Chapter 7

Finite Element Method

The Finite Element Method (FEM) was originated from the field of structural calculation (e.g. stress analysis) in the beginning of the fifties. The terminus - **finite elements** - was introduced by Turner et al. (1956). The concept of the finite element technique consists in a subdivision of a complex structure into small substructures assembling the elements. After its initial development as an engineering tool mathematicians worked out a rigorous formal framework of this method, e.g. concerning consistence of solutions, criteria for numerical stability and convergence behavior as well as accuracy and error bounds. The mathematical background of FEM is functional analysis. The FEM was introduced into the field of computational fluid dynamics (CFD) by Chung (1978), Baker (1983), Huyakorn & Pinder (1983), Diersch (1985) and others. The FEM is a more general approximation technique containing many finite difference schemes as special cases (Chapter 6).

The basic steps in the finite element approximation are:

- Space discretization by finite elements (meshing) (Chapter 7.1)
- Discretization of the integral (weak) formulations of the governing partial differential equations (Chapter 7.2)

$$L(u) \rightarrow \hat{L}(\hat{u}) \tag{7.1}$$

- Generation of interpolation functions to approximate the unknown values of field variables (Chapter 7.3)

$$\hat{u} = f(\hat{u}_1, ..., \hat{u}_n) \tag{7.2}$$

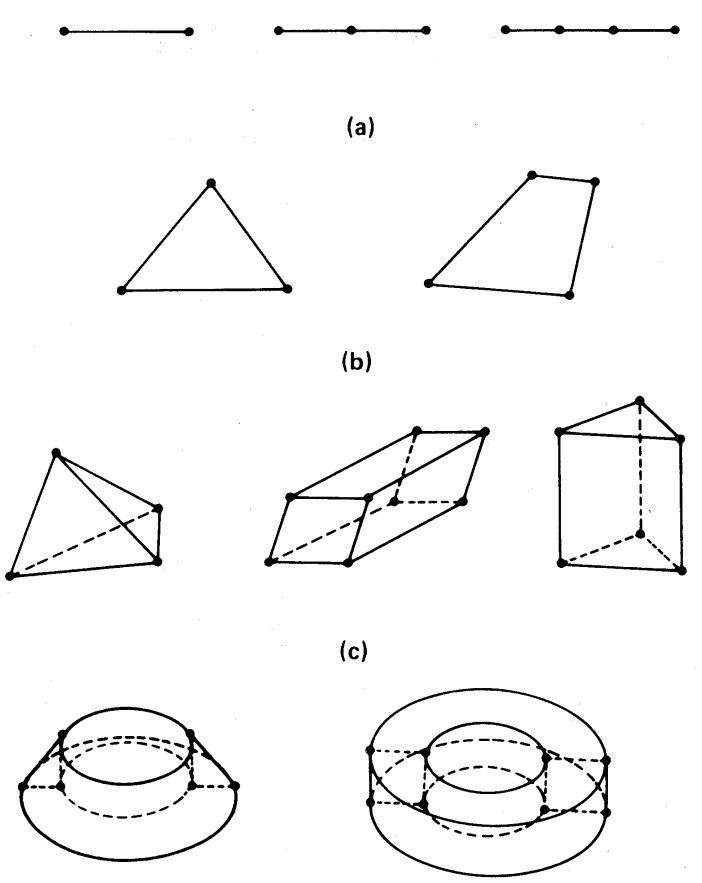

Figure 7.1: Basic element types (Huyakorn & Pinder, 1983)

7.1 Domain Discretization

The basic concept of the finite element method is the subdivision of the computational domain into elements of arbitrary shape and size. The only restriction is that the elements may not overlap and that they have to cover the complete computational domain. The geometric object is finally represented by a mesh of 1-D, 2-D and/or 3-D basic elements (geometric units) such as bars, triangles, quadrilaterals, tetrahedrons, hexahedra and pyramids (Fig. 7.1). Which type of element is most appropriate for a particular problem depends on several factors, such as domain geometry, required accuracy, computational costs etc.). The ability to handle non-uniform and distorted computational domains has always been an important feature of the finite element method. Figures 7.2 and 7.3 show finite element discretizations for fracture networks which are used in

practical computations for fluid flow and mass transport in crystalline rock. In particular, automatic grid adaptation is employed. Within each finite element a certain number of grid points is defined, which are located along the sides/faces of the element. At these points the unknown functions have to be determined. The total number of unknowns (function values, function derivatives) is denoted as the **degree of freedom** of the discrete system.

In general, finite elements are required to have convex shapes. This condition should avoid singularities of the Jacobian, to ensure unique transformation of coordinates.

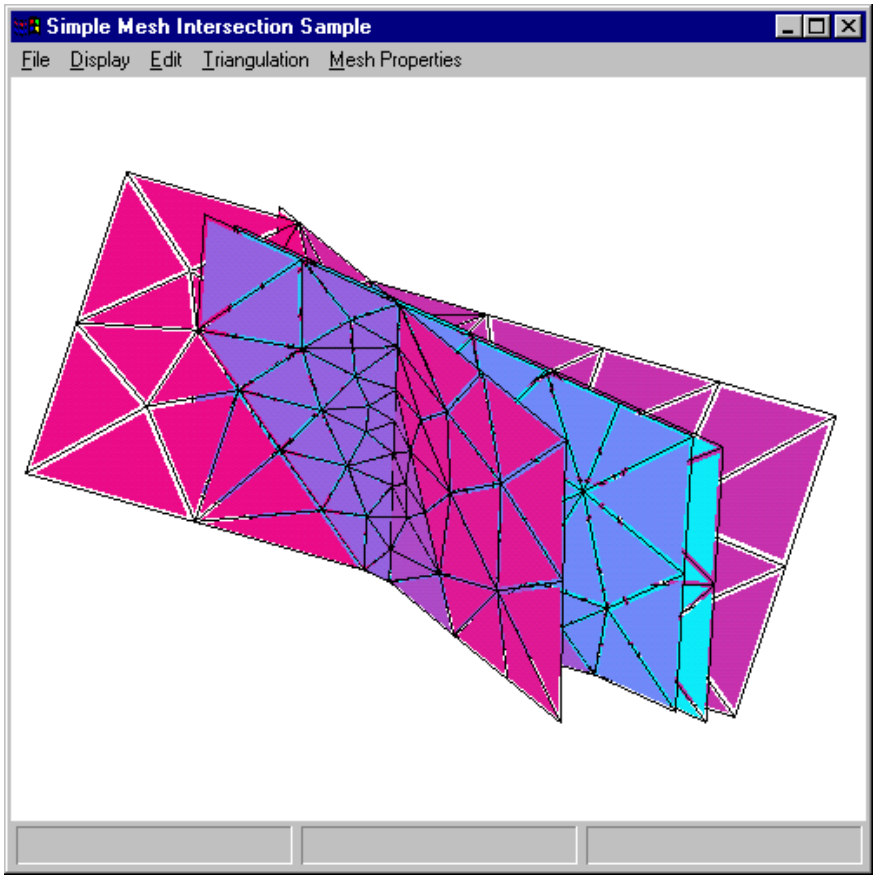

Figure 7.2: Finite element mesh for a fracture system (Kasper et al. 1998)

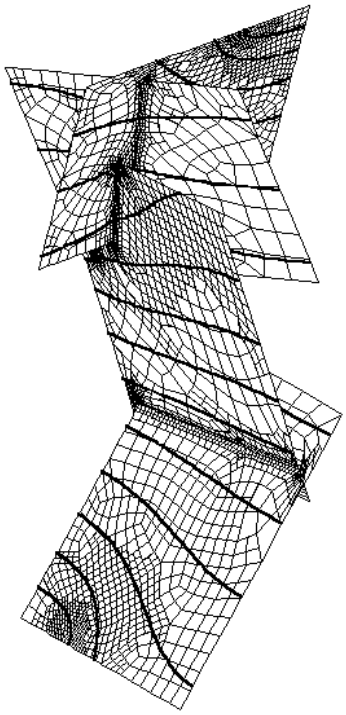

Figure 7.3: Adaptive finite element mesh for a fracture system (Kaiser et al. 1998)

7.2 Equation Discretization

The FEM requires the definition of an integral (weak) formulation of the continuum-mechanics field equations describing the physical problem. In particular, integral formulations are required in situations where discontinuous solutions are possible. In this case, derivatives of the discontinuous quantity are not defined, and, therefore, the integral formulation is the only physical meaningful problem description.

In general, there are two ways to derive integral formulations for continuum problems:

- Variational principles (Chapter 7.2.1)
- Method of weighted residuals (MWR) (Chapter 7.2.2)

A variational principle means that the physical problem can be expressed by the extremum of a functional. Although many physical problems can be described by variational equations (e.g. potential flow), there is no guarantee to find a corresponding variational principle for all problems (e.g. for flow governed by the Navier-Stokes equations). On the other hand, construction of an equivalent integral formulation, i.e. a weak formulation of the governing differential equations, based on the method of weighted residuals is always possible.

Note, that integral formulations have restricted continuity properties of the unknown functions under consideration. Therefore, it is particularly important to define appropriate criteria in order to derive meaningful convergence and error bounds.

7.2.1 Variational Principles

By use of a variational principle the PDE of the original problem is replaced by an equivalent variational problem, which consists in finding a function that maximizes a certain integral quantity, a functional - i.e. a function of an unknown function. A general expression for those functionals is given by the Euler-Lagrange equation

$$F = \int_{\Omega} f(x, u, \frac{\partial u}{\partial x}, \frac{\partial^2 u}{\partial x^2}, ...)dx \tag{7.3}$$

The maximum condition of this functional must reveal again the governing PDE and the boundary condition of the original physical problem. To obtain the maximum condition the calculus of variation is used.

Rayleigh-Ritz method

To obtain an approximate solution for a variational problem the Rayleigh-Ritz method can be used. There it is assumed, the unknown function within the whole domain Ω can be represented by following trial solution

$$\hat{u}(x) = \sum_{i=1}^{m} \phi_i(x) C_i \tag{7.4}$$

where $\phi_i(x)$ are linearly **basis functions** satisfying the boundary conditions, C_i are unknown parameters, m is the number of finite the series terms (grid points).

Therefore, the Euler-Lagrange equation (7.3) becomes

$$F(C_1, ..., C_m) = 0 \tag{7.5}$$

For maximization it has to be

$$\frac{\partial F}{\partial C_i} = 0 \quad , \quad i = 1, ..., m \tag{7.6}$$

which yields a set of algebraic equations for the determination of C_i. Unfortunately for many practical problems, variational principles does not exist.

The main difference between the Rayleigh-Ritz method (RRM) and the finite element method lies in the definition of the basis functions. For FEM, these are element-related functions, whereas for RRM these are valid for the whole domain and have to fit the boundary conditions. The Rayleigh-Ritz method for homogeneous boundary conditions leads to the same discretized equations as the Galerkin method of weighted residuals.

7.2.2 Method of Weighted Residuals

In general the method of weighted residuals (MWR) can be divided into 3 steps:

1. Approximation of the unknown function by a trial solution (kind of approximation),
2. Definition of weighting functions,
3. Derivation of a system of algebraic equations (rules for determination of nodal values).

Approximate solution

Again we assume, the unknown function within the whole domain Ω can be represented by a trial solution $\hat{u}$

$$u \approx \hat{u}(x) = \sum_{i=1}^{m} \phi_i(x) C_i \tag{7.7}$$

where $\phi_i(x)$ are linearly **basis functions** or **interpolation functions** satisfying the boundary conditions, C_i are unknown parameters, m is the number of finite the series terms (grid points).

If an approximate solution $\hat{u}$ is substituted into the original differential equation $L(u) = 0$ it will not identically fulfill this.

$$R(t, x) = L(u) - L(\hat{u}) = L(\hat{u}) \neq 0 \tag{7.8}$$

where R is the **residual**, L is the differential operator, u is the exact solution, and $\hat{u}$ is the approximate solution. The residual is a measure of the accuracy of approximation or of the error introduced by the discretization method. Since the error in general can not be made to vanish simultaneously in all mesh points, an approximate solution can be derived by requiring that some weighted average of the nodal residuals should vanish over the computational domain.

$$< \omega, R >= \int_{\Omega} \omega_i R(t,x) d\Omega = \int_{\Omega} \omega_i L(\hat{u}) d\Omega = 0 \tag{7.9}$$

where ω_i are **weighting functions** or test functions. Now, the unknown values C_i have to be defined in such a way, that the error (i.e. weighted residuals) is minimal.

From the above equations a system of equations can be obtained for determination of the unknown nodal values of the field variable u. For steady PDEs this will be a system of algebraic equations and for unsteady PDEs a system of ordinary differential equations will result. Note, the above equation is closely related to the integral form of the governing equations. The weak formulation allows discontinuities in the solution to appear. The MWR expresses the condition that the projection of the residual in the subspace of weighting functions is zero, i.e. the residual is orthogonal to the subspace. An approximate solution algorithm will converge if R tends to zero for increasing number of weighting functions and nodes (i.e. degree of freedom).

The MWR is a more general procedure to obtain approximate solutions. Different choices of the weighting functions and corresponding algorithms to minimize the residual result in several methods.

Table 7.1: Captured methods for several weighting functions

Method	Weighting function	Remarks
Subdomain method	$\omega_i(x) = \begin{cases} 1 & : \quad x \in \Omega_i \\ 0 & : \quad x \notin \Omega_i \end{cases}$	FVM (conservation)
Collocation method	$\omega_i(x) = \delta(x - x_i)$	FDM
Least-square method	$\omega_i(x) = \partial R / \partial C_i$	Minimization problem
Galerkin method	$\omega_i(x) = \phi_i(x)$	FEM

7.2.3 Galerkin Method

The most popular scheme of MWR is the Bubnov-Galerkin method in which weighting and interpolation functions are selected identically $\omega = \phi$.

