

GAMESS-UK

Session 1

Introduction and Program Basics

Martyn F. Guest, Huub van Dam and Paul Sherwood
CLRC Daresbury Laboratory

m.f.guest@dl.ac.uk, h.j.j.vandam@dl.ac.uk,
p.sherwood@dl.ac.uk

Session 1: Introduction and Program Basics

- Introduction - HPCx and GAMESS-UK
 - Brief overview of QM methods and GAMESS-UK functionality
 - Basics - Input file preparation and job execution
 - Basics 1. Specifying the Geometry and Basis Set
 - Geometry - Internal Coordinates and Cartesians
 - Treatment of Symmetry
 - Basis Sets - Internal, Hybrid and General
 - Basics 2. SCF and DFT Calculations
 - RUNTYPE and SCFTYPE
 - SCF and DFT Wavefunctions - Conventional & Direct
 - Starting Vectors
 - Basics 3. Geometry Optimisation
- Running GAMESS-UK
 - File Usage
 - preparing GAMESS-UK input
 - Invoking GAMESS-UK and the *rungames* script
 - GAMESS-UK output & Visualisation of results
 - Information for the hands-on session

Session 2: More Advanced Options

- Analysing the Wavefunction
 - Properties
 - Graphics
 - Potential Derived Charges
- Density Functional Theory (DFT)
 - Coulomb fitting basis sets
 - Gradient corrections
 - Quadrature selection
- Transition States
- ECPs and Valence Only Calculations
- Relativistic Calculations (ZORA)
- Post Hartree Fock Calculations
 - MP2, MP3
 - MCSCF / CASSCF
 - Direct-CI
 - Conventional Selection-CI
 - CCSD and CCSD(T)
- Solvation (DRF)
- Other RUNTYPEs
- Multiple RUNTYPE specification
- Other Analysis Codes (NBO and AIMPAC)
- Functionality of the Parallel Code
- GAMESS-UK on HPCx
- Trouble Shooting
 - Linear Dependence
 - SCF Convergence
 - Geometry optimisation and Transition States

Introduction: Topics

- WWW pages for GAMESS-UK:
<http://www.dl.ac.uk/CFS>
- Why was GAMESS-UK developed
- On what hardware platforms does the code currently run
- What is the current functionality in GAMESS-UK and what are the limitations and expectations
- Benchmarks and associated cost-effectiveness
- Who is doing the Developments and Support

WWW Pages for GAMESS-UK

- Capabilities <http://www.dl.ac.uk/CFS>
- User's Manual
<http://www.dl.ac.uk/CFS/docs>
- Support
- Tutorial (this material)
<http://www.dl.ac.uk/CFS/tutorials>
- Benchmarks
<http://www.dl.ac.uk/CFS/benchmarks>
- Applications
- FAQ's
<http://www.dl.ac.uk/CFS/FAQ>
- Known Bugs
- Hardware Platforms
- Download
- Programmer's Manual

Why GAMESS-UK was Developed

- Developed as part of CCP1, the collaborative computational project in molecular electronic structure
- Derived from the original GAMESS code obtained from NRCC in 1981 (no gradients in ATMOL). Subsequent developments and functionality quite separate from GAMESS-US.
- Using the parallel software tools from the Pacific Northwest National Laboratory, the parallel code is efficient and portable on a variety of MPP platforms
- Provides broad range of functionality, with competitive capabilities, particularly in the area of DFT and parallel processing.
- Support provided by CCP1 and CFS (Computing for Science)

Hardware Platforms

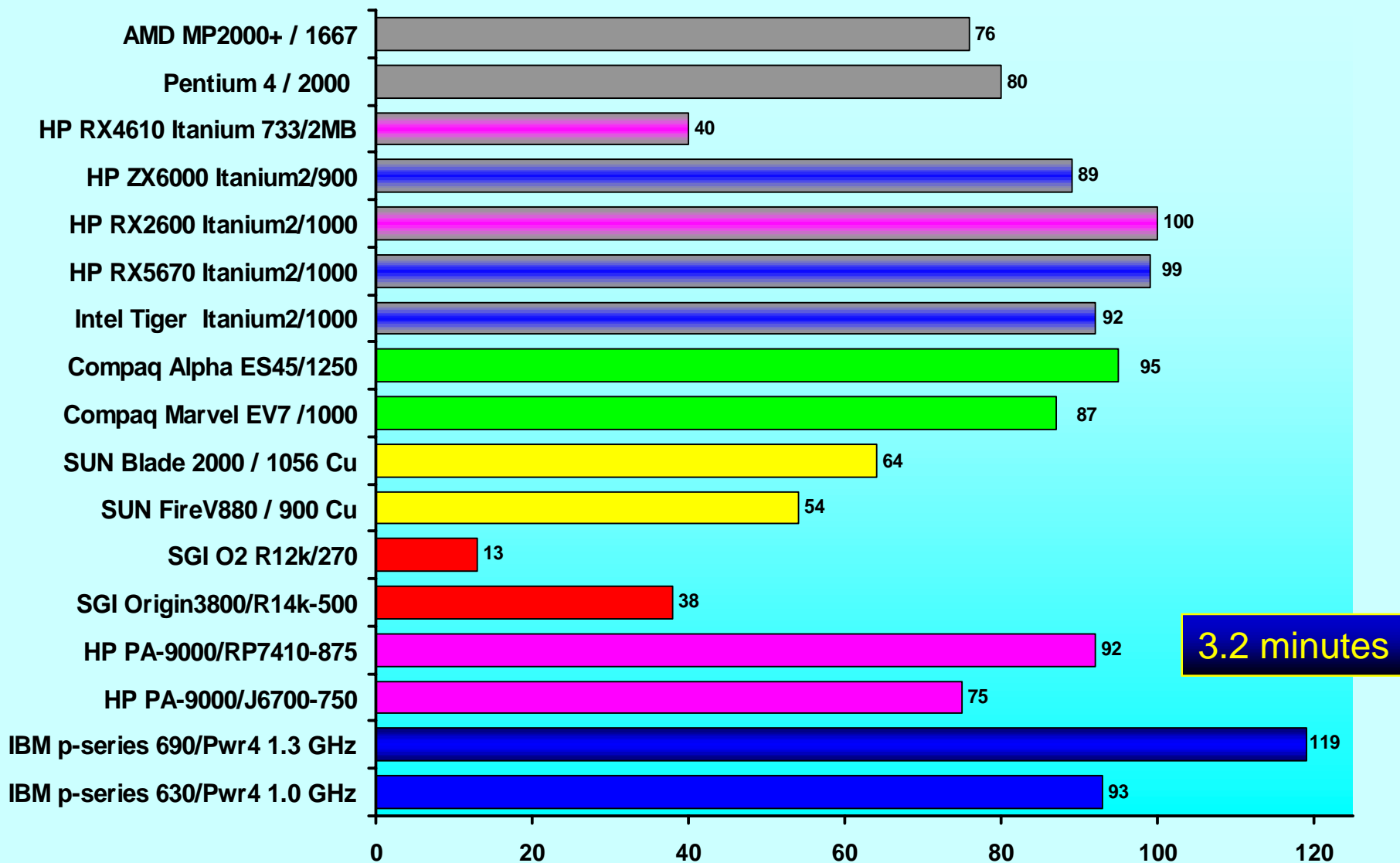
- MPPs
 - ✘ IBM-SP (AIX 5.1D / PSSP 3.4)
 - ✘ Cray T3E (UNICOS/mk 2.0.4.X)
 - ✘ SGI Origin 2000 and 3000 (IRIX 6.5)
 - ✘ Compaq AlphaServer SC (Tru64)
- Desktop Workstations and Clusters of Workstations
 - ✘ SUN (Solaris 2.6 and 2.8),
 - ✘ IBM (AIX 4.X and 5.1),
 - ✘ SGI/R4400 (IRIX5.3), SGI_N32 (IRIX 6.5),
 - R8K/R10K/R12K/R14K
 - ✘ DEC Alpha (Compaq Tru64 V5.1, RedHat Linux*)
 - ✘ PowerPC (RedHat Linux*)
 - ✘ HP (HP UX B10.20 and B11.0)
- IA32 and IA64 systems and associated Commodity Clusters
 - ✘ Pentium III, Pentium 4, AMD Athlon, Itanium-1 and Itanium-2 (PGI, ifc/efc)
 - ✘ PC (RedHat, Suse etc Linux 2.0.X, 2.2.X, egcs g77/gcc)

