Computational Geophysics

A.P. van den Berg

Institute of Earth Sciences
Utrecht University

April 2015

Contents

1 Introduction **1**

2 Numerical modelling of 1-D heat conduction **5**
2.1 A finite difference method with equidistant grid 6
2.1.1 Implementation of natural boundary conditions 8
2.2 A difference method with variable grid spacing 9
2.2.1 Discretization of the equation 9
2.2.2 Implementation of boundary conditions 11

3 Difference methods for 2-D potential problems **15**
3.1 Introduction 15
3.2 A central difference method 16
3.3 A difference method for variable grid spacing and variable coefficient 17
3.4 Implementation of boundary conditions 20
3.4.1 Essential boundary conditions 20
3.4.2 Natural boundary conditions 22

4 The finite element method - an introduction **25**
4.1 Discretization of the domain and solution field 25
4.2 Discretization of the differential equation 28
4.2.1 An example of the Galerkin method with non-local basis functions . 29
4.2.2 Generalisation of the Galerkin method to potential problems in more dimensions 30

5 A finite element solution for the 1-D heat equation **33**
5.1 Discretization of the equation 33
5.1.1 A related Fourier series solution 34
5.2 Structure of the coefficient matrices 35
5.3 Computation of the matrix elements 35
5.3.1 The mass matrix **M** 36
5.3.2 The stiffness matrix **S** 37
5.3.3 The righthand side vector **R** 38
5.4 Implementation of the assembly proces 39
5.5 Solving equations with a tri-diagonal matrix 41
5.6 Implementation of boundary conditions 42
5.6.1 Natural boundary conditions 42
5.6.2 Essential boundary conditions 42
5.7 Steady state problems 43
5.8 Using higher order basis functions 44

6 A finite element solution for multi-dimensional potential problems 47
6.1 Discretization of the equation 48
6.1.1 Element matrices for the 2-D case 50
6.2 Examples of 2-D elements 53
6.2.1 The triangular linear element 53
6.2.2 A quadrilateral element with bi-linear basis functions 56
6.3 Application in various boundary value problems 59
6.3.1 A problem with essential and natural boundary conditions 59
6.3.2 A problem with boundary conditions of type 2 and 3 59

7 Finite element methods for potential equations: geophysical applications 63
7.1 Introduction 63
7.2 Steady state heat conduction 63
7.3 Steady state flow in porous media 64
7.4 Instantaneous viscous flow 64
7.4.1 Governing non-dimensional equations 65
7.4.2 Boundary and initial conditions 65
7.4.3 Potential - streamfunction-vorticity formulation 66

8 Time dependent problems 69
8.1 Introduction 69
8.2 Integration methods 70
8.2.1 The Euler forward method 70
8.2.2 The Euler backward method 71
8.2.3 The Crank-Nicolson method 71
8.3 Stability and convergence of the integration methods 72
8.3.1 Consistency of the integration scheme 73
8.3.2 Stability of the integration scheme 74
8.3.3 Convergence of the integration scheme 75

9 Systems of coupled equations 77
9.1 Model equations for Rayleigh-Benard convection 77
9.1.1 Thermal convection in a porous medium 77
9.1.2 Thermal convection in a viscous fluid layer 79
9.2 Discretization of the governing equations 80
9.3 Steady state convection 80
9.4 Time dependent convection 81
9.4.1 An explicit integration method 82
9.4.2 An implicit method (predictor-corrector) 82

10 Finite element methods for elastic deformation problems 85
10.1 Boundary conditions for elastic problems 85
10.2 Interpolation of vector fields on a grid of nodal points 87
10.3 Expressions for the deformation and stress fields 88
10.4 Discretization of the elastostatic equation 90
10.4.1 Computation of the stiffness matrix 91
10.5 Implementation of boundary conditions 92
10.6 Examples of elastostatic modelling problems 93

11 Finite element methods for viscous flow problems **97**
11.1 Introduction 97
11.2 Boundary conditions for viscous flow problems 99
11.3 Discretization of the Stokes equation using the Galerkin method 100
11.4 The penalty function method 103
11.5 Examples of numerical applications 104
11.5.1 A Poisseuille flow problem 104
11.5.2 An example with forced convection in a subcontinental mantelwedge 106

A Numerical integration with the Gauss-Legendre scheme **107**

B Vector and matrix norms **109**

Chapter 1

Introduction

This course deals with numerical modelling of physical fields and processes in geophysics. The purpose of modelling experiments dealing with geodynamical processes in particular is related to the following circumstances:

- Limited possibilities of direct (in situ) observations in addition to indirect observations as in seismological and gravimetrical measurements. An illustration of this is the limited maximum depth of ≈ 12 km reached by deepdrilling from the Earth's surface.
- The extremely long timescales of these processes - postglacial rebound $10^5\ yr$, mantelconvective overturn $10^8\ yr$, planetary secular cooling $10^9 - 10^{10}\ yr$, makes it practically impossible to monitor the evolution of such processes.

Different ways of proces modelling exist:

1. Physical modelling (laboratory experiments). Examples of geodynamical applications are:
   - Simulation of thermal convection in planetary mantles by studying Rayleigh-Benard convection in laboratory tank experiments [1].
   - Investigation of plastic deformation of layered geological structures resulting in formation of diapirs (salt domes), in centrifuge experiments scaled down for laboratory size objects.
   - 'Sandbox' experiments in a tectonics/structural geology context for the investigation of deformation processes on various scales ranging from single folds of geological layering to full scale lithospheric processes.
2. Theoretical (mathematical) modelling. Theoretical models can be constructed for geophysical problems by formulating mathematical model equations. These equations are often obtained as partial differential equations from physical conservation laws. An example that will be frequently used in following chapters is the (heat)diffusion equation for a static medium. This equation is derived from a (thermal) energy balance equation,

$$\int_V \rho c_p \frac{\partial T}{\partial t} dV = -\int_{\partial V} \mathbf{q} \cdot \mathbf{n} dA + \int_V \rho H dV \tag{1.1}$$

[1] Vatteville, J., van Keken, P.E., Limare, A. and A. Davaille, *Geochmistry Geophysics Geosystems*, **10**, 2009, Q12013, doi:10.1029/2009GC002739.

where $\mathbf{q} \cdot \mathbf{n}$ is the heatflow density across the closed boundary surface ∂V, of the control volume V, where ρ is the density, c_p is the specific heat at constant pressure, ρH is the volumetric density of the internal heat production rate and H the heating rate per unit mass. Such models have a long tradition in geophysics. The model equations are solved analytically [2] and the solution for example a temperature field - expressed as a series solution of the heat equation - is then evaluated numerically. We find an example of this in the Fourier series solution for the initial value problem for a cooling layer where the bottom and top are kept at fixed zero temperature.

$$T(z,t) = \sum_{n=1}^{\infty} A_n \sin(n\pi z/d) \exp\left(-(n\pi/d)^2 \kappa t\right) \tag{1.2}$$

$$A_n = \frac{2}{d} \int_0^d T(z,0) \sin(n\pi z/d)\ dz \tag{1.3}$$

Analytical solution methods are limited to idealized models, for instance with a simple geometry of the domain and with (piecewise) uniform material properties that occur as coefficients in the model equations. An example of the latter is the uniform thermal diffusivity κ in the above Fourier solution [3].

3. In those cases where the mathematical equations can not be solved analytically, numerical modelling can be a good alternative. Characteristics of numerical modelling methods are:

   - The numerical solution for the unknown field is constructed for a discrete set of grid points in the solution domain (space, time). This is known as the domain discretization.
   - Using discretization methods the continuous model equations are transformed into coupled discrete equations for the unknown nodal point values referred to as the degrees of freedom (dof), i.e. the unknowns, of the discrete problem.
   - The discrete equations are solved numerically for given model parameters such as initial conditions, boundary conditions and coefficients of the original equation (like the thermal diffusivity in the heat equation). Because of the size of such problems involving many thousands or even millions of degrees of freedom these solutions are obtained using numerical methods and computers. [4]
   - The result of such computations is a list of numbers corresponding to the nodal point values of the unknown fields. In order to be able to interpret the solution, the results must be visualized using computergraphics tools in a postprocessing step.

In this course a number of numerical methods is presented that can be applied in modelling a range of geodynamical processes. Numerical modelling is defined here as: investigation of the behavior of a mathematical model by means of numerical solution of the governing equations for different values of the model parameters.

Geophysical modelling problems where numerical methods are applied succesfully are:

[2] H.S. Carslaw and J.C. Jaeger, *Conduction of heat in solids*, Oxford University press, 1959

[3] $\kappa = k/(\rho c_p)$ is the thermal diffusivity and k is thermal conductivity.

[4] State of the art computational methods using high-end supercomputers nowadays involve models with $\sim 10^9$ degrees of freedom.

- (Energy/mass) transport models applied in studies of the thermal state of the lithosphere or in models applied in mantle convection studies or in models of fluid flow through porous media. The latter models are used extensively in hydrology and related areas of environmental engineering and in reservoir engineering in oil and gas production or geothermal applications.
- Flow problems: viscous flow models applied to mantle convection and postglacial rebound.
- Deformation problems: elastic and plastic deformation models are applied to problems of widely varying scale, from detailed geological centimeter/meter scale models for deforming layers to large scale lithosphere models with several lithospheric plates.

The (geo)physical applications investigated in this course deal with numerical solution of time and space dependent equations represented by coupled partial differential equations. In introducing these methods we first apply a spatial discretization. This involves replacement of the unknown field with its continuous dependence of the spatial coordinates by a set of dicrete degrees of freedom. In doing so the partial differential equation (PDE) is replaced by a set of ordinary differential equations (ODE), with time as the remaining independent variable. This step is known as semi-discretization of the problem. The system of ODE's is then integrated in time with a numerical integration method starting from given initial values of the degrees of freedom.

Well known discretization methods are:

- finite difference and finite volume methods
- finite element methods
- spectral methods

The first two categories are presented in this course.

Well known integration methods for ODE's are:

- Euler explicit/implicit also known as Euler forward/backward
- Crank-Nicolson
- Runge-Kutta

The first two of these will be presented in this course.

Chapter 2

Numerical modelling of 1-D heat conduction

As first example of a numerical modelling problem we consider the solution of the (heat) conduction problem in a static (non-moving) medium. This problem is encountered for instance when we want to model the temperature distribution in the earth's lithosphere, for given distribution of heat-producing (radiogenic) elements and given heat flux from the mantle into the lithosphere.

In later chapters we will also consider more general problems where besides conductive transport also convective heat transport occurs as in the proces of large scale thermal convection in the earth's mantle or convective heat transport in porous media encountered in geothermal systems.

To keep things simple at this point we only deal with 1-D problems here. Formulating the heatconduction problem for the lithosphere as a 1-D problem is a reasonable approximation. On a sufficiently large scale a horizontally layered model is applicable where heatflow is mainly in the vertical direction. In this setup the temperature depends on only one spatial coordinate, say the depth coordinate z, with $z = 0$ corresponding to the Earth's surface. In time dependent cases we have two independent variables for the single dependent variable, the temperature $T(z, t)$.

The numerical solution methods treated here can be generalized for multi dimensional 2-D and 3-D problems. In Chapter 3 similar methods are treated for 2-D problems.

We start with the time dependent heat equation,

$$\rho c_p \frac{\partial T}{\partial t} = \nabla \cdot k \nabla T + \rho H \tag{2.1}$$

where T is the temperature, k the thermal conductivity, ρ the mass density and c_p the specific heat at constant pressure. Note: at this point the coefficients ρ, c_p, k and ρH are not assumed to be uniform parameters in (2.1).

problem 2.1. *Derive equation (2.1) from a conservation principle for thermal energy expressing the balance between internal heat production and surface heatflux for an arbitrary control volume in a static (i.e. non-moving) medium discussed in the introduction chapter.*

Hint: *Consider the integral heat balance equation (1.1). Use a coordinate free formulation and apply Gauss divergence theoreme and Fouriers law for heat conduction* $\mathbf{J} = -k\nabla T$ *for an arbitrary control volume* V*.*

problem 2.2. *Verify the physical dimensions and units of the terms in the above equation (2.1).*

An important application of the time dependent 1-D heat equation for a static medium is

the half space problem applied to a column of material in a spreading oceanic lithosphere cooling from the top, (Turcotte & Schubert, 2002). [1] In this example the advection term in the time derivative for the (horizontally) moving medium with velocity u, $dT/dt = \partial T/\partial t + u\partial T/\partial x$ is eliminated with an age transformation, $\tau = x/u$, into $\partial T/\partial \tau$. If we also assume a steady state with $\partial T/\partial t = 0$, and we neglect horizontal diffusion of heat [2] we obtain a time dependent equation with the lithospheric age as the time variable.

$$\rho c_p \frac{\partial T}{\partial \tau} = \frac{\partial}{\partial z} k \frac{\partial T}{\partial z} + \rho H \tag{2.2}$$

In the following we will write t instead of τ for the independent time variable.

The thermal problem is specified for a 1-D solution domain, on the interval $[z_0, z_b]$. For the time dependent problem we assume the initial temperature for $t = 0$, to be given in the initial condition

$$T(z, 0) = T_I(z) \tag{2.3}$$

where $T_I(z)$ is a known function. We shall consider two types of boundary conditions. In the first type, known as a Dirichlet condition or also as an essential boundary condition, the temperature is described,

$$T(z_b, t) = T_b(t) \ , \ type\ 1\ (Dirichlet) \tag{2.4}$$

where z_b is a boundary point and $T_b(t)$ is a function of time. In the second type, known as Neumann or natural boundary condition, the first derivative of the temperature or heat flow density q_b is specified,

$$k \frac{\partial T(z_b, t)}{\partial z} = q_b(t) \ , \ type\ 2\ (Neumann) \tag{2.5}$$

where $q_b(t)$ is a given function of time.

We consider two discretization methods. The first, based on a central difference approximation of the spatial derivatives in the heat equation is conceptually simpler. The second method (finite volume or box method) is suitable for more general problems, for instance with variable conductivity k.

For the steady state case ($\partial T/\partial t = 0$) both methods result in a system of linear algebraic equations that can be solved numerically. For the time dependent equation ($\partial T/\partial t \neq 0$) discretization results in a set of ordinary differential equations (ODE) with time t as the independent variable. In Chapter 8 we shall treat a number of integration methods for such sets of ODE's.

2.1 A finite difference method with equidistant grid

First we consider a discretization method for an equidistant grid of nodal points, applied for the special case with uniform coefficient k. The latter assumption implies that the conduction term in (2.2) can be written as a second derivative, $k\partial^2 T/\partial z^2$. The main advantage of this method is its conceptual simplicity. In section 2.2 a more generally applicable method will be introduced.

First we define a 1-D grid on the domain, with uniform distance Δz between the nodal points (equidistant),

$$z_i \ = \ z_0 + i \times \Delta z \ , \ i \ = \ 0, 1, \ldots, N+1 \tag{2.6}$$

[1] D.L. Turcotte and G. Schubert, *Geodynamics*, Cambridge University Press, 2002, 2nd edition.

[2] $\frac{\partial}{\partial x}\left(k \frac{\partial T}{\partial x}\right) \approx 0$

The boundary points are z_0 and z_{N+1} and a part of the grid is shown in Fig. 2.1.

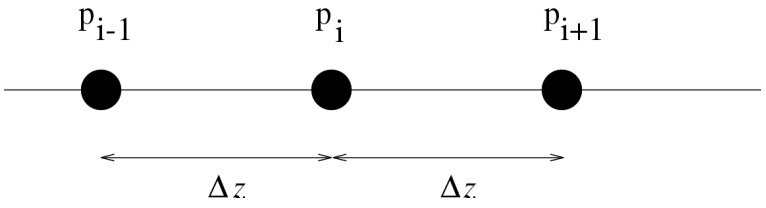

Figure 2.1: *Detail of an equidistant grid showing two segments of length Δz*

.

We define vectors $\mathbf{T}(t)$, $\mathbf{W}(t)$ where the nodalpoint values of the temperature $T_i(t) = T(z_i, t)$ and the heat productivity $W_i(t) = \rho(z_i, t)H(z_i, t)$ are the respective vector elements. We drop the explicit notation of the time dependence for convenience.

The derivative $\partial^2 T/\partial z^2$ in the heat equation is approximated by a central difference formula that can be derived from a Taylor expansion of the temperature field,

$$T(z + \Delta z) = T(z) + \Delta z\frac{\partial T}{\partial z} + \frac{\Delta z^2}{2}\frac{\partial^2 T}{\partial z^2} + \frac{\Delta z^3}{6}\frac{\partial^3 T}{\partial z^3} + \frac{\Delta z^4}{24}\frac{\partial^4 T}{\partial z^4} + \ldots \quad (2.7)$$

$$T(z - \Delta z) = T(z) - \Delta z\frac{\partial T}{\partial z} + \frac{\Delta z^2}{2}\frac{\partial^2 T}{\partial z^2} - \frac{\Delta z^3}{6}\frac{\partial^3 T}{\partial z^3} + \frac{\Delta z^4}{24}\frac{\partial^4 T}{\partial z^4} - \ldots \quad (2.8)$$

eliminating odd powers of Δz, we obtain:

$$T(z + \Delta z) - 2T(z) + T(z - \Delta z) = \Delta z^2\frac{\partial^2 T}{\partial z^2} + \frac{\Delta z^4}{12}\frac{\partial^4 T}{\partial z^4} + \ldots \quad (2.9)$$

$$\frac{\partial^2 T}{\partial z^2} = \frac{T(z - \Delta z) - 2T(z) + T(z + \Delta z)}{\Delta z^2} - \frac{\Delta z^2}{12}\frac{\partial^4 T}{\partial z^4} + \ldots \quad (2.10)$$

Neglecting terms in Δz^2 and higher order in Δz in (2.10) and evaluating the (semi) discretized equation in the nodal points z_i, results in a system of ordinary differential equations (ODE),

$$(\rho c_p)_i \frac{dT_i}{dt} = \frac{k}{\Delta z^2}\left[T_{i-1} - 2T_i + T_{i+1}\right] + W_i\ ,\ i = 1, 2, \ldots, N \quad (2.11)$$

where $(\rho c_p)_i = \rho(z_i)c_p(z_i)$.

In case of prescribed temperature values in both boundary points we have, $T(z_0) = T_0$, $T(z_{N+1}) = T_{N+1}$, with T_0 and T_{N+1} the known boundary values. The resulting equations for the first ($i = 1$) and last ($i = N$) interior points get a contribution containing the boundary values T_0 and T_{N+1}.

$$i = 1 \rightarrow (\rho c_p)_1 \frac{dT_1}{dt} = \frac{k}{\Delta z^2}\left[-2T_1 + T_2\right] + \frac{k}{\Delta z^2}T_0 + W_1 \quad (2.12)$$

$$i = N \rightarrow (\rho c_p)_N \frac{dT_N}{dt} = \frac{k}{\Delta z^2}\left[T_{N-1} - 2T_N\right] + \frac{k}{\Delta z^2}T_{N+1} + W_N \quad (2.13)$$

Evaluation of (2.11) for all the internal nodal points with the above procedure, results in a system of N ordinary differential equations,

$$\mathbf{M}\frac{d\mathbf{T}}{dt} + \mathbf{AT} = \mathbf{R} \quad (2.14)$$

where $\mathbf{M}$ is a diagonal heat capacity matrix, $M_{ij} = (\rho c_p)_i \, \delta_{ij}$, [3] and the righthand side vector is,

$$\mathbf{R} = \left(W_1 + \frac{k}{\Delta z^2} T_0, W_2, \ldots, W_{N-1}, W_N + \frac{k}{\Delta z^2} T_{N+1} \right) \tag{2.15}$$

and $N \times N$ coefficient matrix,

$$\mathbf{A} = \frac{k}{\Delta z^2} \begin{bmatrix} 2 & -1 & 0 & 0 & . & . & . & . \\ -1 & 2 & -1 & 0 & . & . & . & . \\ 0 & -1 & 2 & -1 & . & . & . & . \\ 0 & 0 & -1 & 2 & . & . & . & . \\ . & . & . & . & . & . & . & . \\ . & . & . & . & . & 2 & -1 & 0 \\ . & . & . & . & . & -1 & 2 & -1 \\ . & . & . & . & . & 0 & -1 & 2 \end{bmatrix} \tag{2.16}$$

and vector of unknowns,

$$\mathbf{T}(t) \;=\; (T_1(t), T_2(t), \ldots, T_N(t)) \tag{2.17}$$

problem 2.3. *Verify that (2.14) represents a complete set of equations i.e. the number of degrees of freedom is equal to the number of independent equations.*

problem 2.4. *In a minimal mesh configuration applied to the above problem we have a single internal nodal point besides the two boundary points. Assume absence of internal heat production, $W = 0$. We assume boundary values for the original continuous problem $(T)_{z=0} = 0$ and $(T)_{z=1} = 1$, constant in time. The system of coupled ODE's (2.14) reduces to a single ODE with continous scalar variable $T(t)$. Solve the initial value problem of equation (2.14), for an arbitrary initial temperature value $T(0)$.*
Answer:

$$T(t) = \left(T(0) - \frac{1}{2} \right) \exp\left(-\frac{2\kappa}{\Delta z^2} t \right) + \frac{1}{2} \tag{2.18}$$

Where $\kappa = k/(\rho c_p)$ is the thermal diffusivity.

Verify that the limiting solution is the analytic solution of the corresponding steady state problem, $\lim_{t\to\infty} T(z,t) = 1/2$, for $z = 1/2$.

2.1.1 Implementation of natural boundary conditions

The treatment of natural boundary conditions differs from the one for essential boundary conditions applied in the previous section. We consider a case with prescribed heatflow density q_{N+1} in the nodal point p_{N+1}. The temperature value in p_{N+1} is now also an unknown quantity (degree of freedom) of the problem and we must evaluate the discrete equation (2.11) also in p_{N+1} in order to get a complete set of equations.

A simple way to implement the prescribed heatflow density is to expres the latter in the nodal point temperature values by means of a difference approximation. We approximate the temperature gradient in the expression for the heatflow density by a central difference formula,

$$q_{N+1} = k \left(\frac{\partial T}{\partial z} \right)_{z_{N+1}} = k \frac{T_{N+2} - T_N}{2\Delta z} + O(\Delta z^2) \tag{2.19}$$

[3] δ_{ij} is the Kronecker delta,

$$\delta_{ik} = \begin{cases} 1 & i = k \\ 0 & i \neq k \end{cases}$$

Note that we have used a virtual gridpoint p_{N+2} at a distant Δz outside the domain. For the temperature in this virtual point we find from (2.19) (neglecting the second order term in Δz),

$$T_{N+2} = T_N + 2\frac{q_{N+1}}{k}\Delta z \tag{2.20}$$

Using (2.20) we can formulate the discrete equation in the boundary point p_{N+1}, resulting in,

$$(\rho c_p)_{N+1}\frac{dT_{N+1}}{dt} = \frac{2k}{\Delta z^2}\left[T_N - T_{N+1}\right] + \frac{2k}{\Delta z}\frac{q_{N+1}}{k} + W_{N+1} \tag{2.21}$$

We see that the inhomogeneous boundary condition results in a contribution to the right-hand side vector. Note that the coefficient matrix **A** is no longer symmetric. However symmetry can easily be obtained by dividing the equation for the boundary point by 2.

problem 2.5. *(2.20) shows that (2.19) is equivalent to a linear extrapolation of the temperature field. Show by Taylor expansion that the approximation of the boundary condition applied in (2.19) is indeed of second order accuracy in Δz and show that a forward difference formula results in first order accuracy.*

2.2 A difference method with variable grid spacing

In the derivation of the finite difference method for the heat equation based on central difference approximation of the conduction term we used an equidistant grid of nodal points. We further assumed that the thermal conductivity coefficient was a constant.

To obtain sufficient accuracy in the numerical solution it may be necessary to use many gridpoints in a high resolution mesh, resulting in larger program requirements for memory and compute time. Such mesh refinement is applied on the whole domain in case of an equidistant grid whereas increased resolution may be necessary only on part of the domain where the solution shows strong variations (large gradient). It is clear that in such cases using equidistant grids is not efficient and methods allowing local grid refinement will be more efficient. Different methods exist allowing local refinement. In later chapters we focus on so called finite element methods which offer the most flexibility in local grid refinement of well known discretization methods.

As an example of a method allowing local grid refinement we treat here an other difference method which also includes a simple treatment of variable coefficients $k(z)$. We restrict ourselves again to the 1-D case. A 2-D generalization is introduced in Chapter 3. The method introduced here is known in the literature as a *finite volume method*. We first deal with the steady state problem and shall verify afterwards how this can be extended for time dependent problems.

2.2.1 Discretization of the equation

In the finite volume method the partial differential equation (PDE) is integrated over small grid cell's, the so called finite volumes. In our 1-D case the finite volumes are subintervals I_i, of the complete domain, the interval $I = [0, L]$.

These subintervals $I_i = [z_{m_{i-1}}, z_{m_i}]$, centered at nodalpoint p_i, are illustrated Fig.2.

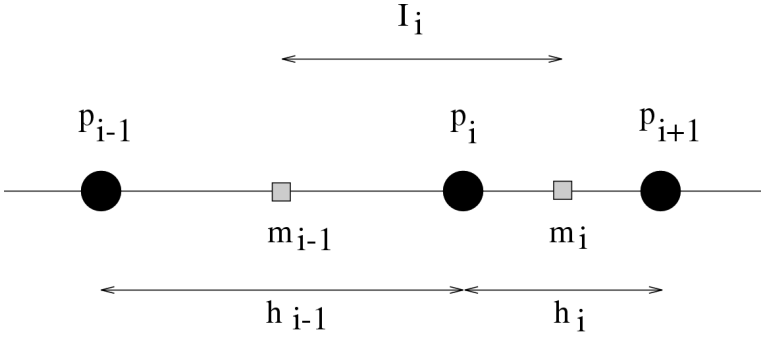

Figure 2.2: *Detail of a 1-D grid with two grid segments and a integration interval of finite volume I_i.*

The mid-points of the grid cells, which span the finite volumes, are labeled m_i. Integration of the steady state heat equation over I_i yields,

$$\int_{z_{m_{i-1}}}^{z_{m_i}} \left(\frac{d}{dz} k(z) \frac{d}{dz} T \; + \; \rho(z) H(z) \right) dz = \left[k(z) \frac{dT}{dz} \right]_{z_{m_{i-1}}}^{z_{m_i}} + \int_{z_{m_{i-1}}}^{z_{m_i}} \rho(z) H(z) dz = 0 \tag{2.22}$$

The remaining derivative in (2.22) is evaluated in the mid-points m_{i-1} and m_i. We approximate these derivatives by their central difference approximations in terms of the neighboring nodal point values and we define $k(z_{m_i}) = k_i$ and $z_{p_{i+1}} - z_{p_i} = h_i$.

$$\left(k \frac{dT}{dz} \right)_{z_{m_i}} \approx k_i \frac{T_{i+1} - T_i}{z_{p_{i+1}} - z_{p_i}} = k_i \frac{T_{i+1} - T_i}{h_i} \tag{2.23}$$

$$\left(k \frac{dT}{dz} \right)_{z_{m_{i-1}}} \approx k_{i-1} \frac{T_i - T_{i-1}}{z_{p_i} - z_{p_{i-1}}} = k_{i-1} \frac{T_i - T_{i-1}}{h_{i-1}} \tag{2.24}$$

The distribution of the heat productivity H is assumed to be known and we define,

$$\int_{z_{m_{i-1}}}^{z_{m_i}} \rho(z) H(z) dz = F_i \tag{2.25}$$

Substitution of (2.23),(2.24) and (2.25) in (2.22) gives,

$$-\frac{k_{i-1}}{h_{i-1}} T_{i-1} \; + \; \left(\frac{k_{i-1}}{h_{i-1}} \; + \; \frac{k_i}{h_i} \right) T_i - \frac{k_i}{h_i} T_{i+1} = F_i \tag{2.26}$$

Equation (2.26) is a linear algebraic equation in the unknown nodal point values of the temperature T_i. By repeating the integration proces for all N finite volumes I_i we obtain a system of linear equations.

The resulting system of equations written in matrix form is,

$$\mathbf{AT} = \mathbf{F} \tag{2.27}$$

problem 2.6. *Verify that the system of equations obtained above is complete in case of prescribed boundary temperatures in $z = 0$ and $z = L$.*

problem 2.7. *Show that the system of equations build from (2.26) is identical to the equations obtained in the previous section for the special case of an equidistant grid and uniform coefficient k and piecewise uniform internal heating H (see problem 2.9).*

problem 2.8. *Show that the matrix* **A** *is a symmetric tri-diagonal matrix, i.e.* $A_{ij} = A_{ji}$, $A_{ij} = 0, |i - j| > 1$.

problem 2.9. *Suppose we whish to apply the finite volume method to a 1-D medium consisting of a stack of layers with uniform conductivity in each layer and that the conductivity is discontinuous in the layer interfaces. The conductivity model in this case is said to be piecewise uniform and consists of a list of discrete layer conductivity values* k_i. *Where would you put the nodal points in this model such that all the necessary entities in the derivation above are well defined?*

problem 2.10. *Suppose we define the heat productivity coefficient* $H(z)$ *by a piecewise constant model. Verify the following formula for the righthand side vector elements,*

$$F_i = \int_{z_{m_{i-1}}}^{z_{m_i}} \rho(z)H(z)dz = \frac{h_{i-1}}{2}\rho(z_{m_{i-1}})H(z_{m_{i-1}}) + \frac{h_i}{2}\rho(z_{m_i})H(z_{m_i}) \tag{2.28}$$

problem 2.11. *Assume that the heatproductivity is concentrated in a point,* $z = z_s$, $W(z) = \rho(z)H(z) = W_s\delta(z - z_s)$, *with* $z_{m_{k-1}} < z_s < z_{m_k}$, *i.e.* $z_s \in I_k$.

Derive for the righthand side vector elements, $F_i = W_s\delta_{ik}$, *where* δ_{ik} *is the Kronecker delta symbol.*

problem 2.12. *Investigate how the steady state equation (2.27) can be extended to a set of ODE's similar to (2.14) for the time dependent case. How would you treat a case with variable heat capacity* ρc_p *in this extension?*

2.2.2 Implementation of boundary conditions

Essential boundary conditions are implemented in the same way as in the discretization method of section 2.1. Essential boundary conditions result in a contribution to the right-hand side vector and a reduction of the number of degrees of freedom of the problem. Here we describe the implementation of natural or Neumann boundary conditions that are used in case the heatflow density is prescribed on the boundary.

We define $(kdT/dz)_{z=z_{N+1}} = q_{N+1}$. The implementation differs from the one described in section 2.1.1 for an equidistant grid. Here we derive the implementation for the boundary condition in nodal point p_{N+1}, illustrated in Fig.3.

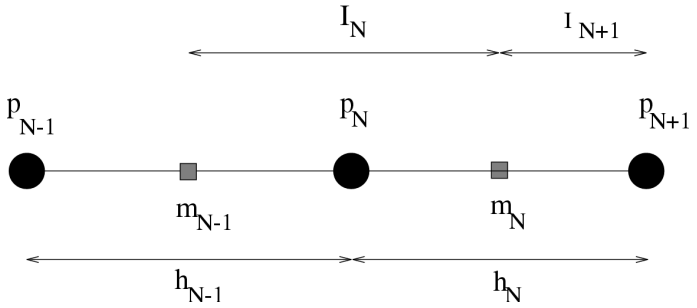

Figure 2.3: *Detail of a 1-D grid with two grid segments including a boundary point and two finite volumes.*

In this case the temperature in the boundary point p_{N+1} is also a degree of freedom of the problem. In order to obtain a complete set of equations we need to introduce an extra equation by integrating over a (half) finite volume from m_N to p_{N+1}. We apply (2.22) for the nodal points p_N and p_{N+1}.

For p_N we get with 2.2.1(2.26)

$$-\frac{k_{N-1}}{h_{N-1}}T_{N-1}+\left(\frac{k_{N-1}}{h_{N-1}}+\frac{k_N}{h_N}\right)T_N-\frac{k_N}{h_N}T_{N+1}=F_N \tag{2.29}$$

The boundary temperature T_{N+1} now appears in the lefthand side as a degree of freedom.

For the boundary point p_{N+1} with prescribed heatflow density we apply the integral formula (2.22) and we choose the integration interval $I_{N+1}=[z_{m_N},z_{p_{N+1}}]$,

$$\begin{aligned}\int_{z_{m_N}}^{z_{p_{N+1}}}\left(\frac{d}{dz}k\frac{d}{dz}T+\rho H\right)dz=\\ \left(k_N\frac{dT}{dz}\right)_{z_{p_{N+1}}}-\left(k_N\frac{dT}{dz}\right)_{z_{m_N}}+\int_{z_{m_N}}^{z_{p_N+1}}\rho H(z)dz\approx\\ q_{N+1}-k_N\frac{T_{N+1}-T_N}{h_N}+F_{N+1}=0\end{aligned} \tag{2.30}$$

This results in an equation for the nodal point p_{N+1},

$$-\frac{k_N}{h_N}T_N+\frac{k_N}{h_N}T_{N+1}=F_{N+1}+q_{N+1} \tag{2.31}$$

Note that the prescribed heatflow density appears as a contribution in the righthand side of the equation.

problem 2.13. *Show that (2.31) is equivalent with the steady state case of (2.21) for the special case with uniform grid spacing $h_i=\Delta z$ and uniform coefficient $k_N=k$.*

problem 2.14. *Consider the problem of the conductive cooling of a hot sphere in a cool environment of uniform and constant temperature, described by the time dependent equation (2.1). We assume that the relevant parameters have a 1-D spherically symmetric distribution, completely described by the radial coordinate r and take the coordinate origin in the centre of the sphere. The corresponding 1-D problem is discretized by defining the values of the radii of N nodal surfaces p_i on the radial axis, illustrated in Fig.2.4. p_1 corresponds to the central point $r=0$ and p_N represents the outer surface $r=R$ of the spherical body. $N-1$ midpoint surfaces m_i are defined halfway each pair of nodal surfaces $p_i,\ p_{i+1}$*

Apply a finite volume approach to transform the PDE (2.4) into a system of ODE's. To this end integrate the PDE over a spherical shell defined by the radial interval $I_i=[r_{m_{i-1}},r_{m_i}]$ in Fig.2.2, that includes the nodal point p_i.

1. *Derive the following ODE system from the integration of the PDE over the spherical shell spanned by the radial interval I_i,*

$$C_i\frac{\partial T_i}{\partial t}=\frac{k^*_{i-1}}{h_{i-1}}T_{i-1}-\left\{\frac{k^*_{i-1}}{h_{i-1}}+\frac{k^*_i}{h_i}\right\}T_i+\frac{k^*_i}{h_i}T_{i+1}+F_i\ ,\quad i=2,\ldots,N-1 \tag{2.32}$$

where $h_i=r_{pi+1}-r_{p_i}$ and $k^*_i=r^2_{m_i}k(r_{m_i})$ and

$$C_i=\frac{1}{3}\left(r^3_{m_i}-r^3_{m_{i-1}}\right)(\rho c_p)_i\ ,\ F_i=\frac{1}{3}\left(r^3_{m_i}-r^3_{m_{i-1}}\right)(\rho H)_i \tag{2.33}$$

Hints:

- *Use the following expression for the conduction term in the PDE,*

$$\nabla \cdot k\nabla T = \frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2 k\frac{\partial T}{\partial r}\right) \tag{2.34}$$

- *Use the following expression for the volume integral over the volume V of a spherical shell of a radially symmetric function $f(\mathbf{x}) = f(r)$, $\int_V f(\mathbf{x})dV = 4\pi\int_{r_{m_{i-1}}}^{r_{m_i}} f(r)r^2dr$. For the time dependent term and the internal heating term, the integral can be interpretd in terms of a local (shell) average that can be approximated in the following way, $4\pi\int f(r)r^2dr = \langle f\rangle V_{shell} \approx f(p_i)V_{shell}$.*
- *Approximate the first order derivatives on the midpoint surfaces that remain after integration, by central differences, expressed in the nodal point values, for instance,*

$$\left(r^2 k\frac{\partial T}{\partial r}\right)_{r=r_{m_i}} \approx r_{m_i}^2 k_i \frac{T_{i+1}-T_i}{r_{p_{i+1}}-r_{p_i}} = k_i^* \frac{T_{i+1}-T_i}{h_i} \tag{2.35}$$

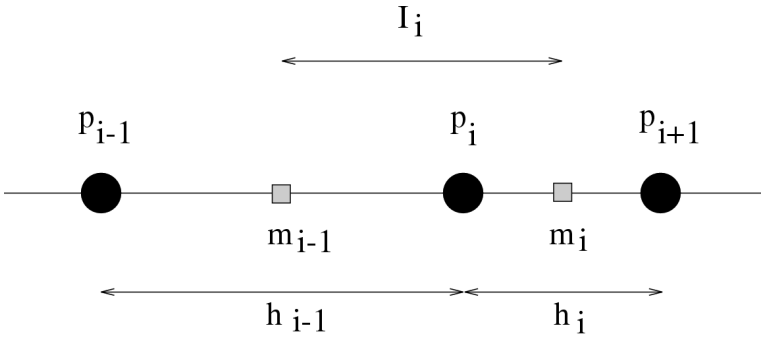

Figure 2.4: *Discretization of the radial axis. Distribution of 1-D nodal surfaces corresponding to radial levels p_i and midpoint surfaces m_i halfway between the nodal surfaces.*

2. *Equation (2.32) pertains to the internal nodal surfaces of the domain. Explain why the resulting set of equations is not complete and how it can be made complete by implementation of apropriate boundary conditions for this problem: an essential boundary condition for the outer surface $r = R$ and a symmetry condition for the centre of the sphere $r = 0$.*

Chapter 3

Difference methods for 2-D potential problems

3.1 Introduction

In the previous chapter numerical solution of 1-D heat diffusion problems was treated. Two alternative methods were introduced for the spatial discretization of the governing partial differential equation. A method based on central differences and a finite volume method. In this chapter we shall extend both methods for application to 2-D problems. Further extensions to 3-D problems are straightforward.

