# Forces: Calculating Them, and Using Them

#### Shobhana Narasimhan JNCASR, Bangalore, India shobhana@jncasr.ac.in



## <u>Outline</u>

- •Forces and the Hellman-Feynman Theorem
- •Stress (v. briefly)
- Techniques for minimizing a function
- •Geometric optimization using forces
- •Frozen Phonon Calculations
- •Molecular Dynamics with forces (v. briefly)



## The Born-Oppenheimer Approximation



Max Born (1882-1970)



R. Oppenheimer (1904-1967)



Shobhana Narasimhan, JNCASR

## The Born-Oppenheimer Approximation

- The many-particle wavefunction  $\psi$  is a function of both nuclear and electronic coordinates.
- Separate out the Hamiltonian into a nuclear and electronic part.
- The wavefunction can then be written as:

$$\Psi_{tot} = \psi_{nuc} \times \psi_{elec}$$

 OK because electrons are much lighter than nuclei and therefore respond 'instantly' on time-scale of nuclear motion.



#### **Forces**

- Need for geometry optimization and molecular dynamics.
- Can also use to get phonons.
- Could get as finite differences of total energy too expensive!
- Use force (Hellmann-Feynman) theorem.
- Richard Feynman's Senior Thesis! (when he was 21...)

AUGUST 15, 1939

PHYSICAL REVIEW

VOLUME 56

Forces in Molecules

R. P. FEYNMAN Massachusetts Institute of Technology, Cambridge, Massachusetts (Received June 22, 1939)

Formulas have been developed to calculate the forces in a molecular system directly, rather than indirectly through the agency of energy. This permits an independent calculation of the slope of the curves of energy *vs.* position of the nuclei, and may thus increase the accuracy, or decrease the labor involved in the calculation of these curves. The force on a nucleus in an atomic system is shown to be just the classical electrostatic force that would be exerted on this nucleus by other nuclei and by the electrons' charge distribution. Qualitative implications of this are discussed.



X

### **Hellmann-Feynman Theorem**

• Want to calculate force on ion *I*:

$$\mathbf{F}_{I}=-rac{d}{d\mathbf{R}_{I}}\langle\Psi|H|\Psi
angle$$

• Get three terms:

$$\mathbf{F}_{I} = -\langle \Psi | \frac{\partial H}{\partial \mathbf{R}_{I}} | \Psi \rangle - \langle \frac{\partial \Psi}{\partial \mathbf{R}_{I}} | H | \Psi \rangle - \langle \Psi | H | \frac{\partial \Psi}{\partial \mathbf{R}_{I}} \rangle$$

• When  $|\Psi\rangle$  is an eigenstate,

$$H|\Psi\rangle = E|\Psi\rangle$$

-Substitute this...



## Hellmann-Feynman Theorem (contd.)

• The force is now given by

$$\mathbf{F}_{I} = -\langle \Psi | \frac{\partial H}{\partial \mathbf{R}_{I}} | \Psi \rangle - E \langle \frac{\partial \Psi}{\partial \mathbf{R}_{I}} | \Psi \rangle - E \langle \Psi | \frac{\partial \Psi}{\partial \mathbf{R}_{I}} \rangle$$
$$= -\langle \Psi | \frac{\partial H}{\partial \mathbf{R}_{I}} | \Psi \rangle - E \frac{\partial}{\partial \mathbf{R}_{I}} \langle \Psi | \Psi \rangle$$

- Note that we can now calculate the force from a calculation at ONE configuration alone – huge savings in time.
- If the basis depends on ionic positions (not true for plane waves), would have extra terms = Pulay forces.
- If |Ψ⟩ is not an exact eigenstate (electronic calculation not well converged), may get big errors in forces calculated using this prescription!



