Test Set for Initial Value Problem Solvers
release 2.4

Francesca Mazzia, Cecilia Magherini
Report 4/2008

Department of Mathematics
University of Bari
ITALY

via E.Orabona 4
I-70125 Bari (ITALY)
Test Set for Initial Value Problem Solvers

Francesca Mazzia*, Cecilia Magherini**

* Dipartimento di Matematica, Università degli studi di Bari, Italy
** Dipartimento di Matematica Applicata, Università di Pisa
(mazzia@dm.uniba.it, cecilia.magherini@ing.unipi.it)

Release 2.4 February 2008

Abstract

The test set for IVP solvers presents a collection of Initial Value Problems to test solvers for implicit differential equations. This test set can both decrease the effort for the code developer to test his software in a reliable way, and cross the bridge between the application field and numerical mathematics, by helping people working in several branches of scientific disciplines in choosing the code most suitable for their problems. This document contains the descriptive part of the test set. It describes the solvers used in the comparisons, the test problems and their origin, and reports on the behavior of the solvers on these problems. The latest version of this document and the software part of the test set is available via the world wide web at http://www.dm.uniba.it/~testset.

The software part serves as a platform on which one can test the performance of a solver on a particular test problem oneself. Instructions how to use this software are in this paper as well.

The idea to develop this test set was discussed at the workshop ODE to NODE, held in Geiranger, Norway, 19--22 June 1995 and was developed by the CWI group. After the workshop ANODE01, held in Auckland, New Zealand, 2001, the testset moved to the University of Bari.

Keywords and Phrases: test problems, software, IVP, IDE, ODE, DAE.

Acknowledgements: This work has been supported by the project "Metodi numerici e software per equazioni differenziali ordinarie: problemi a valori iniziali ed al contorno", sponsored by the University of Bari (2006-2008) and by the PRIN 2004-2006 project "Metodi numerici e software matematico per le applicazioni". We thanks the GNCS-INDAM that supported financially the project form 2001 to 2005 and made possible the use of supercomputer facilities in 2002. We
thanks the CWI group, that formerly maintained the testset, for providing useful software. We thank all contributors to this test set, without whom it would not be possible to collect problems from such a wide variety of application fields.
Contents

I. Introduction

II. Format of problem descriptions

III. Format of solver descriptions

IV. The software part of the test set

Part I - SOLVERS

<table>
<thead>
<tr>
<th>Name</th>
<th>problems type</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>BIMD</td>
<td>ODEs and DAE (index ≤ 3)</td>
<td>I-1-1</td>
</tr>
<tr>
<td>DASSL</td>
<td>ODEs and IDEs/DAE (index ≤ 1)</td>
<td>I-2-1</td>
</tr>
<tr>
<td>GAMD</td>
<td>ODE and DAE (index ≤ 3)</td>
<td>I-3-1</td>
</tr>
<tr>
<td>MEBDFDAE</td>
<td>ODE and DAE (index ≤ 3)</td>
<td>I-4-1</td>
</tr>
<tr>
<td>MEBDFI</td>
<td>ODEs and IDEs/DAE (index ≤ 3)</td>
<td>I-5-1</td>
</tr>
<tr>
<td>PSIDE</td>
<td>ODEs and IDEs/DAE (index upto 3)</td>
<td>I-6-1</td>
</tr>
<tr>
<td>RADAU</td>
<td>ODE and DAE (index ≤ 3)</td>
<td>I-7-1</td>
</tr>
<tr>
<td>RADAU5</td>
<td>ODE and DAE (index ≤ 3)</td>
<td>I-8-1</td>
</tr>
<tr>
<td>VODE</td>
<td>ODE</td>
<td>I-9-1</td>
</tr>
</tbody>
</table>
## Part II - Problems

### ODE Test Problems

<table>
<thead>
<tr>
<th>Name</th>
<th>Dimension</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Problem HIRES</td>
<td>8</td>
<td>II-1-1</td>
</tr>
<tr>
<td>Pollution problem</td>
<td>20</td>
<td>II-2-1</td>
</tr>
<tr>
<td>Ring Modulator</td>
<td>15</td>
<td>II-3-1</td>
</tr>
<tr>
<td>Medical Akzo Nobel problem</td>
<td>400</td>
<td>II-4-1</td>
</tr>
<tr>
<td>EMEP problem</td>
<td>66</td>
<td>II-5-1</td>
</tr>
<tr>
<td>Pleiades problem</td>
<td>28</td>
<td>II-6-1</td>
</tr>
<tr>
<td>Beam</td>
<td>80</td>
<td>II-7-1</td>
</tr>
<tr>
<td>Van der Pol</td>
<td>2</td>
<td>II-8-1</td>
</tr>
<tr>
<td>Oregonator</td>
<td>3</td>
<td>II-9-1</td>
</tr>
<tr>
<td>Robertson</td>
<td>3</td>
<td>II-10-1</td>
</tr>
<tr>
<td>E5</td>
<td>4</td>
<td>II-11-1</td>
</tr>
</tbody>
</table>

### DAE Test Problems

<table>
<thead>
<tr>
<th>Name</th>
<th>Dimension</th>
<th>Index</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chemical Akzo Nobel problem</td>
<td>6</td>
<td>1</td>
<td>II-12-1</td>
</tr>
<tr>
<td>Andrews’ squeezing mechanism</td>
<td>27</td>
<td>3</td>
<td>II-13-1</td>
</tr>
<tr>
<td>Transistor amplifier</td>
<td>8</td>
<td>1</td>
<td>II-14-1</td>
</tr>
<tr>
<td>Charge pump</td>
<td>9</td>
<td>2</td>
<td>II-15-1</td>
</tr>
<tr>
<td>Two bit adding unit</td>
<td>350</td>
<td>1</td>
<td>II-16-1</td>
</tr>
<tr>
<td>Car axis problem</td>
<td>10</td>
<td>3</td>
<td>II-17-1</td>
</tr>
<tr>
<td>Fekete problem</td>
<td>160</td>
<td>2</td>
<td>II-18-1</td>
</tr>
<tr>
<td>Slider crank</td>
<td>24</td>
<td>2</td>
<td>II-19-1</td>
</tr>
<tr>
<td>Water tube system</td>
<td>49</td>
<td>2</td>
<td>II-20-1</td>
</tr>
</tbody>
</table>

### IDE Test Problems

<table>
<thead>
<tr>
<th>Name</th>
<th>Dimension</th>
<th>Index</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAND gate</td>
<td>14</td>
<td>1</td>
<td>II-21-1</td>
</tr>
<tr>
<td>Wheelset</td>
<td>17</td>
<td>2</td>
<td>II-22-1</td>
</tr>
</tbody>
</table>
I Introduction

I.1 The idea behind this test set

Both engineers and computational scientists alike will benefit greatly from having a standard test set for Initial Value Problems (IVPs) which includes documentation of the test problems, experimental results from a number of proven solvers, and Fortran subroutines providing a common interface to the defining problem functions. Engineers will be able to see at a glance which methods will be most effective for their class of problems. Researchers will be able to compare their new methods with the results of existing ones without incurring additional programming workload; they will have a reference with which their colleagues are familiar. This test set tries to fulfill these demands and tries to set a standard for IVP solver testing. We hope that the following features of this test set will enable the achievement of this goal:

- uniform presentation of the problems,
- ample description of the origin of the problems,
- robust interfaces between problem and drivers,
- portability among different platforms,
- contributions by people from several application fields,
- presence of real-life problems,
- being used, tested and debugged by a large, international group of researchers,
- comparisons of the performance of well-known solvers,
- interpretation of the numerical solution in terms of the application field,
- ease of access and use.

There exist other test sets, e.g., NSDTST and STDTST by Enright & Pryce [EP87], PADETEST by Bellen [Bel92], the Geneva test set by Hairer & Wanner [HW] and the Test Frame for Ordinary Differential Equations by Nowak and Gebauer [NG97], which all have their own qualities.

I.2 Structure of this test set

The test set consists of a descriptive part and a software part. The first part describes solvers and test problems and reports on the behavior of the solvers when applied to these problems. Section II explains how this information is presented. The software serves as a platform to test the performance of a solver on a particular test problem by a user of the test set. In Section IV we specify the format of the Fortran subroutines and explain how to run test problems with the help of drivers that make these codes suitable for runs with a number of solvers. Currently, BIMD, DASSL, GAMD, MEBDFDAE, MEBDFI, PSIDE, RADAU, RADAU5 and VODE are supported.

I.3 How to submit new test problems

We invite people to contribute new test problems to this test set. To restrict the amount of time required for the maintainers of the test set to incorporate new problems, it is important that the submissions are in the prescribed format. Firstly, every problem should have a description containing the 4 sections mentioned in Section II, preferably as a LATEX-file. Secondly, a set of Fortran subroutines that is necessary for the implementation has to be supplied in the format specified in Section IV.

Submissions can be sent by e-mail to testset@dm.uniba.it
I.4 How to obtain this test set

The latest release of this test set can be obtained via the WWW page with URL

http://www.dm.uniba.it/~testset,

The first release of this test set appeared in [LSV96], the second release in [LS98], the other releases in [MI03], [MMI06].

I.5 Acknowledgements

We gratefully acknowledge Jacek Kierzenka for his help in defining the interface that that allow the use of the IVP test set problems in the MATLAB environment, the CWI group that set up the first two versions of the testset: P.J. van der Houwen, W. Hoffmann, B.P. Sommeijer, W.M. Lioen, W.A. van der Veen, J.J.B. de Swart, J.E. Frank. In particular we wish to thank P.J. van der Houwen and Walter Lioen, who helped us during the installation procedure.

I.6 People involved

This test set is maintained by the INdAM Bari unit project group Codes and test problems for Differential Equations (coordinator F. Mazzia). The revision 2.4 has been sponsored by the project "Metodi numerici e software per equazioni differenziali ordinarie: problemi a valori iniziali ed al contorno" of the University of Bari (2006). The revision 2.3 has been sponsored by the project PRIN 2004 "Metodi numerici e software matematico per le applicazioni" (coordinator L. Brugnano, local coordinator F. Mazzia) and by the project "Metodi Numerici per equazioni differenziali" (coordinator P. Amadio), sponsored by the University of Bari. In January 2002 a steering committee of A. Bellen (Università di Trieste, Italy), J.R. Cash (Imperial College, London, U.K.), E. Hairer (Université de Genève, Switzerland), F. Krogh (Math à la Carte, Tujunga, California, U.S.A), L. Petzold (University of California, Santa Barbara, U.S.A), B. Simeon, G. Söderlind (Lund University, Sweden), D. Trigiante (Università di Firenze, Italy) and P.J. van der Houwen (formerly at CWI, Amsterdam, The Netherlands) has been set up to oversee this project.

References


II Format of the problem descriptions

Every problem description contains the four sections described below.

II.1 General information

The problem identification is given: the type of problem (IDE, ODE or DAE), its dimension and index. The contributor and any further relevant information are listed too. What is meant here by IDE, ODE, DAE and index, is explained in §IV.

II.2 Mathematical description of the problem

All ingredients that are necessary for implementation are given in mathematical formulas.

II.3 Origin of the problem

A brief description of the origin of the problem, in order to give its physical interpretation. References to the literature are given for further details.

II.4 Numerical solution of the problem

This section contains:

1. Reference solution at the end of the integration interval. The values of (some of) the components of a reference solution at the end of the integration interval are listed.

2. Run characteristics. Integration statistics, if applicable, of runs with BIMD, DASSL, GAMD, MEBDFDAE, MEBDFI, PSIDE, RADAU, RADAU5, and VODE serve to give insight in the numerical difficulty of the problem.

The experiments were done on an Alphaserver DS20E, with a 667 MHz EV67 processor. We used the Fortran 90 compiler with optimization: f90 -O5 <source code>. If a run did not produce correct results then we report what went wrong.

The characteristics are in the following format:

- **solver**
  The name of the numerical solver with which the run was performed.

- **rtol**
  The user supplied relative error tolerance.

- **atol**
  The user supplied absolute error tolerance.

- **h0**
  The user supplied initial step size (if relevant).

- **scd**
  The scd values denote the minimum number of significant correct digits in the numerical solution at the end of the integration interval, i.e.

  \[ \text{scd} := -\log_{10}(|| \text{relative error at the end of the integration interval} ||_{\infty}). \quad (II.1) \]

If some components of the solution vector are not taken into account for the computation of the scd value, or if the absolute error is computed instead of the relative error, then this is specified locally.
II-ii  

**Format of the problem descriptions**

- **mescd**
  
  \[
  \text{mescd} := - \log_{10}( || \text{absolute error} \| \text{./} (\text{atol} / \text{rtol} + || y_{\text{true}} ||) ||_{\infty}).
  \]  

  (II.2)

  where the absolute error is computed at the end of the integration interval, atol and rtol
  are the input tolerances, ytrue is the exact solution at the end of the integration interval
  and ./ and .* are element by element operators. In this case all the components of the
  solution are taken into account.

- **steps**
  
  Total number of steps taken by the solver (including rejected steps due to error test failures
  and/or convergence test failures).

- **accept**
  
  The number of accepted steps.

- **# f and # Jac**
  
  The number of evaluations of the derivative function and its Jacobians, respectively.

- **# LU**
  
  The number of LU-decompositions (not for DASSL). The codes, except for RADAU and
  RADAU5, count the LU-decompositions of systems of dimension \(d\), where \(d\) is the dimension
  of the test problem.

  RADAU and RADAU5 use an \(s\)-stage Radau IIA method. For RADAU5, \(s = 3\) and for
  RADAU, \(s = 3, 5 \text{ or } 7\). Every iteration of the inexact Newton process, used for solving
  systems of non-linear equations, requires the solution of a linear system of dimension \(sd\).

  By means of transformations, this linear system is reduced to \((s + 1)/2\) linear systems of
  dimension \(d\). Of these systems, one system is real, and \((s - 1)/2\) systems are complex. The
  decompositions of all \((s + 1)/2\) linear systems are counted by RADAU and RADAU5 as \(1\)
  LU-decomposition.

- **CPU**
  
  The CPU time in seconds to perform the run on the aforementioned computer. Since
  timings may depend on other processes (like e.g. daemons), we perform 10 runs, discard
  the maximum and minimum values and list the medium of the CPU times.

PSIDE – Parallel Software for Implicit Differential Equations – is a Fortran 77 code for solving
IDE problems. It is developed for parallel, shared memory computers. The integration
characteristics in the tables refer to a one-processor computer. Since PSIDE can do four function
evaluations and four linear system solves concurrently on a computer with four processors, one
may divide the number of function evaluations, decompositions and solves in the tables by four
to obtain the analogous effective characteristics for four-processor machines.

3. **Behavior of the numerical solution.** Plots of (some of) the solution components over (part
of) the integration interval are presented.

4. **Work-precision diagrams.** For every relevant solver, a range of input tolerances and, if
necessary, a range of initial stepsizes, were used to produce plots of the resulting scd or mescd
values, defined in Formulas (II.1) and (II.2), against the number of CPU seconds needed for
the run on the aforementioned computer, with the setting as described before. Here we took
again the medium of the CPU times of 10 runs, after discarding the maximum and minimum
values. The format of these diagrams is as in Hairer & Wanner [HW96, pp. 166–167, 324–325].
The range of input tolerances and initial stepsizes is problem dependent and specified locally.
The input parameters for the runs in the tables with run characteristics are such that these runs
appear in the work-precision diagrams as well. The code PSIDE has been performed only on
one processor.
We want to emphasize that the reader should be careful with using these diagrams for a mutual comparison of the solvers. The diagrams just show the result of runs with the prescribed input on the specified computer. A more sophisticated setting of the input parameters, another computer or compiler, as well as another range of tolerances might change the diagrams considerably.

References

Format of the problem descriptions
III Format of the solver descriptions

Every solver description contains the four sections described below.

III.1 General information

The name of the solver, the type of problem it solves (ODE, IDE or DAE), the authors, the date of the first release, the language, the official link where it is possible to retrieve the software and any further relevant information are listed. What is meant here by IDE, ODE, DAE and index, is explained in §IV.

III.2 Numerical method

General details about the numerical method implemented in the code and references to the literature are given.

III.3 Implementation details

A brief description of the implementation choices used in the codes, like the step variation strategy, the numerical solution of linear and non linear systems and any other useful information together with references to the literature are given.

III.4 How to solve test problems with the solver

This section contains a description of the compiling sequence and explains how to solve test problems in the test set format.
Format of the solver descriptions
IV  The software part of the test set

IV.1 Classification of test problems

We have categorized the test problems in three classes: IDEs, ODEs and DAEs.

In this test set, we call a problem an IDE (system of Implicit Differential Equations) if it is of the form

\[ f(t, y, y') = 0, \quad t_0 \leq t \leq t_{\text{end}}, \]
\[ y, f \in \mathbb{R}^d, \]
\[ y(t_0) \text{ and } y'(t_0) \text{ are given.} \]

A problem is named an ODE (system of Ordinary Differential Equations), if it has the form

\[ y' = f(t, y), \quad t_0 \leq t \leq t_{\text{end}}, \]
\[ y, f \in \mathbb{R}^d, \]
\[ y(t_0) \text{ is given}, \]

whereas the label DAE is given to problems which can be cast in the form

\[ My' = f(t, y), \quad t_0 \leq t \leq t_{\text{end}}, \]
\[ y, f \in \mathbb{R}^d, \quad M \in \mathbb{R}^{d \times d}, \]
\[ y(t_0) \text{ and } y'(t_0) \text{ are given}, \]

where \( M \) is a constant, possibly singular matrix. Note that ODEs and DAEs are subclasses of IDEs.

IV.2 How to perform tests

You can perform one of the following types of tests:

- solve test set problems with solvers that are supported in the test set,
- solve test set problems with your own solver,
- solve your own problem with solvers that are supported in the test set,
- solve a test set problem using the web facility,
- solve your own problem using the web facility,
- solve test set problems using a MATLAB solver,
- solve your own problem in the test set format using a MATLAB solver.

For the first five types of tests, four types of codes are involved: a solver, a driver, a problem code and auxiliary routines, for the last two types of tests the matlab interface of the problem is generated using two auxiliary routines. The solvers available are described in §1-1-1-1-9-1. Currently, there are 9 solvers available:

1. BIMD for ODEs and DAEs of index less than or equal to 3,
2. DASSL for ODEs and IDEs/DAEs of index less than or equal to 1,
3. GAMD for ODEs and DAEs of index less than or equal to 3,
4. MEBDFDAE for ODEs and DAEs of index less than or equal to 3,
5. MEBDFI for ODEs and IDEs/DAEs of index less than or equal to 3,
6. PSIDE for ODEs and IDEs/DAEs of index up to at least 3,
7. RADAU for ODEs and DAEs of index less than or equal to 3,
8. RADAU5 for ODEs and DAEs of index less than or equal to 3, and
9. VODE for ODEs.

These solvers can be obtained via [MM08] in the files bimd.f, ddassl.f, gamd.f90, mebdfd.f, mebdfi.f, pside.f, radau.f, radau5.f and vode.f. These files contain versions of the solvers with which the numerical experiments were conducted. The official links to the solvers, which possibly direct to more recent versions, can be found at [MM08] too.

The drivers bimdd.f, dassld.f, gamdd.f, mebdfdaed.f, mebdfid.f, psided.f, radaud.f, radau5d.f and voded.f, which are available at [MM08], are such that runs can be performed that solve the problem numerically with the aforementioned solvers.

For every test problem, the file problem.f contains a set of nine Fortran 77 subroutines defining the problem. Although the format of the subroutines is the same for all three classes, the meaning of the arguments may depend on the problem class. Section IV.3 describes the format of the problem codes.

The auxiliary linear algebra routines for the solvers are in bimda.f, dassla.f, gamda.f90, psidea.f, radaua.f (for both RADAU and RADAU5) and vodea.f. For MEBDFDAE/MEBFI, the linear algebra routines are included in mebdfdae.f/mebdfi.f. The auxiliary file report.f contains a user interface. All these files are available at [MM08] as well.

IV.2.1 How to solve test problems with available solvers

Compiling

\[
f77 -o dotest solverd.f problem.f solvera.f solver.f report.f,
\]

for the solvers written in Fortran 77, will yield an executable dotest that solves the problem, of which the Fortran routines in the format described in Section IV.3 are in the file problem.f. A complete description of each solver together with some examples are reported in the SOLVERS sections §I-2-1-I-9-1. A makefile is also available in the [MM08] to help in the compilation steps.

IV.2.2 How to solve test problems with your own solver

The following guidelines serve to test your own solver with the test set problems.

- Write your own solver in a format similar to existing solvers in the file own.f.
- (Optional) You may like to put the linear algebra subroutines in a separate file owna.f. In this way you can, for example, use the linear algebra of an existing solver.
- Write driver subroutines in the file ownd.f. If the format of your solver is similar to that of a solver that is already available in the test set, then this will only require minor modifications of the driver routines of that solver.
- Adjust the file report.f as indicated in the comment lines of this file. This will only be a minor modification.
- Compiling

\[
f77 -o dotest ownd.f problem.f own.f owna.f report.f,
\]

will yield an executable dotest that solves the problem, of which the Fortran routines are in the file problem.f.
The software part of the test set

IV.2.3 How to solve your own problem with available solvers

The following guidelines serve to solve your own problem with the solvers that are supported in the test set.

- Write your own problem in a format similar to that of the test set problems in the file `newprob.f`. This format is described precisely in Section IV.3.
- Adjust the file `report.f` as indicated in the comment lines of this file. This will only be a minor modification.
- To solve your problem with, for example, DASSL, compiling
  ```
  f77 -o dotest dassld.f newprob.f ddassl.f dassla.f report.f,
  ```
  will give you the desired executable `dotest`.

IV.2.4 How to solve a test set problem using the web facility

In [MM08], following the link “compile and run on line” it is possible to solve a test set problem online, using the supported solvers. The user input are the relative tolerance `rtol`, the absolute tolerance `atol` and the initial stepsize `h0` for the solvers that need it. As a results the solution computed in the last point, the `scd` and `mescd` and some integration characteristics, as described in §II.4, are displayed. The plots of some component of the solution are also visualized.

IV.2.5 How to solve your own problem using the web facility

In [MM08], following the link “compile and run on line” it is possible to upload a file containing the subroutines describing the problem written using the format described precisely in §IV.3. Then it is possible to choose one of the supported solver for the solution of the problem. The user input are the relative tolerance `rtol`, the absolute tolerance `atol` and the initial stepsize `h0` for the solvers that need it. As a results the solution computed in the last point, the `scd` and `mescd` if the reference solution is available and some integration characteristics, as described in §II.4, are displayed. The plots of the components of the solution defined in the subroutine `setoutput` are also visualized.

IV.2.6 How to solve test set problems using a MATLAB solver

The MATLAB [Mat] function `minterface.m` together with the fortran function `matlab_interface.F`, allow to construct the mex files to run problems in the MATLAB environment. The only restriction is that you need to put the problems and the auxiliary routines in the correct directory. We suggest to download the complete distribution tree of the IVP test set in [MM08] if you want to use the matlab interface.

The MATLAB instruction:

```matlab
MPROB = minterface(problem)
```

returns a function handle to a MEX-Function interface to problem `problem`. If needed, the Fortran MEX-Function interface is automatically generated and compiled, you need a Fortran compiler compatible with the MATLAB environment to complete the compilation steps. Moreover, before using, for the first time, this utilities, at the MATLAB prompt type

```matlab
mex -setup
```

and select the Fortran compiler you want to use.

The interface `mprob` supports the following calling sequences:
The input parameters are the same defined in IV.3 for the fortran functions defining the problem.

The function odetest.m contains a user interface to run and compile the problems in the MATLAB environment. As an example the instruction:

```matlab
>> [sol,stats] = odetest(problem,'ode15s',1e-5,1e-4,1)
```

solves the problem using the matlab solver 'ode15s', with absolute tolerance equal to 1e-5, relative tolerance equal to 1e-4, the first component of the solution is plotted using the MATLAB function 'odeplot'. The output variable sol contains information about the solution.

Use the MATLAB help to have information about the input/output parameters of the functions.

IV.2.7 How to solve you own problem in the test set format using a MATLAB solver

Write your own problem in a format similar to that of the test set problems, as described in Section IV.3 in the file newprob.f. Then put the file in the correct directory in the testset distribution. The instruction:

```matlab
MPROB = minterface(newprob)
```

returns a function handle to a MEX-Function interface to problem newprob.

The instruction odetest

```matlab
>> [sol,stats] = odetest(newprob,'ode15s',1e-5,1e-4,1)
```

will automatically generate the function handle and solve the problem with the MATLAB solver ode15s.

IV.3 Format of the problem codes

The eight subroutines that define the problem are called PROB, INIT, SETTOLERANCES, SETOUTPUT, FEVAL, JEVAL, MEVAL, and SOLUT. The following subsections describe the format of these subroutines in full detail. An additional function PIDATE allows to check the problem interface date, for the current release this function should be equal to:

```fortran
integer function pidate()
    pidate = 20060828
    return
end
```

In the sequel, the variables listed under INTENT(IN), INTENT(INOUT), and INTENT(OUT) are input, update and output variables, respectively.
IV.3.1 Subroutine PROB

This routine gives some general information about the test problem.

```fortran
SUBROUTINE PROB(FULLNM,PROBLM,TYPE,
+ NEQN,NDISC,T,
+ NUMJAC,MLJAC,MUJAC,
+ NUMMAS,MLMAS,MUMAS,
+ IND)
CHARACTER*(*) FULLNM, PROBLM, TYPE
INTEGER NEQN,NDISC,MLJAC,MUJAC,MLMAS,MUMAS,IND(*)
DOUBLE PRECISION T(0:*)
LOGICAL NUMJAC, NUMMAS
C INTENT(OUT) FULLNM,PROBLM,TYPE,NEQN,NDISC,T,NUMJAC,MLJAC,
C + MUJAC,NUMMAS,MLMAS,MUMAS,IND
C +
C Meanings of the arguments:

FULLNM
This character string contains the long name of the problem, e.g. Chemical Akzo Nobel problem.

PROBLM
This character string contains the short name of the problem, e.g. chemakzo, and corresponds to the name of the Fortran source file.

TYPE
This character string takes the value IDE, ODE or DAE, depending on the type of problem.

NEQN
The dimension \(d\) of the problem, which is the number of equations to be solved.

NDISC
The number of discontinuities in time of the function \(f\) or its derivative. The solver is restarted at every such discontinuity by the driver.

T
An array containing time points.
- If NDISC .EQ. 0, then T(0) contains \(t_0\) and T(1) contains \(t_{\text{end}}\).
- If NDISC .GT. 0, then T(0) contains \(t_0\), T(NDISC+1) contains \(t_{\text{end}}\) and T(1) ... T(NDISC) are the time points where the function \(f\) or its derivative has a discontinuity in time.

NUMJAC
To solve the problem numerically, it is necessary to use the partial derivative \(J := \partial f / \partial y\). If \(J\) is available analytically, then NUMJAC = .FALSE. and \(J\) is provided via subroutine JEVAL. If \(J\) is not available, then NUMJAC = .TRUE. and JEVAL is a dummy subroutine. In this case, the solvers approximate \(J\) by numerical differencing.

MLJAC and MUJAC
These integers contain information about the structure of \(J := \partial f / \partial y\). If \(J\) is a full matrix, then MLJAC = NEQN, otherwise MLJAC and MUJAC equal the number of nonzero lower co-diagonals and the number of nonzero upper co-diagonals of \(J\), respectively.

NUMMAS
Only relevant for IDEs.
For IDEs, it is necessary to use the partial derivative $M := \frac{\partial f}{\partial y}$. If $M$ is available analytically, then NUMMAS = .FALSE. and $M$ is provided via subroutine MEVAL. If $M$ is not available, then NUMMAS = .TRUE. and MEVAL is a dummy subroutine. In this case, the solvers have to approximate $M$ by numerical differencing.

- For DAEs and ODEs, NUMMAS is not referenced.

MLMAS and MUMAS
These integers contain information about the structure of the constant matrix $M$ (for DAEs) or the matrix $M := \frac{\partial f}{\partial y}$ (for IDEs).

- For IDEs and DAEs: If $M$ is a full matrix, then MLMAS = NEQN, otherwise MLMAS and MUMAS equal the number of nonzero lower co-diagonals and the number of nonzero upper co-diagonals of $M$, respectively.
- For ODEs, MLMAS and MUMAS are not referenced.

IND
Connected to IDEs and DAEs is the concept of index.

- For ODEs, IND is not referenced.
- For IDEs and DAEs, IND is an array of length NEQN and IND(I) specifies the index of variable I.

IV.3.2 Subroutine INIT
This routine contains the initial values $y(t_0)$ and $y'(t_0)$.

```fortran
SUBROUTINE INIT(NEQN,T,Y,YPRIME,CONSIS)
INTEGER NEQN
DOUBLE PRECISION T,Y(NEQN),YPRIME(NEQN)
LOGICAL CONSIS
C INTENT(IN) NEQN,T
C INTENT(OUT) Y,YPRIME,CONSIS
```

Meaning of the arguments:

**NEQN**
The dimension of the problem.

**Y(NEQN)**
Contains the initial value $y(t_0)$.

**YPRIME(NEQN)**
Only relevant for IDEs and DAEs.

- For IDEs and DAEs, YPRIME contains the initial value $y'(t_0)$.
- For ODEs, YPRIME is not set. If needed by the solver, it is computed in the driver as $y'(t_0) = f(t_0, y_0)$.

**CONSIS**
Only relevant for IDEs and DAEs.
The software part of the test set

- For IDEs and DAEs, CONSIS is a switch for the consistency of the initial values. If CONSIS = .TRUE., then $y(t_0)$ and $y'(t_0)$ are assumed to be consistent. If CONSIS = .FALSE., then $y(t_0)$ and $y'(t_0)$ are possibly inconsistent. Solvers with a facility to compute consistent initial values internally, will try to do so in this case. Currently, all problems in the test set have consistent initial values.
- For ODEs, CONSIS is not referenced.

**IV.3.3 Subroutine SETTOLERANCES**

This routine defines the input tolerances RTOL and ATOL.

```fortran
SUBROUTINE SETTOLERANCES(NEQN,RTOL,ATOL,TOLVEC)
INTEGER NEQN
LOGICAL TOLVEC
DOUBLE PRECISION RTOL(NEQN), ATOL(NEQN)
C INTENT(IN) NEQN
C INTENT(INOUT) RTOL, ATOL
C INTENT(OUT) TOLVEC

Meaning of the arguments:

NEQN
The dimension of the problem.

RTOL
Contains the relative tolerances.
- In input contains the value RTOL(1).
- In output could contain a vector valued RTOL, with different values for the relative tolerances in each component.

ATOL
Contains the absolute tolerances.
- In input contains the value ATOL(1).
- In output could contain a vector valued ATOL, with different values for the absolute tolerances in each component.

TOLVEC
Logical output variable.
- TOLVEC = .TRUE. if all the component of RTOL and ATOL are initialized.
- TOLVEC = .FALSE. if only the first component of RTOL and ATOL is initialized.

**IV.3.4 Subroutine SETOUTPUT**

This routine contains information about the required output.

```fortran
SUBROUTINE SETOUTPUT(NEQN,SOLREF,PRINTSOLOUT,
+ NINDSOL,INDSOL)
LOGICAL SOLREF, PRINTSOLOUT
INTEGER NEQN, NINDSOL
Meaning of the arguments:

**NEQN**
The dimension of the problem.

**SOLREF**
Contains information about the reference solution.

- **SOLREF = .TRUE.** means that the reference solution is available in the function `solut`.
- **SOLREF = .FALSE.** means that the reference solution is not available, the subroutine `SOLOUT` must be a dummy subroutine.

**PRINTSOLOUT**
Contains information about the required output.

- **PRINTSOLOUT = .TRUE.** means that some components of the intermediate computed values of the solution are printed in the output file called `problemSOLVER.txt`.
- **PRINTSOLOUT = .FALSE.** means that no intermediate values are printed.

**NINDSOL**
If **PRINTSOLOUT = .TRUE.**, **NINDSOL** contains the number of components to be printed.

**INDSOL**
If **PRINTSOLOUT = .TRUE.**, **INDSOL(1:NINDSOL)** contains the index of the **NINDSOL** components to be printed.

### IV.3.5 Subroutine FEVAL

This subroutine evaluates the function $f$.

```fortran
SUBROUTINE FEVAL(NEQN,T,Y,YPRIME,F,IERR,RPAR,IPAR)
   INTEGER NEQN,IERR,IPAR(*)
   DOUBLE PRECISION T,Y(NEQN),YPRIME(NEQN),F(NEQN),RPAR(*)
   C   INTENT(IN)  NEQN,T,Y,YPRIME
   C   INTENT(INOUT) RPAR,IPAR
   C   INTENT(OUT)  F,IERR
```

Meaning of the arguments:

**NEQN**
The dimension of the problem.

**T**
The time point where the function is evaluated.

**Y(NEQN)**
The value of $y$ in which the function is evaluated.
The software part of the test set

YPRIME (NEQN)

Only relevant for IDEs.
- For IDEs, this is the value of \( y' \) in which the function \( f \) is evaluated.
- For ODEs and DAEs, YPRIME is not referenced.

F(NEQN)

The resulting function value \( f(T, Y) \) (for ODEs and DAEs), or \( f(T, Y, YPRIME) \) (for IDEs).

IERR

IERR is an integer flag which is always equal to zero on input. Subroutine FEVAL sets IERR = -1 if FEVAL cannot be evaluated for the current values of T, Y and YPRIME. Some solvers have the facility to attempt to prevent the occurrence of IERR = -1, or return to the driver in that case. IERR has an analogous meaning in subroutines JEVAL and MEVAL.

RPAR and IPAR

RPAR and IPAR are double precision and integer arrays, respectively, which can be used for communication between the driver and the subroutines FEVAL, JEVAL and MEVAL. If RPAR and IPAR are not needed, these parameters are ignored by treating them as dummy arguments. RPAR and IPAR have the same meaning in subroutines JEVAL and MEVAL.

IV.3.6 Subroutine JEVAL

This subroutine evaluates the derivative (or Jacobian) of the function \( f \) with respect to \( y \).

```fortran
SUBROUTINE JEVAL(LDIM, NEQN, T, Y, YPRIME, DFDY, IERR, RPAR, IPAR)
INTEGER LDIM, NEQN, IERR, IPAR(*)
DOUBLE PRECISION T, Y(NEQN), YPRIME(NEQN), DFDY(LDIM, NEQN), RPAR(*)
C INTENT(IN)  LDIM, NEQN, T, Y, YPRIME
C INTENT(INOUT) RPAR, IPAR
C INTENT(OUT)  DFDY, IERR
```

Meaning of the arguments:

**LDIM**

The leading dimension of the array DFDY.

**NEQN**

The dimension of the problem.

**T**

The time point where the derivative is evaluated.

**Y(NEQN)**

The value of \( y \) in which the derivative is evaluated.

**YPRIME (NEQN)**

Only relevant for IDEs.
- For IDEs, this is the value of \( y' \) in which the derivative \( \partial f(t, y, y')/\partial y \) is evaluated.
- For ODEs and DAEs, YPRIME is not referenced.

**DFDY (LDIM, NEQN)**

The array with the resulting Jacobian matrix.
- If $\partial f/\partial y$ is a full matrix ($MLJAC = NEQN$), then $DFDY(I,J)$ contains $\partial f_I/\partial y_J$.
- If $\partial f/\partial y$ is a band matrix ($0 \leq MLJAC < NEQN$), then $DFDY(I-J+MUJAC+1,J)$ contains $\partial f_I/\partial y_J$ (LAPACK / LINPACK / BLAS storage).

**IERR, RPAR and IPAR**
See the description of subroutine FEVAL.

### IV.3.7 Subroutine MEVAL

For ODEs, MEVAL is not called and a dummy subroutine is supplied. For DAEs, it supplies the constant matrix $M$. For IDEs, it evaluates the matrix $M := \partial f/\partial y'$.

```
SUBROUTINE MEVAL(LDIM,NEQN,T,Y,YPRIME,DFDDY,IERR,RPAR,IPAR)
INTEGER LDIM,NEQN,IERR,IPAR(*)
DOUBLE PRECISION T,Y(NEQN),YPRIME(NEQN),DFDDY(LDIM,NEQN),RPAR(*)
C INTENT(IN) LDIM,NEQN,T,Y,YPRIME
C INTENT(INOUT) RPAR,IPAR
C INTENT(OUT) DFDDY,IERR
```

#### Meaning of the arguments:

**LDIM**
The leading dimension of the matrix $M$.

**NEQN**
The dimension of the problem.

**T**
The time point where $M$ is evaluated. (For DAEs, $T$ is not referenced.)

**Y(NEQN)**
The value of $y$ in which $M$ is evaluated. (For DAEs, $Y$ is not referenced.)

**YPRIME(NEQN)**
The value of $y'$ in which $M$ is evaluated. (For DAEs, YPRIME is not referenced.)

**DFDDY(LDIM,NEQN)**
This array contains the constant matrix $M$ (for DAEs) or $M := \partial f/\partial y'$ (for IDEs).

- If $M$ is a full matrix ($MLMAS = NEQN$), then $DFDDY(I,J)$ contains $M_{I,J}$ for DAEs and $\partial f_I/\partial y'_J$ for IDEs.
- If $M$ is a band matrix ($0 \leq MLMAS < NEQN$), then $DFDDY(I-J+MUMAS+1,J)$ contains $M_{I,J}$ for DAEs and $\partial f_I/\partial y'_J$ for IDEs. (LAPACK / LINPACK / BLAS storage).

**IERR, RPAR and IPAR**
See the description of subroutine FEVAL.

### IV.3.8 Subroutine SOLUT

This routine contains the reference solution.

```
SUBROUTINE SOLUT(NEQN,T,Y)
INTEGER NEQN
DOUBLE PRECISION T,Y(NEQN)
C INTENT(IN) NEQN,T
C INTENT(OUT) Y
```
Meaning of the arguments:

NEQN  
The dimension of the problem.

T  
The value of \( t \), in which the reference solution is given (normally \( t_{\text{end}} \)).

\( Y(\text{NEQN}) \)  
This array contains the reference solution in \( t = T \).

IV.4 Format of the solver codes

The following guidelines serve to write a solver that could be easily inserted in the test set.

- Write your own solver in a format similar to existing solvers in the file own.f.
- Put the linear algebra subroutines in a separate file owna.f.
- Write driver subroutines in the file ownd.f. If the format of your solver is similar to that of a solver that is already available in the test set, then this will only require minor modifications of the driver routines of that solver.
- Adjust the file report.f as indicated in the comment lines of this file. This will only be a minor modification.

References


The software part of the test set
Part I
Solvers

This part contains a brief description of the solvers used in the comparisons. The description is not meant in substitution of the information given by the authors of the solvers, but just to provide the users with some general specifics of the solvers supported and to collect the most useful bibliography.

Also, some suggestions on how to use the codes in combination with the software provided in the test set home page are given.
1 Solver BiMD

1.1 General information

Authors: C. Magherini and L. Brugnano
first version: October, 2005
last update: September, 2006
language: Fortran 77
availability: the code BiMD is freely available (in the public domain)
official link: http://www.math.unifi.it/~brugnano/BiM/index.html
problem type: ODEs, DAEs up to index 3
IVPtestset files: solver: bimd.f
driver: bimdd.f
auxiliary files: bimda.f (auxiliary routines)

1.2 Numerical method

The code BiMD (written in FORTRAN 77) is based on Blended Implicit Methods of orders 4, 6, 8, 10 and 12. These are a class of $L$-stable Block Implicit Methods defined as a suitable combination (blending) of two equivalent forms of a basic method in order to favorably meet implementation requirements [BT01, BM02, BMM06, Mag04].

1.3 Implementation details

Nonlinear systems are solved by means of an iterative procedure, called blended iteration, based on a nonlinear splitting “naturally” associated to the methods. The strategies for the variation of both the stepsize of integration and the order of the method rely on an estimate of the local truncation errors, obtained through a deferred correction-like procedure, and on an estimate of the convergence properties of the blended iteration. Almost all the details concerning the construction of the code are described in [BM04, BM05, BM06]. The style used during the formulation of the code is very similar to the one used in the codes RADAU and GAM, from which the authors imported some subroutines and comments. Moreover, the name and the meaning of a number of input parameters and local variables have been fully inherited from such codes.

1.4 How to solve test problems with BiMD

Compiling

```
f90 -o dotest bimdd.f problem.f bimda.f bimd.f report.f
```

will yield an executable dotest that solves the problem, of which the Fortran routines in the format described in Section IV.3 are in the file problem.f.

As an example, we perform a test run, in which we solve problem HIRES. Figure I.1.1 shows what one has to do.

References


$ f90 -O5 -o dotest bimdd.f hires.f bimda.f bimd.f report.f
$ ./dotest

Test Set for IVP Solvers (release 2.3)

Solving Problem HIRES using BiMD

User input:

give relative error tolerance:  
id-4

give absolute error tolerance:  
id-4

give initial stepsize:  
id-4

Numerical solution:

<table>
<thead>
<tr>
<th>solution component</th>
<th>scd</th>
<th>mixed</th>
<th>abs</th>
<th>rel</th>
<th>mix - abs,rel</th>
</tr>
</thead>
<tbody>
<tr>
<td>y( 1) = 0.7370390869868376E-003</td>
<td>7.04</td>
<td>7.04</td>
<td>3.90</td>
<td></td>
<td></td>
</tr>
<tr>
<td>y( 2) = 0.1442309432673056E-003</td>
<td>7.75</td>
<td>7.75</td>
<td>3.91</td>
<td></td>
<td></td>
</tr>
<tr>
<td>y( 3) = 0.5886726466999230E-004</td>
<td>7.70</td>
<td>7.70</td>
<td>3.47</td>
<td></td>
<td></td>
</tr>
<tr>
<td>y( 4) = 0.1175514405948055E-002</td>
<td>6.86</td>
<td>6.86</td>
<td>3.93</td>
<td></td>
<td></td>
</tr>
<tr>
<td>y( 5) = 0.2382225700959266E-002</td>
<td>5.38</td>
<td>5.38</td>
<td>2.76</td>
<td></td>
<td></td>
</tr>
<tr>
<td>y( 6) = 0.6222129415035646E-002</td>
<td>4.78</td>
<td>4.77</td>
<td>2.57</td>
<td></td>
<td></td>
</tr>
<tr>
<td>y( 7) = 0.2849335096690541E-002</td>
<td>6.19</td>
<td>6.19</td>
<td>3.64</td>
<td></td>
<td></td>
</tr>
<tr>
<td>y( 8) = 0.2850649043094471E-002</td>
<td>6.19</td>
<td>6.19</td>
<td>3.64</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

used components for scd 8 8 8
scd of Y (maximum norm) 4.78 4.77 2.57

using mixed error yields mescd 4.78
using relative error yields scd 2.57

Integration characteristics:

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>number of integration steps</td>
<td>36</td>
</tr>
<tr>
<td>number of accepted steps</td>
<td>33</td>
</tr>
<tr>
<td>number of f evaluations</td>
<td>559</td>
</tr>
<tr>
<td>number of Jacobian evaluations</td>
<td>30</td>
</tr>
<tr>
<td>number of LU decompositions</td>
<td>36</td>
</tr>
<tr>
<td>CPU-time used:</td>
<td>0.0020 sec</td>
</tr>
</tbody>
</table>

Figure I.1.1: Example of performing a test run, in which we solve problem HIRES with BiMD. The experiment was done on an ALPHAserver DS20E, with a 667MHz EV67 processor. We used the Fortran 90 compiler f90 with the optimization flag -O5.


2 Solver DASSL

2.1 General information

Author: L. Petzold
first version: March 15, 1983
last update: July 11, 2000
language: Fortran 77
availability: the code DASSL is freely available (in the public domain)
official link: http://www.netlib.org/ode/ddassl.f
problem type: IDEs/DAEs of index less or equal to 1
IVP testset files: solver: ddassl.f
driver: dassld.f
auxiliary files: dassla.f (auxiliary linear algebra routines)

2.2 Numerical method

This code implements the Backward Differentiation Formulas of orders one through five to solve an IDE for \( y \) and \( y' \). Values for \( y \) and \( y' \) at the initial time must be given as input. These values must be consistent, (that is, if \( t_0, y_0, y'_0 \) are the given initial values, they must satisfy \( f(t_0, y_0, y'_0) = 0 \) [BCP96].

2.3 Implementation details

The subroutine solves the system from \( t_0 \) to \( t_{\text{out}} \) (final integration time). It allows to continue the solution to get results at additional \( t_{\text{out}} \). This is the interval mode of operation. Intermediate results can also be obtained easily by using the intermediate-output capability. The derivatives are approximated by backward differentiation formulae (BDFs), and the resulting nonlinear system at each time-step is solved by Newton's method. The linear systems are solved using routines from the LINPACK subroutine package. Error handling is accomplished using routines from the SLATEC common mathematical library package. This code is good for stiff ODEs and for DAEs of moderate size, where it is appropriate to treat the Jacobian matrix with dense or banded direct LU decomposition. For large-scale stiff ODE and DAE problems, the user should consider DASPK. For ODE or DAE problems which must stop at the root of a given function of the solution, the user should consider DASKR. The code includes an extensive amount of documentation.

2.4 How to solve test problems with DASSL

Compiling

```
f90 -o dotest dassld.f problem.f ddassl.f dassla.f report.f
```

will yield an executable dotest that solves the problem, of which the Fortran routines in the format described in Section IV.3 are in the file problem.f.

Although DASSL is a code written for problems of index \( \leq 1 \), it can handle some of the higher index problems by adjusting the error control. If possible, this is done in the driver dassld.f.

As an example, we perform a test run, in which we solve problem HIRES. Figure I.2.1 shows what one has to do.
User input:

give relative error tolerance: 1d-4

give absolute error tolerance: 1d-4

Numerical solution:

<table>
<thead>
<tr>
<th>solution component</th>
<th>scd</th>
<th>mixed</th>
<th>abs</th>
<th>rel</th>
<th>mix - abs,rel</th>
</tr>
</thead>
<tbody>
<tr>
<td>y( 1) = 0.7437259735671353E-003</td>
<td>5.18</td>
<td>5.18</td>
<td>2.05</td>
<td></td>
<td></td>
</tr>
<tr>
<td>y( 2) = 0.145514426118115E-003</td>
<td>5.89</td>
<td>5.89</td>
<td>2.04</td>
<td></td>
<td></td>
</tr>
<tr>
<td>y( 3) = 0.6009884916041035E-004</td>
<td>5.92</td>
<td>5.92</td>
<td>1.69</td>
<td></td>
<td></td>
</tr>
<tr>
<td>y( 4) = 0.1188134706173306E-002</td>
<td>4.90</td>
<td>4.90</td>
<td>1.97</td>
<td></td>
<td></td>
</tr>
<tr>
<td>y( 5) = 0.25770466000086416E-002</td>
<td>3.72</td>
<td>3.72</td>
<td>1.10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>y( 6) = 0.682947575510993E-002</td>
<td>3.23</td>
<td>3.23</td>
<td>1.03</td>
<td></td>
<td></td>
</tr>
<tr>
<td>y( 7) = 0.298938592155588E-002</td>
<td>3.86</td>
<td>3.86</td>
<td>1.31</td>
<td></td>
<td></td>
</tr>
<tr>
<td>y( 8) = 0.271061407844423E-002</td>
<td>3.86</td>
<td>3.86</td>
<td>1.31</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

used components for scd 8 8 8

scd of Y (maximum norm) 3.23 3.23 1.03

using mixed error yields mescd 3.23

using relative error yields scd 1.03

Integration characteristics:

number of integration steps 108
number of accepted steps 99
number of f evaluations 173
number of Jacobian evaluations 31

CPU-time used: 0.0010 sec

Figure I.2.1: Example of performing a test run, in which we solve problem HIRES with DASSL. The experiment was done on an ALPHAServer DS20E, with a 667MHz EV67 processor. We used the Fortran 90 compiler f90 with the optimization flag -O5.
References

3 Solver GAMD

3.1 General information

Authors: F. Iavernaro and F. Mazzia
first version: August 1997 (GAM)
last update: February, 2006
language: Fortran 90
availability: the code GAMD is freely available (in the public domain)
official link: http://www.dm.uniba.it/~mazzia/ode/readme.html
problem type: ODEs, DAEs of index less than 3
IVPtestset files: solver: gamd.f90
driver: gamdd.f
auxiliary files: gamda.f90 (auxiliary routines)

3.2 Numerical method

The code GAMD (written in FORTRAN 90) uses the Generalized Adams Methods in block form, of orders 3, 5, 7 and 9. These are A-stable formulae belonging to the class of Boundary Value Methods [BT98, IM99].

3.3 Implementation details

The solution of nonlinear systems is obtained by means of a one-step splitting Newton iteration. The order variation and stepsize selection strategies are based upon an estimation of the local truncation errors for the current, lower and upper order formulae, obtained by means of a deferred correction-like procedure [IM98]. The philosophy and the style used during the formulation of the code are very similar to those characterizing the code RADAU5, from which the authors imported some subroutines, comments and implementation techniques, leaving unchanged the name and the meaning of a number of variables. A preprocessed version of the code GAMD, that allows the user to switch between quadruple and double precision, is also available at the official link http://www.dm.uniba.it/~mazzia/ode/readme.html.

3.4 How to solve test problems with GAMD

Some machines need more virtual memory to compile the subroutine gamda.f90; for example if you are using an ALPHAserver DS20E, with a 667MHz, EV67 processor, execute the following command before the compilation:

```
ulimit -Sd 241000 . Compiling
f90 -o dotest gamdd.f problem.f gamd.f90 gamd.f90 report.f
```

will yield an executable dotest that solves the problem, of which the Fortran routines in the format described in Section IV.3 are in the file problem.f.

As an example, we perform a test run, in which we solve problem HIRES. Figure I.3.1 shows what one has to do.

References

Figure I.3.1: Example of performing a test run, in which we solve problem HIRES with GAMD. The experiment was done on an ALPHAserver DS20E, with a 667MHz EV67 processor. We used the Fortran 90 compiler f90 with the optimization flag -O5.

4 Solver MEBDFDAE

4.1 General information

Author: J. Cash
first version: November, 1998
last update: February, 2006
language: Fortran 77
availability: the code MEBDFDAE is freely available (in the public domain)
official link: http://www.ma.ic.ac.uk/~jcash/IVP_software/mebdftest/mebdfdae.f
problems type: ODEs and DAEs of index less than or equal to 3
IVPtestset files: solver: mebdfdae.f
driver: mebdfi.f
auxiliary files: the linear algebra routines are included in medbdfdae.f.

4.2 Numerical method

The code MEBDFDAE uses the Modified Extended Backward Differentiation Formulas of Cash, that increase the absolute stability regions of the classical BDFs [Cas79, Cas83, Cas03, Hin83, HW96]. These methods are A-stable up to the order 4 and stiffly stable for orders up to 9; therefore they are especially suited for the solution of stiff systems of ODEs [CC92]. The orders of the implemented formulae range from 1 to 8.

4.3 Implementation details

The formulae implemented are three-stages general linear methods with the same Jacobian to be used in the Newton iteration for all the stages. Blas and Lapack auxiliary routines are also used. Versions of this solver for the solutions of ODEs are MEBDF and MEBDFS0, the last one is designed to solve stiff Initial Value Problems for very large sparse systems of ODEs, where the linear equation solver is replaced by the sparse solver YSMP [EGSS77]. Extensions of MEBDFDAE for the solution of very large sparse systems of DAEs is given by the solver MEBDFSD, where the sparse solver used is MA28 [ES77]. A MATLAB translation of MEBDFDAE is available at the official link http://www.ma.ic.ac.uk/~jcash/MATLAB_software/MEBDF.m.

4.4 How to solve test problems with MEBDFDAE

Compiling

f90 -o dotest mebdfdaed.f problem.f mebdfdae.f report.f,

will yield an executable dotest that solves the problem, of which the Fortran routines in the format described in Section IV.3 are in the file problem.f.

As an example, we perform a test run, in which we solve problem HIRES. Figure I.4.1 shows what one has to do.

References


$ f90 -O5 -o dotest mebdfdaed.f hires.f mebdfdae.f report.f
$ dotest

Test Set for IVP Solvers (release 2.3)

Solving Problem HIRES using MEBDFDAE

User input:

give relative error tolerance: 1d-4

give absolute error tolerance: 1d-4

give initial stepsize: 1d-4

Numerical solution:

<table>
<thead>
<tr>
<th>solution component</th>
<th>scd</th>
<th>ignore</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>mixed</td>
<td>abs</td>
</tr>
<tr>
<td>--------------------</td>
<td>-------</td>
<td>-----</td>
</tr>
<tr>
<td>y( 1)</td>
<td>5.33</td>
<td>5.33</td>
</tr>
<tr>
<td>y( 2)</td>
<td>6.03</td>
<td>6.03</td>
</tr>
<tr>
<td>y( 3)</td>
<td>6.05</td>
<td>6.05</td>
</tr>
<tr>
<td>y( 4)</td>
<td>5.06</td>
<td>5.06</td>
</tr>
<tr>
<td>y( 5)</td>
<td>3.84</td>
<td>3.84</td>
</tr>
<tr>
<td>y( 6)</td>
<td>3.32</td>
<td>3.32</td>
</tr>
<tr>
<td>y( 7)</td>
<td>4.08</td>
<td>4.08</td>
</tr>
<tr>
<td>y( 8)</td>
<td>4.08</td>
<td>4.08</td>
</tr>
</tbody>
</table>

used components for scd 8 8 8

scd of Y (maximum norm) 3.32 3.32 1.12

using mixed error yields mescd 3.32
using relative error yields scd 1.12

Integration characteristics:

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>number of integration steps</td>
<td>97</td>
</tr>
<tr>
<td>number of accepted steps</td>
<td>94</td>
</tr>
<tr>
<td>number of f evaluations</td>
<td>168</td>
</tr>
<tr>
<td>number of Jacobian evaluations</td>
<td>21</td>
</tr>
<tr>
<td>number of LU decompositions</td>
<td>21</td>
</tr>
<tr>
<td>CPU-time used:</td>
<td>0.0020 sec</td>
</tr>
</tbody>
</table>

Figure I.4.1: Example of performing a test run, in which we solve problem HIRES with MEBDFDAE. The experiment was done on an ALPHAserver DS20E, with a 667MH EV67 processor. We used the Fortran 90 compiler $f90$ with the optimization flag -O5.


5 Solver MEBDFI

5.1 General information

Authors: T.J. Abdulla and J.R. Cash
first version: October 31, 2003
last update: February, 2006
language: Fortran 77
availability: the code MEBDFI is freely available (in the public domain)
official link: http://www.ma.ic.ac.uk/~jcash/IVP_software/itest/mebdfi.f
problems type: ODEs, DAEs and IDEs of index less than or equal to 3
IVP testset files: solver: mebdfi.f
driver: mebdfid.f
auxiliary files: the linear algebra routines are included in mebdfi.f.

5.2 Numerical method

The code MEBDFI is an extension of MEBDFDAE for the solution of implicit differential equations and uses the Modified Extended Backward Differentiation Formulas of Cash, that increase the absolute stability regions of the classical BDFs [Cas79, Cas83, Cas03, Hin83, HW96]. These methods are A-stable up to the order 4 and stiffly stable for orders up to 9; therefore they are especially suited for the solution of stiff systems of ODEs [CC92]. The orders of the implemented formulae range from 1 to 8.

5.3 Implementation details

The formulae implemented are three-stages general linear methods with the same Jacobian to be used in the Newton iteration for all the stages. Blas and Lapack auxiliary routines are also used. A Fortran 95 translation of MEBDFI made by Bill Paxton is available at the official link of MESA (Modules for Experiments in Stellar Astrophysics) http://theory.kitp.ucsb.edu/~paxton/mesa/mesa_doc/index.html.

5.4 How to solve test problems with MEBDFI

Compiling

