university of groningen

faculty of mathematics
and natural sciences

Numerical Bifurcation Analysis of Large Scale Systems

Dr.ir. F.W. Wubs

Lecture Notes in Applied Mathematics

Academic year 2009-2010

Numerical Bifurcation Analysis of Large Scale Systems

Dr.ir. F.W. Wubs

Institute of Mathematics and Computing Science
P.O. Box 407
9700 AK Groningen
The Netherlands

Contents

Preface

These lecture notes are, like many websites, "under construction". Some parts are based on material obtained from Dr.E.F.F. Botta (University of Groningen), Prof.dr. H.A. van der Vorst (Utrecht University) and Prof.dr. H.A. Dijkstra (Utrecht University). I am grateful to these colleagues for providing me their material.

F.W. Wubs
September 2009

Chapter 1

Classification and well-posedness of PDEs

A PDE gives a relation between the partial derivatives of a function u of the n independent variables $x_1, ..., x_n$. The *order* of the PDE is that of the highest order derivative occurring in it, hence

$$F(x_1, ..., x_n, u, u_{x_1}, ..., u_{x_n}) = 0 \qquad \text{with} \qquad u_{x_i} = \frac{\partial u}{\partial x_i}$$

is a PDE of first order. The PDE is called *quasi-linear* if F is linear in its highest order derivatives. The PDE is *linear* if F is linear in all its arguments, except for the $x_i's$.

Usually a PDE is considered only on a part of $\mathbb{R}^n$. Then, for the uniqueness of the solution we need initial and boundary conditions. These conditions cannot be taken arbitrary. Moreover they should be such that the solution is stable with respect to perturbations in these conditions, which leads to the following definition.

Definition 1.1 *A problem, i.e. PDE plus initial and boundary conditions, is said to be well-posed in the sense of Hadamard if it has a unique solution and is stable with respect to perturbations in the data (i.e. coefficients, forcing).*

While performing the classification below we will also indicate which initial and boundary conditions make a problem well posed.

1.1 First order PDEs

1.1.1 First order scalar PDEs

Consider the initial value problem of the first order consisting of an hyperbolic partial differential equation, i.e. a is real,

$$\frac{\partial u}{\partial t} + a\frac{\partial u}{\partial x} = 0, \qquad a > 0,\ t > 0,\ -\infty < x < \infty \tag{1.1}$$

and initial value $u(x,0) = \phi(x)$. This equation is called the *transport equation*, or *first-order wave equation* It is easy to see that $u(x,t) = f(x - at)$ is the general solution of this equation, which means that the solution is constant for lines

$$x - at = \text{constant}$$

These lines are called characteristics. If we equate the general solution to the initial value we find that the solution of the initial value problem is $u = \phi(x - at)$. One could say that the initial value propagates along the characteristics with speed a.
If a is not constant the characteristics will not be straight anymore, but still u will be constant along a characteristic (at least for homogeneous equations, i.e. there is no forcing). If ϕ contains a jump then this jump is also propagated along the corresponding characteristic.
We could also consider (1.1) for $x > 0$. In that case we need a boundary condition at $x = 0$ to make it well posed, so $u(0,t) = r(t)$. Note however that if a would have been negative then this boundary condition will in general conflict with the initial condition. In computations we usually are working on an interval. Only at one of the end points of the interval a condition is needed, which end point is determined by a.
If a is not constant we have to look at its value at the boundary in order to determine whether a boundary conditions is needed there.

1.1.2 Systems of first order equations

Consider the system

$$\frac{\partial \mathbf{u}}{\partial t} + A\frac{\partial \mathbf{u}}{\partial x} = 0 \tag{1.2}$$

where A is a constant real $n \times n$ matrix. We assume that A is diagonizable, hence there exists a nonsingular Q such that

$$Q^{-1}AQ = \Lambda = \begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_n \end{bmatrix}$$

The system is called *hyperbolic* if all eigenvalues λ_i of A are real. In that case we can, using $\mathbf{w} = \mathbf{Q^{-1}u}$, transform the system (1.2) into n decoupled scalar (transport) equations

$$\frac{\partial w_i}{\partial t} + \lambda_i \frac{\partial w_i}{\partial x} = 0 \ , \qquad i = 1, 2, \ldots, n$$

In this case every λ_i defines a characteristic direction along which w_i's is constant. As in the scalar case the sign of λ_i determines where a boundary condition must be prescribed. However, in general we cannot prescribe the quantity we need at the boundary directly, but the imposed boundary together with the quantity that is propagated along the characteristic towards that boundary from within the domain must specify the desired quantity.

Exercise 1.1 *Bring the linearized shallow water equations to diagonal form and determine where and which boundary conditions have to be applied. The equations have the form*

$$\begin{aligned} u_t &= -\bar{u}u_x - g\zeta_x \qquad (1.3) \\ \zeta_t &= -\bar{u}\zeta_x - Hu_x \qquad (1.4) \end{aligned}$$

where H is the depth, g is the gravity constant, u the velocity and ζ the wave height and $\bar{u}$ the average flow speed.

1.1.3 Higher space dimensions

In two dimensional space we have $u_t = -au_x - bu_y$. This equation needs an initial condition. On a boundary we need to consider the generalization of 1D, if $([a, b], n) < 0$ we need to prescribe u, where n is the outward pointing normal at the boundary, or otherwise stated, if vector $[a, b]$ points into the domain we need to prescribe u.

Exercise 1.2 *Show that this indeed comprises the 1D case.*

1.2 Scalar second order PDEs

Consider the following second order partial differential equations.

$$F(x_1, \cdots, x_n, u, \frac{\partial u}{\partial x_1} \cdots \frac{\partial u}{\partial x_n}) + \sum_{i,j=1}^{n} a_{ij} \frac{\partial^2 u}{\partial x_i \partial x_j} = 0$$

The highest order derivatives determine the type of the equation. Therefore we have to study the matrix A defined by coefficients occurring in the second order terms of the PDE:

$$A = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{pmatrix}$$

For sufficient smooth u this matrix can always be chosen symmetric because then we can permute freely the order of the partial derivatives.
We say that the system is

elliptic if all eigenvalues of A have the same sign,

hyperbolic if all eigenvalues of A have the same sign, except one which has an opposite sign,

parabolic if all eigenvalues of A have the same sign, except one which is zero.

If A depends on x then the type of the PDE can depend on x as well. For example $u_{xx} = xu_{yy}$ is an equation which is elliptic in the region $x < 0$, hyperbolic in the region $x > 0$, and parabolic on the line $x = 0$.

1.2.1 Normal form

Note that the value of the eigenvalues is not important in the type of the PDE, therefore second-order PDEs are equivalent to a so-called normal form of each type of equation. The consequence of this is that when we now when the normal form is well posed then we know it too for all the equivalent forms. The highest order part of the aforementioned types of PDEs have a normal form

$$\begin{array}{rl} \dfrac{\partial^2 u}{\partial \xi_1^2} + \cdots + \dfrac{\partial^2 u}{\partial \xi_n^2} & \qquad \text{(elliptic)} \\ \dfrac{\partial^2 u}{\partial \xi_n^2} - (\dfrac{\partial^2 u}{\partial \xi_i^2} + \cdots + \dfrac{\partial^2 u}{\partial \xi_{n-1}^2}) & \qquad \text{(hyperbolic)} \\ \dfrac{\partial u}{\partial \xi_n} - (\dfrac{\partial^2 u}{\partial \xi_i^2} + \cdots + \dfrac{\partial^2 u}{\partial \xi_{n-1}^2}) & \qquad \text{(parabolic)} \end{array}$$

These normal forms can be obtained through a coordinate transformation. Given a problem

$$\left(\frac{\partial}{\partial x_1} \cdots \frac{\partial}{\partial x_n}\right) A \left(\frac{\partial}{\partial x_1} \cdots \frac{\partial}{\partial x_n}\right)^T u = f$$

apply a coordinate transformation

$$\left(\frac{\partial}{\partial x_1} \cdots \frac{\partial}{\partial x_n}\right)^T = S \left(\frac{\partial}{\partial y_1} \cdots \frac{\partial}{\partial y_n}\right)^T$$

such that $S^T AS$ is diagonal and all diagonal elements have magnitude one or zero. This is a so-called *congruence transformation*, it is known that such a transformation does not change the signs of the eigenvalues (also called *inertia*). It is not hard to find such an S. Since A is real and symmetric, we know that there exists an orthogonal matrix Q that will diagonalize A, hence $Q^T AQ = D$, next we pre and post multiply by the same non-singular diagonal matrix $\hat{D}$ such that we get the desired diagonal matrix. Thus, $S = Q\hat{D}$.

Examples in 2D of PDEs in normal form are

$$\begin{array}{ll} \textit{Poisson equation}: & \dfrac{\partial^2 u}{\partial x^2} + \dfrac{\partial^2 u}{\partial y^2} = f \\ \textit{heat equation}: & \dfrac{\partial u}{\partial t} - \dfrac{\partial^2 u}{\partial x^2} = 0 \\ \textit{wave equation}: & \dfrac{\partial^2 u}{\partial t^2} - \dfrac{\partial^2 u}{\partial x^2} = 0 \end{array}$$

If in the Poisson equation the right-hand side is zero, then it is called *Laplace equation*. For the Poisson and Laplace equation to be well posed we need to prescribe u, u_n or a combination $u + \alpha u_n$, with $\alpha > 0$, which are called *Dirichlet*, *Neumann*, and *Robin* condition, respectively. For the heat equation we need next to the boundary conditions in space, which are the same as those for the elliptic case, an initial condition. The wave equation even needs two initial conditions.

A problem which is not well posed is for instance the backward heat equation $u_t = -u_{xx}$ which is integrated from $t = 0$ to some time $T > 0$ using an initial condition and boundary conditions. This is unstable irrespective of the initial or boundary conditions. In this case we should integrate backwards in time in order to have a stable solution.

Exercise 1.3 *Show that the PDE*

$$a(x, y)u_{xx} + 2b(x, y)u_{xy} + c(x, y)u_{yy} = f(x, y, u, u_x, u_y)$$

is elliptic if $b^2 - ac < 0$, *parabolic if* $b^2 - ac = 0$ *and hyperbolic if* $b^2 - ac > 0$.

1.2.2 Self-adjoint operators and problems

Suppose we have defined a PDE on a domain Ω. Then the *adjoint operator* of an operator L is the operator L^* for which $\int_\Omega vLw - wL^*v \; d\Omega$ only depends on u, v and their derivatives on the boundary of Ω for all sufficiently differentiable (C^2) u, v. A *self-adjoint operator* is an operator for which $L = L^*$. In fact the operators occurring in the Poisson and wave equation shown above are self-adjoint, where for the wave equation, one must take a domain ω in the x, t-space.
For linear second order elliptic PDE's, any self-adjoint operator can be written as

$$Lu = -\sum_{i,j=1}^{n} \frac{\partial}{\partial x_i}\left(a_{ij}\frac{\partial u}{\partial x_j}\right) + qu = -\text{div}(A\text{grad } u) + qu \tag{1.5}$$

For a *self-adjoint problem*, the boundary condition should satisfy $\int_\Gamma (vA\,\text{grad}\,w - wA\,\text{grad}\,v \;\; , \mathbf{n})d\Gamma = 0$ for any v, w which satisfy the same homogeneous boundary conditions. *Homogeneous* boundary conditions have a zero in the right-hand side, so $u = 0,\; u_n = 0$, etc. This formula can be derived from $\int_\Omega vLw - wLv \; d\Omega = 0$ using the above definition for L and Gauss' theorem. Neumann, Dirichlet and Robin boundary conditions all satisfy this condition.

1.2.3 Variational form for elliptic problems

For elliptic self-adjoint problems one can rewrite the problem into a *variational form.* For that we define a *functional*, which is a "function" where you enter a function and which gives back a number. Here we have a functional

$$J[v] = \int_\Omega \left[\tfrac{1}{2}(\,\text{grad}\,v, A\,\text{grad}\,v) + \tfrac{1}{2}qv^2 - fv\right] \; d\Omega + \int_{\Gamma_1} (\tfrac{1}{2}\beta v^2 - \gamma v) \, d\Gamma$$

Here Γ_1 is part of Γ (it may be the whole Γ or just be empty) and q and f are functions from $C(\Omega)$ with $q \geq 0$. Likewise β and γ are functions from $C(\Gamma_1)$ with $\beta \geq 0$. Furthermore, the matrix A is symmetric and positive definite. We will now show that the minimization of this functional over a set of specified v will lead to quite a general class of elliptic problems. The set of v's we are considering is the set of functions for which the function self and its derivative is square integrable on Ω and that satisfy on $\Gamma_2 = \Gamma - \Gamma_1$ the so-called *essential boundary condition* $v = r$, with r

a function prescribed on Γ_2. Let w be an arbitrary function which is zero on Γ_2, and has the same integrability properties as a v. Now, for a minimum (or stationary point) of the functional one should first think of the case where v is a vector. Then for the minimum it holds that the directional derivative of the functional should be zero for any direction. This carries over to the case where v is a function. $J[v]$ has a stationary point for $v = u$ if for $v = u + \varepsilon w$, with ε small it holds that

$$\frac{\mathrm{d}}{\mathrm{d}\varepsilon} J[u + \varepsilon w]\Bigg|_{\varepsilon=0} = 0$$

This leads to

$$\int_\Omega [(\operatorname{grad} w, A \operatorname{grad} u) + wqu - wf]\ d\Omega + \int_{\Gamma_1} w(\beta u - \gamma)\, d\Gamma = 0 \tag{1.6}$$

Exercise 1.4 *: Show that $J[u + \varepsilon w] = J[u] + \varepsilon a(u, w) + \varepsilon^2 b(w)$ where $a(u, w)$ will be zero for the stationary point and $b(w) > 0$ for $w \neq 0$, where q and β should be nonnegative.*

Now we need the identity

$$\int_\Omega (\operatorname{grad} w, A \operatorname{grad} u)\, d\Omega = -\int_\Omega w \operatorname{div}(A \operatorname{grad} u)\, d\Omega + \int_\Gamma (wA \operatorname{grad} u, \mathbf{n})\ \mathbf{d\Gamma}$$

to get rid of the derivatives in front of w. This expression is nothing more than a partial integration (verify this for the 1D case). We also need the fact that on Γ_2 $w = 0$, and hence for the stationary point it holds that

$$-\int_\Omega w\, [\operatorname{div}(A \operatorname{grad} u) - qu + f]\ d\Omega + \int_{\Gamma_1} (w\, [(A \operatorname{grad} u, \mathbf{n}) + \beta\mathbf{u} - \gamma]\ d\Gamma = 0$$

Hence with the notation of (1.5) we have

$$Lu = f \quad \text{in } \Omega \tag{1.7a}$$
$$(A \operatorname{grad} u, \mathbf{n}) + \beta\mathbf{u} = \gamma \quad \text{on } \Gamma_1 \tag{1.7b}$$
$$u = r \quad \text{on } \Gamma_2 \tag{1.7c}$$

Note that when we started off only a boundary condition on Γ_2 was supplied, but that the minimization gives us a boundary condition (1.7b) along Γ_1. We call this boundary condition the *natural boundary condition.* The general name for a PDE that is derived from a minimization, here (1.7a), is called the *Euler-Lagrange differential equation.*
Note that the functional where we started from has only first derivatives in it but that the resulting equation (1.7a) has second order derivatives. Hence, the equation requires more smoothness than the functional. Now, the smoothness of u is determined by f, γ and r. If these admit a solution of the equation then the same solution will be found from the minimization process. However, if they do not admit a solution of the equation but still do for the functional then we say that we have found a *weak solution* of the equations. We call (1.6) the *weak form* of the equation. Note that we can find this weak form from the equation by working in opposite direction. This is interesting because, we could go in this way for quite arbitrary equations.

1.2.4 A very general theorem

Let us again consider (1.6). One could write this in the form

$$a(w,u) =< w,g >$$

Exercise 1.5 *What is $a(w,u)$ and $< w,g >$ here?*

Now w is in a linear space, but u is not (show this). We want to rewrite the problem such that both arguments of $a(*,*)$ are in a linear space and actually in the same space. This is not very difficult. Say u_r is some function that satisfies the essential boundary condition and is of sufficient smoothness. Then we write $u = u_r + \hat{u}$, where $\hat{u}$ is in the same space as w. Now we plug this into the above equation to obtain

$$a(w,\hat{u}) =< w,g > -a(w,u_r) =<< w,\hat{g} >> \tag{1.8}$$

For this equation we want to find the $\hat{u}$ such that it holds for all w, where both functions satisfy, the essential boundary condition and have sufficient smoothness. Now the *Lax-Milgram theorem* gives the conditions for which this problem is well posed [22]. At this place we will not go into the details of that theorem, but just pose the most important condition, which is that for all w

$$a(w,w) \geq c||w||^2 \tag{1.9}$$

for some positive c where we will not specify further the norm, but just comment that this is a kind of positive definiteness condition.

Exercise 1.6 *Show that for a symmetric positive definite matrix A it holds that $(x,Ax) \geq c(x,x)$ where c is the smallest eigenvalue of A.*

So if we can bring a non self-adjoint problem in a weak form with $a(*,*)$ being positive definite in the above sense then in general (the other conditions of the Lax-Milgram do in general not cause problems to prove) the weak form is well posed. Hence, in that case a weak solution of our nonself-adjoint problem exists. Note that in this way we can also determine which boundary conditions make the problem well posed.

Exercise 1.7 *Show that with the introduced terminology the functional of the previous section can be written as*

$$J[\hat{u}] = \frac{1}{2}a(\hat{u},\hat{u}) - << \hat{u},\hat{g} >>$$

1.2.5 The wave equation

Consider the following pure initial value problem

$$\frac{\partial^2 u}{\partial t^2} = a^2 \frac{\partial^2 u}{\partial x^2}, \qquad t > 0,\ -\infty < x < \infty \tag{1.10}$$

with initial conditions $u(x,0) = \phi(x)$ and $u_t(x,0) = \psi(x)$. The general solution is in this case

$$u(x,t) = f(x - at) + g(x + at)$$

Hence, the characteristics are given by the lines

$$x \pm at = \text{constant}$$

The solution satisfying the initial conditions is

$$u(x,t) = \tfrac{1}{2}[\phi(x-at) + \phi(x+at)] + \frac{1}{2a}\int_{x-at}^{x+at} \psi(\xi)\, d\xi$$

From this we observe that the initial conditions on say [0,1] determine the solution within the triangle bounded by the line $t = 0$ and the characteristics through the end points of the interval. This triangle is called the *domain of dependency* of the point $P = (\frac{1}{2}, \frac{1}{2}a^{-1})$, see Fig. 1.1. The domain where the point P is codefining the solution is called the domain of influence of P. The domain of dependence is important

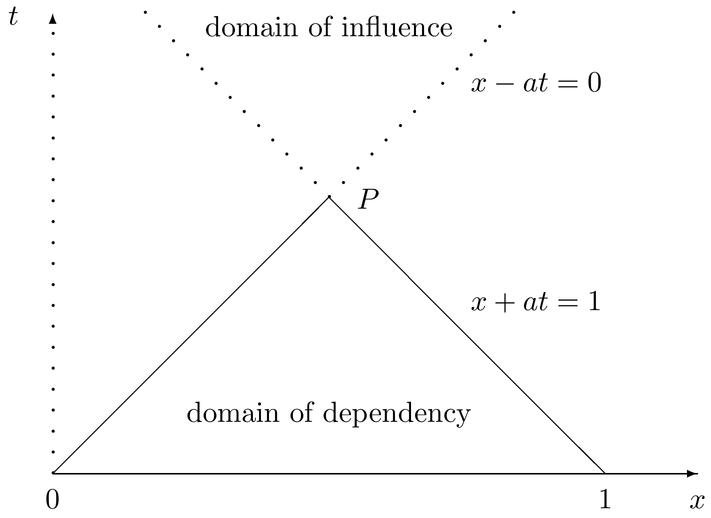


Figure 1.1: Domains of dependency and influence

for numerical methods. It will be clear that a numerical method will not converge for all possible initial conditions if the domain of dependency is not included in the numerical domain of dependency. This lead to the so-called *Courant-Friedrichs-Lewy (CFL) condition* (1928) which states that a numerical method cannot be stable if the numerical domain of dependency does not include the domain of dependency of the continuous equation. Note that this is a necessary condition, so not enough for convergence.

1.2.6 A second-order PDE expressed as a system of first-order PDEs

In general a second-order PDE can transformed to a systemd of first-order PDEs. For instance, by introducing the unknowns

$$u_1 = \frac{\partial u}{\partial x}, \quad u_2 = \frac{\partial u}{\partial y}$$

the Laplace-equation $\Delta u = 0$ can be written as

$$\frac{\partial u_1}{\partial x} + \frac{\partial u_2}{\partial y} = 0, \qquad \frac{\partial u_1}{\partial y} - \frac{\partial u_2}{\partial x} = 0$$

These are the so-called Cauchy-Riemann equations. This is of the form (1.2) with matrix

$$A = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

The eigenvalues of A are $\pm\mathrm{i}$ and hence this is not an hyperbolic system of PDEs. In fact it is elliptic just as its corresponding second-order representation.

Exercise 1.8 *Give a second-order scalar variant of the linearized shallow-water equations (1.3) with* $\bar{\mathrm{u}} = 0$ *and observe that that is also hyperbolic according to the definition for second-order scalar PDEs.*

```
== External Links ==
* http://en.wikipedia.org/wiki/Elliptic_partial_differential_equation
* http://en.wikipedia.org/wiki/Hyperbolic_partial_differential_equation
* http://en.wikipedia.org/wiki/Parabolic_partial_differential_equation
* http://en.wikipedia.org/wiki/Well-posed
```

Chapter 2

Discretization of PDEs

2.1 An overview of discretization strategies for PDEs

Discretization is the process of approximating an infinite dimensional problem by a finite dimensional problem suitable for a computer.
Suppose we have a linear partial differential equation $Lu = f$ on a domain Ω with boundary conditions. We can follow the following strategies to solve the problem:

Finite Difference A straight-forward approach is to cover the domain by a grid, and approximate the PDE by a finite difference equation defined in the grid points using unknowns that are also only defined at the grid points. The basic tool here is the Taylor expansion.

