Konrad-Zuse-Zentrum
für Informationstechnik Berlin

Takustraße 7
D-14195 Berlin-Dahlem
Germany

P. DEUFLHARD

Differential Equations in Technology and Medicine. Computational Concepts, Adaptive Algorithms, and Virtual Labs.

Preprint SC 99–34 (September 1999)

Differential Equations in Technology and Medicine. Computational Concepts, Adaptive Algorithms, and Virtual Labs.

CIME Lectures

Peter Deuflhard

Abstract

This series of lectures has been given to a class of mathematics postdocs at a European summer school on *Computational Mathematics Driven by Industrial Applications* in Martina Franca, Italy (organized by CIME). It deals with a variety of challenging real life problems selected from clinical cancer therapy, communication technology, polymer production, and pharmaceutical drug design. All of these problems from rather diverse application areas share two common features: (a) they have been modelled by various *differential equations* – elliptic, parabolic, or Schrödinger–type partial differential equations, countable ordinary differential equations, or Hamiltonian systems, (b) their numerical solution has turned out to be a real challenge to computational mathematics.

Contents

Introduction

This series of lectures has been given to a class of mathematics postdocs at a European summer school in Martina Franca (organized by CIME). It deals with a variety of challenging real life problems selected from clinical cancer therapy, communication technology, polymer production, and pharmaceutical drug design. All of these problems from rather diverse application areas share two common features: (a) they have been modelled by various *differential equations* – elliptic, parabolic, or Schrödinger–type partial differential equations, countable ordinary differential equations, or Hamiltonian systems, (b) their numerical solution has turned out to be a real challenge to computational mathematics.

Therefore, *before* diving into actual computation, the *computational concepts* to be applied need to be carefully considered. To start with, any numerical analyst must be prepared to totally remodel problems coming from science or engineering – see e.g. Sections 3 and 4 below. The computational problems to be treated should be well–posed, important features of any underlying continuous model should be passed on, if at all possible, to the discrete model, and the computational resources employed (computing time, storage, graphics) should be adequate.

Speaking in mathematical terms, the solutions to be approximated live in appropriate infinite dimensional function spaces, e.g. in Sobolev spaces in Sections 1 and 2, in discrete weighted sequence spaces in Section 3, or in certain statistically weighted function spaces in Section 4. The *mathematical paradigm* advocated throughout this paper is that – already due to mathematical aesthetics – any *infinite* dimensional space should *not* be represented by just a *single finite* dimensional space (with possibly large dimension), but by a well–designed *sequence of finite dimensional spaces*, which successively exploit the asymptotic properties characterizing the original function space. The fascinating, but (for a mathematician) not really surprising experience is that *mathematical aesthetics go directly with computational efficiency.* In other words, a careful and sufficiently ingenious realization of the above paradigm will lead to efficient algorithms that actually work in hard real life problems. The reason for this coincidence of aesthetics and efficiency lies in the fact that function spaces describe some *data redundancy* that can be exploited in the numerical solution process. In order to do so, *adaptivity* of algorithms is one of the leading construction principles. Typically, wherever adaptivity has been successfully realized, algorithms with a computational complexity close to the (unavoidable) complexity of the problem emerge – a feature of extreme importance especially in challenging problems of science, technology, or medicine.

In traditional industrial environments, however, new efficient mathematical algorithms are not automatically accepted – even if they significantly supercede already existing older ones (if not old–fashioned ones) within long established commercial software systems. Exceptions do occur where simulation or optimization is the dominating competition factor. Due to this experience the author's group has put a lot of effort in the design of *virtual labs*. These specialized integrated software systems permit a fast and convenient switch between numerical code and interactive visualization tools (that we also develop, but do not touch here). Sometimes only such virtual labs open the door for new mathematical ideas in hospitals or industrial labs.

1 Partial Differential Equations in Cancer Therapy Planning

The present section deals with partial differential equation (PDE) models arising in medicine (example: cancer therapy hyperthermia) and high frequency electrical engineering (example: radio wave absorption). In this type of application the 3D geometry – say, of human patients – motivates the choice of *tetrahedral* finite element methods (FEM). The clinical setting requires the robust computational solution of problems to prescribed accuracy at highest possible speed on local workstations. Reliability plays the dominant role in medicine, which is a nice parallelism with the intentions of mathematics. Numerical speed is required to permit a fast simulation of different scenarios for different patients. In other words: the situation both requires and deserves the construction of highly efficient algorithms, numerical software, and visualization tools.

1.1 Multilevel Finite Element Methods Revisited

The presentation of this section focusses on elliptic or parabolic PDEs and Maxwell's equations. Mathematically speaking, the stationary solutions of these PDEs live in some Sobolev space like H^{α} or H_{curl} depending on the prescribed boundary conditions, whereas the time dependent solutions live in some scale of these spaces. In view of the above mentioned *paradigm* and the expected *computational complexity*, these spaces are approximated by a sequence of finite element spaces in the frame of multigrid (MG) or multilevel (ML) methods. Before going into the technical details of the real life problems to be presented below, a roadmap of several advanced computational concepts will be given first that have turned out to be important not only for the herein selected applications.

Optimal multigrid complexity. Classical MG methods have been first advocated for actual computation in the 70's by A. Brandt [26] and Hackbusch [52]. The latter author has paved the success path for MG methods by first proving an optimal computational complexity estimate $\mathcal{O}(N)$ for the so–called W–cycle, where N was understood to be the number of nodes in a *uniform* space grid. The same attractive feature was observed in suitable implementations of the simpler V–cycle and later proved by Braess and Hackbusch[23] under certain regularity assumptions and for uniform grids. The subsequent development was then characterized by a successive extension of MG methods from the originally only elliptic problems to larger and larger problem classes.

Adaptive multilevel methods. In quite a number of industrially relevant problems rather localized phenomena occur. In this case, uniform grids are by no means optimal, which, in turn, also means that the classical MG methods on uniform grids could not be regarded as optimal. For this reason, multigrid methods on *adaptive* grids have been developed quite early, probably first by R. Bank [8] in his code PLTMG in the context of problems arising from semiconductor device modelling where sharp local boundary layers arise naturally. Later adaptive MG implementations are the code family KASKADE [14] by the author's group and the code family UG [11] by Wittum, Bastian, and co-workers. UG in particular pays careful attention to parallelization issues [12]. The proof of an optimal computational complexity estimate $\mathcal{O}(N_{ad})$, where N_{ad} is now understood to be the often much smaller adaptive number of nodes, turned out to need more sophisticated proof techniques; for the well–known V–cycle, this challenging task has been performed by Bramble, Pasciak, Wang, and Xu [25] – see also Xu [90]. His rather elegant theoretical tools came from the interpretation of MG methods as abstract *multiplicative* Schwarz methods (equivalent to abstract Gauss–Seidel methods) based on an underlying multilevel splitting in function space.

Hierarchical bases finite element methods. Independent of the classical MG methods, a novel multilevel method based on conjugate gradient iteration with some *hierarchical basis* (HB) *preconditioning* had been suggested in the mid 80's for elliptic PDEs by Yserentant [91]. From the scratch, this new type of algorithm turned out to be competitive with classical MG in terms of computational speed. An *adaptive* 2D version of the new method had been designed and implemented in the late 80's by Deuflhard, Leinen, and Yserentant [39] in the code

KASKADE. On top of that first realization, a more mature version including also 3D has been worked out by Bornemann, Erdmann, and Kornhuber [16]. The present version of KASKADE [14] contains the original HB–preconditioner for 2D and the more recent BPX–preconditioner due to Xu [89, 24] for 3D. For an account of its performance see Section 1.2.

Additive versus multiplicative multigrid methods. After the theoretical milestone paper by Xu [90], the hierarchical basis type methods are now interpreted as abstract *additive* Schwarz methods (equivalent to abstract Jacobi methods) also based on a multilevel decomposition in function space. By construction additive Schwarz methods provide some preconditioning. In this interpretation, which the author prefers to adopt, the classical multigrid methods are then called *multiplicative MG methods*, whereas the HB– or BPX–preconditioned CG methods are called *additive MG methods*. In particular, the BPX–preconditioning and the V–cycle are just the additive and multiplicative counterparts. From theoretical analysis, multiplicative MG methods might require less iterations – which, however, need not imply less computing time (see e.g. the Maxwell MG solvers in Section 1.2 below, Table 1.1). Moreover, if an additive MG involves only one iteration on the finest grid, then multiplicative MG methods cannot gain too much. In the subsequently described elliptic problems, the bulk of computing time is anyway spent in the evaluation of the stiffness matrix elements and the right hand side elements. Summarizing, the question of whether additive or multiplicative MG methods should be preferred, appears to be less important than other conceptual issues – see below. For the orientation of the reader: UG is strictly multiplicative, PLTMG is dominantly multiplicative with some additive options, KASKADE is dominantly additive with some multiplicative code e.g. for eigenvalue problems and the harmonic Maxwell's equations. A common software platform of UG and KASKADE is in preparation.

Cascadic multigrid methods. These rather recent MG methods can be understood as some confluence of additive and multiplicative MG methods. From the additive point of view, cascadic multigrid (CMG) methods are characterized by the simplest possible preconditioner: either no or just a diagonal preconditioner is applied; as a distinguishing feature, coarser levels are visited more often than finer levels – to serve as preconditioning substitutes. From the multiplicative side, CMG methods may be understood as MG methods with an increased number of smoothing iterations on coarser levels, but without any coarse grid corrections. As a first algorithm of this type, a *cascadic conjugate gradient method* (CCG) had been proposed by the author in [30]. The general CMG class with arbitrary smoothers beyond CG has been presented by Bornemann and Deuflhard [20]. In their paper they analyzed CMG in terms of convergence and computational complexity in an adaptive setting – based on first much more restrictive convergence results due to Shaidurov [82]. These CMG methods exhibit good convergence properties only in H^1, but not in L^2 – unlike additive (with appropriate preconditioning) or multiplicative MG methods. Therefore, though being certainly easiest to implement among all MG methods, CMG methods – the youngest members of the MG family – are still in the process of maturing. Just to avoid mixing terms: CMG is different from the code KASKADE, which predominantly realizes additive MG methods.

Local error estimators. Any efficient implementation of *adaptive* MG methods (additive, multiplicative, cascadic) must be based on cheap *local error estimators* or, at least, *local error indicators*. In the best case, these are derived from theoretical *a–posteriori error estimates*. These estimates will be local only, if local (right hand side) perturbations in the given problem remain local – i.e. if the Greens' function of the PDE problem exhibits local behavior. As a consequence of this elementary insight, *adaptive* MG methods will be essentially applicable to linear or nonlinear elliptic or parabolic problems. As for a comparative assessment of the different available local error estimators, there is a beautiful paper by Bornemann, Erdmann, and Kornhuber [17] that gives a unified theoretical framework for most of the popular 2D and 3D error estimators. For orientation: PLTMG uses the triangle oriented estimator of Bank and Weiser [10], KASKADE the edge oriented estimator of Deuflhard, Leinen, Yserentant, and UG the estimator of Babuska and Miller [4].

Adaptive grid refinement. Within adaptive ML methods simplicial grids play a dominant role, since they behave nicely in local refinement processes. In connection with any selected error estimator, the local extrapolation method due to Babuska and Rheinboldt [5] can be applied to determine some threshold value, above which a geometrical element (tetrahedron, triangle, edge) is marked for local refinement. Once this marking has been done, well–designed strategies need to be applied to produce a complete FE grid on the next refinement level. The art of refinement is quite established in 2D (see the "red" and "green" refinements due to Bank et al. [9] or the "blue" refinement due to Kornhuber and Roitzsch [62]). In 3D there is still work left to be done, even though successful strategies due to Rivara[73], Ong[71], or Bey[15] have been around for quite a time.

Multilevel methods for nonlinear elliptic problems. For nonlinear elliptic problems there are two basic lines of MG methods: (I) the *nonlinear MG method*, sometimes also called *full approximation scheme* (FAS), wherein nonlinear residuals are evaluated within MG cycles, and (II) the *Newton MG method*, wherein *linear residuals* are evaluated within the MG method for the solution of the linear systems for the Newton corrections. In [42, 44] DEUFLHARD AND WEISER proposed an adaptive version for the second MG approach based on an *affine conjugate* characterization of nonlinearity via the special Lipschitz condition

$$\|F'(x)^{-1/2}\big(F'(y) - F'(x)\big)(y-x)\| \;\leq\; \omega\|F'(x)^{1/2}(y-x)\|. \tag{1.1}$$

This type of condition enters into certain affine invariant convergence results for both local and global *inexact* Newton methods in the function spaces $W^{p,q}$. The associated code Newton–KASKADE realizes a theoretically backed optimal balance between outer Newton iterations with possible adaptive damping, multilevel discretization, and inner preconditioned CG iterations; its performance is exemplified in Section 1.2 below.

Method of lines for parabolic PDEs. For time dependent PDEs, the most popular approach is still the so–called *method of lines* (MOL), which realizes a *first space / then time* discretization. After space discretization a typically large block–structured system of ordinary differential equations (ODEs) arises, which is then solved by any stiff ODE integrator: in the simplest (but often inefficient) case by an implicit or backward Euler with *constant* timestep, in advanced versions by some implicit multistep code (like DASSL), some implicit Runge–Kutta code (like RADAU 5), or some linearly implicit extrapolation code (like LIMEX) with *adaptive* control of timestep and possibly time discretization order. However, if one aims at dynamically adapted non–uniform space grids in 2D or 3D with MG methods to be applied, which is the typical case in *parabolic* PDEs, then the MOL approach will lead into some mass.

Adaptive Rothe method for linear parabolic PDEs. Starting 91, BORNEMANN [18, 19] suggested to abandon the MOL for parabolic PDEs and to use the so–called ROTHE method instead, which realizes a *first time / then space* discretization. His first papers dealt with initial boundary value problems for linear scalar parabolic PDEs such as

$$u_t = -\Delta u + f(x),\ u(x,0) = \varphi(x),\ u(x,t)\,|_{\partial\Omega} = \psi(t),\ t \geq 0,\ x \in \Omega \subset \mathrm{I\!R}^d\,. \tag{1.2}$$

Upon incorporating the boundary conditions into a linear elliptic operator $\mathcal{A}$ some function U is defined by virtue of the *abstract Cauchy problem*

$$U'(t) = \mathcal{A}U + F,\ U(0) = U_0\ . \tag{1.3}$$

Note that U represents a spatial function living in some *scale of Hilbert spaces* $H^\alpha(\Omega)$. The above abstract ordinary differential equation (ODE) may now be formally discretized for time step τ by some stiff integration scheme. For simplicity, we choose the implicit Euler method, which generates an equation of the type

$$(I - \tau\mathcal{A})\Delta U = \tau F\ . \tag{1.4}$$

This equation represents some (τ–dependent) elliptic boundary value problem, which can be solved by any adaptive multilevel method. Moreover, the available advanced ODE technology may also enter, but now in function space – which means that any error control devices known from finite dimensional ODEs are realized via spatial approximations using an adaptive MLFEM. Summarizing, a substantial advantage of this reversed order of discretization turns out to be that dynamic space grid adaptation and adaptive MG methods within each time layer are, in principle, easy to apply. Moreover, this approach nicely reflects the underlying theoretical structure.

Adaptive Rothe method for nonlinear parabolic PDEs. The above algorithmic approach can be extended to the nonlinear parabolic case. A rather direct extension is obtained on the basis of some abstract stiffness theory presented by the author in [31]. In this paper stiff time discretization of a nonlinear ODE initial value problem, say

$$U' = F(U),\quad U(0) = U_0 \tag{1.5}$$

has been interpreted as a *simplified Newton iteration* for the evolution problem *in function space*. This Newton iteration, in turn, may be formally understood as a Picard iteration for the slightly rewritten ODE

$$U' - \mathcal{A}U = F(U) - \mathcal{A}U,\ U(0) = U_0 \tag{1.6}$$

wherein $\mathcal{A} \approx F'(U_0)$ – i.e. in finite dimension $\mathcal{A}$ is just an approximate Jacobian (n,n)–matrix of the right hand side. From this theoretical insight *linearly implicit* stiff integration methods appear naturally – as opposed to nonlinear stiff discretization schemes like BDF or implicit RK methods. The concept directly carries over to infinite dimension when equations (1.5) and (1.6) are any abstract

Cauchy problem. Upon applying, for simplicity, the linearly implicit Euler discretization to (1.5), we arrive at some linear boundary value problem of the kind

$$(I - \tau\mathcal{A})\Delta U = \tau\big(F(U) - \mathcal{A}U\big) \ . \tag{1.7}$$

Following this line, Lang [63, 64] developed the adaptive multilevel code KARDOS that realizes a linearly implicit (embedded) Runge–Kutta method of low order on each discretization level. In its present form, this most recent code from the KASKADE family is applicable to 3D nonlinear systems of reaction–diffusion equations with mild convection. Generally speaking, since the adaptive Rothe method is fully adaptive in both time and space, it is able to resolve extreme *multiscales* in time and space that often arise in hard real life problems – like e.g. in chemical combustion. An early comparison of the new approach with the more traditional MOL approach (both adaptive 1D implementations) can be found in the survey paper [38]. The Rothe method will play a role in Section 1.2, Section 2, and Section 3.

1.2 Clinical Therapy Planning by Virtual Patients

The so–called *regional hyperthermia* is a rather recent promising cancer therapy based on the *local* heating of tumor tissue to above a threshold value of about 42 °C. At present this therapy is applied in combination with chemotherapy or radiotherapy. The idea is that heated tumor cells are more sensitive to extinction by either rays or drugs. For the medical treatment, the cancer patient is put into an applicator, which essentially consists of a set of 83 (old) or 24 (new) radiofrequency antennas and a water bolus to allow for a low reflection passage of the radio waves into the body – see Fig. 1.1.

