CHAPTER

SIX

ANGLE STYLES

6.1 angle_style amoeba command

6.1.1 Syntax

```
angle_style amoeba
```

6.1.2 Examples

```
angle_style amoeba
angle_coeff * 75.0 -25.0 1.0 0.3 0.02 0.003
angle_coeff * ba 3.6551 24.895 1.0119 1.5228
angle_coeff * ub -7.6 1.5537
```

6.1.3 Description

The *amoeba* angle style uses the potential

$$
\begin{aligned}
E &= E_a + E_{ba} + E_{ub} \\
E_a &= K_2\,(\theta - \theta_0)^2 + K_3\,(\theta - \theta_0)^3 + K_4\,(\theta - \theta_0)^4 + K_5\,(\theta - \theta_0)^5 + K_6\,(\theta - \theta_0)^6 \\
E_{ba} &= N_1(r_{ij} - r_1)(\theta - \theta_0) + N_2(r_{jk} - r_2)(\theta - \theta_0) \\
E_{UB} &= K_{ub}(r_{ik} - r_{ub})^2
\end{aligned}
$$

where E_a is the angle term, E_{ba} is a bond-angle term, E_{UB} is a Urey-Bradley bond term, θ_0 is the equilibrium angle, r_1 and r_2 are the equilibrium bond lengths, and r_{ub} is the equilibrium Urey-Bradley bond length.

These formulas match how the Tinker MD code performs its angle calculations for the AMOEBA and HIPPO force fields. See the *Howto amoeba* page for more information about the implementation of AMOEBA and HIPPO in LAMMPS.

Note that the E_a and E_{ba} formulas are identical to those used for the *angle_style class2/p6* command, however there is no bond-bond cross term formula for E_{bb}. Additionally, there is a E_{UB} term for a Urey-Bradley bond. It is effectively a harmonic bond between the I and K atoms of angle IJK, even though that bond is not enumerated in the "Bonds" section of the data file.

There are also two ways that Tinker computes the angle θ in the E_a formula. The first is the standard way of treating IJK as an "in-plane" angle. The second is an "out-of-plane" method which Tinker may use if the center atom J in the angle is bonded to one additional atom in addition to I and K. In this case, all 4 atoms are used to compute the E_a formula, resulting in forces on all 4 atoms. In the Tinker PRM file, these 2 options are denoted by *angle* versus *anglep*

entries in the "Angle Bending Parameters" section of the PRM force field file. The *pflag* coefficient described below selects between the 2 options.

Coefficients for the E_a, E_{bb}, and E_{ub} formulas must be defined for each angle type via the *angle_coeff* command as in the example above, or in the data file or restart files read by the *read_data* or *read_restart* commands.

These are the 8 coefficients for the E_a formula:

- pflag = 0 or 1
- ubflag = 0 or 1
- θ_0 (degrees)
- K_2 (energy)
- K_3 (energy)
- K_4 (energy)
- K_5 (energy)
- K_6 (energy)

A pflag value of 0 vs 1 selects between the "in-plane" and "out-of-plane" options described above. Ubflag is 1 if there is a Urey-Bradley term associated with this angle type, else it is 0. θ_0 is specified in degrees, but LAMMPS converts it to radians internally; hence the various K values are effectively energy per radian^2 or radian^3 or radian^4 or radian^5 or radian^6.

For the E_{ba} formula, each line in a *angle_coeff* command in the input script lists 5 coefficients, the first of which is "ba" to indicate they are BondAngle coefficients. In a data file, these coefficients should be listed under a "BondAngle Coeffs" heading and you must leave out the "ba", i.e. only list 4 coefficients after the angle type.

- ba
- N_1 (energy/distance^2)
- N_2 (energy/distance^2)
- r_1 (distance)
- r_2 (distance)

The θ_0 value in the E_{ba} formula is not specified, since it is the same value from the E_a formula.

For the E_{ub} formula, each line in a *angle_coeff* command in the input script lists 3 coefficients, the first of which is "ub" to indicate they are UreyBradley coefficients. In a data file, these coefficients should be listed under a "UreyBradley Coeffs" heading and you must leave out the "ub", i.e. only list 2 coefficients after the angle type.

- ub
- K_{ub} (energy/distance^2)
- r_{ub} (distance)

6.1.4 Restrictions

This angle style can only be used if LAMMPS was built with the AMOEBA package. See the *Build package* doc page for more info.

6.1.5 Related commands

angle_coeff

6.1.6 Default

none

6.2 angle_style charmm command

Accelerator Variants: *charmm/intel*, *charmm/kk*, *charmm/omp*

6.2.1 Syntax

```
angle_style charmm
```

6.2.2 Examples

```
angle_style charmm
angle_coeff 1 300.0 107.0 50.0 3.0
```

6.2.3 Description

The *charmm* angle style uses the potential

$$E = K(\theta - \theta_0)^2 + K_{ub}(r - r_{ub})^2$$

with an additional Urey_Bradley term based on the distance r between the first and third atoms in the angle. K, θ_0, K_{ub}, and R_{ub} are coefficients defined for each angle type.

See *(MacKerell)* for a description of the CHARMM force field.

The following coefficients must be defined for each angle type via the *angle_coeff* command as in the example above, or in the data file or restart files read by the *read_data* or *read_restart* commands:

- K (energy)
- θ_0 (degrees)
- K_{ub} (energy/distance^2)
- r_{ub} (distance)

θ_0 is specified in degrees, but LAMMPS converts it to radians internally; hence K is effectively energy per radian^2.

Styles with a *gpu*, *intel*, *kk*, *omp*, or *opt* suffix are functionally the same as the corresponding style without the suffix. They have been optimized to run faster, depending on your available hardware, as discussed on the *Accelerator packages* page. The accelerated styles take the same arguments and should produce the same results, except for round-off and precision issues.

These accelerated styles are part of the GPU, INTEL, KOKKOS, OPENMP, and OPT packages, respectively. They are only enabled if LAMMPS was built with those packages. See the *Build package* page for more info.

You can specify the accelerated styles explicitly in your input script by including their suffix, or you can use the *-suffix command-line switch* when you invoke LAMMPS, or you can use the *suffix* command in your input script.

See the *Accelerator packages* page for more instructions on how to use the accelerated styles effectively.

6.2.4 Restrictions

This angle style can only be used if LAMMPS was built with the MOLECULE package. See the *Build package* doc page for more info.

6.2.5 Related commands

angle_coeff, *pair_style lj/charmm variants*, *dihedral_style charmm*, *dihedral_style charmmfsw*, *fix cmap*

6.2.6 Default

none

(MacKerell) MacKerell, Bashford, Bellott, Dunbrack, Evanseck, Field, Fischer, Gao, Guo, Ha, et al, J Phys Chem, 102, 3586 (1998).

6.3 angle_style class2 command

Accelerator Variants: *class2/kk*, *class2/omp*

6.4 angle_style class2/p6 command

6.4.1 Syntax

```
angle_style class2
```

6.4.2 Examples

```
angle_style class2
angle_coeff * 75.0 25.0 0.3 0.002
angle_coeff 1 bb 10.5872 1.0119 1.5228
angle_coeff * ba 3.6551 24.895 1.0119 1.5228
```

6.4.3 Description

The *class2* angle style uses the potential

$$
\begin{aligned}
E &= E_a + E_{bb} + E_{ba} \\
E_a &= K_2(\theta - \theta_0)^2 + K_3(\theta - \theta_0)^3 + K_4(\theta - \theta_0)^4 \\
E_{bb} &= M(r_{ij} - r_1)(r_{jk} - r_2) \\
E_{ba} &= N_1(r_{ij} - r_1)(\theta - \theta_0) + N_2(r_{jk} - r_2)(\theta - \theta_0)
\end{aligned}
$$

where E_a is the angle term, E_{bb} is a bond-bond term, and E_{ba} is a bond-angle term. θ_0 is the equilibrium angle and r_1 and r_2 are the equilibrium bond lengths.

See *(Sun)* for a description of the COMPASS class2 force field.

Coefficients for the E_a, E_{bb}, and E_{ba} formulas must be defined for each angle type via the *angle_coeff* command as in the example above, or in the data file or restart files read by the *read_data* or *read_restart* commands.

