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
IVPtestset 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_{out} (final integration time). It allows to continue the solution to get results at additional t_{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 ≤ 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.

```
$ f90 -O5 -o dotest dassld.f hires.f ddassl.f dassla.f report.f
$ ./dotest

  Test Set for IVP Solvers (release 2.3)

  Solving Problem HIRES using DASSL

 User input:

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

 Numerical solution:

                                              scd
        solution component                ----------------------------     ignore
                                          mixed       abs        rel    mix - abs,rel
 ---------------------------------        -----      -----      -----   -------------
 y(  1) =  0.7437259735671353E-003         5.18       5.18       2.05
 y(  2) =  0.1455514426118115E-003         5.89       5.89       2.04
 y(  3) =  0.6009984916041035E-004         5.92       5.92       1.69
 y(  4) =  0.1188134706173305E-002         4.90       4.90       1.97
 y(  5) =  0.2577046600086416E-002         3.72       3.72       1.10
 y(  6) =  0.6824947575510993E-002         3.23       3.23       1.03
 y(  7) =  0.2989385921555588E-002         3.86       3.86       1.31
 y(  8) =  0.2710614078444423E-002         3.86       3.86       1.31

 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

[BCP96] K.E. Brenan, S.L. Campbell, and L.R. Petzold. *Numerical Solution of Initial–Value Problems in Differential–Algebraic Equations.* SIAM, second edition, 1996.