A gentle introduction to the Finite Element Method

Francisco-Javier Sayas

April 27, 2015

An introduction

If you haven't been hiding under a stone during your studies of Engineering, Mathematics or Physics, it is very likely that you have already heard about the Finite Element Method. Maybe you even know some theoretical and practical aspects and have played a bit with FEM software. What you are going to find here is a detailed and mathematically biased introduction to several aspects of the Finite Element Method. This is not however a course on the analysis of the method. It is just a demonstration of how it works, written as applied mathematicians usually write it. There are going to be Mathematics involved, but not lists of theorems and proofs. We are also going from the most particular cases towards useful generalizations, from example to theory.

An aspect where this course differs from most of the many introductory books on finite elements is the fact that I am going to begin directly with the two-dimensional case. I have just sketched the one dimensional case in an appendix. Many people think that the one-dimensional case is a better way of introducing the method, but I have an inner feeling that the method losses richness in that very simple situation, so I prefer going directly to the plane.

The course is divided into five lessons and is thought to be read in that order. We cover the following subjects (but not in this order):

- triangular finite elements,
- finite elements on parallelograms and quadrilaterals,,
- adaptation to curved boundaries (isoparametric finite elements),
- three dimensional finite elements,
- assembly of the finite element method,
- some special techniques such as static condensation or mass lumping,
- eigenvalues of the associated matrices,
- approximation of evolution problems (heat and wave equations).

It is going to be around one hundred pages with many figures. Ideas will be repeated over and over, so that you can read this with ease. These notes have evolved during the decade I have been teaching finite elements to mixed audiences of mathematicians, physicists and engineers. The tone is definitely colloquial. I could just claim that these are my classnotes

and that's what I'm like[1]. There's much more than that. First, I believe in doing your best at being entertaining when teaching. At least that's what I try. Behind that there is a deeper philosophical point: take your work (and your life) seriously but, please, don't take yourself too seriously.

I also believe that people should be duly introduced when they meet. All this naming old time mathematicians and scientists only by their last names looks to me too much like the Army. Or worse, high school![2] I think you have already been properly introduced to the great Leonhard Euler, David Hilbert, Carl Friedrich Gauss, Pierre Simon Laplace and George Green[3]. If you haven't so far, consider it done here. This is not about history. It's just good manners. Do you see what I mean by being colloquial?

Anyway, this is not about having fun[4], but since we are at it, let us try to have a good time **while learning**. If you take your time to read these notes with care and try the exercises at the end of each lesson, I can assure that you will have made a significant step in your scientific persona. Enjoy!

These notes were written in its present form during my first year as visiting faculty at the University of Minnesota. They constitute an evolved form of my lecture notes to teach Finite Elements at the graduate level, something I have done for many years in the University of Zaragoza (Spain). The version you are reading now is a revision produced for teaching at the University of Delaware.

[1]To the very common comment *every person has their ways*, the best answer I've heard is *Oh, God, no! We have good manners for that.*

[2]When I was in high school, boys were called by their last names. I was Sayas all over. On the other hand, girls were called by their first names.

[3]You will find here the names of Peter Lejeune Dirichlet, Carl Neumann or Sergei Sobolev, associated to different concepts of PDE theory

[4]Unfortunately too many professional mathematicians advocate fun or beauty as their main motivations to do their job. It is so much better to have a scientific calling than this aristocratic detachment from work...

Lesson 1

Linear triangular elements

1 The model problem

All along this course we will be working with a simple model boundary value problem, which will allow us to put the emphasis on the numerical method rather than on the intricacies of the problem itself. For some of the exercises and in forthcoming lessons we will complicate things a little bit.

In this initial section there is going to be a lot of new stuff. Take your time to read it carefully, because we will be using this material during the entire course.

1.1 The physical domain

The first thing we have to describe is the geometry (the physical setting of the problem). You have a sketch of it in Figure 1.1.

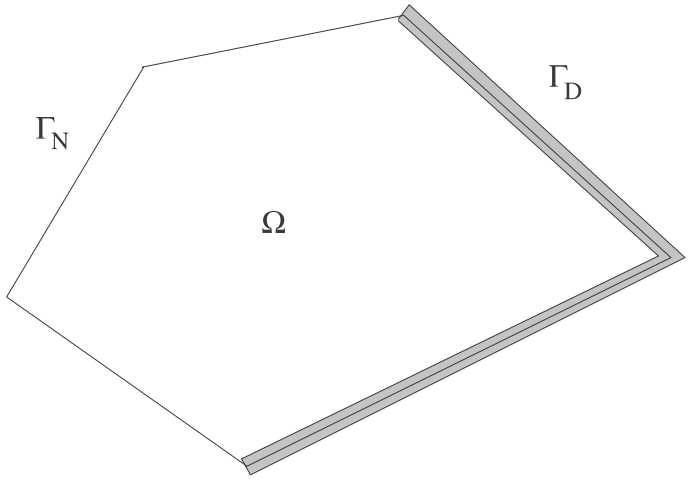


Figure 1.1: The domain Ω and the Dirichlet and Neumann boundaries.

We are thus given a polygon in the plane $\mathbb{R}^2$. We call this polygon Ω. Its boundary is a closed polygonal curve Γ. (There is not much difference if we suppose that there is

one or more holes inside Ω, in which case the boundary is composed by more than one polygonal curve).

The boundary of the polygon, Γ is divided into two parts, that cover the whole of Γ and do not overlap:

- the Dirichlet boundary Γ_D,
- the Neumann boundary Γ_N.

You can think in more mechanical terms as follows: the Dirichlet boundary is where displacements are given as data; the Neumann boundary is where normal stresses are given as data.

Each of these two parts is composed by full sides of the polygon. This is not much of a restriction if you admit the angle of 180 degrees as separating two sides, that is, if you want to divide a side of the boundary into parts belonging to Γ_D and Γ_N, you just have to consider that the side is composed of several smaller sides with a connecting angle of 180 degrees.

1.2 The problem, written in strong form

In the domain we will have an elliptic partial differential equation of second order and on the boundary we will impose conditions on the solution: boundary conditions or boundary values. Just to unify notations (you may be used to different ways of writing this), we will always write the Laplace operator, or Laplacian, as follows

$$\Delta u = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}.$$

By the way, sometimes it will be more convenient to call the space variables (x_1, x_2) rather than (x, y), so expect mixed notations.

The boundary value problem is then

$$\begin{cases} -\Delta u + c\, u = f & \text{in } \Omega, \\ u = g_0 & \text{on } \Gamma_D, \\ \partial_n u = g_1 & \text{on } \Gamma_N. \end{cases}$$

There are new many things here, so let's go step by step:

- The unknown is a (scalar valued) function u defined on the domain Ω.
- c is a non-negative constant value. In principle we will consider two values $c = 1$ and $c = 0$. The constant c is put there to make clear two different terms when we go on to see the numerical approximation of the problem. By the way, this equation is usually called a **reaction-diffusion equation**. The diffusion term is given by $-\Delta u$ and the reaction term, when $c > 0$, is $c\, u$.
- f is a given function on Ω. It corresponds to source terms in the equation. It can be considered as a surface density of forces.

- There are two functions g_0 and g_1 given on the two different parts of the boundary. They will play very different roles in our formulation. As a general rule, we will demand that g_0 is a continuous function, whereas g_1 will be allowed to be discontinuous.

- The symbol ∂_n denotes the exterior normal derivative, that is,

$$\partial_n u = \nabla u \cdot \mathbf{n},$$

  where $\mathbf{n}$ is the unit normal vector on points of Γ pointing always outwards and ∇u is, obviously, the gradient of u.

We are not going to worry about regularity issues. If you see a derivative, admit that it exists and go on. We will reach a point where everything is correctly formulated. And that moment we will make hypotheses more precise. If you are a Mathematician and are already getting nervous, calm down and believe that I know what I'm talking about. Being extra rigorous is not what is important at this precise time and place.

1.3 Green's Theorem

The approach to solve this problem above with the Finite Element Method is based upon writing it in a completely different form, which is sometimes called **weak or variational form**. At the beginning it can look confusing to see all this if you are not used to advanced Mathematics in Continuum Mechanics or Physics. We are just going to show here how the formulation is obtained and what it looks like at the end. You might be already bored in search of matrices and something more tangible! Don't rush! If you get familiarized with formulations and with the notations Mathematicians given to frame the finite element method, many doors will be open to you: you will be able to read a large body of literature that would be ununderstandable to you if you stick to what you already know.

The most important theorem in this process or reformulating the problem is Green's Theorem, one of the most popular results of Vector Calculus. Sometimes it is also called Green's First Formula (there's a popular second one and a less known third one). The theorem states that

$$\int_\Omega (\Delta u)\, v + \int_\Omega \nabla u \cdot \nabla v = \int_\Gamma (\partial_n u)\, v.$$

Note that there are two types of integrals in this formula. Both integrals in the left-hand side are domain integrals in Ω, whereas the integral in the right-hand side is a line integral on the boundary Γ. By the way, the result is also true in three dimensions. In that case, domain integrals are volume integrals and boundary integrals are surface integrals. The dot between the gradients denotes simply the Euclidean product of vectors, so

$$\nabla u \cdot \nabla v = \frac{\partial u}{\partial x_1}\frac{\partial v}{\partial x_1} + \frac{\partial u}{\partial x_2}\frac{\partial v}{\partial x_2}$$

Remark. This theorem is in fact a simple consequence of the Divergence Theorem:

$$\int_\Omega (\operatorname{div} \mathbf{p})\, v + \int_\Omega \mathbf{p} \cdot \nabla v = \int_\Gamma (\mathbf{p} \cdot \mathbf{n})\, v.$$

Here $\operatorname{div} \mathbf{p}$ is the divergence of the vector field $\mathbf{p}$, that is, if $\mathbf{p} = (p_1, p_2)$

$$\operatorname{div} \mathbf{p} = \frac{\partial p_1}{\partial x_1} + \frac{\partial p_2}{\partial x_2}.$$

If you take $\mathbf{p} = \nabla u$ you obtain Green's Theorem. □

1.4 The problem, written in weak form

The departure point for the weak or variational formulation is Green's Theorem. Here it is again

$$\int_\Omega (\Delta u)\, v + \int_\Omega \nabla u \cdot \nabla v = \int_\Gamma (\partial_n u)\, v = \int_{\Gamma_D} (\partial_n u)\, v + \int_{\Gamma_N} (\partial_n u)\, v.$$

Note that we have broken the integral on Γ as the sum of the integrals over the two sub-boundaries, the Dirichlet and the Neumann boundary. You may be wondering what v is in this context. In fact, it is nothing but a test. Wait for comments on this as the section progresses.

Now we substitute what we know in this formula: we know that $\Delta u = c\, u - f$ in Ω and that $\partial_n u = g_1$ on Γ_N. Therefore, after some reordering

$$\int_\Omega \nabla u \cdot \nabla v + c \int_\Omega u\, v = \int_\Omega f\, v + \int_{\Gamma_N} g_1\, v + \int_{\Gamma_D} (\partial_n u)\, v.$$

Note now that I have written all occurrences of u on the left hand side of the equation except for one I have left on the right. In fact we don't know the value of $\partial_n u$ on that part of the boundary. So what we will do is impose that v cancels in that part, that is,

$$v = 0 \qquad \text{on } \Gamma_D.$$

Therefore

$$\int_\Omega \nabla u \cdot \nabla v + c \int_\Omega u\, v = \int_\Omega f\, v + \int_{\Gamma_N} g_1\, v, \qquad \text{if } v = 0 \text{ on } \Gamma_D.$$

Notice now three things:

- We have not imposed the Dirichlet boundary condition ($u = g_0$ on Γ_D) yet. Nevertheless, we have imposed a similar one to the function v, *but in a homogeneous way.*
- As written now, data (f and g_1) are in the right-hand side and coefficients of the equation (the only one we have is c) are in the left-hand side.

- The expression on the left-hand side is linear in both u and v. It is a bilinear form of the variables u and v. The expression on the right-hand side is linear in v.

Without specifying spaces where u and v are, the weak formulation can be written as follows:

$$\left[\begin{array}{l} \text{find } u \text{ such that} \\ u = g_0 \qquad \text{on } \Gamma_D, \\ \displaystyle\int_\Omega \nabla u \cdot \nabla v + c \int_\Omega u\, v = \int_\Omega f\, v + \int_{\Gamma_N} g_1\, v \qquad \text{for all } v \text{ such that } v = 0 \text{ on } \Gamma_D. \end{array}\right.$$

Note how the two boundary conditions appear in very different places of this formulation:

- The Dirichlet condition (given displacements) is imposed apart from the formulation and involves imposing it homogeneously to the testing function v. It is called an **essential boundary condition**.
- The Neumann condition (given normal stresses) appears inside the formulation. It is called a **natural boundary condition**.

Being essential or natural is not inherently tied to the boundary condition: it is related to the role of the boundary condition in the formulation. So when you hear (or say) essential boundary condition, you mean a boundary condition that is imposed apart from the formulation, whereas a natural boundary condition appears inside the formulation. *For this weak formulation of a second order elliptic equation we have*

$$\textit{Dirichlet=essential} \qquad \textit{Neumann=natural}$$

What is v? At this point, you might (you should) be wondering what v is in the formulation. In the jargon of weak formulations, v is called a test function. It tests the equation that is satisfied by u. The main idea is that instead of looking at the equation as something satisfied point-by-point in the domain Ω, you have an averaged version of the equation. Then v plays the role of a weight function, something you use to average the equation. In many contexts (books on mechanics, engineering or physics) v is called a virtual displacement (or virtual work, or virtual whatever is pertinent), emphasizing the fact that v is not the unknown of the system, but something that only exists virtually to write down the problem. The weak formulation is, in that context, a principle of virtual displacements (principle of virtual work, etc). □

1.5 Delimiting spaces

We have reached a point where we should be a little more specific on where we are looking for u and where v belongs. The first space we need is the space of square-integrable functions

$$L^2(\Omega) = \Big\{ f : \Omega \to \mathbb{R} \,:\, \int_\Omega |f|^2 < \infty \Big\}.$$

A fully precise definition of this space requires either the introduction of the Lebesgue integral or applying some limiting ideas. If you know what this is all about, good for you! If you don't, go on: for most functions you know you will always be able to check whether they belong to this space or not by computing or estimating the integral and seeing if it is finite or not.

The second space is one of the wide family of Sobolev spaces:

$$H^1(\Omega) = \left\{ u \in L^2(\Omega) \,:\, \tfrac{\partial u}{\partial x_1}, \tfrac{\partial u}{\partial x_2} \in L^2(\Omega) \right\}.$$

There is a norm related to this space

$$\|u\|_{1,\Omega} = \left(\int_\Omega |\nabla u|^2 + \int_\Omega |u|^2 \right)^{1/2} = \left(\int_\Omega \left| \frac{\partial u}{\partial x_1} \right|^2 + \int_\Omega \left| \frac{\partial u}{\partial x_2} \right|^2 + \int_\Omega |u|^2 \right)^{1/2}.$$

Sometimes this norm is called the energy norm and functions that have this norm finite (that is, functions in $H^1(\Omega)$) are called functions of finite energy. The concept of energy is however related to the particular problem, so it's better to get used to have the space and its norm clearly written down and think of belonging to this space as a type of admissibility condition.

A particular subset of this space will be of interest for us:

$$H^1_{\Gamma_D}(\Omega) = \{ v \in H^1(\Omega) \,:\, v = 0 \quad \text{on } \Gamma_D \}.$$

Note that $H^1_{\Gamma_D}(\Omega)$ is a subspace of $H^1(\Omega)$, that is, linear combinations of elements of $H^1_{\Gamma_D}(\Omega)$ belong to the same space.