We start with the steady state problem, described by a second order partial differential equation and two types of boundary conditions. Extension to the time dependent problem is treated in section 3.3. The problem to be solved can be formulated for multi-dimensional configurations on a solution domain V with boundary $\Gamma = \partial V$ in the following way,

$$-\nabla \cdot c(\mathbf{x})\nabla u = -\partial_x(c\,(\mathbf{x})\partial_x u) - \partial_y(c\,(\mathbf{x})\partial_y u) = f(\mathbf{x}), \quad \mathbf{x} \in V \tag{3.1}$$

We will frequently chose combinations from the following two types of boundary conditions,

$$u(\mathbf{x}) = g(\mathbf{x}),\ \mathbf{x} \in \Gamma_g \tag{3.2}$$

$$c(\mathbf{x})\nabla u(\mathbf{x}) \cdot \mathbf{n}(\mathbf{x}) = h(\mathbf{x}),\ \mathbf{x} \in \Gamma_h \tag{3.3}$$

The two types of boundary conditions considered here are defined separately for the two non-overlapping sub-boundaries, $\Gamma = \Gamma_g \cup \Gamma_h,\ \Gamma_g \cap \Gamma_h = \emptyset$. [1] In (3.2),(3.3) g and h are given functions of the coordinates $\mathbf{x}$.

Boundary conditions (3.2) and (3.3) are known as essential or Dirichlet and natural or Neumann type conditions respectively. In applications in thermal problems essential boundary conditions correspond to prescribed boundary temperatures and in case of natural boundary conditions the heatflow density is prescribed ($\mathbf{q} \cdot \mathbf{n} = -k\nabla T \cdot \mathbf{n} = -k\partial_n T = h$).

problem 3.1. *In (DC) electric exploration methods the electric potential field u is formulated by a Poisson equation. The electric field $\mathbf{E}$ and the electric current density $\mathbf{J}$ are defined as $\mathbf{E} = -\nabla u$, $\mathbf{J} = \sigma\mathbf{E}$, where σ is the electrical conductivity.*

Give a physical interpretation for essential and natural boundary conditions for this problem. What is the interpretation of the right hand side function $f(\mathbf{x})$ in the differential equation (3.1)?

[1] The binary set operator symbols $\cup$ and $\cap$ denote respectively the union and cross-sections of the sets involved and $\emptyset$ represents the empty set.

In 2-D finite difference methods the domain V is discretized with a grid of N nodal points $\mathbf{x}_I = (x_I, y_I)$ and the unknown function $u(\mathbf{x})$ is replaced by a vector of nodal point values,

$$\mathbf{U} = (u(\mathbf{x}_1), u(\mathbf{x}_2), \cdots u(\mathbf{x}_N))^T \tag{3.4}$$

In the previous chapter we have seen how, for a 1-D problem with uniform coefficient $c = 1$, the PDE can be discretized using a central difference approximation for the second derivative. This approach is extended for multi-dimensional problems in the next section.

3.2 A central difference method

We shall use a rectangular geometry of the domain V and we define an equidistant 2-D grid of nodal points by,

$$\mathbf{x}_{ij} = (x_i, y_j) = (x_0 + i \times h, y_0 + j \times h),\ i = 0, 1, \ldots, n_{col}+1,\ j = 0, 1, \ldots, n_{row} = 1 \tag{3.5}$$

(3.5) defines a 2-D *equidistant* grid with a total of $(n_{col} + 2) \times (n_{row} + 2)$ nodal poinys and $n_{col} \times n_{row}$ internal nodal points.

problem 3.2. *Derive the following difference approximation of the 2-D Laplace operator,*

$$\begin{aligned} \nabla^2 u(\mathbf{x}_{ij}) &\approx D_h^2 u(\mathbf{x}_{ij}) \\ &= \frac{(u(x_i + h, y_j) + u(x_i - h, y_j) - 4u(x_i, y_j) + u(x_i, y_j + h) + u(x_i, y_j - h))}{h^2} \\ &= \frac{(u(x_{i+1}, y_j) + u(x_{i-1}, y_j) - 4u(x_i, y_j) + u(x_i, y_{j+1}) + u(x_i, y_{j-1}))}{h^2} \end{aligned} \tag{3.6}$$

Show that the local truncation error in the discretized Laplace operator defined as,

$$E = D_h^2 u(\mathbf{x}_{ij}) - \nabla^2 u(\mathbf{x}_{ij}) \tag{3.7}$$

is of second order in the grid spacing h, i.e. $E = O\left(h^2\right)$.
Hint: *expand the functions in the difference formula in a Taylor series in h in the neighborhood of the grid point $\mathbf{x}_{ij}$. Do this separately for both x and y dependence. Note: in order to obtain an explicit formula for the truncation error it is necessary to expand the Taylor series up to and including the fourth order.*

The degrees of freedom of the discretized problem have been organized in an N vector $\mathbf{U} \in \mathbb{R}^N$ in (3.4) and the sequence of the vector components depends on the nodalpoint numbering of the finite difference mesh, mapping the grid indices i, j (column index, row index) onto the index I of the N-vector $\mathbf{U}$. Assuming n_{row} rows and n_{col} columns in a rectangular grid, a straightforward mapping is obtained by the following column wise numbering of the nodal points $I = (i_{col} - 1)n_{row} + j_{row}$,

$$U_I = u(\mathbf{x}_I) = u(x_{i_{col}}, y_{j_{row}}),\ i_{col} = 1, \ldots, n_{col},\ j_{row} = 1, \ldots, n_{row} \tag{3.8}$$

With this columnwise numbering of the gridpoint values the difference operator is written in terms of the vector elements of $\mathbf{U}$ as follows,

$$D_h^2 \mathbf{U} = U_{I+n_{row}} + U_{I-n_{row}} - 4U_I + U_{I+1} + U_{I-1} \tag{3.9}$$

In case of a boundary value problem with prescribed values of the potential on the entire boundary, a Dirichlet type boundary condition (3.2), the described discretization and nodal point numbering result in a system of linear algebraic equations for the internal

nodal point values of the potential function. The matrix corresponding to these equations is defined **S**.

The bandwidth w of this matrix **S** is defined by the location of non-zero diagonals in the upper and lower triangle matrix, excluding the main diagonal,

$$w = max|I - J|, \quad S_{IJ} \neq 0 \tag{3.10}$$

According to this definition a diagonal matrix has a zero bandwidth and tri-diagonal matrix has a bandwidth of one. Interpreting the expression (3.9) as a matrix row in a system of linear algebraic equations it follows that the bandwidth $w = n_{row}$. From this we find that the smallest bandwidth is obtained by defining the grid columns in the direction of the smallest dimension of the rectangular domain. Efficient algorithms are available for the solution of systems of linear algebraic equations that are based on a so called bandmatrix structure where only matrix elements within the bandwidth defined in (3.10) are stored in computer memory. Minimizing the bandwith of the matrix will result in minimizing the computer requirements (memory, compute time) when using such bandmatrix solvers.

problem 3.3. *The operator D_h^2 in (3.9), applied to the nodal point values of a uniform grid, produces a system of linear algebraic equations for a discrete approximation of (3.1). Consider the special case of a rectangular grid with two rows of internal gridpoints and apply a (grid) columnwise numbering of the degrees of freedom of the problem as in (3.8).*
What is the structure of the matrix **S** *of the resulting system?*

3.3 A difference method for variable grid spacing and variable coefficient

The difference formula for the Laplace operator (3.6) is derived for the special case of a uniform coefficient c and does not apply to the more general case with differential operator $L = \nabla \cdot c(\mathbf{x})\nabla u$. Besides this the grid used in section 3.2 is equidistant.

Here we introduce the more general case with variable coefficient and apply a so called structured grid, defined by the *product* of two 1-D grids with variable nodal point spacing in both coordinate directions. This allows local mesh refinement. We consider a 2-D rectangular domain subdivided in rectangular grid cells spanned by the nodalpoints, illustrated in Fig. 3.1. Similar as in 2.2, but now for a 2-D domain, we integrate the PDE (3.1) over a small area, a finite volume, surrounding a single gridpoint P_0, illustrated in Fig. 3.1.

Relative coordinates of neighboring gridpoints of P_0, shown in Fig. 3.1, are parameterized as follows,

$$\begin{aligned} P_0 &= (0,0) \\ P_1 &= (s_1 h, 0) \\ P_2 &= (0, s_2 h) \\ P_3 &= (-s_3 h, 0) \\ P_4 &= (0, -s_4 h), \quad 0 < s_K \leq 1, \quad K = 1, \ldots, 4 \end{aligned} \tag{3.11}$$

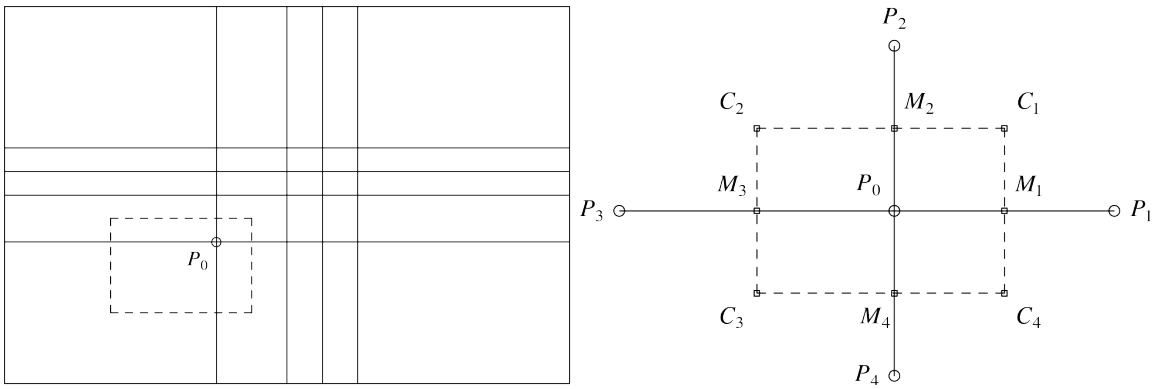

Figure 3.1: *Left: rectangular computational domain showing local mesh refinement and a single (dashed) finite volume. Right: zoom-in finite volume with five grid points in a mesh with variable grid point spacing. $M_K, K = 1, \ldots, 4$ are midpoints positioned halfway two neighboring gridpoints P_0, P_K. A difference equation is derived by integration over the rectangle spanned by the corner points $C_1, \ldots, C_4$.*

The PDE (3.1) is integrated over the rectangular area V spanned by the corner points $C_1, \ldots, C_4$ in Fig.3.1, that are the centre points of the neighboring grid cells.

$$I = \int_V \nabla \cdot c(\mathbf{x}) \nabla u \; dV = \int_{\partial V} c(\mathbf{x}) \partial_j u \; n_j \; dA = - \int_V f(\mathbf{x}) dV \tag{3.12}$$

where ∂V is the closed boundary curve C_1, C_2, C_3, C_4, C_1. The contribution from the vertical boundary segments in (3.12) is

$$I_1 - I_3 = \int_{-s_4 h/2}^{s_2 h/2} \left(c(\mathbf{x}) \partial_x u \right)_{x = s_1 h/2} \; dy - \int_{-s_4 h/2}^{s_2 h/2} \left(c(\mathbf{x}) \partial_x u \right)_{x = -s_3 h/2} \; dy \tag{3.13}$$

Both integrals in (3.13) are approximated using a 'mid-point rule' and the remaining partial derivative is replaced by a central difference approximation,

$$\partial_x u(M_1) \approx \frac{u(P_1) - u(P_0)}{s_1 h} \tag{3.14}$$

The first integral in (3.13) results in,

$$I_1 \approx c(M_1) \left(u(P_1) - u(P_0) \right) \frac{s_2 + s_4}{2 s_1} \tag{3.15}$$

In a similar way we find,

$$I_3 \approx c(M_3) \left(u(P_0) - u(P_3) \right) \frac{s_2 + s_4}{2 s_3} \tag{3.16}$$

The horizontal boundaries result in similar contributions,

$$I_2 = \int_{-s_3 h/2}^{s_1 h/2} \left(c(\mathbf{x}) \partial_y u \right)_{y = s_2 h/2} \; dx \approx c(M_2) \left(u(P_2) - u(P_0) \right) \frac{s_1 + s_3}{2 s_2} \tag{3.17}$$

$$I_4 = \int_{-s_3 h/2}^{s_1 h/2} \left(c(\mathbf{x}) \partial_y u \right)_{y = -s_4 h/2} \; dx \approx c(M_4) \left(u(P_0) - u(P_4) \right) \frac{s_1 + s_3}{2 s_4} \tag{3.18}$$

Putting the segment contributions together [2] in,

$$I = I_1 - I_3 + I_2 - I_4 \tag{3.19}$$

we obtain,

$$I \approx \sum_{K=1}^{4} \alpha_K u(P_K) - \alpha_0 u(P_0) \tag{3.20}$$

The coefficients in (3.20) are given in the Table.

K	α_K
0	$\sum_{K=1}^{4} \alpha_K$
1	$c(M_1)(s_2+s_4)/2s_1$
2	$c(M_2)(s_1+s_3)/2s_2$
3	$c(M_3)(s_2+s_4)/2s_3$
4	$c(M_4)(s_1+s_3)/2s_4$

Table 3.1: *Coefficients of the five-point finite difference 'molecule'.*

Using a 2-D mid-point rule, the right hand side term in (3.12) is approximated by,

$$\int_V f(\mathbf{x})\, dV \approx f(P_0)(s_1+s_3)(s_2+s_4)\frac{h^2}{4} = F \tag{3.21}$$

For the special case of an equidistant mesh the above results reduce to,

$$s_K = 1 \quad \rightarrow \quad \alpha_K = \begin{cases} c(M_K) & ,K = 1,\ldots,4 \\ \sum_{J=1}^{4} c(M_J) & ,K = 0 \end{cases} \quad , \; F = h^2 f(P_0) \tag{3.22}$$

Combination of (3.1),(3.20) and (3.21) results in the following difference equation for the nodalpoint P_0,

$$\alpha_0 u(P_0) - \sum_{K=1}^{4} \alpha_K u(P_K) = F \tag{3.23}$$

By evaluating the finite difference formula (3.23) for every nodal point we obtain a system of linear algebraic equations that can be solved numerically. Each nodal point corresponds to a single equation in this set or to a single row of the corresponding coefficient matrix.

problem 3.4. *Verify that the difference formula (3.23) corresponds to the five-point formula derived in the previous section in the special case of an equidistant mesh and a uniform coefficient* $(c(\mathbf{x}) = c)$*,*

$$4u(P_0) - \sum_{K=1}^{4} u(P_K) = \frac{h^2 f(P_0)}{c} \tag{3.24}$$

problem 3.5. *Extend the derivation of the finite difference formula (3.23) derived for the steady state heat conduction problem (3.1) for the time dependent problem described by,*

$$\rho c_p \frac{\partial T}{\partial t} = \nabla \cdot k \nabla T + H \tag{3.25}$$

Hint: Consider a semi-discretization, leaving the continuous time variable in place to derive a system of first order ordinary differential equations similar to the 1-D case (2.14).

[2] The minus sign for the contributions I_3, I_4 accounts for the direction of the outward pointing normal vector on the corresponding boundary segments. $\nabla u \cdot \mathbf{n} = -\partial_x u$ on the left hand vertical boundary segment and $\nabla u \cdot \mathbf{n} = -\partial_y u$ on the bottom boundary.

problem 3.6. *Derive an expression for the right hand side vector element F for the case where the righthand side function represents a point source $f(\mathbf{x}) = a\delta(\mathbf{x} - \mathbf{x}_s)$, where a is a constant.*

Figure 3.2 shows the structure diagram of an algorithm for filling the coefficient matrix that follows from evaluation of (3.24) in all the internal points of the mesh for the special case of an equidistant mesh and uniform coeffient $c(\mathbf{x}) = 1$. A 2-D rectangular domain V is used and a grid consisting of $n_{col} + 2$ columns and $n_{row} + 2$ rows of nodal points. Furthermore the algorithm assumes that essential boundary conditions are given for all boundary points ($\partial V = \Gamma_g,\ \Gamma_h = \emptyset$). In that case the discretized problem has $n_{row} \times n_{col} = N$ degrees of freedom - one for each internal nodal point. The degrees of freedom are numbered column-wise in the grid of nodal points. This way an N-vector $\mathbf{U}$ is defined of unknown nodal point values.

$$\begin{aligned}\mathbf{U} = \quad & (u(x_1, y_1), u(x_1, y_2), \ldots, u(x_1, y_{n_{row}}), \\ & \cdots, \\ & u(x_{n_{col}}, y_1), u(x_{n_{col}}, y_2), \ldots, u(x_{n_{col}}, y_{n_{row}}),)^T\end{aligned} \tag{3.26}$$

In this case with essential boundary conditions on the complete boundary, evaluating the difference equation (3.24) in every nodal point results in a complete system of equations.

problem 3.7. *Verify that the matrix of the above finite difference equations is symmetric and that the matrix rows outside the main diagonal and four other diagonals contain zero values. Show for the bandwitdh in (3.10): $w = n_{row}$.*

problem 3.8. *Extend the algorithm of Fig. 3.2 with the computation of a right hand side vector for the system of finite difference equations.*

problem 3.9. *Verify how the symmetry of the matrix can be applied to optimize the algorithm of Fig. 3.2.*

problem 3.10. *How could the algorithm of Fig. 3.2 be modified for the case of variable coefficient $c(\mathbf{x})$?*
Hint: consider the equidistant case and apply (3.22).

problem 3.11. *How could the algorithm of Fig. 3.2 be extended for the case of variable coefficient $c(\mathbf{x})$ and variable grid spacing?*

3.4 Implementation of boundary conditions

We distinguish between essential boundary conditions with prescribed values of the solution $u(\mathbf{x})$ and natural boundary conditions where the normal component of the gradient $c(\mathbf{x})\nabla u \cdot \mathbf{n}$ is prescribed in boundary points $\mathbf{x}$.

3.4.1 Essential boundary conditions

An implementation of essential boundary conditions follows directly from the difference equation,

$$\alpha_0 u(P_0) - \sum_{K=1}^{4} \alpha_K u(P_K) = F \tag{3.27}$$

Terms in (3.27) with prescribed values of u in boundary points $\mathbf{x}_K \in \Gamma_g$ can be moved to the right hand side of the equation. Essential boundary conditions thus contribute to

```
ncol - number columns internal nodal points
nrow - number rows internal nodal points
idof - sequence number degree of freedom for nodal point (irow,jcol)

for a constant coefficient and equidistant mesh:
e0 =  4. - main diagonaal element difference formula
e1 = -1. - elements of second diagonal
```

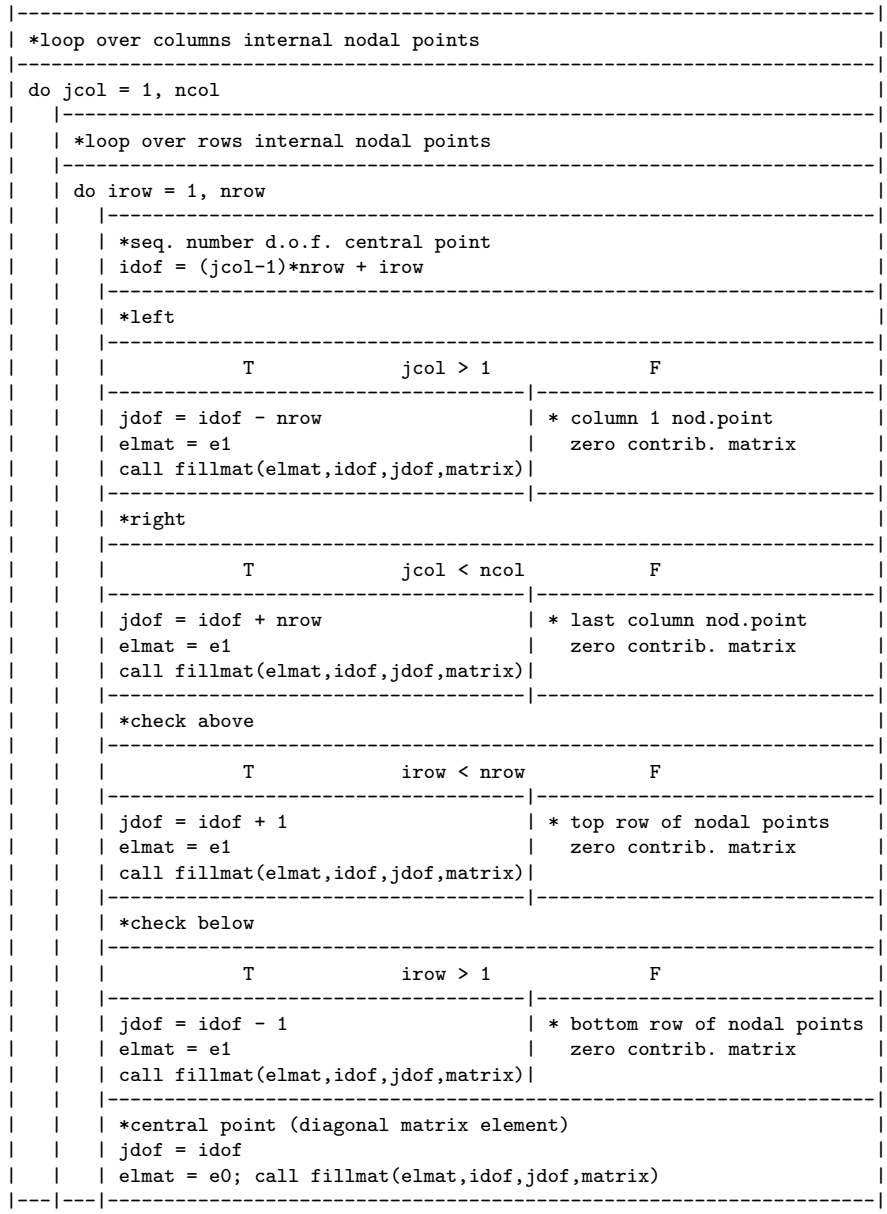

Figure 3.2: *Structure diagram of an algorithm to fill the coefficient matrix of the finite difference equations. The subroutine* `fillmat` *is used for storing the matrix elements in an array* `matrix`*. This way the sparse structure of the matrix can be exploited in an easy way.*

the right hand side vector of the system of equations. This can be made more explicit by partitioning the vector of nodal point values $\mathbf{U} = (\mathbf{U}_f, \mathbf{U}_p)^T$, where $\mathbf{U}_p$ is the vector of prescribed (boundary) nodal point values, and $\mathbf{U}_f$ the vector of remaining (free) unknown nodal point values, the degrees of freedom. The matrix $\mathbf{S}$ and right hand side vector $\mathbf{F}$ partition correspondingly,

$$\begin{pmatrix} \mathbf{S}_{ff} & \mathbf{S}_{fp} \end{pmatrix} (\mathbf{U}_f, \mathbf{U}_p)^T = \mathbf{F}_f \tag{3.28}$$

By writing the multiplications of the partitioned matrix blocks in (3.28) explicitly we see that the vector part of unknown nodal point values $\mathbf{U}_f$ can be solved from the following reduced system of equations,

$$\mathbf{S}_{ff}\mathbf{U}_f = \mathbf{F}_f - \mathbf{S}_{fp}\mathbf{U}_p = \mathbf{R}_f \tag{3.29}$$

Note that $\mathbf{F}_p$ does not occur in (3.29).

problem 3.12. *How could the algorithm in Fig. 3.2 be extended to account for the contribution of inhomogeneous essential boundary conditions in the right hand side vector?*

3.4.2 Natural boundary conditions

Implementation of natural boundary conditions is less straight forward. It is clear that the number of degrees of freedom of the problem is now greater than in the previous case since the nodal point values corresponding to points $\mathbf{x}_K \in \Gamma_h$ are also degrees of freedom. In order to get a complete set of equations, difference equations must be formulated that include these degrees of freedom for $\mathbf{x}_K \in \Gamma_h$. This is done by integrating the differential equation over finite volumes associated with the boundary points $\mathbf{x}_K \in \Gamma_h$, as illustrated in Fig. 3.3. In the integration over the vertical boundaries of the cell, the integral I_3 over the segment $M_2M_4 \subset \Gamma_h$ can be expressed in the known boundary value $c(P_0)\partial_x u(P_0)$. This results in a contribution to the right hand side vector. Note that here the integration is over a reduced area compared to interior grid cells. The expressions for the resulting matrix coefficients differ from the ones for interior nodal points.

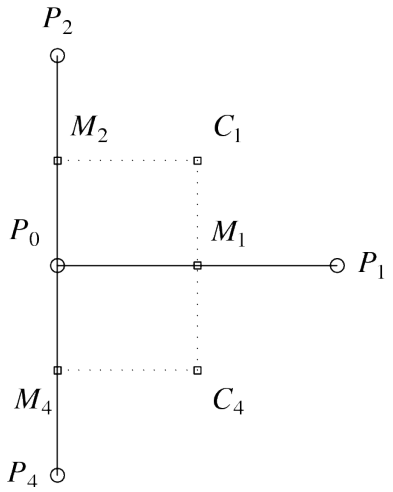

Figure 3.3: *Integration over a grid cell associated with a boundary point $\mathbf{x}_K \in \Gamma_h$. The interior of the computational domain is on the right hand side of the boundary segment P_2, P_0, P_4. The grid line $P_2P_0P_4$ is part of the bondary Γ_h with natural boundary conditions.*

problem 3.13. *Show that, in cases with natural boundary conditions on the whole boundary $\partial V = \Gamma_h$, ($\Gamma_g = \emptyset$), the boundary condition must satisfy the following compatibility condition,*

$$-\int_{\partial V} h(\mathbf{x})\ dA = \int_V f(\mathbf{x})\ dV \tag{3.30}$$

Give a physical interpretation of this condition in the context of steady state heat conduction problems. Show also that the solution is underdetermined by an additive function, say $u_0(\mathbf{x})$, a solution of the homogeneous equation $\nabla \cdot c\nabla u_0 = 0$ that satisfies homogeneous natural boundary conditions $c\nabla u_0 \cdot \mathbf{n} = 0$. [3] *What does this imply for the matrix of the discrete equations?*

[3] For the special case of a spherically symmetric problem, the additive homogeneous solution is a constant. This follows from the following consideration,

$$\frac{1}{r^2}\frac{d}{dr}\left(r^2 c\frac{du_0}{dr}\right) = 0 \Rightarrow r^2 c\frac{du_0}{dr} = A,\ c\frac{du_0}{dr} = \frac{A}{r^2}$$

A finite heat flux in the origin $c\frac{du_0}{dr} < \infty$ requires $A = 0$, $\frac{du_0}{dr} = 0$, with a uniform solution u_0.

Chapter 4

The finite element method - an introduction

In an introduction of the finite element method, two main aspects can be distinguished. These are first the domain discretization and associated discretization of the solution function and second the discretization of the differential equation. In both aspects the finite element method differs from the finite difference/volume methods introduced in the previous chapters.

4.1 Discretization of the domain and solution field

In the finite difference method the unknown function $u(\mathbf{x})$ is discretized by defining a vector of discrete nodal point values,

$$\mathbf{U} = (u(\mathbf{x}_1), \ldots, u(\mathbf{x}_N))^T \tag{4.1}$$

on the grid of nodal points,

$$G = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\} \tag{4.2}$$

In finite difference methods the nodal point values are computed by solving the algebraic finite difference equations, where the unknown vector (4.1) consists of the nodal point values of the field $u(\mathbf{x})$. The value of the solution field outside the nodal points is not directly obtained by these methods.

The finite element method is also based on discretization of the domain V with a grid of nodal points G. The nodal points are connected in such a way that a sub-division of the domain in non-overlapping *elements* e_K is obtained,

$$V = \cup_K e_K, \quad e_K \cap e_J = \emptyset, \quad K \neq J \tag{4.3}$$

In the treatment of the finite difference method we used a rectangular mesh with gridlines parallel to the coordinate axes. Such a so called *structured mesh* can be described as the *product* of two 1-D meshes. This is characteristic for the finite difference method. In the finite element method there is a greater flexibility in the discretization of the domain. It is therefore simpler to apply local grid refinement with the finite element method, such that strong local variations in the solution can be resolved in an efficient way. Restrictions to the domain discretization are that the elements must not be *degenerate*, i.e. the volume of an element should not become too small with respect to the length or surface area of the element boundary.

In the finite element method the discretization of the solution is done by means of an approximating expansion in interpolating basis functions, [1]

$$u(\mathbf{x}) \approx u^h(\mathbf{x}) = \sum_{J=1}^{N} \alpha_J N_J(\mathbf{x}) \tag{4.4}$$

The basis functions N_J are defined in a piecewise way as interpolating functions on the elements. This means that the functions N_J are chosen such that on the individual elements (4.4) takes the form of an interpolation in terms of the element nodal point values of the solution. Different types of interpolation are possible (Lagrange, Hermite, Spline). We shall only consider Lagrange interpolation here. Piecewise Lagrange interpolation can best be illustrated for the 1-D case. Suppose we have 1-D elements with n_e nodal points per element, $x_a, a = 1, \ldots, n_e$, $n_e \geq 2$. For this type of element, interpolating Lagrange polynomials of degree $n_e - 1$ can be defined in the following way,

$$l_a^{n_e-1}(x) = \frac{\prod\limits_{\substack{b=1 \\ b \neq a}}^{n_e} (x - x_b)}{\prod\limits_{\substack{b=1 \\ b \neq a}}^{n_e} (x_a - x_b)} \tag{4.5}$$

problem 4.1. *Writing the products in the (de)nominator of (4.5) explicitly we get,*

$$l_a^{n_e-1}(x) = \frac{(x - x_1)(x - x_2)\ldots(x - x_{n_e})}{(x_a - x_1)(x_a - x_2)\ldots(x_a - x_{n_e})} \tag{4.6}$$

where there are no factors $(x - x_a)$ in the nominator or $(x_a - x_a)$ in the denominator in (4.6).

Verify that $l_a^{n_e-1}(x)$ has the following properties:

1. $l_a^{n_e-1}(x)$ *contains* $n_e - 1$ *factors* $(x - x_b)$, $x_b \neq x_a$, (4.7)
   a polynomial of degree $n_e - 1$
2. $l_a^{n_e-1}(x_a) = 1$ (4.8)
3. $l_a^{n_e-1}(x_c) = \delta_{ac}$, $x_c \in \{x_1, x_2, \ldots, x_{n_e}\}$ (4.9)
   where δ_{ac} is the Kronecker delta.

problem 4.2. *Derive expressions for polynomials of degree one and two that are defined on elements with two and three nodal points per element respectively using (4.5) (see also Fig. 4.1 and 4.2).*

Examples of basis functions, composed of piecewise Lagrange polynomials (4.5) of degree one and two are given in Fig. 4.1 and Fig. 4.2 respectively.

[1] To distinguish between the analytical solution and its approximation in terms of the expansion (4.4) a superscript h is applied to the latter.

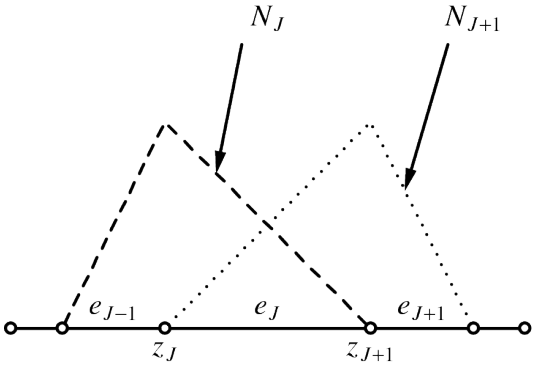

Figure 4.1: *piecewise linear basis functions N_J (dashed) and N_{J+1} (dotted) on a 1-D grid.*

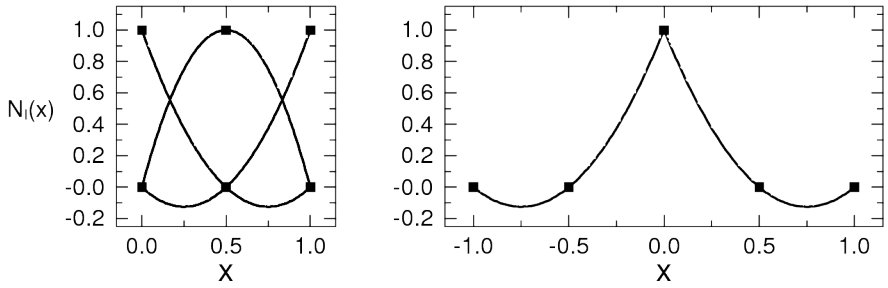

Figure 4.2: *Lagrange interpolating polynomials on a 1-D three-point element (left). A finite element basis function on two neighboring 1-D elements (right).*

The interpolating basis functions $N_J(\mathbf{x})$ in Fig. 4.1 are defined such that they are different from zero only in the neighborhood of the nodal point $\mathbf{x}_J$, with $N_J(\mathbf{x}_J) = 1$. The basis functions have a local *support* S, defined as the set of elements that contain $\mathbf{x}_J$,

$$S(N_J) = \{\cup_K e_K \mid \mathbf{x}_J \in e_K\} \tag{4.10}$$

The N_J are defined piecewise per element. For the 1-D case and using Lagrange polynomials we have,

$$N_J(x) = \begin{cases} l_a^{n_e-1}(x), & x \in S(N_J) \\ 0, & x \ni S(N_J) \end{cases} \tag{4.11}$$

Here a is the local nodal point number ($a = 1, 2, \ldots, n_e$), corresponding to x_J, different from its global nodal point numbering J. $J = 1, 2, \ldots, N$, where N is the total number of nodal points in the domain. For the 1-D case and piecewise basis functions in Fig. 4.1, 4.2, $S(N_J)$ consists of a maximum of two elements.

The interpolating character of the Lagrange functions becomes clear on substitution of a nodal point $\mathbf{x} = \mathbf{x}_I$ in (4.4), and applying (4.9),

$$u(x_I) \equiv U_I = \sum_{J=1}^{N} \alpha_J N_J(x_I) = \sum_{J=1}^{N} \alpha_J \delta_{IJ} = \alpha_I \tag{4.12}$$

The coefficients α_J in the basis function expansion correspond to the nodal point values of the interpolated function u. Outside the nodal points the finite element approximation

(4.4) of the solution u is defined as an interpolation in terms of the nodal point values $u(x_J) = U_J$,

$$u(x) \approx u^h(x) = \sum_{J=1}^{N} U_J N_J(x) \tag{4.13}$$

This is illustrated for the one-dimensional linear element with two nodal points per element ($n_e = 2$),

$$l_1^1(x) = \frac{x - x_2}{x_1 - x_2} = \frac{1}{h}(x_2 - x) \tag{4.14}$$

$$l_2^1(x) = \frac{x - x_1}{x_2 - x_1} = \frac{1}{h}(x - x_1) \tag{4.15}$$

Both parts of this piecewise linear basis function are illustrated in Fig. 4.1.

problem 4.3. *Show that the so called trapezoidal rule for approximation of integrals* [2] *follows directly from an expansion as in (4.13), using equidistant evaluation points x_J. Also derive a corresponding trapezoidal rule for the general case with variable grid spacing.*

4.2 Discretization of the differential equation

The finite element method is introduced here as a special case of the method of Galerkin for solving differential equations. Galerkins method is defined in the following way: for a given differential equation $Lu = f$, the residual function $R = Lu - f$ is multiplied by a weighting function w_I and integrated over the domain,

$$\int_V w_I \left(Lu - f\right) \, dV = 0, \quad I = 1, 2, \ldots \tag{4.17}$$

The linearly independent weighting functions $w_I(\mathbf{x})$ span a linear function space S. When the integral expression in (4.17) is interpreted as a special case of the general functional innerproduct of the functions $p,\ q \in S$,

$$(p \cdot q) = \int_V p(\mathbf{x}) q(\mathbf{x}) \, dV \tag{4.18}$$

then the equation (4.17), $(w_I \cdot R) = 0$, specifies the condition for the solution u that the residue of the differential equation is orthogonal to the vector space S spanned by the weighting functions w_I, in the sense of the inner product (4.18).

