EMSL

Introduction of the NWChem software

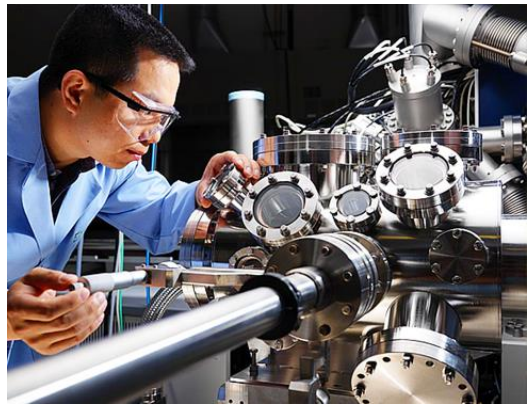
Daniel Mejia-Rodriguez,
Edoardo Aprà & Niri Govind

2023 CyberWorkshop

Environmental Molecular Sciences Laboratory (EMSL)

EMSL

https://www.emsl.pnnl.gov/



A U.S. Department of Energy national user facility

- Molecular-level understanding of physical, chemical, and biological processes
- Addresses DOE-BER environmental and energy production challenges
- ~650 scientific users annually from academia, national labs, and industry
- 150+ state-of-the-art and support instruments

NWChem: Background

- NWChem is part of the Molecular Science Software Suite

- Designed and developed to be a highly efficient and portable **Massively Parallel** computational chemistry package
- Provides computational chemistry solutions that are scalable with respect to chemical system size as well as MPP hardware size

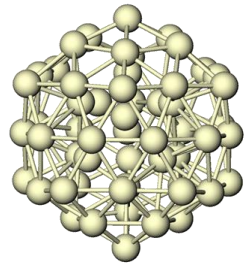
NWChem Overview

- Core development activity at EMSL/PNNL
- Designed for parallel architectures
- Emphasis on modularity, portability, and integration
- Portable – runs on a wide range of computers
  - Supercomputer to Mac or PC with Windows
- Uses Global Arrays/ARMCI for parallelization
- **NWChem 7.2.0 is open-source and freely available**

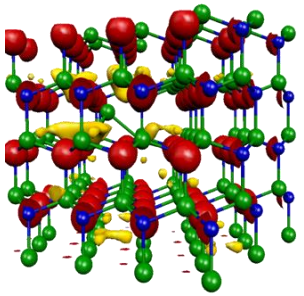
https://nwchemgit.github.io/

http://github.com/nwchemgit/nwchem

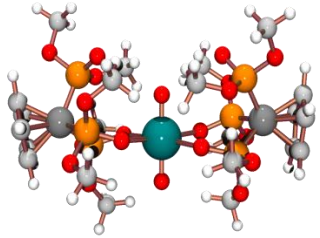
NWChem Science Capabilities

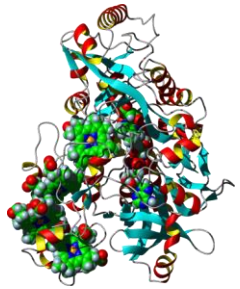


- Provides major modeling and simulation capability for molecular science
  - Broad range of **molecules**, including **bio**molecules, **nano**particles and heavy elements
  - Electronic structure of molecules (non-relativistic, relativistic, ECPs, first and second derivatives)
  - **Solid state** capability (DFT plane-wave, CPMD)
  - Molecular dynamics, molecular mechanics