The Mathematics behind. An even half-trained Mathematician should be wondering what do we mean by the partial derivatives in the definition of $H^1(\Omega)$, since one cannot think of taking the gradient of an arbitrary function of $L^2(\Omega)$, or at least to taking the gradient and finding something reasonable. What we mean by restriction to Γ_D in the definition of $H^1_{\Gamma_D}(\Omega)$ is not clear either, since elements or $L^2(\Omega)$ are not really functions, but classes of functions, where values of the function on particular points or even on lines are not relevant. To make this completely precise there are several ways:

- Define a weak derivative for elements of $L^2(\Omega)$ and what we understand by saying that that derivative is again in $L^2(\Omega)$. Then you move to give a meaning to that restriction of a function in $H^1(\Omega)$ to one part of its boundary.

- Go the whole nine yards and take time to browse a book on distribution theory and Sobolev spaces. It takes a while but you end up with a pretty good intuition of what this all is about.

- Take a shortcut. You first consider the space of functions

$$\mathcal{C}^1(\overline{\Omega}) = \left\{ u \in \mathcal{C}(\overline{\Omega}) \,:\, \tfrac{\partial u}{\partial x_1}, \tfrac{\partial u}{\partial x_2} \in \mathcal{C}(\overline{\Omega}) \right\},$$

which is simple to define, and then you close it with the norm $\|\cdot\|_{1,\Omega}$. To do that you have to know what closing or completing a space is (it's something similar to what you do to define real numbers from rational numbers). Then you have to prove that restricting to Γ_D still makes sense after this completion procedure.

My recommendation at this point is to simply go on. If you are a Mathematician you can take later on some time with a good simple book on elliptic PDEs and will see that it is not that complicated. If you are a Physicist or an Engineer you will probably not need to understand all the details of this. There's going to be a very important result in the next section that you will have to remember and that's almost all. Nevertheless, if you keep on doing research related to finite elements, you should really know something more about this. In due time you will have to find any of the dozens of books on Partial Differential Equations for Scientists and Engineers, and read the details, which will however not be given in the excruciating detail of PDE books for Mathematicians. But this is only an opinion. □

1.6 The weak form again

With the spaces defined above we can finally write our problem in a proper and fully rigorous way:

$$\left[\begin{array}{l} \text{find } u \in H^1(\Omega) \text{ such that} \\ u = g_0 \quad \text{on } \Gamma_D, \\ \displaystyle\int_\Omega \nabla u \cdot \nabla v + c \int_\Omega u\, v = \int_\Omega f\, v + \int_{\Gamma_N} g_1\, v \qquad \forall v \in H^1_{\Gamma_D}(\Omega). \end{array}\right.$$

Let me recall that the condition on the general test function $v \in H^1_{\Gamma_D}(\Omega)$ is the same as

$$v \in H^1(\Omega) \quad \text{such that } v = 0 \text{ on } \Gamma_D,$$

that is, v is in the same space as the unknown u but satisfies a homogeneous version of the essential boundary condition.

The data are in the following spaces

$$f \in L^2(\Omega), \qquad g_1 \in L^2(\Gamma_N), \qquad g_0 \in H^{1/2}(\Gamma_D).$$

We have already spoken of the first of these spaces. The space $L^2(\Gamma_N)$ follows essentially the same idea, with line integrals on Γ_N instead of domain integrals on Ω. The last space looks more mysterious: it is simply the space of restrictions to Γ_D of functions of $H^1(\Omega)$, that is, $g_0 \in H^{1/2}(\Gamma_D)$ means that there exists at least a function $u_0 \in H^1(\Omega)$ such that $u_0 = g_0$ on Γ_D. In fact, all other functions satisfying this condition (in particular our solution u) belong to

$$u_0 + H^1_{\Gamma_D}(\Omega) = \{u_0 + v \,:\, v \in H^1_{\Gamma_D}(\Omega)\} = \{w \in H^1(\Omega) \,:\, w = g_0 \quad \text{on } \Gamma_D\}$$

(can you see why?). Unlike $H^1_{\Gamma_D}(\Omega)$, this set is not a subspace of $H^1(\Omega)$. The only exception is the trivial case, when $g_0 = 0$, since the set becomes $H^1_{\Gamma_D}(\Omega)$.

The fact that g_0 is in $H^{1/2}(\Gamma_D)$ simply means that we are not looking for the solution in the empty set. I cannot give you here a simple and convincing explanation on the name of this space. Sorry for that.

2 The space of continuous linear finite elements

It's taken a while, but we are there! *Numerics start here.* We are now going to discretize all the elements appearing in this problem: the physical domain, the function spaces and the variational/weak formulation.

We are going to do it step by step. At the end of this section you will have the simplest example of a space of finite element functions (or simply finite elements). Many Mathematicians call these elements Courant elements, because Richard Courant introduced them several decades ago with theoretical more than numerical intentions. In the jargon of the business we call them triangular Lagrange finite elements of order one, or simply linear finite elements, or for short (because using initials and short names helps speaking faster and looking more dynamic) $\mathbb{P}_1$ elements.

2.1 Linear functions on a triangle

First of all, let us think for a moment about linear functions. A linear function (or, more properly, affine function) of two variables is the same as a polynomial function of degree at most one

$$p(x_1, x_2) = a_0 + a_1\, x_1 + a_2\, x_2.$$

The set of these functions is denoted $\mathbb{P}_1$. Everybody knows that a linear function is uniquely determined by its values on three different non-aligned points, that is, on the vertices of a (non-degenerate) triangle.

Let us then take an arbitrary non-degenerate triangle, that we call K. You might prefer calling the triangle T, as many people do. However, later on (in Lesson 3) the triangle will stop being a triangle and will become something else, maybe a quadrilateral, and then the meaning of the initial T will be lost. We draw it as in Figure 1.2, marking its three vertices. With this we mean that *a function*

$$p \in \mathbb{P}_1 = \{a_0 + a_1\, x_1 + a_2\, x_2 \,:\, a_0, a_1, a_2 \in \mathbb{R}\}$$

is uniquely determined by its values on these points. Uniquely determined means two things: (a) there is only one function with given values on the vertices; (b) there is in fact one function, that is, the values on the vertices are arbitrary. We can take any values we want and will have an element of $\mathbb{P}_1$ with these values on the vertices. Graphically it is just hanging a flat (linear) function from three non-aligned points.

Thus, a function $p \in \mathbb{P}_1$ can be determined

- either from its three defining coefficients (a_0, a_1, a_2)
- or from its values on the three vertices of a triangle K.

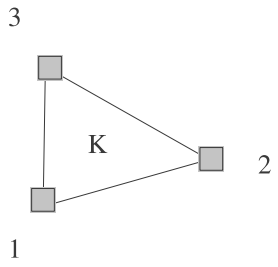


Figure 1.2: A triangle and its three vertices.

Both possibilities state that the space $\mathbb{P}_1$ is a vector space of dimension three. While the first choice (coefficients) gives us a simple expression of the function, the second is more useful for many tasks, in particular for drawing the function. The three values of the function on the vertices will be called the **local degrees of freedom**.

There is another important property that will be extremely useful in the sequel: *the value of $p \in \mathbb{P}_1$ on the edge that joins two vertices of the triangle depends only on the values of p on this two vertices.* In other words, the value of $p \in \mathbb{P}_1$ on an edge is uniquely determined by the degrees of freedom associated to the edge, namely, the values of p on the two vertices that lie on that edge.

2.2 Triangulations

So far we have functions on a single triangle. We now go for partitions of the domain into triangles. A triangulation of Ω is a subdivision of this domain into triangles. Triangles must cover all Ω but no more and must fulfill the following rule:

> *If two triangles have some intersection, it is either a common vertex or a common full edge. In particular, two different triangles do not overlap.*

Figure 1.3 shows two forbidden configurations. See Figure 1.5 to see how a triangulation looks like. There is another rule, related to the partition of Γ into Γ_D and Γ_N:

> *The triangulation must respect the partition of the boundary into Dirichlet and Neumann boundaries.*

This means that an edge of a triangle that lies on Γ cannot be part Dirichlet and part Neumann. Therefore if there is a transition from Dirichlet to Neumann boundaries, there must be a vertex of a triangle in that transition point. Note that this situation has to be taken into account only when there is a transition from Dirichlet to Neumann conditions inside a side of the polygon Ω.

The set of the triangles (that is, the list thereof) will be generally denoted $\mathcal{T}_h$. The subindex h makes reference to the diameter of the triangulation, defined as *the length of the longest edge of all triangles*, that is, the longest distance between vertices of the triangulation.

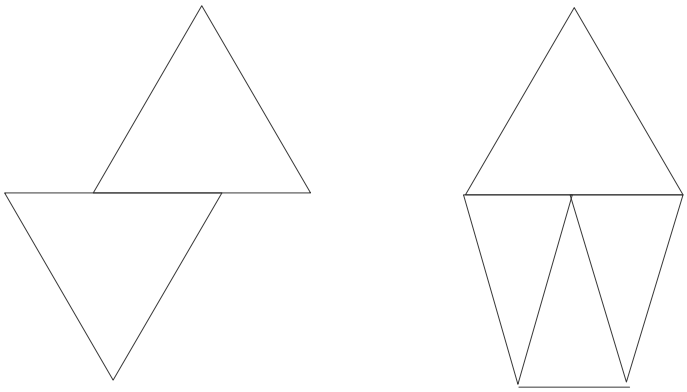


Figure 1.3: Situations not admitted in triangulations. In the second one we see the appearance of what is called a hanging node.

2.3 Piecewise linear functions on a triangulation

We now turn our attention to functions defined on the whole of the polygon Ω that has been triangulated as shown before.

Consider first two triangles sharing a common edge, say K and K' (see Figure 1.6). We take values at the four vertices of this figure and build a function that belongs to $\mathbb{P}_1$ on each of the triangles and has the required values on the vertices. Obviously we can define a unique function with this property. Moreover, since the value on the common edge depends only on the values on the two common vertices, the resulting function is continuous.

We can do this triangle by triangle. We end up with a function that is linear on each triangle and globally continuous. The space of such functions is

$$V_h = \left\{ u_h \in \mathcal{C}(\overline{\Omega}) \,:\, u_h|_K \in \mathbb{P}_1, \quad \forall K \in \mathcal{T}_h \right\}.$$

If we fix values on the set of vertices of the triangulation $\mathcal{T}_h$, there exists a unique $u_h \in V_h$ with those values on the vertices. Therefore an element of V_h is uniquely determined by its values on the set of vertices of the triangulation. The values on the vertices of the whole triangulation are the degrees of freedom that determine an element of V_h. In this context we will call **nodes** to the vertices in their role as points where we take values. (In forthcoming lessons there will be other nodes in addition to vertices.)

Elements of the space V_h are called linear finite element functions or simply $\mathbb{P}_1$ finite elements.

Let us take now a numbering of the set of nodes (that is, vertices) of the triangulation. At this moment any numbering goes[1]. In Figure 1.7 we have a numbering of the nodes of the triangulation of our model domain. The vertices will be generically denoted $\mathbf{p}_i$ with i varying from one to the number of vertices, say N.

[1]And in many instances this will be so to the end of the discretization process. Using one numbering or another has a great influence on the shape of the linear system we will obtain in Section 3, but this shape is relevant only for some choices of the method to solve the corresponding linear system.

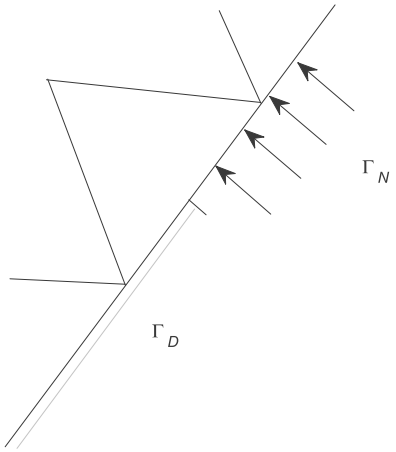


Figure 1.4: A forbidden transition of Dirichlet to Neumann boundary conditions happening inside an edge. Graphical notation for Dirichlet a Neumann boundaries as shown in many Mechanics books are given in the graph.

Because of what we have explained above, if we fix one node (vertex) and associate the value one to this node and zero to all others, there exists a unique function $\varphi_i \in V_h$ that has these values, that is,

$$\varphi_i(\mathbf{p}_j) = \delta_{ij} = \begin{cases} 1, & j = i, \\ 0, & j \neq i. \end{cases}$$

The aspect of one of these functions is shown in Figure 1.8.

Notice that if a triangle K has not $\mathbf{p}_i$ as one of its vertices, φ_i vanishes all over K, since the value of φ_i on the three vertices of K is zero. Therefore, the support of φ_i (the closure of the set of points where φ_i is not zero) is the same as the union of triangles that share $\mathbf{p}_i$ as vertex. In Figure 1.9 you can see the type of supports you can find.

There is even more. Take $u_h \in V_h$. It is simple to see that

$$u_h = \sum_{j=1}^{N} u_h(\mathbf{p}_j)\varphi_j.$$

Why? Let me explain. Take the function $\sum_{j=1}^{N} u_h(\mathbf{p}_j)\varphi_j$ and evaluate it in $\mathbf{p}_i$: you obtain

$$\sum_{j=1}^{N} u_h(\mathbf{p}_j)\varphi_j(\mathbf{p}_i) = \sum_{j=1}^{N} u_h(\mathbf{p}_j)\delta_{ji} = u_h(\mathbf{p}_i).$$

Therefore, this function has exactly the same nodal values as u_h and must be u_h. The fact that two functions of V_h with the same nodal values are the same function is the linear independence of the nodal functions $\{\varphi_i\}$. What we have proved is the fact that $\{\varphi_i : i = 1, \ldots, N\}$ is a basis of V_h and therefore

$$\dim V_h = N = \#\{\text{vertices}\}.$$

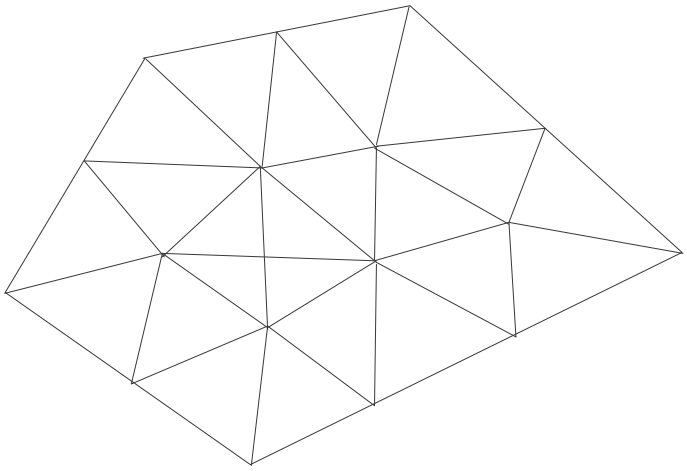


Figure 1.5: A triangulation of Ω.

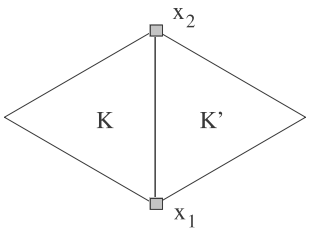


Figure 1.6: Two triangles with a common edge.

There is a particularly interesting aspect of this basis of V_h that makes it special. In general if you have a basis of V_h you know that you can decompose elements of V_h as a unique linear combination of the elements of the basis, that is,

$$u_h = \sum_{j=1}^{N} u_j \, \varphi_j$$

is a general element of V_h. With this basis, the coefficients are precisely the values of u_h on the nodes, that is, $u_j = u_h(\mathbf{p}_j)$. Hence, the coefficients of u_h in this basis are something more than coefficients: there are values of the function on points.

