(25 March 2000)

General Atomic and Molecular Electronic Structure System

GAMESS User's Guide

as prepared at
Department of Chemistry
Iowa State University
Ames, IA 50011

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

Original program assembled by the staff of the NRCC:

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 version of GAMESS is described in

M.W.Schmidt, K.K.Baldridge, J.A.Boatz, S.T.Elbert,
M.S.Gordon, J.H.Jensen, S.Koseki, N.Matsunaga,
K.A.Nguyen, S.J.Su, T.L.Windus, M.Dupuis, J.A.Montgomery
J.Comput.Chem. 14, 1347-1363(1993)

Another information source is
http://www.msg.ameslab.gov/GAMESS/GAMESS.html

There is a GAMESS discussion group originally started
by Gotthard Saghi-Szabo at the University of Maryland. For info, see
http://mineral.umd.edu/gamess-users
The discussions are archived at: http://lacebark.ntu.edu.au/gamess/

Questions about GAMESS may be addressed to:

Mike Schmidt = mike@si.fi.ameslab.gov = 515-294-9796

E-mail is much, much, much preferred to phone calls!

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 CI or MP2 corrections to the energy of these SCF functions.
3. Calculates semi-empirical MNDO, AM1, or PM3 RHF, UHF, or ROHF wavefunctions.
4. Calculates analytic energy gradients for all SCF wavefunctions, plus closed shell MP2 or CI.
5. Optimizes molecular geometries using the energy gradient, in terms of Cartesian or internal coords.
6. Searches for potential energy surface saddle points.
7. Computes the energy hessian, and thus normal modes, vibrational frequencies, and IR intensities. The Raman intensities are an optional follow on job.
8. Obtains anharmonic vibrational frequencies and intensities (fundamentals or overtones).
9. Traces the intrinsic reaction path from a saddle point to reactants or products.
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.
12. Computes radiative transition probabilities.
13. Evaluates spin-orbit coupled wavefunctions.
14. Applies finite electric fields, extracting the molecule's linear polarizability, and first and second order hyperpolarizabilities.
15. Evaluates analytic frequency dependent non-linear optical polarizability properties, for RHF functions.
16. Obtains localized orbitals by the Foster-Boys, Edmiston-Ruedenberg, or Pipek-Mezey methods, with optional SCF or MP2 energy analysis of the LMOs.
17. 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
18. Models solvent effects by
    a. effective fragment potentials (EFP)
    b. polarizable continuum model (PCM)
    c. self-consistent reaction field (SCRF)

19. When combined with the add-on TINKER molecular mechanics program, performs Surface IMOMM or IMOMM QM/MM type simulations. (Anonymous FTP to www.msg.ameslab.gov, directory tinker, file tinker.tar.Z, see simomm.doc contained therein).

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

SCFTYP=	RHF	ROHF	UHF	GVB	MCSCF
Energy	CDP	CDP	CDP	CDP	CDP
analytic Gradient	CDP	CDP	CDP	CDP	CDP
numerical Hessian	CDP	CDP	CDP	CDP	CDP
analytic Hessian	CDP	CDP	-	CDP	-
MP2 energy	CDP	CDP	CDP	-	C
MP2 gradient	CDP	-	-	-	-
CI energy	CDP	CDP	-	CDP	CDP
CI gradient	CD	-	-	-	-
MOPAC Energy	yes	yes	yes	yes	-
MOPAC gradient	yes	yes	yes	-	-

C= conventional storage of AO integrals on disk
D= direct evaluation of AO integrals
P= parallel execution

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.

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 prior to 1992, and now continues at Iowa State University. A number of persons (some of whom have now left the Gordon group) have made contributions: Jerry Boatz, Kim Baldridge, and Shiro Koseki at NDSU; Kiet Nguyen, Jan Jensen, Theresa Windus, Nikita Matsunaga, Shujun Su, Brett Bode, Simon Webb, Wei Chen, Tetsuya Taketsugu, Galina Chaban, and Dmitri Fedorov, Cheol Choi, and Rob Bell at ISU; plus

Frank Jensen at Odense U.,
Mariusz Klobukowski at U.Alberta,
Henry Kurtz at U.Memphis,
Brenda Lam at U.Ottawa,
John Montgomery at United Technologies,
Haruyuki Nakano at U.Tokyo.

It would be difficult to overestimate the contributions Michel Dupuis has made to this program, both 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. Funding for much of the development work on GAMESS is provided by the Air Force Office of Scientific Research.

In late 1987, NDSU and IBM reached a Joint Study Agreement. One goal of this JSA was the development of a version of GAMESS which is 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 the current version of GAMESS.

During the first six months of 1990, Digital awarded a Innovators 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 Digital Unix or VMS windowing environments, and many other X-windows environments as well. The ability to visualize the molecular structures, orbitals, and electrostatic potentials is a significant improvement.

Parallelization of GAMESS began in 1991, with most of the work and design strategy done by Theresa Windus. This multi-year process benefits greatly from the long term support of

GAMESS by the AFOSR, as well as the ARPA sponsorship of the Touchstone Delta experimental computer.

As of July 1, 1992, the development of GAMESS moved to Iowa State University at the Ames Laboratory.

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 has led to the introduction of the Distributed Data Interface style of programming.

The rest of this section gives more specific credit to the sources of various parts of the program.

GAMESS is a synthesis, with many major modifications, of several programs. A large part of the program is from HONDO 5. For sp basis functions, GAUSSIAN76 integrals have been adapted to the HONDO symmetry procedure, while Rys polynomials are used for any higher angular momentum.

