# MDStressLab User Manual version 1.0.0

Nikhil Chandra Admal and E. B. Tadmor

*Department of Aerospace Engineering and Mechanics*
*University of Minnesota, Minneapolis, MN 55455, USA*

## 1. Introduction

This is the user manual for *MDStressLab*, a computer program for post processing molecular statics (MS) and molecular dynamics (MD) results to obtain stress fields using different definitions of the atomistic stress tensor. The latest version of the code and this user manual are available online at `https://mdstresslab.org`. When making use of this code, please cite the following articles by Admal and Tadmor upon which it is based:

1. Admal, N. C., Tadmor, E. B., 2010. A unified interpretation of stress in molecular systems. Journal of Elasticity 100, 63–143
2. Admal, N. C., Tadmor, E. B., 2015a. The non-uniqueness of the atomistic stress tensor and its relationship to the generalized Beltrami representation. Journal of the Mechanics and Physics of Solids. Submitted
3. Admal, N. C., Tadmor, E. B., 2015b. Referential continuum fields in atomistics. In preparation

### *1.1. Stress Definitions*

The most commonly used atomistic stress definitions are the Hardy (1982), virial Maxwell (1870, 1874) and Tsai (1979) stress tensors. For a given choice of weighting function $w$, the Hardy stress $\boldsymbol{\sigma}_w$ at a point $\boldsymbol{x} \in \mathbb{R}^d$ $(d = 2, 3)$ and time $t$ is given by

$$\boldsymbol{\sigma}_w(\boldsymbol{x}, t) = \boldsymbol{\sigma}_{w,\mathrm{v}}(\boldsymbol{x}, t) + \boldsymbol{\sigma}_{w,\mathrm{k}}(\boldsymbol{x}, t), \tag{1}$$

$$\boldsymbol{\sigma}_{w,\mathrm{v}}(\boldsymbol{x}, t) = \frac{1}{2} \sum_{\substack{\alpha,\beta \\ \alpha\neq\beta}} \int_{s=0}^{1} [-\boldsymbol{f}_{\alpha\beta}\, w((1-s)\boldsymbol{x}_\alpha + s\boldsymbol{x}_\beta - \boldsymbol{x}) \otimes (\boldsymbol{x}_\alpha - \boldsymbol{x}_\beta)]\, ds, \tag{2}$$

$$\boldsymbol{\sigma}_{w,\mathrm{k}}(\boldsymbol{x}, t) = -\sum_{\alpha} m_\alpha (\boldsymbol{v}_\alpha^{\mathrm{rel}} \otimes \boldsymbol{v}_\alpha^{\mathrm{rel}}) w(\boldsymbol{x}_\alpha - \boldsymbol{x}), \tag{3}$$

where $\boldsymbol{x}_\alpha \in \mathbb{R}^d$ and $\boldsymbol{v}_\alpha \in \mathbb{R}^d$ denote the position and velocity of particle $\alpha$ in the deformed configuration, $\boldsymbol{f}_{\alpha\beta}$ is the force between particles $\alpha$ and $\beta$, and $\boldsymbol{v}_\alpha^{\mathrm{rel}}$ is the relative velocity of particle $\alpha$ with respect to the particles in its neighborhood.[1] The summations in (2) and (3) are over all particles, and the summation in (2) is a double summation. The Tsai and virial stress tensors are obtained from the Hardy stress as special cases (Admal and Tadmor, 2010). The Tsai stress tensor is obtained from (1) as a limit of $\boldsymbol{\sigma}_w$ for a sequence of weighting functions whose support collapses to a plane $\mathfrak{l}$. For a plane $\mathfrak{l}$ with normal $\boldsymbol{n}$ and area $A$, the Tsai traction vector is given by

$$\boldsymbol{t}(\boldsymbol{x}, \boldsymbol{n}) = \lim_{T\to\infty} \frac{1}{AT} \left[ \int_0^T \sum_{\alpha\beta\cap\mathfrak{l}} \boldsymbol{f}_{\alpha\beta} \frac{(\boldsymbol{x}_\alpha - \boldsymbol{x}_\beta)\cdot\boldsymbol{n}}{|(\boldsymbol{x}_\alpha - \boldsymbol{x}_\beta)\cdot\boldsymbol{n}|}\, dt - \sum_{\alpha\leftrightarrow\mathfrak{l}}^{T} \frac{m_\alpha \boldsymbol{v}_\alpha(t_\leftrightarrow)(\boldsymbol{v}_\alpha(t_\leftrightarrow)\cdot\boldsymbol{n})}{|\boldsymbol{v}_\alpha(t_\leftrightarrow)\cdot\boldsymbol{n}|} \right], \tag{4}$$

where the first summation is over all bonds that cross the plane, the second summation is over all particles that cross the plane, and $\boldsymbol{v}_\alpha(t_\leftrightarrow)$ denotes the velocity of particle $\alpha$ at a time it crosses the plane $\mathfrak{l}$. The virial stress is obtained from the Hardy stress by taking a constant weighting function, and neglecting bonds that cross or exit the averaging domain in the summation given in (1).

---

[1] See Admal and Tadmor (2015a) and Admal and Tadmor (2015b) for the definition of $\boldsymbol{v}_\alpha^{\mathrm{rel}}$.

*Preprint submitted to Journal of Mechanics and Physics of Solids* *May 22, 2015*

The atomistic stress tensors described above are defined on the deformed configuration of particles and correspond to the spatial Cauchy stress (or true stress) defined in continuum mechanics. To make this clear, in the following the prefix "Cauchy" is used when referring to spatial stress measures.

At the same time, in continuum mechanics the stress tensor also has a material description in terms of the first Piola–Kirchhoff stress. In the absence of an explicit deformation map, it is uncommon to give a material description of the atomistic stress tensor. However, in Admal and Tadmor (2015b) we have shown that there exists a material description of the atomistic stress in the form of an atomistic Piola–Kirchhoff stress. In particular, for a given reference and deformed configuration of particles, $\boldsymbol{X}_\alpha$ and $\boldsymbol{x}_\alpha$ respectively, we have shown that the Hardy version of the Piola–Kirchhoff stress is given by

$$\boldsymbol{P}(\boldsymbol{X},t) = \sum_{\substack{\alpha,\beta \\ \alpha<\beta}} \int_{s=0}^{1} [-\boldsymbol{f}_{\alpha\beta}\, w((1-s)\boldsymbol{X}_\alpha + s\boldsymbol{X}_\beta - \boldsymbol{X}) \otimes (\boldsymbol{X}_\alpha - \boldsymbol{X}_\beta)]\, ds, \tag{5}$$

where $\boldsymbol{f}_{\alpha\beta}$ is evaluated for a deformed configuration of particles. Some notable features of the Piola–Kirchhoff atomistic stress is its non-symmetry and the absence of a kinetic contribution. Similar to the derivation of the Cauchy–Tsai and the Cauchy–virial stress tensors, their Piola counterparts can be obtained as special cases of (5).

### *1.2. Non-uniqueness of the atomistic stress tensor*

The non-uniqueness of the potential part of the atomistic stress tensor due to the non-uniqueness of force decomposition, i.e. the non-uniqueness of the forces $\boldsymbol{f}_{\alpha\beta}$, is a well known issue. Admal and Tadmor (2010) show that this non-uniqueness is directly related to the non-uniqueness of the potential energy representation as a function of distances. In Admal and Tadmor (2015a) we propose a decomposition of the atomistic stress into a unique irrotational part (independent of the potential energy representation) and a non-unique solenoidal (divergence-free) part which depends on the choice of the potential energy representation. Additionally, it is shown that this decomposition has an interesting analog in continuum mechanics in the form of a generalized Beltrami representation, which itself is an analog for symmetric tensor fields of the Helmholtz decomposition of vector fields.

### *1.3. MDStressLab stress calculations and required input*

The *MDStressLab* code can calculate fields of the Cauchy and first Piola–Kirchhoff versions of the Hardy, Tsai and virial stress tensor in two and three dimensions.[2] The user defines a grid of points and the stress is evaluated at all points. Additionally, the decomposition studied in Admal and Tadmor (2015a) can be carried out on each of these stress tensors.

Examining the definitions given in equations (1)-(5), we see that the stress tensor code requires the following input:

- Reference and deformed configuration of particles.
- Velocities of particles.
- The species of each particle (used to identify its mass and used by the force calculations with the interatomic model).
- Details of the periodic boundary conditions (PBCs), and the initial and final size of the box used to prescribe PBCs. In the present version, we restrict ourselves to orthogonal periodic boundary conditions.
- The potential representation used to evaluate the forces $\boldsymbol{f}_{\alpha\beta}$. This is an interatomic model compatible with the application programming interface (API) of the Knowledgebase of Interatomic Models (KIM) as described below.
- The weighting function used to evaluate the Hardy and virial stress tensors. In the present version of the code, the compact support of the weighting function is taken to be a sphere (for $d = 3$) or a circle (for $d = 2$) of radius $r_{\text{avg}}$ specified by the user. The Tsai stress tensor at a point $\boldsymbol{x}$ is constructed using the Tsai traction vector evaluated at $\boldsymbol{x}$ with two (three) equal-sized square planes parallel to the two (three) axes for $d = 2$ ($d = 3$). The length of the planes is taken to be equal to $2r_{\text{avg}}$.

---

[2] The units of stress in two dimensions are force per unit length.

### *1.4. Units*

The units used by the *MDStressLab* program are:

- Distance: Å
- Energy: eV
- Time: ps
- Mass: $\mathrm{eV} \cdot \mathrm{ps}^2/\mathrm{Å}^2$

In the next section, the format of the input file to *MDStressLab* is described.

## 2. Input file

Below is a sample input file to *MDStressLab*. This is followed by a detailed explanation of the commands that appear in it.

```
% Read in atomic configuration and species information
read
    spec,species
    conf,config
end

% Set up the grid for computing the stress field
grid
    gfit,300,300,0
end

% Define the KIM model used to compute the atomic interactions
potential
    modl,Pair_LJ_Smooth_Bernardes_Ar__MO_764178710049_000
end

% Specify whether to decompose stress into unique and non-unique parts
uniqueness
    project,T
end

% Setup and begin stress calculation
stress
    pkstr,F
    avgsize,10.0
    virial,F
    tsai,F
    hardy,T
end

stop
```

The input file consists of six stages: `read`, `grid`, `potential`, `uniqueness`, `stress` and `stop`. All lines starting with `%` are comments. Each stage (except `stop`) contains commands and their associated arguments in the following format:

`command,value_1,value_2,...`

Each stage (except `stop`) ends with an ’`end`’ command. The six stages are described separately below.

### *2.1. Stage* '`read`'

In this stage, the state of the system is read as an input from a configuration file and a species file. The two files are expected to have an extension `.data`. The configuration and species files are specified using the commands `conf` and `spec`, respectively. For example, in the input file shown above, the `conf` commands reads in the the configuration file '`config.data`', and the `spec` commands reads in the species file '`species.data`'. The format for the configuration file is shown below.

```
<d=Dim>   <n=Number of atoms>
<Initial box size>   <Final box size>
<Periodic boundary conditions>
<Species_1>   <Position_1>_i   <Velocity_1>_i
<Species_2>   <Position_2>_i   <Velocity_2>_i
.. .. .. .. .. .. .. .. .. .. .. .. .. ..
<Species_n>   <Position_n>_i   <Velocity_n>_i
<Species_1>   <Position_1>_f   <Velocity_1>_f
<Species_2>   <Position_2>_f   <Velocity_2>_f
.. .. .. .. .. .. .. .. .. .. .. .. .. ..
<Species_n>   <Position_n>_f   <Velocity_n>_f
```

The subscripts `i` and `f` shown above correspond to initial and final configurations. The `<PBC>` field accepts a $d$-dimensional logical (boolean) array indicating whether or not (`T/F`) each of the Cartesian directions is periodic. The fields `<Box size_i>` and `<Box size_f>` accept a $d$-dimensional double precision array. The value of the final box size (in all periodic directions) must be greater than twice the cutoff of the interatomic model. The `<Species_k>` field is a string of length 2 of the element name. A snippet of a configuration file of Aluminum atoms is shown below.

```
3   58260
1000.00   1000.00   26.46   1000.00   1000.00   26.46
F   F   T
    Al      -129.658  -148.180  2.6460  0.00  0.00  0.00
    Al      -129.658  -148.180  13.230  0.00  0.00  0.00
    Al      -129.658  -148.180  7.9382  0.00  0.00  0.00
    ..
```

The species file contains masses of different species of atoms. The format of the species file is shown below.

```
<Species_1>     <mass_1>
<Species_2>     <mass_2>
..
<mass_nsp_file> <mass_nsp_file>
```

A sample species file is given below.

```
Al      0.0027964393
Ar      0.0041406102
Si      0.0029111119
```

### *2.2. Stage* '`grid`'

In this stage, the grid on which the atomistic stress will be evaluated is constructed. There are three different options for specifying the grid points: `gfit`, `gfile` and `gposn`. The grid points can be an ordered or a disordered set of points, and each grid point is a $d$-dimensional array. For PBC conditions, all grid points must lie inside the periodic box, and the distance between any grid point and the box sides[3] must be greater than or equal to the size of the averaging domain.[4] The grid commands and their arguments are described next.

- `gfit,<div_1,...,div_d>` (double precision): This command defines a uniform grid fit to the box. The number of divisions in the $x$, $y$ and $z$ directions for the grid are given by the arguments `<div_1`, `<div_2>` and `<div_3>`, respectively. If `<div_3>` is set to zero, the grid is taken to be two-dimensional and positioned at $z = 0$. The generated grid is stored in a file `grid.data` for later use.

  As an example, below is the grid stage from the sample input file given at the start of Section 2:

  ```
  grid
      gfit,300,300,0
  end
  ```

  The above command generates a two-dimensional grid perpendicular to the $z$-axis. The number of divisions in the $x$ and $y$ direction is 300.

- `gfile,<filename>` (character): This command reads the grid definition from the file `<filename>.data`. The format for the grid file is shown below.

  ```
               <Grid point position_1>
               <Grid point position_2>
               .. .. .. .. .. .. .. .. .. .. .. .. .. ..
               <Grid point position_ng>
  ```

  Here `<Grid point position_i>` are the coordinates of the $i$-th grid point (two space delimited double precision numbers for $d = 2$ and three space delimited double precision numbers for $d = 3$).

  Below is an example of a grid definition using `gfile`.

  ```
  grid
      gfile,grid
  end
  ```

  The above command reads position of the grid points from a file named `grid.data`.

- `gdef,<val_1,...,val_d>,<len_1,...,len_d>,<div_1,...,div_d>`: This command defines a grid whose center is given by the argument `<val_1,...,val_3>`. The lengths of the grid in the $x$, $y$ and $z$ directions relative to the box size in each direction are given by `<len_1>`, `<len_2>` and `<len_3>`, respectively. Their values lie in the interval $[0, 1]$. Finally, the number of divisions in the $x$, $y$ and $z$ directions are given by `<div_1>`, `<div_2>` and `<div_3>`, respectively. The total number of grid points generated by `gfit` is equal to `(<div_1>+1)` × `(<div_2>+1)` × `(<div_3>+1)`. If `<div_3>` is equal to zero, the grid is two-dimensional and positioned at $z = 0$. The generated grid is stored in a file `grid.data` for later use.

  Below is an example of a grid definition using `gdef`.

  ```
  grid
      gdef,0.0,0.0,0.0,1.0,0.5,1.0,300,300,0
  end
  ```

---

[3] The periodic box is defined by six sides for $d = 3$ (or four sides for $d = 2$). For three dimensions, three sides of the box are defined by the minimum of the $x$, $y$ and $z$ coordinates of all particles, and three parallel sides are obtained by offsetting the minimum sides in the $x$, $y$ and $z$ directions by $L_x$, $L_y$ and $L_z$, respectively.

[4] We place this restriction due to the following reason. In order to calculate the atomistic stress at a grid point, the program loops over all bonds that cross the averaging domain centered at the grid point. Under periodic boundary conditions, for a grid point close to the boundary of the bounding box, there are three different kinds of bonds that cross the averaging domain: bonds which lie entirely in the box, bonds that cross the boundaries of the box, and bonds that lie entirely in an image of the box. The program *does not* loop over bonds that lie entirely in the image of the box. All grid points that lie at distance greater than $r_{\text{avg}}$ from the box sides, do not contain such bonds in their averaging domains.

The above command generates a two-dimensional grid centered at the origin with lengths in the $x$, $y$ and $z$ directions equal to `<Initial box size_1>`, 0.5×`<Initial box size_2>` and `<Initial box size_3>`, respectively.

### 2.3. *Stage* 'potential'

In this stage, the interatomic model (potential or force field) used to obtain the interatomic forces is specified. *MDStressLab* is a KIM-compliant program, which means that it works with interatomic models that are compatible with the KIM Application Programming Interface (API) (see `https://openkim.org`). In order to use KIM models, it is necessary to install the KIM API framework. This is a prerequisite for using *MDStressLab*. See the `INSTALL_KIM` file for instructions on how to install the KIM API and accompanying model files. The name of the KIM model is given using the command

- `modl,<val>` (Character(len=80)), where `<val>` is the extended KIM ID of the model.

For example, the command

```
potential
    modl,Pair_LJ_Smooth_Bernardes_Ar__MO_764178710049_000
end
```

(taken from the input file given at the start of Section 2) couples *MDStressLab* to a model with the extended KIM ID "Pair_LJ_Smooth_Bernardes_Ar__MO_764178710049_000". This model (Admal, 2015) describes Ar using a modified Lennard–Jones potential.

In order to compute the various notions of the atomistic stress tensor, *MDStressLab* couples to a KIM compliant model and outsources the computation of interatomic forces to the model. During this process, *MDStressLab* and the coupled model constantly exchange information. The KIM API ensures that a consistent exchange of information happens by performing an appropriate hand-shaking procedure at the beginning of the computation. This hand-shaking is performed by comparing the KIM descriptor files[5] of *MDStressLab* and the KIM model. The current version of *MDStressLab* couples with KIM models with the following capabilities:

1. Since the units of *MDStressLab* are fixed, it can only work with models that support these units or models whose units are flexible.
2. The computation within a model requires a neighbor list corresponding to a given configuration of particles. Within the framework of KIM API, a neighbor list and its accessibility come in various forms. For example, a neighbor list can be "half-neighbor" list or a "full-neighbor" list, both of which cay be accessible through a "locator mode", "iterator mode", or a combination of these two modes. The current version of *MDStressLab* offers a half-neighbor list accessible through iterator and locator modes. Therefore, at this point, only models that support half-neighbor list with locator mode or iterator mode can couple to *MDStressLab*.
3. Since *MDStressLab* supports systems with periodic boundary conditions, only models that can support periodic boundary conditions can couple to it. Items 2 and 3 imply, that *MDStressLab* only works with KIM models that support the MI_OPBC_H neighbor boundary condition (NBC).
4. Since the atomistic stress tensor field depends on the derivative of the energy with respect to distances between particles, *MDStressLab* can only couple to models that can compute this derivative. Within the framework of the KIM API, this capability is described by "process_dEdr". Thus only models that support process_dEdr can couple to *MDStressLab*.

### 2.4. *Stage* 'uniqueness'

In this stage, we specify options related to the decomposition of the atomistic stress into an irrotational part and a solenoidal part. This stage consists of only one command:

- `project,<val>` (logical): If `<val>=T` (true), the decomposition of the atomistic stress into an irrotational part and a solenoidal part is performed. Details of the decomposition are given in Admal and Tadmor (2015a).

---

[5] A KIM descriptor file is a standard formatted file associated with a KIM model or KIM-compliant code that describes its capabilities and the information that it can receive and provide.

### *2.5. Stage* '`stress`'

In this stage, input data and options required to compute various notions of the atomistic stress tensor are specified including the size of the averaging domain. The following commands may be used:

- `kinstr,<val>` (logical): If `<val>=T` (true), the kinetic component of the atomistic stresses (Hardy or virial) is evaluated. In the current version, calculation of the kinetic component of the Tsai stress is not supported.
- `pkstr,<val>` (logical): If `<val>=T` (true), the atomistic Piola–Kirchhoff stress is evaluated on the reference configuration. Otherwise the Cauchy stress is evaluated on the deformed configuration.
- `avgsize,<val>` (double precision): The value of $r_{\text{avg}}$ described in Section 1.
- `virial,<val>` (logical): If `<val>=T` (true), the virial stress is evaluated.
- `tsai,<val>` (logical): If `<val>=T` (true), the Tsai stress is evaluated.
- `hardy,<val>` (logical): If `<val>=T` (true), the hardy stress is evaluated.

As an example, below is the stress stage from the sample input file given at the start of Section 2:

```
stress
    pkstr,F
    avgsize,10.0
    virial,F
    tsai,F
    hardy,T
end
```

The above command is used to compute the Cauchy version of the Hardy stress tensor using an averaging sphere of radius 10 Å.

### *2.6. Stage* '`stop`'

This stage terminates the program in an orderly fashion, freeing any allocated storage and exiting gracefully.

## 3. Output files

The output from *MDStressLab* is directed to standard output and contains information about the progress of the simulation. In addition, the various stress tensor fields requested in the input file are outputted to external files as described below.

- The potential part of the Hardy, virial and Tsai stress tensors are outputted to the files `hardy.str`, `virial.str` and `tsai.str`, respectively.
- The kinetic component of an atomistic stress is outputted to a file with a suffix `_kin.str`. For instance, the kinetic component of the Hardy stress is outputted to the file `hardy_kin.str`.
- The decomposition of the potential part of the atomistic stress tensor is given by evaluating the irrotational and the solenoidal parts, and outputting them to files with a suffixes `_i.str` and _s.str respectively, e.g. the irrotational and solenoidal parts of the Hardy stress can be found in the files `hardy_i.str` and `hardy_s.str` respectively.

An `str` output file consist of 12 columns and a number of rows equal to the number of grid points. The first three columns are the coordinates of the the grid points, and the next nine columns are the components of the stress tensor in the following order: 11, 21, 31, 12, 22, 32, 13, 23, 33.

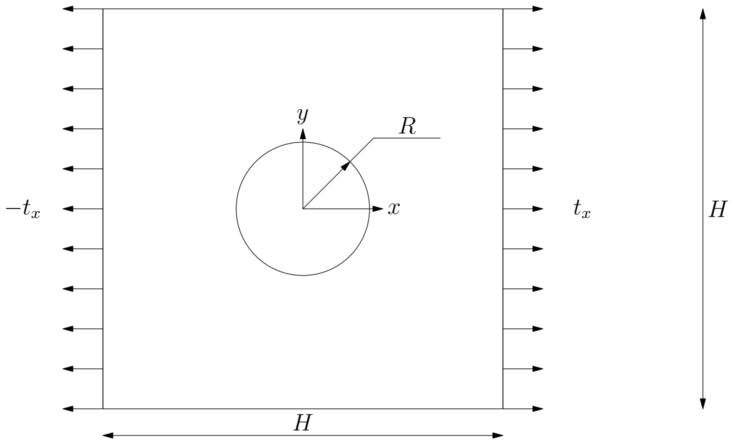

Figure 1: A schematic of the geometry of the studied boundary-value problem, with non-zero traction boundary conditions enforced on a part of the boundary as shown.

## 4. Examples

An example of a rectangular anisotropic medium with a hole subjected to a uniaxial loading under plane strain conditions is provided with the *MDStressLab* code. As part of the example various stress measures are computed and the decomposition into the unique (irrotational) and non-unique (solenoidal) parts is performed. A schematic of the body showing the relevant dimensions and the boundary conditions is shown in Fig. 1. The size of the body, radius of the hole and applied traction shown in Fig. 1 are

$$H = 300\ \text{Å}, \quad R = 50\ \text{Å}, \quad \text{and} \quad \bar{t}_x = 7.117 \times 10^{-5}\,\text{eV}/\text{Å}^3. \tag{6}$$

The body extends to infinity in the positive and negative $z$-directions (i.e. periodic boundary conditions are applied in the out-of-plane direction). The material model is taken to be single crystal Ar in the face-centered cubic (fcc) structure such that the $x$, $y$ and $z$ axes are oriented along the $[100]$, $[010]$ and $[001]$ crystallographic directions. The system is studied at zero temperature.

The boundary-value problem discussed above is inputted to the *MDStressLab* code in the form of reference configuration, final configuration and the potential energy model for the interaction of Ar atoms. The reference (unloaded) configuration of the plate is obtained by stacking $56 \times 56 \times 1$ unit cells with lattice parameter $a_0$ corresponding to the stress free state of the chosen potential model, and removing all atoms that fall inside a circle of radius $50$ Å positioned at the center of the stack, with its out-of-plane axis parallel to the [001] crystallographic axis. The resulting system consists of $11652$ atoms. The interatomic potential is a modified Lennard–Jones pair potential archived in OpenKIM (Admal, 2015; Tadmor et al., 2011). The final configuration is obtained by displacing the atoms in the reference configuration according to the numerical solution of the traction boundary-value problem described above. For a full discussion of the problem, see Admal and Tadmor (2015a).

The example is contained in directory `examples/Plate_w_hole`. The directory contains the following files:

- `README`: Contains instructions for how to run the example.
- `config.data`: Contains the species, positions and velocities of the atoms and details about the periodic boundary conditions. See Section 2.1 for more information.
- `species.data`: Contains the mass of the Ar atom. See Section 2.1 for more information.
- `testrun.in`: The input file to *MDStressLab* setting up the problem and defining what needs to be computed.

## References

Admal, N. C., 2015. Modified Lennard–Jones potential for Argon.
URL `https://openkim.org/cite/MO_764178710049_000`

Admal, N. C., Tadmor, E. B., 2010. A unified interpretation of stress in molecular systems. Journal of Elasticity 100, 63–143.

Admal, N. C., Tadmor, E. B., 2015a. The non-uniqueness of the atomistic stress tensor and its relationship to the generalized Beltrami representation. Journal of the Mechanics and Physics of Solids. Submitted.

Admal, N. C, Tadmor, E. B., 2015b. Referential continuum fields in atomistics. In preparation.

Hardy, R. J., 1982. Formulas for determining local properties in molecular dynamics simulation: Shock waves. The Journal of Chemical Physics 76 (01), 622–628.

Maxwell, J. C., 1870. On reciprocal figures, frames and diagrams of forces. Translations of the Royal Society of Edinburgh XXVI, 1–43.

Maxwell, J. C., 1874. Van der waals on the continuity of the gaseous and liquid states. Nature 10, 477–80.

Tadmor, E. B., Elliott, R. S., Sethna, J. P., Miller, R. E., Becker, C. A., 2011. The potential of atomistic simulations and the Knowledgebase of Interatomic Models. JOM Journal of the Minerals, Metals and Materials Society 63 (7), 17–17.

Tsai, D. H., 1979. The virial theorem and stress calculation in molecular dynamics. The Journal of Chemical Physics 70 (03), 1375–1382.