# *Gamess+DF*

## A Module Incorporating

## Dual-Functional Tamm-Dancoff Approximation,

## MN12-L, MN15-L, MN15 functionals

## into *GAMESS*

# Users Manual

Version 2017
Date of finalization of this version of the software: April 10, 2017
Date of most recent change in this document: April 10, 2017 15:13

Yinan Shu, Aleksandr V. Marenich, Kelsey Parker, and Donald G. Truhlar

*Department of Chemistry, Chemical theory center, and Minnesota Supercomputing Institute, University of Minnesota, Minneapolis, MN 55455-0431, U. S. A.*

Distribution site: *http://comp.chem.umn.edu/gamess+df*

## Contents



## Introduction of *Gamess+DF*

*Gamess+DF* is a module that incorporates the dual-functional Tamm-Dancoff Approximation and new Minnesota functionals into *GAMESS*. The current version of *Gamess+DF* (version 2017) has been developed to work with the latest version (R1) of GAMESS (version of April 1, 2017)

To use *Gamess+DF,* the user needs to obtain the *GAMESS* package from Iowa State University (April 1, 2017 R1 version of *GAMESS*) and *Gamess+DF* (version 2017) from the University of Minnesota.

### Dual-Functional Tamm-Dancoff Approximation

Abbreviations:
LR-TDDFT full linear-response time-dependent Kohn-Sham density functional theory
KS-TDA: Tamm–Dancoff approximation to LR-TDDFT
DF-TDA: dual-functional modification of KS-TDA

The potential energy surfaces (PESs) computed by KS-TDA are known to have incorrect topology near conical intersection (CI) geometries. This is due to the lack of coupling between ground-state KS determinant and TDA singly excited determinants. DF-TDA employs two different functionals: F1 for orbital optimization and F2 for TDA Hamiltonian construction, and hence the coupling is introduced. In many cases, DF-TDA is as accurate as KS-TDA with F2, with 0.1 eV higher excitation energies in a typical case we examined. In addition, DF-TDA predicts the correct dimensionality of PESs near CI seams. Therefore DF-TDA can be used in understanding and modeling electronically nonadiabatic processes involving ground and excited states.

***Gamess+DF* citation:**

Publications including work performed with *Gamess+DF* should cite the software package in one of the following ways:

**Journal of Chemical Physics or World Scientific style**

Y. Shu, A. V. Marenich, K. Parker, and D. G. Truhlar, *Gamess+DF* – version 2017*,* University of Minnesota, Minneapolis, 2017, based on the General Atomic and Molecular Electronic Structure System (GAMESS) as described in M. W. Schmidt, K. K. Baldridge, J. A. Boatz, S. T. Elbert, M. S. Gordon, J. H. Jensen, S. Koseki, N. Matsunaga, K. A. Nguyen, S. J. Su, T. L. Windus, M. Dupuis, and J. A. Montgomery, J. Comput. Chem. **14**, 1347 (1993)

**Elsevier style**

Y. Shu, A. V. Marenich, K. Parker, D. G. Truhlar, *Gamess+DF* – version 2017*,* University of Minnesota, Minneapolis, 2017, based on the General Atomic and Molecular Electronic Structure System (GAMESS) as described in M. W. Schmidt, K. K. Baldridge, J. A. Boatz, S. T. Elbert, M. S. Gordon, J. H. Jensen, S. Koseki, N. Matsunaga, K. A. Nguyen, S. J. Su, T. L. Windus, M. Dupuis, J. A. Montgomery, J. Comput. Chem*.* 14 (1993) 1347.

**ACS style**

Shu, Y.; Marenich, A. V.; Parker, K.; Truhlar, D. G. *Gamess+DF* – version 2017*,* University of Minnesota, Minneapolis, 2017, based on the General Atomic and Molecular Electronic Structure System (GAMESS) as described in Schmidt, M. W.; Baldridge, K. K.; Boatz, J. A.; Elbert, S. T.; Gordon, M. S.; Jensen, J. H.; Koseki, S.; Matsunaga, N.; Nguyen, K. A.; Su, S. J.; Windus, T. L.; Dupuis, M.; Montgomery J. A. *J. Comput. Chem*. **1993**, *14*, 1347.

