AFRL-AFOSR-UK-TR-2013-0026

Heterogeneous Multiscale Methods Applied to Stiff Problems with Varying Scales

Professor Olof Runborg

NADA CSC KTH
Lindstedtsv 3
Stockholm, Sweden 100 44

EOARD Grant 07-3081

Report Date: July 2013

Final Report from 1 June 2007 to 31 May 2012

Air Force Research Laboratory
Air Force Office of Scientific Research
European Office of Aerospace Research and Development
Unit 4515 Box 14, APO AE 09421

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From – To)*
15 July 2013	**Final Report**	**1 June 2007 – 31 May 2012**

4. TITLE AND SUBTITLE

Heterogeneous Multiscale Methods Applied to Stiff Problems with Varying Scales

5a. CONTRACT NUMBER

FA8655-07-1-3081

5b. GRANT NUMBER

Grant 07-3081

5c. PROGRAM ELEMENT NUMBER

61102F

6. AUTHOR(S)

Professor Olof Runborg

5d. PROJECT NUMBER

5d. TASK NUMBER

5e. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

NADA CSC KTH
Lindstedtsv 3
Stockholm, Sweden 100 44

8. PERFORMING ORGANIZATION REPORT NUMBER

N/A

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

EOARD
Unit 4515
APO AE 09421-4515

10. SPONSOR/MONITOR'S ACRONYM(S)

AFRL/AFOSR/IOE (EOARD)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

AFRL-AFOSR-UK-TR-2013-0026

12. DISTRIBUTION/AVAILABILITY STATEMENT

Distribution A: Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

This work explores fast numerical methods for solving rate equations that describe the population densities of chemical species or atomic states. The rate equations are very stiff nonlinear ordinary differential equations, with essentially one slow time scale and a large range of fast scales. We consider implicit multistep and one-step methods. They require the solution of a nonlinear system of equations in each time step with a Newton method. To reduce the cost of this, we use approximations or prefactorization of the Jacobian matrix. Different approximation strategies are explored. The importance of exact discrete conservation is highlighted, leading to an approach where the Jacobian is truncated to banded form and remaining off-diagonal elements are adjusted by a weight that depends on the elements in the full Jacobian. The prefactorization approach uses a QZ decomposition of the leading part of the Jacobian, and a separate treatment of a rank one part. Numerical experiments indicate that these methods give accurate results at a low computational cost.

15. SUBJECT TERMS

EOARD, Mathematical Modeling, Turbulence

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18, NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Kevin Bollino
UNCLAS	UNCLAS	UNCLAS	SAR	31	19b. TELEPHONE NUMBER *(Include area code)* +44 (0)1895 616163

Heterogeneous Multiscale Methods Applied to Stiff Problems with Varying Scales

Award no: FA8655-07-13081

Final Report

Period of performance: 1 June 2007 – 31 May 2013

Olof Runborg

Report date: July 15, 2013

Abstract

We develop fast numerical methods for solving rate equations that describe the population densities of chemical species or atomic states. The rate equations are very stiff nonlinear ordinary differential equations, with essentially one slow time scale and a large range of fast scales. We consider implicit multistep and one-step methods. They require the solution of a nonlinear system of equations in each time step with a Newton method. To reduce the cost of this, we use approximations or prefactorization of the Jacobian matrix. Different approximation strategies are explored. The importance of exact discrete conservation is highlighted, leading to an approach where the Jacobian is truncated to banded form and remaining off-diagonal elements are adjusted by a weight that depends on the elements in the full Jacobian. The prefactorization approach uses a QZ decomposition of the leading part of the Jacobian, and a separate treatment of a rank one part. Numerical experiments indicate that these methods give accurate results at a low computational cost.

Contents

List of Figures

1 Background

This project started out with PIs Prof. Heinz-Otto Kreiss and Dr. Jon Tegner at KTH Royal Institute of Technology. In June 2011 it was transferred to Prof. Olof Runborg, also KTH. On the Air Force side, Dr. Jean-Luc Cambier has been the main investigator.

2 Introduction

The main objective of this project is to investigate fast numerical methods for solving ordinary differential equations (ODEs) with strongly varying time scales. The application of the methods is in the solution of rate equations for the population densities of chemical species or atomic states. This in turn is a key part in the simulation of complex reactive fluid dynamics, where a huge number of instances of the rate equations need to be solved.

For a system with N atomic states the ODEs are of the form

$$\frac{dX}{dt} = (M_0 + yM_1 + y^2M_2)X, \qquad y = z^TX, \qquad M_j \in \mathbb{R}^{N+1\times N+1},$$

where z is a fixed vector and $X \in \mathbb{R}^{N+1}$ contains the population densities of the atomic states and ions.

The rate equations are very stiff ODEs with essentially one slow time scale and many fast scales. The gap between them is large, but the range of the fast scales is also large; see the example in Figure 2, where the a typical solution and the eigenvalues of the Jacobian of the right hand side are plotted. This fact makes these ODEs very difficult to solve numerically.

The huge stiffness ratio of the rate equations precludes the use of standard explicit time-stepping methods. It implies a severe stability requirement that leads to a complexity of $O(N^2\lambda_{\text{large}})$ for a system of N states, where λ_{large} is the largest eigenvalue of the Jacobian. The alternative is to use implicit methods, which have no stability requirements and give better complexity. One can choose the time step based on the smallest eigenvalue λ_{small} but a nonlinear system of equation must be solved in each time step. Since the matrices are full, the dependence on N in the complexity is therefore worse; the cost is $O(N^3\lambda_{\text{small}})$. A third possible approach would be explicit "multiscale" time-stepping methods, such as Chebyshev methods [1, 7], heterogeneous multiscale methods (HMM) [3], projective integration methods [4, 5] and flow averaging methods [9]. These have much lower complexity than standard explicit methods, but require a clear separation between the bulk of the scales and a few fast scales. This is not the scale structure of the rate equations, however, which makes the methods less suitable.

Instead we have explored implicit multistep methods where the nonlinear system of equations is solved using approximations or prefactorization of the Jacobian matrix. The methods then becomes less expensive, with a complexity of only $O(N^2\lambda_{\text{small}})$. The approximation is based on the fact that the Jacobian is

strongly diagonally dominant and can be well approximated by a banded matrix. It must be done carefully, however. In particular we find that the discrete conservation of the solver must be upheld exactly also for the approximated solver. Prefactorization cannot be done by a simple LU decomposition since the ODE is nonlinear, but by treating a rank one part of the Jacobian separately, we only need to consider a system of the type $(A + yB)X = b$ for fixed A, B but varying y. This can be solved fast for any y if we precompute the QZ factorization of A and B.

We also consider one step implicit Runge–Kutta methods for the equations. Approximation and prefactorization cannot be used in the same straightforward way in these methods, however.

This report is organized as follows. In Section 3 the physical model is explained and the governing equations are derived. Some properties of the equations and their consequences for numerical approximation are presented in Section 4. Implicit methods with an approximate Jacobian are discussed in Section 5, while the precomputation strategy using QZ-factorization is described in Section 6. In Section 7 one-step methods are explored. The report concludes with Section 8 where a summary of the results is given together with an outlook on future challenges.

3 Physical Model and Assumptions

We consider a simple atomic system (atomic hydrogen) with N electronic states, with a population number density x_n ($n = 0 \dots N-1$). The ionized state has a number density x_+ and the density of free electrons is x_e. By charge conservation, we have:

$$x_e = \sum_q z_q x_q \tag{1}$$

where the summation runs over all atomic states; thus, $z_q \equiv 0$ for $q = 0, \dots N-1$, since the bound electronic states of H are neutral, and $z_+ \equiv 1$ since the hydrogen ion has a unit charge. Charge conservation thus allows us to express one variable (x_e) in terms of the atomic states via (1). Using chemical element conservation, it is also possible to eliminate one other variable (e.g. x_+); for a total initial number of atoms N_H, we have $\sum_q x_q = N_H = \text{constant}$, leaving only $N-1$ independent variables. However, it is often preferable to keep all variables in the system of equations, with the understanding that some eigenvalues may be zero, expressing the conservation properties of the collisional-radiative kinetic equations.

The complete set of number densities of electronic levels of both neutral and ionized atoms form an Atomic State Density Function (ASDF) which can be expressed in the form of a vector $X = \{x_q, q = 0, \dots N\}$. The rates of change of these population densities are given by three physical processes: a) collisional bound-bound transitions, i.e. excitations and de-excitations; b) collisional bound-free, i.e. ionization and recombination; c) radiative bound-bound. For the latter, we consider radiative deexcitations only, i.e. spontaneous decay

of the excited states; the reverse process of radiation absorption and electronic excitation is neglected. Note that we also have ignored radiative bound-free transitions, i.e. radiative capture and photo-ionization. We also consider only electron-impact collisions. Most of these assumptions and simplifications will be relaxed in future work.

