Spider Help File

This document is a printable version of the *Spider* online Help file. It was last modified April 2001.

Table of Contents

Introduction

Spider and Kalypso

Spider is one of 3 programs which constitute the core of the Kalypso package (the other main programs are **Kalypso** and **Winnow**). The function of Spider is to create the input files needed by the simulation engine, Kalypso. There are 5 or 6 such input files for each simulation project. Their designations and file name extensions are as follows: Target file (*.TRG), Projectile file (*.PRJ), Model file (*.MDL), Run file (*.RUN), Impact file (*.IMP) and (optionally) Inelastic file (*.INL).

Help Features

The Contents button on any Help window allows you to navigate to a specific topic, while the [>>] and [<<] buttons allow you to browse sequentially through all topics in the Help file. The Index button brings up a list of keywords. You can cut and paste Help topics into other Windows applications (use Ctrl-V to paste data into Spider's edit boxes) via the Help window menu, which also allows printing.

Guidelines

Here are some guidelines to follow when creating a simulation project (i.e. a set of 6 input files which will be used by Kalypso).

- First, use the same root for all 6 filenames if possible (TEST.TRG, TEST.PRJ etc.), and create a new directory to store them in.
- Second, begin at the leftmost menu (#1. Target) and work your way to the right (#6. Inelastic).
- Third, work slowly and make good use of the context-sensitive on-line Help. For example, if you position the cursor on an edit box, you can hit **F1** to get more information about the input which is required (and, where relevant, links to lists of representative data).

Using Dialog Boxes

The user supplies data to Spider via a number of dialog boxes, which contain editable input lines, as well as various kinds of 'check-boxes', which are used to select options. There may be an associated set of input lines which are 'enabled' or 'disabled' according to the state of the check-boxes (checked or unchecked). When an input line is disabled (dimmed), this means that the parameter it reflects is irrelevant to your simulation configuration.

When you are satisfied with your data entry, click the Ok button to initiate the procedure to save the data to disk. If you want to abandon the data in the current dialog, and exit without saving, click the Cancel button.

Numeric data entered into input boxes are checked for integrity (format, range) only when focus shifts to another item in the dialog box. An error message will result if the datum has an invalid numeric format (such as '3.1.340') or is out of range. This error must be corrected before you can proceed, even if your next desired action is to click the Cancel button (and abandon the data you entered).

In setting the allowed ranges, the author has in some cases allowed the user to enter physically implausible values; but in general these ranges reflect the physics of the problem. The idea is to protect users from the consequences of incorrect data inputs.

Atomic Data

(expand window to view; see footnote for symbols)

Element	Sym.	Mass (amu)	Z	MP (°C)	Structure, Lattice Consts. (A)
Actinium	Ac	(227)	89	1050	fcc 5.311 (R.T.)
Aluminium	A1	26.98	13	660	fcc 4.0495, (25°C)
Americium	Am	(243)	95	850	hex ABAC 3.642, 11.76 (R.T.)
Antimony	Sb	121.75	51	630	R3m 4.5067, 57°6.5' (25°C)
Argon	Ar	39.95	18	-189	fcc 5.42 (-233°C)
Arsenic	As	74.92	33	814	R3m 4.131, 54°10' (25°C)
Astatine	At	(210)	85	---	---
Barium	Ba	137.34	56	710	bcc 5.019 (R.T.)
Berkelium	Bk	(247)	97	---	---
Beryllium	Be	9.01	4	1280	hcp 2.2856, 3.5832 (25°C)
Bismuth	Bi	208.98	83	271	R3m 4.7457, 57°14.2' (31°C)
Boron	B	10.81	5	2300	tet 8.73, 5.03 (R.T.)
Bromine	Br	79.91	35	-7	ort 4.48, 6.67, 8.72 (-150°C)
Cadmium	Cd	112.40	48	321	hcp 2.9788, 5.6167 (21°C)
Caesium	Cs	132.91	55	29	bcc 6.14 (–10°C)
Calcium	Ca	40.08	20	850	fcc 5.582 (18°C)
Californium	Cf	(249)	98	---	---
Carbon	C	12.01	6	3500	hex graphite 2.4612, 6.7079 (R.T.)
Cerium	Ce	140.12	58	804	fcc 5.1604 (20°C)
Chlorine	C1	35.46	17	-101	tet 8.56, 6.12 (-185°C)
Chromium	Cr	52.00	24	1900	bcc 2.8846 (20°C)
Cobalt	Co	58.93	27	1492	hcp 2.507, 4.069 (R.T.)
Copper	Cu	63.54	29	1083	fcc 3.6147 (20°C)
Curium	Cm	(247)	96	---	---
Dysprosium	Dy	162.50	66	1500	hcp 3.5923, 5.6545 (20°C)

Einsteinium	Es	(254)	99	---	---	
Erbium	Er	167.26	68	1525	hcp	3.559, 5.592 (20°C)
Europium	Eu	151.96	63	900	bcc	4.578 (20°C)
Fermium	Fm	(253)	100	---	---	
Fluorine	F	19.00	9	-220	---	
Francium	Fr	223	87	---	---	
Gadolinium	Gd	157.25	64	1320	hcp	3.6315, 5.777 (20 C)
Gallium	Ga	69.72	31	30	ort	4.520, 7.661, 4.526 (20°C)
Germanium	Ge	72.59	32	958	dia	5.6575 (20°C)
Gold	Au	196.97	79	1063	fcc	4.0783 (25°C)
Hafnium	Hf	178.49	72	2000	hcp	3.1946, 5.0511 (24°C)
Helium	He	4.00	2	-270	---	
Holmium	Ho	164.93	67	1500	hcp	3.5761, 5.6174 (20°C)
Hydrogen	H	1.01	1	-259	hex	3.75, 6.12 (-271°C)
Indium	In	114.82	49	156	tet	3.2512, 4.9467 (20°C)
iodine	I	126.90	53	114	ort	4.792, 7.271, 9.773 (R.T.)
Iridium	Ir	192.22	77	2443	fcc	3.8389 (R.T.)
Iron	Fe	55.85	26	1539	bcc	2.8664 (20°C)
Krypton	Kr	83.80	36	-157	fcc	5.68 (-191°C)
Lanthanum	La	138.91	57	920	hex	ABAC 3.770, 12.131 (20°C)
Lawrencium	Lr	(257)	103	---	---	
Lead	Pb	207.19	82	327	fcc	4.9502 (25°C)
Lithium	Li	6.94	3	180	bcc	3.5092 (20°C)
Lutetium	Lu	174.97	71	1700	hcp	3.5050, 5.5486 (20°C)
Magnesium	Mg	24.31	12	650	hcp	3.2094, 5.2105 (25°C)
Manganese	Mn	54.94	25	1250	cub	8.914 (25°C)
Mendelevium	Md	(256)	101	---	---	
Mercury	Hg	200.59	80	-39	R3m	3.005, 70°32 (-46°C)
Molybdenum	Mo	95.94	42	2620	bcc	3.1469 (20°C)
Neodymium	Nd	144.24	60	1024	hex	ABAC 3.6582, 11.802 (20°C)
Neon	Ne	20.18	10	-249	fcc	4.52 (-268°C)
Neptunium	Np	(237)	93	640	ort	4.723, 4.887, 6.663 (20°C)
Nickel	Ni	58.71	28	1453	fcc	3.524 (18°C)
Niobium	Nb	92.91	41	2420	bcc	3.006 (20°C)
Nitrogen	N	14.01	7	-210	hex	4.03, 6.59 (-234°C)
Nobelium	No	(253)	102	---	---	
Osmium	Os	190.20	76	2700	hcp	2.7353, 4.3191 (20°C)
Oxygen	0	16.00	8	-219	cub	6.83 (-225°C)
Palladium	Pd	106.40	46	1552	fcc	3.8907 (22°C)
Phosphorus	P	30.97	15	44	ort	3.32, 10.52, 4.39 (black) (R.T.)
Platinum	Pt	195.09	78	1769	fcc	3.9239 (20°C)
Plutonium	Pu	(242)	94	-	mon	6.18, 4.82, 10.97, 101.81° (21°C)
Polonium	Po	(210)	84	254	cub	3.345 (10°C)
Potassium	K	39.10	19	63	bcc	5.32 (20°C)
Praseodymium	Pr	140.91	59	935	hex	ABAC 3.6702, 11.828 (20°C)
Promethium	Pm	(147)	61	---	---	
Protactinium	Pa	(231)	91	3000	tet	3.935, 3.238 (R.T.)
Radium	Ra	(226)	88	700	---	
Radon	Rn	(222)	86	-71	---	
Rhenium	Re	186.23	75	3170	hcp	2.760, 4.458 (R.T.)
Rhodium	Rh	102.91	45	1960	fcc	3.8044 (20°C)
Rubidium	Rb	85.47	37	39	bcc	5.70 (20°C)
Ruthenium	Ru	101.07	44	2400	hcp	2.7058, 4.2816 (25°C)
Samarium	Sm	150.35	62	1052	R	8.996, 23°13' (20°C)
Scandium	Sc	44.96	21	1400	hcp	3.3080, 5.2653 (20°C)
Selenium	Se	78.96	34	217	hex	4.3656, 4.9590 (25°C)
Silicon	Si	28.09	14	1410	dia	5.4305 (R.T.)
Silver	Ag	107.87	47	961	fcc	4.0857 (20°C)
Sodium	Na	22.99	11	98	bcc	4.2906 (20°C)
Strontium	Sr	87.62	38	770	fcc	6.0849 (25°C)
Sulphur	S	32.06	16	119	ort	10.414, 10.845, 24.369 (R.T.)
Tantalum	Ta	180.95	73	3000	bcc	3.3026 (20°C)
Technetium	Tc	98.91	43	2700	hcp	2.735, 4.388 (R.T.)
Tellurium	Te	127.60	52	450	hex	4.4566, 5.9268 (25°C)
Terbium	Tb	158.93	65	1450	hcp	3.599, 5.696 (20°C)
Thallium	Tl	204.37	81	304	hcp	3.4566, 5.5248 (18°C)
Thorium	Th	232.04	90	170O	fcc	5.0843 (R.T.)
Thulium	Tm	168.93	69	1600	hcp	3.5372, 5.5619 (20°C)
Tin	Sn	118.69	50	232	tet	5.8315, 3.1814 (25°C)
Titanium	Ti	47.90	22	1680	hcp	2.9506, 4.6788 (25°C)
Tungsten	W	183.85	74	3380	bcc	3.1650 (25°C)
Uranium	U	238.03	92	1133	ort	2.854, 5.869, 4.955 (27°C)
Radium	Y	50.94	23	1920	bcc	3.028 (30°C)
Xenon	Xe	131.30	54	-112	fcc	6.24 (-185°C)
Ytterbium	Yb	173.04	70	824	fcc	5.481 (20°C)
Yttrium	Y	88.91	39	1500	hcp	3.6451, 5.7305 (20°C)

```
Zinc          Zn  65.37     30  419   hcp  2.6649, 4.9468 (25°C)
Zirconium     Zr  91.22     40  1850  hcp  3.2312, 5.1477 (25°C)
```

This compilation is based on a scan of Appendix A9 of A.R. West, Solid State Chemistry and its Applications (J. Wiley, Singapore etc., 1989).