Example: General conservation law

As a first example we consider the general conservation law in differential form (see section 1.1, equation (1.25)).

$$\frac{\partial u}{\partial t} + \nabla \cdot \mathbf{\Phi}^u = \frac{\partial u}{\partial t} + \frac{\partial \mathbf{\Phi}^u_\alpha}{\partial x_\alpha} = q^u \tag{7.10}$$

where $\mathbf{\Phi}^u$ is the flux vector of conservation quantity u and q^u is a source term. The corresponding weak form of the conservation equation in analogy to (10.3) due to the Galerkin method is

$$\int_\Omega \phi_i \frac{\partial u}{\partial t} d\Omega + \int_\Omega \phi_i \nabla \cdot \mathbf{\Phi}^u d\Omega = \int_\Omega \phi_i q^u d\Omega \tag{7.11}$$

The main steps are now: integration by parts of spatial derivation terms and use of Green's theorem in order to convert volume to surface integrals. After these steps we obtain

$$\int_\Omega \phi_i \frac{\partial u}{\partial t} d\Omega - \int_\Omega \mathbf{\Phi}^u \cdot \nabla \phi_i d\Omega + \oint_{\partial\Omega} \phi_i \mathbf{\Phi}^u \cdot d\mathbf{S} = \int_\Omega \phi_i q^u d\Omega \tag{7.12}$$

To simplify the notation, we omit the suffix u in the following. The finite element representations of conservation quantity and flux term are

$$\hat{u}(t,x) = \sum_j \phi_j(x) u_j(t) \quad , \quad \hat{\mathbf{\Phi}} = \sum_j \phi_j(x) \mathbf{\Phi}_j(t) \tag{7.13}$$

The discretized form of the general conservation equation is

$$\sum_i \frac{du_j}{dt} \underbrace{\int_\Omega \phi_i \phi_j d\Omega}_{M_{ij}} - \sum_i \mathbf{\Phi}_j \cdot \underbrace{\int_\Omega \phi_j \nabla \phi_i d\Omega}_{K_{ij}} + \oint_{\partial\Omega} \phi_i \mathbf{\Phi}^u \cdot d\mathbf{S} = \int_\Omega \phi_i q^u d\Omega \tag{7.14}$$

where Ω_j is the subdomain containing node j. The summations extends over all elements containing this node. By this means the original partial differential equation was transformed to an ordinary differential equation. In this representation we have the so-called **mass matrix**

$$M_{ij} = \int_\Omega \phi_i \phi_j d\Omega \tag{7.15}$$

which is a symmetric but not diagonal one and the so-called **stiffness matrix**

$$K_{ij} = \int_\Omega \phi_j \nabla \phi_i d\Omega \tag{7.16}$$

which is no longer symmetric. Note for linear problems, both mass and stiffness matrices are independent of the unknown function u. They will depend then only on the element geometry

We have seen the application of the method of weighted residuals to differential balance equations leads to several matrix-integral expressions that must be evaluated for each finite element. To calculate these expressions we have to know the interpolation functions and their derivatives as functions of coordinate directions.

Example 2: General conservation law with advective and diffusive fluxes

As a first example we consider the general conservation law in differential form (see section 1.1, equation (1.25)).

$$\begin{aligned} \frac{\partial u}{\partial t} + \nabla \cdot \mathbf{\Phi}^u &= \frac{\partial u}{\partial t} + \nabla \cdot \mathbf{\Phi}^u_A + \nabla \cdot \mathbf{\Phi}^u_D \\ &= \frac{\partial u}{\partial t} + \nabla \cdot (\mathbf{v}u) - \nabla \cdot (\alpha \nabla u) = q^u \end{aligned} \tag{7.17}$$

where $\mathbf{\Phi}^u$ is the flux vector of conservation quantity u and q^u is a source term. The corresponding weak form of the conservation equation in analogy to (10.3) due to the Galerkin method is

$$\int_\Omega \phi_i \frac{\partial u}{\partial t}\, d\Omega + \int_\Omega \phi_i \nabla \cdot (\mathbf{v}u)\, d\Omega - \int_\Omega \phi_i \nabla \cdot (\alpha \nabla u)\, d\Omega = \int_\Omega \phi_i q^u\, d\Omega \tag{7.18}$$

The main steps are now: integration by parts of spatial derivation terms and use of Green's theorem in order to convert volume to surface integrals. After these steps we obtain

$$\begin{aligned} &\int_\Omega \phi_i \frac{\partial u}{\partial t} d\Omega - \int_\Omega (\mathbf{v}u) \nabla \phi_i\, d\Omega + \int_\Omega (\alpha \nabla u) \nabla \phi_i\, d\Omega \quad = \\ &\quad - \oint_{\partial\Omega} \phi_i (\mathbf{v}u)\, d\mathbf{S} + \oint_{\partial\Omega} \phi_i (\alpha \nabla u)\, d\mathbf{S} + \int_\Omega \phi_i q^u d\Omega \end{aligned} \tag{7.19}$$

To simplify the notation, we omit the suffix u in the following. The finite element representations of conservation quantity and flux term are

$$\hat{u}(t,x) = \sum_j \phi_j(x) u_j(t) \quad , \quad \nabla \hat{u}(t,x) = \sum_j \nabla \phi_j(x) u_j(t) \tag{7.20}$$

The discretized form of the general conservation equation is

$$\sum_i \frac{du_j}{dt} \underbrace{\int_\Omega \phi_i \phi_j \, d\Omega}_{C_{ij}} - \sum_i (\mathbf{v} u_j) \cdot \underbrace{\int_\Omega \phi_j \nabla \phi_i d\Omega}_{A_{ij}} + \sum_i (\alpha u_j) \cdot \underbrace{\int_\Omega \nabla \phi_j \nabla \phi_i d\Omega}_{D_{ij}}$$
$$= - \oint_{\partial\Omega} \phi_i (\mathbf{v} u_j) \cdot d\mathbf{S} + \oint_{\partial\Omega} \phi_i (\alpha \nabla u_j) \cdot d\mathbf{S} + \int_\Omega \phi_i q^u \, d\Omega \tag{7.21}$$

where Ω_j is the subdomain containing node j. The summations extends over all elements containing this node. By this means the original partial differential equation was transformed to an ordinary differential equation. In this representation we have the so-called **mass matrix**

$$C_{ij} = \int_\Omega \phi_i \phi_j \, d\Omega \tag{7.22}$$

which is a symmetric but not diagonal one and the so-called **stiffness matrix**

$$A_{ij} = \int_\Omega \phi_i \nabla \phi_j \, d\Omega \tag{7.23}$$

which is no longer symmetric. Note for linear problems, both mass and stiffness matrices are independent of the unknown function u. They will depend then only on the element geometry

$$D_{ij} = \int_\Omega \nabla \phi_i \nabla \phi_j \, d\Omega \tag{7.24}$$

We have seen the application of the method of weighted residuals to differential balance equations leads to several matrix-integral expressions that must be evaluated for each finite element. To calculate these expressions we have to know the interpolation functions and their derivatives as functions of coordinate directions.

7.3 Interpolation and Shape Functions

The general idea is to approximate the unknown field variables by linear combinations of defined **interpolation functions** ϕ_i (also called basis or trial functions) and the unknown nodal values of the dependent variable u_i

$$u(t, x) \approx \hat{u}(t, x) = \sum_i u_i(t) \phi_i(x) \tag{7.25}$$

where $u_i(t) = \hat{u}(t, x_i)$ and the summation extends over all nodes of the finite element mesh. The interpolation functions are attached to each node and, therefore, corresponds to the degree of freedom of the discrete system.

As written above, here are several possibilities to define basis functions leading to different methods:

- Collocation or spectral methods: definition of interpolation functions on the whole computation domain (e.g. by trigonometric functions, Legendre or Chebyshev polynomials, splines)
- Standard FEM is based on local interpolation, i.e. interpolation functions are chosen to be locally defined polynomials within each element and being zero outside the considered element. Polynomials are popular because they are simple functions to manipulate mathematically.

7.3.1 General Method for Shape Function Derivation

The general method for derivation of shape functions consists of the following three steps:

1. Polynomial representation of the trial function

$$\hat{\mathbf{u}}(x, y, z) = \mathbf{P}^T(x, y, z)\,\mathbf{a} \tag{7.26}$$

with polynomial $\mathbf{P}$ and general parameter vector $\mathbf{a}$

$$\mathbf{P}^T(x, y, z) = \{1, x, y, z, x^2, xy, xz, y^2, yz, z^2, ...\} \quad , \quad \mathbf{a} = \begin{bmatrix} a_1 \\ a_2 \\ ... \\ a_n \end{bmatrix} \tag{7.27}$$

For all grid points (i.e. nodal degrees of freedom) we have

$$\begin{Bmatrix} u_1 \\ u_2 \\ ... \\ u_m \end{Bmatrix} =$$

$$\begin{bmatrix} 1 & x_1 & y_1 & z_1 & x_1^2 & x_1y_1 & x_1z_1 & y_1^2 & y_1z_1 & z_1^2 & ... \\ 1 & x_2 & y_2 & z_2 & x_2^2 & x_2y_2 & x_2z_2 & y_2^2 & y_2z_2 & z_2^2 & ... \\ ... & ... & ... & ... & ... & ... & ... & ... & ... & ... & ... \\ 1 & x_m & y_m & z_m & x_m^2 & x_my_m & x_mz_m & y_m^2 & y_mz_m & z_m^2 & ... \end{bmatrix} \begin{Bmatrix} a_1 \\ a_2 \\ ... \\ a_n \end{Bmatrix}$$

$$u_i = P_{ij}\, a_j \quad , \quad i = 1, .., m \quad j = 1, .., n$$

2. General parameter vector

Inversion of the above equation gives

$$a_i = P_{ij}^{-1}\, u_j \tag{7.28}$$

3. General shape function

Combining the above equations we obtain

$$\hat{\mathbf{u}}(x,y,z) = \underbrace{P_i^T(x,y,z)P_{ij}^{-1}}_{\mathbf{N}^T}\, u_j \tag{7.29}$$

7.3.2 Shape Function Conditions

The shape functions locally satisfy the following condition

$$\phi^{(e)}(x_i) = \begin{cases} 1 & : \quad x = x_i \\ 0 & : \quad x \in (e) \end{cases} \qquad , \qquad \phi_i^{(e)}(x_j) = \delta_{ij} \tag{7.30}$$

Additionally, the shape functions have to satisfy the following condition

$$\sum_i \phi_i^{(e)}(x) = 1, \quad \forall x \in (e) \tag{7.31}$$

The above equations define the properties of local polynomial interpolation functions for finite element approximations. Different classes of interpolation functions can be constructed according to their inter-element continuity conditions. The most popular are Lagrangian and Hermitian elements. If nodal values of the dependent field variable are defined by the values of the unknown functions, C^0-continuity at the inter-element boundary is sufficient. These elements, which are called **Lagrangian elements**, are appropriate to describe differential equations of order no higher than two. If continuity of first-order derivatives (i.e. C^1-continuity) is required at the inter-element boundaries, we need so-called **Hermitian elements**. In this case, first-order derivatives of the unknown functions are to be considered as additional degrees of freedom of the discrete system.

In the following we show an approach to the construction of interpolation functions for various element types. Due to its specific properties for different element shapes those functions are denoted as **element shape functions**. For the construction of shape functions we have to consider factors such as function completeness, domain geometry, solution accuracy, and computation effort. In general, several function types are possible to this purpose (e.g. trigonometric functions). As noted above polynomials are our favor to represent unknown variables within finite elements.

7.3.3 Isoparametric Mapping

The ability to handle non-uniform and distorted computational domains is an important outline of the finite element method. This feature is based on specific mapping methods.

$$x(r,s,t) = \sum_i N_i(r,s,t)x_i \tag{7.32}$$

where (x, y, z) are physical (also denoted as global) coordinates, (r, s, t) are virtual (also denoted as natural or local) coordinates, and N_i are shape functions, describing the shape of the element. The summation extends over all nodes of the element. This mapping rule performs a geometric transformation of the element between physical and virtual spaces.

We distinguish between subparametric, isoparametric, and superparametric mapping in dependence of the degrees of shape and interpolation functions.

Subparametric mapping	Isoparametric mapping	Superparametric mapping
$\deg(N) < \deg(\phi)$	$\deg(N) = \deg(\phi)$	$\deg(N) > \deg(\phi)$

In next sections we see the use of parametric mapping has particular benefit for the evaluation of integral equations.

7.4 Element Shape Functions

7.4.1 1-D Linear Bar Elements

Shape functions

A simple approximation of the unknown function can be obtained by linear approximation.

$$\hat{u}(x) = a_1 + a_2 x \tag{7.33}$$

Due to the C^0-continuity condition we have at the nodes

$$\begin{aligned} u_1 &= a_1 + a_2 x_1 \\ u_2 &= a_1 + a_2 x_2 \end{aligned} \tag{7.34}$$

or written compactly

$$\left\{ \begin{array}{c} u_1 \\ u_2 \end{array} \right\} = \left[\begin{array}{cc} 1 & x_1 \\ 1 & x_2 \end{array} \right] \left\{ \begin{array}{c} a_1 \\ a_2 \end{array} \right\} \tag{7.35}$$

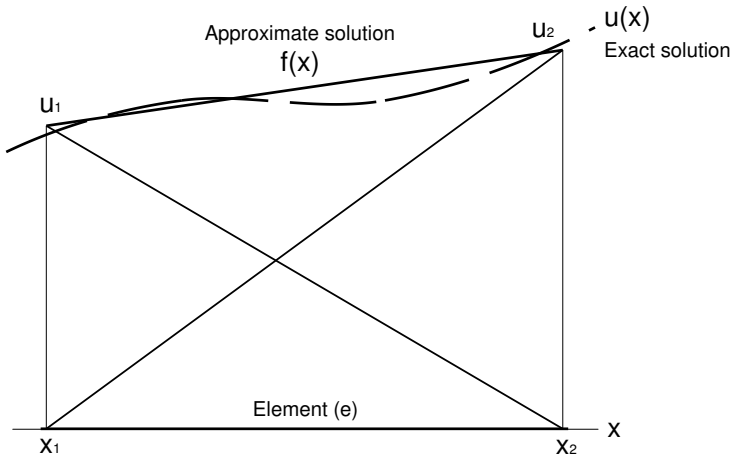

Figure 7.4: Approximate solution by use of 1-D linear shape functions

By inversion of the above matrix equation we can derive relations for the coefficients

$$\left\{ \begin{array}{c} a_1 \\ a_2 \end{array} \right\} = \frac{1}{x_2 - x_1} \left[\begin{array}{cc} x_2 & -x_1 \\ -1 & 1 \end{array} \right] \left\{ \begin{array}{c} u_1 \\ u_2 \end{array} \right\} \tag{7.36}$$

Inserting the relations for the coefficients a_i into equation (7.33) this can be rearranged

$$\hat{u}(x) = \frac{x_2 - x}{x_2 - x_1} u_1 + \frac{x - x_1}{x_2 - x_1} u_2 = N_1(x) u_1 + N_2(x) u_2 \tag{7.37}$$

and rewritten as a trial solution. Therefore, we derived following element-related interpolation functions (i.e. shape functions) for a one-dimensional linear element fulfilling the conditions to shape functions.

$$N_1^{(e)}(x) = \frac{x_2 - x}{x_2 - x_1} \quad , \quad N_2^{(e)}(x) = \frac{x - x_1}{x_2 - x_1} \tag{7.38}$$

By this case we construct a piece-wise linear interpolation of the unknown function (Fig. 7.4).

