HW5 Solutions
Computational Physics
Spring 2004

Gregg Stiesberg

3/8/04

excPZ and excpPZ

Don't forget about the overall sign when taking derivative of the $r_s > 1$ case.

```
Testing derivative for rs=3.000000 ...
      1.0000000000000000   0.7588903969056273
      0.1000000000000000   0.9690540758177550
      0.0100000000000000   0.9968148961568983
      0.0010000000000000   0.9996805550292700
      0.0001000000000000   0.9999680461262946
      0.0000100000000000   0.9999968045325133
      0.0000010000000000   0.9999996806521011

Testing derivative for rs=0.300000 ...
      1.0000000000000000   0.2335609566625678
      0.1000000000000000   0.7517402404681819
      0.0100000000000000   0.9679926664630071
      0.0010000000000000   0.9967038772653853
      0.0001000000000000   0.9996694024173369
      0.0000100000000000   0.9999669303589755
      0.0000010000000000   0.9999966929316751
```

getVxc and getDepsxc

A naive implementation of the formulas without doing some cancellation analytically will reward you with bogus values for higher N values due to the finite numerical precision. Also worth note: some people were getting nan values for the first call to getVxc(); including the math.h header file fixes this problem.

But when it works,

```
N=    40 <Vxc>=  -0.3309179715 <DExc>=   0.0770872901
```

```
N=   400 <Vxc>=  -0.3316095872 <DExc>=   0.0772489404
N=  4000 <Vxc>=  -0.3316096643 <DExc>=   0.0772489584
N= 40000 <Vxc>=  -0.3316096642 <DExc>=   0.0772489584
```

VWN Form

```
N=    40 <Vxc>=  -0.3308110649 <DExc>=   0.0771876972
N=   400 <Vxc>=  -0.3315026793 <DExc>=   0.0773497587
N=  4000 <Vxc>=  -0.3315027563 <DExc>=   0.0773497767
N= 40000 <Vxc>=  -0.3315027563 <DExc>=   0.0773497767
```

test_atoms

If you are having problems here, and your energy diverges for $\alpha = 0.25$ like this:

```
Etot:   -0.487414864276920
Etot:   -0.802772510609264
Root must be bracketed for bisection in rtbis:
        x1=-1.000000, x2=0.000000, f=1.000000, fmid=16.000000
```

then make sure in getphi() that the initial condition for y[1] is the integral of Rho[], the total charge.

The results:

```
Etot:   -0.480662555813063
Etot:   -0.563296956065424
Etot:   -0.517523050845585
Etot:   -0.487893471693243
Etot:   -0.470577675229894
Etot:   -0.460653981521560
Etot:   -0.454932680476997
Etot:   -0.451569568959120
.
.
.
Etot:   -0.445670519782422
Etot:   -0.445670519387741
Etot:   -0.445670519094143
Etot:   -0.445670518875753
Etot:   -0.445670518713314
Etot:   -0.445670518592505
Etot:   -0.445670518502659
```

NOTE: The specified link for comparison is not valid. Get the raw datafiles from the 'retrieve' link at

`http://physics.nist.gov/PhysRefData/DFTdata/notation.html#retrieve_files`

test_atomsC

for $\alpha = 0.25$,

```
Etot:  -37.425748535952742
```

test_getg.c

A great deal of frustration results when you don't have

```
#include "physics.h"
```