The antennas emit radiation at a frequency of about 100 MHz corresponding to a wave length in water of about 30 cm, which – physically speaking – means that *wave optics* and interference phenomena rather than ray optics must be modelled. Heat within the body is produced by absorption of the radio waves and distributed by blood circulation in the tumor as well as in sane tissue. Mathematically speaking, the whole system (*patient, water bolus, applicator, surrounding air*) is modelled by the time harmonic Maxwell's equations in inhomogeneous media and a so–called bio–heat transfer (BHT) partial differential equation describing the heat distribution in the body. The task is to tune the set of radiofrequency antennas optimally such that the heat will concentrate within the tumor of a patient, but not at any hot spots elsewhere.

Figure 1.1: Real patient in hospital (Sigma–60 applicator).

In the project to be reported here, we have been collaborating with internationally renowned oncologists at one of the large Berlin hospitals, the Rudolf–Virchow–Klinikum at the Charité of the Humboldt University. Our task is to support the patient–specific planning of individual therapies. In order to make the method at all useful in a clinical environment, the computational results must be obtained within hours (at most) on a workstation in hospital *to medical reliability.* In addition, any numerical results are to be presented in visual form so that they can be directly interpreted and conveniently handled by medical staff. These requirements made the development of an integrated software package necessary that combines efficient 3D interaction tools with both numerical and computer graphical algorithms. As a prerequisite for the PDE solvers, a rather detailed *virtual patient* needs to be built up from medical imaging input (at present computed tomograms). The system as it stands now is already able to decide about the question whether a given patient can be expected to be successfully treated by hyperthermia using a given applicator. The presentation herein essentially follows the articles [41, 40]

Electric field simulation. We model the antennas by a fixed (angular) frequency ω and the human tissues so that Ohm's law holds. Let the electric field have a representation of the form $Re\,\mathbf{E}(\mathbf{x})e^{i\omega t}$ with a complex amplitude $\mathbf{E}(\mathbf{x})$ defined on a computational domain $\Omega \subset \mathbb{R}^3$. Then the time harmonic Maxwell's equations in terms of the electric field $\mathbf{E}$ and the magnetic field $\boldsymbol{H}$ read

$$\operatorname{curl} \boldsymbol{H} = \quad i\,\omega\epsilon\,\boldsymbol{E}\,, \quad \operatorname{curl} \boldsymbol{E} = -i\,\omega\mu\,\boldsymbol{H}\,, \tag{1.8}$$

where μ is the permeability and $\epsilon = \epsilon' - i\sigma/\omega$ is defined via the generic dielectric constant ϵ' and the conductivity σ. The two equations in (1.8) are combined with the well–known double–curl equation

$$\operatorname{curl}\,(\frac{1}{\mu}\operatorname{curl} \boldsymbol{E}) - \omega^2\epsilon\,\boldsymbol{E} \;=\; 0, \tag{1.9}$$

which will be the basis for the subsequent FE model. An appropriate function space for the differential operator in (1.9) and Dirichlet boundary conditions on the boundary Γ_D is

$$H_{curl} := \{\mathbf{w} \in (L^2(\Omega))^3\,;\;\; \mathbf{curl}\,\mathbf{w} \in (L^2(\Omega))^3,\; \mathbf{w}_t = \mathbf{E}_t^0 \text{ on } \Gamma_D\}.$$

The function space $H_{curl;0}$ is used for homogeneous boundary conditions $\mathbf{w}_t = 0$. We are now ready to give a variational formulation for the desired field $\mathbf{E}$ in the form: Determine $\mathbf{E} \in H_{curl}$ such that for all $\mathbf{w} \in H_{curl;0}$

$$\int_\Omega \{\frac{1}{\mu}\mathbf{curl}\,\mathbf{E}\,\mathbf{curl}\,\mathbf{w} \;-\; \omega^2\epsilon\mathbf{E}\,\mathbf{w}\}\;d\Omega \;-\; \int_{\Gamma_{ext}} \beta(\mathbf{n}\times\mathbf{E})\,(\mathbf{n}\times\mathbf{w})\;d\Gamma \;=\; 0. \tag{1.10}$$

Herein the second integral describes a contribution on the exterior boundary Γ_{ext}. The above bilinear form is coercive for non–vanishing σ, which implies that the problem has a unique solution. Note that the negative part of the integrand plays an important role especially for *high frequency* ω. In the positive semi–definite part, the ample nullspace of the **curl**–operator is rather undesirable, since all standard iterative methods (including MG) are known to preserve nullspace components. This causes a slowing down of convergence, once these nullspace components are present.

For the FE discretization of (1.10) we employ NÉDÉLEC's **curl**–conforming finite elements of lowest order [70] on a tetrahedral triangulation $\mathcal{T}_h$ of the domain – also called Whitney 1–forms or *edge elements*. These elements are easy to refine, which is a necessary prerequisite for any adaptive FEM. They are divergence–free and inherit continuity of the tangential electrical field components from the physical equations so that unwanted spurious discrete solutions [21] are suppressed. Moreover, as pointed out recently by HIPTMAIR [55], they permit a *discrete Helmholtz decomposition*, which turned out to be crucial for the construction of an adaptive multigrid method with so–called hybrid smoothing [13]. As exemplified in

Ref. Depth	Nodes	#Iter Std	M–Hyb	A–Hyb	CPU [min] Std	M–Hyb	A–Hyb
0	128 365	4250	354	413	150	24	20
1	373 084	4832	265	277	800	76	60
2	1 085 269	> 10000	186	194	> 2000	215	160

Table 1.1: Multilevel solvers: standard smoothing (Std), multiplicative hybrid smoothing (M-Hyb), and additive hybrid smoothing (A–Hyb).

Table 1.1 for Gauss–Seidel smoothing, both the multiplicative and the additive hybrid MG versions exhibit optimal multigrid complexity: the number of iterations does not increase with increasing refinement levels; this had not been the case for the former standard (Std) versions. Note that in this case the additive compared to the multiplicative version turns out to require slightly more iterations but slightly less computing time.

Heat Transfer Model. Our present model for the dissipation of heat in the human body (cf. PENNES [72]) assumes *potential flow for the blood* within the various tissues including the tumor. This leads to the so–called *bio–heat transfer equation* (BHT)

$$\rho_t c_t \frac{\partial T}{\partial t} = \operatorname{div}(k \operatorname{grad} T) - W \rho_b \, \rho_t \, c_b \, (T - T_a) + Q \qquad Q = \frac{\sigma}{2} \, |\boldsymbol{E}|^2 \, . \tag{1.11}$$

Herein ρ_t, ρ_b denotes the density of tissue and blood, c_t, c_b the specific heat of tissue and blood, T, T_a the temperature of tissue and arterial blood, k the thermal conductivity of tissue, W the blood perfusion, Q the power deposition within the tissue, and σ the electric conductivity. The thermal effects of *strong* blood vessels are excluded in this simplified model – but will be included in a future stage of the project. For $W = W(x)$, this parabolic PDE is *linear*. We solve it by the adaptive multilevel method KASTIO as implemented within the tool box KASKADE. A more realistic model takes into account that blood flow depends on tissue temperature: experiments in [85] have shown that the blood flow in normal tissues, e.g., skin and muscle, increases significantly when heated up to $41 - 43$°C, whereas in the tumor zone the blood flow decreases with temperature. On this experimental basis, we chose $W = W(T)$ monotonically increasing in muscle and fat tissue, but monotonically decreasing in tumor tissue. The arising *nonlinear* PDE has been solved by the code KARDOS [65] for the time dependent case and by the code Newton–KASKADE [43] in the stationary case – both within the package KASKADE.

Initial Grid Generation. The multilevel FEMs just described require an initial coarse grid, which captures the essential geometric features of the stated problem including a subdomain characterization for the different materials (bone, fat, muscle, ...). The total number of elements should be as large as necessary to state the problem correctly, but as small as possible in order to reduce computational costs. Starting point is a stack of plane CT images (about 60 per patient) containing only density information, which first need to be *segmented* according to physiological and oncological knowledge; this is done by the medical staff. The task then is to construct 3D grids from this type of input. It has turned out in the course of the project, that we had to develop our own fast and robust techniques for grid generation. These techniques include: (a) *extraction of compartment interfaces from segmentation results* by a proper generalization of the marching cubes algorithm [66] to non–binary classifications [54]: a significant speed–up is obtained via lookup–tables; (b) *coarsening of compartment surfaces* to allow for initial grids with as few elements as possible [83]; (c) *tetrahedral mesh generation*: each compartment is filled with tetrahedra starting from its surface by using a *3D–advancing front* method [84].

At present, the whole grid generation process can be performed automatically within about 15 minutes CPU time on a UNIX workstation. A typical coarse grid patient model consists of 40,000 – 60,000 tetrahedra and 8,000 – 10,000 vertices.

Optimization Algorithm. In therapy planning, the antenna parameters for each channel $j = 1, ..., k$ (equivalent to k pairs of coupled antennas) must be computed. We parametrize the complex amplitudes z_j by their real amplitudes a_j and their phases θ_j according to $z_j = a_j \exp(-i\theta_j)$. Then parameters $p = \{\Re z_j, \Im z_j\}$ must be determined such that the following therapeutic goals are achieved:

- within the tumor a therapeutic temperature level $T_t \approx 43°\text{C}$ is maintained,
- regions of healthy tissue are not heated above $T_h \approx 42°\text{C}$.

For most patients both requirements cannot be fulfilled simultaneously. In searching for a compromise we avoid destruction of healthy tissue by the additional constraint that temperature in healthy tissue must not exceed certain limits which depend on the tissue type: 42°C for more sensitive tissue compartments (like bladder or intestine) and 44°C otherwise.

From these goals we arrive at the following objective function

$$f(p) = \int\limits_{\substack{x \in V_{tumor} \\ T(x,p) < T_t}} (T_t - T(x,p))^2 \, dx \; + \int\limits_{\substack{x \notin V_{tumor} \\ T(x,p) > T_h}} (T(x,p) - T_h)^2 \, dx \tag{1.12}$$

to be minimized subject to the constraints

$$T(x,p) \leq T_{lim}(x)\,, \quad x \notin V_{tumor}.$$

In the *linear* heat transfer model, simple superposition of the electric field $\mathbf{E}$ into k modes can be employed, which in $Q \sim \mid \mathbf{E} \mid^2$ leads to k^2 basic modes to be computed in advance, plus one further mode for the basal temperature T_{bas}. For the *nonlinear* bioheat transfer model, we constructed some fixed point iteration [40] that converges at an average contraction rate of $\theta \approx 0.3$. This algorithm exploits the fact that the Maxwell solves are considerably more expensive than the BHT solves.

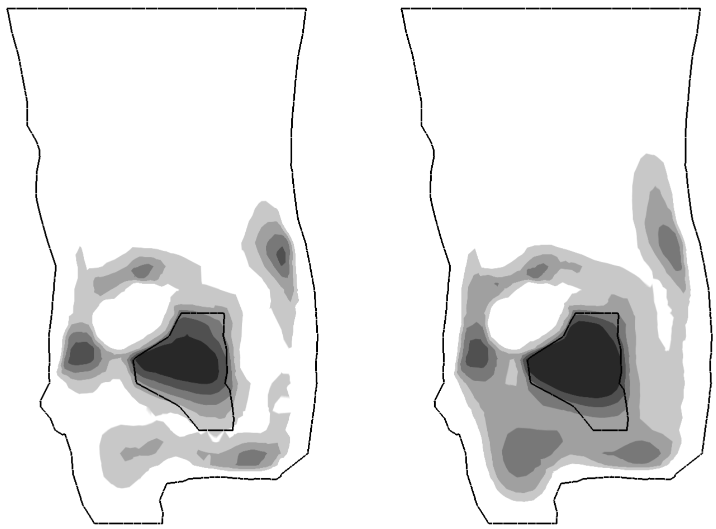

Figure 1.2: Optimized temperature distributions: linear (left) versus nonlinear model (right). Black lines: body outline and tumor contour. Light grey to dark grey: regions from 39°C to 43°C.

The total computational cost for the *nonlinear* case (with n iterations) can be counted to be

$$
\begin{aligned}
cost_{total} \;=\;& k * cost_{Maxwell} + \\
& n * (cost_{nlBHT} + (k^2+1) * cost_{lBHT} + cost_{Opt}) \qquad (1.13)
\end{aligned}
$$

where the notation is certainly self–explaining. We observed $n \approx 6$. The total cost for the *linear* case is obtained by inserting $cost_{nlBHT} = 0$ and $n = 1$ above.

Upon comparing linear versus nonlinear perfusion models, significant differences show up. As can be seen in Fig. 1.2, the nonlinear model predicts a tumor heating, which from the therapeutic point of view is slightly preferable. The nonlinear model also influences the choice of optimal parameters for the k channels.

Old versus new applicator. Our earlier computations have led to considerable improvements over the old applicator (Sigma–60, $k = 4$ channels, circular cross section with larger water bolus) in the form of some new applicator (Sigma–Eye, $k = 12$ channels, eye shaped cross section with smaller water bolus), see Fig. 1.3.

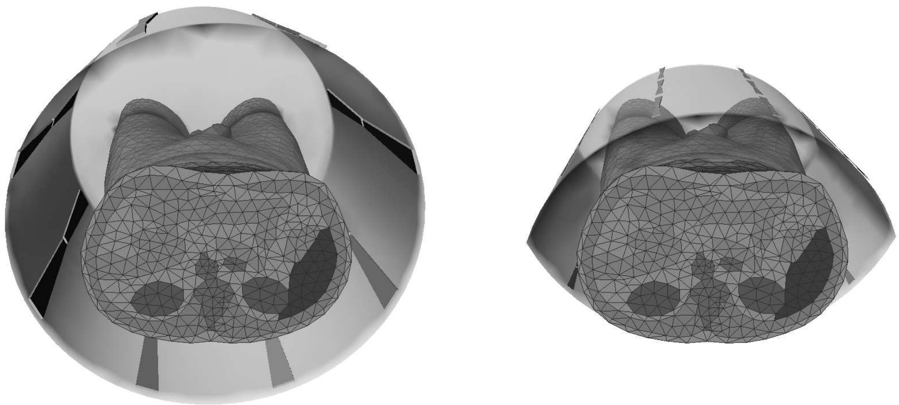

Figure 1.3: Virtual patient in Sigma–60 (left) and Sigma–Eye (right) applicator.

The therapeutic improvement can be seen in Table 1.2, which condenses the information obtained from simulation results for three virtual patients with different tumor locations. In order to illustrate the relative computational weights for the different algorithmic parts, we document some comparative results for both the old and the new applicator in Table 1.3 for the linear heat transfer model. The field computation times per channel of the old Sigma–60 appeared to be ~ 20 minutes as compared to ~ 10 minutes for the new Sigma–Eye, an effect due to the smaller bolus volume. As expected, the temperature computation times roughly scale with k^2.

Virtual patient	part of tumor volume heated to above 43°C	
	Sigma–60	Sigma–Eye
distal (supraanal) rectal carcinoma	17.5%	62.5
highly presacral rectal carcinoma	0.7%	18.4
cervical carcinoma at pelvic wall	24.8%	49.1

Table 1.2: Therapeutic improvement of new (Sigma–Eye) over old (Sigma–60) hyperthermia applicator.

Virtual Lab. The whole integrated software environment HyperPlan now consists of about 300.000 lines of code, wherein only about 120.000 lines are numerical code, the other parts are segmentation algorithms, grid generation methods, and visualization tools. This virtual lab has been recently sold to industry and will be worldwide distributed together with the applicator hardware – increasing the applicator's efficiency significantly.

	Sigma–60 $k = 4$	Sigma–Eye $k = 12$
Segmentation	2 – 4 hours*	
Grid Generation	15 min**	
Field Calculations	80 min**	120 min**
Temperature Calculations	2 min**	20 min**
Optimization	6 sec**	1 min**
	* interactive ** CPU time (SUN UltraSparc)	

Table 1.3: Computation times per patient.

2 Partial Differential Equations in Optical Chip Design

Every netsurfer now and then tends to have the impression that the abbreviation www means *world wide waiting* rather than *world wide web* – despite the tremendous information propagation speed of modern glass fibres. A drastically better performance rate – by many orders of magnitude! – can be expected by future so–called *optical networks.* In such networks all active components (like microlasers) or passive components (like couplers or tapers) are assembled on integrated optical chips. The technological aim is that signal *processing* on such a chip should reach a speed comparable to that of signal *propagation* along the fibre. In the project to be reported here the author's group at ZIB has been collaborating with the Heinrich–Hertz–Institute (HHI) in Berlin and with an industry research lab. As an example, Fig. 2.1 shows a patented optical chip that has been designed by HHI with parameters carefully specified on the basis of ZIB simulations. Its schematic representation is given in Fig. 2.2.

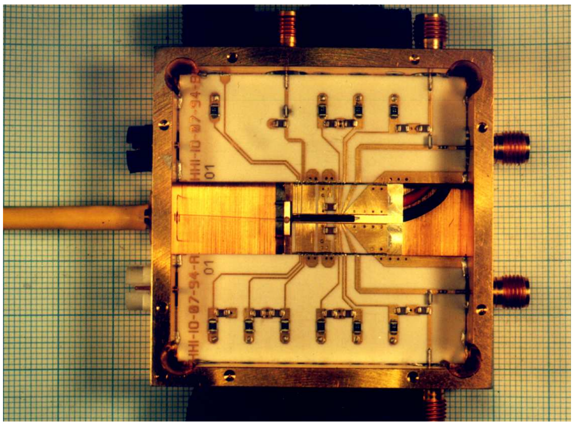

Figure 2.1: Integrated optical chip (central black stripe) mounted on ceramics substrate.