- About 340/year publications citing NWChem





NWChem's core developer team

Edoardo Aprà

DFT & HPC

Eric Bylaska

Plane wave methods

Niri Govind

DFT

Karol Kowalski

Correlated Methods

Daniel Mejía-Rodríguez

DFT/GW

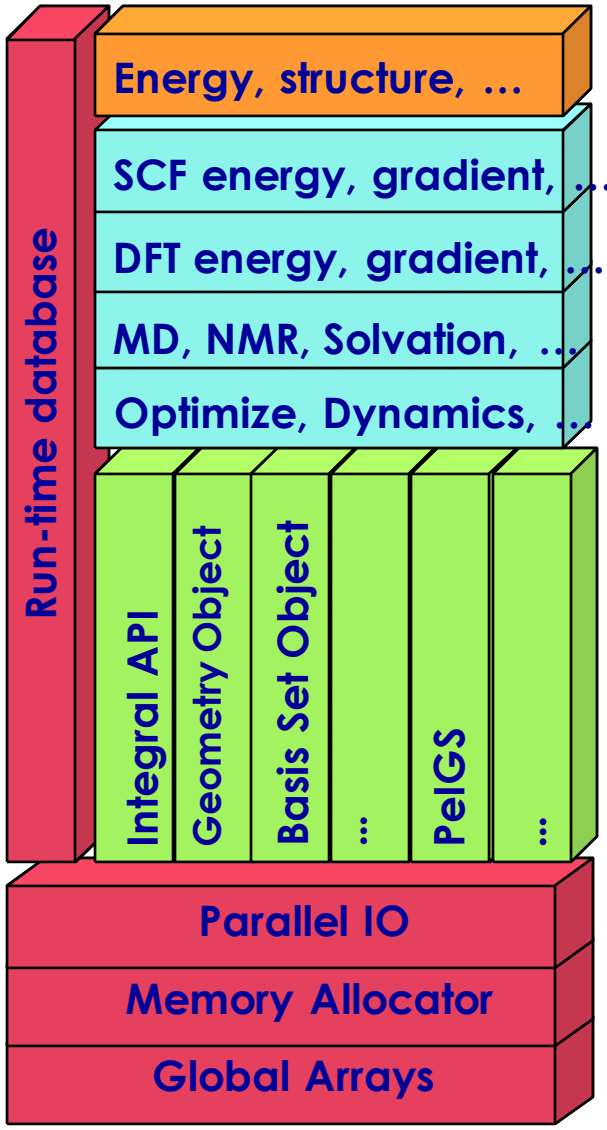
Marat Valiev

QM/MM

NWChem methodologies overview

- NWChem brings a full suite of methodologies to solve large scientific problems
  - High Accuracy Methods → MP, CC, EOMCC, MRCC
    - Ground & Excited States & Linear response
  - Gaussian-based DFT/TDDFT
    - Ground & Excited States, Optimization, Properties (NMR, Electric field gradient, linear response,…)
  - Plane wave based DFT
    - Car-Parinello MD (CPMD), Band Structure, Optimization, etc.
  - Molecular Dynamics, Molecular Mechanics
  - Integrated Methodologies → QM/MM
  - Scripting → Python

NWChem Structure



- **Object-oriented design**
  - abstraction, data hiding, APIs
- **Parallel programming model**
  - non-uniform memory access, Global Arrays, MPI
- **Infrastructure**
  - GA, Parallel I/O, RTDB, MA, …
- **Program modules**
  - communication only through the database
  - persistence for easy restart

NWChem: Ground, Excited-state & response properties of molecular systems

- Electronic structure methods for single point calculations
- Geometry optimization and transition state search
- Vibrational HF and DFT
- Spectroscopic properties (UV/Vis, IR, Raman, X-ray, NMR, EPR, non-linear optical properties)
- Chemical reactions in solutions: solvation models
- Relativistic effects
- Dynamics on a ground state potential energy surface
- Free Energy Sampling Techniques
- partial atomic charges from Electrostatic Potential (ESP)

NWChem: Gaussian DFT

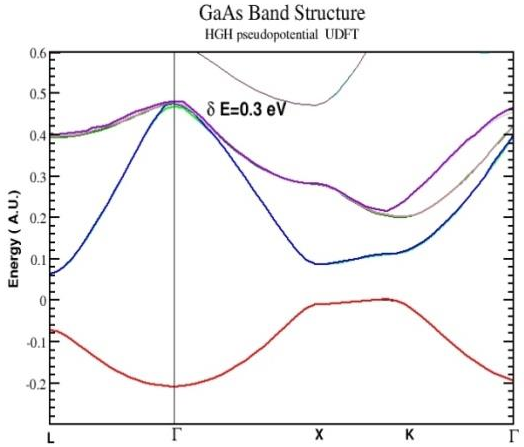
- Gaussian based DFT → Finite systems (molecules, clusters, nanostructures)
  - Wide range of local and non-local exchange-correlation functionals
    - LDA & GGA XC functionals
    - Wide range of hybrid functionals (B3LYP, PBE0, HF exchange, …)
    - Meta-GGA functionals
    - Minnesota functionals (M05, M06, M11, etc …)
    - Range separated functionals
    - DFT + D implementation (long-range empirical vdW)
  - Spin-orbit DFT
    - ECP, ZORA, DK
  - Constrained DFT
  - IR frequencies
  - Linear-response & Real-time TDDFT TDDFT for excited states → Optical spectroscopy (UV/Vis, XAS, …)
  - Various properties (NMR, Linear response, Raman,…)