Extension of the 1e- and 2e- integral routines to handle spdfg basis sets was done by Theresa Windus at North Dakota State University.

The current spdfg gradient package consists of HONDO8 code for higher angular momentum, and the Gaussian80 code for sp bases. The code was adapted into GAMESS by Brett Bode at Iowa State University.

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 Dora 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.

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 Direct Inversion in the Iterative Subspace (DIIS) convergence procedure was implemented by Brenda Lam (then at the University of Houston), for RHF and UHF functions.

The UHF code was taught to do high spin ROHF by John Montgomery at United Technologies, who extended DIIS use to ROHF and the one pair GVB case. Additional GVB-DIIS cases were programmed by Galina Chaban at ISU.

The GVB part is a heavily modified version of GVBONE.

The CI module 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 analytic CI gradient was done by Simon Webb at Iowa State University.

The FULLNR and FOCAS MCSCF programs were contributed by Michel Dupuis of IBM from the HONDO program.

The approximate 2nd order SCF was implemented by Galina Chaban at Iowa State University. SOSCF is provided for RHF, ROHF, GVB, and MCSCF cases.

The sequential MP2 code was adapted from HONDO by Nikita Matsunaga at Iowa State, who also added the RMP2 open shell option in 1992. The 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. Haruyuki Nakano from the U. Of Tokyo interfaced his multireference MCQDPT code to GAMESS during a 1996 visit to ISU.

The parallel MP2 code is a descendent of work done 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, after some grief in developing the necessary DDI infrastructure.

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 numerical force constant computation and normal mode analysis was adapted from 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, with open shell CPHF code written at NDSU. The TCSCF CPHF code is the result of a collaboration between NDSU and John Montgomery at United Technologies. IR intensities and analytic polarizabilities during hessian runs were programmed by Simon Webb at ISU.

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.

Most geometry search procedures in GAMESS (NR, RFO, QA, and CONOPT) were developed by Frank Jensen of Odense University. These methods are adapted to use GAMESS symmetry, and Cartesian or internal coordinates.

The non-gradient optimization so aptly described as "trudge" was adapted from HONDO 7 by Mariusz Klobukowski at U.Alberta, who added the option for CI optimizations.

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 a 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 at Odense University.

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

The radiative transition moment and Zeff spin-orbit coupling modules were written by Shiro Koseki at North Dakota State University, and at Mie University.

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 the active space of two electrons in two orbitals, with assistance from a visit 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.

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 has 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.

Most polarizability calculations in GAMESS were implemented by Henry Kurtz of the University of Memphis. This includes a general numerical differentiation based on application of finite electric fields, and a fully analytic calculation of static and frequency dependent NLO properties for closed shell systems. The latter code was based on a MOPAC implementation by Prakashan Korambath at U. Memphis.

Edmiston-Ruedenberg energy localization is done with a version of the ALIS program "LOCL", modified to run inside GAMESS at NDSU. 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 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.

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 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. 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. Simon Webb at ISU programmed the current self-consistency process for the induced dipoles. The EFP method was sufficiently developed, tested, and described to be released in Sept 1996.

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, at San Diego Supercomputer Center. It was included into GAMESS by Laura Gregerson in March 2000 during a visit to Ames.

The PCM code originates in the group of Jacopo Tomasi at the University of Pisa. Benedetta Mennucci was instrumental in interfacing the PCM code to GAMESS, in 1997, and answering many technical questions about the code, the methodology, and the documentation.

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.

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.

Distribution Policy

Copies of GAMESS will be provided at no charge, to anyone who can reach Mike Schmidt by E-mail, and is not working in a country such as People's Republic of China, North Korea, Cuba, and so on. Your country need not be particularly democratic, but it should at least not have a governmental policy of driving tanks over students.

To get a copy, send E-mail to Mike at the following E-mail address:
mike@si.fi.ameslab.gov
and tell what kind of computer you have. If it happens to be an IBM mainframe, be sure to specify whether it runs VM, MVS, or AIX. You will receive GAMESS by E-mail as a series of files. Please be sure that your mailer's spool directory contains 10 MB of free disk space *before* you ask for GAMESS, so that your incoming mail arrives safely.

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. This is done by signing and returning a straightforward copyright letter. If you know anyone who wants a copy of GAMESS, please refer them to us 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) contact Mike over the net. You should probably allow a year or so to pass for enough significant changes to accumulate.

Input Philosophy

Input to GAMESS may be in upper or lower case. 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 only examples of this category are $DATA, $ECP, $POINTS, and $STONE.
3. Formatted data. This data is never typed by the user, but rather is generated in the correct format by some earlier GAMESS run.

All input groups begin with a $ sign in column 2, followed by a name identifying that group. The group name should be the only item appearing on the input line for any group in category 2 or 3.

All input groups terminate with a $END. For any group in category 2 and 3, the $END must appear beginning in column 2, and thus is the only item on that input 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 spdfg type for computation of the energy or gradient. Some restrictions apply, for example, analytic hessians are limited to spd basis sets.

This program is limited to a total of 500 atoms. The total number of shells cannot exceed 1000, containing no more than 5000 symmetry unique Gaussian primitives. Each contraction can contain no more than 30 gaussians. The total number of contracted basis functions, or AOs, cannot exceed 2047, but one further limit applies: The CI/MCSCF package can use at most 768 orbitals. You may use up to 50 effective fragments, of at most 5 types, containing up to 100 expansion points.

In practice, you will probably run out of CPU or disk 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.

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 slighlty 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 dictionary file. 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.