We shall further use the special case where the weighting functions and the basis functions N_J used in the expansion [3] of the solution u are identical. This is known in the

[2]

$$I = \int_a^b f(x)dx \approx I^h = \sum_{J=1}^{N} w_j f(x_J),\ w_j = \begin{cases} \Delta x/2, & J = 1|N \\ \Delta x, & 1 < J < N \end{cases} \tag{4.16}$$

`http://en.wikipedia.org/wiki/Trapezoidal_rule`

[3] For the general m-dimensional case a generalization of (4.13) is used,

$$u(\mathbf{x}) \approx u^h(\mathbf{x}) = \sum_{J=1}^{N} U_J N_J(\mathbf{x}) \tag{4.19}$$

Where $u(\mathbf{x})$ and $N_J(\mathbf{x})$ are functions of the m spatial coordinates $\mathbf{x} = (x_1, \ldots, x_m)$ of the m-dimensional domain.

literature as the Bubnov-Galerkin method. It can be shown that an approximate solution u^h of (4.17) is obtained with,

$$u(\mathbf{x}) \approx u^h(\mathbf{x}) = \sum_J U_J N_J(\mathbf{x}) \tag{4.20}$$

where $U_J = u^h(\mathbf{x}_J)$, and that the approximation in (4.20) is optimal in the (least square) sense of the euclidean norm induced by the innerproduct in (4.18), i.e. $||Lu^h - f||$ is minimal in the norm,

$$||p||^2 = (p \cdot p) = \int_V p^2(\mathbf{x})\ dV \tag{4.21}$$

4.2.1 An example of the Galerkin method with non-local basis functions

The Galerkin method can be illustrated with a simple example that can be solved analytically by a Fourier series solution. In this example the harmonic (sine) functions of the Fourier expansion are used as basis functions of the linear space of functions on the interval $[0, L]$. In contrast to the basis functions in the finite element method introduced in the previous sections the sine functions applied here do not have a local support.

The example is the 1-D two-point boundary value problem of the steady state equation for heat diffusion in a static medium. In the next chapter a finite element solution for the same problem will be derived. The problem is defined on a domain $[0, L]$ with homogeneous essential boundary conditions,

$$-\frac{d}{dx}k\frac{du}{dx} = f, \quad x \in [0, L], \quad u(0) = u(L) = 0 \tag{4.22}$$

By the choice of the sign in the operator $Lu = -\frac{d}{dx}k\frac{du}{dx}$ (4.22) corresponds to a steady state diffusion problem with a positive (heat)source density function f and temperature u.

The Galerkin principle (4.17) for this case can be written as,

$$\int_0^L N_I\,(Lu - f)\,dx = \int_0^L N_I \left(-\frac{d}{dx}k\frac{du}{dx} - f\right) dx = 0, \quad I = 1, 2, \ldots \tag{4.23}$$

$$\int_0^L \left\{-\frac{d}{dx}\left(N_I k\frac{du}{dx}\right) + \frac{dN_I}{dx}k\frac{du}{dx} - N_I f\right\}\ dx = 0, \quad I = 1, 2, \ldots \tag{4.24}$$

$$\left[-N_I k\frac{du}{dx}\right]_0^L + \int_0^L \frac{dN_I}{dx}k\frac{du}{dx}\ dx - \int_0^L N_I f\ dx = 0, \quad I = 1, 2, \ldots \tag{4.25}$$

The first term in (4.25) is determined by the choice of boundary conditions and/or the basis functions N_I. For the second term we find by substitution of an expansion in the general type basis functions,

$$\begin{aligned}\int_0^L \frac{dN_I}{dx}k\frac{du}{dx}dx &= \int_0^L \frac{dN_I}{dx}k\sum_J \alpha_J\frac{dN_J}{dx}\ dx = \sum_J \int_0^L k\frac{dN_I}{dx}\frac{dN_J}{dx}\ dx\ \alpha_J \\ &= \sum_J S_{IJ}\alpha_J \end{aligned} \tag{4.26}$$

At this point we have converted the PDE (4.22) in a system of linear algebraic equations for the unknown expansion coefficients α_J,

$$\sum_J S_{IJ}\alpha_J = R_I\ ,\ I = 1, 2, \ldots \tag{4.27}$$

with righthand side vector,

$$R_I = \int_0^L N_I f dx + \left[N_I k \frac{du}{dx} \right]_0^L \tag{4.28}$$

Next we choose the basis functions such that they satisfy the homogeneous essential boundary conditions specified in (4.22),

$$N_I(x) = \sin\left(\frac{I\pi x}{L}\right) \rightarrow \frac{dN_I}{dx} = \frac{I\pi}{L}\cos\left(\frac{I\pi x}{L}\right) \tag{4.29}$$

Note that these functions do not have a local support like the piecewise defined finite element basis functions. If we also assume k to be uniform we get for the matrix coefficients,

$$S_{IJ} = \frac{kIJ\pi^2}{L^2}\int_0^L \cos\left(\frac{I\pi x}{L}\right)\cos\left(\frac{J\pi x}{L}\right) dx = \frac{kIJ\pi^2}{2L}\delta_{IJ} \tag{4.30}$$

Because of the *orthogonality* of the cosine functions on the interval $[0, L]$, the matrix $\mathbf{S}$ is found to be a diagonal matrix. We further assume that the righthand side function is uniform, $f(x) = f = constant$. We then obtain the following expression for the righthand side vector elements of the Galerkin equation,

$$F_I = \int_0^L N_I f \; dx = f\int_0^L \sin\left(\frac{I\pi x}{L}\right) \; dx = \begin{cases} \frac{2fL}{I\pi}, & I \; odd \\ 0, & I \; even \end{cases} \tag{4.31}$$

The (Fourier) coefficients of the expansion of $u(x)$ can be obtained by solving the diagonal system of equations,

$$\sum_J S_{IJ}\alpha_J = S_{II}\alpha_I = F_I \;\rightarrow\; \alpha_I = \frac{F_I}{S_{II}} \tag{4.32}$$

$$\alpha_I = \begin{cases} \frac{4fL^2}{I^3\pi^3 k}, & I \; odd \\ 0, & I \; even \end{cases} \tag{4.33}$$

The same solution can be obtained by substitution of a Fourier series expansion directly in the differential equation (4.22) and computation of the corresponding F-series expansion of the uniform right hand side function.

problem 4.4. *Verify the outcome of the Galerkin method by Fourier series expansion of the analytical solution of the above problem*

$$u(x) = \frac{f}{2k}x\,(L - x) \tag{4.34}$$

problem 4.5. *What happened to the boundary contribution term in the Galerkin equation (4.25)?*

4.2.2 Generalisation of the Galerkin method to potential problems in more dimensions

We apply the Galerkin method here to the differential equation introduced in the previous chapter on finite difference methods,

$$-\nabla \cdot c\nabla u = f \tag{4.35}$$

We shall develop the Galerkin method here using a general formulation that is applicable in one, two or three dimensions. We further keep the Lagrangian basis functions N_I general such that the results will be applicable for the general case of the interpolating functions

of the finite elements introduced before. This results in a general formulation of the finite element method for the multi-dimensional steady state heat conduction problem and related Poisson equation for general potential problems.

Applying the Galerkin principle to the differential equation (4.35) we obtain,

$$\int_V N_I \left(-\nabla \cdot c\nabla u - f\right)\ dV = 0,\ I = 1, 2, \ldots, N \tag{4.36}$$

With partial integration of the leading term and applying the divergence theoreme we obtain,

$$\int_{\partial V} -N_I c\nabla u \cdot \mathbf{n}\ dA + \int_V \nabla N_I \cdot c\nabla u\ dV - \int_V N_I f\ dV = 0,\ I = 1, 2, \ldots, N \tag{4.37}$$

We expand u in the second term in finite element basis functions N_J,

$$u(\mathbf{x}) \approx u^h(\mathbf{x}) = \sum_J U_J N_J(\mathbf{x}) \tag{4.38}$$

By substitution of (4.38) in (4.37) we obtain,

$$-\int_{\partial V} N_I c\nabla u \cdot \mathbf{n}\ dA + \sum_J \int_V \nabla N_I \cdot c\nabla N_J\ dV\ U_J = \int_V N_I f\ dV,\ \ I = 1, 2, \ldots, N \tag{4.39}$$

The first term in (4.39) is determined by the boundary conditions and the choice of basis functions (see below). Equation (4.39) represents a system of linear algebraic equations in the unknown coefficients U_J,

$$\mathbf{SU} = \mathbf{F} \tag{4.40}$$

Where the matrix $\mathbf{S}$ is defined as,

$$S_{IJ} = \int_V c\nabla N_I \cdot \nabla N_J\ dV \tag{4.41}$$

and the righthand side vector is,

$$F_I = \int_V N_I f\ dV + \int_{\partial V} N_I c\nabla u \cdot \mathbf{n}\ dA \tag{4.42}$$

The following properties of the above finite element equations can be verified,

1. The matrix $\mathbf{S}$ is symmetric ($S_{IJ} = S_{JI}$) and sparse, i.e. most of the matrix coeffients S_{IJ} are zero, due to the local support of the basis functions. [4]

2. (4.39) represents a complete system of equations in the unknown nodal point values $\mathbf{U} = (U_1, \ldots, U_N)$ in case of natural boundary conditions,

   $$c\nabla u \cdot \mathbf{n} = h(\mathbf{x}),\ \mathbf{x} \in \Gamma_h \subset \partial V \tag{4.43}$$

   where the following compatibility condition holds for the special case with natural b.c. on the whole boundary, i.e. $\Gamma_h = \Gamma$, $\Gamma_g = \emptyset$,

   $$-\int_{\partial V} h(\mathbf{x}) dA = \int_V f(\mathbf{x}) dV \tag{4.44}$$

5

[4] From the integral expression (4.41) it follows that overlapping supports $S_I \cap S_J \neq \emptyset$ are a necessary requirement for a non-zero matrix coefficient S_{IJ}.

[5] Verify that the case with $\Gamma_h = \Gamma$, $\Gamma_g = \emptyset$, does not have a *unique* solution.

3. In the case of essential boundary conditions the number of unknowns (degrees of freedom) is reduced to $N_f = N - N_p$ where N_p is the number of boundary points with a prescribed solution value. The number of Galerkin equations (4.39) is reduced together with the number of degrees of freedom. This is done by choosing only those weighting functions N_I that are zero valued in the subset of the boundary points with prescribed solution values. The boundary integral in (4.39) then vanishes, in a similar way as in section 4.2.1 and we again obtain a (reduced) complete system of equations.

   Inhomogeneous essential boundary conditions produce a contribution to the righthand side vector in a similar way as in the finite difference method. As shown in section 3.4.1 this can be formulated in an explicit way by partitioning the vector of unknown nodal point values $\mathbf{U} = (\mathbf{U}_f, \mathbf{U}_p)$ and the matrix $\mathbf{S}$.

In case of mixed boundary conditions, both essential and natural boundary conditions are prescribed on different parts of the boundary, $\partial V = \Gamma_g \cup \Gamma_h$ and the above, except for the compatibility requirement (4.44), applies to Γ_g and Γ_h separately.

Chapter 5

A finite element solution for the 1-D heat equation

In the previous chapter the finite element method was introduced as a solution method for partial differential equations in one, two and three dimensions. In this chapter we investigate the solution of the 1-D, time dependent, heat diffusion equation in more detail, as an illustration of the general multi-dimensional case.

The equation for time dependent heat conduction problems in a 1-D medium is,

$$\rho c_p \frac{\partial T}{\partial t} - \frac{\partial}{\partial z} k \frac{\partial T}{\partial z} = \rho H(z,t) = f(z,t), \;\; z \in [0, z_{max}], \;\; t \in [0, t_{max}] \tag{5.1}$$

Where T is the temperature, k is the thermal conductivity ρ the mass density, c_P the specific heat and ρH and H are the internal heatproduction rate, respectively per unit volume and per unit mass.

5.1 Discretization of the equation

We apply a *semi-discretization* to the partial differential equation (5.1). This implies that we discretize the spatial domain $[0, z_{max}]$, and the temperature field is expanded in the piecewise linear Lagrangian basis functions $N_J(z)$ introduced in the previous chapter. The expansion coefficients, i.e. the nodal point values of the temperature, are treated as continuous functions of the time variable t,

$$T(z,t) \approx T^h(z,t) = \sum_J T_J^h(t) N_J(z) \tag{5.2}$$

With the use of the Galerkin principle introduced in the previous chapter and substitution of the basis function expansion, the differential equation (5.1) is transformed into a set of ordinary differential equations (ODE). The unknowns or degrees of freedom are the time dependent nodal point values $T_J^h(t)$. Numerical integration methods for systems of ODE's are treated in Chapter 8. In the special case of a steady state conduction problem this approach results in a system of linear algebraic equations. Construction of a solution with the Galerkin method, using testfunctions N_I (Bubnov-Galerkin) starts from,

$$\int_0^{z_{max}} \left\{ \rho c_p \frac{\partial T}{\partial t} - \frac{\partial}{\partial z} \left(k \frac{\partial T}{\partial z} \right) - f \right\} N_I \; dz = 0, \;\; I = 1, 2, \ldots, N, \;\; t \in [0, t_{max}] \tag{5.3}$$

Partial integration of the second term in (5.3) gives,

$$- \left[k N_I \frac{\partial T}{\partial z} \right]_0^{z_{max}} + \int_0^{z_{max}} \left(\rho c_p \frac{\partial T}{\partial t} N_I + k \frac{\partial T}{\partial z} \frac{\partial N_I}{\partial z} - f N_I \right) \; dz = 0 \tag{5.4}$$

Next we expand the temperature in the basis functions N_J,

$$T(z,t) \approx T^h(z,t) = \sum_J T^h(z_J,t) N_J(z) = \sum_J T_J^h(t) N_J(z) \tag{5.5}$$

Substitution in (5.4) and dropping the superscript in T_J^h and the explicit notation of the time dependence, we get,

$$\begin{aligned} - & \left[k N_I \frac{\partial T}{\partial z} \right]_0^{z_{max}} \\ + & \int_0^{z_{max}} \left\{ \sum_J \left(N_J N_I \rho c_p \frac{\partial T_J}{\partial t} + k \frac{\partial N_J}{\partial z} \frac{\partial N_I}{\partial z} T_J \right) - f N_I \right\} dz = 0 \ , \ I = 1,2,.. \end{aligned} \tag{5.6}$$

with $z_1 = 0, z_N = z_{max}$ and $N_I(z_J) = \delta_{IJ}$, we obtain the following system of ODE's,

$$\left(k \frac{\partial T}{\partial z} \right)_0 \delta_{I1} - \left(k \frac{\partial T}{\partial z} \right)_{z_{max}} \delta_{IN} \ + \ \sum_J M_{IJ} \frac{\partial T_J}{\partial t} \ + \ \sum_J S_{IJ} T_J = Q_I, \ I = 1,2,\ldots \tag{5.7}$$

or

$$\mathbf{M} \frac{d}{dt} \mathbf{T} + \mathbf{S} \mathbf{T} = \mathbf{R} \tag{5.8}$$

where the *heat capacity* or *mass matrix* $\mathbf{M}$, the *stiffness matrix* $\mathbf{S}$ and the *righthand side vector* $\mathbf{R}$ are defined as,

$$M_{IJ} = \int_0^{z_{max}} \rho c_p N_I N_J \ dz \tag{5.9}$$

$$S_{IJ} = \int_0^{z_{max}} k \frac{\partial N_I}{\partial z} \frac{\partial N_J}{\partial z} \ dz \tag{5.10}$$

$$Q_I = \int_0^{z_{max}} f N_I \ dz \tag{5.11}$$

$$R_I = Q_I - \left(k \frac{\partial T}{\partial z} \right)_0 \delta_{I1} + \left(k \frac{\partial T}{\partial z} \right)_{z_{max}} \delta_{IN} \tag{5.12}$$

problem 5.1. *Verify the correspondence of the system of ODE's (5.8) with the finite difference equations (2.14).*

5.1.1 A related Fourier series solution

The finite element formulation in 5.1 is a special case of more general applications of the Galerkin method. The Galerkin method is especially useful in case of orthogonal basis functions as was illustrated for potential problems in section 4.2.1. A related example in time dependent problems is found in the Fourier series solution for the problem of a cooling layer of uniform properties, ρc_p, k and of thickness L, of given initial temperature profile and zero boundary temperature $T(0,t) = T(L,t) = 0$ and without internal heating, $(f = 0)$.

$$T(z,t)) = \sum_{n=1}^{\infty} T_n(0) \exp\left(-\frac{t}{\tau_n} \right) \sin\left(\frac{n\pi z}{L} \right) \tag{5.13}$$

where the relaxation times are defined as, $\tau_n = L^2/(\kappa n^2 \pi^2)$, $\kappa = k/(\rho c_p)$ the thermal diffusivity coefficient and where the $T_n(0)$ are the Fourier (sine) coefficients of the given initial temperature profile $T(z,0)$.

problem 5.2. *Derive (5.13) by solving the time dependent Fourier coefficients from the system of ODE's (5.7), with boundary contribution written as,* $-[\kappa N_I \frac{\partial T}{\partial z}]_0^L = 0$*, for a proper choice of the basis functions* N_I*.*
Hint: Use the sine functions as basis functions, $N_J = \sin(J\pi z/L)$*, and the corresponding expression for the diagonal stiffness matrix (4.30). Derive for the diagonal mass matrix,* $M_{IJ} = L/2\ \delta_{IJ}$*.*

5.2 Structure of the coefficient matrices

From the definition (5.9) and (5.10) of the coefficient matrices of (5.6) we can immediately deduce the symmetry property,

$$M_{IJ} = M_{JI},\ S_{IJ} = S_{JI} \tag{5.14}$$

If we choose for the basis functions the piecewise linear (hat) functions introduced in Chapter 4 and displayed in Fig. 5.1, these matrices are also tri-diagonal i.e. there bandwidth is one.

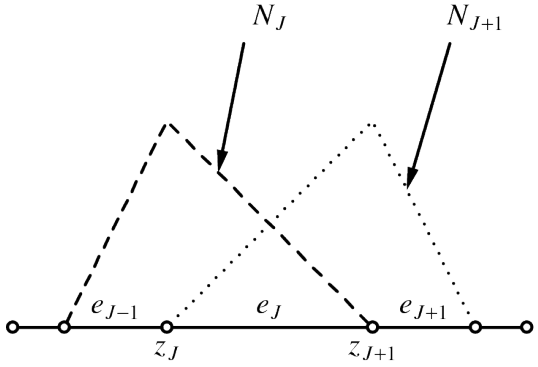

Figure 5.1: *Piecewise linear basis functions* N_J *(dashed) and* N_{J+1} *(dotted) on a 1-D grid. The support of basis function* N_J *is* $e_{J-1} \cup e_J$*. The support of* N_{J+1} *is* $e_J \cup e_{J+1}$*.*

For multi-dimensional problems (2-D,3-D) we find in a similar way that the finite element matrices are sparse, i.e. most of the matrix elements are zero. This follows directly from the expression for the matrix coefficients. For the heatcapacity matrix $\mathbf{M}$, also denoted as the mass matrix in the literature, we have for example,

$$M_{IJ} = \int_V \rho c_p N_I N_J\ dz \tag{5.15}$$

It follows that the matrix element is zero if $S(N_I) \cap S(N_J) = \emptyset$, where $S(N_I)$ is the support of the basis function N_I. This basis function is defined as a piecewise Lagrange interpolating polynomial on the elements that contain nodal $\mathbf{x}_I$, and zero elsewhere. This means that N_I differs from zero only in the direct neighborhood of nodal point $\mathbf{x}_I$ and most of all the possible productfunctions $N_I N_J$ are zero, resulting in a corresponding zero matrix element M_{IJ}. From (5.10) it follows that the stiffness matrix $\mathbf{S}$ has the same sparsity structure as the mass matrix $\mathbf{M}$. This situation resembles the finite difference methods introduced earlier, where only combinations of neighboring nodal points, connected by a *difference molecule* resulted in non-zero matrix coefficients.

5.3 Computation of the matrix elements

The integrals defining the matrix and right hand side vector can be split in a sum of contributions from the individual finite elements e_J, in this 1-D case $e_J = [z_J, z_{J+1}]$. In

this context the complete matrices in (5.9), (5.10) are known as global matrices and the contribution from a single element is known as an element matrix. In finite element computations the global matrices are computed in an *assembly procedure* where the coefficients of the element matrices are added to the corresponding coefficients of the global matrix in a loop over elements. The righthand side vector $\mathbf{R}$ is assembled in a similar way in a loop over elements. In section 5.4 the implementation of the assembly proces will be treated in more detail.

5.3.1 The mass matrix M

The heat capacity- or mass matrix appears in the time dependent term of the differential equations of the finite element solution (5.7). The matrix is defined as the summed contribution of the $N-1$ elements,

$$M_{IJ} = \int_0^{z_{max}} \rho c_p N_I N_J \, dz = \sum_{K=1}^{N-1} \int_{z_K}^{z_{K+1}} \rho c_p N_I N_J \, dz = \sum_{K=1}^{N-1} M_{IJ}^{(K)} \tag{5.16}$$

The mass matrix for element $e_K = [z_K, z_{K+1}]$ is defined as,

$$M_{IJ}^{(K)} = \int_{z_K}^{z_{K+1}} \rho c_p N_I N_J \, dz \tag{5.17}$$

In the following we assume the material properties ρc_p and k to be *piecewise uniform* per element, defined as $\rho(z)c_p(z) = C(z) = C_K, z \in e_K$ and $k(z) = k_K, z \in e_K$.

For the piecewise linear basis functions considered here we see from Fig.5.1 that only the following four coefficients of the element matrix are non-zero, $M_{KK}, M_{KK+1}, M_{K+1K}, M_{K+1K+1}$. Using the local numbering of the nodal points and basis functions on e_K and $h = z_2 - z_1$ we obtain,

$$M_{11} = \int_{z_1}^{z_2} C(z) N_1 N_1 \, dz = C_K \int_{z_1}^{z_2} \left(1 - \frac{z - z_1}{h}\right)^2 dz = C_K \int_0^1 (1-\zeta)^2 \, h \, d\zeta = \frac{h}{3} \tag{5.18}$$

$$\begin{aligned} M_{12} = \int_{z_1}^{z_2} C(z) N_1 N_2 \, dz &= C_K \int_{z_1}^{z_2} \left(1 - \frac{z - z_1}{h}\right)\left(\frac{z - z_1}{h}\right) dz \\ &= C_K \int_0^1 (1-\zeta)\,\zeta h \, d\zeta = \frac{h}{6} \end{aligned} \tag{5.19}$$

Since $M_{11} = M_{22}$ and $M_{12} = M_{21}$ we find for the element matrix, [1]

$$\mathbf{M}^{(K)} = \frac{C_K h_K}{6} \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix} \tag{5.23}$$

[1] A test for a software implementation of the element mass matrix can be devised from the following: An innerproduct of two real valued functions f and g on the interval $[0, z_{max}]$ is defined by,

$$(f \cdot g) = \int_0^{z_{max}} fg \, dz \tag{5.20}$$

Expansion in basis functions gives

$$(f^h \cdot g^h) = \int_0^{z_{max}} \left\{ \sum_I F_I N_I(z) \sum_J G_J N_J(z) \right\} dz = \sum_I \sum_J M_{IJ} F_I G_J = (\mathbf{MG} \cdot \mathbf{F}) \tag{5.21}$$

It follows that the innerproduct (5.20) is exactly represented by (5.21) for piecewise linear functions (including uniform functions), by using expression (5.23).

From (5.21) it follows that the mass matrix is positif definite,

$$(\mathbf{MX} \cdot \mathbf{X}) > 0, \ \mathbf{X} \neq \mathbf{0} \tag{5.22}$$

problem 5.3. *Derive for row number I of the global mass matrix,*

$$M_{IJ} = \begin{cases} \frac{C_{I-1}h_{I-1}}{6} & J = I-1 \\ \frac{C_{I-1}h_{I-1}+C_I h_I}{3} & J = I \\ \frac{C_I h_I}{6} & J = I+1 \\ 0 & |J-I| > 1 \end{cases} \tag{5.24}$$

The lumped mass matrix
In applications the sparse mass matrix is often replaced by an approximating diagonal matrix the so called *lumped* mass matrix, $\mathbf{M}^*$. For the general case in more dimensions we have,

$$M_{IJ} = \int_V C N_I N_J \, dV = \int_V C(\mathbf{x})\Phi(\mathbf{x}) \, dV \tag{5.25}$$

When the basis functions are piecewise Lagrange polynomials of degree p, $\Phi(\mathbf{x})$ is a piecewise polynomial of degree $2p$. This polynomial can be approximated in the usual way by interpolation of order p,

$$\begin{aligned} M_{IJ} &= \int_V C\Phi \, dV \approx \\ M^*_{IJ} &= = \int_V C(\mathbf{x}) \sum_K \Phi_K N_K(\mathbf{x}) \, dV = \sum_K N_I(\mathbf{x}_K) N_J(\mathbf{x}_K) \int_V C(\mathbf{x}) N_K(\mathbf{x}) \, dV \\ &= \sum_K \delta_{IK}\delta_{JK} \int_V C(\mathbf{x}) N_K(\mathbf{x}) \, dV = \delta_{IJ} \int_V C(\mathbf{x}) N_I(\mathbf{x}) \, dV \end{aligned} \tag{5.26}$$

The diagonal element M^*_{II} can be interpreted as a weighted average of the heat capapcity C, evaluated on the support S_I of the basis function N_I.

problem 5.4. *Derive the following expression for the lumped version of the element mass matrix from the mass matrix in (5.23).*
Solution:

$$\mathbf{M}^{*(K)} = \frac{C_K h_K}{2} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \tag{5.27}$$

Verify that the sum of the matrix elements in a row (row sum) is conserved in this matrix lumping procedure.
Hint: make use of the property of the basis functions,

$$\sum_L N_L(\mathbf{x}) = 1 \tag{5.28}$$

Derive for the global lumped mass matrix,

$$M^*_{IJ} = \left(\frac{C_{I-1}h_{I-1} + C_I h_I}{2} \right) \delta_{IJ} \tag{5.29}$$

5.3.2 The stiffness matrix S

The stifness matrix which appears in the diffusion term of equation (5.7) is defined as,

$$S_{IL} = \int_0^{z_{max}} k(z) \frac{\partial N_I}{\partial z} \frac{\partial N_L}{\partial z} \, dz \tag{5.30}$$

The derivatives have the same support as the basis functions,

$$\frac{\partial N_I}{\partial z} = \begin{cases} \frac{1}{h_{I-1}}, & z \in e_{I-1} \\ -\frac{1}{h_I}, & z \in e_I \\ 0, & z \ni e_I \cup e_{I-1} \end{cases} \tag{5.31}$$

We further assume here that the coefficient of thermal conductivity is piecewise constant. The element matrix becomes,

$$S_{IL}^{(K)} = k_K \int_{z_K}^{z_{K+1}} \frac{\partial N_I}{\partial z} \frac{\partial N_L}{\partial z}\, dz \tag{5.32}$$

and substituting (5.31) we get for the diagonal terms,

$$S_{11} = k_K \int_{z_K}^{z_{K+1}} \left(\frac{\partial N_1}{\partial z} \right)^2 dz = \frac{k_K}{h_K}, \quad S_{22} = S_{11} \tag{5.33}$$

For the off-diagonal terms we get,

$$S_{12} = k_K \int_{z_K}^{z_{K+1}} \frac{\partial N_1}{\partial z} \frac{\partial N_2}{\partial z}\, dz = -\frac{k_K}{h_K}, \quad S_{21} = S_{12} \tag{5.34}$$

$$\mathbf{S}^{(K)} = \frac{k_K}{h_K} \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix} \tag{5.35}$$

5.3.3 The righthand side vector R

Here we consider the contribution to the righthand side vector from the righthand side function f of the partial differential equation (5.1) defined as,

$$Q_I = \int_0^{z_{max}} f N_I\, dz \tag{5.36}$$

Corresponding contributions from boundary conditions are defined in (5.7). For a given function f the integral in (5.36) can be evaluated numerically. In a alternative procedure $f(z)$ is expanded in the same basis functions as the solution $T(z)$,

$$\begin{aligned} Q_I &\approx \int_0^{z_{max}} \sum_J F_J N_J(z) N_I(z)\, dz = \sum_J \int_0^{z_{max}} N_J(z) N_I(z)\, dz F_J \\ &= \sum_J A_{IJ} F_J, \quad F_J = f(z_J) \end{aligned} \tag{5.37}$$

where $\mathbf{A}$ is a mass matrix similar to the heat capacity matrix $\mathbf{M}$. This way the righthand side vector is computed by means of a matrix-vector multiplication of the mass matrix and the nodal point vector of the righthanside function of the PDE (5.1). In software implementations the righthandside vector is assembled element-wise by summing element vectors $\mathbf{Q}^K$ in a program loop over elements e_K,

$$Q_I = \sum_K Q_I^K, \quad Q_I^K = \sum_J A_{IJ}^K F_J^K, \quad \mathbf{Q} = \mathbf{AF} \tag{5.38}$$

$$\begin{pmatrix} Q_1^K \\ Q_2^K \end{pmatrix} = \frac{h_K}{6} \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix} \begin{pmatrix} F_1^K \\ F_2^K \end{pmatrix} \tag{5.39}$$

problem 5.5. *How can the vector* $\mathbf{Q}$ *be defined for righthand side function* $f(z) = c\delta(z - z_s)$*, corresponding to a pointsource concentrated in the sourcepoint* z_s*. Where* c *is a constant and* δ *is the Dirac delta function. Show that the number of non-zero vector elements of the right-hand side vector for this case is two.*

5.4 Implementation of the assembly proces

The matrix and right hand side vectors of the discretized equations are computed by summing element contributions. This procedure is known as matrix and vector assembly respectively. As an example of the general procedure we describe here an implementation for a 1-D problem that can be generalized for multi-dimensional problems. A so called location matrix for the discretized domain is used in the implementation. This is an $M \times 2$ matrix, M the number of elements in the 1-D grid. For each element, the corresponding row of the location matrix contains the global sequence numbers of the two degrees of freedom corresponding to the element. In the assembly proces these sequence numbers are used as pointers to the global matrix coefficients where the coefficients of the element matrix are added. Fig. 5.2 shows a structure diagram of an algorithm for the computation of the location matrix in a program array `kelem`.

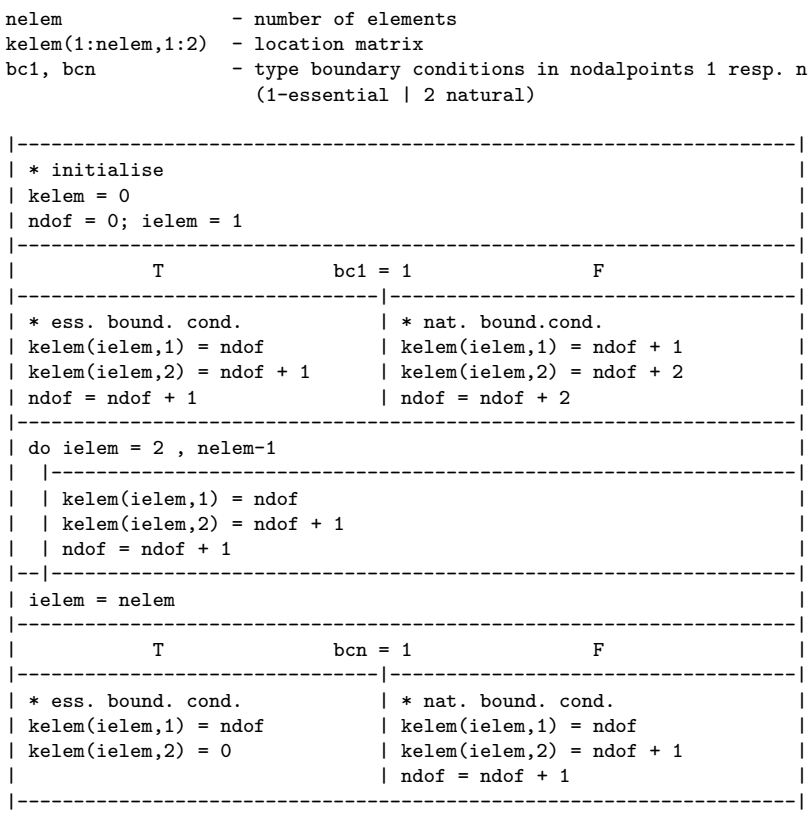

Figure 5.2: *Structure diagram of an algorithm for filling the location matrix in an array* `kelem`*. Note the different treatment of essential and natural boundary conditions.*

The location matrix is used in the assembly process. An algorithm for the assembly of the stiffness matrix and righthand side vector is described in the structure diagram of Fig. 5.3. In this scheme the 2×2 element matrices are computed in an element routine `elem` which contains an implementation of the expression (5.35). The actual summation of the element matrix coefficient to the global matrix coefficients is performed in a procedure `addmat`. In this procedure the special (band) structure of the sparse global matrix can be exploited to obtain effcient memory storage of the matrix array. The element righthand side vector is computed in a routine `elrhs` which contains an implementation of (5.39).

problem 5.6. *Verify that the assembly proces for the element matrices (5.35) results in the same matrix as obtained with the finite volume method in Chapter 2. Hint: compare a single matrix row for both cases.*

```
nelem    - number of elements
coord    - nodalpoint coordinates
kelem    - location matrix
glomat   - global matrix
glovec   - global right hand side vector
elmat    - element matrix
elvec    - element vector
```

```
|-------------------------------------------------------------------|
| do ielem = 1, nelem                                               |
|  |----------------------------------------------------------------|
|  | idof1 = kelem(ielem,1)                                         |
|  | idof2 = kelem(ielem,2)                                         |
|  | * element matrix                                               |
|  | call elem(ielem,coord,elmat)                                   |
|  | * element vector                                               |
|  | call elrhs(ielem,coord,elvec)                                  |
|  |----------------------------------------------------------------|
|  |                  T                 idof1 > 0      F           |
|  |----------------------------------------------------------------|
|  | * diagonal element glob. matr.           | * essent. bound.cond.|
|  | call addmat(idof1,idof1,elmat(1,1),glomat)|                     |
|  | * r.h.s. vector                          |                      |
|  | glovec(idof1)=glovec(idof1)+elvec(1)     |                      |
|  |----------------------------------------------------------------|
|  |                  T                 idof2 > 0      F           |
|  |----------------------------------------------------------------|
|  | * diagonaal element glob. matr.          | * essent. bound.cond.|
|  | call addmat(idof2,idof2,elmat(2,2),glomat)|                     |
|  | * r.h.s. vector                          |                      |
|  | glovec(idof2)=glovec(idof2)+elvec(2)     |                      |
|  |----------------------------------------------------------------|
|  | * element outside main diagonal                                |
|  |----------------------------------------------------------------|
|  |                    T            idof1*idof2 != 0   F           |
|  |----------------------------------------------------------------|
|  | * outside main diagaonal                   | * compute r.h.s.  |
|  | * fill uppertriangle (symmetry)            | contrib.          |
|  | idofmn = min(idof1,idof2)                  | bound.cond.       |
|  | idofmx = max(idof1,idof2)                  |                   |
|  | call addmat(idofmn,idofmx,elmat(1,2),glomat) |                 |
|-------------------------------------------------------------------|
```

Figure 5.3: *Structure diagram of an algorithm for matrix/vector assembly using the location matrix in an array* `kelem`.

