

***GESOL:***  
***Gaussian External Solvation Module Containing***  
***The SMD Solvation Model***

**Users Manual**

Version 2008

Date of finalization of this version of the software: November 23, 2008

Date of most recent change in this document: March 12, 2009

Aleksandr V. Marenich,<sup>a</sup> Gregory D. Hawkins,<sup>a</sup> Daniel A. Liotard,<sup>b</sup>  
Christopher J. Cramer,<sup>a</sup> and Donald G. Truhlar<sup>a</sup>

<sup>a</sup> *Department of Chemistry and Supercomputer Institute, University of Minnesota, Minneapolis,  
MN 55455-0431, U. S. A.*

<sup>b</sup> *Laboratoire de Physico-Chimie Theorique, Universite de Bordeaux 1, 351 Cours de la  
Liberation, 33405 Talence Cedex, France*

Distribution site: <http://comp.chem.umn.edu/gesol>

The code and manual are copyrighted, 2008.

## Contents

<b>1. User agreement.....</b>	<b>3</b>
<b>2. Executive summary.....</b>	<b>4</b>
<b>3. Theory .....</b>	<b>5</b>
<b>3.1. Introduction to the SMD model.....</b>	<b>5</b>
<b>3.2. Applicability of the SMD model implemented in <i>GESOL</i>.....</b>	<b>6</b>
<b>3.3. A note on Poisson solvers .....</b>	<b>7</b>
<b>4. Description of <i>GESOL</i> subprograms.....</b>	<b>9</b>
<b>5. Citation of the <i>GESOL</i> program and the methods used by <i>GESOL</i> .....</b>	<b>12</b>
<b>6. <i>GESOL</i> input description .....</b>	<b>14</b>
<b>7. <i>GESOL</i> output description.....</b>	<b>18</b>
<b>8. <i>GESOL</i> installation and execution.....</b>	<b>20</b>
<b>9. <i>GESOL</i> test suite .....</b>	<b>22</b>
<b>10. Platforms on which <i>GESOL</i> has been tested .....</b>	<b>27</b>
<b>11. Revision history.....</b>	<b>28</b>

## 1. User agreement

*GESOL* is a licensed program, and the use of this program implies acceptance of the terms of the license, which are repeated here for convenience:

- A. No user or site will redistribute the source code or executable code to a third party in original or modified form without written permission of the principal investigator (Donald G. Truhlar). A license does not entitle the licensee to relicense the code or distribute it in original or modified form to parties not covered by the license. The licensee has no ownership rights in the *GESOL* software or in any copyrights for the *GESOL* software or documentation through this license. A user license covers the work of a single research group and the code may be shared and disseminated within a group without requiring permission. Site-license inquiries should be directed to the principal investigator (Donald G. Truhlar).
- B. Publications resulting from using this package will cite the corresponding program. The required references are given in the documentation.
- C. No guarantee is made that this program is bug-free or suitable for specific applications, and no liability is accepted for any limitations in the mathematical methods and algorithms used within the program.
- D. No consulting or maintenance services are guaranteed or implied.
- E. The *Gaussian 03* executable required to use *GESOL* is covered by a separate license.

## 2. Executive summary

*GESOL* is a set of Fortran subroutines and C-Shell scripts that carries out liquid-phase electronic structure calculations using a charge-density (D) based solvation model (SM) called the SMD model. The SMD model in *GESOL* is based on the Polarizable Continuum Model (PCM) of the *Gaussian 03* electronic structure package for the bulk electrostatic component of the solvation free energy and on the Analytical Surface Area (ASA) algorithm (originally developed for the *AMSOL* program, but also contained in full in the *GESOL* program) for the first-solvation-shell component. To run an SMD calculation *GESOL* employs the *External* option of *Gaussian 03*. It requires the user to have a *Gaussian 03* executable (Revisions D or later) installed. It does not require *Gaussian 03* source code, nor does it make any modifications to *Gaussian 03* code. *Gaussian 03* licenses are available from Gaussian, Inc., and *GESOL* licenses are available from the University of Minnesota.

### 3. Theory

#### 3.1. Introduction to the SMD model

The SMD model is a universal continuum solvation model where “universal” denotes applicable to all solvents, and “continuum” denotes that the solvent is not represented explicitly but rather as a dielectric fluid with surface tensions at the solute-solvent interface (“continuum” solvation models are sometimes called “implicit” solvation models). SMD is applicable to any charged or uncharged solute in any solvent or liquid medium for which a few key descriptors are known, and it directly calculates the standard-state free energy of solvation at 298 K and 1 atmosphere of pressure (the user may, if desired, use the standard-state free energy of solvation to calculate free energies of solvation under other conditions, free energies of transfer, partition coefficients, and solubilities). The calculations also yield the polarized solute wave function in solution from which various properties, such as liquid-phase partial charges, may be calculated.

The model separates the standard-state free energy of solvation into three components, as discussed in the next three paragraphs.

The first component is the bulk electrostatic contribution arising from a self-consistent reaction field (SCRF) treatment that involves an integration of the nonhomogeneous Poisson equation for electrostatics in terms of the Polarizable Continuum Model (PCM), more specifically, in terms of the Integral-Equation-Formalism Polarizable Continuum Model (IEFPCM) implemented in *Gaussian 03*. The cavities for the bulk electrostatics calculation are defined by superpositions of nuclear-centered spheres whose sizes are determined by parameters called intrinsic atomic Coulomb radii. The SMD Coulomb radii have been optimized for H, C, N, O, F, Si, P, S, Cl, and Br. For any other atom the SMD model uses the van der Waals radius of Bondi for those atoms for which Bondi defined radii; in cases where the atomic radius is not given in Bondi’s paper (Bondi, A. “Van der Waals volumes and radii,” *J. Phys. Chem.* **1964**, *68*, 441) a radius of 2.0 Å is used. The bulk electrostatic term is sometimes called the electrostatic term, but it should be emphasized that it is calculated from the bulk dielectric constant (bulk relative permittivity), which is not a completely valid description of the solvent in the first solvation shell.

The second contribution to the free energy of solvation is the contribution arising from short-range interactions between the solute and solvent molecules in the first solvation shell. This contribution is sometimes called the cavity–dispersion–solvent-structure (CDS) term, and it is a sum of terms that are proportional (with geometry-dependent proportionality constants called atomic surface tensions) to the solvent-accessible surface areas (SASAs) of the individual atoms of the solute. The SASA of the solute molecule is the area of a surface generated by the center of a spherical effective solvent molecule rolling on the van der Waals surface of the solute molecule. The SASA is calculated with the Analytic Surface Area (ASA) algorithm (see Liotard, D. A.; Hawkins, G. D.; Lynch, G. C.; Cramer, C. J.; Truhlar, D. G. “Improved methods for semiempirical solvation models,” *J. Comput. Chem.* **1995**, *16*, 422). The van der Waals radii

of Bondi are used in this procedure when defined; in cases where the atomic radius is not given in Bondi's paper (Bondi, A. "Van der Waals volumes and radii," *J. Phys. Chem.* **1964**, *68*, 441) a radius of 2.0 Å is used. The solvent radius is set to 0.40 Å for any solvent. Note that the CDS term includes any aspects of solvent structure that are not described by bulk electrostatics, for example, cavitation, dispersion, the partial covalent character of hydrogen bonding, exchange repulsion, and the deviation of the effective dielectric constant in the first solvation shell from its bulk value. The sum of the latter three effects is called the solvent-structure contribution. The semiempirical nature of the CDS term also makes up for errors due to (i) assuming fixed and model-dependent values of the intrinsic Coulomb radii and (ii) any systematic errors in the description of the solute-solvent electrostatic interaction by the nonhomogeneous Poisson equation in terms of the PCM model.