The design of integrated optical components is presently based on two different simplified mathematical models. Their efficient simulation requires the construction of two types of computational methods, the *beam propagation methods* (initial boundary value problems) and the *guided mode methods* (Helmholtz eigenproblems in selected cross section planes). For both of them we have made suggestions to be described now. Typical features coming from the technological problem are its *geometric complexity*, its *multiscale structure*, and the requirement of quite *stringent error tolerances* to control the behavior of the signals over long distances.

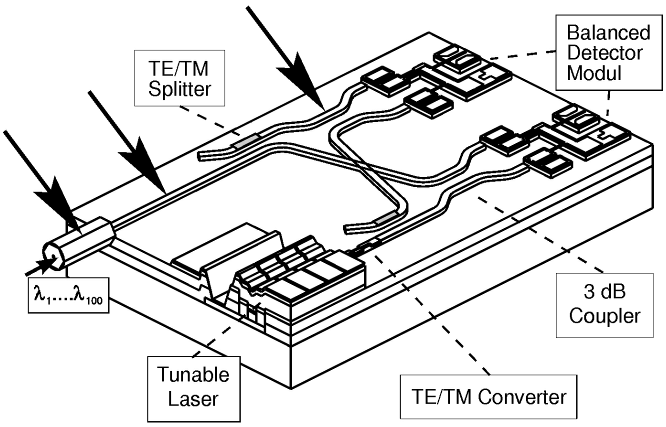

Figure 2.2: Schematic representation of the chip in Fig. 2.1.

2.1 Beam Propagation Analysis

When modelling the *signal propagation* along a glass fibre, the fibre axis naturally arises as a time–like coordinate z. In order to derive the so–called FRESNEL approximation, the electric field E is written in terms of a slowly varying amplitude u as

$$E = ue^{-in_0k_0z} \quad , \tag{2.1}$$

with k_0 the vacuum wave number and n_0 some *effective refraction index* to be specified below. Assume that we start from Maxwell's equations in the form (1.9). Let u depend on the propagation variable z and, for simplicity, only on one cross section variable x. For ease of writing we redefine $x := k_0x, z := k_0z, g := n^2 - n_0^2, c := 2in_0$. The specification of n_0 comes in by some projection argument to take energy conservation in the FRESNEL approximation at least to some extent into account. This leads to

$$n_0^2 = \frac{(n^2u, u) - (\nabla u, \nabla u)}{(u, u)} \quad . \tag{2.2}$$

We thus end up with the normalized *paraxial wave equation* for the transversal electrical mode (TE) in the form

$$\Delta u + g \cdot u = cu_z \quad , \tag{2.3}$$

which obviously is some complex SCHRÖDINGER–type equation. For pure beam propagation, we may even neglect the term u_{zz} so that Δu here means just u_{xx}.

This initial boundary value problem has been solved numerically in several technological projects by F. SCHMIDT [76] using an *adaptive Rothe method.* As already described in Section 1.1 above, this technique offers simultaneous adaptivity in both time and space together with multilevel speed. However, the desirable adaptivity cannot be fully exploited, unless suitable boundary conditions have been constructed, which are to be discussed next.

Discrete transparent boundary conditions. The idea behind the construction of transparent boundary conditions (TB) for wave type equations is to restrict the computations to some region of interest choosing boundary conditions such that waves touching the boundaries just pass these boundaries without any reflections. For some time the canonical approach has been to start from a set of TB derived from the *continuous* model, i.e. from the wave equation itself; these (non–local) boundary conditions were then discretized. However, proceeding like that will often induce discretized reflected waves and even instability [68]. For this reason, we derived a different approach in [78] that we called *discrete* transparent boundary conditions (DTB) – directly based on the Rothe method. Just like in the continuous case, these DTB are also of nonlocal Cauchy type. In addition each linear implicit discretization scheme induces its own DTB.

In order to exemplify the approach, we return to the above PDE (2.3). In the Rothe method the discretization for the time–like variable z, i.e. the direction of propagation, goes first. We deliberately apply the *implicit midpoint rule*, which has the selective feature that it conserves energy also in the discrete case. Consequently, any energy jumps observed in the course of the simulations must originate from the FRESNEL approximation – a convenient and cheap monitor for the validity of the employed model. After z–discretization of (2.3) neighboring time layers $(i, i+1)$ will be related according to (note: $j = \sqrt{-1}$)

$$\begin{aligned}
\frac{\partial^2 u_{i+1}}{\partial x^2} - \lambda_{i+1}^2 u_{i+1} &= -\frac{\partial^2 u_i}{\partial x^2} + \kappa_{i+1}^2 u_i \\
\lambda_{i+1}^2(x) &:= \frac{4jn_0(z_i + \frac{1}{2}\Delta z_{i+1})}{\Delta z_{i+1}} - g(x, z_i + \tfrac{1}{2}\Delta z_{i+1}) \\
\kappa_{i+1}^2(x) &:= -\frac{4jn_0(z_i + \frac{1}{2}\Delta z_{i+1})}{\Delta z_{i+1}} - g(x, z_i + \tfrac{1}{2}\Delta z_{i+1}) \\
\sigma_{i+1}^2(x) &:= 2g(x, z_i + \tfrac{1}{2}\Delta z_{i+1}).
\end{aligned} \tag{2.4}$$

This relation defines a nested sequence of $1D$ *boundary value problems* for successive solutions $u_i(x)$ within some finite region of interest; let $x = a$ denote one of the boundaries. Note that we need not restrict the time steps Δz_i to be constant. Following the lines of [78], the above non–local pattern can be taken into account in terms of certain LAPLACE transforms $U_i(p)$, which can be defined via the recurrence relations

$$U_{i+1}(p) = \frac{u_{i+1}(a) - u_i(a)}{p + \lambda_{i+1}} + U_i(p) - \sigma_{i+1}^2 \frac{U_i(p) - U_i(\lambda_{i+1})}{p^2 - \lambda_{i+1}^2} \,. \tag{2.5}$$

Once this can be solved, the boundary values of the solution u_{i+1} at $x = a$ are defined by

$$\left. \frac{\partial(u_{i+1} - u_i)}{\partial x} \right|_{x=a} + \lambda_{i+1}(u_{i+1}(a) - u_i(a)) = \sigma_{i+1}^2 U_i(\lambda_{i+1}). \tag{2.6}$$

Note that the new boundary conditions at time layer $i+1$ require the old boundary conditions from time layer i and the term $U_i(\lambda_{i+1})$ so that the recurrence (2.5) should be evaluated at $p = \lambda_{i+2}$. Hence, whenever $\lambda_{i+1} = \lambda_{i+2}$ – typically when locally constant stepsizes $\Delta z_{i+1} = \Delta z_{i+2}$ and homogeneous materials occur – then both the denominator and the numerator in the third right hand term of (2.3) vanish: so some limit needs to be taken. Numerical trouble will already arise for nearly zeroes. For this reason, the above recurrence relation turned out to be hard to stabilize numerically. Once this has been achieved (see [78]), the obtained algorithm was easy to realize. Summarizing, this type of DTB goes perfectly together with full adaptivity in space and time.

Remark. An even more elegant derivation of DTB by SCHMIDT AND YEVICK [79] applies some shift operator calculus. An inspired extension of DTB to the 2D Helmholtz equation can be found in the recent paper by SCHMIDT[77], who derives and exploits some type of discrete MIKUSIŃSKI operator calculus.

Numerical illustration. In order to illustrate both the adaptive Rothe method for the paraxial wave equation and the role of the discrete transparent boundary conditions, we give a numerical comparison from [78]. In Fig. 2.3 (left) a Gaussian peak with slight axial deviation is shown to hit homogeneous Dirichlet conditions (metallic boundary); the corresponding nodal flux obtained from the adaptive scheme is presented in Fig. 2.3 (right). Obviously, due to multiple reflections and the associated interference pattern the spatial grids fill up after few time steps making adaptivity dispensable. In contrast to this undesirable behavior, the numerical results obtained with our discrete transparent boundary conditions for the implicit midpoint discretization are pleasing: see Fig. 2.4 (left) and the corresponding nodal flux pattern in Fig. 2.4 (right). Recall that the computational amount is roughly proportional to the number of occurring nodes.

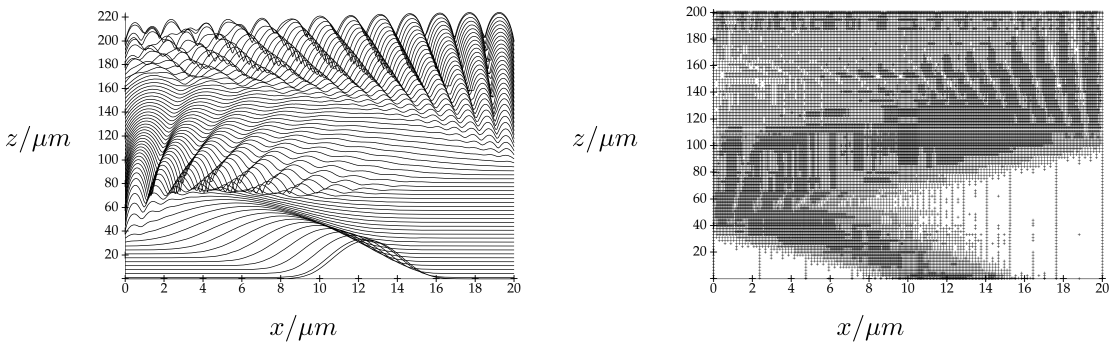

Figure 2.3: Gaussian peak propagation with metallic boundary conditions. Left: solution with interference pattern. Right: corresponding nodal flux.

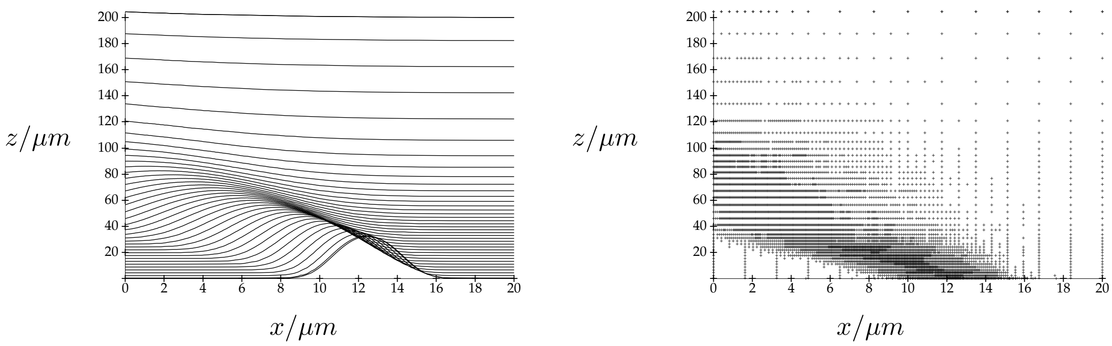

Figure 2.4: Gaussian peak propagation with discrete transparent boundary conditions for the implicit midpoint rule as z–discretization. Left: solution without any reflections. Right: corresponding nodal flux.

2.2 Guided Mode Analysis

As an alternative tool in chip design, engineers study so–called guided modes. For this purpose they select certain cross sections orthogonal to the propagation direction (see e.g. the arrows in Fig. 2.2). In the thus defined planes Maxwell's equations are simplified to (scalar or vectorial) 2D Helmholtz eigenvalue problems. Recall the already mentioned typical features coming from the technological problem like geometric complexity and multiscale structure. In addition, close *eigenvalue clusters* naturally occur that must be resolved to *high accuracy*. Therefore *subspace iteration methods* play a dominant role to assure a proper condition of the numerical problem (compare [50]) and, at the same time, reasonable convergence rates. In what follows we present adaptive FEMs for the complex Helmholtz eigenproblem, which have been recently developed by the author's group and successfully applied to the challenging problem class from integrated optics.

In order to derive the associated mathematical model, we look for solutions that are translation invariant along the propagation direction. They may be found through the ansatz

$$E(x, y, z) = u(x, y)e^{-i\mu z}. \tag{2.7}$$

Upon inserting this form into the paraxial wave equation (2.3) and introducing the notation $\lambda = -\mu^2$, we arrive at the *Helmholtz eigenvalue problem*

$$-\Delta u(x, y) - g(x, y) \cdot u(x, y) = \lambda\, u(x, y), \quad (x, y) \in \Omega \subset \mathsf{IR}^2 \ . \tag{2.8}$$

Eigenfunctions to $\mathrm{Re}(\lambda) \geq 0$ are called *evanescent modes*: they die out along the fibre due to exponential damping. Eigenfunctions to $\mathrm{Re}(\lambda) < 0$ are called *guided modes*: they live along the whole fibre as long as spatial dependencies are ignored. As already stated above, the latter are the objects of interest for the design of integrated optical chips. In weak formulation the above eigenvalue problem reads

$$a(u, v) = \lambda\,(u, v), \quad \forall v \in H, \tag{2.9}$$

with the sesquilinear form $a(u, v) = (\nabla u, \nabla v) - (gu, v)$ and some Sobolev space H chosen according to either Dirichlet or Neumann boundary conditions. For g real (no material losses assumed) the inner product $(\cdot, \cdot)$ is just the L^2–product, $a(\cdot, \cdot)$ is a symmetric bilinear form, and the eigenproblem is *selfadjoint*. If material losses are included into the model – as they should! – then the eigenproblem is generally *non–selfadjoint*.

Selfadjoint Helmholtz eigenproblems. In this case all eigenvalues λ are real and the eigenfunctions form an orthonormal basis. Technological interest focusses on clusters of the q lowest negative eigenvalues, which give rise to the q largest z–frequencies μ in the ansatz (2.7) and thus – via the dispersion relation – to the q lowest (x, y)–frequencies, i.e. to the q smoothest spatial modes. Multigrid methods for the solution of selfadjoint eigenproblems have been around for quite a time – see e.g. the cokernel projection method due to Hackbusch[52] or the *Rayleigh quotient minimization method* (RQM) due to Mandel/McCormick[67]. Asymptotically both approaches are equivalent. For the present challenging technological problem class, however, we nevertheless had to develop our own code, which is based on the latter MG approach. Let in finite dimension the invariant subspace associated with the eigenvalue cluster be represented by some orthogonal matrix U. Then RQM means

$$R(U) = \min_V R(V) \text{ with } R(V) = \text{ trace } \left((V^*BV)^{-1}(V^*AV) \right) . \tag{2.10}$$

In addition to adaptivity, we carefully studied and selected the smoother: we found out that (nonlinear) conjugate gradient methods are less sensitive to clustering of eigenvalues than e.g. Gauss–Seidel or Jacobi methods; moreover, *cg*–methods are better suited to start the subspace iteration – for details we refer to [34]. For coarse grid correction the canonical interpolation would do. By construction, global monotonicity of the MG method with respect to the Rayleigh quotient has been achieved – which, in turn, led to a high robustness of the algorithm even for rather poor initial coarse grids. A simple illustrating 1D example showing this increased robustness of the RQM MG method [67] over the projected MG method [52] can be found in [34].

Non–selfadjoint Helmholtz eigenproblem. In this case the eigenvalues λ are complex lying in a left bounded half stripe region of the complex plane such that $\mathrm{Re}\lambda_1 \leq \mathrm{Re}\lambda_2 \leq \ldots \to \infty$. The eigenvalues with lowest negative real part are those of technological interest. After proper discretization a non–Hermitian eigenvalue problem would arise. Looking over the fence into numerical linear algebra (cf. [50]), we expect to solve such eigenvalue problems via orthogonal transformations to a Schur normal form. As it turns out, we can actually follow this line of thought also in our present operator case, since the *complex* Helmholtz equation differs from its real counterpart only via the complexity of g. In other words, the complex Helmholtz operator is nearly selfadjoint – up to only a compact perturbation. In this situation a completeness result due to Katsnelson [60] states that (a) the corresponding spectrum is discrete and (b) there exists a *Schur basis* $\{u_j\}_{j=1}^{\infty}$ of $L^2(\Omega)$ such that

$$a(v, u_j) = \lambda_j \, (v, u_j) + \sum_{k=1}^{j-1} \tau_{kj} \, (v, u_k) \quad \forall v \in H_0^1(\Omega). \tag{2.11}$$

With this result in mind, we developed a generalization of the above MG method for the selfadjoint case. Of course, the solution of our problem now is no longer a minimum of the Rayleigh quotient, but still a stationary point. In the selfadjoint case, discretization by finite elements had led to a set of nested discrete eigenvalue problems of the type

$$\widetilde{A}\Theta = \widetilde{B}\Theta\widehat{\Lambda}$$

with the orthogonally projected matrices

$$\widetilde{A} = \left(\widetilde{U}\ \widetilde{P}\right)^* A \left(\widetilde{U}\ \widetilde{P}\right) \quad \text{and} \quad \widetilde{B} = \left(\widetilde{U}\ \widetilde{P}\right)^* B \left(\widetilde{U}\ \widetilde{P}\right) \quad .$$

Herein Θ denotes the eigenmode approximations, while the matrices $\widetilde{P}$ represent the simultaneous subspace iteration. The essential extension idea to the non–selfadjoint case now is to replace these projected eigenvalue problems in each smoothing step and each coarse grid correction by projected Schur problems

$$\widetilde{A}\Theta = \widetilde{B}\Theta\widehat{T} \quad ,$$

wherein the $\widehat{T}$ are now triangular matrices. Numerical experience confirms that in this way some smoothing can actually be realized. As in the simpler case the canonical coarse grid correction (interpolation) is taken. The thus modified algorithm resembles a block Arnoldi method [74]. The arising coarse grid problems are of the same type as the original problem but of smaller dimension. Therefore a recursive construction leads to a multigrid algorithm, which is schematically written down here [48].