5.5 Solving equations with a tri-diagonal matrix

For 1-D problems solution procedures of partial differential equations based on discretization methods like the finite difference method or the finite elment method often require the numerical solution of linear algebraic equations with a tri-diagonal matrix. A simple recursion method known as the *Thomas algorithm* can be used to compute such solutions. [2] To derive the algorithm the matrix is written as,

$$\begin{pmatrix} b_1 & c_1 & 0 & \dots & 0 & 0 & 0 \\ a_2 & b_2 & c_2 & \dots & 0 & 0 & 0 \\ 0 & a_3 & b_3 & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & b_{N-2} & c_{N-2} & 0 \\ 0 & 0 & 0 & \dots & a_{N-1} & b_{N-1} & c_{N-1} \\ 0 & 0 & 0 & \dots & 0 & a_N & b_N \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ \vdots \\ u_{N-2} \\ u_{N-1} \\ u_N \end{pmatrix} = \begin{pmatrix} d_1 \\ d_2 \\ d_3 \\ \vdots \\ d_{N-2} \\ d_{N-1} \\ d_N \end{pmatrix} \tag{5.40}$$

We apply Gauss elimination on this system of equations. Eliminate the unknown u_{i-1} from equation number i using equation number $i-1$, working downward and starting in the first column of row number two. Arriving in row number i we have,

$$\alpha_{i-1} u_{i-1} + c_{i-1} u_i = s_{i-1} \tag{5.41}$$

$$a_i u_{i-1} + b_i u_i + c_i u_{i+1} = d_i \tag{5.42}$$

where $\alpha_1 = b_1,\ s_1 = d_1$. Elimination of u_{i-1} gives,

$$\left(b_i - \frac{a_i c_{i-1}}{\alpha_{i-1}} \right) u_i + c_i u_{i+1} = d_i - \frac{a_i s_{i-1}}{\alpha_{i-1}} \tag{5.43}$$

$$\alpha_i = b_i - \frac{a_i c_{i-1}}{\alpha_{i-1}}, \quad s_i = d_i - \frac{a_i s_{i-1}}{\alpha_{i-1}}, \quad i = 2, 3, \dots \tag{5.44}$$

The Gauss elimination process results in a matrix $\mathbf{A}$ with two non-zero coefficients per matrix row, the diagional coefficient, $A_{ii} = \alpha_i$ and $A_{i\ i+1} = c_i$. After the Gauss elimination the solution vector is obtained from the matrix $\mathbf{A}$ by back substitution, applying $c_N = 0$,

$$u_N = \frac{s_N}{\alpha_N} \tag{5.45}$$

$$u_i = \frac{1}{\alpha_i} \left(s_i - c_i u_{i+1} \right),\ i = N-1,\ N-2,\ \dots, 1 \tag{5.46}$$

This can be summarized in the following two-stage procedure. For given vectors $\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d}$:

- compute the vectors $\alpha_i, s_i,\ i = 1, 2, \dots N$
- compute the solution vector $\mathbf{U}$ recursevely, $u_i,\ i = N,\ N-1,\ \dots,\ 1$.

[2] `http://en.wikipedia.org/wiki/Tridiagonal_matrix_algorithm`
`wikibooks.org/wiki/Algorithm_Implementation/Linear_Algebra/Tridiagonal_matrix_algorithm#Fortran_90`

5.6 Implementation of boundary conditions

5.6.1 Natural boundary conditions

Here $\partial T/\partial z$ is given in one of the boundary points $z = 0, z = z_{max}$. This boundary condition can be substituted directly into (5.7). In case natural boundary conditions are given for both boundary points, we get a system of N equations in N unknowns, where N is the number of nodal points of the 1-D mesh.

The special case of a steady state problem with $\partial T/\partial t = 0$ must be considered separately here. We have seen in Chapter 3 that the potential problem with natural boundary condition on the whole boundary requires a compatibility condition for the boundary condition and that the solution, is non-unique (problem 3.13). This can be verified to hold also for the 1-D finite element case treated here, where the fem equations are (5.7),

$$\sum_J S_{IJ}T_J = Q_I + \left(k\frac{\partial T}{\partial z}\right)_{z_{max}} \delta_{IN} - \left(k\frac{\partial T}{\partial z}\right)_0 \delta_{I1},\ I = 1, 2, \ldots, N \tag{5.47}$$

As an example consider the special case with a single linear element,

$$\mathbf{ST} = \frac{k}{h}\begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}\begin{pmatrix} T_1 \\ T_2 \end{pmatrix} = \begin{pmatrix} Q_1 \\ Q_2 \end{pmatrix} + \begin{pmatrix} -q_1 \\ q_2 \end{pmatrix} \tag{5.48}$$

where q_I are the heatflow density values in the nodal points. The element stiffness matrix is singular (det $\mathbf{S}$= 0) and non-unique solutions exist only if a compatibility condition holds for the right-hand side of the equation. This condition is found by summation of the two equations.

$$0 = Q_1 + Q_2 + q_2 - q_1 \tag{5.49}$$

problem 5.7. *Give a physical interpretation of the compatibility condition (5.49).*

problem 5.8. *Verify that homogeneous natural boundary conditions are implicitly accounted for in the finite element method.*

5.6.2 Essential boundary conditions

In case essential boundary conditions apply in both boundary points, the (1-D) problem has $N - 2$ degrees of freedom, corresponding to $N - 2$ internal nodal points. The solution can be written as,

$$\begin{aligned} T(z,t) &\approx \sum_{L=2}^{N-1} T_L(t)N_L(z) + T_1(t)N_1(z) + T_N(t)N_N(z) \\ &= T^*(z,t) + T_1(t)N_1(z) + T_N(t)N_N(z) \end{aligned} \tag{5.50}$$

The function T^* introduced in (5.50) is in a subspace $S_0 \subset S$ of functions with zero boundary values. Where S is the N-dimensional function space spanned by the basis functions $N_J(x), J = 1, \ldots N$. Apply the Galerkin principle to S_0 instead of S, i.e. let $I = 2, 3, \ldots, N - 1$ in the testfunctions. In that case the boundary term in the Galerkin-finite element equations is zero. The terms in T_1 and T_N in the equation (5.7) contain only known quantities,

$$\sum_{J=1}^{N} M_{IJ}\frac{\partial T_J}{\partial t} + \sum_{J=1}^{N} S_{IJ}T_J = Q_I,\ I = 2, 3, \ldots,\ N - 1 \tag{5.51}$$

Therefore these terms contribute to the righthand side vector. This is illustrated in the following example for a steady-state problem,

$$\mathbf{SU} = \mathbf{Q} \tag{5.52}$$

or writing the matrix vector product,

$$\sum_{J=1}^{N} S_{IJ}U_J = Q_I,\ I = 2, \ldots,\ N-1 \tag{5.53}$$

$$\sum_{J=2}^{N-1} S_{IJ}U_J = Q_I - S_{I1}U_1 - S_{IN}U_N,\ I = 2, \ldots,\ N-1 \tag{5.54}$$

for given values of $U_1,\ U_N$. The matrix equation is reduced in size and the righthand side vector modified to,

$$\mathbf{R} = \mathbf{Q} - U_1\mathbf{S_1} - U_N\mathbf{S_N} \tag{5.55}$$

where $\mathbf{S}_1$ and $\mathbf{S}_N$ are the first and last column vectors of the stiffness matrix $\mathbf{S}$

5.7 Steady state problems

An important special case of the equations described in the previous sections occurs if the solution is independent of time. As an example we describe a case with homogeneous essential boundary condition for $z = 0$, $T(0) = 0$. For the other boundary point we consider two possibilities,

1. $$(T)_{z_{max}} = T_m \quad (\textit{essential boundary condition}) \tag{5.56}$$

2. $$(k\partial T/\partial z)_{z_{max}} = q_m \quad (\textit{natural boundary condition}) \tag{5.57}$$

This problem corresponds to a steady state heatconduction problem for a laterally homogeneous layer with vertically varying thermal diffusivity κ, prescribed temperature on the surface $z = 0$ and for z_{max} either a prescribed temperature or a a prescribed heatflow. The equations for both cases with explicit right hand side contributions of the boundary conditions are,

1. $$\sum_{J=2}^{N-1} S_{IJ}T_J = F_I - S_{IN}T_m,\ I = 2, \ldots, N-1 \tag{5.58}$$

2. $$\sum_{J=2}^{N} S_{IJ}T_J = F_I + q_m\delta_{IN},\ I = 2, \ldots, N \tag{5.59}$$

where the righthand side vector $\mathbf{F}$ is defined as,

$$F_I = \int_0^{z_{max}} f(z)N_I(z)\ dz \tag{5.60}$$

and $f(z)$ is the distribution of internal heating. For given f, T_m or q_m this system can be solved for the unknown temperature T.

5.8 Using higher order basis functions

In the previous sections we considered mainly application of linear basis functions, here we shall compare solutions of some simple problems solved by applying various basis functions.

We consider the solution of the one-dimensional Poisson equation, $-d^2u/dz^2 = f$ on the domain $0 \leq z \leq 1$ with essential boundary conditions in the boundary points $z = (0, 1)$.

A solution derived from linear basis functions

To investigate the finite element solution of the 1-D Poisson problem we apply a uniform 1-D mesh consisting of a total of four equidistant nodal points spanning three finite elements with corresponding linear basis functions. Each of the nodal points is associated with a specific basis function with a unit value in the nodal point considered. A global stiffness matrix for this problem can be constructed from the three 2×2 element matrices defined in section 5.3.2 by the assembly process discussed in section 5.4.

problem 5.9. *Derive the following global matrix by assembling the three element matrices,*

$$\mathbf{S} = \frac{1}{h}\begin{pmatrix} 1 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 1 \end{pmatrix} \tag{5.61}$$

where $h = 1/3$ is the length of the elements.

problem 5.10. *As an application of (5.61) we consider first a case with $f = 0$ corresponding to a 1-D Laplace equation. Define the essential boundary conditions as, $u(0) = 0$ and $u(1) = 1$ and derive the following system of equations $\mathbf{SU} = \mathbf{R}$ for the degrees of freedom in the finite element solution corresponding to the internal nodal points,*

$$\frac{1}{h}\begin{pmatrix} 2 & -1 \\ -1 & 2 \end{pmatrix}\begin{pmatrix} U_2 \\ U_3 \end{pmatrix} = \frac{1}{h}\begin{pmatrix} U_1 \\ U_4 \end{pmatrix} = \frac{1}{h}\begin{pmatrix} 0 \\ 1 \end{pmatrix} \tag{5.62}$$

Solve these equations and compare the nodal point solution values with the analytical solution of the corresponding 1-D Laplace equation. Verify that in this case the finite element solution and the analytical solution are identical.

Next we consider an extension of the above problem to a 1-D Poisson equation with a uniform right hand side function $f(z) = H$, $H > 0$ a constant. We specify homogeneous essential boundary conditions $u(0) = u(1) = 0$. This represents the steady state temperature u in a heat conduction problem for a layer with uniform internal heat production rate H and prescribed zero temperature at the top and bottom boundary.

problem 5.11. *Derive the following analytical solution for this Poisson problem,*

$$u(z) = \frac{1}{2}Hz(1 - z) \tag{5.63}$$

The Poisson problem can be solved numerically on the same four-point finite element mesh as before.

problem 5.12. *Verify that the following finite element equations hold for the Poisson problem on the four-point mesh.*

$$\frac{1}{h}\begin{pmatrix} 2 & -1 \\ -1 & 2 \end{pmatrix}\begin{pmatrix} U_2 \\ U_3 \end{pmatrix} = \begin{pmatrix} F_2 \\ F_3 \end{pmatrix} = hH\begin{pmatrix} 1 \\ 1 \end{pmatrix} \tag{5.64}$$

Derive the numerical solution, $U_2 = U_3 = h^2H$ and compare this result with the analytical solution.

A solution derived from second order basis functions

In section 4.1 second order Lagrangian basis functions were introduced, illustrated in Fig. 4.2. The element stiffness matrix of the corresponding 3-point elements for the 1-D Poisson equation $-d^2u/dx^2 = f$ is,

$$\mathbf{S} = \frac{4}{3h^2}\begin{pmatrix} \frac{7}{4} & -1 & -1 \\ -1 & 4 & -1 \\ -1 & -1 & \frac{7}{4} \end{pmatrix} \tag{5.65}$$

We consider a simple example of an application of these second order basis functions in a mimimum 1-D grid consisting of a single (3-point) element to the problem of the previous paragraph with homogeneous essential boundary conditions and uniform righthand side function $f(z) = H$.

The only remaining degree of freedom of this problem is the nodal point value U_2 for the internal nodal point.

problem 5.13. *Derive for the only component of the righthand side vector of the finite element equation,*

$$F_2 = \int_0^1 HN_2 dz = \frac{2}{3}H \tag{5.66}$$

problem 5.14. *Solve the finite element equation and compare the outcome $U_2 = H/8$ with the analytical solution.*

Chapter 6

A finite element solution for multi-dimensional potential problems

In Chapter 5 a finite element solution of the one-dimensional heat equation was presented. The special case of the steady state equation is an example of an elliptic equation like the Poisson equation for the gravitational potential of a mass distribution. This type of equation is quite common in many fields of science and engineering. An other example of an application is in models for steady state groundwater flow. Numerical solutions for 2-D elliptic equations were described in Chapter 3 and in Chapter 4 it was shown, in general terms for multi-dimensional problems, how the finite element method can be used to solve such problems.

Here we shall describe finite element solutions for potential problems and develop the application for two-dimensional problems in more detail. The formulation of the corresponding time dependent problems is similar to the 1-D cases described in Chapter 5.

We start from the generalized Poisson type equation with variable coefficient $c(\mathbf{x})$,

$$-\partial_j \left(c(\mathbf{x})\partial_j u(\mathbf{x})\right) = f(\mathbf{x}) \tag{6.1}$$

This equation applies to the steady state heat conduction problem for a medium with variable conductivity. The Poisson equation follows from (6.1) for a uniform coefficient c with $\nabla c = \mathbf{0}$. For the generalized Laplace differential operator in (6.1) we shall describe a class of element matrices defined by so called isoparametric elements.

In operator notation equation (6.1) is written as,

$$-\nabla \cdot c\nabla u = f \tag{6.2}$$

We consider the following type of boundary conditions,

$$u(\mathbf{x}) = g(\mathbf{x}), \quad \mathbf{x} \in \Gamma_g \tag{6.3}$$

$$c(\mathbf{x})\nabla u(\mathbf{x}) \cdot \mathbf{n} = h(\mathbf{x}), \quad \mathbf{x} \in \Gamma_h \tag{6.4}$$

$$\alpha(\mathbf{x})u(\mathbf{x}) + c(\mathbf{x})\nabla u(\mathbf{x}) \cdot \mathbf{n} = r(\mathbf{x}), \quad \mathbf{x} \in \Gamma_i \tag{6.5}$$

where g, h, α and r are given functions of the spatial coordinates.

problem 6.1. *Consider a steady state heat conduction problem for a rectangular 2-D model of a vertical cross section through the Earth's lithosphere. The temperature at the bottom and top are*

T_m and T_0 respectively. We assume a horizontal symmetry condition on the vertical boundaries. Which formulations from (6.3,6.4) shall we choose to implement the boundary conditions on the four boundaries of the domain? How does this change if we prescribe the mantle heatflow $\mathbf{q}_m$ instead of the mantle temperature?

An example of the mixed type boundary condition (6.5) is found in heat tranport problems in cases where the heatflow density through the boundary Γ_i is assumed to be proportional to the temperature contrast across the boundary (a so called radiation condition). With the heatflow density defined by $\mathbf{q} = -c\nabla u$, we obtain from (6.5),

$$q_n = \alpha \left(u - \frac{r}{\alpha} \right) \tag{6.6}$$

we see that r/α acts as a reference temperature in defining the temperature contrast driving the heatflow across the boundary. α can be interpreted as the inverse of a *thermal resistance* coefficient.

problem 6.2. *We whish to model numerically a physical (lab) Rayleigh-Benard thermal convection experiment. In this experiment viscous fluid in a tank is heated from below and cooled from the top. The fluid layer of thickness h is bounded below and above by copper layers of thickness l. The temperature of the exterior surface of the copper layers is kept at constant values $T_0 + \Delta T$ and T_0 for the bottom and top respectively by separate circuits of heating/cooling liquid in contact with the bottom and top copper plates.*

How can we apply boundary condition type (6.5) to this problem.

Hint: neglect horizontal heat transport in the copper plates and assume that the heatflow density $q_n(x)$ can be described in terms of the local temperature contrast across the copper plates.

6.1 Discretization of the equation

We shall first describe a finite element solution for the elliptic problem based on general element types for 2-D or 3-D problems similar to the description in Chapter 4. In later sections more detailed examples will be given of such solutions for 2-D triangular elements combined with linear basis functions and quadrilateral elements with bi-linear basis functions.

In the Bubnov-Galerkin formulation the partial differential equation is transformed by integration over the domain, resulting in a system of linear algebraic equations.

$$\int_V N_I \left\{ -\nabla \cdot c\nabla u - f \right\} \, dV = 0, \; I = 1, \ldots, N \tag{6.7}$$

Where N is the number of degrees of freedom. Integrating by parts gives,

$$\int_V \left\{ -\nabla \cdot (N_I c\nabla u) + \nabla N_I \cdot c\nabla u - N_I f \right\} \, dV = 0 \tag{6.8}$$

$$-\int_{\partial V} N_I c\nabla u \cdot \mathbf{n} \, dA + \int_V \nabla N_I \cdot c\nabla u \, dV = \int_V N_I f \, dV, \; I = 1, \ldots, N \tag{6.9}$$

Substitution of the expansion in interpolating Lagrangian basis functions,

$$u(\mathbf{x}) = \sum_J U_J N_J(\mathbf{x}), \; U_J = u(\mathbf{x}_J) \tag{6.10}$$

we get for the second term in (6.9)

$$\sum_J \left\{ \int_V \nabla N_I \cdot c\nabla N_J \, dV \right\} U_J = \sum_J S_{IJ} \, U_J \tag{6.11}$$

where the stiffness matrix $\mathbf{S}$ is defined by,

$$S_{IJ} = \int_V \nabla N_I \cdot c \nabla N_J \; dV \tag{6.12}$$

The expression for $\mathbf{S}$ is often rewritten in a different form that is also used in later chapters on vector problems dealing with elastic deformation and viscous flow. To this end we introduce a matrix $\mathbf{B}$ where the matrix columns are defined in terms of the gradient of the finite element basis functions. For a 3-D problem this gives,

$$\mathbf{B} = (\nabla N_1, \ldots, \nabla N_N) = \begin{pmatrix} \partial_1 N_1, \ldots, \partial_1 N_N \\ \partial_2 N_1, \ldots, \partial_2 N_N \\ \partial_3 N_1, \ldots, \partial_3 N_N \end{pmatrix} \tag{6.13}$$

or alternatively,

$$\mathbf{B} = \begin{pmatrix} \partial_1 \cdot \\ \partial_2 \cdot \\ \partial_3 \cdot \end{pmatrix} (N_1, \ldots, N_N) \tag{6.14}$$

The column vectors of the matrix $\mathbf{B}$ are,

$$\mathbf{B}_I = \nabla N_I = (\partial_1 N_I, \partial_2 N_I, \partial_3 N_I)^T \tag{6.15}$$

For the coefficients of the stiffness matrix we get,

$$S_{IJ} = \int_V c \nabla N_I \cdot \nabla N_J \; dV = \int_V \mathbf{B}_I \cdot \mathbf{D}\mathbf{B}_J \; dV = \int_V \mathbf{B}_I^T \mathbf{D}\mathbf{B}_J \; dV \tag{6.16}$$

where $\mathbf{B}_I$ and $\mathbf{B}_J$ are columnvectors and $D_{ij} = c\delta_{ij}$, $i, j = 1, 2, 3$. The global stiffness matrix can be written as a summation of matrices and assembled from the contributions of element matrices,

$$\mathbf{S} = \int_V \mathbf{B}^T \mathbf{D}\mathbf{B} \; dV = \sum_K \int_{e_K} \mathbf{B}^T \mathbf{D}\mathbf{B} \; dV = \sum_K \mathbf{S}^{(K)} \tag{6.17}$$

where $\mathbf{S}^{(K)}$ is the element matrix of element e_K. The summation over elements corresponds to the matrix assembly proces described for 1-D cases in Chapter 5.

problem 6.3. *Derive the following expressions for the element matrix $\mathbf{S}^{(K)}$ for a 1-D element with two degrees of freedom and a 2-D triangular element with three degrees of freedom. For the 1-D element,*

$$\mathbf{S}^{(K)} = \int_{e_K} c \begin{pmatrix} \frac{dN_1}{dz}\frac{dN_1}{dz} & \frac{dN_1}{dz}\frac{dN_2}{dz} \\ \frac{dN_2}{dz}\frac{dN_1}{dz} & \frac{dN_2}{dz}\frac{dN_2}{dz} \end{pmatrix} dz \tag{6.18}$$

And for a 2-D traingular element, associated with 3 degrees of freedom and 3 basis functions introduced in section 6.2.1,

$$\mathbf{S}^{(K)} = \int_{e_K} c \begin{pmatrix} (\partial_x N_1)^2 + (\partial_y N_1)^2 & \partial_x N_1 \partial_x N_2 + \partial_y N_1 \partial_y N_2 & \partial_x N_1 \partial_x N_3 + \partial_y N_1 \partial_y N_3 \\ \partial_x N_2 \partial_x N_1 + \partial_y N_2 \partial_y N_1 & (\partial_x N_2)^2 + (\partial_y N_2)^2 & \partial_x N_2 \partial_x N_3 + \partial_y N_2 \partial_y N_3 \\ \partial_x N_3 \partial_x N_1 + \partial_y N_3 \partial_y N_1 & \partial_x N_3 \partial_x N_2 + \partial_y N_3 \partial_y N_2 & (\partial_x N_3)^2 + (\partial_y N_3)^2 \end{pmatrix} dxdy$$

where local numbering (per element) of the basis functions has been applied.

An expression like (6.17) for the stiffness matrix is common in finite element formulations. A similar matrix occurs in elastic deformation problems and viscous flow problems treated in chapters 10 and 11 respectively.

The righthand side of the equations (6.9) is written as a vector $\mathbf{F}$,

$$\int_V N_I f \; dV = F_I \tag{6.19}$$

The integration over the domain written as a sum of element contributions is,

$$F_I = \sum_K F_I^{(K)} = \sum_K \int_{e_K} N_I f \; dV \tag{6.20}$$

In order to further define the boundary integral in (6.9) we must specify the boundary conditions. Treatment of essential boundary conditions is similar to the treatment for the 1-D special case in Chapter 5. The number of degrees of freedom is reduced by the number of prescribed boundary values and the number of Galerkin equations can then be reduced accordingly. This is done by dropping those test functions that correspond to the boundary points with an essential boundary condition. The remaining testfunctions have zero boundary values on Γ_g resulting in a zero contribution from the boundary integral over Γ_g. The prescribed boundary values give a contribution to the righthand side vector which can be made explicit by partitioning the solution vector and stiffness matrix (see Chapter 5).

In the case of natural boundary conditions (6.4), the boundary integral in (6.9) contains only known functions and it can be subtracted from the righthand side of the equation,

$$F_I^{(2)} = \int_{\Gamma_h} N_I c \nabla u \cdot \mathbf{n} \; dA = \sum_M \int_{b_M} N_I h \; dA = \sum_M F_I^{(2)(M)} \tag{6.21}$$

The boundary integral is split here in a sum over contributions from *boundary elements*, b_M, with $\Gamma_h = \cup_M b_M$.

In the case of mixed boundary conditions (6.5) we find for the boundary integral,

$$\begin{aligned} I_I &= \int_{\Gamma_i} N_I c \nabla u \cdot \mathbf{n} \; dA = \int_{\Gamma_i} N_I \left(r - \alpha u\right) \; dA \\ &= \int_{\Gamma_i} N_I r \; dA - \sum_J \int_{\Gamma_i} \alpha N_I N_J \; dA \; U_J \\ &= F_I^{(3)} - \sum_J S_{IJ}^{(3)} U_J \end{aligned} \tag{6.22}$$

The term $\mathbf{F}^{(3)}$ is a contribution to the righthand side of the finite element equation. The term $\mathbf{S}^{(3)}$ represents a contribution to the stiffness matrix that can be written in terms of boundary element contributions,

$$\mathbf{S}^{(3)} = \sum_M \mathbf{S}^{(3)(M)} \tag{6.23}$$

6.1.1 Element matrices for the 2-D case

Integration for the element matrix $\mathbf{S}^{(K)}$ is often performed by numerical integration. To this end an arbitrary element, spanned by n_e nodal points, is mapped first onto a standard (unit)

element e_u with regular geometry such as a square in the case of a general quadrilateral element (see section 6.2.2). This is done by means of a coordinate transformation from the actual (x, y) coordinates to the so called *natural coordinates* (ξ, η) describing the unit element. We shall apply this procedure to obtain a general scheme that applies for different choices of the basis functions N_I. This general scheme will then be applied to a triangular element with linear basis functions in section 6.2.1 and to a quadrilateral element with bi-linear basis functions in section 6.2.2.

In this procedure a coordinate transformation maps an arbitrary element on the standard element.

$$(x, y) \;\rightarrow\; (\xi(x, y), \eta(x, y)) \tag{6.24}$$

x, y are referred to as the *global* coordinates, applied to the entire domain, ξ and η are referred to as *local* or *natural coordinates* of the element type. The volume integral for the element stiffness matrix (6.17) is rewritten into an integral over the standard element,

$$\mathbf{S}^{(K)} = \int_{e_K} \mathbf{B}^T \mathbf{D} \mathbf{B}\; dV(x, y) = \int_{e_u} \mathbf{B}^T \mathbf{D} \mathbf{B}\; J\; dV(\xi, \eta) = \tag{6.25}$$

where J is the Jacobian determinant of the transformation matrix corresponding to the coordinate transformation,

$$J = det \begin{pmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{pmatrix} = \frac{\partial x}{\partial \xi}\frac{\partial y}{\partial \eta} - \frac{\partial x}{\partial \eta}\frac{\partial y}{\partial \xi} \tag{6.26}$$

Note that we use the same symbol for $\mathbf{B}$ in both coordinate systems. The difference will be clear from the context.

The matrix $\mathbf{B}$ transforms together with the differential form of the gradient operator,

$$\begin{pmatrix} \partial_x \cdot \\ \partial_y \cdot \end{pmatrix} = \begin{pmatrix} \frac{\partial \xi}{\partial x} & \frac{\partial \eta}{\partial x} \\ \frac{\partial \xi}{\partial y} & \frac{\partial \eta}{\partial y} \end{pmatrix} \begin{pmatrix} \frac{\partial \cdot}{\partial \xi} \\ \frac{\partial \cdot}{\partial \eta} \end{pmatrix} = \begin{pmatrix} j_{11} & j_{12} \\ j_{21} & j_{22} \end{pmatrix} \begin{pmatrix} \frac{\partial \cdot}{\partial \xi} \\ \frac{\partial \cdot}{\partial \eta} \end{pmatrix} \tag{6.27}$$

The matrix columns $\mathbf{B}_I$ transform as,

$$\begin{aligned} \mathbf{B}_I &= \begin{pmatrix} \partial_x N_I \\ \partial_y N_I \end{pmatrix} = \begin{pmatrix} \partial_x \cdot \\ \partial_y \cdot \end{pmatrix} N_I \\ &= \begin{pmatrix} j_{11} & j_{12} \\ j_{21} & j_{22} \end{pmatrix} \begin{pmatrix} \frac{\partial \cdot}{\partial \xi} \\ \frac{\partial \cdot}{\partial \eta} \end{pmatrix} N_I = \begin{pmatrix} j_{11} & j_{12} \\ j_{21} & j_{22} \end{pmatrix} \begin{pmatrix} \frac{\partial N_I}{\partial \xi} \\ \frac{\partial N_I}{\partial \eta} \end{pmatrix} \end{aligned} \tag{6.28}$$

The transformation matrix $\mathbf{j}$ is the inverse of the Jacobi matrix $\mathbf{J}$,

$$\mathbf{J} = \begin{pmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{pmatrix} = \begin{pmatrix} \frac{\partial \xi}{\partial x} & \frac{\partial \eta}{\partial x} \\ \frac{\partial \xi}{\partial y} & \frac{\partial \eta}{\partial y} \end{pmatrix}^{-1} = \mathbf{j}^{-1} \tag{6.29}$$

For so called *isoparametric* finite elements the coefficients of the Jacobi matrix $\mathbf{J}$ can be expressed in a simple way in the derivatives of the basis functions. For isoparametric elements the coordinate transformation is defined in a characteristic way in terms of the basis functions as,

$$\mathbf{x}(\xi, \eta) = \sum_{K=1}^{n_e} \mathbf{x}_K N_K(\xi, \eta) \tag{6.30}$$

where the summation is over the nodal points in a single element. For higher order basis functions we can map so called curvi-linear elements in (x, y) space on a standard element

with straight-line boundaries in (ξ, η) space, using an isoparametric transformation. With (6.30) the geometry of the elements is defined in terms of the same basis functions as the problem solution (hence the name isoparametric). This offers a great flexibility in the accurate discretization of a domain with complicated geometry mostly applied to create smooth discretizations of curved boundaries in a finite element mesh.

Applying the transformation according to (6.30) we find,

$$J_{11} = \frac{\partial x}{\partial \xi} = \sum_K x_K N_{K\xi}(\xi, \eta) \tag{6.31}$$

$$J_{12} = \frac{\partial y}{\partial \xi} = \sum_K y_K N_{K\xi}(\xi, \eta) \tag{6.32}$$

$$J_{21} = \frac{\partial x}{\partial \eta} = \sum_K x_K N_{K\eta}(\xi, \eta) \tag{6.33}$$

$$J_{22} = \frac{\partial y}{\partial \eta} = \sum_K y_K N_{K\eta}(\xi, \eta) \tag{6.34}$$

This way the elements of the Jacobi matrix $\mathbf{J}$ have been expressed in the derivatives of the basis functions and the transformation matrix $\mathbf{j}$ follows by explicit inversion of the 2×2 Jacobi matrix by Cramers rule,

$$\mathbf{j} = \mathbf{J}^{-1} = J^{-1} \begin{pmatrix} J_{22} & -J_{12} \\ -J_{21} & J_{11} \end{pmatrix} \tag{6.35}$$

These expressions for general 2-D isoparametric elements are used in the following sections applied to triangular linear and quadrilateral bi-linear elements.

6.2 Examples of 2-D elements

6.2.1 The triangular linear element

As a first example of the computation of element matrices for 2-D elements we consider a triangular element spanned by three nodal points and linear basis functions. The integration in the expression for the element stiffness matrix is performed on a standard triangular element in the (ξ, η) plane. We map an arbitrary element e on the standard element e_u as shown in Fig. 6.1

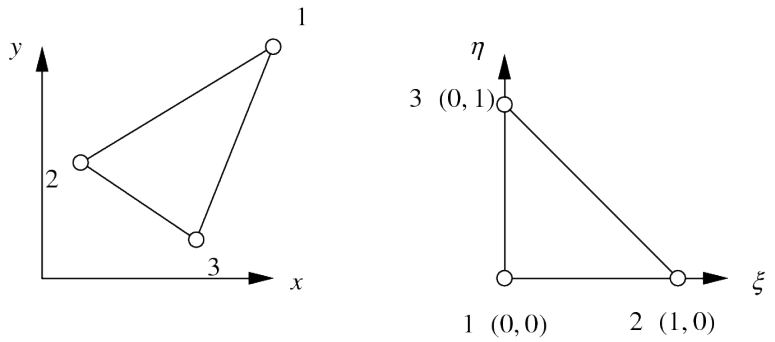

Figure 6.1: *Geometry of a triangular isoparametric element $e(x, y)$ mapped on the standard triangle in the first quadrant of the (ξ, η) plane.*

We assume an isoparametric element. This implies that the coordinate transformation T is defined by the linear basis functions on the standard element in (ξ, η) space,

$$x(\xi, \eta) = x_1 + (x_2 - x_1)\xi + (x_3 - x_1)\eta \tag{6.36}$$

$$y(\xi, \eta) = y_1 + (y_2 - y_1)\xi + (y_3 - y_1)\eta \tag{6.37}$$

where,

$$(0,0) \xrightarrow{T} \mathbf{x}_1, \quad (1,0) \xrightarrow{T} \mathbf{x}_2, \quad (0,1) \xrightarrow{T} \mathbf{x}_3$$

Expressions for the basis functions in the natural coordinates can now be found from the definition of the coordinate transformation T defined in (6.36) and (6.37), by identification with the formal expression for an isoparametric transformation,

$$\mathbf{x}(\xi, \eta) = \sum_{K=1}^{3} \mathbf{x}_K N_K(\xi, \eta) \tag{6.38}$$

Rewriting (6.36) and (6.37), as,

$$\mathbf{x}(\xi, \eta) = \mathbf{x}_1(1 - \xi - \eta) + \mathbf{x}_2\xi + \mathbf{x}_3\eta \tag{6.39}$$

we find the linear basis functions and corresponding derivatives listed in Table 6.1

K	N_K	$N_{K\xi}$	$N_{K\eta}$
1	$1 - \xi - \eta$	-1	-1
2	ξ	1	0
3	η	0	1

Table 6.1: *Linear basis functions and derivatives of the linear isoparametetric triangle.*

problem 6.4. *Verify that the basis functions have the* $\{0|1\}$ *property of Lagrange polynomials on the standard element with*

$$N_K(\xi_J, \eta_J) = \delta_{KJ} \tag{6.40}$$

make a sketch in 3-D perspective of the surface of the basis function value, in (ξ, η) *and* (x, y) *space.*

The constant Jacobi matrix of this transformation is,

$$\mathbf{J} = \begin{pmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{pmatrix} = \begin{pmatrix} x_2 - x_1 & y_2 - y_1 \\ x_3 - x_1 & y_3 - y_1 \end{pmatrix} \tag{6.41}$$

For the uniform Jacobian of the transformation we get,

$$J = det(\mathbf{J}) = (x_2 - x_1)(y_3 - y_1) - (x_3 - x_1)(y_2 - y_1) \tag{6.42}$$

problem 6.5. *Verify that the Jacobian equals twice the surface area of the element triangle. Hint: transform the integral* $\int_e dV(x, y)$ *into an integral over the standard element.*

The integral expression for the stiffness matrix contains the uniform Jacobian. For element K we have,

$$\mathbf{S}^{(K)} = \int_{e_K} \mathbf{B}^T \mathbf{D} \mathbf{B} \; dV(x, y) = J \int_0^1 \int_0^{1-\eta} \mathbf{B}^T \mathbf{D} \mathbf{B} \; d\xi d\eta \tag{6.43}$$

From the definition (6.28) and the uniform derivatives in Table 6.1 it follows that the matrix $\mathbf{B}$ is uniform in ξ, η. Substituting $D_{ij} = c\delta_{ij}$ we get,

$$\mathbf{S}^{(K)} = J \mathbf{B}^T \mathbf{B} \int_{e_u} c(\xi, \eta) \; dV(\xi, \eta) \tag{6.44}$$

The matrix product $\mathbf{B}^T\mathbf{B}$ can be further specified. Substituting, (6.28) and Table 6.1 we get,

$$\mathbf{B} = \mathbf{j} \begin{pmatrix} N_{1\xi} & N_{2\xi} & N_{3\xi} \\ N_{1\eta} & N_{2\eta} & N_{3\eta} \end{pmatrix} = \mathbf{j} \begin{pmatrix} -1 & 1 & 0 \\ -1 & 0 & 1 \end{pmatrix} \tag{6.45}$$

Furthermore we have for the transformation matrix $\mathbf{j} = \mathbf{J}^{-1}$,

$$\mathbf{j} = \frac{1}{J} \begin{pmatrix} (y_3 - y_1) & -(y_2 - y_1) \\ -(x_3 - x_1) & (x_2 - x_1) \end{pmatrix} \tag{6.46}$$

Substitution in (6.45) gives,

$$\mathbf{B} = \frac{1}{J} \begin{pmatrix} (y_2 - y_3) & (y_3 - y_1) & -(y_2 - y_1) \\ (x_3 - x_2) & -(x_3 - x_1) & (x_2 - x_1) \end{pmatrix} \equiv \frac{1}{J} \begin{pmatrix} e_1 & e_2 & e_3 \\ d_1 & d_2 & d_3 \end{pmatrix} \tag{6.47}$$

$$\begin{aligned} \mathbf{B}^T\mathbf{B} &= \frac{1}{J} \begin{pmatrix} e_1 & d_1 \\ e_2 & d_2 \\ e_3 & d_3 \end{pmatrix} \frac{1}{J} \begin{pmatrix} e_1 & e_2 & e_3 \\ d_1 & d_2 & d_3 \end{pmatrix} \\ &= \frac{1}{J^2} \begin{pmatrix} e_1^2 + d_1^2 & sym. & sym. \\ e_1e_2 + d_1d_2 & e_2^2 + d_2^2 & sym. \\ e_1e_3 + d_1d_3 & e_2e_3 + d_2d_3 & e_3^2 + d_3^2 \end{pmatrix} \end{aligned} \tag{6.48}$$

We finally have for the element stiffness matrix of the linear triangular element,

$$\mathbf{S}^{(K)} = C\mathbf{B}^T\mathbf{B} \tag{6.49}$$

where the scalar quantity C is related to the local (element) average of the coefficient $c(\mathbf{x})$.

$$C = \int_{e_u} c(\xi,\eta)\ J\ dV(\xi,\eta) = \int_e c(x,y)\ dV(x,y) \tag{6.50}$$

The integration in (6.50) can be done numerically for a given function c. In the special case of a piecewise uniform coefficient we obtain,

$$C = c\int_e dV = cA_e = \frac{c}{2}J \tag{6.51}$$

problem 6.6. *Verify that a 'degenerate' element e with* $\int_e dV = 0$ *results in a singularity in the element matrix. Software implementations should include a test for this condition in the matrix assembly procedure.*

problem 6.7. *Suppose we whish to compute the temperature distribution and heatflow for a 2-D model. We use a finite element method where the discrete temperature field is computed as a vector of nodal point values, using triangular elements with linear basis functions for the temperature field.*

- *The heatflow density* $\mathbf{q} = -k\nabla T$ *can be computed in a consistent way from the solution vector of nodal point temperature values. Derive an expression for the heatflow density vector written as a matrix-vector product with the nodal point vector of the temperature* $\mathbf{T}$ *and the matrix* $\mathbf{B}$ *defined in (6.13).*
- *Verify that the vector* $\mathbf{q}(\mathbf{x})$ *is piecewise uniform for triangular elements with piecewise linear basis functions.* $\mathbf{q}(\mathbf{x})$ *is therefore discontinuous across element boundaries.*
- *How could we define an approximating nodal point vector for the heatflow density?*

6.2.2 A quadrilateral element with bi-linear basis functions

In the previous section triangular isoparametric elements with linear basis functions were described. Here we will specify the element matrices, introduced for general basis functions in section 6.1.1, for the special case of a quadrilateral element with bi-linear basis functions $N_I(\xi, \eta)$. The resulting expressions are more difficult to handle analytically compared to the case with linear triangular elements and we shall use numerical integration for the evaluation of the expressions for the matrix elements. Numerical integration is common in finite element applications, even in situations were analytical expressions for matrix elements are available (Zienkiewicz, 1977). With such numerical methods it is simple to include variable coefficients $c(\mathbf{x})$. An other advantage of the numerical integration approach is that the software implementation can be more general and less dependent of specific analytical form of the expressions to be integrated.