These are the 4 coefficients for the E_a formula:

- θ_0 (degrees)
- K_2 (energy)
- K_3 (energy)
- K_4 (energy)

θ_0 is specified in degrees, but LAMMPS converts it to radians internally; hence the various K are effectively energy per radian^2 or radian^3 or radian^4.

For the E_{bb} formula, each line in a *angle_coeff* command in the input script lists 4 coefficients, the first of which is "bb" to indicate they are BondBond coefficients. In a data file, these coefficients should be listed under a "BondBond Coeffs" heading and you must leave out the "bb", i.e. only list 3 coefficients after the angle type.

- bb
- M (energy/distance^2)
- r_1 (distance)
- r_2 (distance)

For the E_{ba} formula, each line in a *angle_coeff* command in the input script lists 5 coefficients, the first of which is "ba" to indicate they are BondAngle coefficients. In a data file, these coefficients should be listed under a "BondAngle Coeffs" heading and you must leave out the "ba", i.e. only list 4 coefficients after the angle type.

- ba
- N_1 (energy/distance)
- N_2 (energy/distance)
- r_1 (distance)

- r_2 (distance)

The θ_0 value in the E_{ba} formula is not specified, since it is the same value from the E_a formula.

Note

It is important that the order of the I,J,K atoms in each angle listed in the Angles section of the data file read by the *read_data* command be consistent with the order of the r_1 and r_2 BondBond and BondAngle coefficients. This is because the terms in the formulas for E_{bb} and E_{ba} will use the I,J atoms to compute r_{ij} and the J,K atoms to compute r_{jk}.

Styles with a *gpu*, *intel*, *kk*, *omp*, or *opt* suffix are functionally the same as the corresponding style without the suffix. They have been optimized to run faster, depending on your available hardware, as discussed on the *Accelerator packages* page. The accelerated styles take the same arguments and should produce the same results, except for round-off and precision issues.

These accelerated styles are part of the GPU, INTEL, KOKKOS, OPENMP, and OPT packages, respectively. They are only enabled if LAMMPS was built with those packages. See the *Build package* page for more info.

You can specify the accelerated styles explicitly in your input script by including their suffix, or you can use the *-suffix command-line switch* when you invoke LAMMPS, or you can use the *suffix* command in your input script.

See the *Accelerator packages* page for more instructions on how to use the accelerated styles effectively.

The *class2/p6* angle style uses the *class2* potential expanded to sixth order:

$$E_a = K_2 (\theta - \theta_0)^2 + K_3 (\theta - \theta_0)^3 + K_4 (\theta - \theta_0)^4 + K_5 (\theta - \theta_0)^5 + K_6 (\theta - \theta_0)^6$$

In this expanded term 6 coefficients for the E_a formula need to be set:

- θ_0 (degrees)
- K_2 (energy)
- K_3 (energy)
- K_4 (energy)
- K_5 (energy)
- K_6 (energy)

θ_0 is specified in degrees, but LAMMPS converts it to radians internally; hence the various K are effectively energy per radian^2 or radian^3 or radian^4 or radian^5 or radian^6.

The bond-bond and bond-angle terms remain unchanged.

6.4.4 Restrictions

This angle style can only be used if LAMMPS was built with the CLASS2 package. For the *class2/p6* style LAMMPS needs to be built with the MOFFF package. See the *Build package* doc page for more info.

6.4.5 Related commands

angle_coeff

6.4.6 Default

none

(Sun) Sun, J Phys Chem B 102, 7338-7364 (1998).

6.5 angle_style cosine command

Accelerator Variants: *cosine/omp*, *cosine/kk*

6.5.1 Syntax

```
angle_style cosine
```

6.5.2 Examples

```
angle_style cosine
angle_coeff * 75.0
```

6.5.3 Description

The *cosine* angle style uses the potential

$$E = K[1 + \cos(\theta)]$$

where K is defined for each angle type.

The following coefficients must be defined for each angle type via the *angle_coeff* command as in the example above, or in the data file or restart files read by the *read_data* or *read_restart* commands:

- K (energy)

Styles with a *gpu*, *intel*, *kk*, *omp*, or *opt* suffix are functionally the same as the corresponding style without the suffix. They have been optimized to run faster, depending on your available hardware, as discussed on the *Accelerator packages* page. The accelerated styles take the same arguments and should produce the same results, except for round-off and precision issues.

These accelerated styles are part of the GPU, INTEL, KOKKOS, OPENMP, and OPT packages, respectively. They are only enabled if LAMMPS was built with those packages. See the *Build package* page for more info.

You can specify the accelerated styles explicitly in your input script by including their suffix, or you can use the *-suffix command-line switch* when you invoke LAMMPS, or you can use the *suffix* command in your input script.

See the *Accelerator packages* page for more instructions on how to use the accelerated styles effectively.

6.5.4 Restrictions

This angle style can only be used if LAMMPS was built with the MOLECULE package. See the *Build package* doc page for more info.

6.5.5 Related commands

angle_coeff

6.5.6 Default

none

6.6 angle_style cosine/buck6d command

6.6.1 Syntax

```
angle_style cosine/buck6d
```

6.6.2 Examples

```
angle_style cosine/buck6d
angle_coeff 1  cosine/buck6d  1.978350  4  180.000000
```

6.6.3 Description

The *cosine/buck6d* angle style uses the potential

$$E = K\,[1 + \cos(n\theta - \theta_0)]$$

where K is the energy constant, n is the periodic multiplicity and θ_0 is the equilibrium angle.

The coefficients must be defined for each angle type via the *angle_coeff* command as in the example above, or in the data file or restart files read by the *read_data* or *read_restart* commands in the following order:

- K (energy)
- n
- θ_0 (degrees)

θ_0 is specified in degrees, but LAMMPS converts it to radians internally.

Additional to the cosine term the *cosine/buck6d* angle style computes the short range (vdW) interaction belonging to the *pair_style buck6d* between the end atoms of the angle. For this reason this angle style only works in combination with the *pair_style buck6d* styles and needs the *special_bonds* 1-3 interactions to be weighted 0.0 to prevent double counting.

6.6.4 Restrictions

cosine/buck6d can only be used in combination with the *pair_style buck6d* style and with a *special_bonds* 0.0 weighting of 1-3 interactions.

This angle style can only be used if LAMMPS was built with the MOFFF package. See the *Build package* doc page for more info.

6.6.5 Related commands

angle_coeff

6.6.6 Default

none

6.7 angle_style cosine/delta command

Accelerator Variants: *cosine/delta/omp*

6.7.1 Syntax

```
angle_style cosine/delta
```

6.7.2 Examples

```
angle_style cosine/delta
angle_coeff 2*4 75.0 100.0
```

6.7.3 Description

The *cosine/delta* angle style uses the potential

$$E = K[1 - \cos(\theta - \theta_0)]$$

where θ_0 is the equilibrium value of the angle, and K is a prefactor. Note that the usual 1/2 factor is included in K.

The following coefficients must be defined for each angle type via the *angle_coeff* command as in the example above, or in the data file or restart files read by the *read_data* or *read_restart* commands:

- K (energy)
- θ_0 (degrees)

θ_0 is specified in degrees, but LAMMPS converts it to radians internally.

Styles with a *gpu*, *intel*, *kk*, *omp*, or *opt* suffix are functionally the same as the corresponding style without the suffix. They have been optimized to run faster, depending on your available hardware, as discussed on the *Accelerator packages* page. The accelerated styles take the same arguments and should produce the same results, except for round-off and precision issues.

These accelerated styles are part of the GPU, INTEL, KOKKOS, OPENMP, and OPT packages, respectively. They are only enabled if LAMMPS was built with those packages. See the *Build package* page for more info.

You can specify the accelerated styles explicitly in your input script by including their suffix, or you can use the *-suffix command-line switch* when you invoke LAMMPS, or you can use the *suffix* command in your input script.

See the *Accelerator packages* page for more instructions on how to use the accelerated styles effectively.

6.7.4 Restrictions

This angle style can only be used if LAMMPS was built with the EXTRA-MOLECULE package. See the *Build package* doc page for more info.