Minimum requirements:

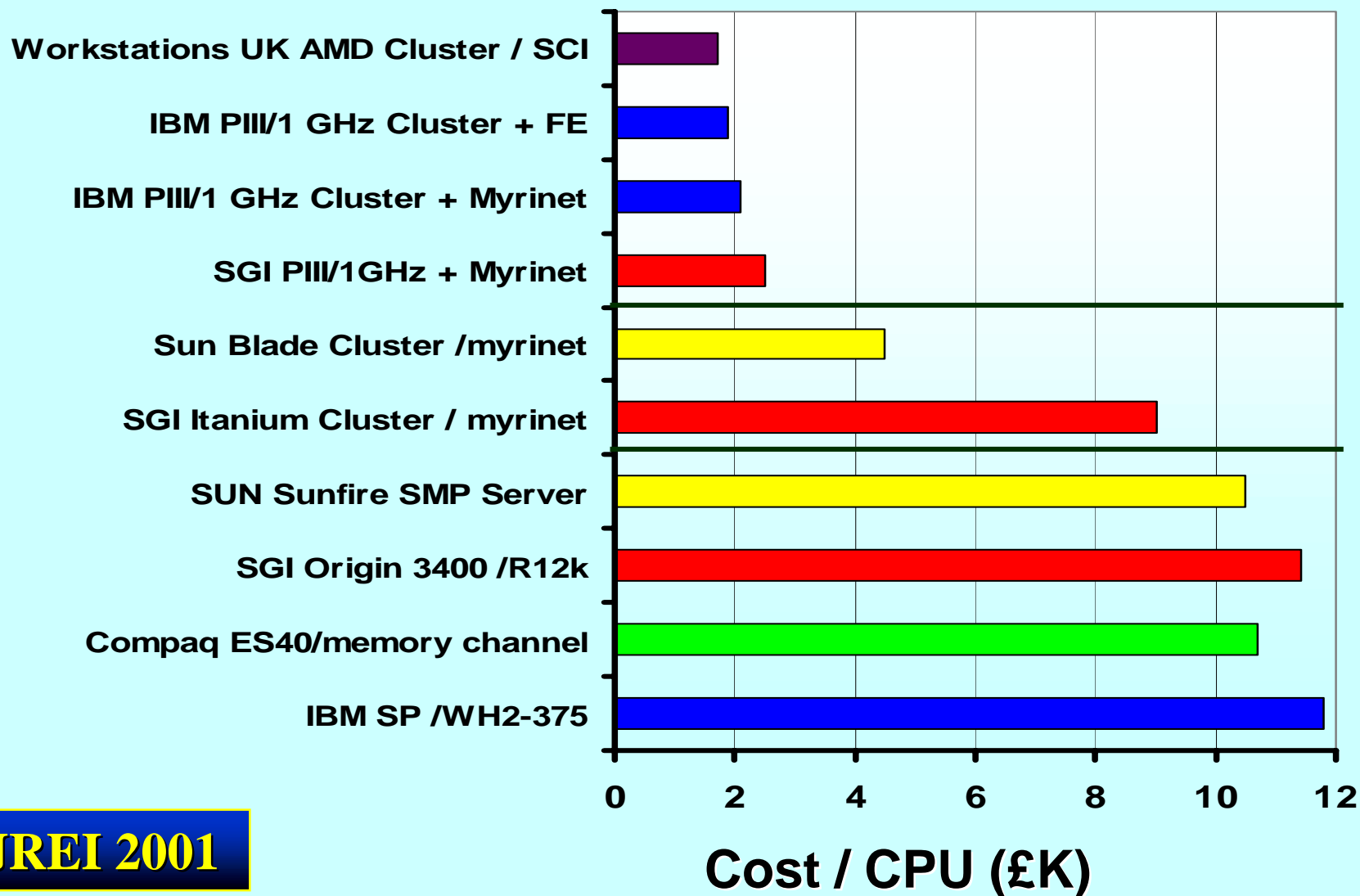
**64-128 MByte RAM
2 GByte disk (SCSI)**

The GAMESS-UK Serial Benchmark

Performance relative to the HP RX2600 Itanium2/1000



Machine Costs: Proprietary and Commodity Solutions



JREI 2001

Introduction to HPCx

- A joint venture between the Edinburgh Parallel Computing Centre (EPCC) at the University of Edinburgh and the Daresbury Laboratory of the Central Laboratory for the Research Councils (CLRC)
- Project funded to £53M (EPSRC, NERC, BBSRC, PPARC)
- Established to operate and support the principal academic and research computing service for the UK
- Principal objective being to provide a *Capability Computing* service to run scientific applications that could not be run on any other available computing platform
- Six-year project with defined performance requirements at year 0, year 2 and year 4 so as to match Moore's Law
- IBM chosen as the technology partner with Power4 based p690 platform and associated infrastructure

HPCx Technology Phase 1

- Phase 1 (Dec. 2002): 3 TFlop/s Rmax Linpack
 - 40 Regatta-H SMP compute systems (1.28 TB memory)
 - 32 x 1.3GHz processors, 32 GB memory; 4 x 8-way LPARs
 - 2 Regatta-H I/O systems
 - 16 x 1.3GHz processors (Regatta-HPC), 4 GPFS LPARS
 - 2 HSM/backup LPARS, 18TB EXP500 fibre-channel global filesystem
 - Switch Interconnect
 - Existing SP Switch2 with "Colony" PCI adapters in all LPARs (20 us latency, 350 MB/s bandwidth)
 - Each compute node has two connections into switch fabric (dual plane)
 - 160 x 8-way compute nodes in total
 - Ranked #9 in the TOP500 list (November 2002)

HPCx Technology Phases 2 & 3

- Phase 2 (2004): 6 TFlop/s Rmax Linpack
 - >40 Regatta-H+ compute systems
 - 32 x 1.8GHz processors, 32 GB memory, full SMP mode (no LPAR)
 - 3 Regatta-H I/O systems (Double the capabilities of Phase 1)
 - "Federation" switch fabric
 - bandwidth quadrupled, ~5-10 microsecond latency, Connect to GX bus directly
- Phase 3 (2006): 12 TFlop/s Rmax Linpack
 - >40 Regatta-H+ compute systems
 - unchanged from Phase 2
 - >40 additional Regatta-H+ compute systems
 - double the existing configuration
 - 4 Regatta I/O systems (Double the capabilities of Phase 2)
- Open to Alternative Technology Solutions (IPF, BlueGene/L ..)

HPCx - Phase 1 Technology at Daresbury



November 2002



July 2002

Overview of QM methods

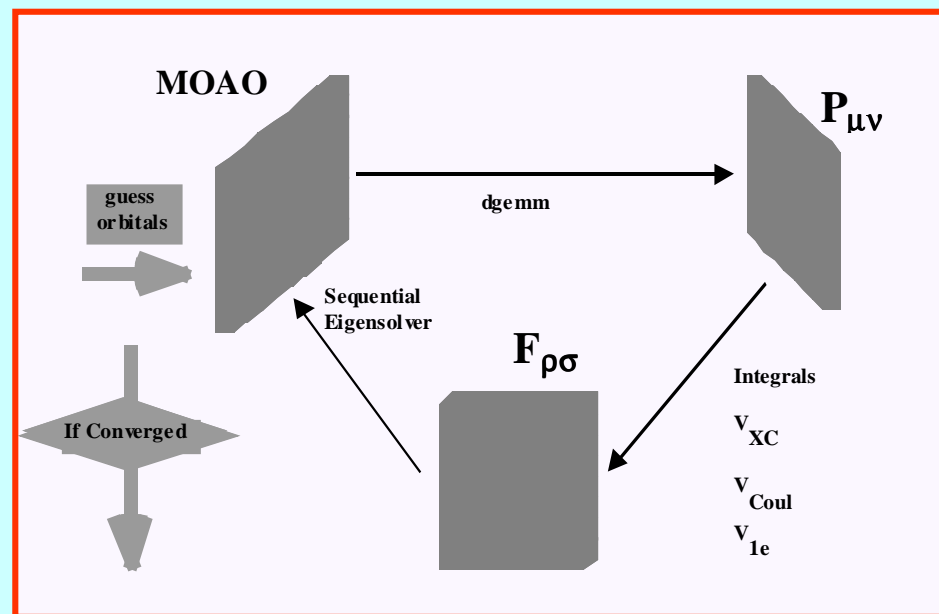
GAMESS-UK functionality

Functionality - Quantum Mechanical Capabilities

- Hartree-Fock energies, gradients, and second derivatives
- Multiconfiguration self-consistent field (MCSCF, CASSCF) energies and gradients
- Density functional theory, energies, gradients and **frequencies**
- Many-body perturbation theory (MP2-MP3) energies plus MP2 gradients
- Coupled-cluster [CCSD and CCSD(T)] energies
- Single and multi-reference configuration interaction energies
- Segmented and generally contracted basis sets, including the correlation-consistent basis sets (EML)
- Effective core potential energies, gradients, and **second derivatives**
- Relativistic Effects (ZORA)
- Solvation Effects (Direct Reaction Field - DRF)
- Variety of Greens Functions-based methods for excitation and ionization energies

SCF Calculations 1.