#### Using H-F Theorem in a (plane-wave) DFT calculation

• Force on ion *I* given by:

$$\mathbf{F}_{I} = -\frac{\partial E(\mathbf{R})}{\partial \mathbf{R}_{I}} = -\left\langle \Psi(\mathbf{R}) \left| \frac{\partial H_{BO}(\mathbf{R})}{\partial \mathbf{R}_{I}} \right| \Psi(\mathbf{R}) \right\rangle$$
$$\mathbf{F}_{I} = -\int n_{\mathbf{R}}(\mathbf{r}) \frac{\partial V_{\mathbf{R}}(\mathbf{r})}{\partial \mathbf{R}_{I}} d\mathbf{r} - \frac{\partial E_{N}(\mathbf{R})}{\partial \mathbf{R}_{I}}$$

where

 $V_{\mathbf{R}}(\mathbf{r}) = (\text{pseudo})\text{potential due to ion cores}$ 

and  $E_N(\mathbf{R})$  = interaction of ions with each other.



#### <u>Stress</u>

- Strain:  $\mathbf{r}_{\alpha} \rightarrow (\delta_{\alpha\beta} + \epsilon_{\alpha\beta})\mathbf{r}_{\beta}$
- Stress:  $\sigma_{\alpha\beta} = -\frac{1}{\Omega} \frac{\partial E}{\partial \epsilon_{\alpha\beta}}$
- Stress Theorem (Nielsen & Martin, 1985) → as for forces, can calculate at a single configuration.
- What if the primitive lattice vectors (specifying unit cell) are not optimal?
  - Forces on atoms may = 0
  - (e.g., an FCC crystal with wrong lattice constant)
  - Stress will not be zero, however.
    - < 0  $\rightarrow$  cell would like to expand.
    - $> 0 \rightarrow$  cell would like to contract.

#### (More in next lecture)



#### **Computing Forces & Stress with PWscf**

tprnfor = .TRUE.

(Set automatically if calculation = `relax' or `md' or `vc-md')

tstress = .TRUE.

(Set automatically if calculation = `vc-md')



#### **Forces Obtained using PWscf**

• e.g., for a point defect (vacancy) in graphene:

Forces	a	cting	on	atoms (R	ty/au)	:				
atom	1	type	1	force	=	0.00997900	-0.00310393	•	0.00000000	
atom	2	type	1	force	=	0.00461455	0.00266421	3	0.00000000	
atom	з	type	1	force	=	0.00230142	0.01019403		0.00000000	
atom	4	type	1	force	=	0.00000000	0.00025653		0.00000000	
atom	5	type	1	force	=	0.00022216	-0.00012827		0.00000000	
atom	6	type	1	force	=	0.00000000	0.01878385		0.00000000	
atom	7	type	1	force	=	0.06589779	0.05927779		0.00000000	
atom	8	type	1	force	=	0.08428497	0.02743027		0.0000000	





### What can one do with forces?

- Optimize ionic positions (geometric relaxation).
- Force constants, vibrational frequencies.
- Molecular dynamics.



## **Geometric Relaxation**



## **Geometric Optimization**

- Want to move the atomic positions around until the lowest-energy equilibrium configuration is obtained.
- At equilibrium,

$$\mathbf{F}_I \equiv -\frac{\partial E(\mathbf{R})}{\partial \mathbf{R}_I} = 0 \quad \text{for all } I.$$

• We are searching for a minimum in a  $3N_{I}$ -dim space.



## **Minimization**

- Relevant to many parts of DFT calculations:
  - Optimizing ionic positions
  - Minimizing energy functional
  - Diagonalizing Hamiltonian
  - Achieving self-consistency (mixing)

A brief foray into numerical methods....

Convention: subscripts  $\leftrightarrow$  coordinates, superscripts  $\leftrightarrow$  iterations



## Minimization in 1-D using gradients

- Consider a function f(x); we want to find  $x_0$ , the value of x where the function has its minimum value.
- Iterative methods: successive approximations  $x^1$ ,  $x^2$ ,  $x^3$ ,..., $x^n$ ..., $x_0$
- Can take several small downhill steps, in the direction opposite the gradient f'(x<sup>n</sup>).



$$x^{n+1} = x^n - \beta^{(n)} f'(x^n)$$

• Might take a long time to converge.

