SC21 STUDENT CLUSTER COMPETITION

QUANTUM ESPRESSO INTRO STUDENT CLUSTER COMPETITION

QUANTUMESPRESSO

YE LUO
Argonne National Laboratory

COLLEEN BERTONI
Argonne National Laboratory

Jun. 28th 2021

MANIFESTO

- QUANTUM ESPRESSO is an integrated suite of Open-Source computer codes for electronic-structure calculations and materials modeling at the nanoscale.
  It is based on density-functional theory, plane waves, and pseudopotentials.

- https://www.quantum-espresso.org/project/manifesto

SCIENCE, ALGORITHMS

QUANTUM MECHANICS

For microscopic material properties

Time-dependent Schrödinger equation *(general)*

$$i\hbar\frac{d}{dt}|\Psi(t)\rangle = \hat{H}|\Psi(t)\rangle$$

Time-independent Schrödinger equation (*single nonrelativistic particle*)

$$\left[\frac{-\hbar^2}{2m}\nabla^2 + V(\mathbf{r})\right]\Psi(\mathbf{r}) = E\Psi(\mathbf{r})$$

Time-independent Schrödinger equation (*general*)

$$\hat{\mathrm{H}}\,|\Psi\rangle = E|\Psi\rangle$$

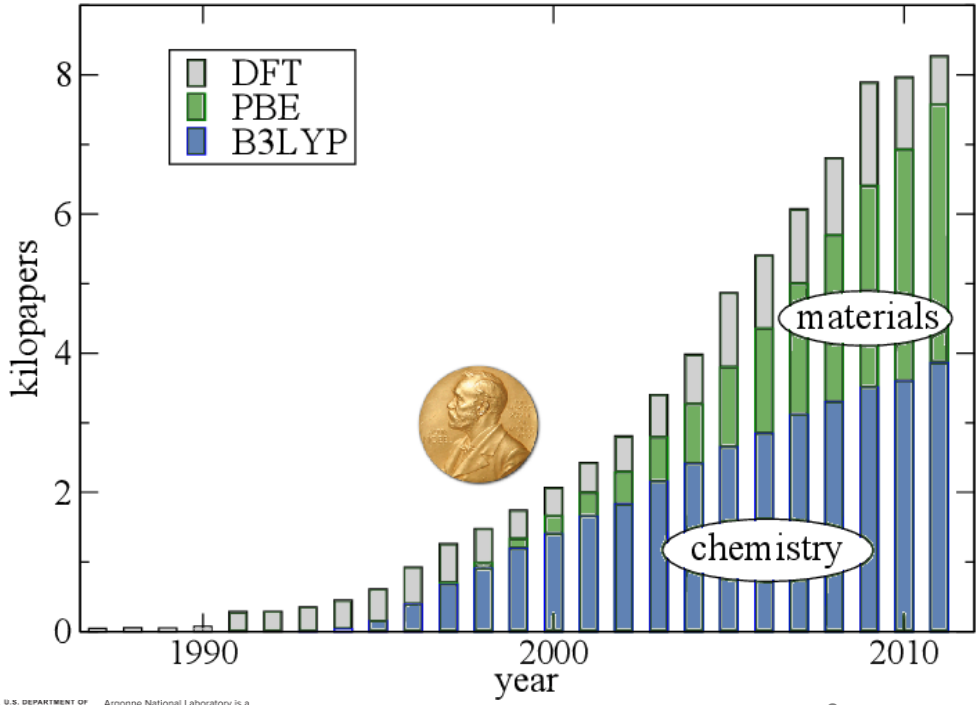
We solve this simplified case which is still extremely difficult

DENSITY FUNCTIONAL THEORY

- **Hohenberg–Kohn theorems**
  - **Theorem 1.** The external potential (and hence the total energy), is a unique functional of the electron density.
  - **Theorem 2.** The functional that delivers the ground-state energy of the system gives the lowest energy if and only if the input density is the true ground-state density.
- **Kohn-Sham equation**
  - The ground-state density of the interacting system of interest can be calculated as ground-state density of an auxiliary non-interacting system in an effective potential
  - Solvable with approximations, LDA

BOOMING PUBLICATIONS

Highly correlated with the availability of HPC clusters



Numbers of papers when DFT is searched as a topic

Perspective on density functional theory
K. Burke, J. Chem. Phys. **136**, 150901 (2012)