- SCF process
 - guess wavefunction
 - iterative solution
- Open Shell calculations
 - UHF
 - Different orbitals for alpha and beta spin electrons
 - Solve alpha and beta secular equations
 - ROHF
 - Same orbitals for different spin,
 - but different occupations (more alpha electrons than beta)
 - GVB (Generalised Valence Bond)
 - Doubly occupied orbitals, singly occupied orbitals, bonding / anti-bonding pairs



SCF Calculations 2.

- Conventional SCF
 - ✘ Store 2-electron integrals on file
 - ✘ Efficiency improved by neglecting small integrals
- Direct SCF
 - ✘ Compute integrals whenever needed
 - ✘ Efficiency improved by pre-screening (taking $C_{\sigma j} C_{\tau j}$ into account)
 - ✘ Δ -density matrix
- Direct vs. Conventional
 - ✘ Conventional:
 - requires less computation
 - ✘ Direct:
 - requires much less disk space
 - allows for dynamic load balancing and avoids I/O bottleneck on parallel machines

Derivatives

- First derivative provides atomic forces
 - geometry optimisation and transition state searches to locate stationary points on potential energy surfaces (PEs)
- Second derivatives
 - vibrational frequencies and infrared intensities
 - analytic form implemented for HF and MP2 (DFT development just completed)
 - characterisation of stationary points (minima, transition states etc.)
- Other derivatives
 - polarisabilities
 - magnetisability
 - Raman intensities

Analysing the Wavefunction

From the converged wavefunction we can compute:

- A variety of one-electron properties
 - dipole moment
 - electrostatic potentials
 - electric field
 - electric field gradient
 - quadrupole moment
 - octupole moment
 - hexadecapole moments
 - spin densities

- **Population analysis** (atomic charges, bond and orbital analysis)
- Generate **localised molecular orbitals** (LMOs)
- **graphical analysis** (electron density, orbital amplitude, electrostatic potential maps)
- **distributed multipole analysis** (DMA)
- Morokuma **energy decomposition analysis**

Density Functional Theory

- If you know the electron density you know everything

- The nuclear positions
- The nuclear charges
- The number of electrons

- Functional incorporates

- Exchange
- Correlation

- Functionals

- Local Density
- Gradient Corrected (GGA)
- Hybrid

- Numerical integration required

- Hartree-Fock

$$K\psi_i(\vec{r}_1) = \sum_j \int \frac{\psi_i(\vec{r}_2)\psi_j^*(\vec{r}_2)}{r_{12}} d\vec{r}_2 \psi_j(\vec{r}_1)$$

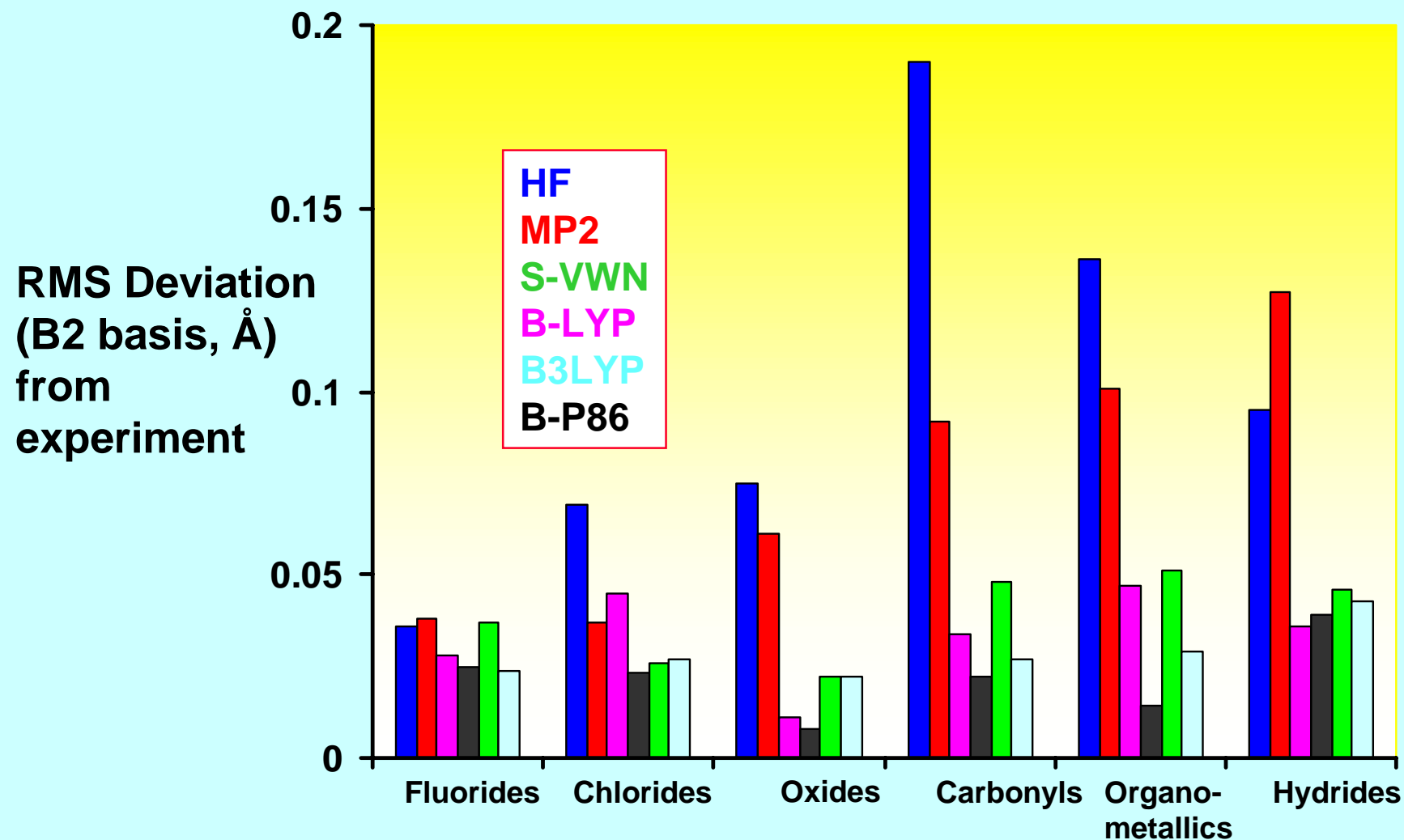
- Density Functional Theory

$$K\psi_i(\vec{r}_1) = f[\rho(\vec{r}_1)]\psi_i(\vec{r}_1)$$

- Hohenberg-Kohn-Sham formalism
- Assumes a charge density, and successively better approximates the Hamiltonian
(cf. Traditional *ab initio* MO methods assume an exact Hamiltonian and successively better approximates the wave-function)

First-Row Transition Metal-Ligand Bond Lengths (M-L)

RMS Deviations from Experiment



Post-HF Calculations

1. Configuration Interaction (CI)

- Configuration expansion
 - ✘ Hartree-Fock: single determinant with electrons in the lowest energy orbitals
 - ✘ Many other determinants possible
 - ✘ Multiple determinant zero-order wavefunctions
 - a small number of determinants
 - near degeneracies
 - non-dynamic correlation
 - GVB, MCSCF, CASSCF
 - ✘ Many determinant wavefunctions
 - a large numbers of determinants (10M+)
 - dynamic correlation
 - MP, CI, CC, Full-CI