Algorithm.

$[U_l, T_l] = \mathrm{MGM}(A_l, B_l, U_l, T_l, l)$

1. presmoothing: $U_l \to \widetilde{U}_l$, $T_l \to \widetilde{T}_l$
2. coarse grid correction: $\widetilde{U}_l \to \widehat{U}_l$, $\widetilde{T}_l \to \widehat{T}_l$
   - compute $A_{l-1} = V_l^* A_l V_l$ and $B_{l-1} = V_l^* B_l V_l$, where in case
     - $l = l_{\max}$: $V_l = \begin{pmatrix} \widetilde{U}_l & P_l \end{pmatrix}$
     - $l < l_{\max}$: $V_l = \begin{pmatrix} \widetilde{U}_l & \begin{matrix} 0 \\ P_l \end{matrix} \end{pmatrix}$
   - if
     - $l > 1$: $[U_{l-1}, T_{l-1}] = \mathrm{MGM}(A_{l-1}, B_{l-1}, \begin{pmatrix} I \\ 0 \end{pmatrix}, \widetilde{T}_l, l-1)$
     - $l = 1$: solve $A_0 U_0 = B_0 U_0 T_0$, $U_0^* B_0 U_0 = I$
   - set $\widehat{U}_l = V_l U_{l-1}$, $\widehat{T}_l = T_{l-1}$
3. postsmoothing: $\widehat{U}_l \to U_l$, $\widehat{T}_l \to T_l$

Herein the matrices A_l and B_l are the (augmented) system and mass matrices corresponding to FE spaces S_l. The matrices U_l with q columns and the upper triangular matrices T_l represent the unknowns. The prolongation matrices P_l perform the interpolation from the finer spaces S_l to the coarser spaces S_{l-1}.

The above algorithm is just a working algorithm in the sense that it has proved to be efficient in rather difficult application problems with multiscale structure and that experience has confirmed optimal MG complexity. A thorough theoretical investigation, however, is still missing.

Multi Quantum Well (MQW) Laser. For illustration purposes we here give the results of computations for a special integrated optical component, a MQW layer as part of some MQW laser. The cross section of the MQW layer is depicted in Fig. 2.5: the left hand structure has some length scale of about $10\mu m$, whereas the right hand zoom shows several monomolecular layers of about $10nm$ thickness.

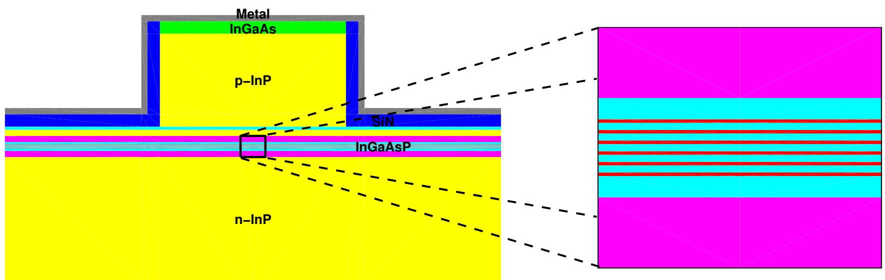

Figure 2.5: Cross section of MQW layer with zoom.

The scale factor of $1000 \approx 2^{10}$ must be spanned by *adaptive* grids – uniform grids would give rise to unreasonably large numbers of nodes. In our adaptive setting we started with coarse grids of about 2.500 nodes and ended up with finest grids of about 15.000 nodes – uniform grid refinement would have produced some estimated 2.500.000 nodes. In this problem $g(x,y) = k_0^2 n^2(x,y)$ with k_0 the vacuum wave number of light and $n(x,y)$ the different refractive indexes, which are complex valued within the MQW layer and the metal layers, but real valued otherwise. The exact design parameters must be kept confidential. In what follows we describe a typical technologically relevant parameter study about the dependence of the eigenvalues upon the imaginary part of some refractive index. We computed the four eigenvalues with lowest real parts for five parameter values – see Fig. 2.6. We also computed the corresponding invariant subspaces. Logarithmic contour plots of the finest mesh approximation of the two Schur modes u_1, u_2 associated with eigenvalues λ_1, λ_2 (see arrows in Fig. 2.6) are given in Fig. 2.7. Observe that u_1 is the symmetric fundamental mode, while u_2 is the anti–symmetric first order mode.

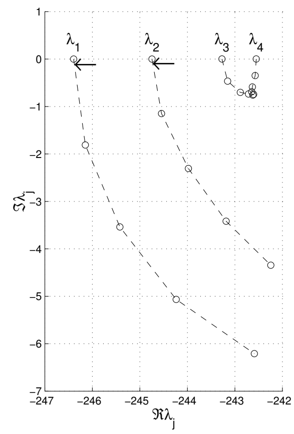

Figure 2.6: Dependence of eigenvalue cluster on imaginary part of refractive index.

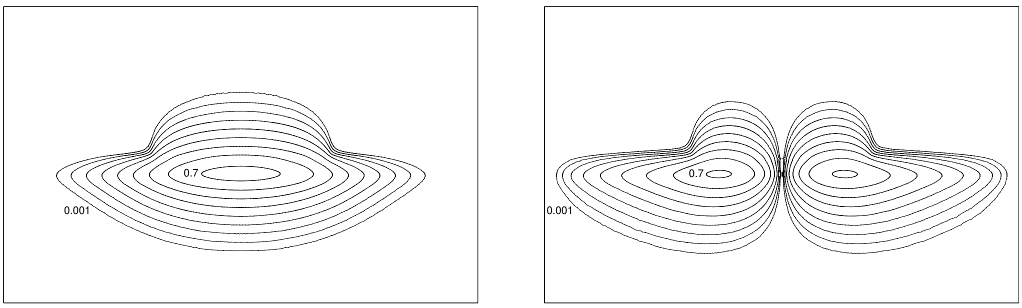

Figure 2.7: Logarithmic contour plots of $|u_1|^2$ and $|u_2|^2$.

3 Countable Ordinary Differential Equations in Polymer Industry

Chemically speaking, *polymers* are a special kind of macromolecules: chains of simple molecules or molecular groups, the *monomers*. The chains typically consist of ten thousand up to ten million monomers, say, and may be linear (the simpler case to be treated here) or even bifurcating. *Copolymers* are built from more than one type of monomer. Polymer materials include a variety of synthetic materials such as lacquers, adhesives, PVC, and MMA. Mathematically speaking, models of *polyreaction kinetics* involve a huge set of ordinary differential equations (ODEs), usually nonlinear and stiff. The numbers of ODEs again range from ten thousand up to ten million – one per each arising polymer length. For a mathematician it is simpler to think of countably infinitely many ODEs to be called *countable* ODEs or just CODEs. Even though CODEs are usually underrated if not totally overlooked in standard mathematical textbooks on differential equations, they play an important role in several scientific fields, e.g. in environmental science (soot formation), astrophysics (cosmic dust), or medicine (pharmacokinetics). In this section we will first describe the CODE models of typical polyreaction mechanisms. A survey of the basic computational approaches will follow. In more detail, we will then present the recent concept of *adaptive discrete Galerkin methods*. This concept has been first proposed in [45] by the author and Wulkow, who has then improved the method considerably in his thesis [87]. On this mathematical and software basis he had started a spin–off firm, which – after the usual critical initial phase – meanwhile consults in chemical industry and research labs all over the world.

3.1 Polyreaction Kinetics

In order to convey an impression of the CODE problem class, we begin with a short list of polyreaction mechanisms that arise in industrial applications. Only under unrealistic simplifications few of these CODEs can be solved in closed analytic form. In industrial applications, however, the mechanisms arise within combinations, which makes an analytical treatment anyway hopeless. That is why two realistic problems are also included here in some detail. As for the notation, let $P_s(t)$ be the concentration of polymers of *chain length* or *polymer degree* s at time t. For ease of writing, we will not distinguish between the chemical species P_s, its concentration $P_s(t)$, and the *chain length distribution* $\{P_s(t)\}_{s=1,2...}$, but just rely on the context.

Chain addition polymerization or free radical polymerization. With $M(t)$ denoting some monomer concentration, this special reaction mechanism is

$$P_s + M \xrightarrow{k_p} P_{s+1}, \quad s = 1, 2, \dots , \tag{3.1}$$

where $k_p > 0$ is the reaction rate coefficient. The kinetics of this reaction is modeled by

$$\begin{aligned} P_1' &= k_p M P_1 \\ P_s' &= k_p M (P_s - P_{s-1}), \quad s = 2, 3, \dots \\ M' &= k_p M \sum_{s=1}^{\infty} P_s \end{aligned} \tag{3.2}$$

with the given initial values

$$P_1(0) = P_{10}, \quad P_s(0) = 0, \quad s = 2, 3, \dots , \quad M(0) = M_0 \, . \tag{3.3}$$

Coagulation and irreversible polycondensation. This mechanism can be described in chemical terms as

$$P_s + P_r \longrightarrow P_{s+r} \tag{3.4}$$

and modeled mathematically by the *nonlinear* CODE

$$P_s' = \frac{1}{2} \sum_{r=1}^{s-1} k_{r,s-r} P_r P_{s-r} - P_s \sum_{r=1}^{\infty} k_{sr} P_{rs} \quad s = 1, 2, \dots \, . \tag{3.5}$$

Once again, the initial distribution $P_s(0)$ is usually given.

Example: Biopolymerization [22]. This problem deals with an attempt to recycle waste of synthetic materials in an ecologically satisfactory way – which is certainly an important problem of modern industrial societies. An attractive idea in this context is to look out for synthetic materials that are both produced and eaten by bacteria – under different environmental conditions, of course. A schematic illustration of the production process within such a bacterial recycling is given in Fig. 3.1: there certain bacteria use fructose as a chemical input to produce polyester (PHB) as chemical output. The macromolecular reaction steps

Figure 3.1: Biopolymerization: bacteria eat sugar and produce polyester. White areas: polyester granules within bacteria cells.

of production and degradation of PHB can be summarized in the *chemical model*

$$\begin{array}{ccccccc} & & E & \xrightarrow{k_a} & A & & \\ A & + & M & \xrightarrow{k_i} & P_1 & & \\ P_s & + & M & \xrightarrow{k_p} & P_{s+1} & & \\ & & P_s & \xrightarrow{k_t} & D_s & + & E \\ & & D_{s+r} & \xrightarrow{k_d} & D_s & + & D_r \end{array}$$

with $s, r = 1, 2, \ldots$. Herein M denotes the monomer fructose, E an enzyme, A the activated enzyme, P_s the so–called "living" and D_s the so–called "dead" PHB–polymer. The *mathematical model* for the above process comprises the CODE system

$$\begin{aligned} E' &= -k_a E + k_t \sum_{r=1}^{s_{max}} P_r \\ A' &= +k_a E - k_i A M \\ M' &= -k_p M \sum_{r=1}^{s_{max}} P_r - k_i A M \\ P_1' &= -k_p M P_1 + k_i A M - k_t P_1 \\ P_s' &= -k_p M (P_s - P_{s-1}) - k_d P_s \qquad , \quad s = 2, 3, \ldots, s_{max} \\ D_s' &= +k_t P_s - k_d (s-1) D_s + 2 k_d \sum_{r=s+1}^{s_{max}} D_r \quad , \quad s = 1, 2, \ldots, s_{max}. \end{aligned}$$

Herein the truncation index s_{max} is not known a priori, practical considerations lead to roughly $s_{max} = 50.000$ – which means that the above system consists of 100.000 ODEs, each of which has roughly the same number of terms in the right side.

Copolymerization. Most industrial synthetic materials are copolymers, typically consisting of three up to seven different sorts of monomers. The mathematical modelling of such systems is often performed in terms of some multidimensionsal ansatz $P_{sr\ldots}$ with s monomers of type A, r monomers of tape B etc. This ansatz, however, leads to an enormous blowup in terms of both computing time and storage. An alternative model has been suggested in the chemical literature (see [46] for reference). In this approach polymers are characterized by their chemically active site at one end of the chain. In the process of numerical solution of the polyreaction CODE enough information about these active sites comes up anyway. In this much simpler framework the following questions of chemical and economical relevance can still be answered: *Which portion of the monomer is consumed in the course of the reaction? What are the time dependent relative distributions of the different polymers?* A typical example with three monomers will be given at the end of Section 3.3 below.

3.2 Basic Computational Approaches

As exemplified in the previous Section 3.1, CODE initial value problems are in general infinite sets of nonlinear ODEs like

$$P_s'(t) = f_s(P_1(t), \ldots, P_s(t), P_{s+1}(t), \ldots) \,, \quad s = 1, \ldots \tag{3.6}$$

given together with initial values $P_s(0)$. Whenever the above right hand side f_s contains only arguments up to P_s, i.e. it has the form $f_s(P_1(t), \ldots, P_s(t))$, then the system can, in principle, be solved one by one – a property called *self–closing* in the literature. If f_s has the general form as above, then it is said to be *open*. The latter case is the typical one in real life applications. In what follows we will survey and assess the basic algorithmic concepts for the numerical solution of open CODEs.

Direct numerical integration. In not too complicated cases direct stiff integration of the reaction kinetics ODEs is still a popular approach. In fact, the author and former co–workers have developed the efficient software package LARKIN (for

LARge chemical KINetics) to tackle such systems, see [6]. Typically, with any such package, a sequential process is performed starting from a small number of ODEs and successively running up to larger and larger numbers. However, compared to direct integration even in this restricted application, the method to be presented in Section 3.3 has proved computational speed–up factors of more than 10.000 together with better accuracies for quantities of industrial relevance – compare [46]. Finally, since all stiff integrators require the Jacobian matrix of the right hand side, this approach suffers from a rather narrow domain of applicability, just think of the 100.000 by 100.000 nonzero Jacobian elements in the above biopolymerization example.

Lumping technique. In this kind of technique linear combinations of components are collected to certain supercomponents, for which then ODEs are derived and solved numerically. A proper collection of components requires a lot of a–priori insight into the process under consideration, sometimes just a $\log(s)$–equilibration is imposed. However, even though this technique is reported to work satisfactorily in some *linear* ODE cases, it is certainly totally unreliable for *nonlinear* ODEs, which represent the bulk of industrially relevant models.

Method of statistical moments. The canonical formulation for distributions – like $P_s(t)$ here – is in terms of statistical moments

$$\mu_k(t)[P] := \sum_{s \geq 1} s^k P_s(t), \qquad k = 0, 1, \ldots\,, \ .$$

Note that *mass conservation* shows up as $\mu_0(t) = \text{const}$. Insertion of the above definition into a polyreaction CODE (see Section 3.1 above) generates an infinite set of ODEs for these moments, a CODE again. It is easy to show that the structural property *open/selfclosing* of the original polyreaction CODE is passed on to the moment CODE. The whole approach is based on the theorem of Stieltjes, which states that the knowledge of *all* moments (if they are bounded) is equivalent to the knowledge of the distribution. If, however, only a finite number N of moments is known, in practice mostly only *few* moments, then there exists a full range of approximations $P_s^{(N)}$ with unclear representation and approximation quality.

One popular method to deal with this lack of information is to specify the distribution in advance. For example, assume that – on the grounds of scientific insight into the given problem – the unknown distribution is expected to be a Poisson distribution

$$P_s = C\, e^{-\lambda s}\, \frac{\lambda^{s-1}}{(s-1)!}\ ,$$

then the unknown parameters λ, C can be determined from just the two moments

$$C = \mu_0, \qquad \lambda = \frac{\mu_1}{\mu_0}\ .$$

Once the two moments are computed, the distribution as a whole seems to be known. Sometimes this kind of specification is also hidden behind so–called *closure relations*. However, strictly speaking, none of these approaches can assure that the stated problem is really solved – without further check of whether the assumptions made are appropriate. An insidious feature of any such approach is that all the thus computed approximations look optically smooth and therefore "plausible" – even when they are totally wrong! That is why these two approaches are only recommended for situations wherein the essential features of the solution are well–studied.

Last but not least, it is not clear how many terms need to be kept in the *truncated* moment CODE. Of course, the approximate moments $\mu_k^{(N)}$ corresponding to truncation index N should be accurate enough within some prescribed tolerance compared to the exact moments μ_k. The choice of truncation index N becomes even hopeless when the reaction rate coefficients depend on the chain length – as e.g. in soot formation, cf. [45, 88].

Monte Carlo method. Markov chains play some role in the computation of stationary solutions of copolymerization problems. In the application studied here, however, Monte Carlo methods require too much computing time in comparison with the method to be presented in Section 3.3. They will play an important role in Section 4.2 below in some different context.

Galerkin methods for continuous PDE models. A rather popular approach to condense the infinite number of ODEs is to model the degree s by some *real* variable, interpreting sums as infinite integrals and thus arriving at integro–partial differential equations, usually with lag terms (like e.g. the neutron transport equation). Following this line, an unknown modelling error for low s is introduced – which is the very part of the models where measurements are typically available. Moreover, depending on the polyreaction mechanism, the arising problem may turn out to be *ill–posed*, which in turn requires some regularization to be carefully studied. In [49], GAJEWSKI/ZACHARIAS suggested *Galerkin methods* in Hilbert space based on modified LAGUERRE polynomials L_k^α for the weight function

$$\Psi(s) = \sigma^\alpha e^{-\sigma} \quad \text{with} \quad \sigma = \beta(t)s, \quad \beta = \frac{\mu_0(t)}{\mu_1(t)} \; . \tag{3.7}$$

The specification of β assures *scaling invariance* in the (continuous) variable s.