The third component is the concentration component. This is zero if the standard state concentration of the solute is the same in the gas phase and solution (for example, if it is one mole per liter in the gas as well as in the solution), and it can be calculated from the ideal-gas formulas when they are not equal. The SMD solvation free energy is output at 298 K for a standard-state concentration of 1 M in both the gaseous and liquid-phase solution phases. Solvation free energies in the literature are often tabulated using a standard-state-gas phase pressure of 1 atm. To convert 1-molar-to-1-molar solvation free energies at 298 K to a standard state that uses a gas-phase pressure of 1 atm and solute standard state concentration of 1 M, add +1.89 kcal/mol to the computed solvation free energy. Note: we use "liquid phase" and "solution phase" as synonyms in this documentation.

### 3.2. Applicability of the SMD model implemented in *GESOL*

The SMD model employs a single set of parameters optimized over six electronic structure methods, namely, M05-2X/MIDI! 6D, M05-2X/6-31G(d), M05-2X/6-31+G(d,p), M05-2X/cc-pVTZ, B3LYP/6-31G(d), and HF/6-31G(d). The model parameters are intrinsic Coulomb radii for the IEFPCM bulk electrostatics and so-called atomic surface tension coefficients for the CDS contribution to the free energy of solvation. The SMD model implemented in *GESOL* may be used for liquid-phase single-point energy calculations, geometry optimizations, and Hessian evaluations with any of the ground-electronic-state electronic structure methods implemented in *Gaussian 03* for which the IEFPCM model is available. In particular, *GESOL* has been tested for single-point energy calculations using the semiempirical AM1 and PM3 methods, Hartree-Fock (HF) theory, density functional theory (DFT), Møller-Plesset perturbation theory (MP2, MP3, MP4SDQ), coupled cluster theory (CCD, CCSD), configuration interaction (QCISD), and the complete active space self-consistent field (CASSCF) for ground electronic states. Analytical gradients can be computed using HF, DFT, MP2, and CASSCF. Analytical second-order derivatives over the bulk-electrostatic contribution to the free energy of solvation can be computed using HF, DFT, and MP2. Note that in all cases when an analytical Hessian is requested the second-order derivatives over the bulk-electrostatic

contribution is analytical, but the smaller contribution from CDS terms is computed by numerical differentiation of the corresponding analytical first-order derivatives.

Liquid-phase geometry optimizations can be carried out using analytical gradients.

### 3.3. A note on Poisson solvers

IEFPCM is an algorithm for solving the nonhomogeneous Poisson equation for continuum solvation calculations in which the solute is represented by its electronic density in real space.<sup>1</sup> The nonhomogeneous Poisson equation is the Poisson equation for a nonhomogeneous dielectric constant.<sup>2</sup> In particular, the dielectric constant is taken as unity inside the solute (because polarization of the solute is included explicitly) and as the bulk dielectric constant outside the solute (in the solvent, which is treated as a continuum (as opposed to discrete medium)).<sup>3</sup>

Although the SMD model was parameterized<sup>4</sup> for the IEFPCM algorithm, it may also be used with other algorithms for solving the nonhomogeneous Poisson equation for continuum solvation calculations in which the solute is represented by its electron density in real space. This includes, for example, the conductor-like screening algorithm<sup>5</sup> (variously called COSMO, GCOSMO, and C-PCM, but here called CPCM). CPCM solves the nonhomogeneous Poisson equation for an infinite dielectric constant (corresponding to the solvent being a conductor rather than a dielectric) and then uses one or another scaling relation to estimate the result for a finite dielectric constant; this is a better approximation at high dielectric constant (e.g., 80) than at low dielectric constant (e.g., below 10). Even for a very low dielectric constant (e.g., 2), the difference between IEFPCM and CPCM is probably less than the errors in either due to the unrealistic treatment of dielectric response at the solute-solvent boundary (discontinuous change in dielectric constant from 1 to the bulk value at a fixed boundary whose location is somewhat arbitrary) and due to some solute charge lying outside the solute cavity;<sup>6</sup> nevertheless the atomic surface tensions were parameterized using IEFPCM, and so that model is preferred when available. We have tested CPCM against IEFPCM using the SMD default parameters for both algorithms (both algorithms are implemented in *Gaussian 03*). The typical difference between IEFPCM and CPCM in *Gaussian 03* for solutes in solvents with dielectric constants greater than or equal to 32 is less than 0.2 kcal/mol for ions and less than 0.1 kcal/mol for neutrals. The typical difference between IEFPCM and CPCM for neutral solutes in solvents with dielectric constants less than 32 is less than 0.5 kcal/mol.

---

<sup>1</sup> Cancès, E.; Mennucci, B.; Tomasi, J. *J. Chem. Phys.* **1997**, *107*, 3032.

<sup>2</sup> Wangsness, R. K. *Electromagnetic Fields*; Wiley: New York, **1979**; p 179.

<sup>3</sup> Mennucci, B.; Cancès, E.; Tomasi, J. *J. Phys. Chem. B* **1997**, *101*, 10506.

<sup>4</sup> Marenich, A. V.; Cramer, C. J.; Truhlar, D. G. *J. Phys. Chem. B* **2008** (submitted).

<sup>5</sup> Klamt, A.; Schüürmann, G. *J. Chem. Soc., Perkin Trans. 2* **1993**, 799; Truong, T. N.; Stefanovich, E. V. *Chem. Phys. Lett.* **1995**, *240*, 253; Baldrige, K.; Klamt, A. *J. Chem. Phys.* **1997**, *106*, 6622; Barone, V.; Cossi, M. *J. Phys. Chem. A* **1998**, *102*, 1995.

<sup>6</sup> Marenich, A. V.; Cramer, C. J.; Truhlar, D. G. *J. Chem. Theory Comput.* **2008**, *4*, 877.

The SMD model should not be used with the generalized Born approximation. The generalized Born approximation (for which many references are given elsewhere<sup>7</sup>) is based on the partial atomic charges of the solute (rather than the solute electronic density in real space), and it has different systematic errors than the models based on the nonlinear Poisson equation. Just as the SMD solvation model was developed for models based on the nonlinear Poisson equation, an earlier model called SM8<sup>8</sup> was developed for use with the generalized Born approximation.