```
f90 -o dotest mebdfid.f problem.f mebdfi.f report.f,
```

will yield an executable dotest that solves the problem, of which the Fortran routines in the format described in Section IV.3 are in the file problem.f.

As an example, we perform a test run, in which we solve problem HIRES. Figure I.5.1 shows what one has to do.

References


$ f90 -O5 -o dotest mebdfid.f hires.f mebdfi.f report.f
$ dotest

Test Set for IVP Solvers (release 2.3)

Solving Problem HIRES using MEBDFI

User input:

give relative error tolerance: 1d-4

give absolute error tolerance: 1d-4

give initial stepsize: 1d-4

Numerical solution:

<table>
<thead>
<tr>
<th>solution component</th>
<th>scd</th>
<th>mixed</th>
<th>abs</th>
<th>rel</th>
<th>mix - abs,rel</th>
</tr>
</thead>
<tbody>
<tr>
<td>y( 1) = 0.7360756579676240E-003</td>
<td>5.98</td>
<td>5.98</td>
<td>2.84</td>
<td></td>
<td></td>
</tr>
<tr>
<td>y( 2) = 0.1440435009167338E-003</td>
<td>6.69</td>
<td>6.69</td>
<td>2.85</td>
<td></td>
<td></td>
</tr>
<tr>
<td>y( 3) = 0.58673650307055238E-004</td>
<td>6.67</td>
<td>6.67</td>
<td>2.44</td>
<td></td>
<td></td>
</tr>
<tr>
<td>y( 4) = 0.117382807712226E-002</td>
<td>5.74</td>
<td>5.74</td>
<td>2.81</td>
<td></td>
<td></td>
</tr>
<tr>
<td>y( 5) = 0.2347013337866003E-002</td>
<td>4.41</td>
<td>4.41</td>
<td>1.78</td>
<td></td>
<td></td>
</tr>
<tr>
<td>y( 6) = 0.6023708667056447E-002</td>
<td>3.67</td>
<td>3.67</td>
<td>1.46</td>
<td></td>
<td></td>
</tr>
<tr>
<td>y( 7) = 0.2893669690773767E-002</td>
<td>4.36</td>
<td>4.36</td>
<td>1.81</td>
<td></td>
<td></td>
</tr>
<tr>
<td>y( 8) = 0.28063003090227050E-002</td>
<td>4.36</td>
<td>4.36</td>
<td>1.81</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

used components for scd 8 8 8

scd of Y (maximum norm) 3.67 3.67 1.46

using mixed error yields mescd 3.67
using relative error yields scd 1.46

Integration characteristics:

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>number of integration steps</td>
<td>92</td>
</tr>
<tr>
<td>number of accepted steps</td>
<td>89</td>
</tr>
<tr>
<td>number of f evaluations</td>
<td>311</td>
</tr>
<tr>
<td>number of Jacobian evaluations</td>
<td>18</td>
</tr>
<tr>
<td>number of LU decompositions</td>
<td>18</td>
</tr>
<tr>
<td>CPU-time used</td>
<td>0.0010 sec</td>
</tr>
</tbody>
</table>

Figure I.5.1: Example of performing a test run, in which we solve problem HIRES with MEBDFI. The experiment was done on an ALPHAserver DS20E, with a 667MH EV67 processor. We used the Fortran 90 compiler f90 with the optimization flag -O5.


6 Solver PSIDE

6.1 General information

Authors: J.J.B. de Swart, W.M. Lioen, and W.A. van der Veen
first version: November 28 1997 (version 1.0)
last update: November 25 1998 (version 1.3)
language: Fortran 77
availability: the code PSIDE is freely available (in the public domain)
official link: http://www.cwi.nl/cwi/projects/PSIDE/
problem type: IDEs/DAEs of index upto at least 3
IVP testset files: solver: pside.f
driver: psided.f
auxiliary files: psidea.f (auxiliary linear algebra routines)

6.2 Numerical method

The code uses the four-stage Radau IIA method.

6.3 Implementation details

PSIDE is a Parallel Software for Implicit Differential Equations [SLV97a, SLV97b]. It has been designed for working on shared memory parallel computers, using the OPENMP parallel tools.

The nonlinear systems are solved by a modified Newton process, in which every Newton iterate itself is computed by means of the Parallel Iterative Linear system Solver for Runge-Kutta (PILSRK) proposed in [HS97]. This process is constructed such that the four stage values can be computed simultaneously, thereby making PSIDE suitable for execution on four processors. Full details about the algorithmic choices and the implementation of PSIDE can be found in [SLV97c].

6.4 How to solve test problems with PSIDE

Compiling

```
f90 -o dotest psided.f problem.f pside.f psidea.f report.f
```

will yield an executable dotest that solves the problem, of which the Fortran routines in the format described in Section IV.3 are in the file problem.f. In order to have the correct solution, before the compilation, change the auxiliary routine I1MACH and D1MACH, in the file dassla.f because they are machine dependent.

As an example, we perform a test run, in which we solve problem HIRES. Figure I.6.1 shows what one has to do.

References


Test Set for IVP Solvers (release 2.3)

Solving Problem HIRES using PSIDE

User input:

give relative error tolerance:  
1d-4  
give absolute error tolerance:  
1d-4  

Numerical solution:

<table>
<thead>
<tr>
<th>solution component</th>
<th>scd</th>
<th>mixed</th>
<th>abs</th>
<th>rel</th>
<th>mix - abs,rel</th>
</tr>
</thead>
<tbody>
<tr>
<td>y(  1) = 0.7371770832059414E-003</td>
<td>7.34</td>
<td>7.34</td>
<td>4.21</td>
<td></td>
<td></td>
</tr>
<tr>
<td>y(  2) = 0.1442575715381605E-003</td>
<td>8.05</td>
<td>8.05</td>
<td>4.20</td>
<td></td>
<td></td>
</tr>
<tr>
<td>y(  3) = 0.58869602269243881E-004</td>
<td>8.06</td>
<td>8.06</td>
<td>3.83</td>
<td></td>
<td></td>
</tr>
<tr>
<td>y(  4) = 0.1175734704403569E-002</td>
<td>7.08</td>
<td>7.08</td>
<td>4.15</td>
<td></td>
<td></td>
</tr>
<tr>
<td>y(  5) = 0.238782323162753E-002</td>
<td>5.83</td>
<td>5.83</td>
<td>3.21</td>
<td></td>
<td></td>
</tr>
<tr>
<td>y(  6) = 0.6244778711349675E-002</td>
<td>5.24</td>
<td>5.24</td>
<td>3.03</td>
<td></td>
<td></td>
</tr>
<tr>
<td>y(  7) = 0.285604371924880E-002</td>
<td>7.34</td>
<td>7.34</td>
<td>4.80</td>
<td></td>
<td></td>
</tr>
<tr>
<td>y(  8) = 0.2849956288075124E-002</td>
<td>7.34</td>
<td>7.34</td>
<td>4.80</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

used components for scd 8 8 8
scd of Y (maximum norm) 5.24 5.24 3.03

using mixed error yields mescd 5.24
using relative error yields scd 3.03

Integration characteristics:

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>number of integration steps</td>
<td>43</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>number of accepted steps</td>
<td>37</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>number of f evaluations</td>
<td>665</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>number of Jacobian evaluations</td>
<td>20</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>number of LU decompositions</td>
<td>168</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

CPU-time used: 0.0029 sec

Figure I.6.1: Example of performing a test run, in which we solve problem HIRES with PSIDE. The experiment was done on an ALPHAserver DS20E, with a 667MHz EV67 processor. We used the Fortran 90 compiler f90 with the optimisation flag -O5.
7 Solver RADAU

7.1 General information

Authors: E. Hairer and G. Wanner
first version: April 23, 1998
last update: January 18, 2002
language: Fortran 77
availability: the code RADAU is freely available (in the public domain)
official link: http://www.unige.ch/~hairer/prog/stiff/radau.f
problem type: ODEs and DAEs of index less than or equal to 3
IVPtestset files: solver: radau.f
driver: radaud.f
auxiliary files: radaua.f (auxiliary linear algebra routines)

7.2 Numerical method

The code RADAU is based on implicit Runge-Kutta methods (Radau IIa) of orders 5, 9 and 13. These methods are L-stable and were firstly implemented in fixed order mode in the code RADAUP [HW96]. It is written for problems of the form \( M y' = f(t, y) \) with a possibly singular matrix \( M \). It is therefore also suitable for the solution of differential-algebraic problems.

7.3 Implementation details

All the implementation techniques described for RADAU5 hold here as well. The code has been provided with an order variation strategy. This is based upon the observation that high order methods perform better than low order methods as soon as the convergence of the simplified Newton iteration is sufficiently fast (a measure of the rate of convergence is the so called contractivity factor) [HW99].

7.4 How to solve test problems with RADAU

Compiling

```
f90 -o dotest radaud.f problem.f radau.f radaua.f report.f,
```

will yield an executable `dotest` that solves the problem, of which the Fortran routines in the format described in Section IV.3 are in the file `problem.f`.

As an example, we perform a test run, in which we solve problem HIRES. Figure I.7.1 shows what one has to do.

References


Figure I.7.1: Example of performing a test run, in which we solve problem HIRES with RADAU. The experiment was done on an ALPHA server DS20E, with a 667MHz EV67 processor. We used the Fortran 90 compiler f90 with the optimization flag -O5.
8 Solver RADAU5

8.1 General information

Authors: E. Hairer and G. Wanner
last update: January 18, 2002
language: Fortran 77
availability: the code RADAU5 is freely available (in the public domain)
official link: http://www.unige.ch/~hairer/prog/stiff/radau5.f
problem type: ODEs and DAEs of index less than or equal to 3
IVPtestset files: solver: radau5.f
driver: radau5d.f
auxiliary files: radaua.f (auxiliary linear algebra routines)

8.2 Numerical method

The code RADAU5 uses an implicit Runge-Kutta method (Radau IIa) of order 5 (three stages) with
step size control and continuous output. It is written for problems of the form \( M y' = f(t, y) \) with
a possibly singular matrix \( M \). It is therefore also suitable for the solution of differential-algebraic
problems.

8.3 Implementation details

Nonlinear systems are solved by a simplified Newton iteration. A similarity transformation on the
inverse of the Butcher array is performed in order to reduce the computational cost associated to
the solution of linear systems (see [HW96], page 121) so that, each time the Jacobian is updated, a
factorization of one real and one complex matrix of the same dimension as that of the continuous
problem is needed.

8.4 How to solve test problems with RADAU5

Compiling

