General Interest Seminar

Gaussian16 and NBO7 on Graham and Cedar

Jemmy Hu

SHARCNET/Compute Canada

Feb. 23, 2022

Gaussian on CC Clusters

- A licensed software available on Graham and Cedar
- Software are installed as Module packages
- Basic module commands

```
module avail                    # to list all available module software
module spider gaussian          # to see what gaussian versions are available
```

```
[jemmyhu@gra-login1 ~]$ module spider gaussian

----------------------------------------------------------------------------------------
----------------------------------------------
  gaussian:
----------------------------------------------------------------------------------------
----------------------------------------------
     Versions:
        gaussian/g03.d01
        gaussian/g09.e01
        gaussian/g16.b01
        gaussian/g16.c01
```

```
[jemmyhu@gra-login1 ~]$ module load gaussian/g16.c01
```

Slurm submit/Run script:

- account type (default or RAC account)
- compute resource (cpus, memory, runtime)
- job type (mpi, openmp, serial)
- software (module load, run command)

Several ways to run Gaussian on Graham/Cedar

- by default, Gaussian runs as a serial job on one cpu, good for small molecules
- shared-memory parallel (use multi-cpus/cores on one socket or node, 1 task with multi-threads, cpu ranges from 2 to cpus per node)

```
#!/bin/bash
#SBATCH --account=def-jemmyhu      # PI's group acount
#SBATCH --mem=10G                  # memory amount, roughly 2 times %mem
#SBATCH --time=00-03:00            # time (DD-HH:MM)
#SBATCH --cpus-per-task=16         # No. of cpus as defined by %nprocs
#SBATCH --output=g16_test.log      # output file

module load gaussian/g16.c01
G16 g16_test.com                   # G16 command
or
g16 < g16_test.com >& g16_test.log # g16 command
```

[~$] sbatch g16_test.sh

Gaussian Basics

Latest version: g16.c01

Resource request (mem, cpu, time)

Gaussian files

file location, .chk

Input sample

Restart

Disk: scratch vs local/tmp

Utilities

https://docs.computecanada.ca/wiki/Gaussian

Gaussian Files

input file: .com, .gjf

run script: .sh

output: .log

runtime files (unnamed): .rwf, .inp, .d2e, .int, .skr

G16: /scratch/user/jobid/

g16: /localscratch/username.jobid.0/

User can name the .chk and .rwf files

Interactive job (test)

[~$] salloc --time=2:0:0 --cpus-per-task=16 --mem=10g --account=def-jemmyhu

(max 3 hrs)

[~$] module load gaussian/g16.c01

[~$] G16 g16_test.com

G16 simplest input: CO-b3lyp.com

```
//Simplest Input, serial, default memory

#p b3lyp/6-311++g(2df,p) opt freq

<< CO-6311++g(2df,p)—B3LYP-Opt-Freq >>

0 1
c
o 1 co

co=1.1281
(blank line)
```

```
//Simplest Input, 16 cpus parallel

%mem=5G
%nproc=16
#p b3lyp/6-311++g(2df,p) opt freq

<< CO-6311++g(2df,p)—B3LYP-Opt-Freq >>

0 1
c
o 1 co

co=1.1281
```

Input file with named .rwf: CO-b3lyp-rwf.com

```
//specify a .chk file

%chk=CO-b3lyp
%mem=5G
%nproc=16
#p b3lyp/6-311++g(2df,p) opt freq

<< CO-6311++g(2df,p)—B3LYP-Opt-Freq >>

0 1
c
o 1 co

co=1.1281
```

```
//specify a .rwf file

%rwf=CO-b3lyp
%chk=CO-b3lyp
%nosave
%mem=5G
%nproc=16
#p b3lyp/6-311++g(2df,p) opt freq

<< CO-6311++g(2df,p)—B3LYP-Opt-Freq >>

0 1
c
o 1 co

co=1.1281
```

The named .chk and/or .rwf files will be created in the current job directory

Checkpointing

1. .chk can be used to
   restart an optimization job
   used for property (freq, etc) calculation afterwards
   used for GaussView