6.7.5 Related commands

angle_coeff , *angle_style cosine/squared*

6.7.6 Default

none

6.8 angle_style cosine/periodic command

Accelerator Variants: *cosine/periodic/omp*

6.8.1 Syntax

```
angle_style cosine/periodic
```

6.8.2 Examples

```
angle_style cosine/periodic
angle_coeff * 75.0 1 6
```

6.8.3 Description

The *cosine/periodic* angle style uses the following potential, which may be particularly used for organometallic systems where $n = 4$ might be used for an octahedral complex and $n = 3$ might be used for a trigonal center:

$$E = \frac{2.0}{n^2} * C \left[1 - B(-1)^n \cos\left(n\theta\right)\right]$$

where C, B and n are coefficients defined for each angle type.

The following coefficients must be defined for each angle type via the *angle_coeff* command as in the example above, or in the data file or restart files read by the *read_data* or *read_restart* commands:

- C (energy)
- B = 1 or -1
- n = 1, 2, 3, 4, 5 or 6 for periodicity

Note that the prefactor C is specified as coefficient and not the overall force constant $K = \frac{2C}{n^2}$. When $B = 1$, it leads to a minimum for the linear geometry. When $B = -1$, it leads to a maximum for the linear geometry.

Styles with a *gpu*, *intel*, *kk*, *omp*, or *opt* suffix are functionally the same as the corresponding style without the suffix. They have been optimized to run faster, depending on your available hardware, as discussed on the *Accelerator packages* page. The accelerated styles take the same arguments and should produce the same results, except for round-off and precision issues.

These accelerated styles are part of the GPU, INTEL, KOKKOS, OPENMP, and OPT packages, respectively. They are only enabled if LAMMPS was built with those packages. See the *Build package* page for more info.

You can specify the accelerated styles explicitly in your input script by including their suffix, or you can use the *-suffix command-line switch* when you invoke LAMMPS, or you can use the *suffix* command in your input script.

See the *Accelerator packages* page for more instructions on how to use the accelerated styles effectively.

6.8.4 Restrictions

This angle style can only be used if LAMMPS was built with the EXTRA-MOLECULE package. See the *Build package* doc page for more info.

6.8.5 Related commands

angle_coeff

6.8.6 Default

none

6.9 angle_style cosine/shift command

Accelerator Variants: *cosine/shift/omp*

6.9.1 Syntax

```
angle_style cosine/shift
```

6.9.2 Examples

```
angle_style cosine/shift
angle_coeff * 10.0 45.0
```

6.9.3 Description

The *cosine/shift* angle style uses the potential

$$E = -\frac{U_{\min}}{2}\left[1 + \cos(\theta - \theta_0)\right]$$

where θ_0 is the equilibrium angle. The potential is bounded between $-U_{\min}$ and zero. In the neighborhood of the minimum $E = -U_{\min} + U_{\min}/4(\theta - \theta_0)^2$ hence the spring constant is $\frac{U_{\min}}{2}$.

The following coefficients must be defined for each angle type via the *angle_coeff* command as in the example above, or in the data file or restart files read by the *read_data* or *read_restart* commands:

- $U_{\min}$ (energy)
- θ (angle)

Styles with a *gpu*, *intel*, *kk*, *omp*, or *opt* suffix are functionally the same as the corresponding style without the suffix. They have been optimized to run faster, depending on your available hardware, as discussed on the *Accelerator packages* page. The accelerated styles take the same arguments and should produce the same results, except for round-off and precision issues.

These accelerated styles are part of the GPU, INTEL, KOKKOS, OPENMP, and OPT packages, respectively. They are only enabled if LAMMPS was built with those packages. See the *Build package* page for more info.

You can specify the accelerated styles explicitly in your input script by including their suffix, or you can use the *-suffix command-line switch* when you invoke LAMMPS, or you can use the *suffix* command in your input script.

See the *Accelerator packages* page for more instructions on how to use the accelerated styles effectively.

6.9.4 Restrictions

This angle style can only be used if LAMMPS was built with the EXTRA-MOLECULE package. See the *Build package* doc page for more info.

6.9.5 Related commands

angle_coeff, *angle_style cosine/shift/exp*

6.9.6 Default

none

6.10 angle_style cosine/shift/exp command

Accelerator Variants: *cosine/shift/exp/omp*

6.10.1 Syntax

```
angle_style cosine/shift/exp
```

6.10.2 Examples

```
angle_style cosine/shift/exp
angle_coeff * 10.0 45.0 2.0
```

6.10.3 Description

The *cosine/shift/exp* angle style uses the potential

$$E = -U_{\min}\frac{e^{-aU(\theta,\theta_0)}-1}{e^{a}-1} \quad \text{with} \quad U(\theta,\theta_0) = -0.5\,(1+\cos(\theta-\theta_0))$$

where $U_{\min}$, θ, and a are defined for each angle type.

The potential is bounded between $[-U_{\min}, 0]$ and the minimum is located at the angle θ_0. The a parameter can be both positive or negative and is used to control the spring constant at the equilibrium.

The spring constant is given by $k = A\exp(A)U_{\min}/[2(\exp(a)-1)]$. For $a > 3$, $\frac{k}{U_{\min}} = \frac{a}{2}$ to better than 5% relative error. For negative values of the a parameter, the spring constant is essentially zero, and anharmonic terms takes over. The potential is furthermore well behaved in the limit $a \to 0$, where it has been implemented to linear order in a for $a < 0.001$. In this limit the potential reduces to the cosineshifted potential.

The following coefficients must be defined for each angle type via the *angle_coeff* command as in the example above, or in the data file or restart files read by the *read_data* or *read_restart* commands:

- U_min (energy)
- θ (angle)

- *A* (real number)

Styles with a *gpu*, *intel*, *kk*, *omp*, or *opt* suffix are functionally the same as the corresponding style without the suffix. They have been optimized to run faster, depending on your available hardware, as discussed on the *Accelerator packages* page. The accelerated styles take the same arguments and should produce the same results, except for round-off and precision issues.

These accelerated styles are part of the GPU, INTEL, KOKKOS, OPENMP, and OPT packages, respectively. They are only enabled if LAMMPS was built with those packages. See the *Build package* page for more info.

You can specify the accelerated styles explicitly in your input script by including their suffix, or you can use the *-suffix command-line switch* when you invoke LAMMPS, or you can use the *suffix* command in your input script.

See the *Accelerator packages* page for more instructions on how to use the accelerated styles effectively.

6.10.4 Restrictions

This angle style can only be used if LAMMPS was built with the EXTRA-MOLECULE package. See the *Build package* doc page for more info.

6.10.5 Related commands

angle_coeff, *angle_style cosine/shift*, *dihedral_style cosine/shift/exp*

6.10.6 Default

none

6.11 angle_style cosine/squared command

Accelerator Variants: *cosine/squared/omp*

6.11.1 Syntax

```
angle_style cosine/squared
```

6.11.2 Examples

```
angle_style cosine/squared
angle_coeff 2*4 75.0 100.0
```

6.11.3 Description

The *cosine/squared* angle style uses the potential

$$E = K[\cos(\theta) - \cos(\theta_0)]^2$$

, which is commonly used in the *DREIDING* force field, where θ_0 is the equilibrium value of the angle, and K is a prefactor. Note that the usual 1/2 factor is included in K.

See *(Mayo)* for a description of the DREIDING force field.

The following coefficients must be defined for each angle type via the *angle_coeff* command as in the example above, or in the data file or restart files read by the *read_data* or *read_restart* commands:

- K (energy)
- θ_0 (degrees)

θ_0 is specified in degrees, but LAMMPS converts it to radians internally.

Styles with a *gpu*, *intel*, *kk*, *omp*, or *opt* suffix are functionally the same as the corresponding style without the suffix. They have been optimized to run faster, depending on your available hardware, as discussed on the *Accelerator packages* page. The accelerated styles take the same arguments and should produce the same results, except for round-off and precision issues.

These accelerated styles are part of the GPU, INTEL, KOKKOS, OPENMP, and OPT packages, respectively. They are only enabled if LAMMPS was built with those packages. See the *Build package* page for more info.

You can specify the accelerated styles explicitly in your input script by including their suffix, or you can use the *-suffix command-line switch* when you invoke LAMMPS, or you can use the *suffix* command in your input script.

See the *Accelerator packages* page for more instructions on how to use the accelerated styles effectively.

6.11.4 Restrictions

This angle style can only be used if LAMMPS was built with the MOLECULE package. See the *Build package* doc page for more info.

6.11.5 Related commands

angle_coeff

6.11.6 Default

none

(Mayo) Mayo, Olfason, Goddard III, J Phys Chem, 94, 8897-8909 (1990).

6.12 angle_style cosine/squared/restricted command

Accelerator Variants: *cosine/squared/restricted/omp*

6.12.1 Syntax

```
angle_style cosine/squared/restricted
```

6.12.2 Examples

```
angle_style cosine/squared/restricted
angle_coeff 2*4 75.0 100.0
```

6.12.3 Description

Added in version 17Apr2024.

The *cosine/squared/restricted* angle style uses the potential

$$E = K[\cos(\theta) - \cos(\theta_0)]^2 / \sin^2(\theta)$$

, which is commonly used in the MARTINI force field, where θ_0 is the equilibrium value of the angle, and K is a prefactor. Note that the usual 1/2 factor is included in K.

See *(Bulacu)* for a description of the restricted angle for the MARTINI force field.

The following coefficients must be defined for each angle type via the *angle_coeff* command as in the example above, or in the data file or restart files read by the *read_data* or *read_restart* commands:

- K (energy)
- θ_0 (degrees)

θ_0 is specified in degrees, but LAMMPS converts it to radians internally.

Styles with a *gpu*, *intel*, *kk*, *omp*, or *opt* suffix are functionally the same as the corresponding style without the suffix. They have been optimized to run faster, depending on your available hardware, as discussed on the *Accelerator packages* page. The accelerated styles take the same arguments and should produce the same results, except for round-off and precision issues.

These accelerated styles are part of the GPU, INTEL, KOKKOS, OPENMP, and OPT packages, respectively. They are only enabled if LAMMPS was built with those packages. See the *Build package* page for more info.

You can specify the accelerated styles explicitly in your input script by including their suffix, or you can use the *-suffix command-line switch* when you invoke LAMMPS, or you can use the *suffix* command in your input script.

See the *Accelerator packages* page for more instructions on how to use the accelerated styles effectively.

6.12.4 Restrictions

This angle style can only be used if LAMMPS was built with the EXTRA-MOLECULE package. See the *Build package* doc page for more info.

6.12.5 Related commands

angle_coeff

6.12.6 Default

none

(Bulacu) Bulacu, Goga, Zhao, Rossi, Monticelli, Periole, Tieleman, Marrink, J Chem Theory Comput, 9, 3282-3292 (2013).

6.13 angle_style cross command

6.13.1 Syntax

```
angle_style cross
```

6.13.2 Examples

```
angle_style cross
angle_coeff 1 200.0 100.0 100.0 1.25 1.25 107.0
```

6.13.3 Description

The *cross* angle style uses a potential that couples the bond stretches of a bend with the angle stretch of that bend:

$$E = K_{SS}\left(r_{12} - r_{12,0}\right)\left(r_{32} - r_{32,0}\right) + K_{BS0}\left(r_{12} - r_{12,0}\right)\left(\theta - \theta_0\right) + K_{BS1}\left(r_{32} - r_{32,0}\right)\left(\theta - \theta_0\right)$$

where $r_{12,0}$ is the rest value of the bond length between atom 1 and 2, $r_{32,0}$ is the rest value of the bond length between atom 3 and 2, and θ_0 is the rest value of the angle. K_{SS} is the force constant of the bond stretch-bond stretch term and K_{BS0} and K_{BS1} are the force constants of the bond stretch-angle stretch terms.

The following coefficients must be defined for each angle type via the *angle_coeff* command as in the example above, or in the data file or restart files read by the *read_data* or *read_restart* commands:

- K_{SS} (energy/distance^2)
- K_{BS0} (energy/distance)
- K_{BS1} (energy/distance)
- $r_{12,0}$ (distance)
- $r_{32,0}$ (distance)

- θ_0 (degrees)

θ_0 is specified in degrees, but LAMMPS converts it to radians internally; hence the K_{BS0} and K_{BS1} are effectively energy/distance per radian.

6.13.4 Restrictions

This angle style can only be used if LAMMPS was built with the YAFF package. See the *Build package* doc page for more info.

6.13.5 Related commands

angle_coeff

6.13.6 Default

none

6.14 angle_style dipole command

Accelerator Variants: *dipole/omp*

6.14.1 Syntax

```
angle_style dipole
```

6.14.2 Examples

```
angle_style dipole
angle_coeff 6 2.1 180.0
```

6.14.3 Description

The *dipole* angle style is used to control the orientation of a dipolar atom within a molecule *(Orsi)*. Specifically, the *dipole* angle style restrains the orientation of a point dipole μ_j (embedded in atom j) with respect to a reference (bond) vector $\vec{r}_{ij} = \vec{r}_i - \vec{r}_j$, where i is another atom of the same molecule (typically, i and j are also covalently bonded).

It is convenient to define an angle gamma between the 'free' vector $\vec{\mu}_j$ and the reference (bond) vector $\vec{r}_{ij}$:

$$\cos\gamma = \frac{\vec{\mu}_j \cdot \vec{r}_{ij}}{\mu_j \, r_{ij}}$$

The *dipole* angle style uses the potential:

$$E = K(\cos\gamma - \cos\gamma_0)^2$$

where K is a rigidity constant and gamma0 is an equilibrium (reference) angle.

The torque on the dipole can be obtained by differentiating the potential using the 'chain rule' as in appendix C.3 of *(AllenTildesley)*:

$$\vec{T}_j = \frac{2K(\cos\gamma - \cos\gamma_0)}{\mu_j r_{ij}} \vec{r}_{ij} \times \vec{\mu}_j$$

Example: if γ_0 is set to 0 degrees, the torque generated by the potential will tend to align the dipole along the reference direction defined by the (bond) vector $\vec{r}_{ij}$ (in other words, $\vec{\mu}_j$ is restrained to point towards atom i).

The dipolar torque $\vec{T}_j$ must be counterbalanced in order to conserve the local angular momentum. This is achieved via an additional force couple generating a torque equivalent to the opposite of $\vec{T}_j$:

$$-\vec{T}_j = \vec{r}_{ij} \times \vec{F}_i$$
$$\vec{F}_j = -\vec{F}_i$$

where $\vec{F}_i$ and $\vec{F}_j$ are applied on atoms i and j, respectively.

The following coefficients must be defined for each angle type via the *angle_coeff* command as in the example above, or in the data file or restart files read by the *read_data* or *read_restart* commands:

- K (energy)
- γ_0 (degrees)

Styles with a *gpu*, *intel*, *kk*, *omp*, or *opt* suffix are functionally the same as the corresponding style without the suffix. They have been optimized to run faster, depending on your available hardware, as discussed on the *Accelerator packages* page. The accelerated styles take the same arguments and should produce the same results, except for round-off and precision issues.

These accelerated styles are part of the GPU, INTEL, KOKKOS, OPENMP, and OPT packages, respectively. They are only enabled if LAMMPS was built with those packages. See the *Build package* page for more info.

You can specify the accelerated styles explicitly in your input script by including their suffix, or you can use the *-suffix command-line switch* when you invoke LAMMPS, or you can use the *suffix* command in your input script.

See the *Accelerator packages* page for more instructions on how to use the accelerated styles effectively.

6.14.4 Restrictions

This angle style can only be used if LAMMPS was built with the DIPOLE package. See the *Build package* doc page for more info.

Note

In the "Angles" section of the data file, the atom ID j defining the direction of the dipole vector to restrain must come before the atom ID of the reference atom i. A third atom ID k must also be provided to comply with the requirement of a valid angle definition. This atom ID k should be chosen to be that of an atom bonded to atom i to avoid errors with "lost angle atoms" when running in parallel. Since the LAMMPS code checks for valid angle definitions, cannot use the same atom ID of either i or j (this was allowed and recommended with older LAMMPS versions).

The *newton* command for intramolecular interactions must be "on" (which is the default except when using some accelerator packages).

> **Note**
>
> This angle style should **NOT** be used with fix shake.

6.14.5 Related commands

angle_coeff, *angle_hybrid*

6.14.6 Default

none

(Orsi) Orsi & Essex, The ELBA force field for coarse-grain modeling of lipid membranes, PloS ONE 6(12): e28637, 2011.

(AllenTildesley) Allen & Tildesley, Computer Simulation of Liquids, Clarendon Press, Oxford, 1987.

6.15 angle_style fourier command

Accelerator Variants: *fourier/omp*

6.15.1 Syntax

```
angle_style fourier
```

6.15.2 Examples

```
angle_style fourier
angle_coeff 75.0 1.0 1.0 1.0
```

6.15.3 Description

The *fourier* angle style uses the potential

$$E = K[C_0 + C_1 \cos(\theta) + C_2 \cos(2\theta)]$$

The following coefficients must be defined for each angle type via the *angle_coeff* command as in the example above, or in the data file or restart files read by the *read_data* or *read_restart* commands:

- K (energy)
- C_0 (real)
- C_1 (real)
- C_2 (real)

Styles with a *gpu*, *intel*, *kk*, *omp*, or *opt* suffix are functionally the same as the corresponding style without the suffix. They have been optimized to run faster, depending on your available hardware, as discussed on the *Accelerator packages* page. The accelerated styles take the same arguments and should produce the same results, except for round-off and precision issues.

These accelerated styles are part of the GPU, INTEL, KOKKOS, OPENMP, and OPT packages, respectively. They are only enabled if LAMMPS was built with those packages. See the *Build package* page for more info.

You can specify the accelerated styles explicitly in your input script by including their suffix, or you can use the *-suffix command-line switch* when you invoke LAMMPS, or you can use the *suffix* command in your input script.

See the *Accelerator packages* page for more instructions on how to use the accelerated styles effectively.

6.15.4 Restrictions

This angle style can only be used if LAMMPS was built with the EXTRA-MOLECULE package. See the *Build package* doc page for more info.

6.15.5 Related commands

angle_coeff

6.15.6 Default

none

6.16 angle_style fourier/simple command

Accelerator Variants: *fourier/simple/omp*

6.16.1 Syntax

```
angle_style fourier/simple
```

6.16.2 Examples

```
angle_style fourier/simple
angle_coeff 100.0 -1.0 1.0
```

6.16.3 Description

The *fourier/simple* angle style uses the potential

$$E = K[1.0 + c\cos(n\theta)]$$

The following coefficients must be defined for each angle type via the *angle_coeff* command as in the example above, or in the data file or restart files read by the *read_data* or *read_restart* commands:

- K (energy)
- c (real)
- n (real)

Styles with a *gpu*, *intel*, *kk*, *omp*, or *opt* suffix are functionally the same as the corresponding style without the suffix. They have been optimized to run faster, depending on your available hardware, as discussed on the *Accelerator packages* page. The accelerated styles take the same arguments and should produce the same results, except for round-off and precision issues.

These accelerated styles are part of the GPU, INTEL, KOKKOS, OPENMP, and OPT packages, respectively. They are only enabled if LAMMPS was built with those packages. See the *Build package* page for more info.

You can specify the accelerated styles explicitly in your input script by including their suffix, or you can use the *-suffix command-line switch* when you invoke LAMMPS, or you can use the *suffix* command in your input script.

See the *Accelerator packages* page for more instructions on how to use the accelerated styles effectively.

6.16.4 Restrictions

This angle style can only be used if LAMMPS was built with the EXTRA-MOLECULE package. See the *Build package* doc page for more info.

6.16.5 Related commands

angle_coeff

6.16.6 Default

none

6.17 angle_style gaussian command

6.17.1 Syntax

```
angle_style gaussian
```

6.17.2 Examples

```
angle_style gaussian
angle_coeff 1 300.0 2 0.0128 0.375 80.0 0.0730 0.148 123.0
```

6.17.3 Description

The *gaussian* angle style uses the potential:

$$E = -k_B T ln \left(\sum_{i=1}^{n} \frac{A_i}{w_i \sqrt{\pi/2}} exp \left(\frac{-2(\theta - \theta_i)^2}{w_i^2} \right) \right)$$

This analytical form is a suitable potential for obtaining mesoscale effective force fields which can reproduce target atomistic distributions *(Milano)*.

The following coefficients must be defined for each angle type via the *angle_coeff* command as in the example above, or in the data file or restart files read by the *read_data* or *read_restart* commands:

- T temperature at which the potential was derived
- n (integer >=1)
- A_1 (> 0, radians)
- w_1 (> 0, radians)
- θ_1 (degrees)
- ...
- A_n (> 0, radians)
- w_n (> 0, radians)
- θ_n (degrees)

6.17.4 Restrictions

This angle style can only be used if LAMMPS was built with the EXTRA-MOLECULE package. See the *Build package* doc page for more info.

6.17.5 Related commands

angle_coeff

6.17.6 Default

none

(Milano) G. Milano, S. Goudeau, F. Mueller-Plathe, J. Polym. Sci. B Polym. Phys. 43, 871 (2005).

6.18 angle_style harmonic command

Accelerator Variants: *harmonic/intel*, *harmonic/kk*, *harmonic/omp*

6.18.1 Syntax

```
angle_style harmonic
```

6.18.2 Examples

```
angle_style harmonic
angle_coeff 1 300.0 107.0
```

6.18.3 Description

The *harmonic* angle style uses the potential

$$E = K(\theta - \theta_0)^2$$

where θ_0 is the equilibrium value of the angle, and K is a prefactor. Note that the usual 1/2 factor is included in K.

The following coefficients must be defined for each angle type via the *angle_coeff* command as in the example above, or in the data file or restart files read by the *read_data* or *read_restart* commands:

- K (energy)
- θ_0 (degrees)

θ_0 is specified in degrees, but LAMMPS converts it to radians internally; hence K is effectively energy per radian^2.

Styles with a *gpu*, *intel*, *kk*, *omp*, or *opt* suffix are functionally the same as the corresponding style without the suffix. They have been optimized to run faster, depending on your available hardware, as discussed on the *Accelerator packages* page. The accelerated styles take the same arguments and should produce the same results, except for round-off and precision issues.

These accelerated styles are part of the GPU, INTEL, KOKKOS, OPENMP, and OPT packages, respectively. They are only enabled if LAMMPS was built with those packages. See the *Build package* page for more info.

You can specify the accelerated styles explicitly in your input script by including their suffix, or you can use the *-suffix command-line switch* when you invoke LAMMPS, or you can use the *suffix* command in your input script.

See the *Accelerator packages* page for more instructions on how to use the accelerated styles effectively.

6.18.4 Restrictions

This angle style can only be used if LAMMPS was built with the MOLECULE package. See the *Build package* doc page for more info.

6.18.5 Related commands

angle_coeff

6.18.6 Default

none

6.19 angle_style hybrid command

Accelerator Variants: *hybrid/kk*

6.19.1 Syntax

```
angle_style hybrid style1 style2 ...
```

- style1,style2 = list of one or more angle styles

6.19.2 Examples

```
angle_style hybrid harmonic cosine
angle_coeff 1 harmonic 80.0 30.0
angle_coeff 2* cosine 50.0
```

6.19.3 Description

The *hybrid* style enables the use of multiple angle styles in one simulation. An angle style is assigned to each angle type. For example, angles in a polymer flow (of angle type 1) could be computed with a *harmonic* potential and angles in the wall boundary (of angle type 2) could be computed with a *cosine* potential. The assignment of angle type to style is made via the *angle_coeff* command or in the data file.

In the *angle_coeff* commands, the name of an angle style must be added after the angle type, with the remaining coefficients being those appropriate to that style. In the example above, the 2 angle_coeff commands set angles of angle type 1 to be computed with a *harmonic* potential with coefficients 80.0, 30.0 for K, θ_0. All other angle types $(2 - N)$ are computed with a *cosine* potential with coefficient 50.0 for K.

If angle coefficients are specified in the data file read via the *read_data* command, then the same rule applies. E.g. "harmonic" or "cosine", must be added after the angle type, for each line in the "Angle Coeffs" section, e.g.

```
Angle Coeffs

1 harmonic 80.0 30.0
2 cosine 50.0
...
```

If *class2* is one of the angle hybrid styles, the same rule holds for specifying additional BondBond (and BondAngle) coefficients either via the input script or in the data file. I.e. *class2* must be added to each line after the angle type. For lines in the BondBond (or BondAngle) section of the data file for angle types that are not *class2*, you must use an angle style of *skip* as a placeholder, e.g.

```
BondBond Coeffs

1 skip
2 class2 3.6512 1.0119 1.0119
...
```

Note that it is not necessary to use the angle style *skip* in the input script, since BondBond (or BondAngle) coefficients need not be specified at all for angle types that are not *class2*.

An angle style of *none* with no additional coefficients can be used in place of an angle style, either in a input script *angle_coeff* command or in the data file, if you desire to turn off interactions for specific angle types.

Styles with a *gpu*, *intel*, *kk*, *omp*, or *opt* suffix are functionally the same as the corresponding style without the suffix. They have been optimized to run faster, depending on your available hardware, as discussed on the *Accelerator packages* page. The accelerated styles take the same arguments and should produce the same results, except for round-off and precision issues.

These accelerated styles are part of the GPU, INTEL, KOKKOS, OPENMP, and OPT packages, respectively. They are only enabled if LAMMPS was built with those packages. See the *Build package* page for more info.

You can specify the accelerated styles explicitly in your input script by including their suffix, or you can use the *-suffix command-line switch* when you invoke LAMMPS, or you can use the *suffix* command in your input script.

See the *Accelerator packages* page for more instructions on how to use the accelerated styles effectively.

6.19.4 Restrictions

This angle style can only be used if LAMMPS was built with the MOLECULE package. See the *Build package* doc page for more info.

Unlike other angle styles, the hybrid angle style does not store angle coefficient info for individual sub-styles in *binary restart files* or *data files*. Thus when restarting a simulation, you need to re-specify the angle_coeff commands.

6.19.5 Related commands

angle_coeff

6.19.6 Default

none

6.20 angle_style lepton command

Accelerator Variants: *lepton/omp*

6.20.1 Syntax

```
angle_style style args
```

- style = *lepton*
- args = optional arguments

args = auto_offset or no_offset
  auto_offset = offset the potential energy so that the value at theta0 is 0.0 (default)
  no_offset = do not offset the potential energy

6.20.2 Examples

```
angle_style lepton
angle_style lepton no_offset

angle_coeff  1  120.0  "k*theta^2; k=250.0"
angle_coeff  2   90.0  "k2*theta^2 + k3*theta^3 + k4*theta^4; k2=300.0; k3=-100.0; k4=50.0"
angle_coeff  3  109.47 "k*theta^2; k=350.0"
```

6.20.3 Description

Added in version 8Feb2023.

Angle style *lepton* computes angular interactions between three atoms with a custom potential function. The potential function must be provided as an expression string using "theta" as the angle variable relative to the reference angle θ_0 which is provided as an angle coefficient. For example *"200.0*theta^2"* represents a *harmonic angle* potential with a force constant K of 200.0 energy units:

$$U_{angle,i} = K(\theta_i - \theta_0)^2 = K\theta^2 \qquad \theta = \theta_i - \theta_0$$

Changed in version 7Feb2024.

By default the potential energy U is shifted so that the value U is 0.0 for $theta = theta_0$. This is equivalent to using the optional keyword *auto_offset*. When using the keyword *no_offset* instead, the potential energy is not shifted.

The Lepton library, that the *lepton* angle style interfaces with, evaluates this expression string at run time to compute the pairwise energy. It also creates an analytical representation of the first derivative of this expression with respect to "theta" and then uses that to compute the force between the angle atoms as defined by the topology data.

The following coefficients must be defined for each angle type via the *angle_coeff* command as in the example above, or in the data file or restart files read by the *read_data* or *read_restart* commands:

- Lepton expression (energy units)
- θ_0 (degrees)

The Lepton expression must be either enclosed in quotes or must not contain any whitespace so that LAMMPS recognizes it as a single keyword. More on valid Lepton expressions below. The θ_0 coefficient is the "equilibrium angle". It is entered in degrees, but internally converted to radians. Thus the expression must assume "theta" is in radians. The potential energy function in the Lepton expression is shifted in such a way, that the potential energy is 0 for a angle $\theta_i == \theta_0$.

6.20.4 Lepton expression syntax and features

Lepton supports the following operators in expressions:

+	Add	-	Subtract	*	Multiply	/	Divide	^	Power

The following mathematical functions are available:

sqrt(x)	Square root	exp(x)	Exponential
log(x)	Natural logarithm	sin(x)	Sine (angle in radians)
cos(x)	Cosine (angle in radians)	sec(x)	Secant (angle in radians)
csc(x)	Cosecant (angle in radians)	tan(x)	Tangent (angle in radians)
cot(x)	Cotangent (angle in radians)	asin(x)	Inverse sine (in radians)
acos(x)	Inverse cosine (in radians)	atan(x)	Inverse tangent (in radians)
sinh(x)	Hyperbolic sine	cosh(x)	Hyperbolic cosine
tanh(x)	Hyperbolic tangent	erf(x)	Error function
erfc(x)	Complementary Error function	abs(x)	Absolute value
min(x,y)	Minimum of two values	max(x,y)	Maximum of two values
delta(x)	delta(x) is 1 for $x = 0$, otherwise 0	step(x)	step(x) is 0 for $x < 0$, otherwise 1

Numbers may be given in either decimal or exponential form. All of the following are valid numbers: *5*, *-3.1*, *1e6*, and *3.12e-2*.

As an extension to the standard Lepton syntax, it is also possible to use LAMMPS *variables* in the format "v_name". Before evaluating the expression, "v_name" will be replaced with the value of the variable "name". This is compatible with all kinds of scalar variables, but not with vectors, arrays, local, or per-atom variables. If necessary, a custom scalar variable needs to be defined that can access the desired (single) item from a non-scalar variable. As an example, the following lines will instruct LAMMPS to ramp the force constant for a harmonic bond from 100.0 to 200.0 during the next run:

```
variable fconst equal ramp(100.0, 200)
bond_style lepton
bond_coeff 1 1.5 "v_fconst * (r^2)"
```

An expression may be followed by definitions for intermediate values that appear in the expression. A semicolon ";" is used as a delimiter between value definitions. For example, the expression:

```
a^2+a*b+b^2; a=a1+a2; b=b1+b2
```

is exactly equivalent to

```
(a1+a2)^2+(a1+a2)*(b1+b2)+(b1+b2)^2
```

The definition of an intermediate value may itself involve other intermediate values. Whitespace and quotation characters ('' and "") are ignored. All uses of a value must appear *before* that value's definition. For efficiency reasons, the expression string is parsed, optimized, and then stored in an internal, pre-parsed representation for evaluation.

Evaluating a Lepton expression is typically between 2.5 and 5 times slower than the corresponding compiled and optimized C++ code. If additional speed or GPU acceleration (via GPU or KOKKOS) is required, the interaction can be represented as a table. Suitable table files can be created either internally using the *pair_write* or *bond_write* command or through the Python scripts in the *tools/tabulate* folder.

Styles with a *gpu*, *intel*, *kk*, *omp*, or *opt* suffix are functionally the same as the corresponding style without the suffix. They have been optimized to run faster, depending on your available hardware, as discussed on the *Accelerator packages* page. The accelerated styles take the same arguments and should produce the same results, except for round-off and precision issues.

These accelerated styles are part of the GPU, INTEL, KOKKOS, OPENMP, and OPT packages, respectively. They are only enabled if LAMMPS was built with those packages. See the *Build package* page for more info.

You can specify the accelerated styles explicitly in your input script by including their suffix, or you can use the *-suffix command-line switch* when you invoke LAMMPS, or you can use the *suffix* command in your input script.

See the *Accelerator packages* page for more instructions on how to use the accelerated styles effectively.

6.20.5 Restrictions

This angle style is part of the LEPTON package and only enabled if LAMMPS was built with this package. See the *Build package* page for more info.

6.20.6 Related commands

angle_coeff , *angle_style table*, *bond_style lepton*,:doc:*dihedral_style lepton <dihedral_lepton>*

6.20.7 Default

none

6.21 angle_style mesocnt command

6.21.1 Syntax

```
angle_style mesocnt
```

6.21.2 Examples

```
angle_style mesocnt
angle_coeff 1 buckling C 10 10 20.0
angle_coeff 4 harmonic C 8 4 10.0
angle_coeff 2 buckling custom 400.0 50.0 5.0
angle_coeff 1 harmonic custom 300.0
```

6.21.3 Description

Added in version 15Sep2022.

The *mesocnt* angle style uses the potential

$$E = K_{\mathrm{H}}\Delta\theta^2, \qquad |\Delta\theta| < \Delta\theta_{\mathrm{B}}$$
$$E = K_{\mathrm{H}}\Delta\theta_{\mathrm{B}}^2 + K_{\mathrm{B}}(\Delta\theta - \Delta\theta_{\mathrm{B}}), \qquad |\Delta\theta| \geq \Delta\theta_{\mathrm{B}}$$

where $\Delta\theta = \theta - \pi$ is the bending angle of the nanotube, K_{H} and K_{B} are prefactors for the harmonic and linear regime respectively and $\Delta\theta_{\mathrm{B}}$ is the buckling angle. Note that the usual 1/2 factor for the harmonic potential is included in K_{H}.

The style implements parameterization presets of K_{H}, K_{B} and $\Delta\theta_{\mathrm{B}}$ for mesoscopic simulations of carbon nanotubes based on the atomistic simulations of *(Srivastava)* and buckling considerations of *(Zhigilei)*.

The following coefficients must be defined for each angle type via the *angle_coeff* command as in the examples above, or in the data file or restart files read by the *read_data* or *read_restart* commands:

- mode = *buckling* or *harmonic*
- preset = *C* or *custom*
- additional parameters depending on preset

If mode *harmonic* is chosen, the potential is simply harmonic and does not switch to the linear term when the buckling angle is reached. In *buckling* mode, the full piecewise potential is used.

Preset *C* is for carbon nanotubes, and the additional parameters are:

- chiral index n (unitless)
- chiral index m (unitless)
- r_0 (distance)

Here, r_0 is the equilibrium distance of the bonds included in the angle, see *bond_style mesocnt*.

In harmonic mode with preset *custom*, the additional parameter is:

- K_{H} (energy)

Hence, this setting is simply a wrapper for *bond_style harmonic* with an equilibrium angle of 180 degrees.

In harmonic mode with preset *custom*, the additional parameters are:

- K_H (energy)
- K_B (energy)
- $\Delta\theta_B$ (degrees)

$\Delta\theta_B$ is specified in degrees, but LAMMPS converts it to radians internally; hence K_H is effectively energy per radian^2 and K_B is energy per radian.

In *buckling* mode, this angle style adds the *buckled* property to all atoms in the simulation, which is an integer flag indicating whether the bending angle at a given atom has exceeded $\Delta\theta_B$. It can be accessed as an atomic variable, e.g. for custom dump commands, as *i_buckled*.

Note

If the initial state of the simulation contains buckled nanotubes and *pair_style mesocnt* is used, the *i_buckled* atomic variable needs to be initialized before the pair_style is defined by doing a *run 0* command straight after the angle_style command. See below for an example.

If CNTs are already buckled at the start of the simulation, this script will correctly initialize *i_buckled*:

```
angle_style mesocnt
angle_coeff 1 buckling C 10 10 20.0

run 0

pair_style mesocnt 60.0
pair_coeff * * C_10_10.mesocnt 1
```

6.21.4 Restrictions

This angle style can only be used if LAMMPS was built with the MOLECULE and MESONT packages. See the *Build package* doc page for more info.

6.21.5 Related commands

angle_coeff

6.21.6 Default

none

(Srivastava) Zhigilei, Wei, Srivastava, Phys. Rev. B 71, 165417 (2005).

(Zhigilei) Volkov and Zhigilei, ACS Nano 4, 6187 (2010).

6.22 angle_style mm3 command

6.22.1 Syntax

```
angle_style mm3
```

6.22.2 Examples

```
angle_style mm3
angle_coeff 1 100.0 107.0
```

6.22.3 Description

The *mm3* angle style uses the potential that is anharmonic in the angle as defined in *(Allinger)*

$$E = K(\theta - \theta_0)^2 \left[1 - 0.014(\theta - \theta_0) + 5.6(10)^{-5}(\theta - \theta_0)^2 - 7.0(10)^{-7}(\theta - \theta_0)^3 + 9(10)^{-10}(\theta - \theta_0)^4\right]$$

where θ_0 is the equilibrium value of the angle, and K is a prefactor. The anharmonic prefactors have units deg^{-n}, for example $-0.014\,\mathrm{deg}^{-1}$, $5.6 \cdot 10^{-5}\,\mathrm{deg}^{-2}$, ...

The following coefficients must be defined for each angle type via the *angle_coeff* command as in the example above, or in the data file or restart files read by the *read_data* or *read_restart* commands:

- K (energy)
- θ_0 (degrees)

θ_0 is specified in degrees, but LAMMPS converts it to radians internally; hence K is effectively energy per radian^2.

6.22.4 Restrictions

This angle style can only be used if LAMMPS was built with the YAFF package. See the *Build package* doc page for more info.

6.22.5 Related commands

angle_coeff

6.22.6 Default

none

6.23 angle_style none command

6.23.1 Syntax

```
angle_style none
```

6.23.2 Examples

```
angle_style none
```

6.23.3 Description

Using an angle style of none means angle forces and energies are not computed, even if triplets of angle atoms were listed in the data file read by the *read_data* command.

See the *angle_style zero* command for a way to calculate angle statistics, but compute no angle interactions.

6.23.4 Restrictions

none

6.23.5 Related commands

angle_style zero

6.23.6 Default

none

6.24 angle_style quartic command

Accelerator Variants: *quartic/omp*

6.24.1 Syntax

```
angle_style quartic
```

6.24.2 Examples

```
angle_style quartic
angle_coeff 1 129.1948 56.8726 -25.9442 -14.2221
```

6.24.3 Description

The *quartic* angle style uses the potential

$$E = K_2(\theta - \theta_0)^2 + K_3(\theta - \theta_0)^3 + K_4(\theta - \theta_0)^4$$

where θ_0 is the equilibrium value of the angle, and K is a prefactor. Note that the usual 1/2 factor is included in K.

The following coefficients must be defined for each angle type via the *angle_coeff* command as in the example above, or in the data file or restart files read by the *read_data* or *read_restart* commands:

- θ_0 (degrees)
- K_2 (energy)
- K_3 (energy)
- K_4 (energy)

θ_0 is specified in degrees, but LAMMPS converts it to radians internally; hence the various K are effectively energy per radian^2 or radian^3 or radian^4.

Styles with a *gpu*, *intel*, *kk*, *omp*, or *opt* suffix are functionally the same as the corresponding style without the suffix. They have been optimized to run faster, depending on your available hardware, as discussed on the *Accelerator packages* page. The accelerated styles take the same arguments and should produce the same results, except for round-off and precision issues.

These accelerated styles are part of the GPU, INTEL, KOKKOS, OPENMP, and OPT packages, respectively. They are only enabled if LAMMPS was built with those packages. See the *Build package* page for more info.

You can specify the accelerated styles explicitly in your input script by including their suffix, or you can use the *-suffix command-line switch* when you invoke LAMMPS, or you can use the *suffix* command in your input script.

See the *Accelerator packages* page for more instructions on how to use the accelerated styles effectively.

6.24.4 Restrictions

This angle style can only be used if LAMMPS was built with the EXTRA-MOLECULE package. See the *Build package* doc page for more info.

6.24.5 Related commands

angle_coeff

6.24.6 Default

none

6.25 angle_style spica command

Accelerator Variants: *spica/omp*, *spica/kk*

6.25.1 Syntax

```
angle_style spica

angle_style spica/omp
```

6.25.2 Examples

```
angle_style spica
angle_coeff 1 300.0 107.0
```

6.25.3 Description

The *spica* angle style is a combination of the harmonic angle potential,

$$E = K(\theta - \theta_0)^2$$

where θ_0 is the equilibrium value of the angle and K a prefactor, with the *repulsive* part of the non-bonded *lj/spica* pair style between the atoms 1 and 3. This angle potential is intended for coarse grained MD simulations with the SPICA (formerly called SDK) parameterization using the *pair_style lj/spica*. Relative to the pair_style *lj/spica*, however, the energy is shifted by ε, to avoid sudden jumps. Note that the usual 1/2 factor is included in K.