Post-HF Calculations 2. Perturbation Theory

- Møller-Plesset 2nd order perturbation theory (MP2) is the most efficient post-HF method
- Size extensive
- Non-variational
- Problems if orbital energies (ϵ_j, ϵ_a) close together
- Functionality available
 - Energy (MP2, MP3)
 - Gradients (MP2, MP3)
 - Analytic frequencies (MP2)
 - Numerical frequencies (MP2, MP3)
- Both direct and conventional MP2 scheme available

- Expand

$$\hat{H} = \hat{H}^{(0)} + \lambda \hat{V}$$

$$E = E^{(0)} + \lambda E^{(1)} + \lambda^2 E^{(2)} + \dots$$

$$\Psi = \Psi^{(0)} + \lambda \Psi^{(1)} + \lambda^2 \Psi^{(2)} + \dots$$

- Second order energy

$$E^{(2)} = \frac{1}{4} \sum_{ijab} \frac{|\langle ij || ab \rangle|^2}{\epsilon_i + \epsilon_j - \epsilon_a - \epsilon_b}$$

Post-HF Calculations 3. MCSCF, CASSCF and CI

- MCSCF, CASSCF (active space)
 - ✘ Multiple determinants
 - ✘ Simultaneous optimisation of both orbitals and configuration coefficients
 - ✘ non-dynamic correlation (zero order wavefunction)
 - ✘ proper dissociation of bonds, near degeneracies
- Choice of active space
 - ✘ Bond dissociation
 - ✘ Near degeneracies
 - ✘ Occupation numbers from a small CI calculation
- Configuration Interaction, CI (dynamic correlation)
 - ✘ Many determinants
 - ✘ Orbitals kept fixed
 - ✘ Configuration coefficients optimised

Post-HF Calculations 4. MRDCI and Direct-CI

- MRDCI
 - ✘ Computes and stores all (Table-CI) or part (semi-direct) of the Hamiltonian matrix
 - ✘ Perturbatively selects the most important determinants
 - ✘ Useful for calculating excited states and UV/Vis spectra
 - ✘ semi-direct implementation extended size of systems amenable to study - 5×10^5 configurations, 20 roots
 - ✘ *“automatic” calculation of UV/Vis spectra*
- Direct-CI
 - ✘ Recomputes most of the Hamilton matrix whenever needed
 - ✘ Uses all single and double excitations from a multireference set
 - ✘ Useful for calculating accurate ground states
 - ✘ Limited range of Excited states also possible
 - ✘ 10^7 - 10^8 configuration state functions
- Full-CI
 - ✘ Useful for benchmark energies

Post-HF Calculations 5. Coupled Cluster

- Exponential expansion of the wavefunction
- Size extensive
- Non-variational
- CCSD (n^6) and CCSD(T) (n^7 scaling)
- Only closed shell energies available
- Most useful for accurate ground state energies
- Now widely used to obtain accurate energetics from DFT geometries

Response Theory

- Time independent reference wavefunction
- Response to a time dependent electric field treated with a perturbation expansion
- Eigenvalues of resulting equations correspond to excitation energies
- Most useful as an efficient formalism to calculate **UV/Vis spectra, ionisation and attachment potentials**
- Accuracy good for single electron excitations
- Tamm-Dancoff Approximation (TDA)
 - **Equivalent to a Singles CI**
- Random Phase Approximation (RPA)
 - **Includes some correlation effects with the reference state**
 - **Excited state gradient and geometry optimisation**
- Multi Configurational Linear Response (MCLR)
 - **RPA using a MCSCF reference wavefunction**

Program Basics

Input file preparation and job execution

Input Preparation

GAMESS-UK reads a short, directive-structured, input from a data file which can be

- ✘ (i) prepared by hand using a text editor
- ✘ (ii) generated by a graphical interface, we will consider two
 - CCP1 Python GUI
 - MOLDEN
- ✘ **GUI Functionality**
 - z-matrix (internal coordinate) editing
 - graphical data display (e.g. orbitals)
 - job submission

- ✘ None of the graphical interfaces currently support the full functionality. This workshop will concentrate on the input files, with GUIs available and demonstrated in the practical sessions.

Invoking GAMESS-UK

- Program reads from stdin - e.g. using /bin/csh

```
% /usr/local/packages/gamessuk/bin/gamess/gamess-uk.serial < test.in  
% setenv ed3 test.ed3  
% /usr/local/packages/gamessuk/bin/gamess/gamess-uk.serial < test.in
```

- Alternatively simply in-line the input, as in the example directories
GAMESS-UK/examples/chap2 etc
- Complications
 - ✘ should run on a fast scratch disk (optimise access to ed7, ed2 etc)
 - ✘ parallel code is invoked differently
 - mpirun, parallel, prun, poe etc
- On workshop systems GAMESS_EXE should point to the correct executable

```
#!/bin/csh  
setenv ed2 mfged2  
setenv ed3 mfged3  
$GAMESS_EXE << EOF  
title  
h2co - TZVP - closed shell SCF  
zmatrix angstrom  
c  
o 1 1.203  
h 1 1.099 2 121.8  
h 1 1.099 2 121.8 3 180.0  
end  
basis tzvp  
enter  
EOF
```

The rungamess Script

- rungamess
 - ✘ creates a scratch directory
 - ✘ sets environment variables from command line arguments
 - ✘ files named <job>.in, <job>.out, <job>.pun etc
- Arguments
 - k ed3 keep file on local disk
 - k ed3=junk.ed3 keep file as specified name
 - t ed7=junk.ed7 keep temporary file (on \$GAMESS_TMP)
 - p 8 number of parallel processors
 - q Submit it a job queue
 - r mrdci Keep files needed to restart an MRDCI run (etc)

ed2, ed3, ed7 ... Mainfile, Dumpfile, Scratchfile etc

rungames - Examples and Environment Variables

```
% rungames test
```

```
% rungames -p 8 -q test
```

```
% rungames -k ed3 -k ed2=/tmp/ed2 test1
```

```
% rungames -k ed4=test1.ed3 test2
```

```
% rungames -r mrdci test1
```

```
% rungames -r mrdci -n test1 test_restart
```

Environment variables

GAMESS_EXE	GAMESS-UK executable
GAMESS_SCR	routing for scratch directory files
GAMESS_TMP	routing for files indicated with -t
GAMESS_PAREXE	GAMESS-UK parallel executable
GAMESS_SUBMODE	How to submit jobs ll pbs nqs
GAMESS_PARMODE	How to run parallel jobs mpi sgimpi poe tcgmsg

Input Structure - A Sample Input

- Predirectives
 - file routing, parallel options etc, memory allocation
- Directive-structure, keyword driven
 - Class 1
 - title, geometry, basis
 - Class 2
 - runtime, scftype, vectors, enter etc
- Many options have defaults, shown in blue.
- Numerous examples of data input are provided in the user manual

input0.in, input1.in

```
core 4000000
title
h2co - default 3-21G basis - SCF
charge 0
multiplicity singlet
zmatrix angstrom
c
o 1 1.203
h 1 1.099 2 121.8
h 1 1.099 2 121.8 3 180.0
end
basis 3-21g
runtime scf
scftype rhf
thresh 5
vectors atoms
enter 1
```

Program Basics

1. Specifying the Geometry and Basis Set

Specification of Geometry

- Cartesian
 - ✘ Easily obtained from modelling software
 - ✘ Can automatically generate internal coordinates for optimisation
- Z-matrix (internal) coordinates
 - ✘ A way to build a geometry from known bond lengths, angles etc
 - ✘ Can optimise chosen set of internal coordinates
 - ✘ Hessian matrices generally better conditioned

Geometry input

- Units either atomic units (au) or Angstrom
- Coordinates (x,y,z)
- Charge
 - positive
 - negative
 - fractional
- Tag <symbol><label>
 - Tag is used to assign basis sets
 - Tag is used in symmetry determination and analysis
- geometry **all**
 - Generate internals

geom0.in

```
title
taut 3 3-21g energy = -297.971122 au
geometry au
  0.00000  0.00000  0.00000  1.0  h
 -1.87385  0.00000  0.00000  7.0  n
 -3.15944 -2.29528  0.00000  7.0  n
 -3.50408  1.96648  0.00000  6.0  c
 -5.53585 -1.64980  0.00000  6.0  c
 -5.89182  1.00597  0.00000  6.0  c
 -2.87960  3.87928  0.00000  1.0  h
 -7.64127  1.98814  0.00000  1.0  h
 -7.42454 -3.37843  0.00000  8.0  o
 -6.76589 -5.08190  0.00000  1.0  h
end
enter
```

Z-matrix input

- Define parameters
- Use of symbolic variables and constants
- Z-matrix conventions: **First atom** will be at (0,0,0), **Second** at (0,0,z), **Third** at (x,0,z)

zmat1.in

Each nucleus (including dummies) is numbered sequentially and specified on a single data line. Nth nucleus (N>3):

TAGN, N1, R1, N2, ANG12, N3, ANG123, ITYPE

TAGN - name and chemical nature of the nucleus

N1 - an integer specifying a previously defined nucleus

R1 - R(N-N1) in the appropriate units.