---

<sup>7</sup> Cramer, C. J.; Truhlar, D. G. *Chem. Rev.* **1999**, *99*, 2161.

<sup>8</sup> Marenich, A. V.; Olson, R. M.; Kelly, C. P.; Cramer, C. J.; Truhlar, D. G. *J. Chem. Theory Comput.* **2007**, *3*, 2011.

## 4. Description of *GESOL* subprograms

The hierarchy of the *GESOL* program is given below:

```
# gesol <input_file> <output_file>
# |
# |----->gesol_header
# |
# |----->gesol_init_f
# |
# |----->$g03root/g03/g03 g03_optgas.com > g03_optgas.log (see a footnote)
# |
# |----->$g03root/g03/g03 g03_shuttle.com > g03_shuttle.log
# |
# |----->gesol_external
# |
# |----->gesol_external_inp_f
# |
# |----->$g03root/g03/g03 g03_external.com > g03_external.log
# |
# |----->gesol_external_out_f
```

Subprogram	Description
<b>gesol</b>	This C-Shell script reads the <i>GESOL</i> input file provided by the user, checks it for errors, initializes the <i>GESOL</i> program, and terminates the <i>GESOL</i> program. It creates the <i>GESOL</i> scratch directory, the <i>GESOL</i> work files containing the SMD parameters, and the <i>Gaussian 03</i> input file called <code>g03_shuttle.com</code> that contains the solute's geometry and the "External" keyword to invoke an external program called <code>gesol_external</code> . Upon terminating the ' <code>g03 g03_shuttle.com</code> ' command this script retrieves the computational results from the <i>Gaussian 03</i> output file called <code>g03_shuttle.log</code> and passes the information to the <i>GESOL</i> output file.
<b>gesol_header</b>	This C-Shell script outputs the <i>GESOL</i> version information.

<b>gesol_init_f</b>	<p>This Fortran code reads the \$GESOL namelist in the <i>GESOL</i> input file, sets up the SMD model parameters, and passes this information to <code>gesol</code>, with the solvent's descriptors being retrieved from a file called <code>gesol_solvent_t</code> (this file is included in the <i>GESOL</i> distribution).</p>
<b>g03</b>	<p>This is a <i>Gaussian 03</i> executable. It is called twice (see a footnote): one time by the <code>gesol</code> script (the <code>g03</code> parent process), another time by <code>gesol_external</code> (the <code>g03</code> child process).</p> <p>The <code>g03</code> parent process reads the input information from the <code>g03_shuttle.com</code> file and writes the computational results into the <code>g03_shuttle.log</code> file. The same process generates an input file with the name <code>Gau-\$PID.EIn</code> for running an external program called <code>gesol_external</code>. The <code>g03</code> parent process reads an output file with the name <code>Gau-\$PID.EOu</code> generated by <code>gesol_external</code> upon its termination. The format of the <code>Gau*</code> files is described at <a href="http://www.gaussian.com/g_ur/k_external.htm">http://www.gaussian.com/g_ur/k_external.htm</a>.</p> <p>The <code>g03</code> child process called by <code>gesol_external</code> carries out a PCM electrostatic calculation using the <code>g03_external.com</code> input file and writes the results in the <code>g03_external.log</code> file.</p> <p>Note that <i>Gaussian 03</i> should be installed prior to execution of the <i>GESOL</i> program. The <code>\$g03root</code> variable must be set up, otherwise the <i>GESOL</i> program will stop.</p>
<b>gesol_external</b>	<p>This C-Shell script creates the <code>g03_external.com</code> input file for the <code>g03</code> child process. Upon the termination of the latter, this script collects the PCM electrostatic data (energies, gradients, and second-order derivatives if requested) from the <code>g03_external.chk</code> checkpoint file and the <code>g03_external.log</code> file generated by the <code>g03</code> child process and passes the electrostatic data augmented with the CDS corrections to the <code>Gau-\$PID.EOu</code> file.</p> <p>In the case of geometry optimization and numerical calculation of gradients and frequencies, the <code>gesol_external</code> script is called once for each geometry.</p>

<b>gesol_external_inp_f</b>	This Fortran code computes the intrinsic Coulomb radii and the CDS contributions to the PCM electrostatic data (energies, gradients, and second-order derivatives if requested) for <code>gesol_external</code> .
<b>gesol_external_out_f</b>	This Fortran code produces the <i>GESOL</i> solvation data output for <code>gesol_external</code> .

\* If a gas-phase geometry optimization is requested, an additional *Gaussian* calculation will be carried out before a solvation calculation. In this case, the solvation calculation will use the optimized gas-phase geometry stored in the file called `g03_optgas.log`.

The typical structure of the `g03_shuttle.com` file:

```
# HF/STO-3G SCF=TIGHT FORCE
# External='gesol_external'

title

0 1
C 0.000000 0.000000 0.000000
O 0.000000 0.000000 1.220000
H 0.935307 0.000000 -0.540000
H -0.935307 0.000000 -0.540000
```

The typical structure of the `g03_external.com` file:

```
%mem=500mb
%nproc=4
%chk=g03_external
#P CIS(Root=1,50-50,Conver=7,EqSolv,full)/6-31G* SCF=TIGHT scrf=(cpcm,read)
# force

title

      0          1
6      0.0000000000      0.0000000000      -0.5425000026
8      0.0000000000      0.0000000000      0.6775000032
1      0.0000000000      0.9353070044      -1.0825000051
1      0.0000000000     -0.9353070044      -1.0825000051

scfvac pcmdoc nodis norep nocav rsolv=0 epsinf=1.00001 surface=sas noaddsph
EPS=78.35000
RADII=UFF MODIFYSFH

1 1.850000 [1.0]
2 1.520000 [1.0]
3 1.200000 [1.0]
4 1.200000 [1.0]
```

This is a brief description of *GESOL* subprograms. In most of the cases, a more detailed description is given in the comment cards in the Fortran code or C-Shell script for each subprogram.

## 5. Citation of the *GESOL* program and the methods used by *GESOL*

Publications including work performed with *GESOL* should cite the *GESOL* software package:

Marenich, A. V.; Hawkins, G. D.; Liotard, D. A.; Cramer, C. J.; Truhlar, D. G. *GESOL* – version 2008, University of Minnesota, Minneapolis, 2008.

