PRAXIS

Samson Cheung
cheung@halcyonsystems.com
December, 2003

Abstract:

This report is to show a terrific performance of a global minimization algorithm under a slight modification. The minimization code is called praxis. It is a multivariable derivative free minimization algorithm.

PRAXIS

This report is a follow-up of the previous report regarding a single variable local minimization algorithm without derivative. The previous report can be found in: http://gmao.gsfc.nasa.gov/intranet/sys_eng/newtech/Minimization.pdf

Having communicated with scientist, it is realized that it is useful to have a multi-variable global minimization algorithm without derivative. It is identified that the PRAXIS algorithm may be useful. PRAXIS is Richard Brent's "principal axis" code for seeking an N-dimensional point X which minimizes a given scalar function F(X). The code is a refinement of Powell's method of conjugate search directions.

It turns out that a professor (John Burkardt) in Mathematics Department at Iowa State University has coded up such algorithm.
http://www.math.iastate.edu/burkardt/f_src/praxis/praxis.html

It turns out that a slight modification can yield a terrific improvement in terms of convergence of the algorithm. Below are the comparisons of Prof. Burkardt's code and the modified one using some classical examples.

The Beale Function

This function is by Beale (1958):

$$f(\mathbf{x}) = \sum_{i=1}^{3} [c_i - x_1(1 - x_2^i)]^2$$

where c_1=1.5, c_2=2.25, and c_3=2.625. This function has a valley approaching the line x_2=1, and has a minimum of 0 at (3, 0.5).

Initial guess is chosen to be (0.1, 0.1).

Original		**Modified**	
Function evaluations	***F(x)***	***Function evaluations***	***F(x)***
11	5.41586	11	5.41586
21	1.62891	21	1.50106
31	0.450770	31	0.363649
43	0.401868E-01	43	0.716855E-01
55	0.458760E-02	55	0.612914E-03
67	0.134551E-04	67	0.118035E-08
79	0.102227E-09	80	0.310457E-21
92	0.150625E-16	97	0.175376E-22
109	0.713752E-20		

This is a simple problem, and the modified code converges much faster than the original code.

The Box Function

This function is by Box (1966):

$$f(\mathbf{x}) = \sum_{i=1}^{10} \left\{ \exp(\frac{-ix_1}{10}) - \exp(\frac{-ix_2}{10}) - x_3 \left[\exp(\frac{-i}{10}) - \exp(-i) \right] \right\}^2$$

This function has minima of 0 at (1,10,1), and also along the line $\{(a,a,0)\}$.

Initial guess: (0., 10., 20.)

Original		Modified	
Function evaluations	***F(x)***	***Function evaluations***	***F(x)***
17	0.229843	17	0.229843
25	0.265456E-01	25	0.265456E-01
38	0.911284E-02	39	0.122307E-01
46	0.895155E-02	48	0.496437E-02
59	0.892969E-02	62	0.453425E-02
68	0.127223E-02	71	0.434882E-02
85	0.103542E-03	87	0.231319E-02
94	0.103465E-03	96	0.277979E-03
108	0.984160E-04	112	0.102584E-03
117	0.656541E-04	121	0.444863E-05
134	0.656541E-04	137	0.444863E-05
142	0.656541E-04	146	0.444863E-05
157	0.307702E-04	162	0.950726E-08
165	0.307702E-04	171	0.950726E-08
182	0.668862E-06	191	0.691807E-13
191	0.668580E-06	205	0.691807E-13
207	0.668426E-06		

The modified version wins big.

The Chebyquad Function

This is a system of j=1,2,…,n equations defined by Flecher (1965).

$$f(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} T_j(x_i) + a_j$$

where Tj is the jth Chebyshev polynomial shifted to [0,1] and

$$a_j = \begin{cases} -1/(j^2-1) & \textit{if } j \textit{ is even} \\ 0 & \textit{if } j \textit{ is odd} \end{cases}$$

Exercise below uses n=8. The minima is approximately 0.00351687372568. Initial guess is given by (0.11111, 0.22222, 0.33333, 0.44444, 0.55556, 0.66667, 0.77778, 0.88889).