Note: need first derivatives



#### Finding zeroes in 1-D: the Newton-Raphson Method

• Use information from first f(x)derivatives to make a series of guesses:

$$f'(x^{n}) = \frac{f(x^{n}) - 0}{x^{n} - x^{n+1}}$$
$$x^{n+1} = x^{n} - \frac{f(x^{n})}{f'(x^{n})}$$

Note: will converge in one step if f is linear

 $x^n$ 

 $x^{n+1}$ 

Х

#### Finding minimum in 1-D: the Newton-Raphson Method

- Looking for a <u>minimum</u> in *f* is equivalent to looking for a <u>zero</u> of *f*.
- So replace *f* by *f* and *f* by *f* "to get:

$$x^{n+1} = x^n - \frac{f'(x^n)}{f''(x^n)}$$

or sometimes:

$$x^{n+1} = x^n - \beta \frac{f'(x^n)}{f''(x^n)}$$

Note: need first derivatives and second derivatives

Note: will converge in one step if f' is linear. So will converge in one step if f is quadratic.



## Minimization in an N-d space

- Consider a function  $f(\mathbf{x})$  of *N* variables  $\mathbf{x} = x_1, x_2, \dots, x_N$
- The gradient is  $\nabla f(\mathbf{x})$ . Note that at a given  $\mathbf{x}$ , this points in direction of maximum increase of  $f(\mathbf{x})$ .
- Want to find  $\mathbf{x}_0$  s.t.  $f(\mathbf{x})$  has its minimum value at  $\mathbf{x}_0$ , i.e.,  $\nabla f(\mathbf{x}) = 0$ .
- Will find iteratively, through a sequence of points x<sup>n</sup> in the *N*-dimensional space that are stepping stones to finding x<sub>0</sub>.
- Convention: subscripts  $\leftrightarrow$  coordinates, superscripts  $\leftrightarrow$  iterations.



## **Steepest Descent**

- Keep going downhill in direction opposite local gradient.
- Could try taking lots of little downhill steps:  $\mathbf{x}^{n+1} = \mathbf{x}^n - \beta \nabla f(\mathbf{x}^n) = \mathbf{x}^n + \beta \mathbf{g}^n$
- Always,  $\mathbf{g}^n$  perpendicular to  $\mathbf{g}^{n+1}$ .
- Better: Once direction  $\widehat{\mathbf{g}^n}$  identified, do line minimization along it.
- Can be slow (may not reach minimum!)
- Problem: when moving along new direction, lose some minimization along old one(s).





## Second derivatives in an N-d space:

The Hessian is an N×N matrix, whose elements are given by:

$$H_{ij} = \partial^2 f(\mathbf{x}) / \partial x_i \partial x_j$$

It gives information about the curvature of the function.



#### **Newton-Raphson Minimization in N-d**





#### **Quasi-Newton-Raphson Methods**

• We had: 
$$\mathbf{x}^{n+1} = \mathbf{x}^n - \beta [\mathbf{H}(\mathbf{x}^n)]^{-1} \nabla f(\mathbf{x}^n)$$

- But H<sup>-1</sup> may be hard/expensive to compute directly.
- Instead, build up estimates for H<sup>-1</sup> by analyzing successive gradient vectors.
- Various prescriptions for doing this, e.g., Broyden method, BFGS method.



## **BFGS Minimization**

Broyden, Fletcher, Goldfarb, Shanno (1970).

$$\mathbf{x}^{n+1} = \mathbf{x}^n - \rho^n \frac{[\mathbf{H}^n]^{-1} \nabla f(\mathbf{x}^n)}{[\mathbf{H}^n]^{-1} \nabla f(\mathbf{x}^n)}$$

Newton-Raphson

Inverse Hessian approximated as:

$$\left[\mathbf{H}^{n+1}\right]^{-1} = \left[\mathbf{H}^{n}\right]^{-1} + \left(\frac{1 + (\boldsymbol{\gamma}^{n})^{T} \left[\mathbf{H}^{n}\right]^{-1} \boldsymbol{\gamma}^{n}}{(\mathbf{s}^{n})^{T} \boldsymbol{\gamma}^{n}}\right) \frac{\mathbf{s}^{n} (\mathbf{s}^{n})^{T}}{(\mathbf{s}^{n})^{T} \boldsymbol{\gamma}^{n}} - \left(\frac{\mathbf{s}^{n} (\boldsymbol{\gamma}^{n})^{T} \left[\mathbf{H}^{n}\right]^{-1} + \left[\mathbf{H}^{n}\right]^{-1} \boldsymbol{\gamma}^{n} (\mathbf{s}^{n})^{T}}{(\mathbf{s}^{n})^{T} \boldsymbol{\gamma}^{n}}\right)$$