The following coefficients must be defined for each angle type via the *angle_coeff* command as in the example above:

- K (energy)
- θ_0 (degrees)

θ_0 is specified in degrees, but LAMMPS converts it to radians internally; hence K is effectively energy per radian^2.

The required *lj/spica* parameters are extracted automatically from the pair_style.

Style *sdk*, the original implementation of style *spica*, is available for backward compatibility.

Styles with a *gpu*, *intel*, *kk*, *omp*, or *opt* suffix are functionally the same as the corresponding style without the suffix. They have been optimized to run faster, depending on your available hardware, as discussed on the *Accelerator packages*

page. The accelerated styles take the same arguments and should produce the same results, except for round-off and precision issues.

These accelerated styles are part of the GPU, INTEL, KOKKOS, OPENMP, and OPT packages, respectively. They are only enabled if LAMMPS was built with those packages. See the *Build package* page for more info.

You can specify the accelerated styles explicitly in your input script by including their suffix, or you can use the *-suffix command-line switch* when you invoke LAMMPS, or you can use the *suffix* command in your input script.

See the *Accelerator packages* page for more instructions on how to use the accelerated styles effectively.

6.25.4 Restrictions

This angle style can only be used if LAMMPS was built with the CG-SPICA package. See the *Build package* doc page for more info.

6.25.5 Related commands

angle_coeff, *angle_style harmonic*, *pair_style lj/spica*, *pair_style lj/spica/coul/long*

6.25.6 Default

none

6.26 angle_style table command

Accelerator Variants: *table/omp*

6.26.1 Syntax

```
angle_style table style N
```

- style = *linear* or *spline* = method of interpolation
- N = use N values in table

6.26.2 Examples

```
angle_style table linear 1000
angle_coeff 3 file.table ENTRY1
```

6.26.3 Description

Style *table* creates interpolation tables of length *N* from angle potential and derivative values listed in a file(s) as a function of angle The files are read by the *angle_coeff* command.

The interpolation tables are created by fitting cubic splines to the file values and interpolating energy and derivative values at each of *N* angles. During a simulation, these tables are used to interpolate energy and force values on individual atoms as needed. The interpolation is done in one of 2 styles: *linear* or *spline*.

For the *linear* style, the angle is used to find 2 surrounding table values from which an energy or its derivative is computed by linear interpolation.

For the *spline* style, a cubic spline coefficients are computed and stored at each of the *N* values in the table. The angle is used to find the appropriate set of coefficients which are used to evaluate a cubic polynomial which computes the energy or derivative.

The following coefficients must be defined for each angle type via the *angle_coeff* command as in the example above.

- filename
- keyword

The filename specifies a file containing tabulated energy and derivative values. The keyword specifies a section of the file. The format of this file is described below.

Suitable tables for use with this angle style can be created by LAMMPS itself from existing angle styles using the *angle_write* command. This can be useful to have a template file for testing the angle style settings and to build a compatible custom file. Another option to generate tables is the Python code in the tools/tabulate folder of the LAMMPS source code distribution.

The format of a tabulated file is as follows (without the parenthesized comments):

```
# Angle potential for harmonic (one or more comment or blank lines)

HAM                          (keyword is the first text on line)
N 181 FP 0 0 EQ 90.0           (N, FP, EQ parameters)
                           (blank line)
1 0.0 200.5 2.5              (index, angle, energy, derivative)
2 1.0 198.0 2.5
...
181 180.0 0.0 0.0
```

A section begins with a non-blank line whose first character is not a "#"; blank lines or lines starting with "#" can be used as comments between sections. The first line begins with a keyword which identifies the section. The line can contain additional text, but the initial text must match the argument specified in the *angle_coeff* command. The next line lists (in any order) one or more parameters for the table. Each parameter is a keyword followed by one or more numeric values.

The parameter "N" is required and its value is the number of table entries that follow. Note that this may be different than the *N* specified in the *angle_style table* command. Let Ntable = *N* in the angle_style command, and Nfile = "N" in the tabulated file. What LAMMPS does is a preliminary interpolation by creating splines using the Nfile tabulated values as nodal points. It uses these to interpolate as needed to generate energy and derivative values at Ntable different points. The resulting tables of length Ntable are then used as described above, when computing energy and force for individual angles and their atoms. This means that if you want the interpolation tables of length Ntable to match exactly what is in the tabulated file (with effectively no preliminary interpolation), you should set Ntable = Nfile.

The "FP" parameter is optional. If used, it is followed by two values fplo and fphi, which are the second derivatives at the innermost and outermost angle settings. These values are needed by the spline construction routines. If not

specified by the "FP" parameter, they are estimated (less accurately) by the first two and last two derivative values in the table.

The "EQ" parameter is also optional. If used, it is followed by a the equilibrium angle value, which is used, for example, by the *fix shake* command. If not used, the equilibrium angle is set to 180.0.

Following a blank line, the next N lines list the tabulated values. On each line, the first value is the index from 1 to N, the second value is the angle value (in degrees), the third value is the energy (in energy units), and the fourth is -dE/d(theta) (also in energy units). The third term is the energy of the 3-atom configuration for the specified angle. The last term is the derivative of the energy with respect to the angle (in degrees, not radians). Thus the units of the last term are still energy, not force. The angle values must increase from one line to the next. The angle values must also begin with 0.0 and end with 180.0, i.e. span the full range of possible angles.

Note that one file can contain many sections, each with a tabulated potential. LAMMPS reads the file section by section until it finds one that matches the specified keyword.

Styles with a *gpu*, *intel*, *kk*, *omp*, or *opt* suffix are functionally the same as the corresponding style without the suffix. They have been optimized to run faster, depending on your available hardware, as discussed on the *Accelerator packages* page. The accelerated styles take the same arguments and should produce the same results, except for round-off and precision issues.

These accelerated styles are part of the GPU, INTEL, KOKKOS, OPENMP, and OPT packages, respectively. They are only enabled if LAMMPS was built with those packages. See the *Build package* page for more info.

You can specify the accelerated styles explicitly in your input script by including their suffix, or you can use the *-suffix command-line switch* when you invoke LAMMPS, or you can use the *suffix* command in your input script.

See the *Accelerator packages* page for more instructions on how to use the accelerated styles effectively.

6.26.4 Restart, fix_modify, output, run start/stop, minimize info

This angle style writes the settings for the "angle_style table" command to *binary restart files*, so a angle_style command does not need to specified in an input script that reads a restart file. However, the coefficient information is not stored in the restart file, since it is tabulated in the potential files. Thus, angle_coeff commands do need to be specified in the restart input script.

6.26.5 Restrictions

This angle style can only be used if LAMMPS was built with the MOLECULE package. See the *Build package* doc page for more info.

6.26.6 Related commands

angle_coeff , *angle_write*

6.26.7 Default

none

6.27 angle_style zero command

6.27.1 Syntax

```
angle_style zero keyword
```

- zero or more keywords may be appended
- keyword = *nocoeff*

6.27.2 Examples

```
angle_style zero
angle_style zero nocoeff
angle_coeff *
angle_coeff * 120.0
```

6.27.3 Description

Using an angle style of zero means angle forces and energies are not computed, but the geometry of angle triplets is still accessible to other commands.

As an example, the *compute angle/local* command can be used to compute the theta values for the list of triplets of angle atoms listed in the data file read by the *read_data* command. If no angle style is defined, this command cannot be used.

The optional *nocoeff* flag allows to read data files with AngleCoeff section for any angle style. Similarly, any *angle_coeff* commands will only be checked for the angle type number and the rest ignored.

Note that the *angle_coeff* command must be used for all angle types. If specified, there can be only one value, which is going to be used to assign an equilibrium angle, e.g. for use with *fix shake*.

6.27.4 Restrictions

none

6.27.5 Related commands

angle_style none

6.27.6 Default

none