Publications including work performed with *GESOL* should also cite the *Gaussian* software package:

Frisch, M. J.; Trucks, G. W.; Schlegel, H. B.; Scuseria, G. E.; Robb, M. A.; Cheeseman, J. R.; Montgomery, J. A., Jr.; Vreven, T.; Kudin, K. N.; Burant, J. C.; Millam, J. M.; Iyengar, S. S.; Tomasi, J.; Barone, V.; Mennucci, B.; Cossi, M.; Scalmani, G.; Rega, N.; Petersson, G. A.; Nakatsuji, H.; Hada, M.; Ehara, M.; Toyota, K.; Fukuda, R.; Hasegawa, J.; Ishida, M.; Nakajima, T.; Honda, Y.; Kitao, O.; Nakai, H.; Klene, M.; Li, X.; Knox, J. E.; Hratchian, H. P.; Cross, J. B.; Bakken, V.; Adamo, C.; Jaramillo, J.; Gomperts, R.; Stratmann, R. E.; Yazyev, O.; Austin, A. J.; Cammi, R.; Pomelli, C.; Ochterski, J. W.; Ayala, P. Y.; Morokuma, K.; Voth, G. A.; Salvador, P.; Dannenberg, J. J.; Zakrzewski, V. G.; Dapprich, S.; Daniels, A. D.; Strain, M. C.; Farkas, O.; Malick, D. K.; Rabuck, A. D.; Raghavachari, K.; Foresman, J. B.; Ortiz, J. V.; Cui, Q.; Baboul, A. G.; Clifford, S.; Cioslowski, J.; Stefanov, B. B.; Liu, G.; Liashenko, A.; Piskorz, P.; Komaromi, I.; Martin, R. L.; Fox, D. J.; Keith, T.; Al-Laham, M. A.; Peng, C. Y.; Nanayakkara, A.; Challacombe, M.; Gill, P. M. W.; Johnson, B.; Chen, W.; Wong, M. W.; Gonzalez, C.; Pople, J. A. *Gaussian 03, Revisions D.01 or E.01*; Gaussian, Inc.: Wallingford, CT, 2004.

In addition, as usual, the user should give literature references for any methods used. A convenient collection of literature references is provided next.

### SMD model

Marenich, A. V.; Cramer, C. J.; Truhlar, D. G. “Universal solvation model based on solute electron density and on a continuum model of the solvent defined by the bulk dielectric constant and atomic surface tensions,” *J. Phys. Chem. B* **2009** (in press).

### ASA algorithm

Liotard, D. A.; Hawkins, G. D.; Lynch, G. C.; Cramer, C. J.; Truhlar, D. G. “Improved methods for semiempirical solvation models,” *J. Comp. Chem.* **1995**, *16*, 422.

### Derivatives of CDS terms

Tunón, I.; Ruiz-López, M. F.; Rinaldi, D.; Bertrán, J. “Computation of hydration free energies using a parameterized continuum model: study of equilibrium geometries and reactive processes in water solution,” *J. Comp. Chem.* **1996**, *17*, 148.

Zhu, T.; Li, J.; Liotard, D. A.; Cramer, C. J.; Truhlar, D. G. "Analytical gradients of a self-consistent reaction-field solvation model based on CM2 atomic charges," *J. Chem. Phys.* **1999**, *110*, 5503.

#### IEFPCM algorithm

Cancès, E.; Mennucci, B.; Tomasi, J. "A new integral equation formalism for the polarizable continuum model: Theoretical background and applications to isotropic and anisotropic dielectrics," *J. Chem. Phys.* **1997**, *107*, 3032.

Mennucci, B.; Tomasi, J. "Continuum solvation models: a new approach to the problem of solute's charge distribution and cavity boundaries," *J. Chem. Phys.* **1997**, *106*, 5151.

Mennucci, B.; Cancès, E.; Tomasi, J. "Evaluation of solvent effects in isotropic and anisotropic dielectrics and in ionic solutions with a unified integral equation method: theoretical bases, computational implementation, and numerical applications," *J. Phys. Chem. B* **1997**, *101*, 10506.

Tomasi, J.; Mennucci, B.; Cancès, E. "The IEF version of the PCM solvation method: an overview of a new method addressed to study molecular solutes at the QM ab initio level," *J. Mol. Struct. (Theochem)* **1999**, *464*, 211.

#### CPCM algorithm

Barone, V.; Cossi, M. "Quantum calculation of molecular energies and energy gradients in solution by a conductor solvent model," *J. Phys. Chem. A* **1998**, *102*, 1995.

Cossi, M.; Rega, M.; Scalmani, G.; Barone, V. "Energies, structures, and electronic properties of molecules in solution with the C-PCM solvation model," *J. Comp. Chem.* **2003**, *24*, 669.

#### CASSCF PCM algorithm

Cossi, M.; Barone, V.; Robb, M. A. "A direct procedure for the evaluation of solvent effects in MC-SCF calculations," *J. Chem. Phys.* **1999**, *111*, 5295.

## 6. *GESOL* input description

This section describes the structure of a *GESOL* input file. Various examples of the *GESOL* input are given in the section entitled “*GESOL* test suite”.

In general, the *GESOL* input is subject to the same syntax rules as used by any *Gaussian 03* input file as described at [http://www.gaussian.com/g\\_ur/m\\_input.htm](http://www.gaussian.com/g_ur/m_input.htm). However, the *GESOL* input has the following features and limitations:

1. The user should use a Unix editor (for instance, vi) to construct a *GESOL* input file.
2. The \$GESOL namelist must be placed in the end of the input file (a type case of the \$GESOL content is not critical). The \$GESOL namelist keywords are described further in this section.
3. The \$GESOL keyword should be followed by the \$END word.
4. Input sections should be separated by a blank line.
5. There should be no link command in the input.
6. A route section (# line) must be presented, but it cannot occupy more than one line.
7. A route section (# line) cannot contain more than six keywords.
8. The *geom* keyword in the route section as well as reading the geometry from a checkpoint file is not permitted.
9. The *scrf* keyword in the route section is not permitted.
10. The keywords in the route section should be separated by at least one space.
11. There should be no space between the options within the same keyword used in the route group. For instance, “opt=(path=7,trustupdate,addredun,qst3)” is correct.
12. A title section must be present, but it cannot occupy more than one line.
13. The use of Cartesian coordinates for molecular geometry specification is recommended. However, the use of redundant internal coordinates through the *Gaussian 03* options *ModRedundant*, *QST2*, and *QST3* is still permitted unless the *gen* or *genecp* keywords are specified.
14. The user should avoid basis set definitions with the use of asterisk. For instance, use “6-31G(d)” instead of “6-31G\*”.