NWChem: High Accuracy Methods

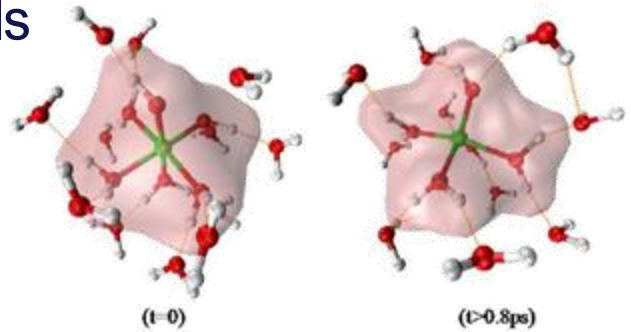
- Coupled Cluster
  - Closed shell coupled cluster [CCSD and CCSD(T)]
  - Tensor contraction engine (TCE)
    - Spin-orbital formalism with RHF, ROHF, UHF reference
    - CCSD,CCSDT, …
    - CCSD(T), CR-CCSD(T), …
    - EOMCCSD,EOMCCSDT
    - Linear response CC (polarizabilities, hyperpolarizabilities)
    - Active-space CCSDt/EOMCCSDt
    - Multi-reference CC : BW-MRCCSD, Mk-MRCCSD, BW-MRCCSD(T), Mk-MRCCSD(T)

NWChem: Plane wave (1)

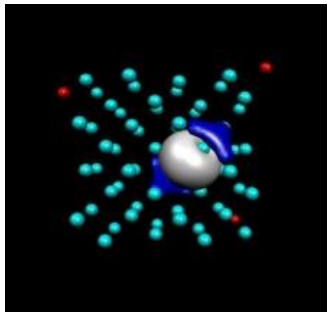
- Plane wave density functional theory
  - Gamma point pseudopotential and projector augmented wave
  - Band structure (with spin-orbit ZORA)
  - Extensive dynamics functionality with Car-Parrinello
  - AIMD QM/MM molecular dynamics, e.g. SPC/E,CLAYFF solid state MD
  - Various exchange-correlation functionals
    - LDA, PBE96, PBE0, B3LYP
    - Exact exchange
  - SIC and OEP



Spin-Orbit splitting in GaAs



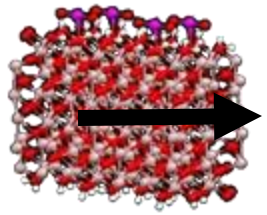
Car-Parrinello provides evidence for five-coordinated $Al(H_2O)_4OH^{2+}$
Swaddle et al, ***Science*, 2005**



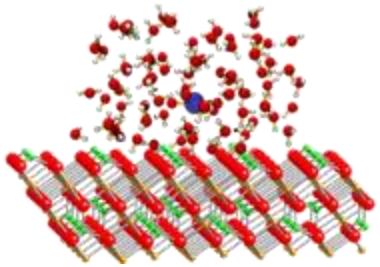
SIC localization

NWChem: Plane wave (2)

