a tutorial on inq

Xavier Andrade, A. Correa and T. Ogitsu
Lawrence Livermore National Laboratory
xavier@llnl.gov
we tricked you into thinking you were going to learn to use an electronic structure program today...
you are going to write your own
History of inq

Project started one year ago
Objective: real-time TDDFT on GPUs

Use modern C++ coding techniques
inq is a library
Users

QBAll

QBAll (interface under development)

inq

Users
Features

Clean design from scratch: 13k lines of code

Designed to run on GPU supercomputers

Plane-wave and pseudopotentials

Modular and extensible implementation
Distribution

Free software
LGPL3 license

Available now from
http://gitlab.com/npneq/inq

Under heavy development
traditional paradigm

developers

electronic structure code

fortran / c / c++

developers

inq library

inq-based programs

c++

users

input file

arbitrary format

users

user scripts

bash / python
Example of an inq "input file"

double distance = 2.0;

vector<atom> geo;
geo.push_back( "N" | vec3d(0.0, 0.0, -distance/2.0));
geo.push_back( "N" | vec3d(0.0, 0.0, distance/2.0));

cell super = cell::cubic(3.0, 3.0, 6.0) | cell::periodic();

systems::ions ions(super, geo);

systems::electrons electrons(ions, basis::cutoff_energy(30.0));

auto result = ground_state::calculate(ions, electrons,
  interaction::dft(),
  scf::conjugate_gradient() | scf::mixing(0.1));
Compiling an inq code

To make compilation easy we provide inc++, a compiler wrapper that passes all the options and libraries required

inc++ nitrogen.cpp -o nitrogen
Conclusions

Not your standard electronic structure code

Use one language, not three

Work in progress, many features missing

Suggestions and contributions are welcome
Today's exercises

Exercise 0: load inq in your terminal

Exercise 1: calculate a potential energy surface and optimize a geometry

Exercise 2: calculate a new observable