MANY DFT CODES AROUND THE WORLD

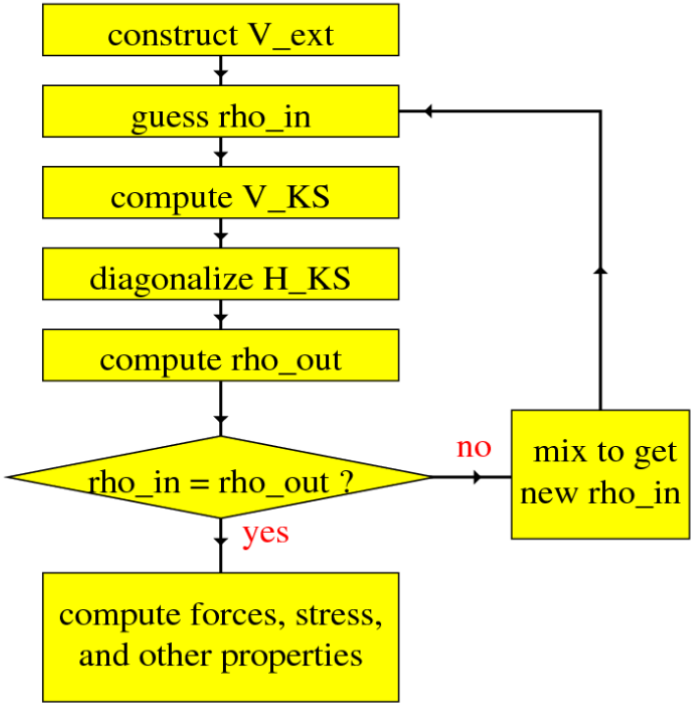
- Local orbital basis codes
  - QUEST: SeqQuest - gaussian basis pseudopotential code
  - SIESTA - numerical atom-centered basis pseudopotential code
  - CRYSTAL - CSE - gaussian basis all-electron code
  - AIMPRO
  - FHI-AIMS - (commerical license) full potential, all-electron, numerical orbitals
  - FPLO
  - OpenMX - GPL - numerical atom-centered basis PP code (Ozaki group)
- All-electron (augmented methods) codes
  - ELK - GPL - FP-LAPW
    (one branch from the old EXCITING code)
  - EXCITING - FP-LAPW, focus on excited state properties (TDDFT, MBPT)
    [license not apparent on website, probably open source]
    (another branch from the old EXCITING code)
  - FLEUR - "freely available" - FLAPW code
  - RSPt - "Open Source" - FP-LMTO
  - WIEN2k - modest fee - full potential LAPW
- Plane wave and related (real space, wavelet, etc.) methods
  - VASP
  - CASTEP
  - CPMD
  - ABINIT - GPL
  - BigDFT - GPL - wavelets
  - Quantum-Espresso (formerly PWscf) - GPL
  - PEtot - GPL
  - DACAPO - GPL
  - Socorro - GPL
  - JDFTx formerly known as DFT++ - GPL
  - Paratec
  - PARSEC - GPL - real space, pseudopotential
  - CP2K - GPL (mixed basis DFT)
  - GPAW - GPL - real-space multigrid PAW code
  - SPHINX
  - QBOX - GPL - plane wave pseudopotential, large parallel

https://dft.sandia.gov/codes_list.html

TECHNICAL DETAILS

PLANEWAVE IMPLEMENTATION

Structure of a self-consistent type code



- Dual space formalism
- V_KS in real-space
- H_KS in reciprocal space
- fast Fourier transform (FFT)
- Iterative solver with subspace diagonalization

Electronic Structure: from BlackBoard to Source Code
Advanced course given by Stefano de Gironcoli
https://cm.sissa.it/phdsection/descriptioncourse.php?ID=38

QE PARALLELIZATION

MPI+OpenMP+GPU

- Course level with MPI
- Fine level with OpenMP threading or GPU
- Some parallelization levels are limited to specific features.

Summary of parallelization levels in quantum ESPRESSO

group	*distributed quantities*	*communications*	*performances*
image	NEB images, phonon modes	very low	linear CPU scaling fair to good load balancing; does not distribute RAM
pool	k-points	low	almost linear CPU scaling, fair to good load balancing; does not distribute RAM
bands	KS orbitals	high	improves scaling
plane-wave	PW, G-vector coecients, R-space FFT arrays	high	good CPU scaling, good load balancing, distributes most RAM
task	FFT on electron states	high	improves load balancing
linear-algebra	subspace hamiltonians and constraints matrices	very high	improves scaling, distributes more RAM
OpenMP	FFT, libraries	intra-node	extends scaling on multicore machines

LOGISTICS

ACCESS

- Git repo https://gitlab.com/QEF/q-e
  - Current develop 98901cc0d67dfed37319187ab6ed10387c1b8f43
  - 6.8 release in July

- Forum for users
  - **users@lists.quantum-espresso.org**

- Bugs and issues
  - Gitlab issues

COMPILER AND LIBRARIES

- Harware support
  - X86_64, ARM, Power CPUs
  - NVIDIA GPU
- Compilers
  - GNU, LLVM
  - Vendor compilers
  - Only NVHPC supports GPU

- Optimized vendor library
  - 1D 2D & 3D batched FFT
  - BLAS/LAPACK
- Parallel eigensolvers
  - ScaLAPACK
  - ELPA

- GPU accelerated counter part

BUILD RECIPES

- Need C and Fortran compilers

  cmake -DCMAKE_C_COMPILER=mpicc -DCMAKE_Fortran_COMPILER=mpif90 ..