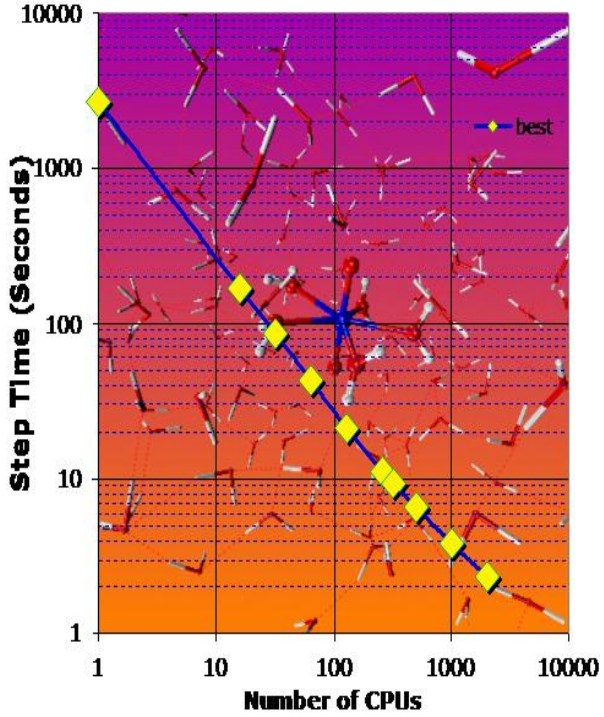
- Can handle charged systems
- A full range of pseudopotentials and a pseudopotential generator
- A choice of state-of-the-art minimizers
- Can also do plane-wave QM/MM



Uranyl on a hydroxylated Al_2O_3 surface



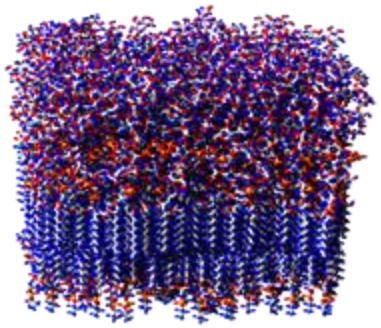
Uranyl in solution interacting with iron oxide



Car-Parrinello plane wave performance, PBE96 GGA Functional, -300 K thermostat, 0.121 fs time step, 122 water molecules-15.6 Å box

NWChem: Molecular Dynamics

- Molecular dynamics
  - Charmm and Amber force fields
  - Various types of simulations:
    - Energy minimization
    - Molecular dynamics simulation including *ab initio* dynamics
    - Free energy calculation
    - Multiconfiguration thermodynamic integration



NWChem: Hybrid QM/MM

- Seamless integration of molecular dynamics with Coupled Cluster and DFT
  - Optimization and transition states
  - QM/MM Potential of Mean Force
  - Modeling properties at finite temperature
    - Excited States with EOMCC, TDDFT
    - Polarizabilities with linear response CC
    - NMR chemical shift with DFT
- QM/MM for pathways
  - NEB-QM/MM approach for Reaction Pathway Calculations
  - Free energy calculation

NWChem: AIMD for Ground & Excited States

New Gaussian basis AIMD module

- Compatible with all Gaussian basis function based electronic structure methods in NWChem
  - Will work with numerical gradients if analytical gradients are absent
- Molecular systems, finite clusters
- Velocity Verlet
- NVE and NVT ensembles
  - Berendsen[1], Langevin[2], and stochastic velocity rescaling[3]
- Standalone program provided to analyze trajectories

[1]Berendsen, *et al. J. Chem. Phys.* **81**, 3684–3690 (1984)
[2]Bussi, Parrinello, *Phys. Rev. E* **75**, 056707 (2007)
[3]Bussi, *et al. J. Chem. Phys.* **126**, 014101 (2007)