In Fig. 7.5 we see two connected 1-D elements. Both elements have node i in common. The shape functions of these elements are defined in physical (global) coordinates by

$$\begin{aligned} N_1^{(1)}(x) &= \frac{x - x_1}{x_2 - x_1} \quad , \quad N_2^{(1)}(x) = \frac{x_2 - x}{x_2 - x_1} \\ N_1^{(2)}(x) &= \frac{x - x_2}{x_3 - x_2} \quad , \quad N_2^{(2)}(x) = \frac{x_3 - x}{x_3 - x_2} \end{aligned} \tag{7.39}$$

Note, the shape functions in local coordinates depend only on the structure of the element and the number of nodal points. The above shape functions define a normalized, universal element.

By use of linear Lagrangian elements, the unknown function $u(x)$ is approximated on element 1 by

$$\hat{u}(x) = N_1^{(1)}(x)u_{i-1} + N_2^{(1)}(x)u_i = u_{i-1} + \frac{x - x_{i-1}}{x_i - x_{i-1}}(u_i - u_{i-1}) \tag{7.40}$$

and on element 2, respectively

$$\hat{u}(x) = N_1^{(2)}(x)u_i + N_2^{(2)}(x)u_{i+1} = u_i + \frac{x - x_i}{x_{i+1} - x_i}(u_{i+1} - u_i) \tag{7.41}$$

Derivatives

The spatial derivative of u is then approximated by

$$\left[\frac{\partial \hat{u}(x)}{\partial x}\right]^{(e)} = \sum_{i=1}^{2} \frac{\partial N_i^{(e)}}{\partial x} u_i = -\frac{u_i - u_{i-1}}{x_i - x_{i-1}} + \frac{u_{i+1} - u_i}{x_{i+1} - x_i} \tag{7.42}$$

and the nodal values at point x_i from the different elements are

$$\left[\frac{\partial \hat{u}(x)}{\partial x}\right]_i^{(1)} = \frac{u_i - u_{i-1}}{x_i - x_{i-1}} \quad , \quad \left[\frac{\partial \hat{u}(x)}{\partial x}\right]_i^{(2)} = \frac{u_{i+1} - u_i}{x_{i+1} - x_i} \tag{7.43}$$

which correspond to first-order backward and forward difference schemes, respectively. Note, the approximate derivatives are not continuous at element boundary, there exists only C^0 continuity. To define a local derivative, we average the individual contributions of the connected elements. If an arithmetic average is taken, we have

$$\left[\frac{\partial \hat{u}(x)}{\partial x}\right]_i^{(e)} = \frac{1}{2}\left(\frac{u_{i+1} - u_i}{x_{i+1} - x_i} - \frac{u_i - u_{i-1}}{x_i - x_{i-1}}\right) \tag{7.44}$$

It is note worthy, that the above local approximation is second-order accurate in space but only on equidistant meshes. However, if we use a weighted average according to

$$\left[\frac{\partial \hat{u}(x)}{\partial x}\right]_i^{(e)} = \frac{1}{x_{i+1} - x_{i-1}}\left((x_{i+1} - x_i)\left[\frac{\partial \hat{u}(x)}{\partial x}\right]_i^{(1)} + (x_i - x_{i-1})\left[\frac{\partial \hat{u}(x)}{\partial x}\right]_i^{(2)}\right)$$

we obtain a central scheme

$$\left[\frac{\partial \hat{u}(x)}{\partial x}\right]_i^{(e)} = \frac{1}{x_{i+1} - x_{i-1}}\left(\frac{x_i - x_{i-1}}{x_{i+1} - x_i}(u_{i+1} - u_i) + \frac{x_{i+1} - x_i}{x_i - x_{i-1}}(u_i - u_{i-1})\right) \tag{7.45}$$

which is second-order accurate in space on arbitrary meshes.

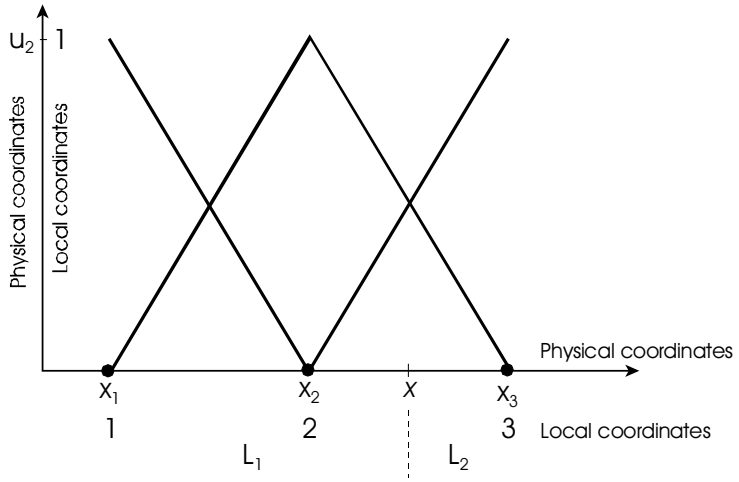

Figure 7.5: 1-D linear elements in local and physical coordinates

Isoparametric mapping

We assume that the coordinate transformation $(x) \rightarrow (r)$ can be conducted by a linear function.

$$x(r) = a_1 + a_2 r \tag{7.46}$$

Additional we require

$$x(r = -1) = x_1 \quad , \quad x(r = +1) = x_2 \tag{7.47}$$

i.e. we use an unified element in local (natural) coordinates as shown in Fig. 7.6. In other words, by this mapping function we convert the $[x_1, x_2]$-space into normalized $[-1, 1]$-space.

Performing the same procedure as in Chapter 7.4.1 we obtain the following relations

$$x(r) = \frac{1-r}{2} x_1 + \frac{r-1}{2} x_2 = N_1(r) x_1 + N_2(r) x_2 \tag{7.48}$$

Therefore, we can write the coordinate transformation by use of shape functions.

$$N_1^{(e)}(r) = \frac{1-r}{2} \quad , \quad N_2^{(e)}(r) = \frac{r-1}{2} \tag{7.49}$$

This means we interpolate the coordinations in the same manner as the unknown function u. We see in local coordinates these shape functions take a simple form and they are independently on physical coordinates. This coordinate transformation allows us to write the trial solution for the unknown function u in terms of local coordinates.

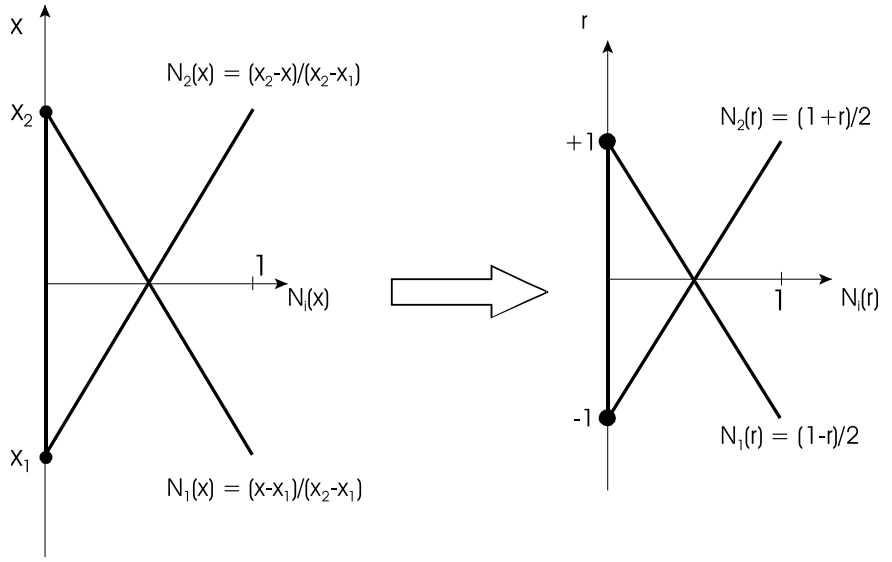

Figure 7.6: Isoparametric coordinate transformation between physical and local coordinates

7.4.2 1-D Quadratic Elements

In general we used Lagrangian polynomials to obtain shape functions. For an one-dimensional element consisting of m nodes we can write the following Lagrangian polynomial.

$$N_i = \prod_{i=1, j\neq i}^{m} \left[\frac{x - x_j}{x_i - x_j}\right] = \frac{(x-x_1)(x-x_2)...(x-x_{i-1})(x-x_{i+1})...(x-x_m)}{(x_i-x_1)(x_i-x_2)...(x_i-x_{i-1})(x_i-x_{i+1})...(x_i-x_m)}$$

The use of linear interpolation functions imposes a constraint on the approximate solution that it must vary linearly between the nodal points. Quadratic interpolation functions are required if linear interpolation failed to approximate the unknown function, e.g. if continuity of first-order derivatives is required. Moreover, it is expected that a smaller interpolation error will be occur if quadratic interpolation is used for the same grid instead of linear one.

Use of three-node elements leads to a quadratic shape functions

$$\begin{aligned} N_1^{(e)}(x) &= \frac{(x-x_2)(x-x_3)}{(x_1-x_2)(x_1-x_3)} \\ N_2^{(e)}(x) &= \frac{(x-x_1)(x-x_3)}{(x_2-x_1)(x_2-x_3)} \\ N_3^{(e)}(x) &= \frac{(x-x_1)(x-x_2)}{(x_3-x_2)(x_3-x_1)} \end{aligned} \tag{7.50}$$

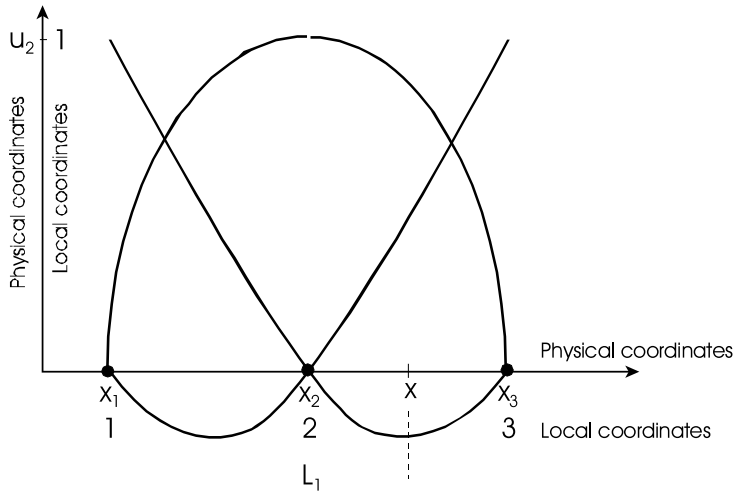

Figure 7.7: 1-D quadratic element in local and physical coordinates

7.4.3 2-D Triangular Elements

We consider a triangular element in physical space (x, y) as shown in Fig. 7.8.

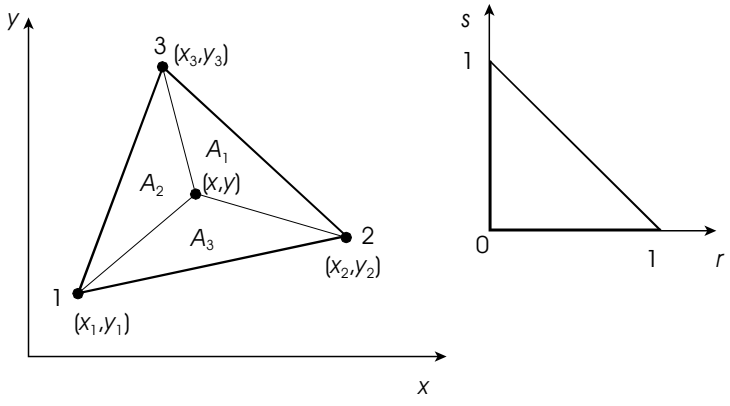

Figure 7.8: Linear triangular element

Local coordinates

Obviously, an arbitrary point in the triangle can be identified by use of the following local (area) coordinates (Fig. 7.8),

$$\begin{aligned} L_1 &= A_1/A \\ L_2 &= A_2/A \\ L_3 &= A_3/A \end{aligned} \tag{7.51}$$

where A is the area of triangular element

$$A = \frac{1}{2}\begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix} \tag{7.52}$$

From geometrical reasons we have

$$1 = L_1 + L_2 + L_3 \tag{7.53}$$

Furthermore we can write

$$L_i = \begin{cases} 1 & : \text{ at node i} \\ 0 & : \text{ at remaining nodes} \end{cases} \tag{7.54}$$

From this condition it can be concluded that

$$\begin{aligned} x &= L_1 x_1 + L_2 x_2 + L_3 x_3 \\ y &= L_1 y_1 + L_2 y_2 + L_3 y_3 \end{aligned} \tag{7.55}$$

Therefore, L_1, L_2 and L_3 can be used as a kind of local normalized coordinates. i.e. the physical coordinates of an arbitrary point within the triangle are given by use of the above local coordinates as above.