N2 - an integer specifying a second nucleus, N2, different from N1, for which the angle (N,N1,N2) will be given.

ANG12 - value of (N,N1,N2), the internuclear angle at N1 between N and N2, in degrees.

N3 - an integer specifying a nucleus for which the dihedral angle (N,N1,N2,N3) will be defined as ANG123.

ANG123 - the internuclear dihedral angle (N,N1,N2,N3) specified (°). It is the angle between the planes (N,N1,N2) and (N1,N2,N3) (sign)

```

TITLE
MoF6 Oh symmetry
ZMAT ANGSTROM
MO
F 1 MOF
F 1 MOF 2 90.0
F 1 MOF 2 90.0 3 90.0
F 1 MOF 2 90.0 3 180.0
F 1 MOF 2 90.0 3 -90.0
F 1 MOF 3 90.0 2 180.0
VARIABLES
MOF 1.814
END
  
```

Z-matrix Examples

A simple D_{3h} system, the CH_3 radical (C-H = 1.120 Å)

zmat4a.in, zmat4b.in

ZMAT ANGS

C

H 1 1.120

H 1 1.120 2 120.0

H 1 1.120 2 120.0 3 180.0

END

zmat2.in

A simple C_{2v} system, H_2O , with O-H = 0.952 Å and HOH = 104.5.

ZMAT ANGS

O

H 1 0.952

H 1 0.952 2 104.5

END

OR

ZMAT ANGS

O

H,1,0.952

H,1,0.952,2,104.5

END

A T_d system, CH_4 (C-H=1.083 Å)

ZMAT ANGS

C

H 1 1.083

H 1 1.083 2 109.471

H 1 1.083 2 109.471 3 120.0

H 1 1.083 2 109.471 4 120.0

END

zmat3.in

ZMAT ANGS

C

H 1 1.082

C 1 1.387 2 120.0

H 3 1.082 1 120.0 2 0.0

C 3 1.387 1 120.0 4 180.0

H 5 1.082 3 120.0 4 0.0

C 5 1.387 3 120.0 6 180.0

H 7 1.082 5 120.0 6 0.0

C 7 1.387 5 120.0 8 180.0

H 9 1.082 7 120.0 8 0.0

C 9 1.387 7 120.0 10 180.0

H 11 1.082 9 120.0 10 0.0

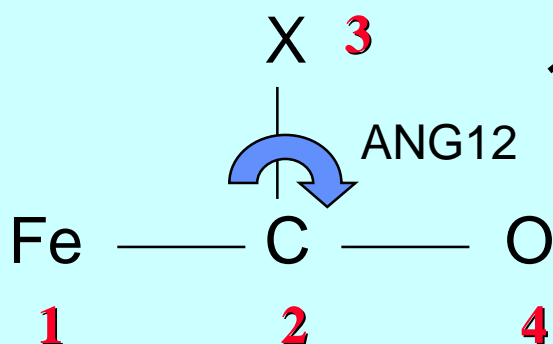
END

zmat5.in

A D_{6h} system, the benzene molecule
with C-C = 1.387 Å and C-H = 1.082 Å

Z-matrix restrictions: Dummy atoms

- Atoms must be specified in terms of previously defined atoms
- Directly-bonded angle ANG12 must be in the range $0 < \text{ANG12} < 180$
- Sometimes definition is easier or more reliable using dummy atoms



```

title
fe(co)5 SCF energy
zmat angstrom
fe
c 1 rceq
x 2 1.00 1 90
o 2 rco 3 90 1 180
....

constants
rceq 1.8273000
rcax 1.8068000
rco 1.1520
end
runtype scf
enter
  
```

zmat6.in

Treatment of Symmetry I.

Before considering aspects of data specification, it is important to have an idea of the methods used in the treatment of molecular symmetry. The aim is to try and optimise performance while maintaining simplicity of related data specification. There are **two levels** at which symmetry is employed;

- **the molecular level**: the program will deduce the point group symmetry based on the geometry provided
- in default, use that information in minimising the number of integrals that need be computed e.g. in SCF calculations.
- The program is capable of handling both Abelian (e.g. C_{2v}) and non-Abelian point groups (e.g. C_{3v}) on an equal footing
- **the orbital level**: both at the AO and MO level, when the symmetry characteristics of MOs will be used in optimising both HF and post HF calculations.
- This requirement is met through the use of symmetry-adapted basis functions. While this technique is limited to Abelian point groups, the program will treat non-Abelian groups by resorting to the optimum Abelian group when handling orbital symmetry (e.g. C_{3v} to C_s)

Treatment of Symmetry II.

- The TAGs used to characterise the component nuclei of the system in either the GEOMETRY or ZMATRIX directive play a vital role in symmetry determination. They are used to establish the effective point group symmetry of the system. Failure to appreciate the rules for TAG specification can lead to a considerable loss in efficiency.
- In **RHF, UHF and Moller Plesset calculations** GAMESS-UK will, based on the molecular point group, generate and retain only the **unique integrals** required, for example, in the process of constructing a 'skeletonised' Fock matrix.
- Such a **symmetry-truncated integral list** is, however, **NOT usable** at present in **pair-GVB, CASSCF, MCSCF, RPA or CI calculations**, and again considerable caution should be taken when using an integral file generated in an earlier SCF run in a subsequent post-HF calculation using the BYPASS directive.
- In **geometry optimisations** the **point group** is derived from the starting geometry, and is **not allowed to change** during the subsequent optimisation. This can lead to problems if the Z-matrix is constructed in such a way as to allow such changes to occur.
- Both **MCSCF and CI modules** assume that **symmetry adaptation** is in operation. If for any reason the SCF MOs of differing irreducible representations become mixed, the post HF calculations may prove unreliable.

Controlling the Point Group Symmetry

In some instances the user need consider lowering the point group determined in default by the program, particularly in the case of degenerate point groups, which for some SCFTYPEs and RUNTYPEs must be a subset of the D_{2h} group. Specifically the appearance of the message.

```
*****
* The molecular point group prohibits use of either
* the requested SCFTYPE or RUNTYPE. Reduce the
* molecular symmetry by modifying the nuclear TAGs
*****
```

The symmetry handling routines within GAMESS-UK assume that any centres with differing TAGs are not related by symmetry. The point group actually adopted in the calculation may be controlled though appropriate TAG specification.

NbCl₅ D_{3h} ; changing the first equatorial chlorine TAG (to CL1) will yield a C_{2v} point group, thus;

```
ZMAT ANGSTROM
NB
CL1 1 REQ
X 2 1.0 1 90
CL 1 REQ 2 120 3 180
CL 1 REQ 2 120 3 0
CL 1 RAX 2 90 3 90
CL 1 RAX 2 90 3 -90
CONSTANTS
REQ 2.338
RAX 2.362
END
```

symmetry1a.in,
 symmetry1b.in

Disabling use of Symmetry

In some applications it is beneficial to present the directives NOSYM and “ADAPT OFF”:

nosym

to disable use of symmetry at the molecular level, as this ensures that the calculation will be performed with the input molecular orientation (ensuring properties such as orbitals, dipole moments etc will be in the input frame).

adapt off

can be presented to disable symmetry adaption if symmetry breaking distortions are expected (e.g. QM/MM).

This may lead to a substantial cost penalty for post-HF calculations on symmetric systems!

