A numerical example for CG_DESCENT conjugate gradient algorithm

Neculai Andrei

Research Institute for Informatics,
8-10, Averescu Avenue, Bucharest, Romania
E-mail: nandrei@ici.ro

In this work I present an unconstrained optimization test function for which the CG_DESCENT algorithm by Hager and Zhang [1] is unable to give a local optimal solution.
The problem is as follows:

$$\min f(x) = \sum_{i=1}^{m}\left(x_i \left(\sum_{j=1}^{n} x_j \right)\right),$$

where $m \le n$. The initial point is $x_0 = [1,1,...1]$. Observe that the function $f(x)$ can be written as:

$$f(x) = (x_1 + x_2 + \cdots + x_m)(x_1 + x_2 + \cdots + x_n).$$

For $m = 10$ and $n = 1000, 2000, \ldots, 10000$, the CG_DESCENT with default settings gives the following results:

CG_DESCENT Algorithm. m=10.

n	iter	nfunc	ngrad	time(c)	f	gnorm	s
1000	2	55	53	0	-0.1584943467908E+78	0.2829126059618E+39	3
2000	2	55	53	0	-0.2501376410966E+79	0.1121130787879E+40	3
3000	2	55	53	0	-0.1260295307017E+80	0.2514461123106E+40	3
4000	2	55	53	2	-0.3973580995694E+80	0.4462902008872E+40	3
5000	2	55	53	1	-0.9687038067406E+80	0.6966454218267E+40	3
6000	2	55	53	0	-0.2006630341308E+81	0.1002512391656E+41	3
7000	2	55	53	2	-0.3714926021979E+81	0.1363890049841E+41	3
8000	2	55	53	1	-0.6334063535750E+81	0.1780779066011E+41	3
9000	2	55	53	2	-0.1014213481831E+82	0.2253178526082E+41	3
10000	2	55	53	3	-0.1545192899102E+82	0.2781090904591E+41	3
TOTAL	20	550	530	0.11(seconds)			

For $m = n/2$ and $n = 1000, 2000, \ldots, 10000$, the CG_DESCENT with default settings gives the following results:

CG_DESCENT Algorithm. m=n/2

n	iter	nfunc	ngrad	time(c)	f	gnorm	s
1000	2	55	53	2	-0.1981409417726E+74	0.3854940787514E+37	3
2000	2	55	53	0	-0.7925637935245E+74	0.7709877974976E+37	3
3000	2	55	53	0	-0.1783268651373E+75	0.1156480643540E+38	3
4000	2	55	53	1	-0.3170255491301E+75	0.1541973434944E+38	3
5000	2	55	53	2	-0.4953523250843E+75	0.1927471992450E+38	3
6000	2	55	53	1	-0.7133072417625E+75	0.2312971218980E+38	3
7000	2	55	53	2	-0.9708903092388E+75	0.2698470435783E+38	3
8000	2	55	53	1	-0.1268101527072E+76	0.3083969661518E+38	3
9000	2	55	53	2	-0.1604940943545E+76	0.3469467432265E+38	3
10000	2	55	53	3	-0.1981408535405E+76	0.3854964817785E+38	3
TOTAL	20	550	530	0.14(seconds)			

For $m = n$ and $n = 1000, 2000, \ldots, 10000$, all is OK:

```
CG_DESCENT Algorithm. m=n.

         n     iter    nfunc    ngrad  time(c)               f                   gnorm         s
 ---------------------------------------------------------------------------------------------------
      1000        2        5        3        0      0.6490858277330E-37   0.5095432573327E-18 0
      2000        2        5        3        0      0.5250599659960E-36   0.1449220433193E-17 0
      3000        2        5        3        0      0.1258114225146E-33   0.2243313821244E-16 0
      4000        2        5        3        0      0.1812770315358E-32   0.8515328097866E-16 0
      5000        2        5        3        1      0.3406574083004E-32   0.1167317280435E-15 0
      6000        2        5        3        0      0.1275991290305E-31   0.2259195689005E-15 0
      7000        2        5        3        0      0.2092766218166E-31   0.2893279259363E-15 0
      8000        2        5        3        0      0.9488472161636E-32   0.1948175778685E-15 0
      9000        2        5        3        0      0.2749947528858E-32   0.1048798842268E-15 0
     10000        2        5        3        0      0.1954040643578E-32   0.8840906386967E-16 0
 ---------------------------------------------------------------------------------------------------
 TOTAL           20       50       30     0.01(seconds)
```

References

[1] Hager, W.W., Zhang, H., (2006) *Algorithm 851: CG_DESCENT, a conjugate gradient method with guaranteed descent*. ACM Transaction on Mathematical Software, 32(1), 113-137.

November 15, 2012