Let us first consider a bound-bound transition, for which the rate of change of the population density for level n is of the form:

$$\frac{dx_n}{dt} = -\alpha^{\uparrow}_{(m|n)} x_n x_e + \beta^{\downarrow}_{(n|m)} x_m x_e \tag{2}$$

The first term on the right (2) describes the loss due to excitation ($\uparrow$) from level n to m, as a result of collisions between free electrons (of number density x_e) and existing states (number density x_n); the second term describes the gain due to collisional deexcitation ($\downarrow$) induced by free electrons (x_e), from the state m, with number density x_m. Hereafter, we will denote the indices on the rates such that the left-most index $(f|$ is the final state and the right index $|i)$ is the initial state. The second term (deexcitation) is the reverse process of the former; if there were only two states to consider, this would be the entire rate of change for level n. All transitions involving the state n must be counted, so the rate of change for excitation and deexcitation alone requires us to sum-up the right hand side of equation (2) over all levels $m \neq n$. Note that for the same transition between the levels n and m, we also have:

$$\frac{dx_m}{dt} = +\alpha^{\uparrow}_{(m|n)} x_n x_e - \beta^{\downarrow}_{(n|m)} x_m x_e \tag{3}$$

Consider now the bound-free collisional transitions. Similarly to (2), the rate of change of the population density for level n due to the ionization and recombination process is:

$$\frac{dx_n}{dt} = -\alpha^{i}_{(+|n)} x_n x_e + \beta^{r}_{(n|+)} x_+ x_e^2 \tag{4}$$

The second term on the right-hand-side depends on x_e^2 to indicate that two electrons must be present in the initial state for collisonal recombination to occur (one electron becomes bound, another is present to receive the energy of recombination, to guarantee energy conservation). Again, conservation properties imply that, for this transition:

$$\frac{dx_+}{dt} = \frac{dx_e}{dt} = -\frac{dx_n}{dt} \tag{5}$$

Finally, a bound-bound radiative decay leads to:

$$\frac{dx_n}{dt} = +A_{(m|n)} x_m = -\frac{dx_m}{dt} \tag{6}$$

We can now combine all terms for the rate of change of the population

density of a level n:

$$\begin{aligned}\frac{dx_n}{dt} = &- \sum_{m>n} \alpha^{\uparrow}_{(m|n)} x_e x_n + \sum_{m>n} \beta^{\downarrow}_{(n|m)} x_e x_m + \sum_{m>n} A_{(n|m)} x_m \\ &+ \sum_{m<n} \alpha^{\uparrow}_{(n|m)} x_e x_m - \sum_{m<n} \beta^{\downarrow}_{(m|n)} x_e x_n - \sum_{m<n} A_{(m|n)} x_n \\ &- \alpha^{i}_{(+|n)} x_e x_n + \beta^{r}_{(n|+)} x_+ x_e^2\end{aligned} \tag{7a}$$

This system can be written in matrix form:

$$\frac{dX}{dt} = (M_0 + x_e M_1 + x_e^2 M_2) \cdot X \tag{8}$$

The right side is non-linear (x_e is a function of X) but note that the first entry on the right is due only to the radiative rates A_{nm}, the only purely linear contribution to the source term, while the last term has the highest degree of non-linearity, but is due entirely to the recombination $\beta^{r}_{(n|+)} x_+ x_e^2$, which is a relatively slow source term.

3.1 Kinetic Rates

It is worth describing here the expressions for the kinetic rates, in order to gain some insight into the structure of the spectrum of eigenvalues of the system (7). Using classical collision theory and the Bohr model for the hydrogen atom, the excitation rates are of the form [10]:

$$\alpha^{\uparrow}_{(m|n)} = (4\pi a_0^2) \bar{v}_e \left(\frac{I_H}{kT}\right)^2 (3 f_{nm}) \psi_{nm} \tag{9}$$

where a_0 is the Bohr radius, I_H=13.6 eV is the Rydberg constant,

$$f_{nm} = \frac{32}{3\pi\sqrt{3}} \frac{1}{n^5} \frac{1}{m^3} \frac{1}{\left(\frac{1}{n^2} - \frac{1}{m^2}\right)^3} \tag{10}$$

is the oscillator strength of the transition $n - m$,

$$\bar{v}_e = \left(\frac{8kT_e}{\pi m_e}\right)^{1/2} \tag{11}$$

is the mean thermal electron velocity and

$$\psi_{nm} = \frac{e^{-\xi_{nm}}}{\xi_{nm}} - E_1(\xi_{nm}) \tag{12}$$

where $\xi_{nm} = E_{nm}/kT$ with $E_{nm} = I_H(1/n^2 - 1/m^2)$ the energy gap between levels and E_1 is the exponential integral:

$$E_1(a) = \int_a^{\infty} \frac{e^{-b}}{b} db \tag{13}$$

At equilibrium ("Boltzmann"), the ratio of population densities is:

$$\frac{x_m^*}{x_n^*} \equiv \mathcal{B}_{nm}(T_e) = \frac{g_m}{g_n} e^{-E_{nm}/kT_e}, \tag{14}$$

where g_n is the degeneracy of state n. The rate of change is null at this equilibrium condition and therefore

$$\beta^{\downarrow}_{(n|m)} = \frac{g_n}{g_m} e^{+\xi_{nm}} \cdot \alpha^{\uparrow}_{(m|n)} \tag{15}$$

The formulation of the rate involves an exponential integral with no simple expression; at low temperature ($\xi_{nm} \gg 1$), we can use the approximation:

$$E_1(a) \simeq \frac{e^{-a}}{a}\left(1 - \frac{1}{a}\right) \tag{16}$$

This is not always the case if we consider a large number of states, as the energy gaps E_{nm} decrease and eventually we have $\xi_{nm} \simeq 1$, in which case a better approximation would be:

$$\psi(\xi_{nm}) \simeq \frac{2}{5}\frac{e^{-\xi_{nm}}}{\xi_{nm}} \tag{17}$$

Nevertheless, we will restrict ourselves to the approximation (16) only, in which case:

$$\alpha^{\uparrow}_{(m|n)} \simeq \left[4\pi a_0^2 \cdot \frac{32}{\pi\sqrt{3}} \cdot \bar{v}_e\right] \frac{e^{-\xi_{nm}}}{n^5 m^3 (n^{-2} - m^{-2})^5} \tag{18}$$

and

$$\beta^{\downarrow}_{(n|m)} \simeq \left[4\pi a_0^2 \cdot \frac{32}{\pi\sqrt{3}} \cdot \bar{v}_e\right] \frac{1}{n^3 m^5 (n^{-2} - m^{-2})^5} \tag{19}$$

The factor in brackets is a scale of the rate of change (dx/dt) (an upper bound); in the limit $\xi_{nm} \to 0$ i.e. for the upper states, the rates approach that value. Another scale is the factor I_H/kT in the definition of ξ_{nm}, which will describe how stiff the system is. If that factor is very low (high temperatures), all rates are of the same order; at low temperatures, the exponential term dominates and the range of time scales is increased. Note also that the system is strongly *diagonally dominant*, in the sense that transitions with small change in quantum number ($m - n \simeq 1$) have a higher rate than those with $m - n \gg 1$. In fact, a fairly good approximation may be to consider a ladder process, i.e transitions between neighboring states only; this approximation may not always be valid however, when other atoms besides Hydrogen are considered, and this will be studied further as we extend the numerical integration schemes to more complex systems.

The ionization rate coefficient is

$$\alpha^{i}_{(+|n)} = (4\pi a_0^2)\bar{v}_e \left(\frac{I_H}{kT}\right)^2 \psi(\xi_n) \tag{20}$$

where $\xi_n = I_n/kT$ and I_n is the ionization potential from level n. The equilibrium for ionization and recombination (the "Saha" equilibrium) involves a different relation:

$$\left(\frac{x_+ x_e}{x_n}\right)^* \equiv \mathcal{S}_n(T_e) = \frac{g_+}{g_n} \underbrace{2\left(\frac{2\pi m_e k T_e}{h^2}\right)^{\frac{3}{2}}}_{\mathbb{Z}_e} e^{-\xi_n} \tag{21}$$

where g_+ is the degeneracy of the ion ground state (for atomic hydrogen, $g_+ \equiv 2$). The factor indentified as $\mathbb{Z}_e$ is the partition function of the free electrons. We cannot generally assume that the equilibrium values are the same for both bound-bound and bound-free processes. Usually we can have Boltzmann equilibrium (14) without Saha equilibrium, but hardly the reverse, mostly because it takes more energy to ionize than to excite; for the upper states close to the ionization limit ($n \gg 1$), the difference is less significant. Using the principle of detailed balance, the reverse (recombination) rate is:

$$\beta^r_{(n|+)} \simeq \left[\frac{4}{\pi}\frac{a_0^2 h^3}{m_e^2 k T_e}\right]\left(\frac{I_H}{kT_e}\right)^2 n^2 \psi(\xi_n) e^{\xi_n} \tag{22}$$