The geometry of an arbitrary quadrilateral element and correspoding standard element is shown in Fig. 6.2.

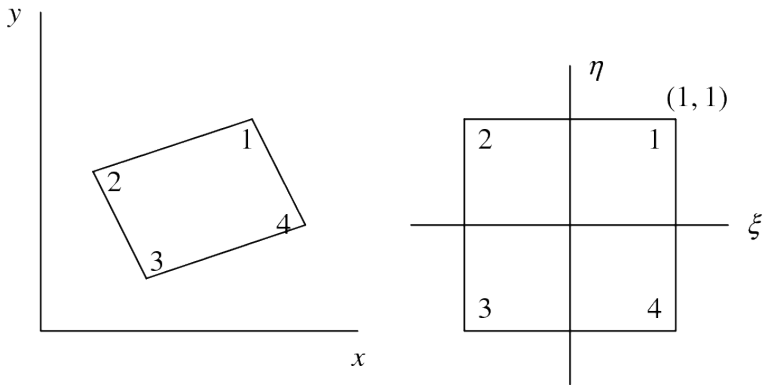

Figure 6.2: *Geometry of a quadrilateral isoparametric element $e(x, y)$ mapped on the unit square of the (ξ, η) plane.*

Assuming bi-linear basis functions $N_I(\xi, \eta)$ implies, for an isoparametric element,

$$x(\xi, \eta) = \alpha_0 + \alpha_1\xi + \alpha_2\eta + \alpha_3\xi\eta \tag{6.52}$$

$$y(\xi, \eta) = \beta_0 + \beta_1\xi + \beta_2\eta + \beta_3\xi\eta \tag{6.53}$$

For the four nodal points (x_L, y_L) we have,

$$x(\xi_L, \eta_L) = x_L \tag{6.54}$$

$$y(\xi_L, \eta_L) = y_L \tag{6.55}$$

In further specifications we do not use the coeffcients (6.52),(6.53) but instead we apply the explicit form of the Lagrangian interpolating bi-linear functions on the standard element. These functions are linear in a single coordinate (bi-linear) and can be written as,

$$N_K = \frac{1}{4}(1 + \xi_K\xi)(1 + \eta_K\eta), \quad K = 1, \ldots, 4 \tag{6.56}$$

problem 6.8. *Verify that the N_K defined in (6.56) has the interpolation property of the Lagrangian polynomials, $N_I(\xi_J, \eta_J) = \delta_{IJ}$ (see Fig. 6.3).*

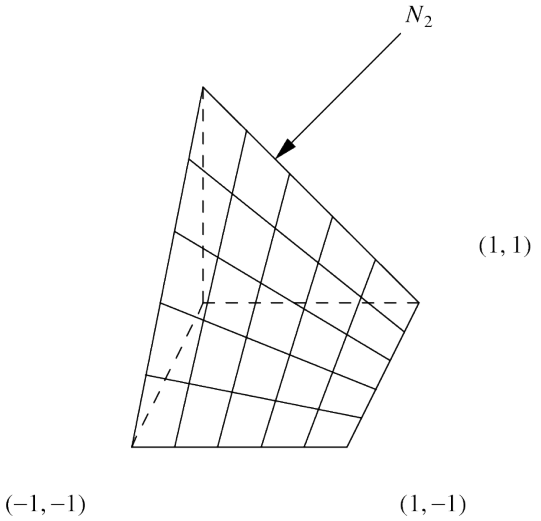

Figure 6.3: *Display of the basis function N_2 on the standard element $[-1,1] \times [-1,1]$.*

The four element basis functions defined in (6.56) and corresponding derivates are listed in Table 6.2.

K	$4N_K$	$4N_{K\xi}$	$4N_{K\eta}$
1	$(1+\xi)(1+\eta)$	$(1+\eta)$	$(1+\xi)$
2	$(1-\xi)(1+\eta)$	$-(1+\eta)$	$(1-\xi)$
3	$(1-\xi)(1-\eta)$	$-(1-\eta)$	$-(1-\xi)$
4	$(1+\xi)(1-\eta)$	$(1-\eta)$	$-(1+\xi)$

Table 6.2: *Bi-linear basis functions and derivatives for a quadrilateral element.*

The derivatives of the basis functions and the Jacobi matrix of the isoparameteric transformation are not constants as in the case of the linear triangular element but functions of the natural coordinates (ξ, η). Because of this the integral expressions for the element stiffness matrix can not be evaluated as easily. In software implementations these integrations are therefore done by numerical integration. In such procedures the integral is replaced by a weighted sum over integrand values. Numerical integration is also convenient in case we have a general functional formulation for a variable coefficient $c(\mathbf{x})$. Analytical evaluation of the integrals for the stiffness matris would not be possible for a general coefficient function $c(\mathbf{x})$.

A general expression for the numerical approximation of an element integral is,

$$\int_e f(\mathbf{x})\ dV \approx \sum_{j=1}^{m} w_j f(\mathbf{x}_j) \tag{6.57}$$

w_j and x_j are weights and evaluation points of the m-point integration scheme. In the appendix a four-point Gauss integration scheme is described that is suitable for the quadrilateral elements considered here. With this numerical scheme the element matrix can be evaluated as,

$$S_{IJ}^{(K)} \quad = \quad \int_{e_u} \mathbf{B}_I^T \mathbf{D} \mathbf{B}_J\ J\ dV(\xi,\eta),\ I,J = 1,\ldots,4$$

$$\approx \sum_{j=1}^{4} w_j \Psi(\xi_j, \eta_j) \tag{6.58}$$

where the integrand function $\Psi = \mathbf{B_I}^T \mathbf{D} \mathbf{B}_J \, J$ is evaluated in the four 'Gauss points'. The matrix product in (6.58) is further specified in the following. The (2×4) matrix $\mathbf{B}$ can be transformed using the data in Table 6.2,

$$\mathbf{B}(\mathbf{x}) = \begin{pmatrix} N_{1x} & \dots & N_{4x} \\ N_{1y} & \dots & N_{4y} \end{pmatrix} \tag{6.59}$$

and after transformation,

$$\mathbf{B}(\xi, \eta)) = \begin{pmatrix} j_{11} N_{1\xi} + j_{12} N_{1\eta} & \dots & j_{11} N_{4\xi} + j_{12} N_{4\eta} \\ j_{21} N_{1\xi} + j_{22} N_{1\eta} & \dots & j_{21} N_{4\xi} + j_{22} N_{4\eta} \end{pmatrix} \tag{6.60}$$

The partial derivatives in (6.60) are given in Table 6.2. De transformation matrix $\mathbf{j}$ can be obtained from the inverse Jacobi matrix,

$$\mathbf{J} = \begin{pmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{pmatrix} \tag{6.61}$$

For the isoparametric element we have,

$$\mathbf{x}(\xi, \eta) = \sum_{I=1}^{4} \mathbf{x}_I N_I(\xi, \eta) \tag{6.62}$$

from which we get,

$$J_{11} = \frac{\partial x}{\partial \xi} = \sum_{I=1}^{4} x_I N_{I\xi} \tag{6.63}$$

$$J_{12} = \frac{\partial y}{\partial \xi} = \sum_{I=1}^{4} y_I N_{I\xi} \tag{6.64}$$

$$J_{21} = \frac{\partial x}{\partial \eta} = \sum_{I=1}^{4} x_I N_{I\eta} \tag{6.65}$$

$$J_{22} = \frac{\partial y}{\partial \eta} = \sum_{I=1}^{4} y_I N_{I\eta} \tag{6.66}$$

This way the Jacobi matrix has been expressed in terms of the element nodal point coordinates (x_I, y_I). The transformation matrix follows from this as,

$$\mathbf{j} = \begin{pmatrix} j_{11} & j_{12} \\ j_{21} & j_{22} \end{pmatrix} = \frac{1}{J} \begin{pmatrix} J_{22} & -J_{12} \\ -J_{21} & J_{11} \end{pmatrix} \tag{6.67}$$

On the element boundaries the basis functions are linear, therefore linear boundary elements can be used for the implementation of boundary conditions of type 2 and 3.

6.3 Application in various boundary value problems

We consider here two modelling problems including the three main types of boundary conditions introduced in section 6.1 were the finite element methods discussed in the previous sections are applied.

In both problems the Poisson equation $-\nabla^2 u = f$ is solved on a square domain $(x, y) \in V = [0, 1] \times [0, 1]$ and boundary $\partial V = \cup_{i=1}^{4} C_i$. The four boundary segments are numbered anti-clock wise starting from the bottom boundary C_1 as illustrated in the diagram in Fig. 6.4. Boundary conditions are posed for the different boundary segments C_i separately.

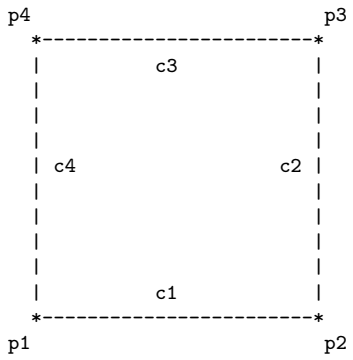

Figure 6.4: *Domain diagram showing the different boundary curves used in specifying the different boundary conditions.*

6.3.1 A problem with essential and natural boundary conditions

In this problem we consider the homogeneous differential equation ($f = 0$) and we apply homogeneous essential boundary conditions $u = 0$ on the top boundary C_3. On the bottom boundary C_1 the normal derivative is prescribed with $\partial u/\partial y = -1$, an inhomogeneous natural boundary condition. In the context of a steady state heat conduction problem this corresponds to a prescribed heatflow. On the vertical boundaries C_2, C_4 we specify horizontal symmetry with $\partial u/\partial x = 0$, corresponding to a zero horizontal heatflux. This is a homogeneous natural boundary condition.

problem 6.9. *Derive the analytical solution for this problem.*
Answer: $u(x, y) = 1 - y$

Hint: first show that a general bi-linear solution exists, $u(x, y) = a + bx + cy + dxy$

6.3.2 A problem with boundary conditions of type 2 and 3

Here the boundary conditions on C_1, C_2 and C_4 are the same as in the previous case. On the top boundary C_3 we apply a boundary condition of type 3,

$$\alpha u + \frac{\partial u}{\partial y} = r \tag{6.68}$$

In section 6.1 it was shown that the first term in (6.68) results in a boundary contribution to the stiffness matrix and that the right hand side term in (6.68) contributes to the right hand side of the finite element equations. For the case of linear elements considered here, simple expressions can be derived for these contributions. Starting from the boundary integral term in the Galerkin equation (6.9) we get,

$$I \;=\; \int_{\Gamma_i} N_I \nabla u \cdot \mathbf{n} \; dA$$

$$
\begin{aligned}
&= \int_{\Gamma_i} N_I \left(r - \alpha u\right)\ dA \\
&= \int_{\Gamma_i} N_I r\ dA - \sum_J \int_{\Gamma_i} \alpha N_I N_J\ dA\ U_J \\
&= F_I^{(3)} - \sum_J M_{IJ}^{(3)}\ U_J
\end{aligned} \tag{6.69}
$$

We further assume that $r(\mathbf{x})$ can be expanded in the basis functions on the boundary and that $\alpha(\mathbf{x})$ is piecewise constant on the boundary elements. This results in the following expression for the right hand side vector contribution,

$$
\begin{aligned}
F_I^{(3)} &= \int_{\Gamma_i} N_I r\ dA \\
&= \int_{\Gamma_i} N_I \sum_J r_J N_J\ dA = \sum_J r_J \int_{\Gamma_i} N_I N_J\ dA \\
&= \sum_J M_{IJ} r_J
\end{aligned} \tag{6.70}
$$

For linear boundary elements corresponding to the linear triangular elements and bi-linear quadrilateral elements described before we have the mass matrix,

$$
\mathbf{M}^{(K)} = \frac{h_K}{6} \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix} \tag{6.71}
$$

From summation over boundary elements e_K we get,

$$
F_I^{(3)} = \sum_K F_I^{(3)(K)} \tag{6.72}
$$

For a boundary element with $e_K \cap S_I \neq \emptyset$, where S_I is the support of the basis function N_I, we have,

$$
\mathbf{F}^{(3)(K)} = \frac{h_K}{6} \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix} \begin{pmatrix} r_1 \\ r_2 \end{pmatrix} \tag{6.73}
$$

For the boundary element contribution to the stiffness matrix (6.70) and piecewise uniform α we derive,

$$
\begin{aligned}
\sum_J M_{IJ}^{(3)} U_J &= \sum_J \int_{\Gamma_i} \alpha N_I N_J\ dA\ U_J \\
&= \sum_J \sum_K \alpha_K \int_{\Gamma_i^{(K)}} N_I N_J\ dA\ U_J = \sum_J \sum_K \alpha_K M_{IJ}^{(K)}\ U_J
\end{aligned} \tag{6.74}
$$

The stiffness matrix contribution of boundary element K is,

$$
\mathbf{S}^{(K)} = \frac{\alpha_K h_K}{6} \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix} \tag{6.75}
$$

problem 6.10. *Derive an analytical solution for the Laplace equation,*

$$
\nabla^2 u = 0 \tag{6.76}
$$

with the following boundary conditions,

$$
\frac{\partial u}{\partial y} = -1,\ \mathbf{x} \in C_1 \tag{6.77}
$$

$$u(\mathbf{x}) = \frac{1}{\alpha}\left(r - \frac{\partial u}{\partial y}\right), \ \mathbf{x} \in C_3 \tag{6.78}$$

$$\frac{\partial u}{\partial x} = 0, \ \mathbf{x} \in C_2 \cup C_4 \tag{6.79}$$

Solution:

$$u(\mathbf{x}) = u(y) = 1 - y + \frac{r+1}{\alpha} \tag{6.80}$$

Simple problems with analytical solutions like (6.80) are useful to 'benchmark' software for the solution of more general differential equations.

Chapter 7

Finite element methods for potential equations: geophysical applications

7.1 Introduction

In this chapter we discuss applications of the finite element methods for elliptic, potential type, boundary value problems introduced in Chapter 6. The following three applications from the geosciences are treated.

Section 7.2 deals with applications in models with purely conductive heat transport, i.e. we exclude convective energy tranport and focus on a static medium. Here we also discuss a solution method for temperature dependent conductivity as an introduction for more generally applicable methods for non-linear formulations of material properties. 7.3 presents an application in models for fluid flow in porous media. Section 7.4 deals with viscous instantaneous (Stokes) flow described by a streamfunction and a vorticity potential.

7.2 Steady state heat conduction

Steady state conductive heat transport is described by the diffusion equation,

$$-\nabla \cdot k(\mathbf{x})\nabla T = \rho H(\mathbf{x}) \tag{7.1}$$

where T is the temperature, k is thermal conductivity, $[k] = \mathrm{Wm^{-1}K^{-1}}$, ρ is the density, ρH is the volumetric rate of internal heat production, $[H] = \mathrm{Wkg^{-1}}$.

Boundary conditions are typically; prescribed temperature (essential boundary condition), $T(\mathbf{x}) = g(\mathbf{x}), \mathbf{x} \in \Gamma_g$ or prescribed boundary heat flow density, $\mathbf{q} \cdot \mathbf{n} = -k\nabla T \cdot \mathbf{n} = h(\mathbf{x}), \mathbf{x} \in \Gamma_h$ (natural boundary condition).

In Chapter 6 finite element equations were presented for problems defined by (7.1) and corresponding boundary condition, resulting in the following system of algebraic equations,

$$\mathbf{ST} = \mathbf{R} \tag{7.2}$$

where the stiffness matrix $\mathbf{S}$ depends on the variable conductivity,

$$S_{IJ} = \int_V \mathbf{B}_I^T \mathbf{D} \mathbf{B}_J dV \tag{7.3}$$

through the matrix $\mathbf{D}$. For the most common isotropic case $D_{ij} = k(\mathbf{x})\delta_{ij}$.

The right hand side vector $\mathbf{R}$ contains contributions from internal heating and inhomogeneous boundary conditions,

$$R_I = \int_V N_I \rho H dV + \int_{\Gamma_h} N_I \mathbf{q} \cdot \mathbf{n} dA + \sum_J S_{IJ}^{fp} T_J^p \tag{7.4}$$

where $\mathbf{T}^p$ is the vector of prescribed nodal point values on Γ_g.

A special case of models with variable $k(\mathbf{x})$ is obtained when the thermal conductivity is temperature dependent. [1] This introduces a non-linearity in equation (7.1) that requires special procedures for the numerical solution. In this non-linear case the stiffness matrix depends on the (unknown) temperature $\mathbf{S} = \mathbf{S}[k(T)]$. For such models an iterative solution based on succesive substitution, also known as Picard iteration, is convenient. [2] Starting from a suitable initial vector for the temperature, $\mathbf{T}^{(0)}$, the system of equations (7.2) is built and solved for the first iteration $\mathbf{T}^{(1)}$. This process is iterated by re-building the matrix and solving for an updated temperature field,

$$\mathbf{S}[\mathbf{T}^{(n)}]\mathbf{T}^{(n+1)} = \mathbf{R}^{(n)} \tag{7.5}$$

The iteration proces is stopped when the solution vector has sufficiently converged, i.e. $||\mathbf{T}^{(n+1)} - \mathbf{T}^{(n)}|| < \epsilon$.

Variable thermal conductivity and in particular temperature dependent thermal conductivity can have an impact in geodynamical models, especially in the cold- and hot boundary layers associated with thermal convection in the Earth's mantle, the lithosphere and the core-mantle boundary region. The main mechanism of conductive heat transport in mantle minerals (lattice vibration), has a negative temperature dependence, $dk/dT < 0$ and positive pressure dependence, $dk/dP > 0$, (Hofmeister, 1999). In the presence of an earth-like geotherm the combined effect of such $k(P,T)$ models is the occurrence of a low-conductivity zone at a sub lithospheric depth, similar to the low-viscosity zone related to the Earth's asthenosphere. This $k(P,T)$ model may have had a significant effect on the secular cooling of the Earth. [3]

7.3 Steady state flow in porous media

7.4 Instantaneous viscous flow

Another application area of finite element methods for elliptic problems in geophysics is in flow problems for highly viscous media with *infinite Prandtl number* [4] governed by the Stokes equation. We restrict the discussion here to imcompressible and isoviscous model cases. Extended formulations for the (weakly) compressible case and for variable viscosity can be found in (Schubert et al., 2001). [5]

[1] A.M. Hofmeister, Mantle values of thermal conductivity and the geotherm from phonon lifetimes, *Science*, **283**, 1699-1706, 1999.

[2] van den Berg, A.P., Yuen, D.A. and J.R. Allwardt, Non-linear effects from variable thermal conductivity and mantle internal heating: implications for massive melting and secular cooling of the mantle, *Phys. Earth Planet. Inter.*, **129**, 359-375, 2002.

[3] van den Berg, A.P., Rainey, E.S.G. and D.A. Yuen, The combined influences of variable thermal conductivity, temperature- and pressure-dependent viscosity and core-mantle coupling on thermal evolution, *Phys. Earth Planet. Inter.*, **149**, 259-278, 2005.

[4] The Prandtl number is defined in terms of viscosity η, density ρ and thermal diffusivity κ as $Pr = \frac{\eta}{\rho\kappa}$.

[5] Schubert, G., Turcotte, D.L. and P. Olson, *Mantle convection in the earth and planets*, Cambridge University Press, 2001.

7.4.1 Governing non-dimensional equations

In geodynamical applications the Stokes equation occurs mainly in mantle convection models, coupled to a *convection-diffusion* equation for the temperature. We focus here on the solution of the Stokes flow equation. Details of the coupled problem including energy transport are presented in Chapter 9.

In the Boussinesq approximation for an incompressible model this is formulated in the following equations. Symbols used are explained in Table 7.1.

$$\nabla \cdot \mathbf{u} = 0 \tag{7.6}$$

$$-\nabla \Delta P + \nabla^2 \mathbf{u} = RaT\mathbf{e_z} \tag{7.7}$$

$$\frac{DT}{Dt} = \frac{\partial T}{\partial t} + \mathbf{u} \cdot \nabla T = \nabla^2 T \tag{7.8}$$

where P is the thermodynamic pressure and ΔP is the dynamic pressure, defined by the gradient $\nabla \Delta P = \nabla P - \rho \mathbf{g}$. The equations have been non-dimensionalized using the following scheme,

$$x_i = x_i^{'} h, t = t^{'} h^2/\kappa, u_i = u_i^{'} \kappa/h, \Delta P = \Delta P^{'} \eta \kappa / h^2, T = T_s + T^{'} \Delta T, \eta = \eta^{'} \eta_0 \tag{7.9}$$

Symbol	Definition	Value	Unit
u_j	flow velocity	-	-
τ_{ij}	deviatoric stress tensor	-	-
T	temperature	-	-
ΔT	temperature scale	-	K
h	spatial scale (domain height)	$3 \cdot 10^6$	m
κ	thermal diffusivity	-	-
η_0	viscosity scale value	10^{22}	Pa s
Ra	thermal Rayleigh number	–	–

Table 7.1: *Physical parameters of the convection models*

The primes in (7.9) indicating non-dimensional quantities are dropped and variables will be non-dimensional in the following, unless stated otherwise.

7.4.2 Boundary and initial conditions

Boundary conditions have to be specified for the velocity and temperature field to complete the problem specification. For time dependent problems with $\partial T/\partial t \neq 0$, we also need an initial condition for the temperature field $T(\mathbf{x}, 0) = T_0(\mathbf{x})$. In most modelling experiments with mantle convection we use impermeable free slip boundaries, described by,

$$u_n = \mathbf{u} \cdot \mathbf{n} = 0 \ , \ impermeable \tag{7.10}$$

$$\frac{\partial u_t}{\partial n} = 0 \ , \ free\ slip \tag{7.11}$$

We will consider Rayleigh-Benard convection in a horizontal layer heated from below. This implies that the top surface of the domain is kept at a fixed low temperature $T = T_s$ (Kelvin) and the bottom temperature at $T = T_s + \Delta T$ (Kelvin). We consider a 2-D rectangular domain with zero heat flux (symmetry) conditions $\partial T/\partial n = \partial T/\partial x = 0$ on the vertical boundaries, similar to a horizontal periodic continuation of the domain.

problem 7.1. *Verify that the above implies constant non-dimensional boundary conditions for the temperature of value 0 and 1 for the top and bottom boundary respectively.*

7.4.3 Potential - streamfunction-vorticity formulation

We make use of a potential formulation in terms of streamfunction and vorticity potentials suitable for 2-D applications. This way the (vector) Stokes equation describing viscous flow can be replaced by two coupled Poisson equations for the stream function ψ and the vorticity ω.

The velocity vector field is defined in terms of the curl of the streamfunction vector potential as $\mathbf{u} = -\nabla \times \mathbf{\Psi}$, where $\nabla \cdot \mathbf{\Psi} = 0$.

The potential formulation for the above problem has the following advantages:

- the incompressible character of the solution is implicit in the use of the streamfunction potential with $\nabla \cdot \mathbf{u} = -\nabla \cdot \nabla \times \mathbf{\Psi} = 0$, which reduces the number of coupled equations to be solved by one.
- The number of physical unknowns is reduced because the pressure will be eliminated from the equations.
- For 2-D problems the second order (vector) Stokes equation is replaced by a scalar fourth order PDE, the bi-harmonic equation, for the stream function potential. This equation can be rewritten as two coupled second order Poisson equations for the scalar streamfunction ψ and vorticity ω respectively. The two resulting Poisson equation can be solved numerically using relatively simple numerical methods.

problem 7.2. *The Stokes equation for an isoviscous ($\eta^{'} = 1$) incompressible fluid can be written in vector form as,*

$$-\nabla P + \nabla^2 \mathbf{u} + \mathbf{F} = \mathbf{0} \tag{7.12}$$

where $\mathbf{F}$ *is the volumetric bodyforce vector. Derive from the above equation the bi-harmonic equation,*

$$\nabla^4 \mathbf{\Psi} = \nabla^2 \nabla^2 \mathbf{\Psi} = -\nabla \times \mathbf{F} \tag{7.13}$$

Hint: apply the curl operator ($\nabla\times$) to (7.12) and use the identity for the Laplace operator of a vector field $\mathbf{A}$*,*

$$\nabla^2 \mathbf{A} = \nabla \nabla \cdot \mathbf{A} - \nabla \times \nabla \times \mathbf{A} \tag{7.14}$$

and the property of the Helmholtz vector potential $\nabla \cdot \mathbf{\Psi} = 0$*.*

Using the vorticity, defined as $\mathbf{\Omega} = \nabla \times \mathbf{u}$, it follows that the 4-th order biharmonic equation (7.13) can be split in two second order equations,

$$\nabla^2 \mathbf{\Omega} = -\nabla \times \mathbf{F} \tag{7.15}$$

$$\nabla^2 \mathbf{\Psi} = \mathbf{\Omega} \tag{7.16}$$

In the following we consider rectangular domain configurations with cartesian coordinates (x, y, z) where the z-axis is aligned with gravity (pointing downward). We focuss on 2-D problems as a special case, of the general 3-D case, where all model properties are uniform in the horizontal y direction, $\partial \cdot / \partial y \equiv 0$ and flow is confined to the vertical (x, z) plane.

For the vorticity this implies $\mathbf{\Omega} = (0, \Omega_y, 0) = (0, \omega, 0)$. For the streamfunction and velocity fields we have, $\mathbf{\Psi} = (\Psi_x, \Psi_y, \Psi_z) = (0, \psi, 0)$,

$$\mathbf{u} = -\nabla \times \mathbf{\Psi} = -\begin{vmatrix} i & j & k \\ \partial_x & \partial_y & \partial_z \\ 0 & \psi & 0 \end{vmatrix} = \begin{pmatrix} \partial_z \psi \\ 0 \\ -\partial_x \psi \end{pmatrix} \tag{7.17}$$

For the righthand side of the vorticity equation (7.15) we get, using $\partial_y \cdot = 0$ and $\mathbf{F} = -RaT\mathbf{e}_z$, $F_x = F_y = 0$,

$$\nabla \times \mathbf{F} = \begin{vmatrix} i & j & k \\ \partial_x & \partial_y & \partial_z \\ F_x & F_y & F_z \end{vmatrix} = \begin{pmatrix} \partial_y F_z - \partial_z F_y \\ \partial_z F_x - \partial_x F_z \\ \partial_x F_y - \partial_y F_x \end{pmatrix} = \begin{pmatrix} 0 \\ Ra\partial_x T \\ 0 \end{pmatrix} \tag{7.18}$$

Substitution of (7.18) in (7.15) we get the following coupled set of equations for the 2-D convection model,

$$\nabla^2 \omega = -Ra\partial_x T \tag{7.19}$$

$$\nabla^2 \psi = \omega \tag{7.20}$$

$$\frac{\partial T}{\partial t} + \mathbf{u} \cdot \nabla T = \nabla^2 T \tag{7.21}$$

It can be shown that the instantaneous flow velocity is perpendicular to the gradient of the streamfunction ψ. This implies that an impermeable boundary represents a streamline. In other words $\psi(\mathbf{x}) = c = constant$, $\mathbf{x} \in \partial V$. Since ψ is determined up to a constant we choose $c = 0$ resulting in homogeneous essential (Dirichlet type) boundary conditions for ψ. From the free slip condition $\partial u_t / \partial n = 0$ it follows that $\partial^2 \psi / \partial n^2 = 0$ and since also $\partial^2 \psi / \partial t^2 = 0$ along an instantaneous streamline we have $\nabla^2 \psi = \omega = 0$. So we have identical boundary conditions for streamfunction and vorticity.

Other models such as for flow driven by *kinematic boundary conditions* i.e. prescribed boundary velocity, the formulation of boundary conditions for the vorticity is complicated and will not be discussed here. Such boundary value problems are more easily set up in *primitive variables* with velocity and pressure as dependent variables as discussed in Chapter 11.

Chapter 8

Time dependent problems

8.1 Introduction

In this chapter we introduce solution methods for time dependent problems (initial value problems). The time dependent energy transport equation is taken as an example problem to be solved. The methods we shall describe are applicable for many problems besides the heat transport problem. The discretized heat diffusion equation introduced in Chapter 2 in the context of finite difference methods and in Chapter 5 of finite element methods is,

$$\mathbf{M}\frac{d}{dt}\mathbf{T} + \mathbf{S}\mathbf{T} = \mathbf{F} \tag{8.1}$$

where $\mathbf{M}$ is a heat capacity or mass matrix, $\mathbf{S}$ is a stiffness matrix and $\mathbf{F}$ a righthand side vector containing contributions from internal heating and inhomogeneous boundary conditions. This is a system of first order ordinary differential equations in the unknown vector of time dependent nodal point values,

$$\mathbf{T}(t) = (T_1(t), T_2(t), \ldots, T_N(t))^T \tag{8.2}$$

where the time t is the remaining independent variable after the (semi)dicretization of the dependence of the spatial coordinates. Such a scheme where all but one coordinates are discretised is also known as *'method of lines'*. This system can be solved (integrated in time) for given initial value $\mathbf{T}(0) = \mathbf{T}_0$.

The numerical solution is computed for a set of discrete points in time, $t_n, n = 1, 2, \ldots, M$. Sometimes the time discretiztion will be equidistant with $t_n = t_0 + n \times \Delta t$, Δt a fixed time step. In many situations however it can be desirable to make the time step Δt dependent of the evolution of the solution. In such cases an adaptive time step scheme is used where the value of Δt is made to decrease when the solution vector changes rapidly in time according to some specified criterion and the time step is increased when the evolution decelerates.

The vectors at the discrete integration times are denoted as,

$$\mathbf{T}(t_n) = \mathbf{T}_n,\ \mathbf{F}(t_n) = \mathbf{F}_n,\ n = 0, 1, 2, \ldots, M \tag{8.3}$$

Different integration schemes can be derived if we interpret the system (8.1) as a sequence of initial value problems where, for given $\mathbf{T}_n = \mathbf{T}(t_n)$, the vector $\mathbf{T}_{n+1} = \mathbf{T}(t_{n+1})$ is to be computed.

8.2 Integration methods

Integration of the system (8.1) over the time interval $[t_{n+1-k}, t_{n+1}]$ and assuming $\mathbf{M}$ to be independent of time we get,

$$\mathbf{M}\left(\mathbf{T}_{n+1} - \mathbf{T}_{n+1-k}\right) = \int_{n+1-k}^{n+1} \left(-\mathbf{S}\mathbf{T} + \mathbf{F}\right)\ dt \tag{8.4}$$

The solutionvector $\mathbf{T}$ in the integral is known for the time values $t_{n+1-k}, t_{n+1-k+1}, \ldots, t_n$ and the vector has to be computed for the new time value t_{n+1}.

The integral in the righthand side of (8.4) can be evaluated numerically and expressed in the values of the solution vector for $t_i \leq t_{n+1}$ in the integration scheme,

$$\int_{t_{n+1-k}}^{t_{n+1}} \left(-\mathbf{S}\mathbf{T} + \mathbf{F}\right)\ dt = \sum_{i=n+1-k}^{n+1} w_i \left(-\mathbf{S}_i\mathbf{T}_i + \mathbf{F}_i\right) \tag{8.5}$$

where the w_i are the weights of the quadrature rule used such as, mid-point, trapezoidal or Simpson rule. This scheme is known as a (k) multistep method. Here the solution vectors $\mathbf{T}_{n+1-k}, \mathbf{T}_{n+1-k+1}, ..., \mathbf{T}_n$ must be stored in computer memory during the computation of $\mathbf{T}_{n+1}$.

In the following we will only consider the more common single step methods, with $k = 1$. In this case we have,

$$\mathbf{M}\left(\mathbf{T}_{n+1} - \mathbf{T}_n\right) = \int_{t_n}^{t_{n+1}} \left(-\mathbf{S}\mathbf{T} + \mathbf{F}\right)\ dt \tag{8.6}$$

The integral in the righthand side of (8.6) contains the unknown vector $\mathbf{T}(t)$. Therefore an approximation is substituted for the integrand. Depending on the type of approximation used, different integration schemes are obtained for the ODE (8.1). If $\mathbf{T}_{n+1}$ occurs only in the lefthand side of the resulting expression the method is called explicit. If the righthand side contains $\mathbf{T}_{n+1}$ the method is called implicit. In the latter case a system of algebraic equations involving the stiffness matrix $\mathbf{S}$ must be solved to obtain $\mathbf{T}_{n+1}$. We describe a number of alternative integration schemes in the following.

8.2.1 The Euler forward method

With this method the integrand in (8.6) is approximated by means of forward extrapolation by a constant vector, equal to the value at t_n. This results in the following scheme,

$$\mathbf{M}\mathbf{T}_{n+1} = \mathbf{M}\mathbf{T}_n + \Delta t\left(\mathbf{F}_n - \mathbf{S}_n\mathbf{T}_n\right) \tag{8.7}$$

In case a diagonal *lumped mass matrix* approximation is used as in (5.26), matrix inversion of $\mathbf{M}$ can be done explicitly, resulting in the following form,

$$\mathbf{T}_{n+1} = \left(\mathbf{I} - \Delta t\mathbf{M}^{-1}\mathbf{S}_n\right)\mathbf{T}_n + \Delta t\mathbf{M}^{-1}\mathbf{F}_n \tag{8.8}$$

This is an explicit scheme, meaning that (with a diagonal mass matrix) no system of algebraic equation has to be solved for the computation of $\mathbf{T}_{n+1}$ from $\mathbf{T}_n$, only matrix vector multiplications and vector summations are used. This means that explicit methods require less compute time per integration step than the implicit methods described below. Besides economy in compute time, explicit schemes also have lower memory requirements, because the sparse structure of the matrix can be exploited in compact memory storage schemes where only non-zero matrix elements are stored. However we shall see below that the explicit Euler method has less favorable stability characteristics.

8.2.2 The Euler backward method

In the Euler backward scheme the integrand in (8.6) is approximated by backward extrapolation from t_{n+1}, with a constant value, resulting in,

$$(\mathbf{M} + \Delta t \mathbf{S}_{n+1})\, \mathbf{T}_{n+1} = \mathbf{M}\mathbf{T}_n + \Delta t \mathbf{F}_{n+1} \tag{8.9}$$

This is an implicit formula because in (8.9) $\mathbf{T}_{n+1}$ must be computed by solving a (non-diagonal) system of linear algebraic equations. When the matrix solver is based on a direct (non-iterative) method like Gauss elimination or LU decomposition the matrix must be stored in memory. In practical applications the memory requirement for matrix storage is more than 50 % of the total program memory for models with a substantial number of degrees of freedom ($> O(10^4)$).

problem 8.1. *Explicit methods do not use matrix solvers, only matrix-vector multiplication. Investigate how much memory can be saved by using a compact storage scheme, storing only non-zero diagonals, with the finite difference matrices described in Chapter 3.*

How can a matrix-vector multiplication be formulated in terms of these non-zero diagonals?

Why is it that the matrix can not be stored in compact format when we use a direct matrix solver? In that case a band matrix storage scheme where only elements within the bandwidth are stored will still result in substantial memory savings.

Below we will see that the Euler backward method has better stability characteristics than the Euler forward method.

