Konrad-Zuse-Zentrum
für Informationstechnik Berlin

Takustraße 7
D-14195 Berlin-Dahlem
Germany

J. LANG AND B. ERDMANN

# Adaptive Linearly Implicit Methods for Heat and Mass Transfer Problems

ZIB-Report 00–21 (July 2000)

# Adaptive Linearly Implicit Methods for Heat and Mass Transfer Problems

J. Lang and B. Erdmann

## 1 Introduction

Dynamical process simulation is the central tool nowadays to assess the modelling process for large scale physical problems arising in such fields as biology, chemistry, metallurgy, medicine, and environmental science. Moreover, successful numerical methods are very attractive to design and control economical plants at low costs in a short time. Due to the great complexity of the established models, the development of fast and reliable algorithms has been a topic of continuing investigation during the last years.

One of the important requirements that modern software must meet today is to judge the quality of its numerical approximations in order to assess safely the modelling process. Adaptive methods have proven to work efficiently providing a posteriori error estimates and appropriate strategies to improve the accuracy where needed. They are now entering into real–life applications and starting to become a standard feature in simulation programs. The present paper reports on one successful way to construct discretization methods adaptive in space and time, which are applicable to a wide range of practically relevant problems.

We concentrate on heat and mass transfer problems which can be written in the form

$$B(x,t,u,\nabla u)\partial_t u = \nabla \cdot (D(x,t,u,\nabla u)\nabla u) + F(x,t,u,\nabla u) \ , \qquad (1)$$

supplemented with suitable boundary and initial conditions. The vector–valued solution $u = (u_1, \ldots, u_m)^T$ is supposed to be unique. This problem class includes the well–known reaction–diffusion equations and the Navier–Stokes equations as well.

In the classical method of lines (MOL) approach, the spatial discretization is done once and kept fixed during the time integration. Discrete solution values correspond to points on lines parallel to the time axis. Since adaptivity in space means to add or delete points, in an adaptive MOL approach new lines can arise and later on disappear. Here, we allow a local spatial refinement in each time step, which results in a discretization sequence first in time then in space. The spatial discretization is considered as a perturbation, which has to be controlled within each time step. Combined with a posteriori error estimates this approach is known as adaptive Rothe method. First theoretical investigations have been made by BORNEMANN [7] for linear parabolic equations. LANG and WALTER

[26] have generalized the adaptive Rothe approach to reaction–diffusion systems. A rigorous analysis for nonlinear parabolic systems is given in LANG [28]. For a comparative study, we refer to DEUFLHARD, LANG, and NOWAK [16].

Since differential operators give rise to infinite stiffness, often an implicit method is applied to discretize in time. We use linearly implicit methods of Rosenbrock type, which are constructed by incorporating the Jacobian directly into the formula. These methods offer several advantages. They completely avoid the solution of nonlinear equations, that means no Newton iteration has to be controlled. There is no problem to construct Rosenbrock methods with optimum linear stability properties for stiff equations. According to their one–step nature, they allow a rapid change of step sizes and an efficient adaptation of the spatial discretization in each time step. Moreover, a simple embedding technique can be used to estimate the error in time satisfactorily. A description of the main idea of linearly implicit methods is given in Section 2.

Stabilized finite elements are used for the spatial discretization to prevent numerical instabilities caused by advection–dominated terms. To estimate the error in space, the hierarchical basis technique has been extended to Rosenbrock schemes in LANG [28]. Hierarchical error estimators have been accepted to provide efficient and reliable assessment of spatial errors. They can be used to steer a multilevel process, which aims at getting a successively improved spatial discretization drastically reducing the size of the arising linear algebraic systems with respect to a prescribed tolerance (BORNEMANN, ERDMANN, and KORNHUBER [8], DEUFLHARD, LEINEN and YSERENTANT [17], BANK and SMITH [2]). A brief introduction to multilevel finite element methods is given in Section 3.

The described algorithm has been coded in the fully adaptive software package KARDOS at the Konrad–Zuse–Zentrum in Berlin. Several types of embedded Rosenbrock solvers and adaptive finite elements were implemented. KARDOS is based on the KASKADE–toolbox [18], which is freely distributed at ftp://ftp.zib.de/pub/kaskade. Nowadays both codes are efficient and reliable workhorses to solve a wide class of PDEs in one, two, or three space dimensions. To demonstrate the performance of our adaptive approach, in Section 4 we will present two practically relevant problems occuring in combustion theory and brine transport in porous media.

## 2 Linearly Implicit Methods

In this section a short description of the linearly implicit discretization idea is given. More details can be found in the books of HAIRER and WANNER [23], DEUFLHARD and BORNEMANN [15], STREHMEL and WEINER [37]. For ease of presentation, we firstly set $B=I$ in (1) and consider the autonomous case. Then we can look at (1) as an abstract Cauchy problem of the form

$$\partial_t u = f(u)\,, \quad u(t_0) = u_0\,, \quad t > t_0\,, \tag{2}$$

where the differential operators and the boundary conditions are incorporated into the nonlinear function $f(u)$. Since differential operators give rise to infinite

stiffness, often an implicit discretization method is applied to integrate in time. The simplest scheme is the implicit (backward) Euler method

$$u_{n+1} = u_n + \tau\, f(u_{n+1})\,, \tag{3}$$

where $\tau = t_{n+1} - t_n$ is the step size and $u_n$ denotes an approximation of $u(t)$ at $t = t_n$. This equation is implicit in $u_{n+1}$ and thus usually a Newton–like iteration method has to be used to approximate the numerical solution itself. The implementation of an efficient nonlinear solver is the main problem for a fully implicit method.

Investigating the convergence of Newton's method in function space, DEUFLHARD [13] pointed out that one calculation of the Jacobian or an approximation of it per time step is sufficient to integrate stiff problems efficiently. Using $u_n$ as an initial iterate in a Newton method applied to (3), we find

$$(I - \tau\, J_n)\, K_n \;=\; \tau f(u_n)\,, \tag{4}$$

$$u_{n+1} \;=\; u_n + K_n\,, \tag{5}$$

where $J_n$ stands for the Jacobian matrix $\partial_u f(u_n)$. The arising scheme is known as the *linearly implicit* Euler method. The numerical solution is now effectively computed by solving the system of linear equations that defines the increment $K_n$. Among the methods which are capable of integrating stiff equations efficiently, linearly implicit methods are the easiest to program, since they completely avoid the numerical solution of nonlinear systems.

One important class of higher–order linearly implicit methods consists of extrapolation methods that are very effective in reducing the error, see DEUFLHARD [14]. However, in the case of higher spatial dimension, several drawbacks of extrapolation methods have shown up in numerical experiments made by BORNEMANN [6]. Another generalization of the linearly implicit approach we will follow here leads to Rosenbrock methods (ROSENBROCK [35]). They have found wide–spread use in the ODE context. Applied to (2) a so–called s–stage Rosenbrock method has the recursive form

$$(I - \tau\gamma_{ii}\, J_n)\, K_{ni} \;=\; \tau f(u_n + \sum_{j=1}^{i-1} \alpha_{ij}\, K_{nj}) + \tau J_n \sum_{j=1}^{i-1} \gamma_{ij}\, K_{nj}\,,\; i = 1(1)s\,, \tag{6}$$

$$u_{n+1} \;=\; u_n + \sum_{i=1}^{s} b_i K_{ni}\,, \tag{7}$$

where the step number $s$ and the defining formula coefficients $b_i$, $\alpha_{ij}$, and $\gamma_{ij}$ are chosen to obtain a desired order of consistency and good stability properties for stiff equations (see e.g. HAIRER and WANNER [23], IV.7). We assume $\gamma_{ii} = \gamma > 0$ for all $i$, which is the standard simplification to derive Rosenbrock methods with one and the same operator on the left–hand side of (6). The linearly implicit Euler method mentioned above is recovered for $s = 1$ and $\gamma = 1$.