Keyword in \$GESOL	Description
EGAS=0	calculates the free energy of solvation relative to the gas-phase total energy computed for each new geometry (default)
EGAS=1	calculates the free energy of solvation relative to the gas-phase total energy computed for the input geometry (this is equivalent to EGAS=0 if only a single-point energy calculation is requested)
EGAS= <i>EGAS_Value</i>	calculates the free energy of solvation relative to <i>EGAS_Value</i> provided by the user

IGAS=0	does not optimize the gas-phase geometry (default)
IGAS=1	optimizes the gas-phase geometry
LCDS=.TRUE.	turns on the CDS corrections (default)
LCDS=.FALSE.	turns off the CDS corrections
LVERB=.TRUE.	turns on the verbose output
LVERB=.FALSE.	turns off the verbose output (default)
PCM='IEFPCM'	invokes the IEFPCM model for bulk electrostatics (default)
PCM='CPCM'	invokes the CPCM model for bulk electrostatics
PCM='NONE'	skips solvation
READRAD='file_name'	requires that the intrinsic Coulomb radii should be read in for each atom in the molecule from the file provided by the user (this option can override the SMD default settings)
READPCM='file_name'	requires that the PCM keywords should be read in from the file provided by the user (this option can override the SMD default settings); by default <i>GESOL</i> uses the following PCM keywords: scfvac, pcmdoc, nodis, norep, nocav, rsolv=0, epsinf=1.00001, surface=sas, noaddsph
SOLVENT='solvent_name'	defines the solvent by solvent name (default: SOLVENT='WATER')

In addition to SOLVENT='WATER', *GESOL* supports the following solvent keywords:

1,1,1-TRICHLOROETHANE	ACETONITRILE	ETHYLBENZENE
1,1,2-TRICHLOROETHANE	ACETOPHENONE	FLUOROBENZENE
1,2,4-TRIMETHYLBENZENE	ANILINE	FORMAMIDE
1,2-DIBROMOETHANE	ANISOLE	FORMICACID
1,2-DICHLOROETHANE	BENZALDEHYDE	N-HEPTANE
1,2-ETHANEDIOL	BENZENE	N-HEXADECANE
1,4-DIOXANE	BENZONITRILE	N-HEXANE
1-BROMO-2-METHYLPROPANE	BENZYLALCOHOL	HEXANOICACID
1-BROMOOCTANE	BROMOBENZENE	IODOBENZENE
1-BROMOPENTANE	BROMOETHANE	IDOETHANE
1-BROMOPROPANE	BROMOFORM	IODOMETHANE
1-BUTANOL	BUTANAL	ISOPROPYLBENZENE
1-CHLOROHEXANE	BUTANOICACID	P-ISOPROPYLTOLUENE
1-CHLOROPENTANE	BUTANONE	MESITYLENE
1-CHLOROPROPANE	BUTANONITRILE	METHANOL
1-DECANOL	BUTYLETHANOATE	METHYLBENZOATE
1-FLUOROCTANE	BUTYLAMINE	METHYLBUTANOATE
1-HEPTANOL	N-BUTYLBENZENE	METHYLETHANOATE
1-HEXANOL	SEC-BUTYLBENZENE	METHYLMETHANOATE

1-HEXENE	TERT-BUTYLBENZENE	METHYLPROPANOATE
1-HEXYNE	CARBONDISULFIDE	N-METHYLANILINE
1-IODOBUTANE	CARBONTETRACHLORIDE	METHYLCYCLOHEXANE
1-ODOHEXADECANE	CHLOROBENZENE	N-METHYLFORMAMIDE ( E / Z MIXTURE )
1-IODOPENTANE	CHLOROFORM	NITROBENZENE
1-IODOPROPANE	A-CHLOROTOLUENE	NITROETHANE
1-NITROPROPANE	O-CHLOROTOLUENE	NITROMETHANE
1-NONANOL	M-CRESOL	O-NITROTOLUENE
1-OCTANOL	O-CRESOL	N-NONANE
1-PENTANOL	CYCLOHEXANE	N-OCTANE
1-PENTENE	CYCLOHEXANONE	N-PENTADECANE
1-PROPANOL	CYCLOPENTANE	PENTANAL
2,2,2-TRIFLUOROETHANOL	CYCLOPENTANOL	N-PENTANE
2,2,4-TRIMETHYLPENTANE	CYCLOPENTANONE	PENTANOICACID
2,4-DIMETHYLPENTANE	DECALIN ( CIS / TRANSMIXTURE )	PENTYLETHANOATE
2,4-DIMETHYLPYRIDINE	CIS-DECALIN	PENTYLAMINE
2,6-DIMETHYLPYRIDINE	N-DECANE	PERFLUOROBENZENE
2-BROMOPROPANE	DIBROMOMETHANE	PROPANAL
2-BUTANOL	DIBUTYLETHER	PROPANOICACID
2-CHLOROBUTANE	O-DICHLOROBENZENE	PROPANONITRILE
2-HEPTANONE	E-1,2-DICHLOROETHENE	PROPYLETHANOATE
2-HEXANONE	Z-1,2-DICHLOROETHENE	PROPYLAMINE
2-METHOXYETHANOL	DICHLOROMETHANE	PYRIDINE
2-METHYL-1-PROPANOL	DIETHYLETHER	TETRACHLOROETHENE
2-METHYL-2-PROPANOL	DIETHYLSULFIDE	TETRAHYDROFURAN
2-METHYLPENTANE	DIETHYLAMINE	TETRAHYDROTHIOPHENE-S,S-DIOXIDE
2-METHYLPYRIDINE	DI IODOMETHANE	TETRALIN
2-NITROPROPANE	DIISOPROPYLETHER	THIOPHENE
2-OCTANONE	CIS-1,2-DIMETHYLCYCLOHEXANE	THIOPHENOL
2-PENTANONE	DIMETHYLDISULFIDE	TOLUENE
2-PROPANOL	N,N-DIMETHYLACETAMIDE	TRANS-DECALIN
2-PROPEN-1-OL	N,N-DIMETHYLFORMAMIDE	TRIBUTYLPHOSPHATE
E-2-PENTENE	DIMETHYLSULFOXIDE	TRICHLOROETHENE
3-METHYLPYRIDINE	DIPHENYLETHER	TRIETHYLAMINE
3-PENTANONE	DIPROPYLAMINE	N-UNDECANE
4-HEPTANONE	N-DODECANE	XYLENE ( MIXTURE )
4-METHYL-2-PENTANONE	ETHANETHIOL	M-XYLENE
4-METHYLPYRIDINE	ETHANOL	O-XYLENE
5-NONANONE	ETHYLETHANOATE	P-XYLENE
ACETICACID	ETHYLMETHANOATE	
ACETONE	ETHYLPHENYLETHER	

The free energy of solvation for a given solute in a particular medium depends on the values of the solvent descriptors defined for the medium. The SMD model uses the following set of solvent descriptors for any nonaqueous solvent:

**DIELEC** dielectric constant,  $\epsilon$ , of solvent

**SOLN** index of refraction at optical frequencies at 293 K,  $n_{20}^D$

<b>SOLA</b>	Abraham's hydrogen bond acidity, $\Sigma\alpha_2^H$
<b>SOLB</b>	Abraham's hydrogen bond basicity, $\Sigma\beta_2^H$
<b>SOLG</b>	$\gamma = \gamma_m / \gamma^0$ (default is 0.0), where $\gamma_m$ is the macroscopic surface tension at air/solvent interface at 298 K, and $\gamma^0$ is $1 \text{ cal mol}^{-1} \text{ \AA}^{-2}$ ( $1 \text{ dyne/cm} = 1.43932 \text{ cal mol}^{-1} \text{ \AA}^{-2}$ )
<b>SOLC</b>	aromaticity, $\phi$ : the fraction of non-hydrogenic solvent atoms that are aromatic carbon atoms
<b>SOLH</b>	electronegative halogenicity, $\psi$ : the fraction of non-hydrogenic solvent atoms that are F, Cl or Br