Finally, the spontaneous emission rates from an upper level m (in sec^{-1}) are:

$$A_{(n|m)} = \left(\frac{8\pi^2 e^2}{m_e c^3}\right)\frac{g_n}{g_m} f_{nm} \tag{23}$$

For hydrogen, this is:

$$A_{(n|m)} = \frac{1.6\ 10^{10}}{m^3 n(m^2 - n^2)} \mathrm{sec}^{-1} \tag{24}$$

4 Properties of the Equations and Numerical Considerations

As shown in (8) the simplified kinetics for a system with N atomic states for a system of ODEs of the form

$$\frac{dX}{dt} = (M_0 + yM_1 + y^2 M_2)X, \qquad y = z^T X, \tag{25}$$

where $X = (x_0, \ldots, x_N) \in \mathbb{R}^{N+1}$ represents the population densities of the various states — x_0 is the density of the ground atomic state, $x_1, \ldots x_{N-1}$ are the densities of excited states and $x_N \equiv x_+$ is the density of ions (we have written $y \equiv x_e$ to express a general conservation property). Moreover, z is the fixed vector $z = (0, \ldots, 0, 1)^T \in \mathbb{R}^{N+1}$ so that $y = x_N$. The matrices $M_j \in \mathbb{R}^{(N+1)\times(N+1)}$ are all fixed with the following properties:

- M_0 is upper triangular. The first and last columns are zero. It has a fast decay off the diagonal and the column sums are all zero.

- M_1 is a full matrix, except that the last column is zero. The submatrix $(1 : N-1, 1 : N-1)$ has fast decay off the diagonal. The column sums are all zero.
- M_2 is a rank one matrix. It is zero, except for the last column, which has sum zero.

See Figure 1 for examples. These properties follow from the physical model described in the previous section. In particular, the zero column sums imply the natural requirement that the total number of electrons is conserved,

$$Q(t) := \sum_{j=0}^{N} x_j(t) = \text{constant}.$$

This is easily seen by left multiplication of (25) by $\mathbf{1} = (1, \ldots, 1)^T \in \mathbb{R}^{N+1}$,

$$\frac{dQ}{dt} = \sum_{j=0}^{N} \frac{dx_j}{dt} = \frac{d}{dt}\mathbf{1}^T \cdot X = \mathbf{1}^T(M_0 + yM_1 + y^2M_2)X = 0,$$

since the column sums $\mathbf{1}^T M_j$ are zero.

4.1 Stiffness and Time Scales

The Jacobian of the right hand side in (25) is given by

$$J(X) = M_0 + yM_1 + y^2M_2 + M_1Xz^T + 2yM_2Xz^T. \tag{26}$$

The eigenvalues of $J(X)$ determine the time scales of the system. Figure 2 shows a typical example of a solution X and the eigenvalues of $J(X)$ as a function of time when $N = 10$. The coarse behavior of the system has the timescale given by the smallest (non-zero) eigenvalue $\lambda_{\text{small}} \approx 10^2$, while the fastest timescales in the system are given by the largest eigenvalues $\lambda_{\text{large}} \approx 10^9$, initially. The large eigenvalues also change over many magnitudes within the relevant computational time. Hence, the system is very stiff, with a stiffness ratio $\lambda_{\text{large}}/\lambda_{\text{small}}$ between 10^6 and 10^9 when $N = 10$. Furthermore, in Figure 3 one can see that λ_{large} grows with N, showing that the stiffness gets worse for larger system sizes N.

4.2 Standard Explicit and Implicit Methods

Standard explicit time-stepping methods cannot be used in these extreme conditions. The stability requirement would demand a time step of the order of $1/\lambda_{\text{large}}$. Since the matrices M_j are full, each step would have a computational cost of $O(N^2)$, giving the total complexity

$$\text{complexity of explicit methods} = O(N^2\lambda_{\text{large}}).$$

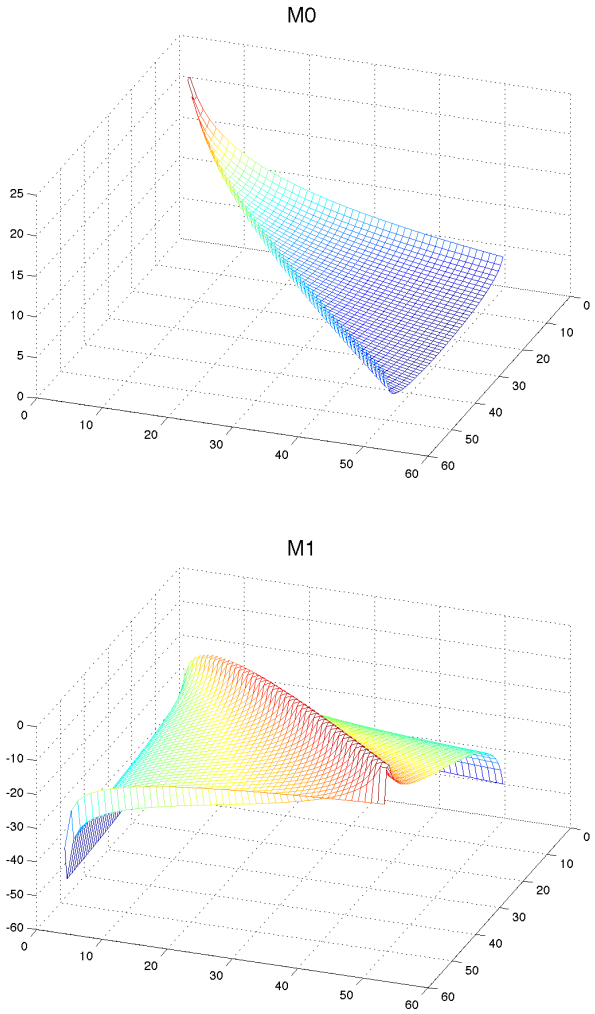
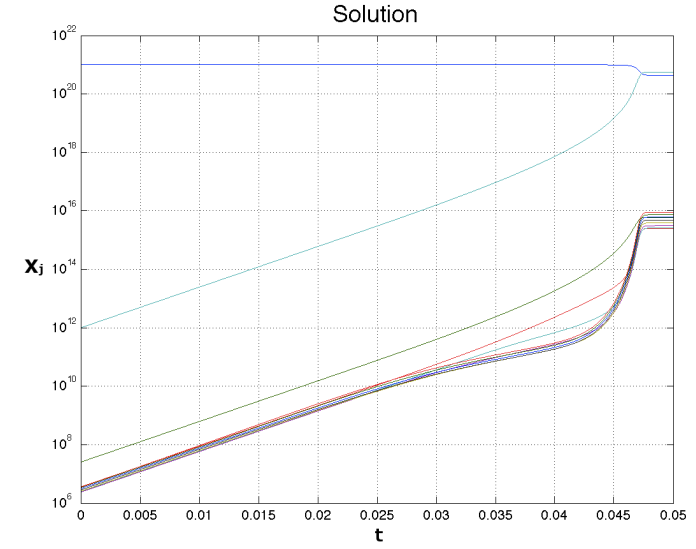
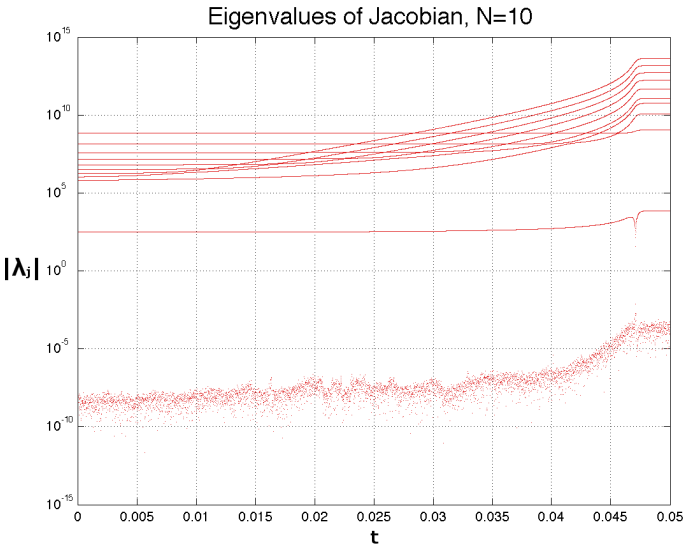


Figure 1: Log plots of element sizes for M_0 and M_1 when $N = 50$.

