



# Interatomic Potentials Repository Project

Lucas M. Hale

Zachary T. Trautt

Chandler A. Becker

National Institute of Standards and Technology

# What is Our Motivation?

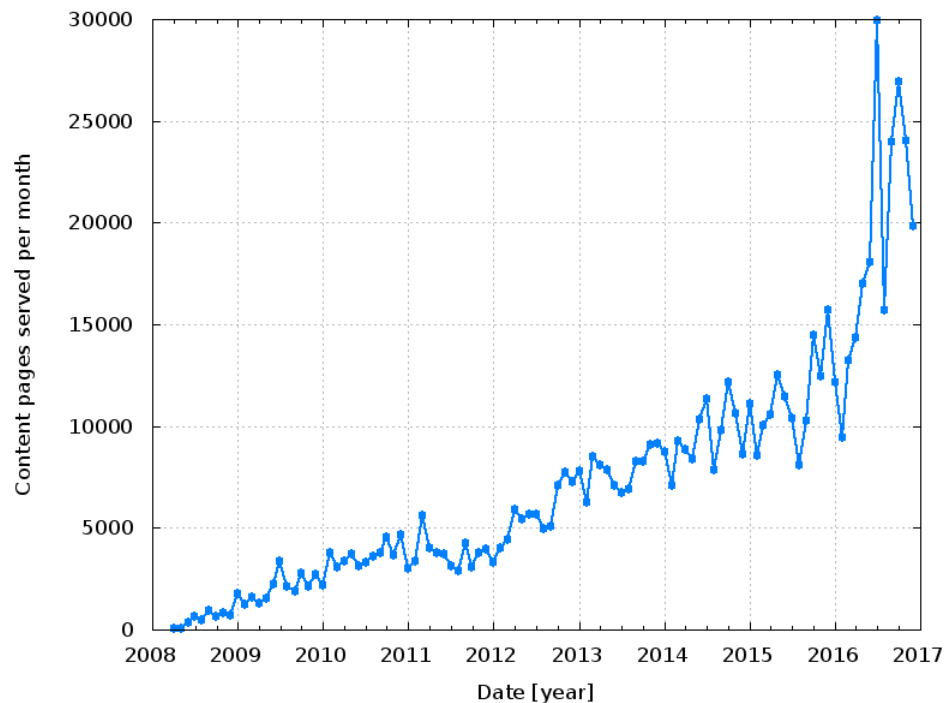
---

Provide tools and resources to make it easier for users to perform high-quality, meaningful molecular dynamics simulations.

# Interatomic Potentials Repository

<https://www.ctcms.nist.gov/potentials/>

- All formats, material systems
- 168+ total potentials
- 130+ LAMMPS-compatible
- Email [potentials@nist.gov](mailto:potentials@nist.gov) to submit



Interatomic Potentials Repository Project

Part of the Materials Genome Initiative

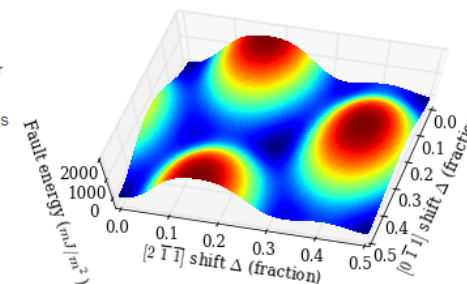
NIST  
National Institute of  
Standards and Technology  
U.S. Department of Commerce

[Home](#) [Tools](#) [References](#) [FAQ](#) [Resources](#) [People](#) [Contact](#)

## Overview

This repository provides a source for interatomic potentials (force fields), related files, and evaluation tools to help researchers obtain interatomic models and judge their quality and applicability. Users are encouraged to download and use interatomic potentials, with proper acknowledgement, and developers are welcome to contribute potentials for inclusion. The files provided have been submitted or vetted by their developers and appropriate references are provided. All classes of potentials (e.g., MEAM, ADP, COMB, Reax, EAM, etc.) and materials are welcome. Interatomic potentials and/or related files are currently available for various metals, semiconductors, oxides, and carbon-containing systems.

If you do not find the element or alloy potential you are seeking, send an email to [potentials@nist.gov](mailto:potentials@nist.gov), and we will post the request. The potentials we have available are posted on the site. Please do not request that files be sent via email. This site reflects what has been submitted, so it is not a complete survey of interatomic potentials for any or all systems. [Here is a list of recent requests for potentials.](#)



If you find this website useful, including using potentials you downloaded, please cite this project in addition to the proper interatomic potential reference:

C.A. Becker, et al., "Considerations for choosing and using force fields and interatomic potentials in materials science and engineering," *Current Opinion in Solid State and Materials Science*, 17, 277-283 (2013). <https://www.ctcms.nist.gov/potentials>

Please note that, due to the wide range of interatomic potential functions and formats, it is the user's responsibility to check that the interatomic potentials produce expected results. More information can be found in the [FAQ](#).

**WHAT IS AVAILABLE?** [Bibliography for the available interatomic potentials at this site.](#) Links are provided to the proper pages to download relevant files.

**OTHER RESOURCES:** [Additional resources \(projects, websites, etc.\)](#) related to interatomic potentials, reference data, and anything else we think might be useful.

**WHAT IS NEW?** [Recent additions](#) to the repository. Please send feedback and contributions to: [potentials@nist.gov](mailto:potentials@nist.gov)

## Interatomic Potentials (Force fields)

To see available potentials containing an element, go to the page for that element.

### Elements

1 H					2 He
3 Li	4 Be				
		5 B	6 C	7 N	8 O
				9 F	10 Ne
11	12				
		13	14	15	16
				17	18

# Potential Information

Select element or system to view  
all associated potentials

Identified by primary citation or  
author

Version and usage notes

Multiple formats allowed

Computed properties

## Gold (Au)

G. Grochola, S.P. Russo, and I.K. Snook, "On fitting a gold embedded atom method potential using the force matching method," J. Chem. Phys., 123 , 204719 (2005). DOI: [10.1063/1.2124667](https://doi.org/10.1063/1.2124667)

Notes: This file was generated by C.A. Becker from the files below that were approved by G. Grochola (RMIT University) and posted with his permission on 21 Feb. 2011. This version is compatible with LAMMPS. Validation and usage information can be found in [Au-Grochola-JCP05-conversion-notes\\_v2.pdf](#). If you use this eam.alloy file, please credit the website in addition to the original reference.

Format: EAM/alloy setfl  
File(s): [Au-Grochola-JCP05.eam.alloy](#)

Notes: These files were approved by G. Grochola.