```
f90 -o dotest radau5d.f problem.f radau5.f radaua.f report.f
```

will yield an executable `dotest` that solves the problem, of which the Fortran routines in the format
described in Section IV.3 are in the file `problem.f`

As an example, we perform a test run, in which we solve problem HIRES. Figure 1.8.1 shows what
one has to do.

References

$ f90 -O5 -o dotest radau5d.f hires.f radau5.f radaua.f report.f
$ ./dotest

Test Set for IVP Solvers (release 2.3)

Solving Problem HIRES using RADAU5

User input:

give relative error tolerance: id-4

give absolute error tolerance: id-4

give initial stepsize: id-4

Numerical solution:

<table>
<thead>
<tr>
<th>solution component</th>
<th>scd</th>
<th>mixed</th>
<th>abs</th>
<th>rel</th>
<th>mix - abs,rel</th>
</tr>
</thead>
<tbody>
<tr>
<td>y( 1) = 0.748515246440879E-003</td>
<td>4.94</td>
<td>4.94</td>
<td>1.81</td>
<td></td>
<td></td>
</tr>
<tr>
<td>y( 2) = 0.146491239496945E-003</td>
<td>5.65</td>
<td>5.65</td>
<td>1.81</td>
<td></td>
<td></td>
</tr>
<tr>
<td>y( 3) = 0.610142628065334E-004</td>
<td>5.67</td>
<td>5.67</td>
<td>1.44</td>
<td></td>
<td></td>
</tr>
<tr>
<td>y( 4) = 0.119676321006783E-002</td>
<td>4.68</td>
<td>4.68</td>
<td>1.75</td>
<td></td>
<td></td>
</tr>
<tr>
<td>y( 5) = 0.273188990794999E-002</td>
<td>3.46</td>
<td>3.46</td>
<td>0.84</td>
<td></td>
<td></td>
</tr>
<tr>
<td>y( 6) = 0.734701764327763E-002</td>
<td>2.96</td>
<td>2.96</td>
<td>0.75</td>
<td></td>
<td></td>
</tr>
<tr>
<td>y( 7) = 0.307462088590754E-002</td>
<td>3.65</td>
<td>3.65</td>
<td>1.10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>y( 8) = 0.262537911409241E-002</td>
<td>3.65</td>
<td>3.65</td>
<td>1.10</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

used components for scd = 8 8 8

scd of Y (maximum norm) = 2.96 2.96 0.75

using mixed error yields mescd = 2.96

using relative error yields scd = 0.75

Integration characteristics:

| number of integration steps | 38 |
| number of accepted steps    | 31 |
| number of f evaluations     | 295 |
| number of Jacobian evaluations | 20 |
| number of LU decompositions | 36 |

CPU-time used: 0.0010 sec

Figure I.8.1: Example of performing a test run, in which we solve problem HIRES with RADAU5. The experiment was done on an ALPHAserver DS20E, with a 667MHz EV67 processor. We used the Fortran 90 compiler f90 with the optimization flag -O5.
9 Solver VODE

9.1 General information

Authors: Peter N. Brown, George D. Byrne and Alan C. Hindmarsh
first version: June 15 1989
last update: April 30, 2000
language: Fortran 77
availability: the code VODE is freely available (in the public domain)
official link: http://www.netlib.org/ode/vode.f
problem type: ODE
IVPtestset files: solver: vode.f
driver: voded.f
auxiliary files: vodea.f (auxiliary linear algebra routines)

9.2 Numerical method

The code is based upon linear multistep methods used with variable coefficients (but fixed leading term) to take account for the stepsize change. It allows the use of Adams and BDFs methods to handle both nonstiff and stiff problems [Byr75].

9.3 Implementation details

VODE [BBA89] is a package based on the EPISODE and EPISODEB packages [HB77, BH76], and on the ODEPACK user interface standard [Hin83], with minor modifications. The code may switch between two different techniques, namely functional iteration and the modified Newton method, to solve nonlinear systems at each time-step. Recently, a FORTRAN 90 version of this solver has been made available at the URL http://www.radford.edu/~thompson/vodef90web/.

9.4 How to solve test problems with VODE

Compiling

\texttt{f90 -o dotest voded.f problem.f vode.f vodea.f report.f},

will yield an executable \texttt{dotest} that solves the problem, of which the Fortran routines in the format described in Section IV.3 are in the file \texttt{problem.f}.

As an example, we perform a test run, in which we solve problem HIRES. Figure I.9.1 shows what one has to do.

References


$f90 -O5 -o dtest voded.f hires.f vode.f vodea.f report.f

dotest
Test Set for IVP Solvers (release 2.3)

Solving Problem HIRES using VODE

User input:
give relative error tolerance:
1d-4
give absolute error tolerance:
1d-4

Numerical solution:

<table>
<thead>
<tr>
<th>solution component</th>
<th>scd</th>
<th>mixed</th>
<th>abs</th>
<th>rel</th>
<th>ignore</th>
</tr>
</thead>
<tbody>
<tr>
<td>y( 1) = 0.7405428802164954E-003</td>
<td>5.47</td>
<td>5.47</td>
<td>2.33</td>
<td></td>
<td></td>
</tr>
<tr>
<td>y( 2) = 0.1449232356407335E-003</td>
<td>6.17</td>
<td>6.17</td>
<td>2.33</td>
<td></td>
<td></td>
</tr>
<tr>
<td>y( 3) = 0.5951034500912568E-004</td>
<td>6.21</td>
<td>6.21</td>
<td>1.98</td>
<td></td>
<td></td>
</tr>
<tr>
<td>y( 4) = 0.1182096389331146E-002</td>
<td>5.19</td>
<td>5.19</td>
<td>2.26</td>
<td></td>
<td></td>
</tr>
<tr>
<td>y( 5) = 0.2483586047844519E-002</td>
<td>4.01</td>
<td>4.01</td>
<td>1.39</td>
<td></td>
<td></td>
</tr>
<tr>
<td>y( 6) = 0.6494848234786107E-002</td>
<td>3.59</td>
<td>3.59</td>
<td>1.39</td>
<td></td>
<td></td>
</tr>
<tr>
<td>y( 7) = 0.2954272405089350E-002</td>
<td>3.98</td>
<td>3.98</td>
<td>1.44</td>
<td></td>
<td></td>
</tr>
<tr>
<td>y( 8) = 0.2745727897959102E-002</td>
<td>3.98</td>
<td>3.98</td>
<td>1.44</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

used components for scd 8 8 8
scd of Y (maximum norm) 3.59 3.59 1.39

using mixed error yields mescd 3.59
using relative error yields scd 1.39

Integration characteristics:

number of integration steps 133
number of accepted steps 131
number of f evaluations 191
number of Jacobian evaluations 10
number of LU decompositions 25

CPU-time used: 0.0010 sec

Figure I.9.1: Example of performing a test run, in which we solve problem HIRES with VODE. The experiment was done on an ALPHA Server DS20E, with a 667MHz EV67 processor. We used the Fortran 90 compiler f90 with the optimization flag -O5.

Part II
Problems

This part is the core of the report. All the test problems collected are described. The problems are ordered as ODEs, DAEs and IDEs.
1 Problem HIRES

1.1 General information

This IVP is a stiff system of 8 non-linear Ordinary Differential Equations. It was proposed by Schäfer in 1975 [Sch75]. The name HIRES was given by Hairer & Wanner [HW96]. It refers to ‘High Irradiance Response’, which is described by this ODE. The parallel-IVP-algorithm group of CWI contributed this problem to the test set. The software part of the problem is in the file hires.f available at [MM08].

1.2 Mathematical description of the problem

The problem is of the form

\[ \frac{dy}{dt} = f(y), \quad y(0) = y_0, \]

with

\[ y \in \mathbb{R}^8, \quad 0 \leq t \leq 321.8122 \]

The function \( f \) is defined by

\[
\begin{pmatrix}
-1.71y_1 + 0.43y_2 + 8.32y_3 + 0.0007 \\
1.71y_1 - 8.75y_2 \\
-10.03y_3 + 0.43y_4 + 0.035y_5 \\
8.32y_2 + 1.71y_3 - 1.12y_4 \\
-1.745y_5 + 0.43y_6 + 0.43y_7 \\
-280y_6y_8 + 0.69y_4 + 1.71y_5 - 0.43y_6 + 0.69y_7 \\
280y_6y_8 - 1.81y_7 \\
-280y_6y_8 + 1.81y_7
\end{pmatrix}
\]

The initial vector \( y_0 \) is given by \((1, 0, 0, 0, 0, 0, 0, 0.0057)^T\).

1.3 Origin of the problem

The HIRES problem originates from plant physiology and describes how light is involved in morphogenesis. To be precise, it explains the ‘High Irradiance Responses’ (HIRES) of photomorphogenesis on the basis of phytochrome, by means of a chemical reaction involving eight reagents. It has been promoted as a test problem by Gottwald in [Got77]. The reaction scheme is given in Figure II.1.1.

\( P_r \) and \( P_{fr} \) refer to the red and far-red absorbing form of phytochrome, respectively. They can be bound by two receptors \( X \) and \( X_0 \), partially influenced by the enzyme \( E \). The values of the parameters were taken from [HW96]

\[
\begin{array}{c|c|c|c|c|c|c}
 k_1 & = & 1.71 & k_3 & = & 8.32 & k_5 & = & 0.035 \\
 k_2 & = & 0.43 & k_4 & = & 0.69 & k_6 & = & 8.32 \\
 k_7 & = & 280 & k_8 & = & 0.69 & k^r & = & 0.69 \\
 c_{k_r} & = & 0.0007
\end{array}
\]

For more details, we refer to [Sch75].

Identifying the concentrations of \( P_r \), \( P_{fr} \), \( P_rX \), \( P_{fr}X \), \( P_rX' \), \( P_{fr}X'E \) and \( E \) with \( y_i, i \in \{1, \ldots, 8\} \), respectively, the differential equations mentioned in §1.2 easily follow. See [SL98] for a more detailed description of this modeling process.

The end point of the integration interval, 321.8122, was chosen arbitrarily [Wan98].
1.4 Numerical solution of the problem

Tables II.1.1–II.1.2 and Figures II.1.2–II.1.6 present the reference solution at the end of the integration interval, the run characteristics, the behavior of the solution over (part of) the integration interval and the work-precision diagrams, respectively. The reference solution was computed by RADAU5 on a Cray C90, using double precision, work(1) = uround = 1.01·10⁻¹⁸, rtol = atol = h0 = 1.1·10⁻¹⁸. For the work-precision diagrams, we used: rtol = 10⁻⁵⁺m/⁴, m = 0, 1, ..., 28; atol = rtol; h0 = 10⁻²·rtol for BIMD, GAMD, MEBDFDAE, MEBDFI, RADAU and RADAU5.

Table II.1.1: Reference solution at the end of the integration interval.

| \(y_1\) | 0.737131257325668·10⁻³ | 0.2386356198831331·10⁻² |
| \(y_2\) | 0.1442485726316185·10⁻³ | 0.6238968252742796·10⁻² |
| \(y_3\) | 0.588872974067575·10⁻⁴ | 0.2849988395185769·10⁻² |
| \(y_4\) | 0.1175651343283149·10⁻² | 0.2850001604814231·10⁻² |

References


Table II.1.2: Run characteristics.

<table>
<thead>
<tr>
<th>solver</th>
<th>rtol</th>
<th>atol</th>
<th>h0</th>
<th>mesd</th>
<th>scd</th>
<th>steps</th>
<th>accept</th>
<th>#f</th>
<th>#Jac</th>
<th>#LU</th>
<th>CPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>BIMD</td>
<td>10^-7</td>
<td>10^-7</td>
<td>10^-9</td>
<td>8.42</td>
<td>6.21</td>
<td>48</td>
<td>47</td>
<td>1395</td>
<td>42</td>
<td>48</td>
<td>0.0039</td>
</tr>
<tr>
<td></td>
<td>10^-10</td>
<td>10^-10</td>
<td>10^-12</td>
<td>11.49</td>
<td>9.28</td>
<td>89</td>
<td>89</td>
<td>2854</td>
<td>82</td>
<td>88</td>
<td>0.0088</td>
</tr>
<tr>
<td>DDASSE</td>
<td>10^-7</td>
<td>10^-7</td>
<td></td>
<td>6.02</td>
<td>3.81</td>
<td>380</td>
<td>369</td>
<td>391</td>
<td>32</td>
<td></td>
<td>0.0039</td>
</tr>
<tr>
<td></td>
<td>10^-10</td>
<td>10^-10</td>
<td></td>
<td>8.99</td>
<td>6.78</td>
<td>1160</td>
<td>1148</td>
<td>1557</td>
<td>45</td>
<td></td>
<td>0.0098</td>
</tr>
<tr>
<td>GAMD</td>
<td>10^-7</td>
<td>10^-7</td>
<td>10^-9</td>
<td>8.51</td>
<td>6.00</td>
<td>38</td>
<td>34</td>
<td>2167</td>
<td>33</td>
<td>38</td>
<td>0.0049</td>
</tr>
<tr>
<td></td>
<td>10^-10</td>
<td>10^-10</td>
<td>10^-12</td>
<td>10.26</td>
<td>7.82</td>
<td>55</td>
<td>50</td>
<td>4164</td>
<td>51</td>
<td>55</td>
<td>0.0098</td>
</tr>
<tr>
<td>MEBDFI</td>
<td>10^-7</td>
<td>10^-7</td>
<td>10^-9</td>
<td>6.45</td>
<td>4.24</td>
<td>218</td>
<td>214</td>
<td>767</td>
<td>29</td>
<td>29</td>
<td>0.0029</td>
</tr>
<tr>
<td></td>
<td>10^-10</td>
<td>10^-10</td>
<td>10^-12</td>
<td>9.51</td>
<td>7.30</td>
<td>420</td>
<td>416</td>
<td>1492</td>
<td>46</td>
<td>46</td>
<td>0.0068</td>
</tr>
<tr>
<td>PSIDE-1</td>
<td>10^-7</td>
<td>10^-7</td>
<td></td>
<td>7.24</td>
<td>4.88</td>
<td>68</td>
<td>60</td>
<td>1208</td>
<td>25</td>
<td>252</td>
<td>0.0039</td>
</tr>
<tr>
<td></td>
<td>10^-10</td>
<td>10^-10</td>
<td></td>
<td>11.06</td>
<td>8.85</td>
<td>152</td>
<td>151</td>
<td>2528</td>
<td>35</td>
<td>344</td>
<td>0.0068</td>
</tr>
<tr>
<td>RADAU</td>
<td>10^-7</td>
<td>10^-7</td>
<td>10^-9</td>
<td>7.11</td>
<td>4.91</td>
<td>51</td>
<td>40</td>
<td>985</td>
<td>22</td>
<td>51</td>
<td>0.0020</td>
</tr>
<tr>
<td></td>
<td>10^-10</td>
<td>10^-10</td>
<td>10^-12</td>
<td>10.65</td>
<td>8.03</td>
<td>69</td>
<td>58</td>
<td>1511</td>
<td>29</td>
<td>68</td>
<td>0.0039</td>
</tr>
<tr>
<td>VODE</td>
<td>10^-7</td>
<td>10^-7</td>
<td></td>
<td>6.19</td>
<td>3.98</td>
<td>415</td>
<td>390</td>
<td>608</td>
<td>9</td>
<td>70</td>
<td>0.0029</td>
</tr>
<tr>
<td></td>
<td>10^-10</td>
<td>10^-10</td>
<td></td>
<td>8.75</td>
<td>6.20</td>
<td>933</td>
<td>880</td>
<td>1224</td>
<td>15</td>
<td>134</td>
<td>0.0059</td>
</tr>
</tbody>
</table>

Figure II.1.2: Behavior of the solution over the integration interval.
Figure II.1.3: Work-precision diagram (scd versus CPU-time).
Figure II.1.4: Work-precision diagram (scd versus CPU-time).
Figure 11.5: Work-precision diagram (mescd versus CPU-time).
Figure II.1.6: Work-precision diagram (mescd versus CPU-time).
2 Pollution problem

2.1 General information

This IVP is a stiff system of 20 non-linear Ordinary Differential Equations. It is the chemical reaction part of the air pollution model developed at The Dutch National Institute of Public Health and Environmental Protection (RIVM) and it is described by Verwer in [Ver94]. The parallel-IVP-algorithm group of CWI contributed this problem to the test set.

The software part of the problem is in the file pollu.f available at [MM08].

2.2 Mathematical description of the problem

The problem is of the form

$$\frac{dy}{dt} = f(y), \quad y(0) = y_0,$$

with

$$y \in \mathbb{R}^{20}, \quad 0 \leq t \leq 60.$$  

The function $f$ is defined by

$$f = \begin{pmatrix}
    - & \sum_{j \in \{1,10,14,23,24\}} r_j + \sum_{j \in \{2,3,8,11,12,22,23\}} r_j \\
    -r_2 - r_3 - r_9 - r_{12} + r_1 + r_{21} \\
    -r_{15} + r_1 + r_{17} + r_{19} + r_{22} \\
    -r_2 - r_{16} - r_{17} - r_{23} + r_{15} \\
    -r_3 + 2r_4 + r_6 + r_7 + r_{13} + r_{20} \\
    -r_6 - r_8 - r_{14} - r_{20} + r_3 + 2r_{18} \\
    -r_4 - r_5 - r_6 + r_{13} \\
    r_4 + r_5 + r_6 + r_7 \\
    -r_7 - r_8 \\
    -r_{12} + r_7 + r_9 \\
    -r_9 - r_{10} + r_8 + r_{11} \\
    r_9 \\
    -r_{11} + r_{10} \\
    -r_{13} + r_{12} \\
    r_{14} \\
    -r_{18} - r_{19} + r_{16} \\
    -r_{20} \\
    r_{20} \\
    -r_{21} - r_{22} - r_{24} + r_{23} + r_{25} \\
    -r_{25} + r_{24} \\
\end{pmatrix},$$

where the $r_i$ are auxiliary variables, given in Table II.2.1. The values of the parameters $k_j$ are in Table II.2.2. Finally, the initial vector $y_0$ is given by

$$y_0 = (0, 0.2, 0, 0.04, 0, 0, 0.1, 0.3, 0.01, 0, 0, 0, 0.07, 0, 0, 0, 0.007, 0, 0, 0)^T.$$

2.3 Origin of the problem

The problem is a chemical model consisting of 25 reactions and 20 reacting compounds. Figure II.2.1 shows the reaction scheme. Writing down the reaction velocities $r_j$ for every reaction equation and making the identification in Table II.2.3, which also lists the concentrations at $t = 0$, one arrives at the system of differential equations (II.2.1). The time interval $[0,60]$ represents the behavior of the reactants sufficiently.
### Table II.2.1: Auxiliary variables.

<table>
<thead>
<tr>
<th>$r_1$</th>
<th>$= k_1 \cdot y_1$</th>
<th>$r_{10}$</th>
<th>$= k_{10} \cdot y_{11} \cdot y_1$</th>
<th>$r_{19}$</th>
<th>$= k_{19} \cdot y_{16}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$r_2$</td>
<td>$= k_2 \cdot y_2 \cdot y_4$</td>
<td>$r_{11}$</td>
<td>$= k_{11} \cdot y_{13}$</td>
<td>$r_{20}$</td>
<td>$= k_{20} \cdot y_{17} \cdot y_6$</td>
</tr>
<tr>
<td>$r_3$</td>
<td>$= k_3 \cdot y_5 \cdot y_2$</td>
<td>$r_{12}$</td>
<td>$= k_{12} \cdot y_{10} \cdot y_2$</td>
<td>$r_{21}$</td>
<td>$= k_{21} \cdot y_{19}$</td>
</tr>
<tr>
<td>$r_4$</td>
<td>$= k_4 \cdot y_7$</td>
<td>$r_{13}$</td>
<td>$= k_{13} \cdot y_{14}$</td>
<td>$r_{22}$</td>
<td>$= k_{22} \cdot y_{19}$</td>
</tr>
<tr>
<td>$r_5$</td>
<td>$= k_5 \cdot y_7$</td>
<td>$r_{14}$</td>
<td>$= k_{14} \cdot y_{1} \cdot y_6$</td>
<td>$r_{23}$</td>
<td>$= k_{23} \cdot y_{1} \cdot y_4$</td>
</tr>
<tr>
<td>$r_6$</td>
<td>$= k_6 \cdot y_7 \cdot y_6$</td>
<td>$r_{15}$</td>
<td>$= k_{15} \cdot y_3$</td>
<td>$r_{24}$</td>
<td>$= k_{24} \cdot y_{19} \cdot y_1$</td>
</tr>
<tr>
<td>$r_7$</td>
<td>$= k_7 \cdot y_9$</td>
<td>$r_{16}$</td>
<td>$= k_{16} \cdot y_4$</td>
<td>$r_{25}$</td>
<td>$= k_{25} \cdot y_{20}$</td>
</tr>
<tr>
<td>$r_8$</td>
<td>$= k_8 \cdot y_{9} \cdot y_6$</td>
<td>$r_{17}$</td>
<td>$= k_{17} \cdot y_4$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$r_9$</td>
<td>$= k_9 \cdot y_{11} \cdot y_2$</td>
<td>$r_{18}$</td>
<td>$= k_{18} \cdot y_{16}$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table II.2.2: Parameter values.

<table>
<thead>
<tr>
<th>$k_1$</th>
<th>$= 0.350$</th>
<th>$k_{10}$</th>
<th>$= 0.900 \cdot 10^4$</th>
<th>$k_{19}$</th>
<th>$= 0.444 \cdot 10^{12}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$k_2$</td>
<td>$= 0.266 \cdot 10^2$</td>
<td>$k_{11}$</td>
<td>$= 0.220 \cdot 10^{-1}$</td>
<td>$k_{20}$</td>
<td>$= 0.124 \cdot 10^4$</td>
</tr>
<tr>
<td>$k_3$</td>
<td>$= 0.123 \cdot 10^3$</td>
<td>$k_{12}$</td>
<td>$= 0.120 \cdot 10^5$</td>
<td>$k_{21}$</td>
<td>$= 0.210 \cdot 10$</td>
</tr>
<tr>
<td>$k_4$</td>
<td>$= 0.860 \cdot 10^{-3}$</td>
<td>$k_{13}$</td>
<td>$= 0.188 \cdot 10$</td>
<td>$k_{22}$</td>
<td>$= 0.578 \cdot 10$</td>
</tr>
<tr>
<td>$k_5$</td>
<td>$= 0.820 \cdot 10^{-3}$</td>
<td>$k_{14}$</td>
<td>$= 0.163 \cdot 10^5$</td>
<td>$k_{23}$</td>
<td>$= 0.474 \cdot 10^{-1}$</td>
</tr>
<tr>
<td>$k_6$</td>
<td>$= 0.150 \cdot 10^5$</td>
<td>$k_{15}$</td>
<td>$= 0.480 \cdot 10^7$</td>
<td>$k_{24}$</td>
<td>$= 0.178 \cdot 10^4$</td>
</tr>
<tr>
<td>$k_7$</td>
<td>$= 0.130 \cdot 10^{-3}$</td>
<td>$k_{16}$</td>
<td>$= 0.350 \cdot 10^{-3}$</td>
<td>$k_{25}$</td>
<td>$= 0.312 \cdot 10$</td>
</tr>
<tr>
<td>$k_8$</td>
<td>$= 0.240 \cdot 10^6$</td>
<td>$k_{17}$</td>
<td>$= 0.175 \cdot 10^{-1}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$k_9$</td>
<td>$= 0.165 \cdot 10^5$</td>
<td>$k_{18}$</td>
<td>$= 0.100 \cdot 10^6$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*†* Notice that this constant has a typing error in [Ver94].

### Figure II.2.1: Reaction scheme.

1. NO2 $\rightarrow$ NO+O3P
2. NO+O3 $\rightarrow$ NO2
3. HO2+NO $\rightarrow$ NO2+OH
4. HCHO $\rightarrow$ 2 HO2+CO
5. HCHO $\rightarrow$ CO
6. HCHO+OH $\rightarrow$ HO2+CO
7. ALD $\rightarrow$ MEO2+HO2+CO
8. ALD+OH $\rightarrow$ C2O3
9. C2O3+NO $\rightarrow$ NO2+MEO2+CO2
10. C2O3+NO2 $\rightarrow$ PAN
11. PAN $\rightarrow$ C2O3+NO2
12. MEO2+NO $\rightarrow$ CH3O+NO2
13. CH3O $\rightarrow$ HCHO+HO2
14. NO2+OH $\rightarrow$ HNO3
15. O3P $\rightarrow$ O3
16. O3 $\rightarrow$ O1D
17. O3 $\rightarrow$ O3P
18. O1D $\rightarrow$ 2 OH
19. O1D $\rightarrow$ O3P
20. SO2+OH $\rightarrow$ SO4+HO2
21. NO3 $\rightarrow$ NO
22. NO3 $\rightarrow$ NO2+O3P
23. NO2+O3 $\rightarrow$ NO3
24. NO3+NO2 $\rightarrow$ N2O5
25. N2O5 $\rightarrow$ NO3+NO2
### Table II.2.3: Identification of variables with species. The square brackets ‘[ ]’ denote concentrations.

<table>
<thead>
<tr>
<th>variable</th>
<th>species</th>
<th>initial value</th>
<th>variable</th>
<th>species</th>
<th>initial value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$y_1$</td>
<td>[NO$_2$]</td>
<td>0</td>
<td>$y_{11}$</td>
<td>[C2O3]</td>
<td>0</td>
</tr>
<tr>
<td>$y_2$</td>
<td>[NO]</td>
<td>0.2</td>
<td>$y_{12}$</td>
<td>[CO2]</td>
<td>0</td>
</tr>
<tr>
<td>$y_3$</td>
<td>[O3P]</td>
<td>0</td>
<td>$y_{13}$</td>
<td>[PAN]</td>
<td>0</td>
</tr>
<tr>
<td>$y_4$</td>
<td>[O3]</td>
<td>0.04</td>
<td>$y_{14}$</td>
<td>[CH3O]</td>
<td>0</td>
</tr>
<tr>
<td>$y_5$</td>
<td>[HO2]</td>
<td>0</td>
<td>$y_{15}$</td>
<td>[HNO3]</td>
<td>0</td>
</tr>
<tr>
<td>$y_6$</td>
<td>[OH]</td>
<td>0</td>
<td>$y_{16}$</td>
<td>[O1D]</td>
<td>0</td>
</tr>
<tr>
<td>$y_7$</td>
<td>[HCHO]</td>
<td>0.1</td>
<td>$y_{17}$</td>
<td>[SO2]</td>
<td>0.007</td>
</tr>
<tr>
<td>$y_8$</td>
<td>[CO]</td>
<td>0.3</td>
<td>$y_{18}$</td>
<td>[SO4]</td>
<td>0</td>
</tr>
<tr>
<td>$y_9$</td>
<td>[ALD]</td>
<td>0.01</td>
<td>$y_{19}$</td>
<td>[NO3]</td>
<td>0</td>
</tr>
<tr>
<td>$y_{10}$</td>
<td>[MEO2]</td>
<td>0</td>
<td>$y_{20}$</td>
<td>[N2O5]</td>
<td>0</td>
</tr>
</tbody>
</table>

### Table II.2.4: Reference solution at the end of the integration interval.

<table>
<thead>
<tr>
<th>variable</th>
<th>solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>$y_1$</td>
<td>0.5646255480022769 $\times 10^{-1}$</td>
</tr>
<tr>
<td>$y_2$</td>
<td>0.134248413042339</td>
</tr>
<tr>
<td>$y_3$</td>
<td>0.4139734331099427 $\times 10^{-8}$</td>
</tr>
<tr>
<td>$y_4$</td>
<td>0.5523140207484339 $\times 10^{-2}$</td>
</tr>
<tr>
<td>$y_5$</td>
<td>0.2018977262302196 $\times 10^{-6}$</td>
</tr>
<tr>
<td>$y_6$</td>
<td>0.4144514863493966 $\times 10^{-6}$</td>
</tr>
<tr>
<td>$y_7$</td>
<td>0.778429118997964 $\times 10^{-1}$</td>
</tr>
<tr>
<td>$y_8$</td>
<td>0.3245075353396018</td>
</tr>
<tr>
<td>$y_9$</td>
<td>0.749401383850106 $\times 10^{-2}$</td>
</tr>
<tr>
<td>$y_{10}$</td>
<td>0.162293157301561 $\times 10^{-7}$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>variable</th>
<th>solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>$y_{11}$</td>
<td>0.113586383257075 $\times 10^{-7}$</td>
</tr>
<tr>
<td>$y_{12}$</td>
<td>0.23050975721339 $\times 10^{-2}$</td>
</tr>
<tr>
<td>$y_{13}$</td>
<td>0.208716282798630 $\times 10^{-3}$</td>
</tr>
<tr>
<td>$y_{14}$</td>
<td>0.1396921016840158 $\times 10^{-4}$</td>
</tr>
<tr>
<td>$y_{15}$</td>
<td>0.8964884856898295 $\times 10^{-2}$</td>
</tr>
<tr>
<td>$y_{16}$</td>
<td>0.4352846369330103 $\times 10^{-17}$</td>
</tr>
<tr>
<td>$y_{17}$</td>
<td>0.689921969626405 $\times 10^{-2}$</td>
</tr>
<tr>
<td>$y_{18}$</td>
<td>0.1007903037365394 $\times 10^{-3}$</td>
</tr>
<tr>
<td>$y_{19}$</td>
<td>0.1772146513969984 $\times 10^{-5}$</td>
</tr>
<tr>
<td>$y_{20}$</td>
<td>0.568293292316392 $\times 10^{-4}$</td>
</tr>
</tbody>
</table>

### 2.4 Numerical solution of the problem

Tables II.2.4–II.2.5 and Figures II.2.2–II.2.6 present the reference solution at the end of the integration interval, the run characteristics, the behavior of the solution over the interval [0,12] and the work-precision diagrams, respectively. The reference solution was computed by RADAU5 on a Cray C90, using double precision, work(1) = uround = 1.01 $\times 10^{-10}$, rtol = atol = h0 = 1.1 $\times 10^{-18}$. For the work-precision diagrams, we used: rtol = $10^{-(5+m/4)}$, m = 0,1,\ldots,32; atol = rtol; h0 = rtol for BIMD, GAMD, MEBDFDAE, MEBDFI, RADAU and RADAU5.

### References


### Table II.2.5: Run characteristics.

<table>
<thead>
<tr>
<th>solver</th>
<th>rtol</th>
<th>atol</th>
<th>h0</th>
<th>mescd</th>
<th>scd</th>
<th>steps</th>
<th>accept</th>
<th>#f</th>
<th>#Jac</th>
<th>#LU</th>
<th>CPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>BIMD</td>
<td>10^{-7}</td>
<td>10^{-7}</td>
<td>10^{-7}</td>
<td>9.25</td>
<td>5.63</td>
<td>25</td>
<td>25</td>
<td>572</td>
<td>22</td>
<td>25</td>
<td>0.0039</td>
</tr>
<tr>
<td></td>
<td>10^{-10}</td>
<td>10^{-10}</td>
<td>10^{-10}</td>
<td>11.73</td>
<td>8.73</td>
<td>41</td>
<td>41</td>
<td>1257</td>
<td>27</td>
<td>41</td>
<td>0.0107</td>
</tr>
<tr>
<td>DDASSL</td>
<td>10^{-7}</td>
<td>10^{-7}</td>
<td>10^{-7}</td>
<td>5.94</td>
<td>4.13</td>
<td>135</td>
<td>135</td>
<td>188</td>
<td>23</td>
<td></td>
<td>0.0039</td>
</tr>
<tr>
<td></td>
<td>10^{-10}</td>
<td>10^{-10}</td>
<td>10^{-10}</td>
<td>9.04</td>
<td>5.91</td>
<td>536</td>
<td>532</td>
<td>669</td>
<td>38</td>
<td></td>
<td>0.0107</td>
</tr>
<tr>
<td>GAMD</td>
<td>10^{-7}</td>
<td>10^{-7}</td>
<td>10^{-7}</td>
<td>8.16</td>
<td>6.31</td>
<td>23</td>
<td>23</td>
<td>625</td>
<td>23</td>
<td>23</td>
<td>0.0049</td>
</tr>
<tr>
<td></td>
<td>10^{-10}</td>
<td>10^{-10}</td>
<td>10^{-10}</td>
<td>11.35</td>
<td>5.36</td>
<td>36</td>
<td>36</td>
<td>1401</td>
<td>36</td>
<td>36</td>
<td>0.0098</td>
</tr>
<tr>
<td>MEBDFI</td>
<td>10^{-7}</td>
<td>10^{-7}</td>
<td>10^{-7}</td>
<td>8.46</td>
<td>6.46</td>
<td>120</td>
<td>118</td>
<td>391</td>
<td>20</td>
<td>20</td>
<td>0.0039</td>
</tr>
<tr>
<td></td>
<td>10^{-10}</td>
<td>10^{-10}</td>
<td>10^{-10}</td>
<td>11.45</td>
<td>9.32</td>
<td>235</td>
<td>235</td>
<td>763</td>
<td>33</td>
<td>33</td>
<td>0.0078</td>
</tr>
<tr>
<td>PSIDE-1</td>
<td>10^{-7}</td>
<td>10^{-7}</td>
<td>10^{-7}</td>
<td>7.51</td>
<td>4.84</td>
<td>31</td>
<td>29</td>
<td>465</td>
<td>9</td>
<td>124</td>
<td>0.0049</td>
</tr>
<tr>
<td></td>
<td>10^{-10}</td>
<td>10^{-10}</td>
<td>10^{-10}</td>
<td>10.64</td>
<td>8.04</td>
<td>63</td>
<td>62</td>
<td>970</td>
<td>12</td>
<td>188</td>
<td>0.0098</td>
</tr>
<tr>
<td>RADAU</td>
<td>10^{-7}</td>
<td>10^{-7}</td>
<td>10^{-7}</td>
<td>5.59</td>
<td>3.78</td>
<td>32</td>
<td>29</td>
<td>227</td>
<td>21</td>
<td>32</td>
<td>0.0029</td>
</tr>
<tr>
<td></td>
<td>10^{-10}</td>
<td>10^{-10}</td>
<td>10^{-10}</td>
<td>10.00</td>
<td>7.75</td>
<td>35</td>
<td>35</td>
<td>449</td>
<td>21</td>
<td>35</td>
<td>0.0049</td>
</tr>
<tr>
<td>VODE</td>
<td>10^{-7}</td>
<td>10^{-7}</td>
<td>10^{-7}</td>
<td>6.61</td>
<td>3.32</td>
<td>149</td>
<td>149</td>
<td>208</td>
<td>4</td>
<td>27</td>
<td>0.0029</td>
</tr>
<tr>
<td></td>
<td>10^{-10}</td>
<td>10^{-10}</td>
<td>10^{-10}</td>
<td>8.79</td>
<td>4.78</td>
<td>393</td>
<td>375</td>
<td>528</td>
<td>7</td>
<td>61</td>
<td>0.0059</td>
</tr>
</tbody>
</table>
Figure II.2.2: Behavior of the solution over the interval $[0,12]$. 
Figure II.2.3: Work-precision diagram (scd versus CPU-time).
Figure II.2.4: Work-precision diagram (soc versus CPU-time).
Figure II.2.5: Work-precision diagram (mescd versus CPU-time).
Figure II.2.6: Work-precision diagram (mescd versus CPU-time).
3 Ring modulator

3.1 General information

The type of the problem depends on the parameter $C_s$. If $C_s \neq 0$, then it is a stiff system of 15 non-linear ordinary differential equations. For $C_s = 0$ we have a DAE of index 2, consisting of 11 differential equations and 4 algebraic equations. The numerical results presented here refer to $C_s = 2 \cdot 10^{-12}$. The problem has been taken from [KRS92], where the approach of Horneber [Hor76] is followed. The parallel-IVP-algorithm group of CWI contributed this problem to the test set.

The software part of the problem is in the file `ringmod.f` available at [MM08].

3.2 Mathematical description of the problem

For the ODE case, the problem is of the form

\[ \frac{dy}{dt} = f(t,y), \quad y(0) = y_0, \]

with

\[ y \in \mathbb{R}^{15}, \quad 0 \leq t \leq 10^{-3}. \]

The function $f$ is defined by

\[
\begin{align*}
    f(t,y) = & \begin{pmatrix}
        C^{-1}(y_8 - 0.5y_{10} + 0.5y_{11} + y_{14} - R^{-1}y_1) \\
        C^{-1}(y_9 - 0.5y_{12} + 0.5y_{13} + y_{15} - R^{-1}y_2) \\
        C^{-1}(-y_{11} + q(U_{D1}) - q(U_{D4})) \\
        C^{-1}(y_{12} + q(U_{D1}) - q(U_{D3})) \\
        C^{-1}(-y_{13} - q(U_{D2}) + q(U_{D4})) \\
        C^{-1}(-R^{-1}y_7 + q(U_{D1}) + q(U_{D2}) - q(U_{D3}) - q(U_{D4})) \\
    \end{pmatrix} \\
    \begin{pmatrix}
        -L_k^{-1}y_1 \\
        -L_k^{-1}y_2 \\
        L_{g1}^{-1}(0.5y_1 - y_3 - R_{g2}y_{10}) \\
        L_{g1}^{-1}(-0.5y_1 + y_4 - R_{g3}y_{11}) \\
        L_{g1}^{-1}(0.5y_2 - y_3 - R_{g2}y_{12}) \\
        L_{g1}^{-1}(-0.5y_2 + y_6 - R_{g3}y_{13}) \\
        L_{g1}^{-1}(-y_1 + U_{in1}(t) - (R_t + R_{g1})y_{14}) \\
        L_{g1}^{-1}(-y_2 - (R_c + R_{g1})y_{15}) \\
    \end{pmatrix}.
\end{align*}
\]

The auxiliary functions $U_{D1}, U_{D2}, U_{D3}, U_{D4}, q, U_{in1}$ and $U_{in2}$ are given by

\[
\begin{align*}
    U_{D1} &= y_3 - y_5 - y_7 - U_{in2}(t), \\
    U_{D2} &= -y_4 + y_6 - y_7 - U_{in2}(t), \\
    U_{D3} &= y_4 + y_5 + y_7 + U_{in2}(t), \\
    U_{D4} &= -y_3 - y_6 + y_7 + U_{in2}(t), \\
    q(U) &= \gamma(e^{8U} - 1), \\
    U_{in1}(t) &= 0.5\sin(2000\pi t), \\
    U_{in2}(t) &= 2\sin(20000\pi t).
\end{align*}
\]

The values of the parameters are:
The initial vector $y_0$ is given by

$$y_0 = (0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0)^T.$$ 

The definition of the function $q(U)$ in (II.3.2) may cause overflow if $\delta U$ becomes too large. In the Fortran subroutine that defines $f$, we set $IERR=-1$ if $\delta U > 300$ to prevent this situation. See page IV-ix of the description of the software part of the test set for more details on $IERR$.

### 3.3 Origin of the problem

The problem originates from electrical circuit analysis. It describes the behavior of the ring modulator, of which the circuit diagram is given in Figure II.3.1. Given a low-frequency signal $U_{in1}$ and a high-frequency signal $U_{in2}$, the ring modulator produces a mixed signal in $U_2$.

Every capacitor in the diagram leads to a differential equation:

$$CU = I.$$
Applying Kirchhoff's Current Law yields the following differential equations:

\[
\begin{align*}
CU_1 &= I_1 - 0.5I_3 + 0.5I_4 + I_7 - R^{-1}U_1, \\
CU_2 &= I_2 - 0.5I_5 + 0.5I_6 + I_8 - R^{-1}U_2, \\
C_iU_3 &= I_3 - q(U_{D1}) + q(U_{D4}), \\
C_iU_4 &= -I_4 + q(U_{D2}) - q(U_{D3}), \\
C_sU_5 &= I_5 + q(U_{D1}) - q(U_{D3}), \\
C_sU_6 &= -I_6 - q(U_{D2}) + q(U_{D4}), \\
C_pU_7 &= -R_p^{-1}U_7 + q(U_{D1}) + q(U_{D2}) - q(U_{D3}) - q(U_{D4}), \\
\end{align*}
\]

where \(U_{D1}, U_{D2}, U_{D3}\) and \(U_{D4}\) stand for:

\[
\begin{align*}
U_{D1} &= U_3 - U_5 - U_7 - U_{in2}, \\
U_{D2} &= -U_4 + U_6 - U_7 - U_{in2}, \\
U_{D3} &= U_4 + U_5 + U_7 + U_{in2}, \\
U_{D4} &= -U_3 - U_6 + U_7 + U_{in2}. \\
\end{align*}
\]

The diode function \(q\) is given by

\[
q(U) = \gamma(e^{\delta U} - 1),
\]

where \(\gamma\) and \(\delta\) are fixed constants.

Every inductor leads to a differential equation as well:

\[
L \dot{I} = U.
\]

Applying Kirchhoff's Voltage Law to closed loops that contains an inductor, results in another 8 differential equations:

\[
\begin{align*}
L_h \dot{I}_1 &= -U_1, \\
L_h \dot{I}_2 &= -U_2, \\
L_{21} \dot{I}_3 &= 0.5U_1 - U_3 - R_{21}I_3, \\
L_{31} \dot{I}_4 &= -0.5U_1 + U_4 - R_{31}I_4, \\
L_{22} \dot{I}_5 &= 0.5U_2 - U_5 - R_{22}I_5, \\
L_{32} \dot{I}_6 &= -0.5U_2 + U_6 - R_{32}I_6, \\
L_{41} \dot{I}_7 &= -U_1 + U_{in1} - (R_i + R_{g1})I_7, \\
L_{42} \dot{I}_8 &= -U_2 - (R_c + R_{g1})I_8. \\
\end{align*}
\]

Initially, all voltages and currents are zero.

Identifying the voltages with \(y_1, \ldots, y_7\) and the currents with \(y_8, \ldots, y_{15}\), we obtain the 15 differential equations (II.3.1). From the plot of \(y_2 = U_2\) in Figure II.3.2 we see how the low and high frequency input signals are mixed by the ring modulator.

### 3.4 Numerical solution of the problem

Tables II.3.2–II.3.3 and Figures II.3.2–II.3.7 present the reference solution at the end of the integration interval, the run characteristics, the behavior of the solution over the integration interval and the work-precision diagrams, respectively. The reference solution was computed using PSIDE with \(\texttt{atol} = \texttt{rtol} = 10^{-13}\). For the work-precision diagrams, we used: \(\texttt{rtol} = 10^{-(4 + m/4)}\), \(m = 0, 1, \ldots, 32\); \(\texttt{atol} = \texttt{rtol}\); \(h0 = 10^{-2}\cdot\texttt{rtol}\) for BIMD, GAMD, MEBDFDAE, MEBDFI, RADAU and RADAU5. The failed runs are in Table II.3.1; listed are the name of the solver that failed, for which values of \(m\) this happened, and the reason for failing.
Table II.3.1: Failed runs.

<table>
<thead>
<tr>
<th>solver</th>
<th>m</th>
<th>reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>RAD A U</td>
<td>0, 1, ..., 26</td>
<td>solver cannot handle IERR=-1.</td>
</tr>
<tr>
<td>RAD A U5</td>
<td>0, 1, ..., 9</td>
<td>solver cannot handle IERR=-1.</td>
</tr>
<tr>
<td>V ODE</td>
<td>0</td>
<td>solver cannot handle IERR=-1.</td>
</tr>
<tr>
<td>V ODE</td>
<td>2</td>
<td>error test failed repeatedly.</td>
</tr>
</tbody>
</table>

Table II.3.2: Reference solution at the end of the integration interval.

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$y_1$</td>
<td>$-0.2339057358486745 \cdot 10^{-1}$</td>
<td>$y_9$</td>
</tr>
<tr>
<td>$y_2$</td>
<td>$-0.7367485485540825 \cdot 10^{-2}$</td>
<td>$y_{10}$</td>
</tr>
<tr>
<td>$y_3$</td>
<td>$0.258295670929169$</td>
<td>$y_{11}$</td>
</tr>
<tr>
<td>$y_4$</td>
<td>$-0.4064465721283450$</td>
<td>$y_{12}$</td>
</tr>
<tr>
<td>$y_5$</td>
<td>$-0.403945665149794$</td>
<td>$y_{13}$</td>
</tr>
<tr>
<td>$y_6$</td>
<td>$0.2607066765422943$</td>
<td>$y_{14}$</td>
</tr>
<tr>
<td>$y_7$</td>
<td>$0.11067610861269975$</td>
<td>$y_{15}$</td>
</tr>
<tr>
<td>$y_8$</td>
<td>$0.2939043432435506 \cdot 10^{-6}$</td>
<td></td>
</tr>
</tbody>
</table>

Table II.3.3: Run characteristics.

<table>
<thead>
<tr>
<th>solver</th>
<th>rtol</th>
<th>atol</th>
<th>h0</th>
<th>mescd</th>
<th>sed</th>
<th>steps</th>
<th>accept</th>
<th>#f</th>
<th>#Jac</th>
<th>#LU</th>
<th>CPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>BIMD</td>
<td>$10^{-1}$</td>
<td>$10^{-4}$</td>
<td>$10^{-6}$</td>
<td>2.89</td>
<td>2.20</td>
<td>19415</td>
<td>19089</td>
<td>455877</td>
<td>17614</td>
<td>19127</td>
<td>3.0793</td>
</tr>
<tr>
<td></td>
<td>$10^{-7}$</td>
<td>$10^{-7}$</td>
<td>$10^{-9}$</td>
<td>7.08</td>
<td>6.28</td>
<td>26590</td>
<td>25880</td>
<td>824318</td>
<td>25865</td>
<td>26585</td>
<td>5.5886</td>
</tr>
<tr>
<td>DDASSL</td>
<td>$10^{-4}$</td>
<td>$10^{-4}$</td>
<td></td>
<td>1.18</td>
<td>0.49</td>
<td>88627</td>
<td>86091</td>
<td>116778</td>
<td>3538</td>
<td></td>
<td>1.4230</td>
</tr>
<tr>
<td></td>
<td>$10^{-7}$</td>
<td>$10^{-7}$</td>
<td></td>
<td>3.22</td>
<td>2.53</td>
<td>252827</td>
<td>249239</td>
<td>318196</td>
<td>7777</td>
<td></td>
<td>4.0123</td>
</tr>
<tr>
<td>GAMD</td>
<td>$10^{-4}$</td>
<td>$10^{-4}$</td>
<td>$10^{-6}$</td>
<td>2.34</td>
<td>1.65</td>
<td>12420</td>
<td>11264</td>
<td>474866</td>
<td>11264</td>
<td>12420</td>
<td>2.7572</td>
</tr>
<tr>
<td></td>
<td>$10^{-7}$</td>
<td>$10^{-7}$</td>
<td>$10^{-9}$</td>
<td>6.11</td>
<td>5.42</td>
<td>18798</td>
<td>16913</td>
<td>1049423</td>
<td>16909</td>
<td>18793</td>
<td>6.0502</td>
</tr>
<tr>
<td>MEBDFI</td>
<td>$10^{-4}$</td>
<td>$10^{-4}$</td>
<td>$10^{-6}$</td>
<td>2.54</td>
<td>1.85</td>
<td>61426</td>
<td>61208</td>
<td>201890</td>
<td>5374</td>
<td>5374</td>
<td>1.6416</td>
</tr>
<tr>
<td></td>
<td>$10^{-7}$</td>
<td>$10^{-7}$</td>
<td>$10^{-9}$</td>
<td>5.28</td>
<td>4.59</td>
<td>148609</td>
<td>148298</td>
<td>483689</td>
<td>12471</td>
<td>12471</td>
<td>3.9831</td>
</tr>
<tr>
<td>PSIDE-1</td>
<td>$10^{-4}$</td>
<td>$10^{-4}$</td>
<td></td>
<td>1.29</td>
<td>0.60</td>
<td>9791</td>
<td>8241</td>
<td>267721</td>
<td>6834</td>
<td>38184</td>
<td>1.6709</td>
</tr>
<tr>
<td></td>
<td>$10^{-7}$</td>
<td>$10^{-7}$</td>
<td></td>
<td>5.21</td>
<td>4.53</td>
<td>55345</td>
<td>45636</td>
<td>886724</td>
<td>3984</td>
<td>11508</td>
<td>5.4696</td>
</tr>
<tr>
<td>RAD A U5</td>
<td>$10^{-7}$</td>
<td>$10^{-7}$</td>
<td>$10^{-9}$</td>
<td>4.49</td>
<td>3.80</td>
<td>102515</td>
<td>93113</td>
<td>545282</td>
<td>12316</td>
<td>54746</td>
<td>3.7742</td>
</tr>
<tr>
<td>V ODE</td>
<td>$10^{-7}$</td>
<td>$10^{-7}$</td>
<td></td>
<td>2.84</td>
<td>2.15</td>
<td>217383</td>
<td>207569</td>
<td>261366</td>
<td>3605</td>
<td>22598</td>
<td>2.4019</td>
</tr>
</tbody>
</table>
References


Figure II.3.2: Behavior of the first eight solution components solution over the integration interval.
Figure II.3.3: Behavior of the last seven solution components solution over the integration interval.
Figure II.3.4: Work-precision diagram (sec versus CPU-time).
Figure II.3.5: Work-precision diagram (scd versus CPU-time).
Figure II.3.6: Work-precision diagram (mescd versus CPU-time).
Figure II.3.7: Work-precision diagram (maxed versus CPU-time).
ODE - Ring modulator
4 Medical Akzo Nobel problem

4.1 General information

The problem consists of 2 partial differential equations. Semi-discretization of this system yields a stiff ODE. The parallel-IVP-algorithm group of CWI contributed this problem to the test set in collaboration with R. van der Hout from Akzo Nobel Central Research.

The software part of the problem is in the file medakzo.f available at [MM08].

4.2 Mathematical description of the problem

The problem is of the form
\[
\frac{dy}{dt} = f(t, y), \quad y(0) = g, \quad (II.4.1)
\]
with
\[y \in \mathbb{R}^{2N}, \quad 0 \leq t \leq 20.\]

Here, the integer \(N\) is a user-supplied parameter. The function \(f\) is given by
\[
f_{2j-1} = \alpha_j \frac{y_{2j+1} - y_{2j-3}}{2\Delta\zeta} + \beta_j \frac{y_{2j-3} - 2y_{2j-1} + y_{2j+1}}{(\Delta\zeta)^2} - ky_{2j-1}y_{2j},
\]
\[
f_{2j} = -ky_{2j}y_{2j-1},
\]
where
\[
\alpha_j = \frac{2(j\Delta\zeta - 1)^3}{c^2}, \quad \beta_j = \frac{(j\Delta\zeta - 1)^4}{c^2}.
\]

Here, \(j\) ranges from 1 to \(N\), \(\Delta\zeta = \frac{1}{N}\), \(y_{-1}(t) = \phi(t)\), \(y_{2N+1} = y_{2N-1}\) and \(g \in \mathbb{R}^{2N}\) is given by
\[
g = (0, v_0, 0, v_0, \ldots, 0, v_0)^T.
\]

The function \(\phi\) is given by
\[
\phi(t) = \begin{cases} 2 & \text{for } t \in (0, 5], \\
0 & \text{for } t \in (5, 20].
\end{cases}
\]

which means that \(f\) undergoes a discontinuity in time at \(t = 5\). Suitable values for the parameters \(k\), \(v_0\) and \(c\) are 100, 1 and 4, respectively.

4.3 Origin of the problem

The Akzo Nobel research laboratories formulated this problem in their study of the penetration of radio-labeled antibodies into a tissue that has been infected by a tumor [Hon94]. This study was carried out for diagnostic as well as therapeutic purposes.

Let us consider a reaction diffusion system in one spatial dimension:
\[
\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} - ku, \quad (II.4.2)
\]
\[
\frac{\partial v}{\partial t} = -kv, \quad (II.4.3)
\]
which originates from the chemical reaction

\[ A + B \xrightarrow{k} C. \]

Here \( A \), the radio-labeled antibody, reacts with substrate \( B \), the tissue with the tumor, and \( k \) denotes the rate constant. The concentrations of \( A \) and \( B \) are denoted by \( u \) and \( v \), respectively. In the derivation of the equations (II.4.2) and (II.4.3) it was assumed that the reaction is governed by mass action kinetics and in addition that the chemical \( A \) is mobile while \( B \) is immobile.

Consider a clean semi-infinite slab, in which the substrate \( B \) is uniformly distributed. When the slab is exposed at its surface to the chemical \( A \), this chemical starts to penetrate into the slab.

To model this penetration, the equations (II.4.2) and (II.4.3) are considered in the strip

\[ S_T = \{(x,t) : 0 < x < \infty, \ 0 < t < T\} \]

for some \( T \), along with the following initial and boundary conditions:

\[ u(x,0) = 0, \ v(x,0) = v_0 \quad \text{for} \ x > 0, \]

where \( v_0 \) is a constant, and

\[ u(0,t) = \phi(t) \quad \text{for} \ 0 < t < T. \]

In order to solve the problem numerically, we transform the variable \( x \) in such a way that the semi-infinite slab is transformed into a finite one. A suitable transformation is provided by the following special family of M"{o}bius transformations:

\[ \zeta = \frac{x}{x+c}, \quad \text{with} \ c > 0. \]

Each transformation in this class transforms \( S_T \) into the slab:

\[ \{(\zeta,t) : 0 < \zeta < 1, \ 0 < t < T\}. \]

In terms of \( \zeta \) the problem now reads:

\[ \frac{\partial u}{\partial t} = \frac{(\zeta - 1)^4 \partial^2 u}{c^2} + \frac{2(\zeta - 1)^3 \partial u}{c^3} - kw, \quad \text{(II.4.4)} \]

\[ \frac{\partial v}{\partial t} = -kw, \quad \text{(II.4.5)} \]

with initial conditions

\[ u(\zeta,0) = 0, \ v(\zeta,0) = v_0 \quad \text{for} \ \zeta > 0, \quad \text{(II.4.6)} \]

and boundary conditions

\[ u(0,t) = \phi(t), \quad \frac{\partial u}{\partial \zeta}(1,t) = 0 \quad \text{for} \ 0 < t < T. \quad \text{(II.4.7)} \]

The last boundary condition is derived from \( \frac{\partial u}{\partial \zeta}(\infty,t) = 0 \).

The system consisting of (II.4.4), (II.4.5), (II.4.6) and (II.4.7) will be written as a system of ordinary differential equations by using the method of lines, i.e. by discretizing the spatial derivatives. We use the uniform grid \( \{\zeta_j\}_{j=1,...,N} \) defined by:

\[ \zeta_j = j \cdot \Delta \zeta, \quad j = 1, \ldots, N, \quad \Delta \zeta = \frac{1}{N}. \]
Let $u_j$ and $v_j$ denote the approximations of $u(\zeta_j, t)$ and $v(\zeta_j, t)$, respectively. Obviously, $u_j$ and $v_j$ are functions of $t$. In terms of the function $u_j$, our choices for the discretization of the spatial first and second order derivatives read

$$\frac{\partial u_j}{\partial \zeta} = \frac{u_{j+1} - u_{j-1}}{2\Delta \zeta}$$

and

$$\frac{\partial^2 u_j}{\partial \zeta^2} = \frac{u_{j-1} - 2u_j + u_{j+1}}{(\Delta \zeta)^2},$$

respectively, where $j = 1, \ldots, N$. Suitable values for $u_0$ and $u_{N+1}$ are obtained from the boundary conditions. They are given by $u_0 = \phi(t)$ and $u_{N+1} = u_N$.

Defining $y(t)$ by $y = (u_1, v_1, u_2, v_2, \ldots, u_N, v_N)^T$, and choosing $T = 20$, this semi-discretized problem is precisely the ODE (II.4.1).

To give an idea of the solution to the PDE (II.4.4)–(II.4.7), Figure II.4.1 plots $u$ and $v$ as function of $x$ and $t$. We nicely see that injection of chemical $A$ (locally) destroys $B$.

![Figure II.4.1: u and v as function of time and space.](image)

### 4.4 Numerical solution of the problem

The numerical experiments were done for the case $N = 200$. In Table II.4.1 we give the value of some components of the reference solution at the end of the integration interval. These components correspond to the values of $u$ and $v$ in $x = 1, 2, 4, 6.0$ and $6.0$. For the complete reference solution we refer to the Fortran subroutine solut. Figure II.4.2 plots the behavior of the solution components $y_i$ for $i \in \{79, 80, 133, 134, 171, 172, 199, 200\}$, which correspond to approximations of the PDE solutions $u$ and $v$ on the grid lines $x = 1, 2, 3$ and 4. Table II.4.2 and Figures II.4.3–II.4.6 show the run
characteristics, and the work-precision diagrams, respectively. The reference solution was computed on the Cray C90, using PSIDE with Cray double precision and atol = rtol = 10^{-10}. For the work-precision diagrams, we used: rtol = 10^{-10+4m-8}, m = 0, 1, \ldots, 40; atol = rtol; h0 = 10^{-5} \cdot rtol for BIMD, GAMD, MEBDFDAE, MEBDFI, RADAU and RADAU5. Since some solution components are zero, all scd values presented here denote absolute precision.

References


Figure II.4.2: Behavior of some solution components over the integration interval.
Figure II.4.3: Work-precision diagram (scd versus CPU-time).
Figure II.4.4: Work-precision diagram (sod versus CPU-time).
Figure II.4.5: Work-precision diagram (mesol versus CPU-time).
Figure II.4.6: Work-precision diagram (mesod versus CPU-time).
5 EMEP problem

5.1 General information
The problem is a stiff system of 66 ordinary differential equations. The ‘Mathematics and the Environment’ project group at CWI contributed this problem to the test set. The software part of the problem is in the file `emep.f` available at [MM08].

5.2 Mathematical description of the problem
The problem is of the form

\[ \frac{dy}{dt} = f(t, y), \quad y(0) = g, \]

with

\[ y \in \mathbb{R}^{66}, \quad 14400 \leq t \leq 417600. \]

The initial vector \( g = (g_i) \) is given by

\[
g_i = \begin{cases} 
1.0 \times 10^9 & \text{for } i = 1, \\
5.0 \times 10^8 & \text{for } i \in \{2, 3\}, \\
3.8 \times 10^{12} & \text{for } i = 4, \\
3.5 \times 10^{13} & \text{for } i = 5, \\
1.0 \times 10^7 & \text{for } i \in \{6, 7, \ldots, 13\}, \\
5.0 \times 10^{11} & \text{for } i = 14, \\
1.0 \times 10^2 & \text{for } i \in \{15, 16, \ldots, 37\}, \\
1.0 \times 10^{-3} & \text{for } i = 38, \\
1.0 \times 10^2 & \text{for } i \in \{39, 40, \ldots, 66\}. 
\end{cases}
\]

The function \( f \) has discontinuities in time at \( t = 3600(4 + 24i) \) and \( t = 3600(-4 + 24i) \) for \( i = 1, 2, 3, 4, 5 \). Since \( f \) is too voluminous to be described here, we refer to the Fortran subroutine `feval` and to [VS94] to get more insight in the function.

5.3 Origin of the problem
The problem is the chemistry part of the EMEP MSC-W ozone chemistry model, which is in development at the Norwegian Meteorological Institute in Oslo, Norway. About 140 reactions with a total of 66 species are involved. Below we give the correspondence between the solution vector \( y \) and the chemical species.

\[
y = (\text{NO, NO}_2, \text{SO}_2, \text{CO, CH}_4, \text{C}_2\text{H}_6, \text{NC}_4\text{H}_{10}, \text{C}_2\text{H}_4, \text{C}_3\text{H}_6, \text{OXYL, HCHO, CH}_3\text{CHO, MEK, O}_3, \text{H}_2\text{O}_2, \text{H}_2, \text{CH}_3\text{O}_2, \text{C}_2\text{H}_5\text{OH, SA, CH}_3\text{O}_2\text{H, C}_2\text{H}_5\text{O}_2, \text{CH}_3\text{CO}_2, \text{PAN, SECC}_4\text{H, MEK}_2, \text{R}_2\text{OOH, ETR}_2, \text{MG}_2\text{LY}_2\text{O}, \text{PRRO}_2, \text{GL}_2\text{OX}_2, \text{OXY}_2, \text{MAL, MAL}_2, \text{OP, OH, OD, NO}_3, \text{N}_2\text{O}_5, \text{ISOPRE, NITRAT, ISRO}_2, \text{MEK}_2, \text{CH}_3\text{O}_2, \text{RC}_2\text{H}_3, \text{OXY}_2, \text{BUR}_2\text{H, ETR}_2, \text{PRRO}_2\text{H, MEK}_2\text{H, MAL}_2\text{H, MACR, ISNI, ISRO}_2\text{H, MAR}_2\text{O, MAPAN, CH}_2\text{C}_2\text{CH}_3, \text{ISON}_3, \text{ISNIR, MVK}_2\text{H, CH}_2\text{CHR, ISNO}_3, \text{ISNIR}_2, \text{MAR}_2\text{O}_2\text{H})^T.
\]
The integration interval covers 112 hours. Rate coefficients are often variable. Some of them undergo a discontinuity at sunrise and sunset, which correspond to $t = 3600(\pm 4 + 24i)$, respectively, for $i = 1, 2, 3, 4, 5$. The unit of the species is number of molecules per cm$^3$, the time $t$ is in seconds. The test problem corresponds to the rural case in [VS94]. From the plot of O3 versus time in Figure II.5.1 we see that in this model the ozone concentration steadily grows over the integration interval. A more elaborate description of the model can be found in [VS94], [Sim93] and [SASJ93].

5.4 Numerical solution of the problem