Basis Set Specification

- Default Cartesian angular functions (1s, 3p, 6d, 10f, 15g) are used throughout GAMESS-UK.
- Option of using spherical-harmonic (5d, 7f, 9g) angular functions is available through specification of the HARMONIC directive.
 - ✦ This is implemented internally through appropriate transformations, and not by computing integrals or derivative integrals over the spherical functions.
- Default basis set is 3-21G if no input provided
- Variety of mechanisms for specifying basis sets through the BASIS directive
- Explicit and hybrid basis sets are available
- Can be selected from the “internal” Library file
 - ✦ single keyword specification
 - BASIS TZVP
- Can be input in general form

Internal Basis Sets

- Wide variety of internal basis sets can be requested through single keyword specification
 - **BASIS** codename
- Minimal Basis
 - **STOnG**, **MINI**
- Split valence (SV)
 - **n-m1G**
- Double-zeta (DZ)
- Triple-zeta (TZV) and Extended
- Polarisation basis sets
- Correlation-consistent basis sets (**CC-PVDZ**, **CC-PVTZ**, **CC-PVQZ**, **CC-PV5Z**).
- ECP basis sets
- DFT Basis sets (**DZVP**, **DZVP2** and **TZVP**)

Using the same family of basis set for all atoms in the molecule.

Examples:

BASIS STO3G

BASIS 6-31G or **BASIS SV 6-31G**

BASIS DZ

or **BASIS DZ AHLRICHS**

BASIS DZP

or **BASIS DZP AHLRICHS**

BASIS TZVP

BASIS 6-311G*

BASIS CC-PVDZ

BASIS ECP STRLC

BASIS DFT DZVP2

basis0.in

Internal Basis Sets - Hybrid Specification

- Request basis sets from more than one of the “built-in” basis sets
- User is responsible for allocating such a basis to each centre using the centre TAGs as specified in the GEOMETRY or ZMATRIX directive

BASIS

basis1 <TAG1>

basis2 <TAG2>

END

- Only the unique TAGs should be specified in this process

- Examples:

```
zmatrix angstrom
```

```
c
```

```
o 1 1.203
```

```
h 1 1.099 2 121.8
```

```
h 1 1.099 2 121.8 3 180.0
```

```
end
```

```
basis
```

```
tzv h
```

```
tzvp o
```

```
tzvp c
```

```
end
```

```
basis
```

```
sv h 3-21g
```

```
sv o 6-31g*
```

```
sv c 6-31g*
```

```
end
```

basis1a.in

basis1b.in

General Basis Set Input

```

title
CH2 3B1 GRHF open shell
mult 3
zmatrix angstrom
c
h 1 1.071
h 1 1.071 2 128.65
end
    
```

Coefficient of gaussian primitive

Exponent of gaussian primitive

```

BASIS
S H
0.032828 13.3615
0.231208 2.0133
0.817238 0.4538
S H
1.0 0.1233
S C
0.002090 4232.61
0.015535 634.882
0.075411 146.097
0.257121 42.4974
0.596555 14.1892
0.242517 1.9666
S C
1.0 5.1477
S C
1.0 0.4962
S C
1.0 0.1533
P C
0.018534 18.1557
0.1154420 3.9864
0.3862060 1.1429
0.6400890 0.3594
P C
1.0 0.1146
END
    
```

basis2a.in

Compatibility is also provided with other QC packages (NWChem, Gaussian) by accepting a reversed ordering of coefficients / exponents in the basis definition lines

basis2b.in

Program Basics

2. SCF and DFT Calculations

- RUNTYPE and SCFTYPE
- SCF Input
- Wavefunctions
- Initial MO vectors
- Direct and conventional SCF algorithms
- DFT
- Analysing the Wavefunction
- Convergence, files, and restarting

RUNTYPE specifications

RUNTYPE (and SCFTYPE) define the computation to be carried out.

RUNTYPE defines the particular task to be undertaken;

Default RUNTYPE is SCF i.e., perform a single point SCF calculation

RUNTYPE SCF

RUNTYPE	Task
INTEGRAL	Single point integral calculation
SCF	Single point SCF calculation
OPTIMIZE	Geometry optimisation (internals)
OPTXYZ	Geometry optimisation (cartesians)
SADDLE	Saddle point location
ANALYSE	Wavefunction analysis
FORCE	Force constant evaluation
HESSIAN	Analytic Force constant evaluation
POLARISABILITY	Polarisability calculation
HYPER	Hyperpolarisability calculation
MAGNET	Magnetisability calculation
RAMAN	Calculation of Raman Intensities
INFRARED	Calculation of IR intensities
TRANSFORM	Integral transformation
CI	CI calculation
GF	Green's Function OVGf calculation
TDA	Green's Function 2ph-TDA calculation
RESPONSE	Response calc. of Excitation Energies

SCFTYPE specifications

- SCFTYPE specifies the form of wavefunction calculation to be employed throughout the nominated task.

SCFTYPE <wavefunction>

- Energies and gradients

- ✘ Closed-shell (RHF)
- ✘ Spin-restricted, high-spin open-shell (GRHF)
- ✘ Spin-unrestricted open-shell (UHF)
- ✘ Generalised Valence Bond (GVB)

- Finite point groups

- 700 functions should be possible on PCs / workstations

- 1200 functions should be routine on parallel machines e.g. Origin3800

- ✘ About 3500 functions, 400 atoms have been run
- ✘ Both GVB and GRHF calculations are performed under the same GVB module

SCFTYPE

RHF

UHF

GVB

MP2

MP3

CASSCF

MCSCF

Wavefunction

Restricted Hartree-Fock

Unrestricted Hartree-Fock

Generalised Valence Bond
& high-spin open-shell (GRHF)

2nd order Moller Plesset

3rd order Moller Plesse

Complete Active Space SCF

2nd order MCSCF

Direct SCF Wavefunctions

DIRECT RHF

Direct-SCF

or simply DIRECT

DIRECT UHF

Direct-UHF

DIRECT GVB

Direct-GVB

SCF Wavefunctions

- Default for closed-shell systems is RHF
- Default for open shell systems is high-spin restricted Hartree Fock
- Must force use of UHF if desired
 - SCFTYPE UHF
 - UHF assumes $(N_a - N_b) = 2 * \text{multiplicity}$
- Orbitals assumed to be in the order
 - (closed-shell) (open-shell) (virtual)
- For more complex open shell restricted Hartree Fock configurations, use OPEN to specify the open shell orbitals and occupancies, e.g.
 - open 1 1 2 2
- Always outputs canonical orbitals

SCF Input

- Intended that default settings should be sufficient
- Defaults
 - Restricted-spin wavefunction
 - Accuracy suitable for non-floppy molecule geometry optimization
 - Symmetry as deduced from the geometry
- Minimal input (all defaults)
- Performs a closed-shell SCF on the formaldehyde molecule in a 6-31G* basis

```
title
h2co - 6-31g* basis
zmatrix angstrom
c
o 1 1.203
h 1 1.099 2 121.8
h 1 1.099 2 121.8 3 180.0
end
basis 6-31g*
enter
```

scf1.in

Simple Open Shell Examples

3B_1 CH2 GRHF

```
title
CH2 3B1 RHF high spin
mult 3
zmatrix angstrom
c
h 1 1.071
h 1 1.071 2 128.65
end
basis 6-31g*
enter
```

scf2a.in

3B_1 UHF SCF

```
title
CH2 3B1 UHF
mult 3
zmatrix angstrom
c
h 1 1.071
h 1 1.071 2 128.65
end
basis 6-31g*
scftype uhf
enter
```

scf2b.in

1B_1 CH2 GRHF

```
title
CH2 1B1 open shell
mult 1
zmatrix angstrom
c
h 1 1.071
h 1 1.071 2 128.65
end
basis 6-31g*
scftype gvb
open 1 1 1 1
enter
```

scf2c.in

Default MO Guess

- Superposition of atomic densities - default
 - Performs atomic SCF on each atom
 - Spherically averages occupations
 - Nearly always the best guess
- When does atomic guess fail?
 - Some ECPs
 - Many calculations on metals, especially open d/f shells
 - Diffuse basis sets
 - Some DFT calculations
- Other Approaches
 - Use eigenvectors from a related calculation
 - Restore from the same or previous Dumpfile (GETQ directive)
 - vectors stored in default sections, or in response to section specified on ENTER