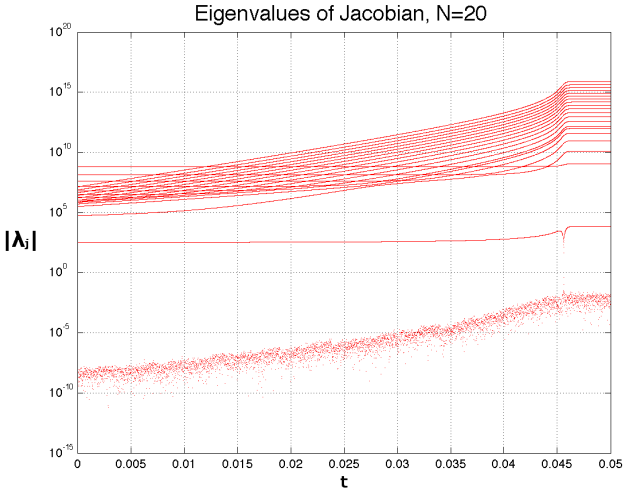


(a)

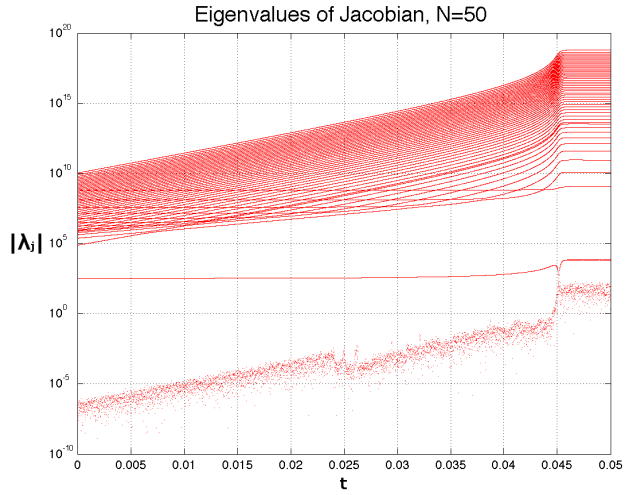


(b)

Figure 2: Rate equations with 10 energy states. a) Population densities as a function of time; b) Eigenvalues λ_j of Jacobian as a function of time. Note the huge span between largest and smallest eigenvalue, and the great changes of magnitudes of the eigenvalues over time. (The noisy data for the smallest eigenvalue is just numerical errors; the exact Jacobian has a zero eigenvalue for all N since its column sum is zero, see below.)



(a)



(b)

Figure 3: Eigenvalues λ_j as a function of time for larger systems: $N = 20$ (a) and $N = 50$ (b).

Implicit methods have no stability requirements and can choose the time step based on accuracy needs only, i.e. proportional to $1/\lambda_{\text{small}}$ such that it just resolves the slow time scale. On the other hand, a nonlinear equation needs to be solved in each time step, which in the end amounts to inverting $I+\alpha J(X)$ for some constant α. This is a full matrix since M_1 is full, giving a cost of $O(N^3)$. Thus

$$\text{complexity of implicit methods} = O(N^3\lambda_{\text{small}}),$$

which is typically much smaller than for explicit methods, but can still be too high because of the cubic dependence on N.

This is thus a particularly difficult type of ODE and solving it fast and at low computational cost presents a great numerical challenge.

4.3 Multiscale Explicit Methods

One approach would be to find non-standard explicit time-stepping methods to solve the rate equations efficiently, based on new multiscale approaches. In the past ten years there has been a renewed interest in developing such explicit methods for stiff ODE problems, for instance Chebyshev methods [1, 7], heterogeneous multiscale methods (HMM) [3], projective integration methods [4, 5], and flow averaging [9]. Although there is a clear gap between the slowest time scale and the next, beyond that there are many fast scales spread out over a large interval. This situation is not handled well by the multiscale methods mentioned above.

5 Implicit Methods with Approximate Jacobian

Implicit methods can be made more efficient if an approximation of the Jacobian $J(X)$ is employed such that $I+\alpha J(X)$ can be inverted rapidly. For this approach we consider the BDF methods, which are standard implicit multistep methods. The first three in this family reads

$$X^{(n+1)} = X^{(n)} + \Delta t F(X^{(n+1)}) \tag{27a}$$

$$X^{(n+1)} = \frac{4}{3}X^{(n)} - \frac{1}{3}X^{(n-1)} + \frac{2}{3}\Delta t F(X^{(n+1)}), \tag{27b}$$

$$X^{(n+1)} = \frac{18}{11}X^{(n)} - \frac{9}{11}X^{(n-1)} + \frac{2}{11}X^{(n-2)} + \frac{6}{11}\Delta t F(X^{(n+1)}), \tag{27c}$$

where $X^{(n)} \approx X(n\Delta t)$.

In order to solve the nonlinear system in each time step one iteration of the Newton method (with initial guess $X^{(n)}$) is employed, which is equivalent to replacing $F(X)$ with a local linearization, namely

$$F(X^{(n+1)}) \quad \Rightarrow \quad F(X^{(n)}) + J(X^{(n)})(X^{(n+1)} - X^{(n)}). \tag{28}$$

This reduces the problem to solving a linear system in each time step, more precisely

$$[I - \Delta t J^{(n)}]X^{(n+1)} = X^{(n)} + \Delta t[F(X^{(n)}) - J^{(n)}X^{(n)}] \tag{29a}$$

$$[I - \frac{2}{3}\Delta t J^{(n)}]X^{(n+1)} = \frac{4}{3}X^{(n)} - \frac{1}{3}X^{(n-1)} + \frac{2}{3}\Delta t[F(X^{(n)}) - J^{(n)}X^{(n)}] \tag{29b}$$

$$[I - \frac{6}{11}\Delta t J^{(n)}]X^{(n+1)} = \frac{18}{11}X^{(n)} - \frac{9}{11}X^{(n-1)} + \frac{2}{11}X^{(n-2)} + \frac{6}{11}\Delta t[F(X^{(n)}) - J^{(n)}X^{(n)}] \tag{29c}$$

The main cost in each iteration is thus the $O(N^3)$ cost to invert $I + \alpha J(X^{(n)})$. The next largest cost is the evaluation of $F(X^{(n)})$ and $J(X^{(n)}) \cdot X^{(n)}$ which are both $O(N^2)$ operations since M_0 and M_1 are full matrices. The idea here is to approximate $J(X) \approx \tilde{J}(X)$ such that $I + \alpha\tilde{J}(X)$ can be inverted with at most $O(N^2)$ operations. That would give

$$\text{complexity of implicit method with approximate } J(X) = O(N^2\lambda_{\text{small}}),$$

which is of course much faster than standard implicit methods. We will now discuss different strategies for this.

Remark: To be more accurate we can make several iterations in the Newton method. Then the above equations are solved multiple time. For instance, if we use K iterations in the BDF1 method we would have

$$[I-\Delta t J^{(n)}]Y_{k+1} = X^{(n)}+\Delta t[F(Y_k)-J(Y_k)Y_k], \qquad Y_0 = X^{(n)}, \qquad X^{(n+1)} = Y_K.$$

5.1 Sherman–Morrison Formula for the Tail of $J(X)$

We split the expression for the Jacobian as

$$J(X) = M_0 + yM_1 + R(X), \qquad R(X) = y^2M_2 + M_1Xz^T + 2yM_2Xz^T, \tag{30}$$

We can write $M_2 = mz^T$ with $m \in \mathbb{R}^{N+1}$ being the last column of M_2. Therefore,

$$R(X) = (3y^2m + M_1X)z^T := \tilde{m}(X)z^T, \tag{31}$$

is a rank one matrix. We now note that for any A we can invert $A + R$ at the same cost as A by using the Sherman–Morrison formula,

$$(A + uv^T)^{-1} = A^{-1} - \frac{A^{-1}uv^TA^{-1}}{1 + v^TA^{-1}u}. \tag{32}$$

Indeed, in order to solve

$$(A + R)X = b,$$

we could first solve $As = \tilde{m}$ and $Aw = b$. Then, by (32), the solution X is given as

$$x = (A+R)^{-1}b = w - \frac{z^T w}{1+z^T s}s = w - \frac{w_N}{1+s_N}s,$$

which is just an $O(N)$ cost once w and s have been computed. The conclusion is that we only need to find a fast, approximate, way to invert the leading part of $I + \alpha J$, namely $I + \alpha(M_0 + yM_1)$.