Now we write the above equations in following compact matrix form

$$\begin{Bmatrix} 1 \\ x \\ y \end{Bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \end{bmatrix} \begin{Bmatrix} L_1 \\ L_2 \\ L_3 \end{Bmatrix} \tag{7.56}$$

Inversion gives

$$\begin{Bmatrix} L_1 \\ L_2 \\ L_3 \end{Bmatrix} = \frac{1}{2A} \begin{bmatrix} x_2y_3 - x_3y_2 & y_2 - y_3 & x_3 - x_2 \\ x_3y_1 - x_1y_3 & y_3 - y_1 & x_1 - x_3 \\ x_1y_2 - x_2y_1 & y_1 - y_2 & x_2 - x_1 \end{bmatrix} \begin{Bmatrix} 1 \\ x \\ y \end{Bmatrix} \tag{7.57}$$

Shape functions

We approximate the unknown function $u(x, y)$ at the triangular element by a linear approximation.

$$\hat{u}(x, y) = a_1 + a_2 x + a_3 y \tag{7.58}$$

At the corner nodes we require that

$$\begin{aligned} u_1 &= a_1 + a_2 x_1 + a_3 y_1 \\ u_2 &= a_1 + a_2 x_2 + a_3 y_2 \\ u_3 &= a_1 + a_2 x_3 + a_3 y_3 \end{aligned} \tag{7.59}$$

or written compactly

$$\left\{ \begin{array}{c} u_1 \\ u_2 \\ u_3 \end{array} \right\} = \left[\begin{array}{ccc} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{array} \right] \left\{ \begin{array}{c} a_1 \\ a_2 \\ a_3 \end{array} \right\} \tag{7.60}$$

By inversion of the above matrix equation we can derive relations for the determination of the unknown interpolation coefficients

$$\left\{ \begin{array}{c} a_1 \\ a_2 \\ a_3 \end{array} \right\} = \frac{1}{2A} \left[\begin{array}{ccc} x_2 y_3 - x_3 y_2 & x_3 y_1 - x_1 y_3 & x_1 y_2 - x_2 y_1 \\ y_2 - y_3 & y_3 - y_1 & y_1 - y_2 \\ x_3 - x_2 & x_1 - x_3 & x_2 - x_1 \end{array} \right] \left\{ \begin{array}{c} u_1 \\ u_2 \\ u_3 \end{array} \right\} \tag{7.61}$$

Inserting the above relations for the coefficients a_i into equation (7.58) this can be rearranged and rewritten as a trial solution for the element

$$\hat{u}(x, y) = N_1(x, y) u_1 + N_2(x, y) u_2 + N_3(x, y) u_3 \tag{7.62}$$

By this way, we derive following element-related interpolation functions (i.e. shape functions) for a linear triangular element.

$$\begin{aligned} N_1(x, y) &= \frac{1}{2A} \left[(x_2 y_3 - x_3 y_2) + (y_2 - y_3) x + (x_3 - x_2) y \right] \\ N_2(x, y) &= \frac{1}{2A} \left[(x_3 y_1 - x_1 y_3) + (y_3 - y_1) x + (x_1 - x_3) y \right] \\ N_2(x, y) &= \frac{1}{2A} \left[(x_1 y_2 - x_2 y_1) + (y_1 - y_2) x + (x_2 - x_1) y \right] \end{aligned} \tag{7.63}$$

or written in a compact matrix form

$$\left\{ \begin{array}{c} N_1 \\ N_2 \\ N_3 \end{array} \right\} = \frac{1}{2A} \left[\begin{array}{ccc} x_2 y_3 - x_3 y_2 & y_2 - y_3 & x_3 - x_2 \\ x_3 y_1 - x_1 y_3 & y_3 - y_1 & x_1 - x_3 \\ x_1 y_2 - x_2 y_1 & y_1 - y_2 & x_2 - x_1 \end{array} \right] \left\{ \begin{array}{c} 1 \\ x \\ y \end{array} \right\} \tag{7.64}$$

Comparing equation (7.81) with the above one, we see the shape functions N_i are simply the area coordinates L_i, i.e. we have isoparametric elements.

Now the derivatives of the shape functions can be easily written down.

$$\frac{\partial \mathbf{N}}{\partial x} = \left\{ \begin{array}{l} \dfrac{\partial N_1}{\partial x} = \dfrac{y_2 - y_3}{2A} \\ \dfrac{\partial N_2}{\partial x} = \dfrac{y_3 - y_1}{2A} \\ \dfrac{\partial N_3}{\partial x} = \dfrac{y_1 - y_2}{2A} \end{array} \right\} \qquad \frac{\partial \mathbf{N}}{\partial y} = \left\{ \begin{array}{l} \dfrac{\partial N_1}{\partial y} = \dfrac{x_3 - x_2}{2A} \\ \dfrac{\partial N_2}{\partial y} = \dfrac{x_1 - x_3}{2A} \\ \dfrac{\partial N_3}{\partial y} = \dfrac{x_2 - x_1}{2A} \end{array} \right\} \tag{7.65}$$

Integration

When element matrices have to be evaluated we are faced with integration of quantities defined in terms of area coordinates. There is the following exact integration expression for linear triangular elements.

$$\iint_{\Delta} L_1^a L_2^b L_3^c dx dy = 2\Delta \frac{a!b!c!}{(a+b+c+2)!} \tag{7.66}$$

7.4.4 2-D Bilinear Quadrilateral Elements

Local coordinates

Again we want to approximate the unknown function by a trial solution based on four nodes of a quadrilateral element (Fig. 7.9)

$$\hat{u}(x) = \sum_{i=1}^{4} N_i(x,y) u_i \tag{7.67}$$

A typical situation, illustrating the advantage of isoparametric mapping is given in following figure. The above considered element is distorted in the physical space (x, y). However, these elements can easily transformed into uniform elements in a virtual space (ξ, η) by using mapping functions according to a Lagrangian interpolation.

Shape functions

For the element interpolation we can use the product of first-degree Lagrangian polynomials

$$N_i(x,y) = L_i(x) L_i(y) = \frac{x - x_i}{x_i - x_{i+1}} \frac{y - y_i}{y_i - y_{i-1}} \tag{7.68}$$

It is most convenient to define shape functions in so-called natural (unified) coordinates (Fig. 7.9) and then transform the relationship into the physical coordinates. The set of shape functions becomes then

$$
\begin{aligned}
N_1(r,s) &= \frac{1}{4}(1+r)(1+s) \\
N_2(r,s) &= \frac{1}{4}(1-r)(1+s) \\
N_3(r,s) &= \frac{1}{4}(1-r)(1-s) \\
N_4(r,s) &= \frac{1}{4}(1+r)(1-s)
\end{aligned}
\tag{7.69}
$$

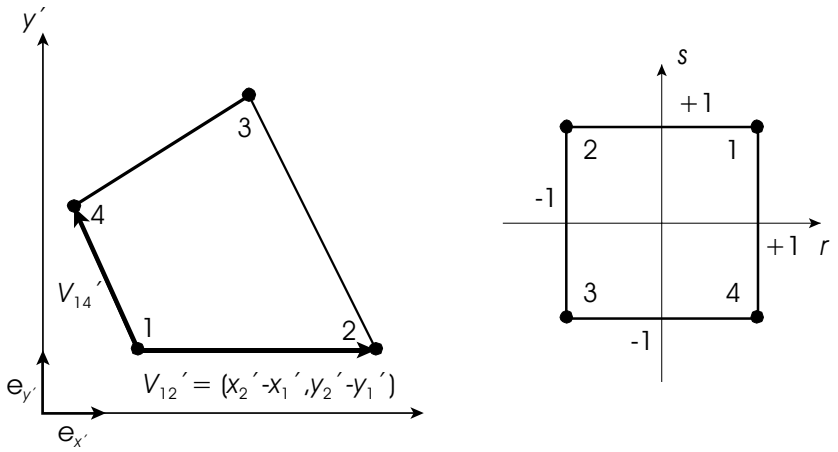

Figure 7.9: Isoparametric mapping of bilinear quadrilateral elements

To approximate derivatives of the unknown function we need to have the derivatives of the interpolation function. To this purpose we have to determine the inverse Jacobian matrix.

$$
\nabla_{xy}\mathbf{N} = [J_{2D}]^{-1}\nabla_{rs}\mathbf{N} \tag{7.70}
$$

with

$$
[J_{2D}]^{-1} = \begin{bmatrix} \dfrac{\partial r}{\partial x} & \dfrac{\partial s}{\partial x} \\ \dfrac{\partial r}{\partial y} & \dfrac{\partial s}{\partial y} \end{bmatrix} \tag{7.71}
$$

$$\nabla_{xy}\mathbf{N} = \begin{bmatrix} \dfrac{\partial N_1}{\partial x} & \dfrac{\partial N_1}{\partial y} \\ \dfrac{\partial N_2}{\partial x} & \dfrac{\partial N_2}{\partial y} \\ \dfrac{\partial N_3}{\partial x} & \dfrac{\partial N_3}{\partial y} \\ \dfrac{\partial N_4}{\partial x} & \dfrac{\partial N_4}{\partial y} \end{bmatrix} \quad , \quad \nabla_{rs}\mathbf{N} = \begin{bmatrix} \dfrac{\partial N_1}{\partial r} & \dfrac{\partial N_1}{\partial s} \\ \dfrac{\partial N_2}{\partial r} & \dfrac{\partial N_2}{\partial s} \\ \dfrac{\partial N_3}{\partial r} & \dfrac{\partial N_3}{\partial s} \\ \dfrac{\partial N_4}{\partial r} & \dfrac{\partial N_4}{\partial s} \end{bmatrix} \tag{7.72}$$

In general we can use the above interpolation function (7.67) to transform coordinates too

$$\begin{aligned} x = x(r,s) &= \sum_{i=1}^{4} N_i(r,s)\, x_i \\ y = y(r,s) &= \sum_{i=1}^{4} N_i(r,s)\, y_i \end{aligned} \tag{7.73}$$

Then we speak about an isoparametric coordinate transformation. In this case the two-dimensional Jacobian matrix is given by

$$[J_{2D}] = \begin{bmatrix} \dfrac{N_i}{\partial r} x_i & \dfrac{N_i}{\partial r} y_i \\ \dfrac{N_i}{\partial s} x_i & \dfrac{N_i}{\partial s} y_i \end{bmatrix} = [\nabla_{rs}\mathbf{N}]\,[A_{2D}] \tag{7.74}$$

Finally we determine $[J_{2D}]^{-1}$ by inversion of the Jacobian.

7.4.5 3-D Tetrahedral Elements

We consider a tetrahedral element in 3-D physical space (x, y, z) as shown in Fig. 7.10. Faces of the tetrahedral element are triangles. In the following we will see, properties of a tetrahedron are very similar to that of triangular elements.

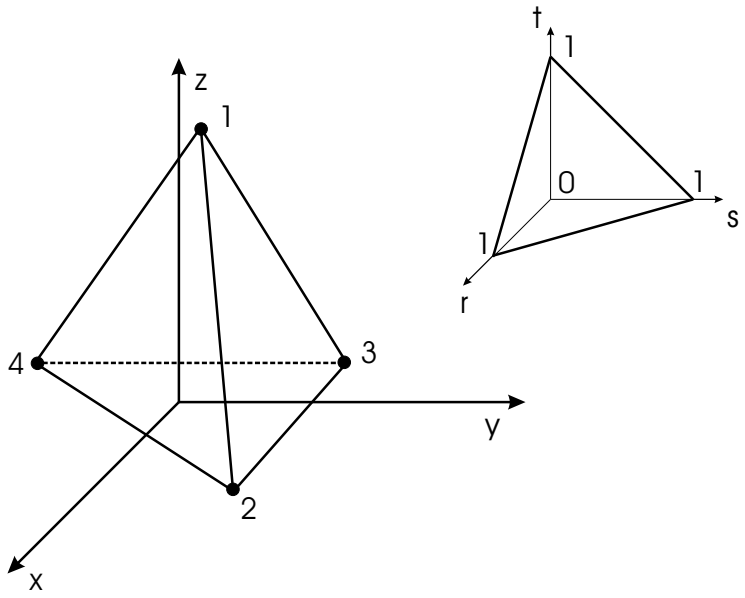

Figure 7.10: Linear tetrahedral element

Local coordinates

Obviously, an arbitrary point in the tetrahedron can be identified by use of the following local (volumetric) coordinates (Fig. 7.10),

$$\begin{aligned} L_1 &= V_1/V \\ L_2 &= V_2/V \\ L_3 &= V_3/V \\ L_4 &= V_4/V \end{aligned} \tag{7.75}$$

where V is the volume of tetrahedral element

$$V = \frac{1}{6}\begin{vmatrix} x_1 & y_1 & z_1 & 1 \\ x_2 & y_2 & z_2 & 1 \\ x_3 & y_3 & z_3 & 1 \\ x_4 & y_4 & z_4 & 1 \end{vmatrix} = \frac{1}{6}\begin{vmatrix} 1 & 1 & 1 & 1 \\ x_1 & x_2 & x_3 & x_4 \\ y_1 & y_2 & y_3 & y_4 \\ z_1 & z_2 & z_3 & z_4 \end{vmatrix} \tag{7.76}$$

For geometrical reasons we have

$$1 = L_1 + L_2 + L_3 + L_4 \tag{7.77}$$

Furthermore we can write

$$L_i \quad = \quad \begin{cases} 1 & : \text{ at node i} \\ 0 & : \text{ at remaining nodes} \end{cases} \tag{7.78}$$

From this condition it can be concluded that

$$\begin{aligned} x &= L_1 x_1 + L_2 x_2 + L_3 x_3 + L_4 x_4 \\ y &= L_1 y_1 + L_2 y_2 + L_3 y_3 + L_4 y_4 \\ z &= L_1 z_1 + L_2 z_2 + L_3 z_3 + L_4 z_4 \end{aligned} \tag{7.79}$$

Therefore, L_1, L_2, L_3 and L_4 can be used as a kind of local normalized coordinates. i.e. the physical coordinates of an arbitrary point within the tetrahedron are given by use of the above local coordinates as above.