Format: EAM table  
File(s):  
F( $\rho$ ): [Au-Grochola-JCP05-F.table](#)  
 $\rho$ (r): [Au-Grochola-JCP05-rou.table](#)  
 $\phi$ (r): [Au-Grochola-JCP05-Phi.table](#)

New! Computed Properties: [2005--Grochola-G--Au](#)

---

X.W. Zhou, R.A. Johnson, and H.N.G. Wadley, "Misfit-energy-increasing dislocations in vapor-deposited CoFe/NiFe multilayers," Phys. Rev. B, 69 , 144113 (2004). DOI: [10.1103/PhysRevB.69.144113](https://doi.org/10.1103/PhysRevB.69.144113)

Notes: This file was generated by C.A. Becker from the files sent by X.W. Zhou (Sandia National Laboratory) and posted with his permission. These files can be used to generate alloy potentials for Cu, Ag, Au, Ni, Pd, Pt, Al, Pb, Fe, Mo, Ta, W, Mg, Co, Ti, and Zr by editing EAM.input. However, as addressed in the reference, these potentials were not designed for use with metal compounds. See the [Zhou04](#) page for more information.

Format: EAM/alloy setfl  
File(s): [Au.set](#)

New! Computed Properties: [2004--Zhou-X-W--Au](#)

---

J.B. Adams, S.M. Foiles, and W.G. Wolfer, "Self-diffusion and impurity diffusion of FCC metals using the 5-frequency model and the Embedded Atom Method," J. Mater. Res. 4 , 102-112 (1989). DOI: [10.1557/JMR.1989.0102](https://doi.org/10.1557/JMR.1989.0102)

Notes: auu6.txt was obtained from <http://enpub.fulton.asu.edu/cms/potentials/main/main.htm> and posted with the permission of J.B. Adams. The name of the file was retained, even though the header information lists the potential as 'universal 4.' This file is compatible with the "pair\_style eam" format in LAMMPS (19Feb09 version).

Format: EAM  
File(s): [auu6.txt](#)

New! Computed Properties: [1989--Adams-J-B--Au](#)

---

G.J. Ackland, G.I. Tichy, V. Vitek, and M.W. Finnis, "Simple N-body potentials for the noble-metals and nickel," Phil. Mag. A 56 , 735-756 (1987). DOI: [10.1080/01418618708204485](https://doi.org/10.1080/01418618708204485)

Notes: The parameters in au moldy were obtained from <http://homepages.ed.ac.uk/graeme/moldy/moldy.html> and posted with the permission of G.J.

# Predicted Properties

Bulk properties

Defect structures

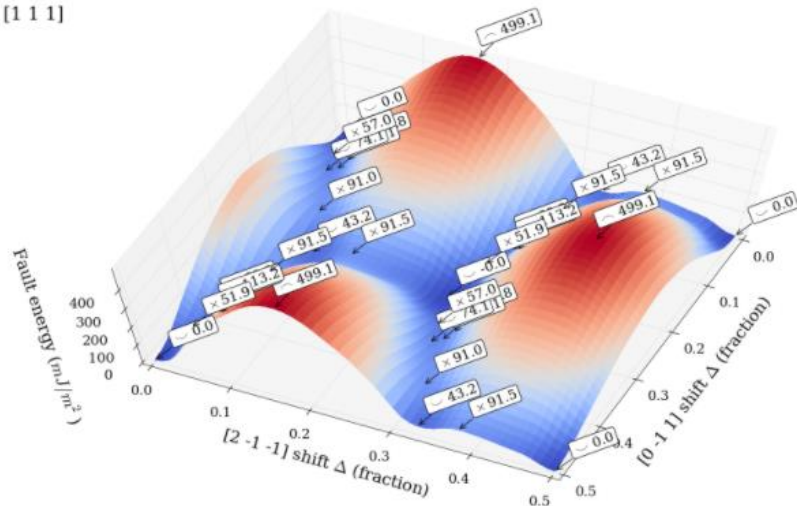
Focus on documenting methodology

0K Crystal Structure Properties for Au

prototype	$E_{\text{coh}}$ (eV)	$a_0$ (Å)	$b_0$ (Å)	$c_0$ (Å)	$C_{11}$ (GPa)	$C_{22}$ (GPa)	$C_{33}$ (GPa)	$C_{12}$ (GPa)	$C_{13}$ (GPa)	$C_{23}$ (GPa)	$C_{44}$ (GPa)	$C_{55}$ (GPa)	$C_{66}$ (GPa)
A1--Cu--fcc	-3.9242	4.0701			201.65			169.53			45.97		
A3'--alpha-La--double-hcp	-3.9194	2.8783		9.3909	229.78		245.94	158.11	142.81		27.30		
A3--Mg--hcp	-3.9147	2.8784		4.6919	227.81		252.83	156.58	135.53		28.73		
A2--W--bcc	-3.8634	3.2218			70.69			162.13			52.46		
A15--beta-W	-3.8236	5.1468			292.22			127.30			0.28		
A5--beta-Sn	-3.6657	5.2037		2.7696	193.04		186.51	5.37	28.26		8.14		-16.28
Ah--alpha-Po--sc	-3.5869	2.6799			210.13			29.73			-4.57		
A4--C--dc	-2.7489	5.6627			270.09			219.65			88.19		

Generalized Planar Fault Energy of pure Gold

Potential descriptor: Au\_GrocholaG\_2005  
 Species: Au  
 Lattice: fcc  
 Fault plane: [1 1 1]

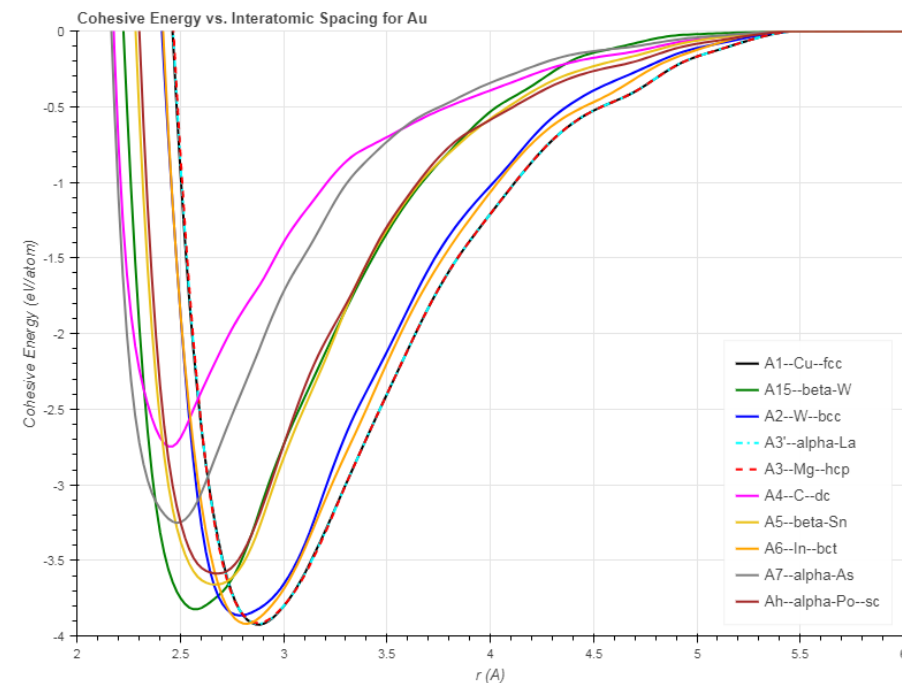


~ Minima  
 × Maxima  
 × Saddle points

Illustration of a complete (1 1 1) fault energy surface with local minima, maxima, and saddle points labeled with their respective fault energy.