An important result. As you can see, when defining the space V_h we have just glued together $\mathbb{P}_1$ functions on triangles. Thanks to the way we have made the triangulation and to the way we chose the local degrees of freedom, what we obtained was a continuous function. One can think, is this so important? Could I take something discontinuous? At this level, the answer is a very loud and clear NO! The reason is the following result that allows us to know whether certain functions are in $H^1(\Omega)$ or not.

Theorem. *Let u_h be a function defined on a triangulation of Ω such that*

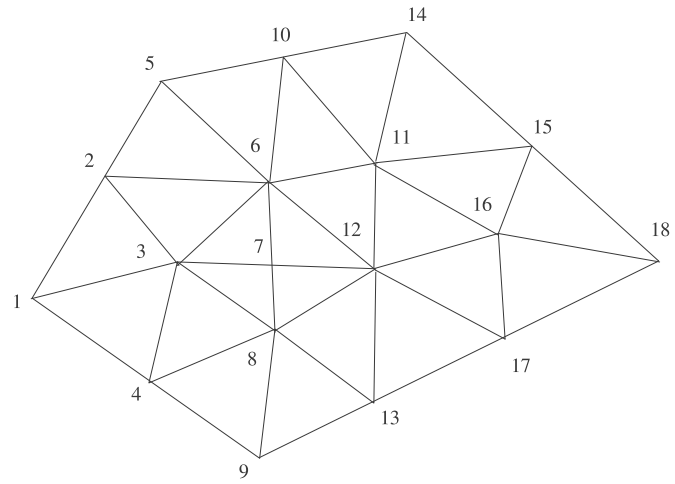


Figure 1.7: Global numbering of nodes.

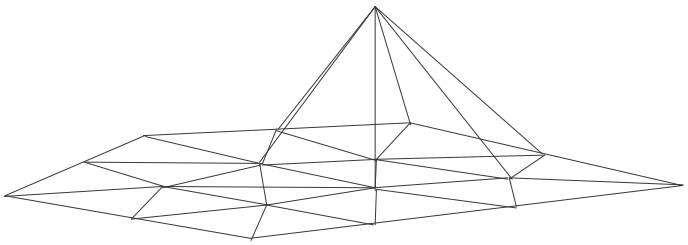


Figure 1.8: The graph of a nodal basis function: it looks like a camping tent.

restricted to each triangle it is a polynomial (or smooth) function. Then

$$u_h \in H^1(\Omega) \qquad \Longleftrightarrow \qquad u_h \text{ is continuous.}$$

There is certain intuition to be had on why this result is true. If you take a derivative of a piecewise smooth function, you obtain Dirac distributions along the lines where there are discontinuities. Dirac distributions are not functions and it does not make sense to see if the are square-integrable or not. Therefore, if there are discontinuities, the function fails to have a square-integrable gradient. □

2.4 Dirichlet nodes

So far we have taken into account the discrete version of the domain Ω but not the partition of its boundary Γ into Dirichlet and Neumann sides. We first need some terminology. A Dirichlet edge is an edge of a triangle that lies on Γ_D. Similarly a **Neumann edge** is an edge of a triangle that is contained in Γ_N. The vertices of the Dirichlet edges are called **Dirichlet nodes**. The doubt may arise in transitions from the Dirichlet to the Neumann part of the boundary. If a node belongs to both Γ_N and Γ_D, it is a Dirichlet node.

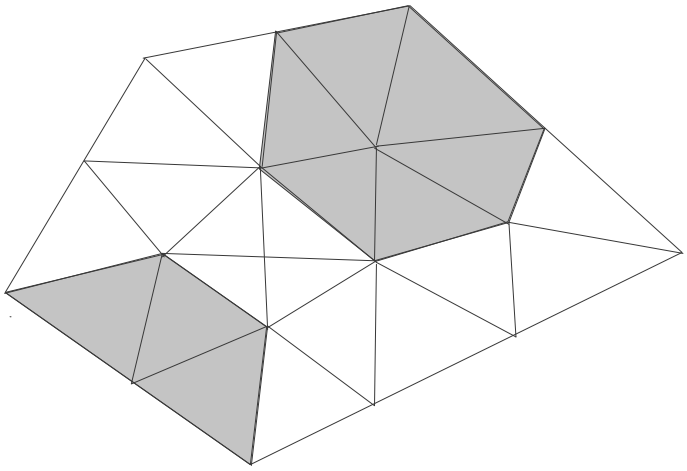


Figure 1.9: Supports of two nodal basis functions.

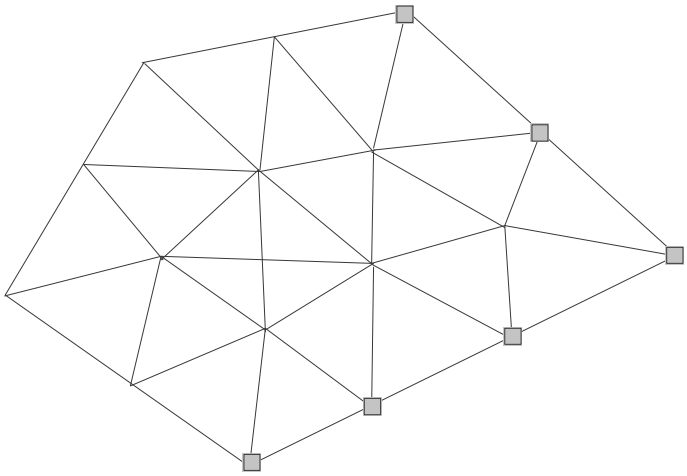


Figure 1.10: Dirichlet nodes corresponding to the domain as depicted in Figure 1.1

In truth, in parallel to what happens with how the Dirichlet and Neumann boundary conditions are treated in the weak formulation, we will inherit two different discrete entities:

- Dirichlet nodes, and
- Neumann edges.

Let us now recall the space

$$H^1_{\Gamma_D}(\Omega) = \{v \in H^1(\Omega) \,:\, v = 0 \quad \text{on } \Gamma_D\}.$$

We might be interested in the space

$$V_h^{\Gamma_D} = V_h \cap H^1_{\Gamma_D}(\Omega) = \{v_h \in V_h \,:\, v_h = 0, \quad \text{on } \Gamma_D\}.$$

Recall now the demand we placed on the triangulation to respect the partition of Γ into Dirichlet and Neumann parts. Because of this, $v_h \in V_h$ vanishes on Γ_D if and only if it vanishes on the Dirichlet edges. Again, since values of piecewise linear functions on edges are determined by the values on the corresponding vertices, we have

$v_h \in V_h$ vanishes on Γ_D if and only if it vanishes on all Dirichlet nodes.

The good news is the fact that we can easily construct a basis of $V_h^{\Gamma_D}$. We simply eliminate the elements of the nodal basis corresponding to Dirichlet nodes. To see that recall that when we write $v_h \in V_h$ as a linear combination of elements of the nodal basis, what we have is actually

$$v_h = \sum_{j=1}^{N} v_h(\mathbf{p}_j)\varphi_j.$$

Therefore $v_h = 0$ on Γ_D if and only if the coefficients corresponding to nodal functions of Dirichlet nodes vanish. To write this more efficiently we will employ two lists, Dir and Ind (as in *independent* or free nodes), to number separately Dirichlet and non-Dirichlet (independent/free) nodes. It is not necessary to number first one type of nodes and then the other, although sometimes it helps to visualize things to assume that we first numbered the free nodes and then the Dirichlet nodes.[2] With our model triangulation numbered as in Figure 1.7 and with the Dirichlet nodes marked in 1.10, the lists are

$$\begin{aligned} \text{Dir} &= \{9, 13, 14, 15, 17, 18\}, \\ \text{Ind} &= \{1, 2, 3, 4, 5, 6, 7, 8, 10, 11, 12, 16\}. \end{aligned}$$

With these lists, an element of V_h can be written as

$$u_h = \sum_{j\in\text{Ind}} u_j\varphi_j + \sum_{j\in\text{Dir}} u_j\varphi_j, \qquad u_j = u_h(\mathbf{p}_j)$$

and an element of $V_h^{\Gamma_D}$ has the form

$$v_h = \sum_{j\in\text{Ind}} v_j\varphi_j.$$

Finally, this proves that

$$\dim V_h^{\Gamma_D} = \#\text{Ind} = \#\{\text{nodes}\} - \#\{\text{Dirichlet nodes}\}.$$

[2]The reason for not doing this is merely practical. The triangulation is built without taking into account which parts of the boundary are Dirichlet and which are Neumann. As we will see in the next Lesson, the numbering of the nodes is inherent to the way the triangulation is given. In many practical problems we play with the boundary conditions for the same domain and it is not convenient to renumber the vertices each time.

3 The finite element method

3.1 The discrete variational problem

After almost fifteen pages of introducing concepts and formulas we can finally arrive at a numerical approximation of our initial problem. Recall that we wrote the problem in the following form

$$\left[\begin{array}{l} \text{find } u \in H^1(\Omega) \text{ such that} \\ u = g_0 \quad \text{on } \Gamma_D, \\ \displaystyle\int_\Omega \nabla u \cdot \nabla v + c \int_\Omega u\, v = \int_\Omega f\, v + \int_{\Gamma_N} g_1\, v \qquad \forall v \in H^1_{\Gamma_D}(\Omega). \end{array}\right.$$

The finite element method (with linear finite elements on triangles) consists of the following discrete version of the preceding weak formulation:

$$\left[\begin{array}{l} \text{find } u_h \in V_h \text{ such that} \\ u_h(\mathbf{p}) = g_0(\mathbf{p}) \quad \text{for every Dirichlet node } \mathbf{p}, \\ \displaystyle\int_\Omega \nabla u_h \cdot \nabla v_h + c \int_\Omega u_h\, v_h = \int_\Omega f\, v_h + \int_{\Gamma_N} g_1\, v_h \qquad \forall v_h \in V_h^{\Gamma_D}. \end{array}\right.$$

As you can easily see we have made three substitutions:

- We look for the unknown in the space V_h instead of on the whole Sobolev space. This means that we have reduced the problem to computing u_h in the vertices of the triangulation (in the nodes) and we are left with a finite number of unknowns.
- We have substituted the Dirichlet condition by fixing the values of the unknowns on Dirichlet nodes. This fact reduces the number of unknowns of the system to the number of free nodes.[3]
- Finally, we have reduced the testing space from $H^1_{\Gamma_D}(\Omega)$ to its discrete subspace $V_h^{\Gamma_D}$. We will show right now that this reduces the infinite number of tests of the weak formulation to a finite number of linear equations.

3.2 The associated system

We write again the discrete problem, specifying the numbering of Dirichlet nodes in the discrete Dirichlet condition:

$$\left[\begin{array}{l} \text{find } u_h \in V_h \text{ such that} \\ u_h(\mathbf{p}_i) = g_0(\mathbf{p}_i) \quad \forall i \in \text{Dir}, \\ \displaystyle\int_\Omega \nabla u_h \cdot \nabla v_h + c \int_\Omega u_h\, v_h = \int_\Omega f\, v_h + \int_{\Gamma_N} g_1\, v_h \qquad \forall v_h \in V_h^{\Gamma_D}. \end{array}\right.$$

[3] This way of substituting the Dirichlet condition by a sort of interpolated Dirichlet condition is neither the only nor the best way of doing this approximation, but it is definitely the simplest, so we will keep it like this for the time being.

Our next claim is the following: the discrete equations

$$\int_\Omega \nabla u_h \cdot \nabla v_h + c \int_\Omega u_h\, v_h = \int_\Omega f\, v_h + \int_{\Gamma_N} g_1\, v_h \qquad \forall v_h \in V_h^{\Gamma_D}$$

are equivalent to the following set of equations

$$\int_\Omega \nabla u_h \cdot \nabla \varphi_i + c \int_\Omega u_h\, \varphi_i = \int_\Omega f\, \varphi_i + \int_{\Gamma_N} g_1\, \varphi_i \qquad \forall i \in \text{Ind}.$$

Obviously this second group of equations is a (small) part of the original one: it is enough to take $v_h = \varphi_i \in V_h^{\Gamma_D}$. However, because of the linearity of the first expression in v_h, if we have the second one for all φ_i, we have the equation for all possible linear combinations of these functions, that is for all $v_h \in V_h^{\Gamma_D}$. Summing up, the method is equivalent to this set of N equations to determine the function u_h:

$$\left[\begin{array}{l} \text{find } u_h \in V_h \text{ such that} \\ u_h(\mathbf{p}_i) = g_0(\mathbf{p}_i) \quad \forall i \in \text{Dir}, \\ \displaystyle\int_\Omega \nabla u_h \cdot \nabla \varphi_i + c \int_\Omega u_h\, \varphi_i = \int_\Omega f\, \varphi_i + \int_{\Gamma_N} g_1\, \varphi_i \qquad \forall i \in \text{Ind}. \end{array}\right.$$

In order to arrive at a linear system, we first have to write u_h in terms of the nodal basis functions

$$u_h = \sum_{j \in \text{Ind}} u_j \varphi_j + \sum_{j \in \text{Dir}} u_j \varphi_j.$$

We next substitute the discrete Dirichlet condition in this expression

$$u_h = \sum_{j \in \text{Ind}} u_j \varphi_j + \sum_{j \in \text{Dir}} g_0(\mathbf{p}_j) \varphi_j.$$

Finally we plug this expression into the discrete variational equation

$$\int_\Omega \nabla u_h \cdot \nabla \varphi_i + c \int_\Omega u_h\, \varphi_i = \int_\Omega f\, \varphi_i + \int_{\Gamma_N} g_1\, \varphi_i,$$

apply linearity, noticing that

$$\nabla u_h = \sum_{j \in \text{Ind}} u_j \nabla \varphi_j + \sum_{j \in \text{Dir}} g_0(\mathbf{p}_j) \nabla \varphi_j$$

and move to the right-hand side what we already know (the Dirichlet data)

$$\begin{aligned} \sum_{j \in \text{Ind}} \left(\int_\Omega \nabla \varphi_j \cdot \nabla \varphi_i + c \int_\Omega \varphi_j \varphi_j \right) u_j \;\; &= \;\; \int_\Omega f\, \varphi_i + \int_{\Gamma_N} g_1\, \varphi_i \\ &\quad - \sum_{j \in \text{Dir}} \left(\int_\Omega \nabla \varphi_j \cdot \nabla \varphi_i + c \int_\Omega \varphi_j \varphi_j \right) g_0(\mathbf{p}_j). \end{aligned}$$

This is a linear system with as many equations as unknowns, namely with $\#\text{Ind} = \dim V_h^{\Gamma_D}$ equations and unknowns. The unknowns are in fact the nodal values of u_h on the free (non-Dirichlet) vertices of the triangulation. After solving this linear system, the formula for u_h lets us recover the function everywhere, not only on nodes.

Remark Unlike the finite difference method, the finite element method gives as a result a function defined on the whole domain and not a set of point values. Reconstruction of the function from computed quantities is in the essence of the method and cannot be counted as a posprocessing of nodal values. □

3.3 Mass and stiffness

There are two matrices in the system above. Both of them participate in the final matrix and parts of them go to build the right-hand side. First we have the **stiffness matrix**

$$w_{ij} = \int_\Omega \nabla\varphi_j \cdot \nabla\varphi_i$$

and second the **mass matrix**

$$m_{ij} = \int_\Omega \varphi_j\, \varphi_i.$$

Both matrices are defined for $i, j = 1, \ldots, N$ (although parts of these matrices won't be used). Both matrices are symmetric. The mass matrix $\mathbf{M}$ is positive definite. The stiffness matrix is positive semidefinite and in fact almost positive definite: if we take an index i and erase the $i-$th row and the $i-$th column of $\mathbf{W}$, the resulting matrix is positive definite.

