

Machine language (Caltech undergrad)

FORTRAN (Caltech grad)

Mathematical Software (Sandia)

$$\text{for } y' = f(t, y), a \leq t \leq b \\ y(a) = \underline{A}$$

- portable FORTRAN

- algorithms

estimate, control error

stiffness

linear algebra

starting step size

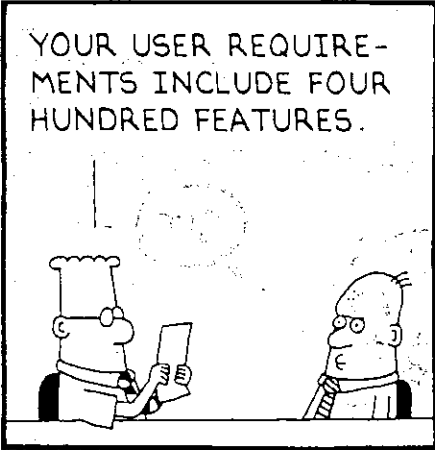
- user interface
answers - where? how accurate?
ask less of users
- documentation
levels
templates
- libraries
uniform appearance
share subprograms

Large scale scientific computing

DVODE - Adams/BDF - Brown,
Byrne, Hindmarsh II.1, II.2, II.3

DILBERT

By Scott Adams



www.dilbert.com scottadams@aol.com

4/14/01 © 2001 United Feature Syndicate, Inc.

New computing environments

- * ease of use
- * graphical interpretation of results
- * interactive computation

Teaching packages

e.g. ODE architect, CODEE group

restricted class of ODEs

graphical input of ICs II.5

Uses RKSUITE/DVODE -

considerable capabilities.

Restricted syntax, tabbed displays, graphics II.6

" Problem Solving Environments (PSEs):
 often where students are first exposed
 to scientific computing and where scientists
 and engineers formulate and solve
 their problems. " W.H. Enright ANODE 2001

CHM6 is a stiff IVP. With new solvers
 do little more than define the IVP.

Maple, Matlab II.7, II.8

- step sizes range over
 14 orders of magnitude
- ode15s is more capable than
 DVODE (and IVPsolve) -
 event location, DAEs, ...

PSEs have capabilities not widely available/exploited in GSC:

- Output as structure so can evaluate $y(x)$ anywhere
- Maple can form $\frac{\partial f}{\partial y}$ analytically.
 IVPsolve uses a method that requires this.
- Matlab has fully integrated sparse matrix technology, so ode15s allows general sparse $\frac{\partial f}{\partial y}$. (DVIDE , IVPsolve do not.)

SUBROUTINE DVODE (F, NEQ, Y, T, TOUT, ITOL, RTOL, ATOL, ITASK,
1 ISTATE, IOPT, RWORK, LRW, IWORK, LIW, JAC, MF)



C-----

C Summary of usage.

C
C Communication between the user and the DVODE package, for normal
C situations, is summarized here. This summary describes only a subset
C of the full set of options available. See the full description for
C details, including optional communication, nonstandard options,
C and instructions for special situations.



C ITASK = 1 for normal computation of output values of Y at t = TOUT.



C MF = Method flag. Standard values are..
C 10 for nonstiff (Adams) method, no Jacobian used.
C 21 for stiff (BDF) method, user-supplied full Jacobian.
C 22 for stiff method, internally generated full Jacobian.
C 24 for stiff method, user-supplied banded Jacobian.
C 25 for stiff method, internally generated banded Jacobian.



C RWORK = Real work array of length at least..
C 20 + 16*NEQ for MF = 10,
C 22 + 9*NEQ + 2*NEQ**2 for MF = 21 or 22,
C 22 + 11*NEQ + (3*ML - 2*MU)*NEQ for MF = 24 or 25.
C LRW = Declared length of RWORK (in user's DIMENSION statement).



C E. The output from the first call (or any call) is..
C Y = Array of computed values of y(t) vector.
C T = Corresponding value of independent variable (normally TOUT).