**Theoretical Chemistry Accounts style**

Shu Y, Marenich AV, Parker K, Truhlar DG (2009) *Gamess+DF* – version 2017*,* University of Minnesota, Minneapolis, 2017, based on the General Atomic and Molecular Electronic Structure System (GAMESS) as described in Schmidt MW, Baldridge KK, Boatz JA, Elbert ST, Gordon MS, Jensen JH, Koseki S, Matsunaga N, Nguyen KA, Su SJ, Windus TL, Dupuis M, Montgomery JA (1993) J. Comput. Chem. 14: 1347

In addition, as usual, the user should give literature references for any methods used: the reference for DF-TDA is

Shu, Y.; Parker, K.; Truhlar, D. G. "Dual-Functional Tamm-Dancoff Approximation: A Simple Linear Response Theory to Correctly Describe the Potential Energy Surfaces In the Vicinity of $S_1/S_0$ Conical Intersection" *J. Phys. Chem. Lett.* **2017**, to be published.

## Introduction to the *Gamess+DF* Package

The user will find the following folders in the *Gamess+DF* package.

1) source

The source folder has eight files,

The following three source files are incorporate DF-TDA module,

"gamess.src", "tddft.src" and "utddft.src". These files are similar to that of gamess.src, tddft.src and utddft.src in GAMESS source file. The three modifed source files have DF-TDA capability implemented.

The following four souce files are incorporate MN12-L, MN15-L, and MN15 functionals,

"dftxca.src", "dftxcc.src", "tddfun.src", "tddxce.src". These files are similar to that of GAMESS source files except for tddxce.src. tddxce.src is a complete new source file that is not in 2016 R1 version of GAMESS (as downloaded at April 1, 2017). These four source files have new functionals implemented.

The last one file is the lked file,

lked file is used to link the objects files, due to the tddxce.src file is a complete new file, we need to include tddxce.o to be linked, hence we modified lked file as well.

2) test_suites

The test_suites folder has a couple test examples. We will give a detailed introduction of each files in the test example in the "How to Run DF-TDA Calculation".

3) utility

The utility folder contains the script and external program required for running a DF-TDA calculation.

Remember the DF-TDA module and new functionals module are independent modules, the users can use either of the module or both.

## Compilation of *Gamess+DF*

The user needs to put the whole GAMESS+DF folder in the GAMESS directory. For example, suppose your GAMESS directory is at /home/David/software/GAMESS. Then the Gamess+DF directory should be at /home/David/software/GAMESS//GAMESS+DF. The reason is, we have set up a script (dftda_assist, used to run DF-TDA calculation, see introduction bellow) uses the utility files in that address.
The *Gamess+DF* source folder contains eight sources files:
DF-TDA module: "gamess.src", "tddft.src", "utddft.src".
These three source files correspond to the TDDFT and UTDDFT modules in GAMESS with additions of DF-TDA capability. Compilation of *Gamess+DF* DF-TDA module requires downloading the original version of GAMESS from Iowa State University and replacing "gamess.src", "tddft.src" and "utddft.src" in folder "source" with *Gamess+DF* "gamess.src", "tddft.src" and "utddft.src".

After the replacement of "gamess.src", "tddft.src" and "utddft.src":
1) If you have not compiled GAMESS yet, you can follow up the normal procedure of compiling GAMESS.
2) If you already complied GAMESS, you can just re-compile the gamess, tddft, and utddft modules and link the object files by the following procedure (which supposes you are at the top directory of GAMESS)

```
./comp gamess
./comp tddft
./comp utddft
./lked gamess 00 > lked.log       # 00 is the version number
```

The executable will be named: "gamess.00.x"

MN12-L, MN15-L, MN15 functionals modules:"dftxca.src", "dftxcc.src", "tddfun.src", "tddxce.src".
Compilation of *Gamess+DF* new functionals module requires downloading the original version of GAMESS from Iowa State University and replacing "dftxca.src", "dftxcc.src", "tddfun.src" in folder "source" with *Gamess+DF* "dftxca.src", "dftxcc.src", "tddfun.src". Also, the user needs to add new source file "tddxce.src" in folder "source".