- Additional feature options
  - QE_ENABLE_OPENMP
  - QE_ENABLE_CUDA, QE_ENABLE_MPI_GPU_AWARE
  - QE_ENABLE_SCALAPACK
  - QE_ENABLE_ELPA

- https://gitlab.com/QEF/q-e/-/wikis/Developers/CMake-build-system

TESTS

- Run QE tests before benchmarking
  - ctest -L “systems--pw” --output-on-failure

- https://gitlab.com/QEF/q-e/-/wikis/Developers/Test-suite-and-test-farm#ctest-experimental

RUN SIMPLE BENCHMARKS

BENCHMARKS

Practice datasets

- https://github.com/QEF/benchmarks
- AUSURF112
  - Relatively small single socket CPU/GPU
  - Used in many publications
- PSIWAT
  - Medium size, good for a few nodes

INPUT FILE

Fortran namelist style

```
&CONTROL
   title = ' DEISA pw benchmark ',
   calculation = 'scf',
   restart_mode = 'from_scratch', !
'restart',
   tprnfor = .TRUE.,
   etot_conv_thr = 1.d-8,
   prefix = 'ausurf'
   pseudo_dir = './'
   outdir = './'
   disk_io = 'none'
   verbosity = 'high'
/
```

disk_io reduces disk read/write verbosity high prints info about memory usage details

Documentation https://www.quantum-espresso.org/Doc/INPUT_PW.html

Parameters changing accuracy may not be modified. For example, **etot_conv_thr**

RUN COMMAND LINE

- Parallel levels are controlled via command line options
  mpirun -np 32 pw.x -nk 4 -ndiag 4 -inp simulation.in
- Manage core/GPU affinity to MPI/OpenMP carefully.
- https://www.quantum-espresso.org/Doc/user_guide.pdf

OUTPUT FILE

```
Program PWSCF v.6.3MaX starts on 12Jul2018 at 21:25:59

This program is part of the open-source Quantum ESPRESSO suite
for quantum simulation of materials; please cite
    "P. Giannozzi et al., J. Phys.:Condens. Matter 21 395502 (2009);
    "P. Giannozzi et al., J. Phys.:Condens. Matter 29 465901 (2017);
     URL http://www.quantum-espresso.org",
in publications or presentations arising from this work. More details at
http://www.quantum-espresso.org/quote
```

MPI OpenMP info

```
Parallel version (MPI & OpenMP), running on      64 processor cores
Number of MPI processes:                16
Threads/MPI process:                     4
```

MPI parallelization levels info

```
MPI processes distributed on     1 nodes
R & G space division:  proc/nbgrp/npool/nimage =      16
Reading input from ausurf.in
```

...

GPU activation info

```
GPU acceleration is ACTIVE.
GPU-aware MPI enabled
```

RESULT CORRECTNESS CHECK

Total energy

Always check energy printout

- “total energy” at every self-consistent cycle
- “! total energy” if converged to required threshold
- Stop early trick

```
&ELECTRONS
    electron_maxstep = 3
```

```
$ grep “total energy”
     total energy              =     -25.49913426 Ry
     total energy              =     -25.49944087 Ry
     total energy              =     -25.49944452 Ry
     total energy              =     -25.49944107 Ry
     total energy              =     -25.49944297 Ry
     total energy              =     -25.49944296 Ry
     total energy              =     -25.49944296 Ry
!    total energy              =     -25.49944296 Ry
```

TIMING INFO

- Built-in timing info printed at the end of run
- Use wall clock time for comparison

```
init_run     :     81.62s CPU     26.85s WALL (       1 calls)
electrons    :   1250.73s CPU    354.11s WALL (       1 calls)

Called by init_run:
wfcinit      :     68.96s CPU     20.45s WALL (       1 calls)
potinit      :      2.40s CPU      1.04s WALL (       1 calls)
hinit0       :      4.51s CPU      1.72s WALL (       1 calls)

PWSCF        : 22m12.48s CPU     6m22.06s WALL
```

PERFORMANCE TIPS

- Use optimized libraries
- Map appropriate parallel levels to the simulation needs.
- Performance tips at the user-guide as well

DEMO

RECIPE

- List of actions
  - Clone repo
  - Build pw.x
  - Run tests
  - Download and run benchmark
- https://github.com/ye-luo/q-e-demo.git

Q&A