Discrete Galerkin methods for CODEs. In [45] the author had suggested to use the above PDE approach only in principle, but to avoid turning the discrete variable s artificially into a continuous one. Upon interpreting discrete inner products as infinite sums, the discrete nature of the problem can be preserved – thus keeping the proper regularization. As a first attempt on this mathematical basis, discrete GALERKIN methods based on *discrete* LAGUERRE polynomials l_k for the weight function

$$\Psi(s) = \rho^s, \qquad \rho < 1 \; . \tag{3.8}$$

were constructed. The specification of ρ via "scaling" of the argument as in (3.7) is not directly possible here, since for a discrete variable s scaling is not a proper concept (see, however, the "moving weight function" concept below). This discrete Galerkin approach turned out to be the starting point for the construction of a new class of rather efficient algorithms to be discussed in detail in the next section.

3.3 Adaptive Discrete Galerkin Methods

For ease of presentation we replace the above nonlinear CODE (3.6) by the linear CODE

$$P_s'(t) = (\mathcal{A}P(t))_s \qquad P_s(0) \quad \text{given} \; . \tag{3.9}$$

Herein the discrete operator $\mathcal{A}$ describing the polyreaction mechanisms may be bounded (rare) or, unbounded (typical). The key to the construction of discrete Galerkin methods is the introduction of a *discrete inner product*

$$(f, g) := \sum_{s=1}^{\infty} f(s)g(s)\Psi(s) \tag{3.10}$$

in terms of some prescribed componentwise positive *weighting function* Ψ. This product induces a set of *orthogonal polynomials* $\{l_j\}, j = 1, 2, \ldots$ satisfying the relations

$$(l_j, l_k) = \gamma_j \delta_{jk} \qquad , \qquad \gamma_j > 0 \qquad j, k = 0, 1, 2, \ldots \tag{3.11}$$

For the solution P of the CODE we naturally try the corresponding ansatz

$$P_s(t) = \Psi(s) \sum_{k=0}^{\infty} a_k(t) l_k(s). \tag{3.12}$$

Moving weight function. There exists an interesting close connection between the statistical moments μ_k and the just introduced coefficients a_k. Upon representing the monomials s^k by the orthogonal set of polynomials, we are able to derive an infinite lower triangular system of algebraic equations of the form

$$\begin{array}{rcl} \mu_0 & = & b_{00}\gamma_0 a_0, \\ \mu_1 & = & b_{10}\gamma_0 a_0 + b_{11}\gamma_1 a_1, \\ \mu_2 & = & \dots\ , \end{array} \tag{3.13}$$

where $b_{kk} \neq 0$ is guaranteed. Herein the moments μ_k arise row–wise, whereas the coefficients a_k arise column–wise. This implies that if all moments are given and bounded, then all coefficients can be computed, which is just the already mentioned Stieltjes theorem. If only N moments are given, then only N coefficients can be computed. On the side of the coefficients, however, we have a reasonable option of *truncating* the expansion (3.12), since they (unlike the moments) are known to *decrease asymptotically*, if the solution P can actually be represented by the above ansatz. In other words, the solution P must be contained in some weighted sequence space, say H_Ψ, with an associated inner product and its induced norm

$$< f, g > := \sum_{s=1}^{\infty} f(s) g(s) / \Psi(s), \qquad \|f\|^2 := < f, f > \ . \tag{3.14}$$

In order to enforce that $P \in H_\Psi$ throughout the whole evolution, we additionally require that $P \approx \mu_0 \Psi$ by imposing the *moving weight function* conditions

$$\nu_0[\Psi] = 1, \qquad \nu_1[\Psi] = \frac{\mu_1[P]}{\mu_0[P]} \tag{3.15}$$

wherein the ν_k denote the statistical moments of the prescribed weight function. The first condition means that Ψ is some *probability density function*, whereas the second one assures some time dependent coupling of the mean values of the unknown distribution P to the known distribution Ψ – hence the name. Insertion of (3.15) into (3.13) leads to the two equivalent conditions

$$a_0(t) = \mu_0(t), \qquad a_1(t) = 0 \tag{3.16}$$

independent of the underlying problem and of the choice of weight function. Summarizing, with these two additional conditions a much smaller number n of degrees of freedom turned out to be sufficient to characterize the dynamics of realistic polyreaction systems.

Method of lines. Upon insertion of the expansion (3.12) into the CODE (3.9), multiplication by the test function $l_j(s)$, summation over s, change of summation order, and use of the above orthogonality relations, we end up with the CODE

$$\gamma_j a_j'(t) = \sum_{k=0}^{\infty} a_k(t)(l_j, \mathcal{A} l_k) \quad j = 0, 1, \dots . \tag{3.17}$$

for the a_k. In the above moving weight function approach the ODE for a_1 can be dropped (recall that $a_1 = 0$) and a new ODE can be created instead, say, for the parameter ρ in the discrete LAGUERRE method based on Ψ as in (3.8). In other words, in this method of lines type approach the moving weight function induces a *moving basis* $l_k(s; \rho(t)), k = 0, 1, \dots$ – similar to moving nodes in PDE applications with moving fronts.

For the numerical realization of (3.17), the inner products

$$(l_j, \mathcal{A} l_k) = \sum_{s=1}^{\infty} l_j(s) \mathcal{A}(s) l_k(s) \Psi(s)$$

must be approximated by a finite number of terms to prescribed accuracy. In rare cases analytical methods allow a closed form representation, which is cheap to evaluate. In most cases, however, numerical approximations turn out to be the only choice. We developed two efficient approximation methods: an *adaptive multigrid summation technique* (compare Section 9.7.2 in the undergraduate numerical analysis textbook [35]), a discrete variant of the MG method described in Section 1.1, and a *Gauss–Christoffel summation*, a discrete variant of Gauss–Christoffel quadrature based on the selected weight function Ψ. As for the applied time discretization, *linearly implicit* schemes in the generally nonlinear CODEs (3.6) turned out to be efficient to tackle the unbounded part of the discrete Fréchet operators of the right hand side. In addition, adaptive timestep and order control is as useful as in ordinary stiff integration.

Adaptive truncation. Truncation of this CODE by setting

$$a_j = 0, \qquad j = n, \quad n+1, \quad \dots$$

leads to the discrete GALERKIN approximation

$$P_s^{(n)}(t) = \Psi(s) \sum_{k=0}^{n} a_k^{(n)}(t) l_k(s). \tag{3.18}$$

In the simpler case of *self–closing* CODEs, the coefficients a_k do *not* depend on the truncation index n. The key to *adaptivivity* in any (global) Galerkin method is that the *truncation error* can be estimated by

$$\begin{aligned} & \left|P^{(n)} - P\right| \doteq \left|P^{(n)} - P^{(n+1)}\right| = \\ & \qquad \left[(a_{n+1}^{(n)})(t)^2\gamma_{n+1} + \sum_{k=0}^{n}\left(a_k^{(n)}(t) - a_k^{(n+1)}(t)\right)^2\gamma_k\right]^{\frac{1}{2}} \ . \end{aligned} \tag{3.19}$$

Note that if we want to assure a condition like

$$\left|P^{(n)} - P^{(n+1)}\right| \leq \ \mathrm{TOL}$$

for some prescribed error tolerance TOL, then we might thus obtain some time dependent number $n(t)$ of terms in the Galerkin approximation.

Adaptive Rothe method. As already tacitly indicated in the above moving weight function conditions, the typical CODE initial value problem lives in some *scale* of Hilbert spaces, say $H_{\rho(t)}, t \geq t_0$ for the discrete Laguerre method, rather than just in single fixed space. This nicely shows up in a special Lipschitz condition, a nonlinear extension of a semi–continuity assumption:

$$\begin{aligned} \|f(t,u) - f(t,v)\|_{\bar{\rho}} & \leq \frac{M}{(\bar{\rho}-\rho)^\gamma}\|u-v\|_\rho, \quad 0 < \gamma \leq 1, \\ \bar{\rho} & > \rho \quad \text{and} \quad u, v \in H_\rho \ . \end{aligned} \tag{3.20}$$

This condition is the essential ingredient of a uniqueness theorem for CODEs – see [87]. In analogy to the discussion for PDEs in Section 1.1 above, the appropriate order of discretization will therefore be *first time discretization, then Galerkin approximation*, which is the so–called Rothe method. Starting from some initial value $u(t) = \phi$ suppose we apply some linearly implicit Euler discretization

$$(I - \tau\mathcal{A})\Delta u = \tau f(\varphi) \ , \quad u_1 = \varphi + \Delta u \ , \tag{3.21}$$

wherein $\mathcal{A}$ is the derivative $f_u(\varphi)$. This is a linear boundary value problem in some discrete sequence space to be treated numerically by a discrete Galerkin method. The numerical realization of the time step control requires an estimate η_1 of the error $\|u_1 - u(t+\tau)\|$ in the norm $\|\cdot\|$ induced by the corresponding inner product. For this purpose we solve the correction equation

$$(I - \tau\mathcal{A})\eta_1 = -\frac{1}{2}\tau^2\mathcal{A}f(\varphi) \ , \quad u_2 = u_1 + \eta_1 \ , \tag{3.22}$$

which obviously has the same structure as (3.21). The approximation u_2 is of order 2 in time, which implies the estimated optimal time step (with safety factor $\sigma < 1$)

$$\tau_{\text{new}} = \tau \sqrt{\frac{\sigma \tau_{\text{tol}}}{\|\eta_1\|}} \,. \tag{3.23}$$

In this realization the truncation index for (3.22) may correspond to a less stringent (absolute) error tolerance than the one for (3.21), since (3.23) requires less accuracy; in this setting the approximation u_1 is regarded as the accepted solution. If the same absolute accuracy is prescribed in both equations, then u_2 is taken as the accepted approximation.

Discrete h-p–method. For sufficiently complex problems in industry *global* Galerkin methods – such as the one given in (3.12) – have meanwhile been clearly outperformed by *local* Galerkin methods (similar as FEMs in PDEs). In order to be able to construct local multilevel bases, the weight function has to be restricted to $\Psi = 1$ on finite subintervals, which induce (discrete) Chebyshev polynomials t_k of degree k. The actually developed method combines adaptive interval refinement (h–method) with adaptive choice of the local degree (p–method) to obtain some rather sophisticated *adaptive h-p–method* on some finite interval. In order to match the asymptotic tail of the distribution, global representations still play a role.

In the course of the years the successive efficient treatment of industrially relevant polyreaction problems has led to a steady increase of modern rather sophisticated algorithmic tools that made their way into the commercial software package Predici$^{\text{TM}}$ of Wulkow [88]. The following copolymerization example with three monomers should give some flavor of such problems.

Example: Radical terpolymerization [46]. Let P_s be the polymer with active end MMA (index a), Q_s the one with styrole (index b), and R_s the one with MSA (index c below). With this notation, the chemical reaction scheme reads

Initialization:

$$\begin{array}{lcl} I & \xrightarrow{k_d} & 2I* \\ I^* + MMA & \xrightarrow{k_I} & P_1^* \\ I^* + S & \xrightarrow{k_I} & Q_1^* \\ I^* + MSA & \xrightarrow{k_I} & R_1 \end{array}$$

Chain growth:

$$
\begin{array}{llll}
P_s^* & + & MMA & \xrightarrow{k_{paa}} & P_{s+1}^* \\
P_s^* & + & S & \xrightarrow{k_{pab}} & Q_{s+1}^* \\
P_s^* & + & MSA & \xrightarrow{k_{pac}} & R_{s+1}^* \\
Q_s^* & + & MMA & \xrightarrow{k_{pba}} & P_{s+1}^* \\
Q_s^* & + & S & \xrightarrow{k_{pbb}} & Q_{s+1}^* \\
Q_s^* & + & MSA & \xrightarrow{k_{pbc}} & R_{s+1}^* \\
R_s^* & + & MMA & \xrightarrow{k_{pca}} & P_{s+1}^* \\
R_s^* & + & S & \xrightarrow{k_{pcb}} & Q_{s+1}^*
\end{array}
$$

Chain termination:

$$
\begin{array}{lllll}
P_s^* & + & P_r^* & \xrightarrow{k_{caa}} & D_{s+r} \\
P_s^* & + & Q_r^* & \xrightarrow{k_{cab}} & D_{s+r} \\
Q_s^* & + & Q_r^* & \xrightarrow{k_{cbb}} & D_{s+r}
\end{array}
$$

For reasons of confidentiality some parts of the mechanism are left out. Initial values and generic reaction rate coefficients are given in [46]. Under the modelling assumption that in this specific polymerization process the living chains live only for a very short time span, the copolymer distribution can be computed as follows. Define e.g. integrated quantities like

$$\bar{P}_s(t) = \int_0^t P_s(t)dt \ .$$

which (up to normalization) counts the number of coplymers with MMA in fixed position s. Then the relation $\bar{P}_s : \bar{Q}_s : \bar{R}_s$ offers detailed insight into the composition of the copolymer. In Fig. 3.2 such compositions are shown for times $t = 360$ min and $t = 1.080$ min. Information of this kind could not have been gained from statistical moment analysis as applied earlier in industry.

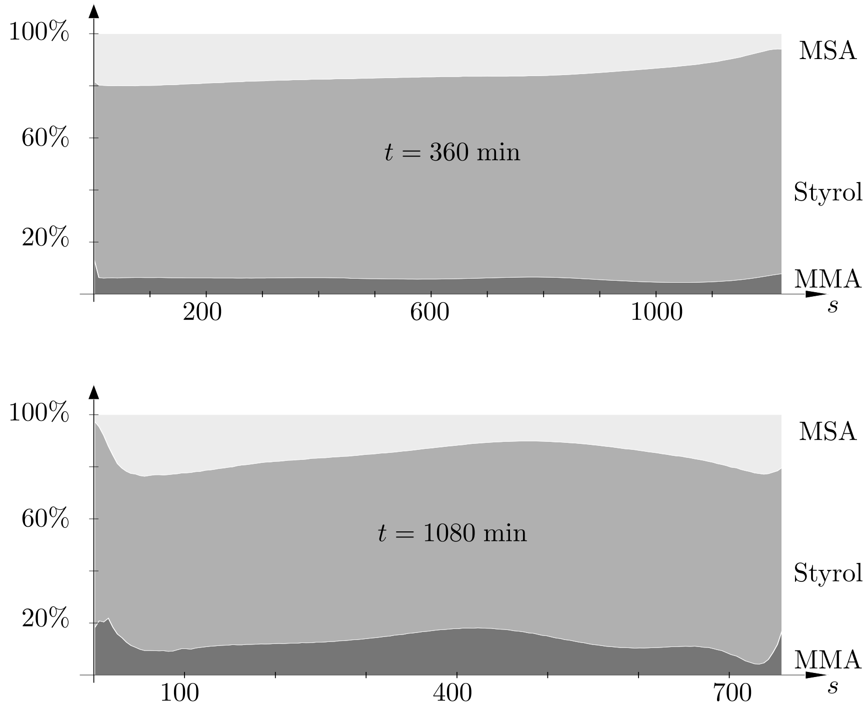

Figure 3.2: Copolymer composition versus chain length for two times t.

4 Hamiltonian Equations in Pharmaceutical Drug Design

The design of highly specific drugs on the computer, the so-called *rational drug design* (as opposed to irrational drug consumption), is a fairly recent dream of biochemistry and pharmaceutical industry. Typically, a lot of heuristics go with this problem, which we skip here. At first mathematical glance, drug design seems to involve the numerical integration of the Hamiltonian differential equations that describe the dynamics of the molecular system under consideration. Following this idea, a huge discrepancy of time scales shows up: phenomena of interest, such as *protein folding* or *active site docking*, occur on a micro- or millisecond scale, whereas present routine computations only cover time spans of up to a few nanoseconds (at best). This gap has stimulated a lot of work in an interdisciplinary field including Numerical Analysis, Statistical Physics, Biochemistry, and Dynamical Systems. The present section reports about some recent and still ongoing collaboration of the author and his group with internationally renowned RNA biochemists including biotech firms in the Berlin region.

Our special contribution to drug design starts from the known insight that the corresponding trajectories are *chaotic*, which means that small perturbations of the initial values of such trajectories *asymptotically* lead to unbounded deviations. In terms of Numerical Analysis this means that the Hamiltonian initial value problems (IVPs) are *ill–conditioned after short time spans*. On the basis of this insight, we suggested a novel concept for the computation of *essential* features of Hamiltonian dynamical systems. The key idea presented in Section 4.1. below is to directly compute *chemical conformations* and rates of conformational changes, interpreting chemical conformations as *almost invariant sets* in the *phase space* (positions *and* momenta) of the corresponding dynamical system. In a first step, this led to an eigenproblem for eigenvalue clusters around the Perron root of the so-called Frobenius-Perron operator associated with the (numerical) flux of the dynamical system.

In a second step, in Section 4.2., we interpreted chemical conformations as objects in the *position space* of the Hamiltonian dynamical system. Moreover, we abandoned the deterministic Hamiltonian systems with given initial values to turn over to ensembles of initial values in the frame of Statistical Physics. This led to the natural construction of a stochastic operator, which appears to be selfadjoint over some weighted L^2-space. Discretization of that operator by means of certain *hybrid Monte Carlo* methods (HMC) generates nearly uncoupled Markov chains that need to be computed. As it turns out, the eigenvectors associated with the Perron cluster of eigenvalues for the stochastic operator contain the desired information about the chemical conformations and their patterns of change.