Due to "tddxce.src" is a complete new source file, we also need to modify lked file such that includes tddxce.o obeject file to be linked. In *Gamess+DF* source folder, we have a modified lked file available. The user needs to replace the original lked file in top directory of GAMESS folder with this new lked file. Just a reminder, the original lked file is at, for example,
/home/David/software/gamess/

After the replacement of these new source files and lked file.
1) If you have not compiled GAMESS yet, you can follow up the normal procedure of compiling GAMESS.

2) If you already complied GAMESS, you can just re-compile the new functional modules and link the object files by the following procedure (which supposes you are at the top directory of GAMESS)

```
./comp dftxca
./comp dftxcc
./comp tddfun
./comp tddxce
./lked gamess 00 > lked.log        # 00 is the version number
```

The executable will be named: "gamess.00.x"

## Calling New Functionals

The keywords of calling new functionals are
DFTTYP=MN12-L for MN12-L functional
DFTTYP=MN15-L for MN15-L functional
DFTTYP=MN15 for MN15 functional

## How to Run a DF-TDA Calculation

The user will find a folder named "test_suites"; in that folder, we have examples of input and output files. Running a DF-TDA calculation can be easily done by executing the script called "dftda_assist" with input variables. This script will run the steps introduced in the following "General Computational Framework" subsection. Introduction of "dftda_assist" will be given in the "Step-by-step Introduction of Running a DF-TDA Calculation" subsection. Please note that the current version of DF-TDA algorithm uses Davidson diagonalizer, which is relatively slow for converging the DF-TDA Hamiltonian, hence we have set the convergence threshold for norm of the residual vector to 1.0E-4.

There is a keyword for the $TDDFT card that can sometimes accelerate the convergence of the Davidson diagonalizer. This keyword is called "OIGG", which stands for "orbital initial gap of ground state"; this corresponds to the initial guess value of the first diagonal element of the TDA configuration interaction Hamiltonian in hartrees. The default is set as -0.003, and this value will be used if the keyword is not mentioned. The diagonal element will affect the preconditioner in Davidson's method. Users can modify this value if they experience slow convergence. However, we believe the default should be good enough for most calcualtions. We are working on a better version of the diagonalizer, and we plan to update the package in a later version of *Gamess+DF*.

**General computational framework**
The current DF-TDA calculation is a three-step calculation (actually four as described next).
Step 1 (denoted as "orb step"): Orbital optimization with F1.

This is a regular KS-DFT calculation with functional F1 in GAMESS.
We suggest using tighter (1.0D-6) SCF convergence than the default by adding the following scf card in GAMESS KS-DFT input:

```
$SCF CONV=1.0D-06 $END
```

We also suggest specifying the grids used in KS-DFT calculation. See the GAMESS manual for grid information.

Step 2 (denoted as "fock step"): Computing fock matrix with F2.

This is a regular KS-DFT calculation in GAMESS with functional F2, but with only one iteration with the optimized orbital from the orb step. Before this step, we need to create the $VEC card from the orb step calculation, and set the following guess card to read the molecular orbital coefficients

```
$GUESS GUESS=MOREAD NORB=XXX $END
```

Please note that the default scf calculation performed by GAMESS will use a smaller grid as the initial guess, and upon convergence it will switch to the grid you requested. Hence it is important that this one iteration calculation is performed with the same grid you used in orb step. Suppose you use NRAD=99 NLEB=590 (which is also the default for meta-GGA functionals in GAMESS) in the orb step, then you need to include the following card so that the initial grid for the SCF calculation will be the same:

```
$DFT NRAD0=99 NLEB0=590 NRAD=99 NLEB=590 $END
```

Intermediate step: Generating coupling terms of Hamiltonian.

In this intermediate step between the fock and dftda steps, the user should use an external script and program to get the coupling terms in the TDA Hamiltonian; the script is called "mntda_dftda", the program is named as mntda.exe. Calling the mntda_dftda script is the only thing one needs to do, as it is designed to call the program mntda.exe and generate FIA files; these files contain the coupling terms of the Hamiltonian.

Step 3 (denoted as "dftda step"): Constructing and diagonalizing the DF-TDA Hamiltonian.