The system can be easily written in terms of these matrices, using the vector

$$b_i = \int_\Omega f\, \varphi_i + \int_{\Gamma_N} g_1\, \varphi_i, \qquad i \in \mathrm{Ind},$$

to obtain

$$\sum_{j\in\mathrm{Ind}} \big(w_{ij} + c\, m_{ij}\big) u_j = b_i - \sum_{j\in\mathrm{Dir}} \big(w_{ij} + c\, m_{ij}\big) g_0(\mathbf{p}_j), \qquad i \in \mathrm{Ind}.$$

This is clearly a square symmetric linear system. If $c = 0$ (then the original equation is the Poisson equation $-\Delta u = f$ and no reaction term appears), only the stiffness matrix appears. Therefore, stiffness comes from diffusion. Likewise mass proceeds from reaction.

The matrix is positive definite except in one special situation: when $c = 0$ and there are no Dirichlet conditions (i.e., $\Gamma_D = \varnothing$, i.e., $\mathrm{Ind} = \{1, \ldots, N\}$ and $V_h^{\Gamma_D} = V_h$). For the pure Neumann problem for the Laplace operator there are some minor solvability issues similar to the occurrence of rigid motions in mechanical problems. Let us ignore this minor complication for now.

Now look again at the figure showing the supports of nodal basis functions (we copy it right here for convenience) and look at the mass matrix

$$m_{ij} = \int_\Omega \varphi_j\, \varphi_i.$$

If the supports of φ_i and φ_j have no intersecting area, the integral defining m_{ij} vanishes. In fact, since the product of φ_i and φ_j is a non-negative function, $m_{ij} = 0$ if and only if

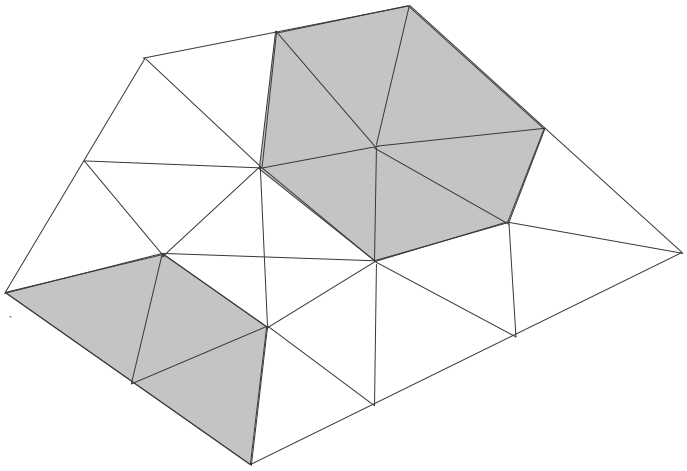


Figure 1.11: Supports of two nodal basis functions

the area of the intersection of the supports is zero[4]. This happens whenever $\mathbf{p}_i$ and $\mathbf{p}_j$ are not vertices of the same triangle.

We say that two nodes are **adjacent** *if they belong to the same triangle.*

In the case of the stiffness matrix we have a similar (maybe weaker result): if the nodes i and j are not adjacent, then $w_{ij} = 0$.

This fact makes the mass and stiffness matrices display a great sparsity character. Given a row i, there are only non-zero entries on positions related to nodes that are adjacent to the $i-$th node.

Going back to the system

$$\sum_{j\in\text{Ind}} \big(w_{ij} + c\, m_{ij}\big) u_j = b_i - \sum_{j\in\text{Dir}} \big(w_{ij} + c\, m_{ij}\big) g_0(\mathbf{p}_j), \qquad i \in \text{Ind},$$

let us remark some simple facts:

- As written now, all data appear in the right-hand side of the system (Neumann data and source terms are in the vector $\mathbf{b}$, Dirichlet data appear multiplying columns of the stiffness-plus-mass matrix).

- Of the full matrices $\mathbf{W}$ and $\mathbf{M}$ we discard rows corresponding to Dirichlet nodes (Dir indices), since no testing is done with the corresponding basis functions. The columns corresponding to these indices are not eliminated though: they are sent to the right-hand side multiplied by the values of the unknown in the Dirichlet nodes, which are known.

[4]By definition the support of a function includes the boundary of the set where the function is non-zero. Therefore, it is possible that the intersection is one edge. The integral is still zero.

4 Exercises

1. **Third type of boundary condition.** Let us consider our usual polygon Ω and the boundary value problem

$$\left[\begin{array}{ll} -\Delta u + u = f & \text{in } \Omega, \\ \partial_n u + k\, u = g & \text{on } \Gamma. \end{array}\right.$$

   Here k is a positive parameter. This type of boundary condition is usually called a boundary condition of the third kind (first being Dirichlet and second Neumann) or a Robin (or Fourier) boundary condition.

   (a) Write down the weak formulation for this problem. Note that the condition is natural and there will not be essential boundary condition in the resulting formulation.

   (b) Write down in detail (as in Sections 3.2/ 3.3) the linear system that has to be solved when we apply the finite element method to this problem. Check that there is a new matrix that can be seen as a boundary-mass matrix. How many non-zero entries does each row of this new matrix have?

   If we take ε very small and the following slightly modified version of the boundary condition

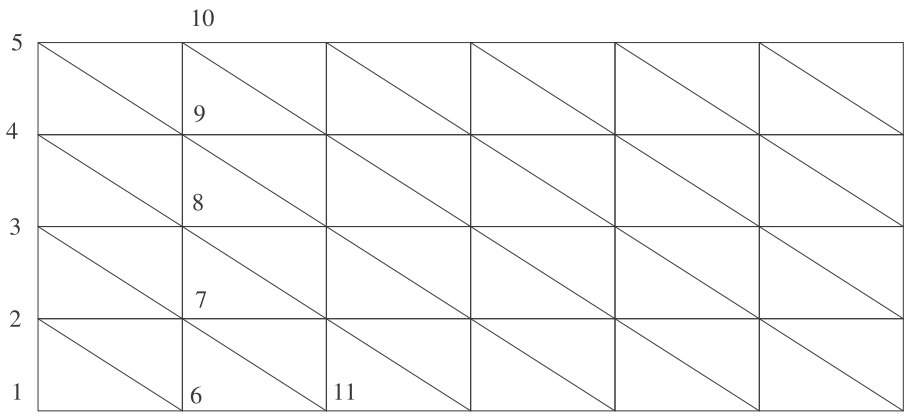
$$\varepsilon \partial_n u + u = g_0, \qquad \text{on } \Gamma$$

   (take $k = \varepsilon^{-1}$ and $g = \varepsilon^{-1} g_0$), we are enforcing the Dirichlet condition in an approximate way. This is done in some commercial and open-source packages.

2. **A particular domain.** Consider the boundary problem of Section 1 on the domain given in the next figure and the following specification for Γ_N and Γ_N

   the left and upper sides have Dirichlet conditions

   and where numbering is done as shown. Let $\mathbf{A} = \mathbf{W} + \mathbf{M}$ be the matrix associated to the system obtained by discretizing with the $\mathbb{P}_1$ finite element method



(a) Write the index sets Dir and Ind.

(b) Write which elements of the 12th row of $\mathbf{A}$ are non-zero.

(c) Identify on the figure the support of the nodal basis function φ_{13}.

(d) What's the size of the system that has to be solved?

(e) We call the profile of the matrix $\mathbf{A}$ to the following vector:

$$m(i) = \inf\{j \,:\, a_{ij} \neq 0\}, \qquad i = 1, \ldots, \#\{\text{nodos}\}$$

that is, $m(i)$ indicates the column number where the first non-zero entry of the ith row is. Compute the profile of $\mathbf{W} + \mathbf{M}$ (without eliminating Dirichlet rows and columns). Draw the form of the matrix using the profile.

(f) In the preceding graph mark which rows and columns will be modified by introduction of Dirichlet conditions. Compute the profile of the reduced matrix (without Dirichlet rows and columns).

(g) What happens if we number nodes horizontally?

Lesson 2

Theoretical and practical notions

1 Assembly

The first lesson left us with a linear system to solve in order to approximate the boundary value problem with the finite element method. There is however the trick question on how to compute all the integrals that appear in the matrix and right-hand side of the system. This is done by a clever process called **assembly** of the system, another of the many good deeds of the finite element method that has made it so extremely popular (as in popular among scientists and engineers, of course) in the last decades.

At this moment we need the polygonal domain Ω and:

- a triangulation $\mathcal{T}_h$,
- a numbering of the nodes $\{\mathbf{p}_i\}$ (nodes are the vertices of the triangles),
- the set of the nodal basis functions $\{\varphi_i\}$.

In this section, nNod will be the global number of nodes.

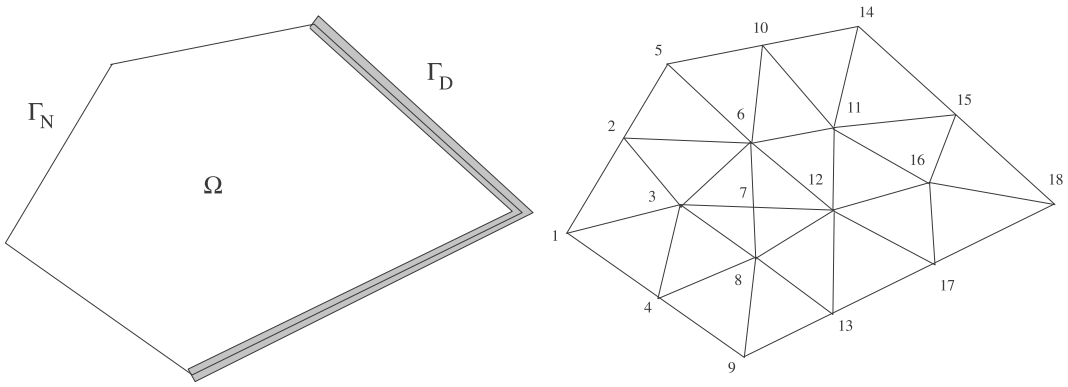


Figure 2.1: Geometry of the problem and triangulation

1.1 The mass and stiffness matrices

We are going to center our attention in the efficient construction of the stiffness matrix

$$w_{ij} = \int_\Omega \nabla\varphi_j \cdot \nabla\varphi_i$$

and of the mass matrix

$$m_{ij} = \int_\Omega \varphi_j \, \varphi_i.$$

Integrals over Ω can be decomposed as the sum of integrals over the different triangles

$$w_{ij} = \int_\Omega \nabla\varphi_j \cdot \nabla\varphi_i = \sum_K \int_K \nabla\varphi_j \cdot \nabla\varphi_i = \sum_K w_{ij}^K.$$

On each triangle we are going to define three local nodal basis functions. First assign a number to each of the three vertices of a triangle K:

$$\mathbf{p}_1^K, \quad \mathbf{p}_2^K, \quad \mathbf{p}_3^K.$$

Then consider the functions

$$N_1^K, \quad N_2^K, \quad N_3^K \quad \in \mathbb{P}_1$$

that satisfy

$$N_\alpha^K(\mathbf{p}_\beta^K) = \delta_{\alpha\beta}, \qquad \alpha, \beta = 1, 2, 3.$$

It is simple to see that the nodal basis function φ_i restricted to the triangle K is either zero (this happens when $\mathbf{p}_i$ is not one of the three vertices of K) or one of the N_α^K functions. More precisely, let n_α be the global number of the local node with number α in the triangle K. This means that

$$N_\alpha^K = \varphi_{n_\alpha}, \qquad \text{on the triangle } K.$$

We can now compute the 3×3 matrix $\mathbf{K}^K$

$$k_{\alpha\beta}^K = \int_K \nabla N_\beta^K \cdot \nabla N_\alpha^K, \qquad \alpha, \beta = 1, 2, 3.$$

This is due to be simple, since the functions N_α^K are polynomials of degree one (unlike the functions φ_i that are only piecewise polynomials). Later on, we will see strategies to do this computation. Note at this moment that computation of this matrix depends only on the triangle K and does not take into account any other element of the triangulation.

Therefore

$$k_{\alpha\beta}^K = w_{n_\alpha n_\beta}^K$$

All other elements of the matrix $\mathbf{W}^K$ are zero. Recall again that $\mathbf{W}^K$ is a nNon $\times$ nNod matrix and that

$$\mathbf{W} = \sum_K \mathbf{W}^K.$$

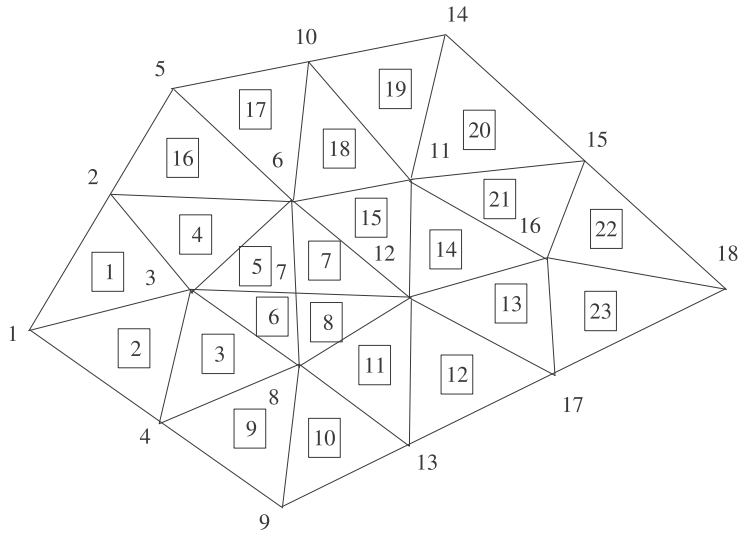
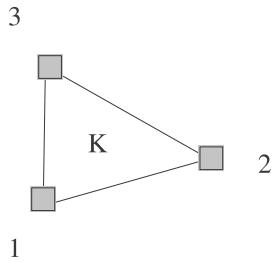


Figure 2.2: A numbering of the triangles.



local		global
1	$\leftrightarrow$	12
2	$\leftrightarrow$	16
3	$\leftrightarrow$	11

Figure 2.3: The 14th triangle and their vertex numberings.

The assembly process requires then a given numbering of triangles as shown in Figure 2.2. The order of this numbering is only used to do the computations but does not modify the shape of the final result.

The process to assemble the mass matrix is the same. Effective assembly of the mass and stiffness matrices can be done at the same time. Instead of computing separately the matrix $\mathbf{K}^K$ and a similar one for the mass matrix, we can directly try to compute the 3×3 matrix with elements

$$\int_K \nabla N_\beta^K \cdot \nabla N_\alpha^K + c \int_K N_\beta^K \, N_\alpha^K, \qquad \alpha, \beta = 1, 2, 3.$$

1.2 The reference element

To compute the elements

$$\int_K \nabla N_\beta^K \cdot \nabla N_\alpha^K \qquad \text{and} \qquad \int_K N_\beta^K \, N_\alpha^K$$

we need: (a) either and effective way of evaluating the functions N_α^K and their gradients; (b) or a closed form for the resulting integrals. Both possibilities are done usually by moving to the so-called reference element.