Atomic mass data are based on the relative atomic mass of 12C = 12.00. These atomic weights are for natural isotopic abundances, except for those given in brackets which refer to short-lived elements without a natural abundance. For these, the mass of the isotope with longest half-life is given.

Crystallographic data originate mainly from International Tables for X-ray Crystallography, Vol. III, p. 278.

R.T. = Room Temperature; fcc = face centred cubic (cubic close packed); hex = hexagonal; hcp = hexagonal close packed; R3m = rhombohedral; bcc = body centred cubic; tet = tetragonal; ort = orthorhombic; cub = cubic; mon = monoclinic; dia = diamond structure.

Surface Debeye Temperatures (Td)

For surfaces/elements for which data are not available (from the table below, or from the LEED literature), use the following rough (and not particularly good!) estimate:

Td(surf) = Td(bulk)/sqrt(2),

for both perpendicular (T) and parallel components, or a 'guesstimate' of your own devising.

For (110) and (111) surfaces, the two parallel (||) surface Debeye temperatures are in general not identical. For the (110) case, differences can be substantial. Kalypso can only work with one || surface Debeye temperature, so you'll have to make a decision about selecting some average value.

```
=============================================
SURF. DEBEYE          D.P. Jackson
TEMPERATURES (K)      Surface Sci. 43 (1974) 431
=============================================
Surface:   (100)       (110)          (111)

           T    ||   T    ||   ||   T    ||   ||

=============================================
Ag       142  203  142  209  137  147  226  226
Al       223  361  223  376  210  221  407  407
Au       110  152  111  156  106  116  168  168
Cr       249  266  255  428  365  252  263  263
Cu       192  292  191  303  181  196  328  328
Fe       227  250  225  399  342  224  236  236
Mo       239  247  250  402  340  248  255  256
Na        79  110   51  168  149   83   78   78
Ni       225  347  225  360  212  230  389  389
Pb       56    78  54    81   54   58   87   87
Pd      189   265  191  273  182  198  294  294
Pt      166   234  167  240  160  174  259  259
V       235   265  227  420  363  229  241  241
W       181   191  186  308  263  184  191  191
=============================================
```

Bulk Debeye Temperatures (T_d/K)

Source: W. Eckstein, 'Computer Simulation of Ion-Surface Interactions, 'Springer-Verlag (1991).

Be	1440	Y	280
B	----	Zr	291
C	420 (graphite), 2230 (diamond)	Nb	275
Na	158	Mo	450
Mg	400	Ru	600
Al	428	Rh	480
Si	640	Pd	274
K	91	Ag	225
Ca	239	Cd	205
Sc	360	In	108
Ti	420	Sn	199 (white), 210(grey)
V	380	Sb	211
Cr	630	Cs	38
Mn	410	Ba	110
Fe	467	Ta	240
Co	445	W	400
Ni	450	Re	430
Cu	343	Os	500
Zn	327	Ir	420
Ga	320	Pt	240
Ge	370	Au	165
As	282	Hg	71.9
Rb	56	Tl	78.5
Sr	147	Pb	105
		Bi	119

File Formats

All files produced by SPIDER are text files, and can be viewed and edited with an ordinary editor, or with Spider's own edit windows. Although direct editing capability is provided by SPIDER, this is not particularly encouraged because it short-circuits SPIDER's error-checking procedures.

However, sometimes you may want to override Spider's default behaviour. A possible example is that you might want to override the formula which is used for calculating mean square atomic displacements. In this case, you can edit the values stored in the .MDL file directly. Another possibility is that you may wish to edit the .TRG file in order to add atoms, vary parameters or delete atoms. (When editing .TRG files, make

sure you don't move the End-of-File position away from the left hand margin, or introduce any blank lines. If you do, Kalypso may report an erroneous number of atoms in your target lattice.)

After being created, files of .PRJ, .RUN, .MDL, .IMP and INL type can be re-read by SPIDER for subsequent updating if needed. The dialog box information required to build .TRG files is not readable from these files, so the Target menu has no 'Open' option.

.TRG File Format

Each line in a .TRG file has a similar format as shown in the following example:

0.00000 0.00000 0.00000 14 28.08550 0 Si

The first 3 numbers are atomic coordinates in Å, followed by the atomic number (Z) and mass (in amu) respectively of the element concerned (in this case Si). The last number (0) is the flags parameter. The symbol (Si) is used for informational purposes only and does not affect the simulation.

Note: The **flags** parameter is an advanced option whose purpose is explained in the Simulation Primer. Normally this parameter will be left at its default value of zero (0).

.PRJ File Format

A .PRJ file is formatted as follows:

1.000 18 39.94800 0 Ar

The first number is the projectile energy in keV, followed by the atomic number and mass respectively of the projectile. The last number is the flags parameter. (The **flags** parameter is an advanced option whose purpose is explained in the file "`Simulation Primer.pdf`". Normally this parameter will be left at its default value of zero (0).) The elemental symbol (Ar) is used for identification purposes only.

.RUN File Format

A .RUN file is formatted as follows:

1 Checkbox parameter
001 Sample frequency parameter (timesteps)
5 Checkbox parameter
90.00 Polar angle of projectile incidence
0.00 Azimuthal angle of projectile incidence
01.0E-0016 Initial simulation timestep (s)

10 Frequency of updating neighbour list
1000.0 10.0 Termination times (fs): maximum, minimum
50 'Max. neighbours' parameter (allowed number of collision partners)
[ke/ep > 20.0] User-defined output expression

.MDL File Format

A .MDL file is formatted as follows:

0 // Projectile-Target core potl type

1.00000000000000E+0000 // Projectile-Target screening length corr

0 // Target atom-Target atom core potential type

1.00000000000000E+0000 // Target-target screening length correction

1 // Attractive Potential type: 0 = SC, 1 = TB [Parameters follow]

1.13000000000000E-0001 1.13000000000000E-0001 1.13000000000000E-0001

2.49180000000000E+0000 2.49180000000000E+0000 2.49180000000000E+0000

1.40870000000000E+0001 1.40870000000000E+0001 1.40870000000000E+0001

3.58740000000000E+0000 3.58740000000000E+0000 3.58740000000000E+0000

0.00000000000000E+0000 0.00000000000000E+0000 0.00000000000000E+0000

1.40050000000000E+0000 1.40050000000000E+0000 1.40050000000000E+0000

1.33500000000000E+0001 1.33500000000000E+0001 1.33500000000000E+0001

1.60000000000000E+0000 // low spline node, A
2.20000000000000E+0000 // high spline node, A
4.00000000000000E+0000 // Potential cut-off, A
1 // Use switching function = 1
3.80000000000000E+0000 // Switch function distance, Rsw
0 // Enable vibrations option
5.87000000000000E+0001 // mass, amu
3.00000000000000E+0002 // Lattice Temp, K
4.50000000000000E+0002 // Td (bulk), K
4.50000000000000E+0002 // Td (perp), K
4.50000000000000E+0002 // Td (par), K
0.00000000000000E+0000 // Surf BE, eV
0.00000000000000E+0000 // Bulk BE, eV
5.00000000000000E-0001 // Termin energy, eV
0 // Option: Test proj only?
0.00390 0.00390 0.00390 // rms vibrational A
0.00390 0.00390 0.00390 // rms vibrational B [*]
2.93981255377077E-0017 -2.44715845426498E-0007 4.21151842135212E+0002
3.81201795131315E+0011 // spline coefs for composite potential

2.80087737609301E-0017 -2.31514429612284E-0007 3.87255816764939E+0002
3.92998181830393E+0011 // spline coefs

2.86944098025965E-0017 -2.38018499566674E-0007 4.03885623685028E+0002
3.87398697736388E+0011 // spline coefs

29 28 // target atomic nos [used by Spider for generating graphs, but not by Kalypso]

[*] Note that you must edit these data yourself by hand if you want to

apply different vibrational corrections to the A and B types of atom (this

remark applies to a compound target only)

.IMP File Format

The first line of a .IMP file is formatted as follows:

1.0000E+0000 1.0000E+0000 3.0 1.000000 1.000000 1.000000 1.000000 90 0 1 1

The first two numbers represent the impact parameter of the projectile in the XY plane (Å). The next number (3.0) represents the height above the surface (3.0 Å) from which the projectile flight begins (z0), relative to the anchor atom in the target lattice (located by default at (0,0,0)).

The remaining numbers (columns 4-11) represent data used by Spider's Impact|Read dialog box; however, these numbers are not used by Kalypso. All information that Spider needs and uses to generate an Impact file is stored in line 1 of the Impact file.

Tip: If you want to archive a very large Impact file created by Spider, just store the first line in a text file to save on disk storage. (This only works if the Impact file generated by Spider has not been edited manually.)

Subsequent lines of the .IMP file (if any) contain only the (x,y,z) coordinate information, like this:

1.0100E+0000 1.0000E+0000 3.0

.INL File Format

// LSS data

```
3   // number of LSS items
29 13.31823299 0.5        // LSS definitions, starting with Z(atom)
28 13.00894464 0.5
18 9.509670314 0.5
11000   // threshold in m/s
```

// Oen-Robinson data

```
5   // number of O-R items
522 0.001123341978 0.5     // OR definitions, starting with Z1*Z2
841 0.001573233319 0.5
504 0.001106750624 0.5
784 0.001510125964 0.5
812 0.001536698237 0.5
1.5   // max. apsis in A
0.3   // gamma
```

// Shapiro-Tombrello data

```
1   // number of S-T items
0 0 0 10 0                // data that start with Z1*Z2=0 are ignored at runtime
```

// Cooling schedule data
250 // Cooling period, fs
300 // Temperature to cool to, K
0 2000 // Times to start/stop cooling, fs

Note that the length of INL files is variable

Editor

Summary

The editor used by Spider functions as a plain-text editor (by default) or as a rich-text format (RTF) editor. The editor mode is selected via the **Editor|Format mode** option on the Spider main window (applies to all new edit windows), or via the **Options|Format mode** menu (on individual editor windows). The main use of the editor is to view the files that Spider generates. Use the **Editor|New** option to create a new document, or **Editor|Open** to open an existing text or RTF file. In general, it is not a good idea to edit the Spider-generated files directly, unless you know what you are doing. (See the File Formats Help topic for information on these files.)

Editor Features

The difference between plain-text and RTF editor modes is in the way the editor stores data to disk. **Do not use the RTF mode for storing any file created by Spider, or it will become unusable by Kalypso.** (If this happens you need to open it again in RTF mode, then switch to Plain Text mode before saving it.) The current mode is indicated on the edit window status line. It is recommended that you save files created in

RTF mode with the .RTF extension. While running in RTF mode, the editor emulates a simple Windows word processor such as the Microsoft Write program. (It is assumed that the reader is familiar with such programs, and no Help on common editor operations is provided here.)

The editor has the standard options for saving, printing and editing documents on the **File** and **Edit** menus respectively. The editor can exchange text and images with the Windows clipboard (and store them on disk when run in RTF mode).

The **Options** menu allows you to switch between Plain Text and RTF editor modes (**Options|Format mode**), and to select the edit window background colour (**Options|Colour**) and font (**Options|Font**).

Target Files

Summary

The [Target|New] menu option brings up a dialog box into which you enter the parameters specifying your target lattice: elemental symbol, atomic number, atomic weight and lattice constant(s). The latter are specified in Å. Two lattice constants are needed to define a hexagonal lattice.