For the general system

$$B(t,u)\partial_t u = f(t,u)\,, \quad u(t_0) = u_0\,, \quad t > t_0\,, \tag{8}$$

an efficient implementation that avoids matrix–vector multiplications with the Jacobian was given by LUBICH and ROCHE [31]. In the case of a time– or solution–dependent matrix $B$, an approximation of $\partial_t u$ has to be taken into account, leading to the generalized Rosenbrock method of the form

$$\begin{aligned}\left(\frac{1}{\tau\gamma}B(t_n,u_n) - J_n\right) U_{ni} \;&=\; f(t_i,U_i) - B(t_n,u_n)\sum_{j=1}^{i-1}\frac{c_{ij}}{\tau}U_{nj} + \tau\gamma_i C_n \\ &+\; (B(t_n,u_n) - B(t_i,U_i))\,Z_i\,, \quad i = 1(1)s\,,\end{aligned} \tag{9}$$

where the internal values are given by

$$t_i = t_n + \alpha_i\tau\,, \quad U_i = u_n + \sum_{j=1}^{i-1} a_{ij}\,U_{nj}\,, \quad Z_i = (1-\sigma_i)z_n + \sum_{j=1}^{i-1}\frac{s_{ij}}{\tau}U_{nj},$$

and the Jacobians are defined by

$$\begin{aligned} J_n \;&:=\; \partial_u(f(t,u) - B(t,u)z)_{|u=u_n,t=t_n,z=z_n}\,, \\ C_n \;&:=\; \partial_t(f(t,u) - B(t,u)z)_{|u=u_n,t=t_n,z=z_n}\,. \end{aligned}$$

This yields the new solution

$$u_{n+1} = u_n + \sum_{i=1}^{s} m_i\,U_{ni}$$

and an approximation of the temporal derivative $\partial_t u$

$$z_{n+1} = z_n + \sum_{i=1}^{s} m_i\,\Big(\frac{1}{\tau}\sum_{j=1}^{i}(c_{ij}-s_{ij})U_{nj} + (\sigma_i - 1)z_n\Big)\,.$$

The new coefficients can be derived from $\alpha_{ij}$, $\gamma_{ij}$, and $b_i$ [31]. In the special case $B(t,u) = I$, we get (6) setting $U_{ni} = \tau\sum_{j=1,\ldots,i}\gamma_{ij}K_{nj}$, $i = 1,\ldots,s$.

Various Rosenbrock solvers have been constructed to integrate systems of the form (8). An important fact is that the formulation (8) includes problems of higher differential index. Thus, the coefficients of the Rosenbrock methods have to be specially designed to obtain a certain order of convergence. Otherwise, order reduction might happen. In [32, 31], the solver ROWDAIND2 was presented, which is suitable for semi–explicit index 2 problems. Among the Rosenbrock methods suitable for index 1 problems we mention ROS2 [12], ROWDA3 [33], ROS3P [29], and RODASP [36]. More informations can be found in [28]. For the convenience of the reader, we give the defining formula coefficients for ROS2

| | |
|---|---|
| $\gamma = 1.707106781186547e+00$ | |
| $a_{11} = 0.000000000000000e+00$<br>$a_{21} = 5.857864376269050e-01$<br>$a_{22} = 0.000000000000000e+00$ | $\alpha_1 = 0.000000000000000e+00$<br>$\alpha_2 = 1.000000000000000e+00$ |
| $c_{11} = 5.857864376269050e-01$<br>$c_{21} = 1.171572875253810e+00$<br>$c_{22} = 5.857864376269050e-01$ | $s_{11} = 0.000000000000000e+00$<br>$s_{21} = 3.431457505076198e-01$<br>$s_{22} = 0.000000000000000e+00$ |
| $\gamma_1 = 1.707106781186547e+00$<br>$\gamma_2 = -1.707106781186547e+00$ | $\sigma_1 = 0.000000000000000e+00$<br>$\sigma_2 = 5.857864376269050e-01$ |
| $m_1 = 8.786796564403575e-01$<br>$m_2 = 2.928932188134525e-01$ | $\hat{m}_1 = 5.857864376269050e-01$<br>$\hat{m}_2 = 0.000000000000000e+00$ |

Table 1: Set of coefficients for ROS2 [12].

and ROWDAIND2 in Tab. 1 and Tab. 2, respectively. Both Rosenbrock solvers have been used in our simulations presented here.

Usually, one wishes to adapt the step size in order to control the temporal error. For linearly implicit methods of Rosenbrock type a second solution of inferior order, say $\hat{p}$, can be computed by a so–called embedded formula

$$\begin{aligned} \hat{u}_{n+1} &= u_n + \sum_{i=1}^{s} \hat{m}_i U_{ni} \,, \\ \hat{z}_{n+1} &= z_n + \sum_{i=1}^{s} \hat{m}_i \left(\frac{1}{\tau} \sum_{j=1}^{i} (c_{ij} - s_{ij}) U_{nj} + (\sigma_i - 1) z_n\right) \,, \end{aligned}$$

where the original weights $m_i$ are simply replaced by $\hat{m}_i$. If $p$ is the order of $u_{n+1}$, we call such a pair of formulas to be of order $p(\hat{p})$. Introducing an appropriate scaled norm $\|\cdot\|$, the local error estimator

$$r_{n+1} = \|u_{n+1} - \hat{u}_{n+1}\| + \|\tau(z_{n+1} - \hat{z}_{n+1})\| \tag{10}$$

can be used to propose a new time step by

$$\tau_{n+1} = \frac{\tau_n}{\tau_{n-1}} \left(\frac{TOL_t \, r_n}{r_{n+1} \, r_{n+1}}\right)^{1/(\hat{p}+1)} \tau_n \,. \tag{11}$$

Here, $TOL_t$ is a desired tolerance prescribed by the user. This formula is related to a discrete PI–controller first established in the pioneering works of GUSTAFFSON, LUNDH, and SÖDERLIND [21, 20]. A more standard step size selection strategy can be found in HAIRER, NØRSETT, and WANNER ([22], II.4).

| | |
|---|---|
| $\gamma = 3.000000000000000e-01$ | |
| $a_{11} = 0.000000000000000e+00$ | $\alpha_1 = 0.000000000000000e+00$ |
| $a_{21} = 1.666666666666667e+00$ | $\alpha_2 = 5.000000000000000e-01$ |
| $a_{22} = 0.000000000000000e+00$ | $\alpha_3 = 1.000000000000000e+00$ |
| $a_{31} = 1.830769230769234e+00$ | $\alpha_4 = 1.000000000000000e+00$ |
| $a_{32} = 2.400000000000000e+00$ | |
| $a_{33} = 0.000000000000000e+00$ | |
| $a_{41} = 1.830769230769234e+00$ | |
| $a_{42} = 2.400000000000000e+00$ | |
| $a_{43} = 0.000000000000000e+00$ | |
| $a_{44} = 0.000000000000000e+00$ | |
| $c_{11} = 3.333333333333333e+00$ | $s_{11} = 0.000000000000000e+00$ |
| $c_{21} = 1.246438746438751e+00$ | $s_{21} = 5.555555555555556e+00$ |
| $c_{22} = 3.333333333333333e+00$ | $s_{22} = 0.000000000000000e+00$ |
| $c_{31} = -1.226780626780621e+01$ | $s_{31} = -4.239316239316217e+00$ |
| $c_{32} = 4.266666666666667e+01$ | $s_{32} = 8.000000000000000e+00$ |
| $c_{33} = 3.333333333333333e+00$ | $s_{33} = 0.000000000000000e+00$ |
| $c_{41} = 5.824628046850726e-02$ | $s_{41} = -4.239316239316217e+00$ |
| $c_{42} = 3.259259259259259e+00$ | $s_{42} = 8.000000000000000e+00$ |
| $c_{43} = -3.703703703703704e-01$ | $s_{43} = 0.000000000000000e+00$ |
| $c_{44} = 3.333333333333333e+00$ | $s_{44} = 0.000000000000000e+00$ |
| $\gamma_1 = 3.000000000000000e-01$ | $\sigma_1 = 0.000000000000000e+00$ |
| $\gamma_2 = 1.878205128205124e-01$ | $\sigma_2 = 1.666666666666667e+00$ |
| $\gamma_3 = -1.000000000000000e+00$ | $\sigma_3 = 2.307692307692341e-01$ |
| $\gamma_4 = 0.000000000000000e+00$ | $\sigma_4 = 2.307692307692341e-01$ |
| $m_1 = 1.830769230769234e+00$ | $\hat{m}_1 = 2.214433650496747e+00$ |
| $m_2 = 2.400000000000000e+00$ | $\hat{m}_2 = 1.831186394371970e+00$ |
| $m_3 = 0.000000000000000e+00$ | $\hat{m}_3 = 8.264462809917363e-03$ |
| $m_4 = 1.000000000000000e+00$ | $\hat{m}_4 = 0.000000000000000e+00$ |

Table 2: Set of coefficients for ROWDAIND2 [32, 31].

