The Abinit project

- Abinit is a robust, full-featured electronic-structure code based on density functional theory, plane waves, and pseudopotentials.
- Abinit is copyrighted and distributed under the GNU General Public License (“copyleft”).
- Brief history
  - The ground-state code was initially developed by Doug Allan at the Corning Research Laboratory
  - Response-function features were initially added by Xavier Gonze at Cornell
  - Development continues by a self-organized international collaboration, overseen by a steering committee and coordinated by Gonze at Université Catholique de Louvain, Belgium.
- Coding is based on modern software engineering principles
  - Design stresses modularity and reusability.
  - Strict coding and documentation style required (“self-documentation”).
  - Suite of automated tests (>400) expands with each added capability (“self-testing”).
  - Highly portable code can be compiled into sequential or parallel executables.

Learning Abinit - Objectives

- Perform ground-state density functional calculations.
  - Total energies
  - Geometry optimization
  - Band structure
  - Charge densities
  - (Electric polarization)
- Perform response-function calculations of total energy 2nd derivatives.
  - Vibrational properties
  - Elastic constants
  - Piezoelectric constants
- Know when to trust your results.
  - Convergence, convergence, convergence!
  - Independent cross-checks
- Know how to learn to use more features on your own.
- Know how to interact with the Abinit worldwide community.

Abinit Ground-State Basics

- Plane-wave formulation of density functional theory using separable pseudopotentials
- Historical approach: three-nested-loop structure
  - (Geometry (Self-consistent potential (Wave functions)))
  - Wave functions and band energies from diagonalizing $\langle \mathbf{k}+\mathbf{G} | H_{KS} | \mathbf{K}+\mathbf{G}' \rangle$
- Car-Parrinello revolution (1985): flatten loop structure
  - (Geometry – potential – wave functions) evolve together by molecular-dynamics algorithm
  - Only evaluate $H_{KS} | \varphi_n \rangle$ to converge occupied $| \varphi_n \rangle$
  - Use FFT's and operate with the potential in real space
- Abinit: non-traditional three-nested-loop structure (effectively flat)
  - Band-by-band preconditioned conjugate-gradient algorithm to converge wave functions
  - Minimizes "residuals" $\left| \left(H_{KS} - \varepsilon_n \right) | \varphi_n \rangle \right|^2$ with orthonormality constraints
  - One conjugate-gradient step for each potential iteration
  - "Subspace Hamiltonian matrix" diagonalization for eigenfunctions

Conjugate-gradient method

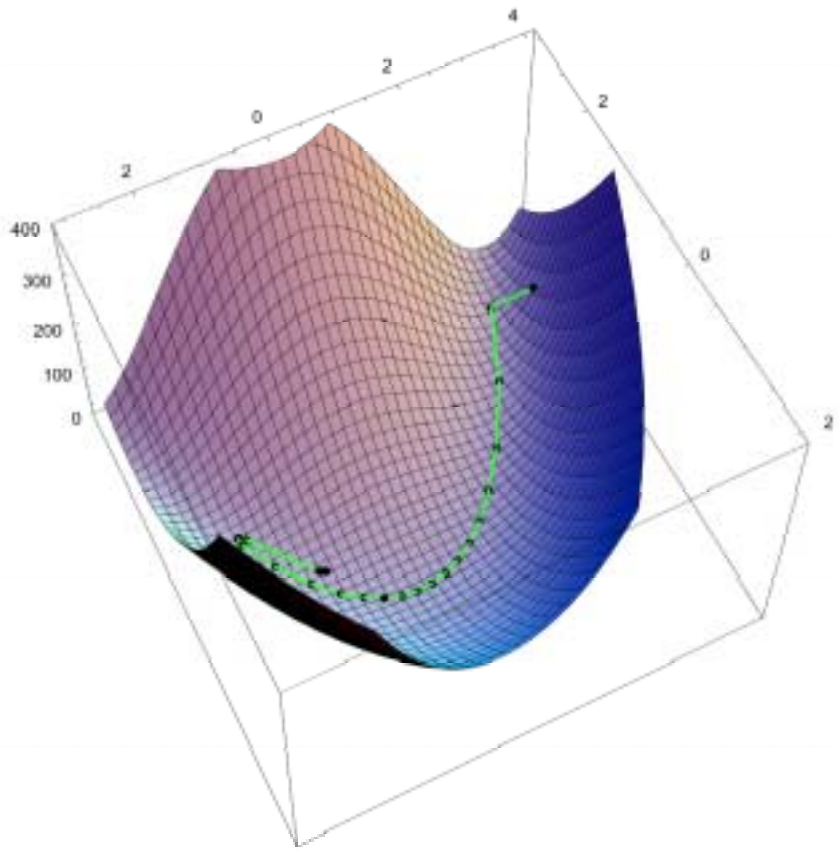
- Simplest minimum-search method is “steepest descents”
  - Follow a path pointing straight downhill from your starting point
  - Stop when you start uphill again, and pick the new downhill direction
  - This sounds smart, but can be highly inefficient, even for something as simple as a quadratic function
- Philosophy of the conjugate-gradient approach
  - Remember what directions you’ve already explored
  - Don’t bother with those directions again
  - N steps gives the exact minimum for an N-dimensional quadratic function
- The wave-function residual problem is nonlinear because of self-consistency, and of high dimension (N = number of plane waves).
- However, the C-G method is vastly more efficient in practice than in theory.
  - Number of steps <<N