The **flags** parameter is an advanced option whose purpose is explained in Chapter 9 of the Simulation Primer. For an elemental target this parameter will normally be left at its default value of zero (0). Do not modify it unless you know what you are doing. For a compound target AB, the flags of type B atoms must be set to 1 (flags of type A atoms remain at 0).

Click here for tables of atomic data.

Procedural Details

The target lattice is generated row-by-row, and layer-by-layer, about some lattice point defined as the origin. The default origin is at (0,0,0), but this can be changed in the dialog box if desired, by modifying the **X-Origin**, **Y-Origin** and **Z-Origin** parameters respectively.

Important: The projectile coordinates at the start of the simulation will be referenced to those of the first ('**anchor**') atom in the .TRG file (normally located at or near (0,0,0)). Lattices generated by Spider's Target|New options perform this task silently. However, if you subsequently Orient a lattice you may have to cut and paste the line corresponding to your desired anchor atom at the top of the .TRG file (the order of the remaining atoms is irrelevant). (Whether or not you have to do this depends on the kind of operations involved in the orientation: rotations in the xy plane, or simple clipping of the edges of the lattice do not affect the anchor atom, but yz and zx rotations do require that you pay attention to the anchor atom.)

Lattice layers are laid down in the -z direction according to the following scheme:

surface of lattice

- o o o o o z = 0
  - o o o o z = -a
- o o o o o z = -2a

The **X-Width** and **Y-Width** parameters in the dialog box specify the number of atomic rows to add on to the lattice on *either side* of the row passing through the origin. Thus an x-width of 1 results in 3 rows in the x-dimension, while an x-width of 4 results in 9 x-rows. The **Z-Width** parameter gives the lattice depth in atomic layers.

The total number of atoms in the lattice is normally given by N = (nz)(2*nx+1)(2*ny+1). For the diamond structure, however, these dimensions refer to diatomic units, and the number of atoms in the lattice will therefore be 2(nz)(2*nx+1)(2*ny+1).

If you need a lattice with an orientation not specified in the menu option, e.g. fcc(210), you'll have to create your own by using the [Target|Orient] option to reorient one of the standard lattices.

Click here for information concerning the (default) azimuthal orientation of the target generated by Spider.

Target: Elemental Symbol

The elemental symbol is a 2-character string, for example: 'Cu' for copper. For vanadium you can use 'V', 'V ' or even ' V'. The elemental symbol is only used for informational purposes, and has no effect on the simulation parameters.

Click here for a table of atomic data.

Target: Atomic Number

The atomic number (Z) is an integer (1-99) which specifies the nuclear charge of your target element. This parameter is used by Kalypso for calculating the short-range (screened Coulombic) potentials in your simulation system.

Click here for a table of atomic data.

Target: Atomic Mass

The target atomic mass represents the mass of the atoms in your target lattice, expressed in atomic mass units (amu). For example, the mass of carbon-12 is 12.000, while that of normal carbon is 12.011.

Click here for a table of atomic data.

Target: Lattice Constants

Lattice constants (expressed in Å) define the interatomic spacing in your target lattice. For a cubic (FCC, BCC or diamond) lattice, only one lattice parameter (*a*) needs to be specified. For a hexagonal (HCP) lattice, two lattice parameters (*a = a1 or a2*, *and c = a3*) need to be specified.

Click here for a table of atomic data.

Target: Default Azimuthal Orientations

Summary

The following list summarises the default crystallographic orientations of the x- and y-edges of lattices generated by the various Target|New options. (The graphics show the surface view (xy plane) of the lattices as they are created by Spider.) In most cases, you will only need the generic form of the edge orientation, according to which (for example) [01-1] becomes <110> (or perhaps <011>, to indicate that it lies in the surface plane.) The notation [1-20] is used to represent the orientation [1,-2,0].

Note that the edges of the diamond lattices are quite 'ragged'. (This is a consequence of the generating algorithm.) They need to be trimmed off using the Display/Orient option.

A good reference, with diagrams, for checking surface crystallographic directions is: M.W. Roberts and C.S. McKee, **Chemistry of the Metal-Gas Interface**, (Clarendon Press, Oxford, 1978), chapter 2.

Face-Centred Cubic (FCC)

(a) FCC (100):

y: [011] 

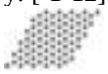
x: [01-1]

(b) FCC (110):

y: [-110] 

x: [001]

(c) FCC (111):
y: [-1-12]

 x: [-110]

Note: all 4 sides of the (111) lattice surface generated by Spider (two of which are tilted at 60 deg. to the x-axis) are <110> rows.

Body-Centred Cubic (BCC)

(a) BCC (100)

y: [-100]

 x: [010]

(b) BCC (110)

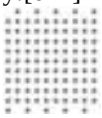
y: [001]

 x: [-110]

Diamond (Dia)

(a) Dia (100):

y:[011]

 x: [01-1]

(b) Dia (110):

y: [-110]

x: [001]

(c) Dia (111):

y: [-1-12]]

 x: [-110]

Hexagonal Close Packed (HCP)

(a) HCP (0001):

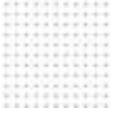
y: [-1010]

 x: [-12-10]

Rocksalt (NaCl)

(a) NaCl(100):

y: [010]

 x[001]

Sphalerite

(a) Sphalerite(100):

(b) Sphalerite(110):

Neighbour Distances

Summary

The **Target|Neighbour Distances** option creates a list of the neighbour distances for a specified atom in a Target file.

The user has to specify the name of the Target file (via the **Select** button), the index of the lattice atom in question (i.e. a line number in the Target file), and the maximum range of neighbour distances (in A) to be collated.

The output of the routine is written to an editor window in the form of a listing of nearest neighbour distances (r) and the corresponding lattice atom index (atom #).

Note: any text file formatted with at least 3 columns of numbers can be processed by this routine, not just Target files.

Applications

This routine has 2 useful applications.

First, it enables you to gain an understanding of the 'Number of Partners' parameter which you will have to set up in the Run file of your project. For example, it is found from the Neighbour Distances routine that bulk Cu atoms have 42 neighbours within a distance of 4.5 A in the undisturbed lattice. If we use a Potential Cut-Off of 3.8 A (slightly larger than the 2^{nd} neighbour distance) then it will be necessary to set the Number of Partners above this number, particularly if the timestep (δt) is large. A better estimate would be to consider how many neighbours lie within a distance $3.8 + \delta r$, where δr is the distance moved by the fastest atom between neighbour updates ($\sim 10 * v * \delta t$, where v is velocity of faste).

Second, the routine can help you with general crystallographic problems which require a knowledge of the coordination numbers of bulk and surface lattice atoms.

Anchor Atom

The term 'anchor' atom refers to the atom whose coordinates are listed in the **first line of the Target file**. The anchor atom must be located at the surface of your lattice. Spider assures this automatically, and assigns the coordinates (0,0,0) to this atom by default. If you edit or orient Target files, be careful to check (with a text editor) that the anchor atom has the coordinates you expect.

When the simulation is run, the projectile initial trajectory and position will be referenced to the coordinates of the anchor atom as specified in the Target file. For example, the impact parameters describing a particular collision will refer to the projectile's interaction with the anchor atom.

Note: Any vibrational correction to the anchor atom's location is applied after the projectile coordinates have been calculated.

Display/Orient Overview

See Also: Crystal Axes, Display/Orient Target & Impact Files

Summary

This Help topic gives an overview of the Display/Orient options which are found on the Target menu and the Impact menu respectively. These options are used to display or modify the data held in .TRG or .IMP files. The procedural options are similar for operations on both sets of files. The sections below consider operations with .TRG and .IMP files separately, including a worked example showing how to create an Impact file for a (111) surface.

Procedure

The Display/Orient window has two modes of operation.

You can rotate, zoom or shift the display without altering the data stored on disk. The '**Display Parameters**' options are found on the left hand side of the window.

The options displayed under the '**Orient Parameters**' title refer to operations which you carry out on disk data for permanent storage (invoked by hitting the **Apply** button). The **Test** button applies the same orientation operation as **Apply**, but affects the display only. You can reset the display to its original condition via the **Reload** button. Only one **Test** or **Apply** operation is accepted between **Reloads**.

The **Impose** operation superimposes coordinate data from a specified .IMP/.TRG file onto an existing .TRG display (Target/Display menu option), or superimposes data from a specified .IMP/.TRG file onto an existing .IMP display (for the Impact/Display menu option). NB. Use the Display/Link menu at the top of the window to specify the file to be linked for the Impose operation.

Target files

The process of orienting a target file is similar to the process of cutting and orienting a crystallite specimen on a laboratory goniometer. You would need to use this procedure, for example, if you wished to perform a simulation involving a non-standard surface like Cu(210). The methodology will be familiar to anyone who has oriented a crystal using spark-cut/Laue diffraction techniques. The lattice to be oriented must first be created by the Target|New Target procedure, and will normally reside in a .TRG file.

In the Orient dialog box, you specify what orientation, cutting and translation operations are to be carried out. Rotations are in general non-commutative, so it may be easier and less error-prone to carry them out one at a time.

After rotation, the lattice crystallite can be truncated ("cut") outside the specified limits. The truncation can take place on a Cartesian (x,y,z) boundary, on a cylindrical (r,z) boundary or on a spherical (r) boundary.

The cylindrical truncation is worth considering under any circumstances, as this shape is computationally more efficient than a microcrystallite with corners.

When **cylindrical** truncation is selected, the specified x,y limits will have no effect: the truncation is performed for rmin < r < rmax [with r = sqrt(x*x+y*y)] and rmin/rmax as specified by you in the input line. Likewise for **spherical** truncation (where now r = sqrt(x*x+y*y+z*z)).

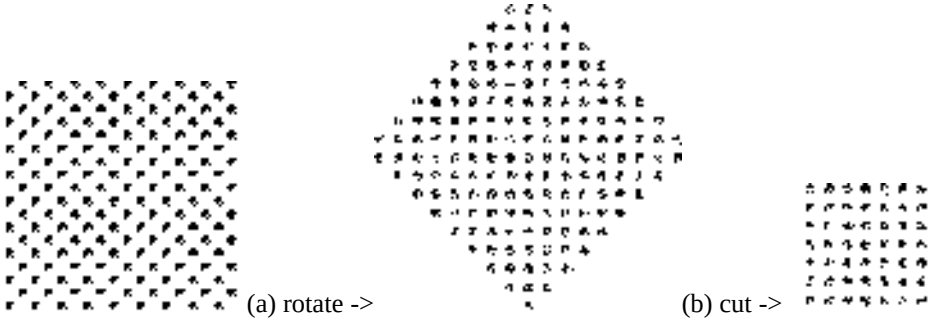
If required, a new origin (dx,dy,dz) can be defined for the coordinate system, i.e. (x,y,z) in the old system becomes (x-dx,y-dy,z-dz).

Important: The projectile coordinates at the start of the simulation will be referenced to those of the first ('anchor') atom in the .TRG file (located by default at or near (0,0,0)). After orienting a lattice you may have to cut and paste the line corresponding to your desired anchor atom at the top of the .TRG file (the order of the remaining atoms is irrelevant). This situation will not arise for an xy rotation, but it will normally require attention after a yz or zx rotation.

Simple example of a target orientation

The task considered here is to create a fcc(100) target with <100> edges (the New Target option generates a lattice with <110> edges by default).