For triangles, the reference element is the triangle with vertices

$$\widehat{\mathbf{p}}_1 = (0,0), \qquad \widehat{\mathbf{p}}_2 = (1,0), \qquad \widehat{\mathbf{p}}_3 = (0,1).$$

To distinguish variables in the reference element and in a general triangle (in this context

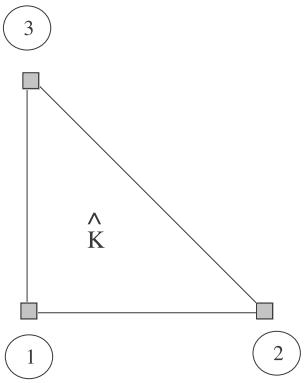


Figure 2.4: The reference element

we say a physical element) it is customary to use the variables (ξ, η) in the reference element and (x, y) in the physical element. In the mathematical literature for FEM it is also usual to put a hat on the name of the variables in the reference element, so that $(\hat{x}, \hat{y})$ would be used to denote coordinates in the reference configuration.

An unimportant detail. Some people prefer to use a different reference triangle, with the same shape but with vertices on $(-1,-1)$, $(1,-1)$ and $(-1,1)$. Some details of the forthcoming computations have to be adapted if this choice is taken. □

The local nodal functions in the reference triangles are three $\mathbb{P}_1$ functions satisfying

$$\widehat{N}_\alpha(\widehat{\mathbf{p}}_\beta) = \delta_{\alpha\beta}, \qquad \alpha, \beta = 1, 2, 3.$$

These functions are precisely

$$\widehat{N}_1 = 1 - \xi - \eta, \qquad \widehat{N}_2 = \xi, \qquad \widehat{N}_3 = \eta$$

or, if you prefer hatting variables (this is the last time we will write both expressions)

$$\widehat{N}_1 = 1 - \widehat{x} - \widehat{y}, \qquad \widehat{N}_2 = \widehat{x}, \qquad \widehat{N}_3 = \widehat{y}$$

Let us now take the three vertices of a triangle K

$$\mathbf{p}_1^K = (x_1, y_1), \qquad \mathbf{p}_2^K = (x_2, y_2), \qquad \mathbf{p}_3^K = (x_3, y_3).$$

The following affine transformation[1]

$$\begin{aligned} \begin{bmatrix} x \\ y \end{bmatrix} &= \underbrace{\begin{bmatrix} x_2 - x_1 & x_3 - x_1 \\ y_2 - y_1 & y_3 - y_1 \end{bmatrix}}_{\mathbf{B}_K} \begin{bmatrix} \xi \\ \eta \end{bmatrix} + \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} \\ &= \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} (1 - \xi - \eta) + \begin{bmatrix} x_2 \\ y_2 \end{bmatrix} \xi + \begin{bmatrix} x_3 \\ y_3 \end{bmatrix} \eta \end{aligned}$$

maps the triangle $\widehat{K}$ bijectively into K. In fact, if we call this transformation F_K, then

$$F_K(\widehat{\mathbf{p}}_\alpha) = \mathbf{p}_\alpha^K, \qquad \alpha = 1, 2, 3.$$

Notice that the second expression we have written for the transformation gives it in terms of the nodal basis functions in the reference domain. You can think of it as a coincidence. In a way it is: the coincidence stems from the fact that the type of functions we are using for finite elements is the same as the functions needed to transform linearly triangles in the plane.

It is simple now to prove that

$$\widehat{N}_\alpha = N_\alpha^K \circ F_K, \qquad \alpha = 1, 2, 3,$$

or, what is the same

$$N_\alpha^K = \widehat{N}_\alpha \circ F_K^{-1}, \qquad \alpha = 1, 2, 3.$$

The $\circ$ symbol is used for composition. In the last expression, what we have is

$$N_\alpha^K(x, y) = \widehat{N}_\alpha(F_K^{-1}(x, y)).$$

Since computing F_K^{-1} is straightforward from the explicit expression for F_K, this formula gives a simple way of evaluating the functions N_α^K. The fact of representing the local basis for the physical in terms of the basis in the reference configuration, $N_\alpha^K = \widehat{N}_\alpha \circ F_K^{-1}$, is referred to as *pushing forward the basis* on the reference element[2].

To evaluate the gradient of N_α^K we have to be more careful, since we have to apply the chain rule. Let us denote briefly gradients as

$$\nabla = \begin{bmatrix} \partial_x \\ \partial_y \end{bmatrix}, \qquad \widehat{\nabla} = \begin{bmatrix} \partial_\xi \\ \partial_\eta \end{bmatrix}.$$

(Note that we are writing gradients as column vectors.) The following formula is the result of applying the chain rule

$$\mathbf{B}_K^\top \big(\nabla \phi \circ F_K\big) = \widehat{\nabla}(\phi \circ F_K).$$

[1]Many mesh generators prepare number triangle locally by ordering nodes counterclockwise. This makes $\det \mathbf{B}_K > 0$.

[2]The opposite process, bringing something from the physical element to the reference one, is called *pull-back*.

$\mathbf{B}_K^\top$ is the transpose of the matrix of the linear transformation F_K. Taking $\phi = N_\alpha^K$ in this expression and moving things a little, we obtain a formula for the gradient of the local basis functions

$$\nabla N_\alpha^K = \mathbf{B}_K^{-\top}\big((\widehat{\nabla}\widehat{N}_\alpha) \circ F_K^{-1}\big).$$

The expression may look complicated but it is very simple to use. If we want to compute the value of the gradient of N_α^K at a point $(x, y) \in K$, we first compute the transformed point $(\xi, \eta) = F_K^{-1}(x, y)$ in the reference triangle, evaluate the gradient of $\widehat{N}_\alpha$ at this point and then multiply it by the matrix $\mathbf{B}_K^{-\top}$, which is the transpose of the inverse of $\mathbf{B}_K$, i.e.,

$$\mathbf{B}_K^{-\top} = \frac{1}{\det \mathbf{B}_K} \begin{bmatrix} y_3 - y_1 & -(y_2 - y_1) \\ -(x_2 - x_1) & x_2 - x_1 \end{bmatrix}$$

with

$$\det \mathbf{B}_K = (x_2 - x_1)(y_3 - y_1) - (y_2 - y_1)(x_3 - x_1)$$

(remember that $|\det \mathbf{B}_K| = 2 \operatorname{area} K$). In fact, for this very elementary method, the gradients of the three basis functions on the reference element are constant vectors

$$\widehat{\nabla}\widehat{N}_1 = \begin{bmatrix} -1 \\ -1 \end{bmatrix}, \qquad \widehat{\nabla}\widehat{N}_2 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \qquad \widehat{\nabla}\widehat{N}_3 = \begin{bmatrix} 0 \\ 1 \end{bmatrix},$$

so computation of the constant vectors ∇N_α^K is very simple, and we don't even have to use the inverse transformation F_K^{-1} for the gradients. We do, however, to evaluate N_α^K.

1.3 Computing with quadrature rules

Depending on the complications of the problem (we are dealing with a very simple model problem), all the computations can be carried out to the reference element or we can try to do things directly on the physical triangle K. Let us mention here two popular quadrature rules for triangles: the three point rule with the vertices

$$\int_K \phi \approx \frac{\operatorname{area} K}{3}\Big(\phi(\mathbf{p}_1^K) + \phi(\mathbf{p}_2^K) + \phi(\mathbf{p}_3^K)\Big)$$

and the midpoints approximation

$$\int_K \phi \approx \frac{\operatorname{area} K}{3}\Big(\phi(\mathbf{m}_1^K) + \phi(\mathbf{m}_2^K) + \phi(\mathbf{m}_3^K)\Big),$$

where $\mathbf{m}_\alpha^K$ are the midpoints of the edges of K. If ϕ is a polynomial of degree one, the first formula gives the exact value. The second formula is even better: if ϕ is a polynomial of degree two, the edge-midpoints formula is exact.

In the very simple case of $\mathbb{P}_1$ elements, we have ∇N_α^K constant and therefore

$$\int_K \nabla N_\beta^K \cdot \nabla N_\alpha^K = (\operatorname{area} K)\, \nabla N_\beta^K \cdot \nabla N_\alpha^K,$$

and this computation is very simple. For the mass matrix, we note that $N_\beta^K\, N_\alpha^K$ is a polynomial of degree two and therefore, the edge-midpoints formula gives the exact value of the integrals

$$\int_K N_\beta^K\, N_\alpha^K$$

with just three evaluations of the functions.

1.4 Doing everything on the reference element

This section gives another idea on how to compute the local mass and stiffness matrices. You can skip it without losing continuity and go to Section 1.5. The change of variables applied to the integral of the local mass matrix gives

$$\int_K N_\beta^K \, N_\alpha^K = |\det \mathbf{B}_K| \int_{\widehat{K}} \widehat{N}_\beta \widehat{N}_\alpha.$$

Therefore everything is done once we have the 3×3 matrix

$$\widehat{\mathbf{K}}_0 = \left[\int_{\widehat{K}} \widehat{N}_\beta \widehat{N}_\alpha \right]_{\alpha,\beta} = \tfrac{1}{24} \begin{bmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{bmatrix}$$

For derivatives, we have to be more careful

$$\begin{aligned} \int_K \nabla N_\beta^K \cdot \nabla N_\alpha^K &= |\det \mathbf{B}_K| \int_{\widehat{K}} \left(\nabla N_\beta^K \circ F_K \right) \cdot \left(\nabla N_\alpha^K \circ F_K \right) = \\ &= |\det \mathbf{B}_K| \int_{\widehat{K}} \left(\mathbf{B}_K^{-\top} \widehat{\nabla} \widehat{N}_\beta \right) \cdot \left(\mathbf{B}_K^{-\top} \widehat{\nabla} \widehat{N}_\alpha \right) = \\ &= |\det \mathbf{B}_K| \int_{\widehat{K}} \mathbf{C}_K \widehat{\nabla} \widehat{N}_\beta \cdot \widehat{\nabla} \widehat{N}_\alpha \end{aligned}$$

where

$$\mathbf{C}_K = \mathbf{B}_K^{-1} \mathbf{B}_K^{-\top} = \begin{bmatrix} c_{11}^K & c_{12}^K \\ c_{12}^K & c_{22}^K \end{bmatrix}$$

is a symmetric 2×2 matrix that depends only on the triangle. If we compute the following 3×3 matrices in the reference element

$$\begin{aligned} \widehat{\mathbf{K}}_{\xi\xi} &= \left[\int_{\widehat{K}} \partial_\xi \widehat{N}_\beta \, \partial_\xi \widehat{N}_\alpha \right]_{\alpha,\beta} = \tfrac{1}{2} \begin{bmatrix} 1 & -1 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \\ \widehat{\mathbf{K}}_{\eta\eta} &= \left[\int_{\widehat{K}} \partial_\eta \widehat{N}_\beta \, \partial_\eta \widehat{N}_\alpha \right]_{\alpha,\beta} = \tfrac{1}{2} \begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix} \\ \widehat{\mathbf{K}}_{\xi\eta} &= \left[\int_{\widehat{K}} \partial_\xi \widehat{N}_\beta \, \partial_\eta \widehat{N}_\alpha \right]_{\alpha,\beta} = \tfrac{1}{2} \begin{bmatrix} 1 & 0 & -1 \\ -1 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \end{aligned}$$

we have

$$\left[\int_K \nabla N_\beta^K \cdot \nabla N_\alpha^K \right]_{\alpha,\beta} = |\det \mathbf{B}_K| \left(c_{11}^K \widehat{\mathbf{K}}_{\xi\xi} + c_{22}^K \widehat{\mathbf{K}}_{\eta\eta} + c_{12}^K (\widehat{\mathbf{K}}_{\xi\eta} + \widehat{\mathbf{K}}_{\xi\eta}^\top) \right).$$

1.5 Right-hand sides

Construction of the right-hand side of the linear system requires the computation of two vectors:

$$\int_\Omega f\,\varphi_i, \qquad \int_{\Gamma_N} g_1\,\varphi_i.$$

In principle, this has to be done for indices of free nodes ($i \in \mathrm{Ind}$), but in practice what is done is to compute them for all i and then discard the elements corresponding to Dirichlet nodes.

The surface forces (source terms) can be treated in a similar way to the stiffness and mass matrices:

$$\int_\Omega f\,\varphi_i = \sum_K \int_K f\,\varphi_i.$$

For each triangle K we compute the vector

$$\int_K f\,N_\alpha^K, \qquad \alpha = 1,2,3$$

and then add these elements in the positions (n_1, n_2, n_3) of the full vector. This process can be done at the same time as the matrix assembly, since it goes triangle by triangle. For the $\mathbb{P}_1$ element, the following extremely simple approximation is enough:

$$\begin{aligned}\int_K f\,N_\alpha^K &\approx \tfrac{1}{3}\sum_{\beta=1}^3 f(\mathbf{p}_\beta^K)\int_K N_\alpha^K = \frac{|\det \mathbf{B}_K|}{3}\sum_{\beta=1}^3 f(\mathbf{p}_\beta^K)\int_{\widehat{K}} \widehat{N}_\alpha \\ &= \frac{|\det \mathbf{B}_K|}{18}\sum_{\beta=1}^3 f(\mathbf{p}_\beta^K).\end{aligned}$$

A simpler options is

$$\int_K f\,N_\alpha^K \approx f(\mathbf{b}^K)\int_K N_\alpha^K = f(\mathbf{b}^K)\frac{|\det \mathbf{B}_K|}{6},$$

where

$$\mathbf{b}^k = \frac{1}{3}(\mathbf{p}_1^K + \mathbf{p}_2^K + \mathbf{p}_3^K)$$

is the barycenter of K. (This second option is wiser when f has discontinuities that are captured by the triangulation, that is, when f is allowed to have jumps across element interfaces.)

The three integrals related to the element K are approximated by the same number. We have actually approximated f by a function that is constant over each triangle: the constant value on the triangle is the average of the values on its vertices (or its value at the barycenter). Otherwise, we can try a quadrature rule to approximate the integrals. It is important at this stage to note that the choice of an adequate quadrature rule has to take into account two facts:

- it has to be precise enough not to lose the good properties of the finite element method, but

- it has to be simple enough not to be wasting efforts in computing with high precision a quantity that is only needed with some precision.

In principle, we could think of using a very precise rule to compute the integrals as exactly as possible. This is overdoing it and forgetting one of the most important principles of well-understood scientific computing: errors from different sources have to be balanced. It doesn't make much sense to spend time in computing exactly a quantity when that number is to be used in the middle of many approximate computations.

The presence of Neumann boundary conditions imposes the computation of the following integrals

$$\int_{\Gamma_N} g_1 \, \varphi_i.$$

This process is made separately to the ones of computing domain integrals for the matrices and the source terms. First of all we have to decompose the Neumann boundary in the set of edges that lie on it (for that we will need a numbering of the Neumann edges):

$$\int_{\Gamma_N} g_1 \, \varphi_i = \sum_L \int_L g_1 \, \varphi_i.$$

Note first that unless $\mathbf{p}_i$ is on the Neumann boundary, this integral vanishes.