Cohesive Energy vs. Interatomic Spacing for Au

[Download data](#)



# Different models for different uses

MEAM



EAM



EAM



Lennard-Jones



"Best" depends on what you want to do

ReaxFF



BOP



DFT



# iprPy Computational Framework

---

<https://github.com/usnistgov/iprPy>

Design scientific calculations that

- are open source with minimum barriers for usage,
- have transparent methodologies supporting knowledge transfer and education,
- produce results that are both human and machine readable,
- allow investigations into method and parameter sensitivity,
- and can be integrated into workflows.

# Python Calculation Scripts

Input parameter file

Calculation script

JSON/XML results

```
calc_surface_energy.in
1 #Run script for calc_surface_energy.py
2
3 #Command lines for LAMMPS (and MPI).
4 #For lammps_command, exclude passing in a script, i.e. no "-in term"
5 lammps_command ..... lmp_mpi
6 mpi_command .....
7
8 #Paths to the potential data model file, and directory containing potential
9 potential_file ..... 2017--Purja-Pun-G-P--Au--LAMMPS--test.xml
10 potential_dir ..... 2017--Purja-Pun-G-P--Au--LAMMPS--test
11
12 #Initial system configuration to load
13 load_file ..... 48fe09d1-0aec-4e8e-8638-a22fae5df0d9.xml
14 load_style ..... system_model
15 load_options ..... key:relaxed-atomic-system
16 symbols .....
17 box_parameters .....
18
19 #System manipulations
20 x_axis .....
21 y_axis .....
22 z_axis .....
23 atomshift .....
24 sizemults ..... 5 5 10
25
26 #Defect parameters
27 surface_model ..... Al--Cu--fcc--110.xml
28 surface_cutboxvector .....
29
30 #Units that input/output values are in
31 length_unit .....
32 pressure_unit .....
33 energy_unit .....
34 force_unit .....
35
36 #Run parameters
37 energytolerance .....
38 forcetolerance .....
39 maxiterations .....
40 maxevaluations .....
41 maxatommotion .....
```

```
calc_surface_energy.py
31
32 def main(*args):
33     """Main function called when script is executed directly."""
34
35     # Read input file in as dictionary
36     with open(args[0]) as f:
37         input_dict = iprPy.tools.parseinput(f, allsingular=True)
38
39     # Interpret and process input parameters
40     process_input(input_dict, *args[1:])
41
42     # Run surface energy
43     results_dict = surface_energy(input_dict['lammps_command'],
44                                 input_dict['initialsystem'],
45                                 input_dict['potential'],
46                                 input_dict['symbols'],
47                                 mpi_command = input_dict['mpi_command'],
48                                 etol = input_dict['energytolerance'],
49                                 ftol = input_dict['forcetolerance'],
50                                 maxiter = input_dict['maxiterations'],
51                                 maxeval = input_dict['maxevaluations'],
52                                 dmax = input_dict['maxatommotion'],
53                                 cutboxvector = input_dict['surface_cutboxvector'])
54
55     # Save data model of results
56     results = iprPy.buildmodel(record_style, 'calc_' + calc_style, input_dict,
57                               results_dict)
58
59     with open('results.json', 'w') as f:
60         results.json(fp=f, indent=4)
61
62 def surface_energy(lammps_command, system, potential, symbols,
63                  mpi_command=None, etol=0.0, ftol=0.0, maxiter=10000,
64                  maxeval=100000, dmax=uc.set_in_units(0.01, 'angstrom'),
65                  cutboxvector='c'):
66     """
67     Evaluates surface formation energies by slicing along one periodic
68     boundary of a bulk system.
69
70     Parameters
71     -----
72     lammps_command : str
73         Command for running LAMMPS.
74     system : atomman.System
75         The system to perform the calculation on.
76     potential : atomman.lammps.Potential
77         The LAMMPS implemented potential to use.
78     symbols : list of str
79         Symbols of atoms in the system.
```

```
results.json
1 {
2     "calculation-surface-energy": {
3         "key": "0549a9ee-5ffb-4397-b90c-113671fefdc4",
4         "calculation": {
5             "iprPy-version": "0.7.2",
6             "LAMMPS-version": "26-Jan-2017-ICMS",
7             "script": "calc_surface_energy",
8             "run-parameter": {
9                 "size-multipliers": {
10                     "a": [
11                         5,
12                         5,
13                     ],
14                     "b": [
15                         5,
16                         5,
17                     ],
18                     "c": [
19                         10,
20                         10,
21                     ]
22                 },
23                 "energytolerance": 0.0,
24                 "forcetolerance": {
25                     "value": 1e-06,
26                     "unit": "eV/angstrom"
27                 },
28                 "maxiterations": 10000,
29                 "maxevaluations": 100000,
30                 "maxatommotion": {
31                     "value": 0.01,
32                     "unit": "angstrom"
33                 }
34             }
35         },
36         "potential-LAMMPS": {
37             "key": "8bfd0a48-8558-46f9-9d20-cd9e92cb83ae",
38             "id": "2017--Purja-Pun-G-P--Au--LAMMPS--test",
39             "potential": {
40                 "key": "ef908258-25d1-439e-8223-3bf4df924ed0",
41                 "id": "2017--Purja-Pun-G-P--Au"
42             }
43         },
44         "system-info": {
45             "family": "Al--Cu--fcc",
46             "artifact": {
47                 "file": "48fe09d1-0aec-4e8e-8638-a22fae5df0d9.xml",
48                 "format": "system model",
49             }
50         }
51     }
52 }
```