This can be achieved by a 45 degree rotation in the xy plane, followed by a cutting operation (expand the Help window to view these diagrams correctly):



Here is what you do in detail:

(a) Specify a z-rotation of 45 degrees;

(b) Specify x and y truncation limits according to the extent of the lattice required. In this example, the minimum x and y values were set to -7 A, and the maximum values were set to +7 A.

To create an entirely new surface orientation, your rotation would involve suitable x and/or y rotations in conjunction with truncations. For instance, rotations of R(xy) = 45 and R(yz) = 45 on a (100) FCC target, combined with suitable truncations (to terminate the surfaces of the lattice) would create a (110) target. Use the Crystal Axes utility (on the Utilities menu) to calculate the rotations required for a given re-orientation operation.

Impact Files

The orientation of an Impact file is a less common operation than orientation of a target file. It is only necessary when the impact region has a non-rectangular shape (as for (111) surfaces). In such cases you need to shape the impact region yourself. In other cases you can always achieve the desired effect by rotating the target lattice.

The orientation of an Impact file is simpler than orientation of a Target file because the Impact file coordinates only change in the xy plane. This means that there is only one axis of rotation (z), for example.

A typical Impact file orientation will now be described.

How do you create a .IMP file for a (111) surface?

The diagram below illustrates the goal of the procedure: a suitable distribution of impact points for projectile normal incidence on a (111) surface. The surface atoms of Al(111) form equilateral triangles. In a simulation we direct projectiles into a symmetry-reduced triangular area. The apexes of this triangle are located at the positions of atoms in 3 different layers of the (111) surface. Spider does not generate the symmetry-reduced .IMP file directly: it has to be trimmed and aligned according to the procedures now described.

Looking down on an Al(111) surface. Red points: surface atoms. Green points: 2nd layer atoms. Blue points: 3rd layer atoms. The mesh of impact points is shown in purple.

Step-by-step procedure

(a) We need the following data for Al: lattice const. (a) = 4.0495 Å, nearest neighbour distance (dnn) = 2.8634 Å.

(b) Select **New** on the **Impact** menu. Name the file (*.IMP) as you like.

Enter the following data into the dialog box:

bx(min) = 0.0 bx(max) = 1.653184 No. impacts = 26

by(min) = 0.0 by(max) = 1.653184 No. impacts = 26

(x,y) angle = 60

Projectile z0 = 3.0

[NB. replace 1.653184 by d_{NN}/sqrt(3) for other fcc crystals]

(c) The number of impact points this data generates is 476. Some of these will be trimmed off later. You may of course select any number other than 26, according to your preferences. Choose the same number for both x and y dimensions to ensure a uniform distribution of points (advisable rather than mandatory).

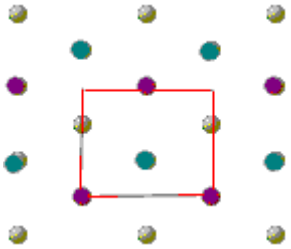
(d) Now select the **Impact|Orient/Display** menu option and load the .IMP file you created in the previous steps. We need to reorient and cut the mesh of points in this file so that they match the dimensions of the reduced symmetry zone. This involves two operations.

(e) Enter 60 for the Z-Axis rotation, and 1.4317 (or dnn/2 for other fcc crystals) for the Max. y-Axis limit. Leave the other limits at their default values. Click the Test button. This will rotate the displayed mesh by 60 degrees, and clip its edge above y = 1.4317. The resulting .IMP file has 306 impact point coordinates left after this operation (compared with 476 originally). If you are satisfied with the test, you can then click the Reload button, followed by the Apply button, to carry out the same operation and save the resulting data to disk under a new name. For the last operation (f), you must first close the display, then reload this newly oriented IMP file.

(f) Enter -30 (minus 30) for the z-Axis rotation, and reset the Max. y-axis value to its default value (100 Å). Click Test. This will rotate the mesh into the final alignment shown in the above figure. Finally, click Reload then Apply to save the oriented IMP file to disk.

Note: the symmetry-reduced impact zone created in the above exercise is only valid for normal primary ion incidence. For oblique angles of incidence you will need to use a larger zone, whose dimensions depend on the primary ion azimuthal direction.

For example, with the ion incident parallel to a [101] direction of a (111) fcc surface, the impact zone is a rectangle of dimensions: x = d; y = d*sqrt(3)/2 (where d is the nearest neighbour distance, and assuming the [101] surface row is parallel to the x-axis). This zone can be created immediately by Spider's Impact|New command, without need for further trimming etc. The figure below shows the region, using different colors to indicate the atoms in the 1st (red), 2nd (green) and 3rd (grey) layers of the fcc (111) surface:



Display/Orient Target & Impact Files

Click an item on the graphic below for information about dialog box controls, or select an item from the following list: Orient Parameters, Rotation, Type of Cut, Boundaries of Cut, New origin, Action Buttons, Image, Move Display, Display Parameters, Perspective, Rotate Display, Zoom, Symbols, Information, Menu Options.

The Display/Orient dialog boxes on the Target and Impact menus respectively are very similar in appearance and function. The difference is that the Target menu version orients/displays the atomic coordinates of a lattice (stored in a .TRG file), whereas the Impact version orients/displays the coordinates in a .IMP file (The .IMP file coordinates are used - in conjunction with angular data from the .RUN file - to initialise the position of the projectile.).

{bmc odhelp.shg}

Orient Parameter

The parameters you specify on the left side of the Display window will determine how the coordinates in your .TRG or .IMP file will be processed. These parameters are completely independent of the Display parameters entered on the right side of the dialog box.

You can get information on a specific parameter from the following list: Rotation, Type of Cut, Boundaries of Cut, New origin.

Operations are carried out according to the "tab order" of the various dialog box input lines: that is, commencing with rotations about the z-, x- any y-axes sequentially, after which the lattice is "cut" and "translated".

Orient Parameters: Rotation

Specify what rotations you wish to apply to the coordinates stored in the currently displayed .TRG or .IMP file. To initiate the rotation, click the button marked 'Apply'. This will allow you to write the rotated coordinate data to disk. If you only want to rotate the display, you should select the Rotate Display option from the right hand column of the Orient/Display window.

Rotations are applied in the order R(x,y), R(y,z), R(z,x), where, for example, R(x,y) represents a rotation in the xy plane (around the z-axis). Only the R(x,y) option is active in the case of a .IMP file.

Orient Parameters: Type of Cut

Specify the shape of the boundaries used to truncate (cut) a set of coordinates defined in a .TRG or .IMP file, by clicking on a shape or on a radio-button option. The shapes available for shaping Impact file data are 2-dimensional, whereas those for Target files are 3-dimensional.

Orient Parameters: Boundaries of Cut

Specify the ranges for truncating (cutting) data in the currently displayed .TRG file (target) or .IMP file (mesh of impact points). For example, if you specify that x-max should be 10 A, then **no atoms will be saved that have an x-coordinate > 10 A**.

Orient Parameters: Position of New Origin

You will rarely use this option for an elemental target. The effect is to displace the coordinates (**R**) of your .TRG/.IMP file to a new position (**R**'): **R**' = **R** - δ**R** . That is: x' = x - δx and so on.

You need to specify the δx, δy and δz components of the translation vector δ**R** (in A).

Orient/Display: Buttons

The functions of the Orient/Display buttons are:

Apply: Initiates an orientation operation based on the current values of Orientation Parameters.

Test: Applies an orientation operation to the display data, so that you can preview the outcome. However, no data are written to disk.

Reload: Restores original data to display by reloading it from the disk.

Impose: (Enabled when you link a file via the Display/Link menu option.) Imposes data from the linked file on the current display. The imposed data do not persist when the display changes.

Exit: Close the Orient/Display window.

Orient/Display: Image

The image in the centre of the Orient/Display window is a graphical representation of the coordinates in a .TRG or .IMP file (as appropriate). In addition, you can superimpose the coordinates from another .IMP/.TRG file using the Display/Link menu option. What you see on screen is determined by the current Display Parameters.

Orient/Display: Move Display

Click on the red arrows to move the image displayed on screen. You can use the Alt+arrow key combination to achieve the same effect.

Display Parameters

Changes to the various display parameter options (including the Move Display arrow buttons) affect only the display you see on the screen. To restore the original, unmodified display you can click on the **Reload** button.

Display Parameters: Perspective

The Perspective option determines the plane from which the .TRG/.IMP file coordinates are viewed. What you actually see will be influenced by any rotations which have previously been applied to the display.

Display Parameters: Rotate Display

The Rotate Display options rotate the coordinates of the currently displayed .TRG or.IMP file (no changes are made to disk data) in any of 3 planes. Rotations are cumulative. Rotations can be made in increments or decrements of 5 degrees by clicking the "up-down" buttons, or they can be specified by editing the appropriate input box, followed by a click on the Apply Rotation button. The Information box displays the **cumulative** total of rotations applied in each plane. You can restore the default data as loaded from disk by clicking the Reload button.

Display Parameters: Zoom

The Zoom buttons change the scale of the graphical display.

Display Parameters: Symbols

Coordinates from a .TRG or .IMP file can be displayed using any of 3 symbol types:

- Icons: Bitmapped images of 'atoms';
- Circles;
- Points: Single pixel display;

Display Parameters: Information

The Information box displays (a) The spatial (x,y,z) coordinates in Angstrom corresponding to the mouse location on the currently displayed image; (b) the cumulative totals of angular rotations applied to the currently displayed image (via the Rotate Display option).

Orient/Display: Menu Options

The available menu options are:

- Display|Background colour: Toggles the background display colour between black and white;
- Display|Link file: Specifies a .TRG or .IMP file which will be 'linked' to the current display, and whose coordinates will be displayed when the Impose button is clicked. This option allows you to directly compare data stored in different files.
- Edit|Copy: Copies the currently displayed image to the Windows clipboard.

Projectile Files

Summary

The **Projectile|New** menu option brings up a dialog box, into which you can enter the parameters which specify the characteristics of the primary projectile in your scattering/bombardment simulation. This data can be saved in the form of a 'Projectile File' with extension .PRJ. The data in such a file can be read and edited at a later time using the **Projectile|Open** menu option.

The parameters specified in a Projectile File are respectively: the elemental symbol of the projectile atom, its atomic number, atomic mass (in amu) and energy (in keV).

The **flags** parameter is an advanced option whose purpose is explained in the Simulation primer (chapter 9). Normally this parameter will be left at its default value of zero (0).

Experimentalists normally employ ions rather than atoms for bombardment experiments. However, it is typically assumed that most ions are neutralised before they collide with the target surface.

Projectile: Elemental Symbol

The elemental symbol is a 2-character string, for example: 'Ar' for argon. For hydrogen you can use 'H', 'H ' or even ' H'. The elemental symbol is only used for informational purposes, and has no effect on the simulation parameters.

Click here for a table of atomic data.

Projectile: Atomic Number

The atomic number (Z) is an integer (1-99) which specifies the nuclear charge of your target element. This parameter is used by Kalypso for calculating the short-range (screened Coulombic) potentials in your simulation system.

Click here for a table of atomic data.

Projectile: Atomic Mass

The target atomic mass represents the mass of the projectile ions/atoms in your simulation system, expressed in atomic mass units (amu). For example, the mass of carbon-12 is 12.000, while that of normal carbon is 12.011.

Click here for a table of atomic data.