In the case of water, there is only one solvent descriptor, the dielectric constant (DIELEC=78.35), which defines the bulk-electrostatics contribution to the free energy of solvation. If the solvent is specified, the program will use the values of the solvent descriptors (DIELEC, SOLA, SOLB, SOLC, SOLG, SOLH, and SOLN), which are stored in a file called `gesol_solvent_t`. However, the user can override the default values by specifying the new (floating-point) values of these descriptors in the namelist \$GESOL in the *GESOL* input file. For a desired solvent, these values can be derived from experiment or from interpolation or extrapolation of data available for other solvents. Solvent parameters for common organic solvents are tabulated in the *Minnesota Solvent Descriptor Database*. The latest version of this database is available at: <http://comp.chem.umn.edu/solvation/mnsddb.pdf>.

By default for a single-point energy calculation in the liquid phase the free energy of solvation is defined as a difference between the liquid-phase total energy and the gas-phase total energy calculated at the input geometry unless the `EGAS=EGAS_Value` option is used. In the latter case *GESOL* uses the gas-phase total energy (*EGAS\_Value*) provided by the user.

If a liquid-phase geometry optimization is requested, it is recommended to use the equilibrium gas-phase geometry as an initial geometry and the `EGAS=1` option in the *GESOL* input file. The user can exploit the `IGAS=1` option to optimize the gas-phase geometry using *GESOL* (this option can be applied only if the input geometry is given in Cartesian coordinates). If the `IGAS=1` option fails to work, the gas-phase geometry should be obtained separately using the original *Gaussian*. If the gas-phase geometry is not a good approximation to the liquid-phase geometry, the user may exploit any other initial geometry, provided the total energy corresponding to the equilibrium gas-phase geometry is specified via the `EGAS` keyword in the *GESOL* input file. In either case the final free energy of solvation is calculated as a difference between the liquid-phase total energy obtained at the liquid-phase equilibrium geometry and the gas-phase total energy obtained at the gas-phase equilibrium geometry. Therefore, the final energy of solvation contains the nuclear reorganization term (in other words, an energy term corresponding to the change in the solute's geometry upon solvation). If the `EGAS=0` option is used, the gas-phase energy will be updated for each optimization step and, therefore, the nuclear reorganization contribution to the free energy of solvation will be omitted.

## 7. *GESOL* output description

This section describes the *GESOL* output structure. Each output file is a concatenation of the following outputs:

1. Header
2. Table with the *GESOL* keywords
3. File `g03_shuttle.com`
4. Beginning of `g03_shuttle.log`
5. File `g03_external.com`
6. File `g03_external.log` if verbose output is requested (`LVERB=.TRUE.`)
7. *GESOL* solvation summary
8. End of `g03_shuttle.log`

Note that in the case of geometry optimization sections 5, 6, 7 are printed for each optimization cycle.

If a gas-phase geometry optimization is requested (`IGAS=1`), an additional *Gaussian* calculation will be carried out before a solvation calculation. An additional file (`g03_optgas.log`) containing the output of the gas-phase geometry optimization will be generated. The gas-phase optimization will be followed by a solvation calculation using the optimized gas-phase geometry.

*GESOL* exploits the `scfvac` option in *Gaussian 03* for a PCM bulk-electrostatics calculation unless the `READPCM` keyword in the *GESOL* input file is applied to specifically override the default setting. If the `scfvac` option is used in the *Gaussian* calculation, the liquid-phase SCRF is preceded by a gas-phase SCF. When the gas-phase SCF has converged, *Gaussian* continues the same SCF procedure to the liquid-phase SCRF. *GESOL* can exploit the `scfvac` option with any of the following electronic structure methods: AM1, PM3, HF, DFT, and CASSCF. For other available methods (MP2, MP3, MP4SDQ, CCD, CCSD, QCISD) the `scfvac` option cannot be used. For any of these post-HF methods *Gaussian 03* equilibrates the solvent reaction field calculated at a given level of theory to the solute electronic density obtained at the HF level, skipping the gas-phase energy calculation at the post-HF level of theory. Nevertheless, in those cases when the `scfvac` option is not used (for instance, when the user overrides the default *GESOL* setting) or cannot be used (like with the methods above), *GESOL* creates an additional file called `g03_externalgas.com` to carry out a separate gas-phase single-point energy calculation. However, this calculation will be skipped if the user provides the gas-phase total energy in the *GESOL* input file using the `EGAS=EGAS_Value` keyword.

The typical *GESOL* solvation summary lists the values of Coulomb atomic radii used for a bulk-electrostatics calculation, van der Waals radii used for a CDS calculation, CDS contributions to the gradients in Cartesian coordinates (if requested), atomic contributions to the solvent accessible surface area and to the CDS energy term, and solvation energy components split over bulk-electrostatics and non-bulk-electrostatics contributions.

An example of solvation energy components is given below for the case when the `scfvac` option is used:

(0) E-EN(g) gas-phase elect-nuc energy	-286.236405 a.u.
(1) E-EN(liq) elect-nuc energy of solute	-286.235727 a.u.
(2) G-P(liq) polarization free energy of solvation	-35.517 kcal/mol
(3) G-ENP(liq) elect-nuc-pol free energy of system	-286.292326 a.u.
(4) G-CDS(liq) cavity-dispersion-solvent structure free energy	0.000 kcal/mol
(5) G-P-CDS(liq) = G-P(liq) + G-CDS(liq) = (2) + (4)	-35.517 kcal/mol
(6) G-S(liq) free energy of system = (1) + (5)	-286.292326 a.u.
(7) DeltaE-EN(liq) elect-nuc reorganization energy of solute molecule (7) = (1) - (0)	0.425 kcal/mol
(8) DeltaG-ENP(liq) elect-nuc-pol free energy of solvation (8) = (3) - (0)	-35.091 kcal/mol
(9) DeltaG-S(liq) free energy of solvation (9) = (6) - (0)	-35.091 kcal/mol

The ENP term refers to the electronic-nuclear-polarization (ENP) contribution to the free energy of solvation. The nuclear relaxation (N) component of the ENP term is equal to the difference between the gas-phase total energy calculated at the gas-phase equilibrium structure and the *gas*-phase total energy calculated at the *liquid*-phase equilibrium structure. If the geometry is assumed to be the same in the gas phase and in the liquid phase (see the `EGAS=0` option in the section entitled “*GESOL* input description”), then the ENP term becomes just an electronic polarization (EP) term. The CDS term refers to the cavity–dispersion–solvent-structure contribution.

