(5 December 2014)

General Atomic and Molecular Electronic Structure System

GAMESS User's Guide
Department of Chemistry
Iowa State University
Ames, IA 50011

literature citations:

"General Atomic and Molecular Electronic Structure System"
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, 1347-1363(1993)
doi:10.1002/jcc.540141112

"Advances in electronic structure theory:
GAMESS a decade later"
M.S.Gordon, M.W.Schmidt
Chapter 41, pp 1167-1189, in
"Theory and Applications of Computational Chemistry,
the first forty years"
C.E.Dykstra, G.Frenking, K.S.Kim, G.E.Scuseria, editors
Elsevier, Amsterdam, 2005.

http://www.msg.chem.iastate.edu/GAMESS/GAMESS.html

Contents of this manual:
        Section 1 - INTRO.DOC - Overview
        Section 2 - INPUT.DOC - Input Description
        Section 3 - TESTS.DOC - Input Examples
        Section 4 - REFS.DOC - Further Information
        Section 5 - PROG.DOC - Programmer's Reference
        Section 6 - IRON.DOC - Hardware Specifics

Contents of Section 1:

Graphical display of results is possible using MacMolPlt, a
back end visualizer of results contained in GAMESS' output
files. MacMolPlt can be downloaded freely from
     http://code.google.com/p/wxmacmolplt

Input files can be prepared using MacMolPlt, or using a
program named Avogadro, which is a molecule builder:
     http://avogadro.openmolecules.net

GAMESS itself, and both graphics programs, run on all
common desktop platforms: MAC OS X, Linux, or Windows.
There is a simple batch queue program named GamessQ to
schedule GAMESS runs on a desktop:
     http://www.msg.chem.iastate.edu/GAMESS/GamessQ
Of course, GAMESS is most often run on a dedicated computer
facility instead of your desktop, but all users can benefit
from installing the two graphics programs on their desktop.

Movies showing how to use GAMESS and GamessQ on desktop
platforms, and other information about what GAMESS can do
are at Jan Jensen's blog,
     http://molecularmodelingbasics.blogspot.com

Capabilities

A wide range of quantum chemical computations are possible using GAMESS, which

1. Calculates RHF, UHF, ROHF, GVB, or MCSCF self-consistent field molecular wavefunctions.

2. Calculates the electron correlation energy correction for these SCF wavefunctions using
   a) Density Functional Theory (DFT),
   b) Valence Bond Theory (VB)
   c) Configuration Interaction (CI),
   c) Many Body Perturbation Theory (MP2),
   e) coupled-cluster (CC) or Equation of Motion CC (EOM-CC) methodologies.
   f) for MCSCF, generates R12 basis set corrections by an interface to the MPQC package.
   See the summary table below for valid combinations.

3. Calculates semi-empirical MNDO, AM1, or PM3 models using RHF, UHF, ROHF, or GVB wavefunctions.

4. Calculates analytic energy gradients for any of the SCF wavefunctions, DFT or TD-DFT, closed or open shell MP2, or closed shell reference CI.

5. Optimizes molecular geometries using the energy gradient, using internal or Cartesian coordinates.

6. Searches for saddle points (transition states) on the potential energy surface.

7. Computes the energy hessian, and thus normal modes, vibrational frequencies, and IR intensities. Raman activities are a follow-up option.

8. Obtains anharmonic vibrational frequencies and intensities (fundamentals or overtones).

9. Traces the intrinsic reaction path from the saddle point towards products, or back to reactants.

10. Traces gradient extremal curves, which may lead from one stationary point such as a minimum to another, which might be a saddle point.

11. Follows the dynamic reaction coordinate, a classical

mechanics trajectory on the potential energy surface.
This is also known as "direct dynamics".

12. Computes excited state energies, wavefunctions, and
    transition dipole moments at various levels:
       a. SCF (e.g. ROHF or MCSCF)
       b. CIS (RHF plus single excitations)
       c. much more general CI functions
       d. time dependent DFT (or TDHF)
       e. Equation of Motion-Coupled Cluster
    with gradients for SCF, CIS, TD-DFT and GUGA CI.

13. Searches for the minimum energy crossing point
    between two intersecting potential energy surfaces,
    which have different spin or space symmetry.

14. Finds conical intersections between surfaces of the
    same spin and space symmetry, at CIS, TDDFT, or MCSCF
    levels. MCSCF-level non-adiabatic coupling matrix
    elements (NACME) between these states may be found.

15. Evaluates relativistic effects, including
       a. scalar corrections, via infinite order two
          component, or 2nd or 3rd order Douglas-Kroll
          transformations. Gradients are available.
       b. spin-orbit coupling matrix elements and the
          resulting spin-mixed wavefunctions.

16. Evaluates the static linear polarizability and the
    first and second order hyperpolarizabilities for all
    wavefunctions, by applying finite electric fields.

17. Evaluates both the static and frequency dependent
    polarizabilities for various non-linear optical
    processes, by analytic means, for RHF wavefunctions.
    Nuclear derivatives of the polarizabilities lead to
    analytic Raman and hyperRaman spectra, also for RHF.
    Imaginary frequency dependent polarizabilities can
    also be obtained, again for RHF only.

18. Obtains localized orbitals by the Foster-Boys,
    Edmiston-Ruedenberg, or Pipek-Mezey methods, with
    optional SCF or MP2 energy analysis of the LMOs.

19. Calculates the following molecular properties:
       a. dipole, quadrupole, and octupole moments
       b. electrostatic potential
       c. electric field and electric field gradients
       d. electron density and spin density

    e. Mulliken and Lowdin population analysis
    f. virial theorem and energy components
    g. Stone's distributed multipole analysis

20. Models solvent effects by discrete particles
    a. effective fragment potentials (EFP)
  or by various continuum models
    b. polarizable continuum model (PCM)
    c. solvation model density (SMD),
      a reparameterization of PCM
    d. surface and simulation of volume polarization
      for electrostatics (SS(V)PE)
    e. conductor-like screening model (COSMO)
    f. self-consistent reaction field (SCRF)
  It is possible to make a layer model consisting of QM
  atoms, surrounded by EFP particles, embedded in PCM.

21. Performs all-electron calculations based on the
    Fragment Molecular Orbital (FMO) method.

22. Models the formation of aperiodic polymers with the
    Elongation Method.

23. Perform QM/MM style HF, DFT, GVB, MCSCF, MP2 and
    TDDFT calculations, using the integrated QuanPol
    program.

24. When combined with the plug-in TINKER molecular
    mechanics program, performs Surface IMOMM (SIMOMM)
    or IMOMM QM/MM type simulations. Download from
    http://www.msg.chem.iastate.edu/GAMESS/GAMESS.html

25. When combined with the plug-in NEO program (Nuclear
    Electron Orbitals), performs quantum mechanics
    computations of nuclear structure. NEO's code is
    included with GAMESS source distributions, see
    the directory ~/gamess/qmnuc.