Original		**Modified**	
Function evaluations	***F(x)***	***Function evaluations***	***F(x)***
29	0.121760E-01	29	0.121760E-01
47	0.958968E-02	47	0.958968E-02
65	0.760727E-02	65	0.760727E-02
83	0.659183E-02	83	0.659183E-02
101	0.642240E-02	101	0.642240E-02
119	0.617844E-02	119	0.617844E-02
137	0.606428E-02	137	0.606428E-02
165	0.583154E-02	166	0.600191E-02
184	0.525177E-02	185	0.525552E-02
203	0.501901E-02	204	0.504464E-02
221	0.493539E-02	223	0.475416E-02
239	0.490479E-02	242	0.462059E-02
257	0.486027E-02	261	0.455464E-02
303	0.469713E-02	309	0.453759E-02
321	0.462121E-02	328	0.451527E-02
341	0.431909E-02	349	0.403702E-02
362	0.424728E-02	370	0.376838E-02
382	0.416881E-02	390	0.368161E-02
403	0.372564E-02	409	0.364803E-02
423	0.368207E-02	428	0.364615E-02
453	0.360621E-02	459	0.356874E-02
471	0.357676E-02	478	0.356315E-02
489	0.355752E-02	497	0.355653E-02
507	0.355213E-02	516	0.354215E-02
525	0.355073E-02	535	0.352900E-02
543	0.353919E-02	554	0.351697E-02
561	0.352838E-02	573	0.351695E-02
591	0.352216E-02	604	0.351691E-02
609	0.352180E-02		
628	0.352176E-02		

The Cube function

This function is by Leon (1966):

$$f(\mathbf{x}) = 100(x_2 - x_1^3)^2 + (1 - x_1)^2$$

This function has a valley follows the curve $x_2 = x_1^3$.

Initial guess is chosen to be (-1.2, -1).

Original		**Modified**	
Function evaluations	***F(x)***	***Function evaluations***	***F(x)***
11	4.26783	11	4.26783
21	3.99472	22	3.99520
31	3.58034	32	3.59668
45	3.23347	45	3.56850
61	2.75867	58	3.54281
74	1.20761	72	3.39729
87	0.556541	86	2.66334
101	0.296619	100	2.17454
113	0.394027E-01	114	0.432901
125	0.212529E-01	127	0.295702
138	0.144709E-01	140	0.227055
152	0.741538E-02	154	0.141716
165	0.356590E-02	167	0.107169
179	0.804133E-03	179	0.805482E-01
191	0.232493E-03	192	0.604526E-01
203	0.188945E-04	206	0.418768E-01
220	0.223398E-12	219	0.305640E-01
236	0.514756E-13	233	0.188854E-01
		246	0.122225E-01
		260	0.709208E-02
		273	0.351223E-02
		299	0.184115E-03
		314	0.140896E-04
		327	0.259374E-07
		340	0.163409E-12
		361	0.280125E-15

This time the modified code loss the competition.

The Fletcher-Powell Helix Function

This function is by Fletcher and Powell (1963):

$$f(\mathbf{x}) = 100\left\{(x_3 - 10\theta)^2 + (r-1)^2\right\} + x_3^2$$

where $r = \sqrt{\left(x_1^2 + x_2^2\right)}$

and $2\pi\theta = \begin{cases} \arctan(x_2 / x_1) & if \;\; x_1 > 0 \\ \pi + \arctan(x_2 / x_1) & if \;\; x_1 < 0 \end{cases}$

This function has a helical valley, and an minima at (-1,0,0).

Original		**Modified**	
Function evaluations	***F(x)***	***Function evaluations***	***F(x)***
14	162.447	14	162.447
23	127.230	23	127.230
36	4.80882	37	3.66733
45	3.91830	46	3.33551
58	3.33798	60	2.87663
66	3.18223	69	2.38632
81	2.79723	87	0.622416
90	2.74019	96	0.614781
105	2.63494	112	0.602013
114	1.58545	122	0.248060
129	0.510812	140	0.168417
137	0.508801	149	0.140159
152	0.488789	165	0.137377
161	0.156296	174	0.412875E-01
176	0.754369E-01	190	0.612504E-03
184	0.444695E-01	199	0.222277E-05
199	0.323890E-01	215	0.159882E-07
208	0.709344E-02	224	0.139028E-11
223	0.128360E-03		
231	0.631415E-07		
246	0.765162E-12		
255	0.186690E-14		

This modified code converges faster with less function calls.

The Hilbert Function

The Hilbert function is given by

$$f(\mathbf{x}) = \mathbf{x}^T \mathbf{A}\mathbf{x}$$

where A is an n by n Hilbert matrix, i.e., $a_{ij} = \dfrac{1}{i + j - 1}$ for $1 \le i, j \le n$.

In our case, n=10, and initial guess is $\vec{x} = 1$.

