Homework III

Greg Nelson, Jeremy Morris and Anup Bhawalkar

March 9, 2005

1 Ex 1

I decided to follow Phil's advice and use Python to solve this problem. It took a few days, actually, to get everything installed properly. Once I did, there was really no problems to speak of. Python is a very easy language to use and to learn.

The Python SciPy package uses lsode along with lapack, blas and atlas to solve any number of problems. As has been mentioned, it is quite easy to use and the documentation is easy to read.

It is interesting to note that for the value $\lambda = 10$, we could not get a solution. For all other values of λ, a solution was found. I have prepared tables for each λ and for all specified values of τ. The tables follow:

λ	= 10		
τ	y_n	rel error	num func eval
0.001	-2.73770834931e+38	5.03235682313e+38	490
0.0001	-6.74204785994e+37	1.23929893987e+38	611
1e-05	6.6614830914e+36	1.22448981447e+37	758
1e-06	2.55025181408e+35	4.68778097584e+35	908
1e-07	9.13084320311e+34	1.67839869085e+35	1150
1e-08	3.15483446583e+33	5.79910301767e+33	1248
1e-09	2.67218816206e+33	4.91191997621e+33	1290
1e-10	1.90107968102e+32	3.49449615643e+32	1513

λ	= 1		
τ	y_n	rel error	num func eval
0.001	5.2087025967	10.5744493962	82
0.0001	-0.203697243843	0.625571067435	102
1e-05	-0.45885856281	0.156542726697	185
1e-06	-0.537365299585	0.0122344724704	180
1e-07	-0.542474196298	0.00284348265288	206
1e-08	-0.544094562149	0.000135015458007	254
1e-09	-0.54402010438	1.85012836365e-06	319
1e-10	-0.54402109841	2.29399383417e-08	421

λ	= 0		
τ	y_n	rel error	num func eval
0.001	-0.54510926401	0.00200020384948	97
0.0001	-0.543962285783	0.000108130191298	99
1e-05	-0.544027896663	1.24733652385e-05	135
1e-06	-0.544021419617	5.67491517115e-07	137
1e-07	-0.544020719282	7.19838061871e-07	257
1e-08	-0.544021093773	3.1461813452e-08	225
1e-09	-0.544021113639	5.05418908237e-09	307
1e-10	-0.544021111116	4.15845243663e-10	409

λ	= -1		
τ	y_n	rel error	num func eval
0.001	-0.544026359389	9.64760345291e-06	103
0.0001	-0.544010327292	1.98220194775e-05	118
1e-05	-0.544019685609	2.6198985494e-06	182
1e-06	-0.544020853437	4.73238849994e-07	205
1e-07	-0.54402107868	5.92059693726e-08	214
1e-08	-0.544021104014	1.26375125595e-08	268
1e-09	-0.544021110935	8.45577532308e-11	285
1e-10	-0.544021110928	7.08184601383e-11	451

λ	= -1000		
τ	y_n	rel error	num func eval
0.001	-0.544024400093	6.04609541539e-06	176
0.0001	-0.544026043682	9.0672819062e-06	236
1e-05	-0.544021391813	5.16382824942e-07	386
1e-06	-0.544021167571	1.04190612665e-07	435
1e-07	-0.54402113072	3.64519975995e-08	1086
1e-08	-0.54402111085	7.31694294275e-11	3806
1e-09	-0.544021110875	2.59637211646e-11	590
1e-10	-0.544021110886	6.89862550795e-12	765

λ	= -1000000		
τ	y_n	rel error	num func eval
0.001	-0.544034209652	2.40776742738e-05	195
0.0001	-0.544023508095	4.40645683754e-06	282
1e-05	-0.544021109227	3.05585539086e-09	348
1e-06	-0.54402111089	6.77128133812e-13	416
1e-07	-0.544021111695	1.48168880288e-09	624
1e-08	-0.544021110159	1.34216857127e-09	683
1e-09	-0.54402111088	1.66914737988e-11	1013
1e-10	-0.544021110889	1.23834042073e-12	1228

λ	= -1000000000		
τ	y_n	rel error	num func eval
0.001	-0.54402735965	1.14862461926e-05	213
0.0001	-0.544031045659	1.82617345927e-05	294
1e-05	-0.544021607996	9.13763812537e-07	420
1e-06	-0.544021075262	6.54881388969e-08	351
1e-07	-0.544021113006	3.89133538762e-09	591
1e-08	-0.544021110718	3.14530711931e-10	862
1e-09	-0.544021110889	1.11017994211e-13	1227
1e-10	-0.54402111089	2.59586192347e-13	1256