Rosenbrock methods offer several structural advantages. They preserve conservation properties like fully implicit methods. There is no problem to construct Rosenbrock methods with optimum linear stability properties for stiff equations. Because of their one–step nature, they allow a rapid change of step sizes and an efficient adaptation of the underlying spatial discretizations as will be seen in the next section. Thus, they are attractive for solving real world problems.

# 3 Multilevel Finite Elements

In the context of PDEs, system (9) consists of linear elliptic boundary value problems possibly advection–dominated. In the spirit of spatial adaptivity a multilevel finite element method is used to solve this system. The main idea of the multilevel technique consists of replacing the solution space by a sequence of discrete spaces with successively increasing dimension to improve their approximation property. A posteriori error estimates provide the appropriate framework to determine where a mesh refinement is necessary and where degrees of freedom are no longer needed. Adaptive multilevel methods have proven to be a useful tool for drastically reducing the size of the arising linear algebraic systems and to achieve high and controlled accuracy of the spatial discretization (see e.g. BANK [1], DEUFLHARD, LEINEN, and YSERENTANT [17], LANG [27]).

Let $T_h$ be an admissible finite element mesh at $t = t_n$ and $S_h^q$ be the associated finite dimensional space consisting of all continuous functions which are polynomials of order $q$ on each finite element $T \in T_h$. Then the standard Galerkin finite element approximation $U_{ni}^h \in S_h^q$ of the intermediate values $U_{ni}$ satisfies the equation

$$(L_n\, U_{ni}^h, \phi) = (r_{ni}, \phi) \quad \text{for all } \phi \in S_h^q\,, \tag{12}$$

where $L_n$ is the weak representation of the differential operator on the left–hand side in (9) and $r_{ni}$ stands for the entire right–hand side in (9). Since the operator $L_n$ is independent of $i$ its calculation is required only once within each time step.

It is a well–known inconvenience that the solutions $U_{ni}^h$ may suffer from numerical oscillations caused by dominating convective and reactive terms as well. An attractive way to overcome this drawback is to add locally weighted residuals to get a stabilized discretization of the form

$$(L_n\, U_{ni}^h, \phi) + \sum_{T \in T_h} (L_n\, U_{ni}^h, w(\phi))_T = (r_{ni}, \phi) + \sum_{T \in T_h} (r_{ni}, w(\phi))_T\,, \tag{13}$$

where $w(\phi)$ has to be defined with respect to the operator $L_n$ (see e.g. FRANCA and FREY [19], LUBE and WEISS [30], TOBISKA and VERFÜRTH [38]). Two important classes of stabilized methods are the streamline diffusion and the more general Galerkin/least–squares finite element method.

The linear systems are solved by direct or iterative methods. While direct methods work quite satisfactorily in one–dimensional and even two–dimensional

applications, iterative solvers such as Krylov subspace methods perform considerably better with respect to CPU–time and memory requirements for large two– and three–dimensional problems. We mainly use the BICGSTAB–algorithm [40] with ILU–preconditioning.

After computing the approximate intermediate values $U_{ni}^h$ a posteriori error estimates can be used to give specific assessment of the error distribution. Considering a hierarchical decomposition

$$S_h^{q+1} = S_h^q \oplus Z_h^{q+1}\,, \tag{14}$$

where $Z_h^{q+1}$ is the subspace that corresponds to the span of all additional basis functions needed to extend the space $S_h^q$ to higher order, an attractive idea of an efficient error estimation is to bound the spatial error by evaluating its components in the space $Z_h^{q+1}$ only. This technique is known as hierarchical error estimation and has been accepted to provide efficient and reliable assessment of spatial errors (BORNEMANN, ERDMANN, and KORNHUBER [8], DEUFLHARD, LEINEN and YSERENTANT [17], BANK and SMITH [2]). In LANG [28], the hierarchical basis technique has been carried over to time–dependent nonlinear problems. Defining an a posteriori error estimator $E_{n+1}^h \in Z_h^{q+1}$ by

$$E_{n+1}^h = E_{n0}^h + \sum_{i=1}^{s} m_i E_{ni}^h \tag{15}$$

with $E_{n0}^h$ approximating the projection error of the initial value $u_n$ in $Z_h^{q+1}$ and $E_{ni}^h$ estimating the spatial error of the intermediate value $U_{ni}^h$, the local spatial error for a finite element $T \in T_h$ can be estimated by $\eta_T := \|E_{n+1}^h\|_T$. The error estimator $E_{n+1}^h$ is computed by linear systems which can be derived from (13). For practical computations the spatially global calculation of $E_{n+1}^h$ is normally approximated by a small element–by–element calculation. This leads to an efficient algorithm for computing a posteriori error estimates which can be used to determine an adaptive strategy to improve the accuracy of the numerical approximation where needed. A rigorous a posteriori error analysis for a Rosenbrock–Galerkin finite element method applied to nonlinear parabolic systems is given in LANG [28]. In our applications we applied linear finite elements and measured the spatial errors in the space of quadratic functions.

In order to produce a nearly optimal mesh, those finite elements $T$ having an error $\eta_T$ larger than a certain threshold are refined. After the refinement improved finite element solutions $U_{ni}^h$ defined by (13) are computed. The whole procedure solve–estimate–refine is applied several times until a prescribed spatial tolerance $\|E_{n+1}^h\| \leq TOL_x$ is reached. To maintain the nesting property of the finite element subspaces coarsening takes place only after an accepted time step before starting the multilevel process at a new time. Regions of small errors are identified by their $\eta$–values.

# 4 Applications

## 4.1 Stability of Flame Balls

The profound understanding of premixed gas flames near extinction or stability limits is important for the design of efficient, clean–burning combustion engines and for the assessment of fire and explosion hazards in oil refineries, mine shafts, etc. Surprisingly, the near–limit behaviour of very simple flames is still not well–known. Since these phenomena are influenced by bouyant convection, typically experiments are performed in a $\mu g$ environment. Under these conditions transport mechanisms such as radiation and small Lewis number effects, the ratio of thermal diffusivity to the mass diffusivity, come into the play. Seemingly stable flame balls are one of the most exciting appearances which were accidentally discovered in drop–tower experiments by RONNEY [34] and confirmed later in parabolic aircraft flights. First theoretical investigations on purely diffusion–controlled stationary spherical flames were done by ZELDOVICH [42]. 40 years later his flame balls were predicted to be unstable [11]. However, encouraged by the above new experimental discoveries, BUCKMASTER and collaborators [9] have shown that for low Lewis numbers flame balls can be stabilized including radiant heat loss which was not considered before (see Fig. 1 for a configuration of a stationary flame ball). Nowadays there is an increasing interest in high–quality $\mu g$ space experiments necessary to assess the steady properties and stability limits of flame balls (see NASA information at http://cpl.usc.edu/SOFBALL/sofball.html).

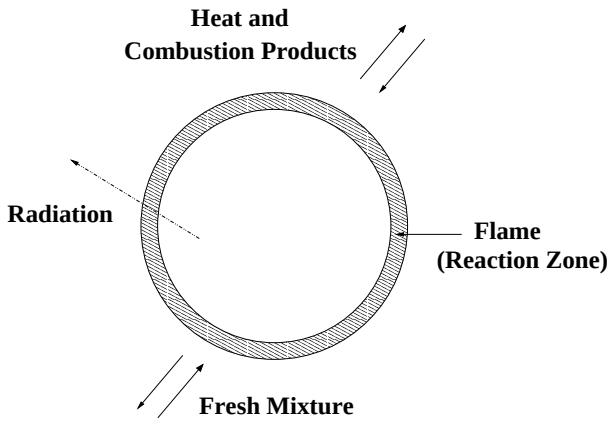


Figure 1: Configuration of a stationary flame ball. Diffusional fluxes of heat and combustion products (outwards) and of fresh mixture (inwards) together with radiative heat loss cause a zero mass–averaged velocity.

Although analytical modelling has identified the key physical ingredients of spherical premixed flames, quantitative confirmation can only come from detailed numerical simulations. Usually, spherically symmetric one–dimensional flame codes are used to investigate steady properties, stability limits, and dynamics of flame balls (see e.g. BUCKMASTER, SMOOKE, and GIOVANGILI [10], WU, RONNEY, COLANTONIO, and VANZANDT [41]). Higher dimensional simulations are very rare due to their great demand for local mesh adaptation in order to resolve the thin reaction layers. In BOCKHORN, FRÖHLICH, and SCHNEIDER [4] and KAGAN, SIVASHINSKI [25] two–dimensional computations of flame balls were presented. Three–dimensional investigations using parallel architectures were published in BOCKHORN, FRÖHLICH, GERLINGER, and SCHNEIDER [5].

