

*Introduction to Finite Element  
Modelling in Geosciences:*  
**Numerical Integration**

D. May & M. Frehner  
ETH Zürich

July 7, 2014

## 1 Overview

In the previous script we saw that 1D finite element discretisation of the diffusion equation may lead to the element matrices

$$\mathbf{MM} = \begin{bmatrix} \int_0^L N_1 N_1 dx & \int_0^L N_1 N_2 dx \\ \int_0^L N_2 N_1 dx & \int_0^L N_2 N_2 dx \end{bmatrix} \quad (1)$$

and

$$\mathbf{KM} = \begin{bmatrix} \kappa \int_0^L \frac{\partial N_1}{\partial x} \frac{\partial N_1}{\partial x} dx & \kappa \int_0^L \frac{\partial N_1}{\partial x} \frac{\partial N_2}{\partial x} dx \\ \kappa \int_0^L \frac{\partial N_2}{\partial x} \frac{\partial N_1}{\partial x} dx & \kappa \int_0^L \frac{\partial N_2}{\partial x} \frac{\partial N_2}{\partial x} dx \end{bmatrix}, \quad (2)$$

where the shape functions  $N_1$  and  $N_2$  were defined as

$$N_1(x) = 1 - \frac{x}{L}, \quad N_2(x) = \frac{x}{L}. \quad (3)$$

As we will see in the coming lectures, many different equations lead to these same (or at least very similar) matrices. A mass matrix  $\mathbf{MM}$  is always associated with discretization of a first-order time derivative (e.g.,  $\partial T/\partial t$ ) whereas the  $\mathbf{KM}$  matrix comes from the discretization of second-order spatial derivatives (e.g.,  $\partial^2 T/\partial x^2$ ). In the example we considered it was possible to evaluate the integrals in Equations (1) & (2) analytically. This was largely facilitated by the fact that we i) considered a 1D problem using ii) linear basis functions and iii) we assumed that  $\kappa$  was constant over each element. Using analytically derived element stiffness matrices can be advantageous as it avoids the need to approximate the integrals which removes one source of numerical error and may also be much more efficient than approximation technique. However, in general, if any of the above three assumptions are violated, using analytically derived stiffness matrices is not feasible. In such circumstances one must *approximate* the integrals appearing within the stiffness matrix. Throughout the remainder of this course, we will utilise integrals which have been approximated numerically.

$n$	$\xi_i$	$w_i$	$k$
1	0	2.0	1
2	$\pm\sqrt{\frac{1}{3}}$	1.0	3
3	0	0.8888889	
	$\pm 0.77459667$	0.5555555	5
4	$\pm 0.33998104$	0.65214515	
	$\pm 0.86113631$	0.34785485	7

Table 1: A selection of different order accuracy 1D Gauss-Legendre quadrature rules.

## 2 Quadrature

The method we will employ to perform numerical integration is a type of quadrature rule. In general, a quadrature rule replaces the integral by a weighted sum over a set of  $n$  quadrature points. For example, the function  $f(\xi)$  integrated over the range  $[a, b]$  is approximated via

$$\int_a^b f(\xi) d\xi \approx \sum_{i=1}^n f(\xi_i)w_i. \quad (4)$$

In Equation (4),  $\xi_i$  is the coordinate (or abscissa) of the  $i^{\text{th}}$  quadrature point and  $w_i$  is the quadrature weight associated with the  $i^{\text{th}}$  quadrature point. Note that the quadrature rule only requires one to evaluate the integrand  $f(\cdot)$  at the location of each quadrature point.

The choice of quadrature scheme defines the values of  $n$ ,  $\xi_i$  and  $w_i$ . The type of quadrature scheme which should be used is entirely dependent on the nature of the integrand  $f(\cdot)$ . In finite element analysis, the integrand is defined via products of the basis functions  $N(\cdot)$  (see **MM**) or from products of the derivative of the basis functions (see **KM**). In general, integrands of this type are polynomials of some order  $k$ , i.e. they contain terms like  $\xi^s$  where  $s = 1, 2, \dots, k$ . The family of *Gauss-Legendre* quadrature schemes have been specifically designed to exactly evaluate polynomial integrands in one dimension. We will denote by the  $n$ -point Gauss-Legendre (GLg) rule as that which uses  $n$  quadrature points. In 1D,  $n$ -point GLg rule *exactly* integrates polynomials of order  $\leq (2n - 1)$ . Tabulated values of  $n$ ,  $\xi_i$  and  $w_i$  for GLg rules which are suitable for polynomials up to third order can be found in most finite element books. In Table 1 we provide the Gauss-Legendre quadrature rules for 1D from  $n = 1 \rightarrow 4$ .