# Why Design Calculations This Way?

## Self-contained calculations

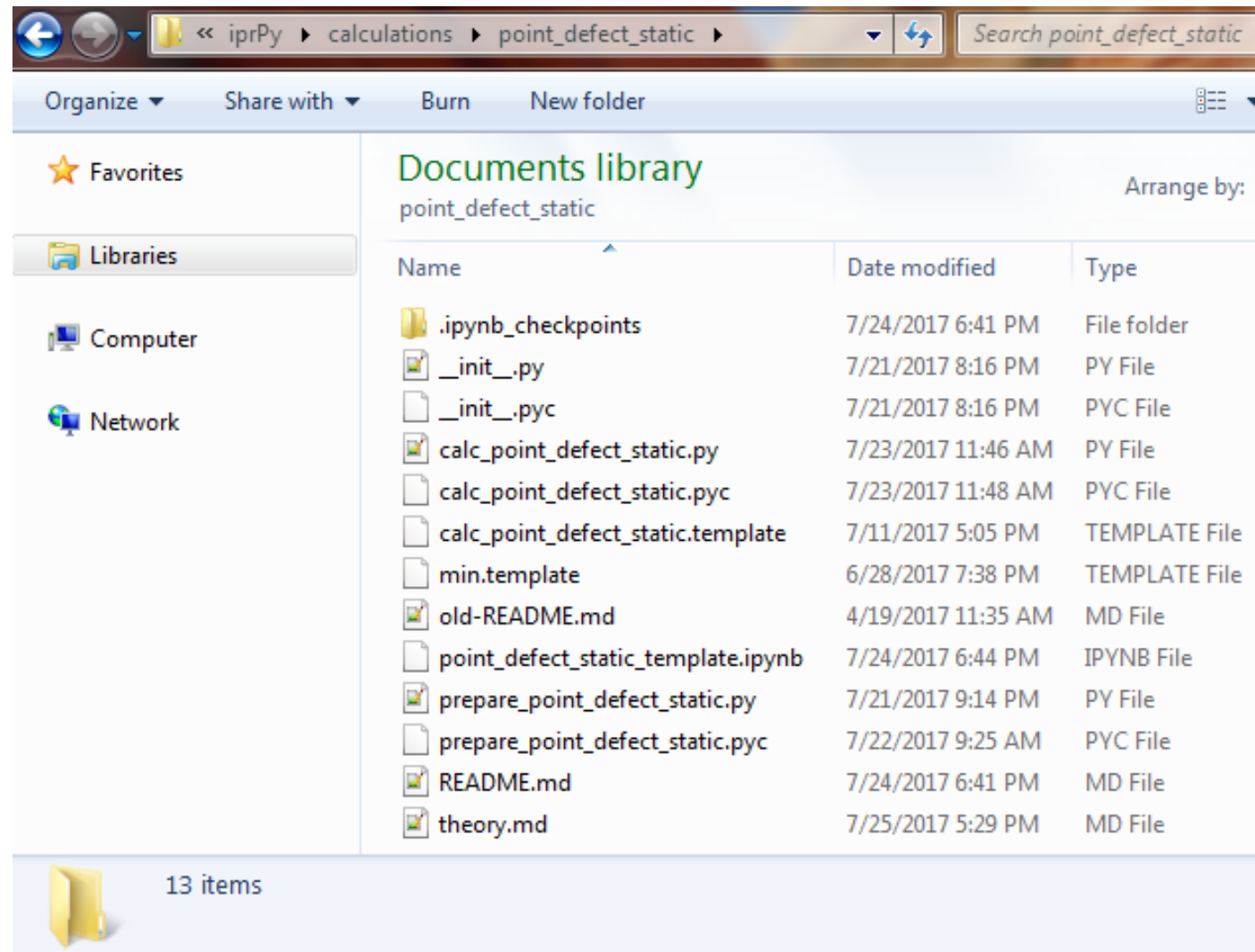
- Archive complete methodology
- Share capability with others
- Transfer knowledge to others

## Clear input parameters

- Focus on terms relevant to calculation
- Integrate into workflows
- Investigate parameter sensitivities

## Structured results

- Machine readable: store in database
- Human readable: understand contents



# Jupyter Demonstration Notebooks

## surface\_energy Calculation

Lucas M. Hale, [lucas.hale@nist.gov](mailto:lucas.hale@nist.gov), Materials Science and Engineering Division, NIST.

Chandler A. Becker, [chandler.becker@nist.gov](mailto:chandler.becker@nist.gov), Office of Data and Informatics, NIST.

Zachary T. Trautt, [zachary.trautt@nist.gov](mailto:zachary.trautt@nist.gov), Materials Measurement Science Division, NIST.

Version: 2017-07-24

[Disclaimers](#)

## Introduction

The `surface_energy` calculation evaluates the formation energy for a free surface by slicing an atomic system along a specific plane.

**Disclaimer #1:** Other atomic configurations at the free surface for certain planar cuts may have lower energies. The atomic relaxation will find a local minimum, which may not be the global minimum. Additionally, the material cut is planar perfect and therefore does not explore the effects of atomic roughness.

**Disclaimer #2:** Currently, the rotation capabilities of `atomman` limit this calculation such that only cubic prototypes can be rotated. Properties of non-cubic structures can still be explored, as long as the configuration being loaded has the plane of interest perpendicular to one of the three box vectors.

## Method and Theory

An initial system is supplied, and a LAMMPS simulation performs an energy/force minimization on the system and estimates the total potential energy of the perfect bulk system,  $E_{bulk}^{total}$ . A corresponding defect system is constructed by changing one of the three boundary conditions from periodic to non-periodic. This effectively slices the system along the boundary plane creating two free surfaces, each with surface area  $A_{surface}$ . The defect system is then relaxed with an energy/force minimization and the total potential energy of the defect system,  $E_{surface}^{total}$ , is measured. The formation energy of the free surface,  $E_{surface}^f$ , is computed in units of energy over area as

$$E_{surface}^f = \frac{E_{surface}^{total} - E_{bulk}^{total}}{2A_{surface}}$$

## 3.2 surface\_energy()

```
def surface_energy(lammps_command, system, potential, symbols,
                  mpi_command=None, etol=0.0, ftol=0.0, maxiter=10000,
                  maxeval=100000, dmax=uc.set_in_units(0.01, 'angstrom'),
                  cutboxvector='c'):
    """
    Evaluates surface formation energies by slicing along one periodic
    boundary of a bulk system.

    Parameters
    -----
    lammps_command : str
        Command for running LAMMPS.
    system : atomman.System
        The system to perform the calculation on.
    potential : atomman.lammps.Potential
        The LAMMPS implemented potential to use.
    symbols : list of str
        The list of element-model symbols for the Potential that correspond to
        system's atypes.
    mpi_command : str, optional
        The MPI command for running LAMMPS in parallel. If not given, LAMMPS
        will run serially.
    etol : float, optional
        The energy tolerance for the structure minimization. This value is
        unitless. (Default is 0.0).
    ftol : float, optional
        The force tolerance for the structure minimization. This value is in
        units of force. (Default is 0.0).
    maxiter : int, optional
        The maximum number of minimization iterations to use (default is
        10000).
    maxeval : int, optional
        The maximum number of minimization evaluations to use (default is
        100000).
    dmax : float, optional
        The maximum distance in length units that any atom is allowed to relax
        in any direction during a single minimization iteration (default is
        0.01 Angstroms).
    cutboxvector : str, optional
        Indicates which of the three system box vectors, 'a', 'b', or 'c', to
        cut with a non-periodic boundary (default is 'c').

    Returns
    -----
    dict
        Dictionary of results consisting of keys:

        - **'dumpfile_base'** (*str*) - The filename of the LAMMPS dump file
          of the relaxed bulk system.
        - **'dumpfile_surf'** (*str*) - The filename of the LAMMPS dump file
          of the relaxed system containing the free surfaces.
        - **'E_total_base'** (*float*) - The total potential energy of the
```