The mathematical model we shall adopt is that of [4] which is based on the constant density assumption. In dimensionless form it reads

$$\begin{aligned} \partial_t T - \nabla^2 T &= w - s\,, \\ \partial_t Y - \frac{1}{Le}\nabla^2 Y &= -w\,, \\ w &= \frac{\beta^2}{2\,Le}\,Y \exp\left(\frac{\beta(T-1)}{1+\alpha(T-1)}\right)\,, \qquad (16) \\ s &= c\,\frac{\overline{T}^4 - \overline{T}_u^4}{(\overline{T}_b - \overline{T}_u)^4}\,. \end{aligned}$$

Here, $T := (\overline{T} - \overline{T}_u)/(\overline{T}_b - \overline{T}_u)$ is the nondimensional temperature determined by the dimensional temperatures $\overline{T}$, $\overline{T}_u$, and $\overline{T}_b$, where the indices $u$ and $b$ refer to the unburnt and burnt state of an adiabatic plane flame, respectively. $Y$ represents the mass fraction of the deficient component of the mixture. The chemical reaction rate $w$ is modelled by an one–step Arrhenius term incorporating the dimensionless activation energy $\beta$, the Lewis number $Le$, and the heat release parameter $\alpha := (\overline{T}_b - \overline{T}_u)/\overline{T}_b$. Heat loss is generated by a radiation term $s$ modelled for the optically thin limit. The strength of the radiative loss is mainly determined by the constant $c$ which depends on the Stefan–Boltzmann constant and the Planck length. These relatively simple equations are widely accepted to capture much of the essential physics of flame balls [9, 10]. Comparisons of analytical treatments to experimental results provide strong evidence of the model's validity.

In the following computations, the conditions have been chosen similar to the experiments made by RONNEY [34] for a 6.5% $H_2$–air flame. We set $\overline{T}_u = 300K$, $\overline{T}_b = 830K$, $Le = 0.3$, $\beta = 10$, and derive $\alpha = 0.64$. In [34], additional $CF_3Br$ as a tracer concentration in the mixture was used to increase the heat loss by radiation. Low concentration of $CF_3Br$ yields cellular instability of the flame balls, whereas for increased heat loss due to an increased concentration of the tracer stable flame balls can be observed. To simulate this behaviour we use different values of $c$ in (16), here $c = 0.01$ and $c = 0.1$.

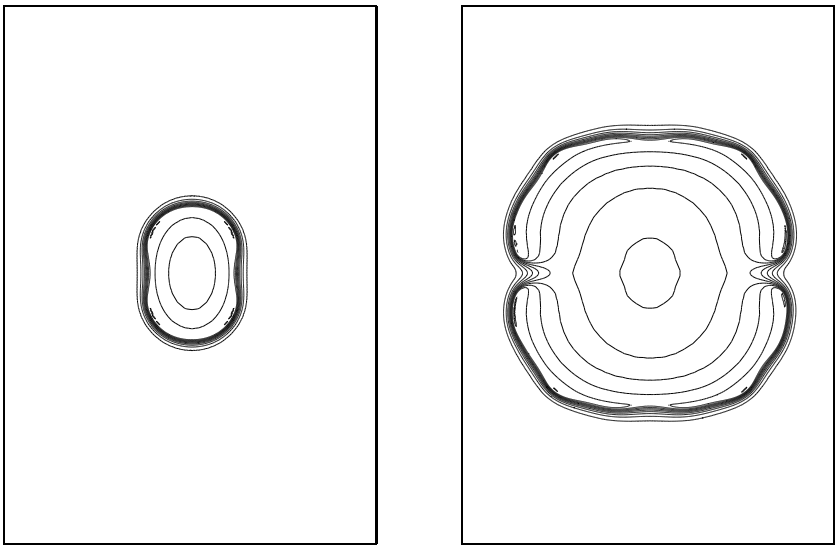

Figure 2: Two–dimensional flame ball with $Le = 0.3$, $c = 0.01$. Iso–thermals $T = 0.1, 0.2, \ldots, 1.0$ at times $t = 10$ and 30.

The computational domain has to be sufficiently large in order to avoid any disturbance caused by the boundary. Typically, sizes of 100 times the flame ball radius are needed to obtain domain–independent solutions due to the long far–field thermal profiles [41]. In those cases, the conductive fluxes at the outer boundary are zero. We consider domains $\Omega = [-L, L]^d$, $d = 2, 3$, with $L = 200$ according to the initial flame radius $r_0 \in [0.2, 2.5]$. As initial conditions we take the analytic solution for a steady plane flame in the high activitation energy limit [4, 5] and use in some calculations a local stretching to generate an elliptic front. In the following we report on two different scenarios, unstable and quasi–stationary flame balls.

*Unstable two–dimensional flame balls.* We set $c = 0.01$ and take an initial elliptic flame with axis' ratio of $1 : 4$. After a short time an instability develops which results in a local quenching of the flame as can be seen in Fig. 2 and Fig. 3. After a while the flame is splitted into two separate smaller flames, which separate again and continue propagating. It can nicely be seen that the dynamic spatial mesh chosen by our adaptive algorithm for TOL $= 0.005$ is well–fitted to the behaviour of the solution. More grid points are automatically placed in regions of high activity in order to resolve the steep solution gradients within the thin reaction layer.

*Quasi–stationary two–dimensional flame balls.* Fixing $c = 0.1$ in (16) and varying the initial radii for a circular flame in a large number of calculations, we found quasi–stationary flame ball configurations. In Fig. 4 we have plotted

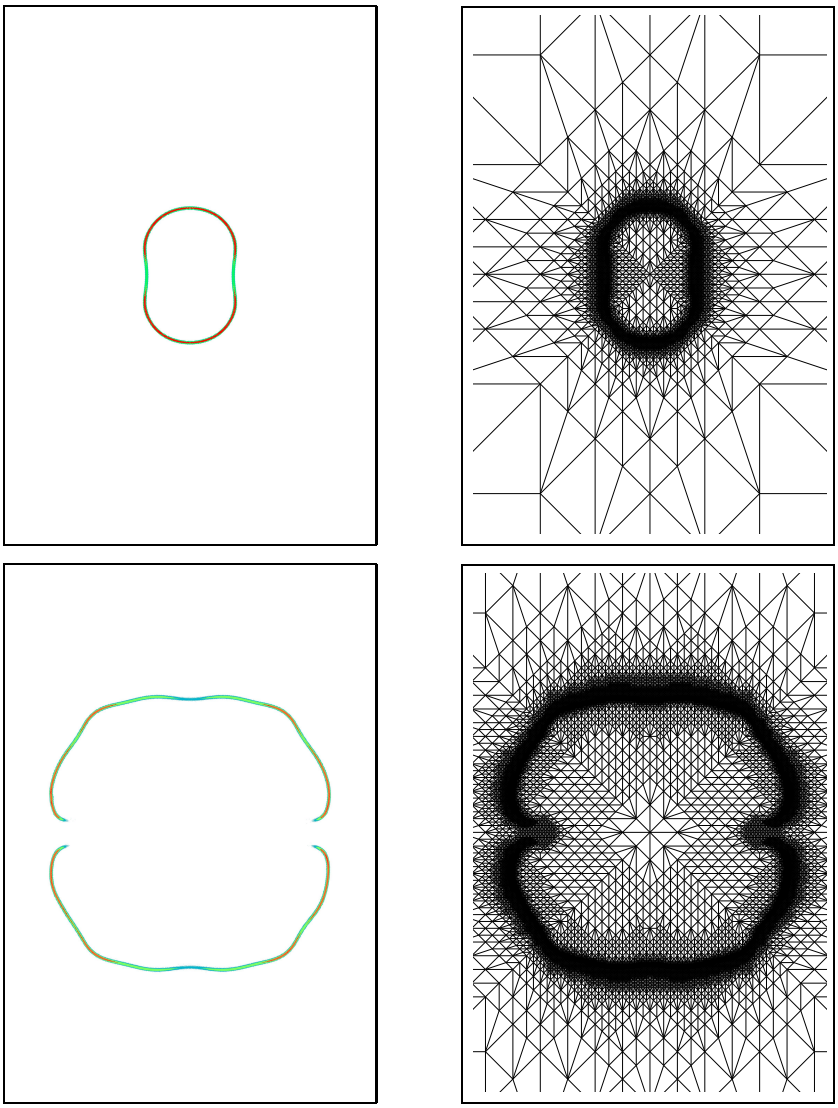

Figure 3: Two–dimensional flame ball with $Le = 0.3$, $c = 0.01$. Reaction rate $w$ at times $t = 10, 30$, and corresponding grids.