NWChem: other functionality

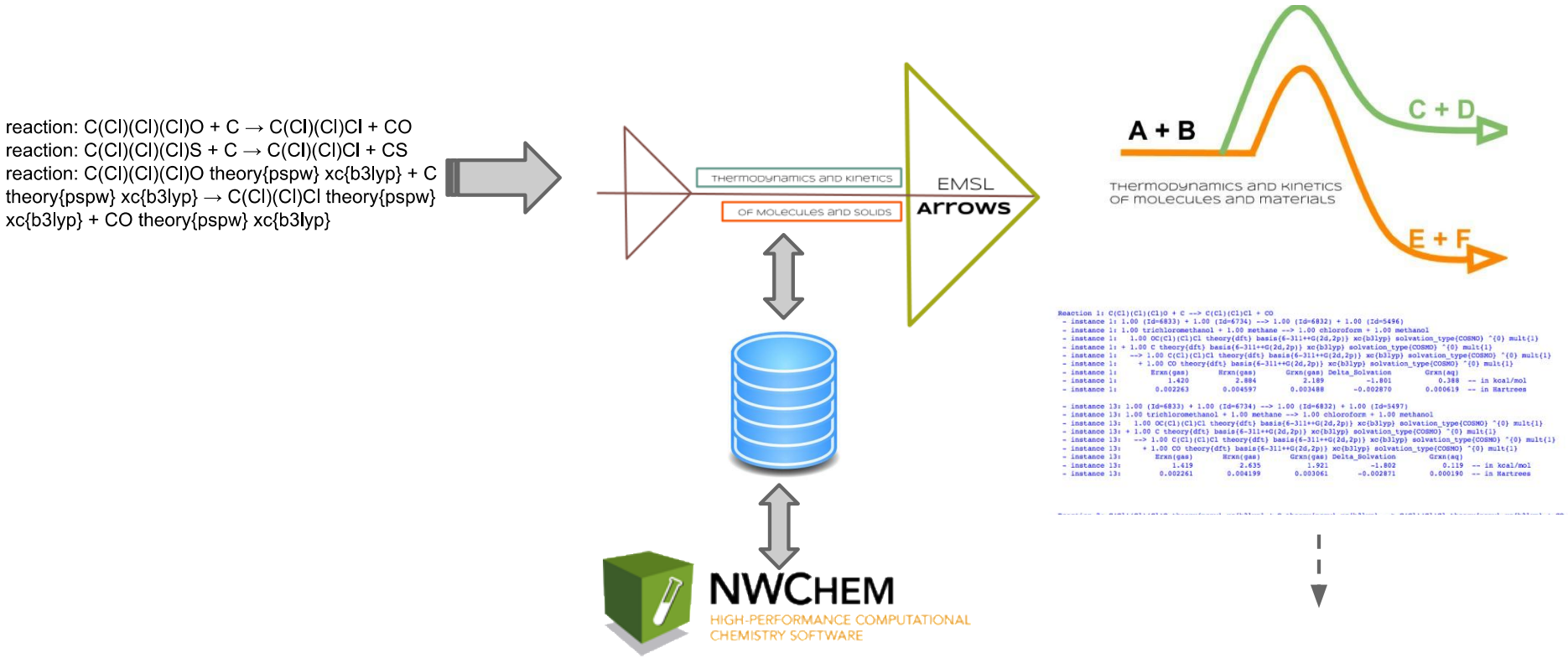
- Other functionality available in NWChem
  - NMR shielding and indirect spin-spin coupling
  - COSMO
  - ONIOM
  - Relativity through spin-orbit ECP, ZORA, and DK
  - Electron transfer
  - Vibrational SCF and DFT for anharmonicity
  - Module for dynamic nucleation theory Monte Carlo
  - Interface with VENUS for chemical reaction dynamics
  - Interface with POLYRATE, Python
  - Interface with NBO

NWChem: Interfaces to other programs

- Recent developments
  - XTB
  - Plumed
  - libxc
  - Simint
  - OpenBLAS
  - Scalapack
  - QIS
  - Molden output

EMSL Arrows: Making molecular modeling accessible

- **EMSL Arrows** is a very simple way to use NWChem.
- The user emails chemical reactions to arrows@emsl.pnnl.gov and
- then an email is sent back with thermodynamic, reaction pathway (kinetic), spectroscopy, and other results.



