. The procedure to achieve such an arrangement is described below:

- Replacing the equations 1 and N for the ones given by the boundary. $u_1 = u_{0,}\, u_n = u_n$

$$A_{1,2:N} := 0 \qquad F_1 := a_{1,1} u_0$$
$$A_{N,1:N-1} = 0 \qquad F_N := a_{N,N} u_0$$

yielding the layout below

$$\begin{bmatrix} a_{1,1} & 0 & \cdots & 0 & 0 \\ a_{2,1} & a_{2,2} & \cdots & 0 & 0 \\ a_{3,1} & a_{3,2} & \cdots & 0 & 0 \\ 0 & \vdots & \ddots & \vdots & \vdots \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ 0 & 0 & \cdots & a_{n-1,n-1} & a_{n-1,n} \\ 0 & 0 & \cdots & 0 & a_{n,n} \end{bmatrix}$$

- Finally getting the terms in the columns 1 below a_{11} and N above $a_{N,N}$ passed to the right side of the equation as below:

$$A_{2:N,1} := 0 \qquad F := F - u_0 A_{2:N,1} - u_n A_{1:N-1,n}$$
$$A_{N,1:N-1} := 0$$

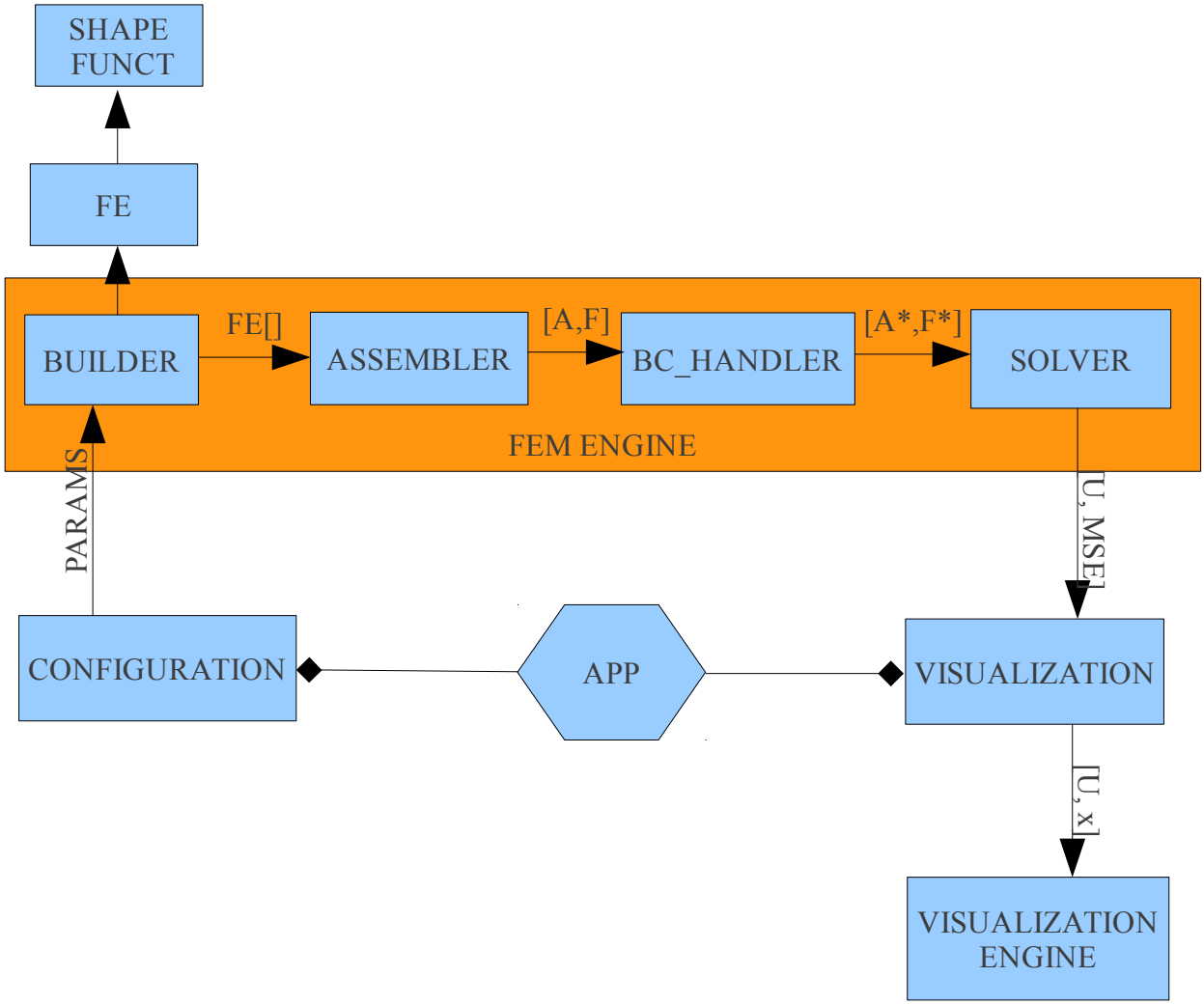
yields to the desired arrangement:

$$\begin{bmatrix} a_{1,1} & 0 & \cdots & 0 & 0 \\ 0 & a_{2,2} & \cdots & 0 & 0 \\ 0 & a_{3,2} & \cdots & 0 & 0 \\ 0 & \vdots & \ddots & \vdots & \vdots \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ 0 & 0 & \cdots & a_{n-1,n-1} & 0 \\ 0 & 0 & \cdots & 0 & a_{n,n} \end{bmatrix}$$

This way we can apply an efficient solver for the system of equations taking advantage of the symmetry of the problem. It can be proved that the arrangement yielding from this procedure is still positive definite.

9. APPLICATION

The FEM Engine, is the core of the application. The tool-chain is depicted below:



BUILDER: Handles the process of building the local systems from the parameters passed by the configuration tool in the application. All the matrices are build in symbolic way by using the package sympy for python, then the integrals are evaluated numerically by using the quadrature described in the section 7. Its output is the space of finite elements, each one with its calculated local stiffness matrix.

ASSEMBLER: Deals with the assembly of the final arrangement from the individual set of elements given by the builder. It uses the procedure described on the seccion 6.4 taking advantage of the COO format.

BC_HANDLER: Since the matrix generated by the assembler is in general singular, we need to apply the boundary conditions to the system. The boundary conditions are in this case, the dirichlet conditions. The algorithm described on the section 8.1.3, yiels a symmetric positive-definite matrix and the force vector, a system ready for

being solved.

SOLVER: The solver uses the system given by the BC_Handler and applies the general conjugate gradient algorithm for solving such a system of equations, yielding the solution of the problem ready for being visualized to the user.

10. NUMERICAL TEST

We proceed now to drive a series of tests over the final implementation to check its correctness. Consider the finite element solution of:

$$-\frac{d^2u}{dx^2}+u=x, \quad u=[0,1]$$

$$u(0)=0, u(b)=0$$

with exact solution:

$$u(x)=-\frac{\sinh(x)}{\sinh(1)}+x$$

The configuration of the main screen looks as follows:

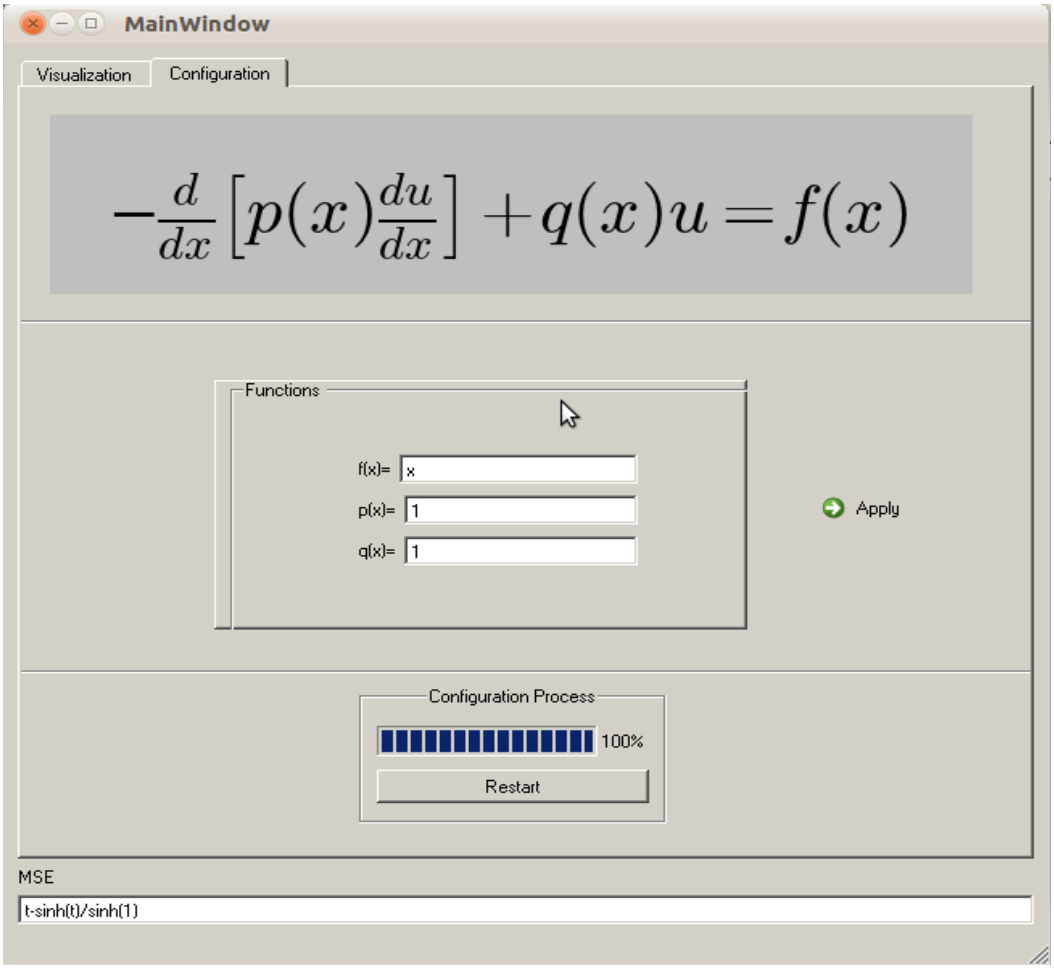


Ilustración 10.1: Main screen

In the implementation, the textbox labeled MSE at the bottom of the screen is the imput of the exact solution with a change of variable from “x” to “t”. The output of the mean square error is also logged. Once the configuration proccess has been finished we procceed to start calculations. At run time, the console screen shows a series of logging messages from the application, namely a message from the shape functions and the grid built upon the given specifications. The table 9.1 shows the resulting Mean square error for a range of number of elements.

N of Elements	MSE
2	9,27529903269900E-11
4	3,97338943539288E-13
8	2,48496886698318E-14
12	1,56061869292804E-14
14	7,05939800547579E-15
15	7,45020840178585E-14
16	3,24856124348579E-13
32	5,32388935333732E-12
64	8,58926884033381E-11
128	1,38322282707754E-9

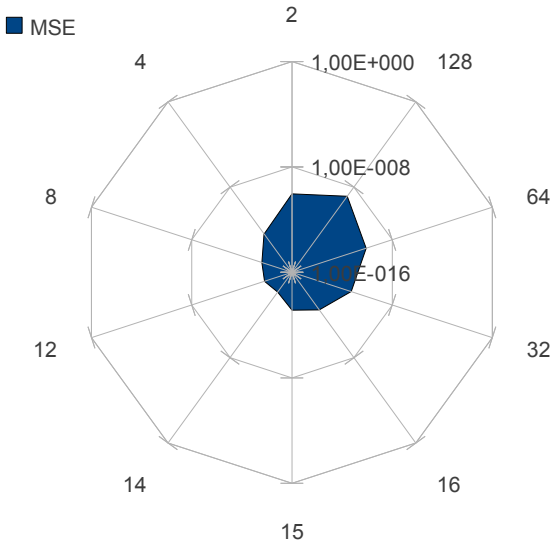


Table 9.1: Mean Square error for a given number of elements

The results of the analisys are very insighful. The solution is very accurate for the minimum number of elements and up to 14 elements the Mean-square-error decreases very slowly, when the numerical accuracy starts to degradate due to the accumulation of round-off error. The graph on the right of Table 9.1 is given in logaritmic scale for the sake of proper visualization of the solution performance.

The figure 9.2 shows the console output in the case of 2 elements, the grid as seen, has 2*N+1 nodes and the shape functions for the left, middle, and right case are called twice and share the rightmost point of the first element (equivalently the leftmost one from the second element).