## Single file combining:

- Documentation
- Code identical to calculation script
- Working example

## 5. Report results

### 5.1 Define units for outputting values

- `length_unit` is the unit of length to display results in.
- `energy_unit` is the unit of energy to display cohesive energies in.
- `e_A_unit` is the energy per area to report the surface energy in.

```
length_unit = 'angstrom'
energy_unit = 'eV'

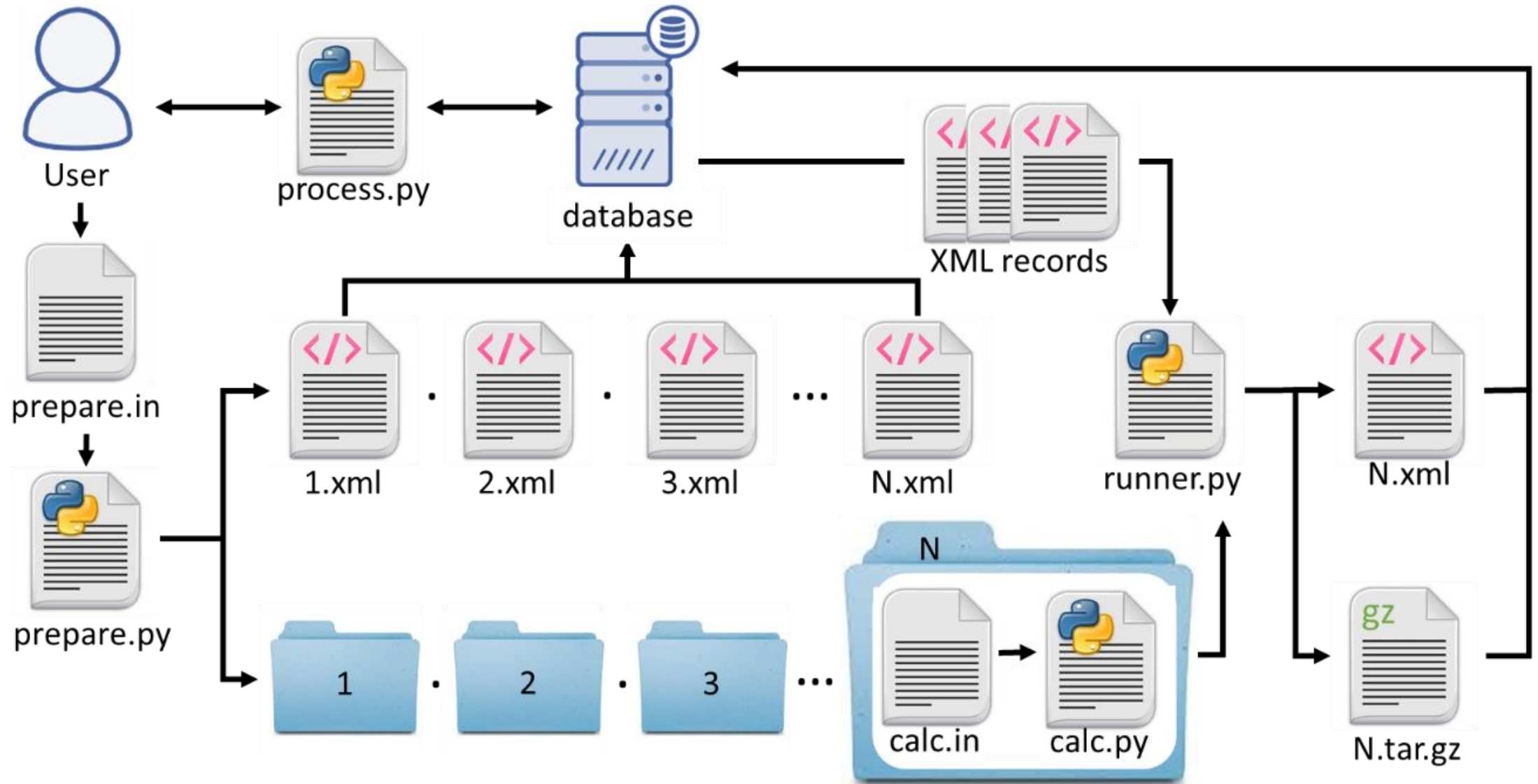
#e_A_unit = energy_unit+'/'+length_unit+'^2'
e_A_unit = 'mJ/m^2'
```

### 5.2 Print $E_{coh}$ , $A_{surface}$ , and $E_{surface}^f$

```
print('E_coh = ', uc.get_in_units(results_dict['E_coh'], energy_unit), energy_un
print('A_surface = ', uc.get_in_units(results_dict['A_surf'], length_unit+'^2'), 1
print('E_surface_f = ', uc.get_in_units(results_dict['E_surf_f'], e_A_unit), e_A_u
```

```
E_coh = -4.44999999835 eV
A_surface = 438.066793081 angstrom^2
E_surface_f = 2049.39393517 mJ/m^2
```