26. When combined with the plug-in XMVB program,
    performs valence bond calculations. Please contact
    Professor Wei Wu of Xiamen University for more
    information: http://ctc.xmu.edu.cn/xmvb/index.html.

27. When combined with the plug-in NBO program, performs
    Natural Bond Orbital analyses. This program is
    available at http://www.chem.wisc.edu/~nbo6, for a
    modest license fee.

28. Interfaces to the Atoms in Molecules Package, for

        the Quantum Theory of Atoms in Molecules analysis.
        See http://aim.tkgristmill.com/index.html

Many of these calculations may be performed in parallel!

A quick summary of the current program capabilities is given below:

SCFTYP=	RHF	ROHF	UHF	GVB	MCSCF
	---	----	---	---	-----
SCF energy	CDFpEP	CDFpEP	CDFpEP	CD-pEP	CDFpEP
SCF analytic gradient	CDFpEP	CDFpEP	CDFpEP	CD-pEP	CDFpEP
SCF analytic Hessian	CDFp--	CDFp--	CDFp--	CD-p--	-D-p-
VB energy	C-----	C-----			
MP2 energy	CDFpEP	CDFpEP	CD-pEP	------	CD-pEP
MP2 gradient	CDFpEP	-D-pEP	CD-pEP	------	------
CI energy	CDFp--	CD-p--	------	CD-p--	CD-p--
CI gradient	CD----	------	------	------	------
CC energy	CDFpE-	CDF-E-	------	------	------
EOMCC excitations	CD--E-	CD--E-	------	------	------

semi-empirical models:

	RHF	ROHF	UHF	GVB	MCSCF
DFT energy	CDFpEP	CD-pEP	CDFpEP	n/a	n/a
DFT gradient	CDFpEP	CD-pEP	CDFpEP	n/a	n/a
DFT Hessian	CD-p--	CD-p--	CD-p--	n/a	n/a
DFTB energy	yes/F	------	yes	n/a	n/a
DFTB gradient	yes/F	------	yes	n/a	n/a
DFTB Hessian	yes	------	yes	n/a	n/a
TD-DFT energy	CDFpEP	------	CDFp--	n/a	n/a
TD-DFT gradient	CDFpEP	------	------	n/a	n/a
MOPAC energy	yes	yes	yes	yes	n/a
MOPAC gradient	yes	yes	yes	no	n/a

```
C= conventional storage of AO integrals on disk
D= direct evaluation of AO integrals whenever needed
F= Fragment Molecular Orbital methodology is enabled.
   "F" pertains to the gas phase; for FMO with PCM or EFP
    there are further restrictions not specified here.
p= parallel execution
E= Effective Fragment Potential discrete solvation
P= Polarizable Continuum Model continuum solvation
        "yes" for MOPAC means "no" for FMO.
```

Numerical gradients and fully or partly numerical Hessians are available for any energy or gradient in this table.

History of GAMESS

GAMESS was put together from several existing quantum chemistry programs, particularly HONDO, by the staff of the National Resources for Computations in Chemistry. The NRCC project (1 Oct 77 to 30 Sep 81) was funded by NSF and DOE, and was limited to the field of chemistry. The NRCC staff added new capabilities to GAMESS as well. Besides providing public access to the code on the CDC 7600 at the site of the NRCC (the Lawrence Berkeley Laboratory), the NRCC made copies of the program source code (for a VAX) available to users at other sites. The original citation for this program was

M. Dupuis, D. Spangler, and J. J. Wendoloski
National Resource for Computations in Chemistry
Software Catalog, University of California:
Berkeley, CA (1980), Program QG01

This manual is a completely rewritten version of the original documentation for GAMESS. Any errors found in this documentation, or the program itself, should not be attributed to the original NRCC authors.

The present version of the program has undergone many changes since the NRCC days. This occurred at North Dakota State University from 1982 up to 1992, and now continues at Iowa State University to the present.

It would be difficult to overestimate the contributions Michel Dupuis has made to this program, in its original form, and since. This includes the donation of code from HONDO, and numerous suggestions for other improvements.

The continued development of this program from 1982 on can be directly attributed to the nurturing environment provided by Professor Mark Gordon, at North Dakota State and then Iowa State University.

It is important to also single out Professor Emeritus Klaus Ruedenberg of Iowa State University, whose group is responsible for the determinant technology lying underneath the MCSCF programs in GAMESS.

Even when our students and postdocs leave Iowa State, many continue to make contributions to GAMESS. In addition, we have also included many codes developed in

other groups over the years, so that the list of authors of GAMESS is actually much longer than the author list of the 1993 J. Comput. Chem. article. A complete list of authors may be found at the top of every log file from a GAMESS run.

Funding of many of the developments in GAMESS from 1982 to the present time was, and is provided by the Air Force Office of Scientific Research. This has always been the backbone of the support for GAMESS.

In late 1987, NDSU and IBM reached a Joint Study Agreement. One goal of this JSA was the development of a version of GAMESS that was vectorized for the IBM 3090's Vector Facility, which was accomplished by the fall of 1988. This phase of the JSA led to a program which is also considerably faster in scalar mode as well. The second phase of the JSA, which ended in 1990, was to enhance GAMESS' scientific capabilities. These additions include analytic hessians, ECPs, MP2, spin-orbit coupling and radiative transitions, and so on.   Everyone who uses the current version of GAMESS owes thanks to IBM in general, and Michel Dupuis of IBM Kingston in particular, for their sponsorship of GAMESS during this JSA.

During the first six months of 1990, Digital awarded a Innovator's Program grant to NDSU. The purpose of this grant was to ensure GAMESS would run on the DECstation, and to develop graphical display programs. As a result, the companion programs MOLPLT, PLTORB, DENDIF, and MEPMAP were modernized for the X-windows environment, and interfaced to GAMESS. These programs now run under the X-windows environments, and many other X-windows environments as well. The ability to visualize the molecular structures, orbitals, and electrostatic potentials is a significant improvement. These graphics programs eventually formed the nucleus of the program MacMolPlt.

Parallelization of GAMESS began in 1991, with most of the early work and design strategy done by Theresa Windus. This benefited greatly from the ARPA sponsorship of the Touchstone Delta experimental computer. Message passing used the TCGMSG library of Robert Harrison in the early years, up to 1999. Parallelization of GAMESS has turned into a multi-year process as detailed below.