Original		**Modified**	
Function evaluations	***F(x)***	***Function evaluations***	***F(x)***
35	0.165461	35	0.165461
57	0.384653E-03	57	0.384653E-03
79	0.448673E-04	79	0.448673E-04
102	0.281960E-05	102	0.281960E-05
124	0.157652E-05	124	0.157652E-05
147	0.218904E-08	147	0.218904E-08
171	0.218418E-08	171	0.218418E-08
204	0.218417E-08	204	0.218417E-08
230	0.218410E-08	230	0.218398E-08

Both versions give the same numbers except the very last iteration. They have the same performance.

The Powell 3D Function

This function is by Powell (1964):

$$f(\mathbf{x}) = 3 - \left(\frac{1}{1+\left(x_1 - x_2\right)^2}\right) - \sin\left(\pi x_2 x_3 / 2\right) - \exp\left\{\left[-\left(\frac{x_1 + x_2}{x_2}\right) - 2\right]^2\right\}$$

Initial guess is chosen to be (0,1,2). Minima is at zero.

Original		**Modified**	
Function evaluations	***F(x)***	***Function evaluations***	***F(x)***
15	0.118973	15	0.118973
23	0.217345E-02	23	0.217345E-02
36	0.169936E-03	37	0.781542E-04
44	0.243988E-04	46	0.136004E-07
59	0.414322E-05	60	0.302037E-08
68	0.232635E-05	69	0.102141E-12
83	0.273059E-06		
91	0.341949E-13		
107	0.444089E-14		
116	0.888178E-15		

It found the true minimum quickly and did not stop prematurely.

The Rosenbrock Function

This function is by Rosenbrock (1960):

$$f(\mathbf{x}) = 100\left(x_2 - x_1^2\right)^2 + \left(1 - x_1\right)^2$$

This is a well known function with a parabolic valley. Descent methods tend to fall into the valley and then follow it around to the minimum at (1,1).

Initial guess: (-1.2, -1.2)

Original		**Modified**	
Function evaluations	***F(x)***	***Function evaluations***	***F(x)***
11	26.1547	11	26.1547
21	1.51442	21	1.62974
32	1.04519	33	1.09536
46	0.696127	46	0.571557
58	0.661417E-03	58	0.297423
72	0.216762E-06	70	0.192514
85	0.230444E-12	84	0.229561E-01
98	0.126401E-14	96	0.592474E-02
120	0.142739E-16	108	0.491755E-03
		120	0.593373E-04
		132	0.193048E-06
		145	0.984390E-10
		166	0.136615E-10

Like the Cube Function case, the modified version took more function evaluations.

The Powell Singular Function

This function is by Powell (1962):

$$f(\mathbf{x}) = (x_1 - 10x_2)^2 + 5(x_3 - x_4)^2 + (x_2 - 2x_3)^4 + 10(x_1 - x_4)^4$$

This function is difficult to minimize, and provides a severe test of the stopping criterion, because the Hessian Matrix at the minimum (x=0) is doubly singular.

Initial guess: (3,-1,0,1).

Original		**Modified**	
Function evaluations	***F(x)***	***Function evaluations***	***F(x)***
18	11.1916	18	11.1916
29	7.91869	29	7.91869
39	7.62476	39	7.62476
55	6.92464	56	3.20524
65	4.24913	67	0.225039
75	1.60041	79	0.729633E-01
91	1.44355	96	0.410838E-01
101	0.683071	107	0.631386E-02
111	0.307229	118	0.619695E-03
129	0.209772	137	0.139044E-03
139	0.328052E-01	148	0.969917E-04
149	0.279276E-01	159	0.546275E-04
167	0.155675E-01	178	0.281707E-05
177	0.992418E-02	189	0.107281E-05
187	0.700558E-02	200	0.131881E-06
207	0.372934E-02	219	0.941237E-07
217	0.608303E-03	230	0.932410E-07
227	0.267952E-03	241	0.619238E-07
245	0.199287E-03	260	0.556904E-07
255	0.308128E-04	271	0.553072E-07
265	0.326976E-05	282	0.100767E-07
283	0.309868E-05		
294	0.288943E-05		
305	0.736648E-06		
323	0.616313E-07		
334	0.442899E-07		
345	0.115466E-07		

The modified code converges faster.