C F. To continue the integration after a successful return, simply
C reset TOUT and call DVODE again. No other parameters need be reset.



```

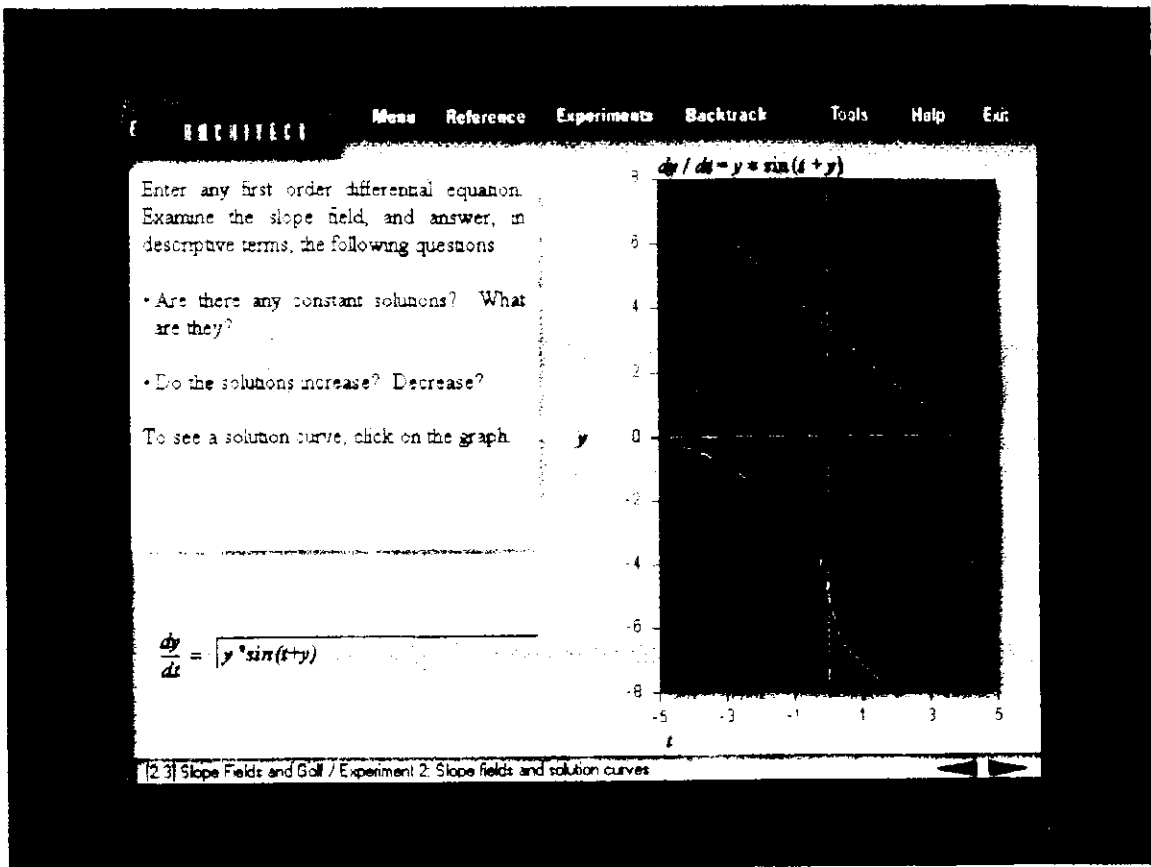
C-----
C Full description of user interface to DVODE.
C
C
C ITASK = An index specifying the task to be performed.
C Input only. ITASK has the following values and meanings.
C 1 means normal computation of output values of y(t) at
C t = TOUT (by overshooting and interpolating).
C 2 means take one step only and return.
C 3 means stop at the first internal mesh point at or
C beyond t = TOUT and return.
C 4 means normal computation of output values of y(t) at
C t = TOUT but without overshooting t = TCRIT.
C TCRIT must be input as RWORK(1). TCRIT may be equal to
C or beyond TOUT, but not behind it in the direction of
C integration. This option is useful if the problem
C has a singularity at or beyond t = TCRIT.
C 5 means take one step, without passing TCRIT, and return.
C TCRIT must be input as RWORK(1).
C
C Note.. If ITASK = 4 or 5 and the solver reaches TCRIT
C (within roundoff), it will return T = TCRIT (exactly) to
C indicate this (unless ITASK = 4 and TOUT comes before TCRIT,
C in which case answers at T = TOUT are returned first).
C
C
C
C MF = The method flag. Used only for input. The legal values of
C MF are 10, 11, 12, 13, 14, 15, 20, 21, 22, 23, 24, 25,
C -11, -12, -14, -15, -21, -22, -24, -25.
C MF is a signed two-digit integer, MF = JSV*(10*METH + MITER).
C JSV = SIGN(MF) indicates the Jacobian-saving strategy..
C JSV = 1 means a copy of the Jacobian is saved for reuse
C in the corrector iteration algorithm.
C JSV = -1 means a copy of the Jacobian is not saved
C (valid only for MITER = 1, 2, 4, or 5).
C METH indicates the basic linear multistep method..
C METH = 1 means the implicit Adams method.
C METH = 2 means the method based on backward
C differentiation formulas (BDF-s).
C MITER indicates the corrector iteration method..
C MITER = 0 means functional iteration (no Jacobian matrix
C is involved).
C MITER = 1 means chord iteration with a user-supplied
C full (NEQ by NEQ) Jacobian.
C MITER = 2 means chord iteration with an internally
C generated (difference quotient) full Jacobian
C (using NEQ extra calls to F per df/dy value).
C MITER = 3 means chord iteration with an internally
C generated diagonal Jacobian approximation
C (using 1 extra call to F per df/dy evaluation).
C MITER = 4 means chord iteration with a user-supplied
C banded Jacobian.
C MITER = 5 means chord iteration with an internally
C generated banded Jacobian (using ML+MU+1 extra
C calls to F per df/dy evaluation).
C If MITER = 1 or 4, the user must supply a subroutine JAC
C (the name is arbitrary) as described above under JAC.
C For other values of MITER, a dummy argument can be used.