8.2.3 The Crank-Nicolson method

With the Crank-Nicolson (CN) integration method, the integral in (8.6) is approximated with a trapezoidal integration rule, [1]

$$\int_{t_n}^{t_{n+1}} f(t)\, dt \approx \frac{\Delta t}{2} \left(f_{n+1} + f_n\right) \tag{8.10}$$

This is related to a linear approximation of the function f on the integration interval. Substitution in (8.6) results in,

$$\mathbf{M}\mathbf{T}_{n+1} = \mathbf{M}\mathbf{T}_n + \frac{\Delta t}{2} \left(\mathbf{F}_{n+1} - \mathbf{S}_{n+1}\mathbf{T}_{n+1} + \mathbf{F}_n - \mathbf{S}_n\mathbf{T}_n\right) \tag{8.11}$$

$$\left(\mathbf{M} + \frac{\Delta t}{2}\mathbf{S}_{n+1}\right) \mathbf{T}_{n+1} = \left(\mathbf{M} - \frac{\Delta t}{2}\mathbf{S}_n\right) \mathbf{T}_n + \frac{\Delta t}{2} \left(\mathbf{F}_{n+1} + \mathbf{F}_n\right) \tag{8.12}$$

This is another example of an implicit integration scheme that contains more operations than the Euler implicit scheme. The CN formula represents a more accurate approximation of the original ordinary differential equation than both the Euler methods. In the following we illustrate the CN scheme in an application to the time dependent heat equation.

The Crank-Nicolson scheme is often applied in models of thermal convection in the Earth's mantle where the discretized energy transport equation is of similar form as the equation for time dependent heat conduction problems described in Chapter 3 and 5. Here we illustrate the application of the CN scheme to the time dependent heat conduction equation. Chapter 9 deals with time integration of the coupled equations for thermal convection.

[1] Eric W. Weisstein. "Newton-Cotes Formulas." From MathWorld–A Wolfram Web Resource. http://mathworld.wolfram.com/Newton-CotesFormulas.html

We assume here that the matrices $\mathbf{M}$ and $\mathbf{S}$ are constant in time. The equation for the CN scheme is then given by (8.12) and can be rewritten as,

$$\mathbf{A}\mathbf{T}_{n+1} = \mathbf{B}\mathbf{T}_n + \mathbf{R}_{n+1} \tag{8.13}$$

In the special case of a fixed time step Δt, and with $\mathbf{M}$, $\mathbf{S}$ independent of time, $\mathbf{A}$ and $\mathbf{B}$ are also constant in time and they can be computed once, outside the program loop over integration time steps to reduce the compute time. The same holds for the decomposition of the matrix $\mathbf{A}$, that is the most time consuming part of a direct matrix solver computation. In case of an adaptive time step, $\mathbf{A}$ and $\mathbf{B}$ must be recomputed from the matrices $\mathbf{M}$ and $\mathbf{S}$ each time when the time step is changed.

The implementation of boundary conditions is done in a similar way as for steady state problems. We partition the solution vector in *free* and *prescribed* components, $\mathbf{T} = (\mathbf{T}_f, \mathbf{T}_p)^T$. The corresponding partitioning of the equation (8.13) can then be written as,

$$\begin{pmatrix} \mathbf{A}_{ff} & \mathbf{A}_{fp} \\ \mathbf{A}_{pf} & \mathbf{A}_{pp} \end{pmatrix} \begin{pmatrix} \mathbf{T}_{f\ n+1} \\ \mathbf{T}_{p\ n+1} \end{pmatrix} = \begin{pmatrix} \mathbf{B}_{ff} & \mathbf{B}_{fp} \\ \mathbf{B}_{pf} & \mathbf{B}_{pp} \end{pmatrix} \begin{pmatrix} \mathbf{T}_{f\ n} \\ \mathbf{T}_{p\ n} \end{pmatrix} + \begin{pmatrix} \mathbf{R}_{f\ n+1} \\ \mathbf{R}_{p\ n+1} \end{pmatrix} \tag{8.14}$$

From the first row of (8.14) we obtain,

$$\mathbf{A}_{ff}\mathbf{T}_{f\ n+1} = \mathbf{B}_{ff}\mathbf{T}_{f\ n} + \mathbf{R}_{f\ n+1} + (\mathbf{B}_{fp}\mathbf{T}_{p\ n} - \mathbf{A}_{fp}\mathbf{T}_{p\ n+1}) \tag{8.15}$$

For the special case where the matrices as well as the boundary conditions are constant in time the term in brackets in (8.15) is constant,

$$(\mathbf{B}_{fp} - \mathbf{A}_{fp})\,\mathbf{T}_p = -\Delta t \mathbf{S}_{fp}\mathbf{T}_p \tag{8.16}$$

The CN scheme for this special case becomes,

$$\mathbf{A}_{ff}\mathbf{T}_{f\ n+1} = \mathbf{B}_{ff}\mathbf{T}_{f\ n} + \mathbf{R}_{f\ n+1} - \Delta t \mathbf{S}_{fp}\mathbf{T}_p \tag{8.17}$$

This is for example applied in bottom heated thermal convection models as in Rayleigh-Benard convection, with a prescribed constant bottom boundary temperature.

problem 8.2. *How can the cold top surface in Rayleigh-Benard convection be treated with the above schemes?*

problem 8.3. *The value of* $\mathbf{T}_{n+1}$ *computed with the Euler-forward scheme (8.8) can be used as a* predictor *in a* predictor-corrector *scheme, resulting in what is known as Heun's method* [2]*. Derive an expression for a corrector for* $\mathbf{T}_{n+1}$ *by applying a trapezium-rule integration in terms of* $\mathbf{T}_n$ *and the predictor value, written as* $\mathbf{T}_{n+1}^{pred}$*.*

8.3 Stability and convergence of the integration methods

Sofar we have not discussed the value of the integration time step Δt. A smaller value of the time step will generally result in a more accurate numerical solution. However the required compute time is proportional to the number of time steps in most methods and for a fixed time interval $[t_0, t_{max}]$, the number of time steps increases with decreasing Δt. Therefore an optimum choice of the time step is desirable, combining accuracy and economy of the required computations.

Besides accuracy, stability of the solution is important. The effect of errors due to numerical round-off can sometimes accumulate resulting in a complete loss of the accuracy

[2] `wiki/Heun's_method`.

within a limited number of time steps. In such cases the integration method has become unstable. It appears that explicit integration methods become unstable for time step values above a critical value. In the following some results are given from numerical mathematics regarding stability and convergence of numerical integration methods for ordinary differential equations.

8.3.1 Consistency of the integration scheme

The single step methods described before can be written as,

$$\mathbf{T}_{n+1} - \mathbf{A}\mathbf{T}_n - \mathbf{R}_n = \mathbf{0} \tag{8.18}$$

In the forward Euler scheme we get from (8.8), for constant matrices,

$$\mathbf{A} = \mathbf{I} - \Delta t \mathbf{M}^{-1}\mathbf{S}, \quad \mathbf{R}_n = \Delta t \mathbf{M}^{-1}\mathbf{F}_n \tag{8.19}$$

Note that the *amplification matrix* $\mathbf{A}$ defined here differs from the matrix $\mathbf{A}$ defined in (8.13). The *local truncation error* E is defined by substitution of the analytical solution of the ODE, denoted by the explicit time dependence, in (8.18). The analytical solution does not solve the discrete equation (8.18) exactly, as expressed in the following error term $\mathbf{E}(t_n)$,

$$\mathbf{T}(t_{n+1}) - \mathbf{A}\mathbf{T}(t_n) - \mathbf{R}_n = \Delta t \mathbf{E}(t_n) \tag{8.20}$$

An algorithm with

$$E(t) = ||\mathbf{E}(t)||_\infty \leq c\Delta t^k, \quad t \in [0, t_{max}], \ c \ constant \tag{8.21}$$

is said to be consistent. Where the vectornorm $||\cdot||_\infty$ is further explained in Appendix B. The exponent k is the (convergence) order of the numerical integration scheme. For the forward Euler scheme in (8.11), $k = 1$ as can be shown by a Taylor expansion of the truncation error in $t = t_n$,

$$\begin{aligned}
\Delta t \mathbf{E}(t_n) &= \mathbf{T}(t_n) + \Delta t \mathbf{T}'(t_n) + \frac{\Delta t^2}{2}\mathbf{T}''(\Theta) - \\
&\quad \left(\mathbf{I} - \Delta t \mathbf{M}^{-1}\mathbf{S}\right)\mathbf{T}(t_n) - \Delta t \mathbf{M}^{-1}\mathbf{F}(t_n) \\
&= \Delta t \mathbf{M}^{-1}\left\{\mathbf{M}\mathbf{T}'(t_n) + \mathbf{S}\mathbf{T}(t_n) - \mathbf{F}(t_n)\right\} + \frac{\Delta t^2}{2}\mathbf{T}''(\Theta) \\
&= \frac{\Delta t^2}{2}\mathbf{T}''(\Theta), \quad \Theta \in [t_n, t_{n+1}]
\end{aligned} \tag{8.22}$$

[3] The local truncation error E follows from (8.22) as,

$$\mathbf{E}(t_n) = \frac{\Delta t}{2}\mathbf{T}''(\Theta), \quad E(t_n) = \frac{\Delta t}{2}||\mathbf{T}''(\Theta)||_\infty \tag{8.23}$$

problem 8.4. *Derive a similar result for the backward Euler scheme.*
Hint: expand $\mathbf{T}(t_n)$ *in* t_{n+1}.

problem 8.5. *Derive for the Crank-Nicolson scheme,* $k = 2$. *Hint: expand the time dependent vectors* $\mathbf{T}(t_{n+1})$, $\mathbf{T}(t_n)$, $\mathbf{F}(t_{n+1})$, $\mathbf{F}(t_n)$ *in the midpoint* $t_{n+1/2} = t_n + \Delta t/2$.

[3] In the derivation of (8.22) the expression in curly brackets equals zero because it represents the residue of the analytical solution substituted in the ODE.

8.3.2 Stability of the integration scheme

Stability of the integration scheme is determined by the amplification matrix $\mathbf{A}$ in (8.18). We consider the effect of a round-off error δ in the initial value of the homogeneous equation and set the term $\mathbf{R}_n = 0$.

$$\begin{aligned}
\mathbf{T}_1 &= \mathbf{A}(\mathbf{T}_0 + \delta) = \mathbf{A}\mathbf{T}_0 + \mathbf{A}\delta \\
\mathbf{T}_2 &= \mathbf{A}^2\mathbf{T}_0 + \mathbf{A}^2\delta \\
\mathbf{T}_{n+1} &= \mathbf{A}^{n+1}\mathbf{T}_0 + \mathbf{A}^{n+1}\delta
\end{aligned} \tag{8.24}$$

The error in the solution at t_{n+1} due to the perturbation δ of the initial value is,

$$\mathbf{e}_{n+1} = \mathbf{T}_{n+1} - \mathbf{A}^{n+1}\mathbf{T}_0 = \mathbf{A}^{n+1}\delta \tag{8.25}$$

A condition for stability of the integration scheme is that this error is bounded,

$$||\mathbf{e}_{n+1}|| = ||\mathbf{A}^{n+1}\delta|| \leq ||\mathbf{A}||^{n+1}||\delta|| < M, \quad for\ all\ n \tag{8.26}$$

Here $||\mathbf{A}||$ is the matrix norm associated with the vector norm used. In Appendix B it is shown that this norm is related to the largest eigenvalue of the matrix. (8.26) expresses the stability requirement for the amplification matrix,

$$||\mathbf{A}|| \leq 1 \tag{8.27}$$

This condition is satisfied if the largest eigenvalue of $\mathbf{A}$, say λ_m is smaller than 1. If we apply this to the forward Euler scheme we obtain, with $\mathbf{F} = \mathbf{0}$,

$$\mathbf{T}_{n+1} = \left(\mathbf{I} - \Delta t\mathbf{M}^{-1}\mathbf{S}\right)\mathbf{T}_n \tag{8.28}$$

We find the following stability condition for the time step,

$$\Delta t < 2\mu_m^{-1} \tag{8.29}$$

where μ_m is the largest eigenvalue of the matrix $\mathbf{M}^{-1}\mathbf{S}$.

problem 8.6. *Derive (8.29) from (8.28).*

The forward Euler method is conditionally stable. It can be shown that for the discrete heat equation $\mu_m = O(h^{-2})$, where h is a characteristic grid spacing. This implies that if we increase the grid resolution, i.e. decrease h, the stability limit for the time step (8.29) decreases as well (quadraticly).

problem 8.7. *Show that the implicit (backward) Euler scheme is unconditionally stable.*

problem 8.8. *Investigate the stability character for both the Euler schemes for the special case of a scalar differential equation,*

$$M\frac{dT}{dt} + ST = F \tag{8.30}$$

where T, M, S, F are positive scalar quantities.

8.3.3 Convergence of the integration scheme

It can be shown that a stable and consistent integration scheme is convergent. Here convergence means that the difference between the numerical solution and the analytical solution of the ODE can be made arbitrarily small by choosing a sufficiently small time step.

For the general integration scheme (8.18) this can be shown as follows,

$$\mathbf{T}_{n+1} - \mathbf{A}\mathbf{T}_n - \mathbf{R}_n = \mathbf{0} \tag{8.31}$$

Substitution of the exact solution of the ODE gives,

$$\mathbf{T}(t_{n+1}) - \mathbf{A}\mathbf{T}(t_n) - \mathbf{R}(t_n) = \Delta t \mathbf{E}(t_n) \tag{8.32}$$

The global error in the numerical solution is defined as,

$$\mathbf{e}_n = \mathbf{T}_n - \mathbf{T}(t_n) \tag{8.33}$$

Taking the difference of (8.31) and (8.32) we have,

$$\mathbf{e}_{n+1} = \mathbf{A}\mathbf{e}_n - \Delta t \mathbf{E}(t_n) \tag{8.34}$$

Backward recursion of (8.34) gives,

$$\mathbf{e}_n = \mathbf{A}\mathbf{e}_{n-1} - \Delta t \mathbf{E}(t_{n-1}) \tag{8.35}$$

$$\mathbf{e}_{n+1} = \mathbf{A}^2\mathbf{e}_{n-1} - \Delta t \mathbf{A}\mathbf{E}(t_{n-1}) - \Delta t \mathbf{E}(t_n) \tag{8.36}$$

$$\mathbf{e}_{n+1} = \mathbf{A}^{n+1}\mathbf{e}_0 - \Delta t \sum_{i=0}^{n} \mathbf{A}^i \mathbf{E}(t_{n-i}) \tag{8.37}$$

with an exact initial value we have $\mathbf{e}_0 = \mathbf{0}$ and we can express the norm of the vector $\mathbf{e}_n$, using an associated matrixnorm $||\mathbf{A}||$ of the amplification matrix $\mathbf{A}$ described in more detail in Appendix B,

$$\begin{aligned}
&||\mathbf{e}_n|| = \Delta t || \sum_{i=0}^{n-1} \mathbf{A}^i \mathbf{E}(t_{n-1-i})|| \leq \Delta t \sum_{i=0}^{n-1} ||\mathbf{A}||^i ||\mathbf{E}(t_{n-1-i})|| \\
&\leq \Delta t \sum_{i=0}^{n-1} ||\mathbf{E}(t_{n-1-i})||, \quad (\textit{stability}, \ ||\mathbf{A}|| \leq 1) \\
&\leq t_n \max_{t\in[t_0,t_{max}]} ||\mathbf{E}(t)||, \ t_n = \sum_{i=0}^{n-1} \Delta t \\
&\leq t_n c \Delta t^k, \ (\textit{consistency})
\end{aligned} \tag{8.38}$$

For given t_n we have $\lim_{\Delta t \to 0} ||\mathbf{e}(t_n)|| = 0$. This shows that a stable and consistent scheme is also convergent. Note that the rate of convergence increases with k.

Chapter 9

Systems of coupled equations

In the previous chapters we have considered models described by a single partial differential equation. In this chapter we deal with models consisting of several partial differential equations describing coupled transport processes. As main examples of such physical processes we take models of thermal convection in a layer heated from below *(Rayleigh-Benard convection)* (R-B). We consider convection in fluid saturated porous media as well as in a purely viscous fluid layer. The latter type plays an important role in geodynamical models. This chapter deals with R-B convection exclusively and we investigate separately steady state convection and time dependent convection problems.

9.1 Model equations for Rayleigh-Benard convection

9.1.1 Thermal convection in a porous medium

Rayleigh-Benard convection in a closed, fluid saturated porous medium is described by a poisson equation for the dynamic pressure driving fluid flow, coupled to a convection diffusion equation describing heat transport.

We start from the Darcy equation for flow in porous media (Turcotte and Schubert, 2001),

$$\mathbf{q} = -\frac{k}{\mu}\left(\nabla p - \rho\mathbf{g}\right) = -\frac{k}{\mu}\left\{\nabla(p_r + \Delta p) - (\rho_r + \Delta\rho)g\mathbf{e}_z\right\} = -\frac{k}{\mu}\left(\nabla\Delta p - \Delta\rho g\mathbf{e}_z\right) \quad (9.1)$$

Where $\mathbf{q}$ is the Darcy flow velocity, p the total fluid pressure, p_r a hydrostatic reference pressure with $\nabla p_r = \rho_r\mathbf{g} = \rho_r g\mathbf{e}_z$ and $\Delta p = p - p_r$ a dynamic pressure. The gradient of Δp together with a buoyancy volume force $\Delta\rho g\mathbf{e}_z$, related to density variation, are the driving forces. In the context of thermal convection problems the density variation is expressed in the temperature variation in a linearized equation of state,

$$\rho = \rho_r(1 - \alpha(T - T_s)) \quad (9.2)$$

A scalar source/sink term can be introduced as the divergence of the Darcy flow field, $s = \nabla\cdot\mathbf{q}$. This term is relevant for example in configurations where fluid is pumped in/out of the medium. In the problem context considered here, with a natural (free) convection problem, the domain is closed (with impermable boundaries) and the source term is set to zero.

Non-dimensional quantities are introduced that will transform the coupled equations for thermal convection to a one-parameter model with the Rayleigh number as a control parameter. This is done with the following scheme: for spatial coordinates and temperature, $\mathbf{x} = h\mathbf{x}'$, $T = T_s + T'\Delta T$ and the non-dimensional temperature perturbation is written as

$\theta = T' = (T - T_s)/\Delta T$. h is the depth of the domain, ΔT the fixed temperature contrast across the layer and T_s the constant surface temperature.

After non-dimensionalization and introducing a convection-diffusion equation describing heat transport, the following coupled equations are obtained, including a fluid source term s,

$$-\nabla^2 \Delta p = Ra \partial_z \theta + s = -Ra \partial_y \theta + s \tag{9.3}$$

$$\frac{\partial \theta}{\partial t} + \mathbf{q} \cdot \nabla \theta - \nabla^2 \theta = 0 \tag{9.4}$$

Material properties are assumed uniform here and equal to their corresponding scale value denoted by a zero subscript in the following. The pressure scale is expressed in the scale values for thermal diffusivity, κ_0, fluid viscosity μ_0 and permeability of the porous medium k_0, as $p_0 = \kappa_0 \mu_0 / k_0$. The source term defined as the divergence of the Darcy flow field has the dimension of inverse time with the scale value $s_0 = 1/t_0 = \kappa_0 / h^2$.

From these definitions the non-dimensional Rayleigh number follows as,

$$Ra = \frac{\rho_0 g_0 \alpha_0 k_0 h \Delta T}{\mu_0 \kappa_0} \tag{9.5}$$

This differs slightly from the definition for the classical convection problem in a viscous medium that contains h^3 in stead of $k_0 h$ (note that the permeability scale k_0 has dimension m^2).

The constitutional equation for the Darcy flux (9.1) is non-dimensionalized as follows,

$$\begin{aligned} \mathbf{q} &= q_0 \mathbf{q}' = -\frac{k}{\mu} \left(\nabla \Delta p + \rho_0 \alpha_0 \Delta T g_0 \theta \mathbf{e}_z \right) = -\frac{k p_0}{\mu h} \left(\nabla' \Delta p' + \frac{h}{p_0} \rho_0 \alpha_0 \Delta T g_0 \theta \mathbf{e}_z \right) \\ &= -\frac{k p_0}{\mu h} \left(\nabla' \Delta p' + \frac{h k_0}{\kappa_0 \mu_0} \rho_0 \alpha_0 \Delta T g_0 \theta \mathbf{e}_z \right) = -\frac{k p_0}{\mu h} \left(\nabla' \Delta p' + Ra \theta \mathbf{e}_z \right) \\ \mathbf{q} &= -\frac{\kappa_0}{h} \mathbf{q}' \end{aligned} \tag{9.6}$$

where $q_0 = \kappa_0 / h$ is identified as the proper scale value. For the non-dimensional Darcy flux we get - dropping the primes for non-dimensional quantities,

$$\mathbf{q} = -\nabla \Delta p - Ra \theta \mathbf{e}_z = -\nabla \Delta p + Ra \theta \mathbf{e}_y \tag{9.7}$$

Boundary conditions

In the following we consider a natural convection problem for a rectangular 2-D domain, with the source term in (9.3) set to $s = 0$. Dimensional temperature at the top surface (depth $z = 0$) is T_s, at the bottom surface (depth $z = h$) it is $T_s + \Delta T$. The corresponding non-dimensional temperature at non-dim. depth $z = 0, 1$ is $T(z = 0) = 0$ and $T(z = 1) = 1$. At the vertical boundaries symmetry conditions apply $\partial_x T = 0$.

For the flow/pressure equation a zero fluid flux condition applies for all boundaries. This can be converted into a corresponding condition for the pressure perturbation (dynamic pressure) Δp that occurs as the unknown field in the flow equation (9.3) as follows,

$$q_n = \mathbf{q} \cdot \mathbf{n} = -\frac{k}{\mu} (\nabla p \cdot \mathbf{n} - \rho \mathbf{g} \cdot \mathbf{n}) = 0 \tag{9.8}$$

which gives for the total pressure gradient,

$$\nabla p \cdot \mathbf{n} = \nabla (p_r + \Delta p) \cdot \mathbf{n} = \rho \mathbf{g} \cdot \mathbf{n} = (\rho_r + \Delta \rho) g \mathbf{e}_z \cdot \mathbf{n} \tag{9.9}$$

and, in terms of the dynamic pressure gradient,

$$\nabla \Delta p \cdot \mathbf{n} = \Delta \rho g \mathbf{e}_z \cdot \mathbf{n} \tag{9.10}$$

This implies homogeneous natural boundary conditions on the vertical boundaries ($\mathbf{e}_z \cdot \mathbf{n} = 0$). Since $\Delta\rho = -\rho_0 \alpha (T - T_s)$ we also have hom.nat. b.c. at the top surface. At the bottom surface the inhomogeneous nat.b.c. is,

$$\nabla \Delta p \cdot \mathbf{n} = \Delta \rho g = -\rho_0 \alpha g (T_s + \Delta T - T_s) = -\rho_0 \alpha g \Delta T \tag{9.11}$$

In its non-dimensional form this inhomogeneous natural boundary condition can be expressed in the Rayleigh number. At the bottom boundary we have,

$$\nabla \Delta p \cdot \mathbf{n} = \frac{\partial \Delta p}{\partial n} = -\frac{\partial \Delta p}{\partial y} = -\frac{\kappa_0 \mu_0}{k_0 h} \frac{\partial \Delta p'}{\partial y'} \tag{9.12}$$

From the right hand side of (9.11) we get

$$-\frac{\kappa_0 \mu_0}{k_0 h} \frac{\partial \Delta p'}{\partial y'} = -\rho_0 \alpha g \Delta T \Rightarrow \frac{\partial \Delta p'}{\partial y'} = \frac{k_0 h \rho_0 \alpha g \Delta T}{\kappa_0 \mu_0} = Ra \Rightarrow \frac{\partial \Delta p'}{\partial n} = -Ra \tag{9.13}$$

[1]

9.1.2 Thermal convection in a viscous fluid layer

In Chapter 7 viscous flow models were presented for the simple case of an isoviscous fluid using a streamfunction-vorticity formulation that resulted in coupled Poisson-type potential equations for the streamfunction, $\psi(\mathbf{x})$, and vorticity, $\omega(\mathbf{x})$, potentials. In this formulation the fluid flow velocity vector, $\mathbf{u}$, is computed from the scalar streamfunction potential as a derived quantity $\mathbf{u} = (u_x, u_z) = (\partial_z \psi, -\partial_x \psi)$. [2]

The streamfunction-vorticity method is particularly suitable for isoviscous problems with free slip-impermeable boundaries. A more general formulation suitable for more complex rheology and types of boundary conditions is given in Chapter 11.

Here we consider similar isoviscous models with free slip-impermeable boundaries as in Chapter 7 as part of the classical R-B convection model.

In the Boussinesq approximation, followed here, the energy transport equation takes the form of a convection-diffusion equation. In non-dimensional form this becomes,

$$\frac{\partial T}{\partial t} = \nabla^2 T - \mathbf{u} \cdot \nabla T + Q \tag{9.15}$$

where Q is the non-dimensional internal heating rate. The time variable occurs explicitly in the energy equation. The time dependence of the flow equations given in Chapter 7 is implicit through the temperature dependence of the buoyancy forces and time dependence of the temperature field.

In the R-B convection model essential boundary conditions are prescribed for the temperature on the horizontal top and bottom boundaries of the convecting layer, with zero and unit non-dimensional temperature on the top and bottom boundary respectively. On the vertical boundaries the heatflow density is set to zero, implying a zero normal temperature derivative, a homogeneous natural boundary condition corresponding to a symmetric horizontal continuation of the domain.

[1] Apparantly we have natural boundary conditions on the complete boundary which poses a problem, because a compatibility condition must be satisfied with the right hand side of the pressure equation (9.3), (related to the fact that Δp is determined up to a harmonic function in (9.3),

$$\int_{\partial\Omega} \nabla \Delta p \cdot \mathbf{n} dS = \int_{\Omega} -Ra \partial_z \theta dV \tag{9.14}$$

[2] Note the similarity with the method used in section 9.1.1 where the Darcy flow vector is computed from the gradient of the dynamic pressure.

9.2 Discretization of the governing equations

In the following we present methods for the solution of the coupled equations, for both types of thermal convection of section 9.1. Steady-state convection is treated as a separate case. Several methods are given for time integration of the coupled equations for the time dependent case.

For both types of convection problem introduced in section 9.1 a flow velocity field is obtained by solving similar, potential type equations followed by computation of the velocity field as a derived quantity from the scalar potentials. In the following we use a single notation, $\mathbf{u}(\mathbf{x})$, for the flow velocity vector field for both types of convection problems.

The heat transport equation (9.15), without the convective term $\mathbf{u} \cdot \nabla T$, has been discretized with the finite element method in Chapter 5 and 6. Equation (9.15) is a convection/diffusion equation and the convective term results in an extra contribution to the stiffness matrix.

problem 9.1. *Derive for the stiffness matrix of the convection/diffusion equation (9.15) with the Galerkin method,*

$$A_{IJ} = \int_V (\nabla N_I \cdot \nabla N_J + N_I \mathbf{u} \cdot \nabla N_J) \; dV \tag{9.16}$$

From (9.16) it follows that the stiffness matrix is no longer symmetric for non-zero convection velocity $\mathbf{u}$. In practice it appears that the numerical solution of (9.15) becomes unstable, for a given velocity field, when the finite element mesh is not of sufficiently high resolution. Besides mesh refinement one can also use finite element implementations of so called upwind techniques, developed originally for finite difference methods, to supress the forementioned instabilities. In these methods the test functions $w_I(\mathbf{x})$ in the Galerkin method are no longer identified with the basis functions $N_J(\mathbf{x})$ (Bubnov-Galerkin). Instead testfunctions are used which are adapted for the local flow velocity $\mathbf{u}$, (Petrov-Galerkin). For a more complete discussion of upwind techniques we refer to the literature. [3]

In the following we shall assume that the finite element mesh is of sufficient resolution to avoid instabillities due to the advective term in the transport equation. We shall first describe a solution for steady state convection.

9.3 Steady state convection

In steady state convection models the time derivative in (9.15) vanishes and the model equations now consist of one (porous media case) or two (viscous case) potential equations coupled with the steady-state convection-diffusion equation,

$$\nabla^2 T - \mathbf{u} \cdot \nabla T + Q = 0 \tag{9.17}$$

Discretization of the PDE's with the finite element method results in the following coupled set of algebraic equations, [4]

$$\mathbf{B\Phi} = \mathbf{F}(\mathbf{T}) \tag{9.18}$$

$$\mathbf{A}(\mathbf{U}(\mathbf{T}))\mathbf{T} = \mathbf{R} \tag{9.19}$$

[3] C. Cuvelier, A. Segal, A.A. van Steenhoven, 1986. Finite element methods and Navier-Stokes Equations, Reidel, Dordrecht.

S.V. Patankar, 1980. Numerical heat transfer and fluid flow, McGraw-Hill

[4] We use a single notation for the (coupled) potential equation(s) for either the dynamic pressure (porous media case) or the streamfunction-vorticity (viscous case).

problem 9.2. *Give expressions for (9.18) for cases of porous media convection and for a viscous medium.*

The equations (9.18) and (9.19) are coupled through the righthand side vector and through the advection velocity $\mathbf{u}$ in the expression for the stiffness matrix of the discrete heat equation (9.16). The coupling through the advection velocity introduces a non-linearity because the velocity depends on the temperature. Steady state solutions of the non-linear system can be computed iteratively by Picard iteration (succesive substitution) using the following algorithm,

1. Compute the velocity for a given temperature field $\mathbf{T}^{(n)}$ by solving, [5]

$$\mathbf{B\Phi}^{(n+1)} = \mathbf{F}(\mathbf{T}^{(n)}), \mathbf{U}^{(n+1)} = \mathbf{D\Phi}^{(n+1)} \tag{9.20}$$

2. Compute the temperature from,

$$\mathbf{A}(\mathbf{U}^{(n+1)})\mathbf{T}^{(n+1)} = \mathbf{R} \tag{9.21}$$

3. Perform a convergence test,

$$\epsilon_{\mathbf{U}} = \frac{||\mathbf{U}^{(n+1)} - \mathbf{U}^{(n)}||}{||\mathbf{U}^{(n+1)}||}, \quad \epsilon_T = \frac{||\mathbf{T}^{(n+1)} - \mathbf{T}^{(n)}||}{||\mathbf{T}^{(n+1)}||} \tag{9.22}$$

4. If $\max(\epsilon_{\mathbf{U}}, \epsilon_T) > \epsilon$ for a predefined accuracy ϵ, continue with step 1. If $\max(\epsilon_{\mathbf{U}}, \epsilon_T) \leq \epsilon$ convergence is reached and $\mathbf{U}^{(n+1)}, \mathbf{T}^{(n+1)}$ are taken as the final solution.

In applications under-relaxation may be necessary to maintain stability of the iteration proces. In that case the solution vector computed in (9.20) and (9.21) are taken as interim results $\mathbf{U}^*$, $\mathbf{T}^*$ and the next iterand is computed from,

$$\mathbf{\Phi}^{(n+1)} = \beta\mathbf{\Phi}^* + (1-\beta)\mathbf{\Phi}^{(n)}, \quad \mathbf{T}^{(n+1)} = \beta\mathbf{T}^* + (1-\beta)\mathbf{T}^{(n)} \tag{9.23}$$

with a relaxation factor $0 < \beta < 1$.

9.4 Time dependent convection

For time dependent convection problems discretization of the model equations leads to the following set of equations,

$$\mathbf{M}\frac{d}{dt}\mathbf{T} + \mathbf{A}(\mathbf{U}(\mathbf{T}))\mathbf{T} = \mathbf{R} \tag{9.24}$$

$$\mathbf{B\Phi} = \mathbf{F}(\mathbf{T}), \mathbf{U} = \mathbf{D\Phi} \tag{9.25}$$

The integration methods for systems of ordinary differential equations described in Chapter 8 can be applied to solve this system for given intial values. However due to the coupling in the advective term not all of these methods can be applied directly.

[5] $\mathbf{D\Phi}$ represents the operation for computing the flow velocity from either the dynamic pressure (porous media) or the streamfunction (viscous flow).

9.4.1 An explicit integration method

Application of the forward Euler method to the coupled system (9.24),(9.25) leads to the following 2-stage algorithm,

1. for given temperature and velocity vectors $\mathbf{T}_n$ and $\mathbf{U}_n$ compute $\mathbf{T}_{n+1}$,

$$\mathbf{T}_{n+1} = \left(\mathbf{I} - \Delta t \mathbf{M}^{-1}\mathbf{A}(\mathbf{U}_n)\right)\mathbf{T}_n + \Delta t \mathbf{M}^{-1}\mathbf{R}_n \tag{9.26}$$

2. compute the velocity field $\mathbf{U}_{n+1}$ for $t = t_{n+1}$ using $\mathbf{T}_{n+1}$ and (9.25).

As for the case of the static (purely conductive) heat equation it can be shown that the Euler forward algorithm is conditionally stable and a commonly applied adaptive timestep criterion is,

$$\Delta t < \min_{K \in \{1,\ldots,N_{elm}\}} \left(\frac{h_K}{|\mathbf{u}_K|}\right) \tag{9.27}$$

where h is a characteristic local element size, $|\mathbf{u}|$ is the local magnitude of the flow velocity field and N_{elm} is the number of elements. This is the so called Courant-Friedrichs-Levy (CFL) criterion. In practice the explicit scheme is applied with variable time step $\Delta t_n = \alpha \Delta t_{CFL}$ with $\alpha < 0.02 - 0.10$.

9.4.2 An implicit method (predictor-corrector)

Because of the more favorable stability properties implicit methods are often preferred to explicit ones. In the geophysical literature on mantle convection implicit methods are commonly used in a so called predicitor-corrector scheme (Christensen, 1984, Hansen and Ebel, 1988, van den Berg et al., 1993), in the following way,

1. For given temperature and velocity vectors $\mathbf{T}_n, \mathbf{U}_n$, compute an approximate prediction for the temperature for $t = t_{n+1}$, T^*_{n+1} with an implicit Euler method (see Chapter 8),

$$\mathbf{M}\Delta t^{-1}\left(T^*_{n+1} - T_n\right) + \mathbf{A}(\mathbf{U}_n)T^*_{n+1} = \mathbf{R}_{n+1} \tag{9.28}$$

   where the stiffness matrix has been approximated by substitution of $\mathbf{U}_n$ instead of $\mathbf{U}_{n+1}$ in the implicit Euler scheme.

2. Compute the corresponding prediction for the velocity $\mathbf{U}^*_{n+1}$ from T^*_{n+1} by solving (9.25).

3. Compute a corrected temperature for $t = t_{n+1}$ using the predicted velocity vector and a Crank-Nicolson (corrector) step,

$$\mathbf{M}\Delta t^{-1}\left(\mathbf{T}_{n+1} - \mathbf{T}_n\right) + \frac{1}{2}\mathbf{A}(\mathbf{U}^*_{n+1})\mathbf{T}_{n+1} + \frac{1}{2}\mathbf{A}(\mathbf{U}_n)\mathbf{T}_n = \frac{1}{2}\left(\mathbf{R}_{n+1} + \mathbf{R}_n\right) \tag{9.29}$$

4. Compute a corrected velocity field U_{n+1} for t_{n+1} with the corrected temperature vector $\mathbf{T}_{n+1}$ by solving (9.25).

The corrector steps can be performed repeatedly in an iterative way, in a predictor-multi-corrector scheme, until the solution vectors converge. This offers the possibility to iterate for non-linear models, such as models with temperature dependent thermal conductivity (van den Berg et al., 2002). [6] The rate of convergence of such schemes strongly depends on the value of the time step Δt. The time step is therefore usually chosen small enough such that a single corrector step provides sufficient accuracy. Because of the higher accuracy of the predictor-corrector (PC) scheme (higher order truncation error of the CN scheme $E_{CN} = c\Delta t^2$) and the stability of the implicit integration methods with respect to the time step value it appears that the (PC) scheme allows a larger time step value compared to the explicit Euler method, $\Delta t = \alpha \Delta t_{CFL}$ with $\alpha \approx 0.5 - 1.0$.

[6] A.P. van den Berg, D.A. Yuen, J.R. Allwardt, Non-linear effects from variable thermal conductivity and mantle internal heating: implications for massive melting and secular cooling of the mantle, *Physics of the Earth and Planetary Interiors*, 129, 359-375, 2002.

Chapter 10

Finite element methods for elastic deformation problems

In the previous chapters we have considered scalar problems. Here we introduce solution methods for elastic deformation problems where the unknown field is the displacement vector field. Such problems occur for instance in geophysical modelling experiments for the elastic part of the lithosphere. Observed flexure of the lithosphere indicates an effective elastic thickness of several tens of kilomters (Turcotte and Schubert, 2002). In seismology these vector problems occur associated with elastic wavefields described by the elastodynamic equation,

$$\rho \frac{d^2 u_i}{dt^2} = \partial_j \sigma_{ij} + \rho F_i \tag{10.1}$$

where u_i is the displacement field, σ_{ij} is the elastic stress tensor and ρF_i is the volumetric bodyforce. In (10.1) and in the following we apply the Einstein summation convention where summation is implied over repeated (small) indices. In the following we shall only consider elastostatic problems where the inertial term in (10.1) is neglected. For geodynamical problems, on a geological timescale this is a good approximation. The finite element methods derived for the static case can be extended for time dependent problems, such as in seismic wave field modelling, by numerical integration methods for ordinary differentialequations. In spectral methods (10.1) is first transformed into an elliptic equation in the frequency domain, similar to the Helmholtz equation, by means of a Fourier transformation (Marfurt, 1984, van den Berg, 1984-88).