# High-Throughput Tools



# Implemented calculations

---

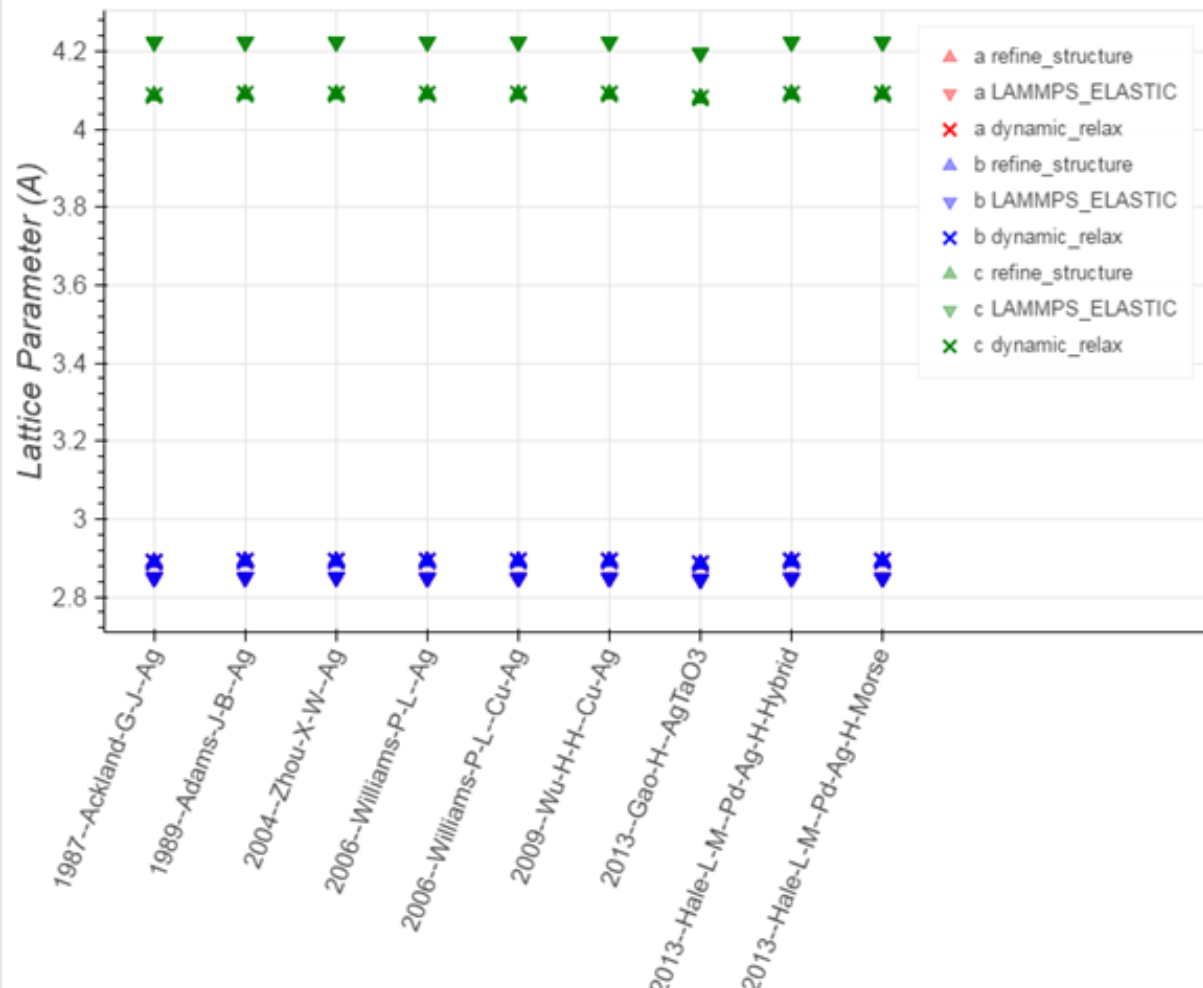
- Cohesive energy versus interatomic spacing
- Static relaxation of bulk systems and elastic constants (2 methods)
- Dynamic relaxation of bulk systems
- Free surface energies
- Generalized stacking faults (2 methods)
- Point defect formation energies
- Dislocation monopole generation

# Method Comparison – Lattice Constants

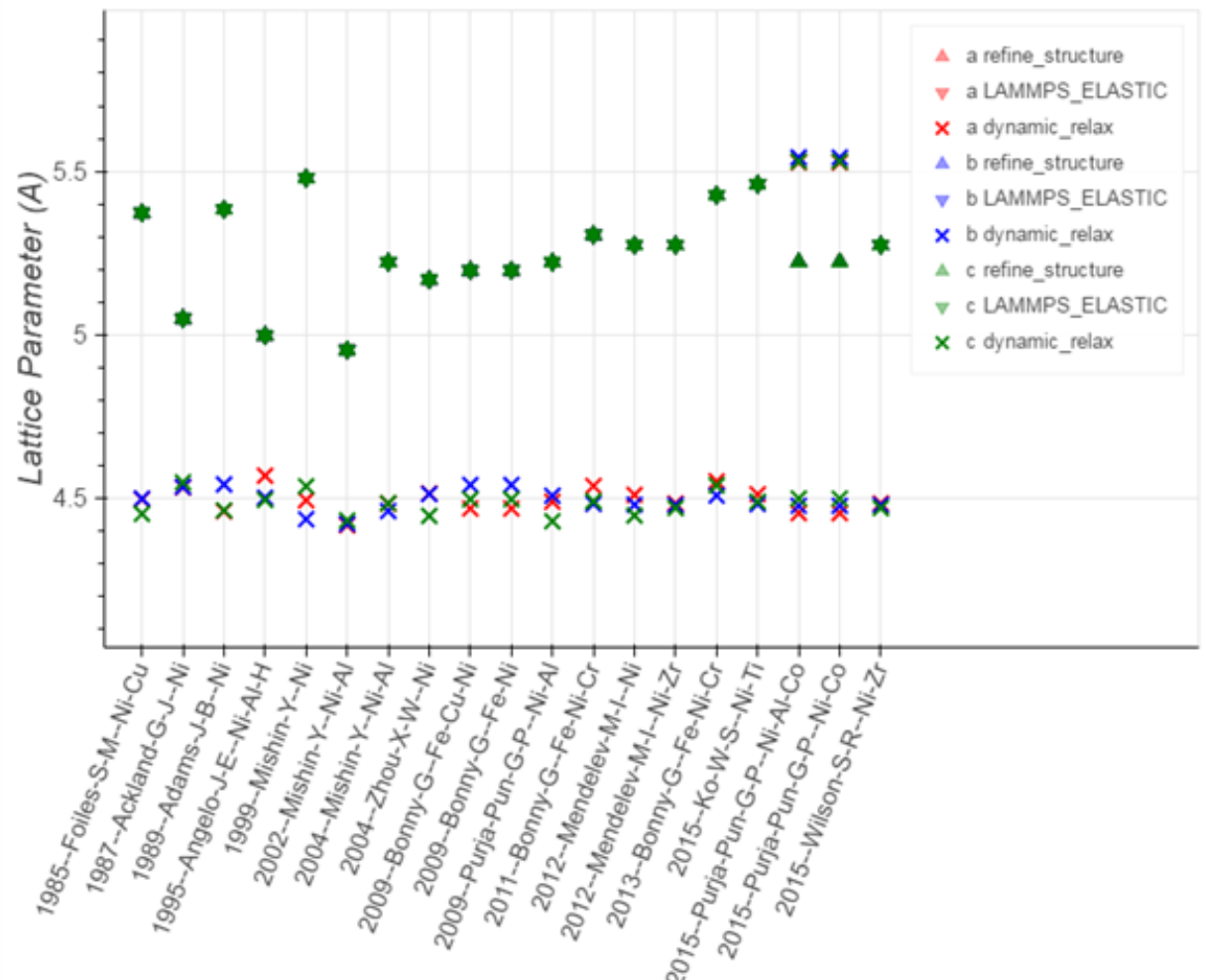
minimize + box\_relax doesn't fully  
relax bct  $\rightarrow$  fcc