The Tridiagonal Function

This function is by Gregory and Karney (1969):

$$f(\mathbf{x}) = \mathbf{x}^T A\mathbf{x} - 2x_1$$

where

$$A = \begin{bmatrix} 1 & -1 & & & & \\ -1 & 2 & -1 & & 0 & \\ & -1 & 2 & -1 & & \\ & & -1 & 2 & & \\ & 0 & & & \ddots & \\ & & & & -1 & 2 \end{bmatrix}$$

This function is useful for testing the quadratic convergence property. The minimum $f(\mathbf{x}^*) = -n$ occurs when x* is the first column of A^{-1}, i.e.,

$$\mathbf{x}^* = (n, n-1, n-2, \cdots, 2, 1)$$

In this exercise n=4; then the minimum is expected to find in n^2 or less linear minimization. The initial guess is (0,0,0,0).

Original		**Modified**	
Function evaluations	***F(x)***	***Function evaluations***	***F(x)***
17	-3.66667	17	-3.66667
31	-4.00000	31	-4.00000
54	-4.00000	48	-4.00000
96	-4.00000	68	-4.00000
111	-4.00000	83	-4.00000

This is a simple problem, and the modified code seems to converge better.

The Watson Function

This function can be found in Kowalik and Osborne (1968):

$$f(\mathbf{x}) = x_1^2 + \left(x_2 - x_1^2 - 1\right)^2 + \sum_{i=2}^{30}\left\{\sum_{j=2}^{n}(j-1)x_j\left(\frac{i-1}{29}\right)^{j-2} - \left[\sum_{j=1}^{n}x_j\left(\frac{i-1}{29}\right)^{j-1}\right]^2 - 1\right\}^2$$

The minimization problem is ill-conditioned, and rather difficult to solve. For n=6, the minimum is f(x')=2.28767005355e-3 at

x'=(-0.015725,1.012435,-0.232992,1.260430,-1.513729,0.992996).

Initial guess is (0,0,0,0,0,0).

Original		Modified	
Function evaluations	***F(x)***	***Function evaluations***	***F(x)***
23	1.04255	23	1.04255
37	0.219368	37	0.219368
51	0.124112E-01	51	0.124112E-01
65	0.119368E-01	65	0.119368E-01
80	0.113415E-01	80	0.113415E-01
102	0.113105E-01	103	0.113050E-01
116	0.112895E-01	118	0.112392E-01
130	0.985389E-02	133	0.112090E-01
144	0.974949E-02	148	0.109475E-01
158	0.963899E-02	163	0.109436E-01
180	0.962064E-02	186	0.109012E-01
194	0.956952E-02	201	0.108700E-01
208	0.908306E-02	216	0.108279E-01
222	0.904106E-02	231	0.869696E-02
236	0.868494E-02	246	0.869654E-02
260	0.867298E-02	273	0.863974E-02
274	0.867003E-02	288	0.857291E-02
288	0.708602E-02	303	0.674010E-02
302	0.355012E-02	318	0.659878E-02
316	0.230718E-02	333	
342	0.230125E-02	358	
356	0.229735E-02	373	
370	0.229389E-02	388	
384	0.229357E-02	403	
399	0.229134E-02	418	
423	0.229064E-02	443	
438	0.229061E-02	458	

The Wood Function

This function can be found in Colville (1968):

$$f(\mathbf{x}) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2 + 90(x_4 - x_3^2)^2 + (1 - x_3)^2$$
$$+10.1\left[(x_2 - 1)^2 + (x_4 - 1)^2\right] + 19.8(x_2 - 1)(x_4 - 1)$$

The function rather like Rosenbrock's, but with four variables. one (2nd column) continue the iteration and produce numbers.

Original		**Modified**	
Function evaluations	***F(x)***	***Function evaluations***	***F(x)***
17	275.571	17	275.571
28	98.5192	28	98.5192
39	77.5998	39	77.5998
55	11.6864	56	46.0155
65	8.61427	67	33.7786
…	…	…	…
267	7.55902	268	5.52427
295	7.48411	290	4.99277
305	7.38568	309	4.87755
…	…	…	…
768	2.45252	776	0.164841E-02
778	2.38478	787	0.112361E-02
788	2.21677	798	0.750649E-05
…	…	…	…
1022	0.471395	879	0.818241E-11
1032	0.376731		
…	…		
1146	0.211333		
1164	0.208038		
…	…		
1374	0.274420E-01		
1392	0.250004E-01		
…	…		
1478	0.105187E-01		
1488	0.103745E-01		
…	…		
1637	0.212891E-06		
1648	0.174424E-06		

Conclusion

Modified code gives faster convergence and save function evaluations.