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
IVPtestset 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

[Cas79] J. Cash. *Stable Recursions with applications to the numerical solution of stiff systems.* Academic Press, New York, 1979.

[Cas83] J. Cash. The integration of stiff initial value problems in o.d.e.s using modified extended backward differentiation formulae. *Comp. and Maths. with Applics.*, 9:645–657, 1983.

```
$ 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:

                                              scd
       solution component                -----------------------------     ignore
                                         mixed       abs        rel    mix - abs,rel
 ---------------------------------       -----      -----      -----   -------------
 y(  1) =  0.7360756579676240E-003        5.98       5.98       2.84
 y(  2) =  0.1440435009167338E-003        6.69       6.69       2.85
 y(  3) =  0.5867365037055238E-004        6.67       6.67       2.44
 y(  4) =  0.1173828077122226E-002        5.74       5.74       2.81
 y(  5) =  0.2347013337886003E-002        4.41       4.41       1.78
 y(  6) =  0.6023708667056447E-002        3.67       3.67       1.46
 y(  7) =  0.2893696909773767E-002        4.36       4.36       1.81
 y(  8) =  0.2806303090227050E-002        4.36       4.36       1.81

 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:

    number of integration steps          92
    number of accepted steps             89
    number of f evaluations             311
    number of Jacobian evaluations       18
    number of LU decompositions          18

 CPU-time used:                        0.0010 sec
```

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`.

[Cas03] J. Cash. Efficient numerical methods for the solution of stiff initial-value problems and differential algebraic equations. *Proc. Roy. Soc. London, A*, 459:797–815, 2003.

[CC92] J. Cash and S. Considine. An mebdf code for stiff initial value problems. *Acm Trans Math Software*, pages 142–158, 1992.

[Hin83] Alan C. Hindmarsh. ODEPACK, a systemized collection of ODE solvers. In R. Stepleman et al., editors, *Scientific Computing*, pages 55–64, Amsterdam, 1983. IMACS, North-Holland Publishing Company.

[HW96] E. Hairer and G. Wanner. *Solving Ordinary Differential Equations II: Stiff and Differential-algebraic Problems.* Springer-Verlag, second revised edition, 1996.