at the top of test_getc.c. The compiler doesn't seem to miss this; but the return values from getc() are garbage without it. Also, why are we using Nmx=40000 to declare arrays, and then only using N=4000 elements? Results are:

```
Iter:   1  Etot:   -111.999999997952  Error:     37.526923194748
Iter:   2  Etot:    -95.821163401420  Error:     21.348086598216
Iter:   3  Etot:    -84.468658331622  Error:      9.995581528418
Iter:   4  Etot:    -78.998916235863  Error:      4.525839432659
Iter:   5  Etot:    -76.495379981061  Error:      2.022303177857
Iter:   6  Etot:    -75.387332052363  Error:      0.914255249159
Iter:   7  Etot:    -74.901367084492  Error:      0.428290281288
Iter:   8  Etot:    -74.682770933856  Error:      0.209694130652

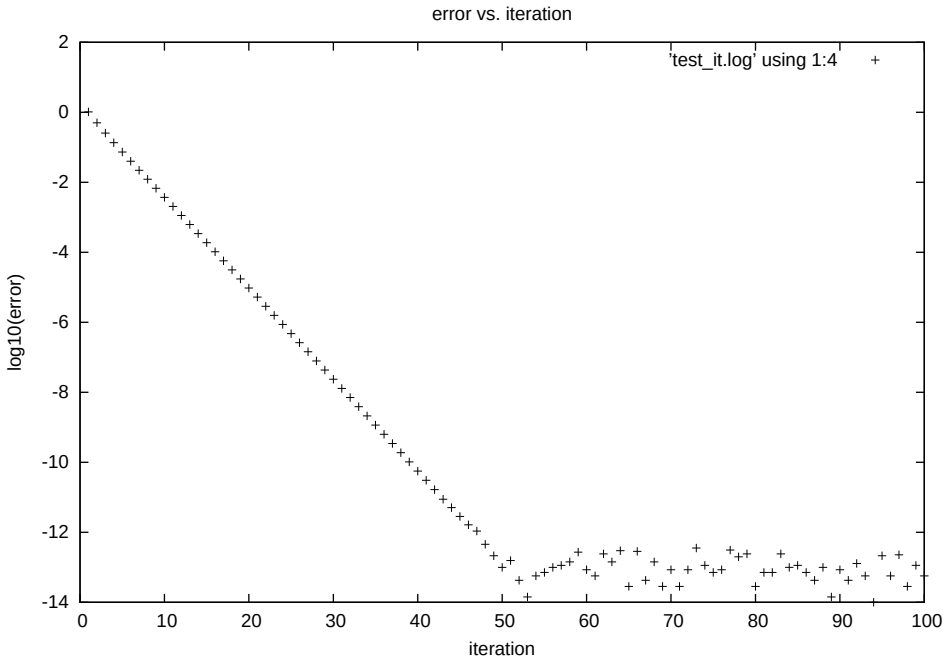
.
.
Iter:  74  Etot:    -74.473076803189  Error:      0.000000000015
Iter:  75  Etot:    -74.473076803194  Error:      0.000000000010
Iter:  76  Etot:    -74.473076803197  Error:      0.000000000006
Iter:  77  Etot:    -74.473076803200  Error:      0.000000000004
Iter:  78  Etot:    -74.473076803202  Error:      0.000000000002
Iter:  79  Etot:    -74.473076803203  Error:      0.000000000001
Iter:  80  Etot:    -74.473076803203  Error:      0.000000000001
```

test_lu.c

This is actually a 4X4 system, not 6X6 as stated in the homework. We can adjust this at will of course.

convergence and runtime

On a PIII 1GHz, time for 50 iterations was 18 seconds.



broyden.c

Results in Gnuplot-ready format: (it, Etot, err, log10(err))

1	-75.500122456620	1.027045653417	0.011589748939
2	-74.580650047499	0.107573244296	-0.968295733314
3	-74.473279967741	0.000203164537	-3.692152097499
4	-74.473077400720	0.000000597516	-6.223650518177
5	-74.473076837394	0.000000034191	-7.466092119822
6	-74.473076803568	0.000000000364	-9.438360162612
7	-74.473076803222	0.000000000018	-10.733102503982
8	-74.473076803204	0.000000000000	-13.069228550159
9	-74.473076803204	0.000000000000	-13.002281760529
10	-74.473076803204	0.000000000000	-13.002281760529
11	-74.473076803204	0.000000000000	-12.525160505809
12	-74.473076803204	0.000000000000	-12.893137291104
13	-74.473076803204	0.000000000000	-13.069228550159
14	-74.473076803204	0.000000000000	-13.847379800543
15	-74.473076803204	0.000000000000	-12.893137291104
16	-74.473076803204	0.000000000000	-13.069228550159
17	-74.473076803204	0.000000000000	-inf
18	-74.473076803204	0.000000000000	-13.370258545823
19	-74.473076803204	0.000000000000	-12.643259817887

20	-74.473076803204	0.000000000000	-13.847379800543
21	-74.473076803204	0.000000000000	-12.568626199590
22	-74.473076803204	0.000000000000	-12.546349804879
23	-74.473076803204	0.000000000000	-12.643259817887
24	-74.473076803204	0.000000000000	-13.546349804879
25	-74.473076803204	0.000000000000	-12.525160505809
26	-74.473076803204	0.000000000000	-13.847379800543
27	-74.473076803204	0.000000000000	-13.370258545823
28	-74.473076803204	0.000000000000	-inf
29	-74.473076803204	0.000000000000	-13.847379800543
30	-74.473076803204	0.000000000000	-13.847379800543

In getg, use Rho in the calculations for Etot, not Rhonew. This makes for a consistent calculation, and as a result the errors are much improved.

Here we see the vast improvement we have made:

