OpenKIM Tutorial

Ellad B. Tadmor, Ryan S. Elliott

Department of Aerospace Engineering and Mechanics

University of Minnesota

Key Collaborators:

- Daniel S. Karls (U. Minnesota)
- Yaser Afshar (U. Minnesota)
- Noam Bernstein (NRL)
- George Karypis (U. Minnesota)
- Yonatan Kurniawan (BYU)
- Mark Transtrum (BYU)
- Mingian Wen (LBL)

NSF CDI (2009–2014); NSF CDS&E (2014–2018); NSF CMMT (2019–)

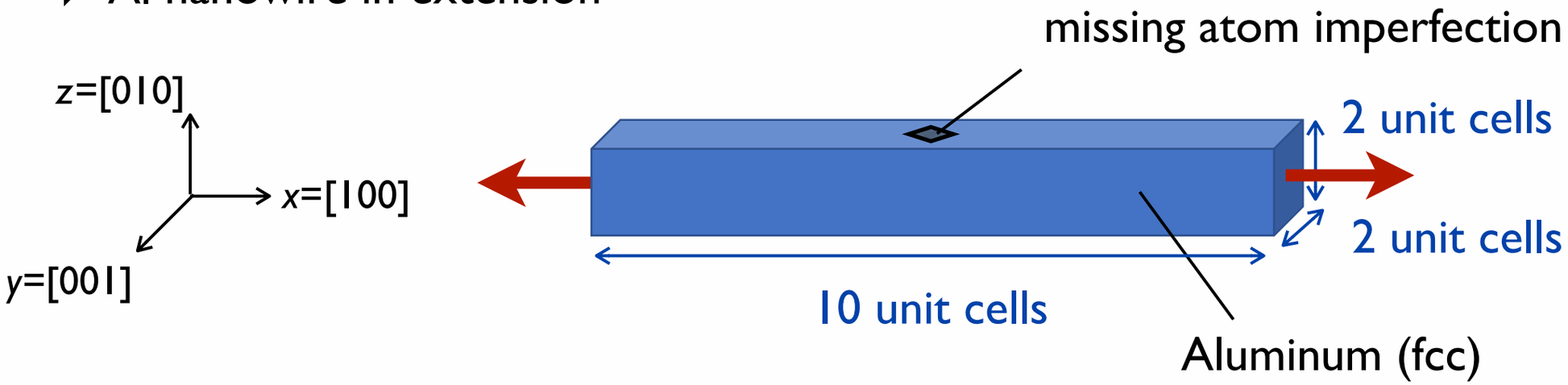
Outline | OpenKIM Tutorial

- LAMMPS Simulation: Nanowire Extension
- OpenKIM Framework and LAMMPS
- Potential Selection Exercise
- Obtaining Properties through OpenKIM Web Queries
- Unit Conversion Handling

- LAMMPS Simulation: Nanowire Extension
  - Setting up and running the simulation
  - Results
- OpenKIM Framework and LAMMPS
- Potential Selection Exercise
- Obtaining Properties through OpenKIM Web Queries
- Summary

Nanowire Extension Simulation

- Al nanowire in extension



  - Aluminum in the face-centered cubic (fcc) structure
  - Stress-strain curve obtained using the Embedded Atom Method (EAM) potential for Al due to Zhu et al., *Acta Materialia*, 49:4005, 2001.
  - Periodic boundary conditions along *x*-direction; free surfaces along *y* and *z*-directions
  - Imperfection introduced by removing one atom to localize deformation
  - Displacement boundary-conditions applied by stretching cell in *x*-direction
  - Stretching to 100% strain in increments of 1%

LAMMPS Script

```
# Aluminum nanowire --- static uniaxial tension along x-direction
#
# Based on an example by Mark Tschopp and Nathan Rhodes available at:
# https://icme.hpc.msstate.edu/mediawiki/index.php/LAMMPS_Fracture
#
# Authors: Daniel S. Karls, Ellad B. Tadmor
# Last revision: 8/7/2019

# Initialization
units metal
dimension 3
boundary p p p
atom_style atomic

# Simulation variables
variable latticetype string "fcc"      # Other options: "fcc", "bcc", "sc"`
variable a0 equal 4.05                 # Equilibrium lattice constant
variable wire_len_x equal 10           # Nanowire length in x-direction (in unit cells)
variable wire_len_y equal 2            # Nanowire length in y-direction (in unit cells)
variable wire_len_z equal 2            # Nanowire length in z-direction (in unit cells)
variable strain_increment equal 0.01   # Increment of strain relative to ref config
                                       # at each load step
variable number_load_steps equal 100   # Number of load steps

# Derived/other variables
variable vacuum_pad equal 5            # Set the vacuum along the y and z directions
variable box_min_y equal -${vacuum_pad}
variable box_max_y equal ${wire_len_y}+${vacuum_pad}
variable box_min_z equal -${vacuum_pad}
variable box_max_z equal ${wire_len_z}+${vacuum_pad}
variable box_yz_area equal (${box_max_y}-${box_min_y})*(${box_max_z}-${box_min_z})*${a0}^2
variable wire_yz_area equal ${wire_len_y}*${wire_len_z}*${a0}^2
```

LAMMPS Script

```
# Construct lattice
lattice ${latticetype} ${a0}
region supercell block 0 ${wire_len_x} &
                       ${box_min_y} ${box_max_y} &
                       ${box_min_z} ${box_max_z} units lattice
create_box 1 supercell
region nanowire block EDGE EDGE 0 ${wire_len_y} 0 ${wire_len_z} units lattice
create_atoms 1 region nanowire

# Create imperfection
variable atomx equal floor(${wire_len_x}/2)
variable atomy equal 0.0
variable atomz equal 0.0
variable atomrad equal 0.1
region void sphere ${atomx} ${atomy} ${atomz} ${atomrad}
delete_atoms region void

# Set mass to dummy value (not used in static minimization)
mass 1 1.0

# Specify potential
pair_style eam/alloy
pair_coeff * * Al_zhou.eam.alloy Al

# First, perform a static minimization with respect to all atomic positions
# under the condition that the box size relax along the axial (x) direction in
# order to achieve zero stress along this dimension.
thermo 10
thermo_style custom step lx ly lz press pxx pyy pzz pe
min_style cg
fix 1 all box/relax x 0.0 fixedpoint 0 0 0
minimize 1.0e-16 1.0e-16 5000 10000
unfix 1
```

LAMMPS Script

```
# Record stress
variable total_strain equal 0.0
variable pressf1 equal -pxx
variable pressf equal ${pressf1}*${box_yz_area}/${wire_yz_area}
print "STEP 0; strain: ${total_strain}; engineering stress (Bar): ${pressf}"
print "${total_strain} ${pressf}" file out_stress-strain_lmp.dat

# Dump the relaxed refenence configuration prior to straining
dump dmp1 all atom 1 out_config_lmp.dump
dump_modify dmp1 scale no
run 0
undump dmp1

#############################################################
# Now execute an incremental strain along the axial dimension
#############################################################

# Define looping variables
variable n loop ${number_load_steps}
label loop

  # Compute stretch factor relative to the deformed length needed to
  # obtain a strain increment of `strain_increment` relative to the
  # reference configuration (i.e. engineering or Lagrangian strain)
  variable stretch_factor equal &
           1.0+${strain_increment}/(1+${n}*${strain_increment})

  # Calculate total strain for this step
  variable total_strain equal ${n}*${strain_increment}

  # Stretch box
  change_box all x scale ${stretch_factor} remap
```

LAMMPS Script

```
# Minimize positions of atoms within cell
minimize 1.0e-10 1.0e-10 1000 10000

# Print out stress
variable pressf1 equal -pxx
variable pressf equal ${pressf1}*${box_yz_area}/${wire_yz_area}
print "STEP ${n}; strain: ${total_strain}; engineering stress (Bar): ${pressf}"
print "${total_strain} ${pressf}" append out_stress-strain_lmp.dat

# Dump relaxed strained configration
dump dmp1 all atom 1 out_config_lmp.dump
dump_modify dmp1 append yes scale no
run 0
undump dmp1

next n
jump SELF loop

print "Simulation complete"
```

EXERCISE: Run the Simulation

- **Goal:** Run the nanowire extension simulation

The LAMMPS input script is on your usb drive as: `in.kim.zhoueam.lmp.nanowire`

Execute the script on your machine:

```
$ cd /home/lammps/KIM-API
$ lmp_serial -in in.kim.zhoueam.lmp.nanowire > out.kim.zhoueam.lmp.nanowire
```

This will create the following files:

- `out_config_lmp.dump` (contains configurations at the end of each load step)
- `out_stress-strain_lmp.dat` (contains stress (bars) vs strain (percent) data)

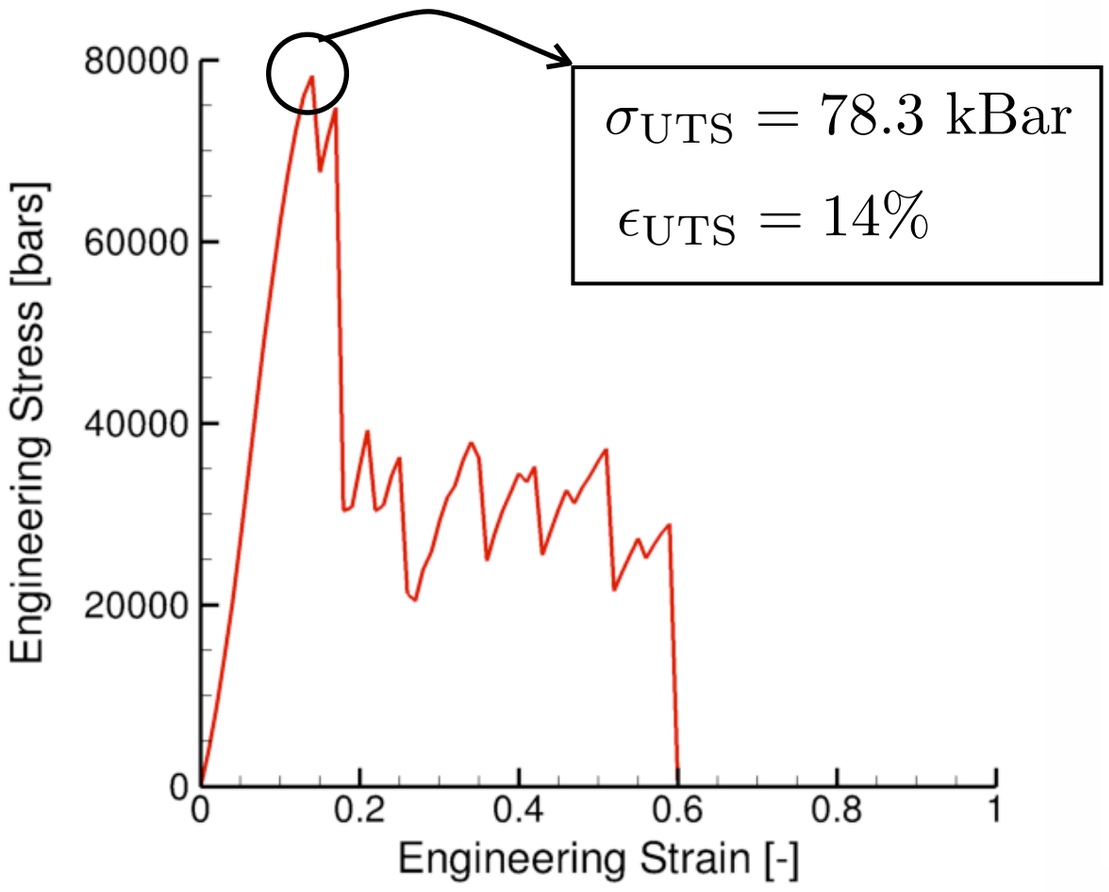
Plot the stress-strain curve using gnuplot, grace or matplotlib. For example:

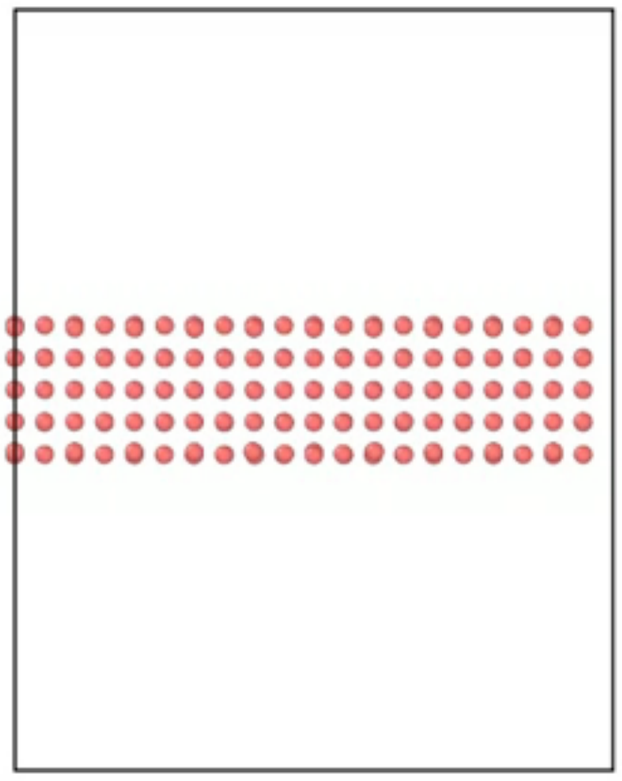
```
$ gnuplot
plot "out_stress-strain_lmp.dat" with lines
```

Raise your hand when you are done.