This is a TDDFT calculation with functional F2. Again, we only include one iteration for the SCF such that the orbitals will not be re-optimized. The dftda calculation is initiated by calling the keyword DFTDA=.TRUE. in TDDFT card, for example,

```
$TDDFT NSTATE=10 MULT=1 DFTDA=.TRUE. $END
```

In addition, when DFTDA=.TRUE. the Tamm-Dancoff approximation will be turned on automatically.

**Step-by-step introduction to running a DF-TDA calculation**

Choice one:

The user can run the three steps of the calculation one-by-one manually. See the next section for sample input files. For the intermediate step, the user needs to run the mntda_dftda script with input variables generated by steps 1 and 2. For example, if your molecule is ethylene, you would run:

```
mntda_dftda ethylene_orb.log ethylene_fock.inp ethylene_fock.log all gamess_executable_address
```

where the gamess_executable_address is where your GAMESS executable script is, for example,

```
/home/David/software/gamess
```

In this example, the final step would be to run ethylene_dftda.inp, which is generated by the mntda_dftda script.

Choice two:

The user can use the dftda_assist script. The dftda_assist script requires three input variables: molecule name, scratch address, and the address of the user's GAMESS executable script (which is where rungms is). For example:

```
dftda_assist ethylene /scratch/David/ /home/David/software/gamess
```

Remember that the molecule name should be the same as your xyz file (the xyz file should be in the standard GAMESS format).

We recommend that the user reads the dftda_assist script and makes modifications if necessary. Within the dftda_assist script, the calculation is broken down into the three steps. At each of the three

steps, we have a command line to execute the GAMESS program; this could change when the rungms script is different. For example, when we execute GAMESS, we have the following command,

```
/home/David/software/gamess/rungms ethylene_orb.inp ${scratch} 1 1 > ethylene_orb.log
```

but this command can change when the rungms script has been set differently because the rungms script requires different inputs, for example, version number of GAMESS, scratch address, etc., and also the order of these input variables can change for different rungms scripts.

The default number of states requested at the dftda step is 10 (nine excited states plus one ground state). The user can modify the number of states in the dftda_assist script by changing the value at NSTATE=10.

When running the dftda_assist script, one needs "head", "middle" and "tail" *partial* input files, which will be used to generate the full input files. The user can find these in test_suites. Let's introduce each of the partial input files with the test_suites example for acetone. In the test_suites/acetone/, we can find the following files:

- acetone.xyz : this is the acetone molecule xyz coordinate in standard GAMESS format
- head_orbital.tmp and tail_orbital.tmp: these two files along with the xyz file are put together at the orb step to generate acetone_orb.inp. The user should modify head_orbital.tmp according to the specifications needed in the user's GAMESS calculation; head_orbital.tmp is where you specify F1.
- head_fock.tmp, middle2_fock.tmp and tail_fock.tmp: these three files along with the xyz file and the middle_fock.tmp file (which is automatically generated by the dftda_assist script) are put together to create acetone_fock.inp during the fock step. Similar to modifying head_orbital.tmp, the user should modify head_fock.tmp according to their needs; head_fock.tmp is the first place where you will specify F2.
- head_dftda.tmp and tail_dftda.tmp: these two files along with the xyz file and the middle_dftda.tmp file (which is automatically generated by the dftda_assist script) are put together to create acetone_dftda.inp during the dftda step. The user should modify head_dftda.tmp to fit their needs. The user needs to specify F2 in this file.

In summary, if the user choses to use the dftda_assist script, they will need the *.tmp files listed above with the specifications for their calculation. Once you have all these files in your directory, you can run a DF-TDA calculation by executing the dftda_assist script.