Next, for each edge consider the two vertices that delimit it: $\mathbf{p}_1^L$ and $\mathbf{p}_2^L$. As we had with triangular elements, we will need the relation between the extremal points of each Neumann edge and the global numbering. If

$$\mathbf{p}_1^L = (x_1, y_1), \qquad \mathbf{p}_2^L = (x_2, y_2),$$

the function

$$[0,1] \ni t \longmapsto \phi_L(t) = (1-t) \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} + t \begin{bmatrix} x_2 \\ y_2 \end{bmatrix}$$

is a parameterization of the segment L. We now consider the following two functions

$$\psi_1 = 1 - t, \qquad \psi_2 = t.$$

They are just the nodal basis functions on the reference element $[0,1]$ for the space of linear polynomials in one dimension. It is simple to see that

$$(\varphi_i \circ \phi_L)(t) = \begin{cases} \psi_1(t), & \text{if } \mathbf{p}_i = \mathbf{p}_1^L, \\ \psi_2(t), & \text{if } \mathbf{p}_i = \mathbf{p}_2^L, \\ 0, & \text{otherwise.} \end{cases}$$

The integrals to be computed are

$$\int_L g_1 \varphi_{n_\alpha} = \text{length}\, L \int_0^1 (g_1 \circ \phi_L)(t) \psi_\alpha(t) \mathrm{d}t, \qquad \alpha = 1, 2$$

(as before n_α denotes the global index for the local node α). We can the use numerical quadrature for this line integral. Alternatively we can approximate

$$\int_L g_1 \, \varphi_{n_\alpha} \approx g_1(\mathbf{m}^L) \int_L \varphi_i = \frac{\text{length}\, L}{2} g_1(\mathbf{m}^L), \qquad \alpha = 1, 2,$$

where $\mathbf{m}^L = \frac{1}{2}(\mathbf{p}_1^L + \mathbf{p}_2^L)$ is the midpoint of L.

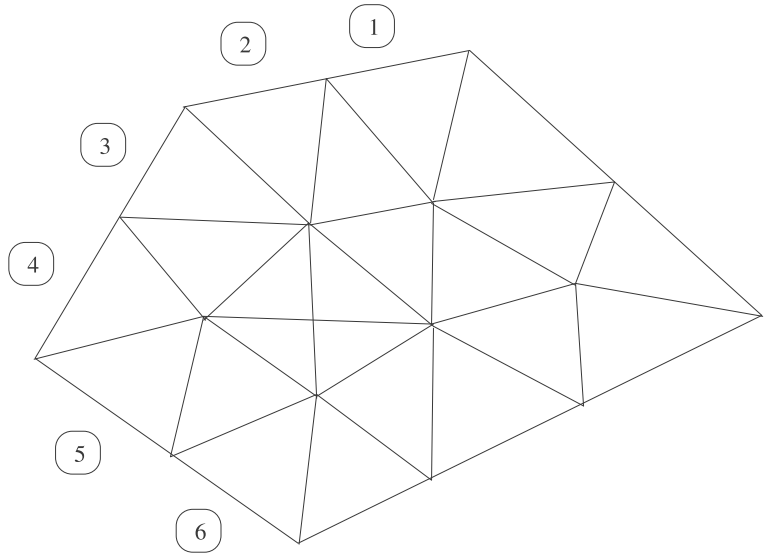
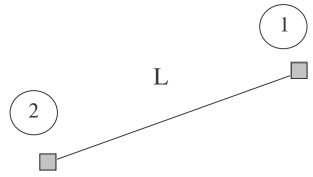


Figure 2.5: A numbering of Neumann edges/elements.



local		global
1	$\leftrightarrow$	10
2	$\leftrightarrow$	5

Figure 2.6: The 2nd Neumann edge and its numberings. For this edge, $n_1 = 10$ and $n_2 = 5$. It is common to number boundary edges positively from the point of view of the interior domain, that is, when going from the first node to the second, we leave the interior domain to the left.

2 A taste of the theory

2.1 Abstract frame

Because many of the ideas that we will develop on and on in this course are quite independent from the particular problem, let us rewrite everything in a slightly more abstract language. We have two spaces,

$$V = H^1(\Omega) \quad \text{and} \quad V_0 = H^1_{\Gamma_D}(\Omega),$$

a bilinear form (related only to the partial differential operator)

$$a(u, v) = \int_\Omega \nabla u \cdot \nabla v + c \int_\Omega u\, v$$

and a linear form where most of the data appear

$$\ell(v) = \int_\Omega f\, v + \int_{\Gamma_N} g_1\, v.$$

Finally there is a linear operator γ that serves us to describe the essential conditions: for us γu is the value of u on the boundary Γ_D. Notice that

$$V_0 = \{v \in V \,:\, \gamma v = 0\}.$$

The problem admits then this simple form

$$\left[\begin{array}{l} \text{find } u \in V \text{ such that} \\ \gamma u = g_0, \\ a(u,v) = \ell(v) \quad \forall v \in V_0 \end{array}\right. .$$

Only when $g_0 = 0$ (or when there's no Γ_D and the whole boundary is a Neumann boundary), the problem reduces to an even simpler one

$$\left[\begin{array}{l} \text{find } u \in V_0 \text{ such that} \\ a(u,v) = \ell(v) \quad \forall v \in V_0. \end{array}\right.$$

Therefore, when the essential condition is homogeneous (or when there is no essential condition), the set where we look for u and the test space are the same. In other cases, the restriction imposed to the tests v is the homogeneous version of the essential condition.

2.2 Well-posedness

Let us recall that the natural norm in our space $V = H^1(\Omega)$ was

$$\|u\| = \|u\|_{1,\Omega} = \left(\int_\Omega |\nabla u|^2 + \int_\Omega |u|^2\right)^{1/2}.$$

There are several conditions that ensure the well-posedness of the problem

$$\left[\begin{array}{l} \text{find } u \in V \text{ such that} \\ \gamma u = g_0, \\ a(u,v) = \ell(v) \quad \forall v \in V_0, \end{array}\right.$$

or of its homogeneous version ($g_0 = 0$)

$$\left[\begin{array}{l} \text{find } u \in V_0 \text{ such that} \\ a(u,v) = \ell(v) \quad \forall v \in V_0. \end{array}\right.$$

Well-posedness means existence and uniqueness of solution and continuity of the solution with respect to the data.

Let us first list the properties that are satisfied in all the situations we are addressing in this course:

- V is a Hilbert space (a vector space, with an inner product so that the space is complete with respect to the associate norm)[3],
- V_0 is a closed subspace of V,
- the bilinear form a is continuous in V, that is, there exists $M > 0$ such that

$$|a(u, v)| \leq M\|u\| \, \|v\|, \qquad \forall u, v \in V,$$

- the linear form ℓ is continuous

$$|\ell(v)| \leq C_\ell \|v\|, \qquad \forall v \in V.$$

As already mentioned, all of these properties are satisfied in our case. In fact

$$C_\ell^2 \leq \int_\Omega |f|^2 + C_\Omega \int_{\Gamma_N} |g_1|^2.$$

There is a last property, called **ellipticity** or **coercivity**, which reads: there exists $\alpha > 0$ such that

$$a(v, v) \geq \alpha \|v\|^2, \qquad \forall v \in V_0.$$

Note that the property is only demanded on the set V_0. In our case it is not satisfied in all situations. In fact, it is satisfied in all but one case:

- if $c > 0$ the property is satisfied with α depending only on c,
- if $c = 0$ and $\operatorname{length} \Gamma_D > 0$, the property is satisfied with α depending on Ω and on the partition of the boundary in Dirichlet and Neumann parts.

If all the properties mentioned above hold, then the problem

$$\left[\begin{array}{l} \text{find } u \in V_0 \text{ such that} \\ a(u, v) = \ell(v) \quad \forall v \in V_0, \end{array} \right.$$

has a unique solution and

$$\|u\| \leq C_\ell / \alpha.$$

If $g_0 \neq 0$ then the problem

$$\left[\begin{array}{l} \text{find } u \in V \text{ such that} \\ \gamma u = g_0, \\ a(u, v) = \ell(v) \quad \forall v \in V_0, \end{array} \right.$$

has a unique solution if there exists a $u_0 \in V$ such that $\gamma u_0 = g_0$. In that case, the continuity of the solution with respect to the data has a more complicated expression

$$\|u\| \leq C_\ell / \alpha + (M/\alpha + 1) \inf \big\{ \|u_0\| \, : \, \gamma u_0 = g \big\}.$$

[3]Maybe this sentence looks too hard. You should know what a vector space and also what an inner (or scalar) product is. When you have an inner product, you have an associated norm and with it a concept of convergence of sequences of elements of V. Completeness is a property that ensures that all Cauchy sequences have a limit. In essence, it means that convergence has to happen inside the space. We cannot have a sequence of elements of V converging to something that is not in V.

Remark. For the pure Neumann problem with $c = 0$

$$\left[\begin{array}{l} -\Delta u = f \quad \text{in } \Omega, \\ \partial_n u = g_1 \quad \text{on } \Gamma, \end{array}\right.$$

we cannot verify the conditions to prove existence and uniqueness. In fact, existence is not guaranteed and we never have uniqueness. First of all, because of the divergence theorem we must have

$$\int_\Omega \Delta u = \int_\Omega \operatorname{div}(\nabla u) = \int_\Gamma \partial_n u$$

and therefore the data have to satisfy the compatibility condition

$$\int_\Omega f + \int_\Gamma g_1 = 0.$$

If this condition is satisfied, there is more that one solution, since constant functions satisfy the problem

$$\left[\begin{array}{l} -\Delta u = 0 \quad \text{in } \Omega, \\ \partial_n u = 0 \quad \text{on } \Gamma. \end{array}\right.$$

□

2.3 Galerkin methods

A Galerkin method for the problem

$$\left[\begin{array}{l} \text{find } u \in V_0 \text{ such that} \\ a(u, v) = \ell(v) \quad \forall v \in V_0, \end{array}\right.$$

consists of the choice of a finite dimensional space

$$V_h^0 \subset V_0$$

and on the consideration of the discrete problem

$$\left[\begin{array}{l} \text{find } u_h \in V_h^0 \text{ such that} \\ a(u_h, v_h) = \ell(v_h) \quad \forall v_h \in V_h^0. \end{array}\right.$$

The $\mathbb{P}_1$ finite element method for the reaction-diffusion problem with homogeneous Dirichlet conditions is therefore an example of Galerkin method[4].

The Galerkin equations are equivalent to a linear system. Let us do here the detailed argument, although you will see that we already did exactly this in Section 3 of the previous lesson.

[4]Galerkin comes from Boris Galerkin. A good pronunciation of the word would be something more like *Galyorkin*, with emphasis on the *lyor* syllable. Most English speakers pronounce it however as if it were an English word.

First we need a basis of V_h^0: $\{\varphi_i \,:\, i \in \text{Ind}\}$. The index set Ind is now anything you want to use in order to number the finite basis of the set. In general we would number form one to the dimension of the space, but in our model problem the numbering proceeds from eliminating some indices from a wider numbering. Then we notice that the abstract set of equations

$$a(u_h, v_h) = \ell(v_h) \quad \forall v_h \in V_h^0$$

is equivalent to

$$a(u_h, \varphi_i) = \ell(\varphi_i) \quad \forall i \in \text{Ind}.$$

Finally, we decompose

$$u_h = \sum_{j \in \text{Ind}} u_j \varphi_j$$

and substitute this expression above to obtain the linear system

$$\sum_{j \in \text{Ind}} a(\varphi_j, \varphi_i) u_j \; = \ell(\varphi_i), \qquad i \in \text{Ind}.$$

There are as many unknowns as there are equations here. In this abstract setting, the values u_j are not nodal values, since an arbitrary basis of a linear space has nothing to do with nodes or evaluations of functions.

If the hypotheses of Section 2.2 hold, this system has a unique solution. Furthermore we have the following result, which is popularly referred to as Céa's Lemma[5]:

$$\|u - u_h\| \leq \frac{M}{\alpha} \inf \left\{ \|u - v_h\| \,:\, v_h \in V_h^0 \right\}.$$

The result might not seem to say much at first sight. There are however some aspects that have to be remarked here:

- The result gives an upper bound of the error between the exact solution u and the approximate solution u_h (the finite element solution) and this error bound is measured in the *energy* norm and not in any other one.

- The term

  $$\inf \left\{ \|u - v_h\| \,:\, v_h \in V_h^0 \right\}$$

  is just an approximation error, completely unrelated to the original problem. It measures how well the (unknown) exact solution can be approximated by elements of the space where we are looking for the solution. Because of how this term is estimated in particular situations (in FEM, for instance) many people call this an interpolation error. We will see a bit of this in the following section. This approximation error is measured also in the energy norm, of course[6].

[5] Céa, as in Jean Céa. French. Do you best with the pronunciation of the name.

[6] There's a well-established tradition to keep the infimium in the right-hand side of Céa's estimate. The infimum is actually a minimum, as guaranteed by elementary functional analysis arguments. Céa's estimate is also called the *quasioptimality* of the Galerkin method.

- The only other constants in the inequality depend on the problem, but not on data. Note however that complicated solutions (solutions that vary a lot, or that have large gradients, or anything you can think of as difficult to grasp with a simple approximation) will not necessarily be approximated as well as simple smooth solutions. Since we do not know the solution (by definition, it is the unknown), how can we have an idea of this error? The answer is the lifetime work of numerical analysts and computational scientists. Just three ideas:
  - for simple smooth solutions, numerical analysis shows usually how error behaves quite precisely, which gives us a hint of the best possible behavior of our method;
  - PDE theory sometimes helps in understanding where things can go wrong and we can do some effort in concentrating approximation in that area;
  - finally, there is a whole new (new as in only thirty years old or so) branch of computational knowledge related to error estimation and adaptivity, allowing you to improve your computations with information you harvest from the already performed computations.

The theoretical frame for the case with non-homogeneous Dirichlet conditions is somewhat more delicate, because we have to go someplace more abstract to write correctly the approximation of the condition

$$u = g_0 \quad \text{on } \Gamma_D$$

by

$$u_h(\mathbf{p}) = g_0(\mathbf{p}) \quad \forall \mathbf{p} \text{ Dirichlet node},$$

without making any use of the particularities of the finite element space $\mathbb{P}_1$. This can be done in several ways, and we are not going to detail them. Particularized to FEM the result will look like this

$$\|u - u_h\| \leq (1 + \frac{M}{\alpha}) \inf \big\{ \|u - v_h\| \,:\, v_h \in V_h, \quad v_h(\mathbf{p}) = g_0(\mathbf{p}) \quad \forall \mathbf{p} \text{ Dirichlet node} \big\}.$$

Note that the approximation error in the right-hand side includes the imposition of the discrete essential boundary condition.

2.4 Convergence of the $\mathbb{P}_1$ finite element method

How does all of this work for the $\mathbb{P}_1$ finite element? Let us go back to the case with homogeneous boundary conditions. As mentioned, the error can be bounded as

$$\|u - u_h\|_{1,\Omega} \leq \frac{M}{\alpha} \inf \big\{ \|u - v_h\|_{1,\Omega} \,:\, v_h \in V_h^0 \big\}.$$

Let us emphasize again that the norm for measuring the error is imposed by the problem (see Section 2.1). Assume now that u is a well-behaved function. For example, that it is continuous. Then we can construct a function $\pi_h u$ by taking nodal values of u on the vertices of the triangulation and creating with them an element of V_h. This

is, obviously, interpolation in V_h, that is, interpolation with continuous piecewise linear functions. Because of the Dirichlet boundary condition u vanishes on Dirichlet nodes, and so does consequently $\pi_h u$. Therefore $\pi_h u \in V_h^0$ and we can use the bound

$$\|u - u_h\|_{1,\Omega} \leq \frac{M}{\alpha}\|u - \pi_h u\|_{1,\Omega}.$$

We have therefore bounded the error of the finite element method by the error of interpolation of the exact solution in the finite element space. A nice thing about this interpolation process is the fact that it is done triangle-by-triangle, so actually, the global error for interpolation is the sum of the errors that we have done element-by-element.