The elastostatic equation solved in the following is given by,

$$\partial_j \sigma_{ij} + \rho F_i = 0 \tag{10.2}$$

problem 10.1. *Some time dependent geodynamical problems may involve the elastostatic equation (10.2). Describe examples of deformation processes on a geological timescale where the time dependence enters through time dependent boundary conditions or the body force term.*

10.1 Boundary conditions for elastic problems

The elastostatic equation (10.2) represents a set of differential equations of the order $2 \times m$, where m is the dimension of the problem i.e. the number of components of the elastic displacement field. In general m boundary conditions are required to specify a unique solution. We consider in particular the following types of boundary conditions,

$$u_i(\mathbf{x}) = g_i(\mathbf{x}),\ \mathbf{x} \in \Gamma_{g_i},\ i = 1, \ldots, m \tag{10.3}$$

$$\sigma_{ij} n_j(\mathbf{x}) = h_i(\mathbf{x}),\ \mathbf{x} \in \Gamma_{h_i},\ i = 1, \ldots, m \tag{10.4}$$

$$\sigma_{ij} n_j(\mathbf{x}) + \alpha_{ij} u_j(\mathbf{x}) = r_i(\mathbf{x}),\ \mathbf{x} \in \Gamma_{r_i},\ i = 1, \ldots, m \tag{10.5}$$

With condition (10.3) vector components of the displacement field are prescribed, an example of an essential boundary condition. Condition (10.4) specifies component i of the traction vector $\mathbf{t}$, $t_i = \sigma_{ij} n_j$, where n_j are the components of the outward pointing normal vectorfield on the boundary. (10.5) is a mixed condition with a linear combination of the local displacement field and the traction field. (10.4) follows as a special case of (10.5) with $\alpha_{ij} = 0$, (10.4) and (10.5) are natural boundary conditions, (10.5) is sometimes referred to as a Robin type condition. This mixed boundary condition can be used in a model where elastic resistive forces act on parts of the boundary. This occurs when modelling deformation of elastic structures floating in a fluid medium. A geophysical application of this is found in models of the isostatic deformation of an elastic lithosphere subject to resistive forces related to the hydrostatic pressure along the bottom interface between the lithosphere and the fluid substratum representing the viscous mantle.

problem 10.2. *Derive a compatibility condition for a case with natural boundary conditions of type (10.4) on the complete boundary surface of the domain.*

Boundary conditions can be used in combinations of the different types in different directions. We limit ourselves here to 2-D configurations where boundary conditions are specified in the direction of the normal vector and tangential vector along the boundary. We further assume that,

$$\Gamma = \partial V = \Gamma_{g_i} \cup \Gamma_{h_i} \cup \Gamma_{r_i},\ i = 1, 2 \tag{10.6}$$

$$\Gamma_{g_i} \cap \Gamma_{h_i} = \Gamma_{g_i} \cap \Gamma_{r_i} = \Gamma_{h_i} \cap \Gamma_{r_i} = \emptyset,\ i = 1, 2 \tag{10.7}$$

[1] This implies that in every point of the 2-D boundary curve Γ two conditions are specified ($i = 1, 2$) and that these conditions refer to different vector components of the displacement or traction fields. This has the following consequences,

1. If both components of the displacement are prescribed in a boundary point $\mathbf{x}$, i.e. $\mathbf{x} \in \Gamma_{g_1} \cap \Gamma_{g_2}$, the traction components in $\mathbf{x}$ are free parameters that can not be prescribed.

2. The opposite also holds: if both the traction components are prescribed both displacement components are free parameters. This occurs in cases with a stress-free (zero-traction) boundary like the Earth's free surface in an elastic model for lithospheric flexure,

   $$t_i = \sigma_{ij} n_j = 0,\ i = 1, 2 \tag{10.9}$$

   both the traction components are prescribed so the displacement components are free parameters.

[1] Note that (10.7) implies,

$$\Gamma_{g_i} \neq \emptyset \leftrightarrow \Gamma_{h_i} = \emptyset \wedge \Gamma_{r_i} = \emptyset \tag{10.8}$$

3. If a traction component is prescribed together with a displacement component, the two must be specified in different component directions. An example of this is a boundary with zero normal displacement and zero tangential traction - a free slip condition - with $u_n = u_j n_j = 0$, $\mathbf{t}_s = 0$, where the subscript s refers to the tangential unit vector $\mathbf{s}$ with $\mathbf{n} \cdot \mathbf{s} = 0$. [2]

10.2 Interpolation of vector fields on a grid of nodal points

In the finite element solution of vector problems piecewise interpolation of the unknowns in terms of the nodal point values is used. In previous chapters we have used the expression,

$$u(\mathbf{x}) \approx u^h(\mathbf{x}) = \sum_{J=1}^{N} N_J(\mathbf{x}) u(\mathbf{x}_J) = \sum_{J=1}^{N} N_J(\mathbf{x}) U_J \tag{10.10}$$

for scalar fields $u(\mathbf{x})$. The nodal point values of the unknown scalar represent the unknown vector $\mathbf{U} \in \mathbb{R}^N$ of the discretized problem where N is the number of nodal points. We shall further drop the superscript h in the approximated field. The difference will be clear from the context.

For a vector field $\mathbf{u}(\mathbf{x}) \in \mathbb{R}^m$, $m = 2, 3$ we can generalize the basis function expansion (10.10) by applying the interpolation component wise. We shall further assume displacement vector fields in $\mathbb{R}^2$.

$$\mathbf{u}(\mathbf{x}) = (u_1(\mathbf{x}), u_2(\mathbf{x}))^T \approx \begin{pmatrix} \sum_{J=1}^{N} N_J(\mathbf{x}) U_{1J} \\ \sum_{J=1}^{N} N_J(\mathbf{x}) U_{2J} \end{pmatrix} \tag{10.11}$$

Where the component wise N-vectors are defined as,

$$\mathbf{U}_i = (u_i(\mathbf{x}_1), u_i(\mathbf{x}_2), \ldots u_i(\mathbf{x}_N))^T \in \mathbb{R}^N, \ i = 1, 2 \tag{10.12}$$

Next we arrange the components of the two N-vectors $\mathbf{U}_1$ and $\mathbf{U}_2$ in a single $2N$-vector,

$$\mathbf{U} = (u_1(\mathbf{x}_1), u_2(\mathbf{x}_1), \ldots u_1(\mathbf{x}_N), u_2(\mathbf{x}_N))^T \in \mathbb{R}^{2N} \tag{10.13}$$

The interpolation in (10.11) can then be written as a matrix-vector multiplication,

$$u_i(\mathbf{x}) \approx \sum_{L=1}^{2N} N_{iL}(\mathbf{x}) U_L \ , \quad i = 1, 2 \tag{10.14}$$

Where the $2 \times 2N$ interpolation matrix $\mathbf{N}$ is defined as,

$$\mathbf{N}(\mathbf{x}) = \begin{pmatrix} N_1(\mathbf{x}) & 0 & \ldots & N_N(\mathbf{x}) & 0 \\ 0 & N_1(\mathbf{x}) & \ldots & 0 & N_N(\mathbf{x}) \end{pmatrix} \tag{10.15}$$

The discretized fields (10.14) defined on the finite element grid with N nodal points form a $2N$ dimensional linear vector space of 2-vectors over $\mathbb{R}$ denoted as S^h. The columnvectors $\mathbf{N}_J$ of the interpolation matrix $\mathbf{N}$ are elements of S^h.

problem 10.3. *Show that the vectorfunctions $\mathbf{N}_J(\mathbf{x})$, defined as the matrix columns of (10.15), are linearly independent. In other words the $\mathbf{N}_J(\mathbf{x})$ represent a basis of the vector space S^h.*
Hint: Show for a linear combination $\mathbf{I}(\mathbf{x}) = \sum_{L=1}^{2N} \alpha_L \mathbf{N}_L(\mathbf{x})$, that $\mathbf{I}(\mathbf{x}) = \mathbf{0} \Leftrightarrow \alpha_L = 0$, $L = 1, \ldots, 2N$, by evaluation of $\mathbf{I}(\mathbf{x})$ in the nodal points of the finite element mesh and applying the property $N_J(\mathbf{x}_I) = \delta_{JI}$ of the Lagrange functions.

[2] The total traction vector $t_i = \sigma_{ij} n_j$ can be written explicitly decomposed in normal and tangential components, $\mathbf{t}_n$ and $\mathbf{t}_s$ with, $\mathbf{t}_n = (\mathbf{t} \cdot \mathbf{n})\mathbf{n}$ or $t_{ni} = \sigma_{kl} n_l n_k n_i$, $\mathbf{t}_s = \mathbf{t} - (\mathbf{t} \cdot \mathbf{n})\mathbf{n} = \sigma\mathbf{n} - (\sigma\mathbf{n} \cdot \mathbf{n})\mathbf{n}$, or $t_{si} = \sigma_{ij} n_j - \sigma_{kl} n_l n_k n_i$,

The vector interpolation scheme (10.14) is equivalent to a function expansion in terms of the basis functions $\mathbf{N}_J$ of the vector space S^h.

$$\mathbf{u}(\mathbf{x}) \approx \mathbf{u}^h(\mathbf{x}) = \sum_{L=1}^{2N} \mathbf{N}_L U_L = \mathbf{NU} \tag{10.16}$$

The expressions (10.15),(10.16) will be used in the following for the derivation of finite element methods for (vector) elastic problems. The resulting solution from these finite element methods is an approximation $\mathbf{u}^h \in S^h$.

10.3 Expressions for the deformation and stress fields

The displacement field $\mathbf{u}(\mathbf{x})$ is associated with a deformation or strain tensor field, defined by the symmetric strain tensor,

$$\epsilon_{ij} = \frac{1}{2}\left(\partial_i u_j + \partial_j u_i\right) \tag{10.17}$$

We consider the 2-D case here and rewrite the symmetric strain tensor of order two as a 3×1 deformation or strain vector ϵ defined by,

$$\epsilon = (\epsilon_1, \epsilon_2, \gamma)^T = (\epsilon_{11}, \epsilon_{22}, 2\epsilon_{12})^T \tag{10.18}$$

This definition of the strain vector is related to the special plane-strain case defined for a 3-D configuration,

$$\epsilon_{i3} = 0, \quad i = 1, 2, 3 \tag{10.19}$$

Deformation in the direction perpendicular to the x, y plane is zero in the case (10.19).

The deformation vector (10.18) can be expressed in the components of the displacement field by definition of a suitable differential operator,

$$\epsilon(\mathbf{x}) = \begin{pmatrix} \partial_1 & 0 \\ 0 & \partial_2 \\ \partial_2 & \partial_1 \end{pmatrix} \begin{pmatrix} u_1(\mathbf{x}) \\ u_2(\mathbf{x}) \end{pmatrix} \tag{10.20}$$

For the discretized problem we find by substitution of the interpolation expression (10.16),

$$\epsilon(\mathbf{x}) = \begin{pmatrix} \partial_1 & 0 \\ 0 & \partial_2 \\ \partial_2 & \partial_1 \end{pmatrix} \mathbf{NU} = \mathbf{BU} \tag{10.21}$$

where the $3 \times 2N$ matrix $\mathbf{B}$ follows from applying the operator ϵ in (10.21) to the column vectors of the matrix $\mathbf{N}$ in (10.15) as,

$$\mathbf{B}(\mathbf{x}) = \begin{pmatrix} \partial_1 N_1 & 0 & \dots & \partial_1 N_N & 0 \\ 0 & \partial_2 N_1 & \dots & 0 & \partial_2 N_N \\ \partial_2 N_1 & \partial_1 N_1 & \dots & \partial_2 N_N & \partial_1 N_N \end{pmatrix} \tag{10.22}$$

The expressions used here for the elastic deformation field are generalizations of the expressions used in the previous chapters for the gradient of a scalar potential field.

The matrix $\mathbf{B}$ transforms the interpolated displacement field $\mathbf{u}$ in the corresponding deformation field ϵ. $\mathbf{B}$ is known as the *strain-displacement* matrix.

From the symmetric stress tensor σ_{ij} we define the stress vector,

$$\sigma = (\sigma_1, \sigma_2, \tau)^T = (\sigma_{11}, \sigma_{22}, \sigma_{12})^T \tag{10.23}$$

We further assume a linear elastic medium where the following linear stress-strain relation holds,

$$\sigma_{ij} = c_{ijkl}(\mathbf{x})\epsilon_{kl}(\mathbf{x}) \tag{10.24}$$

For an isotropic medium, described by two independent elastic parameters, we have,

$$c_{ijkl} = \lambda\delta_{ij}\delta_{kl} + \mu\left(\delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk}\right) \tag{10.25}$$

where λ and μ are the Lamé parameters of the isotropic elastic medium and δ_{pq} is the Kronecker delta symbol.

problem 10.4. *Derive for the 2-D plane-strain case the following stress-strain relation,*

$$\begin{pmatrix} \sigma_1 \\ \sigma_2 \\ \tau \end{pmatrix} = \begin{pmatrix} \lambda+2\mu & \lambda & 0 \\ \lambda & \lambda+2\mu & 0 \\ 0 & 0 & \mu \end{pmatrix} \begin{pmatrix} \epsilon_1 \\ \epsilon_2 \\ \gamma \end{pmatrix} = \mathbf{D}\epsilon \tag{10.26}$$

Hint: first derive from (10.24) and (10.25) the stress-strain relation $\sigma_{ij} = \lambda\delta_{ij}\epsilon_{kk} + 2\mu\epsilon_{ij}$.

In practice the Lamé parameters are often replaced by Youngs modulus E and Poissons ratio ν, [3]

$$\lambda = \frac{\nu E}{(1+\nu)(1-2\nu)} \tag{10.27}$$

$$\mu = \frac{E}{2(1+\nu)} \tag{10.28}$$

The elasticity matrix $\mathbf{D}$ for the plane strain case becomes

$$\mathbf{D} = \frac{E(1-\nu)}{(1+\nu)(1-2\nu)} \begin{pmatrix} 1 & \frac{\nu}{1-\nu} & 0 \\ \frac{\nu}{1-\nu} & 1 & 0 \\ 0 & 0 & \frac{1-2\nu}{2(1-\nu)} \end{pmatrix} \tag{10.29}$$

For the discretized problem the stress vector can now be expressed in the components of the discrete displacement field. From (10.21) and (10.26) we get,

$$\sigma = \mathbf{D}\epsilon = \mathbf{DBU} \tag{10.30}$$

The stress field (10.30) expressed in the nodal point displacement values is similar to corresponding expressions for the heat flow density vector expressed in the nodal point temperature values $\mathbf{q} = -k\nabla T$ in Chapter 6.

[3] see Turcotte & Schubert, Geodynamics, 2002 and `http://scienceworld.wolfram.com/physics` (elasticity),

10.4 Discretization of the elastostatic equation

The elastostatic equation is discretized by replacing the displacement field by means of interpolation on the finite element grid. For a displacement field $\mathbf{u} \in \mathbb{R}^2$ and N nodal points we have $2N$ degrees of freedom (not considering possible essential boundary conditions). By applying a Galerkin method we can also derive $2N$ equations in the $2N$ unknowns. In a similar way as for scalar problems in Chapter 6 we require that the residu of the differential equation is perpendicular to all the interpolation basis functions involved, using a suitable innerproduct, defined on the vector space S^h,

$$\int_V \mathbf{N}_I^T \left(\mathbf{L}[\mathbf{u}] - \mathbf{f}\right) \, dV = 0, \quad I = 1, 2, \ldots, 2N \tag{10.31}$$

$$\int_V N_{iI} \left(\partial_j \sigma_{ij}[\mathbf{u}] + \rho F_i\right) \, dV = 0, \quad I = 1, 2, \ldots, 2N \tag{10.32}$$

$$\int_{\partial V} N_{iI} \sigma_{ij} n_j \, dA - \int_V \partial_j N_{iI} \sigma_{ij}[\mathbf{u}] \, dV + \int_V N_{iI} \rho F_i \, dV = 0, \quad I = 1, 2, \ldots, 2N \tag{10.33}$$

problem 10.5. *Verify that the above expression (10.31)*

$$(\mathbf{F} \cdot \mathbf{G}) = \int_V \mathbf{F}^T \mathbf{G} \, dV \tag{10.34}$$

fulfills the following rules for an innerproduct on a linear space of vector functions on V,

$$(\mathbf{F} \cdot \mathbf{G}) = (\mathbf{G} \cdot \mathbf{F}) \tag{10.35}$$

$$((\mathbf{F} + \mathbf{H}) \cdot \mathbf{G}) = (\mathbf{F} \cdot \mathbf{G}) + (\mathbf{H} \cdot \mathbf{G}) \tag{10.36}$$

$$(\alpha \mathbf{F} \cdot \mathbf{G}) = \alpha (\mathbf{F} \cdot \mathbf{G}) \tag{10.37}$$

$$(\mathbf{F} \cdot \mathbf{F}) \geq 0 \tag{10.38}$$

The first term in (10.33) is determined by the boundary conditions to be specified. The third term represents the righthandside vector of the discrete equations - together with a possible contribution from the first two terms in case of inhomogeneous boundary conditions. The second term in (10.33) results in the stiffness matrix. We rewrite this term making use of the symmetry of the stress tensor,

$$I_2 = \int_V \partial_j N_{iI} \sigma_{ij}[\mathbf{u}] \, dV = \int_V \epsilon_{ij}[\mathbf{N}_I] \sigma_{ij}[\mathbf{u}] \, dV \tag{10.39}$$

In terms of the strain and stress vectors ϵ and σ this becomes,

$$I_2 = \int_V \epsilon^T[\mathbf{N}_I] \sigma[\mathbf{u}] \, dV \tag{10.40}$$

Substitution in (10.40) of the basis function expansion for $\mathbf{u}$ in the stress vector (10.30) and the expression for the strain (10.21) gives,

$$I_2 = \int_V \mathbf{B}_I^T \mathbf{D} \mathbf{B} \, dV \; \mathbf{U} \tag{10.41}$$

Writing explicitly for the matrix product $\mathbf{BU}$ in (10.41),

$$\mathbf{BU} = \sum_{J=1}^{2N} \mathbf{B}_J U_J \tag{10.42}$$

we get,

$$I_2 = \sum_{J=1}^{2N} \int_V \mathbf{B}_I^T \mathbf{D} \mathbf{B}_J \; dV U_J = \sum_{J=1}^{2N} S_{IJ} U_J \tag{10.43}$$

The stiffness matrix is identified from this as,

$$\mathbf{S} = \int_V \mathbf{B}^T \mathbf{D} \mathbf{B} \; dV \tag{10.44}$$

This is a similar expression as introduced for scalar potential problems in Chapter 6.

10.4.1 Computation of the stiffness matrix

Implementation of the expressions for the stiffness matrix (10.44) is done in similar ways as in Chapter 6 for scalar potential problems. Obvious choices for the elements are the linear triangles and bi-linear quadrilaterals. We consider here the stiffness matrix for a quadrilateral isoparametric bi-linear element. A related element for the scalar potential equation has been introduced in section 6.2.2. For element number K we compute the element stiffness matrix as,

$$\mathbf{S}^{(K)} = \int_{e_K} \mathbf{B}^T \mathbf{D} \mathbf{B} \; dV(x,y) = \int_{e_u} \mathbf{B}^T \mathbf{D} \mathbf{B} J \; dV(\xi,\eta) \tag{10.45}$$

where J is the determinant of the Jacobi matrix corresponding to the isoparametric coordinate transformation. We evaluate the integral using numerical Gauss-Legendre quadrature (see Appendix A) on the square standard element in the ξ, η domain,

$$\mathbf{S}^{(K)} = \sum_{j=1}^{m} w_j \mathbf{B}^T(\xi_j,\eta_j) \mathbf{D}(\xi_j,\eta_j) \mathbf{B}(\xi_j,\eta_j) J(\xi_j,\eta_j) \tag{10.46}$$

where (ξ_j, η_j) and w_j are the coordinates and weights of the m integration points The 3×8 strain-displacement matrix of the element is defined by,

$$\mathbf{B}(\mathbf{x}) = \begin{pmatrix} N_{1x} & 0 & \dots & N_{4x} & 0 \\ 0 & N_{1y} & \dots & 0 & N_{4y} \\ N_{1y} & N_{1x} & \dots & N_{4y} & N_{4x} \end{pmatrix} \tag{10.47}$$

The partial derivatives in (10.47) transform with the coordinate transformation according to,

$$\mathbf{B}(\xi,\eta) = \begin{pmatrix} j_{11}N_{1\xi} + j_{12}N_{1\eta} & 0 & \dots \\ 0 & j_{21}N_{1\xi} + j_{22}N_{1\eta} & \dots \\ j_{21}N_{1\xi} + j_{22}N_{1\eta} & j_{11}N_{1\xi} + j_{12}N_{1\eta} & \dots \end{pmatrix} \tag{10.48}$$

The transformation matrix $\mathbf{j}$ and the inverse $\mathbf{J}$ have been defined in Chapter 6. The derivatives of the basis functions in (10.48) are listed in Chapter 6 in Table 2. The coefficients of the matrix $\mathbf{B}$ can now be computed with (10.48). The result is then used in the computation of the element stiffness matrix using (10.46). Finally the global stiffness matrix is computed by adding the contributions from the individual element matrices in a loop over elements in a matrix assembly procedure.

problem 10.6. *Verify that the element stiffness matrix for the quadrilateral element described above is an* 8×8 *matrix.*

10.5 Implementation of boundary conditions

In section 10.1 three types of boundary conditions in use for elastic deformation problems were given,

$$u_i(\mathbf{x}) = g_i(\mathbf{x}),\ \mathbf{x} \in \Gamma_{g_i},\ i = 1, \ldots m \tag{10.49}$$

$$\sigma_{ij} n_j = h_i(\mathbf{x}),\ \mathbf{x} \in \Gamma_{h_i},\ i = 1, \ldots m \tag{10.50}$$

$$\sigma_{ij} n_j + \alpha_{ij} u_j = r_i(\mathbf{x}),\ \mathbf{x} \in \Gamma_{r_i},\ i = 1, \ldots m \tag{10.51}$$

We distinguish between essential (10.49) and natural (10.50),(10.51) boundary conditions. The implementation of the different types is similar as for scalar potential problems.

For a problem with essential boundary conditions we partition the solution vector,

$$\mathbf{U} = (\mathbf{U}_f, \mathbf{U}_p)^T \tag{10.52}$$

and expand the displacement field in a reduced set of basis functions which are zero in the boundary points of Γ_{g_i},

$$\mathbf{u}^h = \mathbf{N}\mathbf{U} = \mathbf{N}_f \mathbf{U}_f + \mathbf{N}_p \mathbf{U}_p \tag{10.53}$$

where $\mathbf{U}_p$ is the vector of prescribed components of the displacement field on the boundary and $\mathbf{U}_f$ is vector of unknown displacement components, the degrees of freedom of the problem. The matrix $\mathbf{N}_f$ contains the matrix columns of $\mathbf{N}$ that are zero on the boundary Γ_{g_i}. $\mathbf{N}_p$ consists of the remaining matrix columns. The Galerkin equations are now evaluated for the reduced set of basis functions, the columns of $\mathbf{N}_f$. The boundary integral contribution of Γ_{g_i} in (10.33) is then equal to zero.

The contribution from the essential boundary conditions to the righthand side vector follows from the partitioning of the solution vector, the righthand side vector and the stiffness matrix. The partitioned equations are,

$$\mathbf{S}\mathbf{U} = \begin{pmatrix} \mathbf{S}_{ff} & \mathbf{S}_{fp} \end{pmatrix} (\mathbf{U}_f, \mathbf{U}_p)^T = \mathbf{F}_f \tag{10.54}$$

This results in a reduced set of equations to be solved for $\mathbf{U}_f$,

$$\mathbf{S}_{ff}\mathbf{U}_f = \mathbf{F}_f - \mathbf{S}_{fp}\mathbf{U}_p = \mathbf{F}_f - \mathbf{F}^{(1)} \tag{10.55}$$

Natural boundary conditions of the type (10.50) result in a contribution to the righthand side vector,

$$\int_{\partial V} N_{iI} \sigma_{ij} n_j \, dA = \int_{\partial V} N_{iI} h_i \, dA = F_I^{(2)} \tag{10.56}$$

The Robin type boundary condition (10.51) results in contribution to the righthand side vector as well as a stiffness matrix contribution.

problem 10.7. *Show for the Robin type boundary condition,*

- *For the righthand side vector,*

$$F_I^{(3)} = \int_{\Gamma_{r_i}} N_{iI} r_i \, dA \tag{10.57}$$

- *For the stifness matrix,*

$$S_{IJ}^{(3)} = \int_{\Gamma_{r_i}} N_{iI} \alpha_{ij} N_{jJ} \, dA \tag{10.58}$$

10.6 Examples of elastostatic modelling problems

We consider two applications of the finite element method for elastostatic problems with different combinations of boundary conditions on a 2-D domain $V = [0,1] \times [0,1]$, and boundary made up of segments $c_i, i = 1, \ldots 4$, illustrated in Fig. 10.1

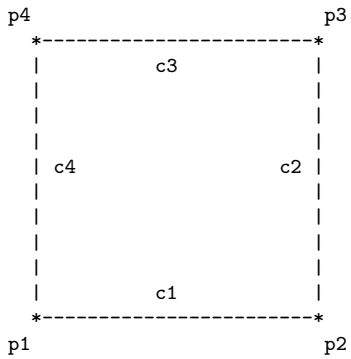

Figure 10.1: *Domain diagram showing the boundary segment curves used in specifying the different boundary conditions.* `p1=(0,0), p3=(1,1).`

The boundary conditions are specified for the individual curves separately. We first consider two simple cases with uniaxial displacement field, $\epsilon_{22} \neq 0$, $\epsilon_{11} = \epsilon_{12} = 0$, i.e. the strain is non-zero along a single main-axis (Turcotte & Schubert, 2002). In both cases the displacement in the elastic medium is computed resulting from a load along the top boundary C_3.

A problem with boundary conditions of types 1 and 2

This problem is defined by setting the volume forces to zero $\rho\mathbf{F} = \mathbf{0}$ and imposing the following boundary conditions,

$$u_1(\mathbf{x}) = u_2(\mathbf{x}) = 0, \ \mathbf{x} \in c_1 \tag{10.59}$$

$$\left.\begin{array}{l} u_1(\mathbf{x}) = 0 \\ \sigma_{12}(\mathbf{x}) = 0 \end{array}\right\} \ \mathbf{x} \in c_2 \cup c_4 \tag{10.60}$$

$$\left.\begin{array}{l} \sigma_{22}(\mathbf{x}) = F(\mathbf{x}) \\ \sigma_{12}(\mathbf{x}) = 0 \end{array}\right\} \ \mathbf{x} \in c_3 \tag{10.61}$$

The displacement along the bottom boundary is set to zero. Along the vertical boundaries the normal displacement is set to zero as well as the tangential traction component (free slip condition). The load is defined by specifying a non-zero vertical component of the boundary traction on the top boundary. This configuration is applicable in a model for the flexure of an elastic lithospheric plate as a response to a variable sediment loading. If we assume a uniform load, the problem becomes a one-dimensional uniaxial one.

First consider the more general problem defined by the elastostatic equation including a uniform vertical body force field and a vertical load along the top boundary curve C_3.

$$\partial_j \sigma_{ij} + f_i = 0, \ i = 1, 2 \tag{10.62}$$

For the $i = 1$ equation we get, using $\partial x \cdot = 0$ and $\mathbf{u} = (u, v), u = 0$,

$$\partial_x \sigma_{xx} + \partial_y \sigma_{xy} = \partial_y (\mu \epsilon_{xy}) = -f_x = 0 \rightarrow \epsilon_{xy} = 0 \tag{10.63}$$

For the y component we have with $\partial x \cdot = 0$ and $\epsilon_{xx} \sim \partial_x u = 0$,

$$\partial_x \sigma_{xy} + \partial_y \sigma_{yy} = \partial_y (\lambda \epsilon_{xx} + (\lambda + 2\mu)\epsilon_{yy}) = (\lambda + 2\mu)\frac{\partial^2 v}{\partial y^2} = -f_y \tag{10.64}$$

This general result, a second order ODE for the vertical displacement can now be applied for the solution of several 1-D problems. For the case with zero body force we substitute $f_2 = 0$ in (10.64) to obtain,

$$\frac{d^2 v}{dy^2} = 0 \rightarrow v(y) = Ay + B, \; v(0) = B = 0 \tag{10.65}$$

From the given uniform traction along $C_3 : \; y = 1$ we get,

$$t_y = \sigma_{yj} n_j = \sigma_{yy} = (\lambda + 2\mu)\epsilon_{yy} = (\lambda + 2\mu)\frac{dv}{dy} = (\lambda + 2\mu)A = F \tag{10.66}$$

This leads to the solution, $A = F/(\lambda + 2\mu), \; B = 0$ or, with $\lambda + 2\mu = \frac{E(1-\nu)}{(1+\nu)(1-2\nu)}$,

$$v(y) = \frac{(1+\nu)(1-2\nu)}{E(1-\nu)} F y \tag{10.67}$$

The corresponding stress field is found from $v(y) = F_y/(\lambda + 2\mu) = cy$,

$$\sigma_2 = \lambda \epsilon_1 + (\lambda + 2\mu)\epsilon_2 = (\lambda + 2\mu)\partial_y (cy) = (\lambda + 2\mu)c = F \tag{10.68}$$

For the above case with uniform F we find a linear displacement field with uniform gradient (strain) and therefore also a uniform stress field. This problem can be used to benchmark a finite element code. In this case the numerical solution should be accurate when using linear finite elements because of the linear nature of the analytical solution.

problem 10.8. *The same elastic column as described above can be loaded under its own weight in a gravity field. The boundary conditions are the same as before with the exception of c_3 which is now a stess-free boundary. Derive the analytical solution for this model,*

$$u_2(y) = \frac{\rho g}{2(\lambda + 2\mu)} \left(y^2 - 2y \right) \tag{10.69}$$

and for the stress field,

$$\sigma_{22}(y) = -\rho g (1 - y) \tag{10.70}$$

where ρ and g are the density of the medium and the gravity acceleration respectively.

A problem with boundary conditions of types 1,2 and 3

This problem is defined by setting the volume forces to zero $\rho \mathbf{F} = \mathbf{0}$ and posing the following boundary conditions,

$$\left.\begin{aligned} -\sigma_{22}(\mathbf{x}) + \alpha u_2(\mathbf{x}) = 0 \\ \sigma_{12}(\mathbf{x}) = 0 \end{aligned}\right\} \; \mathbf{x} \in c_1, \; \alpha > 0 \tag{10.71}$$

where we have used $\sigma_{ij}n_j = -\sigma_{22}$ on c_1.

$$\left.\begin{array}{l} u_1(\mathbf{x}) = 0 \\ \sigma_{12}(\mathbf{x}) = 0 \end{array}\right\} \ \mathbf{x} \in c_2 \cup c_4 \tag{10.72}$$

$$\left.\begin{array}{l} \sigma_{22}(\mathbf{x}) = F(\mathbf{x}) \\ \sigma_{12}(\mathbf{x}) = 0 \end{array}\right\} \ \mathbf{x} \in c_3 \tag{10.73}$$

The boundary condition on c_1 produces a rebound force proportional to the boundary displacement.

problem 10.9. *Derive the analytical solution for the displacement field for the case of uniform load F,*

$$u_2(y) = F\left(\frac{1}{\alpha} + \frac{(1+\nu)(1-2\nu)}{E(1-\nu)}y\right) \tag{10.74}$$

problem 10.10. *The Robin type boundary condition can be used in models for the elastic flexure of the lithosphere that include the effect of hydrostatic rebound forces from the asthenopheric mantle modelled as a fluid half space. Verify the following boundary condition for the bottom of the lithosphere for this case,*

$$\begin{pmatrix} t_1 \\ t_2 \end{pmatrix} + \begin{pmatrix} \alpha_{11} & \alpha_{12} \\ \alpha_{21} & \alpha_{22} \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \begin{pmatrix} r_1 \\ r_2 \end{pmatrix} \tag{10.75}$$

where $t_i = \sigma_{ij}n_j$ and u_i are the boundary traction and displacement fields on the bottom boundary of the lithosphere. Show for a mantle density ρ_m and gravity accelaration g, $\alpha_{i1} = 0$, $\alpha_{i2} = \rho_m g n_i$. The last value can be approximated by $\alpha_{i2} = \rho_m g \delta_{i2}$. What assumption has been made here?

Chapter 11

Finite element methods for viscous flow problems

11.1 Introduction

The Navier-Stokes equation for the flow velocity field of a viscous fluid can be written as,

$$\rho \frac{Du_i}{Dt} = \partial_j \sigma_{ij} + \rho F_i \tag{11.1}$$

where the u_i are the vector components of the velocity field and the Lagrangian time derivative is defined as,

$$\frac{Du_i}{Dt} = \frac{\partial u_i}{\partial t} + u_j \frac{\partial u_i}{\partial x_j} \tag{11.2}$$

Note that the second term on the righthand side represents a non-linearity in the Navier-Stokes equation. For a viscous medium the stress tensor is split in a static and a viscous shear part,

$$\sigma_{ij} = -P\delta_{ij} + \tau_{ij} \tag{11.3}$$

where P is the thermodynamic pressure and τ_{ij} is the viscous shear stress tensor which can be expressed in the viscosity η and the strain rate tensor e_{ij} as,

$$\tau_{ij} = 2\eta e_{ij}, \quad e_{ij} = \frac{1}{2}\left(\partial_i u_j + \partial_j u_i\right) \tag{11.4}$$

For the geodynamically important case with *creeping flow* it can be shown that the inertial term in (11.1) can be neglected. It can be shown by a suitable non-dimensionalization of the equation that the inertial term scales with the inverse Prandtl number of the medium $Pr = \eta_0/(\rho_0\kappa_0)$ where η_0, ρ_0 and κ_0 are the viscosity, density and thermal diffusivity scale values. For the earths mantle $Pr \approx 10^{24}$.

Dropping the inertial term in (11.1) we obtain the Stokes equation which no longer contains an explicit time dependence. The velocity field can still be time dependent however through the time dependence of the body force field ρF_i. An example of this is found in time dependent Rayleigh-Benard convection where the buoyancy term related to the gravitational body force is coupled to a time dependent temperature field. Time dependence of the flow field can also be the result of time dependent driving forces acting on the boundaries, for example in the case of large scale tectonic plates driving asthenopheric flow.

Based on the above we only consider the Stokes flow case in the following,

$$\partial_j \sigma_{ij} + \rho F_i = -\partial_i P + \partial_j \tau_{ij} + \rho F_i = 0 \tag{11.5}$$

For a given body force field $\mathbf{F}$ the Stokes equation (11.5), in a 2-D configuration, contains five dependent variables, P, τ and ρ. The three stress tensor elements in (11.5) are expressed in the velocity components by the constitution equation (11.4), reducing the number of dependent variables to four. In the general case of a compressible fluid the variable density may be dependent on pressure and temperature in a material dependent way. For a complete model description we need an equation of state describing the density as a function of pressure and temperature, adding one equation to the model equations.

$$f(\rho, P, T) = 0 \tag{11.6}$$

To complete the set of model equations we shall apply the continuity equation derived from applying mass conservation,

$$\frac{D\rho}{Dt} + \rho \partial_j u_j = 0 \tag{11.7}$$

problem 11.1. *Verify that (11.5), together with (11.4), (11.6) and (11.7) represent a complete system of equations for 2-D (isothermal) problems and how this result can be extended to 3-D configurations and variable temperature.*

Most mantle convection models are based on the assumption of an incompressible fluid, with $\partial_j u_j = 0$. In this case the differential equation describing the viscous flow velocity field in a high viscosity fluid, the Stokes equation, can be reformulated in terms of scalar potentials, the stream function and vorticity (Turcotte & Schubert, 2002). This has the advantage that the resulting coupled potential equations are relatively simple to solve using numerical methods for scalar potential problems introduced in the previous chapters.

problem 11.2. *Verify that incompressible models still allow a variable density $\rho(\mathbf{x}, t)$, but that $D\rho/Dt = 0$. Give a physical interpretation of this.*

Here we follow a different approach keeping the flow velocity components and the pressure as the variables to be solved. We shall describe finite element methods for the Stokes equation. In the previous chapter finite element methods for vector problems were applied to elasto-static deformation problems. The vector methods introduced in that chapter can be used also for the viscous flow problems. A more extensive treatment of numerical methods for Navier-Stokes problems can be found in (Cuvelier et al., 1986). [1]

[1] C. Cuvelier, A. Segal, A.A. van Steenhoven, *Finite element methods and Navier-Stokes equations*, Reidel, Dordrecht, The Netherlands, 1986.