Table II.5.1 and Figure II.5.1 present the value of reference solution at the end of the integration interval $t = 417600$ and the behavior of the solution over the integration interval of the components of $y$ corresponding to NO, NO2, SO2, CH4, O3 and N2O5 (i.e. $y_1$, $y_2$, $y_3$, $y_5$, $y_{14}$ and $y_{40}$). For the complete reference solution at the end of the integration interval we refer to the Fortran subroutine solut. The values at the horizontal axis in Figure II.5.1 denote the time $t$ in hours modulo 24 hours. Table II.5.2 and Figures II.5.2–II.5.5 contain the run characteristics and the work-precision diagrams, respectively. Since components $y_{36}$ and $y_{38}$ are relatively very small and physically unimportant, we did not include these components in the computation of the scd value. The reference solution was computed using RADAU5 with rtol = $10^{-14}$, atol = $10^{-8}$, h0 = $10^{-8}$. For the work-precision diagrams, we used: rtol = $10^{-2+m/8}$, $m = 0, 1, \ldots, 32$; atol = 1 and h0 = $10^{-7}$ for BIMD, GAMD, MEBDFDAE, MEBDFI, RADAU and RADAU5.

References


Table II.5.2: Run characteristics.

<table>
<thead>
<tr>
<th>solver</th>
<th>rtol</th>
<th>atol</th>
<th>h0</th>
<th>mesc</th>
<th>scd</th>
<th>steps</th>
<th>accept</th>
<th>#f</th>
<th>#Jac</th>
<th>#LU</th>
<th>CPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>BIMD</td>
<td>10^{-1}</td>
<td>1</td>
<td>10^{-7}</td>
<td>1.63</td>
<td>2.48</td>
<td>300</td>
<td>278</td>
<td>4797</td>
<td>212</td>
<td>296</td>
<td>0.3250</td>
</tr>
<tr>
<td></td>
<td>10^{-4}</td>
<td>1</td>
<td>10^{-7}</td>
<td>2.39</td>
<td>2.39</td>
<td>509</td>
<td>485</td>
<td>8292</td>
<td>445</td>
<td>501</td>
<td>0.5553</td>
</tr>
<tr>
<td></td>
<td>10^{-6}</td>
<td>1</td>
<td>10^{-7}</td>
<td>5.12</td>
<td>5.12</td>
<td>808</td>
<td>748</td>
<td>17116</td>
<td>639</td>
<td>793</td>
<td>1.0873</td>
</tr>
<tr>
<td>DDASSL</td>
<td>10^{-2}</td>
<td>1</td>
<td></td>
<td>1.57</td>
<td>1.82</td>
<td>741</td>
<td>701</td>
<td>1340</td>
<td>171</td>
<td>1200</td>
<td>0.1200</td>
</tr>
<tr>
<td></td>
<td>10^{-4}</td>
<td>1</td>
<td></td>
<td>3.48</td>
<td>3.48</td>
<td>1938</td>
<td>1880</td>
<td>3322</td>
<td>254</td>
<td>2557</td>
<td>0.2557</td>
</tr>
<tr>
<td></td>
<td>10^{-6}</td>
<td>1</td>
<td></td>
<td>5.35</td>
<td>5.35</td>
<td>3964</td>
<td>3851</td>
<td>6221</td>
<td>404</td>
<td>4714</td>
<td>0.4714</td>
</tr>
<tr>
<td>GAMD</td>
<td>10^{-2}</td>
<td>1</td>
<td>10^{-7}</td>
<td>2.46</td>
<td>2.46</td>
<td>347</td>
<td>283</td>
<td>10656</td>
<td>283</td>
<td>347</td>
<td>0.4851</td>
</tr>
<tr>
<td></td>
<td>10^{-4}</td>
<td>1</td>
<td>10^{-7}</td>
<td>2.92</td>
<td>2.92</td>
<td>335</td>
<td>300</td>
<td>13551</td>
<td>300</td>
<td>335</td>
<td>0.6188</td>
</tr>
<tr>
<td></td>
<td>10^{-6}</td>
<td>1</td>
<td>10^{-7}</td>
<td>4.66</td>
<td>4.64</td>
<td>607</td>
<td>503</td>
<td>28488</td>
<td>504</td>
<td>607</td>
<td>1.2629</td>
</tr>
<tr>
<td>MEBDFI</td>
<td>10^{-2}</td>
<td>1</td>
<td>10^{-7}</td>
<td>1.17</td>
<td>1.17</td>
<td>649</td>
<td>597</td>
<td>2537</td>
<td>130</td>
<td>130</td>
<td>0.1454</td>
</tr>
<tr>
<td></td>
<td>10^{-4}</td>
<td>1</td>
<td>10^{-7}</td>
<td>3.53</td>
<td>3.53</td>
<td>1320</td>
<td>1252</td>
<td>4834</td>
<td>216</td>
<td>216</td>
<td>0.2772</td>
</tr>
<tr>
<td></td>
<td>10^{-6}</td>
<td>1</td>
<td>10^{-7}</td>
<td>4.80</td>
<td>4.80</td>
<td>2621</td>
<td>2458</td>
<td>9214</td>
<td>406</td>
<td>406</td>
<td>0.5407</td>
</tr>
<tr>
<td>PSIDE-1</td>
<td>10^{-2}</td>
<td>1</td>
<td></td>
<td>1.58</td>
<td>2.39</td>
<td>490</td>
<td>438</td>
<td>6954</td>
<td>175</td>
<td>1908</td>
<td>0.8462</td>
</tr>
<tr>
<td></td>
<td>10^{-4}</td>
<td>1</td>
<td></td>
<td>2.29</td>
<td>2.29</td>
<td>509</td>
<td>447</td>
<td>9241</td>
<td>213</td>
<td>1980</td>
<td>0.9516</td>
</tr>
<tr>
<td></td>
<td>10^{-6}</td>
<td>1</td>
<td></td>
<td>3.97</td>
<td>3.95</td>
<td>769</td>
<td>650</td>
<td>15861</td>
<td>335</td>
<td>2716</td>
<td>1.4240</td>
</tr>
<tr>
<td>RADAU</td>
<td>10^{-2}</td>
<td>1</td>
<td>10^{-7}</td>
<td>1.59</td>
<td>2.57</td>
<td>398</td>
<td>325</td>
<td>3510</td>
<td>224</td>
<td>398</td>
<td>0.6159</td>
</tr>
<tr>
<td></td>
<td>10^{-4}</td>
<td>1</td>
<td>10^{-7}</td>
<td>2.68</td>
<td>2.68</td>
<td>542</td>
<td>492</td>
<td>4815</td>
<td>377</td>
<td>542</td>
<td>0.8433</td>
</tr>
<tr>
<td></td>
<td>10^{-6}</td>
<td>1</td>
<td>10^{-7}</td>
<td>3.62</td>
<td>3.60</td>
<td>463</td>
<td>390</td>
<td>10241</td>
<td>281</td>
<td>463</td>
<td>1.3566</td>
</tr>
<tr>
<td>VODE</td>
<td>10^{-2}</td>
<td>1</td>
<td></td>
<td>0.87</td>
<td>0.87</td>
<td>884</td>
<td>859</td>
<td>1409</td>
<td>62</td>
<td>272</td>
<td>0.1396</td>
</tr>
<tr>
<td></td>
<td>10^{-4}</td>
<td>1</td>
<td></td>
<td>2.49</td>
<td>2.49</td>
<td>2296</td>
<td>2199</td>
<td>3547</td>
<td>64</td>
<td>383</td>
<td>0.2586</td>
</tr>
<tr>
<td></td>
<td>10^{-6}</td>
<td>1</td>
<td></td>
<td>4.51</td>
<td>4.49</td>
<td>4302</td>
<td>4078</td>
<td>6090</td>
<td>82</td>
<td>637</td>
<td>0.4431</td>
</tr>
</tbody>
</table>
Figure II.5.1: Behavior of the solution over the integration interval.
Figure II.5.2: Work-precision diagram (scd versus CPU-time).
Figure II.5.3: Work-precision diagram (sod versus CPU-time).
Figure II.5.4: Work-precision diagram (mescd versus CPU-time).
Figure II.5.5: Work-precision diagram (mescd versus CPU-time).
6 Pleiades problem

6.1 General information

The problem consists of a nonstif system of 14 special second order differential equations rewritten to first order form, thus providing a nonstif system of ordinary differential equations of dimension 28. The formulation and data have been taken from [HWN93]. E. Messina contributed this problem to the test set. Comments to eleonora.messina@unina.it.

The software part of the problem is in the file plei.f available at [MM08].

6.2 Mathematical description of the problem

The problem is of the form

$$z'' = f(z), \quad z(0) = z_0, \quad z'(0) = z'_0,$$  \hspace{1cm} (II.6.1)

with

$$z \in \mathbb{R}^{14}, \quad 0 \leq t \leq 3.$$

Defining $z := (x^T, y^T)^T$, $x, y \in \mathbb{R}^7$, the function $f: \mathbb{R}^{14} \to \mathbb{R}^{14}$ is given by $f(z) = f(x,y) = (f^{(1)}(x,y)^T, f^{(2)}(x,y)^T)^T$, where $f^{(1,2)}: \mathbb{R}^{14} \to \mathbb{R}^7$ read

$$f^{(1)}_i = \sum_{j \neq i} m_j (x_j - x_i)/r_{ij}^2, \quad f^{(2)}_i = \sum_{j \neq i} m_j (y_j - y_i)/r_{ij}^2, \quad i = 1, \ldots, 7.$$  \hspace{1cm} (II.6.2)

Here, $m_i = i$ and

$$r_{ij} = (x_i - x_j)^2 + (y_i - y_j)^2.$$

We write this problem to first order form by defining $w = z'$, yielding a system of 28 non-linear differential equations of the form

$$\begin{pmatrix} z \\ w \end{pmatrix}' = \begin{pmatrix} w \\ f(z) \end{pmatrix},$$  \hspace{1cm} (II.6.3)

with

$$(z^T, w^T)^T \in \mathbb{R}^{28}, \quad 0 \leq t \leq 3.$$

The initial values are

$$\begin{pmatrix} z_0 \\ w_0 \end{pmatrix} = \begin{pmatrix} x_0 \\ y_0 \\ x'_0 \\ y'_0 \end{pmatrix}, \quad \text{where} \quad \begin{array}{l}
 x_0 = (3,3,-1,-3,2,-2,2)^T, \\
 y_0 = (3,-3,2,0,0,-4,4)^T, \\
 x'_0 = (0,0,0,0,0,1.75,-1.5)^T, \\
 y'_0 = (0,0,0,-1.25,1,0,0)^T.
\end{array}$$

6.3 Origin of the problem

The Pleiades problem is a celestial mechanics problem of seven stars in the plane of coordinates $x_i$, $y_i$ and masses $m_i = i$ ($i = 1, \ldots, 7$). We obtain the formulation of the problem by means of some mechanical considerations. Let us consider the body $i$. According to the second law of Newton this star is subjected to the action

$$F_i = m_i p_i'^2,$$  \hspace{1cm} (II.6.4)

where $p_i := (x_i,y_i)^T$. On the other hand, the law of gravity states that the force working on body $i$ implied by body $j$, denoted by $F_{ij}$, is

$$F_{ij} = g \frac{m_i \cdot m_j}{\|p_i - p_j\|^2} \delta_{ij},$$  \hspace{1cm} (II.6.5)
I-6-2 ODE - Pleiades problem

Figure II.6.1: Trajectories of the first and third body on $[0,2]$.

Table II.6.1: Quasi-collisions in Pleiades problem. The squared distance between body $i$ and body $j$ at $t = \tau$ is listed (values taken from [HNW93]).

| $i$ | $j$ | $\tau$ | $||p_i - p_j||^2$ |
|-----|-----|--------|-----------------|
| 1   | 7   | 1.23   | 0.0129          |
| 1   | 3   | 1.46   | 0.0193          |
| 1   | 5   | 1.63   | 0.0031          |
| 1   | 7   | 1.68   | 0.0011          |
| 2   | 6   | 1.94   | 0.1005          |
| 2   | 7   | 2.14   | 0.0700          |

Here, $F_i$, $F_{ij} \in \mathbb{R}^2$, $g$ is the gravitational constant, which is assumed to be one here, and $d_{ij} = \frac{p_i - p_j}{||p_i - p_j||}$ represents the direction of the distance between the two stars. According to the principle of superposition of actions, $F_i$ will be the sum of the interactions between body $i$ and all the others,

$$F_i = \sum_{i \neq j} F_{ij}.$$  \hspace{2cm} (II.6.6)

It is easily checked that (II.6.4)–(II.6.6) and (II.6.2) are the same.

During the movement of the 7 bodies several quasi-collisions occur which are displayed in Table II.6.1. In Figure II.6.1 the behaviors of the bodies 1 and 3 in the interval $[0,2]$ are shown; the circles and the crosses represent data obtained every 0.05 sec, the link ‘-‘ indicates the distance occurring between the two stars at $t = 1.45$.

6.4 Numerical solution of the problem

One should be aware of the fact that the Pleiades problem is a nonstiff ODE. Therefore we also include the results obtained by the nonstiff solver DOPR15[HW96], which is based on an explicit Runge–Kutta method.
Tables II.6.2–II.6.3 and Figures II.6.2–II.6.4 present the reference solution at the end of the integration interval, the run characteristics, the behavior of the solution components $x_1$ and $y_1$ over the integration interval and the work-precision diagrams, respectively. The computation of the scd values is based on the first 14 components, since they refer to the physically important quantities. The reference solution was computed on the Cray C90, using PSIDE with Cray double precision and $\text{atol} = \text{rtol} = 10^{-16}$. For the work-precision diagrams, we used: $\text{rtol} = 10^{-((4+m)/3)}$, $m = 0, 1, \ldots, 24$; $\text{atol} = \text{rtol}$; $h^0 = 10^{-2} \cdot \text{rtol}$ for BIMD, GAMD, RADAU, RADAU5 and MEBDFDAE.

With respect to the RADAU and RADAU5 results in Table II.6.3 and Figures II.6.3–II.6.4, we remark that for generality of the test set drivers, we did not use the facility to exploit the special structure of problems of the form (II.6.3). By setting the input parameter $\text{IWORK}(9)=14$, and adjusting the Jacobian routine appropriately, RADAU and RADAU5 produces considerably better results.

These results are listed for RADAU in Table II.6.4.

References


Table II.6.3: Run characteristics.

<table>
<thead>
<tr>
<th>solver</th>
<th>rtol</th>
<th>atol</th>
<th>h0</th>
<th>mescd</th>
<th>scd</th>
<th>steps</th>
<th>accept</th>
<th>#f</th>
<th>#Jac</th>
<th>#LU</th>
<th>CPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>BIMD</td>
<td>$10^{-4}$</td>
<td>$10^{-4}$</td>
<td>$10^{-6}$</td>
<td>2.69</td>
<td>2.12</td>
<td>113</td>
<td>105</td>
<td>1955</td>
<td>79</td>
<td>110</td>
<td>0.0449</td>
</tr>
<tr>
<td></td>
<td>$10^{-7}$</td>
<td>$10^{-7}$</td>
<td>$10^{-9}$</td>
<td>5.38</td>
<td>4.81</td>
<td>138</td>
<td>127</td>
<td>4013</td>
<td>123</td>
<td>138</td>
<td>0.0888</td>
</tr>
<tr>
<td></td>
<td>$10^{-10}$</td>
<td>$10^{-10}$</td>
<td>$10^{-12}$</td>
<td>8.60</td>
<td>8.42</td>
<td>154</td>
<td>138</td>
<td>6947</td>
<td>129</td>
<td>152</td>
<td>0.1562</td>
</tr>
<tr>
<td>DDASSL</td>
<td>$10^{-4}$</td>
<td>$10^{-4}$</td>
<td></td>
<td>0.80</td>
<td>0.23</td>
<td>428</td>
<td>390</td>
<td>589</td>
<td>49</td>
<td>185</td>
<td>0.0185</td>
</tr>
<tr>
<td></td>
<td>$10^{-7}$</td>
<td>$10^{-7}$</td>
<td></td>
<td>3.43</td>
<td>3.24</td>
<td>1237</td>
<td>1224</td>
<td>1674</td>
<td>59</td>
<td>0.0517</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$10^{-10}$</td>
<td>$10^{-10}$</td>
<td></td>
<td>5.88</td>
<td>5.72</td>
<td>3778</td>
<td>3773</td>
<td>4709</td>
<td>61</td>
<td>0.1425</td>
<td></td>
</tr>
<tr>
<td>DOPRI5</td>
<td>$10^{-4}$</td>
<td>$10^{-4}$</td>
<td></td>
<td>1.06</td>
<td>0.50</td>
<td>100</td>
<td>74</td>
<td>602</td>
<td>0.0059</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$10^{-7}$</td>
<td>$10^{-7}$</td>
<td></td>
<td>4.06</td>
<td>3.49</td>
<td>295</td>
<td>244</td>
<td>1772</td>
<td>0.0176</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$10^{-10}$</td>
<td>$10^{-10}$</td>
<td></td>
<td>8.06</td>
<td>7.83</td>
<td>940</td>
<td>940</td>
<td>5642</td>
<td>0.0366</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GAMD</td>
<td>$10^{-4}$</td>
<td>$10^{-4}$</td>
<td>$10^{-6}$</td>
<td>1.54</td>
<td>0.97</td>
<td>85</td>
<td>69</td>
<td>2751</td>
<td>69</td>
<td>85</td>
<td>0.0566</td>
</tr>
<tr>
<td></td>
<td>$10^{-7}$</td>
<td>$10^{-7}$</td>
<td>$10^{-9}$</td>
<td>4.81</td>
<td>4.57</td>
<td>122</td>
<td>104</td>
<td>5163</td>
<td>104</td>
<td>122</td>
<td>0.1083</td>
</tr>
<tr>
<td></td>
<td>$10^{-10}$</td>
<td>$10^{-10}$</td>
<td>$10^{-12}$</td>
<td>7.65</td>
<td>7.30</td>
<td>183</td>
<td>177</td>
<td>7927</td>
<td>173</td>
<td>183</td>
<td>0.1649</td>
</tr>
<tr>
<td>MEBDFI</td>
<td>$10^{-4}$</td>
<td>$10^{-4}$</td>
<td>$10^{-6}$</td>
<td>1.12</td>
<td>0.56</td>
<td>387</td>
<td>366</td>
<td>1339</td>
<td>56</td>
<td>56</td>
<td>0.0303</td>
</tr>
<tr>
<td></td>
<td>$10^{-7}$</td>
<td>$10^{-7}$</td>
<td>$10^{-9}$</td>
<td>3.84</td>
<td>3.62</td>
<td>835</td>
<td>816</td>
<td>2764</td>
<td>86</td>
<td>86</td>
<td>0.0654</td>
</tr>
<tr>
<td></td>
<td>$10^{-10}$</td>
<td>$10^{-10}$</td>
<td>$10^{-12}$</td>
<td>7.14</td>
<td>6.94</td>
<td>1868</td>
<td>1868</td>
<td>6119</td>
<td>189</td>
<td>189</td>
<td>0.1454</td>
</tr>
<tr>
<td>PSIDE-1</td>
<td>$10^{-4}$</td>
<td>$10^{-4}$</td>
<td></td>
<td>2.23</td>
<td>1.82</td>
<td>102</td>
<td>76</td>
<td>1710</td>
<td>27</td>
<td>364</td>
<td>0.0410</td>
</tr>
<tr>
<td></td>
<td>$10^{-7}$</td>
<td>$10^{-7}$</td>
<td></td>
<td>5.26</td>
<td>4.70</td>
<td>248</td>
<td>223</td>
<td>3187</td>
<td>1</td>
<td>592</td>
<td>0.0712</td>
</tr>
<tr>
<td></td>
<td>$10^{-10}$</td>
<td>$10^{-10}$</td>
<td></td>
<td>8.12</td>
<td>7.55</td>
<td>807</td>
<td>807</td>
<td>9095</td>
<td>1</td>
<td>604</td>
<td>0.1786</td>
</tr>
<tr>
<td>RADAU</td>
<td>$10^{-4}$</td>
<td>$10^{-4}$</td>
<td>$10^{-6}$</td>
<td>2.67</td>
<td>2.11</td>
<td>151</td>
<td>138</td>
<td>1053</td>
<td>132</td>
<td>151</td>
<td>0.0303</td>
</tr>
<tr>
<td></td>
<td>$10^{-7}$</td>
<td>$10^{-7}$</td>
<td>$10^{-9}$</td>
<td>6.20</td>
<td>6.17</td>
<td>112</td>
<td>95</td>
<td>2153</td>
<td>83</td>
<td>112</td>
<td>0.0547</td>
</tr>
<tr>
<td></td>
<td>$10^{-10}$</td>
<td>$10^{-10}$</td>
<td>$10^{-12}$</td>
<td>9.41</td>
<td>9.20</td>
<td>130</td>
<td>119</td>
<td>3001</td>
<td>91</td>
<td>130</td>
<td>0.0742</td>
</tr>
<tr>
<td>VODE</td>
<td>$10^{-4}$</td>
<td>$10^{-4}$</td>
<td></td>
<td>0.40</td>
<td>-0.17</td>
<td>352</td>
<td>325</td>
<td>468</td>
<td>6</td>
<td>57</td>
<td>0.0117</td>
</tr>
<tr>
<td></td>
<td>$10^{-7}$</td>
<td>$10^{-7}$</td>
<td></td>
<td>2.76</td>
<td>2.57</td>
<td>1081</td>
<td>1043</td>
<td>1232</td>
<td>18</td>
<td>94</td>
<td>0.0303</td>
</tr>
<tr>
<td></td>
<td>$10^{-10}$</td>
<td>$10^{-10}$</td>
<td></td>
<td>5.41</td>
<td>5.20</td>
<td>3120</td>
<td>3079</td>
<td>3351</td>
<td>51</td>
<td>203</td>
<td>0.0830</td>
</tr>
</tbody>
</table>

Table II.6.4: Run characteristics obtained by RADAU with exploited special structure.

<table>
<thead>
<tr>
<th>solver</th>
<th>rtol</th>
<th>atol</th>
<th>h0</th>
<th>mescd</th>
<th>scd</th>
<th>steps</th>
<th>accept</th>
<th>#f</th>
<th>#Jac</th>
<th>#LU</th>
<th>CPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>RADAU</td>
<td>$10^{-4}$</td>
<td>$10^{-4}$</td>
<td>$10^{-6}$</td>
<td>1.72</td>
<td>2.11</td>
<td>151</td>
<td>138</td>
<td>1053</td>
<td>132</td>
<td>151</td>
<td>0.0234</td>
</tr>
<tr>
<td></td>
<td>$10^{-7}$</td>
<td>$10^{-7}$</td>
<td>$10^{-9}$</td>
<td>5.13</td>
<td>6.17</td>
<td>112</td>
<td>95</td>
<td>2153</td>
<td>83</td>
<td>112</td>
<td>0.0429</td>
</tr>
<tr>
<td></td>
<td>$10^{-10}$</td>
<td>$10^{-10}$</td>
<td>$10^{-12}$</td>
<td>8.27</td>
<td>9.20</td>
<td>130</td>
<td>119</td>
<td>3001</td>
<td>91</td>
<td>130</td>
<td>0.0586</td>
</tr>
</tbody>
</table>
Figure II.6.2: Behavior of the two solution components corresponding to the first body over the integration interval.
Figure II.6.3: Work-precision diagram (sod versus CPU-time).
Figure II.6.4: Work-precision diagram (sod versus CPU-time).
Figure II.6.5: Work-precision diagram (mescd versus CPU-time).
Figure II.6.6: Work-precision diagram (mescd versus CPU-time).
7 Problem BEAM

7.1 General information
The problem is originally described by a partial differential equation subject to boundary conditions. The semi-discretization in space of this equation leads to a stiff system of $n$ non-linear second order differential equations which is rewritten to first order form, thus providing a stiff system of ordinary differential equations of dimension $2n$. The formulation and data have been taken from [HW96]. The INdAM-Bari Test Set group contributed this problem to the test set. The software part of the problem is in the files beam.f available at [MM08].

7.2 Mathematical description of the problem
The problem is of the form

$$z'' = f(t, z, z'), \quad z(0) = z_0, \quad z'(0) = z'_0,$$

with

$$z \in \mathbb{R}^n, \quad t \geq 0.$$

The function $f : \mathbb{R}^n \to \mathbb{R}^n$ is defined by

$$f(t, z, z') = Cv + Du.$$

Here $C$ is the tridiagonal $n \times n$ matrix whose entries are given by

$$
\begin{align*}
(C)_{11} &= 1, \quad (C)_{nn} = 3, \quad \text{and} \quad (C)_{ll} = 2, \quad l = 2, \ldots, n-1, \\
(C)_{l,l+1} &= -\cos(z_l - z_{l+1}), \quad l = 1, \ldots, n-1, \\
(C)_{l,l-1} &= -\cos(z_l - z_{l-1}), \quad l = 2, \ldots, n,
\end{align*}
$$

and $D$ is the $n \times n$ bidiagonal matrix whose lower and upper diagonal entries are

$$
\begin{align*}
(D)_{l,l+1} &= -\sin(z_l - z_{l+1}), \quad l = 1, \ldots, n-1, \\
(D)_{l,l-1} &= -\sin(z_l - z_{l-1}), \quad l = 2, \ldots, n,
\end{align*}
$$

$v = (v_1, v_2, \ldots, v_n)^T$ is defined by

$$v_l = n^4(z_{l-1} - 2z_l + z_{l+1}) + n^2(\cos(z_l)F_y - \sin(z_l)F_x), \quad l = 1, \ldots, n$$

with $z_0 = -z_1, \quad z_{n+1} = z_n$, and $u$ is the column vector of size $n$ solution of the tridiagonal system

$$Cu = g$$

with $g = Dv + (z_1^2, z_2^2, \ldots, z_n^2)^T$.

We write this problem to first order form by defining $w = z'$, yielding a system of $2n$ non-linear differential equations of the form

$$
\begin{pmatrix}
    z \\
    w
\end{pmatrix}' = 
\begin{pmatrix}
    w \\
    f(t, z, w)
\end{pmatrix}
$$

with

$$(z, w)^T \in \mathbb{R}^{2n}, \quad t \geq 0.$$  

The initial values are

$$
\begin{pmatrix}
    z_0 \\
    w_0
\end{pmatrix} = \begin{pmatrix}
    z'_0 \\
    z_0
\end{pmatrix}, \quad \text{where} \quad \begin{cases}
    z_0 = (0, 0, \ldots, 0)^T \\
    z'_0 = (0, 0, \ldots, 0)^T
\end{cases}
$$
7.3 Origin of the problem

The BEAM problem originates from mechanics and describes the motion of an elastic beam which is supposed inextensible, of length 1 and thin. Moreover, it is assumed that the beam is clamped at one end and a force \( F = (F_u, F_v) \) acts at the free end. As coordinate system it is used the angle \( \theta \) as a function of arc length \( s \) and time \( t \). The beam is then described by the equations

\[
\begin{align*}
\dot{u}(s, t) &= \int_0^s \cos \theta(\sigma, t) \, d\sigma, \\
\dot{v}(s, t) &= \int_0^s \sin \theta(\sigma, t) \, d\sigma.
\end{align*}
\]

In order to obtain the equations of motion for this problem, the Lagrange theory is applied. Let \( T \) be the kinetic and \( U \) the potential energy defined respectively as follows

\[
\begin{align*}
T &= \frac{1}{2} \int_0^1 (\dot{u}(s, t)^2 + \dot{v}(s, t)^2) \, ds \\
U &= \frac{1}{2} \int_0^1 \left( \frac{\theta''(s, t)^2}{\max(s, \max(\sigma))} - F_u(t)u(1, t) - F_v(t)v(1, t) \right) \, ds.
\end{align*}
\]

Here dots and primes denote derivatives with respect to \( t \) and \( s \), respectively.

Using the Hamilton principle, the equations of motion are derived. They are

\[
\begin{align*}
\int_0^1 G(s, \sigma) \cos(\theta(s, t) - \theta(\sigma, t)) \, d\sigma = \\
= \dot{\theta}''(s, t) + \cos \theta(s, t)F_v(t) - \sin \theta(s, t)F_u(t) \tag{II.7.1} \\
- \int_0^1 G(s, \sigma) \sin(\theta(s, t) - \theta(\sigma, t))(\dot{\theta}(\sigma, t))^2 \, d\sigma \\
\theta(0, t) = 0, \quad \dot{\theta}(1, t) = 0
\end{align*}
\]

where

\[
G(s, \sigma) = 1 - \max(s, \sigma)
\]

is the Green function for the problem \(-w''(s) = g(s), w'(0) = w(1) = 0\).

We discretize the integrals with the midpoint rule:

\[
\int_0^1 f(\theta(\sigma, t)) \, d\sigma = \frac{1}{n} \sum_{k=1}^n f(\theta_k), \quad \theta_k = \theta\left((k - \frac{1}{2}) \frac{1}{n}, t\right), \quad k = 1, \ldots, n.
\]

Equations (II.7.1) then become

\[
\begin{align*}
\sum_{k=1}^n a_{\ell k} \ddot{\theta}_k &= n^4(\theta_{k-1} - 2\theta_k + \theta_{k+1}) + n^2(\cos \theta_k F_v - \sin \theta_k F_u) \\
- \sum_{k=1}^n g_{\ell k} \sin(\theta_k - \theta_k) \dot{\theta}_k^2, \quad \ell = 1, \ldots, n, \\
\theta_0 &= -\theta_1, \quad \theta_{n+1} = \theta_n,
\end{align*}
\]

where

\[
\begin{align*}
a_{\ell k} &= g_{\ell k} \cos(\theta_k - \theta_k), \quad g_{\ell k} = n + \frac{1}{2} - \max(\ell, k),
\end{align*}
\]
Table II.7.1: Run characteristics.

<table>
<thead>
<tr>
<th>solver</th>
<th>rtol</th>
<th>atol</th>
<th>h0</th>
<th>mescd</th>
<th>scd</th>
<th>steps</th>
<th>accept</th>
<th>#f</th>
<th>#Jac</th>
<th>#LU</th>
<th>CPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>BIMD</td>
<td>10^{-4}</td>
<td>10^{-4}</td>
<td>10^{-4}</td>
<td>3.53</td>
<td>3.58</td>
<td>60</td>
<td>60</td>
<td>1249</td>
<td>58</td>
<td>59</td>
<td>0.2137</td>
</tr>
<tr>
<td></td>
<td>10^{-7}</td>
<td>10^{-7}</td>
<td>10^{-7}</td>
<td>5.61</td>
<td>6.97</td>
<td>777</td>
<td>777</td>
<td>16197</td>
<td>722</td>
<td>744</td>
<td>2.7308</td>
</tr>
<tr>
<td>DDASLL</td>
<td>10^{-4}</td>
<td>10^{-4}</td>
<td>10^{-4}</td>
<td>1.83</td>
<td>2.29</td>
<td>29120</td>
<td>28928</td>
<td>30700</td>
<td>243</td>
<td>3.1544</td>
<td></td>
</tr>
<tr>
<td></td>
<td>10^{-7}</td>
<td>10^{-7}</td>
<td>10^{-7}</td>
<td>4.63</td>
<td>5.25</td>
<td>51757</td>
<td>51160</td>
<td>56908</td>
<td>768</td>
<td>6.4455</td>
<td></td>
</tr>
<tr>
<td>GAMD</td>
<td>10^{-4}</td>
<td>10^{-4}</td>
<td>10^{-4}</td>
<td>3.58</td>
<td>3.59</td>
<td>49</td>
<td>49</td>
<td>1715</td>
<td>49</td>
<td>49</td>
<td>0.2030</td>
</tr>
<tr>
<td></td>
<td>10^{-7}</td>
<td>10^{-7}</td>
<td>10^{-7}</td>
<td>5.49</td>
<td>6.28</td>
<td>459</td>
<td>458</td>
<td>21156</td>
<td>458</td>
<td>459</td>
<td>2.2321</td>
</tr>
<tr>
<td>MEBDFI</td>
<td>10^{-4}</td>
<td>10^{-4}</td>
<td>10^{-4}</td>
<td>2.56</td>
<td>1.92</td>
<td>578</td>
<td>559</td>
<td>6447</td>
<td>55</td>
<td>55</td>
<td>0.2284</td>
</tr>
<tr>
<td></td>
<td>10^{-7}</td>
<td>10^{-7}</td>
<td>10^{-7}</td>
<td>5.20</td>
<td>5.26</td>
<td>38693</td>
<td>38645</td>
<td>292234</td>
<td>2054</td>
<td>2054</td>
<td>127690</td>
</tr>
<tr>
<td>PSIDE-1</td>
<td>10^{-4}</td>
<td>10^{-4}</td>
<td>10^{-4}</td>
<td>2.52</td>
<td>2.14</td>
<td>42</td>
<td>36</td>
<td>1096</td>
<td>29</td>
<td>168</td>
<td>0.2903</td>
</tr>
<tr>
<td></td>
<td>10^{-7}</td>
<td>10^{-7}</td>
<td>10^{-7}</td>
<td>4.28</td>
<td>5.44</td>
<td>241</td>
<td>208</td>
<td>8006</td>
<td>192</td>
<td>964</td>
<td>1.4806</td>
</tr>
<tr>
<td>RADAU</td>
<td>10^{-4}</td>
<td>10^{-4}</td>
<td>10^{-4}</td>
<td>3.57</td>
<td>2.49</td>
<td>62</td>
<td>55</td>
<td>406</td>
<td>43</td>
<td>61</td>
<td>0.2645</td>
</tr>
<tr>
<td></td>
<td>10^{-7}</td>
<td>10^{-7}</td>
<td>10^{-7}</td>
<td>4.24</td>
<td>5.72</td>
<td>71</td>
<td>71</td>
<td>1653</td>
<td>46</td>
<td>60</td>
<td>0.5632</td>
</tr>
<tr>
<td>VODE</td>
<td>10^{-4}</td>
<td>10^{-4}</td>
<td>10^{-4}</td>
<td>-0.25</td>
<td>1.09</td>
<td>60537</td>
<td>60519</td>
<td>145514</td>
<td>1099</td>
<td>3041</td>
<td>7.3727</td>
</tr>
<tr>
<td></td>
<td>10^{-7}</td>
<td>10^{-7}</td>
<td>10^{-7}</td>
<td>4.40</td>
<td>6.48</td>
<td>58132</td>
<td>57793</td>
<td>139394</td>
<td>967</td>
<td>3338</td>
<td>7.8080</td>
</tr>
</tbody>
</table>

In Hairer & Wanner [HW96] the exterior forces are chosen as

\[ F_u = -\varphi(t), \quad F_v = \varphi(t), \quad \varphi(t) = \begin{cases} 1.5 \sin^2 t, & 0 \leq t \leq \pi, \\ 0, & \pi \leq t, \end{cases} \]

and the initial conditions are taken to be

\[ \theta(s,0) = 0, \quad \dot{\theta}(s,0) = 0. \]

### 7.4 Numerical solution of the problem

The resulting system of ODEs is integrated for \( 0 \leq t \leq 5 \), using \( n = 40 \). Table II.7.1 and Figures II.7.1-II.7.3 present the run characteristics, the behavior of the solution components \( z_{10}, z_{20}, z_{30} \) and \( z_{40} \) over the interval and the work-precision diagrams, respectively. The computation of the scd values is based on the first 40 components, since they refer to the physically important quantities. The reference solution was computed by RADAU on an Alphaserver DS20E, with a 667 MHz EV67 processor, using double precision \( \text{work}(1) = \text{uround} = 1.0 \times 10^{-19}, \text{rtol} = \text{atol} = h0 = 1.1 \times 10^{-18} \).

For the work-precision diagrams, we used: \( \text{rtol} = 10^{-4(m/4)}, m = 0, \ldots, 16; \text{atol} = \text{rtol}; h0 = \text{rtol} \) for BIMD, GAMD, MEBDFDAE, MEBDFI, RADAU and RADAU5.

With respect to the RADAU and RADAU5 results in Table II.7.1 and Figures II.7.2-II.7.5, we remark that for generality of the test set drivers, we did not use the facility to exploit the special structure of problems. By setting the input parameter \( \text{WORK}(9)=40 \), and adjusting the Jacobian routine appropriately, RADAU and RADAU5 produce considerably better results.

These results are listed for RADAU in Table II.7.2.

### References

Table II.7.2: Run characteristics obtained by RADAU with exploited special structure.

<table>
<thead>
<tr>
<th>solver</th>
<th>rtol</th>
<th>atol</th>
<th>h0</th>
<th>mesd</th>
<th>sec</th>
<th>steps</th>
<th>accept</th>
<th>#f</th>
<th>#Jac</th>
<th>#LU</th>
<th>CPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>RADAU</td>
<td>$10^{-4}$</td>
<td>$10^{-4}$</td>
<td>$10^{-4}$</td>
<td>3.42</td>
<td>2.49</td>
<td>62</td>
<td>55</td>
<td>406</td>
<td>43</td>
<td>61</td>
<td>0.0869</td>
</tr>
<tr>
<td></td>
<td>$10^{-7}$</td>
<td>$10^{-7}$</td>
<td>$10^{-7}$</td>
<td>4.24</td>
<td>5.72</td>
<td>71</td>
<td>71</td>
<td>1653</td>
<td>46</td>
<td>60</td>
<td>0.1728</td>
</tr>
</tbody>
</table>

Figure II.7.1: Behavior of the solution components $z_{10}$, $z_{20}$, $z_{30}$ and $z_{40}$ over the integration interval.
Figure II.7.2: Work-precision diagram (sec versus CPU-time).
Figure II.7.3: Work-precision diagram (scd versus CPU-time).
Figure II.7.4: Work-precision diagram (mescd versus CPU-time).
Figure II.7.5: Work-precision diagram (meshed versus CPU-time).
8 Problem VDPOL

8.1 General information

The problem consists of a second order differential equation rewritten to first order form, thus providing a system of ordinary differential equations of dimension 2. It was proposed by B. van der Pol in the 1920’s [vdP20], [vdP26]. The INdAM-Bari Test Set group contributed this problem to the test set. Most of the documentation about this problem has been retrieved from [EP02]. The software part of the problem is in the files vdpol.f and vdpolm.f available at [MM08].

8.2 Mathematical description of the problem

The problem is of the form

\[ z'' = f(z, z'), \quad z(0) = z_0, \quad z'(0) = z'_0, \]

with

\[ z \in \mathbb{R}, \quad t \in [0, T], \]

where the function \( f \) is given by

\[ f(z, z') = \mu (1 - z^2)z' - z, \quad \mu > 0. \]  

(II.8.1)

We write this problem to first order form by defining \( y_1 = z \) and \( y_2 = z' \), yielding a system of 2 nonlinear differential equations of the form

\[ \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}' = \begin{pmatrix} y_2 \\ f(y_1, y_2) \end{pmatrix} \]  

(II.8.2)

with

\( (y_1, y_2)^T \in \mathbb{R}^2, \quad t \in [0, T]. \)

A rescaling of the solutions of (II.8.2) results in the following formulation

\[ \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}' = \begin{pmatrix} y_2 \\ \bar{f}(y_1, y_2) \end{pmatrix}, \]

(II.8.3)

where

\[ \bar{f}(y_1, y_2) = ((1 - y_1^2)y_2 - y_1)/\epsilon, \quad \epsilon > 0. \]

Problem (II.8.2) will be referred to as vdpol\(_\mu\) and problem (II.8.3) as vdpol\(_\epsilon\). The initial values are

\[ \begin{pmatrix} y_1(0) \\ y_2(0) \end{pmatrix} = \begin{pmatrix} z_0 \\ z'_0 \end{pmatrix} \quad \text{where} \quad \begin{cases} z_0 = 2 \\ z'_0 = 0 \end{cases} \]

8.3 Origin of the problem

The VDPOL problem originates from electronics and describes the behaviour of nonlinear vacuum tube circuits. The circuit scheme, designed by Balthazar van der Pol in the 1920’s, is given in Figure II.8.1. This is an RLC loop, but with the passive resistor of Ohm’s Law replaced by an active element which would pump energy into the circuit whenever the amplitude of the current falls below a certain level. In the 1920’s this active element was an array of vacuum tubes, now it is a semiconductor device. The voltage drop at the semiconductor (which used to be RI) is given by a nonlinear function \( f(I) \) of
the current $I$. If we substitute $f(I)$ for $RI$ in the standard RLC-circuit equation $LI'' + RI' + I/C = 0$, the current in the circuit turns out to be modeled by

$$LI'' + f'(I)I' + I/C = 0. \quad (II.8.4)$$

In a 1924 study of oscillator circuits in early commercial radios (at Philips research laboratories), B. van der Pol assumed the voltage drop to be represented by the nonlinear function $f(I) = bI^3 - aI$, which with equation (II.8.4) leads to

$$LI'' + (3bI^2 - a)I' + I/C = 0. \quad (II.8.5)$$

This equation is also closely related to the equation introduced by the British mathematical physicist Lord Rayleigh (John William Strutt, 1842 - 1919) to model the oscillations of a clarinet reed. For more details see [EP02].

If we denote by $\tau$ the time variable in Eq. (II.8.5) and make the substitutions $I = pz$ and $t = \tau/\sqrt{LC}$, the result is

$$\frac{d^2 z}{dt^2} + (3b\mu^2 z^2 - a)\sqrt{C}\frac{dz}{L \, dt} + z = 0.$$

With $p = \sqrt{a/(3b)}$ and $\mu = a\sqrt{C/L}$ this gives the standard form

$$z'' + \mu(z^2 - 1)z' + z = 0$$

of the van der Pol's equation.

The van der Pol equation is often used as a test problem for ODEs solvers. It has two periodic solutions, the constant solution, $z(t) \equiv 0$, that is unstable, and the nontrivial periodic solution (roughly corresponding to the initial conditions $z(0) = 2$, $z'(0) = 0$), that is named ‘limit cycle’ because all the other nontrivial solutions converge to this one as $t \to \infty$.

This qualitative behavior is well shown in the phase plane plot in Figure II.8.2 (for $\mu = 2$), where outward and inward spiral trajectories converge to the limit cycle (the closed curve).

The parameter $\mu > 0$ weights the importance of the nonlinear part of the equation. When $\mu$ is 'large' the approach to the limit cycle is quite rapid (see Figure II.8.3 for $\mu = 10^2$) and the van der Pol equation is more interesting because of the non negligible influence of the nonlinear term. From
an analysis of the behavior of the limit cycle [Sha94] it turns out that it can be described in terms of portions where the solution components change slowly and the problem is quite stiff, alternating with regions of very sharp change (quasi-discontinuities) where it is non-stiff. Thus, the problem switches from stiff to non-stiff with a very sharp changing solution that makes the equation quite challenging for ODEs solvers.

The van der Pol equation may be treated in different ways, the most straightforward is to split the equation into a system of two first order differential equations as in (II.8.2). Note that if the second of the equations is divided by \( \mu \) we get an equation that has the character of a singular perturbation problem. Several other approaches may show other aspects on the nature of this problem. For example Hairer and Wanner [HW96] introduce the following scaling transformation of (II.8.2) to make the steady-state approximation independent of \( \mu \):

\[
\begin{align*}
x &= t/\mu, \\
w_1(x) &= y_1(t), \\
w_2(x) &= \mu y_2(t)
\end{align*}
\]

Substituting in (II.8.2) and using again \( y \) for \( w \) and \( t \) for \( x \), the equation (II.8.3) is obtained with \( \varepsilon = 1/\mu^2 \). The scaled version (II.8.3) has the advantage that a small interval independent of the parameter value can be considered to track at least one period of the solution.

8.4 Numerical solution of the problem

8.4.1 \texttt{vdpol} with \( \mu = 10^3 \) and \( t \in [0, 2\mu] \)

Tables II.8.1, II.8.2 and Figures II.8.4, II.8.6–II.8.9 present the reference solution at the end of the integration interval, the run characteristics, the behavior of the first component of the solution over the integration interval and the work-precision diagrams, respectively. The reference solution was computed by RADAU on an Alphaserver DS20E, with a 667 MHz EV67 processor, using double precision \( \text{work}(1) = \text{uround} = 1.01 \cdot 10^{-19}, \text{rtol} = \text{atol} = h0 = 1.1 \cdot 10^{-18} \). For the work-precision diagrams, we used: \( \text{rtol} = 10^{-(4+m/4)}, \) \( m = 0, 1, \ldots, 32; \text{atol} = \text{rtol}; h0 = 10^{-2} \cdot \text{rtol} \) for BIMD, GAMD, MEBDFDAE, MEBDFI, RADAU and RADAU5.
8.4.2 \textit{vdpol}, with $\epsilon = 10^{-6}$ and $t \in [0, 2]$

Tables II.8.3, II.8.4 and Figures II.8.5, II.8.10-II.8.13 present the reference solution at the end of the integration interval, the run characteristics, the behavior of the first component of the solution over the integration interval and the work-precision diagrams, respectively. The reference solution was computed by RADAU on an Alphaserver DS20E, with a 667 MHz EV67 processor, using double precision $\text{work}(1) = \text{uround} = 1.01 \cdot 10^{-19}$, $\text{rtol} = \text{atol} = h_{0} = 1.1 \cdot 10^{-18}$. For the work-precision

\begin{table}[h]
\centering
\begin{tabular}{|c|c|}
\hline
$y_1$ & $0.170616732170469 \cdot 10^{1}$ \\
$y_2$ & $-0.8928097010248125 \cdot 10^{-3}$ \\
\hline
\end{tabular}
\caption{Reference solution at the end of the integration interval.}
\end{table}
Table II.8.2: Run characteristics.

<table>
<thead>
<tr>
<th>solver</th>
<th>rtol</th>
<th>atol</th>
<th>h0</th>
<th>mescd</th>
<th>scd</th>
<th>steps</th>
<th>accept</th>
<th>#f</th>
<th>#Jac</th>
<th>#LU</th>
<th>CPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>BIMD</td>
<td>10^{-4}</td>
<td>10^{-4}</td>
<td>10^{-6}</td>
<td>4.05</td>
<td>3.57</td>
<td>133</td>
<td>112</td>
<td>2801</td>
<td>104</td>
<td>133</td>
<td>0.0020</td>
</tr>
<tr>
<td></td>
<td>10^{-7}</td>
<td>10^{-7}</td>
<td>10^{-9}</td>
<td>9.18</td>
<td>8.80</td>
<td>224</td>
<td>219</td>
<td>5072</td>
<td>209</td>
<td>224</td>
<td>0.0029</td>
</tr>
<tr>
<td></td>
<td>10^{-10}</td>
<td>10^{-10}</td>
<td>10^{-12}</td>
<td>11.17</td>
<td>10.32</td>
<td>250</td>
<td>248</td>
<td>10151</td>
<td>237</td>
<td>250</td>
<td>0.0078</td>
</tr>
<tr>
<td>DDASSL</td>
<td>10^{-4}</td>
<td>10^{-4}</td>
<td></td>
<td>2.88</td>
<td>2.37</td>
<td>549</td>
<td>507</td>
<td>940</td>
<td>122</td>
<td></td>
<td>0.0020</td>
</tr>
<tr>
<td></td>
<td>10^{-7}</td>
<td>10^{-7}</td>
<td></td>
<td>5.57</td>
<td>5.06</td>
<td>1342</td>
<td>1296</td>
<td>1980</td>
<td>129</td>
<td></td>
<td>0.0049</td>
</tr>
<tr>
<td></td>
<td>10^{-10}</td>
<td>10^{-10}</td>
<td></td>
<td>8.25</td>
<td>7.73</td>
<td>4484</td>
<td>4445</td>
<td>5943</td>
<td>168</td>
<td></td>
<td>0.0166</td>
</tr>
<tr>
<td>GAMD</td>
<td>10^{-4}</td>
<td>10^{-4}</td>
<td>10^{-6}</td>
<td>4.86</td>
<td>4.30</td>
<td>129</td>
<td>90</td>
<td>5133</td>
<td>91</td>
<td>129</td>
<td>0.0039</td>
</tr>
<tr>
<td></td>
<td>10^{-7}</td>
<td>10^{-7}</td>
<td>10^{-9}</td>
<td>7.55</td>
<td>6.71</td>
<td>173</td>
<td>137</td>
<td>9422</td>
<td>141</td>
<td>173</td>
<td>0.0078</td>
</tr>
<tr>
<td></td>
<td>10^{-10}</td>
<td>10^{-10}</td>
<td>10^{-12}</td>
<td>9.53</td>
<td>9.17</td>
<td>235</td>
<td>197</td>
<td>16067</td>
<td>201</td>
<td>235</td>
<td>0.0127</td>
</tr>
<tr>
<td>MEBDFI</td>
<td>10^{-4}</td>
<td>10^{-4}</td>
<td>10^{-6}</td>
<td>3.31</td>
<td>2.86</td>
<td>477</td>
<td>435</td>
<td>1761</td>
<td>83</td>
<td>83</td>
<td>0.0029</td>
</tr>
<tr>
<td></td>
<td>10^{-7}</td>
<td>10^{-7}</td>
<td>10^{-9}</td>
<td>6.11</td>
<td>5.60</td>
<td>1134</td>
<td>1083</td>
<td>3818</td>
<td>118</td>
<td>118</td>
<td>0.0059</td>
</tr>
<tr>
<td></td>
<td>10^{-10}</td>
<td>10^{-10}</td>
<td>10^{-12}</td>
<td>9.06</td>
<td>8.55</td>
<td>2135</td>
<td>2098</td>
<td>7215</td>
<td>208</td>
<td>208</td>
<td>0.0107</td>
</tr>
<tr>
<td>PSIDE-1</td>
<td>10^{-4}</td>
<td>10^{-4}</td>
<td></td>
<td>6.42</td>
<td>3.43</td>
<td>181</td>
<td>149</td>
<td>2811</td>
<td>57</td>
<td>57</td>
<td>0.0029</td>
</tr>
<tr>
<td></td>
<td>10^{-7}</td>
<td>10^{-7}</td>
<td></td>
<td>7.20</td>
<td>6.32</td>
<td>310</td>
<td>293</td>
<td>6141</td>
<td>52</td>
<td>52</td>
<td>0.0059</td>
</tr>
<tr>
<td></td>
<td>10^{-10}</td>
<td>10^{-10}</td>
<td></td>
<td>9.99</td>
<td>9.14</td>
<td>1000</td>
<td>990</td>
<td>15536</td>
<td>109</td>
<td>115</td>
<td>0.0156</td>
</tr>
<tr>
<td>RADAU</td>
<td>10^{-4}</td>
<td>10^{-4}</td>
<td>10^{-6}</td>
<td>4.48</td>
<td>4.28</td>
<td>210</td>
<td>172</td>
<td>1822</td>
<td>144</td>
<td>208</td>
<td>0.0010</td>
</tr>
<tr>
<td></td>
<td>10^{-7}</td>
<td>10^{-7}</td>
<td>10^{-9}</td>
<td>8.56</td>
<td>8.18</td>
<td>240</td>
<td>222</td>
<td>3508</td>
<td>187</td>
<td>238</td>
<td>0.0020</td>
</tr>
<tr>
<td></td>
<td>10^{-10}</td>
<td>10^{-10}</td>
<td>10^{-12}</td>
<td>10.63</td>
<td>9.24</td>
<td>209</td>
<td>176</td>
<td>6240</td>
<td>130</td>
<td>207</td>
<td>0.0039</td>
</tr>
<tr>
<td>VODE</td>
<td>10^{-4}</td>
<td>10^{-4}</td>
<td></td>
<td>3.29</td>
<td>3.08</td>
<td>545</td>
<td>487</td>
<td>779</td>
<td>19</td>
<td>117</td>
<td>0.0020</td>
</tr>
<tr>
<td></td>
<td>10^{-7}</td>
<td>10^{-7}</td>
<td></td>
<td>5.20</td>
<td>4.73</td>
<td>1614</td>
<td>1502</td>
<td>2145</td>
<td>30</td>
<td>223</td>
<td>0.0049</td>
</tr>
<tr>
<td></td>
<td>10^{-10}</td>
<td>10^{-10}</td>
<td></td>
<td>7.49</td>
<td>7.07</td>
<td>4350</td>
<td>4120</td>
<td>5266</td>
<td>72</td>
<td>516</td>
<td>0.0146</td>
</tr>
</tbody>
</table>

diagrams, we used: rtol = 10^{-(4+m/4)} , m = 0,1,\ldots,32; atol = rtol; h0 = 10^{-2} \cdot rtol for GAMD, MEBDFDAE, MEBDFI, RADAU and RADAU5.

Table II.8.3: Reference solution at the end of the integration interval.

\[
\begin{align*}
  y_1 & = 0.1706167732170483 \cdot 10^1 \\
  y_2 & = -0.8928097010247975 \cdot 10^0
\end{align*}
\]

References


Figure II.8.5: Behavior of the solution component $y_1$ over the integration interval (scaled equation)

Table II.8.4: Run characteristics.

<table>
<thead>
<tr>
<th>solver</th>
<th>rtol</th>
<th>atol</th>
<th>h0</th>
<th>mescd</th>
<th>scd</th>
<th>steps</th>
<th>accept</th>
<th>#f</th>
<th>#Jac</th>
<th>#LU</th>
<th>CPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>BIMD</td>
<td>10^{-4}</td>
<td>10^{-4}</td>
<td>10^{-6}</td>
<td>4.31</td>
<td>3.98</td>
<td>170</td>
<td>155</td>
<td>3684</td>
<td>151</td>
<td>170</td>
<td>0.0029</td>
</tr>
<tr>
<td></td>
<td>10^{-7}</td>
<td>10^{-7}</td>
<td>10^{-9}</td>
<td>9.06</td>
<td>8.73</td>
<td>301</td>
<td>293</td>
<td>7631</td>
<td>280</td>
<td>301</td>
<td>0.0059</td>
</tr>
<tr>
<td></td>
<td>10^{-10}</td>
<td>10^{-10}</td>
<td>10^{-12}</td>
<td>11.17</td>
<td>10.84</td>
<td>307</td>
<td>304</td>
<td>13339</td>
<td>292</td>
<td>307</td>
<td>0.0088</td>
</tr>
<tr>
<td>DDASL</td>
<td>10^{-4}</td>
<td>10^{-4}</td>
<td>10^{-6}</td>
<td>2.89</td>
<td>2.56</td>
<td>796</td>
<td>776</td>
<td>1260</td>
<td>127</td>
<td>0.0029</td>
<td></td>
</tr>
<tr>
<td></td>
<td>10^{-7}</td>
<td>10^{-7}</td>
<td>10^{-9}</td>
<td>5.89</td>
<td>5.57</td>
<td>1943</td>
<td>1912</td>
<td>2796</td>
<td>149</td>
<td>0.0078</td>
<td></td>
</tr>
<tr>
<td></td>
<td>10^{-10}</td>
<td>10^{-10}</td>
<td>10^{-12}</td>
<td>8.96</td>
<td>8.64</td>
<td>6166</td>
<td>6110</td>
<td>7973</td>
<td>223</td>
<td>0.0234</td>
<td></td>
</tr>
<tr>
<td>GAMD</td>
<td>10^{-4}</td>
<td>10^{-4}</td>
<td>10^{-6}</td>
<td>5.40</td>
<td>5.08</td>
<td>148</td>
<td>105</td>
<td>6999</td>
<td>105</td>
<td>148</td>
<td>0.0049</td>
</tr>
<tr>
<td></td>
<td>10^{-7}</td>
<td>10^{-7}</td>
<td>10^{-9}</td>
<td>6.52</td>
<td>6.19</td>
<td>163</td>
<td>133</td>
<td>12727</td>
<td>131</td>
<td>163</td>
<td>0.0098</td>
</tr>
<tr>
<td></td>
<td>10^{-10}</td>
<td>10^{-10}</td>
<td>10^{-12}</td>
<td>10.16</td>
<td>9.84</td>
<td>244</td>
<td>216</td>
<td>18095</td>
<td>215</td>
<td>244</td>
<td>0.0137</td>
</tr>
<tr>
<td>MEBDFI</td>
<td>10^{-4}</td>
<td>10^{-4}</td>
<td>10^{-6}</td>
<td>3.86</td>
<td>3.33</td>
<td>638</td>
<td>591</td>
<td>2179</td>
<td>92</td>
<td>92</td>
<td>0.0029</td>
</tr>
<tr>
<td></td>
<td>10^{-7}</td>
<td>10^{-7}</td>
<td>10^{-9}</td>
<td>6.99</td>
<td>6.67</td>
<td>1369</td>
<td>1317</td>
<td>4735</td>
<td>132</td>
<td>132</td>
<td>0.0078</td>
</tr>
<tr>
<td></td>
<td>10^{-10}</td>
<td>10^{-10}</td>
<td>10^{-12}</td>
<td>10.80</td>
<td>10.47</td>
<td>2862</td>
<td>2858</td>
<td>9489</td>
<td>287</td>
<td>287</td>
<td>0.0146</td>
</tr>
<tr>
<td>PSIDE-1</td>
<td>10^{-4}</td>
<td>10^{-4}</td>
<td>10^{-6}</td>
<td>5.70</td>
<td>5.38</td>
<td>235</td>
<td>166</td>
<td>4402</td>
<td>73</td>
<td>780</td>
<td>0.0049</td>
</tr>
<tr>
<td></td>
<td>10^{-7}</td>
<td>10^{-7}</td>
<td>10^{-9}</td>
<td>8.72</td>
<td>8.39</td>
<td>414</td>
<td>386</td>
<td>7896</td>
<td>75</td>
<td>908</td>
<td>0.0078</td>
</tr>
<tr>
<td></td>
<td>10^{-10}</td>
<td>10^{-10}</td>
<td>10^{-12}</td>
<td>11.40</td>
<td>11.07</td>
<td>1388</td>
<td>1365</td>
<td>23066</td>
<td>131</td>
<td>1360</td>
<td>0.0224</td>
</tr>
<tr>
<td>RADAU</td>
<td>10^{-4}</td>
<td>10^{-4}</td>
<td>10^{-6}</td>
<td>4.77</td>
<td>4.44</td>
<td>242</td>
<td>207</td>
<td>2214</td>
<td>165</td>
<td>231</td>
<td>0.0020</td>
</tr>
<tr>
<td></td>
<td>10^{-7}</td>
<td>10^{-7}</td>
<td>10^{-9}</td>
<td>8.28</td>
<td>7.95</td>
<td>186</td>
<td>149</td>
<td>5212</td>
<td>102</td>
<td>173</td>
<td>0.0029</td>
</tr>
<tr>
<td></td>
<td>10^{-10}</td>
<td>10^{-10}</td>
<td>10^{-12}</td>
<td>11.47</td>
<td>11.14</td>
<td>245</td>
<td>215</td>
<td>7589</td>
<td>148</td>
<td>224</td>
<td>0.0049</td>
</tr>
<tr>
<td>VODE</td>
<td>10^{-4}</td>
<td>10^{-4}</td>
<td>10^{-6}</td>
<td>2.93</td>
<td>2.61</td>
<td>788</td>
<td>702</td>
<td>1186</td>
<td>21</td>
<td>181</td>
<td>0.0029</td>
</tr>
<tr>
<td></td>
<td>10^{-7}</td>
<td>10^{-7}</td>
<td>10^{-9}</td>
<td>5.65</td>
<td>5.32</td>
<td>2375</td>
<td>2200</td>
<td>3091</td>
<td>41</td>
<td>345</td>
<td>0.0088</td>
</tr>
<tr>
<td></td>
<td>10^{-10}</td>
<td>10^{-10}</td>
<td>10^{-12}</td>
<td>8.42</td>
<td>8.09</td>
<td>6426</td>
<td>6058</td>
<td>7814</td>
<td>106</td>
<td>794</td>
<td>0.0215</td>
</tr>
</tbody>
</table>


Figure II.8.6: Work-precision diagram (scd versus CPU-time).
Figure II.8.7: Work-precision diagram (sd versus CPU-time).
Figure II.8.8: Work-precision diagram (nsecd versus CPU-time).
Figure II.8.9: Work-precision diagram (msed versus CPU-time).
Figure II.8.10: Work-precision diagram (sec versus CPU-time).
Figure II.8.11: Work-precision diagram (sod versus CPU-time).
Figure II.8.12: Work-precision diagram (mescd versus CPU-time).
Figure II.8.13: Work-precision diagram (mescd versus CPU-time).
9 Problem OREGO

9.1 General information

The problem consists of a stiff system of 3 non-linear Ordinary Differential Equations. The name Orego was given by Hairer & Wanner [HW96] and refers to the Oregonator model which is described by this ODE. The Oregonator model takes its name from the University of Oregon where in the 1972 Field, Kőrös & Noyes [FKN72] proposed this model for the Belousov–Zhabotinskii reaction. The INdAM-Bari Test Set group contributed this problem to the test set. The software part of the problem is in the file orego.f available at [MM08].

9.2 Mathematical description of the problem

The problem is of the form
\[
\frac{dy}{dt} = f(y), \quad y(0) = y_0,
\]
with
\[\begin{align*}
y & \in \mathbb{R}^3, \quad 0 \leq t \leq 360.
\end{align*}\]

The function \( f \) is defined by
\[
f(y) = \begin{pmatrix}
s(y_2 - y_1 y_2 + y_1 - q y_1^2) \\
\frac{1}{s}(-y_2 - y_1 y_2 + y_3) \\
w(y_1 - y_3)
\end{pmatrix}.
\]

The values of the parameters \( s, q \) and \( w \) are
\[\begin{align*}
s &= 77.27 \\
w &= 0.161 \\
q &= 8.375 \cdot 10^{-6}.
\end{align*}\]

The initial vector \( y_0 \) is given by \((1, 2, 3)^T\).

9.3 Origin of the problem

The OREGO problem originates from the celebrated Belousov–Zhabotinskii (BZ) reaction. When certain reactants, like bromous acid, bromide ion and cerium ion, are combined, they exhibit a chemical reaction which, after an induction period of inactivity, oscillates with change in structure and in color, from red to blue and vice versa.

The color changes are caused by alternating oxidation-reductions in which the cerium switches its oxidation state from Ce(III) to Ce(IV).

Field, Kőrös and Noyes formulated the following model for the most important parts of the kinetic mechanism that gives rise to oscillation in the BZ reaction. This mechanism can be summarized as three concurrent processes [Gra02]:

- the reduction of bromate (\( \text{BrO}_3^- \)) to bromine (Br) via the reducing agent bromide (\( \text{Br}^- \)). Bromomalonic acid (BrMA) is produced;

- the increase of hypobromous acid (\( \text{HBrO}_2 \)) at an accelerating rate and the production of Ce(IV).