Guess from Smaller MOs

- Projection guess
- use the MOs from a smaller basis as a guess
- e.g., H₂CO 6-31G guess for 6-31G** calculation
- vectors reside on the Dumpfile
- vectors from the 6-31G calculation are retrieved in the larger calculation
- use of the GETQ directive that points to the location of the 6-31G vectors on the “foreign” dumpfile (allocated using ed4).
 - This requires both location and vector section specification

```
title
h2co - 6-31G basis
zmatrix angstrom
c
o 1 1.203
h 1 1.099 2 121.8
h 1 1.099 2 121.8 3 180.0
end
basis 6-31g
enter
```

scf3a.in

```
title
h2co - 6-31G** basis
zmatrix angstrom
c
o 1 1.203
h 1 1.099 2 121.8
h 1 1.099 2 121.8 3 180.0
end
basis 6-31g**
vectors getq ed4 1 1
enter
```

scf3b.in

Conventional and Direct SCF

- Conventional SCF is the default
- Formats for storage of the 2e-integral files
 - P-supermatrix (closed shells)
 - J+K supermatrix (open shells)
 - 2e-integral format (all SCFTYPEs)
 - ⌘ supermatrix is “fastest” for small cases (< 120 GTOs)
 - ⌘ 2e-integral format the default for larger cases, with the file ≤ 3 times smaller than the corresponding supermatrix file
 - ⌘ care required when same integral file in different SCF calcs.
- To force direct
 - ⌘ **SCFTYPE DIRECT RHF**
- Disk space and elapsed times suggest avoiding use of conventional SCF for large (> 300 GTOs) cases, except:
 - ⌘ using memory to hold integrals on “large” parallel machines (64+ CPUs)

Density Functional Theory

Input for a DFT calculation is essentially that for the corresponding closed-shell RHF or UHF module, with additional keywords that control the DFT specific features. In the simplest case, the user need just introduce a single data with the character string DFT in the first data field to request a DFT rather than HF calculation:

closed-shell DFT calculation

```
TITLE
H2CO - 3-21G DFT (B-LYP DEFAULT)
ZMATRIX ANGSTROM
C
O 1 1.203
H 1 1.099 2 121.8
H 1 1.099 2 121.8 3 180.0
END
DFT
ENTER
```

dft1a.in

open-shell unrestricted UKS

```
TITLE
H2CO+ - 2B1 - 3-21G BASIS UKS
CHARGE 1
MULT 2
ZMATRIX ANGSTROM
C
O 1 1.203
H 1 1.099 2 121.8
H 1 1.099 2 121.8 3 180.0
END
SCFTYPE UHF
DFT
ENTER
```

dft1b.in

There is no restricted RKS for open-shell systems, only UKS

DFT Calculations I.

If the DFT module is switched on without specifying any options then the following functional and quadrature settings will apply;

- the Becke (1988) exchange functional
- the Lee, Yang and Parr (LYP) correlation functional
- quadrature grids designed to obtain a relative error of less than $1.0e-6$ in the number of electrons per atom. These grids are constructed from the logarithmic radial grid and Gauss-Legendre angular grid, using the SSF weighting scheme with screening and MHL angular grid pruning. ("QUADRATURE MEDIUM").
- the gradient of the energy will be evaluated without considering the gradient of the quadrature weights and grid points ("GRADQUAD OFF").

Most important DFT Directives

- The functional
- Accuracy of the numerical integration
 - Low, Medium, High, Very High
- Gradients of the quadrature

DFT Directive Specification

DFT B-LYP QUADRATURE MEDIUM

or

DFT BECKE88

DFT LYP

DFT QUADRATURE MEDIUM

DFT Calculations II.

Specification of Common Functionals

- **B3LYP**; selects the hybrid exchange-correlation energy functional due to Becke.
- **S-VWN or SVWN**; selects the LDA exchange functional and the Vosko, Wilk, and Nusair (VWN) correlation functional.
- **B-LYP or BLYP**; selects the Becke88 exchange energy functional and the Lee, Yang and Parr correlation energy functional.
- **B-P86 or BP86**; selects the Becke88 exchange energy functional and the Perdew 1986 gradient corrected correlation functional.
- **B97**; selects the Becke97 hybrid exchange-correlation energy functional
- **B97-1**; selects the Becke97 hybrid exchange-correlation energy functional as re-parametrised by Hamprecht et al.
- **HCTH**; selects the Hamprecht, Cohen, Tozer & Handy exchange-correlation functional

Specification of Integration Grids

Specify the required grid accuracy

DFT QUADRATURE LOW

The LOW accuracy grid should only be used for preliminary studies; designed to obtain the total number of electrons from the density integration with a relative error of 10^{-4} per atom.

DFT QUADRATURE MEDIUM

The MEDIUM accuracy grid - obtains a relative error of less than 10^{-6} in the N_e per atom.

DFT QUADRATURE HIGH

The HIGH accuracy grid - obtains a relative error of less than 10^{-8} in the N_e per atom.

DFT QUADRATURE VERYHIGH

... only for benchmark calculations.

DFT Calculations III.

DFT B3LYP closed shell calculation

```
TITLE
H2CO - 6-31G* B3LYP // high quad
ZMATRIX ANGSTROM
C
O 1 1.203
H 1 1.099 2 121.8
H 1 1.099 2 121.8 3 180.0
END
BASIS 6-31G*
DFT B3LYP
DFT QUADRATURE HIGH
ENTER
```

dft2.in

DFT HCTH - open-shell UKS -

```
TITLE
H2CO+ - 2B1 - TZVP BASIS HCTH UKS
CHARGE 1
MULT 2
ZMATRIX ANGSTROM
C
O 1 1.203
H 1 1.099 2 121.8
H 1 1.099 2 121.8 3 180.0
END
BASIS TZVP
SCFTYPE UHF
DFT HCTH
ENTER
```

dft3.in

Program Basics

3. Geometry Optimisation

Geometry Optimisation

Three methods are available to search for a minimum on a potential Surface,

1. the recommended method, a quasi-Newton rank-2 update procedure, is driven through the specification

RUNTYPE OPTIMIZE

Performs optimisation in internal co-ordinates, and thus requires initial ZMATRIX and VARIABLES specification of the molecular geometry, or ZMATRIX construction from a set of cartesian co-ordinates supplied under control of the GEOMETRY directive.

2. the second internal coordinate-driven method is that based on the hill-walking algorithm due to Simons and Jorgensen. Intended primarily for transition state usage, it may also be employed in geometry optimisation. The procedure is driven through additional keyword specification on the RUNTYPE directive, thus;

RUNTYPE OPTIMIZE JORGENSEN

3. the third method, perhaps less robust and flexible than the others, is a cartesian-driven update method. This is requested through

RUNTYPE OPTXYZ

Internal Co-ordinates and VARIABLES

RUNTYPE OPTIMIZE

Geometry optimisation is conducted in a system of internal coordinates - bond lengths, bond angles and dihedral angles - defined by the z-matrix.

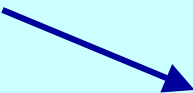
This is controlled through the introduction of so-called VARIABLES in the z-matrix. Any internal coordinate whose value is to be varied during optimisation must be specified as a VARIABLE, and an initial value assigned to it through the VARIABLE definition lines of the ZMATRIX directive.

Consider the data from the SCF computations on formaldehyde:

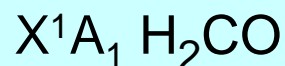
```
ZMATRIX ANGSTROM  
C  
O 1 1.203  
H 1 1.099 2 121.8  
H 1 1.099 2 121.8 3 180.0  
END
```

ZMATRIX required when optimising the geometry

```
ZMATRIX ANGSTROM  
C  
O 1 CO  
H 1 CH 2 HCO  
H 1 CH 2 HCO 3 180.0  
VARIABLES  
CO 1.203  
CH 1.099  
HCO 121.8  
END
```



Simple Optimisation Examples



geom.opt.1.in

```

TITLE
H2CO GEOMETRY TEST
GEOMETRY
0.0000000  0.0000000  0.9998722  6  C
0.0000000  0.0000000  -1.2734689  8  O
0.0000000  1.7650653  2.0942591  1  H
0.0000000  -1.7650653  2.0942591  1  H
END
RUNTYPE OPTXYZ
ENTER
  
```

Optimisation in
cartesian
Coordinates

```

TITLE
H2CO - DZ - OPTIMISATION
ZMATRIX ANGSTROM
C
O 1 CO
H 1 CH 2 HCO
H 1 CH 2 HCO 3 180.0
VARIABLES
CO 1.203
CH 1.099
HCO 121.8
END
BASIS DZ
RUNTYPE OPTIMIZE
ENTER
  
```

geom.opt.3.in

Optimisation in Internal
Coordinates

```

TITLE
H2CO - DZ - JORGENSEN OPT.
ZMATRIX ANGSTROM
C
O 1 CO
H 1 CH 2 HCO
H 1 CH 2 HCO 3 180.0
VARIABLES
CO 1.203
CH 1.099
HCO 121.8
END
BASIS DZ
RUNTYPE OPTIMIZE JORGENSEN
ENTER
  
```

geom.opt.2.in

GAMESS-UK Files:

Usage in SCF and DFT Calculations

File types

- Direct Access
 - ed0, ed1, ed2, ed3, ... , ed39 etc
- Fortran streams
 - ftn001 etc
 - used mainly by post-HF modules
- Formatted
 - punchfile
 - aimpac etc

- ed3
 - Dumpfile, often retained for restarts
 - organised into numbered sections
e.g. vectors 1
enter 2
 - summary at end of job includes section numbers in use
- ed2
 - Mainfile, integrals for conventional SCF
 - Extensive space requirements
- ed7
 - Scratchfile, modest space requirements

Section Specification on the Dumpfile

The **Dumpfile** is organised into **variable length sections**, with the user typically nominating a number of these for data storage e.g., for eigenvectors).