The described approach is presently worked out in collaboration with biochemists that design RNA drugs in their chemical labs. Our aim is to substitute time consuming and costly experiments in the chemical RNA lab by reliable simulations in a *virtual RNA Lab*. First steps in this direction are illustrated in Section 4. 3 at some fairly complex, but moderate size RNA molecule.

4.1 Deterministic Chaos in Molecular Dynamics

In classical textbooks on Molecular Dynamics (MD), see e.g. [1], a single molecule is modelled by a Hamilton function

$$H(q,p) = \tfrac{1}{2} p^T M^{-1} p + V(q) \qquad (4.1)$$

with q the 3D atomic positions, p the (generalized) momenta, M the diagonal mass matrix, and V a differentiable potential function. The Hamilton function is called *separated*, whenever the p-part and the q-part of H are separated as above.

The Hamilton function defined on the phase space $\Gamma \subset \mathbb{R}^{6N}$ induces the canonical equations of motion, the *Hamiltonian equations*

$$\dot{q} = M^{-1}p, \qquad \dot{p} = -\mathrm{grad}\, V \,, \tag{4.2}$$

which describe the dynamics of the molecule in a deterministic way: For given initial state $x_0 = (q(0), p(0))$ the unique formal solution of (4.2) is usually written as $x(t) = (q(t), p(t)) = \Phi^t x_0$ in terms of the *flow* Φ^t. Numerical integration of (4.2) by any one-step method with stepsize τ leads to the discrete solution

$$x_{k+1} = \Psi^\tau x_k \quad \Rightarrow \quad x_k = (\Psi^\tau)^k \, x_0, \tag{4.3}$$

in terms of a *discrete flow* Ψ^τ.

Condition of the initial value problem. Given an initial perturbation δx_0 we are interested in its growth along the flow

$$\delta x(t; x_0) = \Phi^t(x_0 + \delta x_0) - \Phi^t x_0 \ .$$

The *condition number* $\kappa(t)$ of an initial value problem (see the textbook by Deuflhard and Bornemann [32]) may be defined as the worst case error propagation factor in first order perturbation analysis so that (in some suitable norm $|\cdot|$)

$$|\delta x(t; x_0)| \; < \; \kappa(t) |\delta x_0| \quad \text{for all } x_0.$$

It is of utmost importance to keep in mind that $\kappa(t)$ is a quantity characterizing the analytic problem independent of any discretization. A *linear growth* result $\kappa(t) \sim t$ holds for a subclass of so–called *integrable* Hamiltonian systems such as the popular Kepler problem – see Arnold [3]. In real life MD problems, however, *κ increases exponentially.* In order to illustrate this behavior, Fig. 4.1 shows results for the simple Butane molecule. In order to be able to ignore any discretization error effects, unusually small time steps (τ = 0.005 fs) within the Verlet scheme have been chosen. A physically negligible initial perturbation 10^{-4}Å can be seen to overgrow the nominal solution after a time span $T > 500$ fsec, which is significantly shorter than the time spans of physical interest.

Once *long term* trajectories in MD have been identified as ill-conditioned mathematical objects, they should be avoided in actual computation. Only *short term* trajectories should be accepted as numerical input for further scientific interpretation. This seems to be in direct contradiction to the possible prediction of long term behavior of biomolecules! How can we overcome this difficulty?

Warning. There are quite a number of beautiful movies about the dynamics of biomolecules that visualize merely accidental numerical results of long term simulations – and are therefore of doubtful value for chemical interpretation.

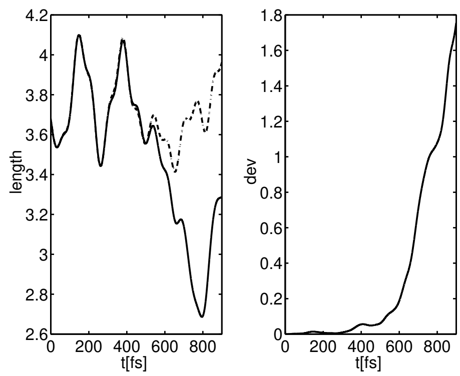

Figure 4.1: Two dynamical simulations for the Butane molecule with initial deviation 10^{-4}Å. Left: Evolutions of the total molecule length (in Å). Right: Dynamics of the deviation (in Å).

Multiscale structure. In order to gain more insight, we proceed to a rather instructive example due to Grubmüller and Tavan [51]. Figure 4.2 describes the dynamics of a polymer chain of 100 CH_2 groups. Possible timesteps for numerical integration are confined to $\tau < 10$ fsec due to fast oscillations. Time scales of physical interest range from 10^3 to 10^5 psec in this problem, which is a factor 10^5 to 10^7 larger. The figure presents six different zoom levels in time, each of which scales up by a factor 10. On the smaller time scales (upper levels) the dynamical behavior is characterized by nonlinear *oscillations* around certain vague "equilibrium positions". On larger and larger time scales these oscillations become less and less important. On the largest time scale (lowest level) we observe a kind of flip–flop behavior between two "conformations".

This observation suggests an alternative to long term trajectory simulation: the *essential dynamical pattern* of a molecular process could as well be modelled by *probabilities* for the molecular system to stay within different *conformations*. From a chemical point of view, conformations describe clusters of *geometric* configurations associated with some specified chemical *functionality*. In a conformation, the large scale geometric structure of the molecule is understood to be conserved, whereas on smaller scales the system may well rotate, oscillate, or fluctuate. In order to understand a chemical system, conformations and their average life spans are the main objects of chemical interest. Therefore the *direct computation of conformational dynamics* seems to be the concept to pursue.

Frobenius–Perron operator. In a first approach, we recurred to the so-called Frobenius-Perron operator U associated with the flow Φ^τ. This operator is defined on the set $\mathcal{M}$ of probability measures over the phase space Γ by virtue of

$$(U\mu)(G) = \mu(\Phi^{-\tau}(G)) \quad \text{for all measurable } G \subset \Gamma \text{ and arbitrary } \mu \in \mathcal{M}. \quad (4.4)$$

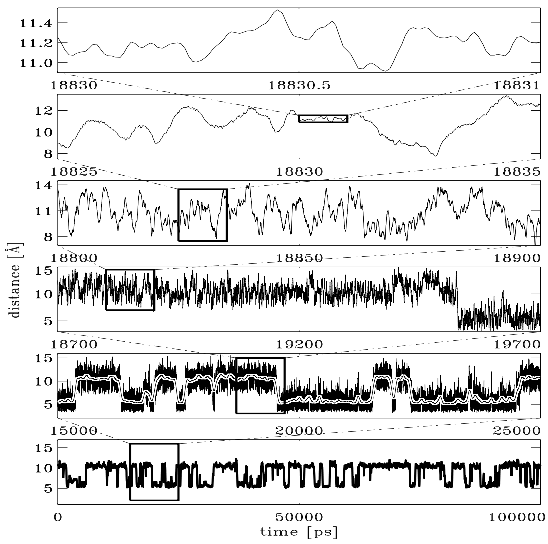

Figure 4.2: MD simulation of a polymer chain of 100 CH_2 groups due to [51]. Time scale zoom factor 10 from level to level.

Invariant sets are the union of all those states that a dynamical system can reach, i.e. they correspond to an *infinite* duration of stay. Moreover, they are fixed points of U associated with the so-called PERRON eigenvalue $\lambda = 1$ of U. In [33], we interpreted conformations as *almost invariant subsets in phase space* corresponding to *finite, but still large* duration of stay – a phenomenon often called *metastability*. In the spirit of earlier work of DELLNITZ/JUNGE [29] we analyzed the connection of almost invariant sets with eigenmodes to real eigenvalue clusters around the Perron root, to be called *Perron clusters* hereafter – see DEUFLHARD, HUISINGA, FISCHER, AND SCHÜTTE [37].

Following HSU [57] the discretization of the Frobenius-Perron operator is done on a box covering $\{G_1, \dots , G_n\}$ associated with characteristic functions χ_{G_i}. Since the flow Φ^τ conserves energy, these boxes would be expected to subdivide the *energy surface* $\Gamma_0(E) = \{x \in \Gamma : H(x) = E\}$. However, a typical discrete flow $(\Psi^{\tau/k})^k$ with k steps will *not* conserve energy exactly – even symplectic discretizations preserve energy only on average over long times. Therefore the boxes have to subdivide *energy cells* defined by

$$\Gamma_\delta(E) = \{x \in \Gamma, \, |H(x) - E| \leq \delta\}$$

in terms of some perturbation parameter δ. With $f = (\Psi^{\tau/k})^k \approx \Phi^\tau$ the discretized

Frobenius-Perron operator $U_n = (u_{i,j})$ can be written componentwise as

$$u_{ij} = \frac{m(f^{-1}(G_i) \cap G_j)}{m(G_j)}, \quad i = 1, \dots, n, \tag{4.5}$$

where m denotes the LEBESGUE measure; roughly speaking, this means that $m(G_j)$ is the volume of the box G_j. The geometric situation as a whole is illustrated in Fig. 4.3. The approximations of the various volumes are performed by some equidistributed Monte Carlo method. In order to speed up computations, a nested sequence of boxes is constructed for an *adaptive multilevel box method* called subdivision algorithm by DELLNITZ/HOHMANN [28].

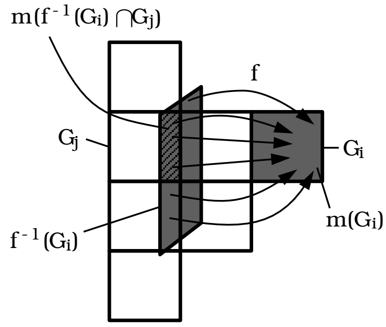

Figure 4.3: Stochastic matrix element $u_{i,j}$ computed from (4.5)

The realization of this first attempt, however, turned out to suffer from two important disadvantages. The first one is of a theoretical nature: since the Frobenius–Perron operator for a deterministic Hamiltonian system is *unitary* in $L^2(\Gamma)$, real eigenvalues inside the unit circle cannot exist. But they did exist and had been computed and interpreted in our method! This comes from the fact that, by subdividing energy cells rather than the energy surface, we had allowed for stochastic perturbations of the deterministic systems; in this more general setting, eigenvalues could show up also inside the unit circle and did contain the information wanted. Consequently, to model this situation correctly, a more general stochastic theory would be needed – which is still missing. Second, this approach obviously causes some *curse of dimension*, which prevents the method to be applicable to realistic molecules. To understand this perhaps unexpected and certainly undesirable effect, recall that the described subdivision method had been successfully developed in [29] for hyperbolic systems, where the dynamics is known to collapse asymptotically to some low dimensional attractor; this feature, however, does not carry over to Hamiltonian systems.

4.2 Identification of Metastable Conformations

This section describes a rather recent improved approach mainly due to Schütte [80, 81]. This approach keeps the conceptual advantages of the Dynamical Systems approach (as given in Section 4.1), but avoids the conceptual disadvantages by exploiting concepts of Statistical Physics instead. Its key new feature is the replacement of the Frobenius-Perron operator by some well-designed Markov operator, a *spatial transition operator* based on Boltzmann statistics. Most experiments in a chemical lab are performed under the conditions of constant temperature and volume, which is known to give rise to the *canonical density*

$$f_0(x) \;=\; \frac{1}{Z}\exp\left(-\beta\, H(x)\right), \quad \text{with} \quad Z = \int_\Gamma \exp\left(-\beta\, H(x)\right) dx, \tag{4.6}$$

where $\beta = 1/k_B\mathcal{T}$, temperature $\mathcal{T}$, and Boltzmann's constant k_B.

We start with some notion of *almost invariant* sets in the language of statistics. For some selected set S let χ_S denote its characteristic function, i.e. $\chi_S(x) = 1$ iff $x \in S$ and $\chi_S(x) = 0$ otherwise. Then the *transition probability* between two subsets S_1, S_2 of the phase space Γ is given by

$$w(S_1, S_2, \tau) \;=\; \frac{1}{\int_{S_1} f_0(x)\,dx}\int\limits_{S_1} \chi_{S_2}(\Phi^\tau x)\, f_0(x)\,dx \qquad S_1, S_2 \subset \Gamma \tag{4.7}$$

By this definition almost invariant sets will be those with $w(S,S,\tau) \approx 1$. In Section 4.1 we had understood chemical conformations as almost invariant sets in phase space. However, in reality these objects are observed in *position space*. That is why we now turn over to characterize conformations as sets B in position space $\Omega \subset \mathbb{R}^{3N}$. Upon allowing for arbitrary momenta p, we are naturally led to focus our interest on the *phase space fiber*

$$\Gamma(B) \;=\; \{(q,p) \in \Gamma, \quad q \in B\}\;. \tag{4.8}$$

With this notation we now call a set $B \subset \Omega$ a *conformation* whenever

$$w(\Gamma(B), \Gamma(B), \tau) \approx 1. \tag{4.9}$$

For H *separable* as in (4.1), f_0 from (4.6) splits into the product

$$f_0(x) \;=\; \underbrace{\frac{1}{Z_p}\exp\left(-\frac{\beta}{2}\,p^T M^{-1} p\right)}_{=\mathcal{P}(p)}\; \underbrace{\frac{1}{Z_q}\exp\left(-\beta\, V(q)\right)}_{=\mathcal{Q}(q)}, \tag{4.10}$$

with $\mathcal{P}$ and $\mathcal{Q}$ normalized such that

$$\int \mathcal{P}(p)dp = \int \mathcal{Q}(q)dq = 1.$$

Upon returning to (4.8), we may specify the conditional probability for a system being in set $B \subset \Omega$ to move to set $C \subset \Omega$ after some time τ as

$$w(\Gamma(B), \Gamma(C), \tau) \; = \; \frac{1}{\int\limits_{\Gamma(B)} f_0(x)\, dx} \int\limits_{\Gamma(B)} \chi_{\Gamma(C)}(\Phi^\tau x)\, f_0(x)\, dx \; . \tag{4.11}$$

Construction of spatial Markov operator. With these preparations we are now ready to confine all terms to position space only. The denominator in (4.11) represents the probability for a system from the statistical ensemble *to be within* $B \subset \Omega$ which can be simplified using $dx = dqdp$ (for H separable) and the normalization of $\mathcal{P}$ to yield

$$\pi(B) \; = \int\limits_{\Gamma(B)} f_0(x)\, dx \; = \int\limits_{B} \mathcal{Q}(q)\, dq \; . \tag{4.12}$$

The analog treatment of the numerator in (4.11) will give rise to the definition of an operator T. Let $\xi_1(q,p) = q$ denote the projection from the variables x to the position variables and introduce an inner product in the *weighted* Hilbert space $L^2_{\mathcal{Q}}$ by

$$\langle u, v \rangle_{\mathcal{Q}} \; = \; \int_\Omega u^*(q)\, v(q)\, \mathcal{Q}(q)\, dq$$

together with its induced norm

$$\|u\|^2_{\mathcal{Q}} = \langle u, u \rangle_{\mathcal{Q}} \; .$$

With this notation we may write

$$\int\limits_{\Gamma(B)} \chi_{\Gamma(C)}(\Phi^\tau x)\, f_0(x)\, dx \; = \; \underbrace{\int\limits_{C} \left\{ \int_{\mathrm{IR}^{3N}} \chi_B(\xi_1 \Phi^\tau(q,p))\, \mathcal{P}(p)\, dp \right\} \mathcal{Q}(q)\, dq}_{=: \langle T\chi_B, \chi_C \rangle_{\mathcal{Q}}} \tag{4.13}$$

thus defining the operator

$$Tu(q) \; = \; \int u(\xi_1 \Phi^\tau(q,p))\, \mathcal{P}(p)\, dp, \tag{4.14}$$

Upon combining (4.11) up to (4.13) we may write the conditional probability of a system being in B to move into C during time τ as

$$w(B, C, \tau) = \frac{\langle T\chi_B, \chi_C \rangle_{\mathcal{Q}}}{\pi(B)} \qquad B, C \subset \Omega \tag{4.15}$$

By construction, T can be interpreted as the restriction of the Frobenius–Perron operator U, see (4.4), to position space via averaging over the momentum part of the canonical distribution. The operator T is defined on the weighted spaces

$$L^r_{\mathcal{Q}}(\Omega) = \{u : \Omega \to \mathcal{C}, \quad \int_\Omega |u(q)|^r \mathcal{Q}(q)\, dq < \infty\}, \qquad r = 1, 2.$$

In terms of these spaces the important properties of T are (Schütte [80]):

1. T is bounded in $L^p_{\mathcal{Q}}(\Omega)$: $\|Tu\|_{\mathcal{Q}} \leq \|u\|_{\mathcal{Q}} \quad p = 1, 2$
2. T is a Markov operator on $L^1_{\mathcal{Q}}(\Omega)$.
3. T is *selfadjoint* in $L^2_{\mathcal{Q}}$, since Φ^τ is *reversible*. Hence, the spectrum $\sigma(T)$ is real-valued and bounded: $\sigma(T) \subset [-1, 1]$.
4. There exists a *Perron cluster* of discrete eigenvalues well-separated from the remaining (continuous) part of the spectrum.

Properties 1-3 hold for Hamiltonian systems in general; property 4 only holds under additional assumptions – which are, however, satisfied in the systems of interest here. Summarizing, T has just the theoretical properties needed as a basis for the computational identification of conformational subsets via the eigenmodes to Perron clusters of eigenvalues.