Rayleigh-Ritz If we have a self-adjoint problem, there is a functional $J(u)$ associated to the PDE. The minimum of the functional over the appropriate function space yields the solution of the PDE. (For convenience the original problem is often transformed to a problem with homogeneous boundary conditions.) A discretization is obtained by approximating u by a finite sum of basis functions (called $\hat{u}$) which are all in the appropriate function space,

$$\hat{u}(x) = \sum_{i=1}^{n} c_i \phi_i(x) \tag{2.1}$$

and do sufficiently accurate numerical integrations for the integrals occurring in the functional. This will result in a minimization of the functional over the only free parameters left: the coefficients c_i. This minimization will result in a linear problem for these coefficients.

If the *support*, i.e. where the function is nonzero except for a few points, of every $\phi_i(x)$ extends over the whole domain then the matrix of the linear problem will in general be full (unless we take eigenfunctions of the operator and boundary conditions as basis functions). An example of this approach is the *pseudo-spectral method*. In this method orthogonal polynomials (discussed in Section 8.2 of [2] or Section 10.1 in [22]) are used as basis functions.

If every $\phi_i(x)$ is only locally nonzero (compact support), then the matrix will become sparse. An important example here is the finite element approach (see Sections 11.5 and 12.4 of Burden and Faires).

Weighted Residuals If $\hat{u}$ is again the expansion (2.1) then in general it is not possible to choose the coefficients such that $L\hat{u} - f$ is zero for all points in the domain, since we have only n coefficients and there are an infinite number of points in the domain. Therefore one requires that $(v_j, L\hat{u} - f) = 0$ for $j = 1, ..., n$, where the innerproduct here is just an integral over the domain. Here, v_j is called *test function* and the space spanned by these functions is called the *test space*. The basis functions are said to be in the *search space*, i.e. the space which is spanned by the basis functions and contains the approximate solution. The particular choice for v_j depends on properties of the problem (operator and boundary conditions). We just list common choices.

Galerkin $v_j = \phi_j$, so the search and test space are equal. If we have a self-adjoint problem, the same linear system for the coefficients will occur as in the Rayleigh-Ritz approach.

Petrov-Galerkin The test space is different from the search space. Some common choices are:

Least squares $v_j = L\phi_j$, which is equivalent to the minimization over the search space of $||L\hat{u} - f||_2^2$. This is an approach which always works but the matrix of the resulting linear system may have a rather high condition number, which may give problems with reaching the required accuracy or the convergence of the iterative method. An advantage is that it leads to a Symmetric Positive Definite (SPD) matrix (in fact it can also be viewed as the Rayleigh-Ritz method applied to $L^*Lu = L^*f$).

Collocation $v_j = \delta(x - x_j)$, which results in the requirement that $L\hat{u} - f$ should be zero in n points in the domain.

Finite Volume Here L assumes a special form $Lu = \text{div}(Mu)$, which comes about in conservation laws. Cover the domain with n so-called disjunct *control volumes* also called finite volume. On control volume j we have that test function v_j is 1 and zero elsewhere. Say $\widehat{Mu}$ is some approximation of Mu and apply the weighted residual approach which leads to

$$\begin{aligned}
\int_\Omega v_j(\text{div}(\widehat{Mu}) - f)d\Omega &= 0 \\
\int_{\Omega_j} \text{div}(\widehat{Mu}) - fd\Omega &= 0 \\
\int_{\Gamma_j} (\widehat{Mu}, n)d\Gamma - \int_{\Omega_j} fd\Omega &= 0
\end{aligned}$$

where Ω_j is the $j - th$ control volume, Γ_j its boundary, and n the unit outward pointing normal on the boundary. So far nothing has been said on the approximation $\widehat{Mu}$. We may use basis functions for u or discretize Mu on a grid. In any case the discretization should be such that on the interface of the j-th control volume with one of its neighbors the approximating *flux* $(\widehat{Mu}, n)$ should be equal up to the sign, which should be different (in the continuous case this is true since the outward pointing normal vectors on the interface are equal except for the sign).

Now in the continuous case we have that $\int_\Omega (\mathrm{div}(Mu) - f) d\Omega = 0$ or $\int_\Gamma (Mu, n) d\Gamma = \int_\Omega f d\Omega$, yielding a condition on the flux on the boundary of the domain. A similar condition is found in the discrete case, since

$$\begin{aligned}\int_\Omega (\mathrm{div}(\widehat{Mu}) d\Omega &= \sum_{j=1}^{n} \int_{\Omega_j} (\mathrm{div}(\widehat{Mu}) d\Omega = \sum_{j=1}^{n} \int_{\Gamma_j} (\widehat{Mu}, n) d\Gamma \\ &= \sum_{k \in K} \int_{\Gamma_k} (\widehat{Mu}, n) d\Gamma\end{aligned}$$

where Γ_k is a part of the outer boundary of the domain. Hence here we have

$$\sum_{k \in K} \int_{\Gamma_k} (\widehat{Mu}, n) d\Gamma = \int_\Omega f d\Omega$$

This condition also shows that no artificial numerical fluxes remain in the interior. This is an important property in applications. If the equation describes conservation of mass, energy, or momentum, then this means that there is no artificial loss or growth in the interior due to numerical errors.

```
== External Links ==
* http://en.wikipedia.org/wiki/Rayleigh-Ritz_method
* http://en.wikipedia.org/wiki/Ritz_method
* http://eom.springer.de/R/r082500.htm SpringerLink - Ritz method
* http://en.wikipedia.org/wiki/Collocation_method
```

2.2 Finite-difference methods for elliptic equations

2.2.1 Finite-difference approximations in 1 dimension

Suppose we want to solve a PDE numerically on an interval $\Omega = [0, 1]$. We define the grid points x_i of an equidistant grid as:

$$x_i = ih, \quad i = 0, 1, ..., N; \quad h = 1/N$$

Furthermore, define:

$$u_i = u(x_i), \ i = 0, 1, ..., N$$

Now assume that $u \in C^4[0,1]$. Using the Taylor series of u, we can now derive a finite difference for the derivatives of u in the grid points x_i using

$$u_{i+1} = u(x_i + h) = u_i + h\,\frac{du}{dx}\bigg|_{x_i} + \frac{h^2}{2!}\frac{d^2u}{dx^2}\bigg|_{x_i} + \frac{h^3}{3!}\,\frac{d^3u}{dx^3}\bigg|_{x_i} + \frac{h^4}{4!}\,\frac{d^4u}{dx^4}\bigg|_{x_i+\xi_1 h} \tag{2.2}$$

and

$$u_{i-1} = u(x_i - h) = u_i - h\,\frac{du}{dx}\bigg|_{x_i} + \frac{h^2}{2!}\frac{d^2u}{dx^2}\bigg|_{x_i} - \frac{h^3}{3!}\,\frac{d^3u}{dx^3}\bigg|_{x_i} + \frac{h^4}{4!}\,\frac{d^4u}{dx^4}\bigg|_{x_i-\xi_2 h} \tag{2.3}$$

where $\xi_1, \xi_2 \in [0,1]$. Add both equations, rearrange and use the continuity of the fourth derivative to obtain

$$\frac{d^2u}{dx^2}\bigg|_{x_i} = \frac{u_{i+1} - 2u_i + u_{i-1}}{h^2} - \frac{h^2}{12}\,\frac{d^4u}{dx^4}\bigg|_{x_i+\xi h} \quad , \quad \xi \in [-1,1] \tag{2.4}$$

In the same way we can substract the equations to obtain

$$\frac{du}{dx}\bigg|_{x_i} = \frac{u_{i+1} - u_i}{2h} - \frac{h^2}{3!}\,\frac{d^3u}{dx^3}\bigg|_{x_i+\tau h} \quad , \quad \tau \in [-1,1]$$

Omitting the terms of $O(h^2)$, we call the right-hand side of both equations the second-order central difference for the derivative of the left-hand side of the equation. Second order means we omitted terms of $O(h^2)$ and we thus have a local truncation error of $O(h^2)$. Here "central" refers to the symmetry around x_i.
In the same way can derive central differences for the third and fourth order derivatives:

$$\frac{d^3u}{dx^3}\bigg|_{x_i} = \frac{u_{i+2} - 2u_{i+1} + 2u_{i-1} - u_{i-2}}{2h^3} + O(h^2)$$

$$\frac{d^4u}{dx^4}\bigg|_{x_i} = \frac{u_{i+2} - 4u_{i+1} + 6u_i - 4u_{i-1} + u_{i-2}}{h^4} + O(h^2)$$

Non-central discretizations can also easily be obtained. For example the forward approximation of the first derivative of u can be obtained directly from the expansion (2.2) yielding

$$\frac{\mathrm{d}u}{\mathrm{d}x}\bigg|_{x_i} = \frac{u_{i+1} - u_i}{h} + O(h)$$

Similarly a backward difference can be obtained from (2.3)

$$\frac{\mathrm{d}u}{\mathrm{d}x}\bigg|_{x_i} = \frac{u_i - u_{i-1}}{h} + O(h)$$

ook wel achterwaartse differentiebenadering genoemd.

Higher order discretizations Suppose we want to increase the accuracy to $O(h^4)$ instead of $O(h^2)$. We can do this by using the Taylor series for $u_{i+2} = u(x_i + 2h)$ and $u_{i-2} = u(x_i - 2h)$ too. If we just add all these (four) Taylor series like before, we will get terms containing the fourth derivative of u. So instead of just adding them all together, we need to make a linear combination, such that we only get second derivatives and other derivates are all cancelled out. In this case this is possible as follows:

$$-16u_{i+1} - 16u_{i-1} + u_{i+2} + u_{i-2}$$

Next, after some rewriting, we obtain this central differences formula for the second derivative of u:

$$\left.\frac{d^2u}{dx^2}\right|_{x_i} = \frac{-u_{i+2} + 16u_{i+1} - 30u_i + 16u_{i-1} - u_{i-2}}{12h^2} + O(h^4)$$

Solving a 1D elliptic equation The numerical solution of the equation of the boundary value problem $u_{xx} = f(x)$, $u(0) = u(1) = 0$, proceeds as follows. We require that in every point x_i the differential equation is satisfied where we replace the derivatives in those points by the difference approximations. This yields a system of difference equations for the values in the grid points. The solution of the difference equation will be denoted by U_i which approximates $u(x_i)$. It is called *grid function*, since it is only defined at the grid points. Likewise, we restrict f to the grid points $f_i = f(x_i)$. This yields the linear system (show this)

$$h^{-2}\begin{bmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 2 \end{bmatrix}\begin{bmatrix} U_1 \\ U_2 \\ . \\ U_{N-2} \\ U_{N-1} \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ . \\ f_{N-2} \\ f_{N-1} \end{bmatrix}$$

After this system is solved, we have found the desired approximations of $u(x_i)$ at the grid points x_i, $i = 1, 2, \dots, N-1$. This approximation can be made more accurate by increasing the number of grid points, thereby increasing the size of the linear system, resulting in a higher computation time. This shows that in the numerical solution of PDEs we always have to find a trade-off between the accuracy and the amount of computer time we want to spent. The game is of course to get the accuracy as high as possible by using accurate difference schemes for the lowest possible amount of computer time and/or memory usage.

This example also shows that we run into problems if fourth-order accurate discretizations need to be used at the boundary. Usually we have to be satisfied with lower-order accuracy near the boundaries.

non-uniform grids Finally we consider difference approximations for the first and second derivative on non-uniform grids. Consider, three subsequent grid points

$$x_{i-1}, x_i, x_{i+1}, \quad \text{met} \quad x_i - x_{i-1} = h_1 \quad \text{en} \quad x_{i+1} - x_i = h_2$$

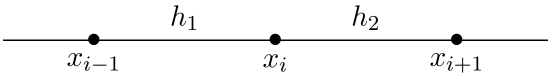


Figure 2.1: Non-uniform grid

as indicated in Fig. 2.1. Analogously to (2.2) and (2.3) we find now the Taylor-expansions

$$\begin{aligned} u(x_i+h_2) &= u_i + h_2 \frac{\mathrm{d}u}{\mathrm{d}x}\bigg|_{x_i} + \frac{h_2^2}{2!}\frac{\mathrm{d}^2u}{\mathrm{d}x^2}\bigg|_{x_i} + \frac{h_2^3}{3!}\frac{\mathrm{d}^3u}{\mathrm{d}x^3}\bigg|_{x_i} + \frac{h_2^4}{4!}\frac{\mathrm{d}^4u}{\mathrm{d}x^4}\bigg|_{x_i+\theta_1 h_2} \\ u(x_i-h_1) &= u_i - h_1 \frac{\mathrm{d}u}{\mathrm{d}x}\bigg|_{x_i} + \frac{h_1^2}{2!}\frac{\mathrm{d}^2u}{\mathrm{d}x^2}\bigg|_{x_i} - \frac{h_1^3}{3!}\frac{\mathrm{d}^3u}{\mathrm{d}x^3}\bigg|_{x_i} + \frac{h_1^4}{4!}\frac{\mathrm{d}^4u}{\mathrm{d}x^4}\bigg|_{x_i-\theta_2 h_1} \end{aligned} \tag{2.5}$$

After multiplication by respectively h_1 and h_2 we find after additions and some algebraic manipulations

$$\frac{\mathrm{d}^2u}{\mathrm{d}x^2}\bigg|_{x_i} = \frac{2u_{i+1}}{h_2(h_1+h_2)} - \frac{2u_i}{h_1h_2} + \frac{2u_{i-1}}{h_1(h_1+h_2)} + \frac{h_1-h_2}{3}\frac{\mathrm{d}^3u}{\mathrm{d}x^3}\bigg|_{x_i} + \cdots \tag{2.6}$$

and in a similar way

$$\frac{\mathrm{d}u}{\mathrm{d}x}\bigg|_{x_i} = \frac{h_1u_{i+1}}{h_2(h_1+h_2)} + \frac{(h_2-h_1)u_i}{h_1h_2} - \frac{h_2u_{i-1}}{h_1(h_1+h_2)} - \frac{h_1h_2}{6}\frac{\mathrm{d}^3u}{\mathrm{d}x^3}\bigg|_{x_i} + \cdots \tag{2.7}$$

When the maximum mesh size is indicated by h, hence, $h_i \le h$, then we see that (2.7) yields an $O(h^2)$ approximation to the first derivative, but that (2.6) only gives a $O(h)$ approximation to the second derivative.

However, often the non-uniform grid will occur as a transformation of a uniform grid. So $x_i = g(t_i)$, where $t_{i+1}-t_i = h$. If that is the case and $g \in C^2$, then the approximation (2.6) will be second-order accurate.

Exercise 2.1 *Show the claim of the last paragraph.*

2.2.2 Finite-difference approximations in 2 dimensions

Now we know how to apply the finite differences method in one dimension, we can easily do the same in two dimensions. Assuming $\Omega = \big\{(x,y) \in R^2; a < x < b,\ c < y < d\big\}$, we again define our grid points to be:

$$(x_i, y_i) = (a+ih, c+jk)\,,\ i = 0,1,...,m;\ j = 0,1,...,n;\ h = \frac{b-a}{m},\ k = \frac{d-c}{n}$$

Now we can just use the results of the one-dimensional part to obtain:

$$\frac{\partial^2u}{\partial x^2}\bigg|_{i,j} = \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h^2} + O(h^2) \tag{2.8}$$

and

$$\left.\frac{\partial^2 u}{\partial y^2}\right|_{i,j} = \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{k^2} + O(k^2) \tag{2.9}$$

where $u_{i,j} = u(x_i, y_j)$. The only difficulty is in case we have mixed derivatives, e.g. u_{xy}. But now we have:

$$\left.\frac{\partial^2 u}{\partial x \partial y}\right|_{i,j} = \frac{\partial}{\partial x}\left[\frac{\partial u}{\partial y}\right]_{i,j}$$

which is, using the central difference for a first derivative, equal to:

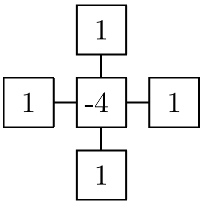
$$\left.\frac{\partial^2 u}{\partial x \partial y}\right|_{i,j} = \frac{1}{2h}\left[\left.\frac{\partial u}{\partial y}\right|_{i+1,j} - \left.\frac{\partial u}{\partial y}\right|_{i-1,j}\right] + O(h^2)$$

Using discretizations for the left derivatives gives the desired result:

$$\left.\frac{\partial^2 u}{\partial x \partial y}\right|_{i,j} = \frac{1}{4hk}\left[u_{i+1,j+1} - u_{i+1,j-1} - u_{i-1,j+1} + u_{i-1,j-1}\right] + O(h^2) + O(k^2)$$

If we replace in a PDE $Lu = f$ the partial derivatives by difference approximations then we obtain also a difference approximation for L denoted by L_h. Very often L_h is schematically written as a *difference molecule* or *stencil.* Here, the occurring weights are shown according to the geometry of the grid. For example for the Laplace operator Δ on a uniform grid with $h = k$, we find with (2.8) and (2.9) a difference approximation Δ_h with error $O(h^2)$, hence $\Delta = \Delta_h + O(h^2)$. The difference molecule is then shown as

$$\Delta_h = h^{-2} \tag{2.10}$$



We call this the 5-point approximation to the Laplace-operator.

Discretization on a non-uniform grid We will look at a lineair elliptic PDE (Lu=f), hence

$$Lu = -au_{xx} - cu_{yy} + du_x + eu_y + qu$$

With a and c positive and $q \geq 0$. In the remaining part we will consider a, c, d, e to be constant, though this is not essential. We use the discretization mentioned in the 1D non-uniform grid part. If we take a five point approach (see Fig. 2.2) we find

$$C_P U_{i,j} - C_W U_{i-1,j} - C_S U_{i,j-1} - C_E U_{i+1,j} - C_N U_{i,j+1} = f_P$$

This leads to the following equations for the coefficients:

$$C_W = \frac{2a + h_2 d}{h_1(h_1 + h_2)},\ C_S = \frac{2c + k_2 e}{k_1(k_1 + k_2)},\ C_E = \frac{2a - h_1 d}{h_2(h_1 + h_2)},$$
$$C_N = \frac{2c - k_1 e}{k_2(k_1 + k_2)},\ C_P = \frac{2a + (h_2 - h_1)d}{h_1 h_2} + \frac{2c + (k_2 - k_1)e}{k_1 k_2} + q \tag{2.11}$$

Thus $C_P = C_W + C_S + C_E + C_N + q$ and with restriction $q \geq 0$ we have $C_P \geq C_W + C_S + C_E + C_N$ If we take h and k small enough all coefficients will be positive The restrictions for h and k to achieve that are $\max_i\ h_i \leq \frac{2a}{|d|}$ and $\max_i\ k_i \leq \frac{2c}{|e|}$ If we write this as a system $Ax = b$ (with x consisting of $u(i,j)$) A will be weakly diagonal dominant, a matrix property favorable for iterative methods. If the coefficient in front of the first derivative in L is big, then the condition on h can be quite restrictive. In that case, one could use an *upwind discretization* for the first derivative. Here upwind means that we pick all the information in the direction where the wind is coming from. In this case this has to do with the coefficients in front of the first and second derivative. In any case it should be chosen such that the diagonal entry increases in order to get again a weakly diagonally dominant matrix. Here we take for the term du_x

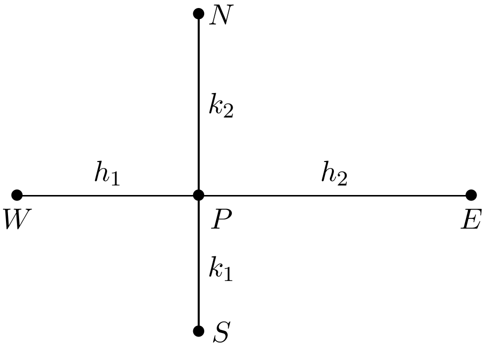


Figure 2.2: non-uniform grid in 2D

$$d\frac{U_{i,j} - U_{i-1,j}}{h_1} \quad \text{if} \quad d > 0$$

and

$$d\frac{U_{i+1,j} - U_{i,j}}{h_2} \quad \text{if} \quad d < 0$$

similarly eu_y. The price of this approach is that we only have a first-order accurate discretization and on top of that we have introduced *artificial diffusion.* This can be seen by making a Taylor expansion

$$\frac{U_{i,j} - U_{i-1,j}}{h_1} = u_x|_{i,j} + h_1 u_{xx}|_{i,j} + O(h_1^2)$$

So the term du_x introduces dissipation of magnitude dh_1 which may be rather big with respect to the real diffusion.

2.2.3 Discretization near the boundary

First we will consider a square area Ω with boundary Γ and a uniform grid. If we look at the five point formula

$$C_P U_{i,j} - C_W U_{i-1,j} - C_S U_{i,j-1} - C_E U_{i+1,j} - C_N U_{i,j+1} = f_P$$

We see that when $i = 1, m-1$ or $j = 1, n-1$ at least one of the terms is a boundary point. If we have a Dirichlet condition $u(a, y) = r(y)$ on the y-axis we can replace the term $C_W U_{0,j}$ by $C_W r(y_j)$. We then move this known term to the right and we get

$$C_P U_{1,j} - C_S U_{1,j-1} - C_E U_{2,j} - C_N U_{1,j+1} = f_P + C_W r(y_i)$$

If we have Neumann boundary conditions we have to use fictive grid points (x_{-1}, y_j), $j = 1, ..., n-1$. With the help of these point we also use the five point formula. We than eliminate the fictive grid points in the formula using the fact that $\frac{U_{1,j} - U_{-1,j}}{2h} = 0$ For a point P we then find that

$$C_P U_{0,j} - C_S U_{0,j-1} - (C_E + C_W) U_{1,j} - C_N U_{0,j+1} = f_P$$

If we have Robin boundary conditions we can do the same thing. Note that in both cases the diagonal dominance of the matrix remains intact. We will see later that this property ensures a unique solution, at least if at least at one point a Dirichlet condition is applied.

For a non-square boundary we also have to use fictive points and inter- and extrapolation.

Finally, we remark that at the boundary we may usually take the order of accuracy one lower than in the internal domain. It can be shown that the order of convergence is still that of the discretization in the internal domain.