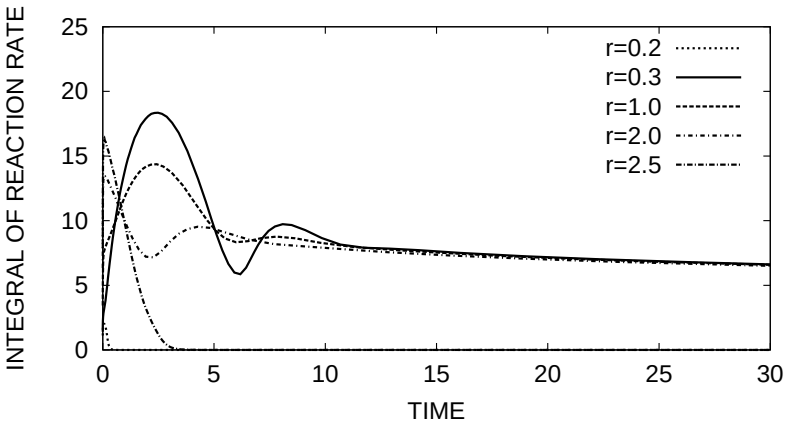


Figure 4: Two–dimensional flame ball with $Le = 0.3$, $c = 0.1$. Integrated reaction rate for different initial radii.

the evolution of the integrated reaction rate $\int_\Omega w(t,x,y)dxdy$ for selected initial radii. For too small and too large radii the flame is quickly extinguished. In between we observe a convergence process to a quasi–steady state characterized by a very slow decrease of the integrated reaction rate. The corresponding flame diameter is around 2. Similar results for $c = 0.05$ were reported in [4].

*Splitting of three–dimensional flame balls.* In the three–dimensional case we get a more complex pattern formation. Just to give an impression we select one typical example taken from [5]. Starting with an ellipsoid having axis' ratio of $1 : 1 : 2$, the flame ball is splitted along the z–axis due to the thermo–diffusive instabilities and further splitting occur afterwards (see Fig. 5). Although we were able to detect certain parameter regions for extinction resulting from excessive heat loss, we have not found configurations that are stable for longer time periods yet. This is the subject of current research.

## 4.2 Brine Transport in Porous Media

High–level radioactive waste is often disposed in salt domes. The safety assessment of such a repository requires the study of groundwater flow enriched with salt. The observed salt concentration can be very high with respect to seawater, leading to sharp and moving freshwater–saltwater fronts. In such a situation, the basic equations of groundwater flow and solute transport have to be modified (Hassanizadeh and Leijnse [24]). We use the physical model proposed by Trompert, Verwer, and Blom [39] for a non–isothermal, single–phase, two–component saturated flow. It consists of the brine flow equation, the salt

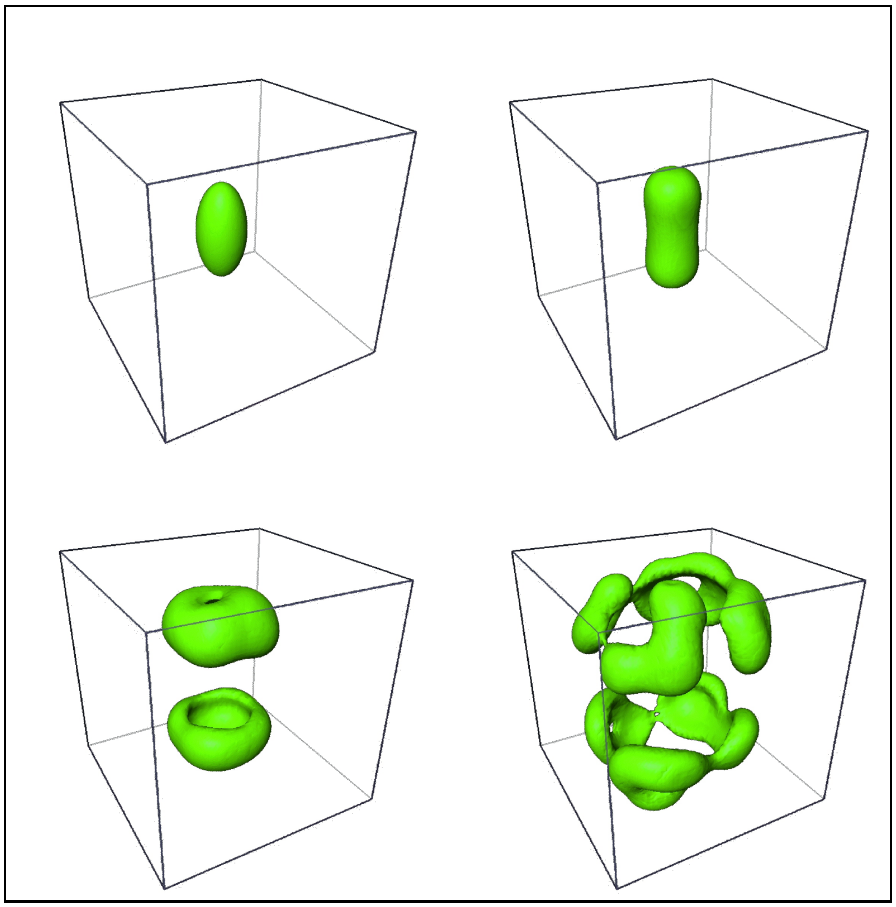

Figure 5: Three–dimensional flame ball with $Le = 0.3$, $c = 0.1$. Iso–thermals $T = 0.8$ at times $t = 0.0, 2.0, 5.0,$ and $8.0$.

transport equation, and the temperature equation and reads

$$n\rho\,(\beta\,\partial_t p + \gamma\,\partial_t w + \alpha\,\partial_t T) + \nabla\cdot(\rho\mathbf{q}) = 0\,, \tag{17}$$

$$n\rho\,\partial_t w + \rho\mathbf{q}\cdot\nabla w + \nabla\cdot(\rho J^w) = 0\,, \tag{18}$$

$$(nc\rho + (1-n)\rho^s c^s)\partial_t T + \rho\, c\,\mathbf{q}\cdot\nabla T + \nabla\cdot J^T = 0\,, \tag{19}$$

supplemented with the state equations for the density $\rho$ and the viscosity $\mu$ of the fluid

$$\begin{aligned}\rho &= \rho_0 \exp\left(\alpha(T-T_0) + \beta(p-p_0) + \gamma w\right)\,,\\ \mu &= \mu_0\,(1.0 + 1.85w - 4.0w^2)\,.\end{aligned}$$

Here, the pressure $p$, the salt mass fraction $w$, and the temperature $T$ are the independent variables, which form a coupled system of nonlinear parabolic equations.

The Darcy velocity $\mathbf{q}$ of the fluid is defined as

$$\mathbf{q} = -\frac{K}{\mu}\left(\nabla p - \rho\mathbf{g}\right)\,,$$

where $K$ is the permeability tensor of the porous medium, which is supposed to be of the form $K = \mathrm{diag}(k)$, and $\mathbf{g}$ is the acceleration of gravity vector. The salt dispersion flux vector $J^w$ and the heat flux vector $J^T$ are defined as

$$J^w = -\left((nd_m + \alpha_T|\mathbf{q}|)\,I + \frac{\alpha_L - \alpha_T}{|\mathbf{q}|}\mathbf{q}\mathbf{q}^T\right)\nabla w\,,$$

$$J^T = -\left((\kappa + \lambda_T|\mathbf{q}|)\,I + \frac{\lambda_L - \lambda_T}{|\mathbf{q}|}\mathbf{q}\mathbf{q}^T\right)\nabla T\,,$$

where $|\mathbf{q}| = \sqrt{\mathbf{q}^T\mathbf{q}}$.