Discretization of Markov operator. We proceed as in the previous section by introducing a set of boxes, this time in position space rather than phase space. Let $\{G_1, \ldots, G_n\} \subset \Omega$ denote a *covering* of Ω. In view of a Galerkin approximation in $L^2_{\mathcal{Q}}$ we define the basis $\mathcal{V}_n = \text{span}\{\chi_1, \ldots, \chi_n\}$ in terms of the characteristic functions $\chi_i = \chi_{G_i}$. Moreover, let $\pi(G_i) = \pi_i$. In this basis we may construct an (n, n)-matrix $P = (p_{ij})$ via spatial transition probabilities as

$$p_{ij} = \frac{\langle T\chi_i, \chi_j \rangle_{\mathcal{Q}}}{\pi_i} = w(G_i, G_j, \tau) \qquad i, j \in \{1, \ldots, n\}. \tag{4.16}$$

By construction, this matrix P is row-wise stochastic and, since T is self-adjoint, also *reversible*. Upon observing the condition of *detailed balance* in the form

$$\pi_i\, p_{ij} = \pi_j\, p_{ji}, \qquad \forall i, j \in \{1, \ldots, n\}.$$

the matrix is *symmetric* with respect to a weighted discrete inner product, which implies that $\sigma(P) \subset [-1, 1]$, real. Moreover, there exists a Perron cluster of eigenvalues, if the boxes are well-chosen to really cover the relevant position space.

Hybrid Monte Carlo realization. The above introduced MARKOV operator T in $L^1_{\mathcal{Q}}(\Omega)$ can also be interpreted as a transition operator that generates a MARKOV *chain* $\{q_k\}_{k=0,1,\dots}$ via the *discrete stochastic dynamical system*

$$q_{k+1} = \xi_1 \Phi^\tau(q_k, p_k), \quad k = 0, 1, \dots, \tag{4.17}$$

wherein the momenta p_k in each step are chosen randomly from the distribution $\mathcal{P}$. Obviously, upon changing from the Frobenius-Perron operator U to the transition operator T, we also change from the discrete *deterministic* dynamical system (4.3) to its stochastic counterpart (4.17). Iterations of (4.17) realize sequences $\{q_k\}$ that are asymptotically distributed according to $\mathcal{Q}$. This feature can be exploited by application of suitable *Monte Carlo* (MC) methods. After M samples $\{q_j\}$ from (4.17) the $\mathcal{Q}$-expectation value of any *spatial observable* $\mathcal{A} : \Omega \to \mathbb{R}$ is approximated by an averaged sum such that

$$\Big| \frac{1}{M} \sum_{j=1}^{M} \mathcal{A}(q_j) - \int_{\Omega} \mathcal{A}(q) \mathcal{Q}(q)\, dq \Big| \le C\, M^{-1/2}, \tag{4.18}$$

with a constant C not explicitly depending on $\dim(\Gamma) = 6N$. In the present context, the observable will be any of the $\frac{1}{2}n(n+1)$ integrand factors $T\chi_i\chi_j$ arising in (4.16). As the integrand contains the (short term numerical) flux, a *hybrid Monte Carlo* (HMC) method is preferably applied, which is a compromise between trajectory evaluation and MC. In more detail, the matrix elements p_{ij} are asymptotically approximated by virtue of *relative frequencies*

$$\frac{\#\,(q_k \in G_i \wedge q_{k+1} \in G_j)}{\#\,(q_k \in G_i)} \to w(G_i, G_j, \tau) = p_{ij} \qquad i, j \in \{1, \dots, n\}. \tag{4.19}$$

The theoretical result (4.18) raises the expectation that we have eventually overcome the *curse of dimension.* However, the well-known undesirable effect of *critical slowing down* of the iteration due to local trapping is still possible. For this reason, we developed an *adaptive temperature* variant called ATHMC – see FISCHER, CORDES, AND SCHÜTTE [47].

Essential degrees of freedom. In realistic RNA drug molecules we have to face around $N > 100$ atoms – see Fig. 4.5. If we subdivide each of the $3N$ state variables into m pieces, then the dimension of the stochastic matrix P would be $n = m^{3N}$ – which would be just too much even for a Krylov space iterative eigenproblem solver. Therefore, in the spirit of a suggestion due to AMADEI, LINSSEN, AND BERENDSEN [2], we start with a long term HMC series and apply some covariance analysis to it; this technique then helps to reduce the total number of state variables to a subset of $d << 3N$ *essential* variables so that only $n = m^d$ discretization boxes are needed. A realistic example will be presented in the subsequent Section 4.3.

Eigenvalue Cluster analysis. Assume that we know how to evaluate the entries of P. So we are finally left with the numerical solution of the eigenproblem

$$P\alpha = \lambda\alpha \quad \text{with} \quad \alpha = (\alpha_1, \ldots, \alpha_n) \ .$$

for a cluster of eigenvalues around the Perron eigenvalue $\lambda = 1$. One question is that we will not know in advance how many of the eigenvalues close to 1 should be included into the Perron cluster. The basic insight from [37] is that the k eigenvalues in the Perron cluster should be interpretable as a perturbed k-fold Perron root. The decision about k is quite subtle and has to be made by careful examination of the matrix $\mathcal{W}$ defined below in (4.22). Suppose now that this decision has been made. Then the conformational sets can be computed via appropriate linear combinations of the eigenmodes corresponding to the Perron cluster. Each conformation, say B, is represented as a set of indices, say I_B, that mark the associated boxes belonging to the spatial conformation. Given such a conformation as subset $B \subset \Omega$, the *probability* for the dynamical system *to stay within* B can easily be evaluated via the relation

$$w(B, B, \tau) = \frac{1}{\sum\limits_{i \in I_B} \pi_i} \sum_{i,j \in I_B} \pi_i \, p_{ij} \ . \tag{4.20}$$

This should be distinguished from the *probability* for the system *to be within* a conformation (compare also (4.12))

$$\pi(B) = \sum_{i \in I_B} \pi_i \ . \tag{4.21}$$

Finally, with k conformations $B_1, B_2, \ldots, B_k$ identified, the *transition rates between conformations* can be arranged in the (k,k)–matrix

$$\mathcal{W} = (w_{ij}) = \big(w(B_i, B_j, \tau)\big) \ , \quad i, j = 1, \ldots, k \tag{4.22}$$

which, together with the vector $\{\pi(B_1), \ldots, \pi(B_k)\}$, contains the core information of conformation dynamics. Details of this Perron cluster analysis omitted here can be found in our paper [37].

4.3 Virtual RNA Lab

In the frame of our collaboration with RNA technologists, all the above described mathematical techniques (and more) together with a variety of state of the art tools in scientific visualization are to be collected within some virtual RNA lab. Such a virtual lab is an integrated software package that permits a convenient switch between numerical code, fast visualization, and 3D graphic interaction. In order to give some flavor and at the same time to illustrate the above mathematical methods, a moderate size RNA molecule, the trinucleotide r(ACC), will be presented in some detail.

Hamilton function. The function H defined in (4.1) consists of the kinetic energy (p=momenta, M = mass tensor) and of the potential energy terms V like the covalent energy terms for bond stretching, angle bending, out-of-plane oscillations, dihedral torsions, non-bonded Lennard-Jones, and Coulomb terms. We have used the semi-emperical force field GROMOS96 [86]. The set of parameters had been adapted by the GROMOS96 group to quantum chemical calculations and experimental observations. The whole set was refined self-consistently to reproduce experimental results on chemical systems.

$$\begin{aligned} H(q,p) &= \tfrac{1}{2}p^T M^{-1} p + \\ & \sum_{k,l} V_{\text{bond}}(q_k,q_l) + \\ & \sum_{k,l,j} V_{\text{angle}}(q_k,q_l,q_j) + \\ & \sum_{k,l,j,m} V_{\text{out-of-plane}}(q_k,q_l,q_j,q_m) + \\ & \sum_{k,l,j,m} V_{\text{dihedral}}(q_k,q_l,q_j,q_m) + \\ & \sum_{k,l} V_{\text{Lennard-Jones}}(q_k,q_l) + \\ & \sum_{k,l} V_{\text{Coulomb}}(q_k,q_l) \end{aligned}$$

The *short term trajectories* needed in the computational model of Section 4.2 were realized by $m = 40$ steps of a Verlet discretization with local timestep $\tau/m = 2$ fsec, which means $\tau = 0.08$ psec.

Dimension reduction. The moderate size RNA molecule r(ACC) has $N = 70$ atoms, which means a phase space dimension $6N = 420$ for the Hamiltonian dynamical system and half of that for the position space. From chemical insight, a set of 37 *torsion angles* is responsible for the existence of different chemical conformations – see Fig. 4.4 left. By means of $M = 320.000$ sampling points within an ATHMC run, the covariance analysis due to [2] supplied a set of only $d = 4$ *essential* variables!

Perron cluster analysis. Upon subdividing two of the essential variables by 2, the other two by 3, we end up with only $n = 36$ boxes in position space. The approximation of the $\sim \frac{1}{2}n^2$ elements of the stochastic matrix P required $M = 128.000$ samplings of short term trajectories. The numerical eigenproblem solution for the $(36, 36)$-matrix P yielded the following candidates for the Perron cluster:

k	1	2	3	4	5	6	7	8	9	...
λ_k	1.000	0.999	0.989	0.974	0.963	0.946	0.933	0.904	0.805	...

Table 4.1: Perron eigenvalue cluster for r(ACC).

A first significant gap can be clearly observed after $\lambda_2 = 0.999$, which would have led to $k = 2$ conformations. A careful further analysis via the correlation matrix $\mathcal{W}$, however, led to the decision $k = 8$, which also shows a remarkable gap to the rest of the spectrum. The eight conformations {D1c, D1t, ... D4c, D4t} actually show significant structural differences, which supply a lot of insight to the chemical expert. In Table 4.2 we give the additional information about the probabilities for the dynamical system *to be within a conformation* (first row) and the probabilities *to stay within a conformation* (second row). In Fig. 4.4 two of the computed conformations are presented. Observe that D2c and D3c differ in a turn of the torsion angle χ and a flip–flop in the pseudo–angle P, which has appeared as an essential variable from our computation.

conformations	D1c	D1t	D2c	D2t	D3c	D3t	D4c	D4t
prob. to be within	0.107	0.011	0.116	0.028	0.320	0.038	0.285	0.095
prob. to stay within	0.986	0.938	0.961	0.888	0.991	0.949	0.981	0.962

Table 4.2: Probabilities $\pi(B_i)$ due to (4.21) and w_{ii} due to (4.20), $i = 1, \ldots, 8$.

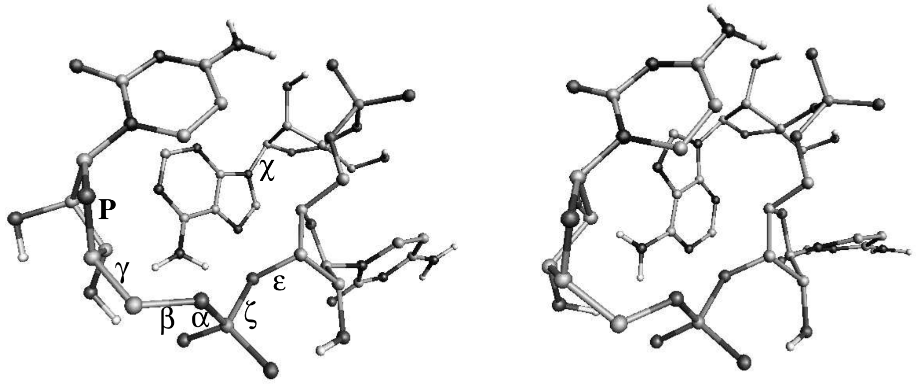

Figure 4.4: Conformations D3c (left) and D2c (right) of the r(ACC) molecule. Left: 6 out of 37 torsion angles. Look at the torsion angle χ and the pseudo-angle P to compare.

Conformational transition rates. The transition probabilities *between* the conformations are collected in the $(8,8)$-matrix $\mathcal{W}$ from (4.22) which here reads

$$\begin{pmatrix}
0.986 & 5.3_{10}{-3} & 1.4_{10}{-5} & 8.1_{10}{-3} & 5.3_{10}{-4} & 0 & 0 & 0 \\
5.1_{10}{-2} & 0.938 & 1.0_{10}{-2} & 4.6_{10}{-6} & 2.6_{10}{-4} & 6.3_{10}{-4} & 0 & 1.1_{10}{-7} \\
4.0_{10}{-5} & 3.1_{10}{-3} & 0.949 & 4.8_{10}{-2} & 0 & 3.6_{10}{-7} & 3.6_{10}{-5} & 2.6_{10}{-5} \\
2.7_{10}{-3} & 1.6_{10}{-7} & 5.7_{10}{-3} & 0.991 & 7.7_{10}{-8} & 0 & 8.0_{10}{-4} & 1.2_{10}{-9} \\
4.9_{10}{-4} & 2.5_{10}{-5} & 0 & 2.1_{10}{-7} & 0.961 & 1.3_{10}{-2} & 2.5_{10}{-2} & 7.3_{10}{-4} \\
0 & 2.6_{10}{-4} & 5.0_{10}{-7} & 0 & 5.6_{10}{-2} & 0.888 & 3.5_{10}{-3} & 5.2_{10}{-2} \\
0 & 0 & 4.7_{10}{-6} & 9.0_{10}{-4} & 1.0_{10}{-2} & 3.4_{10}{-4} & 0.981 & 7.4_{10}{-3} \\
0 & 1.4_{10}{-8} & 1.0_{10}{-5} & 4.0_{10}{-9} & 9.0_{10}{-4} & 1.5_{10}{-2} & 2.2_{10}{-2} & 0.962
\end{pmatrix}$$

Recall that the diagonal elements of $\mathcal{W}$ are the same as the second row in Table 4.2, whereas the first row in Table 4.2 does not show up in $\mathcal{W}$.

Extension to larger molecules. The above moderate size molecule is a quite good candidate to sharpen the mathematician's knife in this kind of collaboration. Biomolecules of real interest to our partners are larger – see e.g. the *hammerhead molecule* as represented in Fig. 4.5.

This type of biomolecule consists of about $150 - 200$ nucleotides; each nucleotide gives rise to roughly 12 torsion angles, which sums up to about 2.000 torsion angles that might generate conformations and should therefore be subdivided! Fortunately, chemists report that usually *the larger the molecules, the stiffer their structure.* If this is reliable, then covariance analysis for the associated HMC data will bring up "not too many" essential degrees of freedom that should still be tractable within our algorithmic framework. Future developments will include parallelization of the ATHMC part, a hierarchical framework to sample all relevant spatial configuration data, and a telescoping of the method with some kind of subspace multigrid eigenproblem solver similar to the one presented in [48].

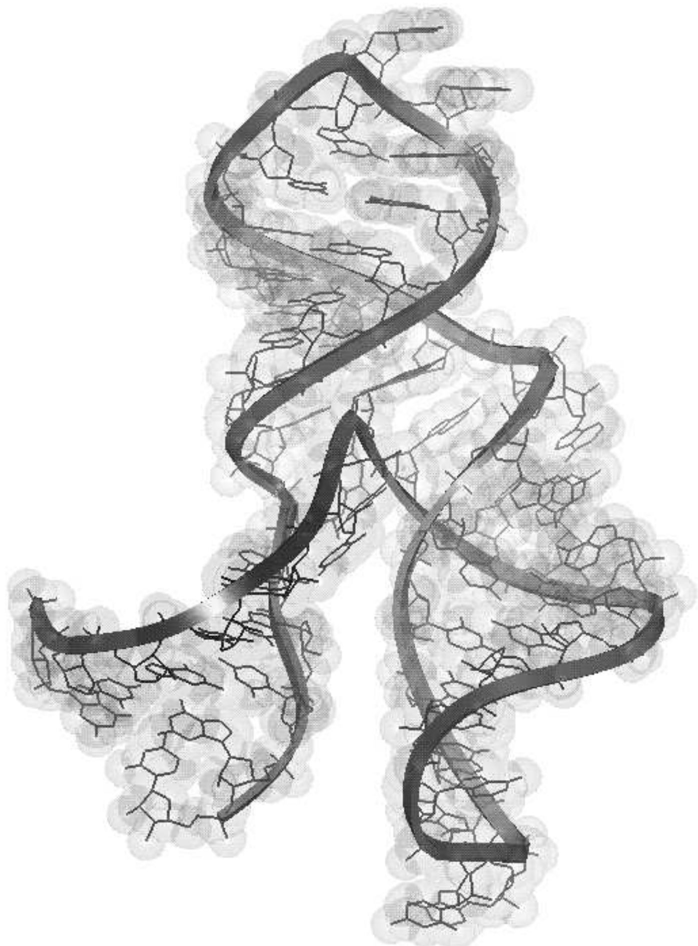

Figure 4.5: Hammerhead molecule, a flexible candidate for RNA drug design.

References

[1] M. P. Allen and D. J. Tildesley: *Computer Simulations of Liquids.* Clarendon, Oxford (1990).

[2] A. Amadei, A. Linssen, and H. Berendsen: *Essential dynamics of proteins.* Proteins, Vol. 17 (1993).

[3] V.I. Arnold: *Mathematical Methods of Classical Mechanics.* Second Edition. Springer (1989).

[4] I. Babuška, A. Miller: *A feedback finite element method with a–posteriori error estimation: part I.* Comput. Meth. Appl. Mech. Engrg. **61** pp. 1–40 (1987).

[5] I. Babuška, W.C. Rheinboldt: *Estimates for adaptive finite element computations.* SIAM J. Numer. Anal. **15** pp. 736–754 (1978).

[6] G. Bader, U. Nowak, P. Deuflhard: *An advanced simulation package for large chemical reaction systems.* In: R. Aiken (ed.), Stiff Computation, Oxford University Press, pp. 255–264 (1985).

[7] A. Bachem, M. Jünger, R. Schrader (eds.): *Mathematik in der Praxis.* Springer–Verlag (1995).

