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 beetwen 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  gamda.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

[BT98] L. Brugnano and D. Trigiante. *Solving Differential Problems by Multistep Initial and Boundary Value Methods.* Gordon & Breach, Amsterdam, 1998.

```
$ f90 -O5 -o dotest gamdd.f hires.f gamda.f90  gamd.f90 report.f
$ dotest

  Test Set for IVP Solvers (release 2.3)

  Solving Problem HIRES using GAMD90

 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) =  0.7370189658683070E-003       6.95       6.95       3.82
 y(  2) =  0.1442269592313960E-003       7.67       7.67       3.82
 y(  3) =  0.5886363518265143E-004       7.63       7.63       3.40
 y(  4) =  0.1175477661507891E-002       6.76       6.76       3.83
 y(  5) =  0.2381655379215545E-002       5.33       5.33       2.71
 y(  6) =  0.6221249713391935E-002       4.75       4.75       2.55
 y(  7) =  0.2848304918830136E-002       5.77       5.77       3.23
 y(  8) =  0.2851695081169868E-002       5.77       5.77       3.23

 used components for  scd                   8          8          8
 scd of Y (maximum norm)                 4.75       4.75       2.55


 using mixed error yields mescd          4.75
 using relative error yields scd                               2.55

 Integration characteristics:

    number of integration steps        29
    number of accepted steps           24
    number of f evaluations           967
    number of Jacobian evaluations      24
    number of LU decompositions        29

 CPU-time used:                       0.0020 sec
```

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`*.*

[IM98] F. Iavernaro and F. Mazzia. Solving ordinary differential equations by generalized adams methods: properties and implementation techniques. *Appl. Num. Math.*, 28:107–126, 1998.

[IM99] F. Iavernaro and F. Mazzia. Block-boundary value methods for the solution of ordinary differential equations. *SIAM J. Sci. Comput.*, 21(1):323–339, 1999.