Results

- Results for the Zhou et al. (2001) EAM potential for Al





Visualization using Ovito

- How do we know if this is the "correct" potential for this simulation?

  How does the choice of potential affect the results?

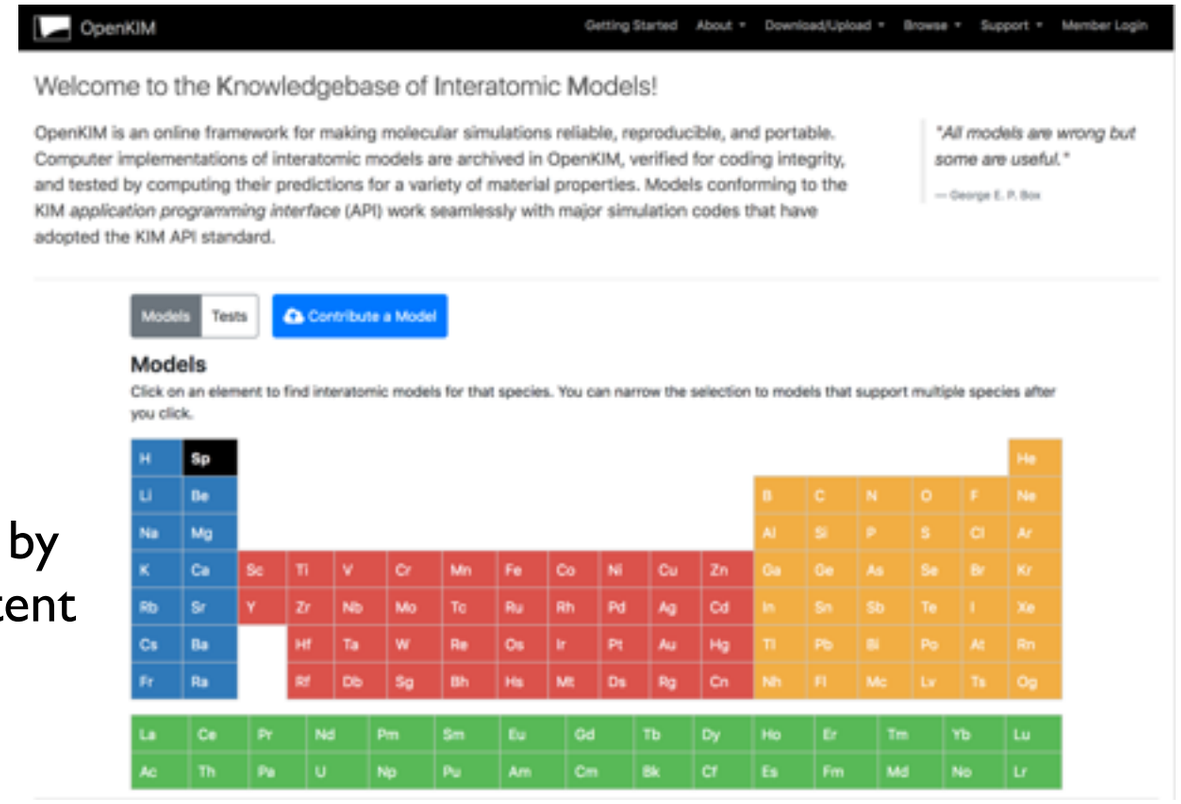
- LAMMPS Simulation: Nanowire Extension
  - Setting up and running the simulation
  - Results
- OpenKIM Framework and LAMMPS
  - What is OpenKIM?
  - Interatomic potentials on OpenKIM.org
  - Running LAMMPS with OpenKIM potentials
- Potential Selection Exercise
- Obtaining Properties through OpenKIM Web Queries
- Summary

Open Knowledegebase of Interatomic Models

OpenKIM by the numbers

https://openkim.org

- The OpenKIM repository currently contains:
  - 390 interatomic potentials
  - 2600 property tests
  - 9 verification checks
  - 75,000 reference data items
- OpenKIM has 560 members from 43 countries
- The OpenKIM website (openkim.org) is visited by 500-1000 researchers per month and KIM content is downloaded 10,000 per month
- The OpenKIM Directory of Model Developers (openkim.org/model-developer-directory/) includes 64 groups involved in interatomic potential development.



OpenKIM is funded by the NSF

OpenKIM is a member of DateCite

PIs: Ellad Tadmor, Ryan Elliott, George Karypis, Mark Transtrum

Advisory Board: Pietro Asinari, Laura Bartolo, Michael Baskes, Betsy Rice, Sadasivan Shankar, Aidan Thompson

Models on openkim.org

OpenKIM | Getting Started | About | Download/Upload | Browse | Support | tadmor | 1

Welcome to the **K**nowledgebase of **I**nteratomic **M**odels!

OpenKIM is an online framework for making molecular simulations reliable, reproducible, and portable. Computer implementations of interatomic models are archived in OpenKIM, verified for coding integrity, and tested by computing their predictions for a variety of material properties. Models conforming to the KIM *application programming interface* (API) work seamlessly with major simulation codes that have adopted the KIM API standard.

> *"All models are wrong but some are useful."*
> — George E. P. Box

Models | Tests | Contribute a Model or Data

Models

Click on an element to find interatomic models for that species. You can narrow the selection to models that support multiple species after you click.

H	Sp																He
Li	Be											B	C	N	O	F	Ne
Na	Mg											Al	Si	P	S	Cl	Ar
K	Ca	Sc	Ti	V	Cr	Mn	Fe	Co	Ni	Cu	Zn	Ga	Ge	As	Se	Br	Kr
Rb	Sr	Y	Zr	Nb	Mo	Tc	Ru	Rh	Pd	Ag	Cd	In	Sn	Sb	Te	I	Xe
Cs	Ba		Hf	Ta	W	Re	Os	Ir	Pt	Au	Hg	Tl	Pb	Bi	Po	At	Rn
Fr	Ra		Rf	Db	Sg	Bh	Hs	Mt	Ds	Rg	Cn	Nh	Fl	Mc	Lv	Ts	Og

La	Ce	Pr	Nd	Pm	Sm	Eu	Gd	Tb	Dy	Ho	Er	Tm	Yb	Lu
Ac	Th	Pa	U	Np	Pu	Am	Cm	Bk	Cf	Es	Fm	Md	No	Lr

Models on openkim.org

Cu

Extended KIM ID	Title
EAM_Dynamo_AcklandTichyVitek_1987_Cu__MO_179025990738_005	Finnis-Sinclair potential (LAMMPS cubic hermite tabulation) for Cu developed by Ackland et al. (1987) v005
EAM_Dynamo_AcklandTichyVitek_1987v2_Cu__MO_762798677854_000	Finnis-Sinclair potential (LAMMPS cubic hermite tabulation) for Cu developed by Ackland et al. (1987), version 2 refitted for radiation studies v000
EAM_Dynamo_AcklandVitek_1990_Cu__MO_642748370624_000	Finnis-Sinclair potential (LAMMPS cubic hermite tabulation) for Cu developed by Ackland and Vitek (1990) v000
EAM_Dynamo_AdamsFoiles_1989Universal6_Cu__MO_145873824897_000	EAM potential (LAMMPS cubic hermite tabulation) for Cu (Universal6) developed by Adams and Foiles (1989) v000

-
-
-
-

EAM_Dynamo_ZhouWadleyJohnson_2001_CuTa__MO_547744193826_000	EAM potential (LAMMPS cubic hermite tabulation) for the Cu-Ta system developed by Zhou, Wadley and Johnson (2001) v000
EAM_NN_Johnson_1988_Cu__MO_887933271505_002	EAM Potential (analytical nearest-neighbor) for Cu developed by Johnson (1988) v002
EMT_Asap_MetalGlass_BaileySchiotzJacobsen_2004_CuMg__MO_228059236215_001	EMT potential for Cu-Mg metallic glasses developed by Bailey, Schiotz, and Jacobsen (2004) v000
EMT_Asap_MetalGlass_CuMgZr__MO_655725647552_002	Effective Medium Theory potential for CuMg and CuZr alloys, in particular metallic glasses.
EMT_Asap_MetalGlass_PaduraruKenoufiBailey_2007_CuZr__MO_987541074959_001	EMT potential for Cu-Zr metallic glasses developed by Paduraru et al. (2007) v000
EMT_Asap_Standard_JacobsenStoltzeNorskov_1996_AlAgAuCuNiPdPt__MO_115316750986_001	EMT potential for Al, Ni, Cu, Pd, Ag, Pt and Au developed by Jacobsen, Stoltze, and Norskov (1996) v000
EMT_Asap_Standard_JacobsenStoltzeNorskov_1996_Cu__MO_396616545191_001	EMT potential for Cu developed by Jacobsen, Stoltze, and Norskov (1996) v000
EMT_Asap_Standard_Jacobsen_Stoltze_Norskov_AlAgAuCuNiPdPt__MO_118428466217_002	Standard Effective Medium Theory potential for face-centered cubic metals as implemented in ASE/Asap.
LJ_ElliottAkerson_2015_Universal__MO_959249795837_003	Efficient 'universal' shifted Lennard-Jones model for all KIM API supported species developed by Elliott and Akerson (2015) v003
MEAM_2NN_Fe_to_Ga__MO_145522277939_001	Model parameterization of 2NN MEAM model
Morse_Shifted_GirifalcoWeizer_1959HighCutoff_Cu__MO_151002396060_002	Morse potential (shifted) for Cu by Girifalco and Weizer (1959) using a high-accuracy cutoff distance v002
Morse_Shifted_GirifalcoWeizer_1959LowCutoff_Cu__MO_673777079812_002	Morse potential (shifted) for Cu by Girifalco and Weizer (1959) using a low-accuracy cutoff distance v002
Morse_Shifted_GirifalcoWeizer_1959MedCutoff_Cu__MO_173787283511_002	Morse potential (shifted) for Cu by Girifalco and Weizer (1959) using a medium-accuracy cutoff distance v002
Pair_Morse_Modified_MacDonaldMacDonald_Cu__MO_034823476734_000	Modified Morse pair potential for copper due to MacDonald and MacDonald

Models on openkim.org

EAM_NN_Johnson_1988_Cu__MO_887933271505_002

Title	EAM Potential (analytical nearest-neighbor) for Cu developed by Johnson (1988) v002
Description	Analytical nearest-neighbor EAM model for Cu by Johnson
Species	Cu
Contributor	Ryan
Maintainer	Ryan
Author	Ryan S. Elliott
Publication Year	2018
Item Citation	This Model originally published in [1] is archived in OpenKIM [2-4]. [1] Johnson RA. Analytic nearest-neighbor model for fcc metals. Physical Review B. 1988Mar;37(8):3924–31. doi:10.1103/PhysRevB.37.3924 [2] Elliott RS. EAM Potential (analytical nearest-neighbor) for Cu developed by Johnson (1988) v002. OpenKIM; 2018. doi:10.25950/3ccd9f3b [3] Tadmor EB, Elliott RS, Sethna JP, Miller RE, Becker CA. The potential of atomistic simulations and the Knowledgebase of Interatomic Models. JOM. 2011;63(7):17. doi:10.1007/s11837-011-0102-6 [4] Elliott RS, Tadmor EB. Knowledgebase of Interatomic Models (KIM) Application Programming Interface (API). OpenKIM; 2011. doi:10.25950/ff8f563a Click here to download the above citation in BibTeX format.
Short KIM ID	MO_887933271505_002
Extended KIM ID	EAM_NN_Johnson_1988_Cu__MO_887933271505_002
DOI	10.25950/3ccd9f3b https://doi.org/10.25950/3ccd9f3b https://search.datacite.org/works/10.25950/3ccd9f3b
KIM Item Type	Portable Model
KIM API Version	2.0
Programming Language(s)	100.00% C
Previous Version	EAM_NN_Johnson_1988_Cu__MO_887933271505_001

Models on openkim.org

Further down the model page for
EAM_NN_Johnson_1988_Cu__MO_887933271505_002

Verification Check Dashboard

(Click here to learn more about Verification Checks)

Grade	Name	Category	Brief Description	Full Results	Aux File(s)
P	**vc-species-supported-as-stated**	mandatory	The model supports all species it claims to support; see full description.	Results	Files
P	**vc-periodicity-support**	mandatory	Periodic boundary conditions are handled correctly; see full description.	Results	Files
P	**vc-permutation-symmetry**	mandatory	Total energy and forces are unchanged when swapping atoms of the same species; see full description.	Results	Files
A	**vc-forces-numerical-derivative**	consistency	Forces computed by the model agree with numerical derivatives of the energy; see full description.	Results	Files
F	**vc-dimer-continuity-c1**	informational	The energy versus separation relation of a pair of atoms is C1 continuous (i.e. the function and its first derivative are continuous); see full description.	Results	Files
P	**vc-objectivity**	informational	Total energy is unchanged and forces transform correctly under rigid-body translation and rotation; see full description.	Results	Files
P	**vc-inversion-symmetry**	informational	Total energy is unchanged and forces change sign when inverting a configuration through the origin; see full description.	Results	Files
P	**vc-memory-leak**	informational	The model code does not have memory leaks (i.e. it releases all allocated memory at the end); see full description.	Results	Files
P	**vc-thread-safe**	mandatory	The model returns the same energy and forces when computed in serial and when using parallel threads for a set of configurations. Note that this is not a guarantee of thread safety; see full description.	Results	Files

Models on openkim.org

- Further down the model page for
  EAM_NN_Johnson_1988_Cu__MO_887933271505_002

Cubic Crystal Basic Properties Table

Species: Cu

	Model	Lattice Constant [Å]	Cohesive Energy [eV]	c11 [GPa]	c12 [GPa]	c44 [GPa]
bcc Expand	EAM_NN_Johnson_1988_Cu__MO_887933271505_002	2.8593961074900003	3.6063831577047547	146.26088738200002	137.952181442	91.9367817649002
diamond Expand	EAM_NN_Johnson_1988_Cu__MO_887933271505_002	5.450421601530001	2.4241832490740016	N/A	N/A	N/A
fcc Expand	EAM_NN_Johnson_1988_Cu__MO_887933271505_002	3.61472985148	3.5400001233123688	184.172808464	115.324864335	68.8519693905
sc Collapse	EAM_NN_Johnson_1988_Cu__MO_887933271505_002	2.3724498152700004	3.263473577969841	270.847253148	24.4996165814	-17.5854303931
	EAM_Dynamo_AcklandTichyVitek_1987_Cu__MO_179025990738_005	2.41274794936	2.934987746337366	296.901843605	90.83341720560001	54.1961445689
	EAM_Dynamo_Ackl	2.39571615309	2.94757703382590	186.096823759000	61.7775700379000	24.2576222484000

Models on openkim.org

Further down the model page for
EAM_NN_Johnson_1988_Cu__MO_887933271505_002

Tests

ElasticConstantsCubic__TD_011862047401_004

Computes the cubic elastic constants for some common crystal types (fcc, bcc, sc) by calculating the hessian of the energy density with respect to strain. An estimate of the error associated with the numerical differentiation performed is reported.

Test	Test Results	Link to Test Results page	Benchmark time
ElasticConstantsCubic_bcc_Cu__TE_091603841600_004	expand	view	2602
ElasticConstantsCubic_fcc_Cu__TE_188557531340_004	expand	view	3665
ElasticConstantsCubic_sc_Cu__TE_319353354686_004	expand	view	3079

Full results page.

Expand a property synopsis.

Usertime muliplied by the Whetstone Benchmark. This number can be used (approximately) to compare the performance of different models independently of the architecture on which the test was run.

Models on openkim.org

Further down the model page for
EAM_NN_Johnson_1988_Cu__MO_887933271505_002

Tests

ElasticConstantsCubic__TD_011862047401_004

Computes the cubic elastic constants for some common crystal types (fcc, bcc, sc) by calculating the hessian of the energy density with respect to strain. An estimate of the error associated with the numerical differentiation performed is reported.

Test	Test Results	Link to Test Results page	Benchmark time
ElasticConstantsCubic_bcc_Cu__TE_091603841600_004	expand	view	2602
ElasticConstantsCubic_fcc_Cu__TE_188557531340_004	expand	view	3665
ElasticConstantsCubic_sc_Cu__TE_319353354686_004	collapse	view	3079

instance-id: 1
Isothermal elastic constants for a cubic crystal at constant temperature and stress
(For more information, see the property definition elastic-constants-isothermal-cubic-crystal-npt)

Crystal type = ["sc"]

a = 2.37244981527 angstrom
Species = ["Cu"]
Basis atom coordinates = [[0.0 0.0 0.0]]

Temperature = 0 K
Cauchy stress = [0 0 0 0 0 0] GPa

c11 = 270.8472531475441 GPa
c12 = 24.49961658139509 GPa
c44 = -17.585430393063543 GPa

Elastic constants (note that c44 is negative indicating the sc structure is unstable).

Models on openkim.org

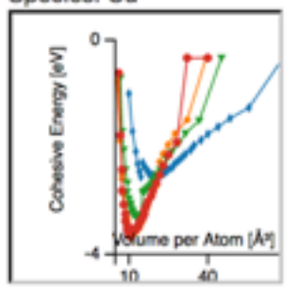
Further down the model page for
EAM_NN_Johnson_1988_Cu__MO_887933271505_002

Visualizers (in-page)

Cohesive Energy Graph

This graph shows the cohesive energy versus volume-per-atom for the current mode for four mono-atomic cubic phases (body-centered cubic (bcc), face-centered cubic (fcc), simple cubic (sc), and diamond). The curve with the lowest minimum is the ground state of the crystal if stable. (The crystal structure is enforced in these calculations, so the phase may not be stable.) Graphs are generated for each species supported by the model.

Species: Cu

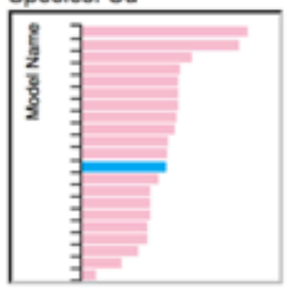


Click on any thumbnail to get a full size image.

Diamond Lattice Constant

This bar chart plot shows the mono-atomic face-centered diamond lattice constant predicted by the current model (shown in the unique color) compared with the predictions for all other models in the OpenKIM Repository that support the species. The vertical bars show the average and standard deviation (one sigma) bounds for all model predictions. Graphs are generated for each species supported by the model.

Species: Cu



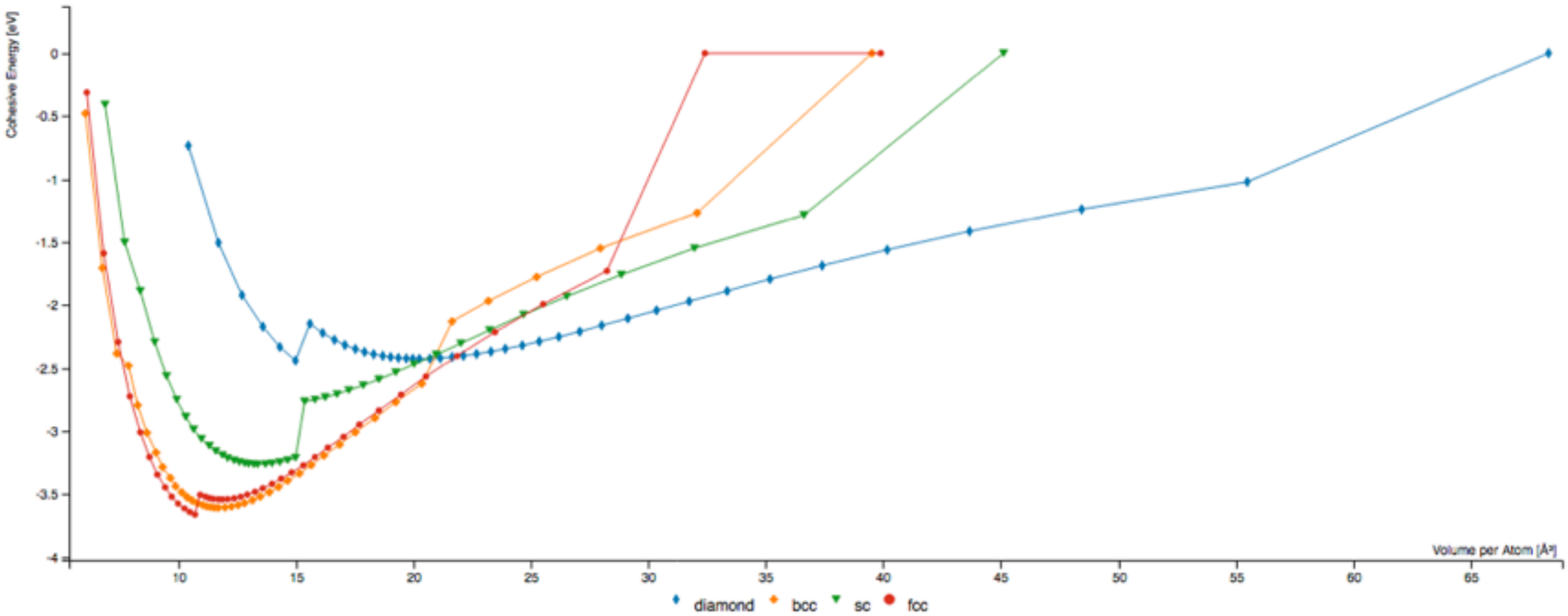
Click on any thumbnail to get a full size image.

Models on openkim.org

OpenKIM

Model: EAM_NN_Johnson_1988_Cu__MO_887933271505_002
Species: Cu

This graph shows the cohesive energy versus volume-per-atom for the current model for four mono-atomic cubic phases (body-centered cubic (bcc), face-centered cubic (fcc), simple cubic (sc), and diamond). The curve with the lowest minimum is the ground state of the crystal if stable. (The crystal structure is enforced in these calculations, so the phase may not be stable.) The curves below are for the species specified above.



Models on openkim.org

- Further down the model page for EAM_NN_Johnson_1988_Cu__MO_887933271505_002

° **Wiki**

Description

This **Model** implements the potential developed by R.A. Johnson for fcc metals as described in the reference above (see Source Citations). In particular, this model is applied to copper (Cu).

Parameters

Symbols (matching the reference):

$$r_e, \phi_e, \gamma, f_e, \beta, E_c, \alpha, \rho_e.$$

Corresponding variables in code:
JEAM_R0, JEAM_PHI0, JEAM_GAM, JEAM_G0, JEAM_BET, JEAM_EC, JEAM_ALF, JEAM_RHO0, where the prefix JEAM emphasizes the fact that each variable corresponds to the "Johnson Embedded Atom Potential".

Warning: The model uses other parameters DIM, SPECCODE and MODEL_CUTOFF denoting the dimensionality of the space (3 by default), the number of species (1, by default) and the cut-off radius (3.5 Angstrom by default), respectively. Default values have been hardcoded and, in principle, they should not be modified.

Details

The total potential energy of a system of N atoms is assumed to take the form $E = \sum_{i=1}^{N} E_i$, such that

$$E_i = \sum_{i=1}^{N} \left[F(\rho_i) + \frac{1}{2} \sum_{j=1}^{m} \phi(r_{ij}) \right],$$

and

$$\rho_i = \sum_{j=1}^{m} f(r_{ij}),$$

where E_i denotes the energy per atom i, $F(\rho_i)$ is the embedding function contribution, $\frac{1}{2} \sum_{j=1}^{m} \phi(r_{ij})$ is the two-body contribution to the energy, ρ_i stands for the electron density at atom i, and $f(r_{ij})$ is the atomic electron density of atom j as a function of the distance from its center r_{ij}, while j is one of the m neighbors of the atom i.

...

Running LAMMPS with KIM Models

- Only two changes are needed to the LAMMPS script to run with a KIM model:

1. Replace the `units` command with a `kim_init` command:

```
# Initialization
units metal
```

Selects the KIM model and sets the units

```
# Initialization
kim_init EAM_Dynamo_ZhouWadleyJohnson_2001_Al__MO_049243498555_000 metal
```

2. Replace the `pair_style` and `pair_coeff` commands with a `kim_interactions` command:

```
# Specify potential
pair_style eam/alloy
pair_coeff * * Al_zhou.eam.alloy Al
```

Define the atom style to species mapping

```
# Initialization
kim_interactions Al
```

Outline | OpenKIM Tutorial

- LAMMPS Simulation: Nanowire Extension
  - Setting up and running the simulation
  - Results
- OpenKIM Framework and LAMMPS
  - What is OpenKIM?
  - Interatomic potentials on OpenKIM.org
  - Running LAMMPS with OpenKIM potentials
- Potential Selection Exercise
  - Select a suitable interatomic potential from OpenKIM.org
  - Rerun simulation with new potential
- Obtaining Properties through OpenKIM Web Queries
- Summary

EXERCISE: Potential Selection using OpenKIM

- **Goal:** Select a suitable interatomic potential

Material properties that are important for the potential to get right for a tensile test:

Property	Symbol [units]	DFT value (0 K)
Lattice constant	a_0 [Å]	4.049 [1] Vitos et al., *Surf. Sci.*,
Elastic constants	C_{11} [GPa]	108.2 [2] Bercegeay et al., *Phys. Rev.*
	C_{12} [GPa]	56.6 [2]
	C_{44} [GPa]	30.5 [2]
Surface energies	γ_{100} [eV/Å^2]	0.0748 [1]
	γ_{110} [eV/Å^2]	0.0793 [1]
	γ_{111} [eV/Å^2]	0.0841 [1]
Unstable stacking energy	γ_{us} [eV/Å^2]	0.010 [3] Kibey et al., *Acta Mater..*,

Look through the Al potentials on openkim.org, select one that you think is best.

Raise your hand when you are done.

Comparison of Potentials - Errors in %

SLIDE REMOVED

EXERCISE: Potential Selection using OpenKIM

- **Goal:** Run the nanowire extension simulation with the selected potential

Step 1: Copy the input script to a new file and save Zhou plot files:

```
$ cp in.kim.zhoueam.lmp.nanowire in.kim
$ mv out_stress-strain_lmp.dat out_stress-strain_lmp_zhou.dat
$ mv out_config_lmp.dump out_config_lmp_zhou.dump
```

Step 2: Edit the new script to use the KIM model you selected.

```
$ nano in.kim
```

Step 3: Run simulation

```
$ lmp_serial -in in.kim > out.kim
```

Step 4: Plot results, comparing the new potential with Zhou EAM. In gnuplot:

```
plot “out_stress-strain_lmp.dat” with lines, “out_stress-strain_lmp_zhou.dat”
with lines
```

Raise your hand when you are done.

Results with Other Potentials

SLIDE REMOVED

Citation Information for KIM Model

When running LAMMPS with a KIM model, citation information in BibTex format is contained in the `log.cite` file:

```
This LAMMPS simulation made specific use of work described in the following references.  See http://
lammps.sandia.gov/cite.html for details.

@Comment
{
\documentclass{article}
\usepackage{url}
\begin{document}
This Model originally published in \cite{MO_049243498555_000a, MO_049243498555_000b} is archived in
OpenKIM~\cite{MO_049243498555_000, MD_120291908751_005, tadmor:elliott:2011, elliott:tadmor:2011}.
\bibliographystyle{vancouver}
\bibliography{kimcite-MO_049243498555_000.bib}
\end{document}
}

@Article{MO_049243498555_000a,
  author = {Zhou, X.W. and Wadley, H.N.G. and Johnson, R.A. and Larson, D.J. and Tabat, N. and Cerezo, A.
and Petford-Long, A.K. and Smith, G.D.W. and Clifton, P.H. and Martens, R.L. and Kelly, T.F.},
  doi = {10.1016/S1359-6454(01)00287-7},
  journal = {Acta Materialia},
  number = {19},
  pages = {4005 - 4015},
  title = {Atomic scale structure of sputtered metal multilayers},
  volume = {49},
  year = {2001},
}

...
```

Citation Information for KIM Model

When running LAMMPS with a KIM model, citation information in BibTeX format is contained in an auto-generated `log.cite` file:

```
.
.
.
@Misc{MO_049243498555_000,
  author       = {Ellad Tadmor},
  title        = {{EAM} potential ({LAMMPS} cubic hermite tabulation) for {A}l developed by {Z}hou, {W}
adley and {J}ohnson (2001) v000},
  doi          = {10.25950/84f935c7},
  howpublished = {OpenKIM, \url{https://doi.org/10.25950/84f935c7}},
  keywords     = {OpenKIM, Model, MO_049243498555_000},
  publisher    = {OpenKIM},
  year         = 2018,
}

@Misc{MD_120291908751_005,
  author       = {Ryan S. Elliott},
  title        = {{EAM} {M}odel {D}river for tabulated potentials with cubic {H}ermite spline interpolation
as used in {LAMMPS} v005},
  doi          = {10.25950/68defa36},
  howpublished = {OpenKIM, \url{https://doi.org/10.25950/84f935c7}},
  keywords     = {OpenKIM, Model Driver, MD_120291908751_005},
  publisher    = {OpenKIM},
  year         = 2018,
}
.
.
.
```

Outline | OpenKIM Tutorial

- LAMMPS Simulation: Nanowire Extension
  - Setting up and running the simulation
  - Results
- OpenKIM Framework and LAMMPS
  - What is OpenKIM?
  - Interatomic potentials on OpenKIM.org
  - Running LAMMPS with OpenKIM potentials
- Potential Selection Exercise
  - Select a suitable interatomic potential from OpenKIM.org
  - Rerun simulation with new potential
- Obtaining Properties through OpenKIM Web Queries
  - LAMMPS kim_query mechanism
  - Query exercise
- Summary

Obtaining Properties through OpenKIM Web Queries

- In the simulation that you ran, a hard-wired lattice constant was used:

```
...
# Simulation variables
variable latticetype string "fcc"      # Other options: "fcc", "bcc", "sc"`
variable a0 equal 4.05                 # Equilibrium lattice constant
...
lattice ${latticetype} ${a0}
...
```

But actually each interatomic potential predicts a different equilibrium lattice constant:

EAM_CubicNaturalSpline_ErcolessiAdams_1994_Al__MO_800509458712_002	4.032
EAM_Dynamo_AngeloMoodyBaskes_1995_NiAlH__MO_418978237058_005	4.05
EAM_Dynamo_CaiYe_1996_AlCu__MO_942551040047_005	4.05
EAM_Dynamo_ErcolessiAdams_1994_Al__MO_123629422045_005	4.032
EAM_Dynamo_FarkasJones_1996_NbTiAl__MO_042691367780_000	3.869
EAM_Dynamo_JacobsenNorskovPuska_1987_Al__MO_411692133366_000	3.988
...	...

You can do a simulation to determine *a0*, look it up and substitute into script, or ...

Obtaining Properties through OpenKIM Web Queries

- Using OpenKIM queries within a LAMMPS script:

```
kim_init EAM_Dynamo_ZhouWadleyJohnson_2001_Al__MO_049243498555_000 metal
...
# Simulation variables
variable latticetype string "fcc"      # Other options: "fcc", "bcc", "sc"`
kim_query
kim_query a0 get_lattice_constant_cubic crystal=["fcc"] species=["Al"] units=["angstrom"]
...
lattice ${latticetype} ${a0}
...
```

The `kim_query` command performs a web query to openkim.org and retrieves

- For the model specified in `kim_init` (`EAM_Dynami_ZhouWadley...`)
- The requested query (`get_lattice_constant_cubic`) with parameters
  - `crystal = [“fcc”]`
  - `species = [“Al”]`
  - `units = [“angstrom”]`
- Result placed in the variable `a0`

For the Zhou et al. model this will result in an fcc lattice with `a0=4.081654928624631 Å`

EXERCISE: Rerun Simulation with Query

- **Goal:** Run the nanowire extension simulation with the selected potential

Step 1: Copy your input script to a new file and save your plot files:

```
$ cp in.kim in_query.kim
$ mv out_stress-strain_lmp.dat out_stress-strain_lmp_my.dat
$ mv out_config_lmp.dump out_config_lmp_my.dump
```

Step 2: Edit the new script and replace the "`variable a0`" command with a `kim_query`:

```
$ nano in_query.kim
```

Step 3: Run simulation

```
$ lmp_serial -in in_query.kim > out_query.kim
```

Step 4: Plot results, comparing the new results with the previous results. In gnuplot:

```
plot "out_stress-strain_lmp.dat" with lines, "out_stress-strain_lmp_my.dat"
with lines
```

Raise your hand when you are done.

Results with Query

SLIDE REMOVED

Outline | OpenKIM Tutorial

- LAMMPS Simulation: Nanowire Extension
  - Setting up and running the simulation
  - Results
- OpenKIM Framework and LAMMPS
  - What is OpenKIM?
  - Interatomic potentials on OpenKIM.org
  - Running LAMMPS with OpenKIM potentials
- Potential Selection Exercise
  - Select a suitable interatomic potential from OpenKIM.org
  - Rerun simulation with new potential
- Obtaining Properties through OpenKIM Web Queries
  - LAMMPS kim_query mechanism
  - Query exercise
- Summary

Summary

- Simulation results are strongly affected by the choice of interatomic potential.
- OpenKIM archives many interatomic potentials (models) on openkim.org.
  Each KIM model has the features:
  - Archival storage with provenance control
  - A unique KIM ID and a DOI that can be cited in publications
  - Results for property predictions and verification checks on coding integrity
  - Can be downloaded and used directly with LAMMPS
- KIM models are integrated with LAMMPS:
  - A command is provided to install KIM models. (Automatic when installing from binary)
  - Usage is simple:

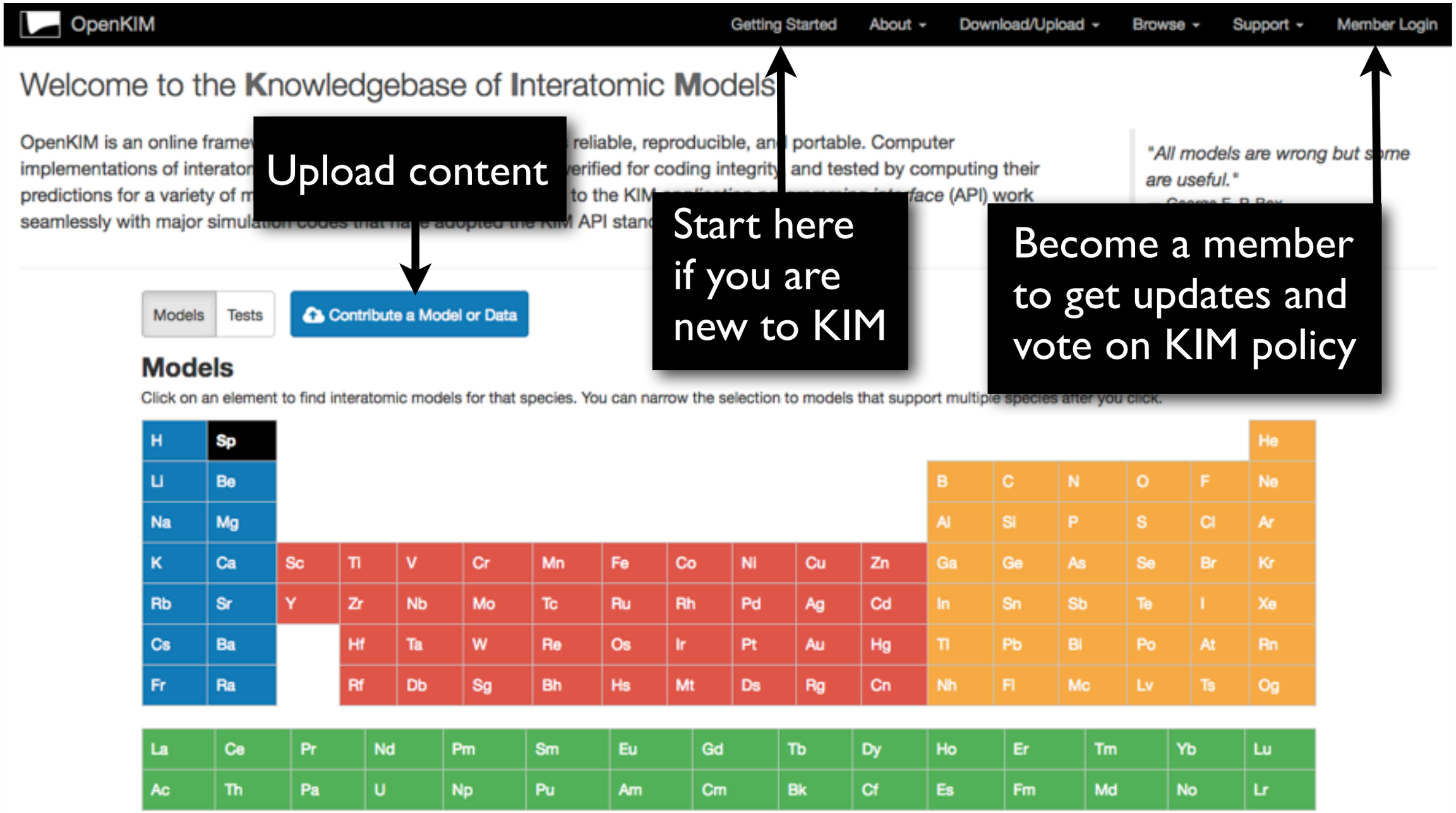
```
kim_init EAM_Dynamo_ZhouWadleyJohnson_2001_Al__MO_049243498555_000 metal
...
kim interactions Al
```

  - Web queries can be performed from within LAMMPS scripts to get properties:

```
kim_query a0 get_lattice_constant_cubic crystal=["fcc"] species=["Al"] units=["angstrom"]
```

  - A mechanism for unit conversion is provided.

OpenKIM.org



https://openkim.org