To be used in a general manner, tabulated quadrature rules assume an integration domain. GLg rules typically assume that the integration is performed over the domain  $[-1, 1]$ . The choice of integration domain is arbitrary. To use the tabulated rule in practice, one needs to define a coordinate transform from the domain of interest (say  $[a, b]$  in Equation (4), or  $[0, L]$  in Equation (1)) to the domain used by the tabulated quadrature rule. To illustrate the necessary steps to numerically evaluate a finite element stiffness matrix via GLg quadrature, we now consider evaluating

$$\int_0^L N_1(x)N_1(x) dx. \quad (5)$$

The first task is to transform the limit of integration, from what we will refer to as the global coordinate system  $0 \leq x \leq L$  to local coordinate system used by the quadrature rule given by  $-1 \leq \xi \leq 1$ . Note that the shape functions defined in Equation (3) are written in terms of the global coordinate  $x$  varying over the interval  $[0, L]$ . We can linearly map the  $[0, L]$  interval for  $x$  onto the  $[-1, 1]$  interval for  $\xi$  using the following transformation:

$$x = \frac{L}{2} (\xi + 1). \quad (6)$$

The reader should check that this transformation gives us the desired limits of  $[-1, +1]$ . This is achieved by inserting  $x = 0$  (and  $x = L$ ) into Eq. (6) and solving for  $\xi$ . Substituting Equation (6) into Equation (3) leads to the shape functions defined in terms of local coordinates

$$N_1(\xi) = \frac{1}{2} (1 - \xi), \quad N_2(\xi) = \frac{1}{2} (1 + \xi). \quad (7)$$

To perform the integration over the local coordinate system  $[-1, +1]$ , we must first transform the variable of integration. This is obtained by differentiating Equation (6) with respect to  $\xi$  and rearranging to yield

$$dx = \frac{L}{2} d\xi. \quad (8)$$

Thus, one can rewrite the integral in Equation (5) as

$$\int_0^L N_1(x) N_1(x) dx = \int_{-1}^1 N_1(\xi) N_1(\xi) \frac{L}{2} d\xi. \quad (9)$$

As will be seen in the following lectures, the term  $\frac{L}{2}$  appearing in Equation (9), which is responsible for converting the integral from the global to the local coordinate system, is referred to as the determinant of the Jacobian,  $\det(\mathbf{J})$ .

Now we are ready to numerically approximate the integral using the summation formula in Equation (4). For example, using the 2 point Gauss-Legendre rule (i.e.,  $n = 2$ , see Table 1) the integral can be computed as

$$\begin{aligned} \int_{-1}^1 N_1(\xi) N_1(\xi) \frac{L}{2} d\xi &\approx N_1\left(-\sqrt{1/3}\right) N_1\left(-\sqrt{1/3}\right) \times \frac{L}{2} \times 1.0 \\ &\quad + N_1\left(\sqrt{1/3}\right) N_1\left(\sqrt{1/3}\right) \times \frac{L}{2} \times 1.0 \\ &= \frac{L}{2} (0.6220084681 \times 1.0 + 0.04465819869 \times 1.0) \\ &= 0.333L, \end{aligned}$$

noting that  $N_1(\xi)N_1(\xi)$  evaluated at  $\sqrt{1/3}$  is 0.04465819869 and  $N_1(\xi)N_1(\xi)$  at  $\xi = -\sqrt{1/3}$  is 0.6220084681 and the weighting factors are both 1.0 (see Table 1). Note that this result is identical to the result obtained by the analytic calculation (i.e.,  $L/3$ ).

As a second example, consider calculation of the term

$$\int_0^L \frac{\partial N_1}{\partial x} \frac{\partial N_1}{\partial x} dx, \quad (10)$$

appearing in Equation (2). Note firstly that, by using the chain rule,

$$\frac{\partial N_1}{\partial x} = \frac{\partial N_1}{\partial \xi} \frac{\partial \xi}{\partial x}. \quad (11)$$

In general, the derivative of the global coordinate with respect to local coordinate (i.e.,  $\partial x/\partial \xi$ ) is known as the Jacobian (denoted  $\mathbf{J}$ ). The inverse relation,  $\partial \xi/\partial x$  (appearing in Equation (11)) is known as the inverse of the Jacobian, and this must always be calculated to convert derivatives from global coordinates to local coordinates.