In basic courses on numerical methods you will have seen that it is possible to estimate the error of interpolation without knowing the solution, but that this bound of the error is proportional to some quantity depending on a high order derivative of the function that is interpolated. You will have seen this in one space dimension. In several space dimensions, it is a bit more difficult but not so much. The result is the following: there exists a constant C that depends on the minimum angle of the triangulation such that

$$\|u - \pi_h u\|_{1,\Omega} \leq Ch\Big(\int_\Omega |\partial_{xx}u|^2 + |\partial_{xy}u|^2 + |\partial_{yy}u|^2\Big)^{1/2},$$

where h is the size of the longest edge of the triangulation. The expression on the right-hand side is an example of a Sobolev seminorm. It is denoted usually as

$$|u|_{2,\Omega} = \Big(\int_\Omega |\partial_{xx}u|^2 + |\partial_{xy}u|^2 + |\partial_{yy}u|^2\Big)^{1/2}.$$

The whole bound is

$$\|u - u_h\|_{1,\Omega} \leq C'h|u|_{2,\Omega}$$

with the constant C' depending on the coefficients of the problem, on the geometry of the physical setting and on the smallest angle. If the triangles are very flat (the ratio between the longest edge and the inradius[7] is very small), the constant gets to be very large.

First of all, let us remark that the error bound requires the second derivatives of the solution to be square-integrable, which is not always the case. Second, note that if u is a polynomial of degree one, this error bound is zero and u_h is exactly u. You can use this as a way of constructing exact solutions to validate your own coding of the method. Third, the fact that the bound is proportional to h makes the method a **method of order one**. This means that if you make the longest edge half its size, you should only expect the error to be divided by two. Be aware that the argument on error decrease is done on the bound, since the error itself is unknown. In fact the error could decrease much faster, but in principle you should not expect this to happen.

3 Quadratic elements

Its very low order makes the $\mathbb{P}_1$ method not very attractive. Just to expect having an additional digit in precision you should have edges ten times shorter, which amounts to

[7]Inradius is the geometric term for the radius of the inscribed circumference.

increasing dramatically the number of unknowns. Instead, it is often recommended to use a higher order method, which is exactly what we are going to do right now.

3.1 Local and global descriptions

Let us consider the space of polynomials in two variables with degree at most two

$$\mathbb{P}_2 = \big\{a_0 + a_1\, x + a_2\, y + a_2\, x^2 + a_4\, y^2 + a_5\, x\, y \ : \ a_0, \ldots, a_5 \in \mathbb{R}\big\}.$$

An element of $\mathbb{P}_2$ is determined by six independent parameters (the quantities a_i), that is, the space $\mathbb{P}_2$ has dimension equal to six. Let us take a triangle K and let us mark six points as **nodes**:

- the three vertices of the triangle,
- the midpoints of the three edges.

The following result is easy to prove: *a function in $\mathbb{P}_2$ is uniquely determined by its values on the six nodes of the triangle.* Take now two points $\mathbf{p}_1$ and $\mathbf{p}_2$. The function

$$[0,1] \ni t \longmapsto (1-t)\,\mathbf{p}_1 + t\,\mathbf{p}_2$$

parameterizes linearly the segment between these two points. If $p \in \mathbb{P}_2$, then a simple computation shows that

$$p((1-t)\mathbf{p}_1 + t\,\mathbf{p}_2) \in \mathbb{P}_2(t) = \big\{b_0 + b_1\, t + b_2\, t^2 \ : \ b_0, b_1, b_2 \in \mathbb{R}\big\},$$

that is, seen on any segment (on any straight line actually), an element of $\mathbb{P}_2$ is a parabolic function, which, as everyone knows, is determined by three different points. Therefore *the value of a function in $\mathbb{P}_2$ on an edge of the triangle is uniquely determined by its three values on the nodes that lie on that edge* (two vertices and one midpoint).

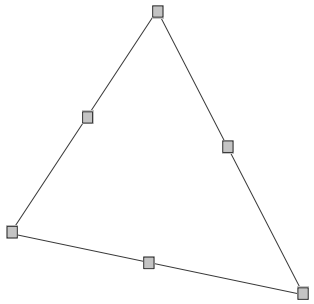


Figure 2.7: The nodes (local degrees of freedom) of a $\mathbb{P}_2$ triangle.

Because of this last property, we can glue together two $\mathbb{P}_2$ triangles as we did in the $\mathbb{P}_1$ case. Take a triangulation in the usual conditions, fix values of a function in all the nodes (vertices and midpoints) and on each triangle construct the only function in $\mathbb{P}_2$

that matches the given values. The resulting function is continuous. In fact it is a general element of the space

$$V_h = \left\{ u_h \in \mathcal{C}(\overline{\Omega}) \,:\, u_h|_K \in \mathbb{P}_2, \quad \forall K \in \mathcal{T}_h \right\}.$$

All the arguments presented in Lesson 1 hold also here. The dimension of this space is

$$\dim V_h = \#\{\text{vertices}\} + \#\{\text{edges}\},$$

since there is one midpoint per edge.

As before, we give a global numbering to the set of nodes (vertices and midpoints of edges): $\{\mathbf{p}_1, \ldots, \mathbf{p}_N\}$ and construct the functions $\varphi_i \in V_h$ satisfying

$$\varphi_i(\mathbf{p}_j) = \delta_{ij} = \begin{cases} 1, & j = i, \\ 0, & j \neq i. \end{cases}$$

For the same reasons as in the $\mathbb{P}_1$ case, these functions constitute a basis of V_h and any function of this space can be expressed as

$$u_h = \sum_{j=1}^{N} u_h(\mathbf{p}_j)\varphi_j.$$

There are two types of basis functions now:

- those associated to vertices, whose support is the set of triangles surrounding the vertex,
- those associated to midpoints, whose support is the set of two triangles (only one if the edge is on the boundary) that share the edge.

Take the usual triangulation and make yourself some drawing of the form of the supports of the nodal basis functions.

The concept of a Dirichlet node is the same: it is any node on a Dirichlet edge, Dirichlet edges being edges on the Dirichlet boundary Γ_D. The following result is then a straightforward consequence of the fact that value on edges is determined by degrees of freedom on edges:

$v_h \in V_h$ vanishes on Γ_D if and only if it vanishes on all Dirichlet nodes.

Therefore, it is very simple to construct a basis of

$$V_h^{\Gamma_D} = V_h \cap H^1_{\Gamma_D}(\Omega) = \{v_h \in V_h \,:\, v_h = 0 \quad \text{on } \Gamma_D\}$$

by simply ignoring nodal basis functions φ_i associated to Dirichlet nodes. Can you notice that I am copy-pasting formulas from Lesson 1?

Very important. The whole of Section 3 in Lesson 1 can be read with these adapted concepts. There's nothing new at all, but the different concepts of local spaces and nodes. *You should have a detailed look again at that section* to convince yourself that this is so. In particular pay attention to mass and stiffness matrices and note that the number of adjacent nodes for each node is increased with respect to $\mathbb{P}_1$ triangles (we will explore this in an exercise). □

Bookkeeping for quadratic elements. Counting local and global degrees of freedom on quadratic elements gets us into a new world of (minor) difficulties. So far we had the lists of *vertices* of the triangulation, and the lists of *elements*. For quadratic elements, we need to number the edges of the triangulation. This is a list of what vertices of the triangulation are the vertices surrounding each of the edges. This list gives an automatic numbering of the midpoints of the edges, which are nodes in the $\mathbb{P}_2$ elements. We can then consider that the list of all nodes is built as follows:

- first all the vertices,
- then all the (midpoints of) the edges.

With this numbering, the degrees of freedom corresponding to the midpoints of the edges come at the end. We also have to relate the local and the global lists. This can be easily done with yet another list: we now produce the list of the three edges (global numbering) for each of the elements. (We can typically think that the first edge is the opposed to the first vertex, etc.) At the time of the assembly, the indices referred to vertices are taken from the list of elements, and the indices referred to midpoints are taken from the list of edges, adding the number of vertices, so that it is correlative.

3.2 The reference element

If we want to implement the $\mathbb{P}_2$ we need to compute the usual integrals for the mass and stiffness matrices (an also, of course, the right-hand sides, that include the influence of data). For that, we need a way of evaluating the nodal basis functions on each triangle.

Since the argument is, again, exactly the same as for the $\mathbb{P}_1$ element, let us work now in the opposite sense. In the reference triangle we mark the six nodes as shown in Figure 2.8. As usual (ξ, η) are the coordinates in the reference configuration.

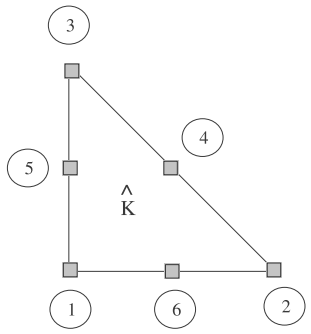


Figure 2.8: The $\mathbb{P}_2$ reference triangle.

Each of the functions

$$\widehat{N}_1 = (1-\xi-\eta)(1-2\xi-2\eta), \qquad \widehat{N}_2 = \xi(2\xi-1), \qquad \widehat{N}_3 = \eta(2\eta-1)$$

$$\widehat{N}_4 = 4\xi\eta, \qquad \widehat{N}_5 = 4\eta(1-\xi-\eta), \qquad \widehat{N}_6 = 4\xi(1-\xi-\eta)$$

takes the unit value on the corresponding node (the one numbered with the subindex) and vanishes in all other five nodes.

Let's do again some copy–pasting. The functions

$$N_\alpha^K = \widehat{N}_\alpha \circ F_K^{-1}, \qquad \alpha = 1, \ldots, 6$$

have the same property as the $\widehat{N}_\alpha$ functions, only on the triangle K, that is mapped from $\widehat{K}$ via the linear transformation F_K. These functions are polynomials of degree two (do you see why?) and therefore

$$N_\alpha^K = \varphi_{n_\alpha}, \qquad \text{in } K$$

where n_α is the global index corresponding to the local node α in K. Here is again the formula for the gradient

$$\nabla N_\alpha^K = \mathbf{B}_K^{-\top}\big((\widehat{\nabla}\widehat{N}_\alpha) \circ F_K^{-1}\big).$$

Note that now $\widehat{\nabla}\widehat{N}_\alpha$ is not constant, so the inverse transformation F_K^{-1} is needed also to evaluate the gradient.

We then compute the local matrices, which are 6×6 matrices,

$$\int_K \nabla N_\beta^K \cdot \nabla N_\alpha^K \qquad \text{and} \qquad \int_K N_\beta^K \, N_\alpha^K,$$

put the elements in the global positions

$$\int_K \nabla \varphi_{n_\beta} \cdot \nabla \varphi_{n_\beta} \qquad \text{and} \qquad \int_K \varphi_{n_\beta} \, \varphi_{n_\alpha}$$

and add the contributions of all triangles to assemble the full stiffness and mass matrices.

3.3 Convergence

The general error bound

$$\|u - u_h\|_{1,\Omega} \leq (1 + \frac{M}{\alpha}) \inf \big\{ \|u - v_h\|_{1,\Omega} \,:\, v_h \in V_h,\ v_h(\mathbf{p}) = g_0(\mathbf{p})\ \forall \mathbf{p} \text{ Dirichlet node} \big\}.$$

still holds here. In the case of homogeneous Dirichlet conditions, we can use the same arguments as in the preceding section to obtain a full bound like

$$\|u - u_h\|_{1,\Omega} \leq C h^2 |u|_{3,\Omega},$$

where:

- the constant C depends on the PDE operator, on the geometry and on the smallest angle (becoming worse as the triangles become flatter)
- the new Sobolev seminorm $|u|_{3,\Omega}$ uses the third order partial derivatives of u.

The result is valid only when this last seminorm is finite, which is much more to require than what we had at the beginning. Note that the **order two** in energy norm ($H^1(\Omega)$ norm) is good news, since using smaller triangles really pays off and the gain of precision is due to be much faster than in the $\mathbb{P}_1$ case. In the final exercise of this section we will explore what's the price to be paid (there's no free lunch, you know).

4 Cubic elements and static condensation

4.1 The $\mathbb{P}_3$ element

Can we do better than order two? The answer is yes, and besides, it is easy to do better. We will just give some hints on the order three case, because something new appears and we really want to deal with new ideas instead of doing the same thing over and over. Look

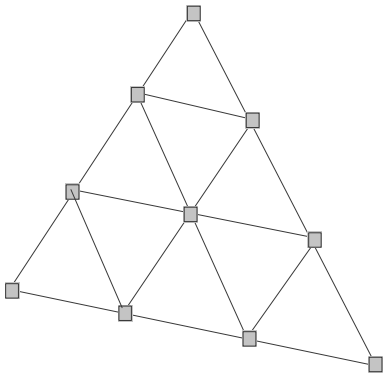


Figure 2.9: The $\mathbb{P}_3$ triangle

first at Figure 2.9. There are ten nodes in the triangle:

- the three vertices,
- two points per side, at relative distances 1/3 and 2/3 from the vertices,
- the barycenter, which is computed by averaging the coordinates of the three vertices

$$\tfrac{1}{3}\mathbf{v}_1^K + \tfrac{1}{3}\mathbf{v}_2^K + \tfrac{1}{3}\mathbf{v}_3^K .$$

Note also that each edge has four nodes on it. The local space is that of polynomials of degree up to three $\mathbb{P}_3$. Instead of writing a general element of this space, let us list the monomials that are used:

$$\begin{array}{cccc} 1 & & & \\ x & y & & \\ x^2 & xy & y^2 & \\ x^3 & x^2y & xy^2 & y^3 \end{array}$$

Count them. Ten monomials (i.e., ten coefficients) and ten nodes. Well, that's a surprise! Two other surprises:

- *a function in $\mathbb{P}_3$ is uniquely determined by its values on the ten nodes of the triangle,*
- *the value of a function in $\mathbb{P}_3$ on an edge of the triangle is uniquely determined by its four values on the nodes that lie on that edge.*

Note that a $\mathbb{P}_3$ function restricted to a segment (straight line) is a cubic function of one variable.

We are almost done here. We can construct the spaces V_h, the nodal basis functions φ_i, the subspace $V_h^{\Gamma_D}$ by eliminating Dirichlet nodes, etc. The dimension of V_h is

$$\#\{\text{vertices}\} + 2\,\#\{\text{edges}\} + \#\{\text{triangles}\}.$$

4.2 Static condensation

There is however a new entity here, and that's the very isolated interior node. I say isolated because that node is only adjacent to the other nodes of the same triangle. This has some consequences at the practical level that we are going to explore right now.

Let φ_i be the nodal basis function associated to a node that is the barycenter of the triangle K. Then $\operatorname{supp}\varphi_i = K$. Therefore

$$a(\varphi_j, \varphi_i) = \int_K \nabla\varphi_j \cdot \nabla\varphi_i + c \int_K \varphi_j\,\varphi_i \qquad \forall j,$$

and

$$\ell(\varphi_i) = \int_K f\,\varphi_i$$

(do you see why there is no Neumann term here?) which means that once we have gone through the element K in the assembly process, we will have finished the i-th row of the system, with no contributions from other elements. The idea of **static condensation** is simple: get rid of that equation and unknown in the same process of assembly.