2.2.4 Nonlinearity

In finite differencies nonlinearity is not very difficult to handle. Let us consider the example equation

$$uu_x + u_{xx} = f(x)$$

on [0,1] and with Dirichlet boundary conditions. On a uniform mesh, using central discretizations, we obtain

$$u_i(u_{i+1} - u_{i-1})/(2h) + (u_{i+1} - 2u_i + u_{i-1})/h^2 = f_i$$

for $i = 1, ..., N$, $h = 1/N$. This is only one possibility. For instance, one could also write the example equation write in a divergence form

$$\frac{\partial}{\partial x}(\frac{1}{2}u^2 + u_x) = f(x).$$

Then we could employ the finite volume approach shown in the next section.

2.2.5 An example of a finite volume discretization

In this section we illustrate the finite-volume approach by discretizing the self-adjoint operator defined by

$$Lu = -\operatorname{div}(A \operatorname{grad} u)$$

Now it is trivial that on any part of the domain $\Omega_1 \subset \Omega$ we have

$$\int_{\Omega_1} Lu \, d\Omega_1 = \int_{\Omega_1} f \, d\Omega_1$$

Using Gauss' theorem on the left-hand side we obtain

$$-\int_{\Gamma_1} (A \operatorname{grad} u, \mathbf{n}) \, \mathbf{d\Gamma_1} = \int_{\mathbf{\Omega_1}} \mathbf{f} \, \mathbf{d\Omega_1} \tag{2.12}$$

where Γ_1 is the boundary of Ω_1. Here $(A \operatorname{grad} u, \mathbf{n})$ is often called the *flux*, because it usually represent an amount of some entity (mass, heat, etc.) crossing through the boundary per time unit.

We will now consider the specific case where $A = diag([a, c])$, so we have the PDE

$$-\frac{\partial}{\partial x}\left[a\frac{\partial u}{\partial x}\right] - \frac{\partial}{\partial y}\left[c\frac{\partial u}{\partial y}\right] = f \tag{2.13}$$

Assume that Ω is covered by a uniform grid with in both x- and y-direction a mesh size h. In Fig. 2.3 a part of the grid around the point $P = (x_i, y_j)$ is drawn.

Here Ω_1 is the dashed square from which the sides halve the lines connecting P and its neighbors N_1, E_1, S_1 and W_1. For the integrals occurring in (2.12) we apply the midpoint rule. Moreover observe that the normals are [0,1], [1,0], [0,-1] and [-1,0] if we start at N_1 and walk in clockwise direction, which shows that apart from the sign we need only one component of the gradient at these points. Together this yields the following half discretization

$$\left[-a\frac{\partial u}{\partial x}\bigg|_{O_1} + a\frac{\partial u}{\partial x}\bigg|_{W_1}\right] h + \left[-c\frac{\partial u}{\partial y}\bigg|_{N_1} + c\frac{\partial u}{\partial y}\bigg|_{Z_1}\right] h = f_P h^2$$

Note that no matter how we are going to discretize the remaining derivatives, if we do it in exactly the same way in neighboring cells, e.g. $au_x|_{O_1}$ is equal to $au_x|_{W_1}$ of the cell at the right, then if we just sum up all the equations all the internal fluxes cancel and only the flux over the boundary of the domain remains.

To continue the discretization we use the central discretization

$$\frac{\partial u}{\partial x}\bigg|_{O_1} = (U_{i+1,j} - U_{i,j})/h \tag{2.14}$$

which yields the discretization

$$C_P U_{i,j} - C_W U_{i-1,j} - C_S U_{i,j-1} - C_E U_{i+1,j} - C_N U_{i,j+1} = f_P h^2$$

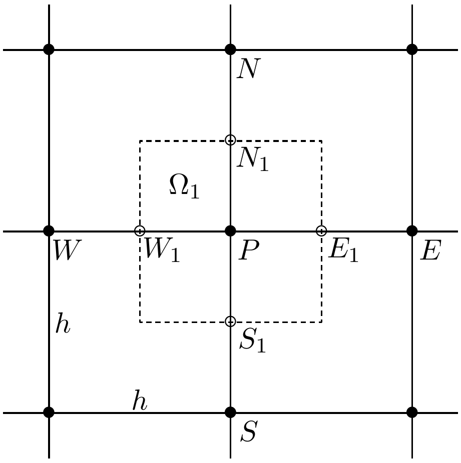


Figure 2.4

Figure 2.3: A control volume

with

$$C_W = a(x_i - \tfrac{1}{2}h, y_j), \quad C_S = c(x_i, y_j - \tfrac{1}{2}h)$$

$$C_E = a(x_i + \tfrac{1}{2}h, y_j) \quad C_N = c(x_i, y_j + \tfrac{1}{2}h)$$

$$C_P = C_W + C_S + C_E + C_N$$

From the fact that C_N is equal to the C_S of the volume on top of this one and similar for the horizontal direction, it follows that the matrix will be symmetric. This can be exploited in the solution process.

Exercise 2.2 *What changes if in y direction we have a mesh size k. Show that the resulting matrix is still symmetric.*

Exercise 2.3 *Give the finite volume discretization of the convection-diffusion equation as written in the last paragraph of the previous section.*

Staggered grid For a number of systems of PDEs it appears to be handy and advantageous to not define the various unknowns occurring in it in the same grid points. So if for instance in the above example a and b would depend on another variable, say v, then it would be nice if that variable would be defined at the midpoints of the control volume faces, so at N_1, E_1, S_1 and W_1. Of course, there is another equation for v that needs to be discretized. Staggered grids are often applied in Computational Fluid Dynamics [21].

2.2.6 The global discretization error

In the previous we have seen how a PDE $Lu = f$ defined on a domain Ω, with boundary conditions at its boundary can be approximated by a system of difference equations $L_hU = f_h$. In general the exact solution u will not satisfy this equation and the residual

$$\tau_h = L_hu - f_h = L_h(u - U) \tag{2.15}$$

which is called *discretization error.* This error could be estimated using Taylor expansions expressed in the mesh size h and the partial derivatives of u. When for $h \to 0$ also $\tau_h \to 0$, the difference scheme is *consistent* and if it holds that $\tau_h = O(h^p)$, then the difference scheme is *consistent of order p.*

As a consequence of the local discretization error also the grid function U will differ from the exact solution u restricted to the grid points. The difference

$$v_h = u - U \tag{2.16}$$

is called the *global discretization error.* We have *convergence* if the global discretization error tends to zero if the mesh size is decreased.

The main point is that L_h should be a stable operator. So in general if we consider $L_hU = f_h$ and the same problem with a slight perturbation in the right-hand side $L_h\tilde{U} = f_h + \delta_h$, then the difference $e_h = \tilde{U} - U$ which is a solution of $L_he_h = \delta_h$ should be bounded in δ_h. It is from 2.15 that the restriction of u to the grid is a solution of the perturbed problem $L_hu = f_h + \tau_h$, hence from stability we find convergence. It appears quite general to be the case that convergence can be proven from stability and consistency. It is called after Lax and Richtmyer or also the *equivalence theorem*, see the famous book of Richtmyer and Morton [23].

In this case stability follows if

$$(y, L_hy) \geq c||y||^2 \tag{2.17}$$

for arbitrary grid functions y that satisfy the Dirichlet boundary conditions. (It is related to (1.9.) In fact c is the minimum eigenvalue of the matrix associated to L_h. Taking $y = e_h = L_h^{-1}\delta_h$, the following inequalities hold: $||L_h^{-1}\delta_h||||\delta_h|| \geq (L_h^{-1}\delta_h, \delta_h) \geq c||L_h^{-1}\delta_h||^2$ where the first one follows using Cauchy-Schwartz. So $||e_h|| = ||L_h^{-1}\delta_h|| \leq \frac{1}{c}||\delta_h||$.

It is beyond the scope to really prove stability of discretizations in general, but one can quite easily observe that some discretization maybe prune to instability. As we are usually looking to smooth solutions the discrete operator will due to consistency very much be acting the same on smooth functions as the original continuous operator will do. The difference is expected for fast-oscillating wave-like functions. If the difference operator is nearly zero for such wave-like functions and much smaller than when applied to a smooth function of the same magnitude while the continuous form is not, then this indicates an instable discretization (this is equivalent of getting a very small c in (2.17). This occurs for example for discretizations where the stencil of the discretization does not connect the odd points to the even points of the grid, e.g. the central discretization for u_x, or similar for red and black points in checker board form on the grid. It depends on the application whether the instability is disastrous for the computation but one should be aware of that. Moreover, there exist ways to stabilize discretizations.

```
== External Links ==
* http://en.wikipedia.org/wiki/Finite_difference_method
* http://en.wikipedia.org/wiki/Five-point_stencil
* http://en.wikipedia.org/wiki/Maximum_principle
```

2.3 Finite element discretization for elliptic equations

The starting point for the finite element method is a weak form such as shown in (1.8). Let us state it here as follows. We are looking for a solution u in a linear space V such that for all $v \in V$ it holds that

$$a(v, u) = (v, f) \tag{2.18}$$

The space is such that all elements of it satisfy the essential boundary condition (in homogeneous form). Now we like to find solutions of the form $\hat{u} = \sum_{j=1}^{N} c_j \phi_j(x)$, where all $\phi_i(x)$, $i = 1, ..., N$ are also in V. In fact they span a subspace of V. This subspace is the search space and written as V_h. We now want (2.18) to hold on V_h. This means that the test space is equal to the search space, and hence this is the Galerkin approach. This gives the linear system

$$Ac = b$$

where $A_{ij} = a(\phi_i, \phi_j)$ and $b_i = (\phi_i, f)$.

Exercise 2.4 *Show this by substituting ϕ_i for v and $\hat{u}$ for u in (2.18).*

Exercise 2.5 *Show that if $a(*, *)$ satisfies (1.9) for all v in V then A is positive definite, and if $a(u, v) = a(v, u)$ for arbitrary u and v in V that A is symmetric.*

Now we only need to define the basis functions. The most common choice is to use interpolation polynomials as basis functions. We could use polynomials that perform an interpolation over the whole domain or use *piecewise polynomial interpolation.* (For the construction of piecewise polynomials we partition the domain in smaller parts (the pieces) and define a low order interpolation on each part, next we require at the edges of the parts some form of continuity.) The former needs high degree polynomials when we have many interpolation points and may suffer from the *Runge-phenomenon* (resulting in oscilatory behavior) if sharp gradients are present in the real solution. Another problem is to find interpolating polynomials on irregularly shaped domains. But, if no sharp gradients are present and the domain is "nice", then this may do a good job and it is favorable to use orthogonal polynomials as a basis. However, if we expect strong gradients and the domain is quite irregular, piecewise interpolating polynomials are much more flexible. Now assume we have defined the piecewise interpolating polynomials we want to use, e.g. a linear approximation on each part of the domain, then this spans a space V_h. Then the next step is to find a nice basis for these

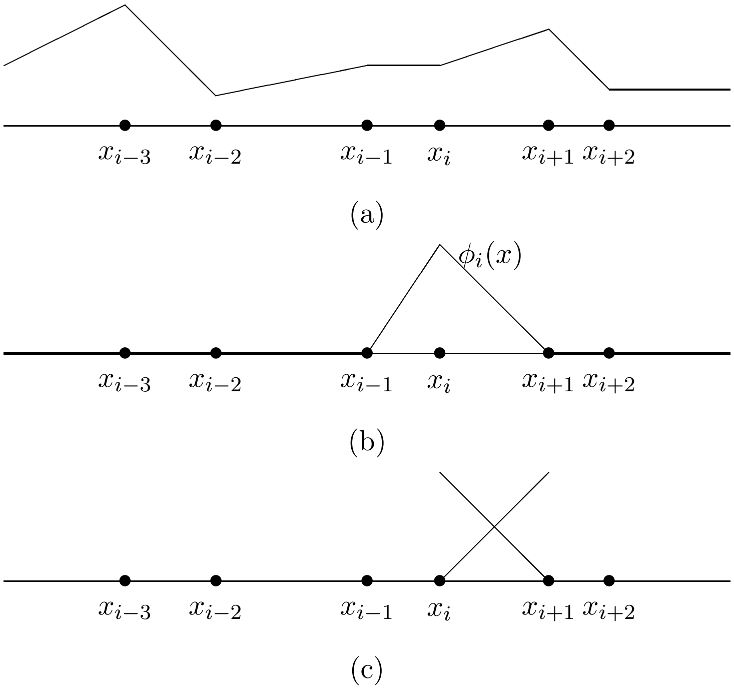


Figure 2.4: (a) Partitioning and piecewise linear polynomial, (b) basis function, (c) element basis functions

polynomials which also spans the space V_h. The nice thing is that there exists a basis in which the basis functions have a very local support and are the building blocks of the finite element approach.
Let us consider the 1D case. Say we partition the domain [0,1] in N parts (not necessarily equal), which are called elements. Now we require the piecewise interpolating function to be linear on each element and that it is continous from one element to a neighboring element (Fig. 2.4.a). As basis function we take a function that is 1 at some interface of 2 neighboring elements and zero at all other interfaces (Fig. 2.4.b). Finally, we take from these basis functions the restriction to one element which yields the *element basis functions* (Fig. 2.4.c). Generally speaking a finite element is defined by a part of the domain and the interpolation used on the element. Where for the latter we should think of the element basis functions.

Exercise 2.6 *Show that if the c_j, $j = 1, ..., N$ are known that then we can compute the function values of the interpolating polynomial easily by using the element basis functions.*

Now the question arises for what kind of equations can the solution be represented by a piecewise linear function $f(x)$. Note that the derivatives of this function are in general not differentiable at an interface. However if we consider the integral $\int (f')^2 dx$ then there is a sequence of 1 times differentiable $\{f_n\}$ that converges to f and moreover

$\int (f_n')^2 dx$ converges and is precisely the integral of $\int (f')^2 dx$ where we have excluded all the interfaces. As in second-order PDEs we arive at an $a(*,*)$ with first-order derivatives which is of the type of integral considered, we can handle second-order PDEs with piecewise linear basis functions.
All piecewise *Lagrangian interpolation*, which means interpolation on function values, have a discontinuity in the first derivative. Hence can only be applied to first and second-order PDEs.
Piecewise Hermite interpolation opens the way to construct polynomials which are continuous in both the function and derivatives. In this case next to function values, also values of derivatives are used to define the interpolating polynomial. For instance if on an interface both function value and derivative are interpolated then we have a continuous differentiable function. In that case we can also handle fourth-order PDEs.

Order of accuracy The order of accuracy that can be obtained by piecewise interpolating functions depends on

1. the order of the differential equation q,
2. the degree of interpolating polynomial p and the smoothness of it m, and
3. the order of the derivative of the unknown function k you are looking to, i.e. u $(k = 0)$ or u_x $(k = 1)$ or u_{xx} $(k = 2)$, etc.

If $2m \geq q$ (as seen in the smoothness discussion above) and $p + 1 - q \geq 0$ then the order of accuracy in the approximation of $\frac{\partial^k u}{\partial x^k}$ is $p + 1 - k$. So for second-order PDEs $(q = 2)$ using piecewise linear interpolation $(m = 1,\ p = 1)$, we have that the function value $(k = 0)$ is approximated to second order. So if the mesh size is halved in all directions then the error decreases by about a factor four.

2.3.1 Some finite elements

Exercise 2.7 *For a 1D domain, draw the element basis functions for quadratic interpolation on each element. In this case next to the end points also the midpoint is used in the interpolation.*

Exercise 2.8 *A cubic interpolation polynomial can be fixed by giving at the end points of an interval both the function value and its derivative. Give the equations that define the four element basis functions.*

Exercise 2.9 *Consider a triangulation of a 2D domain, i.e. the domain is partitioned in triangles. Assume that we want to do a linear approximation on each triangle, which is in fact a plane determined by the function value in its three corners. Make a sketch of the basis function associated to this.*

Exercise 2.10 *Try to find the discription of the quadratic and cubic 2D triangular elements in a book or on the internet and sketch the location of interpolation points. In which way is this a generalization of the corresponding 1D elements?*

2.3.2 Handling constraints

The handling of constraints can be nicely done by using Lagrange multipliers as thought in calculus courses. Here it is introduced in the finite dimensional case.
Say we want to minimize $J(x) = \frac{1}{2}(x, Ax) - (b, x)$ with A SPD under the constraint $Bx = c$. We can bring this constraint within the minimization using Lagrange multipliers. We now have to minimize

$$\hat{J}(x, \mu) = \frac{1}{2}(x, Ax) - (b, x) + (\mu, Bx - c)$$

The minimization over x and μ leads to the following linear equations to be solved

$$\begin{aligned} Ax - b + B^T\mu &= 0, \\ Bx &= c \end{aligned}$$

If A is not symmetric but positive definite, then we still can use the last equation to incorporate a constraint.

Exercise 2.11 *Say we want to minimize the following expression over u, v which both satisfy homogeneous boundary conditions at all boundaries*

$$\frac{1}{2}(u, -\Delta u) - (f_1, u) + \frac{1}{2}(v, -\Delta v) - (f_2, v)$$

subject to the constraint $u_x + v_y = 0$. What equations are to be solved after minimization? If you did it right you found the Stokes equations for incompressible fluid flow.

2.3.3 Setup of FE code

The assembly process, connectivity array, element matrices, stiffness matrix, static condensation.

2.4 Properties

2.4.1 Some matrix properties

There exist a few useful tools to detect important properties of matrices occurring in discretizations. For instance to determine whether a matrix is non-singular.
We start off with a few definitions.

Definition 2.1 (Spectrum) *The spectrum is the set of all eigenvalues of a matrix A and denoted by $\sigma(A)$.*

It is obvious that all eigenvalues are in a disc with the origin as center and the radius the biggest eigenvalue. This leads to the definition

Definition 2.2 (Spectral radius) *The spectral radius $\rho(A)$ is $\max\limits_{\lambda \in \sigma(A)} |\lambda|$.*

Definition 2.3 (Similarity) *Two matrices A and B are called similar if there exists a non-singular matrix Q such that $B = Q^{-1}AQ$.*

The operation with Q on A is called a *similarity transformation.*

Theorem 2.4 (Similar matrices) *Two similar matrices have the same spectrum and spectral properties.*

Definition 2.5 (Permutation matrix) *A matrix P is an identity matrix in which the rows are permuted.*

An example of a permutation matrix is

$$P = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

Premultiplication of A with P enforces a permutation of the rows of A. In the case of the example all rows shift down one and the last row becomes the first. Similarly, postmultiplication enforces a permutation of the columns.
The transformation PAP^T results in a mutual permutation where the rows and columns are permuted in the same way. This ensures that a diagonal of the original is also a diagonal element of the permuted matrix. It also holds that $P^{-1} = P^T$, hence the special transformation is a similarity transformation.

Definition 2.6 (Irreducibility) *A matrix A is called irreducible if there is no permutation matrix P such that*

$$PAP^T = \begin{bmatrix} F & G \\ O & H \end{bmatrix} \tag{2.19}$$

where F and H are square matrices and O a zero matrix.

A is *reducible* if it is not irreducible. In that case the system $A\mathbf{x} = \mathbf{b}$ can be split into two or more systems that can be solved one after another.

Theorem 2.7 (Gerschgorin) *Let A be an arbitrary (complex) matrix of order N and let every row define a radius by*

$$\Lambda_i = \sum_{\substack{j=1 \\ j\neq i}}^{N} |a_{ij}|, \qquad i = 1, 2, \ldots, N$$

then the eigenvalues λ of A are in the union of the Gerschgorin-discs

$$|z - a_{ii}| \leq \Lambda_i, \qquad i = 1, 2, \ldots, N$$

Proof. Let λ be an eigenvalue of A with corresponding eigenvector $\mathbf{x}$. We normalize $\mathbf{x}$ such that for the biggest element is holds that $|x_r| = 1$. From the row corresponding to this element in $A\mathbf{x} = \lambda\mathbf{x}$ it follows that

$$a_{rr}x_r + \sum_{\substack{j=1 \\ j\neq r}}^{N} a_{rj}x_j = \lambda x_r$$

rearranging and taking norms we have

$$|\lambda - a_{rr}| = |\sum_{\substack{j=1 \\ j\neq r}}^{N} a_{rj}x_j| \leq \sum_{\substack{j=1 \\ j\neq r}}^{N} |a_{rj}||x_j| \leq \sum_{\substack{j=1 \\ j\neq r}}^{N} |a_{rj}| = \Lambda_r \tag{2.20}$$

Hence λ is in the disc corresponding to the row where the eigenvector is biggest. From this it follows that any λ is in a Gerschgorin-disc. ■

A corollary from this theorem is that for an arbitrary matrix A of order N it holds that

$$\rho(A) \leq \max_i \sum_{j=1}^{N} |a_{ij}| = \|A\|_\infty \tag{2.21}$$

and because A^T has the same eigenvalues as A it also holds that

$$\rho(A) \leq \max_j \sum_{i=1}^{N} |a_{ij}| = \|A\|_1 \tag{2.22}$$

Theorem 2.8 (Taussky) *For an irreducible matrix A of order N it holds that a point λ on the boundary of the uninion of all Gerschgorin-discs can only be an eigenvalue of A if it is on the boundary of each disc (and the corresponding eigenvector has components of equal magnitude)*

Proof. Suppose that the eigenvalue λ of A is on the boundary of the union of Gerschgorin-discs. This means that for each disc λ is on its boundary or outside of it. Hence, using the notation of the previous theorem

$$|\lambda - a_{ii}| \geq \Lambda_i, \; i = 1, 2, \ldots, N$$

Combining with (2.20) we see that this is only possible if we have equality, hence

$$\sum_{\substack{j=1 \\ j \neq r}}^{N} |a_{rj}||x_j| = \sum_{\substack{j=1 \\ j \neq r}}^{N} |a_{rj}|$$

Which is only possible if $|x_s| = 1$ for all s for which $a_{rs} \neq 0$. So the eigenvalue must be on the boundary of the disc where the eigenvector is biggest and moreover the eigenvector must have values of equal magnitude for all its components that are used on this row. The last means that we could also have taken the row s in stead of the row r in (2.20), and that also here we will find equality as above and that the elements of the eigenvector used in this row should be of equal magnitude 1. Due to the fact that the matrix is irreducible we can reach every row in the matrix, herewith proving that the eigenvalue should be on the boundary of all the discs and that the magnitude of all the components of the corresponding eigenvector should be equal. ■

Usually we use this theorem to show that a matrix is nonsingular. However, if we have to deal with a symmetric real matrix, then we know that all eigenvalues and eigenvectors are real. So if the circles go through one point on the boundary of the union, then this point is an eigenvalue if the components of the eigenvector can be chosen plus or minus one. If not one vector of such type is an eigenvector, the point is still not an eigenvalue.