Here we have a sudden change in color from red to blue;

- the reduction of Cerium catalyst Ce(IV) to Ce(III). Here we have a gradual change in color from blue to red.
Table II.9.1: Reference solution at the end of the integration interval.

<table>
<thead>
<tr>
<th>$t$</th>
<th>$X = y_1$</th>
<th>$Y = y_2$</th>
<th>$Z = y_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>360</td>
<td>0.1000814870318523 · 10$^1$</td>
<td>0.1228178521549917 · 10$^1$</td>
<td>0.1320554942846706 · 10$^1$</td>
</tr>
</tbody>
</table>

Table II.9.2: Failed runs.

<table>
<thead>
<tr>
<th>solver</th>
<th>$m$</th>
<th>reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>VODE</td>
<td>2,4</td>
<td>error test failed repeatedly</td>
</tr>
</tbody>
</table>

Then, from this mechanism the following Oregonator scheme is obtained

\[
\begin{align*}
A + Y & \rightarrow X + P & r = k_3 A Y \\
X + Y & \rightarrow 2P & r = k_2 X Y \\
A + X & \rightarrow 2X + 2Z & r = k_4 A X \\
2X & \rightarrow A + P & r = k_4 X^2 \\
B + Z & \rightarrow \frac{1}{f} Y & r = k, B Z
\end{align*}
\]

Here using the conventional notation as in [FKN72] the assignments and the effective concentration are

- hypobromous acid $[\text{HBrO}_2] = X = 5.025 \times 10^{-11}$
- bromide $[\text{Br}^-] = Y = 3.0 \times 10^{-7}$
- cerium $^{4}\text{[CE(IV)]} = Z = 2.412 \times 10^{-8}$
- bromate $[\text{BrO}_3^-] = A$
- all oxidizable organic species $[\text{Org}] = B$
- $[\text{HOBr}] = P$

The reaction rate equations for the intermediate species $X$, $Y$, and $Z$ are

\[
\begin{align*}
\frac{dX}{dt} &= s (Y - XY + X - qX^2) \\
\frac{dY}{dt} &= \frac{1}{s} (-Y - XY + fZ) \\
\frac{dZ}{dt} &= w(X - Z).
\end{align*}
\]

with $f = 1$, and $s$, $w$, and $q$ as in the previous subsection.

9.4 Numerical solution of the problem

Tables II.9.1, II.9.3 and Figures II.9.1–II.9.7 present the reference solution at the end of the integration interval, the run characteristics, the behavior of the solution over the integration interval and the work-precision diagrams, respectively. The reference solution was computed by RADAU on an Alphaserver DS20E, with a 667 MHz EV67 processor, using double precision work(1) = uround $= 1.01 \cdot 10^{-19}$, rtol = atol $= h_0 = 1.1 \cdot 10^{-18}$, atol $= h_0 = 1.1 \cdot 10^{-40}$. For the work-precision diagrams, we used: rtol $= 10^{-(4+m/4)}$, $m = 0, 1, \ldots, 32$; atol $= rtol$; $h_0 = 10^{-2}$·rtol for BMD, GAMD, MEBDFDAE, MEBDFI, RADAU and RADAU5. The failed runs are in Table II.9.2; listed are the name of the solver that failed, for which values of $m$ this happened, and the reason for failing.
### Table II.9.3: Run characteristics.

<table>
<thead>
<tr>
<th>solver</th>
<th>rtol</th>
<th>atol</th>
<th>h0</th>
<th>mescd</th>
<th>scd</th>
<th>steps</th>
<th>accept</th>
<th>#f</th>
<th>#Jac</th>
<th>#LU</th>
<th>CPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>BIMD</td>
<td>$10^{-4}$</td>
<td>$10^{-4}$</td>
<td>$10^{-4}$</td>
<td>3.85</td>
<td>3.85</td>
<td>235</td>
<td>224</td>
<td>4393</td>
<td>215</td>
<td>235</td>
<td>0.0049</td>
</tr>
<tr>
<td></td>
<td>$10^{-7}$</td>
<td>$10^{-7}$</td>
<td>$10^{-9}$</td>
<td>7.87</td>
<td>7.86</td>
<td>347</td>
<td>339</td>
<td>9629</td>
<td>334</td>
<td>347</td>
<td>0.0107</td>
</tr>
<tr>
<td></td>
<td>$10^{-10}$</td>
<td>$10^{-10}$</td>
<td>$10^{-12}$</td>
<td>11.29</td>
<td>11.29</td>
<td>373</td>
<td>367</td>
<td>16863</td>
<td>339</td>
<td>373</td>
<td>0.0176</td>
</tr>
<tr>
<td>DDASSL</td>
<td>$10^{-4}$</td>
<td>$10^{-4}$</td>
<td>$10^{-4}$</td>
<td>2.62</td>
<td>2.62</td>
<td>889</td>
<td>813</td>
<td>1505</td>
<td>124</td>
<td></td>
<td>0.0039</td>
</tr>
<tr>
<td></td>
<td>$10^{-7}$</td>
<td>$10^{-7}$</td>
<td>$10^{-7}$</td>
<td>5.58</td>
<td>5.57</td>
<td>2725</td>
<td>2671</td>
<td>4210</td>
<td>189</td>
<td></td>
<td>0.0137</td>
</tr>
<tr>
<td></td>
<td>$10^{-10}$</td>
<td>$10^{-10}$</td>
<td>$10^{-10}$</td>
<td>8.66</td>
<td>8.66</td>
<td>8192</td>
<td>8098</td>
<td>11119</td>
<td>274</td>
<td></td>
<td>0.0381</td>
</tr>
<tr>
<td>GAMD</td>
<td>$10^{-4}$</td>
<td>$10^{-4}$</td>
<td>$10^{-4}$</td>
<td>3.61</td>
<td>3.61</td>
<td>219</td>
<td>162</td>
<td>8510</td>
<td>163</td>
<td>219</td>
<td>0.0088</td>
</tr>
<tr>
<td></td>
<td>$10^{-7}$</td>
<td>$10^{-7}$</td>
<td>$10^{-9}$</td>
<td>6.90</td>
<td>6.89</td>
<td>251</td>
<td>205</td>
<td>16050</td>
<td>208</td>
<td>251</td>
<td>0.0176</td>
</tr>
<tr>
<td></td>
<td>$10^{-10}$</td>
<td>$10^{-10}$</td>
<td>$10^{-10}$</td>
<td>9.50</td>
<td>9.50</td>
<td>291</td>
<td>268</td>
<td>22034</td>
<td>270</td>
<td>291</td>
<td>0.0234</td>
</tr>
<tr>
<td>MEBDFI</td>
<td>$10^{-4}$</td>
<td>$10^{-4}$</td>
<td>$10^{-4}$</td>
<td>3.34</td>
<td>3.33</td>
<td>733</td>
<td>687</td>
<td>2707</td>
<td>103</td>
<td>103</td>
<td>0.0049</td>
</tr>
<tr>
<td></td>
<td>$10^{-7}$</td>
<td>$10^{-7}$</td>
<td>$10^{-9}$</td>
<td>6.39</td>
<td>6.39</td>
<td>1586</td>
<td>1529</td>
<td>5399</td>
<td>174</td>
<td>174</td>
<td>0.0107</td>
</tr>
<tr>
<td></td>
<td>$10^{-10}$</td>
<td>$10^{-10}$</td>
<td>$10^{-12}$</td>
<td>9.59</td>
<td>9.59</td>
<td>3248</td>
<td>3232</td>
<td>10754</td>
<td>345</td>
<td>345</td>
<td>0.0205</td>
</tr>
<tr>
<td>PSIDE-1</td>
<td>$10^{-4}$</td>
<td>$10^{-4}$</td>
<td>$10^{-4}$</td>
<td>4.74</td>
<td>4.73</td>
<td>221</td>
<td>178</td>
<td>4096</td>
<td>128</td>
<td>836</td>
<td>0.0059</td>
</tr>
<tr>
<td></td>
<td>$10^{-7}$</td>
<td>$10^{-7}$</td>
<td>$10^{-7}$</td>
<td>7.06</td>
<td>7.06</td>
<td>441</td>
<td>407</td>
<td>9235</td>
<td>148</td>
<td>1164</td>
<td>0.0117</td>
</tr>
<tr>
<td></td>
<td>$10^{-10}$</td>
<td>$10^{-10}$</td>
<td>$10^{-10}$</td>
<td>10.77</td>
<td>10.47</td>
<td>1450</td>
<td>1412</td>
<td>26255</td>
<td>219</td>
<td>1788</td>
<td>0.0332</td>
</tr>
<tr>
<td>RADAU</td>
<td>$10^{-4}$</td>
<td>$10^{-4}$</td>
<td>$10^{-6}$</td>
<td>3.42</td>
<td>3.12</td>
<td>268</td>
<td>222</td>
<td>3416</td>
<td>200</td>
<td>267</td>
<td>0.0029</td>
</tr>
<tr>
<td></td>
<td>$10^{-7}$</td>
<td>$10^{-7}$</td>
<td>$10^{-9}$</td>
<td>7.48</td>
<td>7.48</td>
<td>267</td>
<td>216</td>
<td>6859</td>
<td>192</td>
<td>265</td>
<td>0.0059</td>
</tr>
<tr>
<td></td>
<td>$10^{-10}$</td>
<td>$10^{-10}$</td>
<td>$10^{-12}$</td>
<td>9.83</td>
<td>9.82</td>
<td>261</td>
<td>202</td>
<td>12917</td>
<td>176</td>
<td>257</td>
<td>0.0098</td>
</tr>
<tr>
<td>VODE</td>
<td>$10^{-4}$</td>
<td>$10^{-4}$</td>
<td>$10^{-4}$</td>
<td>2.15</td>
<td>2.15</td>
<td>1196</td>
<td>1101</td>
<td>1820</td>
<td>38</td>
<td>236</td>
<td>0.0049</td>
</tr>
<tr>
<td></td>
<td>$10^{-7}$</td>
<td>$10^{-7}$</td>
<td>$10^{-7}$</td>
<td>4.73</td>
<td>4.73</td>
<td>3083</td>
<td>2858</td>
<td>4348</td>
<td>64</td>
<td>454</td>
<td>0.0117</td>
</tr>
<tr>
<td></td>
<td>$10^{-10}$</td>
<td>$10^{-10}$</td>
<td>$10^{-10}$</td>
<td>7.51</td>
<td>7.51</td>
<td>7890</td>
<td>7430</td>
<td>9903</td>
<td>133</td>
<td>970</td>
<td>0.0293</td>
</tr>
</tbody>
</table>
Figure II.9.1: Behavior of the solution component $y_1$ over the integration interval.

Figure II.9.2: Behavior of the solution component $y_2$ over the integration interval.

Figure II.9.3: Behavior of the solution component $y_3$ over the integration interval.
References


Figure II.9.4: Work-precision diagram (sod versus CPU-time).
Figure II.9.5: Work-precision diagram (scd versus CPU-time).
Figure II.9.6: Work-precision diagram (mescd versus CPU-time).
Figure II.9.7: Work-precision diagram (mescd versus CPU-time).
10 Problem ROBER

10.1 General information
The problem consists of a stiff system of 3 non-linear ordinary differential equations. It was proposed by H.H. Robertson in 1966 [Rob66]. The name ROBER was given by Hairer & Wanner [HW96]. The INdAM-Bari Test Set group contributed this problem to the test set. The software part of the problem is in the file rober.f available at [MM08].

10.2 Mathematical description of the problem
The problem is of the form

\[
\frac{dy}{dt} = f(y), \quad y(0) = y_0,
\]

with

\[ y \in \mathbb{R}^3, \quad t \in [0, T], \]

The function \( f \) is defined by

\[
f(y) = \begin{pmatrix}
-0.04y_1 + 10^4y_2y_3 \\
0.04y_1 - 10^4y_2y_3 - 3 \cdot 10^7y_2^2 \\
3 \cdot 10^7y_2^2
\end{pmatrix}
\]

The initial vector \( y_0 \) is given by \((1, 0, 0)^T\).

10.3 Origin of the problem
The ROBER problem describes the kinetics of an autocatalytic reaction given by Robertson (1966) [Rob66]. The structure of the reactions is given in Table II.10.1, where \( k_1, k_2, k_3 \) are the rate constants and \( A, B \) and \( C \) are the chemical species involved. Under some idealized conditions [Aik85] and the assumption that the mass action law is applied for the rate functions, the following mathematical model consisting of a set of three ODEs can be set up

\[
\begin{pmatrix}
y_1' \\
y_2' \\
y_3'
\end{pmatrix} = \begin{pmatrix}
-k_1y_1 + k_3y_2y_3 \\
k_1y_1 - k_2y_2^2 - k_3y_2y_3 \\
k_2y_2^2
\end{pmatrix},
\]

with \((y_1(0), y_2(0), y_3(0))^T = (y_{01}, y_{02}, y_{03})^T\), where \( y_1, y_2, y_3 \) denote the concentrations of \( A, B \) and \( C \) respectively and \( y_{01}, y_{02}, y_{03} \) are the concentrations at time \( t = 0 \).

The ROBER problem is very popular in numerical studies [Eds74] and it is often used as a test problem in the stiff integrators comparisons. The numerical values of the rate constants used in the test problem are \( k_1 = 0.04, k_2 = 3 \cdot 10^7 \) and \( k_3 = 10^4 \), and the initial concentrations \( y_{01} = 1, y_{02} = \ldots \)
0, \ y_0 = 0. The large difference among the reaction rate constants is the reason for stiffness. As is typical for problems arising in chemical kinetics this special system has a small very quick initial transient. This phase is followed by a very smooth variation of the components where a large stepsize would be appropriate for a numerical method.

Originally the problem was posed on the interval \( 0 \leq t \leq 40 \), but it is convenient to integrate it on much longer intervals. As a matter of fact Hindmarsh discovered that many codes fail if \( t \) becomes very large. In this case if \( y_2 \) accidentally becomes negative, it then tends to \(-\infty\), causing overflow (see [HW96]).

![Graphs of y1 and y2 over time](image.jpg)

Figure II.10.1: Behavior of the solution on \([0, 10^{11}]\)

### 10.4 Numerical solution of the problem

The system of ODEs is integrated for \( t \in [0, 10^{11}] \). Tables II.10.3–II.10.4 and Figures II.10.1–II.10.5 present the reference solution at the end of the integration interval, the run characteristics, the behavior of the components of the solution over part of the integration interval and the work-precision diagrams, respectively. The reference solution was computed by RADAU on an Alphaserver DS20E, with a 667 MHz EV67 processor, using double precision \( \text{work}(1) = \text{uround} = 1.01 \cdot 10^{-19} \), \( \text{rtol} = \text{atol} = \)
ODE - Problem Rober

Table II.10.2: Failed runs.

<table>
<thead>
<tr>
<th>solver</th>
<th>m</th>
<th>reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>DASGL</td>
<td>5, 8, 10, 11, 13, ... 32</td>
<td>error test failed repeatedly</td>
</tr>
</tbody>
</table>

$h_0 = 1.1 \cdot 10^{-18}$. For the work-precision diagrams, we used: $rtol = 10^{-4m/4}$, $m = 0, 1, \ldots, 32$; $atol = 10^{-4}rtol$; $h_0 = 10^{-2}rtol$ for BIMD, GAMD, MEBDFDAE, MEBDFI, RADAU and RADAU5. The failed runs are in Table II.10.2; listed are the name of the solver that failed, for which values of $m$ this happened, and the reason for failing.

Table II.10.3: Reference solution at the end of the integration interval.

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$y_1$</td>
<td>0.3083340149701255 $\cdot 10^{-7}$</td>
</tr>
<tr>
<td>$y_2$</td>
<td>0.833397010334713 $\cdot 10^{-13}$</td>
</tr>
<tr>
<td>$y_3$</td>
<td>0.999999791665050</td>
</tr>
</tbody>
</table>

Table II.10.4: Run characteristics.

<table>
<thead>
<tr>
<th>solver</th>
<th>rtol</th>
<th>atol</th>
<th>$h_0$</th>
<th>mescd</th>
<th>scd</th>
<th>steps</th>
<th>accept</th>
<th>#f</th>
<th>#Jac</th>
<th>#LU</th>
<th>CPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>BIMD</td>
<td>$10^{-4}$</td>
<td>$10^{-8}$</td>
<td>$10^{-6}$</td>
<td>6.70</td>
<td>3.02</td>
<td>101</td>
<td>100</td>
<td>1904</td>
<td>97</td>
<td>101</td>
<td>0.0020</td>
</tr>
<tr>
<td></td>
<td>$10^{-7}$</td>
<td>$10^{-11}$</td>
<td>$10^{-9}$</td>
<td>10.07</td>
<td>6.39</td>
<td>132</td>
<td>131</td>
<td>3883</td>
<td>125</td>
<td>132</td>
<td>0.0039</td>
</tr>
<tr>
<td></td>
<td>$10^{-10}$</td>
<td>$10^{-14}$</td>
<td>$10^{-12}$</td>
<td>13.70</td>
<td>10.12</td>
<td>150</td>
<td>157</td>
<td>6529</td>
<td>148</td>
<td>159</td>
<td>0.0068</td>
</tr>
<tr>
<td>DDASGL</td>
<td>$10^{-4}$</td>
<td>$10^{-8}$</td>
<td>$10^{-6}$</td>
<td>4.51</td>
<td>0.83</td>
<td>473</td>
<td>453</td>
<td>682</td>
<td>62</td>
<td>62</td>
<td>0.0020</td>
</tr>
<tr>
<td></td>
<td>$10^{-7}$</td>
<td>$10^{-11}$</td>
<td>$10^{-9}$</td>
<td>7.15</td>
<td>3.47</td>
<td>1278</td>
<td>1252</td>
<td>1549</td>
<td>108</td>
<td>108</td>
<td>0.0059</td>
</tr>
<tr>
<td>GAMD</td>
<td>$10^{-4}$</td>
<td>$10^{-8}$</td>
<td>$10^{-6}$</td>
<td>6.27</td>
<td>2.59</td>
<td>62</td>
<td>62</td>
<td>2165</td>
<td>62</td>
<td>62</td>
<td>0.0020</td>
</tr>
<tr>
<td></td>
<td>$10^{-7}$</td>
<td>$10^{-11}$</td>
<td>$10^{-9}$</td>
<td>9.94</td>
<td>6.05</td>
<td>93</td>
<td>91</td>
<td>4883</td>
<td>89</td>
<td>92</td>
<td>0.0059</td>
</tr>
<tr>
<td></td>
<td>$10^{-10}$</td>
<td>$10^{-14}$</td>
<td>$10^{-12}$</td>
<td>12.41</td>
<td>8.73</td>
<td>169</td>
<td>169</td>
<td>9427</td>
<td>166</td>
<td>169</td>
<td>0.0107</td>
</tr>
<tr>
<td>MEBDFI</td>
<td>$10^{-4}$</td>
<td>$10^{-8}$</td>
<td>$10^{-6}$</td>
<td>6.25</td>
<td>2.56</td>
<td>401</td>
<td>398</td>
<td>1299</td>
<td>72</td>
<td>72</td>
<td>0.0029</td>
</tr>
<tr>
<td></td>
<td>$10^{-7}$</td>
<td>$10^{-11}$</td>
<td>$10^{-9}$</td>
<td>8.95</td>
<td>5.27</td>
<td>804</td>
<td>802</td>
<td>2611</td>
<td>98</td>
<td>98</td>
<td>0.0049</td>
</tr>
<tr>
<td></td>
<td>$10^{-10}$</td>
<td>$10^{-14}$</td>
<td>$10^{-12}$</td>
<td>11.53</td>
<td>7.85</td>
<td>1614</td>
<td>1612</td>
<td>5252</td>
<td>186</td>
<td>186</td>
<td>0.0107</td>
</tr>
<tr>
<td>PSIDE-1</td>
<td>$10^{-4}$</td>
<td>$10^{-8}$</td>
<td>$10^{-6}$</td>
<td>5.75</td>
<td>2.07</td>
<td>56</td>
<td>55</td>
<td>1295</td>
<td>36</td>
<td>224</td>
<td>0.0020</td>
</tr>
<tr>
<td></td>
<td>$10^{-7}$</td>
<td>$10^{-11}$</td>
<td>$10^{-9}$</td>
<td>9.03</td>
<td>5.35</td>
<td>158</td>
<td>154</td>
<td>3128</td>
<td>39</td>
<td>496</td>
<td>0.0039</td>
</tr>
<tr>
<td></td>
<td>$10^{-10}$</td>
<td>$10^{-14}$</td>
<td>$10^{-12}$</td>
<td>11.29</td>
<td>7.61</td>
<td>570</td>
<td>563</td>
<td>9772</td>
<td>50</td>
<td>744</td>
<td>0.0127</td>
</tr>
<tr>
<td>RADAU</td>
<td>$10^{-4}$</td>
<td>$10^{-8}$</td>
<td>$10^{-6}$</td>
<td>6.74</td>
<td>3.06</td>
<td>114</td>
<td>112</td>
<td>811</td>
<td>108</td>
<td>113</td>
<td>0.0010</td>
</tr>
<tr>
<td></td>
<td>$10^{-7}$</td>
<td>$10^{-11}$</td>
<td>$10^{-9}$</td>
<td>9.35</td>
<td>5.67</td>
<td>112</td>
<td>110</td>
<td>1852</td>
<td>104</td>
<td>112</td>
<td>0.0020</td>
</tr>
<tr>
<td></td>
<td>$10^{-10}$</td>
<td>$10^{-14}$</td>
<td>$10^{-12}$</td>
<td>11.21</td>
<td>7.53</td>
<td>108</td>
<td>106</td>
<td>3420</td>
<td>92</td>
<td>108</td>
<td>0.0029</td>
</tr>
<tr>
<td>VODE</td>
<td>$10^{-4}$</td>
<td>$10^{-8}$</td>
<td>$10^{-6}$</td>
<td>3.66</td>
<td>-0.02</td>
<td>593</td>
<td>576</td>
<td>830</td>
<td>12</td>
<td>100</td>
<td>0.0020</td>
</tr>
<tr>
<td></td>
<td>$10^{-7}$</td>
<td>$10^{-11}$</td>
<td>$10^{-9}$</td>
<td>6.70</td>
<td>3.02</td>
<td>1292</td>
<td>1220</td>
<td>1686</td>
<td>22</td>
<td>199</td>
<td>0.0049</td>
</tr>
<tr>
<td></td>
<td>$10^{-10}$</td>
<td>$10^{-14}$</td>
<td>$10^{-12}$</td>
<td>9.59</td>
<td>5.91</td>
<td>3306</td>
<td>3138</td>
<td>3873</td>
<td>56</td>
<td>408</td>
<td>0.0127</td>
</tr>
</tbody>
</table>
References


Figure II.10.2: Work-precision diagram (sec versus CPU-time.)
Figure II.10.3: Work-precision diagram (sec versus CPU-time.)
Figure H.10.4: Work-precision diagram (mescd versus CPU-time.
Figure II.10.5: Work-precision diagram (mescl versus CPU-time.)
11 Problem E5

11.1 General information

The problem consists of a stiff system of 4 non-linear ordinary differential equations. It was proposed by Datta in 1967. The name E5 was given by Enright, Hull and Lindberg (1975) [EHL75]. The formulation and data have been taken from [HW96]. The Bari Test Set group contributed this problem to the test set. The software part of the problem is in the file e5.f available at [MM08].

11.2 Mathematical description of the problem

The problem is of the form

\[ \frac{dy}{dt} = f(y), \quad y(0) = y_0, \]

with \( y \in \mathbb{R}^4, \quad t \in [0, T], \)

The function \( f \) is defined by

\[ f(y) = \begin{pmatrix} -Ay_1 - By_1 y_3 \\ Ay_1 - M C y_2 y_3 \\ Ay_1 - By_1 y_3 - M C y_2 y_3 + C y_4 \\ By_1 y_3 - C y_4 \end{pmatrix} \] (II.11.1)

where \( A = 7.89 \cdot 10^{-10}, \quad B = 1.1 \cdot 10^7, \quad C = 1.13 \cdot 10^3, \quad \text{and} \quad M = 10^6. \)

The initial vector \( y_0 \) is given by \((1.76 \cdot 10^{-3}, 0, 0, 0)^T.\)

11.3 Origin of the problem

The E5 problem is a model for chemical pyrolysis studied by Datta in 1967 and describes a reaction involving six reactants. The reaction scheme is given in Table II.11.1, where \( A_i, \quad i = 1, \ldots, 6 \) are the chemical species and \( k_1, \quad k_2, \quad k_3, \quad k_4 \) the rate of reaction constants. According to mass action kinetics,

\[
\begin{array}{cccc}
A_1 & k_1 & A_2 + A_3 \\
A_2 + A_3 & k_2 & A_5 \\
A_1 + A_3 & k_3 & A_4 \\
A_4 & k_4 & A_3 + A_6
\end{array}
\]

Table II.11.1: Reaction scheme for problem E5

the corresponding mathematical model is the following

\[
\begin{aligned}
y_1' &= -k_1 y_1 - k_3 y_1 y_3 \\
y_2' &= k_1 y_1 - k_2 y_2 y_3 \\
y_3' &= k_1 y_1 - k_2 y_2 y_3 - k_3 y_1 y_3 + k_4 y_4 \\
y_4' &= k_3 y_1 y_3 - k_4 y_4 \\
y_5' &= k_2 y_2 y_3 \\
y_6' &= k_4 y_4
\end{aligned}
\] (II.11.2)
where \( y_i \) are the concentrations of the reactants \( A_i \). This set of ODEs is one of the test problems in the stiff integrator comparison by Enright, Hull and Lindberg (1975) [EHL75]. The rate constants used in the test problem were \( k_1 = 7.89 \cdot 10^{-10} \), \( k_2 = 1.13 \cdot 10^8 \), \( k_3 = 1.1 \cdot 10^7 \), \( k_4 = 1.13 \cdot 10^3 \) and the initial values were all set to zero except for \( y_1(0) = 1.76 \cdot 10^{-3} \). The vastly different rates of reaction that occur in the same system are the cause for stiffness. With rate constants inserted in (II.11.2) the system (II.11.1) is obtained [Aik85]. Note that the differential equation possesses the invariant \( y_2 - y_3 - y_4 = 0 \) and it is recommended to use the relation \( y_3' = y_2' - y_4' \) in the function subroutine in order to avoid eventual cancellation of digits [HW96].

Although the problem was originally posed on the interval \( 0 \leq t \leq 1000 \), it is often integrated on a much longer interval because of the interesting properties of the solutions for \( t \) large [HW96]. In 1981 Shampine [Sha81] observed that since the solution components are badly scaled (\( |y_1| \leq 2 \cdot 10^{-3} \) and the magnitude of all the other components doesn’t exceed \( 4 \cdot 10^{-10} \)), a scalar absolute error control is quite unsuitable and a componentwise scaled absolute error control would be recommendable for this problem.

### 11.4 Numerical solution of the problem

The system of ODEs is integrated for \( t \in [0, 10^{13}] \). Tables II.11.3–II.11.4 present the reference solution at the end of the integration interval and the run characteristics, Figures II.11.1–II.11.3 present the behavior of the components of the solution over the integration interval and the work-precision diagrams, respectively. The work precision diagrams were computed using the mescd since the solution at the end of the integration interval is very close to zero. For the same reason, the scd column in Table II.11.4 has been skipped. The reference solution was computed by RADAU on an Alphaserver DS20E, with a 667 MHz EV67 processor, using double precision \( \text{work}(1) = \text{uround} = 1.01 \cdot 10^{-10}, rtol = h0 = 1.1 \cdot 10^{-18}, atol = 1.1 \cdot 10^{-40} \). For the work-precision diagrams, we used: \( rtol = 10^{-4+m/4} \), \( m = 0, 1, \ldots , 32 \); \( atol = 1.7 \cdot 10^{-24} \); \( h0 = 10^{-2} \cdot rtol \) for BIMD, GAMD, MEBDF-DAE, MEBDFI, RADAU and RADAU5. The failed runs are in Table II.11.2; listed are the name of the solver that failed, for which values of \( m \) this happened, and the reason for failing.

### Table II.11.2: Failed runs.

<table>
<thead>
<tr>
<th>solver</th>
<th>( m )</th>
<th>reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>DASSL</td>
<td>0,1,2,6,7,8,9,11,13, 14,16,\ldots ,32</td>
<td>error test failed repeatedly</td>
</tr>
</tbody>
</table>

### Table II.11.3: Reference solution at the end of the integration interval.

| \( y_1 \) | 0.1112903278711829 \cdot 10^{-200} |
| \( y_2 \) | 0.8867655517642120 \cdot 10^{-22} |
| \( y_3 \) | 0.8854814626268838 \cdot 10^{-22} |
| \( y_4 \) | 0.0000000000000000 |

### References

Figure II.11.1: Behavior of the solution over the integration interval in double logarithmic scale.


Table II.11.4: Run characteristics.

<table>
<thead>
<tr>
<th>solver</th>
<th>rtol</th>
<th>atol</th>
<th>h0</th>
<th>mescd</th>
<th>socd</th>
<th>steps</th>
<th>accept</th>
<th>#f</th>
<th>#Jac</th>
<th>#LU</th>
<th>CPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>BIMD</td>
<td>10^{-4}</td>
<td>1.110^{-24}</td>
<td>10^{-6}</td>
<td>4.98</td>
<td>2.70</td>
<td>169</td>
<td>169</td>
<td>3438</td>
<td>162</td>
<td>169</td>
<td>0.0049</td>
</tr>
<tr>
<td></td>
<td>10^{-7}</td>
<td>1.110^{-24}</td>
<td>10^{-9}</td>
<td>8.34</td>
<td>3.05</td>
<td>174</td>
<td>174</td>
<td>6409</td>
<td>168</td>
<td>174</td>
<td>0.0068</td>
</tr>
<tr>
<td></td>
<td>10^{-10}</td>
<td>1.110^{-24}</td>
<td>10^{-12}</td>
<td>11.77</td>
<td>3.48</td>
<td>287</td>
<td>287</td>
<td>10726</td>
<td>282</td>
<td>287</td>
<td>0.0156</td>
</tr>
<tr>
<td>DDASSL</td>
<td>10^{-7}</td>
<td>1.110^{-24}</td>
<td></td>
<td>7.55</td>
<td>2.26</td>
<td>2516</td>
<td>2468</td>
<td>3443</td>
<td>148</td>
<td></td>
<td>0.0137</td>
</tr>
<tr>
<td>GAMD</td>
<td>10^{-4}</td>
<td>1.110^{-24}</td>
<td>10^{-6}</td>
<td>5.52</td>
<td>3.24</td>
<td>103</td>
<td>101</td>
<td>4977</td>
<td>99</td>
<td>103</td>
<td>0.0068</td>
</tr>
<tr>
<td></td>
<td>10^{-7}</td>
<td>1.110^{-24}</td>
<td>10^{-9}</td>
<td>8.19</td>
<td>2.90</td>
<td>125</td>
<td>125</td>
<td>9167</td>
<td>122</td>
<td>125</td>
<td>0.0117</td>
</tr>
<tr>
<td></td>
<td>10^{-10}</td>
<td>1.110^{-24}</td>
<td>10^{-12}</td>
<td>11.13</td>
<td>2.84</td>
<td>154</td>
<td>154</td>
<td>13497</td>
<td>154</td>
<td>154</td>
<td>0.0166</td>
</tr>
<tr>
<td>MEBDFI</td>
<td>10^{-4}</td>
<td>1.110^{-24}</td>
<td>10^{-6}</td>
<td>5.16</td>
<td>2.87</td>
<td>653</td>
<td>644</td>
<td>2145</td>
<td>86</td>
<td>86</td>
<td>0.0049</td>
</tr>
<tr>
<td></td>
<td>10^{-7}</td>
<td>1.110^{-24}</td>
<td>10^{-9}</td>
<td>8.13</td>
<td>2.85</td>
<td>1048</td>
<td>1043</td>
<td>3423</td>
<td>122</td>
<td>122</td>
<td>0.0088</td>
</tr>
<tr>
<td></td>
<td>10^{-10}</td>
<td>1.110^{-24}</td>
<td>10^{-12}</td>
<td>10.56</td>
<td>2.27</td>
<td>1782</td>
<td>1779</td>
<td>5823</td>
<td>188</td>
<td>188</td>
<td>0.0137</td>
</tr>
<tr>
<td>PSIDE-1</td>
<td>10^{-4}</td>
<td>1.110^{-24}</td>
<td></td>
<td>3.94</td>
<td>1.65</td>
<td>137</td>
<td>112</td>
<td>3160</td>
<td>69</td>
<td>544</td>
<td>0.0049</td>
</tr>
<tr>
<td></td>
<td>10^{-7}</td>
<td>1.110^{-24}</td>
<td></td>
<td>7.99</td>
<td>2.71</td>
<td>255</td>
<td>243</td>
<td>5181</td>
<td>173</td>
<td>944</td>
<td>0.0078</td>
</tr>
<tr>
<td></td>
<td>10^{-10}</td>
<td>1.110^{-24}</td>
<td></td>
<td>11.46</td>
<td>3.18</td>
<td>707</td>
<td>704</td>
<td>13278</td>
<td>286</td>
<td>1512</td>
<td>0.0195</td>
</tr>
<tr>
<td>RADAU</td>
<td>10^{-4}</td>
<td>1.110^{-24}</td>
<td>10^{-6}</td>
<td>4.72</td>
<td>2.43</td>
<td>100</td>
<td>99</td>
<td>2220</td>
<td>80</td>
<td>100</td>
<td>0.0029</td>
</tr>
<tr>
<td></td>
<td>10^{-7}</td>
<td>1.110^{-24}</td>
<td>10^{-9}</td>
<td>8.42</td>
<td>3.14</td>
<td>148</td>
<td>145</td>
<td>3123</td>
<td>118</td>
<td>144</td>
<td>0.0039</td>
</tr>
<tr>
<td></td>
<td>10^{-10}</td>
<td>1.110^{-24}</td>
<td>10^{-12}</td>
<td>11.79</td>
<td>3.51</td>
<td>142</td>
<td>132</td>
<td>5733</td>
<td>106</td>
<td>141</td>
<td>0.0059</td>
</tr>
<tr>
<td>VODE</td>
<td>10^{-4}</td>
<td>1.110^{-24}</td>
<td></td>
<td>3.17</td>
<td>0.88</td>
<td>1238</td>
<td>1149</td>
<td>1718</td>
<td>27</td>
<td>260</td>
<td>0.0059</td>
</tr>
<tr>
<td></td>
<td>10^{-7}</td>
<td>1.110^{-24}</td>
<td></td>
<td>6.67</td>
<td>1.39</td>
<td>2655</td>
<td>2484</td>
<td>3464</td>
<td>47</td>
<td>397</td>
<td>0.0107</td>
</tr>
<tr>
<td></td>
<td>10^{-10}</td>
<td>1.110^{-24}</td>
<td></td>
<td>9.69</td>
<td>1.41</td>
<td>4003</td>
<td>3836</td>
<td>4776</td>
<td>70</td>
<td>458</td>
<td>0.0156</td>
</tr>
</tbody>
</table>
Figure II.11.2: Work-precision diagram (messd versus CPU-time).
Figure II.11.3: Work-precision diagram (mesod versus CPU-time).
12 Chemical Akzo Nobel problem

12.1 General information

This IVP is a stiff system of 6 non-linear DAEs of index 1 and has been taken from [Sto98]. The parallel-IVP-algorithm group of CWI contributed this problem to the test set in collaboration with W.J.H. Stortelder. We acknowledge the remarks of Dotsikas Ioannis, which improved the formulation of this problem considerably. The software part of the problem is in the file chemakzo.f available at [MM08].

12.2 Mathematical description of the problem

The problem is of the form

\[ M \frac{dy}{dt} = f(y), \quad y(0) = y_0, \quad y'(0) = y'_0, \]

with

\[ y \in \mathbb{R}^6, \quad 0 \leq t \leq 180. \]

The matrix \( M \) is of rank 5 and given by

\[ M = \begin{pmatrix}
1 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
\end{pmatrix} \]

and the function \( f \) by

\[ f(y) = \begin{pmatrix}
-2r_1 + r_2 & -r_3 & -r_4 & -\frac{1}{2}r_5 + F_m \\
-2r_1 & r_2 & -r_3 & -r_4 \\
r_1 & -r_3 & +r_3 & -2r_4 \\
r_2 & -r_3 & +r_3 & +r_5 \\
K_s \cdot y_1 \cdot y_4 - y_0 \\
\end{pmatrix}, \]

where the \( r_i \) and \( F_m \) are auxiliary variables, given by

\[ \begin{align*}
    r_1 &= k_1 \cdot y_1^4 \cdot y_2^2, \\
r_2 &= k_2 \cdot y_3 \cdot y_4, \\
r_3 &= k_3 \cdot y_1 \cdot y_5, \\
r_4 &= \frac{k_4}{K} \cdot y_1 \cdot y_4^2, \\
r_5 &= k_4 \cdot y_6^2 \cdot y_2^3, \\
F_m &= kIA \cdot \left( \frac{p(\text{CO}_2)}{H} - y_2 \right). 
\end{align*} \]

The values of the parameters \( k_1, k_2, k_3, k_4, K, kIA, p(\text{CO}_2) \) and \( H \) are

\[ \begin{align*}
    k_1 &= 18.7, & k_4 &= 0.42, & K_s &= 115.83, \\
k_2 &= 0.58, & K &= 34.4, & p(\text{CO}_2) &= 0.9, \\
k_3 &= 0.09, & kIA &= 3.3, & H &= 737. 
\end{align*} \]
The consistent initial vectors are 
\[ y_0 = \begin{pmatrix} 0.444, 0.00123, 0, 0.007, 0, K_s \cdot y_{0,1} \cdot y_{0,4} \end{pmatrix}^T \quad y'_0 = f(y_0). \]

It is clear from the definition of \( r_1 \) and \( r_5 \) that the function \( f \) cannot be evaluated for negative values of \( y_2 \). In the Fortran subroutine that defines \( f \), we set \( \text{IERR}=-1 \) if \( y_2 < 0 \) to prevent this situation. See page IV-ix of the description of the software part of the test set for more details on \( \text{IERR} \).

### 12.3 Origin of the problem

The problem originates from Akzo Nobel Central Research in Arnhem, The Netherlands. It describes a chemical process, in which 2 species, FLB and ZHU, are mixed, while carbon dioxide is continuously added. The resulting species of importance is ZLA. In the interest of commercial competition, the names of the chemical species are fictitious. The reaction equations, as given by Akzo Nobel [CBS93], are given in Figure II.12.1. The last reaction equation describes an equilibrium

![Reaction scheme for Chemical Akzo Nobel problem.](image)

\[ K_s = \frac{[\text{FLB.ZHU}]}{[\text{FLB}] \cdot [\text{ZHU}]} \]

The value of \( K_s \) plays a role in parameter estimation. The other equations describe reactions with velocities given by

\[ r_1 = k_1 \cdot [\text{FLB}]^4 \cdot [\text{CO}_2]^2, \]

\[ r_2 = k_2 \cdot [\text{FLBT}] \cdot [\text{ZHU}], \]

\[ r_3 = \frac{k_3}{K} \cdot [\text{FLB}] \cdot [\text{ZLA}], \]

\[ r_4 = k_3 \cdot [\text{FLB}] \cdot [\text{ZHU}]^2, \]

\[ r_5 = k_4 \cdot [\text{FLB.ZHU}]^2 \cdot [\text{CO}_2]^2, \]

respectively. Here the square brackets \([\cdot]\) denote concentrations. One would expect from the reaction scheme in Figure II.12.1, that reaction velocities \( r_1 \), \( r_4 \) and \( r_5 \) would read

\[ r_1 = k_1 \cdot [\text{FLB}]^2 \cdot [\text{CO}_2]^2, \]

\[ r_4 = k_3 \cdot [\text{FLB}] \cdot [\text{ZHU}]^2 \cdot [\text{CO}_2], \]

\[ r_5 = k_4 \cdot [\text{FLB.ZHU}] \cdot [\text{CO}_2]^2. \]
However, it turns out that the chemical process under consideration is modeled more appropriately using (II.12.1)-(II.12.3).

The inflow of carbon dioxide per volume unit is denoted by $F_m$, and satisfies

$$F_m = kLA \cdot \left( \frac{p(\text{CO}_2)}{H} - [\text{CO}_2] \right),$$

where $kLA$ is the mass transfer coefficient, $H$ is the Henry constant and $p(\text{CO}_2)$ is the partial carbon dioxide pressure. $p(\text{CO}_2)$ is assumed to be independent of $[\text{CO}_2]$. The parameters $k_1$, $k_2$, $k_3$, $k_4$, $K$, $kLA$, $K_s$, $H$ and $p(\text{CO}_2)$ are given constants.

The process is started by mixing 0.444 mol/liter FLB with 0.007 mol/liter ZHU. The concentration of carbon dioxide at the beginning is 0.00123 mol/liter. Initially, no other species are present. The simulation is performed on the time interval $[0, 180 \text{ minutes}]$.

Identifying the concentrations $[\text{FLB}]$, $[\text{CO}_2]$, $[\text{FLB}]$, $[\text{ZHU}]$, $[\text{ZLA}]$, $[\text{FLB}:\text{ZHU}]$ with $y_1, \ldots, y_6$, respectively, one easily arrives at the mathematical formulation of the preceding section.

### 12.4 Numerical solution of the problem

Tables II.12.1-II.12.2 and Figures II.12.2-II.12.6 present the reference solution at the end of the integration interval, the run characteristics, the behavior of the solution over the integration interval and the work-precision diagrams, respectively. The reference solution was computed by PSIDE on a Cray C90, using double precision, $\text{rtol} = \text{atol} \cdot 10^{-10}$. To get more insight in the exact behavior of the second component, we included a plot of $y_2$ on $[0, 3]$ in Figure II.12.2. For the work-precision diagrams, we used: $\text{rtol} = 10^{-8+m/4}$, $m = 0, 1, \ldots, 20$; $\text{atol} = \text{rtol}$; $h_0 = \text{rtol}$ for BIMD, GAMD, MEBDFDAE, MEBDFI, RADAU and RADAU5. The failed runs are in Table II.12.3; listed are the name of the solver that failed, for which values of $m$ this happened, and the reason for failing.

*Apart from $H$, which is generally known, all parameters have been estimated by W. Stortelder [Sto95].

<table>
<thead>
<tr>
<th>solver</th>
<th>rtol</th>
<th>atol</th>
<th>$h_0$</th>
<th>mesed</th>
<th>scd</th>
<th>steps</th>
<th>accept</th>
<th>#f</th>
<th>#Jac</th>
<th>#LU</th>
<th>CPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>BIMD</td>
<td>$10^{-10}$</td>
<td>$10^{-10}$</td>
<td>$10^{-10}$</td>
<td>12.39</td>
<td>10.61</td>
<td>41</td>
<td>41</td>
<td>1177</td>
<td>41</td>
<td>41</td>
<td>0.0039</td>
</tr>
<tr>
<td>DDASSL</td>
<td>$10^{-10}$</td>
<td>$10^{-10}$</td>
<td>$10^{-10}$</td>
<td>10.04</td>
<td>8.33</td>
<td>522</td>
<td>515</td>
<td>649</td>
<td>38</td>
<td></td>
<td>0.0039</td>
</tr>
<tr>
<td>GAMD</td>
<td>$10^{-10}$</td>
<td>$10^{-10}$</td>
<td>$10^{-10}$</td>
<td>11.89</td>
<td>9.82</td>
<td>35</td>
<td>35</td>
<td>1737</td>
<td>35</td>
<td>35</td>
<td>0.0039</td>
</tr>
<tr>
<td>MEBDFI</td>
<td>$10^{-10}$</td>
<td>$10^{-10}$</td>
<td>$10^{-10}$</td>
<td>11.42</td>
<td>9.76</td>
<td>274</td>
<td>273</td>
<td>916</td>
<td>32</td>
<td>32</td>
<td>0.0029</td>
</tr>
<tr>
<td>PSIDE-1</td>
<td>$10^{-10}$</td>
<td>$10^{-10}$</td>
<td>$10^{-10}$</td>
<td>11.41</td>
<td>9.91</td>
<td>87</td>
<td>85</td>
<td>1671</td>
<td>15</td>
<td>204</td>
<td>0.0039</td>
</tr>
<tr>
<td>RADAU</td>
<td>$10^{-10}$</td>
<td>$10^{-10}$</td>
<td>$10^{-10}$</td>
<td>10.71</td>
<td>8.39</td>
<td>43</td>
<td>41</td>
<td>696</td>
<td>30</td>
<td>43</td>
<td>0.0010</td>
</tr>
</tbody>
</table>
Table II.12.3: Failed runs.

<table>
<thead>
<tr>
<th>solver</th>
<th>m</th>
<th>reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSIDE-1</td>
<td>14,16,17,18,19,20</td>
<td>stepsize too small</td>
</tr>
</tbody>
</table>

References


Figure II.12.2: Behavior of the solution over the integration interval.
Figure II.12.3: Work-precision diagram (sod versus CPU-time).
Figure II.12.4: Work-precision diagram (sod versus CPU-time).
Figure II.12.5: Work-precision diagram (mescd versus CPU-time).
Figure II.12.6: Work-precision diagram (msecd versus CPU-time).
13 Andrews’ squeezing mechanism

13.1 General information

The problem is a non-stiff second order DAE of index 3, consisting of 14 differential and 13 algebraic equations. It has been promoted as a test problem by Giles [Gil78] and Manning [Man81]. The formulation here corresponds to the one presented in Hairer & Wanner [HW96]. The parallel-IVP-algorithm group of CWI contributed this problem to the test set. The software part of the problem is in the file andrews.f available at [MM08].

13.2 Mathematical description of the problem

The problem is of the form

\[ K \frac{dy}{dt} = \phi(y), \quad y(0) = y_0, \quad y'(0) = y'_0, \quad (II.13.1) \]

where

\[ y = \begin{pmatrix} q \\ \dot{q} \\ \lambda \end{pmatrix}, \quad K = \begin{bmatrix} I & O & O \\ O & I & O \\ O & O & O \end{bmatrix}, \quad \phi(y) = \begin{pmatrix} \dot{q} \\ \dot{\lambda} \\ M(q)\dot{q} - f(q, \dot{q}) + G^T(q)\lambda \\ g(q) \end{pmatrix}. \]

Here,

\[ 0 \leq t \leq 0.03, \]
\[ q \in \mathbb{R}^7, \]
\[ \lambda \in \mathbb{R}^6, \]
\[ M : \mathbb{R}^7 \rightarrow \mathbb{R}^{7 \times 7}, \]
\[ f : \mathbb{R}^{14} \rightarrow \mathbb{R}^7, \]
\[ g : \mathbb{R}^7 \rightarrow \mathbb{R}^6, \]
\[ G = \frac{\partial y}{\partial \dot{q}}. \]

The function \( M(q) = (M_{ij}(q)) \) is given by:

\[
\begin{align*}
M_{11}(q) &= m_1 \cdot r^2 + m_2 (rr^2 - 2da \cdot rr \cdot \cos q_2 + da^2) + I_1 + I_2, \\
M_{21}(q) &= M_{12}(q) = m_2 (da^2 - da \cdot rr \cdot \cos q_2) + I_2, \\
M_{22}(q) &= m_2 \cdot da^2 + I_2, \\
M_{33}(q) &= m_3 (sa^2 + sb^2) + I_3, \\
M_{44}(q) &= m_4 (e - ea)^2 + I_4, \\
M_{54}(q) &= M_{45}(q) = m_4 ((e - ea)^2 + zt(e - ea) \sin q_4) + I_4, \\
M_{55}(q) &= m_4 (zt^2 + 2zt(e - ea) \sin q_4 + (e - ea)^2) + m_5 (ta^2 + tb^2) + I_4 + I_5, \\
M_{66}(q) &= m_6 (zf - fa)^2 + I_6, \\
M_{76}(q) &= M_{67}(q) = m_6 ((zf - fa)^2 - u(zf - fa) \sin q_6) + I_6, \\
M_{77}(q) &= m_6 ((zf - fa)^2 - 2u(zf - fa) \sin q_6 + u^2) + m_7 (ua^2 + ub^2) + I_6 + I_7, \\
M_{ij}(q) &= 0 \text{ for all other cases.}
\end{align*}
\]
The function $f = (f_i(q, \dot{q}))$ reads:
\[
\begin{align*}
    f_1(q, \dot{q}) &= \text{mom} - m_2 \cdot d \cdot r \cdot \dot{q}_2 (\dot{q}_2 + 2\dot{q}_1) \sin q_2, \\
    f_2(q, \dot{q}) &= m_2 \cdot d \cdot r \cdot \dot{q}_2^2 \cdot \sin q_2, \\
    f_3(q, \dot{q}) &= F_x (sc \cdot \cos q_3 - sd \cdot \sin q_3) + F_y (sd \cdot \cos q_3 + sc \cdot \sin q_3), \\
    f_4(q, \dot{q}) &= m_4 \cdot z (e - ca) \dot{q}_5^2 \cdot \cos q_4, \\
    f_5(q, \dot{q}) &= -m_4 \cdot z (e - ca) \dot{q}_4 (\dot{q}_4 + 2\dot{q}_5) \cos q_4, \\
    f_6(q, \dot{q}) &= -m_5 \cdot u (z f - fa) \dot{q}_6^2 \cdot \cos q_6, \\
    f_7(q, \dot{q}) &= m_6 \cdot u (z f - fa) \dot{q}_6 (\dot{q}_6 + 2\dot{q}_7) \cos q_6.
\end{align*}
\]

$F_x$ and $F_y$ are defined by:
\[
\begin{align*}
    F_x &= F(xd - xc), \\
    F_y &= F(yd - yc), \\
    F &= -c_0 (L - l_0)/L, \\
    L &= \sqrt{(xd - xc)^2 + (yd - yc)^2}, \\
    xd &= sd \cdot \cos q_3 + sc \cdot \sin q_3 + xb, \\
    yd &= sd \cdot \sin q_3 - sc \cdot \cos q_3 + yb.
\end{align*}
\]

The function $g = (g_i(q))$ is given by:
\[
\begin{align*}
    g_1(q) &= rr \cdot \cos q_1 - d \cdot \cos (q_1 + q_2) - ss \cdot \sin q_3 - xb, \\
    g_2(q) &= rr \cdot \sin q_1 - d \cdot \sin (q_1 + q_2) + ss \cdot \cos q_3 - yb, \\
    g_3(q) &= rr \cdot \cos q_1 - d \cdot \cos (q_1 + q_2) - e \cdot \sin (q_4 + q_6) - zt \cdot \cos q_5 - xa, \\
    g_4(q) &= rr \cdot \sin q_1 - d \cdot \sin (q_1 + q_2) + e \cdot \cos (q_4 + q_6) - zt \cdot \sin q_5 - ya, \\
    g_5(q) &= rr \cdot \cos q_1 - d \cdot \cos (q_1 + q_2) - z f \cdot \cos (q_6 + q_7) - u \cdot \sin q_7 - xa, \\
    g_6(q) &= rr \cdot \sin q_1 - d \cdot \sin (q_1 + q_2) - z f \cdot \sin (q_6 + q_7) + u \cdot \cos q_7 - ya.
\end{align*}
\]

The constants arising in these formulas are given by:
\[
\begin{array}{cccc}
    m_1 &=& 0.04325 & I_1 &=& 2.194 \cdot 10^{-6} & ss &=& 0.035 \\
    m_2 &=& 0.00365 & I_2 &=& 4.410 \cdot 10^{-7} & sa &=& 0.01874 \\
    m_3 &=& 0.02373 & I_3 &=& 5.255 \cdot 10^{-5} & sb &=& 0.01043 \\
    m_4 &=& 0.00706 & I_4 &=& 5.667 \cdot 10^{-7} & sc &=& 0.018 \\
    m_5 &=& 0.07050 & I_5 &=& 1.169 \cdot 10^{-5} & sd &=& 0.02 \\
    m_6 &=& 0.00706 & I_6 &=& 5.667 \cdot 10^{-7} & sa &=& 0.02308 \\
    m_7 &=& 0.05498 & I_7 &=& 1.912 \cdot 10^{-5} & tb &=& 0.00916 \\
    xa &=& -0.06934 & d &=& 0.028 & u &=& 0.04 \\
    ya &=& -0.00227 & da &=& 0.0115 & ua &=& 0.01228 \\
    xb &=& -0.03635 & e &=& 0.02 & ub &=& 0.00449 \\
    yb &=& 0.03273 & ea &=& 0.01421 & z f &=& 0.02 \\
    xc &=& 0.014 & rr &=& 0.007 & zt &=& 0.04 \\
    yc &=& 0.072 & ra &=& 0.00092 & fa &=& 0.01421 \\
    c_0 &=& 4.530 & l_0 &=& 0.07785 & mom &=& 0.033
\end{array}
\]

Consistent initial values are
\[
y_0 = (q_0, \dot{q}_0, \ddot{q}_0, \lambda_0)^T \quad \text{and} \quad y'_0 = (\ddot{q}_0, \dddot{q}_0, \dddot{q}_0, \lambda_0)^T.
\]
where

\[
q_0 = \begin{pmatrix}
-0.061713890014276449635894845801 \\
0.45527981916307038025912382449 \\
0.226683901658858846747185609 \\
0.48736497954384253022559893330 \\
-0.226683901658858846747185609 \\
1.23054744454982119249735015568
\end{pmatrix},
\]

\[
\dot{q}_0 = (0, 0, 0, 0, 0, 0)^T,
\]

\[
\ddot{q}_0 = \begin{pmatrix}
14222.443919954138705911625887 \\
-10666.8329399655854029433719415 \\
0 \\
0 \\
0 \\
0
\end{pmatrix},
\]

\[
\lambda_0 = \begin{pmatrix}
98.5665703962410896057654982170 \\
-6.12268834425566265503114393122 \\
0 \\
0 \\
0 \\
0
\end{pmatrix},
\]

\[
\dot{\lambda}_0 = (0, 0, 0, 0, 0, 0)^T.
\]

The index of the \(q, \dot{q}, \ddot{q}\) and \(\lambda\) components in \(y\) is 1, 2, 3 and 3, respectively.

### 13.3 Origin of the problem

Formulation (II.13.1) can be rewritten as

\[
M(q)\ddot{q} = f(q, \dot{q}) - G^T(q)\lambda, \\
0 = g(q),
\]

which is the general form of a constrained mechanical system. More precisely, the problem describes the motion of 7 rigid bodies connected by joints without friction. It was promoted by [Gil78] and [Man81] as a test problem for numerical codes. [HW96, pp. 530–536] describes the system and the modeling process in full detail.

### 13.4 Numerical solution of the problem

The Jacobian \(\frac{\partial \phi}{\partial y}\), needed by the numerical solver, was approximated by

\[
\begin{pmatrix}
0 & I & 0 & 0 \\
0 & 0 & I & 0 \\
0 & 0 & 0 & M \\
0 & 0 & M & G^T \\
G & 0 & 0 & 0
\end{pmatrix},
\]

which means that we neglect the derivatives of \(f(q, \dot{q})\) as well as those of \(M(q)\) and \(G(q)\). Note that the evaluation of such a Jacobian does not cost anything, because \(M\) and \(G\) are already computed in the evaluation of \(\phi\). However, we did not exploit this in the numerical computations.
Tables II.13.2–II.13.3 and Figures II.13.1–II.13.5 present the reference solution at the end of the integration interval, the run characteristics, the behavior of the solution over the integration interval and the work-precision diagrams, respectively. In computing the scd values, only the first seven components were considered, since they refer to the physically important quantities, in computing the mescd values all the components were considered. The reference solution was computed on the Cray C90, using PSIDE with Cray double precision and atol = rtol = 10^{-14}. For the work-precision diagrams, we used: rtol = 10^{-\left(4+m/8\right)}, m = 0, 1, \ldots, 48; atol = rtol; h0 = rtol for BIMD, GAMD, MEBDFDAE, MEBDFI, RADAU and RADAU5.

The failed runs are in Table II.13.1; listed are the name of the solver that failed, for which values of $m$ this happened, and the reason for failing.

\begin{table}[h]
\centering
\begin{tabular}{|c|c|c|}
\hline
solver & $m$ & reason \\
\hline
GAMD & 3, 4, 6 & stepsizes too small \\
RADAU & 55, 56 & stepsizes too small \\
\hline
\end{tabular}
\caption{Failed runs.}
\end{table}

References


[MM08] F. Mazzia and C. Magherini. \textit{Test Set for Initial Value Problem Solvers, release 2.4}. Department of Mathematics, University of Bari and INdAM, Research Unit of Bari, February 2008. Available at \url{http://www.dm.uniba.it/~testset}.

\begin{table}[h]
\centering
\begin{tabular}{|c|c|}
\hline
$y_1$ & 0.158107719629904 \cdot 10^2 \\
$y_2$ & -0.1575637105984298 \cdot 10^2 \\
$y_3$ & 0.4082224013073101 \cdot 10^{-1} \\
$y_4$ & -0.5347301163226948 \\
$y_5$ & 0.524409658805304 \\
$y_6$ & 0.5347301163226948 \\
$y_7$ & 0.1048080741042263 \cdot 10^1 \\
\hline
\end{tabular}
\caption{Reference solution (first 7 components) at the end of the integration interval.}
\end{table}
Table II.13.3: Run characteristics.

<table>
<thead>
<tr>
<th>Solver</th>
<th>rtol</th>
<th>atol</th>
<th>h0</th>
<th>mescd</th>
<th>sscd</th>
<th>steps</th>
<th>accept</th>
<th>#f</th>
<th>#Jac</th>
<th>#LU</th>
<th>CPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>BIMD</td>
<td>10^{-4}</td>
<td>10^{-4}</td>
<td>10^{-4}</td>
<td>0.27</td>
<td>3.05</td>
<td>46</td>
<td>41</td>
<td>1034</td>
<td>41</td>
<td>46</td>
<td>0.0185</td>
</tr>
<tr>
<td></td>
<td>10^{-7}</td>
<td>10^{-7}</td>
<td>10^{-7}</td>
<td>2.82</td>
<td>5.38</td>
<td>122</td>
<td>122</td>
<td>2553</td>
<td>122</td>
<td>122</td>
<td>0.0459</td>
</tr>
<tr>
<td>GAMD</td>
<td>10^{-4}</td>
<td>10^{-4}</td>
<td>10^{-4}</td>
<td>0.35</td>
<td>2.82</td>
<td>82</td>
<td>58</td>
<td>2281</td>
<td>58</td>
<td>82</td>
<td>0.0293</td>
</tr>
<tr>
<td></td>
<td>10^{-7}</td>
<td>10^{-7}</td>
<td>10^{-7}</td>
<td>1.53</td>
<td>4.54</td>
<td>128</td>
<td>116</td>
<td>5176</td>
<td>116</td>
<td>128</td>
<td>0.0093</td>
</tr>
<tr>
<td>MEBDFI</td>
<td>10^{-4}</td>
<td>10^{-4}</td>
<td>10^{-4}</td>
<td>-1.11</td>
<td>0.37</td>
<td>118</td>
<td>108</td>
<td>466</td>
<td>23</td>
<td>23</td>
<td>0.0078</td>
</tr>
<tr>
<td></td>
<td>10^{-7}</td>
<td>10^{-7}</td>
<td>10^{-7}</td>
<td>1.25</td>
<td>3.50</td>
<td>300</td>
<td>287</td>
<td>1222</td>
<td>38</td>
<td>38</td>
<td>0.0195</td>
</tr>
<tr>
<td>PSIDE-1</td>
<td>10^{-4}</td>
<td>10^{-4}</td>
<td></td>
<td>0.22</td>
<td>2.95</td>
<td>92</td>
<td>75</td>
<td>1675</td>
<td>52</td>
<td>52</td>
<td>0.0410</td>
</tr>
<tr>
<td></td>
<td>10^{-7}</td>
<td>10^{-7}</td>
<td></td>
<td>2.10</td>
<td>4.98</td>
<td>113</td>
<td>93</td>
<td>2637</td>
<td>63</td>
<td>63</td>
<td>0.0615</td>
</tr>
<tr>
<td>RADAU</td>
<td>10^{-4}</td>
<td>10^{-4}</td>
<td>10^{-4}</td>
<td>-0.84</td>
<td>1.36</td>
<td>96</td>
<td>56</td>
<td>810</td>
<td>54</td>
<td>54</td>
<td>0.0137</td>
</tr>
<tr>
<td></td>
<td>10^{-7}</td>
<td>10^{-7}</td>
<td>10^{-7}</td>
<td>0.47</td>
<td>4.45</td>
<td>114</td>
<td>95</td>
<td>1292</td>
<td>90</td>
<td>90</td>
<td>0.0195</td>
</tr>
</tbody>
</table>

Figure II.13.1: Behavior of the solution modulo $2\pi$ over the integration interval.
Figure II.13.2: Work-precision diagram (scd versus CPU-time).
Figure II.13.3: Work-precision diagram (sod versus CPU-time).
Figure II.13.4: Work-precision diagram (mescd versus CPU-time).
Figure II.13.5: Work-precision diagram (nescd versus CPU-time).
DAE – Andrews’ squeezing mechanism
14 Transistor amplifier

14.1 General information

The problem is a stiff DAE of index 1 consisting of 8 equations. P. Rentrop has received it from K. Glashoff & H.J. Oberle and has documented it in [RRS89]. The formulation presented here has been taken from [HLR89]. The parallel-IVP-algorithm group of CWI contributed this problem to the test set.

The software part of the problem is in the file transamp.f available at [MM08].

14.2 Mathematical description of the problem

The problem is of the form

\[ M \frac{dy}{dt} = f(t,y), \quad y(0) = y_0, \quad y'(0) = y'_0, \]

with

\[ y \in \mathbb{R}^8, \quad 0 \leq t \leq 0.2 \]

The matrix \( M \) is of rank 5 and given by

\[
M = \begin{pmatrix}
-C_1 & C_1 & 0 & 0 & 0 & 0 & 0 & 0 \\
C_1 & -C_1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -C_2 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -C_3 & C_3 & 0 & 0 & 0 \\
0 & 0 & 0 & C_3 & -C_3 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -C_4 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -C_5 & C_5 \\
0 & 0 & 0 & 0 & 0 & 0 & C_5 & -C_5 \\
\end{pmatrix},
\]

and the function \( f \) by

\[
f(t, y) = \begin{pmatrix}
-\frac{U_e(t)}{R_0} + \frac{y_8}{R_0} \\
-\frac{V_b}{R_2} + y_2\left(\frac{1}{R_1} + \frac{1}{R_3}\right) - (\alpha - 1)g(y_2 - y_3) \\
g(y_2 - y_3) + \frac{y_3}{R_5} \\
-\frac{V_b}{R_4} + \frac{y_6}{R_5} + \alpha g(y_2 - y_3) \\
-\frac{V_b}{R_5} + y_5\left(\frac{1}{R_0} + \frac{1}{R_6}\right) - (\alpha - 1)g(y_5 - y_6) \\
g(y_5 - y_6) + \frac{y_6}{R_6} \\
-\frac{V_b}{R_6} + \frac{y_5}{R_6} + \alpha g(y_5 - y_6) \\
\frac{y_6}{R_6}
\end{pmatrix},
\]