EMSL Arrows can calculate

- DFT, PSPW, BAND, HF, MP2, CCSD(T)
- Reaction thermodynamics for molecular systems
- Reaction paths for molecular systems
- IR, Raman spectra, UV-vis for molecular systems, phonon spectra for materials systems
- NMR and EXAFS spectra for molecular and materials systems
- Energetics, structures, and band structures of crystals using the Crystal Open Database
- A variety of datafiles can be returned including XYZ files, CIF files, NWChem output files

NWChem Best Practices – Installation Tips

- Packages available from Linux distributions (e.g. RPM)
  - Mostly well built and read for desktop computers
- Container images (Docker, Singularity, Shifter, Podman)
- Conda
- Homebrew
- When starting from source, be sure of
  - Use the most efficient GA/ARMCI
  - Link with optimized BLAS and Scalapack
  - Play nicely with compilers …
  - Validate installation with Q&A suite
  - Follow the compilation hints from the NWChem website
  - We can help you, please contact us at
    https://nwchemgit.github.io

NWChem Best Practices – Input files

- Keep it simple
  - Most of the defaults are suited for most cases
- Use direct algorithms
  - HF/DFT direct algorithm prevent I/O
  - Default is I/O that might not be suitable for your HW
- Use memory based algorithms (a.k.a in-core)
  When enough aggregate memory is available,

NWChem Input Basics

- Minimal input (all defaults)

```
geometry
  n     0.00 0.00 0.00
  n     0.00 0.00 1.08
end

basis
  n library cc-pvdz
end

task scf
```

- Performs a closed-shell SCF on the N_2 molecule

Geometry Input: Units

- Input can be in Angstrom

```
geometry  # units are in angstroms
  C    0   0        0
  H    0   0.9885 -0.4329
  H    0 -0.9885   0.4329
end
```

or atomic units

```
geometry units au  # change units to a.u.
  C    0   0        0
  H    0   1.868 -0.818
  H    0 -1.868   0.818
end
```

Geometry Input: Symmetry

- CH2 molecule with C_{2v} symmetry

```
geometry units au #input using symmetry
  C   0  0     0
  H   0  1.868 -0.818
  symmetry c2v
end
```

- C_{60} with I_h symmetry

```
geometry  #bonds = 1.4445 and 1.3945 Angstrom
    symmetry Ih
    c -1.2287651   0.0    3.3143121
end
```

Geometry Input: autosym and autoz

- By default NWChem will:
  - Attempt to find ***symmetry*** if none is specified
  - Attempt to build a ***z-matrix*** from cartesian coordinates (for the geometry optimization)
  - ***Center*** the molecule in the reference frame
  - The input below turns off these three steps (not recommended!)

```
geometry noautoz noautosym nocenter
   C    0   0        0
   H    0   0.9885 -0.4329   #Angstroms
   H    0 -0.9885   0.4329
end
```

Geometry Input: zmatrix

- Geometry can be specified using a z-matrix format

```
geometry
     zmatrix
       O
       H1 O 0.95
       H2 O 0.95 H1 108.0
     end
 end
```

Geometry Input: zmatrix

- Distances and angles can be specified with variables

```
geometry
    zmatrix
      O
      H1 O doh
      H2 O doh H1 ahoh
      variables
        ahoh 108.0
      constant
        doh 0.95
    end
 end
```

Geometry Input: zcoord

- Forcing internal coordinates (use with care …)

```
geometry
  Si         0.0000E+00  0.0000E+00  0.0000E+00
  H         -0.9436E+00 -0.8807E+00  0.7319E+00
  H          0.7373E+00 -0.8179E+00 -0.9932E+00
  H         -0.7835E+00  0.1038E+01 -0.7137E+00
  Si         0.1699E+01  0.1556E+01  0.1695E+01
  H          0.7715E+00  0.2377E+01  0.2511E+01
  H          0.2544E+01  0.6805E+00  0.2539E+01
  H          0.2514E+01  0.2381E+01  0.7713E+00
end

### fix the Si-Si distance to 4.0 angstroms ###
geometry adjust  # initial state
  zcoord
    bond 1 4 4.00 r constant
  end
end
```

Geometry Input: system

- Crystal lattice, used in plane wave code, for 3-D periodic systems (crystals)

```
geometry units angstroms center noautosym
  noautoz print
  system crystal
    lat_a 3.625d0       #diamond
    lat_b 3.625d0
    lat_c 3.625d0
    alpha 90.0d0
    beta  90.0d0
    gamma 90.0d0
  end
  C     -0.50000d0  -0.50000d0  -0.50000d0
  C      0.00000d0   0.00000d0  -0.50000d0
  C      0.00000d0  -0.50000d0   0.00000d0
  C     -0.50000d0   0.00000d0   0.00000d0
  C     -0.25000d0  -0.25000d0  -0.25000d0
  C      0.25000d0   0.25000d0  -0.25000d0
  C      0.25000d0  -0.25000d0   0.25000d0
  C     -0.25000d0   0.25000d0   0.25000d0
end
```