λ	= -1000000000000		
τ	y_n	rel error	num func eval
0.001	-0.544027556507	1.1848101396e-05	232
0.0001	-0.54402630527	9.54812295661e-06	291
1e-05	-0.544021608166	9.140765339e-07	439
1e-06	-0.544021052637	1.07077067453e-07	443
1e-07	-0.544021066726	8.11802870779e-08	617
1e-08	-0.544021110515	6.8779790181e-10	835
1e-09	-0.544021111132	4.45455212075e-10	835
1e-10	-0.544021110887	5.12764360764e-12	1090

2 Ex 2

All of us did this problem. Greg and Anup used Matlab, Jeremy used Python. It appears that it will not solve for $\tau = 10^4, 10^6$ and that for other values of τ, the relative error could not be controlled very well. Here is the table of values we got:

τ = 0.001	func eval = 183	jac evals = 21
y_n	rel err	
4.44394727881e-05	0.0211554726948	
4.4439472774e-23	0.0211554730034	
0.999955560527	4.44394727879e-05	
τ = 1e-05	func eval = 225	jac evals = 24
y_n	rel err	
4.54217522001e-05	0.000480671175107	
4.54217522106e-23	0.000480671406205	
0.999954578248	4.54217521997e-05	
τ = 1e-07	func eval = 268	jac evals = 27
y_n	rel err	
4.54163073966e-05	0.000360741398717	
4.54163073869e-23	0.000360741183282	
0.999954583693	4.54163073955e-05	
τ = 1e-08	func eval = 253	jac evals = 24
y_n	rel err	
4.54146827413e-05	0.000324955983173	
4.54146827452e-23	0.000324956070088	
0.999954585317	4.54146827404e-05	
τ = 1e-09	func eval = 261	jac evals = 24
y_n	rel err	
4.54141766296e-05	0.00031380813126	
4.54141766347e-23	0.000313808243022	
0.999954585823	4.54141766291e-05	
τ = 1e-10	func eval = 262	jac evals = 25
y_n	rel err	
4.54139531356e-05	0.000308885349291	
4.54139531086e-23	0.000308884754617	
0.999954586047	4.54139531348e-05	

3 Ex 3

The problem number three is solved using MATLAB. MATLAB has various built in functions for solving ode. There are both stiff and non stiff solvers in MATLAB. These solvers are listed below

3.1 Non stiff solvers

3.1.1 ode45

ode45 is based on an explicit Runge-Kutta formula. It is one step solver. Order of accuracy is medium

3.1.2 ode23

ode23 is an implementation of an explicit Runge-Kutta method. This is also a one step solver. Order of accuracy is low. It may be better than ode45 at crude tolerances.

3.1.3 ode113

This is is a variable order Adams-Bashforth-Moulton PECE solver. ode113 is a multistep solver. Order of accuracy is low to high

3.2 Stiff Solvers

3.2.1 ode15s

ode15s is a variable order solver based on the numerical differentiation formulas (NDFs). Optionally, it uses the backward differentiation formulas (BDFs). Multistep solver. Order of accuaracy is medium to low.

3.2.2 ode23s

ode23s is based on a modified Rosenbrock formula of order 2. It is one step solver.

3.3 Answers

We have used ode15s for solving the problem number three.

variable	τ	num func eval	num jac eval	sol at right end point	actual rel err
a				5.361539722881250E-01	5.875661029014800E-04
b	1.E-03	124	7	4.493028635601460E-06	1.212544108853530E-03
c				4.638415346832410E-01	6.783195588222940E-04
a				5.359622574621910E-01	2.297818077519060E-04
b	1.E-04	139	5	4.489596634936950E-06	4.477677418664900E-04
c				4.640332529411610E-01	2.652728664715950E-04
a				5.358353618961680E-01	7.034726525601020E-06
b	1.E-05	216	4	4.487526497475790E-06	1.353525934597570E-05
c				4.641601505773340E-01	8.121276878172860E-06
a				5.358383558902350E-01	1.447239738137550E-06
b	1.E-06	302	4	4.487575057373640E-06	2.714322554818030E-06
c				4.641571565347080E-01	1.670772805278280E-06
a				5.358391783166700E-01	8.759858701323540E-08
b	1.E-07	378	9	4.487588053517790E-06	1.816978364097830E-07
c				4.641563340952770E-01	1.011287877017780E-07
a				5.358391344547030E-01	5.741991875109990E-09
b	1.E-08	496	8	4.487587291162220E-06	1.181688802527790E-08
c				4.641563779580060E-01	6.628893037607300E-09
a				5.358391313779190E-01	1.035966729203040E-15
b	Most Accurate	7720	5	4.487587238132900E-06	9.437509538050960E-16
c	Solution			4.641563810348490E-01	5.979789732449030E-16