5.2 Direct Truncation of M_0, M_1

Since both M_0 and M_1 has a fast decay off the diagonal, a natural approach to speed up their inversion would be to truncate the matrices to banded form, with a small bandwidth, i.e. setting most of the matrices to zero, only leaving a few non-zero diagonals untouched. We introduce the direct truncation operator

$$\tilde{A} = \text{trunc}(A, p) \quad \Rightarrow \quad \tilde{A}_{ij} = \begin{cases} A_{ij}, & |i-j| < p, \\ 0, & |i-j| \geq p. \end{cases}$$

Then we let $\tilde{M}_j = \text{trunc}(M_j, p)$ and approximate

$$M_0 + yM_1 \approx \tilde{M}_0 + y\tilde{M}_1.$$

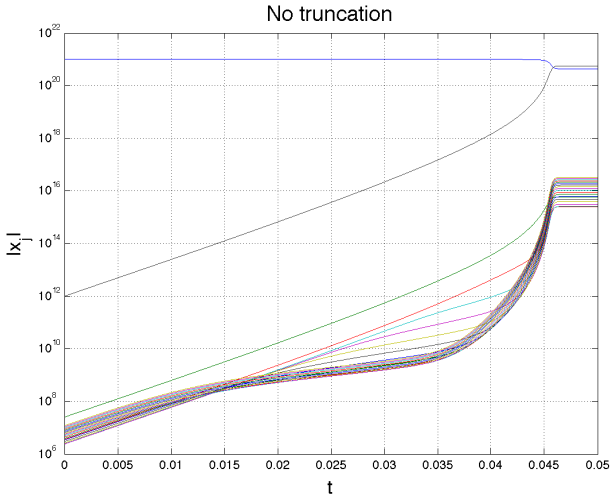
Since $I+\alpha(\tilde{M}_0+y\tilde{M}_j)$ is a banded matrix with bandwidth p the cost of inverting it is $O(Np^2)$. Hence, we could allow ourselves to take $p \sim \sqrt{N}$ to have a cost of $O(N^2)$ as we needed.

Unfortunately, the direct truncation method yields very poor results. A typical example of a solution with directly truncated M_j is shown in Figure 4. The sharp transition to equilibrium at around $t = 0.045$ is smeared out and the equilibrium is not reached until much later. A more quantitative study of the errors is made in Figures 5 and 6. Here the parameter K represents the number of iterations in the Newton method when solving the nonlinear systems of equations in each BDF time step. The error reported in these, and subsequent, plots is the maximum componentwise relative error between the solution with and without truncation,

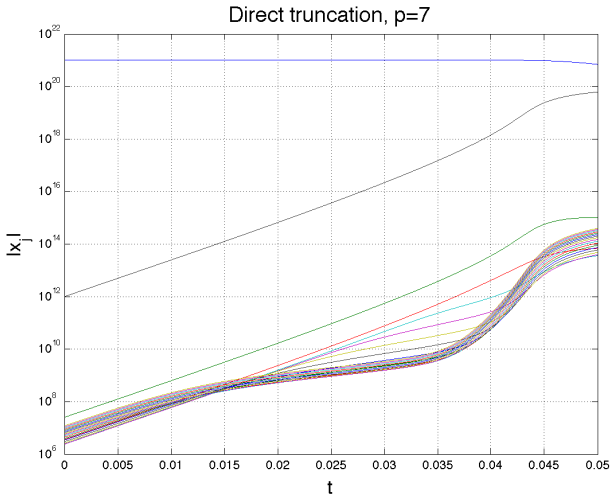
$$\text{error} = \max_j \frac{|x_j - \tilde{x}_j|}{|x_j|}.$$

The final time is selected inside the transition region to equilibirum, $T = 0.045$. One can see that the methods remains stable also for severe truncations (small p) but the accuracy is not good. To come below 10% error more than 15 of the 20 diagonals must be kept in BDF1 with $\Delta t = 10^{-4}$, for example. For $p < 10$ the relative error is close to one for all method. One can note though that the error is reduced if more iterations (larger K) or smaller time steps Δt are used.

The underlying reason for the bad results with direct truncation has to do with the conservation properties of the discrete approximation, which we dicsuss next.



(a)



(b)

Figure 4: Solution example with (a) and without (b) direct trunction. Parameters used were $N = 20$, $\Delta t = 10^{-5}$ and $p = 7$.

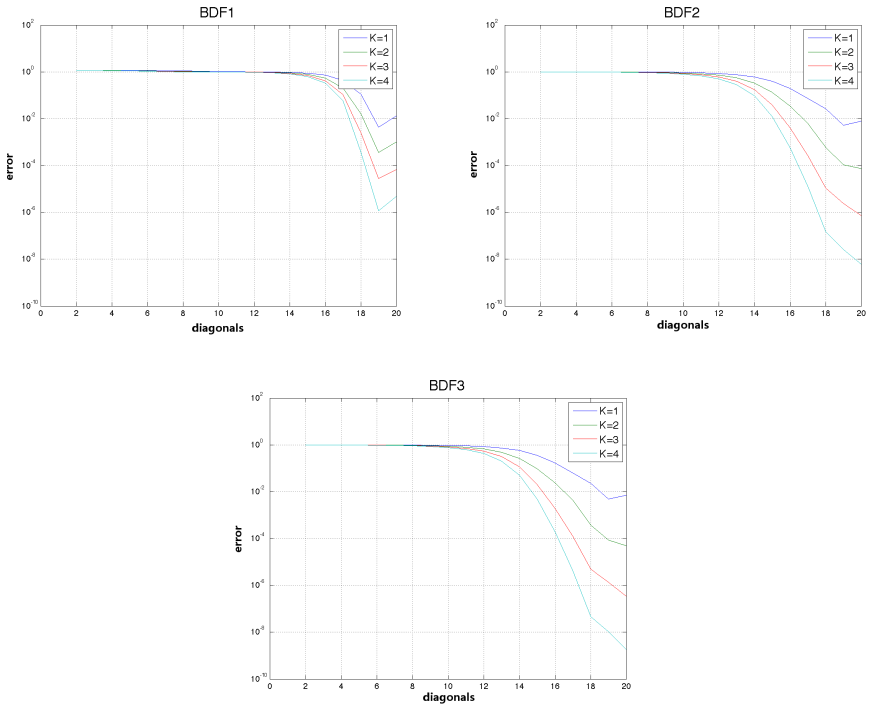


Figure 5: Error with direct truncation as a function of the number of diagonals p and Newton iterations K for the BDF methods. Parameters used were $N = 20$, $\Delta t = 10^{-4}$ and final time $T = 0.045$.

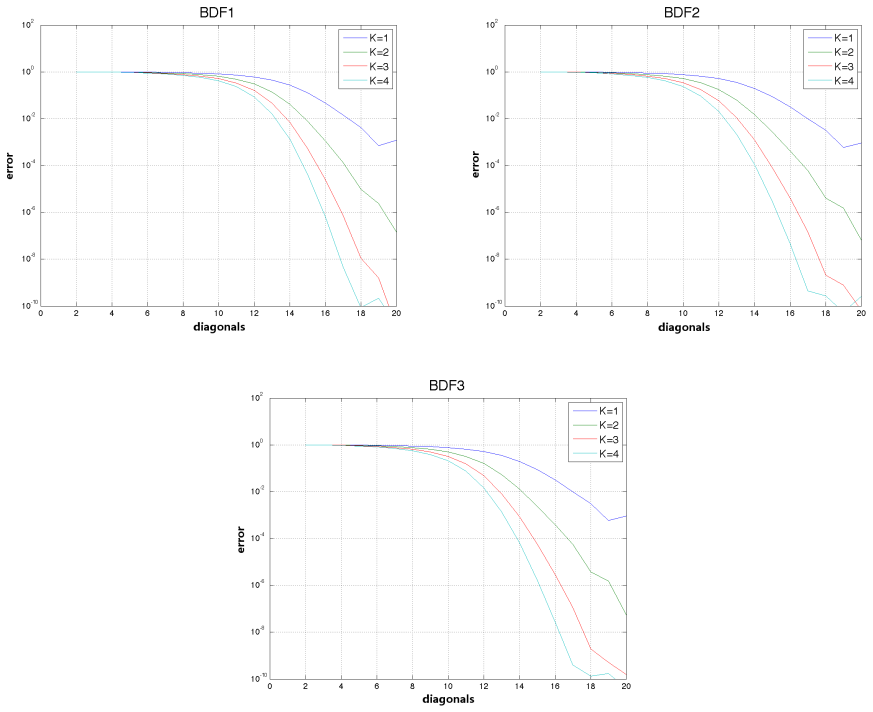


Figure 6: Error with direct truncation as a function of the number of diagonals p and Newton iterations K for the BDF methods. Parameters used were $N = 20$, $\Delta t = 10^{-5}$ and final time $T = 0.045$.