```

```

C RWORK = A real working array (double precision).
C The length of RWORK must be at least
C  $20 + NYH*(MAXORD + 1) + 3*NEQ + LWM$  where
C NYH = the initial value of NEQ,
C MAXORD = 12 (if METH = 1) or 5 (if METH = 2) (unless a
C smaller value is given as an optional input),
C LWM = length of work space for matrix-related data..
C  $LWM = 0$  if MITER = 0,
C  $LWM = 2*NEQ**2 + 2$  if MITER = 1 or 2, and MF.gt.0,
C  $LWM = NEQ**2 + 2$  if MITER = 1 or 2, and MF.lt.0,
C  $LWM = NEQ - 2$  if MITER = 3,
C  $LWM = (3*ML+2*MU+2)*NEQ + 2$  if MITER = 4 or 5, and MF.gt.0,
C  $LWM = (2*ML+MU+1)*NEQ + 2$  if MITER = 4 or 5, and MF.lt.0.
C (See the MF description for METH and MITER.)
C Thus if MAXORD has its default value and NEQ is constant,
C this length is..
C  $20 + 16*NEQ$  for MF = 10,
C  $22 + 16*NEQ + 2*NEQ**2$  for MF = 11 or 12,
C  $22 + 16*NEQ + NEQ**2$  for MF = -11 or -12,
C  $22 + 17*NEQ$  for MF = 13,
C  $22 + 18*NEQ + (3*ML+2*MU)*NEQ$  for MF = 14 or 15,
C  $22 + 17*NEQ + (2*ML+MU)*NEQ$  for MF = -14 or -15,
C  $20 + 9*NEQ$  for MF = 20,
C  $22 + 9*NEQ + 2*NEQ**2$  for MF = 21 or 22,
C  $22 + 9*NEQ + NEQ**2$  for MF = -21 or -22,
C  $22 + 10*NEQ$  for MF = 23,
C  $22 + 11*NEQ + (3*ML+2*MU)*NEQ$  for MF = 24 or 25.
C  $22 + 10*NEQ + (2*ML+MU)*NEQ$  for MF = -24 or -25.
C The first 20 words of RWORK are reserved for conditional
C and optional input and optional output.
C
C The following word in RWORK is a conditional input..
C RWORK(1) = TCRIT = critical value of t which the solver
C is not to overshoot. Required if ITASK is
C 4 or 5, and ignored otherwise. (See ITASK.)
C
C LRW = The length of the array RWORK, as declared by the user.
C (This will be checked by the solver.)

```





```

/* A model for lead in the body:
   x1 = amount in blood
   x2 = amount in tissue
   x3 = amount in bones */
x1' = -k01*x1 + k03 + k21*x1 + k12*x2 +
x2' = k21*x1 - (k02 + k12)*x2
x3' = k31*x1 - k13*x3

k01 = 0.0011; k03 = 0.0162
k21 = 0.0111; k12 = 0.0124
k31 = 0.0039; k13 = 0.00035

```

```

/* Lead input:
   I1 = 49.3*(1 - step(t,400))

```

5:1 INS

Initial Conditions	Integration
t: 0	Interval: 800
x1: 0	# Points: 100
x2: 0	<input type="button" value="Clear Last"/> <input type="button" value="Clear All"/>
x3: 0	<input type="button" value="←"/> Solve <input type="button" value="→"/>

Enable Graphical IC