Static methods predict diamond cubic  
nickel as stable

A6--In--bct Ag

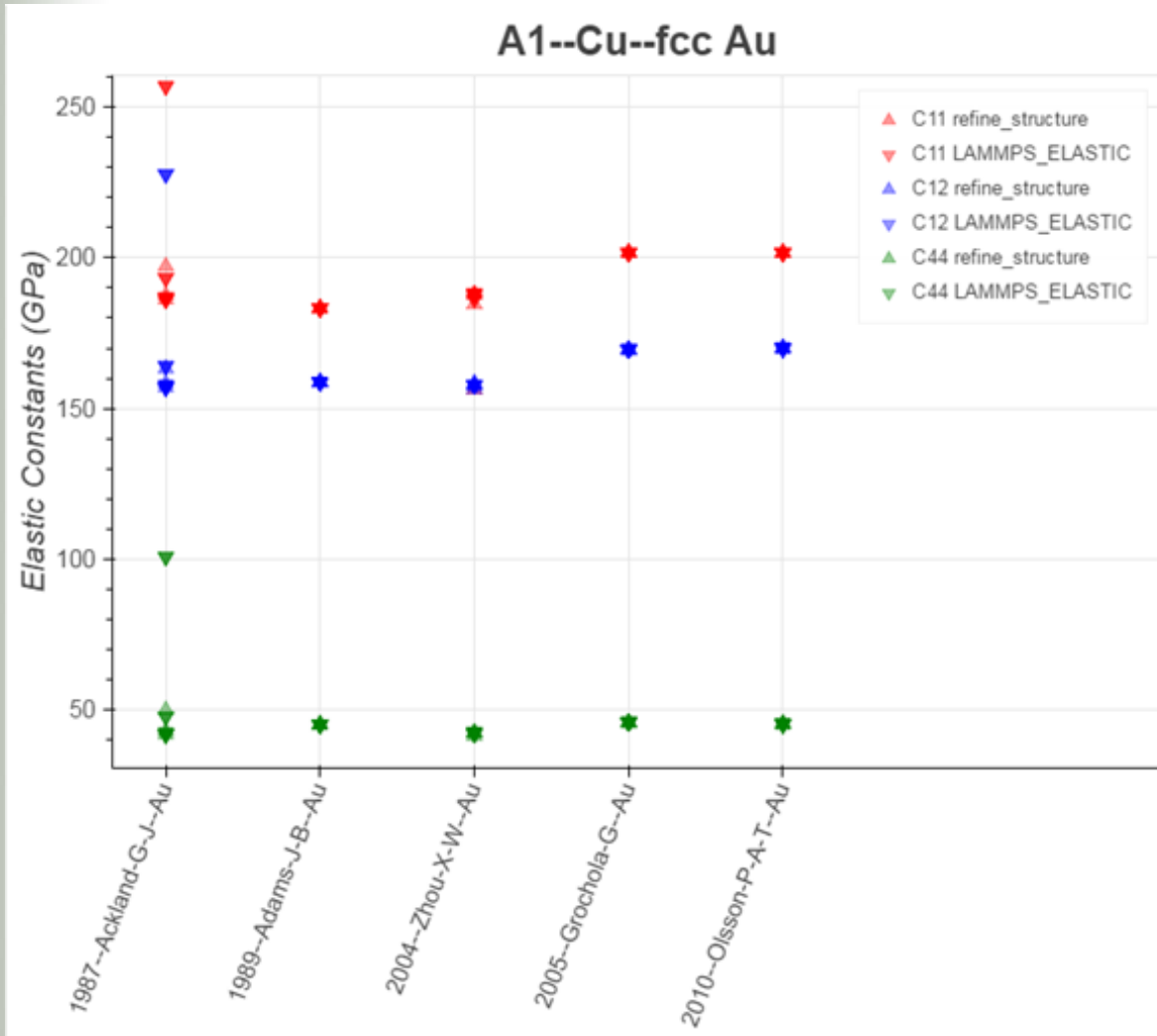


A4--C--dc Ni

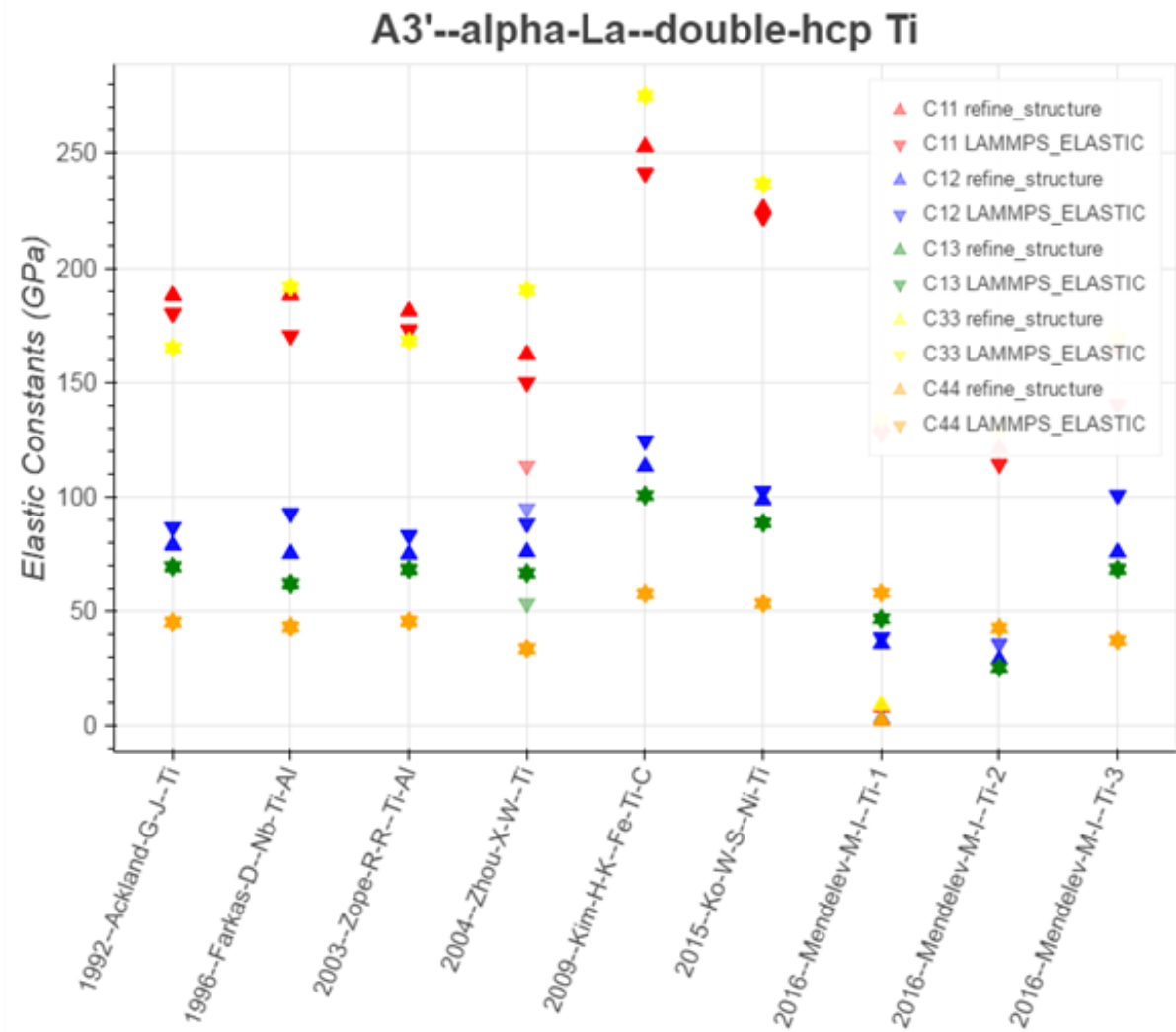


# Method Comparison – Elastic Constants

Sensitivity to strain range



Sensitivity to method



# Minimize Barriers for Usage

Only requirements are Python 2.7 and system-independent packages

Treat calculations and databases modularly

Define common inputs for similar calculations

Full documentation (finishing up now)

