On Developing Mathematical Software

IWTAM, Bari, Italy, December 2002

Fred T. Krogh — fkrogh@mathalacarte.com

- Part 1 — a list of random thoughts
- Part 2 — gives some examples from our work
- Part 3 — plans for a new integrator
- Paper for Parts 1 & 2 is available at

http://mathalacarte.com/fkrogh

20 Restricted 3 Body Problems
Tolerances $10^{-2} - 10^{-12}$
(From Philip Sharp)

Method	CPU Secs.
RK Suite	1.280
dxrk8	.366
step	1.770
diva	.974
odex	.486

Accuracy for the various codes was comparable.

These results are in the slides but not in the paper as we don't regard our Runge Kutta code, dxrk8, to be noticeably better than DOP853 the code it was derived from. Diva is a variable order Adams code of ours which integrates second order equations without requiring that they be broken up into first order equations. We believe that most (but not all) of the difference between diva and step is due to that fact. The results do show that even for these differential equations the choice of method and the coding details can make a significant difference.

Part 1: The List

1. Internal output

2. Output at the interface

3. Detailed error messages

4. The fine details: cancellation, loops, if tests, etc.

5. Understandable test cases (tune with care)

6. Verify results with care

7. Take advantage of your mistakes

The List (Continued)

8. Keep consistent with units

9. Test the extremes

10. The algorithm matters (is theory useful?)

11. Work on what doesn't work. E.g. using for BDF: $\delta\mathbf{y}^T\delta\mathbf{f}/(||\delta\mathbf{y}||\ ||\delta\mathbf{f}||)$

12. Toss out what doesn't work

13. Don't give up too soon

14. Your subconscious is a powerful tool; learn to use it

The List (Continued)

15. Test assumptions

16. Keep a variable dictionary in comments

17. Write user documentation first

18. Know what performance you should expect to get

19. Don't pay too much (just enough) attention to others

Most Important: See setbacks as learning opportunities, and as the stimulus necessary for the generation of new ideas.

Part 2: Examples

1. Early variable order Adams (DVDQ)

2. Quadrature (more reliable and half the function evals.)

3. 1-D zero finding

Total Number of Function Evaluations in
Solving All the Problems Listed in Table I of
TOMS 21, 327–344(1995) (from Dr. Shi)

tol	BR	DE	M	R	LE	2.1	2.2
10^{-7}	2804	2808	2839	7630	2694	3154	2950
10^{-10}	2905	2963	2992	7768	2821	3338	3060
10^{-15}	2975	3196	3261	8014	3061	3448	3151
0	3008	2998	3146	8230	3165	3509	3219

tol	2.3	2.4	2.5	4.1	4.2	DZ
10^{-7}	2645	2791	2687	2696	2650	2100
10^{-10}	2789	2922	2819	2835	2786	2177
10^{-15}	2948	3015	2914	2908	2859	2236
0	3029	3060	2954	2950	2884	2255

Examples (Continued)

5. Nonlinear least squares (Reliable and no restriction on how much or how many times function values can increase)

6. Current project – Optimization

NV	NE	NB	NBA	NI	NIA	M		#
60	20	40	20	40	20	0	LP	200
100	20	100	80	0	0	0	LP	200
500	200	500	300	0	0	0	LP	1
240	80	240	80	160	80	0	LP	2
60	20	40	20	40	20	0	LP	200
100	20	100	80	0	0	0	LP	200
500	200	500	300	0	0	0	LP	1
240	80	240	80	160	80	0	LP	2
300	0	300	200	0	0	400	LS	10
300	0	300	200	0	0	400	QP	10
300	0	300	50	0	0	400	LS	10
300	0	300	50	0	0	400	QP	10
200	0	160	50	160	50	300	LS	10
200	0	160	50	160	50	400	QP	10
200	40	60	30	20	10	200	LS	10
200	40	60	30	20	10	400	QP	10

Examples (Continued)

	dopt		lssol	
	Error	Secs.	Error	Secs.
LP	1.1E−12	8.2	3.9E−13	17.4
LP	2.2E−12	5.3	1.9E−11	12.2
LP	3.1E−13	11.8	3.2E−14	54.4
LP	1.0E−13	13.6	4.3E−14	29.6
LP	1.6E−12	9.1	3.5E−13	26.5
LP	1.0E−12	6.6	1.6E−12	14.4
LP	1.6E−14	13.8	1.3E−14	58.6
LP	4.0E−14	12.1	5.1E−14	62.9
LS	2.4E−15	42.6	2.7E−15	57.7
QP	2.2E−15	13.5	3.4E−15	18.5
LS	4.6E−15	37.5	5.8E−15	57.9
QP	5.3E−15	8.5	1.2E−14	19.4
LS	9.9E−15	40.2	7.6E−15	116.9
QP	1.9E−14	32.4	9.9E−15	108.4
LS	1.1E−14	10.4	5.9E−15	23.2
QP	1.3E−14	6.9	6.4E−15	20.0

Part 3: New ODE Code

(Joint work with Philip Sharp)

- Is there anything left to do?
- Where can big gains be made?
- What new features might be useful?

ODE Code: Efficiency

1. **Groups, Groups, Groups!**
2. Separate linear algebra
3. Improved starting
4. Option for internal independent variable
5. Multirate (Different groups, different step-sizes)
6. Action lists
7. Transition tables
8. "Save" variables in user space

ODE Code: Features

9. Non stiff ODE's (Adams)

10. Stiff ODE's (BDF)

11. DAE's (Adams and/or BDF)

12. Method switching

13. Delay Equations

14. Direct integration of higher order equations

15. Use of extra derivatives

16. Mixtures of the above.

ODE Code: Features (Continued)

17. Saving the solution

18. G-Stops (interpolation & extrapolation)

19. Save/Restore stepsize and order

20. Reverse communication available

21. Control over what is interpolated

22. And more ...