Writing the system of the three balance equations (17)–(19) in the form (8), we find for the $3\times 3$ matrix $B$

$$B(p,w,T) = \begin{pmatrix} n\rho\beta & n\rho\gamma & n\rho\alpha \\ 0 & n\rho & 0 \\ 0 & 0 & nc\rho + (1-n)\rho^s c^s \end{pmatrix}\,.$$

Since the compressibility coefficient $\beta$ is very small, the matrix $B$ is nearly singular and, as known (HAIRER and WANNER [23], VI.6), linearly implicit time integrators suitable for differential algebraic systems of index 1 do not give precise results. This is mainly due to the fact that for $\beta = 0$ the matrix $B$ becomes singular and additional consistency conditions have to be satisfied to avoid order reduction. We have applied the Rosenbrock solver ROWDAIND2 [31], which handles both situations, $\beta = 0$ and $\beta \neq 0$.

| | | | |
|---|---|---|---|
| $n$ | porosity | 0.4 | |
| $k$ | permeability | $10^{-10}$ | $m^2$ |
| $d_m$ | molecular diffusion | 0.0 | $m^2 s^{-1}$ |
| $\alpha_T$ | transversal dispersivity | 0.002 | $m$ |
| $\alpha_L$ | longitudinal dispersivity | 0.01 | $m$ |
| $c$ | heat capacity | 4182 | $J\,kg^{-1}K^{-1}$ |
| $c^s$ | solid heat capacity | 840 | $J\,kg^{-1}K^{-1}$ |
| $\kappa$ | heat conductivity | 4.0 | $J\,s^{-1}m^{-1}K^{-1}$ |
| $\lambda_T$ | transversal heat conductivity | 0.001 | $J\,m^{-2}K^{-1}$ |
| $\lambda_L$ | longitudinal heat conductivity | 0.01 | $J\,m^{-2}K^{-1}$ |
| $\rho^s$ | solid density | 2500 | $kg\,m^{-3}$ |
| $\rho_0$ | freshwater density | 1000 | $kg\,m^{-3}$ |
| $T_0$ | reference temperature | 290 | $K$ |
| $p_0$ | reference pressure | $10^5$ | $kg\,m^{-1}s^{-2}$ |
| $\alpha$ | temperature coefficient | $-3.0 \cdot 10^{-4}$ | $K^{-1}$ |
| $\beta$ | compressibility coefficient | $4.45 \cdot 10^{-10}$ | $m\,s^2 kg^{-1}$ |
| $\gamma$ | salt coefficient | $ln(1.2)$ | |
| $\mu_0$ | reference viscosity | $10^{-3}$ | $kg\,m^{-1}s^{-1}$ |

Table 3: Parameters of the two–dimensional brine transport model.

An additional feature of the model is that the salt transport equation (18) is usually dominated by the advection term. In practice, global Peclet numbers can range between $10^2$ and $10^4$, as reported in [39]. On the other hand, the temperature and the flow equation are of standard parabolic type with convection terms of moderate size.

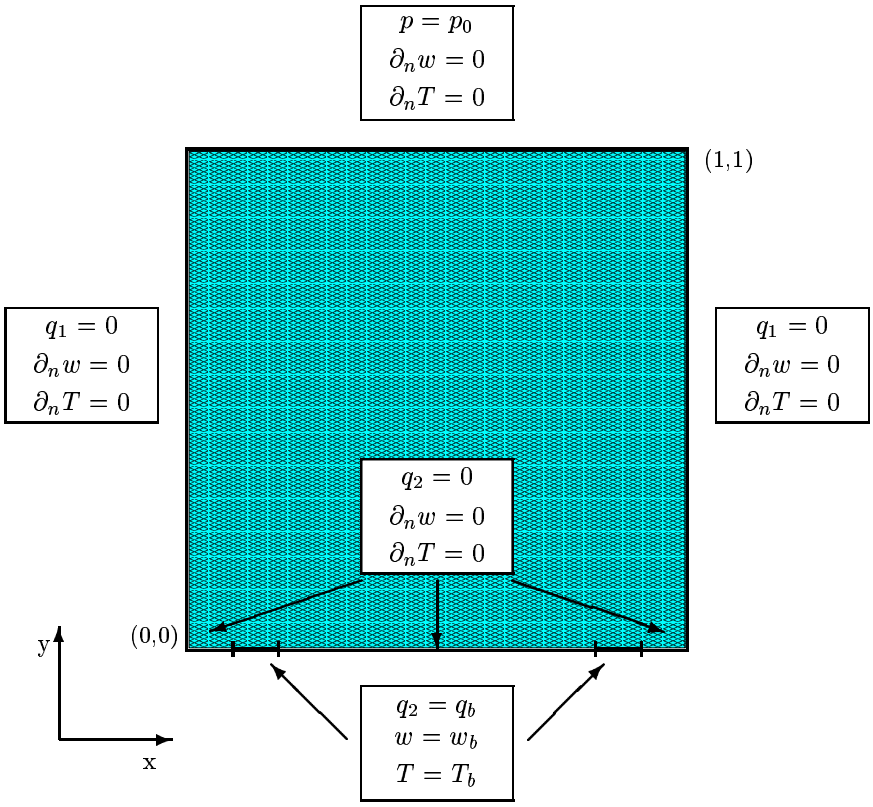


Figure 6: Two–dimensional brine transport. Computational domain and boundary conditions for $t > 0$. The two gates where warm brine is injected are located at $(x, y) : \frac{1}{11} \le x \le \frac{2}{11}$, $\frac{9}{11} \le x \le \frac{10}{11}$, $y = 0$.

*Two–dimensional warm brine injection.* This problem was taken from [39]. We consider a (very) thin vertical column filled with a porous medium. This justifies the use of a two–dimensional flow domain $\Omega = \{(x, y) : 0 < x, y < 1\}$ representing a vertical cross–section. The acceleration of gravity vector points downward and takes the form $\mathbf{g} = (0, -g)^T$, where the gravity constant $g$ is set to 9.81. The initial values at $t = 0$ are

$$p(x, y, 0) = p_0 + (1 - y)\rho_0 g, \quad w(x, y, 0) = 0\,, \text{ and } T(x, y, 0) = T_0\,.$$

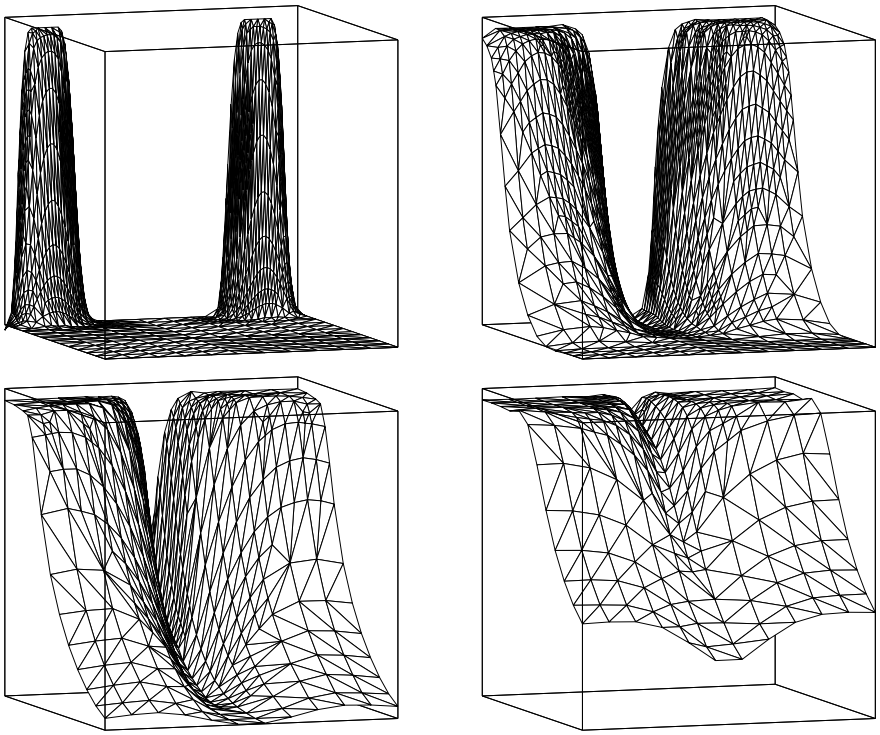

Figure 7: Two–dimensional brine transport. Distribution of salt concentration at $t = 500, 5000, 10000$, and 20000 with corresponding spatial grids.

The boundary conditions are described in Fig. 6. We set $w_b = 0.25$, $T_b = 292.0$, and $q_b = 10^{-4}$. The remaining parameters used in the model are given in Table 3.

Warm brine is injected through two gates at the bottom. This gives rise to sharp fronts between salt and fresh water, which have to be resolved with fine meshes in the neighbourhood of the gates, see Fig. 7. Later the solutions smooth out with time until the porous medium is filled completely with brine. Our computational results are comparable to those obtained in [39] with a method of lines approach coupled with a local uniform grid refinement. In Fig. 8 we show the time steps and the degrees of freedom chosen by the KARDOS solver to integrate over $t \in [0, 2 \cdot 10^4]$. The curves nicely reflect the high dynamics at the beginning in both, time and space, while larger time steps and coarser grids are selected in the final part of the simulation.