where

$$\mathbf{\gamma}^n \equiv \nabla f(\mathbf{x}^n) - \nabla f(\mathbf{x}^{n-1})$$
 and  $\mathbf{s}^n \equiv [\mathbf{H}(\mathbf{x}^n)]^{-1} \nabla f(\mathbf{x}^n)$ 

(No doubt you will agree that this is a good point at which to end our foray into numerical methods...)

#### **Back to the Problem of Ionic Relaxation**

- Function f to be minimized is total energy E<sub>tot</sub>.
- Points **x** in  $3N_I$ -d space correspond to set of ionic coordinates  $(x_1, y_1, z_1, x_2, y_2, z_2, \dots, x_{NI}, y_{NI}, z_{NI})$ .
- Gradients ∇*f*(**x**) correspond to set of 3 components of forces on the N<sub>I</sub> ions.
- Forces can be computed using Hellmann-Feynman theorem.
- Now use a minimization scheme to find the ionic positons that give the lowest value of  $E_{tot}$ , which is also when the forces on all ions are (close to) zero.





## **Ionic Relaxation in PWscf**

•Tell the program to carry out ionic relaxation, and say which method to use to find minimum





## **Ionic Relaxation in PWscf (contd.)**

•Say which atoms are to be moved & in which directions.

•e.g., for a four-layer Al(001) slab:

ATON	AIC_POS		Allow these					
Al	0.000	0.000	2.828	0	0	1		atoms to move
Al	0.500	0.500	2.121	0	0	1	$\int$	only along z
Al	0.000	0.000	1.414	0	0	0	7	
Al	0.500	0.500	0.707	0	0	0		Fix these atoms



## **Equilibrium Geometries**

- In some cases, there are lots of equilibrium geometries (corresponding to local minima in the energy landscape).
- Finding the <u>global</u> minimum can be challenging!

e.g., NO adsorbed on a 5-atom Rh cluster:



Ghosh, Pushpa, de Gironcoli & Narasimhan



#### Force Constants and Vibrational Frequencies



## The Vibrational Frequency of a Diatomic Molecule

• Can obtain from energies or from forces:

e.g., for CO:







Sananda Biswas



## What about a triatomic molecule?

• More useful to have forces than only total energy...





## **"Frozen-Phonon" Calculations**

Take "snapshots" of a crystal as it is vibrating in a particular mode.



- Use Hellmann-Feynman theorem to compute forces as a function of displacement.
- Force vs. displacement gives phonon frequency.
- Note: Can instead use density functional perturbation theory (tomorrow!)



#### **The Force-Constant Tensor**

 $\Phi_{\alpha,\beta}(\mathbf{R}_i - \mathbf{R}_j) \sim \text{force induced on atom } j \text{ in direction } \beta,$ upon moving atom i in direction  $\alpha$ 



Can be obtained easily if forces can be computed.



### **Getting Phonon Frequencies**

• Dynamical Matrix:

$$\widetilde{\mathbf{D}}(\mathbf{k}) = \sum_{\mathbf{R}} e^{i\mathbf{k}\cdot\mathbf{R}} \Phi(\mathbf{R})$$

• Diagonalize to get phonon frequencies:

$$\mathsf{Det}\left[\widetilde{\mathbf{D}}(\mathbf{k}) - \widetilde{\mathbf{M}}\,\omega^2(\mathbf{k})\right] = 0$$



## **Molecular Dynamics**



#### **Forces and Motion**

• Newton's equations of motion:

$$\mathbf{F}_I = m_I \mathbf{a}_I = m_I \frac{d^2 \mathbf{R}_I}{dt^2}$$

- If forces can be computed, can integrate these to get ionic positions as a function of time.
- Molecular Dynamics (next week...)







#### May the Force Be With You....