If the `scfvac` option is not used the *GESOL* output of energy components is simplified to the following:

(0) E-EN(g) gas-phase elect-nuc energy	-286.682983 a.u.
(1) G-ENP(liq) elect-nuc-pol free energy of system	-286.795025 a.u.
(2) G-CDS(liq) cavity-dispersion-solvent structure free energy	3.384 kcal/mol
(3) G-S(liq) free energy of system = (1) + (2)	-286.789632 a.u.
(4) DeltaG-S(liq) free energy of solvation (4) = (3) - (0)	-66.924 kcal/mol

If verbose output is requested (`LVERB=.TRUE.`) the *GESOL* program will generate two checkpoint files: `g03_geom.chk` (with geometry parameters, forces, and a computed Hessian if requested) and `g03_morb.chk` (with computed molecular orbitals after the last SCF calculation).

## 8. GESOL installation and execution

The *GESOL* software package comes as a single tar file (`gesol.tar`) that should be placed in the user's directory and then untarred by executing the following command:

```
>tar xvf gesol.tar
```

There will be a directory called `gesol` containing two subdirectories (`gesol/bin` and `gesol/tests`). The subdirectory `gesol/bin` contains the following files:

```
-rwxr-x---      comp_gesol
-rwxr-x---      gesol
-rwxr-x---      gesol_external
-rw-r-----    gesol_external_inp_f.F
-rw-r-----    gesol_external_out_f.F
-rwxr-x---      gesol_header
-rw-r--r--     gesol_init_f.F
-rw-r-----    gesol_solvent_t
```

The user should set the working directory to `gesol/bin` and then compile the Fortran codes (`*.F`) by executing the following command:

```
>./comp_gesol
```

The user should define the full path to `gesol/bin/gesol` through the `$PATH` environment variable in a dot-file called `.bashrc` or in `.cshrc` (depending on which shell is used), for instance, by adding a line:

```
export PATH=$PATH:$HOME/gesol/bin
```

or

```
setenv PATH ${PATH} : $HOME/gesol/bin
```

The user should also assign the `$DIR_GESOL` variable to the full path to the `gesol/bin` directory by adding a line in `.bashrc` or in `.cshrc`:

```
export DIR_GESOL=$HOME/gesol/bin
```

or

```
setenv DIR_GESOL $HOME/gesol/bin
```

It is recommended to define the scratch space through the `$SCR_GESOL` variable. If `$SCR_GESOL` is not set up by the user, the program will set it to `'/scratch*/$USER/'` if such a directory exists (otherwise it will set the scratch space to the current directory).

The user can run the *GESOL* program by executing the following command:

```
>gesol <input> <output>
```

To assure that the *GESOL* kit has been properly installed, the user is encouraged to run the whole *GESOL* test suite that can be found in `gesol/tests`. This directory contains 42 input examples (`*.inp`). The corresponding output files (`*.out`) are located in

gesol/tests/output. A Perl script called `compare.pl` can be used to compare the user-obtained output files against those in `gesol/tests/output` by executing the following command:

```
>./compare.pl *.out
```

A difference in the last digit is acceptable. Test18, Test19, Test23, and Test27 may produce a difference in the second-third digit that is acceptable too. All the tests using the Minnesota (M05) suite of density functionals will fail if *GESOL* is run with a revision of *Gaussian 03* earlier than Revision E.01.

Note that *Gaussian 03* should be installed prior to execution of the *GESOL* program. The `$g03root` variable must be set up, otherwise the *GESOL* program will stop.

## 9. *GESOL* test suite

The *GESOL* test suite contains 42 representative input/output files located in `gesol/tests`. This test suite includes single-point energy calculations, local minimum and transition state geometry optimizations, and Hessian calculations for the ground electronic states in the liquid phase. The tested electronic structure methods include AM1, PM3, HF, DFT, MP2, MP3, MP4SDQ, CCD, CCSD, QCISD, and CASSCF. The open-shell electronic systems  $\text{Ru}(\text{H}_2\text{O})_6^{3+}$  and  $\text{CH}_3\text{O}^\bullet$  are tested using spin-unrestricted Hartree-Fock (UHF) wavefunctions.

### **test1.inp**

HF/STO-3G single-point solvation energy and analytical gradients for the cyclohexanamine cation in water

### **test1.am1.inp**

AM1 single-point solvation energy for the cyclohexanamine cation in water

### **test1.pm3.inp**

PM3 single-point solvation energy for the cyclohexanamine cation in water

### **test1.ccd.inp**

CCD single-point solvation energy for the cyclohexanamine cation in water

### **test1.ccsd.inp**

CCSD single-point solvation energy for the cyclohexanamine cation in water

### **test1.mp3.inp**

MP3 single-point solvation energy for the cyclohexanamine cation in water

### **test1.mp4.inp**

MP4(SDQ) single-point solvation energy for the cyclohexanamine cation

### **test1.qcisd.inp**

QCISD single-point solvation energy for the cyclohexanamine cation in water

### **test2.inp**

B3LYP/3-21G analytical geometry optimization and analytical Hessian calculation in the liquid phase for the cyclohexanamine cation using the CalcFC option (the initial Hessian is calculated), LCDS=.FALSE. (no CDS terms are included), READRAD='radii.dat' (Coulomb radii are read in from the file called `radii.dat`)

### **test3.inp**

B3LYP/6-31+G(d,p) analytical Hessian calculation for methanol in the gas phase

**test3.m052x.inp**

M05-2X/6-31+G(d,p) analytical Hessian calculation for methanol in the gas phase

**test4.inp**

B3LYP/6-31+G(d,p) analytical geometry optimization and analytical Hessian calculation for methanol in benzene using the IGAS=1 option (the gas-phase geometry is optimized first)

**test4.m052x.inp**

M05-2X/6-31+G(d,p) analytical geometry optimization and analytical Hessian calculation for methanol in benzene using the IGAS=1 option (the gas-phase geometry is optimized first)

**test5.inp**

B3LYP/6-31+G(d,p) analytical geometry optimization and analytical Hessian calculation for methanol in methylene chloride using the IGAS=1 option (the gas-phase geometry is optimized first)

**test5.m052x.inp**

M05-2X/6-31+G(d,p) analytical geometry optimization and analytical Hessian calculation for methanol in methylene chloride using the IGAS=1 option (the gas-phase geometry is optimized first)

**test6.inp**

B3LYP/6-31+G(d,p) analytical geometry optimization and analytical Hessian calculation for methanol in methanol using the IGAS=1 option (the gas-phase geometry is optimized first)

**test6.m052x.inp**

M05-2X/6-31+G(d,p) analytical geometry optimization and analytical Hessian calculation for methanol in methanol using the IGAS=1 option (the gas-phase geometry is optimized first)