2. In G16, .rwf can be used to restart a job,
   particularly to restart a job failed in the freq calculation step

Restart

Restart opt job from .chk

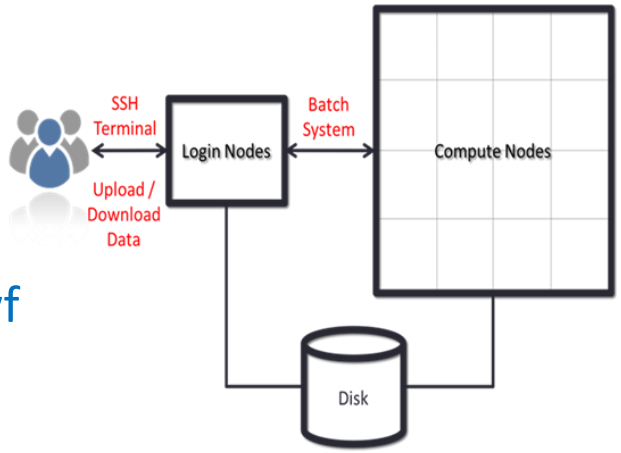
```
%chk=Cu_test.chk
%mem=10g
%nproc=16
#B3LYP/DGDZVP Opt Geom=AllCheck Guess=Read
(one blank line)
```

Restart freq from .rwf

```
%rwf=/scratch/jemmyhu/Cu_test.rwf
%NoSave
%chk=Cu_test.chk
%mem=10g
%nproc=16
#p restart
(one blank line)
```

File structure: scratch (G16) vs localscratch (g16)

	module load gaussian/g16.c01 G16 g16_test.com	**module load gaussian/g16.c01** g16 < g16_test.com >& g16_test.log
Runtime files (.rwf, .inp, .d2e, .int, .skr)	$SCRATCH /scratch/userid/jobid/	$SLURM_TMPDIR (/localscratch/username.jobid.0/)
Disk size	> 1TB	< 1TB (800-900GB?)
Speed		faster
File life time	Files stay when job failed. Clear up jobid folder yourself	removed right after the job is done
For restart	.rwf left for restart	None. Name .chk and .rwf for possible restart



Job location/folder: .com, .log, .sh; named .chk, .rwf

File structure example: scratch (G16) vs localscratch (g16)

```
[jemmyhu@gra766 g16.c01]$ squeue -u jemmyhu
             JOBID     USER          ACCOUNT         NAME  ST  TIME_LEFT NODES CPUS TRES_PER_N MIN_MEM NODELIST
(REASON)
          57611549  jemmyhu     def-jemmyhu_cpu    g16_nbo7.sh   R    2:53:35     1    8       N/A      8G gra766 (None)
          57611551  jemmyhu     def-jemmyhu_cpu    G16_test.sh   R    2:58:30     1    8       N/A      8G gra750 (None)
```

g16 for job 57611549

```
[jemmyhu@gra-login1 ~]$ ssh gra766

[jemmyhu@gra766 /]$ cd /localscratch

[jemmyhu@gra766 localscratch]$ cd jemmyhu.57611549.0
[jemmyhu@gra766 jemmyhu.57611549.0]$ ls
Gau-28440.inp  Gau-28451.d2e  Gau-28451.int  Gau-28451.rwf  Gau-28451.skr
[jemmyhu@gra766 jemmyhu.57611549.0]$ pwd
/localscratch/jemmyhu.57611549.0
```

G16 for job 57611551

```
[jemmyhu@gra766 scratch]$ ls
57611477  57611551
```

Disk error

Erroneous write. write 122880 instead of 4239360.
fd = 3
writwa
writwa: File exists

Erroneous write. write -1 instead of 3648000.

fd = 4
writwa
writwa: No space left on device

Solutions:

1. If you start the job with g16, /tmp does not have enough space, switch to G16

2. If you use G16, one /scratch disk array does not have enough space, configure lfs striping to stripe data over multiple OSTs, e.g., default count is 1, set to be 3:

   lfs setstripe -c 3 /scratch/yourname/

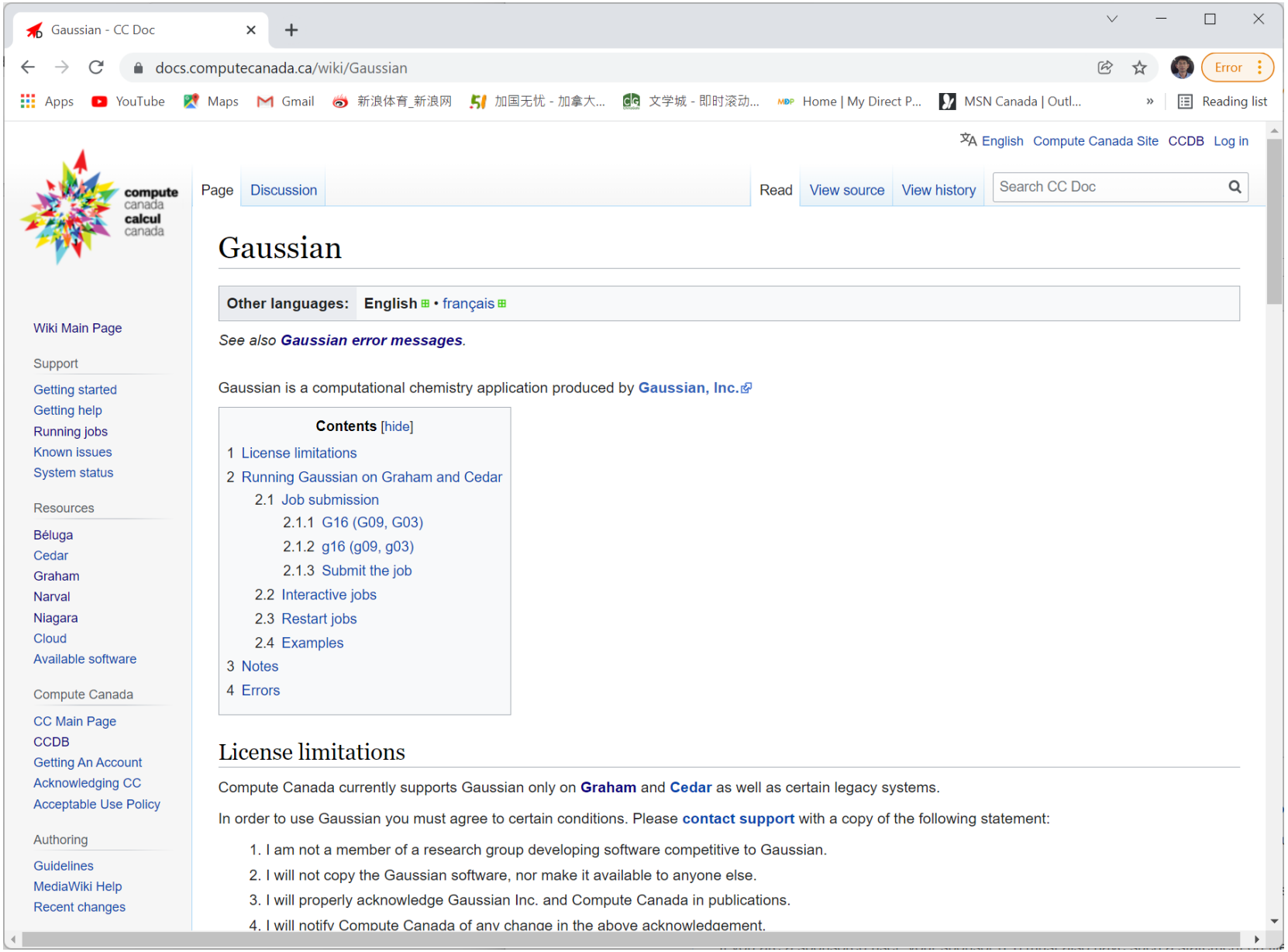
Gaussian Utilities

Utilities

- formchk
- cubegen
- freqchk
- ……

use the same version

direct run from command line



https://docs.computecanada.ca/wiki/Gaussian

Gaussian.com | Expanding the lim

gaussian.com

Apps YouTube Maps Gmail 新浪体育_新浪网 加国无忧 - 加拿大... 文学城 - 即时滚动... Home | My Direct P... MSN Canada | Outl... YouTube Reading list

Expanding the limits of computational chemistry

News Products Support Our People Orders/Contact

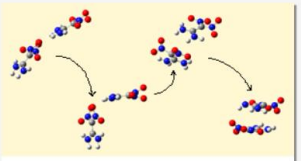
Search Entire Site

Quick Links

Citations • Release Notes
Keywords • FAQ/Tips
Videos: YouTube English • YouTube Japanese • 优酷 Gaussian频道
Computer Requirements
Prices
Sales Agents
Section 508 VPAT

New Chemistry in Gaussian 16

Gaussian 16 expands the range of molecules and types of chemical problems that you can model. More...



GaussView 6 in Action

Become familiar with GaussView 6's wide array of new features through brief video demos. More...

Upcoming Workshops

25 Feb 2022: Online Workshop

Full Workshop Schedule
Inquire About a Workshop
Photos: Hyderabad, Ulm, Shenzen

NBO: Natural Bond 0rbital

The NBO program performs the analysis of a many-electron molecular wavefunction in terms of localized electron-pair bonding units.