where \( g \) and \( U_e \) are auxiliary functions given by

\[ g(x) = \beta (e^{\frac{x}{T_T}} - 1) \quad \text{and} \quad U_e(t) = 0.1 \sin (200\pi t). \]
The values of the technical parameters are:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$U_b$</td>
<td>6</td>
</tr>
<tr>
<td>$U_F$</td>
<td>0.026</td>
</tr>
<tr>
<td>$\alpha$</td>
<td>0.99</td>
</tr>
<tr>
<td>$\beta$</td>
<td>$10^{-6}$</td>
</tr>
<tr>
<td>$R_0$</td>
<td>1000</td>
</tr>
<tr>
<td>$R_k$</td>
<td>9000  for $k = 1, \ldots, 9$,</td>
</tr>
<tr>
<td>$C_k$</td>
<td>$k \cdot 10^{-6}$ for $k = 1, \ldots, 5$.</td>
</tr>
</tbody>
</table>

Consistent initial values at $t = 0$ are

$$
y_0 = \begin{pmatrix}
0 \\
\frac{U_b}{R_2} + 1 \\
\frac{U_b}{R_2} + 1 \\
\frac{U_b}{R_2} + 1 \\
\frac{U_b}{R_2} + 1 \\
\frac{U_b}{R_2} + 1 \\
0
\end{pmatrix}, \quad y'_0 = \begin{pmatrix}
51.338775 \\
51.338775 \\
-24.9757667 \\
-24.9757667 \\
-10.0564433 \\
-10.0564433 \\
\end{pmatrix}
$$

The first, fourth and seventh component of $y'_0$ were determined numerically. All components of $y$ are of index 1.

The definition of the function $g(x)$ in (II.14.1) may cause overflow if $x_U$ becomes too large. In the Fortran subroutines feval and jeval that define the function $f$ and the partial derivatives of $f$ with respect to $y$, respectively, we set $IERR=-1$ if $x_U > 300$ to prevent this situation. See page IV-ix of the description of the software part of the test set for more details on $IERR$.

### 14.3 Origin of the problem

The problem originates from electrical circuit analysis. It is a model for the transistor amplifier. The diagram of the circuit is given in Figure II.14.1. Here $U_e$ is the input signal and $U_b$ is the amplified signal.
output voltage. The circuit contains two transistors of the form depicted in Figure II.14.2. As a simple model for the behavior of the transistors we assume that the currents through the gate, drain and source, which are denoted by \( I_G, I_D \) and \( I_S \), respectively, are

\[
I_G = (1 - \alpha)g(U_G - U_S),
\]

\[
I_D = \alpha g(U_G - U_S),
\]

\[
I_S = g(U_G - U_S),
\]

where \( U_G \) and \( U_S \) denote the voltage at the gate and source, respectively, and \( \alpha = 0.99 \). For the function \( g \) we take

\[
g(U_i - U_j) = \beta(e^{\frac{U_i - U_j}{U_F}} - 1),
\]

where \( \beta = 10^{-6} \) and \( U_F = 0.026 \).

To formulate the governing equations, Kirchhoff’s Current Law is used in each numbered node. This law states that the total sum of all currents entering a node must be zero. All currents passing through the circuit components can be expressed in terms of the unknown voltages \( U_1, \ldots, U_8 \). Consider for instance node 1. The current \( I_{C_1} \) passing through capacitor \( C_1 \) is given by

\[
I_{C_1} = \frac{d}{dt}(C_1(U_2 - U_1)),
\]

and the current \( I_{R_0} \) passing through the resistor \( R_0 \) by

\[
I_{R_0} = \frac{U_2 - U_1}{R_0}.
\]

Here, the currents are directed towards node 1 if the current is positive. A similar derivation for the
other nodes gives the system:

node 1: \[ \frac{d}{dt}(C_1(U_2 - U_1)) + \frac{U_1(t)}{R_c} - \frac{U_1}{R_c} = 0, \]

node 2: \[ \frac{d}{dt}(C_1(U_1 - U_2)) + \frac{U_1}{R_c} - U_2 \left( \frac{1}{R_1} + \frac{1}{R_2} \right) + (\alpha - 1)g(U_2 - U_3) = 0, \]

node 3: \[ -\frac{d}{dt}(C_2U_3) + g(U_2 - U_3) - \frac{U_2}{R_3} = 0, \]

node 4: \[ -\frac{d}{dt}(C_3(U_4 - U_5)) + \frac{U_3}{R_4} - \frac{U_3}{R_4} - \alpha g(U_2 - U_3) = 0, \]

node 5: \[ \frac{d}{dt}(C_3(U_4 - U_5)) + \frac{U_4}{R_5} - U_5 \left( \frac{1}{R_5} + \frac{1}{R_6} \right) + (\alpha - 1)g(U_5 - U_6) = 0, \]

node 6: \[ -\frac{d}{dt}(C_4U_6) + g(U_5 - U_6) - \frac{U_5}{R_7} = 0, \]

node 7: \[ -\frac{d}{dt}(C_5(U_7 - U_8)) + \frac{U_6}{R_8} - \frac{U_6}{R_8} - \alpha g(U_5 - U_6) = 0. \]

node 8: \[ -\frac{d}{dt}(C_5(U_7 - U_8)) + \frac{U_7}{R_9} = 0. \]

The input signal \( U_e(t) \) is

\[ U_e(t) = 0.1 \sin(200\pi t). \]

To arrive at the mathematical formulation of the preceding subsection, one just has to identify \( U_i \) with \( y_i \).

From the plot of output signal \( U_8 = y(8) \) in Figure II.14.2 we see that the amplitude of the input signal \( U_e \) is indeed amplified.

### 14.4 Numerical solution of the problem

Tables II.14.2–II.14.3 and Figures II.14.3–II.14.4 present the reference solution at the end of the integration interval, the run characteristics, the behavior of the solution over the integration interval and the work-precision diagrams, respectively. The reference solution was computed on the Cray C90, using PSIDE with Cray double precision and \( \text{atol} = \text{rtol} = 10^{-14} \). For the work-precision diagrams, we used: \( \text{rtol} = 10^{-4(4+m/8)} \), \( m = 0, \ldots, 40 \); \( \text{atol} = \text{rtol} \); \( h0 = 10^{-5} \cdot \text{rtol} \) for BIMD, GAMD, MEBDFDAE, MEBDFI, RADAU and RADAU5.

The failed runs are in Table II.14.1; listed are the name of the solver that failed, for which values of \( m \) this happened, and the reason for failing.

### References

Table II.14.2: Reference solution at the end of the integration interval.

<table>
<thead>
<tr>
<th>$y_1$</th>
<th>$y_2$</th>
<th>$y_3$</th>
<th>$y_4$</th>
<th>$y_5$</th>
<th>$y_6$</th>
<th>$y_7$</th>
<th>$y_8$</th>
</tr>
</thead>
<tbody>
<tr>
<td>-0.5562145012262709 · 10^{-2}</td>
<td>0.3006522471903042 · 10</td>
<td>0.284995878608128 · 10</td>
<td>0.2926422536206241 · 10</td>
<td>0.2704617863010554 · 10</td>
<td>0.2761837778393145 · 10</td>
<td>0.4770927631616772 · 10</td>
<td>0.123695868091548 · 10</td>
</tr>
</tbody>
</table>

Table II.14.3: Run characteristics.

<table>
<thead>
<tr>
<th>solver</th>
<th>rtol</th>
<th>atol</th>
<th>h0</th>
<th>mescd</th>
<th>scd</th>
<th>steps</th>
<th>accept</th>
<th>#f</th>
<th>#Jac</th>
<th>#LU</th>
<th>CPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>BIMD</td>
<td>10^{-4}</td>
<td>10^{-4}</td>
<td>10^{-6}</td>
<td>5.85</td>
<td>5.63</td>
<td>466</td>
<td>408</td>
<td>8423</td>
<td>408</td>
<td>466</td>
<td>0.0322</td>
</tr>
<tr>
<td></td>
<td>10^{-7}</td>
<td>10^{-7}</td>
<td>10^{-9}</td>
<td>8.62</td>
<td>8.34</td>
<td>618</td>
<td>575</td>
<td>19632</td>
<td>575</td>
<td>618</td>
<td>0.0752</td>
</tr>
<tr>
<td>DDASSL</td>
<td>10^{-4}</td>
<td>10^{-4}</td>
<td>10^{-6}</td>
<td>4.60</td>
<td>3.08</td>
<td>9759</td>
<td>6026</td>
<td>18381</td>
<td>739</td>
<td>618</td>
<td>0.1113</td>
</tr>
<tr>
<td></td>
<td>10^{-7}</td>
<td>10^{-7}</td>
<td>10^{-9}</td>
<td>7.24</td>
<td>5.49</td>
<td>40810</td>
<td>23859</td>
<td>77402</td>
<td>33678</td>
<td>0.4743</td>
<td></td>
</tr>
<tr>
<td>GAMD</td>
<td>10^{-4}</td>
<td>10^{-4}</td>
<td>10^{-6}</td>
<td>6.30</td>
<td>5.83</td>
<td>373</td>
<td>276</td>
<td>17204</td>
<td>276</td>
<td>373</td>
<td>0.0517</td>
</tr>
<tr>
<td></td>
<td>10^{-7}</td>
<td>10^{-7}</td>
<td>10^{-9}</td>
<td>8.58</td>
<td>7.37</td>
<td>374</td>
<td>325</td>
<td>34320</td>
<td>326</td>
<td>374</td>
<td>0.1064</td>
</tr>
<tr>
<td>MEBDFI</td>
<td>10^{-4}</td>
<td>10^{-4}</td>
<td>10^{-6}</td>
<td>5.06</td>
<td>4.80</td>
<td>1580</td>
<td>1486</td>
<td>5949</td>
<td>256</td>
<td>256</td>
<td>0.0303</td>
</tr>
<tr>
<td></td>
<td>10^{-7}</td>
<td>10^{-7}</td>
<td>10^{-9}</td>
<td>7.25</td>
<td>6.99</td>
<td>3628</td>
<td>3513</td>
<td>13324</td>
<td>419</td>
<td>419</td>
<td>0.0703</td>
</tr>
<tr>
<td>PSIDE-1</td>
<td>10^{-4}</td>
<td>10^{-4}</td>
<td>10^{-6}</td>
<td>5.02</td>
<td>4.76</td>
<td>516</td>
<td>362</td>
<td>9742</td>
<td>253</td>
<td>2008</td>
<td>0.0351</td>
</tr>
<tr>
<td></td>
<td>10^{-7}</td>
<td>10^{-7}</td>
<td>10^{-9}</td>
<td>7.50</td>
<td>7.23</td>
<td>835</td>
<td>653</td>
<td>21914</td>
<td>419</td>
<td>2724</td>
<td>0.0732</td>
</tr>
<tr>
<td>RADAU</td>
<td>10^{-7}</td>
<td>10^{-7}</td>
<td>10^{-9}</td>
<td>7.11</td>
<td>6.83</td>
<td>1775</td>
<td>1551</td>
<td>1782</td>
<td>1541</td>
<td>1775</td>
<td>0.0517</td>
</tr>
</tbody>
</table>


Figure II.14.3: Behavior of the solution over the integration interval.
Figure II.14.4: Work-precision diagram (scd versus CPU-time).
Figure II.14.5: Work-precision diagram (sod versus CPU-time).
Figure II.14.6: Work-precision diagram (mescd versus CPU-time).
Figure II.14.7: Work-precision diagram (mescd versus CPU-time).
15 Charge pump

15.1 General information
The problem is a stiff DAE of index 2, consisting of 3 differential and 6 algebraic equations. It has been contributed by Michael Günther, Georg Denk and Uwe Feldmann [GDF95].

The software part of the problem is in the file pump.f available at [MM08].

15.2 Mathematical description
The problem is of the form

\[ M \frac{dy}{dt} = f(t,y(t)), \quad y(0) = y_0, \quad y'(0) = y'_0, \]

with

\[ y \in \mathbb{R}^9, \quad 0 \leq t \leq 1.2 \cdot 10^{-6}. \]

The 9×9 matrix \( M \) is the zero matrix except for the minor \( M_{1,3,1,5} \), that is given by

\[
M_{1,3,1,5} = \begin{pmatrix}
1 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 1
\end{pmatrix}.
\]

The function \( f \) is defined by

\[
f(t,y) = \begin{pmatrix}
-y_9 \\
0 \\
-y_6 + V_m(t) \\
y_1 - Q_G(v) \\
y_2 - C_S \cdot y_7 \\
y_3 - Q_S(v) \\
y_4 - C_D \cdot y_8 \\
y_5 - Q_D(v)
\end{pmatrix},
\]

with \( v := (v_1, v_2, v_3) = (y_6, y_6 - y_7, y_6 - y_8) \), \( C_D = 0.4 \cdot 10^{-12} \) and \( C_S = 1.6 \cdot 10^{-12} \). The functions \( Q_G, Q_S \) and \( Q_D \) are given by:

1. If \( v_1 \leq V_{FB} = U_{T0} - \gamma \sqrt{\Phi} - \Phi \), then

\[
Q_G(v) = C_\alpha (v_1 - V_{FB}),
\]

\[
Q_S(v) = Q_D(v) = 0,
\]

with \( C_\alpha = 4 \cdot 10^{-12} \), \( U_{T0} = 0.2 \), \( \gamma = 0.035 \) and \( \Phi = 1.01 \).

2. If \( v_1 > V_{FB} \) and \( v_2 \leq U_{TE} = U_{T0} + \gamma (\sqrt{\Phi} - U_{BS} - \sqrt{\Phi}) \), then

\[
Q_G(v) = C_\alpha \gamma \left( \sqrt{\gamma/2}^2 + v_1 - V_{FB} - \gamma/2 \right),
\]

\[
Q_S(v) = Q_D(v) = 0.
\]

3. If \( v_1 > V_{FB} \) and \( v_2 > U_{TE} \), then

\[
Q_G(v) = C_\alpha \left( \frac{2}{3} (U_{GDT} + U_{GST} - \frac{U_{GDT}U_{GST}}{U_{GDT} + U_{GST}}) + \gamma \sqrt{\Phi - U_{BS}} \right),
\]

\[
Q_S(v) = Q_D(v) = -\frac{1}{2} \left( Q_G - C_\alpha \gamma \sqrt{\Phi - U_{BS}} \right).
\]
Here, $U_{BS}$, $U_{GST}$ and $U_{GDT}$ are given by

$$
U_{BS} = v_2 - v_1,
U_{GST} = v_2 - U_{TE},
U_{GDT} = \begin{cases} v_3 - U_{TE} & \text{for } v_3 > U_{TE}, \\ 0 & \text{for } v_3 \leq U_{TE}. \end{cases}
$$

The function $V_{in}(t)$ is defined using $	au = (10^9 \cdot t) \mod 120$ by

$$
V_{in}(t) = \begin{cases} 0 & \text{if } \tau < 50, \\ 20(\tau - 50) & \text{if } 50 \leq \tau < 60, \\ 20 & \text{if } 60 \leq \tau < 110, \\ 20(120 - \tau) & \text{if } \tau \geq 110. \end{cases}
$$

This means that the function $f$ has discontinuities in its derivative at $\tau = 50, 60, 90, 110, 120$.

Consistent initial values are

$$
y_0 = (Q_G(0, 0, 0), 0, Q_S(0, 0, 0), 0, Q_D(0, 0, 0), 0, 0, 0, 0)^T \quad \text{and} \quad y'_0 = (0, 0, 0, 0, 0, 0, 0, 0, 0)^T.
$$

The index of the first eight variables is 1, whereas the index of $y_9$ is 2.

### 15.3 Origin of the problem

The Charge-pump circuit shown in Figure II.15.1 consists of two capacitors and an $n$-channel MOS-transistor. The nodes gate, source, gate, and drain of the MOS-transistor are connected with the nodes 1, 2, 3, and Ground, respectively. In formulating the circuit equations, the transistor is replaced by four non-linear current sources in each of the connecting branches. They model the transistor.

![Circuit diagram of Charge-pump circuit](taken from [GDF95])

After inserting the transistor model in the circuit, we get the final circuit, which can be obtained from the circuit in Figure II.15.1 by applying the following changes:

- Remove the transistor and replace it by a solid line between the nodes 2 and 3. The point where the lines 2–3 and 1–Ground cross each other becomes a node, which will be denoted by $T$.

- Add current sources between nodes 1 and $T$, between 2 and $T$ and between 3 and $T$. There should also be a current source between the ground and node $T$, but as the node Ground does not enter the circuit equations, it will not be discussed. The currents produced by these sources are written as the derivatives of charges: current from 1 to $T$: $Q'_G$, from $T$ to 2: $Q'_S$ and from $T$ to 3: $Q'_D$. Here, the functions $Q_G, Q_S$ and $Q_D$ depend on the voltage drops $U_1, U_1 - U_2$ and $U_1 - U_3$, where $U_i$ denotes the potential in node $i$. 


The unknowns in the circuit are given by:

- The charges produced by the current sources: \( Y_{T1}, Y_{T2}, Y_{T3} \). They are aliases for respectively \( Q_G, Q_S \) and \( Q_D \). Consequently, \( Y_{T1} \) is the current between node \( T \) and node \( i \).
- The charges \( Y_S \) and \( Y_D \) in the capacitors \( C_S \) and \( C_D \).
- Potentials in nodes 1 to 3: \( U_1, U_2, U_3 \).
- The current through the voltage source \( V_{in}(t) \): \( I \).

In terms of these physical variables, the vector \( y \) introduced earlier reads

\[
y = (Y_{T1}, Y_S, Y_{T2}, Y_D, Y_{T3}, U_1, U_2, U_3, I)^T.
\]

Now, the following equations hold:

\[
Y'_{T1} = -I, \\
Y'_S + Y'_{T2} = 0, \\
Y'_D + Y'_{T3} = 0, \\
U_1 = V_{in}(t).
\]

The charges depend on the potentials and are given by

\[
Y_{T1} = Q_G(U_1, U_1 - U_2, U_1 - U_3), \\
Y_S = C_S \cdot U_2, \\
Y_{T2} = Q_S(U_1, U_1 - U_2, U_1 - U_3), \\
Y_D = C_D \cdot U_3, \\
Y_{T3} = Q_D(U_1, U_1 - U_2, U_1 - U_3).
\]

The functions \( Q_G, Q_S \) and \( Q_D \) are given in the previous section.

**Remark**: the potential \( U_1 \) is known. Here, it is treated as an unknown in order to keep the formulation general and leaving open the possibility to extend the circuit. In addition, removing \( U_1 \) by hand contradicts a Computer Aided Design (CAD) approach in circuit simulation.

### 15.4 Numerical solution of the problem

The various components differ enormously in magnitude. Therefore, the absolute and relative input tolerances \( \text{atol} \) and \( \text{rtol} \) were chosen to be component-dependent. Furthermore, we neglect the index 2 variable \( y_9 \) in the error control of DASSL. This leads to the following input tolerances:

\[
\text{atol}(i) = \text{Tol} \cdot 10^{-6} \quad \text{for} \quad i = 1, \ldots, 5, \\
\text{atol}(i) = \text{Tol} \quad \text{for} \quad i = 6, \ldots, 8, \\
\text{rtol}(i) = \text{Tol} \quad \text{for} \quad i = 1, \ldots, 8, \\
\text{atol}(9) = \text{rtol}(9) = 1000 \quad \text{for} \quad \text{DASSL}, \\
\text{atol}(9) = \text{rtol}(9) = \text{Tol} \quad \text{for} \quad \text{other solvers}.
\]

The reference solution was computed using quadruple precision GAMD on an Alphaserver DS20E, with a 667 MHz EV67 processor, \( \text{atol} = 10^{-18}, \text{rtol} = 10^{-37} \).

Table II.15.1 and Figures II.15.3-II.15.4 present the run characteristics and the work-precision diagram, respectively. For the computation of the number of significant correct digits (scd), only the first component is taken into account. The second up to eighth component are ignored because these components are zero in the true solution; the ninth component is neglected because it was excluded.
Table II.15.1: Run characteristics.

<table>
<thead>
<tr>
<th>solver</th>
<th>Tol</th>
<th>mescd</th>
<th>scd</th>
<th>steps</th>
<th>accept</th>
<th>#f</th>
<th>#Jac</th>
<th>#LU</th>
<th>CPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>BIMD</td>
<td>10^{-5}</td>
<td>7.34</td>
<td>16.00</td>
<td>711</td>
<td>454</td>
<td>8827</td>
<td>454</td>
<td>711</td>
<td>0.0478</td>
</tr>
<tr>
<td></td>
<td>10^{-7}</td>
<td>8.65</td>
<td>16.00</td>
<td>1125</td>
<td>688</td>
<td>15367</td>
<td>688</td>
<td>1125</td>
<td>0.0820</td>
</tr>
<tr>
<td>DDASSL</td>
<td>10^{-1}</td>
<td>0.93</td>
<td>0.14</td>
<td>447</td>
<td>438</td>
<td>604</td>
<td>369</td>
<td></td>
<td>0.0088</td>
</tr>
<tr>
<td></td>
<td>10^{-3}</td>
<td>5.42</td>
<td>16.00</td>
<td>983</td>
<td>833</td>
<td>1659</td>
<td>853</td>
<td></td>
<td>0.0215</td>
</tr>
<tr>
<td></td>
<td>10^{-5}</td>
<td>6.71</td>
<td>3.43</td>
<td>1737</td>
<td>1487</td>
<td>2903</td>
<td>1309</td>
<td></td>
<td>0.0361</td>
</tr>
<tr>
<td></td>
<td>10^{-7}</td>
<td>6.09</td>
<td>3.32</td>
<td>3059</td>
<td>2587</td>
<td>4945</td>
<td>2058</td>
<td></td>
<td>0.0595</td>
</tr>
<tr>
<td>GAMD</td>
<td>10^{-1}</td>
<td>2.11</td>
<td>1.51</td>
<td>320</td>
<td>200</td>
<td>3735</td>
<td>200</td>
<td>320</td>
<td>0.0166</td>
</tr>
<tr>
<td></td>
<td>10^{-3}</td>
<td>2.85</td>
<td>2.69</td>
<td>350</td>
<td>220</td>
<td>4786</td>
<td>220</td>
<td>350</td>
<td>0.0205</td>
</tr>
<tr>
<td></td>
<td>10^{-5}</td>
<td>4.78</td>
<td>5.12</td>
<td>620</td>
<td>370</td>
<td>14800</td>
<td>320</td>
<td>570</td>
<td>0.0547</td>
</tr>
<tr>
<td></td>
<td>10^{-7}</td>
<td>4.94</td>
<td>4.75</td>
<td>870</td>
<td>510</td>
<td>22340</td>
<td>410</td>
<td>770</td>
<td>0.0791</td>
</tr>
<tr>
<td>PSIDE-1</td>
<td>10^{-1}</td>
<td>1.17</td>
<td>0.37</td>
<td>938</td>
<td>839</td>
<td>9843</td>
<td>140</td>
<td>3752</td>
<td>0.0742</td>
</tr>
<tr>
<td></td>
<td>10^{-5}</td>
<td>2.64</td>
<td>4.47</td>
<td>1366</td>
<td>1068</td>
<td>13424</td>
<td>160</td>
<td>5424</td>
<td>0.1005</td>
</tr>
<tr>
<td></td>
<td>10^{-7}</td>
<td>9.05</td>
<td>16.00</td>
<td>2425</td>
<td>1555</td>
<td>24331</td>
<td>300</td>
<td>9616</td>
<td>0.1835</td>
</tr>
</tbody>
</table>

from DASSL's error control. For the mescd we consider all the components. The first component of the reference solution equals \(0.1262800429876759 \times 10^{-12}\) at the end of the integration interval. We remark that the magnitude of this component is at most \(10^{-10}\). For the work-precision diagram, we used: \(\text{Tol} = 10^{-1+(m/2)}, \ m = 0, 1, \ldots, 14; \ h0 = 10^{-6}\). \text{Tol} for BIMD, GAMD, MEBDFDAE, MEBDFI, RADAU and RADAU5. From Table II.15.1 and Figure II.15.3 we see that the numerical solution computed by DASSL results for some rather large values of \text{Tol} in a \text{scd} value of 15.4, which equals the accuracy of the reference solution.

Figure II.15.2 shows the behavior of the solution over the integration interval. Only the last four components have been plotted, since they are the physically important quantities. The other five components refer to charge flows inside the transistor, which are quantities the user is not interested in. These components have a similar behavior as the components 6, 7 and 8, but their magnitude is at most \(10^{-10}\).

The failed runs are in Table II.15.2; listed are the name of the solver that failed, for which values of \(m\) this happened, and the reason for failing.

References


Table II.15.2: Failed runs.

<table>
<thead>
<tr>
<th>solver</th>
<th>$m$</th>
<th>reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>BIMD</td>
<td>0</td>
<td>floating invalid</td>
</tr>
<tr>
<td>BIMD</td>
<td>4</td>
<td>too many consecutive Newton failures</td>
</tr>
<tr>
<td>BIMD</td>
<td>3, 5, 7</td>
<td>floating divide by zero</td>
</tr>
<tr>
<td>DASSL</td>
<td>2</td>
<td>error test failed repeatedly</td>
</tr>
<tr>
<td>DASSL</td>
<td>4, 7</td>
<td>floating overflow</td>
</tr>
<tr>
<td>DASSL</td>
<td>14</td>
<td>corrector failed to converge repeatedly</td>
</tr>
<tr>
<td>MEBDFDAE</td>
<td>0, 1, ..., 14</td>
<td>steps size too small</td>
</tr>
<tr>
<td>MEBDFI</td>
<td>0, 1, ..., 10</td>
<td>floating invalid</td>
</tr>
<tr>
<td>MEBDFI</td>
<td>11, 12, 13, 14</td>
<td>steps size too small</td>
</tr>
<tr>
<td>PSIDE-1</td>
<td>4, 13, 14</td>
<td>steps size too small</td>
</tr>
<tr>
<td>RADAU</td>
<td>0, 1, ..., 14</td>
<td>steps size too small</td>
</tr>
<tr>
<td>RADAU95</td>
<td>0, 1, ..., 10</td>
<td>floating invalid</td>
</tr>
<tr>
<td>RADAU95</td>
<td>11, ..., 14</td>
<td>steps size too small</td>
</tr>
</tbody>
</table>
Figure II.15.2: Behavior of the solution over the integration interval.
Figure II.15.3: Work-precision diagram (sec versus CPU-time).
Figure II.15.4: Work-precision diagram (mescd versus CPU-time).
16 Two bit adding unit

16.1 General Information

The problem is a stiff DAE of index 1, consisting of 175 differential equations and 175 algebraic equations. It has been contributed by M. Günther [Gün95, Gün98].

The software part of the problem is in the file tba.f available at [MM08].

16.2 Mathematical description of the problem

The problem is of the form

\[ \frac{dy}{dt} = f(t, x), \]
\[ 0 = y - g(x), \]  

where

\[ y, x \in \mathbb{R}^{175}, \quad f : \mathbb{R}^{351} \rightarrow \mathbb{R}^{350}, \quad g : \mathbb{R}^{350} \rightarrow \mathbb{R}^{350}, \quad 0 \leq t \leq 320, \quad y(0) = y_0, \quad x(0) = x_0. \]

Since the functions \( f(t, x) \) and \( g(x) \) and the (consistent) initial values \( y_0 \) and \( x_0 \) are too voluminous to be printed here, we refer to the subroutines \texttt{feval} and \texttt{init} for their definitions. The function \( f \) has discontinuities in its derivative at \( t = 0, 5, 10, \ldots, 320 \). The index of the components of \( x \) and \( y \) equals 1.

The function \( f \) contains several square roots. It is clear that the function can not be evaluated if one of the arguments of one of these square roots becomes negative. To prevent this situation, we set \( \texttt{IERR} = -1 \) in the Fortran subroutine that defines \( f \) if this happens. See page IV-ix of the description of the software part of the test set for more details on \( \texttt{IERR} \).

16.3 Origin of the problem

The two bit adding unit computes the sum of two base-2 numbers (each two digits long) and a carry bit. These numbers are fed into the circuit in the form of input signals. As a result the circuit gives their sum coded as three output signals.

The two bit adding unit circuit is a digital circuit. These circuits are used to compute boolean expressions. This is accomplished by associating voltages with boolean variables. By convention the boolean is true if the voltage exceeds 2V, and false if it is lower than 0.8V. In between the boolean is undefined. Using CMOS technique, however, sharper bounds are possible for the representation of booleans.

Digital circuits that compute elementary logical operations are called gates. An example of a gate is the NAND gate of test problem 9. This circuit is used to compute the logical expression \( \neg(V_1 \land V_2) \), where \( V_1 \) and \( V_2 \) are the booleans that are fed into the circuit as input signals.

The two bit adding unit is depicted in Figure II.16.1. In this figure the symbols ‘&’, ‘\( \geq 1 \)’ and a little white circle respectively stand for the AND, OR and NOT gate. A number of input signals and output signals enter and leave the circuit. Each signal is described by a time-dependent voltage and the boolean it represents. For these two quantities we shall use one symbol: the symbol of this boolean variable. Which one of the two quantities is meant by the symbol, is always clear from the context. With this convention, the input signals are referred to by the boolean variable they represent.

The circuit is designed to perform the addition

\[ A_1 A_0 + B_1 B_0 + C_{in} = C S_1 S_0. \]
The input signals representing the two numbers and the carry bit $C_{in}$ are fed into the circuit at the nodes indicated by $A0$, $A1$, $B0$, $B1$ and $Cin$. Here, a bar denotes the logical inversion. The output signals are delivered by the nodes indicated by $S0$, $S1$ and $C$.

In Figure II.16.1, a number of boxes are drawn using dashed lines. Each of them represents one
Table II.16.1: Characteristics of the gates that occur in the two bit adding unit.

<table>
<thead>
<tr>
<th>Name</th>
<th>logical expression</th>
<th># nodes</th>
<th># times</th>
</tr>
</thead>
<tbody>
<tr>
<td>NOR</td>
<td>((V_1 \lor V_2))</td>
<td>3 \cdot 4 + 1 = 13</td>
<td>3</td>
</tr>
<tr>
<td>NAND</td>
<td>((V_1 \land V_2))</td>
<td>3 \cdot 4 + 2 = 14</td>
<td>1</td>
</tr>
<tr>
<td>ANDOI</td>
<td>((V_1 \lor (V_2 \land V_3)))</td>
<td>4 \cdot 4 + 2 = 18</td>
<td>5</td>
</tr>
<tr>
<td>ORANI</td>
<td>((V_1 \land (V_2 \lor V_3)))</td>
<td>4 \cdot 4 + 2 = 18</td>
<td>1</td>
</tr>
</tbody>
</table>

of the following gates: the NOR (first box to the left in the top-row), the ORANI gate (the box besides \(S_1\)), the NAND (the box besides the ORANI gate) and the ANDOI (the box at the bottom). The circuit diagram of the NAND-gate is given in test problem 9. For the circuit diagrams of the NOR, ANDOI and ORANI gate see Figures II.16.2, II.16.3 and II.16.4. What logical expressions they compute, is listed in Table II.16.1. The fourth column in this table lists the number of times the gate occurs in the big circuit. The third column tabulates the number of nodes in the gate. These nodes consist of two types. The first type of nodes consists of the internal nodes of the transistors due to the MOS transistor model of Shichmann and Hodges [SH68]. Each transistor has four internal nodes that are also the links between transistor and the rest of the circuit. The second type of nodes comprises the usual nodes that are used to link circuit components together. These nodes are indicated by a number placed inside a square. To prevent any misunderstanding, we remark that the big dots in Figures II.16.2-II.16.4 do not represent nodes.

The connection of a gate with the rest of the circuit consists of the input nodes and the output node of the gate. The input signals enter the gate at the nodes with symbol \(V_1\), \(V_2\) and \(V_3\). The output signal leaves the gate from one of the numbered nodes. To ensure stability of the circuit, such an output node is always connected to a capacitance (we refer to the Fortran driver: CLOAD denoting the value of a load capacitance for the logical gates, and COUT for the output nodes \(S_0, S_1\) and \(\bar{C}\)). Finally, three enhancement transistors are coupled with the ANDOI gate at the bottom for a correct treatment of \(C_{in}\). This yields 12 internal nodes and two additional nodes, because the three transistors are coupled in series. Counting all nodes we have \(3 \cdot 13 + 1 \cdot 14 + 5 \cdot 18 + 1 \cdot 18 + 14 = 175\) nodes.

Applying Kirchhoff’s law to all nodes yields a system of 175 equations. This system is an integral form DAE of the special form

\[
A \cdot q(V) = f(t, V).
\]

The function \(q\) is a generally nonlinear function of node potentials \(V\), which describes the charges stored in all charge storing elements [GDF96]. Assembling the charge flow at each node by an incidence matrix \(A\), the dynamic part \(A \cdot q(V)\) equals the contribution of static currents denoted by \(f(t, V)\). If all load capacitances at the output nodes are nonzero, then the integral form DAE has differential index 0. If only one of the load capacitances equals zero, the generalized capacitance matrix \(A \cdot \partial q(V)/\partial V\) is singular, yielding a system of differential index 1. This shows the regularization effects by applying additional capacitances. Here, we use CLOAD=0 and COUT=2.0.

To make this problem suitable for the solvers used in this test set, the variable \(Q = A \cdot q(V)\) of assembled charges is introduced leading to

\[
\dot{Q} = f(t, V),
\]

\[
0 = Q - Aq(V).
\]

This transformation of the integral form DAE into a linearly implicit system raises the differential index by one. However, in the case of singular load capacitances, no higher index effects are detected in the sense of an appropriate perturbation index [Gün98].

Some of the 175 variables have a special meaning. These are the voltage variables of the nodes...
that deliver the output signals. The output signals $S_0$, $S_1$ and $C$ are given by the variables $x_{49}$, $x_{130}$ and $x_{148}$, respectively. Only these variables are of interest to the engineer.

In the next section we shall see the two bit adder in operation. Every 10 units of time the addition

$$A_1 A_0 + B_1 B_0 + C_{in} = C S_1 S_0,$$

is carried out. The numbers that are added are represented by the input signals depicted in Figure II.16.5. The outcome of the addition is represented by output signals given in Figure II.16.6. Often the output signals need time to adjust to changes in the input signal. Therefore, only during certain periods the sum is correctly represented by the output signals. The two bit adding unit has been designed in such a way that after each 10 units of time the output signal represents the sum correctly.

To see the two bit adding unit performing an addition let us see what happens at $t = 200$. Then the input signals read:

$$\overline{A_0} = 0, \quad \overline{A_1} = 1, \quad \overline{B_0} = 0, \quad \overline{B_1} = 0, \quad C_{in} = 1,$$

and the output signals are

$$S_0 = 1, \quad S_1 = 0, \quad \overline{C} = 0.$$

Recall, that a bar denotes the logical inverse. Clearly, the addition $01 + 11 + 1 = 101$ has been carried out.

### 16.4 Numerical solution of the problem

M. Günther provided the source code that defines the problem.

Table II.22.2 lists the voltages of the output signals in the reference solution. For the complete reference solution at $t = 320$ we refer to subroutine SOLUT. Since these components refer to the output signals $S_0$, $S_1$ and $\overline{C}$, they are the physically relevant quantities.

<table>
<thead>
<tr>
<th>$x_{49}$</th>
<th>$x_{130}$</th>
<th>$x_{148}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.2040419147264534</td>
<td>0.4997238455712048 · 10</td>
<td>0.2038985905095614</td>
</tr>
</tbody>
</table>

Table II.16.2: Value at the end of the integration interval of the components of the reference solution that correspond to the output signals.

Table II.16.4 and Figures II.16.6–II.16.10 present the run characteristics, the behavior of the output signals over the integration interval and the work-precision diagram, respectively. In computing the sec values, only $x_{49}$, $x_{130}$ and $x_{148}$ were considered, since they refer to the physically important quantities.

The reference solution was computed using RADAU5 without restarts in the discontinuities in time of the derivative of the problem defining function $f$, with $\text{rtol} = \text{atol} = 10^{-5}$ and $h_0 = 4 \cdot 10^{-5}$.

For the work-precision diagram, we used: $\text{rtol} = 10^{-(2+m/8)}$, $m = 0, 1, \ldots, 32$; $\text{atol} = \text{rtol}; h_0 = 10 \cdot \text{rtol}$ for BIMD, GAMD, MEBDFDAE, MEBDFI, RADAU and RADAU5. The failed runs are in Table II.16.3; listed are the name of the solver that failed, for which values of $m$ this happened, and the reason for failing.

**Remark**

M. Günther also wrote a special purpose solver called CHORAL, which stands for CHarge-ORiented ALgorithm [Gün95, Gün98] for integrating equations of the form

$$\frac{dy}{dt} = f(t, x),$$

$$0 = y - q(x).$$
Table II.16.3: Failed runs.

<table>
<thead>
<tr>
<th>solver</th>
<th>m</th>
<th>reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>BIMD</td>
<td>27, 32</td>
<td>more than nmax steps are needed</td>
</tr>
<tr>
<td>DASSL</td>
<td>30, 31</td>
<td>corrector failed to converge repeatedly</td>
</tr>
<tr>
<td>GAMD</td>
<td>25, 29</td>
<td>steps size too small</td>
</tr>
<tr>
<td>MEBDFDAE</td>
<td>0, 1</td>
<td>steps size too small</td>
</tr>
<tr>
<td>MEBDFDAE</td>
<td>2, 18</td>
<td>illegal function call</td>
</tr>
<tr>
<td>PSIDE-1</td>
<td>0, 24</td>
<td>steps size too small</td>
</tr>
<tr>
<td>RADAU</td>
<td>0, 1, 17</td>
<td>solver cannot handle IERR=-1.</td>
</tr>
<tr>
<td>RADAU5</td>
<td>0, 1, 17</td>
<td>solver cannot handle IERR=-1.</td>
</tr>
</tbody>
</table>

Most equations occurring in circuit analysis are of this form. In these equations the variables \( y \) and \( x \) represent respectively (assembled) charges and voltages. CHORAL is based on Rosenbrock-Wanner methods, while the special structure of the problem is exploited. The code eliminates the \( y \) variables, reducing the linear algebra work to solving systems of order 175 instead of 350. Correspondingly, a step size prediction and error control based directly on node potentials and currents is offered. For more information see

http://www.math.uni-wuppertal.de/~guenther.

References


Table II.16.4: Run characteristics.

<table>
<thead>
<tr>
<th>solver</th>
<th>rtol</th>
<th>atol</th>
<th>h0</th>
<th>mescd</th>
<th>scd</th>
<th>steps</th>
<th>accept</th>
<th>#f</th>
<th>#Jac</th>
<th>#LU</th>
<th>CPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>BIMD</td>
<td>10^{-2}</td>
<td>10^{-2}</td>
<td>10^{-1}</td>
<td>2.11</td>
<td>3.23</td>
<td>836</td>
<td>679</td>
<td>12440</td>
<td>679</td>
<td>836</td>
<td>29.4664</td>
</tr>
<tr>
<td></td>
<td>10^{-4}</td>
<td>10^{-4}</td>
<td>10^{-3}</td>
<td>4.44</td>
<td>6.58</td>
<td>1688</td>
<td>1621</td>
<td>24239</td>
<td>1621</td>
<td>1688</td>
<td>62.4786</td>
</tr>
<tr>
<td>DDASSL</td>
<td>10^{-2}</td>
<td>10^{-2}</td>
<td>10^{-2}</td>
<td>1.55</td>
<td>2.40</td>
<td>1892</td>
<td>1779</td>
<td>3674</td>
<td>786</td>
<td>21.3276</td>
<td></td>
</tr>
<tr>
<td></td>
<td>10^{-4}</td>
<td>10^{-4}</td>
<td></td>
<td>3.60</td>
<td>4.54</td>
<td>6036</td>
<td>5736</td>
<td>9380</td>
<td>866</td>
<td>27.7379</td>
<td></td>
</tr>
<tr>
<td>GAMD</td>
<td>10^{-2}</td>
<td>10^{-2}</td>
<td>10^{-1}</td>
<td>2.72</td>
<td>4.66</td>
<td>735</td>
<td>597</td>
<td>20213</td>
<td>597</td>
<td>735</td>
<td>28.1996</td>
</tr>
<tr>
<td></td>
<td>10^{-4}</td>
<td>10^{-4}</td>
<td>10^{-3}</td>
<td>2.68</td>
<td>3.32</td>
<td>1332</td>
<td>1225</td>
<td>43250</td>
<td>1234</td>
<td>1332</td>
<td>61.4255</td>
</tr>
<tr>
<td>MEBDFI</td>
<td>10^{-2}</td>
<td>10^{-2}</td>
<td>10^{-1}</td>
<td>1.96</td>
<td>3.14</td>
<td>2065</td>
<td>1818</td>
<td>194700</td>
<td>533</td>
<td>533</td>
<td>19.1911</td>
</tr>
<tr>
<td></td>
<td>10^{-4}</td>
<td>10^{-4}</td>
<td>10^{-3}</td>
<td>3.01</td>
<td>3.36</td>
<td>5269</td>
<td>4851</td>
<td>363601</td>
<td>982</td>
<td>982</td>
<td>38.9239</td>
</tr>
</tbody>
</table>

Figure II.16.2: Circuit diagram of the NOR gate (taken from [Gün95]).

Figure II.16.3: Circuit diagram of the ANDOI gate (taken from [Gün95]).
Figure II.16.4: Circuit diagram of the ORANI gate (taken from [Gün95]).
Figure II.16.5: The input signals $\overline{A}_0$, $\overline{A}_1$, $\overline{B}_0$, $\overline{B}_1$, and $C$. 
Figure II.16.6: Behavior of the output signals $S_0$, $S_1$, and $\overline{C}$ over the integration interval.
Figure II.16.7: Work-precision diagram (sod versus CPU-time).
Figure II.16.8: Work-precision diagram (scd versus CPU-time).
Figure II.16.9: Work-precision diagram (mescd versus CPU-time).
Figure II.16.10: Work-precision diagram (meskel versus CPU-time).
17 The car axis problem

17.1 General information

The problem is a stiff DAE of index 3, consisting of 8 differential and 2 algebraic equations. It has been taken from [Sch94]. Since not all initial conditions were given, we have chosen a consistent set of initial conditions. The software part of the problem is in the file caraxis.f available at [MM08].

17.2 Mathematical description of the problem

The problem is of the form

\[ p' = q, \quad (I.17.1) \]
\[ Kq = f(t, p, \lambda), \quad p, q \in \mathbb{R}^4, \quad \lambda \in \mathbb{R}^2, \quad 0 \leq t \leq 3, \quad (I.17.2) \]
\[ 0 = \phi(t, p), \quad (I.17.3) \]

with initial conditions

\[ p(0) = p_0, \quad q(0) = q_0, \quad p'(0) = q'_0, \quad \lambda(0) = \lambda_0 \text{ and } \lambda'(0) = \lambda'_0. \]

The matrix \( K \) reads \( \varepsilon^2 \frac{M}{2} I_4 \), where \( I_4 \) is the 4 \( \times \) 4 identity matrix. The function \( f : \mathbb{R}^7 \to \mathbb{R}^4 \) is given by

\[
\begin{pmatrix}
(L_0 - L_t) \frac{x_l}{L_t} + \lambda_1 x_b + 2\lambda_2 (x_l - x_r) \\
(L_0 - L_t) \frac{y_l}{L_t} + \lambda_1 y_b + 2\lambda_2 (y_l - y_r) - \frac{\varepsilon^2 M}{2} \\
(L_0 - L_r) \frac{x_r}{L_r} - x_b \\
(L_0 - L_r) \frac{y_r}{L_r} - y_b - 2\lambda_2 (x_l - x_r)
\end{pmatrix}
\]

Here, \((x_l, y_l, x_r, y_r)^T := p, \) and \( L_t \) and \( L_r \) are given by

\[ \sqrt{x_l^2 + y_l^2} \text{ and } \sqrt{(x_r - x_b)^2 + (y_r - y_b)^2}. \]

Furthermore, the functions \( x_b(t) \) and \( y_b(t) \) are defined by

\[ x_b(t) = \sqrt{L^2 - y_b^2(t)}, \quad (I.17.4) \]
\[ y_b(t) = h \sin(\omega t). \quad (I.17.5) \]

The function \( \phi : \mathbb{R}^5 \to \mathbb{R}^2 \) reads

\[
\phi(t, p) = \begin{pmatrix}
x_l x_b + y_l y_b \\
(x_l - x_r)^2 + (y_l - y_r)^2 - L^2
\end{pmatrix}
\]

The constants are listed below.

<table>
<thead>
<tr>
<th>( \frac{L}{L_0} )</th>
<th>( \varepsilon )</th>
<th>( M )</th>
<th>( h )</th>
<th>( \tau )</th>
<th>( \omega )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( = 1 )</td>
<td>( = 10^{-2} )</td>
<td>( = 10 )</td>
<td>( = 10^{-1} )</td>
<td>( = \pi/5 )</td>
<td>( = 10 )</td>
</tr>
</tbody>
</table>

Consistent initial values are

\[ p_0 = \begin{pmatrix}
0 \\
1/2 \\
1/2
\end{pmatrix}, \quad q_0 = \begin{pmatrix}
-1/2 \\
0 \\
-1/2
\end{pmatrix}, \quad q'_0 = -2 M^2 f(0, p_0, \lambda_0), \quad p'_0 = q_0, \quad \lambda_0 = \lambda'_0 = (0, 0)^T. \]

The index of the variables \( p, q \) and \( \lambda \) is 1, 2 and 3, respectively.
17.3 Origin of the problem

The car axis problem is an example of a rather simple multibody system, in which the behavior of a car axis on a bumpy road is modeled by a set of differential-algebraic equations.

A simplification of the car is depicted in Figure II.17.1. We model the situation that the left wheel at the origin $(0,0)$ rolls on a flat surface and the right wheel at coordinates $(x_b,y_b)$ rolls over a hill of height $h$ every $\tau$ seconds\(^1\). This means that $y_b$ varies over time according to (II.17.5). The length of the axis, denoted by $L$, remains constant over time, which means that $x_b$ has to fulfill (II.17.4). Two springs carry over the movement of the axis between the wheels to the chassis of the car, which is represented by the bar $(x_l,y_l)-(x_r,y_r)$ of mass $M$. The two springs are assumed to be massless and have Hooke’s constant $1/e^2$ and length $L_0$ at rest.

There are two position constraints. Firstly, the distance between $(x_l,y_l)$ and $(x_r,y_r)$ must remain constantly $L$ and secondly, for simplicity of the model, we assume that the left spring remains orthogonal to the axis. If we identify $p$ with the vector $(x_l,y_l,x_r,y_r)^T$, then we see that Equation (II.17.3) reflects these constraints.

Using Lagrangian mechanics, the equations of motions for the car axis are given by

$$\frac{M}{2} \frac{d^2 p}{dt^2} = F_H + G^T \lambda + F_g.$$  

Here, $G$ is the $2 \times 4$ Jacobian matrix of the function $\phi$ with respect to $p$ and $\lambda$ is the 2-dimensional vector containing the so-called Lagrange multipliers. The factor $M/2$ is explained by the fact that the mass $M$ is divided equally over $(x_l,y_l)$ and $(x_r,y_r)$. The force $F_H$ represents the spring forces:

$$F_H = -\left(\cos(\alpha_l)F_l, \sin(\alpha_l)F_l, \cos(\alpha_r)F_r, \sin(\alpha_r)F_r\right)^T,$$

\(^1\)In the source fortran file the variable $r$ stands for $h$.
where $F_l$ and $F_r$ are the forces induced by the left and right spring, respectively, according to Hooke’s law:

$$F_l = (L_l - L_0) / \varepsilon^2,$$
$$F_r = (L_r - L_0) / \varepsilon^2.$$  

Here, $L_l$ and $L_r$ are the actual lengths of the left and right spring, respectively:

$$L_l = \sqrt{x_l^2 + y_l^2},$$
$$L_r = \sqrt{(x_r - x_b)^2 + (y_r - y_b)^2}.$$  

Furthermore, $\alpha_l$ and $\alpha_r$ are the angles of the left and right spring with respect to the horizontal axis of the coordinate system:

$$\alpha_l = \arctan(y_l / x_l),$$
$$\alpha_r = \arctan((y_r - y_b) / (x_r - x_b)).$$  

Finally, $F_g$ represents the gravitational force

$$F_g = -(0, 1, 0, 1)^T \frac{M}{2} g.$$  

The original formulation [Sch94] sets $g = 1$. We rewrite (II.17.6) as a system of first order differential equations by introducing the velocity vector $q$, so that we obtain the first order differential equations (II.17.1) and

$$\frac{M}{2} \frac{dq}{dt} = F_H + G^T \lambda + F_g.$$  

Setting $f = F_H + G^T \lambda + F_g$, it is easily checked that multiplying (II.17.7) by $\varepsilon^2$ yields (II.17.2).

To arrive at a consistent set of initial values $p_0$, $q_0$ and $\lambda_0$, we have to solve the system of equations consisting of the constraint

$$\phi(t_0, p_0) = 0,$$  

and the $1$ up to $k - 1$ times differentiated constraint (II.17.8), where $k$ is the highest variable index. To facilitate notation, we introduce $\tilde{p} := (t, p)^T$ and its derivative $\tilde{q} := \frac{dp}{dt} = (1, q^T)^T$. The Jacobian of $\phi$ with respect to $\tilde{p}$ will be denoted by $G$. Here, $k = 3$, yielding the additional conditions

$$\tilde{G}(\tilde{p}_0)\tilde{q}_0 = 0$$  

and

$$\phi_{\tilde{p}\tilde{p}}(\tilde{p}_0)(\tilde{q}_0, \tilde{\lambda}_0) + \tilde{G}(\tilde{p}_0)\tilde{q}_0' = 0,$$  

where $\phi_{\tilde{p}\tilde{p}}$ denotes the second derivative of $\phi$ with respect to $\tilde{p}$. Using (II.17.6) and the fact that the first component of $\tilde{q}_0'$ vanishes, the latter condition equals

$$\phi_{\tilde{p}\tilde{p}}(\tilde{p}_0)(\tilde{q}_0, \tilde{\lambda}_0) + \frac{2}{M} G(p_0) (F_H(p_0) + G^T(p_0) \lambda_0 + F_g(p_0)) = 0.$$  

(II.17.10)
The equations (II.17.8)–(II.17.10) are solved for
\[ x_r = L, \]
\[ x_l = 0, \]
\[ y_r = y_l = L_0, \]
\[ x'_r = x'_l = -\frac{L_0}{L} \frac{2\pi}{\tau} h, \]
\[ y'_r = \frac{L^2 \tau}{2\pi h^2 M} (2\lambda_1 - \lambda_2), \]
\[ y'_l = \frac{L^2 \tau}{2\pi h^2 M} (2\lambda_2 - \lambda_1) \pm \frac{L}{\varepsilon} \sqrt{\frac{-8\lambda_2 + 2\lambda_1}{M}}. \]

Choosing \( \lambda_1 = \lambda_2 = 0 \), we arrive at the initial conditions listed in §17.2.

**Table II.17.1: Reference solution at the end of the integration interval.**

| \( y_1 \) | 0.49345578427502890422 \( \cdot 10^{-1} \) |
| \( y_2 \) | 0.496989460301711353861 |
| \( y_3 \) | 0.10417452488542151681 \( \cdot 10 \) |
| \( y_4 \) | 0.37391102779361256927 |
| \( y_5 \) | -0.77058368404972357970 \( \cdot 10^{-1} \) |
| \( y_6 \) | 0.744686658723778553466 \( \cdot 10^{-2} \) |
| \( y_7 \) | 0.17556815753723222276 \( \cdot 10^{-1} \) |
| \( y_8 \) | 0.770341043779251977643 |
| \( y_9 \) | -0.47368659084893324729 \( \cdot 10^{-2} \) |
| \( y_{10} \) | -0.11046803312573468808 \( \cdot 10^{-2} \) |

**17.4 Numerical solution of the problem**

Tables II.17.1–II.17.2 and Figures II.17.2–II.17.3 present the reference solution at the end of the integration interval, the run characteristics, the behavior of some solution components over the integration interval and the work-precision diagrams, respectively. The reference solution was computed on using quadruple precision GAMD on an Alphaserver DS20E, with a 667 MHz EV67 processor. atol = rtol = \( h_0 = 10^{-24} \). For the work-precision diagrams, we used: rtol = \( 10^{-4} \cdot (4 + m/4) \), \( m = 0, 1, \ldots, 24 \); atol = rtol; h0 = rtol for BIMD, GAMD, MEBDFDAE, MEBDFI, RADAU and RADAU5.

**References**


Table II.17.2: Run characteristics.

<table>
<thead>
<tr>
<th>solver</th>
<th>rtol</th>
<th>atol</th>
<th>h0</th>
<th>mescl</th>
<th>scd</th>
<th>steps</th>
<th>accept</th>
<th>#f</th>
<th>#Jac</th>
<th>#LU</th>
<th>CPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>BIMD</td>
<td>10(^{-4})</td>
<td>10(^{-4})</td>
<td>10(^{-4})</td>
<td>2.19</td>
<td>0.34</td>
<td>71</td>
<td>71</td>
<td>1693</td>
<td>71</td>
<td>71</td>
<td>0.0088</td>
</tr>
<tr>
<td></td>
<td>10(^{-7})</td>
<td>10(^{-7})</td>
<td>10(^{-7})</td>
<td>5.47</td>
<td>3.34</td>
<td>138</td>
<td>138</td>
<td>4511</td>
<td>138</td>
<td>138</td>
<td>0.0224</td>
</tr>
<tr>
<td></td>
<td>10(^{-10})</td>
<td>10(^{-10})</td>
<td>10(^{-10})</td>
<td>8.01</td>
<td>5.35</td>
<td>235</td>
<td>235</td>
<td>9669</td>
<td>235</td>
<td>235</td>
<td>0.0488</td>
</tr>
<tr>
<td>GAMD</td>
<td>10(^{-4})</td>
<td>10(^{-4})</td>
<td>10(^{-4})</td>
<td>1.98</td>
<td>0.39</td>
<td>39</td>
<td>39</td>
<td>2169</td>
<td>39</td>
<td>39</td>
<td>0.0088</td>
</tr>
<tr>
<td></td>
<td>10(^{-7})</td>
<td>10(^{-7})</td>
<td>10(^{-7})</td>
<td>4.82</td>
<td>2.64</td>
<td>98</td>
<td>98</td>
<td>7167</td>
<td>98</td>
<td>98</td>
<td>0.0293</td>
</tr>
<tr>
<td></td>
<td>10(^{-10})</td>
<td>10(^{-10})</td>
<td>10(^{-10})</td>
<td>6.50</td>
<td>3.84</td>
<td>179</td>
<td>179</td>
<td>18771</td>
<td>179</td>
<td>179</td>
<td>0.0742</td>
</tr>
<tr>
<td>MEBDFI</td>
<td>10(^{-4})</td>
<td>10(^{-4})</td>
<td>10(^{-4})</td>
<td>0.88</td>
<td>-0.23</td>
<td>280</td>
<td>278</td>
<td>1246</td>
<td>27</td>
<td>27</td>
<td>0.0059</td>
</tr>
<tr>
<td></td>
<td>10(^{-7})</td>
<td>10(^{-7})</td>
<td>10(^{-7})</td>
<td>4.65</td>
<td>3.34</td>
<td>650</td>
<td>648</td>
<td>2797</td>
<td>47</td>
<td>47</td>
<td>0.0137</td>
</tr>
<tr>
<td></td>
<td>10(^{-10})</td>
<td>10(^{-10})</td>
<td>10(^{-10})</td>
<td>4.21</td>
<td>2.08</td>
<td>1393</td>
<td>1391</td>
<td>5624</td>
<td>85</td>
<td>85</td>
<td>0.0264</td>
</tr>
<tr>
<td>PSIDE-1</td>
<td>10(^{-4})</td>
<td>10(^{-4})</td>
<td>10(^{-4})</td>
<td>0.83</td>
<td>-0.28</td>
<td>55</td>
<td>54</td>
<td>1403</td>
<td>42</td>
<td>42</td>
<td>0.0098</td>
</tr>
<tr>
<td></td>
<td>10(^{-7})</td>
<td>10(^{-7})</td>
<td>10(^{-7})</td>
<td>4.41</td>
<td>2.27</td>
<td>179</td>
<td>172</td>
<td>4103</td>
<td>83</td>
<td>83</td>
<td>0.0273</td>
</tr>
<tr>
<td></td>
<td>10(^{-10})</td>
<td>10(^{-10})</td>
<td>10(^{-10})</td>
<td>7.25</td>
<td>4.86</td>
<td>625</td>
<td>612</td>
<td>13751</td>
<td>115</td>
<td>115</td>
<td>0.0869</td>
</tr>
<tr>
<td>RADAU</td>
<td>10(^{-4})</td>
<td>10(^{-4})</td>
<td>10(^{-4})</td>
<td>1.34</td>
<td>0.19</td>
<td>98</td>
<td>97</td>
<td>850</td>
<td>95</td>
<td>95</td>
<td>0.0039</td>
</tr>
<tr>
<td></td>
<td>10(^{-7})</td>
<td>10(^{-7})</td>
<td>10(^{-7})</td>