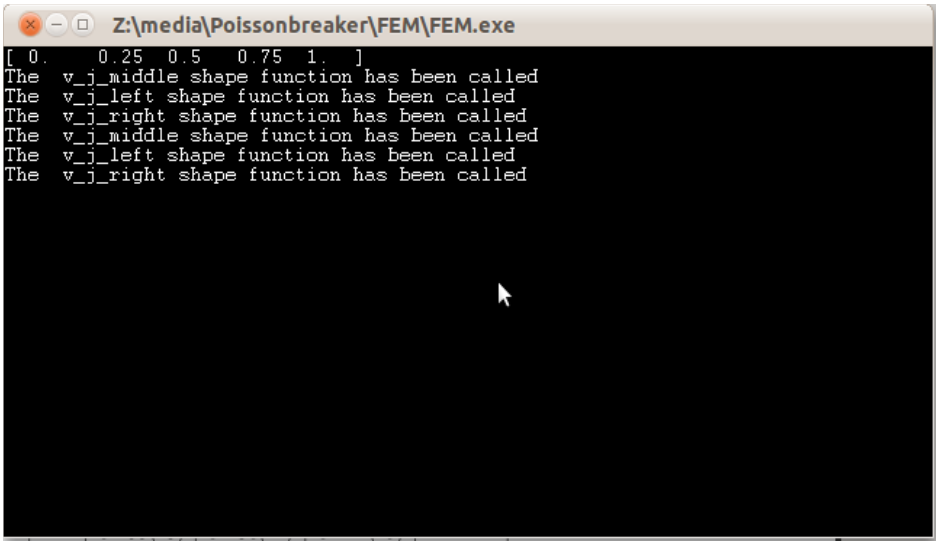


Ilustración 10.2: Logging on console

The figures from 10.3 to 10.5 shows graphically the behavior of the solution under the increasing of number of elements. Its noticeable how the elements being very close to each other as in the figure 10.4 degrades the correct behavior of the method.