Exercise 2.12 *Show that the discretization of the Laplace operator on some domain with Neumann boundary conditions using the standard five-point stencil has an eigenvalue 0, and that it has not as soon as at one point we have a Dirichlet condition.*

We can use this to show that the class of weakly diagonally dominant matrices are non-singular. This is defined as follows

Definition 2.9 (Weak diagonal dominance) *A matrix A of order N is called weakly diagonally dominant when*

$$|a_{ii}| \geq \sum_{\substack{j=1 \\ j \neq i}}^{N} |a_{ij}| \quad \textit{for} \quad i = 1, 2, \ldots, N$$

and the $>$-sign should hold for at least one of the equations.

Theorem 2.10 *If A is an irreducible, weakly diagonally dominant matrix of order N, then A is non-singular.*

Proof. There is at least one disc for which 0 is outside. Hence according to Taussky's theorem 0 cannot be an eigenvalue. Hence the matrix is non-singular. ■

Definition 2.11 (Positive Definiteness) *A matrix is A is called positive definite if* $(x, Ax) > 0$ *for all nonzero* x

Theorem 2.12 *If A is real symmetric, then positive definiteness equivalent to having positve eigenvalues.*

Theorem 2.13 *If A has positive diagonal elements and moreover is symmetric, irreducible, and weakly diagonally dominant, then it is positive definite.*

Proof. Proof The eigenvalues of a symmetric matrix are real and from the location of the Gershgorin discs we know that all eigenvalues are on the positive real axis. ■

In simulation also non-negativeness of solutions is an issue. For instance, we would like to keep a concentration positive during the computations. It is clear that when we multiply a vector with positive elements by a matrix with positive elements then the the outcome is also positive. But what if we want to solve a system where the right-hand side is positive, for which matrices is the solution positive? Let us first define the class of non-negative matrices.

Definition 2.14 (Non-negativeness) *A matrix A is non-negative if* $(A \geq 0)$ *if each element is non negative, and A is positive* $(A > 0)$ *if each element is positive.*

Definition 2.15 (Monotony) *A matrix A is monotone if its inverse exists and is non-negative*

Theorem 2.16 *A matrix A is monotone if and only if from* $A\mathbf{x} \geq \mathbf{0}$ *it follows that* $\mathbf{x} \geq 0$.

Proof. Let A be monotone and $\mathbf{y} = \mathbf{Ax} \geq \mathbf{0}$. Then it follows straight-forwardly that $\mathbf{x} = \mathbf{A^{-1}y} \geq \mathbf{0}$. Now the other way around. First we show nonsingularity. Let $A\mathbf{x} \geq \mathbf{0} \;\Rightarrow\; \mathbf{x} \geq \mathbf{0}$ and suppose $\mathbf{z}$ is a nonzero singular vector so $A\mathbf{z} = \mathbf{0}$. Then from our outset, z must have non-negative elements. However $-z$ is also a singular vector and hence also the elements of this vector should be non-negative. From this it follows that $z = 0$, so the matrix is non-singular. Furhermore the inverse of A is the solution of the system $AX = I$, which just means that the columns of the inverse follow from the solution of $Ax = e_i$ for $i = 1, ..., N$ where e_i is a unit vector. Since the unit vector is non-negative the columns of the inverse of A are non-negative and hence $A^{-1} \geq 0$.■

Definition 2.17 (M-matrix) *A matrix A is an M-matrix if A is monotone and* $a_{ij} \leq 0$ $(\forall i \neq j)$.

Theorem 2.18 *An M-matrix has positive diagonal elements.*

Proof. Let $A^{-1} = (b_{ij})$ then all $b_{ij} \geq 0$. From $AA^{-1} = I$ it follows that

$$a_{ii}b_{ii} + \sum_{\substack{j=1 \\ j\neq i}}^{N} a_{ij}b_{ji} = 1, \qquad i = 1, 2, \ldots, N$$

In the sum all $a_{ij} \leq 0$ and all $b_{ji} \geq 0$, hence

$$a_{ii}b_{ii} = 1 + \sum_{\substack{j=1 \\ j\neq i}}^{N} |a_{ij}b_{ji}| \geq 1, \qquad i = 1, 2, \ldots, N$$

Since $b_{ii} \geq 0$ it follows from this that $a_{ii} > 0, \quad i = 1, 2, \ldots, N$. ■

Theorem 2.19 *An irreducible, weakly diagonally dominant matrix A with positive diagonal elements and $a_{ij} \leq 0$ $(\forall i \neq j)$ is an M-matrix.*

Proof. From Theorem 2.10 we know that A is non-singular, hence it only remains to show that $A^{-1} \geq 0$. We write $A = D - B$ met D een diagonaalmatrix en de diagonaal van B gelijk aan nul. From application of Tausky's theorem to $D^{-1}B$ it follows that $\rho(D^{-1}B) < 1$. Hence the sum

$$S = \sum_{k=0}^{\infty} (D^{-1}B)^k$$

converges and

$$S = (I - D^{-1}B)^{-1}$$

From $D^{-1} \geq 0$ and $B \geq 0$ it follows that $D^{-1}B \geq 0$, hence, also $S \geq 0$. From

$$A^{-1} = (I - D^{-1}B)^{-1}D^{-1} = SD^{-1}$$

it finally follows that also $A^{-1} \geq 0$. ■

```
== External Links ==
* http://en.wikipedia.org/wiki/Similar_matrix
* http://mathworld.wolfram.com/ReducibleMatrix.html
* http://en.wikipedia.org/wiki/Gershgorin_circle_theorem
* http://planetmath.org/encyclopedia/MMatrix.html
```

2.4.2 Maximum principles and monotony

Consider the Laplace equation $\Delta u = 0$. Then from Gauss' theorem we have

$$\int_{\Gamma} u_n d\Gamma = 0$$

where n is the outward normal. This says that if $u_n \neq 0$ on the whole boundary that it must be on parts positive and on other parts negative. This holds for the boundary of any volume in the definition domain of the Laplace equation. Hence if we take this volume very small around a point. Then we see that this point cannot be an extremum. Since in some directions u is increasing and in others it is decreasing. Since this can be any point in the intenal of the domain this means that we cannot have an extremum inside the domain hence it must be on the boundary.
Consider now the Poisson equation $-\Delta u = f$ with $f \leq 0$. Then from Gauss' theorem we have

$$\int_\Gamma u_n d\Gamma \geq 0$$

If we now contract the volume to a point we see that this point can be a minimum since u may be increasing in all directions away from it. However we cannot have a maximum in this case because then u_n must be negative in any direction. So in this case we cannot have a maximum in the internal domain. Similar if $f \geq 0$ we cannot have a minimum in the internal domain.
Next consider the case $-\Delta u + qu = f$ where $q \geq 0$ and $f \leq 0$ then we have that

$$\int_\Gamma u_n d\Gamma \geq \int_\Omega qud\Omega$$

Now we again we contract the volume and see using the previous that if u is positive that no maximum is possible. Similar for positive f no minimum is possible.
Now we consider a generaal discrete case:

$$C_P\, U_{i,j} - C_W U_{i-1,j} - C_S U_{i,j-1} - C_E U_{i+1,j} - C_N U_{i,j+1} = f_P$$

where $C_P \geq C_W + C_S + C_E + C_N$ and all the coefficients are non-negative. At a Neumann boundary some of them may be zero. This can be rewritten to

$$\begin{array}{rl} & (C_P - C_W - C_S - C_E - C_N)U_{i,j} \\ + & C_W(U_{i,j} - U_{i-1,j}) + C_S(U_{i,j} - U_{i,j-1}) + C_E(U_{i,j} - U_{i+1,j}) + C_N(U_{i,j} - U_{i,j+1}) \\ = & f_P \end{array}$$

which is a discrete analogue of the Gauss' theorem. Now consider again three cases.
If $f_P = 0$ for all internal points of the computational domain and C_P is equal to the sum of the other coefficients, then the first coefficient in the equation cancels and we have that $U_{i,j}$ cannot be an extremum for any internal point. Since if one of differences is positive then another must be negative.
If $f_P \leq 0$ for all internal points of the computational domain and C_P is equal to the sum of the other coefficients, then again the first coefficient in the equation cancels. Reasoning the same as in the continuous case we cannot have a maximum in the internal domain. Similar we cannot have a minimum if $f_P \geq 0$
If $f_P \leq 0$ for all internal points of the computational domain and C_P is bigger than the sum of the other coefficients, then we have a positive coefficient in front of $U_{i,j}$. And reasoning the same as in the continuous case now leads to the observation that no positive maximum is possible; similarly no negative minimum if $f_P \geq 0$.

The consequence of this is that if on a Diriclet boundary we prescribe a positive value and furthermore if the right-hand side is positive, then the solution must be positive. Since if it would become negative it must have a negative minimum in the internal or at the Neumann boundary, which conflicts with the above assertion.
Note that we have already seen the same conclusion for monotone matrices. So these properties are related.

2.5 Time dependent equations

2.5.1 Method of lines

In the following sections we will treat a number of methods to solve systems of ordinary differential equations (ODEs). In this section, we will show how we transform a PDE into a system of ODEs, by looking to the one dimensional heat equation

$$\frac{\partial u}{\partial t} = a\frac{\partial^2 u}{\partial x^2} + f, \quad a > 0,\ t > 0,\ 0 < x < 1 \tag{2.23}$$

The initial condition is $u(x,0) = \phi(x)$ and for convenience we will assume homogeneous boundary conditions $u(0,t) = u(1,t) = 0$.
In the method of lines we first discretize in space. We will treat the finite difference and finite element method subsequently.

Finite difference method We partition the x-interval in m equal parts, hence $h = 1/m$ and $x_i = ih,\ i = 0, 1, \dots m$. According to (2.4) it holds for sufficient smooth u

$$u_{xx}(x,t) = \frac{u(x+h,t) - 2u(x,t) + u(x-h,t)}{h^2} - \tfrac{1}{12}h^2 u_{xxxx}(x,t) + O(h^4)$$

Now, if we use the notation $u_j(t) = u(x_j,t)$, then we get after substitution in (2.23) for $x = x_j,\ j = 1, 2, \dots, m-1$

$$\frac{d}{dt}u_j(t) = a\frac{u_{j+1}(t) - 2u_j(t) + u_{j-1}(t)}{h^2} - \tfrac{1}{12}ah^2 u_{xxxx}(x_j,t) + O(h^4) + f_j(t)$$

After discarding the local discretization errors we arrive at the system of ODEs

$$\frac{d}{dt}U_j = a\frac{U_{j-1} - 2U_j + U_{j+1}}{h^2} + f_j(t), \qquad j = 1, 2, \dots, m-1$$

In which the functions $U_j(t)$ approximate $u_j(t)$, hence they approximate the solution $u(x,t)$ along the lines $x = x_j,\ j = 1, 2, \dots, m-1$. This is why this approach got the name *method of lines*, sectie 6.9).
We can also put the difference equations in matrix vector form. For that we introduce the vector

$$\mathbf{U(t) = [U_1(t), U_2(t), \dots, U_{m-1}(t)]^T}$$

and similarly a vector $\mathbf{F}$ and the $(m-1)\times(m-1)$ matrix

$$A = \frac{a}{h^2}\begin{bmatrix} -2 & 1 & & & \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ & & & 1 & -2 \end{bmatrix}$$

Now we can write the system of ODEs as

$$\frac{d}{dt}\mathbf{U} = \mathbf{AU} + \mathbf{F} \tag{2.24}$$

Finite element method The only difference with the approach we followed for the elliptic equations is that now the coefficients in the sum of the basis functions will depend on t. So we write $u_N(x,t) = \sum_{j=1}^N c_j(t)\phi_j(x)$. In the Galerkin approach we plug this into the equation and test with the ϕ_i. This yields here

$$\sum_{j=1}^N [\frac{d}{dt}c_j(t)(\phi_i,\phi_j) + ac_j(\phi_i',\phi_j') - (\phi_i, f)] = 0$$

which in system form can be written as

$$M\frac{d}{dt}\mathbf{c} = \mathbf{Ac} + \mathbf{F} \tag{2.25}$$

where here $M_{ij} = (\phi_i,\phi_j)$, $A_{ij} = (\phi_i',\phi_j')$ and $F_i = (\phi_i, f)$. The matrix M is called the *mass matrix*, while A is as before the stiffness matrix.
It is interesting to see that in both cases we have a similar system

$$M\frac{d}{dt}\mathbf{c} = \mathbf{Ac} + \mathbf{F}, \qquad \mathbf{c(0)} = \mathbf{c_0} \tag{2.26}$$

where in the finite difference case M is just the identity.

2.5.2 Stability investigation

In order to make a judicious choice for a method we must consider the stability problem for (2.26). In this case it is enough to consider what happens if we perturb the initial value a bit. So let us denote the solution following from the perturbation by $\hat{c}(t)$ which has initial condition $\hat{c}(0) = c_0 + \varepsilon$. Since $\hat{c}$ also satisfies the ODEs exactly, we can simply subtract the two systems to obtain a system for the difference $e(t) = \hat{c}(t) - c(t)$, which assumes the form

$$M\frac{d}{dt}\mathbf{e} = \mathbf{Ae}, \qquad \mathbf{e(0)} = \varepsilon \tag{2.27}$$

We can now use standard theory from ODEs to solve this system. We first try to break it into independent scalar equations by setting $e = \exp(\lambda t)v$ where v does not depend on time anymore. This leads to the generalized eigenvalue problem

$$\lambda M v = Av$$

In our case M is non-singular and therefore the eigenvalues are just the standard eigenvalues of $M^{-1}A$. Once, we have found the eigenpairs (λ_i, v_i) we build the matrix $V = [v_1, ..., v_N]$ and use it to bring the original equation to diagonal form, yielding the scalar equations

$$\frac{d}{dt}\hat{e}_i = \lambda_i \hat{e}_i$$

with $\hat{e} = V^{-1}e$. The initial condition transforms into the condition $\hat{e}(0) = V^{-1}\varepsilon$. From this scalar equation we can study the stability of our ODE. Many physical systems are dissipative. If it is given some initial condition the system turns into rest after a while. This means that the eigenvalues in such a system have negative real parts. We like numerical methods to have the same property. In a numerical method we are just doing some approximation to the time derivative and therefore the reduction to scalar equations can be done precisely in the same way using the same V. So we find the equations

$$DDT\hat{E}_i = \lambda_i \hat{E}_i, \quad \text{with } \hat{E}_i(0) = \hat{e}_i(0)$$

where DDT is a short we use here to denote the discretization of the time derivative which is not specified further. This means that for stability of the numerical method we just can look to the equation

$$\frac{d}{dt}u = \lambda u$$

which is called the *test equation*, and where λ will run through the eigenvalues of the eigenvalue problem given above.

Localization of eigenvalues

For the choice of the method we would like to know where the spectrum of the problem is. For this we can use two paths: the matrix method and the difference method.

Matrix method In this case we start from the matrix. We inspect the following

Symmetry If the matrix is symmetric we know that the eigenvalues are real, if it is skew-symmetric and real $A^T = -A$ the eigenvalues are purely imaginary.

Positive definiteness If the matrix is following from a finite element discretization of a positive definite PDE problem then the discretization inherits this property. If it follows from a finite difference equation then we can use Gershgorin's and Taussky's theorem to study the positive definiteness. This usually only works for M matrices. In the more general case, for instance if discretizations of higher accuracy are used it becomes more sophisticated.

Note that for a real matrix positive definiteness has only to be considered for the symmetric part of the matrix, because $(x, Ax) = (x, \frac{1}{2}(A + A^T)x) + (x, \frac{1}{2}(A - A^T)x)$, where the second term is purely imaginary.

Spectral radius The spectral radius can be found using Gershgorin's theorem or by the infinity norm of the matrix.

If we apply this to the matrix above we have that both M and A are symmetric. For the finite difference case we can show by Taussky's theorem that $-A$ is positive definite. For the finite element case it follows from the outset that both M and $-A$ are positive definite. The finite difference case we find with both mentioned approaches that the spectral radius of $-A$ is bounded by $4a/h^2$. For the finite element case in order to bound the spectral radius of $M^{-1}A$ is more sophisticated. Since M is positive definite we can bound this by the ratio of the spectral radius of A and the minimum eigenvalue of M. The latter is found by the Gershgorin theorem.

Difference method In the difference method we take the difference operator according to A as starting point. If these difference equations are the same for all the grid points we can try to solve the corresponding eigenvalue problem. Now if we assume that there are no boundary conditions, then the Fourier component $\exp(ij\theta)$ is an eigen(grid)function of the difference operator. Let's do this for the above finite difference example. So we want to solve the eigenvalue problem

$$a\frac{u_{j+1}(t) - 2u_j(t) + u_{j-1}(t)}{h^2} = \lambda u_j$$

Now we plug in $u_j = \exp(ij\theta)$ and find

$$a\frac{e^{i\theta} - 2 + e^{-i\theta}}{h^2}\exp(ij\theta) = \lambda\exp(ij\theta)$$

After some algebraic manipulations on the left-hand enumerator we find

$$\lambda = -4a\frac{\sin^2(\theta/2)}{h^2}$$

And also here we find that the eigenvalue is real and in the interval $[-4a/h^2, 0]$. This approach is less rigorous then the matrix method, since we have neglected the boundary conditions. Nevertheless in many cases it gives a good indication of the location of the eigenvalues in the complex plane.

2.5.3 Some time integrators

In this section we show a number of time integrators that can solve a system of ODEs of the form $du/dt = f(t, u)$ with an initial condition given. In all cases we will define a grid in time direction with stepsize Δt.

Forward Euler method The forward Euler method is defined by

$$w_{n+1} = w_n + \Delta t f(t_n, w_n)$$

where w_0 is given. Using Taylor series one can show that this method is first-order accurate. As indicated above, for stability we can apply it to the test equation, which gives

$$w_{n+1} = w_n + \Delta t\lambda w_n = (1 + \Delta t\lambda)w_n$$

Note that this is a recurrence and hence

$$w_n = (1 + \Delta t\lambda)^n w_0$$

Since for stability we want that an initial perturbation damps out again, we want that w_n tends to zero if n tends to infinity. This is the case only if

$$|1 + \Delta t\lambda| < 1 \tag{2.28}$$

For the finite difference example above we know that the eigenvalues are in the interval $[-4a/h^2, 0]$. Then this condition is satisfied if

$$\Delta t < \frac{h^2}{2a}$$

Hence we get a time step restriction which can become quite severe for small h.
In general one defines $z = \Delta t\lambda$ and studies for which z

$$|1 + z| < 1$$

For equality we find in the complex plane a circle with center -1 and radius 1. The inequality holds for all z in this circle. This part is called the *region of absolute stability* of the Euler method.

Exercise 2.13 *Show that this method is not suited for problems with purely imaginary eigenvalues.*

The forward Euler method is an *explicit method* since we can just fill in a known w_n in f, do an addition and we have w_{n+1}. This contrasts with an *implicit method* where we will have the unknown w_{n+1} in f, which in general leads to the solution of a nonlinear system of equations. The big difference between explicit and implicit methods is that the time step for explicit methods is always bounded in a form similar to that for the Euler method, while for implicit methods this need not be the case. This may make it worthwhile to use an implicit method.

Backward Euler The backward Euler method is an implicit method which has the form

$$w_{n+1} = w_n + \Delta t f(t_{n+1}, w_{n+1})$$

and is like the Forward Euler method first-order accurate.

Exercise 2.14 *Show that here the stability analysis leads to* $|1 - z| > 1$ *and draw the region of absolute stability in this case.*

From the region of absolute stability we observe that this method has no restriction on the time step for our model problem. In fact it will not have a restriction for any problem with eigenvalues from which the real part is negative. Methods with such a property are called *A-stable.*
This method is the simplest example of the *Backward Differentiation Formulas (BDFs)* where the f is always and only evaluated at t_{n+1} and for the discretization of the time derivative values in the past are used, which explains the name. In general BDF(k) uses k values of w in the past and has order of accuracy k.

Exercise 2.15 *Search for plots of the region of stability of the BDF(k) methods in books or on the internet. For which values of k are they A-stable and for which values can they solve our model parabolic problem without a restriction on the time step?*

Trapezoidal method This method has the symmetric form

$$w_{n+1} = w_n + \frac{\Delta t}{2}[f(t_n, w_n) + f(t_{n+1}, w_{n+1})]$$

which causes that this method is second-order accurate in time.

Exercise 2.16 *Show that the stability analysis leads to* $|2 - z| > |2 + z|$ *and that also this method is A-stable.*

The trapezoidal method is in PDE context often called *Crank-Nicholson method.*

Theta methods The theta method is a generalization of the trapezoidal method and has the form

$$w_{n+1} = w_n + \Delta t[(1 - \theta)f(t_n, w_n) + \theta f(t_{n+1}, w_{n+1})]$$

Observe that for $\theta = 0, 1/2, 1$ we get the forward Euler, trapezoidal method, and the backward Euler, respectively. It is A-stable for $\theta \geq 1/2$.

Exercise 2.17 *Show that the factor occurring in the recurrence relation for the trapezoidal method is tending to one for z tending to infinity.*

It is useful in adding some damping to the trapezoidal method, which it lacks for example in our model problem when the time step is chosen big.
There exists also a variant of the theta method called the implicit theta method

$$w_{n+1} = w_n + \Delta t[f(t_{n+\theta}, (1 - \theta)w_n + \theta w_{n+1})]$$

This method only differs from the first one for nonlinear problems. Hence the stability analysis is equal. However, it is known to be slightly better for non-linear problems.