Basis Set Input: Using libraries

- Atoms can be defined by symbol and name

```
basis
      O library cc-pvdz
      H1 library cc-pvdz file /home/me/nwchem/libraries/
      H2 library sto-3g
end
```

- * (asterisk) can be used to state that all atoms in the system should be using the same basis set type

```
basis
      * library cc-pvdz
end
```

Basis Set Input: Explicit basis sets

- Basis set input can be done with exponents and coefficients

```
basis spherical
  H s
    13.0100 0.019685
    1.9620   0.137977
    0.4446   0.478148
    0.1220   0.501240
 H s
    0.1220   1.000000
 H p
    0.7270 1.000000
end
```

Basis Set Input: Explicit basis sets

- Libraries and explicit input can be used together

```
basis spherical
  * library cc-pvdz
H p
  0.007270 1.000000
end
```

- Main library in sync with the obsolete Basis Set Exchange
  - https://bse.pnl.gov
- Alternative library in sync with the current Basis Set Exchange
  - https://www.basissetexchange.org/

Task Input

- The Task directive tells NWChem what it should do

```
task scf                    # default is energy
task scf energy

task dft optimize
task dft saddle
task ccsd frequencies

task pspw optimize

task md dynamics
```

Task Input

- Tasks are performed in sequence as listed in input

```
task scf energy

task dft optimize ignore  # ignore if failed, go
  to next task

task dft saddle

task ccsd frequencies
```

Restarting a calculation

- To restart NWChem will need certain files, that should be saved in permanent directory

```
start ne
permanent_dir /users/me
geometry
   ne 0 0 0
end
basis
   ne library cc-pvdz
end
task scf

restart ne
permanent_dir/users/me
scf
   thresh 1e-8
end
task scf
```

Setting memory and charge keyword

- If NWChem fails with an error asking for more memory, you can set it explicitly

```
memory 2400 mb
```

- Remember, memory is per processor!
- By default, molecules have a neutral charge (0)

```
charge -1
```

NWChem website

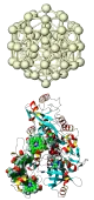
https://nwchemgit.github.io

NWChem

NWChem: Open Source High-Performance Computational Chemistry

The NWChem software contains computational chemistry tools that are scalable both in their ability to efficiently treat large scientific problems, and in their use of available computing resources from high-performance parallel supercomputers to conventional workstation clusters.

NWChem can handle:



- Biomolecules, nanostructures, and solid-state
- From quantum to classical, and all combinations
- Ground and excited-states
- Gaussian basis functions or plane-waves
- Scaling from one to thousands of processors
- Properties and relativistic effects

NWChem is actively developed by a consortium of developers and maintained by the The Environmental Molecular Sciences Laboratory (EMSL) located at the Pacific Northwest National Laboratory (PNNL) in Washington State. Researchers interested in contributing to NWChem should review the Developers page. The code is distributed as open-source under the terms of the Educational Community License version 2.0 (ECL 2.0).

The NWChem development strategy is focused on providing new and essential scientific capabilities to its users in the areas of kinetics and dynamics of chemical transformations, chemistry at interfaces and in the condensed phase, and enabling innovative and integrated research at EMSL. At the same time continued development is needed to enable NWChem to effectively utilize architectures of tens of petaflops and beyond.

Latest NWChem release

NWChem version 7.2.0 is the latest release available for download from the link https://github.com/nwchemgit/nwchem/releases.

EMSL Arrows

Are you just learning how to use NWChem and would like to have an easy way to generate input decks, check your output decks against a large database of calculations, perform simple thermochemistry calculations, calculate the NMR and IR spectra of a modest size molecule,

Github repository

https://github.com/nwchemgit/nwchem

- Source code repository
- Release files
- Documentation in wiki format
- Issue channel for bug reports

Questions?

Thank you