The DoD awarded a CHSSI grant to ISU in 1996 to extend that scalability of existing parallel methods, and more importantly develop new techniques. This brought Graham

Fletcher on board as a postdoc, and led to the introduction of the Distributed Data Interface (DDI) programming model. The first version of DDI, written at ISU, was used from June 1999 to May 2004. Ryan Olson, with help from Alistair Rendell of Australian National University, rewrote DDI entirely in C, adding optimizations for the commonplace SMP nodes, especially System V memory use. Dmitri Fedorov of the National Institute for Advanced Industrial Science and Technology added the concept of subgroups at the same time. This combined new version of DDI has been the message passing support layer for GAMESS since May 2004.

The DoE awarded a SciDAC grant to ISU in 2002 to enable additional scientific capabilities in GAMESS, with emphasis on scalable algorithms. To date, this has supported parallelization of the EFP solvent molecule, and new codes for analytic MCSCF Hessians, and open shell MP2 gradients.

Some summary of these various grants and initiatives is in order. The 1982 version of GAMESS contained roughly 80,000 lines of FORTRAN code, implementing the present five wavefunction types, and analytic nuclear gradients for each, enabling geometry optimization and transition state search, and numerically differentiated frequencies. The only electron correlation method available was GUGA based CI computation. All computations were in the gas phase.

By 2005, GAMESS had grown to roughly 650,000 lines of FORTRAN. Analytic hessian computation is now routine at the SCF levels. Electron correlation is now treated with direct determinant CI codes, and in addition perturbation theory, density functional, or coupled cluster methods (with analytic gradients for some of these) may be used. New AO integral codes, including effective core potentials are used, and direct AO integral computation is possible. Discrete and continuum models for solvated molecules are provided, and there is an associated program for surface chemistry. Additional chemistry runs are provided, such as reaction paths and dynamical trajectories, IR and Raman spectra, anharmonic vibrational corrections, static or frequency dependent polarizabilities, transition moments, and spin-orbit couplings. Scalar relativistic corrections can be applied to any computation. Improvements or complete rewrites have been made for geometry searches, SCF convergers, internal coordinates, ease of use, available basis sets, and so on. The majority of these computations can be run on parallel computers.

The rest of this section gives more specific credit to the sources of various parts of the program. The order here is partly chronological, and partly logical. If you are one of the programmers, and discover your contributions have not been detailed properly, please let us know.

* * * *

GAMESS is a synthesis, with many major modifications, of several programs. A large part of the program originate from HONDO 5.

For sp basis functions, modified Gaussian76 s,p,L shell code is used. Both the sp rotated axis integrals and the sp gradient packages were modified in 2001 by Jose Maria Sierra of Synstar Computer Services in Madrid, Spain. The sp integral routines were modified in 2003 and in 2004 by Kazuya Ishimura of the Institute for Molecular Science to use McMurchie-Davidson quadratures for the basic axes-1 integrals, after which they are rotated ala Hehre/Pople. For spd functions, the s,p,d,L shell rotated axis code written by Kazuya Ishimura of the Institute for Molecular Science is used. For integral quartets with higher angular momentum, the s,p,d,f,g Electron Repulsion Integral Calculator (ERIC) code written by Graham Fletcher at NASA/Eloret in 2004 is used, provided the total angular momentum of the quartet is no more than 5. Both rotated axis codes, the sp gradient code, and ERIC share a common, fully accurate evaluation of Fm(t) integrals, and have been tested for very small (down to 0.005) and very large (1.0d+11) Gaussian exponents. The Rys polynomial program of Michel Dupuis is used to handle the general integral case: s,p,d,f,g, or L shells. HONDO 1e- and 2e- Rys routines were redimensioned to handle up to g shells by Theresa Windus at North Dakota State University in 1991. AO integrals by Rys quadrature for energy and property values were extended to s,p,d,f,g,h,i (or L) shells between 2005-2013 by Graham Fletcher, Mike Schmidt, and Joe Ivanic.

Any sp gradient integrals are done with Jose Sierra's modified version of the Gaussian80 code due to Schlegel. The spdfg gradient package consists of Michel Dupuis' Rys Polynomial code, and was adapted into GAMESS by Brett Bode at Iowa State University in 1994.

The use of quantum fast multipole methods for avoiding long range integral evaluation in large molecules was programmed by Cheol Choi at Iowa State and at Kyungpook National University, and included in GAMESS in 2001.

The ECP code goes back to Louis Kahn, with gradient modifications originally made by K.Kitaura, S.Obara, and K.Morokuma at IMS in Japan. The code was adapted to HONDO by Stevens, Basch, and Krauss, from whence Kiet Nguyen adapted it to GAMESS at NDSU. Modifications for f functions were made by Drora Cohen and Brett Bode. This code was completely rewritten to use spdfg basis sets, to exploit shell structure during integral evaluation, and to add the capability of analytic second derivatives by Brett Bode at ISU in 1997-1998. Jose Sierra of Synstar removed the last few bugs from this in 2003.

The Model Core Potential (MCP) codes originate from the University of Alberta and the University of Kyushu. MCP energy code was interfaced to GAMESS in 2003 by Mariusz Klobukowski (UofA). Many model core potentials, and their associated valence basis sets, were added as a basis library by Mariusz in 2005. Hirotoshi Mori and Eisaku Miyoshi (KyuDai) developed the nuclear gradient code for MCP with the assistance of a JSPS grant, and this code was included in GAMESS in March 2007. The ZFK family of model core potentials for p-block elements was added to GAMESS by Toby Zeng in April 2010.

Changes in the manner of entering the basis set, and the atomic coordinates (including Z-matrix forms) are due to Jan Jensen at North Dakota State University.

The direct SCF implementation was done at NDSU, guided by a pilot code for the RHF case by Frank Jensen.

The UHF code was taught to do high spin ROHF by John Montgomery at United Technologies in 1988, who extended the DIIS converger to ROHF and the one pair GVB case. Jason Byrd and John Montgomery implemented the Constrained UHF method in 2013, note that CUHF is an alternative way to produce high spin ROHF results.

The GVB code is a heavily modified version of GVBONE.

Valence Bond theory calculations are implemented as a plug-in program named VB2000, authored by Jiabo Li, Brian Duke, and Roy McWeeny. As of spring 2012, the VB2000 source code is distributed within GAMESS source code distributions, and by default is compiled into GAMESS. Examples and program documentation are found in the vb2000 subdirectory.

The SCF for Molecular Interactions option was added to GAMESS in 1997 by Antonino Famulari, during a summer visit from the University of Milan. This two fragment code was replaced with a multi-fragment code from Maurizio Sironi of the University of Milan in 2004.

The Direct Inversion in the Iterative Subspace (DIIS) convergence procedure was implemented by Brenda Lam (then at the University of Houston) in 1986, for RHF and UHF functions. Additional GVB-DIIS cases were programmed by Galina Chaban at ISU. The approximate second order SCF converger was implemented by Galina Chaban at Iowa State University in 1995, and was provided for RHF, ROHF, GVB, and MCSCF cases. The FULLNR and FOCAS MCSCF convergers were contributed by Michel Dupuis from his HONDO program. A parallel implementation of the FULLNR converger was written by Graham Fletcher at Eloret in 2002. The Jacobi orbital rotation scheme for MCSCF orbital optimization was written by Joe Ivanic and Klaus Ruedenberg at Iowa State University in 2001.