Therefore, one can rewrite Equation (10) as

$$\begin{aligned} \int_0^L \frac{\partial N_1}{\partial x} \frac{\partial N_1}{\partial x} dx &= \int_0^L \left( \frac{\partial N_1}{\partial \xi} \frac{\partial \xi}{\partial x} \right) \left( \frac{\partial N_1}{\partial \xi} \frac{\partial \xi}{\partial x} \right) dx \\ &= \int_{-1}^1 \left( \frac{\partial N_1}{\partial \xi} \frac{\partial \xi}{\partial x} \right) \left( \frac{\partial N_1}{\partial \xi} \frac{\partial \xi}{\partial x} \right) \frac{L}{2} d\xi \\ &= \int_{-1}^1 \frac{\partial N_1}{\partial \xi} \frac{\partial N_1}{\partial \xi} \frac{2}{L} d\xi, \end{aligned} \quad (12)$$

where we used the fact that  $\partial x/\partial \xi = \frac{L}{2}$  (see Equation (8)). Note that  $\partial N_1/\partial \xi = -1/2$ , which is constant (see Equation (7)). Thus by using the 2-point GLg quadrature rule, the term represented by Equation (12) can be integrated to give

$$\begin{aligned} \int_{-1}^1 \frac{\partial N_1}{\partial \xi} \frac{\partial N_1}{\partial \xi} \frac{2}{L} d\xi &\approx \frac{2}{L} \left( \left( -\frac{1}{2} \right) \left( -\frac{1}{2} \right) \times 1.0 + \left( -\frac{1}{2} \right) \left( -\frac{1}{2} \right) \times 1.0 \right) \\ &= \frac{1}{L}, \end{aligned} \quad (13)$$

which is once again the exact result.

From these simple examples, we see that numerical integration gives very accurate results with minimal computation (in this case using only 2 integration points). This accuracy is due the strength of the Gauss-Legendre method and the fact that the functions being integrated are polynomials of a known order.

For the remainder of this course we will assume that shape functions and their derivatives are given in terms of the local coordinates on the interval  $[-1, 1]$ . Moreover, we will use Gauss-Legendre quadrature which also requires integration over the interval  $[-1, 1]$ . Based on these assumptions, the following steps must be carried out in order to perform numerical integration of the terms in the element matrices:

1. Define the number of integration points and obtain the integration points and weights. Evaluate the shape functions and their derivatives (defined in terms of local coordinates) at each integration point, and save the results for later use.
2. Do a loop over all elements and initialize the element matrices  $\mathbf{MM}$  and  $\mathbf{KM}$ .
3. Within the element loop, start a loop over all integration points. One must convert derivatives from the local coordinates to the global coordinates using the inverse of the Jacobian, by performing an operation such as that in Equation (11).

4. Perform the vector multiplication involving the shape functions or their derivatives (evaluated at integration points), multiplied by the relevant weight, and multiplied by the determinant of the Jacobian (to convert integrals from global to local coordinates, see Equation (9)), which leads to the matrices  $\mathbf{MM}$  and  $\mathbf{KM}$ . The result should be added to the multiplication from proceeding integration points.
5. Once one has exited the loop over integration points, the integration is complete (i.e., the element matrices have been integrated).
6. The element matrices can then be added to the global element stiffness matrix. This process is then continued for each element.

In the 1D example discussed here, it might seem to be an overkill to use numerical integration. As you will see soon, however, doing this has immense advantages for 2D and 3D cases, as it allows you to easily deform your mesh, without writing a new code. Therefore it is good to practice on a simple example.

### 3 Exercises

1. Perform the numerical integration of Equation (10) with a 3-point quadrature rule, instead of 2-point rule. How do the results differ?
2. Using the program developed in the last session, write a FEM code for solving the diffusion equation in 1D where the element matrices and vectors are integrated numerically. Check that the results obtained are equivalent.
3. Read the following chapter from the book of Kwon & Bang. Explain which order quadrature rule should be used if one was to use quadratic elements to solve the diffusion equation? Modify your code to solve the 1D diffusion equation with quadratic elements, instead of linear elements. The 1D quadratic shape functions defined in the local coordinates system are given by

$$N_1(\xi) = \frac{1}{2}\xi(\xi - 1) \quad (14)$$

$$N_2(\xi) = 1 - \xi^2 \quad (15)$$

$$N_3(\xi) = \frac{1}{2}\xi(\xi + 1). \quad (16)$$

Note that each 1D quadratic element is composed of three nodes, instead of two nodes (as with the linear elements). Compare the results of the solution obtained using a 3-point GLq quadrature rule and a 2-point GLq quadrature rule.