The **sections** are characterised by integers (in the range 1 - 350), which may be specified by the user through data input. This has normally involved such specification at two points (or more) in the data input, through the **VECTORS** and **ENTER** directive.

Both directives are used to control the reading and writing of eigenvectors, with **VECTORS** used in restart jobs to specify the location of suitable vectors for input to some SCF process, and **ENTER** used in both startup and restart jobs to specify where generated eigenvectors are to be stored.

The current code provides a set of default values so that the user may avoid the task of nominating sections. These defaults are a function of SCFTYPE.

Default Vector Sections as a function of SCFTYPE

SCFTYPE	Default Section Numbers	
Closed-shell SCF	1	
UHF	2	3
Open-shell RHF	4	5
GVB / GRHF	4	5
CASSCF	6	7
MCSCF	8	9

Routing of files

- By default files are deleted at the end of the job
- File specifications provide names and cause files to be retained
 - ✘ set environment variable outside job
 - ✘ use file directive
 - ✘ Use -t, -k -r options to rungames (described earlier)

environment variables

```
#!/bin/csh
setenv ed2 mfged2
setenv ed3 mfged3
../../bin/games << EOF
title
.....

enter
EOF
```

file directive within dataset

```
#!/bin/csh
../../bin/games << EOF
file ed2 mfged2 keep
file ed3 mfged3 keep
title
.....
EOF
```

Restarting Calculations

- Need to keep the Dumpfile
- Restart directive
 - **RESTART NEW**
 - provide new geometry
 - load old vectors, hessian
 - **RESTART <task> .. As on runtime**
e.g. **RESTART OPTIM**
 - calculation resumes the specified task, e.g. a geometry optimisation at the last stored geometry
 - **RESTART**
 - perform a new task but use the geometry as stored on the dumpfile
 - **RESTART <task> REGEN**
 - resume the task but regenerate all integral files

```

TITLE
H2CO - 3-21G DEFAULT BASIS
ZMATRIX ANGSTROM
C
O 1 1.203
H 1 1.099 2 121.8
H 1 1.099 2 121.8 3 180.0
END
ENTER

RESTART
TITLE
H2CO+ - 2B2 - 3-21G DEFAULT BASIS - UHF
CHARGE 1
MULT 2
ZMATRIX ANGSTROM
C
O 1 1.203
H 1 1.099 2 121.8
H 1 1.099 2 121.8 3 180.0
END
SCFTYPE UHF
ENTER
  
```

restart1a.in

Restart

restart1b.in

Preparing GAMESS-UK input

1. Using Molden

- Z-matrices can be prepared using Molden
 - Molden \Rightarrow ZMAT Editor
 - Restrictions
 -
- Can save z-matrix in GAMESS-UK form.
- Molden can also start simple interactive calculations
 - Molden \Rightarrow ZMAT Editor \Rightarrow Submit Job

Hands-on session

✧ Machines

```
ssh -l course01 login.hpcx.ac.uk
```

✧ HPCx

- access through rlogin/PuTTY ssh
 - **User-ids** **course01** ⇒ **course20**
 - **Passwords** **dlcode03**
- all non-graphical applications + CHARMM/GAMESS-UK
- ChemShell binary incorporates GULP

✧ Compaq DS20: tca16, DS10: tca13

- access through xterms served from tca13
 - **User-ids** **hpc01** ⇒ **hpc10**
 - **Passwords** **abc123**
- all non-graphical applications + CHARMM/GAMESS-UK
- ChemShell binary incorporates GULP

✧ 6 PCs running vista eXeed

- will run CCP1 GUI , PuTTY for login to HPCx

Directory structure

- home directories on HPCx are
 /hpcx/home/z004/z004/course01 etc
- Binaries for HPCx are in
 - /usr/local/packages/gamessuk/bin/gamess-uk.serial and
 - /usr/local/packages/gamessuk/bin/gamess-uk
- .profile should set up paths for you so that
 - GAMESS-UK can be invoked using `rungamess`
 - ChemShell can be invoked using `chemsh`
 - ALPHA Molden can be invoked as `/workshop/alpha/bin/molden (....)`
- Some pre-prepared datasets based on the examples given in these tutorials are under
 ~/serial and ~/parallel

Simple LoadLeveler (HPCx)

- `llsubmit <script>`
 - prints jobid
- `llq | grep course01`
 - listing jobs (jobids and status : R = Running, I = Waiting, C=Exiting)

\$ llq

Id	Owner	Submitted	ST	PRI	Class	Running On
13f41.2618.0	lcc33	4/2 04:04	R	50	par2_6	11f36
11f41.2921.0	chan	4/2 10:00	R	50	par8_6	12f35
14f42.2565.0	phil	4/2 10:04	R	50	par16_12	13f31
.....						
12f42.910.0	dph0vts	4/2 18:35	I	50	par16_6	
13f41.2656.0	dph0vts	4/2 18:35	I	50	par16_6	
13f42.2621.0	natalia	4/2 18:37	I	50	par2_1	

52 job step(s) in queue, 35 waiting, 0 pending, 17 running, 0 held, 0 preempted

- `llcancel <jobid>`
 - Delete job (either running or queued)
- Further documentation

http://www.hpcx.ac.uk/support/documentation/UserGuide/HPCxuser/Batch_Processing.html

IBM's Loadleveler on HPCx

16 processor parallel job

```
#!/bin/sh
#@ shell = /bin/sh
#@ job_type = parallel
#@ account_no = z004
#@ job_name = test1
#@ output = ~course00/para
#@ input = ~course00/para
#@ error = ~course00/para
#@ wall_clock_limit = 00:30
#@ tasks_per_node = 8
#@ node = 2
#@ node_usage=not_shared
#@ network.LAPI = csss,not_shared,US
#@ queue
cd ~/work
/usr/bin/poe /usr/local/packages/gamessuk/bin/gamess-uk
```

```
export LAPI_USE_SHM=yes
export MP_SHARED_MEMORY=yes
export MP_EUILIB=us
export MP_EUIDEVICE=csss
export MP_MSG_API=lapi
export MP_CSS_INTERRUPT=yes
export AIXTHREAD_SCOPE=S
export MP_POLLING_INTERVAL=25000
export RT_GRQ=ON

export MP_EUILIBPATH=/usr/local/lib/ha19:$MP_EUILIBPATH
export LIBPATH=$MP_EUILIBPATH:$LIBPATH
```

```
rungamess -p 16 -q -T 20 test1
```

Visualisation and GAMESS-UK

1. Using Molden

- Molden reads data from the GAMESS-UK output file
 - ✘ read output (loads basis, vectors etc)
 - Molden \Rightarrow read
 - ✘ optimised structure
 - Displayed by default
 - ✘ calculating and display orbitals
 - Molden \Rightarrow dens. Mode \Rightarrow orbital
 - ✘ frequencies
 - change the input to use `RUNTYPE FORCE`
 - run the calculation, read the output
 - Molden \Rightarrow Norm. Mode

Visualisation and GAMESS-UK

2. Using the CCP1 Python GUI

- Available on Windows Systems:
- Open vtkgraph.py link using Python
- Can read .xyz or .pdb files (or start from a single atom)
- Z-matrix editor
 - ✘ must be fully connected by bonds to work
- GAMESS-UK calculation editor
 - ✘ Provide parameters
 - ✘ Job page - choose hpcx or localhost (Pentium on Windows)