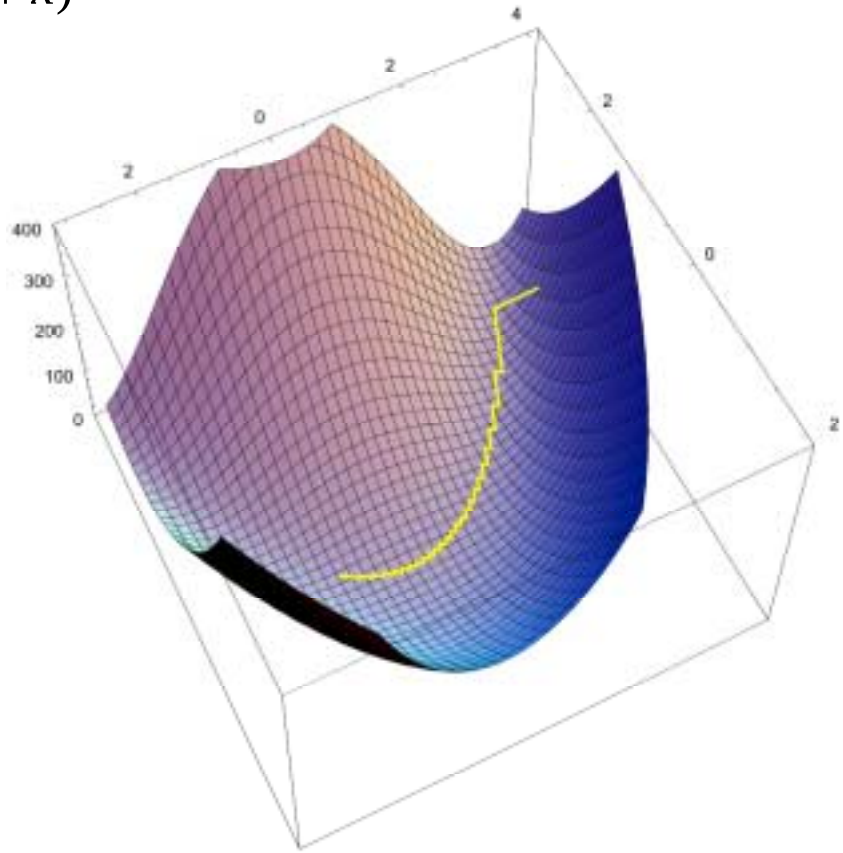
Conjugate-gradient method illustrated

Conjugate-gradient search

$$f(x,y) = 50(y - 0.25x^2) + (1+x)^2$$

Steepest-descent search





Rutgers
Department of Physics and Astronomy

Lucent Technologies
Bell Labs Innovations

Abinit potential convergence

- Potential residual for i^{th} iteration is $\left(V_{OUT}^{i} - V_{IN}^{i}\right)$
- Minimize residual-squared with conjugate-gradient or one of several mixing algorithms (which are in fact a form of steepest-descent).
- Iterative steps for potential and wave functions alternate
  - Wave functions never “start from scratch”
  - Accurate wave-function convergence is never “wasted” on a poorly converged potential