The Ames Laboratory determinant full CI code was written by Joe Ivanic and Klaus Ruedenberg. As befits code written by an Australian living in Iowa, it was interfaced to GAMESS during an extremely cordial visit to Australia National University in January 1998. An update by Joe in October 2000 exploits Abelian point group symmetry. A general CI program based on selected determinants was added by Joe and Klaus in July 2001. After moving from Ames Laboratory at ISU to the Advanced Biomedical Computing Center of the National Cancer Institute-Frederick, Fort Detrick, Joe wrote a determinant based program for second order CI, in 2002. In early 2003, Joe added the Occupation Restricted Multiple Active Space determinant CI program, again written at NCI.

The GUGA CI is based on Brooks and Schaefer's unitary group program which was modified to run within GAMESS, using a Davidson eigenvector method written by Steve Elbert.

Programming of the GUGA analytic CI gradient was done by Simon Webb in 1996 at Iowa State University.

The CIS gradient program was written in 2003 by Simon Webb of the Advanced Biomedical Computing Center of the National Cancer Institute in Frederick. Transition moments were added by Simon and Pooja Arora in June 2005.

     The sequential MP2 and UMP2 energy code was adapted
from HONDO in 1994 by Nikita Matsunaga at ISU. Nikita
programmed the RMP open shell energy in 1992. The ZAPT
open shell energy was programmed by Rob Bell in 1999. The
serial closed shell MP2 gradient code is also from HONDO,
and was adapted to GAMESS in 1995 by Simon Webb and Nikita
Matsunaga. In 1996, Simon Webb added the frozen core
gradient option at ISU. The parallel closed shell MP2 code
is a descendant of work for GAMESS-UK by Graham Fletcher,
Alistair Rendell, and Paul Sherwood at Daresbury. This was
adapted to GAMESS at ISU by Graham Fletcher in 1999.
Serial and parallel codes for the spin unrestricted UMP2
gradient were programmed by Christine Aikens at ISU, in
2002. Christine Aikens added a parallel spin-restricted
open shell (ZAPT) gradient code in 2005. Programs for
parallel closed shell MP2 energy (2006) and gradient (2007)
using disk storage were written by Kazuya Ishimura at the
Institute for Molecular Science (IMS) in Okazaki. The
parallel Resolution of the Identity MP2 program by Michio
Katouda, also from IMS, was added in 2010.

     Credits for multiconfigurational PT follow. Haruyuki
Nakano, then at the University of Tokyo, interfaced his
multireference MCQDPT code (based on CSFs) to GAMESS during
a 1996 visit to ISU, this is MRPT=MCQDPT. Parallelization
of the Tokyo multireference PT code was done by Hiroaki
Umeda at Mie University, and included into GAMESS in 2001.
A determinant based code which is equivalent to MRMP/MCQDPT
was programmed in 2005 by Joe Ivanic of the National Cancer
Institute, this is MRPT=DETMRPT. In 2008, Haruyuki Nakano
of the University of Kyushu contributed a general MCSCF
reference quasi-degenerate perturbation theory code,
MRPT=GMCPT, which is capable of treating various non-CAS
references, including those of the ORMAS type. In 2012,
Luke Roskop of Iowa State University extended MRPT=DETMRPT
to the case of ORMAS reference functions.

     The grid-free DFT energy and gradient code was written
by Kurt Glaesemann at Iowa State University, starting from
the code of Almlof and Zheng, adding four center overlap
integrals, a gradient program, developing the auxiliary
basis option, and adding some functionals. This was
included in GAMESS in 1999.

     The grid based DFT program was introduced in 2001 at
the University of Tokyo, by Takao Tsuneda, Muneaki Kamiya,
Susumu Yanagisawa, and Dmitri Fedorov. The original program
is from Nevin Oliphant, Hideo Sekino, and Rod Bartlett at
QTP. Many improvements were made to this early program at

U. Tokyo: using point group symmetry, switching from coarse
to fine grids, functional development, and parallelization.
Sarom Sok at ISU added many new functionals in 2007, 2008,
and 2009, some with the help of Huub van Dam's density
functional repository. Sarom added the Truhlar group's
meta-GGA M06 and M08 functionals in 2008 and 2009, using
source code from U.Minnesota. Roberto Peverati of the
University of Zurich added Grimme's dispersion correction
in 2008. Roberto added "wB97" range separated GGA, "B97"
style GGA and metaGGA, and B2-PLYP in 2009, and he added
the M11 metaGGA family in spring 2011. Federico Zahariev
at ISU included the TPSS family of meta-GGAs in 2008 and
2009. Kiet Nguyen at Wright-Patterson AFB added CAM-B3LYP
in 2009. The HPTi project (Jean-Philippe Blaudeau, Shawn
Brown, Mike Lasinksi, Nick Romero, Anthony Yau) enabled the
use of Lebedev or Standard Grid-1 grids in April 2008, and
Janssen's grids in May 2009.

    The time dependent DFT program originated in the group
of Takao Tsuneda at the University of Tokyo, and was
included into GAMESS in the fall of 2006 by Mahito Chiba at
AIST in Tsukuba. This group also included the "long range
correction" option (aka "range separation") for both ground
and excited states. The analytic TD-DFT gradient for
singlet excited states from a closed shell reference was
added by Mahito Chiba in August 2007. Mahito Chiba, in
collaboration with Dmitri Fedorov, also developed FMO
functionality in TD-DFT energies. The TD-DFT energy for
UHF ground states was added by Soohaeng Yoo at Iowa State,
in February 2008. Tamm/Dancoff approximation coding was
done by Federico Zahariev at ISU in 2010. The HPTi project
parallelized the closed shell TD-DFT energy and gradient
programs in April 2008. Sarom Sok and Federico Zahariev
have developed higher density derivatives for many
functionals, allowing them to be used in TD-DFT energies
and gradients. Federico has also developed the corrections
to the TD-DFT eigenvalue equation needed for meta-GGA
excitation energies in 2010. The two-photon absorption
cross-sections were programmed by Federico Zahariev at ISU
in spring 2012.

    TD-DFT solvation effects include EFP1 discrete
solvation, added to the closed shell TD-DFT excitation
energies in 2008 by Soohaeng Yoo, and to its gradient in
2010 by Noriyuki Minezawa at ISU. C-PCM solvent effects on
TD-DFT closed shell excitation energies were added by
Mahito Chiba in December 2008, with PCM modifications to
this gradient by Yali Wang and Hui Li in November 2009.
The combined TD-DFT/EFP/PCM solvation model was finished in

November 2010 by Nandun Thellamurege and Hui Li at U.
Nebraska.