<td>3.73</td>
<td>2.51</td>
<td>289</td>
<td>288</td>
<td>259</td>
<td>282</td>
<td>282</td>
<td>0.0127</td>
</tr>
<tr>
<td></td>
<td>10(^{-10})</td>
<td>10(^{-10})</td>
<td>10(^{-10})</td>
<td>5.99</td>
<td>4.22</td>
<td>179</td>
<td>178</td>
<td>4281</td>
<td>169</td>
<td>169</td>
<td>0.0166</td>
</tr>
</tbody>
</table>
Figure II.17.2: Behavior of \((x_l, y_l)\) and \((x_r, y_r)\) over the integration interval.
Figure II.17.3: Work-precision diagram (sod versus CPU-time).
Figure II.17.4: Work-precision diagram (scd versus CPU-time).
Figure II.17.5: Work-precision diagram (mescd versus CPU-time).
Figure II.17.6: Work-precision diagram (mescl versus CPU-time).
18 Fekete problem

18.1 General information

The problem is an index 2 DAE from mechanics. The dimension is $8N$, where $N$ is a user supplied integer. The numerical tests shown here correspond to $N = 20$. The problem is of interest for the computation of the elliptic Fekete points [Par95]. The parallel-IVP-algorithm group of CWI contributed this problem to the test set, in collaboration with W. J. H. Stortelder. The software part of the problem is in the file fekete.f available at [MM08].

18.2 Mathematical description of the problem

The problem is of the form

$$M \frac{dy}{dt} = f(y), \quad y(0) = y_0, \quad y'(0) = y'_0,$$

with

$$y, f \in \mathbb{R}^{8N}, \quad 0 \leq t \leq t_{end}.$$

Here, $t_{end} = 1000$, $N = 20$ and $M$ is the (constant) mass matrix given by

$$M = \begin{pmatrix} I_{6N} & 0 \\ 0 & 0 \end{pmatrix},$$

where $I_{6N}$ is the identity matrix of dimension $6N$. For the definition of the function $f$, we refer to §18.3.

The components $y_{0,i}$ of the initial vector $y_0$ are defined by

$$\begin{pmatrix} y_{0,3(j-1)+1} \\ y_{0,3(j-1)+2} \\ y_{0,3(j-1)+3} \end{pmatrix} = \begin{pmatrix} \cos(\omega_j) \cos(\beta_j) \\ \sin(\omega_j) \cos(\beta_j) \\ \sin(\beta_j) \end{pmatrix} \quad \text{for} \quad j = 1, \ldots, N,$$

where

$$\beta_j = \begin{cases} \frac{3}{8} \pi & \text{and} \quad \omega_j = \frac{2j}{9} \pi + \frac{1}{13} \pi & \text{for} \quad j = 1, \ldots, 3, \\ \frac{1}{5} \pi & \text{and} \quad \omega_j = \frac{2(j-3)}{9} \pi + \frac{1}{25} \pi & \text{for} \quad j = 4, \ldots, 10, \\ -\frac{2}{10} \pi & \text{and} \quad \omega_j = \frac{2(j-10)}{9} \pi + \frac{1}{7} \pi & \text{for} \quad j = 11, \ldots, 16, \\ -\frac{3}{10} \pi & \text{and} \quad \omega_j = \frac{2(j-17)}{9} \pi + \frac{1}{7} \pi & \text{for} \quad j = 17, \ldots, 20, \end{cases}$$

and

$$y_{0,i} = 0 \quad \text{for} \quad i = 3N + 1, \ldots, 6N,$$

$$y_{0,6N+j} = \frac{1}{2} (p_j(0), \hat{f}_j) \quad \text{for} \quad j = 1, \ldots, N,$$

$$y_{0,i} = 0 \quad \text{for} \quad i = 7N + 1, \ldots, 8N,$$

where

$$p_j = \begin{pmatrix} y_{3(j-1)+1} \\ y_{3(j-1)+2} \\ y_{3(j-1)+3} \end{pmatrix}, \quad \hat{f}_j = \begin{pmatrix} f_{3(j-1)+1}((p(0), 0, \ldots, 0)^T) \\ f_{3(j-1)+2}((p(0), 0, \ldots, 0)^T) \\ f_{3(j-1)+3}((p(0), 0, \ldots, 0)^T) \end{pmatrix},$$

and $p = (y_1, y_2, \ldots, y_{8N})^T$. The initial derivative vector reads $y'_0 = f(y_0)$. These definitions of $y_0$ and $y'_0$ yield consistent initial values. The first $6N$ components are of index 1, the last $2N$ of index 2.
18.3 Origin of the problem

This problem is of interest for the computation of the elliptic Fekete points. Let us define the unit sphere in \( \mathbb{R}^3 \) by \( S^2 \) and for any configuration \( x := (x_1, x_2, \ldots, x_N)^T \) of points \( x_i \in S^2 \), the function

\[
V(x) := \prod_{i<j} \|x_i - x_j\|_2. \tag{II.18.3}
\]

We denote the value of \( x \) for which \( V \) reaches its global maximum by \( \hat{x} = (\hat{x}_1, \ldots, \hat{x}_N) \). The points \( \hat{x}_1, \hat{x}_2, \ldots, \hat{x}_N \) are called the elliptic Fekete points of order \( N \). For example, for \( N = 4 \), the points of the optimal solution form a tetrahedron. But, in case of 8 points, intuition fails; the elliptic Fekete points do not form a cube in this case. A cube where, for example, the upper plane is rotated over \( 45^\circ \) with respect to the bottom plane, gives already a larger value of \( V \). It turns out (see e.g. [Par95]) that \( \hat{x} \) is difficult to compute as solution of a global optimization problem. For reasons that will become clear later, we differentiate \( \log(V) \) with respect to \( x_k \) and apply the method of Lagrange multipliers, to see that \( \hat{x} \) fulfills

\[
\nabla_k \log(V(x)) \big|_x = \hat{x} = \sum_{j \neq k} \frac{\hat{x}_k - \hat{x}_j}{\|\hat{x}_k - \hat{x}_j\|_2} = \zeta_k \hat{x}_k, \tag{II.18.4}
\]

where the \( \zeta_k \) are Lagrange multipliers.

We now discuss the Fekete points from another point of view. Consider on \( S^2 \) a number of \( N \) particles, on which two forces are invoked: a repulsive force, by which the particles will start to move away from each other, and an adhesion force, by which the particles will reach a stationary state after a certain period of time.
We denote the position in Cartesian coordinates of particle $i$ at time $t$ by $p_i(t)$ and the configuration of $N$ points at time $t$ by $p(t) = (p_1(t), \ldots, p_N(t))^T$. The stationary configuration is assumed to be obtained at $t = t_{\text{stat}}$ and will be denoted by $\hat{p} := (\hat{p}_1, \hat{p}_2, \ldots, \hat{p}_N)$, where $\hat{p}_i := p_i(t_{\text{stat}})$. The repulsive force on particle $i$ caused by particle $j$ is defined by

$$F_{ij} = \frac{p_i - p_j}{\|p_i - p_j\|_2^2}.$$ 

Note that the choice $\gamma = 3$ can be interpreted as an electrical force working on particles with unit charge. The adhesion force working on particle $i$ is denoted by $A_i$ and given by

$$A_i = -\alpha q_i.$$ 

Here, $q$ is the velocity vector and $\alpha$ is valued 0.5.

We can compute the configuration of the particles as function of time, given that the particles cannot leave the unit sphere, as solution of the DAE system

\begin{align*}
p' &= q, \quad \text{(II.18.5)} \\
q' &= g(p, q) + G^T(p)\lambda, \quad \text{(II.18.6)} \\
0 &= \phi(p), \quad \text{(II.18.7)}
\end{align*}

where $G = \partial \phi / \partial p$ and $\lambda \in \mathbb{R}^N$. The function $\phi : \mathbb{R}^{3N} \rightarrow \mathbb{R}^N$ represents the constraint, which states that the particles remain on the unit sphere:

$$\phi_i(p) = p_{i,1}^2 + p_{i,2}^2 + p_{i,3}^2 - 1.$$ 

The function $g : \mathbb{R}^{6N} \rightarrow \mathbb{R}^{3N}$ is given by $g = (g_i), i = 1, \ldots, N, \text{ where}$

$$g_i(p, q) = \sum_{j \neq i} F_{ij}(p) + A_i(q).$$

The term $G^T(p)\lambda$ in (II.18.6) represents the normal force which keeps the particle on $S^2$.

Since we know that the speed of the final configuration at $t = t_{\text{stat}}$ is 0, we can substitute $q = 0$ and $p = \hat{p}$ in formula (II.18.6), thus arriving at

$$0 = \sum_{j \neq i} F_{ij}(\hat{p}) + G^T(\hat{p})\lambda,$$

which is equal to

$$\sum_{i \neq j} \frac{\hat{p}_i - \hat{p}_j}{\|\hat{p}_i - \hat{p}_j\|_2^2} = -2\lambda_i \hat{p}_i. \quad \text{(II.18.8)}$$

Comparing (II.18.4) and (II.18.8) tells us that computing $\hat{p}$ for $\gamma = 2$ gives the local optima of the function $V$ in (II.18.3). In [PSS97], it is showed that computing $\hat{p}$ by solving the system (II.18.5)--(II.18.7) and then substituting $x = \hat{p}$ in (II.18.3), results in values of $V$ that are very competitive with those obtained by global optimization packages. For more details on elliptic Fekete points, we refer to [Par95] and [SS93].

The DAE system mentioned before is of index 3. To arrive at a more stable formulation of the problem, we stabilize the constraint (see [BCP98, p. 153]) by replacing (II.18.5) by

$$p' = q + G^T(p)\mu,$$

where $\mu \in \mathbb{R}^N$, and appending the differentiated constraint

$$0 = G(p)q. \quad \text{(II.18.10)}$$
The system (II.18.1), (II.18.2), (II.18.3), (II.18.4) is now of index 2; the variables \( p \) and \( q \) are of index 1, the variables \( \lambda \) and \( \mu \) of index 2. We cast the system in the form (II.18.1) by setting \( y = (p, q, \lambda, \mu)^T \) and \( f(y) = f(p, q, \lambda, \mu) = (q + G^T \mu, q + G^T \lambda, Gq)^T \), where \( p_i \) is in Cartesian coordinates.

The choice for the initial configuration as defined in §18.2 is a rough attempt to spread out the points over the sphere. To arrive at a consistent set of initial values we choose \( q(0) = 0 \), yielding \( \mu(0) = 0 \) and \( \phi_i''(0) = (2p_i(0), q_i(0)) = 0 \). Consequently,

\[
\phi_i''(0) = (2p_i(0), q_i(0))
\]

\[
= (2p_i(0), g_i(p(0), q(0)) + 2\lambda_i(0)p_i(0)).
\]

Requiring \( \phi_i''(0) = 0 \) gives

\[
\lambda_i(0) = -\frac{\langle p_i(0), g_i(p(0), q(0)) \rangle}{2\langle p_i(0), p_i(0) \rangle} = -\frac{1}{2} \langle p_i(0), g_i(p(0), q(0)) \rangle.
\]

The initial derivative vector \( y_0' \) can be chosen equal to \( f(y_0) \). For \( N \leq 20 \), \( t_{\text{stat}} \leq 1000 \), therefore we chose \( t_{\text{end}} = 1000 \).

In Figure II.18.1 the final configuration for 20 points is plotted.

### 18.4 Numerical solution of the problem

All the tests concern the case with \( N = 20 \). Tables II.18.1–II.18.2 and Figures II.18.1–II.18.6 present the reference solution at the end of the integration interval (first 12 components), the run characteristics, the behavior of the first 6 solution components over the interval [0, 20] and the work-precision diagrams, respectively. In computing the scd values, only the first sixty components were considered, since they refer to the position of the particles. The reference solution was computed using RADAU5, \( \text{rtol} = 10^{-12} \), \( \text{atol} = 10^{-12} \), and \( h0 = 10^{-12} \). For the work-precision diagrams, we used: \( \text{rtol} = 10^{-2+2m/10} \), \( m = 0, 1, \ldots, 64 \); \( \text{atol} = \text{rtol} \); \( h0 = \text{rtol} \) for BIMD, GAMD, MEBDFDAE, MEBDFI, RADAU and RADAU5.

### References


### Table II.18.2: Run characteristics.

<table>
<thead>
<tr>
<th>solver</th>
<th>rtol</th>
<th>atol</th>
<th>h₀</th>
<th>mescd</th>
<th>scd</th>
<th>steps</th>
<th>accept</th>
<th>#f</th>
<th>#Jac</th>
<th>#LU</th>
<th>CPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>BIMD</td>
<td>10^{-2}</td>
<td>10^{-2}</td>
<td>10^{-2}</td>
<td>4.12</td>
<td>2.63</td>
<td>30</td>
<td>29</td>
<td>415</td>
<td>29</td>
<td>30</td>
<td>0.2450</td>
</tr>
<tr>
<td></td>
<td>10^{-3}</td>
<td>10^{-3}</td>
<td>10^{-3}</td>
<td>5.36</td>
<td>4.19</td>
<td>43</td>
<td>43</td>
<td>668</td>
<td>43</td>
<td>43</td>
<td>0.3445</td>
</tr>
<tr>
<td></td>
<td>10^{-4}</td>
<td>10^{-4}</td>
<td>10^{-4}</td>
<td>6.69</td>
<td>5.33</td>
<td>65</td>
<td>65</td>
<td>1094</td>
<td>65</td>
<td>65</td>
<td>0.5124</td>
</tr>
<tr>
<td>GAMD</td>
<td>10^{-2}</td>
<td>10^{-2}</td>
<td>10^{-2}</td>
<td>4.16</td>
<td>2.99</td>
<td>26</td>
<td>24</td>
<td>526</td>
<td>24</td>
<td>26</td>
<td>0.2147</td>
</tr>
<tr>
<td></td>
<td>10^{-3}</td>
<td>10^{-3}</td>
<td>10^{-3}</td>
<td>4.79</td>
<td>3.78</td>
<td>26</td>
<td>26</td>
<td>947</td>
<td>26</td>
<td>26</td>
<td>0.3006</td>
</tr>
<tr>
<td></td>
<td>10^{-4}</td>
<td>10^{-4}</td>
<td>10^{-4}</td>
<td>5.76</td>
<td>4.45</td>
<td>38</td>
<td>38</td>
<td>1319</td>
<td>38</td>
<td>38</td>
<td>0.4119</td>
</tr>
<tr>
<td>MEBDFI</td>
<td>10^{-2}</td>
<td>10^{-2}</td>
<td>10^{-2}</td>
<td>3.56</td>
<td>2.10</td>
<td>60</td>
<td>57</td>
<td>192</td>
<td>15</td>
<td>15</td>
<td>0.1064</td>
</tr>
<tr>
<td></td>
<td>10^{-3}</td>
<td>10^{-3}</td>
<td>10^{-3}</td>
<td>4.58</td>
<td>3.23</td>
<td>129</td>
<td>128</td>
<td>428</td>
<td>18</td>
<td>18</td>
<td>0.1513</td>
</tr>
<tr>
<td></td>
<td>10^{-4}</td>
<td>10^{-4}</td>
<td>10^{-4}</td>
<td>5.81</td>
<td>4.81</td>
<td>218</td>
<td>216</td>
<td>707</td>
<td>23</td>
<td>23</td>
<td>0.2176</td>
</tr>
<tr>
<td>PSIDE-1</td>
<td>10^{-2}</td>
<td>10^{-2}</td>
<td></td>
<td>3.66</td>
<td>2.20</td>
<td>73</td>
<td>53</td>
<td>693</td>
<td>16</td>
<td>16</td>
<td>1.3137</td>
</tr>
<tr>
<td></td>
<td>10^{-3}</td>
<td>10^{-3}</td>
<td></td>
<td>4.40</td>
<td>3.19</td>
<td>88</td>
<td>59</td>
<td>779</td>
<td>11</td>
<td>11</td>
<td>1.4357</td>
</tr>
<tr>
<td></td>
<td>10^{-4}</td>
<td>10^{-4}</td>
<td></td>
<td>5.32</td>
<td>4.12</td>
<td>114</td>
<td>75</td>
<td>967</td>
<td>9</td>
<td>9</td>
<td>1.7363</td>
</tr>
<tr>
<td>RADAU</td>
<td>10^{-2}</td>
<td>10^{-2}</td>
<td>10^{-2}</td>
<td>4.43</td>
<td>1.97</td>
<td>33</td>
<td>30</td>
<td>274</td>
<td>27</td>
<td>27</td>
<td>0.5065</td>
</tr>
<tr>
<td></td>
<td>10^{-3}</td>
<td>10^{-3}</td>
<td>10^{-3}</td>
<td>4.11</td>
<td>2.65</td>
<td>43</td>
<td>41</td>
<td>315</td>
<td>38</td>
<td>38</td>
<td>0.5993</td>
</tr>
<tr>
<td></td>
<td>10^{-4}</td>
<td>10^{-4}</td>
<td>10^{-4}</td>
<td>5.36</td>
<td>4.29</td>
<td>61</td>
<td>58</td>
<td>442</td>
<td>54</td>
<td>54</td>
<td>0.7662</td>
</tr>
</tbody>
</table>


Figure II.18.2: Behavior of the solution over the integration interval.
Figure II.18.3: Work-precision diagram (sed versus CPU-time).
Figure II.18.4: Work-precision diagram (sod versus CPU-time).
Figure II.18.5: Work-precision diagram (mesd versus CPU-time).
Figure II.18.6: Work-precision diagram (mescd versus CPU-time).
19 Slider Crank

19.1 General Information

This problem was contributed by Bernd Simeon, March 1998. The slider crank shows some typical properties of simulation problems in flexible multibody systems, i.e., constrained mechanical systems which include both rigid and elastic bodies. It is also an example of a stiff mechanical system since it features large stiffness terms in the right hand side. Accordingly, there are some fast variables with high frequency oscillations.

This problem is originally described by a second order system of differential-algebraic equations (DAEs), but transformed to first order and semi-explicit system of dimension 24. The index of the problem is originally 3, but an index 1 and index 2 formulation are supplied as well. By default, the subroutines provide the index 2 formulation.

Comments to simeon@ma.tum.de.

The software part of the problem is in the file crank.f available at [MM08].

19.2 Mathematical description of the problem

The original problem has the form

\[
M(p, q) \begin{pmatrix} \dot{p} \\ \dot{q} \end{pmatrix} = f(p, \dot{p}, q, \dot{q}) - G(p, q)^T \lambda, \\
0 = g(p, q) + r(t),
\]

where \(0 \leq t \leq 0.1\), \(p \in \mathbb{R}^3\), \(q \in \mathbb{R}^4\), \(\lambda \in \mathbb{R}^3\), \(M : \mathbb{R}^7 \to \mathbb{R}^7 \times \mathbb{R}^7\), \(f : \mathbb{R}^{14} \to \mathbb{R}^7\), \(g : \mathbb{R}^7 \to \mathbb{R}^3\), \(r : \mathbb{R} \to \mathbb{R}^3\), and \(G = \partial g/\partial (p, q)\). The matrix \(M(p, q)\) is symmetric positive semi-definite and rank \(M(p, q)\) is 3, which implies that the DAE (II.19.1) is of index 3. For the index 2 formulation, the position constraints are replaced by the velocity constraints

\[
0 = \frac{d}{dt} \begin{pmatrix} g(p, q) + r(t) \end{pmatrix} = G(p, q) \begin{pmatrix} \dot{p} \\ \dot{q} \end{pmatrix} + \dot{r}(t).
\]

(II.19.2)

Additionally, the system is transformed to first order and semi explicit form

\[
\begin{pmatrix} \dot{p} \\ \dot{q} \end{pmatrix} = \begin{pmatrix} v_p \\ v_q \end{pmatrix}, \\
\begin{pmatrix} \dot{v}_p \\ \dot{v}_q \end{pmatrix} = \begin{pmatrix} \dot{a}_p \\ \dot{a}_q \end{pmatrix},
\]

(II.19.3)

\[
0 = M(p, q) \begin{pmatrix} a_p \\ a_q \end{pmatrix} - f(p, v_p, q, v_q) + G(p, q)^T \lambda, \\
0 = G(p, q) \begin{pmatrix} v_p \\ v_q \end{pmatrix} + \dot{r}(t),
\]

which increases the dimension of the problem to 24. If we define \(y := (p, q, v_p, v_q, a_p, a_q, \lambda)^T\), then the consistent values are given by \(y(0) := y_0\) and \(y'(0) := y'_0\). The components of \(y_0\) are zero, except for
The first 14 components of $y_0'$ read $y_{0,i}' = y_{0,i+7}$, $i = 1, \ldots, 14$; the last 10 are zero.

For the index 2 formulation, the index of the variables $p$, $q$, $v_p$ and $v_q$ equals 1 and that of $a_p$, $a_q$ and $\lambda$ equals 2. The equations are given in detail in the next subsections, in which already some references to the origin of the problem, treated in §19.3, are given.

19.2.1 Equations of motion

The position or gross motion coordinates $p$ are

$$p := \begin{pmatrix} \phi_1 \\ \phi_2 \\ x_3 \end{pmatrix}$$

The deformation coordinates $q$ (of the elastic connecting rod, see below) are

$$q := \begin{pmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{pmatrix}$$

The mass matrix $M$ reads

$$M(p,q) = \begin{pmatrix} M_r(p) + M_c(p,q) & C(p,q)^T \\ C(p,q) & M_\Delta \end{pmatrix}$$

with rigid motion mass matrix

$$M_r(p) = \begin{pmatrix} J_1 + m_3 l_1^2 & 1/2 l_1 l_2 m_2 \cos(\phi_1 - \phi_2) & 0 \\ 1/2 l_1 l_2 m_2 \cos(\phi_1 - \phi_2) & J_2 & 0 \\ 0 & 0 & m_3 \end{pmatrix},$$

coupling blocks

$$M_c(p,q) = \begin{pmatrix} 0 & \rho l_1(\cos(\phi_1 - \phi_2)c_1^T + \sin(\phi_1 - \phi_2)c_2^T)q & 0 \\ 0 & q^T M_\Delta q + 2\rho c_{12}^T q & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

and

$$C(p,q)^T = \begin{pmatrix} \rho c_{12}^T + \rho q B^T \end{pmatrix}.$$
and elastic body space discretization mass matrix

\[
M_\Delta = \rho d h l_2 \begin{pmatrix}
1/2 & 0 & 0 & 0 \\
0 & 1/2 & 0 & 0 \\
0 & 0 & 8/1 & 1 \\
0 & 0 & 1/2 & 0
\end{pmatrix}.
\]

The forces are given by

\[
f(p, \dot{p}, q, \dot{q}) = \left( f_r(p, \dot{p}) + f_c(p, \dot{p}, q, \dot{q}) \right) - \text{grad } W_\Delta(q) - D_\Delta \dot{q},
\]

where the rigid motion terms are collected in

\[
f_r(p, \dot{p}) = \begin{pmatrix}
-1/2 l_1 (\gamma(m_1 + 2m_2) \cos(\phi_1 + l_2 m_2 \phi_2^2 \sin(\phi_1 - \phi_2))) \\
-1/2 l_2 \gamma m_2 \cos(\phi_2) + 1/2 l_1 l_2 m_2 \phi_2 \sin(\phi_1 - \phi_2) \\
0
\end{pmatrix}.
\]

For the force term \( f_c(p, \dot{p}, q, \dot{q}) \) we have

\[
\begin{pmatrix}
\rho l_1 \phi_2^2 (-\sin(\phi_1 - \phi_2) c_1^T + \cos(\phi_1 - \phi_2) c_2^T) \dot{q} - 2\rho l_1 \dot{\phi}_2 (\cos(\phi_1 - \phi_2) c_1^T + \sin(\phi_1 - \phi_2) c_2^T) \dot{q} \\
\rho l_1 \phi_2^2 (\sin(\phi_1 - \phi_2) c_1^T - \cos(\phi_1 - \phi_2) c_2^T) \dot{q} - 2\rho \phi_2 c_2^T \dot{q} - 2\phi_2^2 \dot{q} M_\Delta \dot{q} \\
-\rho \dot{q}^T B \dot{q} - \rho \gamma (\cos \phi_2 c_1^T \dot{q} - \sin \phi_2 c_2^T \dot{q}) \\
0
\end{pmatrix},
\]

and for \( f_\Delta(p, \dot{p}, q) \) the expression

\[
\ddot{\phi}_2 M_\Delta q + \rho \left( \ddot{\phi}_2^2 c_{12} + l_1 \dot{\phi}_1^2 (\cos(\phi_1 - \phi_2) c_1 + \sin(\phi_1 - \phi_2) c_2) + 2\dot{\phi}_2 B \dot{q} \right) - \rho \gamma \left( \sin \phi_2 c_1 + \cos \phi_2 c_2 \right) .
\]

The gradient of the elastic potential \( W_\Delta(q) \) in case of linear elasticity (which is the default) is

\[
\text{grad } W_\Delta(q) = K_\Delta q
\]

with stiffness matrix

\[
K_\Delta = E d h / h_2 \begin{pmatrix}
\pi^4/24 (h/l_2)^2 & 0 & 0 & 0 \\
0 & \pi^4/2 (h/l_2)^2 & 0 & 0 \\
0 & 0 & 16/3 & -8/3 \\
0 & 0 & -8/3 & 7/3
\end{pmatrix}.
\]

Alternatively, in case of the nonlinear beam model (IPAR(1) = 1, see below), it holds

\[
\text{grad } W_\Delta(q) = K_\Delta q + k_\Delta(q).
\]

\[
k_\Delta(q) = 1/2 \pi^2 E d h / h_2 \begin{pmatrix}
q_1 q_4 - \beta q_2 (-4q_3 + 2q_4) \\
4q_2 q_4 - \beta q_1 (-4q_3 + 2q_4) \\
4\beta q_1 q_2 \\
1/2 q_1^2 + 2\beta^2 q_2
\end{pmatrix}, \quad \beta = 80/(9\pi^2).
\]

The damping matrix \( D_\Delta \) is by default zero. The coupling matrices and vectors arising from the space discretization read

\[
B = d h l_2 \begin{pmatrix}
0 & 0 & -16/\pi^3 & 8/\pi^3 - 1/\pi \\
0 & 0 & 0 & 1/(2\pi) \\
16/\pi^3 & 0 & 0 & 0 \\
1/\pi - 8/\pi^3 & -1/(2\pi) & 0 & 0
\end{pmatrix}.
\]
and
\[
\begin{align*}
    c_1 &= d h l_2 (0, 0, 2/3, 1/6)^T, \\
    c_2 &= d h l_2 (2/\pi, 0, 0, 0)^T, \\
    c_{12} &= d h l_2^2 (0, 0, 1/3, 1/6)^T, \\
    c_{21} &= d h l_2^2 (1/\pi, -1/(2\pi), 0, 0)^T.
\end{align*}
\]
Finally, the position constraints \(0 = g(p, q) + r(t)\) are given by
\[
\begin{align*}
    0 &= l_1 \sin \phi_1 + l_2 \sin \phi_2 + q_4 \sin \phi_2, \\
    0 &= x_3 - l_1 \cos \phi_1 - l_2 \cos \phi_2 - q_4 \cos \phi_2, \\
    0 &= \phi_1 - \Omega t.
\end{align*}
\]

\textbf{19.2.2 Parameters}

For the simulation, the following data are used:
- The bodies have lengths \(l_1 = 0.15\), \(l_2 = 0.30[m]\).
- The masses of the bodies are \(m_1 = 0.36\), \(m_2 = 0.151104\), \(m_3 = 0.075552[kg]\).
- The moments of inertia are \(J_1 = 0.002727\), \(J_2 = 0.0045339259[kg m^2]\).
- The flexible connecting rod has height and width \(h = d = 0.008[m]\).
- The mass density \(\rho = 7870[kg/m^3]\), and Young’s modulus \(E = 2 \cdot 10^{11}[N/m^2]\).
- The gravity constant was set to zero since gravitation plays no role here, \(\gamma = 0\).
- The angular velocity of the prescribed crank motion is \(\Omega = 150[rad/s]\).

\textbf{19.3 Origin of the problem}

The planar slider crank mechanism, see Figure II.19.1, consists of a rigid crank (body 1), an elastic connecting rod (body 2), a rigid sliding block (body 3) and two revolving and one translational joint. Koppens \[Kop89\] and Jahnke \[JPD93\] investigated this example using an ODE model in minimum coordinates. In \[Sim96\], an alternative DAE approach is introduced.
The mathematical model outlined above is derived in two steps. First, the elastic connecting rod is discretized in space. The geometry of the rod allows to apply an Euler-Bernoulli beam
\[
\begin{align*}
    u_1(x, y) &= w_1(x) - yw_2'(x), \\
    u_2(x, y) &= w_2(x),
\end{align*}
\]
to describe the longitudinal and lateral displacements \(u_1\) and \(u_2\) of material point \((x, y)\) in the body-fixed coordinate system. For the longitudinal displacement \(w_1\) of the neutral fiber, a simple quadratic model
\[
w_1(x) = \xi^2(-4q_3 + 2q_4) + \xi(4q_3 - q_4), \quad \xi = x/l_2,
\]
is sufficient to show the basic effects. The lateral displacement \(w_2\) is approximated by the first two sinus shape functions
\[
w_2(x) = \sin(\pi \xi) q_1 + \sin(2\pi \xi) q_2.
\]
These functions satisfy the boundary conditions \(w_1(0) = 0, w_2(0) = 0, w_2(l_2) = 0\). Accordingly, the body-fixed coordinate system’s origin is placed in \((x, y) = (0, 0)\), and its \(x\)-axis passes through the point \((l_2 + w_1(x), 0)\).

As already mentioned in §19.2, we provide two versions of the problem. The first one (default) assumes linear elasticity while the second takes the coupling of longitudinal and lateral displacements in terms of \(k_\Delta(q)\) into account. Set \(\text{IPAR(1)} = 1\) to switch to this nonlinear beam model. See below for a comparison of the results.

In the second step, the equations of motion of the overall multibody system are assembled. Due to the choice of \(\phi_2\) as gross motion coordinate, there is no constraint equation necessary to express the revolving joint between crank and connecting rod. The revolving joint between sliding block and connecting rod and the translational joint lead to two constraints that depend on the deformation variable \(q_4\). The third constraint equation defines the crank motion using \(r(t) = (0, 0, -\Omega t)^T\). Here, other functions for the crank motion could also be prescribed.

The model described so far features no dissipation. Consequently, the solutions show a purely oscillatory behavior. We supply also a nonzero damping matrix \(D_\Delta\) which can be activated by setting \(\text{IPAR(2)} = 1\). Then, 0.5 percent dissipation is included in the right hand side of the elastic connecting rod.

In §19.4, we investigate the dynamic behavior of the slider crank model corresponding to the nonlinear model without damping with the initial values listed in §19.2, which were calculated such that the motion is almost smooth, using an asymptotic expansion technique [Sim97]. In Figure II.19.4 we see the behavior of the numerical solution for this setting of the model. A close look at these plots reveals that both lateral displacements \(q_1, q_2\) as well as longitudinal displacements \(q_3, q_4\) still show some small oscillations. The corresponding frequencies as solutions of the eigenvalue problem \(\omega^2 M_\Delta q = K_\Delta q\) are
\[
\omega_1 = 1277, \quad \omega_2 = 5107, \quad \omega_3 = 6841, \quad \omega_4 = 24613 \ [rad/s].
\]
In particular, \(q_3\) and \(q_4\) are characterized by the relatively large frequency \(\omega_4\). Any explicit discretization in time will need step sizes smaller than the shortest period of oscillation, even for tracking a smooth solution. On the other hand, the challenge for implicit methods is to be able to take larger steps. In this simulation the gross motion coordinates \(p\) differ only slightly from the motion of a mechanism with rigid connecting rod.

The subroutines that describe the model offer several possibilities to test other variants of the model than those tested in §19.4. We now discuss some of them.

**Oscillatory solution**

We provide also a second set of initial values (subroutine init2) which lead to a strongly oscillatory solution. Here, the initial deformation as well as the corresponding velocity were set to zero, \(q(0) = \ldots \)

The mathematical model outlined above is derived in two steps. First, the elastic connecting rod is discretized in space. The geometry of the rod allows to apply an Euler-Bernoulli beam
\[
\begin{align*}
    u_1(x, y) &= w_1(x) - yw_2'(x), \\
    u_2(x, y) &= w_2(x),
\end{align*}
\]
to describe the longitudinal and lateral displacements \(u_1\) and \(u_2\) of material point \((x, y)\) in the body-fixed coordinate system. For the longitudinal displacement \(w_1\) of the neutral fiber, a simple quadratic model
\[
w_1(x) = \xi^2(-4q_3 + 2q_4) + \xi(4q_3 - q_4), \quad \xi = x/l_2,
\]
is sufficient to show the basic effects. The lateral displacement \(w_2\) is approximated by the first two sinus shape functions
\[
w_2(x) = \sin(\pi \xi) q_1 + \sin(2\pi \xi) q_2.
\]
These functions satisfy the boundary conditions \(w_1(0) = 0, w_2(0) = 0, w_2(l_2) = 0\). Accordingly, the body-fixed coordinate system’s origin is placed in \((x, y) = (0, 0)\), and its \(x\)-axis passes through the point \((l_2 + w_1(x), 0)\).

As already mentioned in §19.2, we provide two versions of the problem. The first one (default) assumes linear elasticity while the second takes the coupling of longitudinal and lateral displacements in terms of \(k_\Delta(q)\) into account. Set \(\text{IPAR(1)} = 1\) to switch to this nonlinear beam model. See below for a comparison of the results.

In the second step, the equations of motion of the overall multibody system are assembled. Due to the choice of \(\phi_2\) as gross motion coordinate, there is no constraint equation necessary to express the revolving joint between crank and connecting rod. The revolving joint between sliding block and connecting rod and the translational joint lead to two constraints that depend on the deformation variable \(q_4\). The third constraint equation defines the crank motion using \(r(t) = (0, 0, -\Omega t)^T\). Here, other functions for the crank motion could also be prescribed.

The model described so far features no dissipation. Consequently, the solutions show a purely oscillatory behavior. We supply also a nonzero damping matrix \(D_\Delta\) which can be activated by setting \(\text{IPAR(2)} = 1\). Then, 0.5 percent dissipation is included in the right hand side of the elastic connecting rod.

In §19.4, we investigate the dynamic behavior of the slider crank model corresponding to the nonlinear model without damping with the initial values listed in §19.2, which were calculated such that the motion is almost smooth, using an asymptotic expansion technique [Sim97]. In Figure II.19.4 we see the behavior of the numerical solution for this setting of the model. A close look at these plots reveals that both lateral displacements \(q_1, q_2\) as well as longitudinal displacements \(q_3, q_4\) still show some small oscillations. The corresponding frequencies as solutions of the eigenvalue problem \(\omega^2 M_\Delta q = K_\Delta q\) are
\[
\omega_1 = 1277, \quad \omega_2 = 5107, \quad \omega_3 = 6841, \quad \omega_4 = 24613 \ [rad/s].
\]
In particular, \(q_3\) and \(q_4\) are characterized by the relatively large frequency \(\omega_4\). Any explicit discretization in time will need step sizes smaller than the shortest period of oscillation, even for tracking a smooth solution. On the other hand, the challenge for implicit methods is to be able to take larger steps. In this simulation the gross motion coordinates \(p\) differ only slightly from the motion of a mechanism with rigid connecting rod.

The subroutines that describe the model offer several possibilities to test other variants of the model than those tested in §19.4. We now discuss some of them.

**Oscillatory solution**

We provide also a second set of initial values (subroutine init2) which lead to a strongly oscillatory solution. Here, the initial deformation as well as the corresponding velocity were set to zero, \(q(0) = \ldots \)
$v_1(0) = 0$, which is equivalent to consistent initial values on a rigid motion trajectory. Figure II.19.2 plots the behavior of $q_1$, $q_2$ and $q_4$ for this setting. Both lateral and longitudinal modes oscillate now with different frequencies.

**Nonlinear beam model and damping**

The left and right plot in Figure II.19.3 show the effects of setting IPAR(1) = 1 and IPAR(2) = 1, respectively. On the left, the difference between linear and nonlinear beam model is illustrated, with initial values close to the smooth motion. In particular, the components $q_3$ and $q_4$ change if the nonlinear model is employed. At points of maximum bending, the longitudinal displacement has now much smaller minima. If we increase the crank’s angular velocity, the resulting forces acting on the connecting rod are much larger and we can then even observe how the sharp needles turn into a singularity, the buckling phenomenon.

On the right of Figure II.19.3, the damping was activated by IPAR(2) = 1, with initial values on a rigid motion trajectory (init2). Obviously, the oscillation shown in Figure II.19.2 on the right is now slowly damped out.
19.4 Numerical solution of the problem

The results presented here refer to index 2 formulation of the linear model without damping, using the initial values corresponding to a smooth solution.

Tables II.19.2–II.19.3 and Figures II.19.4–II.19.8 present the reference solution at the end of the integration interval, the run characteristics, the behavior of some of the solution components over the integration interval and the work-precision diagrams, respectively. In computing the scd values, only the first seven and the last three components were taken into account, since they refer to the physically important quantities. The reference solution was computed using MEBDFI with atol = $10^{-14}$ and rtol = $10^{-14}$ and h0 = $10^{-12}$. For the work-precision diagrams, we used: rtol = $10^{-(4+m/4)}$, $m = 0, \ldots, 24$; atol = rtol; h0 = $10^{-2}$ · rtol for BMD, GAMD, MEBDFDAE, MEBDFI, RADAU5 and RADAU. The failed runs are in Table II.19.1; listed are the name of the solver that failed, for which values of $m$ this happened, and the reason for failing.

Remarks

- The slider crank is an example for a stiff mechanical system given in DAE form. See Lubich [Lub93] for an investigation of such systems and the implications for numerical methods in the ODE case.

- The nonlinear beam model leads to a higher computational effort but does not provoke convergence failures of Newton's method in RADAU5, as might be expected in case of nonlinear stiffness terms.
As an alternative to stiff solvers, it is still possible to apply methods based on explicit discretizations, e.g., half-explicit or projection methods for constrained mechanical systems. The code MDOP5 [Sim95], a projection method based on DOPRI5, uses 2260 integration steps to solve this problem in the default setting, with atol = 10^{-6} and rtol = 10^{-5}, and initial values close to the smooth motion. Thus, the stiffness is not that severe in case of this carefully chosen one-dimensional elastic body model.

There is also an extended version of the slider crank with a two-dimensional FE grid for the connecting rod. There, explicit methods do not work any longer. An animation of the system motion can be found at http://www.mathematik.tu-darmstadt.de/~simeon/.

### References


Figure II.19.4: Behavior of the $i$th solution component; $i \in \{2, 3, \ldots, 7, 22, 23, 24\}$.


Figure II.19.5: Work-precision diagram (scd versus CPU-time).
Figure II.19.6: Work-precision diagram (scd versus CPU-time).
Figure II.19.7: Work-precision diagram (nescd versus CPU-time).
Figure II.19.8: Work-precision diagram (mescd versus CPU-time).
II-19-14

DAE - Slider crank
20 Water tube system

20.1 General information

This IVP is an index 2 system of 49 non-linear Differential-Algebraic Equations and describes the water flow through a tube system, taking into account turbulence and the roughness of the tube walls. The parallel-IVP-algorithm group of CWI contributed this problem to the test set in cooperation with B. Koren (CWI) and Paragon Decision Technology B.V.

The software part of the problem is in the file `water.f` available at [MM08].

20.2 Mathematical description of the problem

The problem is of the form

\[ M \frac{dy}{dt} = f(t,y), \quad y(0) = y_0, \quad y'(0) = y'_0, \]  

where \( 0 \leq t \leq 17 \cdot 3600 \) and \( y \in \mathbb{R}^{49} \). Furthermore,

\[ M = \begin{bmatrix} M^\phi & O & O \\ O & O & O \\ O & O & M^p \end{bmatrix}, \]  

where \( M^\phi \in \mathbb{R}^{18 \times 18} \) and \( M^p \in \mathbb{R}^{13 \times 13} \) are given by

\[ M^\phi_{i,j} = \begin{cases} v_i & \text{for } i = j, \\ 0 & \text{otherwise.} \end{cases} \]

\[ M^p_{i,j} = \begin{cases} C_5 & \text{for } i = j = 1, \\ C_8 & \text{for } i = j = 2, \\ 0 & \text{otherwise.} \end{cases} \]

The first 38 components of \( y \) are of index 1, the last 11 are of index 2. For the definition of \( f \) and the values of \( C_5, C_8 \) and \( v \) we refer to \S 20.3.

The initial vectors \( y_0 \) and \( y'_0 \) are given by

\[ y_0 = \begin{cases} 0 & \text{for } i = 1, 2, \ldots, 18 \\ 0.047519404529185289807 & \text{for } i = 19, 20, \ldots, 36 \end{cases} \quad \text{and} \quad y'_0 = (0, \ldots, 0)^T. \]  

The function \( f \) contains several square roots. It is clear that the function can not be evaluated if one of the arguments of one of these square roots becomes negative. To prevent this situation, we set \( \text{IERR}=-1 \) in the Fortran subroutine that defines \( f \) if this happens. See page IV-ix of the the description of the software part of the test set for more details on \( \text{IERR} \).

20.3 Origin of the problem

This test example describes how water flows through a water tube system. The system is represented by a set of nodes, which are connected by tubes. The structure of the water tube system is depicted in Figure II.20.1. There are two types of nodes: normal nodes and buffer nodes, to which a buffer is attached. We denote the set of all nodes by \( N \), and the set of buffer nodes by \( B \). For the system under consideration, \( B = \{5, 8\} \). The rectangles in Figure II.20.1 represent the buffers. The pipes are in the horizontal plane; the buffers are connected to the nodes perpendicular to this plane. The pipes from the buffer nodes to the rectangles are virtual; in reality the buffers are directly attached to the buffer nodes. In the model every node can have inflow and outflow, which are denoted by \( e_i^\text{in}(t) \) and \( e_i^\text{out}(t) \). In our example, inflow occurs only at node 1 and node 13, whereas only node 10 has outflow.
The unit of time in the model is second. Defining the time in hours by $\hat{t} = t/3600$, these flows are defined by

$$e_{11}^{in}(t) = (1 - \cos(e^{-\hat{t}} - 1))/200,$$
$$e_{13}^{in}(t) = (1 - \cos(e^{-\hat{t}} - 1))/80,$$
$$e_{10}^{out}(t) = \hat{t}^2(3\hat{t}^2 - 92\hat{t} + 720)/10^6.$$

Figure II.20.2 shows plots of these flows as function of $\hat{t}$. Note that the outflow has a peak at 8 AM and is increasing again after 3 PM.

Although it seems that node 6 and node 9 could be omitted, we include them in the model, to leave open the possibility that these nodes have inflow or outflow. The arrows in Figure II.20.1 denote the direction in which we compute the flows. For example, if there is a flow from node 4 to node 3, then this flow will be negative.

To model the flow of the water, we introduce some symbols, which are listed in Table II.20.1. The roughness $k_{i,j} = 2 \cdot 10^{-4}$ is measured as the average height of the obstacles on the tube wall. The structure $S_{i,j}$ is defined as

$$S_{i,j} = \begin{cases} 
1 & \text{if there is a tube from } i \text{ to } j, \\
0 & \text{otherwise.}
\end{cases}$$
Figure II.20.2: Inflows and outflow in m³/s as function of time in hours.

From Figure II.20.1 we see that

\[
S = \begin{bmatrix}
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
\]

Some of the quantities in Table II.20.1 can be computed directly from others:

\[
\mu = \nu \cdot \rho, \\
\phi_{i,j}(t) = u_{i,j}(t) \cdot A_{i,j}, \\
A_{i,j} = \pi \cdot \frac{d^2_{ij}}{4}, \\
m_{i,j} = A_{i,j} \cdot l_{i,j} \cdot \rho, \\
R_{i,j}(t) = u_{i,j}(t) \cdot d_{i,j} \cdot \nu.
\]

The definition of \(R_{i,j}(t)\) was taken from [Sch78, p. 816].
Table II.20.1: List of symbols for modeling flow in tubes.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Unit</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\phi_{i,j}(t))</td>
<td>(m^3/s)</td>
<td>flow through tube from (i) to (j) at time (t)</td>
</tr>
<tr>
<td>(u_{i,j}(t))</td>
<td>(m/s)</td>
<td>mean velocity of flow through tube from (i) to (j) at time (t)</td>
</tr>
<tr>
<td>(F_{i,j}(t))</td>
<td>(N)</td>
<td>total force on water in tube from (i) to (j) at time (t)</td>
</tr>
<tr>
<td>(F_{i,j}^a(t))</td>
<td>(N)</td>
<td>adhesion force on water in tube from (i) to (j) at time (t)</td>
</tr>
<tr>
<td>(\lambda_{i,j}(t))</td>
<td>-</td>
<td>coefficient of resistance of tube from (i) to (j) at time (t)</td>
</tr>
<tr>
<td>(R_{i,j}(t))</td>
<td>(N/m^2)</td>
<td>Reynolds number of flow through tube from (i) to (j) at time (t)</td>
</tr>
<tr>
<td>(p_i(t))</td>
<td>(N/m^2)</td>
<td>pressure in (i) at time (t)</td>
</tr>
<tr>
<td>(S_{i,j})</td>
<td>-</td>
<td>incidence matrix for structure of the tube system</td>
</tr>
<tr>
<td>(m_{i,j})</td>
<td>(kg)</td>
<td>mass of water in tube from (i) to (j)</td>
</tr>
<tr>
<td>(d_{i,j})</td>
<td>(m)</td>
<td>diameter of tube from (i) to (j)</td>
</tr>
<tr>
<td>(l_{i,j})</td>
<td>(m)</td>
<td>length of tube from (i) to (j)</td>
</tr>
<tr>
<td>(A_{i,j})</td>
<td>(m^2)</td>
<td>area of tube from (i) to (j)</td>
</tr>
<tr>
<td>(k_{i,j})</td>
<td>(m)</td>
<td>roughness of wall of tube from (i) to (j)</td>
</tr>
<tr>
<td>(e^\text{in}_{i}(t))</td>
<td>(m^3/s)</td>
<td>inflow at (i) at time (t)</td>
</tr>
<tr>
<td>(e^\text{out}_{i}(t))</td>
<td>(m^3/s)</td>
<td>outflow at (i) at time (t)</td>
</tr>
<tr>
<td>(B_i) ((i \in B))</td>
<td>(m^2)</td>
<td>area of buffer (i)</td>
</tr>
<tr>
<td>(R^\text{crit})</td>
<td>-</td>
<td>critical Reynolds number</td>
</tr>
<tr>
<td>(g)</td>
<td>(m/s^2)</td>
<td>gravity constant</td>
</tr>
<tr>
<td>(\rho)</td>
<td>(kg/m^3)</td>
<td>density of water</td>
</tr>
<tr>
<td>(\mu)</td>
<td>(kg/(m \cdot s))</td>
<td>viscosity of water</td>
</tr>
<tr>
<td>(v)</td>
<td>(m^2/s)</td>
<td>kinematic viscosity of water</td>
</tr>
<tr>
<td>(v)</td>
<td>(kg/m^4)</td>
<td>auxiliary vector, see (II.20.15)</td>
</tr>
</tbody>
</table>

We now explain how to model the flow through a tube, using Newton’s second Law, which states that

\[
m_{i,j} \frac{du_{i,j}(t)}{dt} = F_{i,j}(t).
\] (II.20.4)

Assuming that gravity has no influence on the water flow in all tubes (remember that the pipes are in the horizontal plane), we see from Figure II.20.3 that the total force on the water in a tube equals

\[
F_{i,j}(t) = A_{i,j}(p_i(t) - p_j(t)) - F_{i,j}^a(t).
\] (II.20.5)
The magnitude of the adhesion force depends on the type of flow. For laminar flows ($|R_{i,j}(t)| \leq R_{crit}^i$), we use the formula [Sch78, p. 12]

$$F_{i,j}^{in}(t)/A_{i,j} = 32\mu \cdot l_{i,j} \cdot u_{i,j}(t)/d_{i,j}^2. \quad (\text{II.20.6})$$

For turbulent flows ($|R_{i,j}(t)| > R_{crit}^i$), we have [Sch78, p. 597]

$$F_{i,j}^{in}(t)/A_{i,j} = \lambda_{i,j}(t) \cdot \rho \cdot l_{i,j} \cdot u_{i,j}(t)^2/d_{i,j}. \quad (\text{II.20.7})$$

where the resistance $\lambda_{i,j}(t)$ is computed from Colebrook and White’s formula [Sch78, p. 621]:

$$0 = \frac{1}{\sqrt{\lambda_{i,j}(t)}} - 1.74 + 2 \log \left( \frac{2k_{i,j}}{d_{i,j}} + \frac{18.7}{|R_{i,j}(t)|\sqrt{\lambda_{i,j}(t)}} \right). \quad (\text{II.20.8})$$

Although for laminar flows the adhesion force does not depend on the resistance coefficient (cf. (II.20.6)), we have to choose a value for $\lambda_{i,j}$ in case of laminar flows. We compute this value by replacing $R_{i,j}$ in (II.20.8) by $R_{crit}$, i.e., we choose the value such that if a flow changes from laminar into turbulent, the resistance coefficient changes gradually.

For the normal nodes, Kirchoff’s law holds, which states that

$$\forall n \in \mathcal{N} - \mathcal{B} : \quad 0 = \sum_{i \mid |S_{i,n}| = 1} \phi_{i,n}(t) + e_{i,n}^{in}(t) - \sum_{j \mid |S_{n,j}| = 1} \phi_{n,j}(t) - e_{n,j}^{out}(t) \quad (\text{II.20.9})$$

For the buffer nodes, we add a term $\psi_n(t)$ that represents the flow to the buffer:

$$\forall n \in \mathcal{B} : \quad \psi_n(t) = \sum_{i \mid |S_{i,n}| = 1} \phi_{i,n}(t) + e_{i,n}^{in}(t) - \sum_{j \mid |S_{n,j}| = 1} \phi_{n,j}(t) - e_{n,j}^{out}(t) \quad (\text{II.20.10})$$

We now explain how to compute $\psi_n(t)$. A buffer can be interpreted as the water column in Figure II.20.4, with ground area $B_n$ and height $h$. Due to the flow $\psi_n(t)$ the height of the buffer changes at a rate $\psi_n(t)/B_n$. The difference between the pressure at the top and bottom of the column satisfies

$$p_n - p_0 = g \cdot \rho \cdot h.$$ 

Consequently, the pressure difference changes at a rate given by

$$\frac{d(p_n - p_0)}{dt} = g \cdot \rho \cdot \frac{dh}{dt} = g \cdot \rho \cdot \frac{\psi_n(t)}{B_n}. \quad (\text{II.20.11})$$

Notice that the pressure $p_0$ is constant and therefore drops out in this formula. Substituting (II.20.11) in (II.20.10) gives

$$\forall n \in \mathcal{B} : \quad C_n \frac{dp_n(t)}{dt} = \sum_{i \mid |S_{i,n}| = 1} \phi_{i,n}(t) + e_{i,n}^{in}(t) - \sum_{j \mid |S_{n,j}| = 1} \phi_{n,j}(t) - e_{n,j}^{out}(t), \quad (\text{II.20.12})$$

where the quantity $C_n := B_n/(\rho \cdot g)$ can be interpreted as the capacity of the buffer at node $n$.

We arrive at the formulation in §20.2 by setting

$$y = \begin{pmatrix} \phi_{1,2}(t), \phi_{2,3}(t), \phi_{2,6}(t), \phi_{3,4}(t), \phi_{3,5}(t), \phi_{4,5}(t), \phi_{5,10}(t), \phi_{6,5}(t), \phi_{7,4}(t), \\ \phi_{7,8}(t), \phi_{8,5}(t), \phi_{8,10}(t), \phi_{9,8}(t), \phi_{11,9}(t), \phi_{11,12}(t), \phi_{12,7}(t), \phi_{12,8}(t), \phi_{13,11}(t), \\ \lambda_{1,2}(t), \lambda_{2,3}(t), \lambda_{2,6}(t), \lambda_{3,4}(t), \lambda_{3,5}(t), \lambda_{4,5}(t), \lambda_{5,10}(t), \lambda_{6,5}(t), \lambda_{7,4}(t), \\ \lambda_{7,8}(t), \lambda_{8,5}(t), \lambda_{8,10}(t), \lambda_{9,8}(t), \lambda_{11,9}(t), \lambda_{11,12}(t), \lambda_{12,7}(t), \lambda_{12,8}(t), \lambda_{13,11}(t), \\ p_5(t), p_8(t), p_1(t), p_2(t), \ldots, p_4(t), p_6(t), p_9(t), p_{10}(t), \ldots, p_{13}(t) \end{pmatrix}^T. \quad (\text{II.20.13})$$
All pressures are of index 2, except for those at the buffer nodes. The reordering of the pressures in (II.20.13) is such that the elements in \( y \) appear in order of increasing index, as required by RADAU, RADAU5 and MEBDFDAE.

The first 18 equations in (II.20.1) are obtained by first substituting (II.20.5) in (II.20.4). Next, we divide both sides by \( A_{i,j} \), thus yielding

\[
\frac{\rho \cdot l_{i,j}}{A_{i,j}} \frac{d\phi_{i,j}(t)}{dt} = p_i(t) - p_j(t) - F_{i,j}(t)/A_{i,j}. \tag{II.20.14}
\]

Finally, (II.20.6) and (II.20.7) are substituted in (II.20.14). Consequently, if we define \( V_{i,j} = \rho \cdot l_{i,j}/A_{i,j} \), then the vector \( v \) in (II.20.2) is given by

\[
v = (V_{1,2}, V_{2,3}, V_{2,6}, V_{3,4}, V_{3,5}, V_{4,5}, V_{5,10}, V_{6,5}, V_{7,4}, V_{7,8}, V_{8,5}, V_{8,10}, V_{9,8}, V_{11,12}, V_{12,7}, V_{12,8}, V_{13,11})^T. \tag{II.20.15}
\]

The next 18 equations in (II.20.1) equal (II.20.8), whereas the last 13 equations are given by (II.20.9) and (II.20.12).

In this model, all tubes and buffers are equal with characteristics as specified in Table II.20.2. Moreover, we assume that the temperature is constant. The values for the physical constants are listed in Table II.20.3. The values for \( R_{\text{crit}} \) was taken from [Sch78, p. 39].

We now discuss how we derived the initial conditions in (II.20.3). First we note that (II.20.9) is an index 2 constraint. Therefore, the initial values also have to satisfy the once differentiated constraint (the so-called hidden constraint)

\[
\forall n \in N - B : \quad 0 = \sum_{i|S_{i,n}=1} \phi_{i,n}(t) + e_{i,n}^{\text{in}}(t) - \sum_{j|S_{n,j}=1} \phi_{n,j}(t) - e_{n,j}^{\text{out}}(t). \tag{II.20.16}
\]