Now we write the above equations in following compact matrix form

$$\begin{Bmatrix} 1 \\ x \\ y \\ z \end{Bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ x_1 & x_2 & x_3 & x_4 \\ y_1 & y_2 & y_3 & y_4 \\ z_1 & z_2 & z_3 & z_4 \end{bmatrix} \begin{Bmatrix} L_1 \\ L_2 \\ L_3 \\ L_4 \end{Bmatrix} \tag{7.80}$$

Inversion gives

$$\begin{Bmatrix} L_1 \\ L_2 \\ L_3 \\ L_4 \end{Bmatrix} = \frac{1}{6V} \begin{bmatrix} a_1 & b_1 & c_1 & d_1 \\ a_2 & b_2 & c_2 & d_2 \\ a_3 & b_3 & c_3 & d_3 \\ a_4 & b_4 & c_4 & d_4 \end{bmatrix} \begin{Bmatrix} 1 \\ x \\ y \\ z \end{Bmatrix} \tag{7.81}$$

with

$$\begin{aligned} a_i &= \begin{vmatrix} x_{i+1} & y_{i+1} & z_{i+1} \\ x_{i+2} & y_{i+2} & z_{i+2} \\ x_{i+3} & y_{i+3} & z_{i+3} \end{vmatrix}, \quad -b_i = \begin{vmatrix} 1 & y_{i+1} & z_{i+1} \\ 1 & y_{i+2} & z_{i+2} \\ 1 & y_{i+3} & z_{i+3} \end{vmatrix} \\ c_i &= \begin{vmatrix} x_{i+1} & 1 & z_{i+1} \\ x_{i+2} & 1 & z_{i+2} \\ x_{i+3} & 1 & z_{i+3} \end{vmatrix}, \quad d_i = \begin{vmatrix} x_{i+1} & y_{i+1} & 1 \\ x_{i+2} & y_{i+2} & 1 \\ x_{i+2} & y_{i+3} & 1 \end{vmatrix} \end{aligned} \tag{7.82}$$

where i is running in a cyclic permutation $\{1, 2, 3, 4\}$.

Shape functions

Following the same procedure as described for the triangular element (sec 7.4.3), shape functions and local coordinates are identical.

$$\begin{Bmatrix} N_1 \\ N_2 \\ N_3 \\ N_4 \end{Bmatrix} \equiv \begin{Bmatrix} L_1 \\ L_2 \\ L_3 \\ L_4 \end{Bmatrix} = \frac{1}{6V} \begin{bmatrix} a_1 & b_1 & c_1 & d_1 \\ a_2 & b_2 & c_2 & d_2 \\ a_3 & b_3 & c_3 & d_3 \\ a_4 & b_4 & c_4 & d_4 \end{bmatrix} \begin{Bmatrix} 1 \\ x \\ y \\ z \end{Bmatrix} \tag{7.83}$$

Derivatives of shape functions are

$$\frac{\partial N_i}{\partial x} = \frac{b_i}{6V} \quad , \quad \frac{\partial N_i}{\partial y} = \frac{c_i}{6V} \quad , \quad \frac{\partial N_i}{\partial z} = \frac{d_i}{6V} \tag{7.84}$$

Integration

When element matrices have to be evaluated we are faced with integration of quantities defined in terms of area coordinates. There is the following exact integration expression for linear tetrahedral elements.

$$\iiint_V L_1^a L_2^b L_3^c L_4^d \, dxdydz = 6V \frac{a!b!c!d!}{(a+b+c+d+3)!} \tag{7.85}$$

7.4.6 3-D Triangular Prismatic Elements

We consider a triangular prism element in 3-D space (x, y, z) as shown in Fig. 7.11. This element is a combination of triangles and quadrilaterals. Those elements have many practical applications, e.g. if 2-D discretizations have to be extended into three dimensions.

Local coordinates

We use local (area) coordinates for triangles.

$$\begin{aligned} L_1 &= N_1^\Delta = A_1/A \\ L_2 &= N_2^\Delta = A_2/A \\ L_3 &= N_3^\Delta = A_3/A \end{aligned} \tag{7.86}$$

where A is the area of triangular element basis.

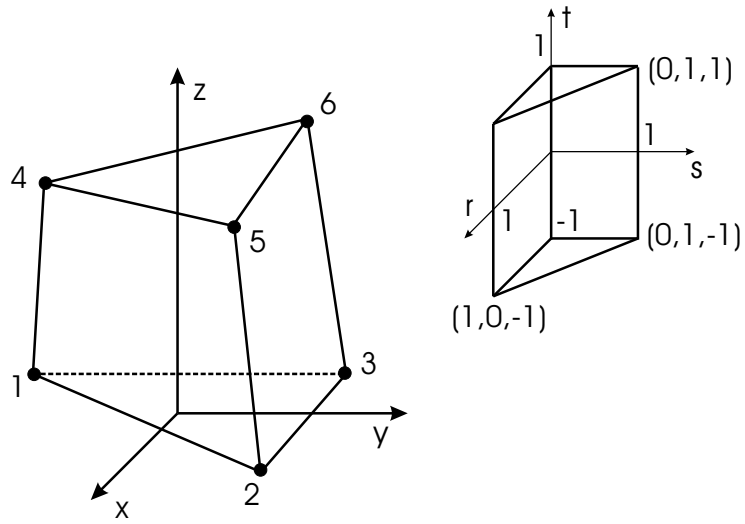

Figure 7.11: Triangular prismatic element

Shape functions

Shape functions of triangle based prism elements can be derived by superposition of shape functions for triangles and line elements.

$$\begin{aligned} N_1 &= N_1^{\Delta}(1-t) \quad , \quad N_2 = N_2^{\Delta}(1-t) \quad , \quad N_3 = N_3^{\Delta}(1-t) \\ N_4 &= N_1^{\Delta}(1+t) \quad , \quad N_5 = N_2^{\Delta}(1+t) \quad , \quad N_6 = N_3^{\Delta}(1+t) \end{aligned} \tag{7.87}$$

Derivatives of shape functions are

$$\nabla \mathbf{N} = \begin{Bmatrix} \frac{\partial N_1}{\partial x} & \frac{\partial N_1}{\partial y} & \frac{\partial N_1}{\partial z} \\ \frac{\partial N_2}{\partial x} & \frac{\partial N_2}{\partial y} & \frac{\partial N_2}{\partial z} \\ \frac{\partial N_3}{\partial x} & \frac{\partial N_3}{\partial y} & \frac{\partial N_3}{\partial z} \\ \frac{\partial N_4}{\partial x} & \frac{\partial N_4}{\partial y} & \frac{\partial N_4}{\partial z} \\ \frac{\partial N_5}{\partial x} & \frac{\partial N_5}{\partial y} & \frac{\partial N_5}{\partial z} \\ \frac{\partial N_6}{\partial x} & \frac{\partial N_6}{\partial y} & \frac{\partial N_6}{\partial z} \end{Bmatrix} = \begin{bmatrix} \frac{\partial N_1^{\Delta}}{\partial x}(1-t) & \frac{\partial N_1^{\Delta}}{\partial y}(1-t) & -N_1^{\Delta} J_z \\ \frac{\partial N_2^{\Delta}}{\partial x}(1-t) & \frac{\partial N_2^{\Delta}}{\partial y}(1-t) & -N_2^{\Delta} J_z \\ \frac{\partial N_3^{\Delta}}{\partial x}(1-t) & \frac{\partial N_3^{\Delta}}{\partial y}(1-t) & -N_3^{\Delta} J_z \\ \frac{\partial N_1^{\Delta}}{\partial x}(1+t) & \frac{\partial N_1^{\Delta}}{\partial y}(1+t) & N_1^{\Delta} J_z \\ \frac{\partial N_2^{\Delta}}{\partial x}(1+t) & \frac{\partial N_2^{\Delta}}{\partial y}(1+t) & N_2^{\Delta} J_z \\ \frac{\partial N_3^{\Delta}}{\partial x}(1+t) & \frac{\partial N_3^{\Delta}}{\partial y}(1+t) & N_3^{\Delta} J_z \end{bmatrix} \tag{7.88}$$

with N_i^{Δ} the 2-D shape functions of triangular elements (7.83) and the Jacobian

for line elements

$$J_z = \frac{dt}{dz} = \frac{2A}{A_1|\mathbf{z}_{14}| + A_2|\mathbf{z}_{25}| + A_3|\mathbf{z}_{36}|} \\ \frac{2}{\Delta z} \tag{7.89}$$

for prismatic elements.

Integration

When element matrices have to be evaluated we are faced with integration of quantities defined in terms of area coordinates. Ratke (1996) outlined the following procedure.

$$\begin{aligned} \iiint_V f_1(N_1^\Delta, N_2^\Delta, N_3^\Delta) f_2(N_z) dV &= \iint_\Delta f_1 \left(\int_{z_u}^{z_o} f_2 dz \right) dA \\ &= \iint_\Delta f_1 \, J_z^{-1} \left(\int_{-1}^{+1} f_2 dt \right) dA \\ &= \iint_\Delta f_1 \, J_z^{-1} dA \times \int_{-1}^{+1} f_2 \, dt \end{aligned} \tag{7.90}$$

Now we can use the integration formula (7.66) for triangular integrals.

7.4.7 3-D Hexahedral Element

We consider a hexahedral element in 3-D space (x, y, z) as shown in Fig. 7.12. It is a straightforward generalization of the 2-D bilinear quadrilateral element presented in section 7.4.4.

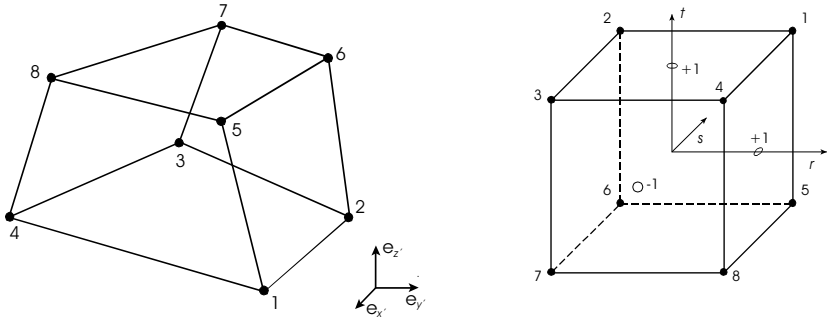

Figure 7.12: Hexahedron in global and local coordinates

Local coordinates

We use isoparametric mapping for coordinate transformation between global and local coordinates (Fig. 7.12).

$$\begin{Bmatrix} x \\ y \\ z \end{Bmatrix} = \sum_{i=1}^{8} N_i(r,s,t) \begin{Bmatrix} x_i \\ y_i \\ z_i \end{Bmatrix} \tag{7.91}$$

Shape functions

For the element interpolation we can use the product of first-degree Lagrangian polynomials

$$N_i(x,y,z) = L_i(x)\, L_i(y)\, L_i(z) \tag{7.92}$$

Shape functions in local and global element coordinates. can be defined in following way.

$$\begin{aligned}
N_1 &= \frac{1}{8}(1+r)(1+s)(1+t) \quad , \quad N_1 = \frac{x-x_2}{x_1-x_2}\frac{y-y_4}{y_1-y_4}\frac{z-z_5}{z_1-z_5} \\
N_2 &= \frac{1}{8}(1-r)(1+s)(1+t) \quad , \quad N_2 = \frac{x-x_3}{x_2-x_3}\frac{y-y_1}{y_2-y_1}\frac{z-z_6}{z_2-z_6} \\
N_3 &= \frac{1}{8}(1-r)(1-s)(1+t) \quad , \quad N_3 = \frac{x-x_4}{x_3-x_4}\frac{y-y_2}{y_3-y_2}\frac{z-z_7}{z_3-z_7} \\
N_4 &= \frac{1}{8}(1+r)(1-s)(1+t) \quad , \quad N_4 = \frac{x-x_1}{x_4-x_1}\frac{y-y_3}{y_4-y_3}\frac{z-z_8}{z_4-z_8} \\
N_5 &= \frac{1}{8}(1+r)(1+s)(1-t) \quad , \quad N_5 = \frac{x-x_6}{x_5-x_6}\frac{y-y_8}{y_5-y_8}\frac{z-z_1}{z_5-z_1} \\
N_6 &= \frac{1}{8}(1-r)(1+s)(1-t) \quad , \quad N_6 = \frac{x-x_7}{x_6-x_7}\frac{y-y_5}{y_6-y_5}\frac{z-z_2}{z_6-z_2} \\
N_7 &= \frac{1}{8}(1-r)(1-s)(1-t) \quad , \quad N_7 = \frac{x-x_8}{x_7-x_8}\frac{y-y_6}{y_7-y_6}\frac{z-z_3}{z_7-z_3} \\
N_8 &= \frac{1}{8}(1+r)(1-s)(1-t) \quad , \quad N_8 = \frac{x-x_5}{x_8-x_5}\frac{y-y_7}{y_8-y_7}\frac{z-z_4}{z_8-z_4}
\end{aligned} \tag{7.93}$$

Implementation: MOmega3D(omega,r,s,t)