The screenshot shows the documentation for iprPy 0.7.2. The page title is 'Calculations'. The left sidebar contains a 'Table Of Contents' with the following items: 'Calculations' (expanded), 'Location of Calculations', 'Run a Calculation' (expanded), 'Run the Jupyter Notebook', 'Run the calculation script', and 'Prepare and run in high-throughput'. Below the sidebar are links for 'Previous topic', 'iprPy Tutorial', 'Next topic', 'Input Parameter File', 'This Page', 'Show Source', and a 'Quick search' box with a 'Go' button. The main content area has a blue header for 'Calculations' and a paragraph stating: 'The iprPy framework is designed around calculations. In design, all implemented calculations behave the same way:'. This is followed by a numbered list of four points: 1. Every calculation is an independent and self-contained unit of work. In other words, it can run in isolation from all other calculations. 2. Each calculation exists as or is executed by calling a Python script. 3. The calculation reads in all variable inputs from an input parameter file. 4. Upon successful completion, the calculation generates an XML- or JSON-formatted results record. Below this is another blue header for 'Location of Calculations' and a paragraph explaining that calculations are located in the iprPy/calculations directory. This is followed by another blue header for 'Run a Calculation' and a paragraph stating that a calculation can be performed in one of three ways, followed by a numbered list of three points: 1. Run the Jupyter Notebook, 2. Run the calculation script, and 3. Prepare and run in high-throughput.

iprPy 0.7.2 documentation » iprPy Tutorial » [previous](#) | [next](#) | [modules](#) | [index](#)

## Calculations

The iprPy framework is designed around calculations. In design, all implemented calculations behave the same way:

1. Every calculation is an independent and self-contained unit of work. In other words, it can run in isolation from all other calculations.
2. Each calculation exists as or is executed by calling a Python script.
3. The calculation reads in all variable inputs from an input parameter file.
4. Upon successful completion, the calculation generates an XML- or JSON-formatted results record.

## Location of Calculations

The calculations are located in the iprPy/calculations directory. This location lets the iprPy package identify the calculations and load them as a calculation style. Doing so makes it possible for objects of the `Calculation` class to access components and interact with the different calculation styles in a common way.

Within the iprPy/calculations directory, each calculation is placed in a subdirectory matching the calculation's name. Collecting all files for a given calculation together into a single folder allows for the calculations to be treated modularly.

## Run a Calculation

A calculation can be performed in one of three ways

1. Run the Jupyter Notebook
2. Run the calculation script
3. Prepare and run in high-throughput

# Other Resources

## Materials Resource Registry

<https://materials.registry.nist.gov/>

- Yellow pages for materials science
- All materials science, not just MD
- Search for resources
- FREE ADVERTISING!
- Add/edit your content by
  - Requesting an account or
  - Contact me

The screenshot displays the NIST Materials Resource Registry website. At the top, the NIST logo is on the left, and navigation links for Home, Services, Login, Help, and Contact are on the right. The main heading is "Materials Resource Registry". Below this is a search bar containing the text "LAMMPS". Underneath the search bar are seven icons representing different resource categories: All Resources (with a green checkmark), Organizations, Data Collections, Datasets, Services (with an API icon), Informational Sites, and Software. Below the icons, it says "Search criteria used (Clear all):" and "Type". On the right side, it indicates "5 results".

**TYPE** (Clear)

- Organization (0)
- Collection (0)
- Dataset (0)
- Service (0)
- Software (5)
- Web Site (0)

**ORIGIN OF DATA** (Clear)

**MATERIAL TYPE** (Clear)

**STRUCTURAL FEATURE** (Clear)

**PROPERTY ADDRESSED** (Clear)

**MPInterfaces**  
Richard Hennig - Hennig group, University of Florida  
<http://henniggroup.github.io/MPInterfaces/>  
Subject keyword(s): python, VASP, VASPsol, MP tools, 2D materials, hetero-structures, nanoparticles, ligands  
MPInterfaces is a python package that enables high throughput Density Functional Theory (DFT) analysis of arbitrary material interfaces (ligand capped nanoparticles, surfaces in the presence of solvents and hetero-structure interfaces) using VASP, VASPsol, LAMMPS, materialsproject database as well as their open source tools and a little bit of ASE.

**JARVIS-FF**  
Kamal Choudhary - National Institute of Standards and Technology (NIST)  
<http://www.ctcms.nist.gov/~knc6/periodic.html>  
Subject keyword(s): Force-field, Molecular dynamics, Repository, MPInterfaces, Materials project, Density functional theory  
The JARVIS-FF computes the energetics and mechanical properties of the materials based on available interatomic potentials/force-fields. The calculations are done using LAMMPS software with a generalized setup given in "in.elastic" script of LAMMPS distribution. The structures for computation were downloaded from density functional theory database-... [show more](#)

**Virtual Diffraction Characterization**  
Shawn P. Coleman  
<https://sites.google.com/site/shawnpatrickcoleman/research>  
Subject keyword(s): molecular dynamics, x-ray diffraction, selected area electron diffraction, XRD, SAED, MD, monte carlo, MC  
A newly proposed algorithm for generating virtual diffraction data was developed and introduced into the LAMMPS molecular dynamics simulator. The algorithm creates in-situ diffraction patterns from atomistic simulations utilizing a full Fourier analysis via explicit computation of the structure factor equation and does not require a priori knowledge... [show more](#)



# We Want to Work With You!

---

**Atomistic Simulations for Industrial Needs Workshop will return next summer!**

Interatomic Potential Repository: <https://www.ctcms.nist.gov/potentials/>

Download potentials

Add potentials: email [potentials@nist.gov](mailto:potentials@nist.gov)

iprPy Framework: <https://github.com/usnistgov/iprPy>

Download and use the calculations

Implement calculations into your workflows **just cite us!**

Add new calculations

Use design principles for research

Materials Resource Registry: <https://materials.registry.nist.gov/>

Explore resources

Add and advertise your work