Von Neumann analysis Instead of analysing the space direction (by trying to localize the spectrum) and afterwards considering the stability of the time integrator. One could of course also discretize both at the same time. If one applies the Fourier analysis to the space discretization one defines the *amplification factor* ρ which is in the case of our example problem and using the Euler method nothing more then $\rho(\theta) = 1 - \Delta t 4a\frac{\sin^2(\theta/2)}{h^2}$ and we require in line with the analysis above (2.28) that the magnitude of the amplification factor should be less than one. This approach is called the Von Neumann analysis.

Algebraic Differential Equations If in a system of PDEs one or more of the equations do not contain a time derivatives, then these equations form a constraint to the solution. After space discretization they become an algebraic constraint and then we arrive at so-called algebraic differential equations. For such equations special methods exist. In the above mentioned methods it is best to apply the constraint immediately to the new time level.

Chapter 3

Solution of sparse systems

PDEs discretized by finite elements or finite differences often lead to very sparse systems of equations. Since the solution of systems is the bottleneck in many simulations, it is worthwhile to try to exploit the sparsity. Here, the ultimate goal is that the amount of work will be proportional to the number of unknowns. First we treat the solution of linear systems and next we consider the nonlinear case.

3.1 Direct methods for sparse linear systems

In general there are two approaches *direct methods* in which the solution is solved to machine accuracy in one step and *iterative methods* in which the solution is approximated in a number of steps, in each of which the accuracy is improved, until a user set tolerance on the accuracy is met. In this chapter we will look to the direct approach. Consider the basic problem

$$A\mathbf{x} = \mathbf{b} \tag{3.1}$$

where A is a matrix of order N. The usual way to solve this is with Gaussian elimination with pivoting (GEP). Here we construct a permutation matrix P, a lower triangular matrix L and an upper triangular matrix U such that $PA = LU$. The permutation matrix P shows which rows of A are permuted due to the pivoting. The amount of work to construct this factorization is for large N about $\frac{2}{3}N^3$ flops. If A is symmetric this can be exploited. In that case we can find a factorization of the form LDL^T where L has ones on its diagonals and D is a diagonal matrix with 1×1 and 2×2 blocks on the diagonal. This halves the amount of work. If A is also positive definite we can make a Cholesky factorization LL^T. It is known that for weakly diagonally dominant matrices, hence for M-matrices, there is no need for pivoting. Neither this is needed for symmetric positive definite matrices.

The amount of work and *complexity*, i.e. how the amount of work behaves as a function of the number of unknowns, depends on the ordering of the matrix. In order to keep the work low we should try to keep the fill low. The *fill* are the elements occurring in the L and U which where not there in the original matrix. Hence one is looking for *fill-reducing orderings*.

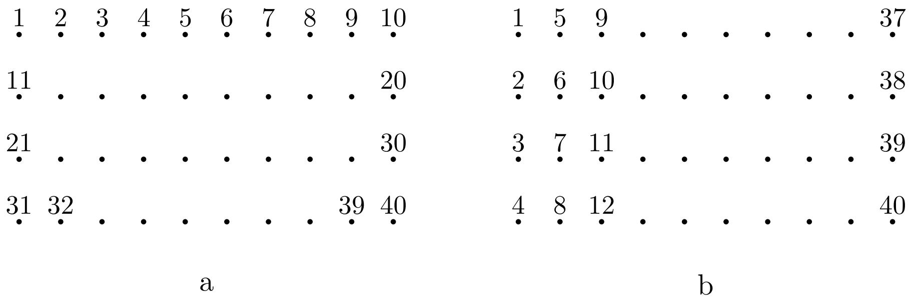


Figure 3.1: (a) Row wise ordering, (b) column wise ordering

Let us first consider the structure of the matrix due to the discretization of the Laplace equation on a rectangular domain Ω using Dirichlet boundary conditions. The sides of the domain are parallel to the x- and y-axis. The domain Ω is covered by a uniform grid with in both directions an equal mesh width h and in x- and y-direction N_x and N_y internal grid points, respectively. We number the internal grid points in *lexicographical ordering* as depicted for the case $N_x = 10$ and $N_y = 4$ in Fig. 3.1. If we use the standard 5-point difference molecule (2.10) then in the i^{th} row of $AU = b$ we have a connection to U_{i-N_x} and U_{i+N_x}, but not to any U_j with $j < i - N_x$ or $j > i + N_x$. This means that the *bandwidth* of the matrix is $2N_x + 1$. It is easy to show that without pivoting there will be no fill outside the band. This bandwidth can be a lot less than that of a full matrix. The amount of work is now approximately $2N_x^2N$ flops. If $N_x \gg N_y$ then is advantageous to take the ordering as in Fig. 3.1.a, because in that case the bandwidth is $2N_y + 1$. The original matrix has also a lot of zeros within the band, but during the elimination process this is lost.

Exercise 3.1 *Make a picture of the matrix for both orderings.*

A *symmetric reordering*, which we are considering here, of a matrix can be described in terms of a permutation matrix: PAP^T. A few symmetric orderings by which we can reduce the fill are the following

Reversed Cuthill-McKee The idea behind standard Cuthill-McKee is to minimize the bandwidth. We saw already that this is advantageous if one uses lexicographical orderings. It appeared that a reversion of the ordering created slightly better results in some cases. However, for our problem this would not make any difference.

(Approximate) Minimum degree This is based on the fact that a row with a low number of elements cannot produced much new fill. Since, the number of elements per row varies during the elimination, one tries to pick the row with the lowest number of elements as pivot row. More details can be found in ([26], 1967) and [8].

Table 3.1: Amount of work and number of nonzeros in L for various orderings

Numbering	flops/1000				nnz(L)/1000			
$N =$	100	400	1600	6400	100	400	1600	6400
Random	35	968	78944	4477865	1.5	14	216	3110
Lex. graphical	11	165	2603	41302	1.0	8	64	512
Rev. Cuthill-McKee	7	96	1410	21510	0.8	6	45	351
Checkerboard	6	85	1299	20559	0.7	5	36	270
Nested dissection	7	78	804	7637	0.8	5	28	153
Minimum degree	5	53	590	7337	0.7	4	22	126

Table 3.2: Order of operations of Nested Dissection on Poisson problem on a hypercube ($N = n^d$ unknowns

	1D		2D		3D		dD	
factorization	n	N	n^3	$N\sqrt{N}$	n^6	N^2	$n^{3(d-1)}$	$N^{3(d-1)/d}$
storage	n	N	$n^2 \log_2(n)$	$N \log_2(N)$	n^4	$N^{4/3}$	$n^{2(d-1)}$	$N^{2(d-1)/d}$

Nested dissection This is a divide and conquer strategy ([10], 1973). Let us describe it shortly because it is also related to domain decomposition approaches. In this technique one simply starts off by splitting the domain in about 2 equal parts, which are separated by a number of unknowns that have a connection to both domains. An unknown in one of the domains does not have a direct connection to any unknown in the other domain; it is only connected via an unknown on the separator. The unknowns on the separator are put last in the vector of unknowns. This process is repeated on each of the two domains recursively.

To illustrate the influence of ordering on the amount of work and the number of nonzeros in the factorization we consider in Table 3.1 a Cholesky factorization for the described Laplace problem for a number of grid resolutions $N_x = N_y = 10,\ 20,\ 40$ and 80. These results show that it pays off to use a fill-reducing ordering. The problems shown are still quite small For bigger problems Nested Dissection will eventually do a better job than minimum degree. In Table 3.1 we show how the complexity behaves for nested dissection on a Poisson problem. To improve accuracy one usually employs *iterative refinement.* Here the residual $r = b - A\hat{x}$ is computed, where $\hat{x}$ is the solution found from the exact solve. Next we compute a correction Δx from $LU\Delta x = r$ and update $\hat{x}$ by adding Δx. This can be repeated a few times till r is small enough. One uses higher precision to compute the residual in order to avoid loss of significant digits. Some concluding remarks.

- Direct methods are robust. So any problem can be solved by it.
- An LU-factorization is expensive in time. The storage of the factorization is still moderate in the 2 and 3D case.

- Solution time of $LUx = b$ is proportional to the storage and therefore comparatively cheap with respect to factorization. Hence, if the matrix is constant and the right-hand side changes then reusing the factorization makes it a very attractive alternative to iterative methods.
- In 2D direct methods are usually faster than iterative methods for systems up to 10.000-100.000 unknowns. In 3D the break even occurs for much smaller systems.
- Direct methods are quite well developed nowadays and usually available in mathematical kernels coming with compilers. Sometimes the user can choose the ordering, but usually it is set to nested dissection or (approximate) minimum degree. Some implementations however maybe simply faster or better adapted to the hardware (cache, distributed memory). Therefore, if possible one could try a few and just pick the fasted one.

```
== External Links ==
* http://en.wikipedia.org/wiki/Degree_%28graph_theory%29 degree
* http://en.wikipedia.org/wiki/Sparse_matrix
* http://en.wikipedia.org/wiki/Cuthill-McKee_algorithm
* http://en.wikipedia.org/wiki/Minimum_degree_algorithm
```

3.2 Handling nonlinear equations

To solve nonlinear equations Newton's method and its variants is by far the most popular approach. In general a Newton method to solve the equation $f(x) = 0$ has the following form given an x_0 perform the iteration,

1. evaluate $f(x_k)$,
2. build an (approximate) Jacobian matrix $J(x_k)$ of f,
3. solve the linear system $J(x_k)\Delta x_k = f(x_k)$,
4. update x_k: $x_{k+1} = x_k + \alpha_k \Delta x_k$.

We discuss each step shortly. In the first step one should be aware of the fact that the attainable accuracy of a Newton method is determined by the accuracy by which one can evaluate f. Try to formulate the problem such that loss of significant digits is as small as possible. Building a Jacobian matrix in the second step may be expensive, therefore often the Jacobian is kept fixed for a number of Newton steps until the convergence deteriorates. Another way is to evaluate the Jacobian only on a subspace. In the third step we have to solve usually a big system, for which the building of a factorization for the Jacobian matrix is quite expensive. One could decide here to reuse the factorization of the Jacobian as long as the convergence does not deteriorate. As long as we are using a direct method to solve the linear system this is equivalent to keeping the Jacobian fixed for a number of steps, but in an iterative procedure, with incomplete factorizations one could choose to update the Jacobian but not its factorization. The fourth step shows a *damping parameter* α_k. By choosing this parameter judiciously one can assure that $f(x_k)$ is becoming smaller in each step, which is not necessarily the case for $\alpha_k = 1$ in the standard Newton method. Hence, by introducing an α_k one can make a *locally convergent method globally convergent*.

Chapter 4

Continuation of steady states

The starting point is a given set of partial differential equations which can be written in operator form as

$$\mathcal{M}\frac{\partial \mathbf{u}}{\partial t} + \mathcal{L}\mathbf{u} + \mathcal{N}(\mathbf{u}) = \mathcal{F} \tag{4.1}$$

where $\mathcal{L}$, $\mathcal{M}$ are linear operators, $\mathcal{N}$ is a nonlinear operator, $\mathbf{u}$ is the vector of dependent quantities and $\mathbf{F}$ contains the forcing of the system. To get a well-posed problem, appropriate boundary conditions have to be added to this set of equations. A typical problem will be given in Section 4.5 and it also serves as a testcase for illustrating the methods.

The computational approach can be divided into three separate parts. First a discrete representation of the model equations has to be obtained through some kind of discretization procedure (Fig. 4.1). This leads to a set of ordinary differential equations

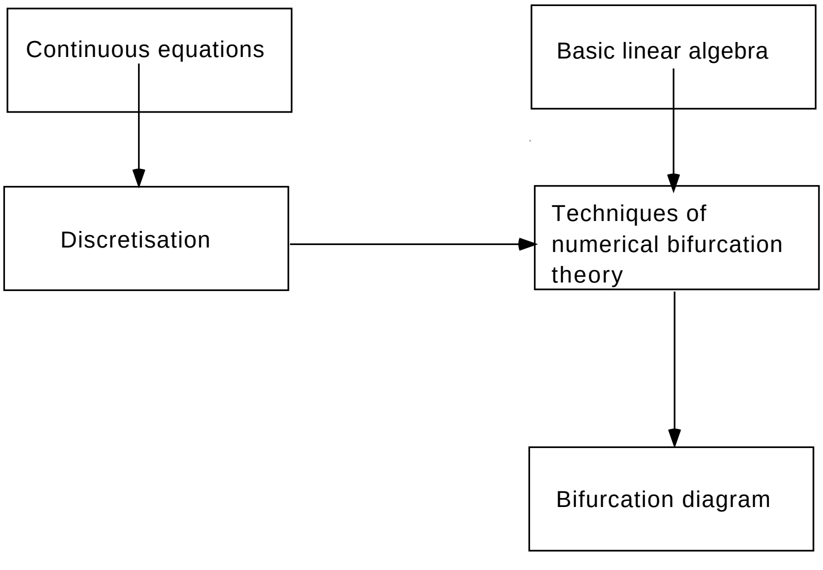


Figure 4.1: *Sketch of the scheme of the computational work involved to compute bifurcation diagrams.*

with or without algebraic constraints

$$\mathcal{M}_N \frac{d\mathbf{x}}{dt} + \mathcal{L}_N \mathbf{x} + \mathcal{N}_N(\mathbf{x}) = \mathcal{F}_N \tag{4.2}$$

where the subscript N denotes the space discretized variant and the vector x denotes the values of $\mathbf{u}$ on a grid in finite differences or the coefficients in the expansion in basisfunctions in Finite Elements. The second part of the computational work is to apply specific techniques of numerical bifurcation theory (Fig. 4.1). These are the same techniques which are used in the packages for the smaller dimensional systems, such as AUTO, MATCONT and CONTENT.
Usually, the scheme below is followed

(i) Determine the fixed points $\bar{\mathbf{u}}$ of the system of equations when parameters are changed, i.e., solve the problem

$$\mathcal{L}_N \bar{\mathbf{x}} + \mathcal{N}_N(\bar{\mathbf{x}}) = \mathcal{F}_N \tag{4.3}$$

This will be done using continuation methods which are presented in Section 4.1. Their description follows closely texts in [25] and [19].

(ii) When one is able to compute a branch of steady solutions in a control parameter, one wants to know whether a bifurcation point has been crossed, whether other branches exist and if yes, how they can be reached. Practical techniques to do so are provided in Section 4.2.

(iii) If a steady state is computed, one wants to assess its linear stability. With $\mathbf{x} = \bar{\mathbf{x}} + \tilde{\mathbf{x}}$, linearizing (4.1) around $\bar{\mathbf{x}}$ and separating $\tilde{\mathbf{x}}(t) = \hat{\mathbf{x}}\ e^{\sigma t}$ gives an eigenvalue problem of the form

$$(\mathcal{L}_N + \mathcal{N}_{Nx}(\mathbf{x}))\hat{\mathbf{x}} = -\sigma \mathcal{M}_N \hat{\mathbf{x}} \tag{4.4}$$

where $\mathcal{N}_{Nx}$ is the derivative of the operator $\mathcal{N}_N$ with respect to $\mathbf{x}$. The solution of these eigenvalue problems is discussed in Section 4.3

(iv) Finally, one wants to compute trajectories of the model under investigation, either in the regime where bifurcation behavior (and fixed points) are known, or to compute periodic orbits. As a spin-off of the methodology above, implicit time-dependent methods will be discussed in Section 4.4

As will turn out, an important part of the computational work is the solution of large linear systems of equations. The success of the latter methods mainly determines the dimension of the dynamical system which can be handled. Whereas for small dimensional dynamical systems robust direct techniques (section 3.1) can be used, for giant dimensional systems one must turn to sophisticated (and less robust) iterative techniques.

4.1 Pseudo-arclength continuation

To determine steady solutions of (4.2), we need to solve the set of nonlinear algebraic equations

$$\Phi(\mathbf{x},\lambda) = \mathcal{L}_N\mathbf{x} + \mathcal{N}_N(\mathbf{x}) - \mathcal{F}_N = 0 \tag{4.5}$$

where λ indicates a control parameter which appears in the operators $\mathcal{L}_N$ and/or $\mathcal{N}_N$ and/or $\mathcal{F}_N$.

For reasons which will be made clear below, it is advantageous to parametrize branches of solutions with an arclength parameter s as sketched in Fig. 4.2. A branch γ of steady solutions $(\mathbf{x}(s),\lambda(s)),\ s \in [s_a, s_b]$ is a smooth one-parameter family of solutions of (4.5). Since an extra degree of freedom is introduced by the arclength s, a normalization condition of the form

$$\Sigma(\mathbf{x}(s),\lambda(s),s) = 0 \tag{4.6}$$

is needed to close the system of equations. We thus end up to solve a system of nonlinear algebraic equations of dimension $N+1$ for the $N+1$ unknowns $(\mathbf{x}(s),\lambda(s))$. But where to start on the branch and how to choose the normalization?

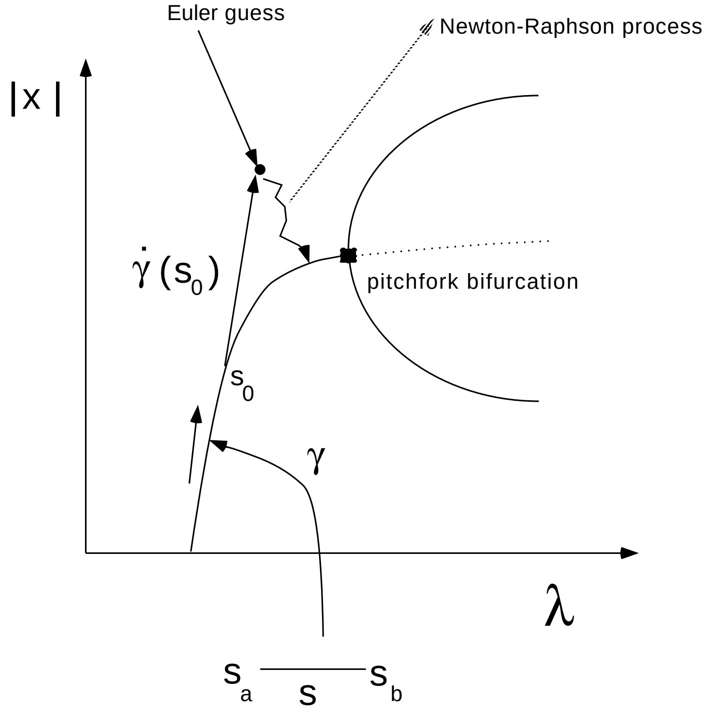


Figure 4.2: *Sketch of the parametrization of branches of steady solutions by an arclength parameter s and the tangent $\dot\gamma$ along the branch in a typical bifurcation diagram.*

In the more specialized literature [25,], several alternatives are described. In many applications, some trivial state can always be found, for example for zero forcing and/or a motionless solution. With respect to the normalization issue, we consider the geometry of the problem. Aim is to determine the range of a curve $\gamma\ : I \subseteq \mathbb{R} \rightarrow \mathbb{R}^{N+1}$, with

$\gamma(s) = (\mathbf{x}(s), \lambda(s))$ such that (4.5) is satisfied. Assuming that we now know, at some point s_0, a solution $(\mathbf{x}_0, \lambda_0)$, then the tangent space of the curve at $s = s_0$ is spanned by the vector $\dot\gamma(s_0) = (\dot{\mathbf{x}}(s_0), \ \dot\lambda(s_0))^T$ (Fig. 4.2). As a normalization condition, it turns out to be advantageous (the reason being the solution method in the next section) to take a normalization of the length of the tangent,

$$\dot{\mathbf{x}}_0^T \dot{\mathbf{x}}_0 + \dot\lambda_0^2 = 1 \tag{4.7}$$

In some applications, the initial tangent is analytically available. For example, in a problem where the motionless solution exists is a solution for all λ, we find $\dot{\mathbf{x}}_0 = 0$ and $\dot\lambda_0 = 1$.

A more general way of computing the tangent is the following. By differentiating $\Phi(\gamma(s)) = 0$ to s we find

$$[\Phi_x \ \ \Phi_\lambda]\dot\gamma(s) = \begin{pmatrix} \frac{\partial \Phi_1}{\partial x_1} & \cdots & \frac{\partial \Phi_1}{\partial x_N} & \frac{\partial \Phi_1}{\partial \lambda} \\ \frac{\partial \Phi_1}{\partial x_N} & \cdots & \frac{\partial \Phi_N}{\partial x_N} & \frac{\partial \Phi_N}{\partial \lambda} \end{pmatrix} \dot\gamma(s) = 0 \tag{4.8}$$

If $(\mathbf{x}_0, \lambda_0)$ is not a bifurcation point, then $\dim(\ker([\Phi_x \ \ \Phi_\lambda])) = 1$ and therefore $[\Phi_x \ \Phi_\lambda]$ has rank N. Hence, we can determine $\dot\gamma(s_0)$ as the null space of the $N(N+1)$ matrix $[\Phi_x \ \Phi_\lambda]$.

First, the matrix $[\Phi_x \ \Phi_\lambda]$ is triangulated into the form

$$\begin{pmatrix} * & * & * & * \\ 0 & * & * & * \\ 0 & 0 & * & * \end{pmatrix} \tag{4.9}$$

where this matrix (an $*$ indicates a possible nonzero element) is shown for $N = 3$. The last row cannot be entirely zero, and therefore the (permuted) tangent vector $\mathbf{v} = (\dot{\mathbf{x}}_0, \dot\lambda_0)$ can be computed by solving

$$\begin{pmatrix} * & * & * & * \\ 0 & * & * & * \\ 0 & 0 & * & * \\ 0 & 0 & 0 & 1 \end{pmatrix} \mathbf{v} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \tag{4.10}$$

and its length is normalized as in (4.7).

Once $\mathbf{x}_0, \lambda_0$, $\dot{\mathbf{x}}_0$ and $\dot\lambda_0$ are determined. a further point on the same branch can be calculated by taking

$$\Sigma(\mathbf{x}, \lambda, s) = \dot{\mathbf{x}}_0^T(\mathbf{x} - \mathbf{x}_0) + \dot\lambda_0(\lambda - \lambda_0) - (s - s_0) \tag{4.11}$$

and solve the total system of equations (4.5) and (4.11) given a prescribed steplength $\Delta s = s - s_0$. In this form, the continuation method is called a pseudo-arclength method [13,]. The name derives from the fact that (4.11) is an approximation to (4.7). The advantage of this method is that the Jacobian of the extended system (4.5)-(4.11) is non-singular at saddle node bifurcations, whereas the Jacobian Φ_x is. Hence, one can easily follow a branch around a saddle node bifurcation [13,].