**Sample Input Files**

The three calculation input files (corresponding to the three steps of the DF-TDA process) are called:

```
{molecule name}_orb.inp
{molecule name}_fock.inp
```

and

```
{molecule name}_dftda.inp
```

The following is an example of a three-step calculation for ethylene, in particular a DF-TDA/M11:MN15 calculation. We set the grid as NRAD=120 and NLEB=2030, and we use the 6-311+G(2d,p) basis set. The $VEC card data is removed from the ethylene_fock.inp and ethylene_dftda.inp to save space.

ethylene_orb.inp:

```
 $CONTRL SCFTYP=RHF RUNTYP=ENERGY EXETYP=RUN UNITS=ANGS
  MULT=1 NOSYM=1 ICHARG=0 DFTTYP=M11 COORD=UNIQUE MAXIT=500
  ISPHER=0 $END
 $SYSTEM MWORDS=1000 $END
 $SCF CONV=1.0D-06 $END
 $DFT NRAD=120 NLEB=2030 $END
 $BASIS GBASIS=N311 NGAUSS=6 NDFUNC=2 NPFUNC=1 DIFFSP=.TRUE. $END
 $DATA
STEP 1: Calculation of M11 DFT orbitals
C1
H 1.0  0.000000  0.923274  1.238289
H 1.0  0.000000 -0.923274  1.238289
H 1.0  0.000000  0.923274 -1.238289
H 1.0  0.000000 -0.923274 -1.238289
C 6.0  0.000000  0.000000  0.668188
C 6.0  0.000000  0.000000 -0.668188
 $END
```

ethylene_fock.inp (we didn't show $VEC card here explicitly):

```
$CONTRL SCFTYP=RHF RUNTYP=ENERGY EXETYP=RUN UNITS=ANGS
  MULT=1 NOSYM=1 ICHARG=0 DFTTYP=MN15 COORD=UNIQUE
  MAXIT=1 NPRINT=5 ISPHER=0 $END
 $SYSTEM MWORDS=1000 $END
 $DFT NRAD0=120 NLEB0=2030 NRAD=120 NLEB=2030 $END
 $BASIS GBASIS=N311 NGAUSS=6 NDFUNC=2 NPFUNC=1 DIFFSP=.TRUE. $END
 $GUESS GUESS=MOREAD NORB=82 $END
 $DATA
STEP 2: Calculation of the MN15 Fock matrix using M11 DFT orbitals
C1
H 1.0  0.000000  0.923274  1.238289
H 1.0  0.000000 -0.923274  1.238289
H 1.0  0.000000  0.923274 -1.238289
H 1.0  0.000000 -0.923274 -1.238289
C 6.0  0.000000  0.000000  0.668188
C 6.0  0.000000  0.000000 -0.668188
 $END
 $VEC
```

```
……
 $END
```

ethylene_dftda.inp (we didn't show $VEC card here explicitly):

```
$CONTRL SCFTYP=RHF RUNTYP=ENERGY EXETYP=RUN UNITS=ANGS
  MULT=1 NOSYM=1 ICHARG=0 COORD=UNIQUE TDDFT=EXCITE
  MAXIT=1 DFTTYP=MN15 ISPHER=0 $END
 $SYSTEM MWORDS=2000 $END
 $DFT NRAD0=120 NLEB0=2030 NRAD=120 NLEB=2030 $END
 $BASIS GBASIS=N311 NGAUSS=6 NDFUNC=2 NPFUNC=1 DIFFSP=.TRUE. $END
$GUESS GUESS=MOREAD NORB=82 $END
 $TDDFT NSTATE=10 MULT=1 DFTDA=.T. $END
 $DATA
STEP 3: DF-TDA
C1
H 1.0  0.000000  0.923274  1.238289
H 1.0  0.000000 -0.923274  1.238289
H 1.0  0.000000  0.923274 -1.238289
H 1.0  0.000000 -0.923274 -1.238289
C 6.0  0.000000  0.000000  0.668188
C 6.0  0.000000  0.000000 -0.668188
 $END
 $VEC
……
 $END
```

**How to read the output**

The DF-TDA energies are shown in file {molecule name}_dftda.log. The output is very similar to that of a regular TDDFT calculation.