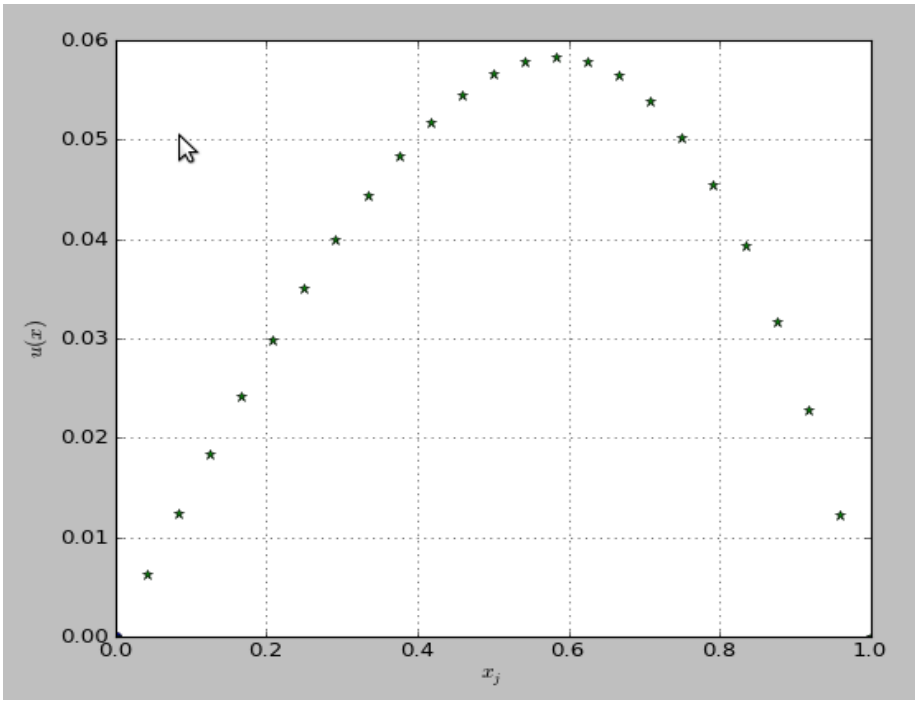


Ilustración 10.3: Solution for 12 elements

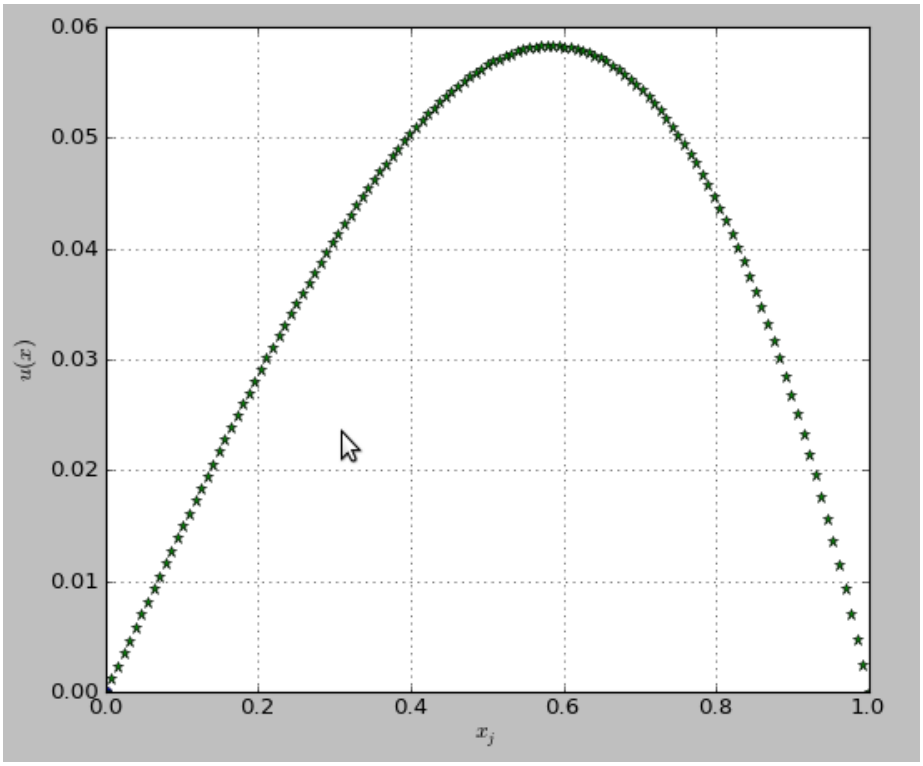


Ilustración 10.4: Solution for 32 elements

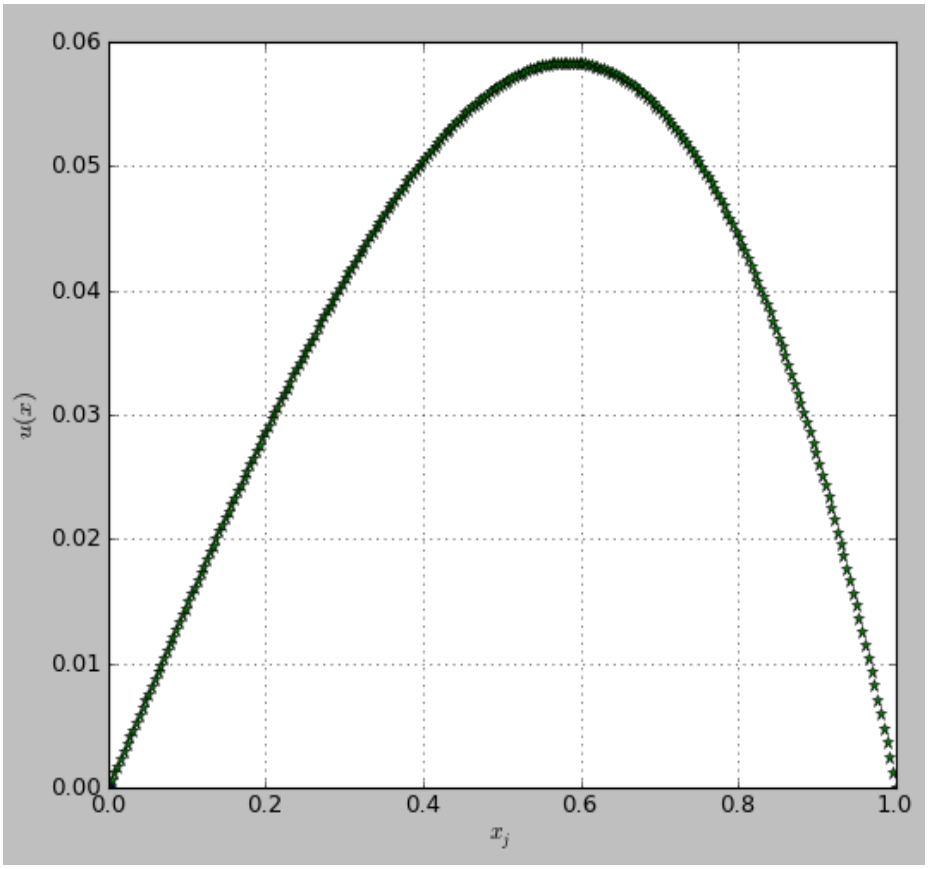


Ilustración 10.5: Solution for 64 elements

It has been proved then the high accuracy of the method for solving a second order differential equation when both, the solution and the piecewise-parabolic elements are “close” as in the case above. In the studied case the initial rate of error reduction with the increase of elements was relatively slow, so we could have chosen very early a point for breaking further calculations. Nevertheless it has to be said that is a good exercise for gaining a better insight on the method.