4.1.1 The Euler-Newton method

To solve the equations (4.5) and (4.11), an Euler predictor/Newton corrector algorithm is applied. Let the steady state which is already known be indicated by $\mathbf{x}^0$, then a good guess for the next steady state is the Euler predictor given by

$$\mathbf{x}^1 = \mathbf{x}_0 + \Delta s \ \ \dot{\mathbf{x}}_0 \tag{4.12}$$

$$\lambda^1 = \lambda_0 + \Delta s \ \ \dot{\lambda}_0 \tag{4.13}$$

where again the dot indicates differentiation to s. Now in the Newton-Raphson method, for $k = 1, 2, ...$ $\mathbf{x}^k$ and λ^k are updated by

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \Delta\mathbf{x}^{k+1} \tag{4.14}$$

$$\lambda^{k+1} = \lambda^k + \Delta\lambda^{k+1} \tag{4.15}$$

where $(\Delta\mathbf{x}^{k+1}, \Delta\lambda^{k+1})$ are solved from the correction equation

$$\begin{pmatrix} \Phi_x(\mathbf{x}^k, \lambda^k) & \Phi_\lambda(\mathbf{x}^k, \lambda^k) \\ \dot{\mathbf{x}}_0^T & \dot{\lambda}_0 \end{pmatrix} \begin{pmatrix} \Delta\mathbf{x}^{k+1} \\ \Delta\lambda^{k+1} \end{pmatrix} =$$

$$= \begin{pmatrix} -\Phi(\mathbf{x}^k, \lambda^k) \\ \Delta s - \dot{\mathbf{x}}_0^T(\mathbf{x}^k - \mathbf{x}_0) - \dot{\lambda}_0(\lambda^k - \lambda_0) \end{pmatrix} \tag{4.16}$$

Hence, within each iteration, a linear system of equations has to be solved. If the Newton-Raphson process has converged up to a desired accuracy, a new steady state has been found.

One can split the solution of (4.16) into two steps in which only linear systems with Φ_x are solved. Let $\mathbf{r} = -\Phi(\mathbf{x}^k, \lambda^k)$ and $r_{N+1} = \Delta s - \dot{\mathbf{x}}_0^T(\mathbf{x}^k - \mathbf{x}_0) - \dot{\lambda}_0(\lambda^k - \lambda_0)$, then if $\mathbf{z}_1$ and $\mathbf{z}_2$ are solved from

$$\Phi_x(\mathbf{x}^k, \lambda^k)\mathbf{z}_1 \quad = \quad \mathbf{r} \tag{4.17}$$

$$\Phi_x(\mathbf{x}^k, \lambda^k)\mathbf{z}_2 \quad = \quad \Phi_\lambda(\mathbf{x}^k, \lambda^k) \tag{4.18}$$

then the solution $(\Delta\mathbf{x}^{k+1}, \Delta\lambda^{k+1})$ is found from

$$\Delta\lambda^{k+1} \quad = \quad \frac{r_{N+1} - \dot{\mathbf{x}}_0^T\mathbf{z}_1}{\dot{\lambda}_0 - \dot{\mathbf{x}}_0^T\mathbf{z}_2} \tag{4.19}$$

$$\Delta\mathbf{x}^{k+1} \quad = \quad \mathbf{z}_1 - \Delta\lambda^{k+1}\mathbf{z}_2 \tag{4.20}$$

One of the problems involved is the determination of the Jacobian matrix Φ_x and the derivative vector Φ_λ. One can do this in, at least, four ways: (i) 'by hand', (ii) symbolically using Mathematica or Maple, (iii) use automatic differentiation software which provides the code for the Jacobian matrix Φ_x from that of the right hand side Φ — an example of such a program is ADIFOR (http://www-unix.mcs.anl.gov/autodiff/ADIFOR/) — or (iv) compute it numerically by finite differences through

$$\frac{\partial\Phi_k}{\partial x_l} \approx \frac{\Phi_k(x_l + \epsilon) - \Phi_k(x_l)}{\epsilon} \tag{4.21}$$

for $k = 1, .., N; l = 1, ..., N$ and small ϵ.

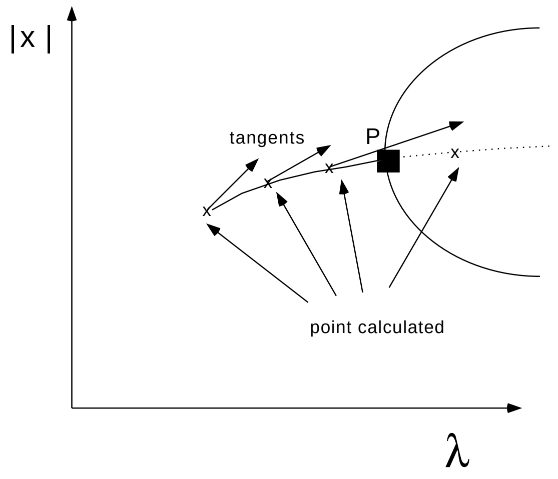


Figure 4.3: *Example of computations of steady states versus a parameter λ on a branch passing through a pitchfork bifurcation P.*

4.2 Detection and Switching

In the previous section, a method has been described to perform steady state continuation in a single parameter. Suppose, we have computed the points on a branch of steady solutions as indicated in Fig. 4.3 by varying a parameter λ. In this case, the method would just pass the pitchfork bifurcation point P (Fig. 4.3). How do we determine that this bifurcation has occurred?

One way to do this is to solve the eigenvalue problem associated with the stability of the steady state at each point. We know that for a pitchfork bifurcation, a single real eigenvalue must cross the imaginary axis. Hence, by monitoring these eigenvalues, the pitchfork bifurcation can be detected. In many applications, however, the solution of the eigenvalue problem is computationally expensive. Hence, simpler and cheaper indicator functions may be desired and some of these are described below.

4.2.1 Detection of bifurcations

To determine simple codimension-1 bifurcation points (transcritical, pitchfork and saddle node bifurcations), the determinant of the Jacobian matrix (det Φ_x) can be monitored. However, for many large dimensional problems this determinant is expensive to compute and other alternatives must be considered. In [25], a family of test functions τ_{pq} is obtained as follows: let Φ_{pq} be the Jacobian matrix Φ_x in which the p^{th} row is replaced by the q^{th} unit vector. If we solve the linear system

$$\Phi_{pq}\mathbf{v} = \mathbf{e}_p \tag{4.22}$$

for $\mathbf{v}$, where $\mathbf{e}_p$ is the p^{th} unit vector, then it can be shown [25,] that

$$\tau_{pq} = \mathbf{e}_p^T \Phi_x \mathbf{v} \tag{4.23}$$

changes sign when Φ_x is singular. In principle, the choices of q and p are arbitrary as long as Φ_{pq} is nonsingular. Of course, for any solution method, it is advantageous that Φ_{pq} and Φ_x have the same structure. However, in specific problems, not all values of q and p can be chosen and it is advisable to make a choice based on the knowledge of the (symmetry properties of the) solutions of the particular problem.
Saddle node bifurcations can be easily detected by following $\dot{\lambda}$ along a branch, where the dot indicates differentiation to the arclength parameter s. For Hopf bifurcation points, also more sophisticated methods exist [16,], but usually these points are determined by solving the linear stability problem which is discussed in the next section. In this case, a complex conjugate pair of eigenvalues $\sigma = \sigma_r + i\ \sigma_i$ crosses the imaginary axis and a zero of the function $\sigma_r(\lambda)$ has to be calculated to obtain the location of the Hopf bifurcation.
Once a change in sign is found in one of the scalar quantities, $\dot{\lambda}, \det\Phi_x$, τ_{pq} or $\sigma_r(\lambda)$, between two points along a branch, say s_a and s_b, a secant process can be used to locate the zero of each function exactly. In more detail, let either function be indicated by $f(s)$ then a zero of $f(s)$ is determined by

$$s_{l+1} = s_l - f(s_l)\frac{s_l - s_{l-1}}{f(s_l) - f(s_{l-1})} \tag{4.24}$$

$$s_0 = s_a \quad ; \quad s_1 = s_b \tag{4.25}$$

When $s_a \neq 0$, the stopping criterion on the iteration can be chosen as $\mid s_{l+1} - s_l \mid / s_a < \ \varepsilon$, where ε must be chosen to achieve the desired accuracy. In some cases, a larger ε must be taken because the matrix Φ_x may become nearly singular. It is recommended to check *a postiori* that the value of $f(s)$ is substantially smaller than the value of this function at both s_a and s_b.

4.2.2 Branch switching

If, for example, $\det(\Phi_x)$ changes sign but $\dot{\lambda}$ does not, a simple bifurcation point (transcritical or pitchfork) is detected. Subsequently, a branch switch process can be started to locate solutions on the nearby branch. In Fig. 4.4, this situation is sketched near a pitchfork bifurcation. Let $\hat{\Phi}_x$ be the Jacobian matrix at the bifurcation point $(\mathbf{x}_*, \lambda_*)$ just after the secant iteration (see the previous section) has converged. Furthermore, let the tangent along the already known branch in $s = s_a$ be indicated by $(\dot{\mathbf{x}}_0, \dot{\lambda}_0)$. First, the null vector ϕ of $\hat{\Phi}_x$ is calculated, for example by inverse iteration [1,]; the latter method is described in Section **??** Next, a vector $(\hat{\mathbf{x}}, \hat{\lambda})$ is constructed which is orthogonal to $(\dot{\mathbf{x}}_0, \dot{\lambda}_0)$ by solving

$$\begin{pmatrix} \hat{\Phi}_x & \hat{\Phi}_\lambda \\ \dot{\mathbf{x}}_0^T & \dot{\lambda}_0 \end{pmatrix} \begin{pmatrix} \hat{\mathbf{x}} \\ \hat{\lambda} \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ 0 \end{pmatrix} \tag{4.26}$$

The solution of this problem is easily determined to be

$$\hat{\lambda} = \frac{-\dot{\mathbf{x}}_0^T \phi}{\dot{\lambda}_0 - \dot{\mathbf{x}}_0^T \mathbf{z}} \ ; \ \hat{\mathbf{x}} = \phi - \hat{\lambda}\mathbf{z}$$

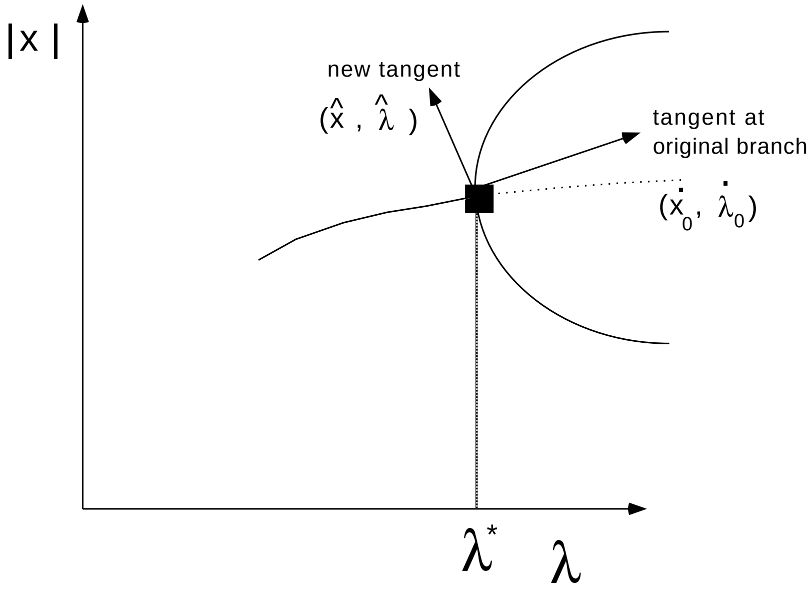


Figure 4.4: *Example of branch switching near a pitchfork bifurcation.*

where $\mathbf{z}$ is the solution of $\hat{\Phi}_x\mathbf{z} = \hat{\Phi}_\lambda$. To determine a point on the new branch (Fig. 4.4), the Newton process is started with Euler - predictor

$$\mathbf{x}^1 = \mathbf{x}_* \pm \Delta s\ \hat{\mathbf{x}}\ ;\ \lambda^1 = \lambda_* + \Delta s\ \hat{\lambda} \tag{4.27}$$

The ± indicates that points can be found on either side of the known branch. When a point on a new branch is found, the pseudo-arclength procedure is again used to compute additional points on this branch.

If one already anticipates a pitchfork bifurcation, one can also determine the other branch by a technique which makes use of the imperfections. Suppose, two points A and B on a branch are computed where the stability is different (Fig. 4.5a) or where some τ_{pq} from (4.23) changes sign. Now one knows that, associated with a pitchfork bifurcation, there is an internal symmetry of the system. By introducing an additional parameter p_s which breaks the symmetry (for example, introducing some asymmetric component in the forcing), the pitchfork no longer exists for small p_s. One continues a few steps into this parameter from point A up to $p_s = \varepsilon$. Then, a point C on the bifurcation diagram as in Fig. 4.5b is obtained. Next, the parameter λ is increased up to the value of λ at point B; in this way point D is reached (Fig. 4.5b). As a last step, p_s is continued back to zero and point E is obtained (Fig. 4.5c). By following the branch back in λ, the pitchfork is easily found as the point where $\dot{\lambda}$ changes sign.

4.2.3 Finding isolated branches

In many applications, there exist branches of steady state solution that are disconnected from the branch containing a trivial starting solution; these branches are the

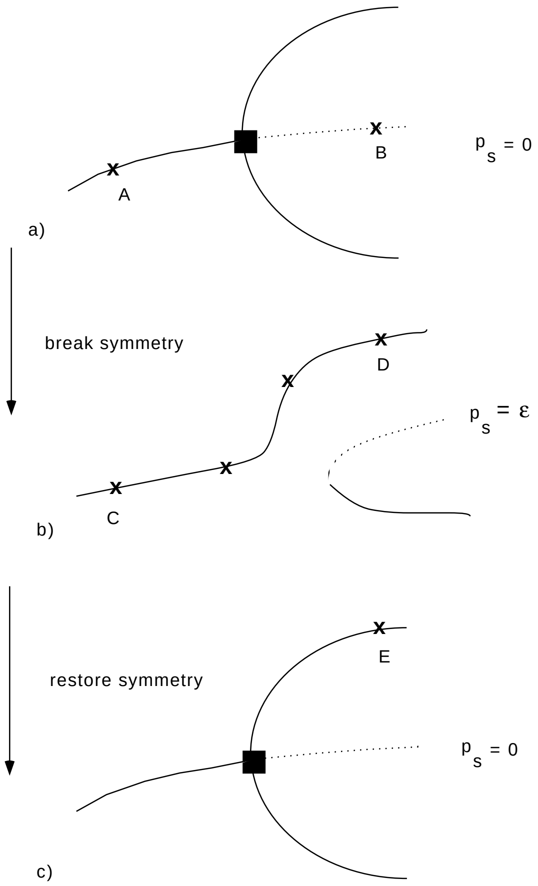


Figure 4.5: *Example of how knowledge of imperfections can be used to locate bifurcation points. The control parameter is λ. (a) Symmetric situation with computed points A and B, where a sign switch in one of the indicator functions has been detected. (b) The imperfect pitchfork bifurcation is created by adding artificial asymmetry into the set of equations using a parameter p_s. Point A is followed up to point C in p_s. As a next step, one continues from A to D for a value of λ approximately up to the value at B. (c) Finally, symmetry is restored, point D is followed up to E and the pitchfork can be found as the point where $\dot{\lambda}$ changes sign.*

so-called isolated branches. One can already anticipate that in a dynamical system in which there is no symmetry, it is likely that isolated branches are present.
There are at least four methods to compute these isolated branches but it is never guaranteed that one will find all branches with either of these methods. Two of those are more or less trial and error while in the latter two, a more systematic approach is followed.

(i) Transient integration.
In this approach, a set of initial conditions is chosen and a transient computation is started, for example by using an implicit method as in Section 4.4. If one is lucky, one of the initial conditions is in the attraction basin of a steady state on the isolated branch and once found (Fig. 4.6a), one can continue tracing this branch using the pseudo-arclength continuation method.

(ii) Isolated Newton-Raphson search.
One can also start a Newton-Raphson process uncoupled from the pseudo-arclength continuation from several chosen starting points. Since the convergence of the Newton-Raphson process is only good when one is near the steady state, this method may not work very well, but again, if one is very lucky an isolated branch might be found (Fig. 4.6b).

(iii) Two-parameter continuation.
In many cases, a second parameter can be varied such that the isolated branch targeted connects to an already known branch. An important example is where there are values of the second parameter for which the dynamical system has a particular symmetry and pitchfork bifurcations are present. Once the connection is present, the isolated branch can be computed by restoring the second parameter to its original value (Fig. 4.6c).

(iv) Residue continuation.
This method is a special case of a two-parameter continuation where one starts with a guess of the solution on the isolated branch, say indicated by $\mathbf{x}_G$, at some value of a parameter λ. Because this is no steady solution, it follows that

$$\mathbf{f}(\mathbf{x}_G, \lambda) = \mathbf{r}_G \neq 0$$

where $\mathbf{r}_G$ is the nonzero residue. One now defines a second (so-called 'homotopy') parameter α and considers the equations

$$\mathbf{f}(\mathbf{x}, \lambda) - (1 - \alpha)\mathbf{r}_G = 0$$

For $\alpha = 0$, the solution is given by $\mathbf{x}_G$ (by construction) and hence this is the starting point of the pseudo-arclength continuation. By tracing the steady solution branch from $\alpha = 0$ to $\alpha = 1$, we may eventually find an isolated branch (Fig. 4.6d).

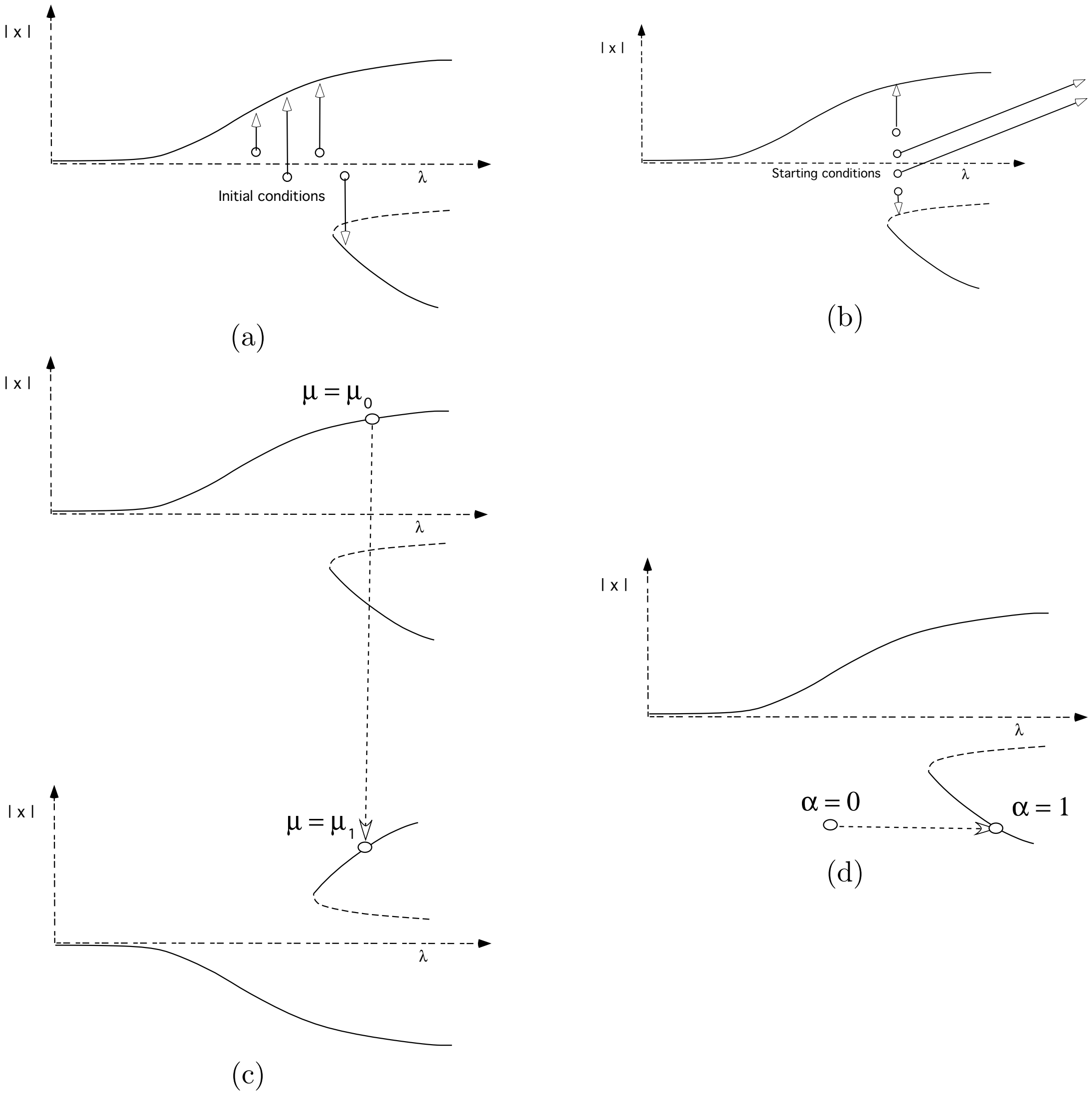


Figure 4.6: *Illustrations of the computation of isolated branches using four different methods. (a) Transient integration; the open circles indicate the initial conditions and the arrows the direction of the trajectories. Note that only stable steady states can be reached. (b) Isolated Newton-Raphson search; the open circles indicate the starting points. The two large arrows indicate a possible divergence of the Newton-Raphson process. (c) Two-parameter continuation; a pitchfork occurs for $\mu_0 < \mu < \mu_1$. (d) Residue continuation, where α is the 'homotopy' parameter.*

4.3 Linear Stability Problem

Suppose a stationary solution $\bar{\mathbf{x}}$ at a certain value of λ has been determined. Then its linear stability is investigated by considering perturbations $\mathbf{x} = \bar{\mathbf{x}} + \tilde{\mathbf{x}}$. Substituted into the general equations (4.70) and omitting quadratic terms in the perturbations quantities, one gets

$$\mathcal{M}\frac{\partial \tilde{\mathbf{x}}}{\partial t} + \mathcal{L}\tilde{\mathbf{x}} + \mathcal{N}_x(\bar{\mathbf{x}})\tilde{\mathbf{x}} = 0 \tag{4.28}$$

where the subscript N in (4.70) has been omitted for clarity and $\mathcal{N}_x$ is the Jacobian matrix. These equations admit solutions of the form $\tilde{\mathbf{x}} = \hat{\mathbf{x}}\ e^{\sigma t}$. The linear stability problem of a particular steady state leads, after discretization, to a generalized matrix eigenvalue problem of the form

$$A\mathbf{x} = \sigma B\mathbf{x} \tag{4.29}$$

with $A = \mathcal{L} + \mathcal{N}_x(\bar{\mathbf{x}})$ and $B = -\mathcal{M}$. The matrix B may be singular. For example in the incompressible Navier-Stokes equations, time derivatives are absent in the continuity equation and hence zeroes on the diagonal of B appear. The pair (A,B) is called a matrix pensil and some properties of the spectrum of matrix pensils are given in

4.4 Implicit Time Integration

In many models, there is an explicit time marching procedure, which can be represented by, using (4.70),

$$\mathcal{M}_N\mathbf{x}^{n+1} = \mathcal{M}_N\mathbf{x}^n + \Delta t\ \mathcal{G}(\mathbf{x}^n) \tag{4.30}$$

where $\mathcal{G}_N = \mathcal{F}_N - (\mathcal{L}_N + \mathcal{N}_N)$. Explicit schemes allow relatively easy implementation of all kinds of physical processes and details of boundary conditions, but suffer from a substantial drawback. The time step is limited because of numerical amplification of truncation errors (through well-known stability criteria) rather than by the changes in the actual solution [24,]. This limitation is even more restrictive as the spatial resolution increases. These properties are extremely undesirable for studies of PDEs where the spin-up takes almost all of the computing time. A long spin-up is for instance needed if there are parts of the solution with large time scales.