Projectile: Energy

The projectile energy (or primary energy) is the kinetic energy of the projectile species at the commencement of the simulation (t = 0). In the Projectile file menu dialog box, the user should specify this energy in keV (1 keV = 1000 eV).

Run File Overview

See Also:

Sampling Frequency Output Dynamic Variables Information

Projectile Incidence Periodic Parameters
Cut-Off Parameters

Summary

Apart from the specification of the projectile angles of incidence, the parameters stored in the .RUN file chiefly influence the way in which the simulation model is implemented at the *computational* level. This is in contrast to the .MDL file parameters, which determine the type of *physical* model which underlies the simulation. In particular, the .RUN file parameters determine how Kalypso manages its neighbour lists, how much dynamic memory it allocates for the simulation, what kind of output is generated, and when the run ends.

Run: Sampling Frequency

See Also: Run File Overview

Overview

The purpose of running the Kalypso simulations is to derive information from the dynamical variables of the system. The user of Spider has the option of specifying when this information will be recorded. The options are:

(a) To record dynamical variables information upon termination of each simulation run, or

(b) To record them periodically, as each collision cascade evolves, or

(c) Both (a) and (b).

(a) is the most common option.

If you choose to store periodic information (options (b) or (c)), you should also specify the Sampling Frequency in the input line indicated (see below).

The 'Limit to one record' means that at most only one record will be written for each atom (on the first occasion that it satisfies the output condition. This option is rarely useful. (It has no effect if option (a) above is selected.)

Hints

The most common option is option (a). For example, a sputtering simulation would require an estimate of the number of atoms sputtered in each run after (say) 300 fs had elapsed. There is no need to record information about the evolving collision cascade. Likewise, a simulation of an ISS experiment would focus on the asymptotic ion trajectories.

Option (b) would typically be chosen if particle trajectories were the focus of interest. For example, you might periodically wish to record 'snapshots' of the system every 10 timesteps. In this case, enter the number '10' as the Sampling Frequency parameter.

Run: Output Dynamical Variables Information

See Also: Run File Overview

Overview

The user is required to specify what kind of data should be recorded by the simulation. The purpose of these options is to avoid unecessarily writing data to disk that won't be used later. For example, if the purpose of the simulation is to calculate sputtering coefficients, you can simply record data for ejected particles. If your simulation is modelling ion scattering processes you can record data for the projectile only.

There are several choices, one of which can be selected by clicking the appropriate button. Each of the options records dynamical variables information in a standard record format. The difference between the options is that they record the information for different groups of particles. Once again, the purpose of having these options is to avoid recording data irrelevant to the purpose of the user.

Options

- All particles in system: always records data for every particle in the system, including the projectile.
- Projectile only: always records data for the projectile, but no other particles.
- All particles with KE > 1 eV: records data for particles with kinetic energies above 1 eV.
- All particles with KE > 10 eV: records data for particles with kinetic energies above 10 eV.
- All particles with KE > 100 eV: records data for particles with kinetic energies above 100 eV.
- All emitted particles: records data for all particles with z > 0.5 A, and vz > 0.0, including the projectile.

- Projectile emission:records data for the projectile only, provided z > 0.5 A, and vz > 0.0.
- All particles emitted within 5 degrees of surface normal: similar to the 'All emitted particles' option; however, this option only records data for those particles emitted within a 5 degree cone around the surface normal.
- User-programmed option: see below.

User-Programmed Option

The User-Programmed Option is a new feature in Spider 2.1. This is an advanced option which allows the user to specify a condition which must be fulfilled by a particle's dynamical variables before its data are recorded. The conditional expression specified by the user will be interpreted by Kalypso at run-time.

The language in which conditional expressions are framed is the same as the query language used by Winnow. For a brief overview, click here. For a formal definition, see the file `winnow.doc` found in the `\docs` directory.

As a simple example, the following expression will achieve the recording of data only for particles which leave the surface (arbitrarily bounded at 0.5 A):

[rz > 5.0e-11] & [vz > 0.0]

Here `rz` represents the particle's z-coordinate, while `vz` represents the z-component of its velocity vector. The '`&`' symbol is the logical AND operator. The output produced by this specification is identical to the 'All emitted particles' option described above. In pseudo-code this specification is programmed as follows:

```
for n = 0 to NATOMS do
  begin
    if (rz[n] > 5.0e-11) AND (vz[n] > 0.0) then
    WriteToFile(rx,ry,rz,px,py,pz...)
  end;
```

Likewise, the expression: [ke/e*p* > 100.0], is only satisfied if a particle's kinetic energy (`ke/e`$_p$, where e*p* is the proton charge) is greater than 100 eV. Therefore the meaning of this expression is to specify that data are only to be recorded for those particles with more than 100 eV kinetic energy. In psuedo-code:

```
for n = 0 to NATOMS do
  begin
    KE = (sqr(px[n]+sqr(py[n])+sqr(pz[n]))/(2*mass[n]);
    if (KE/ep > 100.0) then WriteToFile(rx,ry,rz,px,py,pz...)
  end;
```

User-programmed expressions may be up to 255 characters in length. Various examples of expressions are provided in the Help topic associated with the expression memo box, which is accessed via the **Samples** button. Note that comments bracketed within curly braces `{like this}` will be ignored by the expression parser, and they can be used as required to improve readibility.

Note. The validity of the syntax of any user-programmed option is not checked by Spider. Instead, syntax checking is carried out by Kalypso at run-time.

Run: Projectile Incident Angles

See Also: Run File Overview

Summary

The initial direction of the projectile trajectory is specified by means of its altitudinal (φ) and azimuthal (ϕ) angles respectively in the same (x,y,z) coordinate system which describes the .TRG file lattice sites. For normal incidence, φ = 90. The azimuthal angle is measured as an anti-clockwise rotation in the (x,y) plane, starting from the positive x-axis (ϕ = 0).

If you select ϕ <> 0, you will need to have an understanding of how the specified value of ϕ relates to the major crystallographic axes (like <100> or <110>) of your lattice file. In other words, you need to know where these axes lie in the (x,y) plane in order to set the value of ϕ that corresponds with your experimental interest.

The author strongly recommends that you always set f = 0.0, and that you rotate your lattice to achieve the correct crystallographic alignment with the beam.

The reason for this strong statement is that it is all too easy to forget that the shape of the 'reduced impact zone' depends on the relative azimuthal beam-lattice alignment. Users should appreciate that when the latter is changed, the Impact file specifications should also be changed. (If you use the entire bulk unit cell dimensions to define the impact zone, this requirement no longer holds.)

The recommended procedure for setting up the azimuthal alignment is:

(a) Set ϕ = 0.0;

(b) Rotate your lattice so that the axis of interest lies parallel to the x-direction;

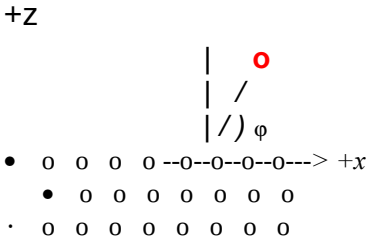
(c) Identify the 'reduced impact zone';

(d) Prepare a .IMP file corresponding to the reduced impact zone.

If you are not sure about the azimuthal orientation of your lattice with respect to the (x,y) plane, you can inspect it visually via the Target|Display/Orient menu option, and read the Help topic that summarises the default azimuthal orientations of lattices generated by Spider.

Example

The following illustration depicts the starting trajectory for a projectile (**o**) with ϕ = 0 and φ ~ 70 (note that in this configuration both vx and vz components of the projectile velocity are initially negative):



The choice of a value for φ is not difficult, and is governed by the experimental set-up. Normally ϕ is specified in terms of a crystallographic direction: e.g. a projectile may be incident on Cu(001) along a <100> azimuth ('<100> incidence'). The number you select for ϕ must be chosen with reference to the actual alignment of the .TRG file which you intend to use for the simulation. By default, the **Target|FCC|(100)** menu option generates a (001) surface with <110> rows parallel to the x and y axes. In this case, you should specify ϕ = 0.0 if you want <110> incidence, and ϕ = 45 if you want <100> incidence. Equivalently, as recommended above, you can rotate your Cu(001) target by 45 degrees to achieve <100> incidence with the same value of ϕ = 0.0.

Symbols

Position vector of the projectile: (x[0],y[0],z[0])

Position vector of the anchor target atom, read from line # 1 of the the .TRG file: (x[1],y[1],z[1])

Velocity vector of the projectile: (vx[0],vy[0],vz[0])

Magnitude of the initial velocity of projectile: v0 = sqrt(vx[0]2 + vy[0]2 + vz[0]2)

Run: Periodic Parameters

See Also: Run File Overview

Summary

Depending on the projectile energy and other characteristics, different simulation systems will evolve over different timescales. The 'Periodic Parameters' are parameters relating to the passage of time. Although their values are not critical to the success of the simulation, they have to be appropriately chosen for the user's specific system in order to ensure efficient computation. The individual parameters (**Initial Timestep**, **Neighbour Update Time**, **Termination Time**) will be considered in the following sections.

Initial Timestep

The Initial Timestep (expressed in seconds) represents the width of the time integration interval, δt, used by Kalypso.

The timestep is chosen in order to effect a compromise between computational speed and integration accuracy. (The larger the timestep, the faster the simulation.) The timestep scales roughly with velocity, i.e. as sqrt(m/E), where m and E are respectively the mass and energy of the projectile.

Kalypso gradually increases the timestep as the atoms in the collision cascade slow down, in such a way that the maximum displacement δr (for the fastest particle) stays approximately constant for every timestep. So by selecting an Initial Timestep, δt, you are also in effect specifying the spatial distance, δr, to be used for the entire simulation. (You can use the Velocity Reckoner gadget to quickly calculate displacements per timestep corresponding to particular energies.)

Typically δt is set so that δr is no more than 2-5% of the interatomic distance in the target atomic lattice (or ca. 0.1 A, although shorter values are needed for very energetic collisions). For example, a timestep of 5.0E-17 s or less would be suitable for 5 keV Ar projectiles (v = 1.6 A/fs).

If the timestep is too large, there is an unacceptably high error in the energy conservation reported by Kalypso. This effect is most significant for hard collision configurations, where the potential changes most rapidly with position. The optimum choice of timestep depends on the kind of information to be extracted from the simulation. For processes associated with hard collisions (e.g. impact collision ion-scattering spectroscopy) a value of perhaps 3.0E-17 is appropriate for the 5 keV Ar-Cu system, whereas for processes with larger cross-sections, a value of 8.0E-17 might be adequate.

It should be mentioned that small impact parameter projectile-target collisions are a rare event. The average energy conservation error reported by Kalypso is dominated by large impact parameter collisions (on the order of 1 A).

If the timestep is increased, you may also find that Kalypso aborts after reporting the message "Too many partners: increase number of partners...". You should not be alarmed. This occurs because longer timesteps require larger neighbour lists. The solution is to increase the "Maximum No. of Partners" parameter in the Run dialog (or to reduce the timestep, which may have been set too large). This allows Kalypso to allocate an appropriate amount of memory for the neighbour list.

Neighbour Update Time

The Update Time is expressed in 'timesteps' (integration cycles). This parameter determines how neighbour lists are managed. (A neighbour list is a list of current and possible interaction partners for each atom, which is compiled on the basis of system velocities, current interatomic distances and potential ranges.)

During a simulation run, neighbour lists are updated periodically, according to the Neighbour Update time parameter (1= every timestep, 10 = every 10^{th} timestep etc). A typical value for this parameter is around 10. No error will result from setting a sub-optimal Update Time.

However, if the lists are updated too frequently, the calculations run more slowly; if the lists are updated too infrequently, then Kalypso has to track far more potential collision partners which requires more memory, and may slow the calculations unacceptably. Note that if the neighbour update time is increased, you will normally have to increase the parameter specifying the number of partners (neighbours) that Kalypso can track (otherwise you will get an error message when you run).

To set an optimum Update Time, you generally should carry out a few preliminary runs to test the sensitivity of the calculation speed to this parameter.

Termination Times

The **Minimum Termination** Time specifies the minimum time for which the simulation will be run, regardless of what happens to the dynamical variables. There are situations in which this may be useful. It is a good idea to use a non-zero value of this parameter in impact collision ISS simulations, for reasons explained elsewhere.

The **Maximum Termination Time** specifies a time (in fs) at which Kalypso will automatically terminate the current run, regardless of the state of the system dynamical variables. The termination time should not be shorter than the timescale of the process which is being modelled.

One reason you need this parameter is to prevent infinite looping of the calculations if for some reason the alternative termination condition (low energy) is not met. The energy-termination condition does not always give satisfactory results because it only takes account of the atoms within a 'lattice' region defined by the initial target atom coordinates. If all target atoms are liable to move outside this region (e.g. if the target consists of a single atom) then you must control the simulation termination via a time condition.

A value of 1000 fs or more would be typical for sputtering studies. After this time, any atoms that are going to be sputtered would already be on their journey out of the surface. There is no point in wasting computing resources by calculating beyond this point. (A much shorter time would be appropriate for ion scattering, because of the abrupt nature of the interaction.)

You should attempt to combine both time and energy termination conditions to control the exit point of your simulations. For instance, in a study of sputtering you could specify that the simulation should terminate if t > 1000 fs, or if the most energetic particle in the lattice region has < 1 eV kinetic energy. A careful choice of these exit conditions may speed the calculations.

Run: Cut-Off Parameters

See Also: Run File Overview, NeighbourDistances

Summary

The simulation algorithms used by Kalypso only take account of interactions over a finite range specified by the **Potential Cut-Off** parameter (R*c*). A consequence of this is that each atom interacts at any time with only those neighbouring atoms that lie within a sphere of radius R*c*. Kalypso maintains a list of these atoms, as well as any others that may potentially move into interaction range between list updates. The **Maximum No. of Partners** parameter represents the number of interaction partners that may be tracked for each atom in the lattice. Some guidelines for setting these parameters are given below.

Potential Cut-Off

A typical choice for the Potential Cut-Off would be some distance between the second and third nearest neighbour distance in your lattice. For example, the nearest and next-nearest neighbours in Cu are 2.56 and 3.62 A respectively. A Potential Cut-Off of around 3.8 A would be appropriate for this case. The maximum value of the Potential Cut-Off allowed by Kalypso is 8.0 A, but you will probably never use such a large value.

For exploratory work, or for calculations involving huge lattices, you can consider reducing the Potential Cut-Off to a value between the first and second nearest neighbour distances (e.g. 3.0 A for Cu). If you are only interested in hard-collisions (as in Ion Scattering Spectrometry) you should probably set a Potential Cut-Off which reflects this (say 1.0 A), because you will benefit from a great increase in computational speed.

Calculations run more slowly as the potential cut-off is increased, because this leads to a larger number of significant pairwise interactions. Memory usage by the computer is also higher, since these extra interaction partners have to be tracked via the neighbour lists.

Maximum Number of Partners

This is admittedly an awkward parameter to set. It is required to keep 's memory usage flexible for users with PCs that don't have an abundance of memory. If you have 32 MB then set this parameter permanently to a very large value (100 say) and don't worry about it.

The significance of the parameter is that it is a measure of the number of neighbouring atoms that have to be included in the biggest neighbour list. (In other words, it determines the amount of memory which will be allocated for the lists.)

The neighbour list of a given atom is a list of all other atoms in the system which are currently within interaction range, or might possibly move within interaction range before the next update of the neighbour list.

The value might be up to 100 depending on the simulation conditions, including: the potential cut-off distance, the projectile velocity, the structure/size of the target lattice and the Neighbour Update Time. This value is somewhat higher than the number of atoms in the primary and secondary coordination shells because the neignbour list also has to take account of moving atoms near the border of the shell. or example, there are 78 atoms within a distance of 1.58a in an undisturbed fcc lattice (a is the lattice parameter).

There is no harm in setting the Number of Partners too high, if your PC has the memory available. Otherwise, consult the example files that are shipped with , or carry out some exploratory runs.

Kalypso will generate an error message at run-time ('Too many partners required') if it finds that the neighbour lists need more partners than you specified. The simulation will then abort gracefully. You can sometimes get the same error message if you set the Initial Timestep to a spuriously high value (for instance, 1 fs for 5 keV Ar projectiles). Whenever Kalypso runs (in verbose mode), it will inform you how many partners were actually required so that you can familiarise yourself with the size of the parameter (look for the message: Maximum no. of collision partners required was:...'). If you still get the 'Increase number of partners' error message after setting this parameter to above 100, it probably means that your time step is too large and has produced an integration error. This may not happen on every run (the error may arise after hundreds of runs have successfully completed).

In general, if you increase the primary projectile energy, the number of partners required will also increase (because of higher speeds of atoms in the collision cascade). Simulation runs involving hard collisions will also require a higher number of partners than those involving large-impact parameter collisions (because of greater lattice disruptions).

Run: User-Programmed Option (Samples)

See also: Output Dynamical Variables

Tips

You can select (with the mouse) and copy (Ctrl-C) the examples given here, and paste them into the memo box of the Run file dialog window using Ctrl-V. The examples are used to illustrate the syntax of the program language, not because they have some special signifcance.

Remember that (a) **all expressions use SI units**; (b) **real numbers must be expressed with a decimal point** (10.0) or with exponents (1e1) but not as integers (10).

The predefined expression `ke/ep` refers to the particle kinetic energy expresed in eV.

Example 1. Store data for a single lattice atom (atom #5).

[rw = 5]

Comment: The meaning is 'row-number (in Target file) = 5'. For the projectile, rw = 0. For the anchor atom, rw = 1.

You can specify that data be stored for lattice atoms #1 to #100 by the following expression:

[rw >= 1] & [rw <= 100]

Example 2. Store data for ejected atoms moving within 45 degrees (NB. tan 45 = 1.0) of the surface normal.

[vz > 0.0] & [rz > 0.5e-11] & [pz/sqrt(px*px+py*py) > 1.0]

Comment: Run-time error will occur if you test this on a particle with px = py = 0.0. To avoid this, you can add the condition '`[abs(px)+abs(py) <> 0.0]`' to the front of the expression (where it will be tested first).

Example 3. Store data for the projectile alone, but only when its kinetic energy is greater than 10 eV and less than or equal to 50 eV.

[rw = 0] & [ke/ep > 10.0] & [ke/ep <= 50.0]

Example 4. (Applies to a composite lattice consisting of Cu + Ni atoms.) Store data for Cu atoms (mass 63.5) but not Ni atoms (mass 58.7). First you must convert amu to kg (1 amu = 1.66056E-27 kg). Then only Cu atoms have mass > 63 amu (= 1.046E-25 kg):

`[ms > 1.046E-25]`

Comment: Don't use equalities like `[ms = 1.0E-25]` with floating point numbers.

Example 5. Store data only after 200 fs (2*10-13 s) have elapsed.

[ti > 2e-13]

Example 6. Store data for atoms located within a radial distance of 15 Angstrom from a cylinder axis perpendicular to the surface which passes through the origin (0,0,0).

[rx*rx + ry*ry < 2.25e-18]

Comment: Can be expressed equivalently as: `[sqrt(rx*rx+ry*ry) <= 1.5e-9]`

Example 6. Store data for (scattered) projectiles that have an altitudinal angle of 44-46 degrees, and an azimuthal angle of 10-12 degrees.

[rw = 0] & [altd >=44.0] & [altd <= 46.0] & [phi >= 10.0] & [phi <= 12.0]

Comment: normally it is better to filter angualr variables at the end of the simulation using Winnow (you may need the data you removed later).

User-Programmed Option

What it Does

The user-programmed option allows you to specify more precisely what kind of output is stored on disk. To use this feature, you need to be familiar with Winnow's query language. The query language defines a conditional expression, which is used at run-time to determine what output should be written to the disk. This process is analogous to the Filter operation used by Winnow, and can be regarded as a 'rough cut' of the information generated by the system, mainly for the purpose of saving disk storage space.

A simple example of a conditional expression is: [rz > 1.0E-10]. In words, this means: "the z-coordinate is greater than 1.0 Å". When writing output data to disk, Kalypso will first check whether this expression is TRUE or FALSE for each particle. If the expression is TRUE, an output record is stored for that particle; otherwise no action is taken. The output file (dynvars.snk) thus consists of records which satisfy the user-programmed conditional expression. In this example, the output file would consist of records for all particles (including the projectile) that were found to be located more than 1.0 Å above the surface.

For a detailed specification of the query language, see the relevant Winnow Help topic. The following summary is not comprehensive.

How to Specify a User-Programmed Expression

In building up conditional expressions consisting of dynamical variables, you can use any combination of the following predefined symbols: rx, ry, rz (position), vx, vy, vz (velocity), px, py, pz (momentum), lx, ly, lz (angular momentum), ke (kinetic energy), ti (time elapsed), ms (particle mass), all of which are expressed in SI units. The expression ke/ep (where the symbol ep is the proton charge), is used to express energies in eV. The symbol pi represents the number 3.1416, and is useful in trigonometric expressions. Other useful symbols are:

(a) rw: the 'row number' which indexes the particle. For the projectile, rw = 0. For the 1^{st} atom defined in the .TRG file, rw = 1; for the 2^{nd}, rw = 2, and so on.

(b) rn: the 'run number' which indexes the run of your simulation (a simulation will normally consist of many different runs, corresponding to different impact parameters). The run number corresponds to the current row of your .IMP file, starting with rn = 1 for the topmost row.

(c) alt, altd:the altitudinal angle of motion expressed in radians or degrees respectively.

(d) phi, phid:the azimuthal angle of motion expressed in radians or degrees respectively.

The logical operators are & (logical AND), | (logical OR) and ! (logical NOT).

Conditional expressions must be enclosed within square braces [thus], which may be nested to any desired degree. Floating point numbers must be specified in the format: 1.0 or 1e10, but not plain 1. Upper/lower case expressions are equivalent.

Examples of User-Programmed Expressions

User programmed expressions are restricted to 255 characters in length, including comments. Comments, which are ignored, are nested between curly braces {thus}. Blank spaces are also ignored, and can be used to improve readability. A few examples of user-programmed expressions will now be given, with explanations.

Specifying Particles

To store data for projectile and target atom #1 only, use: `[rw = 0] | [rw = 1]`. The meaning of this expression is: "rw = 0 or rw = 1" (note the use of '|' for the logical 'or' operation), which refers either to the projectile (rw = 0) or to target atom #1 (rw = 1). You might use this expression if you are only interested in following the trajectories of the projectile and its initial collision partner. Equivalent expressions are: [rw < 2] and [rw <= 1], since rw is always greater than or equal to zero. Suppose you have a target consisting initially of 81 surface atoms (9 by 9 rows). Then, you could store data for these atoms alone (excluding the projectile) by specifying the expression: `[rw > 0] & [rw < 82]`. Note the use of '&' for logical 'and'.