        Incorporation of enough MOPAC version 6 routines to run
PM3, AM1, and MNDO calculations from within GAMESS was done
by Jan Jensen at North Dakota State University. The RM1
parameterization was added by Melissa Gajewski in 2010, at
U. Alberta. Caspar Steinmann interfaced MOPAC to PCM in
spring 2013, and also enabled parallel execution.

        The numerical force constant computation and normal
mode analysis was adapted from Andy Komornicki's GRADSCF
program, with decomposition of normal modes in internal
coordinates written at NDSU by Jerry Boatz.

        The code for the analytic computation of RHF Hessians
was contributed by Michel Dupuis of IBM from HONDO 7. High
and low spin restricted open shell CPHF code was written at
NDSU in 1989. The TCSCF CPHF code is the result of a
collaboration between NDSU and John Montgomery, then at
United Technologies, in 1990. Analytic IR intensities and
polarizabilities (during hessian runs) were programmed by
Simon Webb at ISU in 1995. Analytic Hessians for MCSCF
wavefunctions based on determinants were coded, and enabled
for parallel execution, by Tim Dudley at ISU, and included
into GAMESS in April 2004, with a souped-up version added
in March 2006.

        Code for Raman intensity prediction was written at
Tokyo Metropolitan University in April 2000.

        The vibrational SCF and MP2 anharmonic frequency code
for fundamental modes and overtones was written by Galina
Chaban, Joon Jung, and Benny Gerber at U.California-Irvine
and Hebrew University of Jerusalem, and included in GAMESS
in 2000. The solver was modified to perform degenerate
perturbation theory for more accurate results by Nikita
Matsunaga at Long Island University in 2001.

        Delocalized internal coordinates were implemented by
Jim Shoemaker at the Air Force Institute of Technology in
1997, and put online in GAMESS by Cheol Choi at ISU after
further improvements in 1998.

        Most of the geometry search procedures (OPTIMIZE and
SADPOINT) were developed by Frank Jensen of the University
of Aarhus. These methods are adapted to use GAMESS
symmetry, and Cartesian or internal coordinates. Numerical
differentiation of the energy to obtain gradients and

Hessians which may be used in OPTIMIZE or SADPOINT searches was programmed by Ryan Olson at ISU in 2003. The MEX procedure for searching for minimum energy crossing points between two surfaces was programmed by Jeremy Harvey and Nikita Matsunaga, and finally included into GAMESS in 2006. The non-gradient optimization (so aptly named TRUDGE) was adapted from HONDO 7 by Mariusz Klobukowski at U.Alberta, this may be more interesting for its exponent optimization option.

The intrinsic reaction coordinate pathfinder was written at North Dakota State University, and modified later for new integration methods by Kim Baldridge. The Gonzales-Schelegel IRC stepper was incorporated by Shujun Su at Iowa State, based on pilot code from Frank Jensen.

The code for the Dynamic Reaction Coordinate was developed by Tetsuya Taketsugu at Ochanomizu U. and U. of Tokyo, and added to GAMESS by him at ISU in 1994.

The two algorithms for tracing gradient extremals were programmed by Frank Jensen, now at the University of Aarhus.

The program for Monte Carlo generation of trial structures along with a simulated annealing protocol was written by Paul Day at Wright-Patterson Air Force Base. Modifications to this were made by Pradipta Bandyopadhyay at ISU, and the code was included in 2001.

The surface scanning option was implemented by Richard Muller at the University of Southern California.

Static polarizabilities for any type of energy value are bases on a code from Henry Kurtz of the University of Memphis. This uses a numerical differentiation based on application of finite electric fields. The program was added in 1992, and was modified by Sanka Ghosh to produce all tensor components in 2005.

Henry Kurtz' program for the fully analytic calculation of static and frequency dependent polarizabilities for NLO properties for closed shell systems was included in 1994, based on a MOPAC implementation by Prakashan Korambath at U. Memphis.

An extended TDHF package for the analytic computation of static and frequency dependent polarizabilities, and also their nuclear derivatives, plus Raman and hyperRaman

spectra prediction was written by Olivier Quinet and Benoit
Champagne at the Facultes Universitaires Notre-Dame de la
Paix, and coworker Bernard Kirtman at UC-Santa Barbara.
Financial support for this was provided by Belgium. This
package was added to GAMESS in February 2005.

   Ivana Adamovic programmed the imaginary frequency
polarizability computation for closed shell functions in
2005, at ISU.

   Edmiston-Ruedenberg energy localization is done with a
version of the ALIS program "LOCL", modified at NDSU to run
inside GAMESS. Foster-Boys localization is based on a
highly modified version of QCPE program 354 by D.Boerth,
J.A.Hasmall, and A.Streitweiser. John Montgomery
implemented the Pipek/Mezey population localization. The
LCD SCF decomposition and the MP2 decomposition were
written by Jan Jensen at Iowa State in 1994.

   Point Determined Charges were implemented by Mark
Spackman at the University of New England, Australia.

   The Morokuma decomposition was implemented by Wei Chen
at Iowa State University, in 1995. The Localized Molecular
Orbital Energy Decomposition Analysis was implemented by
Peifeng Su and Hui Li at the University of Nebraska in
2009.

   The radiative transition moment and effective nuclear
charge spin-orbit coupling modules were written by Shiro
Koseki at North Dakota State University in 1990.

   The full Breit-Pauli spin-orbit coupling integral
package was written by Thomas Furlani. This code was
incorporated into GAMESS by Dmitri Fedorov at Iowa State
University in 1997, who generalized the spin-orbit coupling
matrix element code generously provided by Thomas Furlani
(restricted to an active space of two electrons in two
orbitals), with assistance from visits to ISU by Thomas
Furlani and Shiro Koseki. Dmitri Fedorov has since
generalized the full two electron approach to allow for any
spins, for more than two spin multiplicities at a time, and
a partial treatment of the the two electron terms that runs
in time similar to the one electron operator. Space and
spin symmetries are exploited to speed up the runs. Dmitri
Fedorov programmed the SO-MCQDPT options at the University
of Tokyo in 2001. Density matrix calculation for spin-
orbit coupled states was programmed by Toby Zeng and

Mariusz Klobukowski at the University of Alberta, and added
to GAMESS in April 2010.

Inclusion of relativistic effects by means of the
Douglas-Kroll transformation was developed to the third
order by Takahito Nakajima and Kimihiko Hirao at the
University of Tokyo. It was implemented into GAMESS by
Takahito Nakajima and Dmitri Fedorov, including energy,
semi-analytic nuclear gradient, and spin-orbit coupling (at
the first order, DK1). The program was included with
GAMESS in November 2003.