A nice spin-off of continuation methods is the immediate availability of implicit time integration schemes. Using a time step Δt, and a time index n, this scheme becomes for $\omega \in [0, 1]$,

$$\mathcal{M}_N\frac{\mathbf{x}^{n+1} - \mathbf{x}^n}{\Delta t} + (1-\omega)\mathcal{F}(\mathbf{x}^n) + \omega\mathcal{F}(\mathbf{x}^{n+1}) = 0 \tag{4.31}$$

For $\omega = 1/2$ and $\omega = 1$, these are the Crank-Nicholson method and backward Euler method, respectively [1,].

The equations for $\mathbf{x}^{n+1}$ are solved by the Newton-Raphson technique and lead to the same type of numerical problems as that for the steady state computation. It is well-known that the second-order Crank-Nicholson scheme is unconditionally stable for linear equations. This does not mean that one can take any time step, since this quantity is limited by two factors. One of those is accuracy: although the scheme is second-order accurate in time, large discretization errors occur when too large time steps are used. A second limitation on the time step is the convergence domain of the Newton-Raphson process, which does not necessarily converge for every chosen time step. For many applications, however, much larger time steps can be taken than in explicit models. For more details about time integration see Section 2.5.

4.5 A Prototype Problem

The problem below has been used as a testproblem during a workshop on "Application of Continuation Methods in Fluid Mechanics" in 1998 (see [12]). It is a relatively simple problem, and hence techniques can be easily illustrated. The physics of the problem is also very transparent, making it a nice prototype system to use here.

4.5.1 Introduction

The Rayleigh-Bénard problem is one of the 'classics' in fluid dynamics and one in the area of cellular convection. It is motivated by results from a (conceptually) simple experiment (Fig. 4.7). A rectangular container is filled with a viscous liquid such as silicone oil. Air is situated above the upper surface of the liquid and the temperature far from the air-liquid interface is nearly constant. When the initially motionless liquid is heated from below, the liquid remains motionless below a critical value of the vertical temperature gradient. In this case, the heat transfer through the layer is only by heat conduction. When the temperature gradient slightly exceeds the critical value, the liquid is set into motion and after a while the flow organizes itself into cellular patterns (Fig. 4.8).

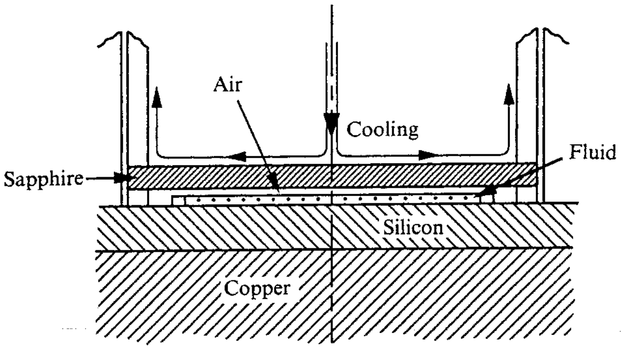


Figure 4.7: *Sketch of the experimental set-up; the liquid is situated on the (heated) silicon block and separated from the (cooled) sapphire block by a small air gap [15,].*

The motion of the liquid can also be detected by measuring the horizontally averaged vertical heat flux. A measure for the increase of heat transport due to convection is the Nusselt number Nu. This dimensionless scalar is the ratio of the heat transfer due to combined conduction and convection and the heat transfer due to conduction only; Nu = 1 in case of conduction only. In Fig. 4.9, Nu is plotted as a function of the

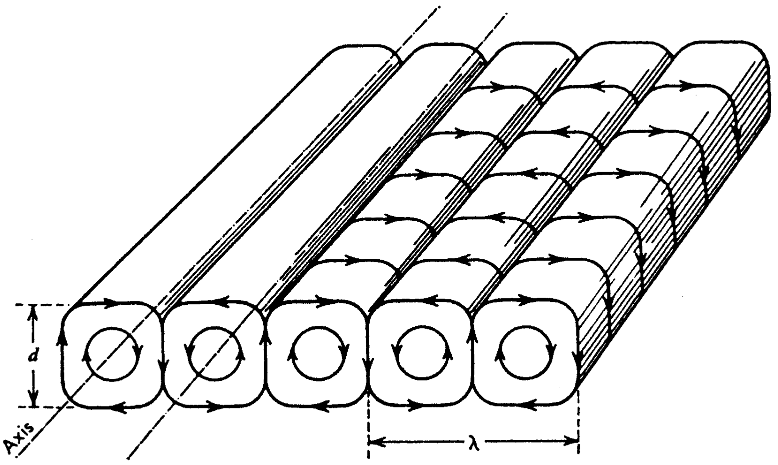


Figure 4.8: *Example of a flow pattern consisting of cellular rolls (also called roll cells) arising in a liquid heated from below. [14,].*

vertical temperature difference over the layer. The onset of convection in the liquid is shown by the increase of Nu above unity. From the experimental data, one can guess that some bifurcation is involved where the steady motionless state becomes unstable and new cellular type of solutions stabilize. From the symmetry properties of the flow — one can imagine to rotate the container over 180° and get the same experimental results — a pitchfork bifurcation is anticipated. One of the relevant problems with respect to the experiment is to determine the vertical temperature gradient associated with this bifurcation point.

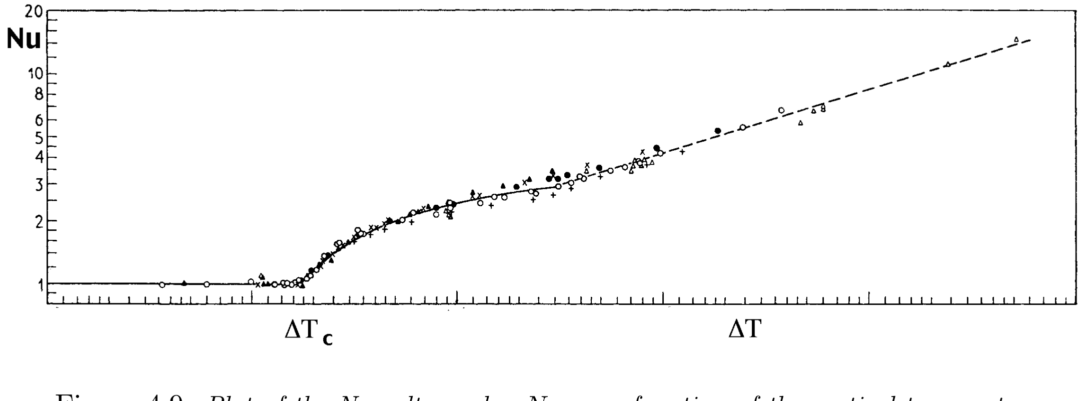


Figure 4.9: *Plot of the Nusselt number Nu as a function of the vertical temperature difference ΔT; $Nu = 1$ if the heat transport is by conduction only and when $Nu > 1$ there is convection in the liquid; ΔT_c is the critical temperature gradient [3,].*

4.5.2 Model

The equations governing the flow are

$$\rho_0 \left[\frac{\partial \mathbf{v}_*}{\partial t_*} + \mathbf{v}_*.\nabla \mathbf{v}_* \right] = -\nabla p_* + \mu \nabla^2 \mathbf{v}_* - \rho_* g \mathbf{e}_3 \tag{4.32}$$

$$\nabla \cdot \mathbf{v}_* = 0 \tag{4.33}$$

$$\rho_0 C_p \left[\frac{\partial T_*}{\partial t_*} + \mathbf{v}_*.\nabla T_* \right] = \lambda_T \nabla^2 T_* \tag{4.34}$$

In these equations, (x_*, y_*, z_*) are the Cartesian coordinates of a point in the liquid layer, t_* denotes time, $\mathbf{v}_* = (u_*, v_*, w_*)$ is the velocity vector, p_* denotes pressure, $\mathbf{e}_3$ the unit vector in z-direction and T_* is the temperature. Finally, ρ_0, g, C_p, μ and λ_T are the reference density, the acceleration due to gravity, the heat capacity, the dynamic viscosity and the thermal conductivity, respectively. The thermal diffusivity κ and kinematic viscosity ν are given by $\nu = \mu/\rho_0$, $\kappa = \lambda_T/(\rho_0 C_p)$ and all these quantities will be assumed constant. A linear equation of state

$$\rho_* = \rho_0(1 - \alpha_T(T_* - T_0)) \tag{4.35}$$

is assumed, where α_T is the thermal compressibility coefficient and T_0 a reference temperature. The lower boundary of the liquid is considered to be a very good conducting boundary on which the temperature is constant T_B, and no-slip conditions apply. On the lateral walls (at $x_* = 0, L_x$ and $y_* = 0, L_y$) no-flux and no-slip conditions are prescribed. Let the non-deforming gas-liquid interface be located at $z_* = d$, then the

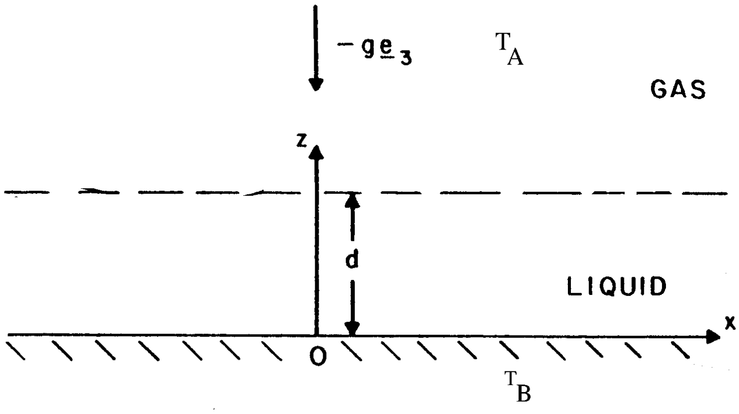


Figure 4.10: *Sketch of the model set-up and boundary conditions of the prototype problem.*

boundary conditions become (Fig. 4.10)

$$\begin{aligned} z_* = d &: \frac{\partial u_*}{\partial z_*} = \frac{\partial v_*}{\partial z_*} = w_* = 0 \; ; \; \lambda_T \frac{\partial T_*}{\partial z_*} = h(T_A - T_*) && (4.36) \\ z_* = 0 &: T_* = T_B \; ; \; u_* = v_* = w_* = 0 && (4.37) \\ x_* = 0, L_x &: u_* = v_* = w_* = \frac{\partial T_*}{\partial x_*} = 0 && (4.38) \\ y_* = 0, L_y &: u_* = v_* = w_* = \frac{\partial T_*}{\partial y_*} = 0 && (4.39) \end{aligned}$$

where h is an interfacial heat transfer coefficient and T_A is the temperature of the gas far from the interface.

4.5.3 Motionless solution

For $\bar{\mathbf{v}}_* = 0$, there is a steady state given by

$$\bar{T}_*(z_*) = T_B - \beta z_* \; ; \; \beta = \frac{h(T_B - T_A)}{\lambda_T + hd} \tag{4.40}$$

The quantity β is the vertical temperature gradient over the layer. The corresponding pressure distribution is readily determined from (4.32) and if one chooses $T_0 = T_A$, this gives

$$\bar{p}_*(z_*) = p_0 + \rho_0 g([\alpha_T(T_B - T_A) - 1]z_* - \frac{\alpha_T \beta}{2} z_*^2) \tag{4.41}$$

This motionless solution is characterized by only conductive heat transfer and is easily realized in laboratory experiments. Note that such a motionless solution exists for all values of the vertical temperature difference $\Delta T = \beta d$. Hence, according to theory presented in section 3.7, we would not expect saddle node bifurcations to occur on the branch of motionless solutions.

4.5.4 Dimensionless equations

The equations are non-dimensionalized using scales κ/d for velocity, d^2/κ for time and d for length. Moreover a dimensionless temperature T is introduced through $T_* = (T_B - T_A)T + T_A$ and a dimensionless pressure p through $p_* = p_0 + p(\mu\kappa/d^2)$. This leads to the non-dimensional problem

$$\begin{aligned} Pr^{-1}\left[\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v}.\nabla\mathbf{v}\right] &= -\nabla p + \nabla^2\mathbf{v} + Ra\ T\ \mathbf{e}_3 && (4.42) \\ \nabla \cdot \mathbf{v} &= 0 && (4.43) \\ \frac{\partial T}{\partial t} + \mathbf{v}.\nabla T &= \nabla^2 T && (4.44) \end{aligned}$$

with boundary conditions

$$z = 1 \quad : \quad \frac{\partial u}{\partial z} = \frac{\partial v}{\partial z} = w = 0; \frac{\partial T}{\partial z} = -Bi\, T \tag{4.45}$$

$$z = 0 \quad : \quad T = 1 \; ; \; u = v = w = 0 \tag{4.46}$$

$$x = 0, A_x \quad : \quad u = v = w = \frac{\partial T}{\partial x} = 0 \tag{4.47}$$

$$y = 0, A_y \quad : \quad u = v = w = \frac{\partial T}{\partial y} = 0 \tag{4.48}$$

In the equations (4.44)-(4.48), the dimensionless parameters Pr (Prandtl), Ra (Rayleigh), A_x, A_y (Aspect ratios) and Bi (Biot) appear which are defined as

$$\begin{aligned} Ra &= \frac{\alpha_T g(T_B - T_A)d^3}{\nu\kappa}; \; Pr = \frac{\nu}{\kappa}; \; Bi = \frac{hd}{\lambda_T} \\ A_x &= L_x/d; \; A_y = L_y/d \end{aligned} \tag{4.49}$$

and hence there are five parameters in this system of equations. This number reduces to four in the two-dimensional case since one of the aspect ratios is infinite.
The dimensionless motionless solution is given by

$$\bar{u} = \bar{v} = \bar{w} = 0 \quad ; \quad \bar{T}(z) = 1 - z\frac{Bi}{Bi+1} \tag{4.50}$$

$$\bar{p}(z) = Ra\left[z - \frac{Bi}{(1+Bi)}\frac{z^2}{2}\right] \tag{4.51}$$

and this is a solution for all values of Ra and Bi which makes it an ideal starting point for the computations below.

4.6 Computation of Steady Solutions

In this section, the present methods to determine steady state solutions in parameter space. To illustrate the discretization methods in section 4.6.1, the example problem from the previous section is used. In section 4.2.2, the pseudo-arclength continuation method is described.

4.6.1 Discretization

For the problem at hand, many type of discretization methods have been used, i.e. finite differences, finite elements and spectral methods. To illustrate the use of finite differences, consider the two-dimensional case in the prototype problem above, i.e. restricting to solutions $\mathbf{v} = (u, v, w)$ which are independent of y and with $v = 0$. A staggered grid is used with u, w at boundaries and p, T at center points of the grid cells (see Fig. 4.11a). The horizontal momentum equation is enforced at $u-$points, the vertical momentum equation at $w-$points and the continuity and temperature equation at center (p, T) points.

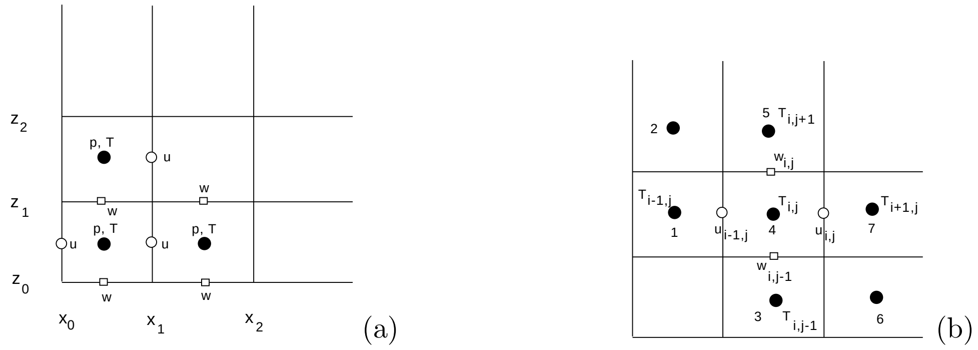


Figure 4.11: *(a) Sketch of the staggered grid, with points $i = 0, ..., I$ and $j = 0, ..., J$ in the x, z-direction, respectively. (b) Local stencil around the point $T_{i,j}$.*

For the discretization, it is efficient to define the discrete operators on a local stencil and subsequently assemble the operators over the whole domain. This is particularly useful when the nonlinear interactions in a model are at most quadratic, such as in the Navier-Stokes equations. In the latter case, the nonlinear operator $\mathcal{N}$ in (4.1) can be written as $\mathcal{N}(\mathbf{u})\mathbf{u}$. For each variable, a local stencil is defined such as in Fig. 4.11b for the temperature point $T_{i,j}$. As an example, consider the discretization of the horizontal diffusion operator, which is linear, using central differences. In this case, at point (i, j)

$$\frac{\partial^2 T}{\partial x^2} \approx \frac{T_{i+1,j} + T_{i-1,j} - 2T_{i,j}}{\Delta x^2} \tag{4.52}$$

According to the stensil (Fig. 4.11b), one now defines local operators $L^{TT}_{i,j}[1, \cdots, 7]$. The first superscript in L^{TT} indicates which equation is handled (in the case the temperature equation). The second superscript indicates by which unknown the coefficient has to be multiplied to get the right equations; in this case, temperature. The index $[*]$ refers to the stencil points and hence

$$L^{TT}_{i,j}[1] = \frac{1}{\Delta x^2} \tag{4.53}$$

$$L^{TT}_{i,j}[7] = \frac{1}{\Delta x^2} \tag{4.54}$$

$$L^{TT}_{i,j}[4] = -\frac{2}{\Delta x^2} \tag{4.55}$$

with all other $L^{TT}_{i,j}[*]$ being zero. The local operator is then built up as

$$\sum_{l=1}^{7} L^{TT}_{i,j}[l]T[l] \tag{4.56}$$

where $T[l]$ refers to the stensil around (i, j) with, for example, $T[1] = T_{i-1,j}$ and $T[5] = T_{i,j+1}$.