The matrix of shape function derivatives is therefore given as follows

$$\nabla N_{rst} = \begin{bmatrix} \frac{\partial N_1}{\partial r} & \frac{\partial N_1}{\partial s} & \frac{\partial N_1}{\partial t} \\ \frac{\partial N_2}{\partial r} & \frac{\partial N_2}{\partial s} & \frac{\partial N_2}{\partial t} \\ \frac{\partial N_3}{\partial r} & \frac{\partial N_3}{\partial s} & \frac{\partial N_3}{\partial t} \\ \frac{\partial N_4}{\partial r} & \frac{\partial N_4}{\partial s} & \frac{\partial N_4}{\partial t} \\ \frac{\partial N_5}{\partial r} & \frac{\partial N_5}{\partial s} & \frac{\partial N_5}{\partial t} \\ \frac{\partial N_6}{\partial r} & \frac{\partial N_6}{\partial s} & \frac{\partial N_6}{\partial t} \\ \frac{\partial N_7}{\partial r} & \frac{\partial N_7}{\partial s} & \frac{\partial N_7}{\partial t} \\ \frac{\partial N_8}{\partial r} & \frac{\partial N_8}{\partial s} & \frac{\partial N_8}{\partial t} \end{bmatrix} \tag{7.94}$$

$$= \frac{1}{8} \begin{bmatrix} (1+s)(1+t) & (1+r)(1+t) & (1+s)(1+t) \\ -(1+s)(1+t) & (1-r)(1+t) & (1+s)(1+t) \\ -(1-s)(1+t) & -(1-r)(1+t) & (1-s)(1+t) \\ (1-s)(1+t) & -(1+r)(1+t) & (1-s)(1+t) \\ (1+s)(1-t) & (1+r)(1-t) & (1+s)(1-t) \\ -(1+s)(1-t) & (1-r)(1-t) & (1+s)(1-t) \\ -(1-s)(1-t) & -(1-r)(1-t) & (1-s)(1-t) \\ (1-s)(1-t) & -(1+r)(1-t) & (1-s)(1-t) \end{bmatrix}$$

Implementation: MGradOmega3D(gradient,r,s,t)

Shape function derivatives in global coordinates $(x-y-z)$ can be determined by use of the inverse Jacobian matrix $[J_{3D}]^{-1}$.

$$\nabla_{xyz}\mathbf{N} = [J_{3D}]^{-1}\nabla_{rst}\mathbf{N} \tag{7.95}$$

The Jacobian matrix in 3-D case is given by

$$[J_{3D}] = \begin{bmatrix} \frac{\partial x}{\partial r} & \frac{\partial y}{\partial r} & \frac{\partial z}{\partial r} \\ \frac{\partial x}{\partial s} & \frac{\partial y}{\partial s} & \frac{\partial z}{\partial s} \\ \frac{\partial x}{\partial t} & \frac{\partial y}{\partial t} & \frac{\partial z}{\partial t} \end{bmatrix} = [\nabla_{rst}\mathbf{N}]\,[A_{3D}] \tag{7.96}$$

which can be rewritten, if using the isoparametric coordinate transformation eqn (7.91), as

$$[J_{3D}] = \begin{bmatrix} \frac{\partial N_i}{\partial r} x_i & \frac{\partial N_i}{\partial r} y_i & \frac{\partial N_i}{\partial r} z_i \\ \frac{\partial N_i}{\partial s} x_i & \frac{\partial N_i}{\partial s} y_i & \frac{\partial N_i}{\partial s} z_i \\ \frac{\partial N_i}{\partial t} x_i & \frac{\partial N_i}{\partial t} y_i & \frac{\partial N_i}{\partial t} z_i \end{bmatrix} \tag{7.97}$$

Implementation: Calc3DElementJacobianMatrix(element,r,s,t,invjac,detjac)

The required inverse Jacobian $[J_{3D}]^{-1}$ has to be calculated by formulas for matrix inversion.

Implementation: M3Invertiere(matrix)

7.5 Network Simulation

7.5.1 Pipe Networks - 1-D Elements in $\mathcal{R}^3$

We consider linear one-dimensional elements in three-dimensional space. Those elements may be used e.g. for pipe network modeling. The following figure shows the physical (x, y, z) and the virtual (or natural) systems of coordinates (r).

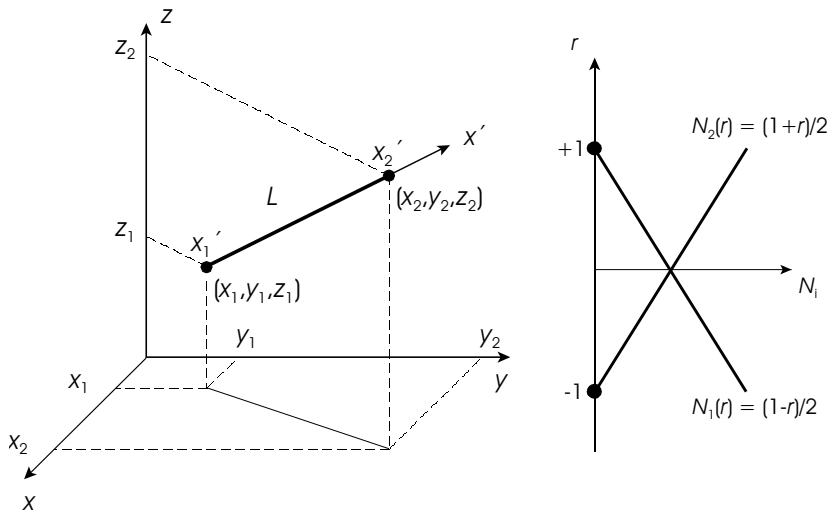

Figure 7.13: 1-D element in $\mathcal{R}^3$ and in local isoparametric coordinates

Derivatives of shape functions of isoparametric elements are

$$\begin{aligned} \frac{dN_1}{dx'} &= \frac{dr}{dx'}\frac{dN_1}{dr} = [J_{1D}^{-1}]\frac{dN_1}{dr} = -\frac{1}{L} \\ \frac{dN_2}{dx'} &= \frac{dr}{dx'}\frac{dN_2}{dr} = [J_{1D}^{-1}]\frac{dN_2}{dr} = \frac{1}{L} \end{aligned} \tag{7.98}$$

The Jacobian matrix of the 1-D element in local physical coordinates is given by

$$[J_{1D}] = \left[\frac{dx'}{dr}\right] = \left[\frac{dN_1}{dr}\ \frac{dN_2}{dr}\right]\begin{bmatrix} x'_1 \\ x'_2 \end{bmatrix} = \left[\frac{dN}{dr}\right]^T [A'_{1D}] \tag{7.99}$$

Nodal coordinates have to transformed into the global coordinate system by use of following transformation matrices

$$\begin{aligned} [A'_{1D}] &= \begin{bmatrix} x'_1 \\ x'_2 \end{bmatrix} = [\cos(x',x)\ \cos(x',y)\ \cos(x',z)]\begin{bmatrix} x_1 & x_2 \\ y_1 & y_2 \\ z_1 & z_2 \end{bmatrix} \\ &= [{}_{1D}T_{3D}]\ [A_{3D}]^T \end{aligned} \tag{7.100}$$

The Jacobian matrix (1st rank tensor) of the 1-D element can be written in global physical coordinates as follows

$$\begin{aligned} [J_{1D}] &= \left[\frac{dN}{dr}\right]^T [{}_{1D}T_{3D}][A_{3D}]^T \\ &= \frac{1}{2}\sqrt{(x_2-x_1)^2+(y_2-y_1)^2+(z_2-z_1)^2} = \frac{L}{2} \end{aligned} \tag{7.101}$$

The determinant of the Jacobian matrix is simply

$$\det[J_{1D}] = \frac{L}{2} \tag{7.102}$$

and therefore the inverse of the Jacobian matrix is given by

$$[J_{1D}^{-1}] = \frac{2}{L} \tag{7.103}$$

Alternative approach

Alternatively, the derivatives of shape functions in global coordinates can be calculated by the following one-step transformation.

$$\begin{Bmatrix} \frac{\partial N}{\partial x} \\ \frac{\partial N}{\partial y} \\ \frac{\partial N}{\partial z} \end{Bmatrix} = \begin{bmatrix} \frac{\partial x'}{\partial x} & \frac{\partial x'}{\partial y} & \frac{\partial x'}{\partial z} \\ \frac{\partial y'}{\partial x} & \frac{\partial y'}{\partial y} & \frac{\partial y'}{\partial z} \\ \frac{\partial z'}{\partial x} & \frac{\partial z'}{\partial y} & \frac{\partial z'}{\partial z} \end{bmatrix} \begin{Bmatrix} \frac{\partial N}{\partial x'} \\ \frac{\partial N}{\partial y'} \\ \frac{\partial N}{\partial z'} \end{Bmatrix} \tag{7.104}$$

$$= \begin{bmatrix} cos(x',x) & cos(x',y) & cos(x',z) \\ cos(y',x) & cos(y',y) & cos(y',z) \\ cos(z',x) & cos(z',y) & cos(z',z) \end{bmatrix} \begin{Bmatrix} \frac{\partial N}{\partial x'} \\ 0 \\ 0 \end{Bmatrix}$$

$$= \begin{Bmatrix} cos(x',x) \\ cos(y',x) \\ cos(z',x) \end{Bmatrix} \frac{\partial N}{\partial x'}$$

7.5.2 Fracture Networks - 2-D Triangular Elements in $\mathcal{R}^3$

We consider linear triangular elements in three-dimensional space. Those elements may be used e.g. for fracture network modeling. The following figure shows several coordinate systems: global 3-D coordinates (x, y, z) , 2-D element coordinates (x', y'), and 2-D element tensor coordinates (x'', y''),

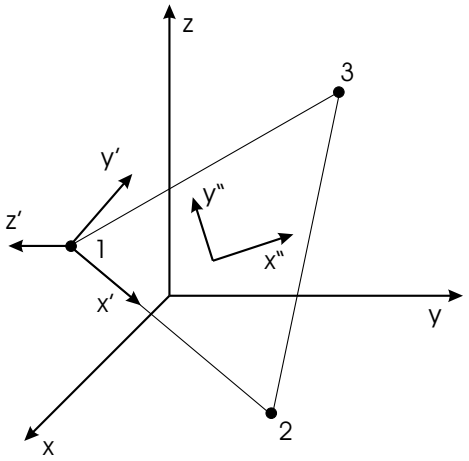

Figure 7.14: 2-D triangular element in $\mathcal{R}^3$, several coordinate systems

Element Area

Calculation of triangle area is very simple using the following formula, which is well known from analytic geometry.

$$A = \frac{1}{2}|\mathbf{x}_{12} \times \mathbf{x}_{13}| = \begin{vmatrix} 1 & 1 & 1 \\ x_2 - x_1 & y_2 - y_1 & z_2 - z_1 \\ x_3 - x_1 & y_3 - y_1 & z_3 - z_1 \end{vmatrix} \tag{7.105}$$

Implementation: CalcElementVolumeTriangle()

Derivatives of Shape Functions

Frequently, we need to have the derivatives of shape functions for several element matrices expressing fluxes of quantities (e.g. conductance, advection, diffusion, stiffness matrices).

$$\nabla_{x'y'}\mathbf{N} = \begin{bmatrix} \frac{\partial N_1}{\partial x'} & \frac{\partial N_1}{\partial y'} \\ \frac{\partial N_2}{\partial x'} & \frac{\partial N_2}{\partial y'} \\ \frac{\partial N_3}{\partial x'} & \frac{\partial N_3}{\partial y'} \end{bmatrix} = \frac{1}{2A} \begin{bmatrix} y'_2 - y'_3 & x'_3 - x'_2 \\ y'_3 - y'_1 & x'_1 - x'_3 \\ y'_1 - y'_2 & x'_2 - x'_1 \end{bmatrix} \tag{7.106}$$

Note, we have written the derivatives in local element coordinates based on eqn (7.65). Now we have only to rewrite this expression in global coordinates $(x - y - z)$ using the following coordinate transformation.

Coordinate Transformation

The transformation between global (3-D: $x - y - z$) and local (2-D: $x' - y'$) coordinates is given by

$$[A'_{2D}] = [A_{3D}]\,[{}_{2D}T_{3D}] \tag{7.107}$$

with:

- Local coordinate matrix

$$[A'_{2D}] = \begin{bmatrix} x'_1 & y'_1 \\ x'_2 & y'_2 \\ x'_3 & y'_3 \end{bmatrix} \tag{7.108}$$

- Coordinate transformation matrix

$$[{}_{2D}T_{3D}] = \begin{bmatrix} cos(x', x) & cos(y', x) \\ cos(x', y) & cos(y', y) \\ cos(x', z) & cos(y', z) \end{bmatrix} \tag{7.109}$$

- Global coordinate matrix

$$[A_{3D}] = \begin{bmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ x_3 & y_3 & z_3 \end{bmatrix} \tag{7.110}$$

Implementation: Calc2DElementCoordinatesTriangle()

Sometimes, material tensors are related to specific directions, i.e. they have anisotropic behavior. We assume that $x'' - y''$ are the principal directions (Fig.

7.14). In order to evaluate element matrices in local element coordinates, we have to transform those tensors into these coordinates.

$$\mathbf{k}' = \mathbf{T}\,\mathbf{k}''\,\mathbf{T}^T \tag{7.111}$$

with coordinate transformation matrix:

$$\mathbf{T} = \begin{bmatrix} cos(x'', x') & cos(x'', y') \\ cos(y'', x') & cos(y'', y') \end{bmatrix} \tag{7.112}$$

7.6 Diffusion Equation

7.6.1 Steady Diffusion Equation

Example: 1-D steady diffusion equation

As an example we consider the one-dimensional diffusion equation

$$\frac{\partial}{\partial x}(\alpha \frac{\partial u}{\partial x}) = q \tag{7.113}$$

We develop a Galerkin weak formulation with linear elements. The discretized nodal equation is written as

$$-\sum_i u_i \int_{\Omega_j} \alpha \nabla N_i \cdot \nabla N_j d\Omega + \oint_{\partial\Omega_j} N_j \alpha \frac{\partial u}{\partial n} dS = \int_{\Omega_j} N_j q d\Omega \tag{7.114}$$

Using linear shape functions

$$N_1(x) = \frac{1-x}{x_2 - x_1} \quad , \quad N_2(x) = \frac{x}{x_2 - x_1} \tag{7.115}$$

the weak formulation for the 1-D problem is

$$-\sum_{j=i-1} u_j \int_{i-1} \alpha \frac{\partial N_j}{\partial x} \frac{\partial N_i}{\partial x} dx - \int_{i-1} N_i q dx = 0 \tag{7.116}$$

The derivatives of the shape functions are given by

$$\frac{\partial N_1}{\partial x} = -\frac{1}{x_2 - x_1} \quad , \quad \frac{\partial N_2}{\partial x} = \frac{1}{x_2 - x_1} \tag{7.117}$$

Performing the integration we obtain

$$\alpha_{i+\frac{1}{2}} \frac{u_{i+1} - u_i}{\Delta x^2} - \alpha_{i-\frac{1}{2}} \frac{u_i - u_{i-1}}{\Delta x^2} = \frac{q_{i-1} + 4q_i + q_{i-1}}{6} \tag{7.118}$$

For a linear variation of the diffusion coefficient we have

$$\alpha_{i+\frac{1}{2}} = \frac{1}{\Delta x} \int_{x_i} \alpha dx = \frac{\alpha_{i+1} + \alpha_i}{2} \tag{7.119}$$

Note the left-hand-side is identical to a central second-order finite difference scheme (6.11).

$$\frac{u_{i+1} - 2u_i + u_{i-1}}{\Delta x^2} \tag{7.120}$$

It is a general rule for the case of uniform meshes that elements of order p lead to $p + 1$-order accurate discretizations.