5.3 Discrete Conservation

As mentioned above, in the exact ODE solution the total number of electrons is conserved. In the discrete case, we define

$$Q^{(n)} := \mathbf{1}^T X^{(n)}.$$

Then, $Q^{(n)}$ is constant also for the BDF methods. For example, in BDF3,

$$\begin{aligned} Q^{(n+1)} &= \mathbf{1}^T X^{(n+1)} \\ &= \frac{18}{11}\mathbf{1}^T X^{(n)} - \frac{9}{11}\mathbf{1}^T X^{(n-1)} + \frac{2}{11}\mathbf{1}^T X^{(n-2)} + \frac{6}{11}\Delta t \mathbf{1}^T F(X^{(n+1)}) \\ &= \frac{18}{11}Q^{(n)} - \frac{9}{11}Q^{(n-1)} + \frac{2}{11}Q^{(n-2)}, \end{aligned}$$

which is a difference equation with the solution $Q^{(n)} = Q^{(0)}$ for all $n > 1$ if initial data is taken such that $Q^{(-2)} = Q^{(-1)} = Q^{(0)}$. Morever, exact discrete conservation holds also for the approximate BDF methods where F is linearized. This follows in a similar way upon noting that

$$\mathbf{1}^T J(X) = \mathbf{1}^T (M_0 + yM_1 + y^2 M_2 + M_1 X z^T + 2yM_2 X z^T) = 0,$$

again since $\mathbf{1}^T M_j = 0$. Then, again

$$\begin{aligned} Q^{(n+1)} &= \mathbf{1}^T [I - \frac{6}{11}\Delta t J^{(n)}] X^{(n+1)} \\ &= \frac{18}{11}\mathbf{1}^T X^{(n)} - \frac{9}{11}\mathbf{1}^T X^{(n-1)} + \frac{2}{11}\mathbf{1}^T X^{(n-2)} + \frac{6}{11}\Delta t \mathbf{1}^T [F(X^{(n)}) - J^{(n)} X^{(n)}] \\ &= \frac{18}{11}Q^{(n)} - \frac{9}{11}Q^{(n-1)} + \frac{2}{11}Q^{(n-2)}. \end{aligned}$$

(Note that this holds true also if multiple iterations are used, $K > 1$, since it holds for each iteration.)

However, if we directly truncate M_0 and M_1 as in the previous section they no longer have exact column sum zero, $\mathbf{1}^T \tilde{M}_j \neq 0$ and then neither has the approximate Jacobian. There is no longer exact discrete conservation, which for this problem leads to bad performance of the numerical method.

5.4 Weighted Truncation of M_0, M_1

Given the discussion about conservation in the previous section, the methods would be improved if the truncation is done such that the column sums of the matrices are unaffected. This can for instance be done by re-weighting the off-diagonal elements as follows

$$\tilde{A} = \mathrm{trunc}_w(A, p) \quad \Rightarrow \quad \tilde{A}_{ij} = \begin{cases} A_{ii}, & i = j, \\ w_j A_{ij}, & 0 < |i-j| < p, \\ 0, & |i-j| \geq p, \end{cases}$$

where the weights are given by

$$w_j = \frac{A_{ii} - \sum_{i=0}^{N} A_{ij}}{A_{ii} - \sum_{|i-j|<p}^{N} A_{ij}}.$$

It is easy to check that $\mathbf{1}^T A = \mathbf{1}^T \tilde{A}$. We use this truncation on the full matrix[1]

$$\tilde{M} = \mathrm{trunc}_w(M_0 + yM_1, p).$$

Thus, with this truncation strategy we enforce discrete conservation and we expect a better behavior of the numerical solution. Indeed, much better results are obtained. The corresponding result with weighted truncation for the example in Figure 4 is shown in Figure 7. Here, the transition to equilibrium is captured without problems. The quantitative study corresponding to Figures 5 and 6 is shown in Figures 8 and 9. Small errors can be obtained also with moderately sized p. For example in BDF3 with $K = 1$ and $\Delta t = 10^{-5}$ one can truncate to just 3 diagonals and still obtain a relative error close to 1%. On the other hand, the methods become unstable (huge relative error) for small enough p or large enough Δt. This seems to be different from the direct truncation. However, the conclusions made for direct truncation about the influence of K and Δt hold true also for weighted truncation: larger K and smaller Δt give smaller error.

6 Implicit Methods with Precomputation

As was shown in the previous section the main cost of a standard implicit method is to invert the matrix $I + \alpha J$ in each iteration. Via the Sherman–Morrison formula the work is reduced to inverting the leading order matrix $I + \alpha(M_0 + yM_1)$. Instead of using an approximation $\tilde{M}_0 + y\tilde{M}_1$ to lessen this cost, one can precompute a factorization of the matrix. The usual LU decomposition does not help, however, since the parameter y will be different in every time step. Instead we use the generalized Schur factorization which provides a simultaneous decomposition of $I + \alpha M_0$ and αM_1.

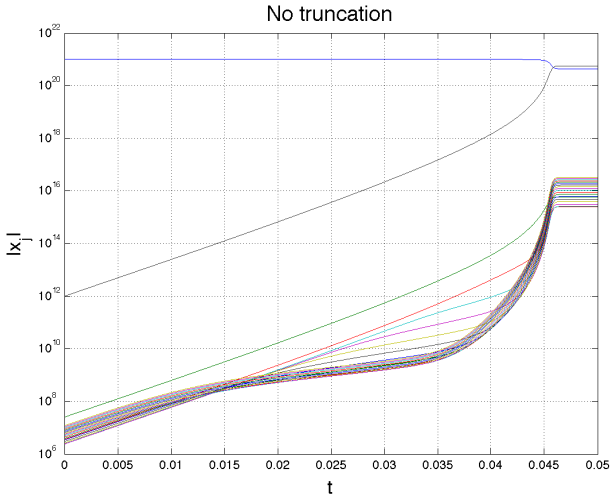
The main theoretical result behind this approach is the generalized Schur decomposition theorem saying that for any two $N \times N$ matrices A and B there are two orthogonal matrices Q and Z such that both QAZ and QBZ are upper triangular. There is also a stable algorithm to compute these matrices: the QZ-algorithm [6]. In our case, we find the orthogonal matrices related to $I + \alpha M_0$ and αM_1 such that

$$Q(I + \alpha M_0)Z = m_0, \qquad \alpha Q M_1 Z = m_1,$$

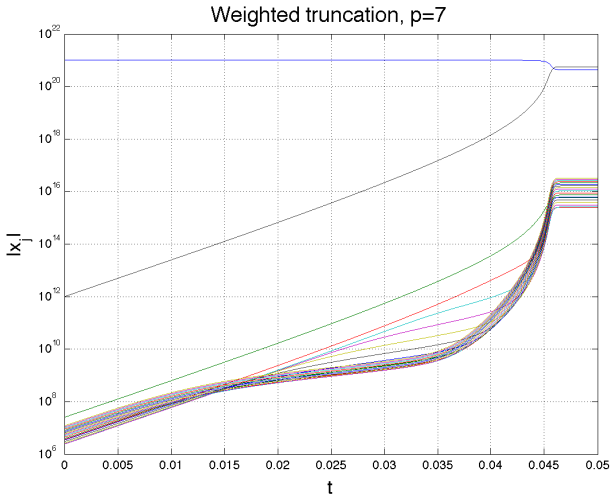
where both m_0 and m_1 are upper triangular. Then, to solve

$$(I + \alpha(M_0 + yM_1))x = b,$$

[1] Since the truncation operation is not linear in this case, the result is different from truncating M_0 and M_1 individually.



(a)



(b)

Figure 7: Solution example with (a) and without (b) weighted trunction. Parameters used were $N = 20$, $\Delta t = 10^{-5}$ and $p = 7$.

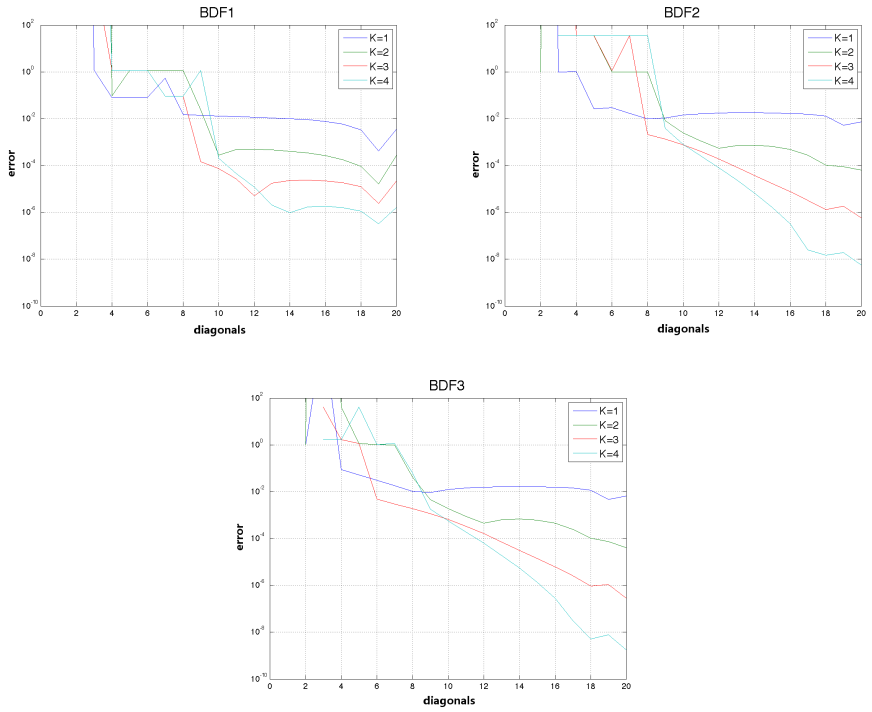


Figure 8: Error with weighted truncation as a function of the number of diagonals p and Newton iterations K for the BDF methods. Parameters used were $N = 20$, $\Delta t = 10^{-4}$ and final time $T = 0.045$.