**test7.inp**

B3LYP/6-31+G(d,p) analytical geometry optimization and analytical Hessian calculation for methanol in water using the PCM keywords provided in the file `pcm.dat` (option READPCM) and the EGAS=-115.734872 option (it means that the free energy of solvation at each step of the liquid-phase geometry optimization is calculated relative to the user-provided gas-phase total energy value that is -115.734872; the gas-phase SCF calculation is skipped)

**test7.m052x.inp**

M05-2X/6-31+G(d,p) analytical geometry optimization and analytical Hessian calculation for methanol in water using the PCM keywords provided in the file `pcm.dat` (option READPCM) and the EGAS=-115.706492 option (it means that the free energy of solvation at each step of the liquid-phase geometry optimization is calculated relative to the user-provided gas-phase total energy value that is -115.706492; the gas-phase SCF calculation is skipped)

**test8.inp**

MP2(Full)/6-31+G(d,p) analytical geometry optimization and analytical Hessian calculation for methanol in water using the EGAS=0 option (it means that the free energy of solvation at each step of the liquid-phase geometry optimization is calculated relative to the gas-phase total energy updated for each step during the optimization)

**test9.inp**

MP2(Full)/6-31+G(d,p) analytical geometry optimization and analytical Hessian calculation for methanol in water using the EGAS=1 option (it means that the free energy of solvation at each step of the liquid-phase geometry optimization is calculated relative to the gas-phase total energy obtained at the first optimization step for the input geometry which in this example is the equilibrium gas-phase geometry)

**test10.inp**

HF/3-21G analytical gradients for methanol in acetonitrile

**test11.inp**

HF/3-21G numerical gradients for methanol in acetonitrile using the default *Gaussian 03* value for differentiation step

**test12.inp**

HF/6-31G(d) single-point energy for dimethyl 4-bromo-2,5-dichlorophenyl thiophosphate in water

**test13.inp**

HF/6-31G(d) single-point energy for dimethyl 4-bromo-2,5-dichlorophenyl thiophosphate in octanol

**test14.inp**

HF/6-31G(d) single-point energy for tetramethylsilane in water using LVERB=.TRUE. (verbose output) in the \$GESOL namelist and '# P' in the *Gaussian 03* route line (for more verbose output)

**test15.inp**

HF/6-31G(d) single-point energy for tetramethylsilane in water using the PCM='CPCM' option (CPCM model for electrostatics)

**test16.inp**

HF/6-31G(d) single-point energy for tetramethyl silane in water using the CPCM model for electrostatics (option PCM='CPCM') and the PCM keywords provided in the file pcm.dat (option READPCM) as well as LVERB=.TRUE. (verbose output) in the \$GESOL namelist and '# P' in the *Gaussian 03* route line (for more verbose output)

**test17.inp**

HF/6-31G(d) single-point energy for 1-heptanol in ortho-dichlorobenzene with the solvent parameters (DIELEC, SOLA, SOLB, SOLC, SOLG, SOLH, SOLN) defined by the user in the \$GESOL namelist

**test18.inp**

BLYP/6-31B(d) single-point energy for aniline in carbontet using the GEN keyword in the *Gaussian 03* route line and the 6-31B(d) basis set stored externally in the file `basis.dat`

**test19.inp**

BLYP/6-31B(d) single-point energy for aniline in carbontet using the GEN keyword in the *Gaussian 03* route line and the 6-31B(d) basis set entered in the *GESOL* input file as a separate section

**test20.inp**

Spin-unrestricted BLYP single-point energy for the Ru(III) hexaaqua complex using the GEN keyword in the *Gaussian 03* route line and the effective core potential on Ru

**test21.inp**

HF analytical Hessian calculation for the Ru(II) hexaaqua complex in the gas phase using LVERB=.TRUE. to obtain the checkpoint file `g03_geom.chk` with the initial Hessian for the next test

**test22.inp**

HF analytical geometry optimization for the Ru(II) hexaaqua complex in water using the initial Hessian obtained in the previous test (option `opt=ReadFC`) and analytical calculation of frequencies for the liquid-phase equilibrium geometry

**test23.inp**

HF/6-31+G(d) analytical optimization of the transition state of the  $\text{NH}_3 + \text{CH}_3\text{Cl}$  reaction in water using internal coordinates defined with the `ModRedundant` option and analytical Hessian calculation

**test24.inp**

HF/6-31+G(d) analytical optimization of the transition state of the  $\text{NH}_3 + \text{CH}_3\text{Cl}$  reaction in water using Cartesian coordinates (no `ModRedundant`) and analytical Hessian calculation

**test25.inp**

HF/3-21G analytical optimization of the  $\text{SiH}_2 + \text{H}_2$  transition state (*Gaussian 03* Test Job 302) using the `opt=qst2` option and testing the IEFPCM bulk electrostatics only (`LCDS=.FALSE.`)

**test26.inp**

HF/3-21G analytical optimization of the  $\text{SiH}_2 + \text{H}_2$  transition state (*Gaussian 03* Test Job 302) using the `opt=qst2` option and the SMD default settings for aqueous solution (`LCDS=.TRUE.`, `SOLVENT='WATER'`)

**test27.inp**

Spin-unrestricted HF/3-21G analytical optimization of the transition state of the  $\text{CH}_2\text{OH} \rightarrow \text{CH}_3\text{O}$  isomerization (*Gaussian 03* Test Job 413) in n-hexane using the `opt=qst3` option

**test28.inp**

HF/STO-3G single-point energy for  $\text{CH}_2\text{O}$  in water (*Gaussian* Test Job 616) using `LVERB=.TRUE.` to obtain the checkpoint file `g03_morb.chk` which contains the molecular orbital guess for the next test

**test29.inp**

CASSCF/STO-3G analytical gradients for the ground electronic state of  $\text{CH}_2\text{O}$  in water (*Gaussian* Test Job 616) using the `guess=read` keyword in the route line to read the initial molecular orbitals from test28 and using the PCM keywords provided in the file `pcm.dat` (option `READPCM`)

**test30.inp**

CASSCF/STO-3G numerical gradients for the ground electronic state of  $\text{CH}_2\text{O}$  in water (*Gaussian* Test Job 616) with the `iop(1/39=-1)` keyword in the route line to use a user-provided value of differentiation step as well as using the `guess=read` keyword in the route line to read the initial molecular orbitals from test28, using the PCM keywords provided in the file `pcm.dat` (option `READPCM`), and using the `EGAS=1` option

## 10. Platforms on which *GESOL* has been tested

The current version of *GESOL* has been tested with *Gaussian 03* (Revisions D.01 and E.01) on the following platforms:

- IBM pSeries 690 and pSeries 655 Nodes (Power 4 processors), running AIX version 5.3
- SGI Altix Intel Itanium 2, running SUSE Linux 2.6.5
- IBM Blade Center AMD Opteron Linux Cluster, running SUSE Linux 2.6.5
- SGI Altix XE 1300, running SUSE Linux 2.6.16

## **11. Revision history**

Version 2008 is the first version.