Inclusion of relativistic effects by the Relativistic
scheme of Elimination of Small Components (RESC) method,
was developed by Takahito Nakajima and Kimihiko Hirao at
the University of Tokyo. This code was written by Takahito
Nakajima and consequently adapted into GAMESS by Dmitri
Fedorov, who extended the methodology in March 2000 to the
computation of gradients. RESC provides both scalar (spin
free) and vector (spin-dependent) relativistic corrections.

The Normalized Elimination of Small Components (NESC)
was programmed by Dmitri Fedorov at ISU and the University
of Tokyo. Special thanks are due to Kenneth Dyall for his
assistance in providing check values. Extension of NESC to
include gradient computation was also done by Dmitri.

Finally, inclusion of scalar relativstic corrections to
an infinite order two-component (IOTC) transformation was
added in September 2010, by Maria Barysz of Nicholas
Copernicus University.

Development of the EFP method began in the group of
Walt Stevens at NIST's Center for Advanced Research in
Biotechnology (CARB) in 1988. Walt is the originator of
this method, and has provided both guidance and some early
financial support to ISU for its continued development.
Mark Gordon's group's participation began in 1989-90 as
discussions during a year Mark spent in the DC area, and
became more serious in 1991 with a visit by Jan Jensen to
CARB. At this time the method worked for the energy, and
gradient with respect to the ab initio nuclei, for one
fragment only. Jan has assisted with most aspects of the
multi-fragment development since. Paul Day at NDSU and ISU
derived and implemented the gradient with respect to
fragments, and programmed EFP geometry optimization, from
1992-1994. Wei Chen at ISU debugged many parts of the EFP
energy and gradient, developed the code for following IRCs,
improved geometry searches, and fitted much more accurate

repulsive potentials, from 1995-1996. Simon Webb at ISU
programmed the current self-consistency process for the
induced dipoles in 1994. The EFP method was sufficiently
developed, tested, and described, to be released in
September 1996, with an RHF level potential for water.
Code for charge penetration was added by Mark Freitag in
2001, and made numerically stabile by Lyuda Slipchenko in
2006. Ivana Adamovic included a DFT level EFP for water in
2002. Parallelization of the EFP codes was done by Heather
Netzloff in 2005.

    The second EFP theory (called EFP2) was begun in 1996
by Jan Jensen, who programmed an analytic formula for the
exchange repulsion. Hui Li replaced this with a faster,
more accurate code in 2005. Ivana Adamovic programmed a
dispersion term for EFP2 in 2005. Hui Li added the charge
transfer term for EFP2 in 2005.

    Two other methods using the EFP model are available. A
combination of EFP + PCM energies (an onion-like solution
model) was programmed by Pradipta Bandyopadhyay in 2000.
The use of EFPs to model biological systems, including a
boundary across a covalent bond, was coded at the
University of Iowa in 2000, by Jan Jensen, Visvaldas
Kairys, and Hui Li.

    The SCRF solvent model was implemented by Dave Garmer
at CARB, and was adapted to GAMESS by Jan Jensen and Simon
Webb at Iowa State University.

    The COSMO model was developed by Andreas Klamt and Kim
Baldridge, starting at the San Diego Supercomputer Center,
and later at University of Zurich. It was included into
GAMESS by Laura Brovold in March 2000 during a visit to
Ames. Subsequent additions were made by Yohann Potier and
Roberto Peverati, at the University of Zurich, and included
in GAMESS in June 2010.

    The PCM code originated in the group of Jacopo Tomasi
at the University of Pisa. Benedetta Mennucci was
instrumental in interfacing the original D-PCM code to
GAMESS in 1997, and answering many technical questions
about the code, the methodology, and the documentation. In
2000, Benedetta Menucci provided code implementing an
improved IEF solver for the PCM surface charges. The
changes to implement iterative solution of the PCM
equations for large molecules, and to provide an accurate
nuclear gradient were carried out by Hui Li and Jan Jensen
at the University of Iowa in 2001-2004, along with the

parallelization. This included implementation of two new surface tessellation schemes, GEPOL-AS and GEPOL-RT. Hui and Jan also implemented the Conductor-PCM method, and extended the PCM methodology to all types of SCF functions. Hui Li's research group at the University of Nebraska implemented the following improvements: FIXPVA tessellation with smooth switching functions for reliable geometry optimizations (Peifeng Su, 2008), extension of FIXPVA to cavitation, repulsion, and dispersion (2009), heterogenous CPCM (Dejun Si, 2009), closed shell PCM/TDDFT gradients (Yali Wang, 2009), closed shell PCM/MP2 gradients (Dejun Si, 2010), open shell PCM/MP2 gradients (Dejun Si, September 2010), and combined EFP/PCM solvation for all single reference MP2 gradients (Nandun Thellamurege and Dejun Si, November 2010). The SMD modifications to the PCM model are due to Alek Marenich, Junjun Liu, Chang-Guo Zhan, Christopher Cramer, and Don Truhlar at U. Minnesota (November 2010).

   The Surface and Volume Polarization for Electrostatics continuum solvation model is written by Dan Chipman of Notre Dame University, using several integral routines written by Michel Dupuis for the SVP model included in HONDO. The SVP model was added to GAMESS in June 2005.

   The SIMOMM model for surface chemistry is based on the Tinker program of Jay Ponder's group, and is available as a plug-in option. The treatment is QM embedded in a MM background. The coding for this was done by Jim Shoemaker at the Air Force Institute of Technology, and finished by Cheol Ho Choi at ISU. The interface to GAMESS was completed in 1998.

   The Coupled-Cluster (CC) and Equation of Motion Coupled-Cluster (EOMCC) programs included in GAMESS are due to Piotr Piecuch, Karol Kowalski, Marta Wloch, Jeffrey Gour, and Jesse Lutz of Michigan State University (MSU), and Stanislaw A. Kucharski and Monika Musial of the University of Silesia. In addition to a number of standard CC and EOMCC methods, including the older CCSD, CCSD(T), and EOMCCSD approaches, the CC codes incorporated in GAMESS are capable of performing renormalized (R) and completely renormalized (CR) CCSD[T] and CCSD(T) calculations for the ground state, the ground-state calculations employing the rigorously size extensive completely renormalized non-iterative triples CR-CCSD(T)_L = CR-CC(2,3) approach. The combined corrections due to triply and quadruply excited clusters are available in the factorized forms of the CCSD(TQ), renormalized CCSD(TQ), and completely