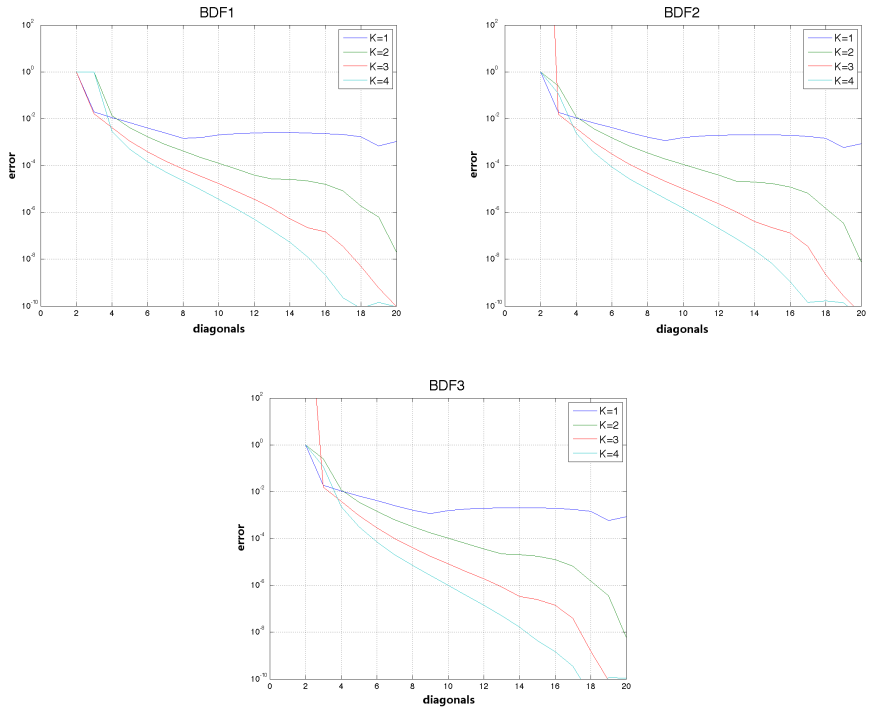


Figure 9: Error with weighted truncation as a function of the number of diagonals p and Newton iterations K for the BDF methods. Parameters used were $N = 20$, $\Delta t = 10^{-5}$ and final time $T = 0.045$.

we multiply by Q and Z from the left and right, respectively, to get

$$Q(I + \alpha(M_0 + yM_1))ZZ^*x = Qb \quad \Rightarrow \quad (I + \alpha(m_0 + ym_1))(Z^*x) = Qb.$$

Hence, x is obtained from

$$(I + \alpha(m_0 + ym_1))\tilde{x} = Qb, \qquad x = Z\tilde{x},$$

where the system matrix $I + \alpha(m_0 + ym_1)$ is now upper triangular. The cost of computing x is thus given by the cost of two matrix multiplications by orthogonal matrices and one linear solve with an upper triangular matrix. These all have costs of the order N^2 operations. The QZ-algorithm itself costs $O(N^3)$. The total complexity of this approach is thus

$$\text{complexity of implicit method with precomputation} = O(N^3 + N^2\lambda_{\text{small}}).$$

Thus when $N \leq \lambda_{\text{small}}$ or when very high accuracy is needed (small Δt) this is as fast as the methods based on approximate Jacobians. Since there is no approximation involved, the precomputation method is, however, more robust and accurate.

6.1 Solution Steps

The steps to evaluate X given by

$$(I - \Delta t J(\tilde{X}))X = b,$$

in the BDF methods (29) can be summarized as follows.

- Compute the generalized Schur factorization of $I+\alpha M_0$ and αM_1 to obtain m_0, m_1, Q and Z.
- Let $y = \tilde{X}_N$ and compute

  $$\tilde{m} = 3y^2 m + M_1\tilde{X},$$

  according to (31). (Here m is the last column of M_2.)
- Solve

  $$(I + \alpha(m_0 + ym_1))\tilde{s} = Q\tilde{m}, \qquad (I + \alpha(m_0 + ym_1))\tilde{w} = Qb,$$

- Let $s = Z\tilde{s}$ and $w = Z\tilde{w}$.
- The solution is given by

  $$X = w - \frac{w_N}{1 + s_N}s.$$

7 One Step Methods

In the multistep methods discussed above, initial data must be given for several time steps, which is inconvenient for the application where the rate equations are used; they are coupled with a CFD solver where only one data point is given each time they need to be solved. The simplest way to get around this problem would be to use a *one-step* method, which only requires one initial data. We thus investigated the use of implicit Runge–Kutta (IRK) methods, focusing on 2-stage methods.

For an autonomous ODE $X' = F(X)$ the update step from $X^{(n)} \to X^{(n+1)}$ in a 2-stage IRK is defined by the parameters $a_{k,\ell}$ and b_k as follows. First, find ξ_1, ξ_2 such that

$$\xi_1 = F\left(X^{(n)} + \Delta t[a_{11}\xi_1 + a_{12}\xi_2]\right), \tag{33a}$$

$$\xi_2 = F\left(X^{(n)} + \Delta t[a_{21}\xi_1 + a_{22}\xi_2]\right), \tag{33b}$$

and, second, compute

$$X^{(n+1)} = X^{(n)} + \Delta t(b_1\xi_1 + b_2\xi_2). \tag{34}$$

Note that since ξ_1, ξ_2 appear in both the left and right hand side of (33) the methods are implicit. As in the multistep case we therefore need to solve this equation by Newton's method with zero as initial guess. Upon replacing F by its linearized version as in (28) we get

$$\begin{pmatrix}\xi_1\\ \xi_2\end{pmatrix} = \begin{pmatrix}F(X^{(n)}) + \Delta t J(X^{(n)})[a_{11}\xi_1 + a_{12}\xi_2]\\ F(X^{(n)}) + \Delta t J(X^{(n)})[a_{21}\xi_1 + a_{22}\xi_2]\end{pmatrix},$$

which leads to the linear system for ξ_1 and ξ_2,

$$\left[I - \Delta t\begin{pmatrix}a_{11}J(X^{(n)}) & a_{12}J(X^{(n)})\\ a_{21}J(X^{(n)}) & a_{22}J(X^{(n)})\end{pmatrix}\right]\begin{pmatrix}\xi_1\\ \xi_2\end{pmatrix} = \begin{pmatrix}F(X^{(n)})\\ F(X^{(n)})\end{pmatrix}. \tag{35}$$

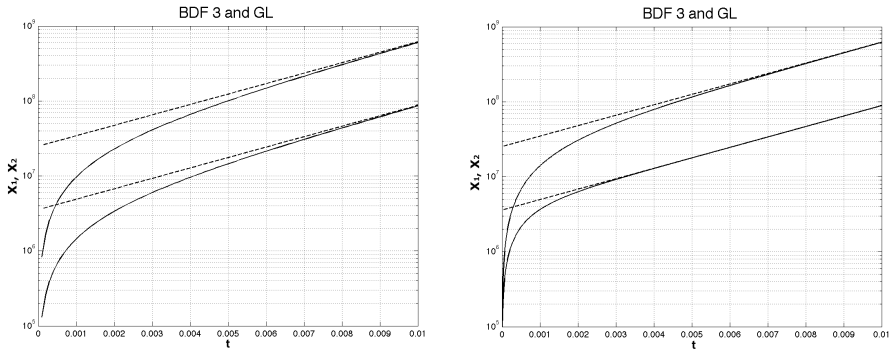
This system must be solved in each time step. The solution is then used to update $X^{(n)}$ as in (34).

We first tested the classical fourth order *Gauss-Legendre* (GL4) IRK method, in which $b_1 = b_2 = 1/2$ and

$$a_{11} = a_{22} = \frac{1}{4}, \qquad a_{12} = \frac{1}{4} - \frac{\sqrt{3}}{6}, \qquad a_{21} = \frac{1}{4} + \frac{\sqrt{3}}{6}.$$

The results were not satisfactory, however. The GL4 method is very accurate when the solution is on the slow manifold, but it is bad at damping the fast, unresolved, modes compared to the BDF methods. This is precisely the typical situation at the initial part of the computation. An example of this problem is shown in Figure 10.