Specifying Energy Ranges

Filtering out particles in a certain energy range is easy, and involves the imposition of conditions on the term `ke/ep` (particle kinetic energy expressed in eV; ep is the proton charge). For example, to save information only for particles moving with kinetic energies >= 1 eV and < 10 eV, use the expression: `[ke/ep >= 1.0] & [ke/ep < 10.0]`. To save information for projectiles moving in the same energy range, apply the expression: : `[rw = 0] & [ke/ep >= 1.0] & [ke/ep < 10.0]`.

Specifying Angular Ranges

Frequently, the simulation is concerned with particles emitted into a certain angular range. You can approach this problem in more than one way, but only the most elegant is considered here. This involves the use of the predfined variable `altd`, which represents the altitudinal angle expressed in degrees.

(1) For example, suppose you are interested in altitudinal angle (psi) emission of all target atoms (i.e. [rw > 0]) in the range 75-90 deg. (where 90 deg. is the surface normal). In order to be counted, a particle must satisfy several conditions:

(a) [rz > 4.0E-10] (here we assume that emitted atoms are more than 4.0 Å above the surface);

(b) [pz > 0.0] (particle is travelling away from the surface);

(c) $\varphi > 75$ degrees (the altitudinal emission angle, φ, never exceeds 90 degrees).

The final user-programmed expression is therefore of the form: [rz > 4.0E-10] & [pz > 0.0] & [altd > 75.0].

Model File

Introduction

The Model file stores the parameters which specify the physical model on which your simulation is to be based. These include the functional form of the short range (repulsive) and long range (attractive) interaction potentials, lattice vibration (temperature) effects and the energy parameters (the surface/bulk binding energy and the termination energy).

The Model file is quite complex, so work systematically. Help on its fields can be found in the **User Guide**, and in the **Spider UI** file. Parameters for the attractive potentials are found in chapter 10 of the **Simulation Primer**.

Some general points to note:

- If your target is an element, you can ignore any fields which make reference to the type B atoms [Z(B), V(B-B) and V(A-B)].
- Don't forget to set the cut-off distance before you view the graphs of the potential
- The switching function should normally be used, and Rsw should be located below the cut-off distance, but above the last shell of atoms within the range of the potential.

Model: Screened Coulombic Potential

Summary

The user of Kalypso is required to choose the type of potential employed by the simulations. The short range potential used by Kalypso is classified as a screened Coulombic potential. There are 3 varieties of these in common use, which are named after their developers: the (Ziegler-Biersack-Littmark) ZBL potential, the Moliere-Firsov potential and the Moliere-Lindhard potential. The Screening Length Correction is an adjustable parameter which may be used to modify the range of these potentials to improve their accuracy.

The user has to select (a) one form of screened Coulombic potential to represent the projectile atom-target atom interaction; (b) one form of screened Coulombic potential to represent the target atom-target atom interaction; (c) two 'screening length adjustment' factors which typically fall in the range 0.7-1.0 (1.0 is normal for the ZBL potential). The target-target and projectile-target interactions do not have to be of the same form.

Review of Screened Coulombic Potentials

The short-range, repulsive interatomic potential takes the form of a screened Coulombic potential:

$$V(r) = Z1.Z2.e2.\rho(r/a)/4\pi\varepsilon r$$

The screening function $\rho(r/a)$ has the properties $\rho(0) = 1$, $\rho(\text{infinity}) = 0$. It is normally represented in the general form:

$\rho(r/a) = \Sigma\, \alpha[i].\exp(-\beta\,[i].r/a)$,

i (= 1-3 or 1-4)

where the $\alpha\,[i]$ and $\beta\,[i]$ are coefficients ($\Sigma\, \alpha[i] = 1$) and *a* is the screening length (defined below).

The Moliere potentials and the ZBL potential define their respective parameters ($\alpha\,[i]$, $\beta\,[i]$ and *a)* in various different ways.

The Moliere potential has two variants, according to the way in which the screening length (*a*, in A) is chosen:

a = 0.4685/(Z1^0.5+Z2^0.5)^2/3 (Firsov form)
a = 0.4685/sqrt(Z1^0.33+Z2^0.33) (Lindhard' form)

In practice this distinction is irrelevant since (judging from the literature) every user of this potential seemingly adds his or her own **screening length correction**. The screening length correction is a 'correction' factor (< 1.0) which is used to scale the screening length parameter in screened Coulombic potentials, particularly the Moliere potential. A typical value for this correction for the Moliere potential would be 0.8, usually chosen by fits to experimental data e.g. impact collision ion scattering spectrometry.

The screening length (*a*, in A) for the ZBL potential is defined as:

a = 0.4685/(Z1^0.23 + Z2^0.23)

For the ZBL potential, you would normally choose a value of 1.0 for the screening length correction, unless you have reason to do otherwise.

You should be wary of changing the screening length correction parameter arbitrarily, as it has enormous effects on the potential and unrealistic values could conceivably undermine the credibility of the simulations.

Which potential is best? This question cannot be answered with rigour. However, if you are not going to search for an optimum screening length correction, the ZBL potential is probably the easiest choice because it is normally used in 'unadjusted form', i.e. with a screening length correction of 1.0.

However, the Moliere potential with adjusted screening length may well be closer to reality than the unadjusted ZBL potential (see Fauster, Hartwig and Duerr, Appl. Phys. A 45 63 1988 for a review of this topic).

The problem is knowing what correction to apply to the Moliere screening length. In early work a factor of 0.8-0.9 was typical, but more recently even values < 0.7 seem not to excite comment: e.g. 0.68 was used for 4 keV Ar-Ni in Phys. Rev. Lett. 69 1992 1391.

Such 'correction' factors have an enormous effect on the potential, because they involve exponentiation. In fact, the effect of the corrections may be more significant than some of the terms in the original potential!

The ZBL potential is generally used without any screening length correction, and this fixed form may be regarded as an advantage or a disadvantage, depending on how highly you rate the potential. For the Ar-Cu system at least, this author notes that the ZBL potential fits closely the ab initio potential calculated by Broomfield et al. (Surface Sci. 202 (1988) 320). However, the classical MD studies of the Ar-Cu system by Harrison's group were mostly carried out using the Moliere potential (with a screening length correction of around 0.85).

In any serious study of an ion-surface collision one should experiment a little with the potential, to see whether this has any influence on the outcome of the simulation averages.

Model: Lattice Vibrations

Lattice Vibrations

You can choose to include 'lattice vibrational effects' in your simulation. What this means is that when runs, small displacements calculated according to the Debeye lattice model will be applied to the atoms in your target lattice. The displacements are applied by Kalypso randomly along each coordinate axis using a Gaussian distribution of deviates d**r** (= **i**dx + **j**dy + **k**dz). The random number generator is seeded according to the option chosen by the user in Kalypso.

The calculation of vibrational displacements requires specification of the temperature of the target lattice, as well as the lattice bulk and surface Debeye temperatures. Debeye temperatures are empirical parameters used to describe lattice dynamics and thermodynamics. (Click here for tables of bulk and surface Debeye temperature data.) The definition of the Debeye temperature,which is needed to characterise thermal displacements, is different for bulk and surface atoms. The latter are often given a perpendicular Debeye temperature, and a parallel Debeye temperature, depending on the direction of displacement with respect to the surface.

If the Lattice Vibrations option is not selected, the simulations will always use the 'ideal' lattice specified by the .TRG file coordinates. **You should generally disable the Lattice Vibrations option when experimenting with parameter settings, in order to eliminate the 'random' element associated with the thermal displacements** (which prevents any two simulations from ever being quite the same).

Model: Specific Energies

Summary

The user is required to specify three parameters with the dimension of energy: the Surface and 'Bulk' Binding Energies and the Termination Energy respectively. The first two parameters represent the change in potential experienced by an atom as it leaves the surface/side faces of the target crystallite. The last parameter controls the termination of the simulation. See below for further details.

Surface Binding Energy

This parameter should not be needed in a classical dynamics simulation if the interaction potential completely described the system dynamics. However, if you wish to include a surface binding energy term (E*s*) in your simulation, specify it here.

The surface binding energy correction implemented by Kalypso takes the form of subtracting an amount E*s* from the kinetic energy associated with the vertical (z) velocity component of particles ejected from the lattice, i.e. moving beyond the cut-off distance of the potential. The net effect of the correction is to apply a 'refraction' to the trajectories of particles that leave the surface. Particles whose z-velocity is too small to escape the surface energy barrier are reflected elastically (i.e. vz is changed to -vz). Specify a value of 0.0 for the Surface Binding Energy if you don't want to include a surface energy correction (the usual practice).

'Bulk' Binding Energy

The name for this parameter may be confusing. It is similar to the Surface Binding Energy (see preceding), but refers to an energy correction applied when the partice leaves any of the target sides (excluding the top and bottom surfaces). 'Side-face binding energy' might be a better description. As with the surface energy correction, particles may be either reflected or refracted by the barrier. This feature will rarely be needed.

Termination Energy

The significance of the Termination Energy is that the simulation performed by Kalypso will terminate if the maximum kinetic energy of all particles within the lattice becomes equal to, or falls below the threshold specified here. For sputtering studies, a typical termination energy would be 2-4 eV, but this would be much higher for ion scattering spectrometry (ISS).

The **Test Projectile Only** check-box option allows you to base the termination criterion on the energy of the projectile only (this may be useful for ISS simulation, for instance). This option is disabled by default.

The remaining remarks in this section refer to special situations (a) when you want to set a Termination Energy which is of the same order of magnitude as the projectile primary energy, or (b) when the system being modelled has very few atoms in it.

Special Considerations

(a) High Termination Energies

For ISS (and other techniques involving detection of fast atomic projectiles), you need to think carefully about the physics of the scattering process when setting a Termination Energy. Ideally, the termination energy should be set as high as is feasible, in order to minimise computation time. However, you need to be careful not to terminate hard collisions too early.

Consider a primary projectile (mass m1, energy E0) incident on a target with mass m2.

In a direct impact (centre-to-centre) collision, the maximum energy transferred to the target (E*max*) is:

E*max* = E0*(4m1.m2)/(m1+m2)2 .

At the turning (apsidal) point of the collision, the energy transferred to the target atom is around half this amount. Also, at the turning point of the collision, the **kinetic energy of a light projectile is zero**. Most of the incident energy is momentarily stored as potential energy. Thus, you need to be sure to set the termination energy less than half of E*max* as defined above, or the calculation is liable to terminate spuriously in the middle of a hard collision. Kalypso checks your inputs for this problem, and notifies you, unless the Minimum Termination Time parameter (in the Run file) is greater than zero: in this case, Kalypso assumes that you have set this parameter at such a value that the problem just described is avoided (i.e. greater than the time needed to complete the collision's close encounter).

For example, E/E0 is around 0.9480 for the Ar-Cu system, and 0.2230 for the He-Cu system. Suppose the projectile energy is 1000 eV, you need to set the termination energy somewhat below E*max*/2 = 948/2 eV for Ar and 223/2 eV for He. Otherwise, direct-impact collisions would cause the simulation to terminate near the apsis (turning point).