[8] R.E. Bank: *PLTMG: A Software Package for Solving Elliptic Partial Differential Equations. Users' Guide 8.0* SIAM (1998).

[9] R.E. Bank, A.H. Sherman, A. Weiser: *Refinement algorithms and data structures for regular local mesh refinement.* In: R. Stapleman et al (eds.), Scientific Computing, North–Holland, pp. 3–17 (1983).

[10] R.E. Bank, A. Weiser: *Some a posteriori error estimates for elliptic partial differental equations.* Math. Comp. **44**, pp. 283–301 (1985).

[11] P. Bastian, K. Birken, K. Johannsen, S. Lang, N. Neuss, H. Rentz-Reichert, C. Wieners: *UG – A flexible software toolbox for solving partial differential equations.* Computing and Visualization in Science, Vol. 1, pp. 27–40 (1997).

[12] P. Bastian, G. Wittum: *Adaptive multigrid methods: The UG concept.* In: W. Hackbusch and G. Wittum (eds.), Adaptive Methods – Algorithms, Theory and Applications, Series Notes on Numerical Fluid Mechanics, Vol. 46, pp. 17–37, Vieweg, Braunschweig (1994).

[13] R. Beck, P. Deuflhard, R. Hiptmair, B. Wohlmuth, R.H.W. Hoppe: *Adaptive Multilevel Methods for Edge Element Discretizations of Maxwell's Equations.* Surveys of Mathematics for Industry, accepted for publication (1999).

[14] R. Beck, B. Erdmann, R. Roitzsch: *KASKADE 3.0 – User's Guide.* `ftp://ftp.zib.de/pub/kaskade` (1996).

[15] J. Bey: *Tetrahedral Grid Refinement.* Computing, Vol. 55, No. 4, pp. 355–378 (1995).

[16] F.A. Bornemann, B. Erdmann, R. Kornhuber: *Adaptive multilevel methods in three space dimensions.* Int. J. Num. Meth. in Eng. **36**, pp. 3187–3203 (1993).

[17] F.A. Bornemann, B. Erdmann, R. Kornhuber: *A posteriori error estimates for elliptic problems in two and three space dimensions.* SIAM J. Numer. Anal. **33**, pp. 1188–1204 (1996).

[18] F. Bornemann: *An adaptive multilevel approach to parabolic equations I. General theory and implementation.* IMPACT Comput. Sci. Engrg. **2**, pp. 279–317 (1990).

[19] F. Bornemann: *An adaptive multilevel approach to parabolic equations II. Variable–order time discretization based on a multilevel error correction.* IMPACT Comput. Sci. Engrg. **3**, pp. 93–122 (1991).

[20] F. Bornemann, P. Deuflhard: *The Cascadic Multigrid Method for Elliptic Problems.* Numer. Math. 75, Springer International, pp. 135–152 (1996).

[21] A. Bossavit: *Solving Maxwell's equations in a closed cavity and the question of spurious modes.* IEEE Trans. Mag., 26(2), pp. 702–705 (1990).

[22] R. Bradel, A. Kleinke, K.-H. Reichert: *Molar Mass Distribution of Microbial Poly (D-3-Hydroxybutyrate) in the Course of Intracellular Synthesis and Degradation.* Makromol. Chem., Rapid Commun. **12**, p. 583 (1991).

[23] D. Braess, W. Hackbusch: *A new convergence proof for the multigrid method including the V-cycle.* SIAM. J. Numer. Anal., 20, pp. 967–975 (1983).

[24] J. Bramble, J. Pasciak, J. Xu: *Parallel multilevel preconditioners.* Math. Comp. **55**, pp. 1–22 (1990).

[25] J. Bramble, J. Pasciak, J. Wang, J. Xu: *Convergence estimates for multigrid algorithms without regularity assumptions.* Math. Comp. **57**, pp. 23–45 (1991).

[26] A. Brandt: *Multi–level adaptive solutions to boundary–value problems.* Math. Comp. **31**, pp. 333–390 (1977).

[27] M.–O. Bristeau, G. Etgen, W. Fitzigibbon, J.–L. Lion, J. Periaux, M. Wheeler (eds.): *Computational Science for the 21st Century.* Tours, France. Wiley–Interscience–Europe (1997).

[28] M. Dellnitz, A. Hohmann: *A subdivision algorithm for the computation of unstable manifolds and global attractors.* Numer. Math. 75, pp. 293–317 (1997).

[29] M. Dellnitz, O. Junge: *On the approximation of complicated dynamical behavior.* SIAM J. Num. Anal. 36(2), pp. 491–515 (1999).

[30] P. Deuflhard: *Cascadic Conjugate Gradient Methods for Elliptic Partial Differential Equations. Algorithm and Numerical Results.* In [61], pp. 29–42 (1994).

[31] P. Deuflhard: *Uniqueness Theorems for Stiff ODE Initial Value Problems.* In: D.F. Griffiths and G.A. Watson (eds.): *Numerical Analysis 1989*, Longman Scientific & Technical, Harlow, Essex, UK, pp. 74–205 (1990).

[32] P. Deuflhard and F. Bornemann: *Numerische Mathematik II — Integration gewöhnlicher Differentialgleichungen.* Walter de Gruyter, Berlin, New York (1994).

[33] P. Deuflhard, M. Dellnitz, O. Junge, Ch. Schütte: *Computation of Essential Molecular Dynamics by Subdivision Techniques.* In [36], pp. 98–115 (1998).

[34] P. Deuflhard, T. Friese, F. Schmidt, R. März, H.–P. Nolting: *Effiziente Eigenmodenberechnung für den Entwurf integriert–optischer Chips.* In [56], pp. 267–279 (1996).

[35] P. Deuflhard, J. Hohmann: *Numerical analysis. A First Course in Scientific Computation.* Berlin, New York: de Gruyter (1995).

[36] P. Deuflhard, A. Hermans, B. Leimkuhler, A.E. Mark, S. Reich, and R.D. Skeel (eds.): *Computational Molecular Dynamics: Challenges, Methods, Ideas.* Lecture Notes in Computational Science and Engineering, Vol. **4**, Springer-Verlag (1998).

[37] P. Deuflhard, W. Huisinga, A. Fischer, and Ch. Schütte: *Identification of almost invariant aggregates in reversible nearly uncoupled Markov chains.* Lin. Alg. Appl., accepted.

[38] P. Deuflhard, J. Lang, U. Nowak: *Recent Progress in Dynamical Process Simulation.* In: H. Neunzert (ed.): *Topics in Industrial Mathematics*, Wiley & Teubner Publishers, pp. 122–137 (1996).

[39] P. Deuflhard, P. Leinen, H. Yserentant: *Concepts of an Adaptive Hierarchical Finite Element Code.* IMPACT Comp. Sci. Eng. 1, pp. 3–35 (1989).

[40] P. Deuflhard, M. Seebass: *Adaptive Multilevel FEM as Decisive Tools in the Clinical Cancer Therapy Hyperthermia.* In: Choi–Hong Lai, Peter Bjørstad, Mark Cross, O. Widlund (eds.), Procs. 11th International Conference on Domain Decomposition Methods (DD11), UK, 1998 (to appear 1999).

[41] P. Deuflhard, M. Seebass, D. Stalling, R. Beck, H.C. Hege: *Hyperthermia Treatment Planning in Clinical Cancer Therapy: Modelling, Simulation, and Visualization.* Plenary Keynote talk, 15th IMACS World Congress 1997. In: Achim Sydow (ed.), Vol. 3, *Computational Physics, Chemistry and Biology.* Wissenschaft and Technik Verlag, pp. 9–17 (1997).

[42] P. Deuflhard, M. Weiser: *Local Inexact Newton Multilevel FEM for Nonlinear Elliptic Problems.* In [27], pp. 129–138 (1997).

[43] P. Deuflhard, M. Weiser, M. Seebass: *A New Nonlinear Elliptic Multilevel FEM Appied to Regional Hyperthermia.* Konrad Zuse Zentrum, Preprint SC 98-35 (1998).

[44] P. Deuflhard, M. Weiser: *Global Inexact Newton Multilevel FEM for Nonlinear Elliptic Problems.* In [53], pp. 71–89 (1998).

[45] P. Deuflhard, M. Wulkow: *Computational Treatment of Polyreaction Kinetics by Orthogonal Polynomials of a Discrete Variable.* IMPACT Comp. Sci. Eng.1, pp. 269–301 (1989).

[46] P. Deuflhard, M. Wulkow: *Simulationsverfahren für die Polymerchemie.* In [7], pp. 117–136 (1995).

[47] A. Fischer, F. Cordes, and Ch. Schütte: *Hybrid Monte Carlo with adaptive temperature in mixed canonical ensemble: Efficient conformational analysis of RNA.* J. Comput. Chem., 19(15), pp. 1689-1697 (1998).

[48] T. Friese, P. Deuflhard, F. Schmidt: *A Multigrid Method for the Complex Helmholtz Eigenvalue Problem.* In: Procs. 11th International Conference on Domain Decomposition Methods (DD11), UK, 1998 (to appear 1999).

[49] H. Gajewski, K. Zacharias: *On an Initial Value Problem for a Coagulation Equation with Growth Term.* Math. Nachr. 109, pp. 135–156 (1982).

[50] G.H. Golub, C.F. Van Loan: *Matrix Computations.* The Johns Hopkins University Press, Baltimore, London, 2nd Edition (1989).

[51] H. Grubmüller, P. Tavan. *Molecular dynamics of conformational substates for a simplified protein model.* J. Chem. Phys. **101** (1994).

[52] W. Hackbusch: *Multi–Grid Methods and Applications.* Springer Verlag, Berlin, Heidelberg, New York (1995).

[53] W. Hackbusch, G. Wittum (eds.): *Multigrid Methods, Lecture Notes in Computational Science and Engineering.* Vol. 3, Springer–Verlag (1998).

[54] H.-C. Hege, M. Seebaß, D. Stalling, M. Zöckler: *A Generalized Marching Cubes Algorithm Based On Non-Binary Classifications.* ZIB Preprint SC 97–05 (1997).

[55] R. Hiptmair: *Multilevel Preconditioning for Mixed Problems in Three Dimensions.* Ph.D. Thesis, Wissner, Augsburg (1996).

[56] K.–H. Hoffmann, W. Jäger, T. Lohmann, H. Schunk (eds.): *MATHEMATIK Schlüsseltechnologie für die Zukunft.* Springer (1997).

[57] C.S. Hsu: *Cell–to–Cell Mapping. A Method of Global Analysis for Nonlinear Systems.* In: F. John, J.E. Marsden, L. Sirovich (eds.): *Applied Mathematical Sciences*, Vol. 64, Springer–Verlag.

[58] W. Huisinga, Ch. Best, R. Roitzsch, Ch. Schütte, and F. Cordes: *From Simulation Data to Conformational Ensembles: Structure and Dynamics Based Methods.* Konrad-Zuse-Zentrum Berlin. Preprint SC 98–36 (1998).

[59] P. Iedema, M. Wulkow, H. Hoefsloot: *Modeling Molecular Weight and Degree of Branching Distribution of Low Density Polyethylene.* Submitted to Macromolecules (1999).

[60] V.E. Katsnelson: *Conditions under which systems of eigenvectors of some classes of operators form a basis.* Funkt Anal. Appl.**1**, pp. 122–132 (1967).

[61] D.E. Keyes, J. Xu (eds.): *Domain Decomposition Methods in Scientific and Engineering Computing.* AMS Series Contemporary Mathematics, Vol. 180, Providence (1994).

[62] R. Kornhuber, R. Roitzsch: *On adaptive grid refinement in the presence of internal or boundary layers.* IMPACT Comput. Sci. Engrg. **2**, pp. 40–72 (1990).

[63] J. Lang: *Adaptive FEM for Reaction–Diffusion Equations.* Appl. Numer. Math. 26, pp. 105–116 (1998).

[64] J. Lang: *Adaptive Multilevel Solution of Nonlinear Parabolic PDE Systems. Theory, Algorithm, and Applications.* Habilitation Thesis, Free University of Berlin, 1999. Konrad–Zuse-Zentrum Berlin, Preprint SC 99–20 (1999).

[65] J. Lang, B. Erdmann, M. Seebass: *Impact of Nonlinear Heat Transfer on Temperature Distribution in Regional Hyperthermia.* Accepted for publication in IEEE Transaction on Biomedical Engineering (1999).

[66] W. E. Lorensen, H. E. Cline: *Marching Cubes: A high resolution 3D surface construction algorithm.* Computer Graphics 21:4, pp. 163-169 (1987).

[67] J. Mandel, S.F. McCormick: *A multilevel variational method for $Au = \Lambda Bu$ on composite grids.* J. Comp. Phys. **80**, pp. 442-452 (1989).

[68] B. Mayfield: *Nonlocal Boundary Conditions for the Schrödinger Equation.* PhD thesis, University of Rhode Island, Providence, RI, (1989).

[69] A.H.E. Müller, D. Yan, M. Wulkow: *Molecular Parameters of Hyperbranched Polymers Made by Self–condensing Vinyl Polymerization. I. Molecular weight distribution.* Macromolecules **30**, pp. 7015 (19997).

[70] J.C. Nédeléc: *Mixed finite elements in* $\mathrm{I\!R}^3$. Numer. Math. 35, pp. 315–341 (1980).

[71] M.E. Go Ong: *Hierarchical Basis Preconditioners for Second Order Elliptic Problems in Three Ddimensions.* PhD Thesis, University of California, Los Angeles (1989).

[72] H.H. Pennes: *Analysis of tissue and arterial blood temperatures in the resting human forearm.* J. Appl. Phys. 1, pp. 93-122 (1948).

[73] M.C. Rivara: *Algorithms for refining triangular grids suitable for adaptive and multigrid techniques.* Int. J. Numer. Meth. Engrg. 20, pp. 745-756 (1984).

[74] Y. Saad: *Numerical Methods for Large Eigenvalue Problems.* Manchester University Press (1992).

[75] J.M. Sanz-Serna, M.P. Calvo: *Numerical Hamiltonian Problems.* Chapman & Hall (1994).

[76] F. Schmidt: *An adaptive approach to the numerical solution of Fresnel's wave equation.* IEE Journal of Lightwave Technology, 11(9), pp. 1425–1435 (1993).

[77] F. Schmidt: *Computation of discrete transparent boundary conditions for the 2D Helmholtz equation.* Konrad–Zuse–Zentrum Berlin. Preprint SC 97-64 (1997).

[78] F. Schmidt, P. Deuflhard: *Discrete Transparent Boundary Conditions for the Numerical Solution of Fresnel's equation.* Computers Math. Applic. **29**, No. 9, pp. 53–76 (1995).

[79] F. Schmidt, D. Yevick: *Discrete transparent boundary conditions for Schrödinger–type equations.* J. Comput. Phys., Vol. 134, pp. 96–107 (1997).

[80] Ch. Schütte: *Conformational Dynamics: Modelling, Theory, Algorithm, and Application to Biomolecules.* Habilitation Thesis. Konrad–Zuse–Zentrum Berlin, Preprint SC 99–18 (1999).

[81] Ch. Schütte, A. Fischer, W. Huisinga, and P. Deuflhard: *A Direct Approach to Conformational Dynamics based on Hybrid Monte Carlo.* J. Comput. Phys. **151**, Special Issue on Computational Molecular Biophysics, pp. 146–168 (1999).

[82] V.V. Shaidurov: *Some estimates of the rate of convergence for the cascadic conjugate–gradient method.* Preprint, Otto–von–Guericke–Universität. Magdeburg (1994).

[83] M. Seebass, D. Stalling, M. Zöckler, H.-C. Hege, P. Wust, R. Felix, P. Deuflhard: *Surface Mesh Generation for Numerical Simulations of Hyperthermia Treatments.* In: Procs. 16th Annual Meeting of the European Society for Hyperthermic Oncology, p. 146, Berlin (1997).

[84] M. Seebass, D. Stalling, J. Nadobny, P. Wust, R. Felix, P. Deuflhard: *Three-Dimensional Finite Element Mesh Generation for Numerical Simulations of Hyperthermia Treatments*, Proc. 7th Int. Congress on Hyperthermic Oncology, Rome, Vol. 2, pp. 547-548 (1996).

[85] C.W. Song, A. Lokshina, J.G. Rhee, M. Patten, and S.H. Levitt: *Implication of blood flow in hyperthermic treatment of tumors.* IEEE Trans. Biomed. Engrg. 31, pp. 9-16 (1984).

[86] W.F. van Gunsteren, S.R. Billeter, A.A. Eising, P.H. Hünenberger, P. Krüger, A.E. Mark, W.R.P. Scott, and I.G. Tironi: *Biomolecular Simulation: The GROMOS96 Manual and User Guide*. vdf Hochschulverlag AG, ETH Zürich (1996).

[87] M. Wulkow: *Numerical Treatment of Countable Systems of Ordinary Differential Equations.* Dissertation, Konrad–Zuse–Zentrum Berlin. Technical Report TR 90–08 (1990).

[88] M. Wulkow: *Feature article – The Simulation of Molecular Weight Distributions in Polyreaction Kinetics by Discrete Galerkin Methods.* Macromol. Theory Simul. **5**, pp. 393–416 (1996).

[89] J. Xu: *Theory of Multilevel Methods.* PhD Thesis, Report No. AM 48, Department of Mathematics, Pennsylvania State University (1989).

[90] J. Xu: *Iterative methods by space decomposition and subspace correction.* SIAM Review **34** pp. 581–613 (1992).

[91] H. Yserentant: *On the multilevel splitting of finite element spaces.* Numer. Math. **58** pp. 163–184 (1986).