Next consider the nonlinear horizontal advection term for heat, which is discretized at $T-$ points as

$$\frac{\partial(uT)}{\partial x} = u_{i,j}\frac{T_{i+1,j} + T_{i,j}}{2\Delta x} - u_{i-1,j}\frac{T_{i,j} + T_{i-1,j}}{2\Delta x} \tag{4.57}$$

This term is a part of the nonlinear operator in the $T-$ equation associated with the operator $\mathcal{N}$ in (4.1). One defines a local nonlinear operator $N_{i,j}^{TT}[1,\cdots,7]$ as

$$N_{i,j}^{TT}[1] = -\frac{u_{i-1,j}}{2\Delta x} \tag{4.58}$$

$$N_{i,j}^{TT}[7] = \frac{u_{i,j}}{2\Delta x} \tag{4.59}$$

$$N_{i,j}^{TT}[4] = \frac{u_{i,j} - u_{i-1,j}}{2\Delta x} \tag{4.60}$$

with other stensil coefficients zero. The discretized equations of the local nonlinear operator are built up according to

$$\sum_{l=1}^{7} N_{i,j}^{TT}[l]T[l] \tag{4.61}$$

In this way, it is relatively easy to include boundary conditions. For example, imagine the implementation of a no-flux condition ($\partial T/\partial x = 0$) for the temperature at $x = 0$. Using central differences, this becomes

$$s\frac{\partial T}{\partial x} = 0 \Rightarrow T_{0,j} = T_{1,j} \tag{4.62}$$

s If the total stensil coefficient is indicated by $A^{TT} = L^{TT} + N^{TT}$, then the boundary condition can be accounted for by correcting the stensil coefficient $A_{1,j}^{TT}[4]$ as

$$s\tilde{A}_{1,j}^{TT}[4] = A_{1,j}^{TT}[4] + A_{1,j}^{TT}[1] \tag{4.63}$$

s and thereafter setting $A_{1,j}^{TT}[1] = 0$.
The boundary condition for temperature at $z = 0$ is discretized as

$$sT = 1 \Rightarrow \frac{1}{2}(T_{i,0} + T_{i,1}) = 1 \Rightarrow T_{i,0} = 2 - T_{i,1} \tag{4.64}$$

s This can be accounted for by correcting the stensil coefficient $A_{i,1}^{TT}[4]$ as

$$s\tilde{A}_{i,1}^{TT}[4] = A_{i,1}^{TT}[4] - A_{i,1}^{TT}[3] \tag{4.65}$$

s including a forcing term $F_{i,1}^{T} = 2\ A_{i,1}^{TT}[3]$ and setting $A_{i,1}^{TT}[3] = 0$ thereafter. Assembly of the total operators can be accomplished by one big loop over the grid points and the stencil points.
¿From (4.4), it can be seen that not only the discretized operator $\mathcal{N}$ is needed, but also its derivative $\mathcal{N}_u$ around a certain solution $(\bar{u}, \bar{w}, \bar{p}, \bar{T})$. For the horizontal advection operator in (4.57), this derivative becomes

$$\frac{\partial(\bar{u}T)}{\partial x} + \frac{\partial(u\bar{T})}{\partial x} \tag{4.66}$$

When discretized with central differences, the coefficients for the first term are similar to those in the operator $N_{i,j}^{TT}$ in (4.60) but with u substituted by $\bar{u}$. For the second term, an additional operator $N_{i,j}^{TU}$ is needed, which is defined by

$$N_{i,j}^{TU}[1] = -\frac{\bar{T}_{i-1,j} + \bar{T}_{i,j}}{2\Delta x} \tag{4.67}$$

$$N_{i,j}^{TU}[4] = \frac{\bar{T}_{i+1,j} + \bar{T}_{i,j}}{2\Delta x} \tag{4.68}$$

such that the term from this operator in the Jacobian matrix is built up as

$$\sum_{l=1}^{7} \left[N_{i,j}^{TT}[l]T[l] + N_{i,j}^{TU}[l]u[l] \right] \tag{4.69}$$

where again $T[l]$ and $u[l]$ refer to stensil point values, i.e. $u[4] = u_{i,j}$. Corrections due to boundary conditions and assembly of the matrices can be accomplished in the same way for the other operators in (4.70).

In this way, the discretized equations can be written as a nonlinear system of ordinary differential equations with algebraic constraints which has the form

$$\mathcal{M}_N \frac{d\mathbf{x}}{dt} + \mathcal{L}_N \mathbf{x} + \mathcal{N}_N(\mathbf{x}) = \mathcal{F}_N \tag{4.70}$$

where $\mathbf{x}$ indicates the total N-dimensional vector of unknowns, and where the operators depend on parameters and their subscript N indicates that they are discrete equivalents of the continuous operators. In the two-dimensional prototype problem, $\mathbf{x}$ is given by

$$\mathbf{x} = (u_{0,0}, w_{0,0}, p_{0,0}, T_{0,0}, u_{1,0}, ..., T_{I-1,J}, u_{I,J}, w_{I,J}, p_{I,J}, T_{I,J}) \tag{4.71}$$

and $N = 4\ (I+1)\ (J+1)$.

4.7 Application to the Prototype Problem

In this section, a typical application of the methods above is presented for the Rayleigh-Bénard problem as described in section 4.6. All results below were computed with a version of the code BOOM, which has been developed in my group over the years. The BOOM (Dutch for 'tree' and abbreviation for Bifurcation Analysis ('Onderzoek' in Dutch) of Ocean Models) code combines the continuation method with a choice of eigenvalue solvers and iterative linear systems solvers. The user has to supply the discretized operators $\mathcal{L}_N$, $\mathcal{N}_N$, $\mathcal{M}_N$, $\mathcal{F}_N$ and the Jacobian matrix.

A starting point $(\mathbf{x}_0, \lambda_0)$ has to be prescribed and the number of eigenvalues m_e to compute within the linear stability analysis has to be chosen. The sequence of computations is the following:

1. Compute the tangent vector $(\dot{\mathbf{x}}_0, \dot{\lambda}_0)$, if necessary (sometimes it can be analytically determined).

2. Compute the Euler guess with chosen steplength Δs

$$\mathbf{x} = \mathbf{x}_0 + \Delta s\ \dot{\mathbf{x}}_0$$
$$\lambda = \lambda_0 + \Delta s\ \dot{\lambda}_0$$

3. Solve the system of nonlinear algebraic equations for the steady equations using the Newton-Raphson method. Within each Newton iteration, one (or two) systems of linear equations have to be solved with a chosen method (direct, iterative).

4. When the previous step has converged, the generalized eigenvalue problem $A\mathbf{x} = \sigma B\mathbf{x}$ is solved for the first m_e eigenvalues closest to the imaginary axis using a chosen eigenvalue solver (SIT, JDQZ).

5. Compute a desired number of testfunctions to monitor properties of the flow and to monitor whether bifurcations have occurred (real part of eigenvalues, testfunctions τ_{pq} as in (4.23), determinant of Jacobian matrix). Take action, if a bifurcation point is detected, for example proceed with branch switching.

The two-dimensional case of the test problem (Fig. 4.12) is considered for a liquid with $Pr = 1$ which is heated from below in a container of aspect ratio $A = 10$. For water, with $\kappa = 10^{-7}\text{m}^2/\text{s}$ and $\nu = 10^{-6}\text{m}^2/\text{s}$, the Prandtl number is about 10. Results for this problem have been presented extensively in section 4 of [28] for the case $Bi = \infty$ and no-slip conditions at all walls, using the original primitive equation formulation with unknowns (u, w, p, T). Here, also results can be found on the performance of the iterative methods GMRES and BICGSTAB, either for the steady equations as well as within the JDQZ method. These results indicate that the methods used are indeed efficient for the prototype problem but since so many parameters are involved, they are not presented here; interested readers should consult [28].

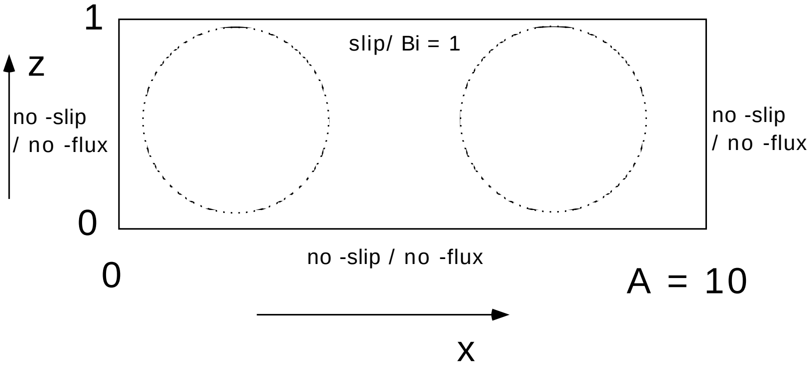


Figure 4.12: *Set-up of the two-dimensional configuration of the prototype problem.*

For the two-dimensional case, a more efficient formulation of the prototype problem was used in [5]. A streamfunction-vorticity formulation can be used, where the stream-

$I \times J$ (Grid)	aspect ratio	Bi	lateral walls	Ra_c
128×16	$A = 10$	$Bi = 1$	no-slip	1589.76
256×16	$A = 10$	$Bi = 1$	no-slip	1566.30
512×16	$A = 10$	$Bi = 1$	no-slip	1563.78
16×16	$A = \pi/a_c$	$Bi = 1$	slip	1555.58
32×32	$A = \pi/a_c$	$Bi = 1$	slip	1544.50
64×64	$A = \pi/a_c$	$Bi = 1$	slip	1541.98
$\infty \times \infty$	$A = \pi/a_c$	$Bi = 1$	slip	1541.18
[20,]	$A = \pi/a_c$	$Bi = 1$	slip	1541.14
256×16	$A = 10$	$Bi = 5$	no-slip	1620.10
256×16	$A = 10$	$Bi = 10$	no-slip	2019.02

Table 4.1: *Grid test of the value of the first bifurcation point. The first three rows show the convergence of the value of Ra for the case considered here. In the next five rows, a comparison with analytically determined values can be made for a special aspect ratio $a_c = \pi/\sqrt{2}$ and slip conditions at the lateral boundaries [20,]. The last two rows show the sensitivity of the location of the first bifurcation point with Bi.*

function ψ and the vertical component of the vorticity vector ζ are defined as

$$u = \frac{\partial \psi}{\partial z} \quad ; \quad w = -\frac{\partial \psi}{\partial x} \tag{4.72}$$

$$\zeta = \frac{\partial w}{\partial x} - \frac{\partial u}{\partial z} \tag{4.73}$$

This reduces the number of unknowns per point from 4 (u, w, p, T) to 3 (ψ, ζ, T). The equations in this formulation are easily derived by taking the rotation of the momentum equations (4.32) and become

$$Pr^{-1}\left[\frac{\partial \zeta}{\partial t} + \frac{\partial (u\zeta)}{\partial x} + \frac{\partial (w\zeta)}{\partial z}\right] = \nabla^2 \zeta + Ra \frac{\partial T}{\partial x} \tag{4.74}$$

$$\zeta = -\nabla^2 \psi \tag{4.75}$$

with boundary conditions

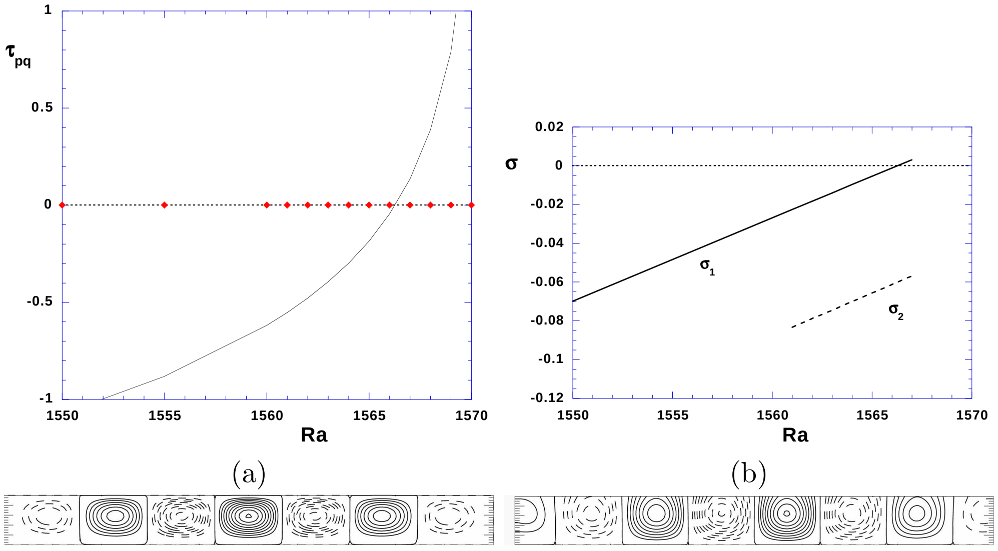
$$x = 0, A \quad : \quad \frac{\partial T}{\partial x} = \psi = \gamma \frac{\partial \psi}{\partial x} + (1-\gamma)\zeta = 0 \tag{4.76}$$

$$z = 0 \quad : \quad T - 1 = \psi = \frac{\partial \psi}{\partial z} = 0 \tag{4.77}$$

$$z = 1 \quad : \quad \frac{\partial T}{\partial z} + Bi\, T = \psi = \zeta = 0. \tag{4.78}$$

where $\gamma = 0$ and $\gamma = 1$ give slip and no-slip conditions, respectively. Details of the discretization, on a non-staggered grid, can be found in [4] and [5].

First aim of the computations is to find the critical temperature gradient (or critical value of Ra) for fixed Bi. Hence, we take $Bi = 1$, $\gamma = 1$, $\lambda = Ra$, start at the



(a)

(b)

(c)

(d)

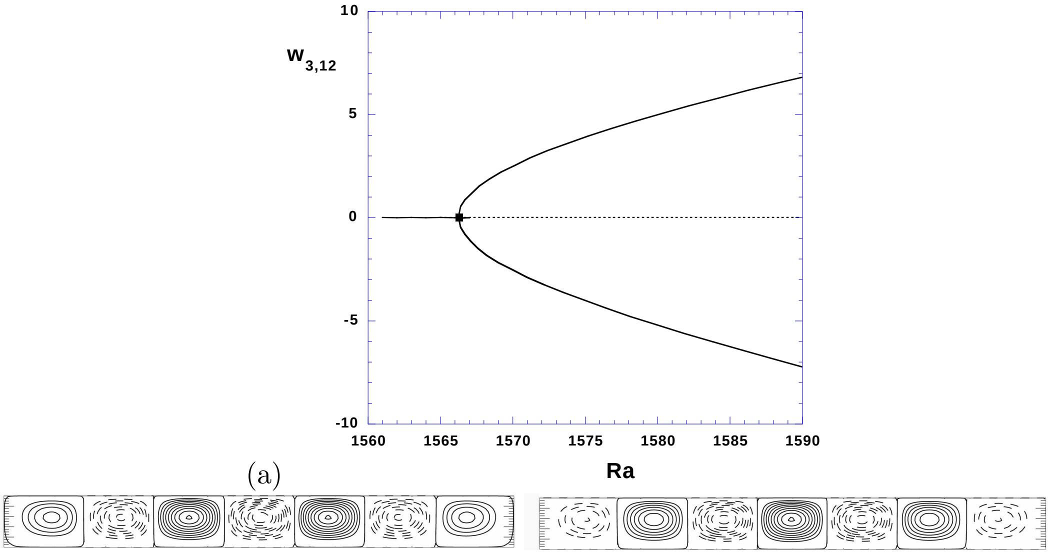
Figure 4.13: *(a) Computation of the testfunction τ_{pq} along the primary branch of motionless flow. A zero of this function may indicate a bifurcation point. The diamonds indicate the actual points computed along the motionless state. (b) First two eigenvalues as a function of Ra along the same branch. Indeed, a real eigenvalue passes the imaginary axis at the same location where the zero of τ_{pq} appears. (c) Pattern of the streamfunction of the eigenvector corresponding to σ_1, just at the point where the imaginary axis is crossed. (d) Pattern of the temperature of the same eigenvector.*

motionless solution (4.40) and prescribe the initial tangent as $(\dot{\mathbf{x}}_0, \dot{\lambda}_0) = (\mathbf{0}, 1)$. The latter can be used because the motionless solution is a solution for all values of Ra. The version of the code applied has an iterative solver (BICGSTAB combined with ILU-preconditioning) and the SIT eigenvalue solver [5,]. In the latter paper, also the performance of the preconditioner (see Technical Box 4.4) can be found.

In Fig. 4.13, the computation along the primary branch (i.e. the motionless solution) is displayed using a 256×16 grid. Note that the dimension of the dynamical system is $3 \times 257 \times 17 = 13,107$. A particular testfunction τ_{pq} (4.23) for $p = q = 256 \times 16 + 1 = 4,097$, is shown in Fig. 4.13a and goes through zero near $Ra = 1565$. In this figure, the points actually computed are indicated by the diamonds. The first two eigenvalues, which are both real, are shown along this motionless solution in Fig. 4.13b, indicating that the motionless solution becomes unstable near $Ra = 1565$, since one eigenvalue crosses the imaginary axis. Patterns of the streamfunction and temperature perturbation which destabilize the motionless state (the eigenvector associated with σ_1) are plotted in Fig. 4.13c and Fig. 4.13d, respectively. The pattern consists of seven cells and the solution for the streamfunction is symmetric with respect to the mid-axis of the container. Note that the pattern with counter-rotating cells is also an eigenvector associated with σ_1.

For each application, it is recommended to check whether the chosen resolution is sufficient to get accurately enough results. If the discretization is consistent then, for an infinitely fine grid, the results of the continuous problem are approached. To check the convergence of the numerical discretization procedure and to be able to extrapolate to the continuous problem, the value of Ra at the first bifurcation is determined for a number of grid sizes; the result is shown in Table 4.1. One can see that there is convergence and that a 256×16 is a reasonable grid to perform the computations. In this case, a comparison with analytical solutions is also possible for a particular aspect ratio and value of Bi if the boundary conditions on the sidewalls (4.76) are taken to be slip conditions ($\gamma = 0$). The sensitivity of the bifurcation point with Bi is illustrated in the last two rows of Table 4.1. Note that the value of Ra at these bifurcation points does not depend on Pr since the eigenvalues σ are all real.

Because of the reflection symmetry through the mid-axis ($x = A/2$), a pitchfork bifurcation is expected to occur at $Ra = 1566$. The bifurcation structure for $Pr = 1.0$ is plotted in the weakly nonlinear regime in Fig. 4.14a. On the vertical axis, the vertical velocity at the gridpoint $(3, 12)$ - near the upper left corner - is plotted. The slightly supercritical patterns near the primary bifurcation point are shown in the Figs. 4.14b-c for streamfunction and temperature, respectively. At the first primary bifurcation point ($Ra = 1566$) the motionless solution becomes unstable to the 7-cell pattern (Fig. 4.14b) which stabilizes in a supercritical pitchfork bifurcation. Also its symmetry related pattern stabilizes (Fig. 4.14c) and both patterns are stable up to end of the computational domain. For the three-dimensional case, similar results can be calculated and an overview of the complete solution of this problem is presented in [9].



(a)

(b)

(c)

Figure 4.14: *(a) Bifurcation diagram and (b-c) cellular solutions for the streamfunction arising at the first pitchfork bifurcation.*

Bibliography

[1] K. E. Atkinson. *An Introduction to Numerical Analysis.* John Wiley and Sons, 1976.

[2] R.L. Burden and J.D. Faires. *Numerical Analysis.* Brooks/Cole, 2001.

[3] S. Chandrasekhar. *Hydrodynamic and Hydromagnetic Stability.* Clarendon Press, Oxford, U.K., 1961.

[4] H. A. Dijkstra. On the structure of cellular solutions in rayleigh-bénard-marangoni flows in small-aspect-ratio containers. *J. Fluid Mech.*, 243:73–102, 1992.

[5] H. A. Dijkstra, M. J. Molemaker, A van der Ploeg, and E. F. F. Botta. An efficient code to compute nonparallel flows and their linear stability. *Comp. Fluids*, 24:415–434, 1995.

[6] E. J. Doedel. AUTO: A program for the automatic bifurcation analysis of autonomous systems. In *Proc. 10th Manitoba Conf. on Numerical Math. and Comp.*, volume 30, pages 265–274, 1980.

[7] E. J. Doedel and L. S. Tuckermann. *Numerical Methods for Bifurcation Problems and Large-Scale Dynamical Systems.* Springer-Verlag, New York, 2000.

[8] I.S. Duff, A.M. Erisman, and J.K. Reid. *Direct methods for sparse matrices.* Monographs on numerical analysis. Oxford science publications, 1986.

[9] A. Yu. Gelfgat. Different modes of Rayleigh-Benard instability in two- and three dimensional rectangular enclosures. *J. Comp. Physics*, 156:300–324, 1999.

[10] A. George. Nested dissection of a regular finite-element mesh. *SIAM J. Numer. Anal.*, 10:345–363, 1973.

[11] J. Guckenheimer and S. Kim. Computational environments for exploring dynamical systems. *Bifurcation and Chaos*, 1:269–276, 1991.

[12] C. A. Katsman, M. J. Schmeits, and H. A. Dijkstra. Application of continuation methods in physical oceanography. In D. Henry and A. Bergeon, editors, *Continuation methods in Fluid Dynamics*, pages 155–166. Vieweg, 2000.

[13] H. B. Keller. Numerical solution of bifurcation and nonlinear eigenvalue problems. In P. H. Rabinowitz, editor, *Applications of Bifurcation Theory.* Academic Press, New York, U.S.A., 1977.

[14] E. L. Koschmieder. *Bénard Cells and Taylor Vortices.* Cambridge University Press, Cambridge, UK, 1993.

[15] E. L. Koschmieder and D. W. Switzer. The wavenumbers of supercritical surface-tension-driven Bénard convection. *J. Fluid Mech.*, 240:533–548, 1992.

[16] Y. A. Kuznetsov. *Elements of Applied Bifurcation Theory.* Springer Verlag, New York, U.S.A., 1995.

[17] K. Lust and D. Roose. Computation and bifurcation analysis of periodic solutions of large-scale systems. In E. Doedel and L. S. Tuckermann, editors, *Numerical Methods for Bifurcation Problems and Large-Scale Dynamical Systems*, pages 265–302. Springer, 2000.

[18] K. Lust, D. Roose, A. Spence, and A. R. Champneys. An adaptive Newton-Picard algorithm with subspace iteration for computing periodic solutions of large-scale dynamical systems. *SIAM J. Scientific Computing*, 19:1188–1209, 1998.

[19] A. H. Nayfeh and B. Balachandran. *Applied Nonlinear Dynamics.* John Wiley, New York, U.S.A., 1995.

[20] D. A. Nield. Surface tension and buoyancy effects in cellular convection. *J. Fluid Mech.*, 19:341–352, 1964.

[21] R. Peyret and T.D. Taylor. *Computational methods for fluid flow.* Springer-Verlag, 1983.

[22] A. Quarteroni, R. Sacco, and F. Saleri. *Numerical Mathematics.* Springer, 2007.

[23] R.D. Richtmyer and K.W. Morton. *Difference methods for initial value problems.* Interscience, 1967.

[24] P. Roache. *Computational Fluid Dynamics.* Hermosa Publishing, Albequerque, NM, U.S.A., 1976.

[25] R. Seydel. *Practical Bifurcation and Stability Analysis: From Equilibrium to Chaos.* Springer-Verlag, New York, U.S.A., 1994.

[26] W.F. Tinney and J.W. Walker. Direct solutions of sparse network equations by optimally ordered triangular factorization. In *Proc. IEEE 55*, pages 1801–1809, 1967. Proceedings, Reading.

[27] H.A. van der Vorst. *Computational Methods for large Eigenvalue Problems*, volume VIII of *Handbook of Numerical Analysis*, pages 3–179. North-Holland (Elsevier), 2002.

[28] J. J. Van Dorsselaer. Computing eigenvalues occurring in continuation methods with the Jacobi-Davidson QZ method. *J. Comp. Physics*, 138:714–733, 1997.