Abinit atomic geometry optimization

- Atomic forces and stress on unit cell are calculated analytically
  - These converge along with the potential and wave functions
  - When they are adequately converged, take an optimization step
- Broyden method for efficient optimization
  - Assume energy is a quadratic function of the coordinate error from optimum (including unit cell size and shape)

$$E = E^{\min} + \sum_{i,j=1}^{3N_{atom}+6} \left(X_i - X_i^{\min} \right) J_{ij} \left(X_j - X_j^{\min} \right)$$

  - Use J matrix and forces/stresses to predict next guess at $X^{\min}$
  - Build improved approximations to inverse J matrix using forces and coordinates from past steps (J is called the Jacobian or Hessian)
- Start new potential-wave function cycle with old wave functions
  - Very efficient when geometry steps are small near optimum
- Finish when forces and stresses approach zero within tolerance test

Many convenience features

- Dataset input by keywords and default values
  - Thorough documentation of input variables (keywords)
  - Flexible input format
  - Comments for documentation permitted
- Dataset error checking
  - Allowed values for input variables
  - Consistency among input variables
  - BUT misspelled keywords usually will result in defaults rather than errors
- Metalanguage for chaining different datasets
- Automatic symmetry analysis
  - Space group identified from unit cell and atomic positions
  - Irreducible Brillouin-zone wedge and k-point sample weights generated
- Dynamic memory allocation with automatic fall-back to disk if needed
- Automatic naming and renaming of output files for bookkeeping

Course Syllabus

- Abinit has self-guiding tutorials with hyperlinks to documentation
  - We will go through 6 of these at an instructor-accelerated pace
  - Become familiar with help files and input variable documentation
- Four ground-state tutorials
  1. H2 molecule – total energy, electronic enertgies, charge density, bond length, atomization energy
  2. H2 moleucle – convergence studies, LDA vs. GGA
  3. Si crystal – k samples, lattice optimization, band structure, convergence
  4. Al metal – occupation numbers, Fermi-level smearing, surface energy
- Lecture outlining theory background for response-function calculations
- Two response-function tutorials
  - Phonons in crystalline AlAs
  - Elastic and piezoelectric properties of AlAs (including ground-state treatment of electrical polarization), elastic properties of Al metal
- Final exam

Pseudopotentials for Abinit

- Abinit reads numerical tables for several types of psp's
  - Descriptions in Infos/Psp_info directory
- Psps_for_tests directory has samples for the tests and tutorials
- Psps_for_tests/HGH has Hartwigsen-Goedecker-Hutter psp's for most elements
- www.abinit.org/Psps has links to several psp collections and to two publicly available codes for generating your own
- Pseudopotential properties and issues
  - Plane-wave convergence may differ a lot among different psp's for the same element
  - May or may not include x-c potential nonlinear core corrections
  - May or may not treat semicore states as valence
- **Never trust a psp until you test it yourself for know properties of simple systems as analogous as possible to your research problem.**
  - Remember, however, that density functional theory has its limitations

Learning more about Abinit

- The Infos directory – repository of all packaged documentation
  - Infos/dirs_and_files is a guide to everything
- Infos/Features/features_v4.4.html – brief description of all present capabilities of Abinit
- Infos/*help*, Infos/*manual* files in a variety of formats (ascii text, html, postscript)
- Infos/FAQ.html – questions and responses culled from the Abinit forum listserve
- Test_v* - Included for automated testing but have examples of every type of calculation Abinit can do
  - README files give brief descriptions
  - t*.in and Refs/t*.out – sample input and output files (usually very poorly converged)
  - RunTests.cnf – at end, lists psp's in Psps_for_tests directory for each test
- Read the sources – the ultimate learning tool for more experienced users in the Src_* directories

Interacting with the Abinit community

- Browse the web site www.abinit.org
  - Executables for many platforms as well as full source tree and documentation can be downloaded
- Join the Forum, the Abinit users' mailing list
  - Read Infos/netiquette.html before submitting a question
  - Browse the archives
  - Help other users; this is strictly a volunteer operation
- Read Infos/acknowledgements.html if you're writing a paper with Abinit results
- Report bugs; better yet, fix them and report the fix
  - Read Infos/problem_report first
- Add new features
  - Under GPL, you can always do this for your own use BUT
  - It's better to share by having your feature incorporated in official releases
  - Browse the Infos/Notes_for_coding directory first
  - Join the Developers' mailing list