The program carries out the determination of
- NAOs: natural atomic orbitals
- NHOs: natural hybrid orbitals
- NBOs: natural bond orbitals
- NLMOs: natural localized molecular orbitals

and uses these to perform
- NPA: natural population analysis (NPA),
- NBO energetic (deletions) analysis,
- NRT: natural resonance theory
- NCS: natural chemical shielding

and other tasks pertaining to localized analysis of wavefunction properties

What NBO version is in Gaussian package?

Gaussian has built-in NBO-3.1, the default NBO in Gaussian when using keyword pop=nbo

What NBO versions are on Graham and Cedar?

NBO is not installed as an independent package, it's built-in with Gaussian package on the clusters

Versions:

gaussian/g03.d01
gaussian/g09.e01 - NBO6
gaussian/g16.b01 - NBO6
gaussian/g16.c01 - NBO7

Recommend to use the latest versions

What's new in NBO7?

NEDA for Gaussian

Natural Energy Decomposition Analysis (NEDA) is implemented for Gaussian 16, Rev. C.01

Improved NRT and $CHOOSE Algorithms

improved "solver" (QPNRT algorithm) for variational optimization of NRT resonance weights and bond orders, a more flexible and robust $CHOOSE algorithm to obtain the density contribution for each idealized resonance bonding pattern

Natural Poly-electron Population Analysis (NPEPA)

2nd-order Reduced Density Matrix Elements (RDM2)

Resonance Natural Bond Orbitals (RNBO)

input

https://gaussian.com/relnotes/

•**[REV C]** NBO version 7 is supported. There are new options to the **Population** keyword:

- **Pop=NPA7**, Natural Population Analysis
- **Pop=NBO7**, full Natural Bond Orbital Analysis
- **Pop=NBO7Read,** full NBO with NBO input read from the input stream
- **Pop=NBO7Delete,** NBO analysis of the effects of deletion of some interactions
- **Pop=NEDA**, Natural Energy Decomposition Analysis

input to the NBO portion consists simply of one or more keywords enclosed in NBO keylists in the ESS input file.
The main NBO keylist (the $NBO keylist) is of the form:

```
$NBO ...(keywords)... $END
```

Simple examples of such $NBO keylists are

```
$NBO dipole nrt $END
$NBO file=myjob archive naonbo ncs=0.05 $END
```

Other keylist identifiers include

$CORE, $CHOOSE, $NRTSTR, $DEL, and $NPEPA

Keylists cannot be nested, and each new keylist must begin on a newline.

The NBO program reads the keywords of each keylist to set various job options, then interrogates the ESS program for information concerning the wavefunction to perform the requested tasks

NBO input is terminated by a blank line.

Thus, keylist input may appear as follows:

```
$nbo  steric  $end
$del
  delete 1 element 3 11
$end
<blank line>
```

Example

NBO analysis with geometry optimization

```
# rhf/3-21g opt pop=nbo7

Methylamine...rhf/3-21g optimization

0 1
C
N   1   1.47
H   1   1.09   2   109.4712
H   1   1.09   2   109.4712   3   120. 0
H   1   1.09   2   109.4712   3   240. 0
H   2   1.01   1   109.4712   3    60. 0
H   2   1.01   1   109.4712   3   300. 0
```

numerical optimization with deletions

```
# rhf/3-21g opt=z-matrix pop=nbo7del

Methylamine...reoptimization with LEWIS deletions

0 1
C
N   1   CN
H   1   CHa   2   NCHa
…
H   2   NH    1   CNH    3  -beta 0

CN        1.4712
CHa       1.0901
…
alfa     121.4224
beta      64.2403

$nbo  print=0  nbosum  $end      ! Turn off all printing but NBO Summary
$choose                          ! Fix Lewis structure for all geometries
   lone 2 1 end
   bond s 1 2 s 1 3 s 1 4 s 1 5 s 2 6 s 2 7 end
$end
$del  lewis  $end                ! Perform deletions analysis
```