*Three–dimensional pollution with salt water.* Here, we consider Problem III of [3] and simulate a salt pollution of fresh water flowing from left to right through a tank $\Omega = \{(x, y, z) : 0 \leq x \leq 2.5, 0 \leq y \leq 0.5, 0 \leq z \leq 1.0\}$ filled with a porous medium. The flow is supposed to be isothermal ($\alpha = 0$) and

incompressible ($\beta = 0$). Hence, the problem consists now of two PDEs with a singular $2 \times 2$ matrix $B(p, w)$ multiplying the vector of temporal derivatives. The acceleration of gravity vector takes the form $\mathbf{g} = (0, 0, -g)^T$.

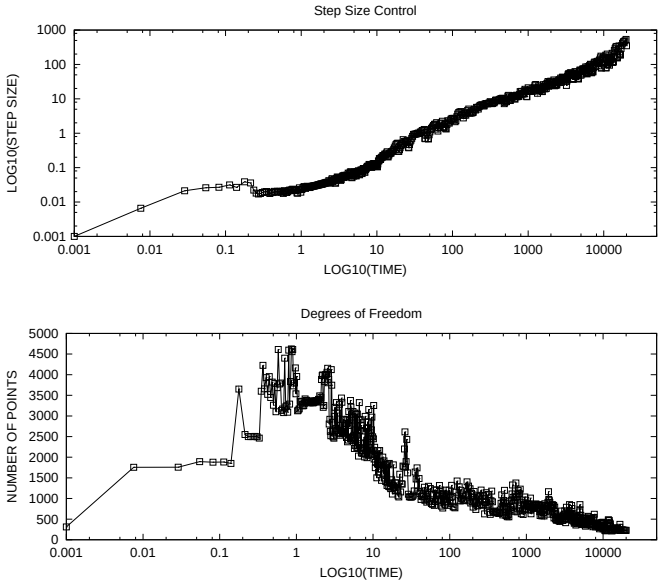


Figure 8: Two–dimensional brine transport. Evolution of time steps and number of spatial discretization points for $TOL_t = TOL_x = 0.005$.

The brine having a salt mass fraction $w_b = 0.0935$ is injected through a small slit $S = \{(x, y, 1) : 0.375 \leq x \leq 0.4375,\ 0.25 \leq y \leq 0.3125\}$ at the top of the tank. We note that the slit chosen here differs slightly from that used in [3]. The initial values at $t = 0$ are taken as

$$p(x, y, z, 0) = p_0 + (0.03 - 0.012x + 1.0 - z)\rho_0 g, \quad w(x, y, z, 0) = 0\,,$$

and the boundary conditions are

$$\begin{aligned}
&p = p(x, y, z, 0)\,,\ w = 0\,, && \text{on } x = 0\,,\\
&p = p(x, y, z, 0)\,,\ \partial_n w = 0\,, && \text{on } x = 2.5\,,\\
&q_2 = 0\,,\ \partial_n w = 0\,, && \text{on } y = 0 \text{ and } y = 1\,,\\
&q_3 = 0\,,\ \partial_n w = 0\,, && \text{on } z = 0 \text{ and } \{z = 1\} \setminus S\,,\\
&\rho q_3 = -4.95 \cdot 10^{-2}\,,\ w = w_b = 0.0935\,, && \text{on } S\,.
\end{aligned}$$

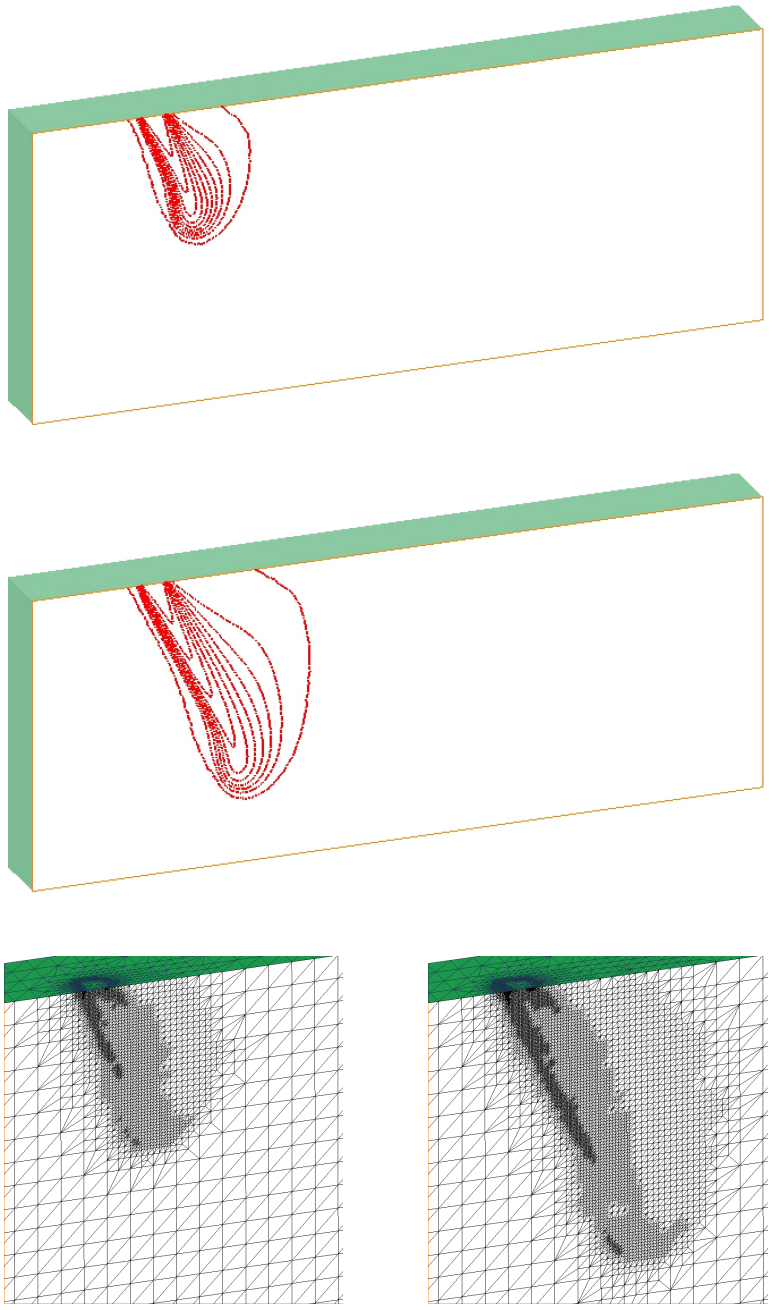

Figure 9: Three–dimensional brine transport. Level lines of the salt concentration $w = 0.0, 0.01, \ldots, 0.09$, in the plane $y = 0.28125$ after two hours (top) and four hours (middle), and corresponding spatial grids (bottom) in the neighborhood of the inlet.

The parameters used in the three–dimensional simulation are given in Tab. 4. Additionally, the state equation for the viscosity of the fluid is modified to

$$\mu = \mu_0(1.0 + 1.85w - 4.1w^2 + 44.5w^3)\,.$$

In Fig. 9 we show the distribution of the salt concentration in the plane $y = 0.28125$ after two and four hours. The pollutant is slowly transported by the flow while sinking to the bottom of the tank. The steepness of the solution is higher in the back of the pollution front, which causes fine meshes in this region. Despite the dominating convection terms no wiggles are visible, especially at the inlet. An interesting observation is the unexpected drift in front of the solution – a phenomenon which was also observed by BLOM and VERWER [3].

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| $n$ | = | 0.35 | $\gamma$ | = | $ln(2)$ | $nd_m$ | = | $10^{-9}$ |
| $\kappa$ | = | $7.18 \cdot 10^{-11}$ | $\alpha_T$ | = | 0.001 | $\alpha_L$ | = | 0.01 |
| $p_0$ | = | 0.0 | $\mu_0$ | = | 0.001 | $\rho_0$ | = | 1000 |

Table 4: Parameters of the three–dimensional brine transport model.

# 5 Conclusion

Dynamical process simulation of complex real–life problems advises the use of modern algorithms, which are able to judge the quality of their numerical approximations and to determine an adaptation strategy to improve their accuracy in both the time and the space discretization. The paper presented a combination of efficient linearly implicit time integrators of Rosenbrock type and error–controlled grid improvement based on a multilevel finite element method. This approach leads to a minimization of the degrees of freedom necessary to reach a prescribed error tolerance. The savings in computing time are substantial and allow the solution of even complex problems in a moderate range of time.

# References


[1] R.E. Bank, PLTMG: A Software Package for Solving Elliptic Partial Differential Equations - User's Guide 8.0, *SIAM*, 1998.

[2] R.E. Bank and R.K. Smith, A Posteriori Error Estimates Based on Hierarchical Bases, *SIAM J. Numer. Anal.* **30** (1993), 921–935.

[3] J.G. Blom and J.G. Verwer, VLUGR3: A Vectorizable Adaptive Grid Solver for PDEs in 3D, I. Algorithmic Aspects and Applications, *Appl. Numer.Math.* **16** (1994), 129–156.

[4] H. Bockhorn, J. Fröhlich, and K. Schneider, An Adaptive Two–Dimensional Wavelet–Vaguellette Algorithm for the Computation of Flame Balls, *Combust. Theory Modelling* **3** (1999), 177–198.

[5] H. Bockhorn, J. Fröhlich, W. Gerlinger, and K. Schneider, Numerical Investigations on the Stability of Flame Balls, *in: K. Papailiou et al. (eds.), Computational Fluid Dynamics '98, Vol. 1, John Wiley & Sons* (1998), 990–995.

[6] F.A. Bornemann, An Adaptive Multilevel Approach to Parabolic Equations. II. Variable–Order Time Discretization Based on a Multiplicative Error Correction, *IMPACT of Comput. in Sci. and Engrg.* **3** (1991), 93–122.

[7] F.A. Bornemann, An Adaptive Multilevel Approach to Parabolic Equations. III. 2D Error Estimation and Multilevel Preconditioning, *IMPACT of Comput. in Sci. and Engrg.* **4** (1992), 1–45.

[8] F.A. Bornemann, B. Erdmann, and R.Kornhuber, A Posteriori Error Estimates for Elliptic Problems in Two and Three Space Dimensions, *SIAM J. Numer. Anal.* **33** (1996), 1188–1204.

[9] J.D. Buckmaster, G. Joulin, and P.D. Ronney, Effects of Heat Loss on the Structure and Stability of Flame Balls, *Combust. Flame* **79** (1990), 381–392.

[10] J.D. Buckmaster, M. Smooke, and V. Giovangili, Analytical and Numerical Modelling of Flame–Balls in Hydrogen–Air Mixtures, *Combust. Flame* **94** (1993), 113–124.

[11] J.D. Buckmaster, S. Weeratunga, The Stability and Structure of Flame–Bubbles, *Comb. Sci. Tech.* **35** (1984), 287–296.

[12] K. Dekker and J.G. Verwer, Stability of Runge–Kutta methods for stiff nonlinear differential equations, *North–Holland Elsevier Science Publishers*, 1984.

[13] P. Deuflhard, Uniqueness Theorems for Stiff ODE Initial Value Problems, *in: D.F. Griffiths and G.A. Watson (eds.), Numerical Analysis 1989, Proceedings of the 13th Dundee Conference, Pitman Research Notes in Mathematics Series 228, Longman Scientific and Technical* (1990), 74–87.

[14] P. Deuflhard, Recent Progress in Extrapolation Methods for Ordinary Differential Equations, *SIAM Rev.* **27** (1985), 505–535.

[15] P. Deuflhard and F. Bornemann, Numerische Mathematik II, Integration Gewöhnlicher Differentialgleichungen, *De Gruyter Lehrbuch, Berlin, New York*, 1994.

[16] P. Deuflhard, J. Lang, and U. Nowak, Adaptive Algorithms in Dynamical Process Simulation, *in: H. Neunzert (ed.), Progress in Industrial Mathematics at ECMI'94, Wiley–Teubner* (1996), 122–137.

[17] P. Deuflhard, P. Leinen, and H. Yserentant, Concepts of an Adaptive Hierarchical Finite Element Code, *IMPACT of Comput. in Sci. and Engrg.* **1** (1989), 3–35.

[18] B. Erdmann, J. Lang, and R. Roitzsch, KASKADE Manual, Version 2.0, *Report TR93–5, Konrad–Zuse–Zentrum für Informationstechnik Berlin*, 1993.

[19] L.P. Franca and S.L. Frey, Stabilized Finite Element Methods, *Comput. Methods Appl. Mech. Engrg.* **99** (1992), 209–233.

[20] K. Gustafsson, Control–Theoretic Techniques for Stepsize Selection in Implicit Runge-Kutta Methods, *ACM Trans. Math. Software* **20** (1994), 496–517.

[21] K. Gustafsson, M. Lundh, and G. Söderlind, A PI Stepsize Control for the Numerical Solution of Ordinary Differential Equations. *BIT* **28** (1988), 270–287.

[22] E. Hairer, S.P. Nørsett, and G. Wanner, Solving Ordinary Differential Equations I, Nonstiff Problems, *Springer–Verlag, Berlin, Heidelberg, New York*, 1987.

[23] E. Hairer and G. Wanner, Solving Ordinary Differential Equations II, Stiff and Differential-Algebraic Problems, *Second Revised Edition, Springer–Verlag, Berlin, Heidelberg, New York*, 1996.

[24] S.M. Hassanizadeh and T. Leijnse, On the Modeling of Brine Transport in Porous Media, *Water Resources Research* **24** (1988), 321–330.

[25] L. Kagan and G. Sivashinski, Self–Fragmentation of Nonadiabatic Cellular Flames, *Combust. Flames* **108** (1997), 220–226.

[26] J. Lang and A. Walter, A Finite Element Method Adaptive in Space and Time for Nonlinear Reaction–Diffusion Systems, *IMPACT of Comput. in Sci. and Engrg.* **4** (1992), 269–314.

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

[28] J. Lang, Adaptive Multilevel Solution of Nonlinear Parabolic PDE Systems. Theory, Algorithm, and Applications, *appears in: Lecture Notes in Computational Science and Engineering, Springer–Verlag, Berlin, Heidelberg, New York*, 2000.

[29] J. Lang and J. Verwer, ROS3P - an Accurate Third–Order Rosenbrock Solver Designed for Parabolic Problems, *Report MAS–R0013, CWI, Amsterdam*, 2000.

[30] G. Lube and D. Weiss, Stabilized Finite Element Methods for Singularly Perturbed Parabolic Problems, *Appl. Numer. Math.* **17** (1995), 431–459.

[31] Ch. Lubich and M. Roche, Rosenbrock Methods for Differential–Algebraic Systems with Solution–Dependent Singular Matrix Multiplying the Derivative, *Comput.* **43** , 325–342, (1990).

[32] M. Roche, Runge–Kutta and Rosenbrock Methods for Differential–Algebraic Equations and Stiff ODEs, *PhD thesis, Université de Genève*, 1988.

[33] M. Roche, Rosenbrock Methods for Differential Algebraic Equations, *Numer. Math.* **52** (1988), 45–63.

[34] P.D. Ronney, Near–Limit Flame Structures at Low Lewis Number, *Combust. Flame* **82** (1990), 1–14.

[35] H.H. Rosenbrock, Some General Implicit Processes for the Numerical Solution of Differential Equations, *Computer J.* (1963), 329–331.

[36] G. Steinebach, Order–Reduction of ROW–methods for DAEs and Method of Lines Applications, *Preprint 1741, Technische Hochschule Darmstadt, Germany*, 1995.

[37] K. Strehmel and R. Weiner, Linear–implizite Runge–Kutta–Methoden und ihre Anwendungen, *Teubner Texte zur Mathematik 127, Teubner Stuttgart, Leipzig*, 1992.

[38] L. Tobiska and R. Verfürth, Analysis of a Streamline Diffusion Finite Element Method for the Stokes and Navier–Stokes Equation, *SIAM J. Numer. Anal.* **33** (1996), 107–127.

[39] R.A. Trompert, J.G. Verwer, and J.G. Blom, Computing Brine Transport in Porous Media with an Adaptive–Grid Method, *Int. J. Numer. Meth. Fluids* **16** (1993), 43–63.

[40] H.A. van der Vorst, BI–CGSTAB: A fast and smoothly converging variant of BI–CG for the solution of nonsymmetric linear systems, *SIAM J. Sci. Stat.* **13** (1992), 631–644.

[41] M.S. Wu, P.D. Ronney, R.O. Colantonio, and D.M. Vanzandt, Detailed Numerical Simulation of Flame Ball Structure and Dynamics, *Combust. Flame* **116** (1999), 387–397.

[42] Ya.B. Zeldovich, Theory of Combustion and Detonation of Gases, *Academy of Sciences (USSR)*, 1944.