Let us consider that the 0-th node is the barycenter of K. Let $\mathbf{K}^K$ and $\mathbf{b}^K$ be the local matrix and right–hand side contributions from the triangle K

$$k^K_{\alpha\beta} = \int_K \nabla N^K_\beta \cdot \nabla N^K_\alpha + c \int_K N^K_\beta\, N^K_\alpha, \qquad b^K_\alpha = \int_K f\, N^K_\alpha, \qquad \alpha,\beta = 0,\dots,9.$$

Now we decompose the matrix and the vector into blocks, separating the contribution from the interior node from all others:

$$\begin{bmatrix} \mathbf{K}^K_{00} & \mathbf{K}^K_{01} \\ \mathbf{K}^K_{10} & \mathbf{K}^K_{11} \end{bmatrix}, \qquad \begin{bmatrix} \mathbf{b}^K_0 \\ \mathbf{b}^K_1 \end{bmatrix},$$

with

$$\mathbf{K}^K_{00} = \begin{bmatrix} k^K_{0,0} \end{bmatrix}, \qquad \mathbf{K}^K_{01} = \begin{bmatrix} k^K_{0,1} & \dots & k^K_{0,9} \end{bmatrix}, \qquad \mathbf{b}^K_0 = \begin{bmatrix} b^K_0 \end{bmatrix}$$

$$\mathbf{K}^K_{10} = \begin{bmatrix} k^K_{1,0} \\ \vdots \\ k^K_{9,0} \end{bmatrix}, \qquad \mathbf{K}^K_{11} = \begin{bmatrix} k^K_{1,1} & \dots & k^K_{1,9} \\ \vdots & & \vdots \\ k^K_{9,1} & \dots & k^K_{9,9} \end{bmatrix}, \qquad \mathbf{b}^K_1 = \begin{bmatrix} b^K_1 \\ \vdots \\ b^K_9 \end{bmatrix}.$$

You will be wondering why are we calling matrices to blocks 1×1 (scalars) and 1×9 or 9×1 (row or column vectors). The reason is twofold: first, the role these scalars and vectors are playing are the ones of blocks in a matrix so we'd better use block notation,

independent of their shape; second, we will thus be able to recycle all that comes right now for more complicated situations.

The (ten) equations related to the nodes of the element K are

$$\begin{aligned}&\mathbf{K}_{00}^K\mathbf{u}_0^K+\mathbf{K}_{01}^K\mathbf{u}_1^K=\mathbf{b}_0^K,\\&\mathbf{K}_{10}^K\mathbf{u}_0^K+(\mathbf{K}_{11}^K+\mathbf{A})\mathbf{u}_1^K+\mathbf{B}\mathbf{u}_{\text{other}}=\mathbf{b}_1^K+\mathbf{b}.\end{aligned}$$

The unknowns are separated in the same blocks (1 plus 9) and are denoted with local numbering, that is $\mathbf{u}_0^K$ is the unknown associate to the barycenter of K and $\mathbf{u}_1^K$ is the column vector of the nine unknowns associated to all the other nodes on K.

- The matrix $\mathbf{A}$ includes all contributions from other elements to nodes of K. It will be added in the assembly process when we go through these elements.
- The block $\mathbf{B}$ includes all contributions from other triangles to other unknowns (generically written as $\mathbf{u}_{\text{other}}$), that is, unknowns on nodes that are not on K but are adjacent to those on K.
- Finally, $\mathbf{b}$ includes all contributions from other triangles and possibly also from Neumann edges, to the right-hand side.

Now we can write $\mathbf{u}_0^K$ (which, in this case, is just the unknown corresponding to the barycenter of K) as

$$\mathbf{u}_0^K=\left(\mathbf{K}_{00}^K\right)^{-1}\mathbf{b}_0^K-\left(\mathbf{K}_{00}^K\right)^{-1}\mathbf{K}_{01}^K\mathbf{u}_1^K$$

and substitute this expression in the block of the remaining equations for the triangle K (the non–interior unknowns), obtaining

$$\left(\mathbf{K}_{11}^K-\mathbf{K}_{10}^K\left(\mathbf{K}_{00}^K\right)^{-1}\mathbf{K}_{01}^K+\mathbf{A}\right)\mathbf{u}_1^K+\mathbf{B}\mathbf{u}_{\text{other}}=\mathbf{b}_1^K-\mathbf{K}_{10}^K\left(\mathbf{K}_{00}^K\right)^{-1}\mathbf{b}_0^K+\mathbf{b}$$

This means that instead of assembling the full (10×10) block from K and its corresponding right–hand side, we can forget about the interior nodes (just one) on condition of assembling

$$\mathbf{K}_{\text{cond}}^K=\mathbf{K}_{11}^K-\mathbf{K}_{10}^K\left(\mathbf{K}_{00}^K\right)^{-1}\mathbf{K}_{01}^K,\qquad\mathbf{b}_{\text{cond}}^K=\mathbf{b}_1^K-\mathbf{K}_{10}^K\left(\mathbf{K}_{00}^K\right)^{-1}\mathbf{b}_0^K$$

instead of the original matrix. Once we have solved the system, the interior variables are solved using the local equations

$$\mathbf{K}_{00}^K\mathbf{u}_0^K+\mathbf{K}_{01}^K\mathbf{u}_1^K=\mathbf{b}_0^K,$$

that work element–by–element.

Remark. This is a method for implementing the $\mathbb{P}_3$ FEM in a way that the information of the interior nodes is incorporated to the assembly process directly without having to use the corresponding unknown. This doesn't mean that the node is not there. We only compute it separately after having added its contribution to assembly directly. So don't

confuse this, which is nothing else than an implementation trick, with some finite elements (in the class of the so-called exotic or serendipity elements) that avoid interior nodes. □

Maybe I've left you wondering about that strange Algebra in the assembly process and it somehow rings a bell. It should. Write the extended matrix

$$\left[\begin{array}{cc|c} \mathbf{K}_{00}^K & \mathbf{K}_{01}^K & \mathbf{b}_0^K \\ \mathbf{K}_{10}^K & \mathbf{K}_{11}^K & \mathbf{b}_1^K \end{array}\right]$$

and apply Gaussian block elimination (the $\mathbf{K}_{00}^K$ block is just 1×1, so this is just Gauss elimination) you obtain

$$\left[\begin{array}{cc|c} \mathbf{K}_{00}^K & \mathbf{K}_{01}^K & \mathbf{b}_0^K \\ \mathbf{0} & \mathbf{K}_{11}^K - \mathbf{K}_{10}^K \left(\mathbf{K}_{00}^K\right)^{-1} \mathbf{K}_{01}^K & \mathbf{b}_1^K - \mathbf{K}_{10}^K \left(\mathbf{K}_{00}^K\right)^{-1} \mathbf{b}_0^K \end{array}\right].$$

Ta daaaa! There they are. The blocks you wanted. Again, our diagonal block was a scalar, so this was easy. What would have happened if it was a matrix? Do you have to compute that inverse and apply all that Algebra? No, you don't. Gauss block elimination is a nice way of writing the result of Gauss elimination. The point is you apply row elimination to create all those zeros, with no row changes and without trying to create any other zeros. Blocks of the form

$$\mathbf{K}_{11}^K - \mathbf{K}_{10}^K \left(\mathbf{K}_{00}^K\right)^{-1} \mathbf{K}_{01}^K$$

are called Schur complements. If the original matrix is symmetric and positive definite, they are still symmetric and positive definite.

4.3 Convergence, $\mathbb{P}_4$ and higher

We haven't mentioned convergence of the $\mathbb{P}_3$ method yet. In the best possible conditions, this is a method of order three in the $H^1(\Omega)$ Sobolev norm:

$$\|u - u_h\|_{1,\Omega} \leq Ch^3 |u|_{4,\Omega}$$

(can you guess what's in $|u|_{4,\Omega}$?). These best possible conditions include the fact that triangles do not become too flat, since the constant C becomes worse and worse as triangles get flatter and flatter. Note that if you apply static condensation to the $\mathbb{P}_3$ you complicate the assembly process but you end up with a system of order

$$\#\{\text{vertices}\} + 2\,\#\{\text{edges}\}$$

(minus the number of Dirichlet nodes), which is smaller than the one you obtain without condensation. There is an additional advantage of applying condensation. With the usual information of a grid generator (you will have to read the Appendix for that) you can easily construct a coherent numbering including vertices and edges, which works for $\mathbb{P}_2$ elements. Going from $\mathbb{P}_2$ to $\mathbb{P}_3$ means that you have to double the number of unknowns per edge (which is easy) and add the triangles. The numbering of triangles becomes then

relevant. It is not, I insist, for the assembly process. If you apply static condensation, you avoid the unknowns related to barycenter and the numbering of vertices-and-edges is enough for the $\mathbb{P}_3$ element.

The $\mathbb{P}_4$ element is constructed easily following these lines:

- You divide each edge into five equally sized pieces. Then you join these new points on different sides with lines that run parallel to the edges. With that you have created a grid of 15 nodes: three vertices, three points per edge, three interior points, placed on the intersections of the interior lines.
- The space is $\mathbb{P}_4$, which has dimension 15. Everything goes on as usual.
- The three interior nodes can be treated with static condensation: the $\mathbf{K}_{00}^K$ blocks are now 3×3 blocks. With this you reduce in three times the number of triangles the size of the global system to be solved without affecting convergence.
- Order of the method is.... four! (That was easy)

It is possible to create $\mathbb{P}_k$ methods for arbitrary k. You will find people around that will assert that these methods are useless or just of theoretical interest. Be warned: maybe they find them useless, but some other people work with really high order methods and find many advantages in them[8]. However, if you go from $\mathbb{P}_4$ upwards, you implement the method in a very different way. Nodal bases are not the best choice in that case and there is a different way of constructing node-free bases. We will deal with this in Lesson 7.

5 Exercises

1. **Basis functions for the $\mathbb{P}_2$ element.** Try to sketch the form of the nodal basis functions for a $\mathbb{P}_2$ finite element space (similar as Figure 1.8). Note that there are two different types of functions, those associated to vertices and those associated to midpoints of edges.

2. **The plane elasticity system.** The problem of plane deformations in linear elasticity can be reduced to the variational problem:[9]

$$\left[\begin{array}{l}
\text{find } u_1, u_2 \in H^1(\Omega) \text{ such that} \\
u_1 = g_x, \quad u_2 = g_y \qquad \text{on } \Gamma_D, \\
\displaystyle\int_\Omega \Big((\lambda+2\mu)\frac{\partial u_1}{\partial x} + \lambda\frac{\partial u_2}{\partial y}\Big)\frac{\partial v}{\partial x} + \mu\Big(\frac{\partial u_1}{\partial y} + \frac{\partial u_2}{\partial x}\Big)\frac{\partial v}{\partial y} = \displaystyle\int_\Omega v\, f_x + \int_{\Gamma_N} v\, t_x \quad \forall v \in H^1_{\Gamma_D}(\Omega), \\
\displaystyle\int_\Omega \mu\Big(\frac{\partial u_1}{\partial y} + \frac{\partial u_2}{\partial x}\Big)\frac{\partial v}{\partial x} + \Big(\lambda\frac{\partial u_1}{\partial x} + (\lambda+2\mu)\frac{\partial u_2}{\partial y}\Big)\frac{\partial v}{\partial y} = \displaystyle\int_\Omega v\, f_y + \int_{\Gamma_N} v\, t_y \quad \forall v \in H^1_{\Gamma_D}(\Omega),
\end{array}\right.$$

   where:

[8] Be always prepared to find opinionated people in the scientific computing community. Sometimes they are right, sometimes they are partially right, sometimes they are plain wrong.

[9] **Warning.** For reasons that are not so easy to explain as many people think, $\mathbb{P}_1$ elements are never used in elasticity problems because their performance is rather bad. Note that in what you have done here $\mathbb{P}_1$ or $\mathbb{P}_k$ is all the same, so you can be applying this to $\mathbb{P}_2$ elements, which work well for this problem.

- Ω is the plane section of the cylindrical solid
- Γ_D is the part of the boundary of Ω where we know displacements $g_0 = (g_x, g_y)$
- Γ_N is the part of the boundary where we know normal stresses $t = (t_x, t_y)$
- $f = (f_x, f_y)$ are the volume forces
- λ and $\mu = G$ are the Lamé parameters

$$\lambda = \frac{\nu E}{(1+\nu)(1-2\nu)}, \qquad \mu = \frac{E}{2(1+\nu)}$$

- $H^1_{\Gamma_D}(\Omega) = \{v \in H^1(\Omega) \,:\, v|_{\Gamma_D} = 0\}$.

We are given a triangulation $\mathcal{T}_h$, the associated $\mathbb{P}_1$ nodal basis functions (φ_i), etc. We call Ind and Dir to the usual index sets. We approximate the pair (u_1, u_2) by the discrete functions

$$u_h^1 = \sum_j u_j^1 \varphi_j, \qquad u_h^2 = \sum_j u_j^2 \varphi_j$$

Alternating tests with the two variational equations, and grouping both unknowns on the same node (u_j^1, u_j^2) prove that the resulting finite element system can be written in the form

$$\sum_{j \in \mathrm{Ind}} A_{ij} \begin{bmatrix} u_j^1 \\ u_j^2 \end{bmatrix} = F_i + T_i - \sum_{j \in \mathrm{Dir}} A_{ij} \begin{bmatrix} g_j^1 \\ g_j^2 \end{bmatrix}, \qquad i \in \mathrm{Ind}\,.$$

where A_{ij} are 2×2 matrices. What's the dimension of the system? Prove that $A_{ij}^\top = A_{ji}$ and deduce that the system is symmetric.

3. **Comparison of $\mathbb{P}_1$ and $\mathbb{P}_2$.** Consider the simple triangulation depicted in Figure 2.10

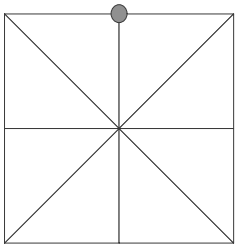


Figure 2.10: A simple triangulation with a marked node

(a) If you consider the Neumann problem there (no Dirichlet nodes), how many unknowns are there in the system corresponding to the $\mathbb{P}_2$ method?

(b) What are the adjacent nodes to the node that is marked on the figure?

(c) A red refinement of a triangle consists of taking the midpoints of the edges and joining them to create four triangles per triangle (see Figure 2.11). If you apply a red refinement to all the elements of the triangulation above and the apply the $\mathbb{P}_1$ element, how many unknowns do you have in the system? Which nodes are adjacent to the same marked nodes in this new triangulation for the $\mathbb{P}_1$ method?

(d) **Discussion.** The error of the $\mathbb{P}_2$ method is bounded by something times h^2. The error of the $\mathbb{P}_1$ method on the uniform red refinement is something else times $h/2$. The constant (the unspecified something) for each case is different. In principle, when the triangulation is fine enough h^2 wins over $h/2$ (it is smaller). With the same number of unknowns one method is better than the other. Where's the difference?

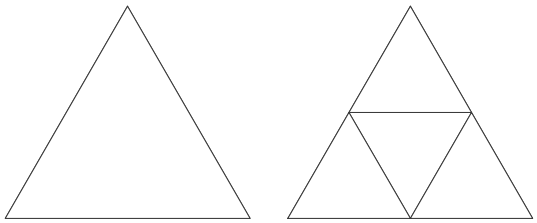


Figure 2.11: A red refinement of a triangle

4. **Bookkeeping for $\mathbb{P}_2$ elements.** Consider the triangulation given in Figure 2.10.

(a) Number vertices, edges, and elements. Construct all the lists that you would need to use $\mathbb{P}_2$ FEM:

- the list of vertices for each element,
- the list of vertices for each edge,
- the list of edges for each element.

For the list of vertices for each edge, *choose always a positive orientation for boundary edges.* This means that if you go from the first edge to the second, you are leaving the exterior domain to the right.

(b) Additionally, build a matrix (list) with the same shape as the one of edges-counted-by-element where you specify if the orientation of the edge: positive or negative. This means the following. If the first edge of an element connects the nodes n_1 and n_2 in the element (counting counterclockwise) and the edge is listed as n_1 going to n_2 you assign a + sign. If the edge is listed as n_2 going to n_1, you assign a minus sign.

(c) Using the above information, choose one element, and say where you would assemble a local mass matrix in the global matrix.