The underlying reason for the poor performance turned out the be that GL4 is not an *L-stable* method, i.e. its stability function does not decay to zero at



(a) $\Delta t = 10^{-4}$ (b) $\Delta t = 10^{-5}$

Figure 10: Initial time evolution of two typical population densities x_1, x_2 in an $N = 20$ system. Initial data is zero for these densities, i.e. far off the slow manifold. The solution using BDF3 (solid lines) is satisfactory for both the large and small time-step; it reaches the slow manifold more or less immediately, as it should. (The true relaxation time is ca 10^{-6}.) The solution using GL4 (dashed lines), on the other hand, takes very long time to reach the slow manifold, even with the small time-step. The solution with Radau Ia cannot be distinguished from the BDF3 solution.

infinity. Instead we used the L-stable third order *Radau Ia* IRK method, for which $b_1 = 1/4$, $b_2 = 3/4$ and

$$a_{11} = a_{21} = \frac{1}{4}, \qquad a_{12} = -\frac{1}{4}, \qquad a_{22} = \frac{5}{12}.$$

In this case we obtained very good results. The solution plots corresponding to those in Figure 10 coincide perfectly with the ones for BDF3.

7.1 Properties of the Linear System of Equations

Using Kronecker notation the system matrix in (35) can be written

$$I - \Delta t A \otimes J(X^{(n)}), \qquad A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix},$$

and with the expresssions in (30) and (31) we get

$$I - \Delta t A \otimes J(X^{(n)}) = I - \Delta t A \otimes M_0 - y \Delta t A \otimes M_1 - \Delta t A \otimes (\tilde{m} z^T),$$

where

$$A \otimes (\tilde{m} z^T) = \begin{pmatrix} a_{11}\tilde{m} \\ a_{21}\tilde{m} \end{pmatrix} \begin{pmatrix} z \\ 0 \end{pmatrix}^T + \begin{pmatrix} a_{12}\tilde{m} \\ a_{22}\tilde{m} \end{pmatrix} \begin{pmatrix} 0 \\ z \end{pmatrix}^T.$$

This shows that the matrix is again of the form

$$I + \alpha(\mathcal{M}_0 + y\mathcal{M}_1) + \mathcal{R},$$

where

$$\mathcal{M}_0 = A \otimes M_0, \qquad \mathcal{M}_0 = A \otimes M_1.$$

and $\mathcal{R}$ is now of rank two (instead of one). To treat $\mathcal{R}$ we therefore need to apply the Sherman–Morrison formula twice, iteratively. The same kind of precomputation as for multistep method can be used for the leading part $I + \alpha(\mathcal{M}_0 + y\mathcal{M}_1)$.

It should be noted, however, that the matrix form is less favorable when more iterations in the Newton method is used, i.e. when $K > 1$. Then the Jacobian of F is evaluated at two different points, leading to a system matrix with two parameters y_1 and y_2. The QZ approach cannot be used on this system.

Moreover, the strategy to approximate the Jacobian using truncation gives a different sparsity pattern. Instead of a banded matrix, we get a 2×2 block structured matrix where each block is banded. Solving this fast is not as straightforward. In the tests that we have performed we have also observed that, accuracy wise, the one-step Runge–Kutta methods are in general more sensitive to truncation than the multistep BDF methods.

8 Conclusions and Future Work

We have explored implicit methods to solve the stiff rate equations. To make the methods efficient we use an approximate Jacobian or a precomputed factorization of the Jacobian.

The Jacobian is strongly diagonally dominant and can be well approximated by a banded matrix. Implicit methods then becomes relatively inexpensive. However, the results are sensitive to precisely how the approximation is done. We have found that it is of vital importance to approximate the Jacobian with a method which maintains the exact discrete conservation of the solver. By using a weighted truncation to banded form we achieve this and obtain accurate results at a low computational cost.

In many cases the generalized Schur factorization of the Jacobian can be used to speed up the time stepping of the ODEs. This is a more robust and accurate approach than approximating the Jacobian. It is preferable if the matrix structure of the Jacobian allows it, and if the accuracy requirements in the ODE solver is high enough so that the prefactorization cost is small compared to the time stepping costs.

One step methods are simpler to initialize than multistep methods, which is an advantage when the ODEs are coupled to the flow solver. Because of the extreme stiffness of the system it is important to use L-stable one-step methods, such as the Radau Ia implicit Runge–Kutta method. The linear system of equations that must be solved in each time step involves the same Jacobian as for multistep method, but it has in general a blocked matrix structure, which means that the approximation and prefactorization strategies above must be adapted.

The main directions for future work are:

- Better understanding of the errors in the weighted truncation. How should the number of diagonals and iterations in the Newton solver be chosen to make the methods robust and accurate? Some adaptive approach may be necessary. Here one could get ideas from *dynamic sparsing* techniques for stiff ODEs [8] or methods for finding sparse preconditioners in PDE problems [2]. In these methods a sparse approximation of the Jacobian is found dynamically based on stability criteria for the ODE or spectral properties of the matrices.

- Choice of implicit Runge–Kutta method. There are many options for L-stable Runge–Kutta methods (e.g. the Rosenbrock family). These should be explored and tested for accuracy, robustness and speed. The resulting matrix structure for the linear system of equation is of particular concern. *Diagonal* implicit Runge–Kutta methods (DIRK) could be an attractive alternative, since the matrix structure is then block triangular.

- More complex and larger systems of rate equations with multiple species. This leads not only to larger ODE systems, but also to different matrix structures, for which our truncation and precomputation strategies must be adapted.

- Dealing with extreme stiffness. For larger systems ($N \approx 100$) the stiffness ratio often exceeds 10^{15} which means that matrix condition numbers are of the same order. Double precision is then not sufficient to obtain any useful accuracy. Some alternative modeling may be necessary for these systems, for instance approximating them by differential algebraic equations (DAE).

- Splitting methods. The complexity could potentially also be reduced by using splitting methods for M_0, M_1 and M_2. This would mean setting $\tilde{M}(x) = yM_1 + y^2M_2$ and do a time-stepping of the type

$$\begin{aligned}
\tilde{X}^{(n)} &= X^{(n)} + \frac{\Delta t}{2}\tilde{M}(\tilde{X}^{(n)})\tilde{X}^{(n)}, \\
\tilde{\tilde{X}}^{(n)} &= \tilde{X}^{(n)} + \Delta t M_0 \tilde{\tilde{X}}^{(n)}, \\
X^{(n+1)} &= \tilde{\tilde{X}}^{(n)} + \frac{\Delta t}{2}\tilde{M}(X^{(n+1)})X^{(n+1)}.
\end{aligned}$$

  This approach should be investigated. It may e.g. be revealing to study the linear problem here, where $\tilde{M}(x) := \tilde{M}(x_0)$.

References

[1] A. Abdulle, Fourth order Chebyshev methods with recurrence relation, *SIAM J. Sci. Comput.*, Vol. 23, No. 6, pp. 2041-2054, 2002.

[2] T. M. Austin, M. Brezina, B. Jamroz, C. Jhurani, T. A. Manteuffel, J. Ruge, Semi-automatic sparse preconditioners for high-order finite element methods on non-uniform meshes. *J. Comput. Phys.*,231 (14) (2012) 4694 ? 4708.

[3] W. E, B. Engquist, The Heterogenous multiscale methods, *Commun. Math. Sci.*, 1 (2003), 87-132.

[4] K. Eriksson, C. Johnson, A. Logg, Explicit time-stepping for stiff ODEs, *SIAM J. Sci. Comput.* 25 (2003) pp. 1142-1157.

[5] C. W. Gear, I. G. Kevrekidis, Projective methods for stiff differential equations: problems with gaps in their eigenvalue spectrum. *SIAM J. Sci. Comp.* 24(4):109-110 (2003).

[6] G. Golub, C. Van Loan, Matrix computations. Johns Hopkins University Press, 1996.

[7] V. Lebedev, How to solve stiff systems of differential equations by explicit methods, in Numerical methods and applications, Ed. G.I. Marchuk, CRC Press, Boca Raton, Ann Arbor, London, Tokyo (1994), 45-80.

[8] U. Nowak, Dynamic Sparsing in stiff extrapolation methods, Konrad-Zuse-Zentrum report SC 92–21, 1992.

[9] H. Owhadi, J. E. Marsden, M. Tao, Non-intrusive and structure preserving multiscale integration of stiff ODEs, SDEs and Hamiltonian systems with hidden slow dynamics via flow averaging. *SIAM Multiscale Model. Simul.* 8 (2010), pp. 1269–1324.

[10] Ya. B. Zel'dovich and Yu. P. Raizer, *Physics of shock waves and high temperature hydrodynamic phenomena*, 3rd edition, Dover, New-York (2002).