We are free to choose initial flows $\phi_{i,j}(0)$ as long as they satisfy (II.20.9); we chose these all equal to zero. This means that the resistance coefficients equal the value for the case of laminar flows, i.e., 0.047519... The pressures at the buffer nodes, which can be selected freely, are chosen to be $10^5 + g \cdot \rho$, which corresponds to initial heights of one meter in the water columns, assuming that $p_0$ in Figure II.20.4 equals one bar. From (II.20.12) it follows that $p_n'(0) = 0$, $n \in B$ (note that the in- and outflows are initially zero). The initial pressures $p_n(0)$, $n \in N - B$, and the initial derivative flows $\phi'_n(0)$ follow from (II.20.14) and (II.20.16). Since the derivatives of the in- and outflows are initially zero, the initial values in (II.20.3) satisfy these equations. The other initial values, $\lambda_{i,j}'(0)$ and $l_n(0)$, $n \in N - B$, appear neither in the system, nor in the hidden constraints, and can be chosen freely. We set these equal to 0.

Several observations can be made from the behavior of the flows, resistance coefficients and pressures, which are plotted in Figure II.20.6–II.20.8:

- The rise and fall of the outflow in node 10 cause the flows to node 10 to change from laminar to turbulent and back, as can be seen from the resistance coefficients $\lambda_{5,10}$ and $\lambda_{8,10}$, which correspond to $y_{25}$ and $y_{30}$.

- At 8 AM, the pressures in the buffer nodes drop below their original level, which means that some of the water that was present in the buffers initially, is used to meet the peak demand.

- The time period in which the flows to node 10 have become laminar again (this period is indicated by the vertical dashed lines in the plots of $y_{25}$ and $y_{30}$, causes an irregular behavior (indicated again by dashed lines) of the solution components $y_3$, $y_6$, $y_9$, $y_{10}$ and $y_{11}$ which correspond to the flow from node 3 to node 4 and the flows in the cycle 4–7–8–5, respectively.

- Some of the flows contain high-frequent oscillations of small amplitude. To see this more clearly, we plotted $\phi_{3,4}$ for $6878 < t \leq 17 \cdot 3600$ in Figure II.20.5.

### 20.4 Numerical solution of the problem

Tables II.20.4–II.20.5 and Figures II.20.6–II.20.8 present the reference solution at the end of the integration interval, the run characteristics, the behavior of the solution over the integration interval and the work-precision diagrams, respectively.
Since the 13 last solution components (the pressures) are so much larger in magnitude than the other components, we used the following vector-valued input tolerances:

\[
\begin{align*}
\text{atol}(i) &= \text{atol} & & \text{for } i = 1, \ldots, 36, \\
\text{atol}(i) &= 10^6 \cdot \text{atol} & & \text{for } i = 37, \ldots, 49, \\
\text{rtol}(i) &= \text{rtol} & & \text{for } i = 1, \ldots, 49.
\end{align*}
\]

The reference solution was computed by PSIDE with \( \text{rtol} = \text{atol} = 10^{-14} \). For the work-precision diagrams, we used: \( \text{rtol} = 10^{-(4+m/4)}, m = 0, 1, \ldots, 24; \text{atol} = \text{rtol}; h_0 = \text{rtol} \) for BIMD, GAMD, MEBDFDAE, MEBDFI, RADAU and RADAU5.

The failed runs are in Table II.20.6; listed are the name of the solver that failed, for which values of \( m \) this happened, and the reason for failing.

References


Table II.20.4: Reference solution at the end of the integration interval.

\[
\begin{array}{cccc}
  y_1 & 0.229848289674730 \cdot 10^{-002} & y_26 & 0.47519404352918529 \cdot 10^{-001} \\
  y_2 & 0.188984650746585 \cdot 10^{-002} & y_27 & 0.47519404352918529 \cdot 10^{-001} \\
  y_3 & 0.1109503645730845 \cdot 10^{-002} & y_28 & 0.47519404352918529 \cdot 10^{-001} \\
  y_4 & 0.158620100314825 \cdot 10^{-003} & y_29 & 0.47519404352918529 \cdot 10^{-001} \\
  y_5 & 0.103002640715102 \cdot 10^{-002} & y_30 & 0.424921433601160 \cdot 10^{-001} \\
  y_6 & 0.8710606306836165 \cdot 10^{-003} & y_31 & 0.473236439609648 \cdot 10^{-001} \\
  y_7 & 0.3243571482093489 \cdot 10^{-002} & y_32 & 0.473236439609648 \cdot 10^{-001} \\
  y_8 & 0.1109503645730845 \cdot 10^{-002} & y_33 & 0.427000218868241 \cdot 10^{-001} \\
  y_9 & 0.712086206521341 \cdot 10^{-003} & y_34 & 0.47519404352918529 \cdot 10^{-001} \\
 y_{10} & 0.6414613953833099 \cdot 10^{-003} & y_35 & 0.47519404352918529 \cdot 10^{-001} \\
 y_{11} & 0.941697549524347 \cdot 10^{-003} & y_36 & 0.365142702667565 \cdot 10^{-001} \\
 y_{12} & 0.340342851906511 \cdot 10^{-002} & y_37 & 0.11111368591478108 \cdot 10^{-006} \\
 y_{13} & 0.239769310739305 \cdot 10^{-002} & y_38 & 0.11111270045592387 \cdot 10^{-006} \\
 y_{14} & 0.239769310739305 \cdot 10^{-002} & y_39 & 0.11111270178730254 \cdot 10^{-006} \\
 y_{15} & 0.334584143041580 \cdot 10^{-002} & y_40 & 0.11111369851929858 \cdot 10^{-006} \\
 y_{16} & 0.135350017035444 \cdot 10^{-002} & y_{41} & 0.11111369255355337 \cdot 10^{-006} \\
 y_{17} & 0.19950214318736 \cdot 10^{-002} & y_{42} & 0.1111136932658045 \cdot 10^{-006} \\
 y_{18} & 0.574622074193575 \cdot 10^{-002} & y_{43} & 0.1111136921703983 \cdot 10^{-006} \\
 y_{19} & 0.4751940452918529 \cdot 10^{-001} & y_{44} & 0.1111127012140091 \cdot 10^{-006} \\
 y_{20} & 0.4751940452918529 \cdot 10^{-001} & y_{45} & 0.1111127441515807 \cdot 10^{-006} \\
 y_{21} & 0.4751940452918529 \cdot 10^{-001} & y_{46} & 0.11111255158881087 \cdot 10^{-006} \\
 y_{22} & 0.4751940452918529 \cdot 10^{-001} & y_{47} & 0.1111127879394227 \cdot 10^{-006} \\
 y_{23} & 0.4751940452918529 \cdot 10^{-001} & y_{48} & 0.11111270969517164 \cdot 10^{-006} \\
 y_{24} & 0.4751940452918529 \cdot 10^{-001} & y_{49} & 0.1111129833871779 \cdot 10^{-006} \\
 y_{25} & 0.4311196778792902 \cdot 10^{-001} & & & \\
\end{array}
\]

Table II.20.5: Run characteristics.

<table>
<thead>
<tr>
<th>solver</th>
<th>rtol</th>
<th>atol</th>
<th>h0</th>
<th>mesed</th>
<th>sed</th>
<th>steps</th>
<th>accept</th>
<th>#f</th>
<th>#Jac</th>
<th>#LU</th>
<th>CPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMD</td>
<td>10^{-4}</td>
<td>10^{-4}</td>
<td>10^{-4}</td>
<td>3.55</td>
<td>1.23</td>
<td>17</td>
<td>16</td>
<td>250</td>
<td>16</td>
<td>17</td>
<td>0.0420</td>
</tr>
<tr>
<td>GAMD</td>
<td>10^{-7}</td>
<td>10^{-7}</td>
<td>10^{-7}</td>
<td>6.05</td>
<td>3.45</td>
<td>333</td>
<td>314</td>
<td>5830</td>
<td>314</td>
<td>333</td>
<td>0.8989</td>
</tr>
<tr>
<td></td>
<td>10^{-10}</td>
<td>10^{-10}</td>
<td>10^{-10}</td>
<td>9.22</td>
<td>7.32</td>
<td>673</td>
<td>586</td>
<td>17454</td>
<td>586</td>
<td>673</td>
<td>2.1101</td>
</tr>
<tr>
<td>MEBDFI</td>
<td>10^{-4}</td>
<td>10^{-4}</td>
<td>10^{-4}</td>
<td>3.51</td>
<td>1.18</td>
<td>18</td>
<td>16</td>
<td>340</td>
<td>16</td>
<td>18</td>
<td>0.0439</td>
</tr>
<tr>
<td></td>
<td>10^{-7}</td>
<td>10^{-7}</td>
<td>10^{-7}</td>
<td>5.94</td>
<td>3.40</td>
<td>233</td>
<td>202</td>
<td>8038</td>
<td>204</td>
<td>233</td>
<td>0.7642</td>
</tr>
<tr>
<td></td>
<td>10^{-10}</td>
<td>10^{-10}</td>
<td>10^{-10}</td>
<td>9.32</td>
<td>7.18</td>
<td>554</td>
<td>458</td>
<td>22918</td>
<td>448</td>
<td>536</td>
<td>1.9744</td>
</tr>
<tr>
<td>PSIDE-1</td>
<td>10^{-4}</td>
<td>10^{-4}</td>
<td>10^{-4}</td>
<td>3.85</td>
<td>1.83</td>
<td>81</td>
<td>77</td>
<td>1197</td>
<td>18</td>
<td>18</td>
<td>0.0488</td>
</tr>
<tr>
<td></td>
<td>10^{-7}</td>
<td>10^{-7}</td>
<td>10^{-7}</td>
<td>6.32</td>
<td>3.30</td>
<td>1267</td>
<td>1171</td>
<td>13926</td>
<td>192</td>
<td>192</td>
<td>0.5846</td>
</tr>
<tr>
<td></td>
<td>10^{-10}</td>
<td>10^{-10}</td>
<td>10^{-10}</td>
<td>9.09</td>
<td>7.18</td>
<td>3189</td>
<td>3037</td>
<td>28403</td>
<td>351</td>
<td>351</td>
<td>1.2561</td>
</tr>
<tr>
<td></td>
<td>10^{-4}</td>
<td>10^{-4}</td>
<td>10^{-4}</td>
<td>4.37</td>
<td>2.45</td>
<td>64</td>
<td>50</td>
<td>799</td>
<td>16</td>
<td>244</td>
<td>0.1015</td>
</tr>
<tr>
<td></td>
<td>10^{-7}</td>
<td>10^{-7}</td>
<td>10^{-7}</td>
<td>5.80</td>
<td>3.09</td>
<td>134</td>
<td>104</td>
<td>2320</td>
<td>40</td>
<td>468</td>
<td>0.2723</td>
</tr>
<tr>
<td></td>
<td>10^{-10}</td>
<td>10^{-10}</td>
<td>10^{-10}</td>
<td>7.86</td>
<td>5.45</td>
<td>827</td>
<td>719</td>
<td>14105</td>
<td>39</td>
<td>1292</td>
<td>1.2102</td>
</tr>
</tbody>
</table>
Figure II.20.6: Behavior of flows over the integration interval.
Figure II.20.7: Behavior of resistance coefficients over the integration interval.
Figure II.20.8: Behavior of pressures over the integration interval.

Table II.20.6: Failed runs.

<table>
<thead>
<tr>
<th>solver</th>
<th>( m )</th>
<th>reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>RADAU</td>
<td>0, 6, 8, 9, 11, 12, 13, 14, 16, \ldots, 20, 24</td>
<td>solver cannot handle ( \text{IER}=1 ).</td>
</tr>
<tr>
<td>RADAU5</td>
<td>6</td>
<td>stepsize too small</td>
</tr>
</tbody>
</table>
Figure II.20.9: Work-precision diagram (scd versus CPU-time).
Figure II.20.10: Work-precision diagram (soc versus CPU-time).
Figure II.20.11: Work-precision diagram (mesed versus CPU-time).
Figure II.20.12: Work-precision diagram (mesed versus CPU-time).
21 NAND gate

21.1 General information

The problem is a system of 14 stiff IDEs of index 1. It has been contributed by Michael Günther and Peter Rentrop [GR96].

The software part of the problem is in the file nan.d available at [MM08].

21.2 Mathematical description of the problem

The problem is of the form:

\[ C(y(t)) \frac{dy}{dt} = f(t, y(t)), \quad y(0) = y_0, \quad y'(0) = y'_0 \]  

with

\[ y \in \mathbb{R}^{14}, \quad 0 \leq t \leq 80. \]

The equations are given by:

\[ C_{GS} \cdot \dot{y}_5 - \dot{y}_1 = \frac{i_{DS}^D(y_2 - y_1, y_5 - y_1, y_3 - y_5, y_5 - y_2, y_4 - V_{DD}) + \frac{y_1 - y_5}{R_{GS}}}{y_{14}} \]  

(II.21.2)

\[ C_{GD} \cdot \dot{y}_5 - \dot{y}_2 = -i_{DS}^D(y_2 - y_1, y_5 - y_1, y_3 - y_5, y_5 - y_2, y_4 - V_{DD}) + \frac{y_2 - V_{DD}}{R_{GD}}, \]  

(II.21.3)

\[ C_{BS}(y_3 - y_5) \cdot (\dot{y}_5 - \dot{y}_3) = \frac{y_3 - V_{BB}}{R_{BS}} - i_{BS}^D(y_3 - y_5), \]  

(II.21.4)

\[ C_{BD}(y_4 - V_{DD}) \cdot (\dot{y}_5 - \dot{y}_4) = \frac{y_4 - V_{BB}}{R_{BD}} - i_{BD}^D(y_4 - V_{DD}), \]  

(II.21.5)

\[ C_{GS} \cdot \dot{y}_9 + C_{GD} \cdot \dot{y}_2 + C_{BS}(y_3 - y_5) \cdot (\dot{y}_5 - \dot{y}_3) - (C_{GS} + C_{GD} + C_{BS}(y_3 - y_5) + C_5) \cdot \dot{y}_5 \]  

- \[ C_{BD}(y_9 - y_5) \cdot (\dot{y}_5 - \dot{y}_9) = \frac{y_9 - y_5}{R_{GS}} + i_{BS}^D(y_3 - y_5) + \frac{y_9 - y_{10}}{R_{GD}} + i_{BD}^D(y_9 - y_5), \]  

(II.21.6)

\[ C_{GS} \cdot \dot{y}_6 = -i_{DS}^E(y_7 - y_6, V_1(t) - y_6, y_8 - y_{10}, V_1(t) - y_7, y_6 - y_5) + C_{GS} \cdot \dot{V}_1(t) - \frac{y_6 - y_{10}}{R_{GS}}, \]  

(II.21.7)

\[ C_{GD} \cdot \dot{y}_7 = i_{DS}^E(y_7 - y_6, V_1(t) - y_6, y_8 - y_{10}, V_1(t) - y_7, y_6 - y_5) + C_{GD} \cdot \dot{V}_1(t) - \frac{y_7 - y_8}{R_{GD}}, \]  

(II.21.8)

\[ C_{BS}(y_8 - y_{10}) \cdot (\dot{y}_8 - \dot{y}_{10}) = \frac{y_8 - V_{BB}}{R_{BS}} + i_{BS}^E(y_8 - y_{10}), \]  

(II.21.9)

\[ C_{BD}(y_9 - y_5) \cdot (\dot{y}_9 - \dot{y}_5) = \frac{y_9 - V_{BB}}{R_{BD}} + i_{BD}^E(y_9 - y_5), \]  

(II.21.10)

\[ C_{BS}(y_8 - y_{10}) \cdot (\dot{y}_8 - \dot{y}_{10}) - C_{BD}(y_{14} - y_{10}) \cdot (\dot{y}_{10} - \dot{y}_{14}) + C_{10} \cdot \dot{y}_{10} \]  

= \frac{y_8 - y_{10}}{R_{GS}} + i_{BS}^E(y_8 - y_{10}) + \frac{y_8 - y_{12}}{R_{GD}} + i_{BD}^E(y_{14} - y_{10}), \]  

(II.21.11)

\[ C_{GS} \cdot \dot{y}_{11} = -i_{DS}^E(y_{12} - y_{11}, V_2(t) - y_{11}, y_{13}, V_2(t) - y_{12}, y_{14} - y_{10}) + C_{GS} \cdot \dot{V}_2(t) - \frac{y_{11}}{R_{GS}}, \]  

(II.21.12)
\[ C_{GD} \cdot y_{12} = i_{DS}^E(y_{12} - y_{11}, V_2(t) - y_{11}, y_{13}, V_2(t) - y_{12}, y_{14} - y_{10}) + C_{GD} \cdot V_2(t) - \frac{y_{12} - y_{10}}{R_{GD}}, \quad (II.21.13) \]

\[ C_{BS}(y_{13}) \cdot y_{13} = -\frac{y_{13} - V_{BB}}{R_{BS}} + i_{BS}^E(y_{13}), \quad (II.21.14) \]

\[ C_{BD}(y_{14} - y_{10}) \cdot (y_{14} - y_{10}) = -\frac{y_{14} - V_{BB}}{R_{BS}} + i_{BD}^E(y_{14} - y_{10}). \quad (II.21.15) \]

The functions \( C_{BD} \) and \( C_{BS} \) read

\[ C_{BD}(U) = C_{BS}(U) = \begin{cases} C_0 \cdot \left(1 - \frac{V}{\phi_B}\right)^{\frac{-2}{\phi_B}} & \text{for } U \leq 0, \\ C_0 \cdot \left(1 + \frac{V}{\phi_B}\right)^{\frac{-2}{\phi_B}} & \text{for } U > 0 \end{cases} \]

with \( C_0 = 0.24 \cdot 10^{-4} \) and \( \phi_B = 0.87 \).

The functions \( i_{BS}^D \) and \( i_{BS}^E \) have the same form denoted by \( i_{BS} \). The only difference between them is that the constants used in \( i_{BS} \) depend on the superscript \( D \) and \( E \). The same holds for the functions \( i_{BD}^D \) and \( i_{BD}^E \). The functions \( i_{BS}, i_{BD} \) and \( i_{DS} \) are defined by

\[ i_{BS}(U_{BS}) = \begin{cases} -i_s \cdot \left(\exp\left(\frac{U_{BS}}{kT}\right) - 1\right) & \text{for } U_{BS} \leq 0, \\ 0 & \text{for } U_{BS} > 0 \end{cases} \]

\[ i_{BD}(U_{BD}) = \begin{cases} -i_s \cdot \left(\exp\left(\frac{U_{BD}}{kT}\right) - 1\right) & \text{for } U_{BD} \leq 0, \\ 0 & \text{for } U_{BD} > 0 \end{cases} \]

\[ i_{DS}(U_{DS}, U_{GS}, U_{BS}, U_{GD}, U_{BD}) = \begin{cases} GDS_+(U_{DS}, U_{GS}, U_{BS}) & \text{for } U_{DS} > 0, \\ 0 & \text{for } U_{DS} = 0, \\ GDS_-(U_{DS}, U_{GD}, U_{BD}) & \text{for } U_{DS} < 0, \end{cases} \]

where

\[ GDS_+(U_{DS}, U_{GS}, U_{BS}) = \]

\[ \begin{cases} 0 & \text{for } U_{GS} - U_{TE} \leq 0, \\ -\beta \cdot (1 + \delta \cdot U_{DS}) \cdot (U_{GS} - U_{TE})^2 & \text{for } 0 < U_{GS} - U_{TE} \leq U_{DS}, \\ -\beta \cdot U_{DS} \cdot (1 + \delta \cdot U_{DS}) \cdot (2 \cdot (U_{GS} - U_{TE}) - U_{DS}) & \text{for } 0 < U_{DS} < U_{GS} - U_{TE}, \end{cases} \]

with

\[ U_{TE} = U_{T0} + \gamma \cdot \left(\sqrt{\Phi - U_{BS}} - \sqrt{\Phi}\right), \quad (II.21.16) \]

and

\[ GDS_-(U_{DS}, U_{GD}, U_{BD}) = \]

\[ \begin{cases} 0 & \text{for } U_{GD} - U_{TE} \leq 0, \\ \beta \cdot (1 - \delta \cdot U_{DS}) \cdot (U_{GD} - U_{TE})^2 & \text{for } 0 < U_{GD} - U_{TE} \leq -U_{DS}, \\ -\beta \cdot U_{DS} \cdot (1 - \delta \cdot U_{DS}) \cdot (2 \cdot (U_{GD} - U_{TE}) + U_{DS}) & \text{for } 0 < -U_{DS} < U_{GD} - U_{TE}, \end{cases} \]

with

\[ U_{TE} = U_{T0} + \gamma \cdot \left(\sqrt{\Phi - U_{BD}} - \sqrt{\Phi}\right). \quad (II.21.17) \]

The constants used in the definition of \( i_{BS}, i_{BD} \) and \( i_{DS} \) carry a superscript \( D \) or \( E \). Using for example the constants with superscript \( E \) in the functions \( i_{BS} \) yields the function \( i_{BS}^E \). These constants are shown in Table II.21.1. The other constants are given by
Table II.21.1: Dependence of constants on $D$ and $E$ for $i_{BS}$, $i_{BD}$ and $i_{DS}$.

<table>
<thead>
<tr>
<th></th>
<th>$E$</th>
<th>$D$</th>
<th>$E$</th>
<th>$D$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$i_s$</td>
<td>$10^{-14}$</td>
<td>$10^{-14}$</td>
<td>$\beta$</td>
<td>$1.748 \cdot 10^{-3}$</td>
</tr>
<tr>
<td>$U_T$</td>
<td>25.85</td>
<td>25.85</td>
<td>$\gamma$</td>
<td>0.035</td>
</tr>
<tr>
<td>$U_{T0}$</td>
<td>0.2</td>
<td>-2.43</td>
<td>$\delta$</td>
<td>0.02</td>
</tr>
<tr>
<td>$\phi$</td>
<td>1.01</td>
<td>1.28</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\[
V_{BB} = -2.5, \\
V_{DD} = 5, \\
C_0 = C_{10} = 0.5 \cdot 10^{-4}, \\
R_{GS} = R_{GD} = 4, \\
R_{BS} = R_{BD} = 10, \\
C_{GS} = C_{GD} = 0.6 \cdot 10^{-4}.
\]

The functions $V_1(t)$ and $V_2(t)$ are

\[
V_1(t) = \begin{cases} 
20 - tm & \text{if } 15 < tm \leq 20, \\
5 & \text{if } 10 < tm \leq 15, \\
(tm - 5) & \text{if } 5 < tm \leq 10, \\
0 & \text{if } tm \leq 5,
\end{cases}
\]

with $tm = t \mod 20$ and

\[
V_2(t) = \begin{cases} 
40 - tm & \text{if } 35 < tm \leq 40, \\
5 & \text{if } 20 < tm \leq 35, \\
(tm - 15) & \text{if } 15 < tm \leq 20, \\
0 & \text{if } tm \leq 15,
\end{cases}
\]

with $tm = t \mod 40$. From these definitions for $V_1(t)$ and $V_2(t)$ we see that the function $f$ in \((\text{II.21.1})\) has discontinuities in its derivative at $tm = 5, 10, 15, 20$. Therefore, we restart the solvers at $t = 5, 10, \ldots, 75$.

Consistent initial values are given by $y_0^t = 0$ and

\[
y_1 = y_2 = y_5 = y_7 = 5.0, \\
y_3 = y_4 = y_6 = y_9 = y_{10} = y_{13} = y_{14} = V_{BB} = -2.5, \\
y_6 = y_{10} = y_{12} = 3.62385, \\
y_{11} = 0.
\]

All components of $y$ are of index 1.

It is clear from Formulas \((\text{II.21.16})\) and \((\text{II.21.17})\) that the function $f$ can not be evaluated if one of the values $\Phi - U_{BS}$, $\Phi - U_{BD}$ or $\Phi$ becomes negative. To prevent this situation, we set $\text{IERR}=-1$ in the Fortran subroutine that defines $f$ if this happens. See page IV-ix of the the description of the software part of the test set for more details on $\text{IERR}$.

### 21.3 Origin of the problem

The NAND gate in Figure II.21.1 consists of two $n$-channel enhancement MOSFETs (ME), one $n$-channel depletion MOSFET (MD) and two load capacitances $C_0$ and $C_{10}$. MOSFETs are special transistors, which have four terminals: the drain, the bulk, the source and the gate, see also Figure II.21.3. The drain voltage of MD is constant at $V_{DD} = 5[V]$. The bulk voltages are constantly $V_{BB} = -2.5[V]$. The gate voltages of both enhancement transistors are controlled by two voltage...
Figure II.21.1: Circuit diagram of the NAND gate (taken from [GR96])

<table>
<thead>
<tr>
<th>V1</th>
<th>LOW</th>
<th>HIGH</th>
<th>HIGH</th>
<th>LOW</th>
</tr>
</thead>
<tbody>
<tr>
<td>V2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LOW</th>
<th>HIGH</th>
<th>HIGH</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

sources $V_1$ and $V_2$. Depending on the input voltages, the NAND gate generates a response at node 5 as shown in Figure II.21.2. If we represent the logical values 1 and 0 by high respectively low voltage levels, we see that the NAND gate executes the Not AND operation. This behavior can be explained from Figure II.21.1 as follows. Roughly speaking, a transistor acts as a switch between drain and source; it closes if the voltage between gate and source drops below a certain threshold value. The circuit is constructed such that the voltage at node 10 drops to zero unless $V_1$ is high and $V_2$ is low, in which case it is approximately $5[V]$. This means that as soon either $V_1$ or $V_2$ is low, then the corresponding enhancement transistors lock; the voltage at node 5 is high at $V_{DD} = 5[V]$ due to MD.
If both $V_1$ and $V_2$ exceed a given threshold voltage, then a drain current through both enhancement transistors occurs. The MOSFETs open and the voltage at node 5 breaks down. The response is low.

In the circuit analysis the three MOSFETs are replaced by the circuit shown in Figure II.21.3. Here, the well-known companion model of Shichmann and Hodges [SH68] is used. The characteristics of the circuit elements can differ depending on the MD or ME case. This circuit has four internal nodes indicated by 1, 2, 3 and 4. The static behavior of the transistor is described by the drain current $i_{DS}$. To include secondary effects, load capacitances like $R_{GS}$, $R_{GD}$, $R_{BS}$, and $R_{BD}$ are introduced. The so-called $pn$-junction between source and bulk is modeled by the diode $i_{BS}$ and the non-linear capacitance $C_{BS}$. Analogously, $i_{BD}$ and $C_{BD}$ model the $pn$-junction between bulk and drain. Linear gate capacitances $C_{GS}$ and $C_{GD}$ are used to describe the intrinsic charge flow effects roughly.

To formulate the circuit equations, we note that the circuit consists of 14 nodes. These 14 nodes are the nodes 5 and 10 and the 12 internal nodes of the three transistors. For every node a variable is introduced that represents the voltage in that node. Table II.21.2 shows the variable–node correspondence. In terms of these voltages the circuit equations are formulated by using the Kirchoff Current Law (KCL) along with the transistor model shown in Figure II.21.3. In Figure II.21.4, we check the behavior of the NAND gate by plotting $V_1$ and $V_2$ together with the numerical value for the voltage at node 5, which is obtained as $y_{10}$ in §21.4. The picture confirms that the NAND gate produces a high signal in the intervals $[0, 5]$, $[10, 15]$, $[20, 25]$, $[40, 45]$, $[50, 55]$ and $[60, 65]$, whereas the output signal
21.4 Numerical solution of the problem

Tables II.21.3–II.21.4 and Figures II.21.5–II.21.7 present the reference solution at the end of the integration interval, the run characteristics, the behavior of the solution over the integration interval and the work-precision diagram, respectively. In computing the scd values, only \( y_5 \), the response of the gate at node 5, was considered. The reference solution was computed on the Cray C90, using PSIDE with Cray double precision and \( \text{atol} = \text{rtol} = 10^{-16} \). For the work-precision diagram, we used:

\[
\text{rtol} = 10^{-\left(4+m/8\right)}, \quad m = 0, 1, \ldots, 64; \quad \text{atol} = \text{rtol}, \quad h_0 = \text{rtol} \text{ for MEBDFI.}
\]
Table II.21.3: Reference solution at the end of the integration interval.

<table>
<thead>
<tr>
<th>y1</th>
<th>0.4971088699385777 \cdot 10^4</th>
<th>y8</th>
<th>-0.250077409198803 \cdot 10^4</th>
</tr>
</thead>
<tbody>
<tr>
<td>y2</td>
<td>0.4999752103929311 \cdot 10^4</td>
<td>y9</td>
<td>-0.249998781491227 \cdot 10^4</td>
</tr>
<tr>
<td>y3</td>
<td>-0.2499998781491227 \cdot 10^4</td>
<td>y10</td>
<td>-0.2400289583878100</td>
</tr>
<tr>
<td>y4</td>
<td>-0.2499999999999991 \cdot 10^4</td>
<td>y11</td>
<td>-0.2399099999996269 \cdot 10^{-3}</td>
</tr>
<tr>
<td>y5</td>
<td>0.4970837023296724 \cdot 10^4</td>
<td>y12</td>
<td>-0.2491214032073855</td>
</tr>
<tr>
<td>y6</td>
<td>-0.2091214032073855</td>
<td>y13</td>
<td>-0.2499999999999991 \cdot 10^4</td>
</tr>
<tr>
<td>y7</td>
<td>0.4970593243278363 \cdot 10^4</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table II.21.4: Run characteristics.

<table>
<thead>
<tr>
<th>solver</th>
<th>rtol</th>
<th>atol</th>
<th>h0</th>
<th>mescd</th>
<th>sde</th>
<th>steps</th>
<th>accept</th>
<th>#f</th>
<th>#Jac</th>
<th>#LU</th>
<th>CPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>DDASSL</td>
<td>10^{-1}</td>
<td>10^{-4}</td>
<td></td>
<td>3.69</td>
<td>5.25</td>
<td>1037</td>
<td>951</td>
<td>1639</td>
<td>246</td>
<td>0.0459</td>
<td></td>
</tr>
<tr>
<td></td>
<td>10^{-7}</td>
<td>10^{-4}</td>
<td></td>
<td>6.22</td>
<td>8.81</td>
<td>3825</td>
<td>3604</td>
<td>5207</td>
<td>638</td>
<td>0.1376</td>
<td></td>
</tr>
<tr>
<td>MEBDFI</td>
<td>10^{-4}</td>
<td>10^{-4}</td>
<td>10^{-4}</td>
<td>3.76</td>
<td>4.57</td>
<td>1120</td>
<td>1006</td>
<td>7693</td>
<td>249</td>
<td>249</td>
<td>0.0683</td>
</tr>
<tr>
<td></td>
<td>10^{-7}</td>
<td>10^{-7}</td>
<td>10^{-7}</td>
<td>6.24</td>
<td>7.50</td>
<td>3786</td>
<td>3429</td>
<td>24487</td>
<td>755</td>
<td>755</td>
<td>0.2255</td>
</tr>
<tr>
<td>PSIDE-1</td>
<td>10^{-4}</td>
<td>10^{-4}</td>
<td></td>
<td>2.39</td>
<td>3.33</td>
<td>464</td>
<td>411</td>
<td>6574</td>
<td>109</td>
<td>1796</td>
<td>0.0927</td>
</tr>
<tr>
<td></td>
<td>10^{-7}</td>
<td>10^{-4}</td>
<td></td>
<td>5.28</td>
<td>8.48</td>
<td>773</td>
<td>643</td>
<td>13134</td>
<td>222</td>
<td>2760</td>
<td>0.1796</td>
</tr>
</tbody>
</table>

References


Figure II.21.5: Behavior of the solution over the integration interval.
Figure II.21.6: Work-precision diagram (sod versus CPU-time).
Figure II.21.7: Work-precision diagram (nsecd versus CPU-time).
22 Wheelset

22.1 General Information

The wheelset is an IDE of dimension 17 which shows some typical properties of simulation problems in contact mechanics, i.e., friction, contact conditions, stiffness, etc. This problem is originally described by an index 3 IDE with additional index 1 equations, but can be reduced to index 2. Test results are based on the index-2 formulation. This problem was contributed by Bernd Simeon, Claus Führer, Peter Rentrop, Nov. 1995. Comments to simeon@na.tum.de or Claus.Fuehrer@na.lu.se. See also [SFR91].

The software part of the problem is in the file wheel.f available at [MM08].

22.2 Mathematical description of the problem

The index 3 formulation of the wheelset problem reads

\[
\begin{align*}
\dot{p} &= v, \\
M(p) \begin{pmatrix} \dot{v} \\ \dot{\beta} \end{pmatrix} &= \begin{pmatrix} f(u) - (\partial g_1(p,q)/\partial p)^T C \lambda \\ d(u) \end{pmatrix}, \\
0 &= g_1(p,q), \\
0 &= g_2(p,q),
\end{align*}
\]  

where \( u := (p,v,\beta,q,\lambda)^T \in \mathbb{R}^{17}, v, \beta \in \mathbb{R}, q \in \mathbb{R}^4, \lambda \in \mathbb{R}^2 \) and \( C \) is a scalar constant. Furthermore, \( M: \mathbb{R}^6 \to \mathbb{R}^6 \times \mathbb{R}^6, \beta: \mathbb{R}^{17} \to \mathbb{R}^6, d: \mathbb{R}^{17} \to \mathbb{R}, g_1: \mathbb{R}^6 \to \mathbb{R}^2 \) and \( g_2: \mathbb{R}^6 \to \mathbb{R}^4 \). The integration interval is from 0 to 10 [s].

For the index 2 formulation of the problem (II.22.3) is replaced by

\[
0 = (\partial g_1(p,q)/\partial p) v.
\]

The non-zero components of the consistent initial values \( u(0) := u_0 \) and \( v(0) := u_0' \) are given by

| \( u_{0,1} \) | 0.1494100000000000 10^{-2} |
| \( u_{0,2} \) | 0.4008900000000000 10^{-6} |
| \( u_{0,3} \) | 0.1124100000000000 10^{-5} |
| \( u_{0,4} \) | -0.2857300000000000 10^{-3} |
| \( u_{0,5} \) | 0.2643900000000000 10^{-3} |
| \( u_{0,6} \) | -1.9752588940112850 |
| \( u_{0,7} \) | -1.0898971028112760 10^{-3} |
| \( u_{0,8} \) | 7.8855083626142858 10^{-2} |

| \( u_{0,9} \) | 7.4122380357667139 10^{-6} |
| \( u_{0,10} \) | 0.1521364296121248 |
| \( u_{0,11} \) | 7.563406395172940 10^{-6} |
| \( u_{0,12} \) | 0.1490635714733819 |
| \( u_{0,13} \) | -0.8339300000000000 10^{-2} |
| \( u_{0,14} \) | -0.7414400000000000 10^{-2} |
| \( u_{0,15} \) | 0.1490635714733819 |
| \( u_{0,16} \) | -0.8339300000000000 10^{-2} |
| \( u_{0,17} \) | -0.7414400000000000 10^{-2} |

The other components of \( u_0 \) and \( u_0' \) are zero. For the index 3 formulation, the index of variables \( p, v, \beta, q \) and \( \lambda \) equals 1, 2, 2, 1 and 3. For the index 2 problem, these numbers read 1, 1, 1, 1 and 2.

The equations are given in detail in the next subsections, in which some references to the origin of the problem, treated in §22.3, are already given. Table II.22.1 lists all problem parameters.

22.2.1 Differential equations

The position coordinates \( p \) are defined as

\[
p := \begin{pmatrix} x \\
y \\
z \\
\theta \\
\phi \end{pmatrix}
\]

lateral displacement
vertical displacement
longitudinal displacement
yaw angle
roll angle
and the contact variables as $q^T = (\psi_L \xi_L \psi_R \xi_R)$ with

$$\xi_{LR} \quad \text{coordinate of the contact point left/right,}$$

$$\psi_{LR} \quad \text{shift angle left/right.}$$

The first three equations in (II.22.2) yield the momentum equations:

$$m_R \ddot{x} = m_R \left( 2 v_0 \kappa \cos \alpha \dot{z} + v_0^2 \kappa \cos \alpha (1 + \kappa (x \cos \alpha - y \sin \alpha)) \right) + T_{L_1} + T_{R_1} + Q_1 - m_R \dot{y} \sin \alpha - b_{1,1} \lambda_1 - b_{1,2} \lambda_2 - 2 c_x x,$$

$$m_R \ddot{y} = -m_R \left( 2 v_0 \kappa \sin \alpha \dot{z} + v_0^2 \kappa \sin \alpha (1 + \kappa (x \cos \alpha - y \sin \alpha)) \right) + T_{L_2} + T_{R_2} + Q_2 - m_R \dot{y} \cos \alpha - b_{2,1} \lambda_1 - b_{2,2} \lambda_2,$$

$$m_R \ddot{z} = m_R \left( -2 v_0 \kappa (x \cos \alpha - y \sin \alpha) + v_0^2 \kappa^2 \dot{z} \right) + T_{L_3} + T_{R_3} + Q_3 + F_A - b_{3,1} \lambda_1 - b_{3,2} \lambda_2,$$

where $b_{i,j}$ denotes the $(i, j)$ element of the constraint Jacobian $\partial g_i(p, q)/\partial p$. The next three equations yield the spin equations:

$$I_2 \dot{\theta} \cos \varphi = -\dot{\varphi} \sin \varphi + v_0 \kappa \left( \varphi (\sin \alpha \cos \theta \cos \varphi + \cos \alpha \sin \varphi) - \dot{\theta} \sin \alpha \sin \theta \sin \varphi \right) - I_1 (\omega_0 + \beta) (\dot{\varphi} - v_0 \kappa \sin \theta \sin \alpha) - (I_1 - I_2) \left( \dot{\theta} \sin \varphi - v_0 \kappa (\cos \theta \cos \varphi \sin \alpha + \sin \varphi \cos \alpha) \right)$$

$$+ \left[ - (\xi_L \sin \theta + R(\xi_L) \sin \psi_L \cos \theta \cos \varphi) T_{L_1} - R(\xi_L) \sin \psi_L \sin \varphi T_{L_2} + (\xi_L \cos \theta + R(\xi_L) \sin \psi_L \sin \theta \cos \varphi) T_{L_3} \right]$$

$$+ \left[ \text{corresponding terms of the right side} \right] - \cos \theta \sin \varphi M_1 + \cos \varphi M_2 + \sin \theta \sin \varphi M_3 - b_{4,1} \lambda_1 - b_{4,2} \lambda_2,$$

$$I_2 \dot{\varphi} = I_2 \theta v_0 \kappa \sin \alpha \cos \theta$$

$$+ I_1 (\omega_0 + \beta) \left( \dot{\theta} \cos \varphi + v_0 \kappa (\cos \theta \sin \varphi \sin \alpha - \cos \varphi \cos \alpha) \right) + (I_1 - I_2) \left( \dot{\theta} \sin \varphi - v_0 \kappa (\cos \theta \cos \varphi \sin \alpha + \sin \varphi \cos \alpha) \right)$$

$$+ \left[ - (\xi_L \cos \theta \sin \varphi - R(\xi_L) \cos \psi_L \cos \theta \cos \varphi) T_{L_1} + (\xi_L \cos \varphi + R(\xi_L) \cos \psi_L \sin \varphi) T_{L_2} + (\xi_L \sin \theta \sin \varphi - R(\xi_L) \cos \psi_L \sin \theta \cos \varphi) T_{L_3} \right]$$

$$+ \left[ \text{corresponding terms of the right side} \right] + \sin \theta M_1 + \cos \theta M_3 - b_{5,1} \lambda_1 - b_{5,2} \lambda_2.$$
\[ I_1 (\dot{\beta} + \dot{\theta} \sin \varphi) = \theta \dot{\phi} \cos \varphi - v_0 \kappa (\cos \alpha \cos \varphi \sin \alpha \cos \theta \sin \varphi - \dot{\theta} \sin \alpha \cos \varphi) \]

\[ + \left[ -R(\xi_L) (\cos \psi_L \sin \theta + \sin \psi_L \cos \theta \sin \varphi) T_L, \right. \]

\[ +R(\xi_L) \sin \psi_L \cos \varphi T_{L_2} \]

\[ -R(\xi_L) (\cos \psi_L \cos \theta - \sin \psi_L \sin \theta \sin \varphi) T_{L_3} \left. \right] \]

\[ + \left[ \text{corresponding terms of the right side} \right] \]

\[ + \cos \theta \cos \varphi M_1 + \sin \varphi M_2 - \sin \theta \cos \varphi M_3 + L_A. \]

The forces \( Q \) and moments \( M \) of the wagon body satisfy the following equations:

\[ Q_1 = \frac{m_A g}{\cos \alpha} \left( \frac{v_0 \alpha}{g} - \tan \alpha \right) \quad \text{(lateral force)}, \]

\[ Q_2 = -m_A g \cos \alpha \left( \frac{v_0 \alpha}{g} \tan \alpha + 1 \right) \quad \text{(vertical force)}, \]

\[ Q_3 = -2c_z z \quad \text{(longitudinal force)}, \]

\[ M_1 = 0 \]

\[ M_2 = Q_3 x \quad \text{(yaw moment)}, \]

\[ M_3 = -h_A Q_1 \quad \text{(roll moment)}, \]

\[ 0 = \cos \theta M_1 - \sin \theta M_3 \quad \text{(no pitch moment)}. \]

The creep forces \( T_{L_1,2,3} \) and \( T_{R_1,2,3} \) of the left and right contact point are obtained via the transformation

\[
\begin{pmatrix}
T_{L_1|R} \\
T_{L_2|R} \\
T_{L_3|R}
\end{pmatrix} =
\begin{pmatrix}
\sin \theta & \cos \theta \cos \Delta_{L|R} & \cos \theta \sin \Delta_{L|R} \\
0 & \pm \sin \Delta_{L|R} & \cos \Delta_{L|R} \\
\cos \theta & -\sin \theta \cos \Delta_{L|R} & \pm \sin \theta \sin \Delta_{L|R}
\end{pmatrix}
\begin{pmatrix}
T_{1|L} \\
T_{2|L} \\
0
\end{pmatrix},
\]

where \( T_{1|L} \) and \( T_{2|L} \) denote the creep forces with respect to the local reference frame of the contact point and \( \pm \) stands for the left and right side, respectively. The creep forces are approximated by

\[ T_{1|L} := -\mu N_{L|R} \tan \left( \frac{Gc_{11}c^2}{\mu N_{L|R}} \nu_1 \right), \]

\[ T_{2|L} := -\mu N_{L|R} \tan \left( \frac{Gc_{22}c^2}{\mu N_{L|R}} \nu_2 + \frac{Gc_{23}c^3}{\mu N_{L|R}} \varphi_3 \right), \]

and corrected by

\[ \text{if} \quad T_1^2 + T_2^2 > (\mu N)^2, \quad \text{then} \quad \hat{T}_1 := \frac{T_1}{\sqrt{T_1^2 + T_2^2}} \mu N \quad \text{and} \quad \hat{T}_2 := \frac{T_2}{\sqrt{T_1^2 + T_2^2}} \mu N. \]

The constant parameters

\[ \mu, G, C_{11}, C_{22}, C_{23} \]

(friction coefficient, glide module, Kalker coefficients) are listed in Table II.22.1. For the computation of \( c \), the size of contact ellipse, which uses the parameters \( \sigma, \tilde{G} \) and \( c \), we refer to [Jas87]. For alternative creep force models see also [Jas87].

The normal forces \( N \) are given by

\[
\begin{pmatrix}
N_L \\
N_R
\end{pmatrix} = \gamma \begin{pmatrix}
\cos \Delta_R & -\sin \Delta_R \\
-\cos \Delta_L & -\sin \Delta_L
\end{pmatrix} \begin{pmatrix}
b_{1,1} & b_{1,2} \\
b_{2,1} & b_{2,2}
\end{pmatrix} \begin{pmatrix}
\lambda_1 \\
\lambda_2
\end{pmatrix},
\]

where \( \gamma \) is the load ratio.
where

\[ \gamma := \frac{1}{\sin \Delta_L \cos \Delta_R + \sin \Delta_R \cos \Delta_L}. \]

Here, \( \Delta_{L|R} \) denotes the contact angles and is defined as

\[
\tan \Delta_L = \frac{(R(\xi_L) \cos \varphi - \sin \varphi \cos \psi_L) \cos \theta + \sin \psi_L \sin \theta}{-R(\xi_L) \sin \varphi - \cos \psi_L \cos \varphi};
\]
\[
\tan \Delta_R = \frac{(R(\xi_R) \cos \varphi - \sin \varphi \cos \psi_R) \cos \theta + \sin \psi_R \sin \theta}{+R(\xi_R) \sin \varphi + \cos \psi_R \cos \varphi}.
\]

For the creepages we have the relations

\[
v_1 = \frac{1}{v_{rol}} (\sin \theta v_{r1} + \cos \theta v_{r3})
\]
\[
v_2 = \frac{1}{v_{rol}} (\cos \theta \cos \Delta_{L|R} v_{r1} \pm \sin \Delta_{L|R} v_{r2} - \sin \theta \cos \Delta_{L|R} v_{r3})
\]
\[
\varphi_3 = \frac{1}{v_{rol}} \left( \mp \sin \Delta_{L|R} (\omega + \beta - v_0 \kappa \sin \alpha) + \cos \Delta_{L|R} (\theta - v_0 \kappa \cos \alpha) \right)
\]

where \( v_{r1,2,3} \) (relative velocity at the contact point) and \( v_{rol} \) (rolling velocity) are given by (correspondingly for the right side)

\[
v_{r1} = \dot{x} - \dot{\theta}(R(\xi_L)(\sin \theta \sin \varphi \cos \psi_L + \cos \theta \sin \psi_L) + \xi_L \sin \theta \cos \varphi) - \varphi \cos \theta(\xi_L \sin \varphi - R(\xi_L) \cos \varphi \cos \psi_L) + (\omega_0 + \beta)R(\xi_L)(-\sin \theta \cos \varphi \sin \psi_L) + v_0 \kappa \cos \alpha(R(\xi_L)(\sin \theta \sin \varphi \cos \psi_L + \cos \theta \sin \psi_L) + \xi_L \sin \theta \cos \varphi - z),
\]
\[
v_{r2} = \dot{y} + \dot{\varphi}(\xi_L \cos \varphi + R(\xi_L) \sin \varphi \cos \psi_L) + (\omega_0 + \beta)R(\xi_L) \cos \varphi \sin \psi_L + v_0 \kappa \sin \alpha(z - \xi_L \sin \theta \cos \varphi - R(\xi_L)(\sin \theta \sin \varphi \cos \psi_L + \cos \theta \sin \psi_L)),
\]
\[
v_{r3} = \dot{z} + v_0 + v_0 \kappa(x \cos \alpha - y \sin \alpha) - \dot{\theta}(\xi_L \cos \theta \cos \varphi \sin \psi_L - \sin \theta \sin \psi_L) + \varphi \sin \theta(\xi_L \sin \varphi - R(\xi_L \cos \varphi \cos \psi_L) + (\omega + \beta)R(\xi_L)(\sin \theta \sin \varphi \sin \psi_L - \cos \theta \cos \psi_L) - v_0 \kappa \sin \alpha(\xi_L \sin \varphi - R(\xi_L) \cos \varphi \cos \psi_L) + v_0 \cos \alpha(\xi_L \cos \theta \cos \varphi + R(\xi_L)(\cos \theta \sin \varphi \cos \psi_L - \sin \theta \sin \psi_L)),
\]

and

\[
v_{rol} = \frac{1}{2} \left\| \begin{pmatrix} -2 \dot{x} + 2v_0 \kappa \cos \alpha \\ -2 \dot{y} - 2v_0 \kappa \sin \alpha \\ -2 \dot{z} - 2v_0 - 2v_0 \kappa(x \cos \alpha - y \sin \alpha) \end{pmatrix} \right\|_2.
\]

### 22.2.2 Constraints

The constraints (II.223) read

\[
\begin{pmatrix} G(\xi_L) - y - \xi_L \sin \varphi + R(\xi_L) \cos \varphi \cos \psi_L \\ G(\xi_R) - y - \xi_R \sin \varphi + R(\xi_R) \cos \varphi \cos \psi_R \end{pmatrix} = 0
\]
with profile functions $R$ (wheel) and $G$ (rail), see Figure II.22.1,

\[ R(\xi) = \rho_0 + \tan \delta_0 (a_0 - |\xi|) \quad \text{for } a_0 - \Delta a < |\xi| < b_0; \]

\[ G(\xi) = \sqrt{\rho_1^2 - (|\xi| - a_0 - \rho_1 \sin \delta_0)^2} - \rho_0 - \cos \delta_0 \rho_1 \quad \text{for } c_1 < |\xi| < c_2. \]

Here, $\xi$ stands for the left or right coordinate $\xi_{L/R}$, respectively, and $\hat{\xi}$ is defined by

\[ \hat{\xi}_{L/R} := x + \xi_{L/R} \cos \theta \cos \varphi + R(\xi_{L|R}) (\cos \theta \sin \varphi \cos \psi_{L|R} - \sin \theta \sin \psi_{L|R}). \]

The constraints (II.22.4) read

\[ G'(\hat{\xi}_L) (R'(\xi_L) \sin \varphi + \cos \varphi \cos \psi_L) + R'(\xi_L) \cos \theta \cos \varphi - \cos \theta \sin \varphi \cos \psi_L + \sin \theta \sin \psi_L = 0, \]

\[ R'(\xi_L) \sin \theta \cos \varphi - \sin \theta \sin \varphi \cos \psi_L - \cos \theta \sin \psi_L = 0, \]

\[ G'(\hat{\xi}_R) (R'(\xi_R) \sin \varphi + \cos \varphi \cos \psi_R) + R'(\xi_R) \cos \theta \cos \varphi - \cos \theta \sin \varphi \cos \psi_R + \sin \theta \sin \psi_R = 0, \]

\[ R'(\xi_R) \sin \theta \cos \varphi - \sin \theta \sin \varphi \cos \psi_R - \cos \theta \sin \psi_R = 0, \]

where \( G'(\hat{\xi}_{L|R}) := \frac{d}{d\xi_{L|R}} G(\hat{\xi}_{L|R}), R'(\xi_{L|R}) := \frac{d}{d\xi_{L|R}} R(\xi_{L|R}). \)

### 22.3 Origin of the problem

The motion of a simple wheelset on a rail track exhibits a lot of the difficulties which occur in the simulation of contact problems in mechanics. The state space form approach for this class of problems
Table II.22.1: Parameter values according to [Jas90], where a hardware blogic model, scaled 1:4, is investigated.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Meaning</th>
<th>Unit</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>(m_R)</td>
<td>mass wheelset</td>
<td>kg</td>
<td>16.08</td>
</tr>
<tr>
<td>(g)</td>
<td>gravity constant</td>
<td>m/s²</td>
<td>9.81</td>
</tr>
<tr>
<td>(v_0)</td>
<td>nominal velocity</td>
<td>m/s</td>
<td>30.0</td>
</tr>
<tr>
<td>(F_A)</td>
<td>propulsion force</td>
<td>N</td>
<td>0</td>
</tr>
<tr>
<td>(L_A)</td>
<td>propulsion moment</td>
<td>kg m²</td>
<td>0</td>
</tr>
<tr>
<td>(\kappa)</td>
<td>describes track geometry</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>(\alpha)</td>
<td>describes track geometry</td>
<td>rad</td>
<td>0</td>
</tr>
<tr>
<td>(\omega_0)</td>
<td>nominal angular velocity</td>
<td>1/s</td>
<td>(v_0/\rho_0)</td>
</tr>
<tr>
<td>(I_1)</td>
<td>lateral moment of inertia</td>
<td>kg m²</td>
<td>0.0605</td>
</tr>
<tr>
<td>(I_2)</td>
<td>vertical moment of inertia</td>
<td>kg m²</td>
<td>0.366</td>
</tr>
<tr>
<td>(m_A)</td>
<td>mass of wagon body</td>
<td>kg</td>
<td>0.0</td>
</tr>
<tr>
<td>(h_A)</td>
<td>height of wagon body</td>
<td>m</td>
<td>0.2</td>
</tr>
<tr>
<td>(c_x)</td>
<td>spring constant</td>
<td>N/m</td>
<td>6400.0</td>
</tr>
<tr>
<td>(c_z)</td>
<td>spring constant</td>
<td>N/m</td>
<td>6400.0</td>
</tr>
<tr>
<td>(x_l)</td>
<td>width of wheelset/2</td>
<td>m</td>
<td>0.19</td>
</tr>
<tr>
<td>(\delta_0)</td>
<td>cone angle/2</td>
<td>rad</td>
<td>0.0262</td>
</tr>
<tr>
<td>(\rho_0)</td>
<td>nominal radius</td>
<td>m</td>
<td>0.1</td>
</tr>
<tr>
<td>(a_0)</td>
<td>gauge/2</td>
<td>m</td>
<td>0.1506</td>
</tr>
<tr>
<td>(\rho_1)</td>
<td>radius track</td>
<td>m</td>
<td>0.06</td>
</tr>
<tr>
<td>(\mu)</td>
<td>friction coefficient</td>
<td></td>
<td>0.12</td>
</tr>
<tr>
<td>(G)</td>
<td>glide module</td>
<td>N/m²</td>
<td>7.92 \cdot 10^{10}</td>
</tr>
<tr>
<td>(C_{11})</td>
<td>Kalker coefficient</td>
<td></td>
<td>4.72772197</td>
</tr>
<tr>
<td>(C_{22})</td>
<td>Kalker coefficient</td>
<td></td>
<td>4.27526987</td>
</tr>
<tr>
<td>(C_{23})</td>
<td>Kalker coefficient</td>
<td></td>
<td>1.97203505</td>
</tr>
<tr>
<td>(\hat{G})</td>
<td>parameter for computation of contact ellipse</td>
<td></td>
<td>0.7115218</td>
</tr>
<tr>
<td>(e)</td>
<td>parameter for computation of contact ellipse</td>
<td></td>
<td>1.3537956</td>
</tr>
<tr>
<td>(\sigma)</td>
<td>parameter for computation of contact ellipse</td>
<td></td>
<td>0.28</td>
</tr>
<tr>
<td>(C)</td>
<td>scaling factor for Lagrange multipliers</td>
<td></td>
<td>10⁴</td>
</tr>
</tbody>
</table>

requires simplifications and table look ups in order to eliminate the nonlinear constraints. The above example provides thus an alternative by using the IDE approach.

Figure II.22.2 shows the mechanical model. The coordinates \(p\) denote the displacements and rotations of the wheelset with respect to the reference frame which is centered in the middle of the track. The wheelset is subjected to

- the gravity and centrifugal forces;
- creep forces in the contact points of wheel and rail;
- forces of the wagon body, which is represented by a frame connected to the wheelset via springs and dampers and proceeding with constant speed \(v_0\);
- constraint forces which enforce the contact of wheel and rail on both sides.

We are particularly interested in a complete and correct formulation of the nonlinear constraint equations. An elimination of the constraints without severe simplifications or the introduction of tables for the dependent variables is impossible. In this example thus a reduction to state space form involves various obstacles, whereas the IDE formulation is straightforward.
Equations (II.22.1)–(II.22.2) stand for the kinematic and dynamic equations with positive definite mass matrix $M(p)$. By means of the profile functions $R$ and $G$ which describe the cross sections of wheel and rail depending on the contact points we first express the constraint equations as $0 = g_1$, see Figure II.22.3. These constraints are of index 3 and enforce that the contact points of wheel and rail coincide on both sides. Additionally, we have to guarantee that wheel and rail do not intersect, which is accomplished by the conditions $0 = g_2$. Note that $\partial g_2 / \partial q$ is regular, which means that we can apply formally the implicit function theorem to eliminate the additional contact variables $q$ and that these constraints are of index 1. The equations of motion of the wheelset are then derived by applying the formalism of Newton and Euler. Here we used the property that this class of contact problems $(\partial g_1 / \partial q) \dot{q} \equiv 0$. This also implies that if we, in order to get the index 2 formulation, differentiate the constraint (II.22.3) with respect to $t$, then we get

$$0 = \frac{d g_1}{dt}(p, q) = \frac{\partial g_1}{\partial p} \dot{p} + \frac{\partial g_1}{\partial q} \dot{q} = \frac{\partial g_1}{\partial p} \dot{p} - \frac{\partial g_1}{\partial q} \left( \frac{\partial g_2}{\partial q} \right)^{-1} \frac{\partial g_2}{\partial p} \dot{p},$$

which simplifies to (II.22.5).

**Remarks**

- $N(p, q, \lambda) \in \mathbb{R}^2$ denotes the normal forces which act in the contact points. They are necessary to evaluate the creep forces.
- The variable $\beta \in \mathbb{R}$ denotes the deviation of the angular velocity and is given by an additional differential equation.
- The parameters $\kappa$ and $\alpha$ describe the track geometry. The setting $\kappa = \alpha = 0$ refers to a straight track.
The constant $C$ in (II.22.2) means that we internally scaled the Lagrange multipliers.

The initial values correspond to a setting in which the dynamic behavior of the wheelset model is investigated when the wheelset starts with an initial deflection in lateral direction ($x$-direction) of $0.14941 \text{ [cm]}$. In [Jas90], a limit cycle was observed for this problem and the model data given above. This type of limit cycle, the so-called hunting motion, is a well known phenomenon in railway vehicle dynamics. In Figure II.22.4 we see this limit cycle as computed by DASSL applied to the index-2 formulation of the problem. The results are in good agreement with those given in [Jas90], which were obtained by a state space form approach and with measurements on a hardware model.

### 22.4 Numerical solution of the problem

Tables II.22.2–II.22.3 present the reference solution at the end of the integration interval, and the run characteristics, respectively. Figure II.22.5 shows the the behavior of the components of $p$ and the angular velocity $\beta$ over the integration interval. Figures II.22.6–II.22.7 contain the work-precision diagrams. For this diagrams, we used: $\text{rtol} = 10^{-4+m/8}$, $m = 0,1,\ldots,48$; $\text{atol} = \text{rtol}$, $h0 = \text{rtol}$ for MEBDFI.
Remarks

- The Jacobian was computed internally by the solvers.
- For the runs with DASSL, we excluded the Lagrange multipliers from the error control by setting $\text{atol}(16)=\text{atol}(17)=\text{rtol}(16)=\text{atol}(17)=10^{10}$.
- The reference solution was computed using DASSL with $\text{atol} = \text{rtol} = 10^{-9}$ for $p, v$ and $q$, and $\text{atol} = \text{rtol} = 10^{10}$ for $\lambda$. 
Table II.22.3: Run characteristics.

<table>
<thead>
<tr>
<th>solver</th>
<th>rtol</th>
<th>atol</th>
<th>h0</th>
<th>mescd</th>
<th>scd</th>
<th>steps</th>
<th>accept</th>
<th>#f</th>
<th>#Jac</th>
<th>#LU</th>
<th>CPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>DDASSL</td>
<td>10^{-4}</td>
<td>10^{-4}</td>
<td></td>
<td>1.35</td>
<td>0.15</td>
<td>5949</td>
<td>5117</td>
<td>10304</td>
<td>1407</td>
<td>0.3250</td>
<td></td>
</tr>
<tr>
<td></td>
<td>10^{-5}</td>
<td>10^{-5}</td>
<td></td>
<td>2.78</td>
<td>1.40</td>
<td>9888</td>
<td>8667</td>
<td>16150</td>
<td>1815</td>
<td>0.4782</td>
<td></td>
</tr>
<tr>
<td></td>
<td>10^{-6}</td>
<td>10^{-6}</td>
<td></td>
<td>3.67</td>
<td>2.32</td>
<td>16010</td>
<td>14298</td>
<td>25256</td>
<td>2577</td>
<td>0.7213</td>
<td></td>
</tr>
<tr>
<td>MEBDFI</td>
<td>10^{-4}</td>
<td>10^{-4}</td>
<td>10^{-4}</td>
<td>1.32</td>
<td>0.12</td>
<td>5758</td>
<td>5188</td>
<td>42694</td>
<td>1185</td>
<td>1185</td>
<td>0.4031</td>
</tr>
<tr>
<td></td>
<td>10^{-5}</td>
<td>10^{-5}</td>
<td>10^{-5}</td>
<td>3.93</td>
<td>2.59</td>
<td>9317</td>
<td>8485</td>
<td>64945</td>
<td>1765</td>
<td>1765</td>
<td>0.6266</td>
</tr>
<tr>
<td></td>
<td>10^{-6}</td>
<td>10^{-6}</td>
<td>10^{-6}</td>
<td>4.89</td>
<td>3.22</td>
<td>13240</td>
<td>12255</td>
<td>86260</td>
<td>2248</td>
<td>2248</td>
<td>0.8560</td>
</tr>
<tr>
<td>PSIDE-1</td>
<td>10^{-4}</td>
<td>10^{-4}</td>
<td></td>
<td>1.53</td>
<td>0.42</td>
<td>1276</td>
<td>945</td>
<td>22090</td>
<td>547</td>
<td>4920</td>
<td>0.5134</td>
</tr>
<tr>
<td></td>
<td>10^{-5}</td>
<td>10^{-5}</td>
<td></td>
<td>2.81</td>
<td>1.67</td>
<td>2335</td>
<td>1507</td>
<td>39204</td>
<td>608</td>
<td>8752</td>
<td>0.8384</td>
</tr>
<tr>
<td></td>
<td>10^{-6}</td>
<td>10^{-6}</td>
<td></td>
<td>4.52</td>
<td>3.34</td>
<td>3070</td>
<td>2068</td>
<td>54074</td>
<td>571</td>
<td>10736</td>
<td>1.0775</td>
</tr>
</tbody>
</table>

Figure II.22.5: Behavior of some solution components over the integration interval.

References


Figure II.22.6: Work-precision diagram (sod versus CPU-time).
Figure II.22.7: Work-precision diagram (mescd versus CPU-time).