Example: 2-D steady diffusion equation

In order to demonstrate the method of weighted residuals we consider the 2-D steady diffusion equation

$$\frac{\partial}{\partial x}(\alpha \frac{\partial u}{\partial x}) + \frac{\partial}{\partial y}(\alpha \frac{\partial u}{\partial y}) = q \tag{7.121}$$

The residual resulting from the approximation of the solution

$$R(\hat{u}) = \frac{\partial}{\partial x}(\alpha \frac{\partial \hat{u}}{\partial x}) + \frac{\partial}{\partial y}(\alpha \frac{\partial \hat{u}}{\partial y}) - q = \nabla \cdot (\alpha \nabla \hat{u}) - q \tag{7.122}$$

The method of weighted residuals yields

$$\int_\Omega \omega R(\hat{u}) d\Omega = \int_\Omega \omega (\nabla \cdot (\alpha \nabla \hat{u}) - q) d\Omega \tag{7.123}$$

Next we perform integration by part of the second order derivative terms.

$$\int_\Omega \omega \nabla \cdot (\alpha \nabla \hat{u}) d\Omega = -\int_\Omega \nabla \omega \cdot (\alpha \nabla \hat{u}) d\Omega + \int_\Omega \nabla (\omega \alpha \nabla \hat{u}) d\Omega \tag{7.124}$$

Using the Gauss-Ostrogradskian theorem for volume integrals we obtain

$$\int_\Omega \nabla (\omega \alpha \nabla \hat{u}) d\Omega = \oint \omega \alpha \frac{\partial \hat{u}}{\partial n} dS \tag{7.125}$$

Finally we derived the following weak formulation of the 2-D diffusion equation.

$$-\int_\Omega \nabla \omega \cdot (\alpha \nabla \hat{u}) d\Omega + \oint \omega \alpha \frac{\partial \hat{u}}{\partial n} dS = \int_\Omega \omega q d\Omega \tag{7.126}$$

7.6.2 Unsteady Diffusion Equation

A an example we consider isothermal flow of homogeneous liquids in a porous medium (e.g. saturated groundwater flow). Readers who are interested in more details of the derivation of the governing equations should refer to Pinder & Gray (1983) and Diersch (1985).

Governing Equations

We start from the differential form of the balance equation of the fluid mass (giving the exact solution for the unknown function - fluid pressure) which belongs to the parabolic PDEs (see Chapter 4.4).

$$S_0^p \frac{\partial p}{\partial t} - \frac{\partial}{\partial x_\alpha}\left(\frac{k_{\alpha\beta}}{\mu}(\frac{\partial p}{\partial x_\beta} + \rho g \frac{\partial z}{\partial x_\beta})\right) = Q_\rho \quad , \quad \alpha,\beta = 1,2,3 \tag{7.127}$$

Galerkin Method

The use of the method of weighted residuals (see 7.2.2) provides us an integral formulation of the flow equation to obtain the approximate solution.

$$\int_\Omega \omega_i \left[S_0^p \frac{\partial \hat{p}}{\partial t} - \frac{\partial}{\partial x_\alpha}\left(\frac{k_{\alpha\beta}}{\mu}(\frac{\partial \hat{p}}{\partial x_\beta} + \rho g \frac{\partial \hat{z}}{\partial x_\beta})\right)\right] d\Omega = \int_\Omega \omega_i Q_\rho d\Omega \tag{7.128}$$

The unknown function is approximated by a trial solution (7.7). Note in the following we use Einstein summation convention for repeating indices.

$$\hat{p}(t, x_\alpha) = \sum_{j=1}^{m} \phi_j(x_\alpha)\, p_j(t) \equiv \phi_j(x_\alpha)\, p_j(t) \tag{7.129}$$

According to the Galerkin method we use identical weighting ω and interpolation functions ϕ.

$$\int_\Omega \phi_i \left[S_0^p \phi_j \frac{dp_j}{dt} - \frac{\partial}{\partial x_\alpha}\left(\frac{k_{\alpha\beta}}{\mu}(\frac{\partial \phi_j}{\partial x_\beta} p_j + \rho g \frac{\partial \phi_j}{\partial x_\beta} z_j)\right)\right] d\Omega = \int_\Omega \phi_i Q_\rho d\Omega \tag{7.130}$$

Partial integration $\phi \nabla A = \nabla(\phi A) - A \nabla \phi$ and the Gauss-Ostrogradskian integral theorem are used to reduce the order of the derivatives.

$$\int_\Omega \phi_i \left[S_0^p \phi_j \frac{dp_j}{dt} + \frac{\partial \phi_j}{\partial x_\alpha}\left(\frac{k_{\alpha\beta}}{\mu}(\frac{\partial \phi_j}{\partial x_\beta} p_j + \rho g \frac{\partial \phi_j}{\partial x_\beta} z_j)\right)\right] d\Omega = - \int_{\partial\Omega} \phi_i q_n dS + \int_\Omega \phi_i Q_\rho d\Omega \tag{7.131}$$

with the outward flux vector

$$q_n = -\frac{k_{\alpha\beta}}{\mu}(\frac{\partial \hat{p}}{\partial x_\beta} + \rho g \frac{\partial z}{\partial x_\beta})n_\alpha \tag{7.132}$$

We rearrange the above equation to put the unknowns terms only to the left hand side.

$$\begin{aligned} \int_\Omega \phi_i \left[S_0^p \phi_j \frac{dp_j}{dt} + \frac{\partial \phi_j}{\partial x_\alpha} \frac{k_{\alpha\beta}}{\mu} \frac{\partial \phi_j}{\partial x_\beta} p_j \right] d\Omega = \\ - \int_\Omega \phi_i \frac{\partial \phi_j}{\partial x_\alpha} \frac{k_{\alpha\beta}}{\mu} \rho g \frac{\partial \phi_j}{\partial x_\beta} z_j d\Omega - \int_{\partial\Omega} \phi_i q_n dS + \int_\Omega \phi_i Q_\rho d\Omega \end{aligned} \tag{7.133}$$

The above equation forms a global system of equations where the number of equations corresponds to the number of grid points.

$$\boxed{C_{ij}\frac{dp_j}{dt} + K_{ij}p_j = r_i} \quad , \quad i,j = 1,..,np \tag{7.134}$$

with

$$\begin{aligned} C_{ij} &= \int_\Omega \phi_i S_0^p \phi_j \, d\Omega \\ K_{ij} &= \int_\Omega \frac{\partial \phi_i}{\partial x_\alpha} \frac{k_{\alpha\beta}}{\mu} \frac{\partial \phi_j}{\partial x_\beta} \, d\Omega \\ r_i &= -\rho g K_{ij} z_i - \int_{\partial\Omega} \phi_i q_n dS + \int_\Omega \phi_i Q_\rho d\Omega \end{aligned} \tag{7.135}$$

Finite Element Approach

Decomposition of the computation domain into finite elements

$$\Omega = \bigcup_{e=1}^{ne} \Omega^e \quad , \quad \partial\Omega = \bigcup_{e=1}^{ne} \partial\Omega^e \tag{7.136}$$

means that the global matrices can be composed by its element contributions.

$$\begin{aligned} C_{ij} &= \sum_{e=1}^{ne} C_{ij}^e \\ K_{ij} &= \sum_{e=1}^{ne} K_{ij}^e \\ r_i &= \sum_{e=1}^{ne} r_i^e \end{aligned} \tag{7.137}$$

Interpolation functions correspond now to individual elements and shape functions are used for interpolation at the element level.

$$\hat{p}^e = N_j(x_\alpha)p_j(t) \quad , \quad N_j = \begin{cases} 1 & , \quad j = i \\ 0 & , \quad j \neq i \end{cases} \tag{7.138}$$

$$\begin{aligned} C^e_{ij} &= \int_{\Omega^e} N_i S^p_0 N_j \, d\Omega \\ K^e_{ij} &= \int_{\Omega^e} \frac{\partial N_i}{\partial x_\alpha} \frac{k_{\alpha\beta}}{\mu} \frac{\partial N_j}{\partial x_\beta} \, d\Omega^e \\ r^e_i &= -\rho g K^e_{ij} z_i - \int_{\partial\Omega^e} N_i q_n dS + \int_{\Omega^e} N_i Q_\rho d\Omega \end{aligned} \tag{7.139}$$

Evaluation of Element Matrices in Local Coordinates

Transformation from physical (x, y, z) to local (r, s, t) coordinates, e.g. by isoparametric functions, allows to evaluate the element matrices in unit coordinates.

$$C^e_{ij} = \int_{\Omega^e} N_i S^p_0 N_j \, d\Omega \tag{7.140}$$

$$\begin{aligned} \mathbf{C}^e &= \int_{\Omega^e} \mathbf{N} S^p_0 \mathbf{N} \, d\Omega \\ &= \int_{-1}^{+1}\int_{-1}^{+1}\int_{-1}^{+1} \mathbf{N} S^p_0 \mathbf{N} \underbrace{\det \mathbf{J} \, drdsdt}_{d\Omega^e} \end{aligned}$$

$$K^e_{ij} = \int_{\Omega^e} \frac{\partial N_i}{\partial x_\alpha} \frac{k_{\alpha\beta}}{\mu} \frac{\partial N_j}{\partial x_\beta} \, d\Omega^e \tag{7.141}$$

$$\begin{aligned} \mathbf{K}^e &= \int_{\Omega^e} \nabla\mathbf{N} \frac{1}{\mu} \mathbf{k} \nabla\mathbf{N}^T \, d\Omega^e \\ &= \int_{-1}^{+1}\int_{-1}^{+1}\int_{-1}^{+1} \underbrace{\nabla\mathbf{N}(\mathbf{J}^{-1})^T}_{\nabla\mathbf{N}(x,y,z)} \frac{1}{\mu} \underbrace{\mathbf{T}^T\mathbf{k}\mathbf{T}}_{\mathbf{k}(x,y,z)} \underbrace{\nabla\mathbf{N}^T\mathbf{J}^{-1}}_{\nabla\mathbf{N}^T(x,y,z)} \underbrace{\det \mathbf{J} \, drdsdt}_{d\Omega^e} \end{aligned}$$

$$r^e_i = -\rho g K^e_{ij} z_i - \int_{\partial\Omega^e} N_i q_n dS + \int_{\Omega^e} N_i Q_\rho d\Omega \tag{7.142}$$

$$\mathbf{r}^e = -\rho g \mathbf{K^e z} - \int_{\partial\Omega^e} \mathbf{N} q_n dS + \int_{-1}^{+1}\int_{-1}^{+1}\int_{-1}^{+1} \mathbf{N} Q_\rho \det \mathbf{J} \, drdsdt$$

with $\mathbf{J}$ the Jacobian matrix and $\mathbf{T}$ the ordinary coordinate transformation matrix.

More details of finite element formulations for subsurface problems, in particular evaluation of element matrices for different flow and transport elements, can be found in Kolditz et al. (2001).

7.7 Transport Equation

7.7.1 Unsteady Transport Equation

We start from the differential form of the balance equation of mass transport (3.40). The weighted residual method applied to this equation yields the following weak formulation for the unknown functions.

Galerkin Method

$$\int_\Omega \omega \left[\frac{\partial (n\hat{C})}{\partial t} + \frac{\partial (q_\alpha \hat{C})}{\partial x_\alpha} - \frac{\partial}{\partial x_\alpha} \left(n D_{\alpha\beta} \frac{\partial \hat{C}}{\partial x_\beta} \right) \right] d\Omega \quad = \quad \int_\Omega \omega (n Q_C) d\Omega \tag{7.143}$$

where ω are weighting functions, and Ω is the solution domain. Approximate solutions of the unknown functions are specified by a hat.

After applying partial differentiation, we use Gauss-Ostrogradskian integral theorem to reduce second-order derivatives

$$\int_\Omega \left[\omega \frac{\partial (n\hat{C})}{\partial t} + \omega \frac{\partial (q_\alpha \hat{C})}{\partial x_\alpha} + \frac{\partial \omega}{\partial x_\alpha} \left(n D_{\alpha\beta} \frac{\partial \hat{C}}{\partial x_\beta} \right) \right] d\Omega = \int_\Gamma \omega q_n d\Gamma + \int_\Omega \omega (n Q_C) d\Omega \tag{7.144}$$

with tracer flux vector

$$q_n \quad = \quad n D_{\alpha\beta} \frac{\partial \hat{C}}{\partial x_\beta} n_\alpha \tag{7.145}$$

where $\partial\Omega$ is the segment of the domain boundary at which Neumann or Cauchy (flux type) boundary conditions have to be specified.

$$\int_\Omega \left[\omega \frac{\partial (n\hat{C})}{\partial t} + \omega \frac{\partial (q_\alpha \hat{C})}{\partial x_\alpha} + \frac{\partial \omega}{\partial x_\alpha} \left(n D_{\alpha\beta} \frac{\partial \hat{C}}{\partial x_\beta} \right) \right] d\Omega = \int_\Gamma \omega q_n d\Gamma + \int_\Omega \omega \left(n R d \frac{A}{V} (C_s - \hat{C}) \right) d\Omega \tag{7.146}$$

$$\int_\Omega \left[\omega n\frac{\partial \hat{C}}{\partial t} + \omega\hat{C}\frac{\partial n}{\partial t} + \omega\frac{\partial(q_\alpha \hat{C})}{\partial x_\alpha} + \frac{\partial \omega}{\partial x_\alpha}\left(nD_{\alpha\beta}\frac{\partial \hat{C}}{\partial x_\beta}\right)\right] d\Omega$$
$$+ \int_\Omega \omega(nRd\,\frac{A}{V}\,\hat{C})d\Omega = \int_\Gamma \omega q_n d\Gamma + \int_\Omega \omega(nRd\,\frac{A}{V}\,C_s)d\Omega \qquad (7.147)$$

The unknown function is approximated by a trial solution

$$\hat{C}(t, x_\alpha) = \sum_{i=1}^{np} \phi_i(x_\alpha)C_i(t) = \phi_i(x_\alpha)C_i(t) \qquad (7.148)$$

where ϕ_i are interpolation functions, and np is the number of grid points. Indices denote values of the unknown functions at the grid points.