renormalized CCSD(TQ) models. For excited states, completely renormalized EOMCCSD(T) (CR-EOMCCSD(T)) and EOM-CR-CC(23) calculations are possible. Electron attachment and detachments (including excitations) are available as IP-EOM and EA-CC methods. The one-body reduced density matrices, dipole moments, transition dipole moments, and oscillator strengths are available at the CCSD and EOMCCSD levels, for RHF. The ground-state CC, R-CC, and CR-CC programs were initially incorporated into GAMESS in May 2002. The excited-state EOMCC and CR-EOMCC programs were incorporated in April 2004. Quadruples corrections and CCSD/EOM-CCSD density matrices were added in June 2005. The CR-CC(2,3) ground-state approach was added in January 2006. Parallel computation of CCSD and CCSD(T) for closed shell references was enabled by Ryan Olson and Jonathan Bentz at Iowa State, in October 2006. Open shell CCSD and CR-CCL based on ROHF reference orbitals was added in May 2007. CR-EOML and IP-EOMCC2/EA-EOMCC2 were included in October 2009, and active triples for IP/EA calculations were finished in September 2010. Open shell reference EOM-CCSD was completed in October 2011. All of these programs were developed with the support of the US Department of Energy, Office of Basic Energy Sciences, SciDAC Computational Chemistry Program and the Chemical Sciences, Geosciences, and Biosciences Division. Additional support has been provided by the NSF's ITR program and the Alfred P. Sloan Foundation.

The GIAO computation of NMR properties for closed shell molecules was programmed by Mark Freitag at Iowa State University, and included in GAMESS in November 2003.

The code for the Fragment Molecular Orbital (FMO) method incorporated and distributed as a part of the standard GAMESS package since May 2004 is being developed at the National Institute of Advanced Industrial Science and Technology (AIST, Japan) by Dmitri Fedorov and Kazuo Kitaura. The FMO method is the successor of the EDA scheme developed by K. Kitaura and K. Morokuma (known in GAMESS as Morokuma-Kitaura decomposition), however, the FMO code was written independently. In GAMESS only the full FMO method is incorporated whereas in the literature one can also find a simplified approach suited for molecular crystals. Since "FMO" is also used to mean "Frontier Molecular Orbitals" and the concept of fragments is also introduced in the EFP method (see above), it is stressed here that the FMO method bears no relation to either of the two methods, that is to say, it is independent of the two, but might be combined

with either of them in the future just as EFPs are used in
e.g. RHF.

   The Nuclear Electron Orbital (NEO) plug-in code is
developed in the group of Sharon Hammes-Schiffer at
Pennsylvania State University, with programming by Simon P.
Webb, Tzvetelin Iordanov, Mike Pak, and Chet Swalina. The
initial release in 2006 permits HF and MP2 level treatment
of nuclear wavefunctions.

   The elongation method, coded and linked to the standard
GAMESS package since April 2006, is a method to mimic the
mechanism of the polymerization/copolymerization in
experiment. Attacking monomers approach a starting chain,
one by one and the electron structure is determined in the
interactive region. Thus, one can perform very efficient
calculations for the electronic structure of huge random
(aperiodic) polymers. The elongation method was first
proposed by A. Imamura and Y. Aoki in 1990s. The present
code was written by Feng Long Gu, Jacek Korchowiec, Marcin
Makowski, and Yuriko Aoki at the Department of Molecular
and Material Sciences, Faculty of Engineering Sciences, at
Kyushu University.

   The Divide and Conquer SCF, MP2, and CCSD programs were
developed at Waseda University, and were included in GAMESS
in January 2009. The code was written by Masato Kobayashi,
Tomoko Akama, and Hiromi Nakai.

   The quantum chemistry polarizable force field program
(QuanPol) was written by Hui Li, Nandun Thellamurege and
Dejun Si at the University of Nebraska-Lincoln. These
authors finished the initial implementation of QuanPol in
August 2011, under an NSF support.

   Many of the options just mentioned have been programmed
to run in parallel, on systems ranging from Linux clusters
to high-end parallel systems. The same software interface
sits between the quantum chemistry in GAMESS and any such
hardware, namely the Distributed Data Interface (DDI).
This implements a mechanism for using the memory of the
entire system to store the large arrays appearing in
quantum chemistry codes. The first version of DDI was due
to Graham Fletcher and Mike Schmidt, introduced in 1999.
The second version of DDI is due to Ryan Olson of ISU, and
Alistair Rendell of the Australian National University, and
includes optimizations for SMP systems, along with other
improvements for some high end systems. The second version
also includes the 'group' scheme, presently used only in

FMO jobs. This DDI was introduced into GAMESS in April
2004, with public release in June 2004.

Distribution Policy

To get a copy, please fill out the application form
available at
      http://www.msg.chem.iastate.edu/GAMESS/GAMESS.html

Persons receiving copies of GAMESS are requested to
acknowledge that they will not make copies of GAMESS for
use at other sites, or incorporate any portion of GAMESS
into any other program, without receiving permission to do
so from ISU. If you know anyone who wants a copy of
GAMESS, please refer them to the web site above, for the
most up to date version available.

No large program can ever be guaranteed to be free of
bugs, and GAMESS is no exception. If you would like to
receive an updated version (fewer bugs, and with new
capabilities), simply return to the web site mentioned.
You should probably allow a half year or so to pass for
enough significant changes to accumulate. The web page
always contains a short synopsis of the most recent
changes.

Input Philosophy

Input to GAMESS may be in upper or lower case. All input groups begin with a $ sign in column 2, meaning exactly column 2 or else it is not detected, followed by a name identifying that group. There are three types of input groups in GAMESS:

1. A pseudo-namelist, free format, keyword driven group. Almost all input groups fall into this first category.

2. A free format group which does not use keywords. The first line of these will contain only the group name, followed by several lines of positional data usually with no keywords, and a last line containing " $END" only. The only members of this category are $DATA, $ECP, $MCP, $GCILST, $POINTS, $STONE, and the EFP related data $EFRAG, $FRAGNAME, $FRGRPL, and $DAMPGS.

3. Formatted data. This data is NEVER typed by the user, but rather is generated in the correct format by some earlier GAMESS run. Like category 2, the first line contains only the group name, and the last line is a separate $END line.

Type 1 groups may have keyword input on the same line as the group name, and the $END may appear anywhere.

Because each group has a unique name, the groups may be given in any order desired. In fact, multiple occurrences of category 1 groups are permissible.

                              * * *

Most of the groups can be omitted if the program defaults are adequate. An exception is $DATA, which is always required. A typical free format $DATA group is

```
 $DATA
STO-3G test case for water
CNV       2

OXYGEN        8.0
    STO  3

HYDROGEN      1.0    -0.758      0.0      0.545
    STO  3
```

    $END

     Here, position is important. For example, the atom
name must be followed by the nuclear charge and then the
x,y,z coordinates. Note that missing values will be read
as zero, so that the oxygen is placed at the origin.
The zero Y coordinate must be given for the hydrogen,
so that the final number is taken as Z.