11.2 Boundary conditions for viscous flow problems

Boundary conditions are implemented in a similar way as in the finite element method for elastic problems presented in the previous chapter. If m is the dimension of the problem, i.e. the number of vector components of the flow velocity field, the number of boundary conditions to be specified is also m. We will consider in particular the following types of boundary conditions (see also Chapter 7),

$$u_i(\mathbf{x}) = g_i(\mathbf{x}),\ \mathbf{x} \in \Gamma_{g_i} \tag{11.8}$$

$$\sigma_{ij} n_j = h_i(\mathbf{x}),\ \mathbf{x} \in \Gamma_{h_i} \tag{11.9}$$

$$\sigma_{ij} n_j + \alpha_{ij} u_j = r_i(\mathbf{x}),\ \mathbf{x} \in \Gamma_{r_i} \tag{11.10}$$

With condition (11.8) the components of the flow velocity field are prescribed, this is an essential boundary condition. Condition (11.9) specifies the components of the traction vector that consists of a pressure (normal) contribution and a shear contribution $t_i = \sigma_{ij} n_j = -P n_i + \tau_{ij} n_j$. The third type (11.10) can be used to prescribe a linear combination of the flow velocity and the traction vector. (11.9) follows as a special case from (11.10) in case $\alpha_{ij} = 0$. (11.9) and (11.10) are natural boundary conditions.

problem 11.3. *Verify how the following physical boundary conditions can be implemented by proper choices from the above three conditions (11.8),(11.9),(11.10),*

- *A no-slip boundary where the fluid sticks to a rigid boundary surface.*
- *A stress-free boundary where the fluid can move freely without any resistive forces from the boundary.*
- *A free-slip impermeable boundary where the fluid can not penetrate the boundary surface and the fluid experiences no (shear) resistance in the direction tangential to the boundary.*
- *An impermeable boundary with intermediate shear conditions where the tangential shear depends on the tangential (slip) velocity component.*

The boundary conditions can be applied component wise in combinations of the different types described above. We only consider 2-D cases here where components of the relevant fields are specified either in the direction of the normal- or the tangential vector at the boundary. We further assume,

$$\Gamma = \partial V = \Gamma_{g_i} \cup \Gamma_{h_i} \cup \Gamma_{r_i},\ i = 1, 2 \tag{11.11}$$

$$\Gamma_{g_i} \cap \Gamma_{h_i} = \Gamma_{g_i} \cap \Gamma_{r_i} = \Gamma_{h_i} \cap \Gamma_{r_i} = \emptyset\ ,\quad i = 1, 2 \tag{11.12}$$

This implies that in each point of the 2-D boundary curve Γ two conditions are specified and that both conditions apply to different component directions. Some consequences of these rules are:

- If both components of the velocity are prescribed in a boundary point i.e. $\mathbf{x} \in \Gamma_{g_1} \cap \Gamma_{g_2}$, then the tractioncomponents in $\mathbf{x}$ are degrees of freedom and can no longer be prescribed. This represents a kinematic boundary condition.

- The opposite is also true: if both traction components are prescribed, $\mathbf{x} \in \Gamma_{h_1} \cap \Gamma_{h_2}$ then both velocity components are degrees of freedom. Such a situation occurs in problems with free boundaries of a viscous medium, as applicable for the earth's surface in a viscous mantle model that includes a moving top boundary. In most modelling applications this condition is replaced by an approximate one of an impermeable free-slip boundary, where the location of the boundary is fixed. This is done for practical reasons so that computations can be done on a single (eulerian) grid of nodal points and remeshing in relation to the deformation of the domain boundaries can be avoided.

- In cases where, besides a traction component, also a velocity component is prescribed, both prescribed vectors must be in different directions. As an example consider a boundary with prescribed (zero) normal velocity component and free slip,

$$u_n = u_j n_j = 0 \tag{11.13}$$

and for the tangential stress expressed in the normal vector,
$\mathbf{t}_s = \mathbf{t} - (\mathbf{t} \cdot \mathbf{n})\mathbf{n} = \sigma\mathbf{n} - (\sigma\mathbf{n} \cdot \mathbf{n})\mathbf{n}$,

$$t_{si} = \sigma_{ij} n_j - \sigma_{kl} n_l n_k n_i = 0 \tag{11.14}$$

where $\mathbf{s}$ is de tangential (unit) vector, $s_j n_j = 0$ (see also section 10.1).

In section 11.5 applications of these boundary conditions in finite element models are given.

11.3 Discretization of the Stokes equation using the Galerkin method

We consider here four combinations of the boundary conditions treated in 11.2, where the domain boundary is split in four corresponding sub boundaries,

$$\partial V = \Gamma = \cup_{i=1}^{4} \Gamma_i, \ \Gamma_i \cap \Gamma_j = \emptyset, \ i \neq j \tag{11.15}$$

The following combination of boundary conditions is used for the four sub boundaries, where components in the direction of the normal and tangential vector are denoted by subscripts n and t respectively.

$$\begin{array}{cccc} u_n(\mathbf{x}) = g_{1n}(\mathbf{x}) & u_t(\mathbf{x}) = g_{1t}(\mathbf{x}) & \mathbf{x} \in \Gamma_1 \\ u_n(\mathbf{x}) = g_{2n}(\mathbf{x}) & \sigma_{nt}(\mathbf{x}) = h_{2t}(\mathbf{x}) & \mathbf{x} \in \Gamma_2 \\ \sigma_{nn}(\mathbf{x}) = h_{3n}(\mathbf{x}) & u_t(\mathbf{x}) = g_{3t}(\mathbf{x}) & \mathbf{x} \in \Gamma_3 \\ \sigma_{nn}(\mathbf{x}) = h_{4n}(\mathbf{x}) & \sigma_{nt}(\mathbf{x}) = h_{4t}(\mathbf{x}) & \mathbf{x} \in \Gamma_4 \end{array} \tag{11.16}$$

The boundary tractions are here written as stress components.

We first apply the Galerkin method to derive a discretized equation from the continuity equation (11.7) for an incompressible fluid with $\nabla \cdot \mathbf{u} = 0$. We will use scalar basis functions N_I^P, to be specified below, to represent the scalar pressure.

$$\int_V N_I^P \nabla \cdot \mathbf{u} \, dV = 0, \ I = 1, 2, \ldots \tag{11.17}$$

The Galerkin principle applied to the Stokes equation (11.5) gives,

$$\int_V N_{iI}\,(\partial_j \sigma_{ij} + \rho F_i)\;\, dV = 0,\;\; I = 1, 2, \ldots \tag{11.18}$$

where the $\mathbf{N}_I$ are the vector basis functions introduced in Chapter 10, applied here to expand the flow velocity field in the nodal point values,

$$u_i(\mathbf{x}) = \sum_{I=1}^{2N} U_I N_{iI}(\mathbf{x}) \tag{11.19}$$

Integration by parts in (11.18) gives,

$$\int_{\partial V} N_{iI}\sigma_{ij} n_j \; dA - \int_V \sigma_{ij}\partial_j N_{iI} \; dV + \int_V N_{iI}\rho F_i \; dV = 0,\;\; I = 1, 2, \ldots \tag{11.20}$$

In the next step we split the total stress tensor in the volume integral in pressure and viscous shear stress, $\sigma_{ij} = -P\delta_{ij} + \tau_{ij}$. Substitution of $\tau_{ij}\partial_j N_{iI} = \tau_{ij}[\mathbf{u}]e_{ij}[\mathbf{N}_I]$ and decomposition of the boundary traction vector $\sigma_{ij}n_j$ and the basis vector N_{iI} in normal and tangential components results in,

$$\begin{aligned}&\int_{\partial V} (N_{nI}\sigma_{nn} + N_{tI}\sigma_{nt})\;\, dA + \int_V (P\partial_i N_{iI} - \tau_{ij}[\mathbf{u}]e_{ij}[\mathbf{N}_I])\;\, dV + \\ &\int_V N_{iI}\rho F_i \; dV = 0,\;\; I = 1, 2, \ldots\end{aligned} \tag{11.21}$$

The boundary integral in (11.21) is determined by the boundary conditions in (11.16).

problem 11.4. *Verify the derivation of the boundary integral contribution by splitting the relevant vectors in normal and tangential components,* $t_i = \sigma_{ij}n_j$, $\mathbf{t} = \sigma_{nn}\mathbf{n} + \sigma_{nt}\mathbf{s}$ *and* $\mathbf{N}_I = N_{nI}\mathbf{n} + N_{tI}\mathbf{s}$.
Hint: consider the innerproduct $(\mathbf{a}\cdot\mathbf{b}) = ((\mathbf{a}_n + \mathbf{a}_t)\cdot(\mathbf{b}_n + \mathbf{b}_t))$.

The boundary surface Γ is split in the four sub boundaries defined in (11.15), (11.16). On Γ_1 the velocity field $\mathbf{u}$ is prescribed. We shall use only those basis functions that are zero on Γ_1 and as a result the boundary integral contribution from Γ_1 is cancelled.

In a similar way on Γ_2 the normal component of the velocity is prescribed, $u_n = g_{2n}$ and the reduced set of basis functions are chosen such that $N_{nI} = 0$ on Γ_2 and the remaining contribution to the boundary integral for Γ_2 is $N_{tI}\sigma_{nt} = N_{tI}h_{2t}$.

For Γ_3 we take basis functions with $N_{tI} = 0$ and find for the remaining part of the integrand $N_{nI}\sigma_{nn} = N_{nI}h_{3n}$ and on the boundary Γ_4 we have the integrand value $N_{nI}h_{4n} + N_{tI}h_{4t}$.

Substitution of these specific contributions from the four sub boundaries in (11.21) gives,

$$\begin{aligned}\int_V \{P\partial_i N_{iI} - \tau_{ij}[\mathbf{u}]e_{ij}[\mathbf{N}_I]\}\;\, dV \;\; = \;\; & -\int_V N_{iI}\rho F_i \; dV \\ & -\int_{\Gamma_2} N_{tI}h_{2t} \; dA \\ & -\int_{\Gamma_3} N_{nI}h_{3n} \; dA \\ & -\int_{\Gamma_4} (N_{nI}h_{4n} + N_{tI}h_{4t})\;\, dA, \\ & I = 1, 2, \ldots\end{aligned} \tag{11.22}$$

By substitution of the basis function expansion for P and $\mathbf{u}$ in the pressure and shear stress terms P and $\tau_{ij}[\mathbf{u}]$, (11.22) is transformed into a system of algebraic equations. For this substitution we need an expression for $\tau_{ij} = 2\eta e_{ij}$. In an analogous way as for elastic problems we define the strain rate vector as,

$$\mathbf{e} = (e_1, e_2, \gamma)^T = (e_{11}, e_{22}, 2e_{12})^T \tag{11.23}$$

We express the strain rate vector in the nodalpoint values of the velocity field components in a similar way as was done for the strain vector in Chapter 7, using the matrix $\mathbf{B}$ introduced in Chapter 7 as $\mathbf{e}(\mathbf{x}) = \mathbf{B}(\mathbf{x})\mathbf{U}$. In the present context the matrix $\mathbf{B}$ is a *strainrate-velocity* matrix.

problem 11.5. *An expressions for the stiffness matrix can be derived by substitution of the basis function expansion,* $\mathbf{u}(\mathbf{x}) = \mathbf{NU}$ *(10.16), in the stress-strainrate term* $\tau_{ij}e_{ij}$ *in (11.22).* $\mathbf{N}$ *is the* $m \times mN$ *interpolation matrix defined in Chapter 10.* $m = 2$ *or* 3 *is the number of components of the velocity field. Derive the following result,*

$$\int_V \tau_{ij}[\mathbf{u}] e_{ij}[\mathbf{N}_I] \; dV = \sum_{J=1}^{mN} \int_V \mathbf{B}_I^T \mathbf{D} \mathbf{B}_J \; dV \; U_J = \sum_{J=1}^{mN} S_{IJ} U_J \tag{11.24}$$

where the diagonal matrix $\mathbf{D}$ *is defined in the stress-strainrate relation* $\mathbf{t} = \mathbf{De}$*, and the stress and strain-rate are written as m+1-vectors as in Chapter 10.*

For a 2-D flow field with $m = 2$*,* $\mathbf{u} = (u_1, u_2)^T$*, the strain rate,* $e_{ij} = 1/2(\partial_i u_j + \partial_j u_i)$*, is rewritten as a 3-vector* $\mathbf{e} = (e_1, e_2, \gamma) = (e_{11}, e_{22}, 2e_{12})^T$*. The shear stress* $\tau_{ij} = 2\eta e_{ij}$ *is written as the 3-vector* $\mathbf{t} = (t_1, t_2, t_3) = (\tau_{11}, \tau_{22}, \tau_{12})^T$ *and,*

$$\begin{aligned}
\begin{pmatrix} t_1 \\ t_2 \\ t_3 \end{pmatrix} &= \begin{pmatrix} \tau_{11} \\ \tau_{22} \\ \tau_{12} \end{pmatrix} = 2\eta \begin{pmatrix} e_{11} \\ e_{22} \\ e_{12} \end{pmatrix} = 2\eta \begin{pmatrix} e_1 \\ e_2 \\ \gamma/2 \end{pmatrix} \\
\mathbf{t} &= 2\eta \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1/2 \end{pmatrix} \begin{pmatrix} e_1 \\ e_2 \\ \gamma \end{pmatrix} = \mathbf{De}
\end{aligned} \tag{11.25}$$

The discretized form of the continuity equation (11.7) is obtained by rewriting the divergence of the velocity field as,

$$\nabla \cdot \mathbf{u} \approx \sum_{J=1}^{mN} \partial_j N_{jJ} U_J \tag{11.26}$$

Integrating the divergence operator, weighted by the pressure basis functions, in the Galerkin scheme we obtain,

$$\int_V N_K^P \partial_j u_j \; dV \approx \sum_{J=1}^{mN} L_{KJ} U_J = 0, \;\; K = 1, 2, \ldots, N_P \tag{11.27}$$

where N_P is the number of pressure basis functions involved and the $N_P \times mN$ matrix $\mathbf{L}$ is defined as,

$$L_{KJ} = \int_V N_K^P \partial_j N_{jJ} \; dV, K = 1, \ldots, N_p \; , J = 1, \ldots, mN \tag{11.28}$$

The pressure term in (11.22) is rewritten by substitution of the special scalar basis function expansion $P(\mathbf{x}) = \mathbf{N}^P(\mathbf{x})\mathbf{P}$, where $\mathbf{N}^P$ is the $1 \times N_P$ row vector of the scalar

basis functions used for the pressure and N_P the number of nodal points where the discrete pressure values are defined.

$$\int_V P\partial_i N_{iI}\ dV \approx \sum_{M=1}^{N_P} L^T_{IM} P_M \tag{11.29}$$

and the $mN \times N_P$ matrix $\mathbf{L}^T$ is defined as

$$L^T_{IM} = \int_V (\partial_i N_{iI})\, N^P_M\ dV, I = 1, \ldots, mN\ , M = 1, \ldots, N_P \tag{11.30}$$

Summarizing we see that the Stokes and continuity equations for the incompressible fluid result in the following discrete algebraic equations,

$$\mathbf{SU} - \mathbf{L}^T\mathbf{P} = \mathbf{F} \tag{11.31}$$

$$\mathbf{LU} = \mathbf{0} \tag{11.32}$$

11.4 The penalty function method

Direct numerical solution of the coupled equations (11.31) and (11.32) for the pressure and flow velocity often leads to numerical problems. The reason for this is that (11.32) does not contain the pressure such that the combined matrix has a zero block submatrix on the diagonal. For this reason the combined equations are usually not solved directly but the pressure degrees of freedom are first eliminated by applying an approximation of the continuity equation. In stead of the original continuity equation an approximate version is used,

$$\epsilon P + \partial_j u_j = 0 \tag{11.33}$$

for a small value of the *penalty parameter* ϵ, typically $\epsilon = 10^{-6}$. Application of the Galerking principle to (11.33) using scalar basis functions $N^P(\mathbf{x})$ results in the following equation,

$$\epsilon\mathbf{M}^P\mathbf{P} + \mathbf{LU} = 0 \tag{11.34}$$

problem 11.6. *Derive the following expression for the pressure-massmatric* $\mathbf{M}^P$*,*

$$M^P_{KM} = \int_V N^P_K N^P_M\ dV \tag{11.35}$$

Elimination of the pressure from (11.34) gives,

$$\mathbf{P} = -\epsilon^{-1}\left(\mathbf{M}^P\right)^{-1}\mathbf{LU} \tag{11.36}$$

and substitution in the discrete Stokes equation (11.31) results in,

$$\mathbf{SU} + \epsilon^{-1}\mathbf{L}^T\left(\mathbf{M}^P\right)^{-1}\mathbf{LU} = \mathbf{S}'\mathbf{U} = \mathbf{F} \tag{11.37}$$

These equations can be solved for the velocity values $\mathbf{U}$ and the pressure values $\mathbf{P}$ can then in principle be derived from (11.36).

In case the pressure is represented as piecewise uniform (per element) the element pressure values are not computed from (11.36) but instead by integrating the velocity flux over the individual closed element boundaries.

problem 11.7. *Show that the element pressure value can be obtained by integrating the normal component of the velocity over the element boundary. Hint: apply the divergence theorem to (11.33). This way the pressure can be computed in a postprocessing step from the numerical solution of (11.37).*

11.5 Examples of numerical applications

11.5.1 A Poisseuille flow problem

As an example of a simple application of a *Stokes flow* problem with several types of boundary conditions we consider a 1-D Poisseuille channel flow problem in a 2-D rectangular domain, illustrated in Fig. 11.1. We prescribe a uniform horizontal inflow velocity at the lefthand vertical boundary, $u(\mathbf{x}) = u_0$, $v(\mathbf{x}) = 0$ at $x = 0$. On the horizontal boundaries we have a no-slip condition with $\mathbf{u} = (u, v) = (0, 0)$. On the outflow boundary we prescribe a zero vertical velocity component and a so called *fully developed flow* condition, implying a vanishing horizontal derivative of the velocity , $\partial u/\partial x = 0$. This outflow condition assumes that the effect of the inflow conditions on the outflow boundary is negligable and that the vertical velocity profile corresponds to the solution of the 1-D problem. This requires that the domain is of sufficient aspect ratio (horizontal length over vertical height).

First consider a case without gravity ($g = 0$). In this case the hydrostatic pressure is zero and the hydrodynamic pressure can be written as $P = Ax$, with a uniform driving pressure gradient $\partial P/\partial x = A$. In the Stokes equation the pressure is determined up to an arbitrary constant and we will assume a zero pressure on the outflow boundary. From the developed flow assumption it follows that homogeneous natural boundary conditions apply for the outflow boundary,

$$\frac{\partial u}{\partial x} = 0 \;\rightarrow\; \tau_{nn} = \tau_{xx} = 2\eta\frac{\partial u}{\partial x} = 0 \;\rightarrow\; \sigma_{nn} = -P + \tau_{nn} = -P = 0 \tag{11.38}$$

A vector plot of the solution of this problem is shown in Fig. 11.1.

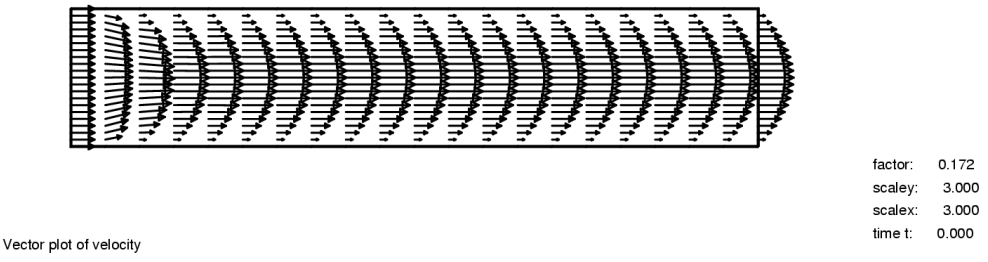

Figure 11.1: *Vector plot of the numerical solution of a 2-D Poisseuille problem.*

Note that the boundary effect, due to the difference between uniform inflow profile and the parabolic 1-D analytical solution, disappears quickly, away from the inflow boundary.

In cases where the effect of a gravitational bodyforce $g > 0$ is included the pressure condition on the outflow boundary is modified due to a non-zero hydrostatic pressure component. For a uniform density we have $\sigma_{nn} = -P + \tau_{nn} = -\rho g z \neq 0$, where z is the depth coordinate with respect to the zero pressure top boundary. A correct implementation of this modified condition results in the same velocity field as shown in Fig. 11.1. Fig. 11.2 shows the result of including the gravitational body force $\rho\mathbf{F}$, with the incorrect uniform pressure condition $\sigma_{nn} = 0$, taken the same as in the model without gravity.

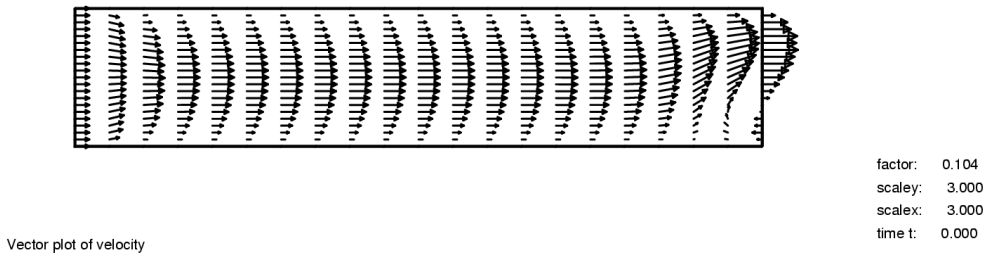

Figure 11.2: *Vector plot of the numerical solution of a 2-D Poisseuille problem including gravity, with boundary condition* $\sigma_{nn} = 0$.

problem 11.8. *Derive the equation for the horizontal flow velocity of the channel flow problem from the Stokes equation for viscous flow* $-\partial_i P + \partial_j \tau_{ij} + \rho g_i = 0$,

$$-\frac{dP_1}{dx} + \eta \frac{d^2u}{dz^2} = 0 \tag{11.39}$$

where P_1 *is the dynamic pressure which is related to the total thermodynamic pressure* P *and the hydrostatic pressure* $P_0 = \rho g z$ *as* $P = P_0 + P_1$.

Show that in the 1-D 'fully developed flow case' the dynamic pressure gradient, $\partial_x P = A$, *is a uniform quantity and that the dynamic pressure* P_1 *is independent of the depth* z.

Hint: First use the vertical component of the Stokes equation to derive $\partial_z P_1 = 0$. *Next apply the horizontal component of the Stokes equation to derive* $\partial_x \partial_x P = 0$ *and from this* $\partial_x \partial_x P_1 = 0$, *using* $P_0 = \rho g z$.

The above model can be used in one-dimensional approximations of flow in the upper mantle.

problem 11.9. *Derive the analytical solution for the 1-D channel flow problem with velocity boundary conditions,*

$$(u)_{z=0} = u_0, \quad (u)_{z=h} = 0 \tag{11.40}$$

(Poisseuille problem $(A \neq 0, u_0 = 0)$, *Couette problem* $(A = 0, u_0 \neq 0, u(h) = 0)$, *where* h *is the channel depth, and* A *is the driving pressure gradient* $\partial_x P = A$.

Answer:

$$u(z) = \left(u_0 - \frac{Ah^2}{2\eta_0} \frac{z}{h} \right) \left(1 - \frac{z}{h} \right) \tag{11.41}$$

In a related model we consider a configuration with $u_0 \neq 0$ and closed loop return flow in an elongated 2-D rectangular domain. We assume that the aspect ratio of the domain $\lambda >> 1$ such that the effect of the vertical boundaries can be neglected in most of the domain and the 1-D fully developed flow description (11.39) holds. For this model a condition for the pressure gradient A can be derived from a zero flux condition applied to a vertical cross-section.

problem 11.10. *Solve the remaining integration constant* $B = \frac{Ah^2}{2\eta_0 u_0}$ *from a mass balance condition for a vertical profile of the horizontal (non-dimensional) velocity,* $u^{'} = u/u_0$, $z^{'} = z/h$,

$$\int_0^1 u'(z')dz' = 0 \tag{11.42}$$

Answer: $B = 3$ *and* $A = \frac{6\eta_0 u_0}{h^2}$.

11.5.2 An example with forced convection in a subcontinental mantel-wedge

Convection in the Earth's mantle can be driven by thermally induced density variations but also by enforced plate velocities of lithospheric plates confining the mantle flow. This is known as kinematically driven mantle flow. We illustrate this here with an example with forced flow in a continental mantle wedge above a subducting lithopsheric plate. The plate velocity is prescribed here along the wedge/plate boundary in terms of the tangential and normal components ($u_t = u_0,\ u_n = 0$). Numerical results of a model calculation are shown in Fig. 11.3.

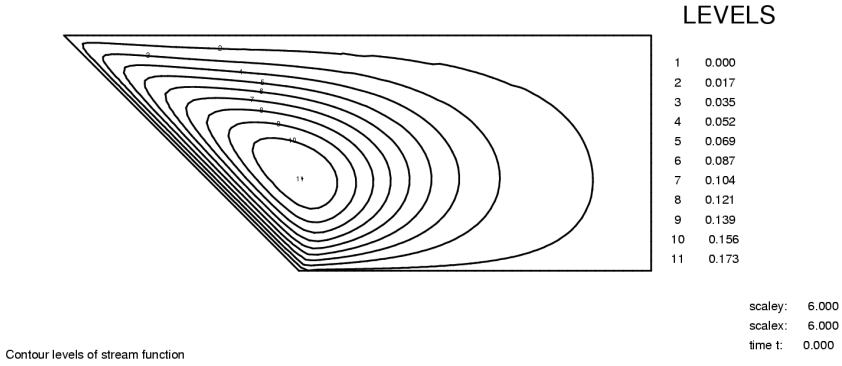

Figure 11.3: *Streamline plot of a numerical mantle wedge model.*

The top horizontal boundary of the mantle wedge has a no-slip condition. On the bottom and righthand side boundary free slip impermeable conditions are specified, $\tau_{nt} = 0, u_n = 0$.

The contour lines of the streamfunction illustrate the steady state flow pattern in the mantle wedge. Comparable simple models have been used to explain the presence of volcanic arcs above subduction zones. In this explanation the flow in the wedge is assumed to focus hot material in the corner region where, aided by fluids from dehydration along the subducting slab, partial melting might result in active volcanism (Iwamori, JGR, 1997, 102, 14.803-14.820).

Appendix A

Numerical integration with the Gauss-Legendre scheme

Element matrices and vectors are usually computed using numerical integration. The value of the integral is written as a weighted sum of integrand values,

$$\int_e f(\mathbf{x})dV = \sum_{j=1}^{m} w_j f(\mathbf{x}_j) \tag{A.1}$$

The $\mathbf{x}_j$ and w_j are the evalution points and integration weights defining the numerical integration scheme. In finite element applications the integrand function $f(\mathbf{x})$ typically contains combinations of derivatives of base functions and coefficients of the differential equation such as thermal conductivity.

Two types of approaches are used in finite element applications. In the first group of the Newton-Cotes methods, evaluation points are the element nodal points. In the second class of the Gauss integration methods the evaluation points are located in the interior of the element.

Here we introduce an often applied two-point Gauss-Legrendre (GL2) integration scheme on a 1-D interval. Two-dimensional integration on a rectangle can than be simply written as repeated integration in the two coordinates. This 1-D GL2 scheme can be derived as the two-point formula which integrates exactly a third order polynomial on the interval $[-1, 1]$. Define the polynomial as,

$$u(x) = a + bx + cx^2 + dx^3 \tag{A.2}$$

Integration over $[-1, 1]$ gives,

$$I = \int_{-1}^{1} u(x)dx = 2(a + \frac{1}{3}c) \tag{A.3}$$

The numerical scheme is specified as,

$$I^h = \sum_{j=1}^{2} w_j u(x_j) \tag{A.4}$$

The integration weights w_j and evaluation points x_j are now derived from the requirements that (A.4) exactly represents (A.3). This leads to,

$$w_1(a + bx_1 + cx_1^2 + dx_1^3) + w_2(a + bx_2 + cx_2^2 + dx_2^3) = 2(a + \frac{1}{3}c) \tag{A.5}$$

Equating the coefficients of a, b, c, d in (A.5) we obtain,

$$\begin{array}{rcrcl} w_1 & + & w_2 & = & 2 \\ w_1 x_1 & + & w_2 x_2 & = & 0 \\ w_1 x_1^2 & + & w_2 x_2^2 & = & \frac{2}{3} \\ w_1 x_1^3 & + & w_2 x_2^3 & = & 0 \end{array} \tag{A.6}$$

From symmetry of the integration points and weights it is found that $w_1 = w_2 = 1$ and $x_1 = -x_2 = 1/\sqrt{3} = 0.577350269\ldots$.

This 1-D numerical integration scheme can be extended to a four-point scheme for application with quadrilateral elements introduced in chapter 6 by repeated integration in the two coordinate directions.

$$I = \int_{-1}^{1}\int_{-1}^{1} u(x,y)dxdy \approx \sum_{j=1}^{4} w_j u(x_j, y_j) \tag{A.7}$$

where $x_1 = x_4 = -x_2 = -x_3 = y_1 = y_4 = -y_2 = -y_3 = 1/\sqrt{3}$

problem A.1. *The 1-D two-point Gauss scheme is apparantly exact for third order polynomials. Verify that the 2-D four-point scheme is sufficiently accurate for the calculation of the element matrix for quadrilateral elements introduced in chapter 6. Which condition must be imposed on the coefficient $c(\mathbf{x})$ in this context?*

Appendix B

Vector and matrix norms

Vector norms have the following characteristics that will be applied in the rest of this appendix,

$$||\mathbf{A}|| > 0, \ \forall \mathbf{A} \neq \mathbf{0} \tag{B.1}$$

$$||\alpha \mathbf{A}|| = |\alpha| \mathbf{A} \tag{B.2}$$

$$||\mathbf{A} + \mathbf{B}|| \leq ||\mathbf{A}|| + ||\mathbf{B}|| \tag{B.3}$$

A general class of vector norms that can be used on an n dimensional vector space are the so called Hölder norms,

$$||\mathbf{x}||_p = \left(\sum_{i=1}^{n} |x_i|^p \right)^{\frac{1}{p}} \tag{B.4}$$

with examples,

$$p = 1, \ ||\mathbf{x}||_1 = \sum_{i=1}^{n} |x_i|, \ \textit{linear norm} \tag{B.5}$$

$$||\mathbf{x}||_2 = \left(\sum_{i=1}^{n} |x_i|^2 \right)^{\frac{1}{2}}, \ \textit{Euclidic norm} \tag{B.6}$$

In the limiting case $p \to \infty$ we obtain the supremum norm,

$$||\mathbf{x}||_\infty = \max_{1 \leq i \leq n} |x_i| \tag{B.7}$$

These vector norms on a vector space $\mathbb{R}^n$ induce a so called associated matrix norm on the linear space of $n \times n$ matrices $\mathbf{A}$ defined as,

$$||\mathbf{A}|| = \sup_{\mathbf{x} \neq \mathbf{0}} \frac{||\mathbf{A}\mathbf{x}||}{||\mathbf{x}||} = \sup_{||\mathbf{x}||=1} ||\mathbf{A}\mathbf{x}|| \tag{B.8}$$

The last equality follows from the following argument. Suppose $\mathbf{x}_0$ is the maximizing vector. Then we have,

$$||\mathbf{A}|| = \frac{||\mathbf{A}\mathbf{x}_0||}{||\mathbf{x}_0||} = ||\mathbf{A}(||\mathbf{x}_0||^{-1}\mathbf{x}_0)|| = ||\mathbf{A}\mathbf{y}_0||, \ ||\mathbf{y}_0|| = 1 \tag{B.9}$$

and

$$\frac{||\mathbf{A}\mathbf{x}_0||}{||\mathbf{x}_0||} \geq \frac{||\mathbf{A}\mathbf{x}||}{||\mathbf{x}||} \;\Rightarrow\; ||\mathbf{A}\mathbf{y}_0|| \geq ||\mathbf{A}\mathbf{y}|| \;,\; ||\mathbf{y}|| = 1 \tag{B.10}$$

The associated matrixnorm is 'compatible', a property defined as,

$$||\mathbf{A}\mathbf{x}|| \leq ||\mathbf{A}||\;||\mathbf{x}|| \tag{B.11}$$

This follows from,

$$||\mathbf{A}|| = \sup_{||\mathbf{x}||=1} ||\mathbf{A}\mathbf{x}|| \geq \frac{||\mathbf{A}\mathbf{x}||}{||\mathbf{x}||} \Rightarrow ||\mathbf{A}\mathbf{x}|| \leq ||\mathbf{A}||\;||\mathbf{x}|| \tag{B.12}$$

A consequence of (B.12) is that the matrixnorm is also multiplicative,

$$||\mathbf{A}\mathbf{B}|| = ||\mathbf{A}\mathbf{B}\mathbf{x}_0|| = ||\mathbf{A}(\mathbf{B}\mathbf{x}_0)|| \leq ||\mathbf{A}||\;||\mathbf{B}\mathbf{x}_0)|| \leq ||\mathbf{A}||\;||\mathbf{B}|| \tag{B.13}$$

In the following two examples of vectornorms and associated matrixnorms are given.

1. The supremum matrix norm follows from the definition of the vectornorm as follows,

$$||\mathbf{A}||_\infty = \sup_x \frac{||\mathbf{A}\mathbf{x}||_\infty}{||\mathbf{x}||_\infty} = \sup_x \left\{ \frac{\max_i |\sum_i a_{ik}x_k|}{\max_k |x_k|} \right\} = \max_i \sum_k |a_{ik}| \tag{B.14}$$

2. The Euclidian matrix norm of a matrix $\mathbf{A}$ is equal to square root of the largest eigenvalue of the symmetric matrix $\mathbf{A}^T\mathbf{A}$. This follows from,

$$||\mathbf{A}||_2 = \sup_{||\mathbf{x}||=1} ||\mathbf{A}\mathbf{x}||_2 = \sup_{||\mathbf{x}||=1} (\mathbf{A}\mathbf{x}\cdot\mathbf{A}\mathbf{x})^{\frac{1}{2}} = \sup_{||\mathbf{x}||=1} (\mathbf{A}^T\mathbf{A}\mathbf{x}\cdot\mathbf{x})^{\frac{1}{2}} \tag{B.15}$$

Here $\mathbf{A}^T\mathbf{A}$ is symmetric and positive definit which means that it can be diagonalized on a basis of orthogonal eigenvectors $\mathbf{e}_j$ corresponding to eigenvalues $\lambda_j \geq 0$. Expanding the maximizing vector in the eigenvectors $\mathbf{x}_0 = \sum_j \alpha_j \mathbf{e_j}$, we get,

$$\begin{aligned} ||\mathbf{A}||_2 &= \left(\sum_j \lambda_j\alpha_j\mathbf{e}_j \cdot \sum_k \alpha_k\mathbf{e}_k \cdot \right)^{\frac{1}{2}} = \left(\sum_j \lambda_j\alpha_j^2 \right)^{\frac{1}{2}} \\ &\leq \left(\lambda_{\max} \sum_j \alpha_j^2 \right)^{\frac{1}{2}} = \lambda_{\max}^{\frac{1}{2}} \left(||\mathbf{x}_0||_2^2 \right)^{\frac{1}{2}} = \lambda_{\max}^{\frac{1}{2}} ||\mathbf{x}_0||_2 = \lambda_{\max}^{\frac{1}{2}} \end{aligned} \tag{B.16}$$

It follows that the maximizing vector is obtained by $\mathbf{x}_0 = \mathbf{e}_{max}$, the eigenvector corresponding to the largest eigenvalue λ_{max}.

In the special case of a symmetric matrix we have $\mathbf{A}^T\mathbf{A} = \mathbf{A}^2$. Now the eigenvalues of the matrices $\mathbf{A}^T\mathbf{A}$, (λ_j), and $\mathbf{A}$, (μ_j), are related in a simple way $\mu_j = \sqrt{\lambda_j}$. This has been applied in Chapter 8 in the analysis of the stability characteristics of the Euler forward integration scheme for ordinary differential equations, where the stability condition $||\mathbf{A}|| < 1$ was imposed, equivalent with $\lambda_{max} < 1$ or equivalently $\mu_{max} < 1$.

problem B.1. *The concept of the matrix norm and the maximizing vector of unit length* $\mathbf{x}_0$ *can be illustrated by investigating how the unit cicle is mapped by a symmetrix matrix in a 2-D example with a* 2×2 *matrix.*

$$\mathbf{A} = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix} \tag{B.17}$$

Show that this matrix has the following eigenvalues $\lambda_1 = 1, \lambda_2 = 3$ *and eigenvectors* $\mathbf{e}_1 = (\frac{1}{2}\sqrt{2}, -\frac{1}{2}\sqrt{2})^T$, $\mathbf{e}_2 = (\frac{1}{2}\sqrt{2}, \frac{1}{2}\sqrt{2})^T$.

Show that the the unitcircle is mapped by $\mathbf{A}$ *on an ellips with major axis* $\mathbf{e}_2$ *and minor axis* $\mathbf{e}_1$. *The norm of this symmetric matrix corresponds to the larger eigenvalue* $\lambda = 3$ *and equals the half major axis of the ellips.*