At the end of {molecule name}_dftda.log file, you will see, for example, the following output,

```
                    SUMMARY OF TDDFT RESULTS

 STATE          ENERGY    EXCITATION  TRANSITION DIPOLE, A.U.  OSCILLATOR
               HARTREE        EV       X      Y      Z    STRENGTH
 0    A       0.0000000000   0.029 <-- OLD GROUND STATE
 1    A      -0.0010664958   0.000   -0.0003 -0.0001 -0.0032   -0.000
 2    A       0.2752792109   7.520    0.2630 -0.0000 -0.0000    0.013
 3    A       0.2941603974   8.034    0.0001  0.0000  0.0000    0.000
 4    A       0.2983217687   8.147   -0.0001 -0.0000 -0.0000    0.000
 5    A       0.3017944013   8.241   -0.0002 -0.0000 -0.0000    0.000
 6    A       0.3037963454   8.296   -0.0000 -0.0000  1.6185    0.531
 7    A       0.3470002608   9.471   -0.0000 -0.0000 -0.0000    0.000
 8    A       0.3639436026   9.932   -0.0000  0.0005  0.0001    0.000
 9    A       0.3668716864  10.012   -0.0000 -0.0000 -0.0000    0.000
10    A       0.3927884318  10.717    0.0000 -0.9539  0.0009    0.238
```

The STATE0, will always be "`0.0000000000`" because it is zeroed-out. STATE1 is our new, coupled ground state; hence when you read this output, STATE1 will be our DF-TDA ground state. STATE2 is our DF-TDA first excited state. The energies shown here are all relative energies. They are relative to the non-coupled, one iteration SCF energy of functional F2 with F1 KS-DFT optimized orbitals. This energy is read from the end of SCF calculation in {molecule name}_dftda.log file. For example, you will see the following output:

```
     -------------------------
         R-M06 SCF CALCULATION
     -------------------------

 NUCLEAR ENERGY =       33.2981002202
 MAXIT =    1    NPUNCH=    2
 EXTRAP=T  DAMP=F  SHIFT=F  RSTRCT=F  DIIS=T  DEM=F  SOSCF=F
    DENSITY MATRIX CONVERGENCE THRESHOLD=  1.00E-06
 COARSE -> FINE DFT GRID SWITCH THRESHOLD=  3.00E-04 (SWITCH IN $DFT)
            HF -> DFT SWITCH THRESHOLD=  0.00E+00 (SWOFF IN $DFT)
 MEMORY REQUIRED FOR RHF ITERS=   1638465 WORDS.

 EXCHANGE FUNCTIONAL   =M06
 CORRELATION FUNCTIONAL=M06
```

```
    DFT THRESHOLD        =.684E-09
 FOR AN EULER-MACLAURIN QUADRATURE USING 120 RADIAL POINTS:
 SMALLEST GAUSSIAN PRIMITIVE EXPONENT=       0.1027410000 OF TYPE -S-
 ON ATOM NUMBER   1 HAS RADIAL NORMALIZATION=  1.000000
  LARGEST GAUSSIAN PRIMITIVE EXPONENT=    4563.2400000000 OF TYPE -S-
 ON ATOM NUMBER   5 HAS RADIAL NORMALIZATION=  1.000000

 ITER EX DEM    TOTAL ENERGY       E CHANGE  DENSITY CHANGE    DIIS ERROR
   1  0  0     -78.5389572905   -78.5389572905   0.033075663   0.000000000

 SCF IS UNCONVERGED, TOO MANY ITERATIONS
    TIME TO FORM FOCK OPERATORS=      9.2 SECONDS (      9.2 SEC/ITER)
    TIME TO SOLVE SCF EQUATIONS=      0.0 SECONDS (      0.0 SEC/ITER)

 FINAL R-M06 ENERGY IS       0.0000000000 AFTER   1 ITERATIONS
 DFT EXCHANGE + CORRELATION ENERGY =       -9.0511790234
 TOTAL ELECTRON NUMBER           =       15.9999999974
 TOTAL KINETIC ENERGY DENSITY     =       78.1182321874
```

You can see that there is only 1 iteration, and the total energy nearby is the reference energy, which is "-78.5389572905" in our case.

Hence, from the TDDFT calculation, we know that the DF-TDA ground state has an excitation energy of -0.0010664958 hartrees, and the first excited state has an excitation energy of 0.2752792109 hartrees. Adding these two excitation energies to the reference, the absolute DF-TDA energies for $S_0$ and $S_1$ are:

$S_0$ -78.5389572905 + (-0.0010664958) = -78.5400237863

$S_1$ -78.5389572905 + (0.2752792109) = -78.2636780796