According to the Galerkin method we use identical weighting ω_i and interpolation functions ϕ_i.

$$\int_\Omega \left[\phi_i n\phi_j\frac{dC_j}{dt} + \phi_i\frac{\partial n}{\partial t}\phi_j C_j + \phi_i\frac{\partial(q_\alpha\phi_j)}{\partial x_\alpha}C_j + \frac{\partial \phi_i}{\partial x_\alpha}(nD_{\alpha\beta}\frac{\partial \phi_j}{\partial x_\beta}C_j)\right] d\Omega$$
$$+ \int_\Omega \phi_i(nRd\frac{A}{V})\phi_j C_j d\Omega = \int_\Gamma \phi_i q_n d\Gamma + \int_\Omega \phi_i(nRd\,\frac{A}{V}\,C_s)d\Omega \qquad (7.149)$$

Algebraic equations

If rearranging the above equations to put unknowns terms to the left hand side only, they form system of algebraic equations with respect to grid values of the unknown functions.

$$\boxed{C_{ij}\frac{dC_j}{dt} + (B_{ij} + D_{ij} + E_{ij} + F_{ij})C_j = r_j} \qquad i, j = 1, .., np \qquad (7.150)$$

with

$$C_{ij} = \int_\Omega \phi_i n\phi_j\, d\Omega \qquad \text{Tracer capacitance matrix}$$
$$B_{ij} = \int_\Omega \phi_i\frac{\partial(q_\alpha\phi_j)}{\partial x_\alpha}\, d\Omega \qquad \text{Tracer advection matrix}$$
$$D_{ij} = \int_\Omega \frac{\partial \phi_i}{\partial x_\alpha}(nD_{\alpha\beta})\frac{\partial \phi_j}{\partial x_\beta}\, d\Omega \qquad \text{Tracer diffusion matrix}$$
$$E_{ij} = \int_\Omega \phi_i\frac{\partial n}{\partial t}\phi_j\, d\Omega$$
$$F_{ij} = \int_\Omega \phi_i(nRd\frac{A}{V})\phi_j\, d\Omega \qquad \text{Tracer solution matrix}$$
$$r_j = \int_\Gamma \phi_i q_n d\Gamma + \int_\Omega \phi_i(nRd\frac{A}{V}C_s)d\Omega \qquad \text{RHS vector} \qquad (7.151)$$

Finite Element Approach

Decomposition of the computation domain into finite elements

$$\Omega = \bigcup_{e=1}^{ne} \Omega^e \tag{7.152}$$

means that the global matrices can be composed by its element contributions.

Interpolation functions correspond now to individual elements and shape functions are used for interpolation at the element level.

$$\hat{C}^e(x_\alpha, t) = \sum_{j=1}^{n} N_j(x_\alpha) C_j(t) \tag{7.153}$$

where n is the node number per element.

Element matrices:

$$\begin{aligned}
C^e_{ij} &= \int_{\Omega^e} N_i n N_j \, d\Omega^e \\
B^e_{ij} &= \int_{\Omega^e} N_i q_\alpha \frac{\partial N_j}{\partial x_\alpha} \, d\Omega^e \\
D^e_{ij} &= \int_{\Omega^e} \frac{\partial N_i}{\partial x_\alpha} (n D_{\alpha\beta}) \frac{\partial N_j}{\partial x_\beta} \, d\Omega^e \\
E^e_{ij} &= \int_{\Omega^e} N_i \frac{dn}{dt} N_j \, d\Omega^e \\
F^e_{ij} &= \int_{\Omega^e} N_i (nRd\frac{A}{V}) N_j \, d\Omega^e \\
r^e_j &= \int_{\Gamma^e} N_i q_n d\Gamma^e + \int_{\Omega^e} N_i (nRd\frac{A}{V} C_s) d\Omega^e
\end{aligned} \tag{7.154}$$

Transformation to Local Element Coordinates

Transformation from physical (x, y, z) to local (r, s, t) coordinates, e.g. by isoparametric functions, allows to evaluate the element matrices in unit coordinates.

$$\begin{aligned}
C^e_{ij} &= \int_{\Omega^e} N_i n N_j \, d\Omega^e \\
\mathbf{C}^e &= \int_{\Omega^e} \mathbf{N} n \mathbf{N} \, d\Omega^e
\end{aligned}$$

$$= \int_{-1}^{+1}\int_{-1}^{+1}\int_{-1}^{+1} \mathbf{N} n \mathbf{N} \underbrace{\det \mathbf{J}\, drdsdt}_{d\Omega^e} \tag{7.155}$$

$$\begin{aligned} B_{ij}^e &= \int_{\Omega^e} N_i q_\alpha \frac{\partial N_j}{\partial x_\alpha}\, d\Omega^e \\ \mathbf{B}^e &= \int_{\Omega^e} \mathbf{N}\,\mathbf{q}\,\nabla \mathbf{N}^T\, d\Omega^e \\ &= \int_{-1}^{+1}\int_{-1}^{+1}\int_{-1}^{+1} \mathbf{N} \underbrace{\mathbf{T}\mathbf{q}}_{\mathbf{q}(x,y,z)} \underbrace{\nabla \mathbf{N}^T \mathbf{J}^{-1}}_{\nabla \mathbf{N}^T(x,y,z)} \underbrace{\det \mathbf{J}\, drdsdt}_{d\Omega^e} \end{aligned} \tag{7.156}$$

$$\begin{aligned} D_{ij}^e &= \int_{\Omega^e} \frac{\partial N_i}{\partial x_\alpha}(nD_{\alpha\beta})\frac{\partial N_j}{\partial x_\beta}\, d\Omega^e \\ \mathbf{D}^e &= \int_{\Omega^e} \nabla \mathbf{N}\,(n\mathbf{D}^T)\,\nabla \mathbf{N}^T\, d\Omega^e \\ &= \int_{-1}^{+1}\int_{-1}^{+1}\int_{-1}^{+1} \underbrace{\nabla \mathbf{N}(\mathbf{J}^{-1})^T}_{\nabla \mathbf{N}(x,y,z)} \underbrace{\mathbf{T}^T(n\mathbf{D}^T)\mathbf{T}}_{n\mathbf{D}^T(x,y,z)} \underbrace{\nabla \mathbf{N}^T \mathbf{J}^{-1}}_{\nabla \mathbf{N}^T(x,y,z)} \underbrace{\det \mathbf{J}\, drdsdt}_{d\Omega^e} \end{aligned} \tag{7.157}$$

$$\begin{aligned} E_{ij}^e &= \int_{\Omega^e} N_i \frac{dn}{dt} N_j\, d\Omega^e \\ \mathbf{E}^e &= \int_{\Omega^e} \mathbf{N}\frac{dn}{dt}\mathbf{N}\, d\Omega^e \\ &= \int_{-1}^{+1}\int_{-1}^{+1}\int_{-1}^{+1} \mathbf{N}\frac{dn}{dt}\mathbf{N} \underbrace{\det \mathbf{J}\, drdsdt}_{d\Omega^e} \end{aligned} \tag{7.158}$$

$$\begin{aligned} F_{ij}^e &= \int_{\Omega^e} N_i (nRd\frac{A}{V}) N_j\, d\Omega^e \\ \mathbf{F}^e &= \int_{\Omega^e} \mathbf{N}(nRd\frac{A}{V})\mathbf{N}\, d\Omega^e \\ &= \int_{-1}^{+1}\int_{-1}^{+1}\int_{-1}^{+1} \mathbf{N}(nRd\frac{A}{V})\mathbf{N} \underbrace{\det \mathbf{J}\, drdsdt}_{d\Omega^e} \end{aligned} \tag{7.159}$$

$$\begin{aligned} r_j^e &= \int_{\Gamma^e} N_i q_n d\Gamma^e + \int_{\Omega^e} N_i (nRd\frac{A}{V}C_s) d\Omega^e \\ \mathbf{r}^e &= \int_{\Gamma^e} \mathbf{N} q_n d\Gamma^e + \int_{-1}^{+1}\int_{-1}^{+1}\int_{-1}^{+1} \mathbf{N}(nRd\frac{A}{V}C_s) \underbrace{\det \mathbf{J}\, drdsdt}_{d\Omega^e} \end{aligned} \tag{7.160}$$

with $\mathbf{J}$ the Jacobian matrix and $\mathbf{T}$ the ordinary coordinate transformation matrix.

7.8 Problems

Domain Discretization (Section 7.1)

1 What are basic elements for domain discretization in several dimensions ?

Equation Discretization (Section 7.2)

2 Explain the difference between variational methods and weighted residual methods for discretization of differential equations.

3 What is the role of weighting functions for with respect to different discretization methods. Hint: use Tab. 7.1.

4 Derive eqn (7.131) from eqn (7.130). Hint: Use the chain rule for partial differentiation.

Interpolation and Shape Functions (Section 7.3)

5 Explain the difference between interpolation and shape functions.

6 Explain the termins global (physical) and local (natural) coordinates of elements. What is isoparametric mapping of element coordinates ?

Element Shape Functions in Local Coordinates (Section 7.4)

7 Calculate shape functions $\mathbf{N}$ and its derivatives $\nabla\mathbf{N}$ for a 1-D linear element in point $r = 1/3$

8 Calculate shape functions $\mathbf{N}$ and its derivatives $\nabla\mathbf{N}$ for a 1-D quadratic element in point $r = 1/3$

9 Calculate shape functions $\mathbf{N}$ and its derivatives $\nabla\mathbf{N}$ for a 2-D bilinear element in Gaussian point $r = 1/\sqrt{3}, s = -1/\sqrt{3}$

Coordinate Transformation using the Jacobian Matrix (Sections 7.4, 7.5)

10 Calculate the Jacobian J_{1D} as well as the inversion J_{1D}^{-1} for a 1-D linear element $P_1 = (0, 0, 0)$ and $P_2 = (1, 2, 3)$ in point $r = 1/3$.

11 Calculate the Jacobian J_{2D} as well as the inversion J_{2D}^{-1} for a 2-D linear element in Gaussian point $r = 1/\sqrt{3}, s = -1/\sqrt{3}$.

$$[A'_{2D}]^T = \begin{bmatrix} 0 & 2 & 2 & 0 \\ 0 & 0 & 2 & 1 \end{bmatrix} \tag{7.161}$$

12 Calculate the Jacobian J_{2D} as well as the inversion J_{2D}^{-1} for a 2-D bilinear element in Gaussian point $r = 0, s = 0$.

$$[A_{3D}]^T = \begin{bmatrix} +1 & +1 & -1 & -1 \\ -0.866 & +0.866 & +0.866 & -0.866 \\ -0.35 & +0.65 & +0.65 & -0.35 \end{bmatrix} \tag{7.162}$$

13 Calculate the Jacobian J_{3D} as well as the inversion J_{3D}^{-1} for a 3-D trilinear element in Gaussian point $r = 0, s = 0, t = 0$.

$$[A_{3D}]^T = \begin{bmatrix} +1 & +1 & -1 & -1 & +1 & +1 & -1 & -1 \\ -1 & +1 & +1 & -1 & -1 & +1 & +1 & -1 \\ -1 & -1 & -1 & -1 & +1 & +1 & +1 & +1 \end{bmatrix} \tag{7.163}$$

Element Matrices (Section 7.6)

14 Derive the flow capacitance and conductance matrices for 1-D linear elements.

15 Derive the flow capacitance and conductance matrices for 2-D linear triangular elements.

16 Derive the flow capacitance and conductance matrices for 2-D bilinear quadrilateral elements.

17 Derive the flow capacitance and conductance matrices for 3-D tetrahedral elements.

18 Derive the flow capacitance and conductance matrices for 3-D triangle based prismatic elements.

19 Derive the flow capacitance and conductance matrices for 3-D trilinear hexahedral elements.

20 Derive the mass and heat advection matrices for 1-D linear elements.

21 Derive the mass and heat advection matrices for 2-D linear triangular elements.

22 Derive the mass and heat advection matrices for 2-D bilinear quadrilateral elements.

23 Derive the mass and heat advection matrices for 3-D tetrahedral elements.

24 Derive the mass and heat advection matrices for 3-D triangle based prismatic elements.

25 Derive the mass and heat advection matrices for 3-D trilinear hexahedral elements.

Element Resultants

26 Derive the velocity vector (fluid mass flux) for 1-D linear elements.

27 Derive the velocity vector (fluid mass flux) for 2-D linear triangular elements.

28 Derive the velocity vector (fluid mass flux) for 2-D bilinear quadrilateral elements.

29 Derive the velocity vector (fluid mass flux) for 3-D tetrahedral elements.

30 Derive the velocity vector (fluid mass flux) for 3-D triangle based prismatic elements.

31 Derive the velocity vector (fluid mass flux) for 3-D trilinear hexahedral elements.

Solving Equation Systems

32 Conduct the numerical analysis of steady-state groundwater flow through a vertical column (solving the equation system manually).

33 Conduct the numerical analysis of steady-state groundwater flow through an inclined column.

34 Conduct the numerical analysis of transient groundwater flow through a vertical column (solving the equation system manually).

35 Conduct the stability analysis for 1-D saturated groundwater flow.