     The free format scanner code used to read $DATA is
adapted from the ALIS program, and is described in the
documentation for the graphics programs which accompany
GAMESS. Note that the characters ;>! mean something
special to the free format scanner, and so use of these
characters in $DATA and $ECP should probably be avoided.

     Because the default type of calculation is a single
point (geometry) closed shell SCF, the $DATA group shown
is the only input required to do a RHF/STO-3G water
calculation.

                            * * *

     As mentioned, the most common type of input is a
namelist-like, keyword driven, free format group. These
groups must begin with the $ sign in column 2, but have no
further format restrictions. You are not allowed to
abbreviate the keywords, or any string value they might
expect. They are terminated by a $END string, appearing
anywhere. The groups may extend over more than one
physical card. In fact, you can give a particular group
more than once, as multiple occurrences will be found and
processed. We can rewrite the STO-3G water calculation
using the keyword groups $CONTRL and $BASIS as

 $CONTRL SCFTYP=RHF RUNTYP=ENERGY $END
 $BASIS GBASIS=STO NGAUSS=3 $END
 $DATA
STO-3G TEST CASE FOR WATER
Cnv   2

Oxygen   8.0   0.0   0.0   0.0
Hydrogen   1.0   -0.758   0.0   0.545
 $END

     Keywords may expect logical, integer, floating point,
or string values. Group names and keywords never exceed 6
characters. String values assigned to keywords never

exceed 8 characters. Spaces or commas may be used to
separate items:

 $CONTRL MULT=3 SCFTYP=UHF,TIMLIM=30.0 $END

      Floating point numbers need not include the decimal,
and may be given in exponential form, i.e. TIMLIM=30,
TIMLIM=3.E1, and TIMLIM=3.0D+01 are all equivalent.

      Numerical values follow the FORTRAN variable name
convention. All keywords which expect an integer value
begin with the letters I-N, and all keywords which expect
a floating point value begin with A-H or O-Z. String or
logical keywords may begin with any letter.

      Some keyword variables are actually arrays. Array
elements are entered by specifying the desired subscript:

 $SCF NO(1)=1 NO(2)=1 $END

      When contiguous array elements are given this may be
given in a shorter form:

 $SCF NO(1)=1,1 $END

      When just one value is given to the first element of
an array, the subscript may be omitted:

 $SCF NO=1 NO(2)=1 $END

      Logical variables can be .TRUE. or .FALSE. or .T.
or .F. The periods are required.

      The program rewinds the input file before searching
for the namelist group it needs. This means that the
order in which the namelist groups are given is
immaterial, and that comment cards may be placed between
namelist groups.

      Furthermore, the input file is read all the way
through for each free-form namelist so multiple occurrences
will be processed, although only the LAST occurrence of a
variable will be accepted. Comment fields within a
free-form namelist group are turned on and off by an
exclamation point (!). Comments may also be placed after
the $END's of free format namelist groups. Usually,
comments are placed in between groups,

 $CONTRL SCFTYP=RHF RUNTYP=GRADIENT $END

```
--$CONTRL EXETYP=CHECK $END
  $DATA
molecule goes here...
```

     The second $CONTRL is not read, because it does not
have a blank and a $ in the first two columns. Here a
careful user has executed a CHECK job, and is now running
the real calculation. The CHECK card is now just a
comment line.

                              * * *

     The final form of input is the fixed format group.
These groups must be given IN CAPITAL LETTERS only! This
includes the beginning $NAME and closing $END cards, as
well as the group contents. The formatted groups are
$VEC, $HESS, $GRAD, $DIPDR, and $VIB. Each of these is
produced by some earlier GAMESS run, in exactly the
correct format for reuse. Thus, the format by which they
are read is not documented in section 2 of this manual.

                              * * *

     Each group is described in the Input Description
section. Fixed format groups are indicated as such, and
the conditions for which each group is required and/or
relevant are stated.

     There are a number of examples of GAMESS input given
in the Input Examples section of this manual.

Input Checking

Because some of the data in the input file may not be
processed until well into a lengthy run, a facility to
check the validity of the input has been provided. If
EXETYP=CHECK is specified in the $CONTRL group, GAMESS
will run without doing much real work so that all the
input sections can be executed and the data checked for
correct syntax and validity to the extent possible. The
one-electron integrals are evaluated and the distinct row
table is generated. Problems involving insufficient
memory can be identified at this stage. To help avoid the
inadvertent absence of data, which may result in the
inappropriate use of default values, GAMESS will report
the absence of any control group it tries to read in CHECK
mode. This is of some value in determining which control
groups are applicable to a particular problem.

The use of EXETYP=CHECK is HIGHLY recommended for the
initial execution of a new problem.

Program Limitations

GAMESS can use an arbitrary Gaussian basis of spdfghi type for computation of the energy, spdfg for analytic nuclear gradients, or spd for analytic nuclear hessians, in the gas phase. Additional restrictions apply, for example, if solvent models, core potentials, scalar or spin-orbit relativistic effects are used.

This program is limited to a total of 2,000 atoms. The total number of symmetry unique basis set shells cannot exceed 5,000, containing no more than 20,000 Gaussian primitives. Each contraction must contain no more than 30 Gaussians. The total number of contracted basis functions, or AOs, cannot exceed 8192. You may use up to 1050 effective fragments, of at most 5 types, containing no more than 2000 multipole/polarizability/other expansion points.

In practice, you will probably run out of CPU time or disk storage before you encounter any of these limitations. See Section 5 of this manual for information about changing any of these limits, or minimizing program memory use.

Except for these limits, the program is basically dimension limitation free. Memory allocations other than these limits are dynamic, from the storage requested by the input.

Restart Capability

The program checks for CPU time, and will stop if time is running short. Restart data are printed and punched out automatically, so the run can be restarted where it left off.

At present all SCF modules will place the current orbitals on the punch file if the maximum number of iterations is reached. These orbitals may be used in conjunction with the GUESS=MOREAD option to restart the iterations where they quit. Also, if the TIMLIM option is used to specify a time limit just slightly less than the job's batch time limit, GAMESS will halt if there is insufficient time to complete another full iteration, and the current orbitals will be punched.

When searching for equilibrium geometries or saddle points, if time runs short, or the maximum number of steps is exceeded, the updated hessian matrix is punched for restart. Optimization runs can also be restarted with the direct access file DICTNRY. See $STATPT for details.

Force constant matrix runs can be restarted from cards. See the $VIB group for details.

The two electron integrals may be reused. The Newton-Raphson formula tape for MCSCF runs can be saved and reused.

* * * *

The binary file restart options are rarely used, and so may not work well (or at all). Restarts which change the card input (adding a partially converged $VEC, or updating the coordinates in $DATA, etc.) are far more likely to be successful than restarts from the DAF file.