Output

```
********************************** NBO 7.0 **********************************
      N A T U R A L  A T O M I C  O R B I T A L  A N D
     N A T U R A L  B O N D  O R B I T A L  A N A L Y S I S
****************************** Jemmy Hu (S102210) ******************************
 (c) Copyright 1996-2021 Board of Regents of the University of Wisconsin System
    on behalf of the Theoretical Chemistry Institute.  All rights reserved.

       Cite this program [NBO 7.0.10 (8-Feb-2021)] as:

       NBO 7.0.  E. D. Glendening, J. K. Badenhoop, A. E. Reed,
       J. E. Carpenter, J. A. Bohmann, C. M. Morales, P. Karafiloglou,
       C. R. Landis, and F. Weinhold, Theoretical Chemistry Institute,
       University of Wisconsin, Madison, WI (2018)

Filename set to /scratch/jemmyhu/57734053/Gau-2187

Job title: Methylamine...rhf/3-21g optimization


NATURAL POPULATIONS:  Natural atomic orbital occupancies

 NAO Atom No lang   Type(AO)    Occupancy      Energy
-----------------------------------------------------
  1    C  1  s     Cor( 1s)     1.99978     -11.17840
  2    C  1  s     Val( 2s)     1.08927      -0.19713
  3    C  1  s     Ryd( 3s)     0.00071       1.95478
  4    C  1  px    Val( 2p)     1.21779      -0.06924
```

NATURAL BOND ORBITALS (Summary):

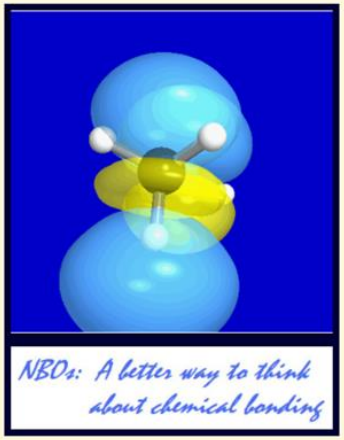
NBO	Occupancy	Energy	Principal Delocalizations (geminal,vicinal,remote)
Molecular unit 1 (CH5N)			
------ Lewis ----------			
1. CR (1) C 1	1.99978	-11.17840	18(g)
2. CR (1) N 2	1.99984	-15.42894	22(g)
3. LP (1) N 2	1.97257	-0.40269	11(v),12(v),13(v)
4. BD (1) C 1- N 2	1.99876	-0.80619	
5. BD (1) C 1- H 3	1.99852	-0.65507	20(v)
6. BD (1) C 1- H 4	1.99370	-0.65295	14(v)
7. BD (1) C 1- H 5	1.99370	-0.65295	15(v)
8. BD (1) N 2- H 6	1.99539	-0.77276	12(v)
9. BD (1) N 2- H 7	1.99539	-0.77276	13(v)
------ non-Lewis ----------			
10. BD*(1) C 1- N 2	0.00009	0.57358	

Examples on Graham in
/home/jemmyhu/tests/test_Gaussian/g16/g16.c01/NBO7

Natural Bond Orbital 7.0 Homepage

Site Editors: Eric Glendening, Clark Landis, Frank Weinhold

"It is nice to know that the computer understands the problem, but I would like to understand it too." - Eugene Wigner



NATURAL BOND ORBITALS

Indispensable tools for the modern computational chemist...

INTRODUCTORY LINKS

NBO INTRO SLIDESHOW

WHAT ARE NBOS?

NBO KEYWORDS/OUTPUT

NBO TUTORIALS

WHAT'S NEW IN NBO 7?

NBO 7.0 PROGRAM

The Natural Bond Orbital (NBO) program *NBO 7.0* is a discovery tool for

NEWS...

[Check here for latest bugfix and update announcements]

February 8, 2021:

New release ***NBO 7.0.10*** (8-Feb-2021) is now available for download by all licensed users. This release includes a parallel implementation of NRT for Linux and MacOS (see the *PARALLEL.NRT* file in the nbo7 root directory for details) and bug fixes for gfortran 6-10. By default, the NRT algorithm now cycles to convergence (NRTCYC=0) and eliminates resonance forms that do not contribute to the resonance hybrid for six consecutive cycles (NRTCULL=6).

May 21, 2020:

New release ***NBO 7.0.9*** (21-May-2020) is now available for download by all licensed *NBO7* and *NBOPro7@Jmol* users. The update includes fixes for NLMO and RNBO analyses, T_h point-group symmetry, new NRTPR keyword, and new NBOMEM environment variable for dynamic memory allocation for GenNBO G09NBO and G16NBO interface programs (making source-code edits largely unnecessary for large systems that exceed default memory limits of the compiled binary program).

February 21, 2020:

The e-order form for *NBO 7.0* products has been revised to include a link to the current *License Agreement*. In addition, the ordering process for a *Site License* now requires identification and title of the institutional officer with appropriate signatory authority, as well as the "Responsible Individual" (if different from the "Authorized Company or Institutional Signatory") who directly receives the downloaded program