It is especially important to avoid the fallacy of setting a termination energy just below the experimental region of interest, unless you have specified a suitable Minimum Termination Time. For example, the ISS peak for 90 degree scattering of 1000 eV He+ from Cu occurs at about 880 eV. However, you must set the Termination Energy below 110 eV, based on the above considerations. However, the use of a Minimum Termination Time, if correctly set, can save the trouble of following the simulation down to this safe energy.

(b) Few-body Systems

Particles which were not 'contained' by the lattice during the simulation, and emitted atoms, will NOT be included in the energy evaluation. Why not? Well, these are moving in free space, and might have considerable velocities - so there is no point in waiting for them to slow down.

For few-body systems it is possible for all atoms to be dispersed with high kinetic energy beyond the domain of the original 'lattice' (if it can be called that).

Under such circumstances, a low energy cut-off is not a reliable control parameter. It is advisible to put the energy cut-off to zero (literally) and rely on the Termination Time parameters (.RUN file) to control the calculations.

Impact File

Overview

The Impact file is used by Kalypso in setting up the projectile's initial position at the start of each simulation run. Each line of data in the Impact file defines a set of (bx, by, z0) coordinates. Kalypso takes the (bx, by) location and combines it with the incident angle information found in the Run file to generate an appropriate initial (x, y) projectile position.

The initial projectile vertical position (height above the surface) is computed as the (z0 parameter) + (the z coordinate of the Target file anchor atom).

Overall, then, the projectile initial coordinates are set as follows:

x[0] = x[1] + z0*cotan φ * cos ϕ + bx
y[0] = y[1] + z0*cotan φ * sin ϕ + by
z[0] = z[1] + z0 (click for symbol explanations).

The set of (bx,by,z0) data in the .IMP file conceptually defines a mesh or grid of originating points (x[0], y[0], z[0]) which have a one-to-one correspondence with a specific mesh of 'impact points' (x[1] +bx, y[1]+by, z[1]) at the sample surface regardless of the angular variables.

This set of impact points (each of which is a 3-dimensional analogue of the impact parameter) should be representative of all possible points, so it should cover a symmetrically complete lattice zone, which can either be defined by the bulk unit cell boundaries, or more usually, by a smaller zone which is equivalent by symmetry, which we term here the 'reduced impact zone'. Some examples of reduced impact zones are given in the Discussion below.

For non-normal incidence, the shape of a reduced impact zone **does** change if the target azimuthal alignment changes with respect to the beam (either because the beam was rotated, or because the target was rotated). (See the Discussion below for more on this.) **Unless you always base your .IMP file on the bulk unit cell (which is inefficient for high symmetry alignments), you need to make a separate .IMP file for every azimuthal configuration.**

Setting Up .IMP Files

Impact parameter coordinates (bx, by) will be written to a .IMP file within the ranges specified in the Impact dialog box input lines, viz:

bx(min) <= bx <= bx(max)
by(min) <= by < by(max) .

The total number of coordinates that will be written to the .IMP file is the product of the '**Number of Impacts**' parameters for the x and y axes respectively (N*x**N*y*).

The inputs must satisfy the condition that bx(min) < bx(max), and by(min) < by(max). If the 'number of impacts' along an axis is 1, then the lower limit (bx(min) or by(min)) is used.

The algorithm used to generate .IMP files is constructed with surface periodicity in mind. It aims to 'pepper' (bx, by) uniformly in the interval specified, starting from the lower limit: the border points are considered to be physically equivalent, as on a periodic surface, so they are only 'hit' once (at the lower limit).

For example, if the range along one axis is specified as 0.0 (min) to 1.0 (max), and the Number of Impacts = 2, the .IMP file will record the points 0.0 and 0.5 for that axis, **not** 0.0 and 1.0. If the Number of Impacts was 10, the points 0.0, 0.1...0.9 would be recorded.

Press the **Refresh** button to get a visual representation of the distribution of impact points that will result from the current dialog box parameters. Use the Impact|Display/Orient option to inspect an existing .IMP file, and modify it (by truncation or reorientation) if necessary. (Use **Copy** to copy the graphic to the Windows clipboard.)

The Templates button displays a Help topic which contains data for creating Impact files for a number of standard projectile-target configurations.

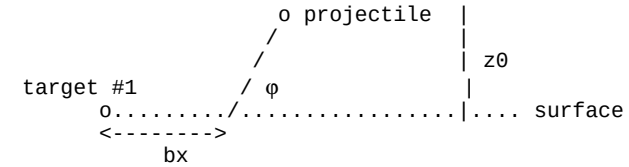
A typical Kalypso simulation will involve sampling many impact parameters within a symmetry-reduced zone of the surface primitive cell, and might require typically 300-500 such impacts for a sputter yield study, or tens of thousands for an ISS (direct impact collision) study. The number of samples required depends on the process cross-section and the level of statistical certainty you desire.

If the projectile in your simulation is not incident along the surface normal, you will need to know what is the default azimuthal orientation of the lattice .TRG file.

Discussion

The impact parameters (bx, by) specified in the IMP file refer to the first target atom specified in the TRG file (located at (0,0,0) by default). Kalypso will automatically position the projectile relative to this atom as follows:

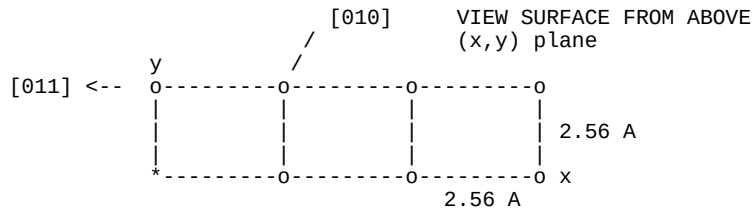
Viewed from side, in the xz plane



To use Kalypso effectively, you must have a thorough understanding of the symmetry properties of your surface. **In particular, what is important is the periodicity of surface structure from the viewpoint of the incoming projectile.** When designing an Impact file, you first need to identify a representative region of your surface which suffices to represent all possible scattering configurations. A surface region defined by the borders of the bulk unit cell always serves to satisfy this condition. But if there are elements of symmetry present in the experimental set-up it is possible to find a 'reduced symmetry zone' of smaller area. (These ideas are discussed in the following paper: D.E. Harrison Jr., C.E. Carlston and G.D. Magnuson, Phys. Rev. 139 (1963) A737.)

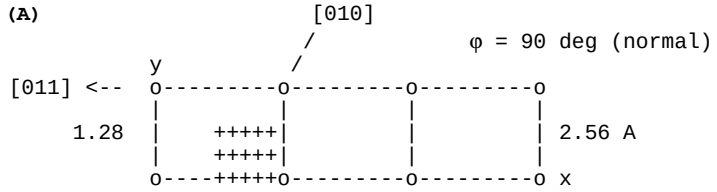
Here are some examples for the Cu(100) surface that illustrate the idea of a reduced symmetry zone. (Read this while looking at a picture of the Cu(100) surface.)

The Cu fcc lattice has lattice constant a = 3.62 A. The **surface** layer consists of square cells of nearest neigbours (d = 2.56 A or 3.62/sqrt(2)):

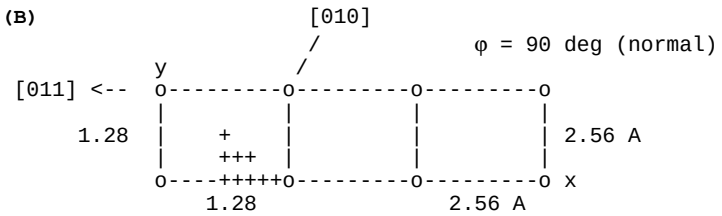


(The above view shows the coordinate system relative to the target atomic positions which SPIDER generates by default.)

For **normal** projectile incidence, you might think initially that we need to sample only ¼ of the area of the 'surface unit cell', i.e. the shaded region (0,0) to (1.28,1.28):

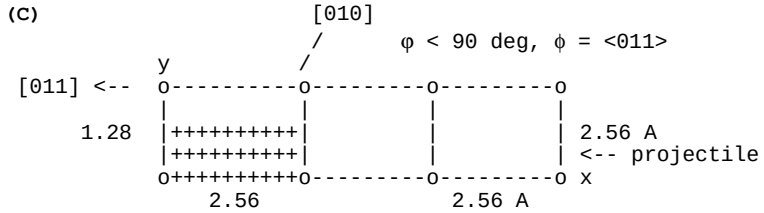


However, the area of the above zone can still be be reduced by 50% without loss of information. This new region has a triangular shape defined by three corners of the symmetry zone shown in the previous diagram (A):



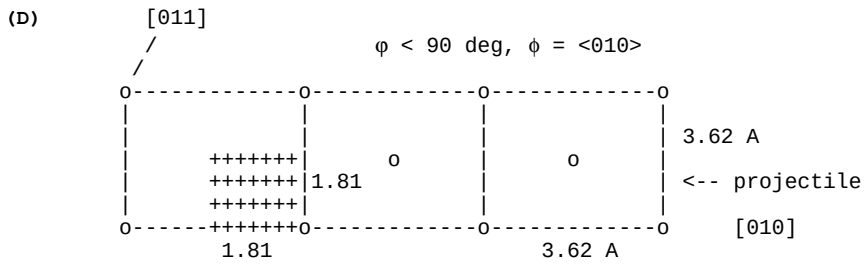
[Note: a similar diagram can be derived for bcc(100) surfaces.] It is more computationally efficient to sample the area shown in (B) than in (A), because otherwise you are (in effect) duplicating each run. Lattice vibration and 'containment effects will, however, break the symmetry to some extent and give you different results from 'symmetrically equivalent' trajectories. You can generate a mesh equivalent to (B) using Spider's Impact|Display/Orient option with data from a template.

However, for projectile incidence parallel to a <011> direction, we need to sample ½ of the surface cell, i.e. the shaded region (0,0) to (2.56,1.28):



This requirement arises because of the presence of the second Cu atomic layer. (It now matters on which side of the surface cell the projectile arrives.)

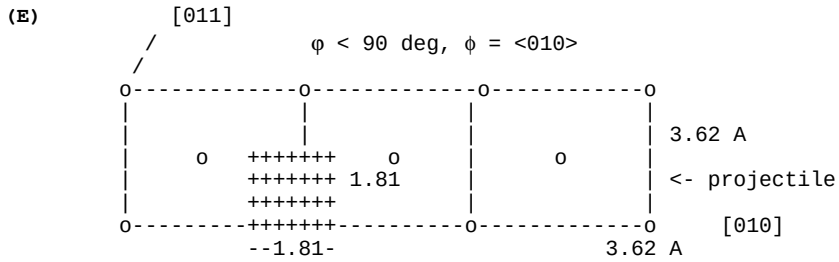
For a projectile incident along a [010] azimuth, the symmetry-reduced sampling zone is square, with dimensions 1.81x1.81 A2:



Note that the sampling area (shaded) is the same size as that for the <011> azimuth (2.56x1.28 = 1.81x1.81). To carry out a simulation involving the [010] azimuth, you would have to rotate the coordinates in the TRG file produced by SPIDER, using the Target|Display/Orient option (or, alternatively (and not recommended), rotate the IMP file coordinates).

For azimuthal directions other than [010] or [011] type, you have to sample the ENTIRE bulk unit cell i.e. a range of impact parameters from (0,0) to (3.62,3.62) for [010] incidence. This is because you get no opportunity to exploit symmetry.

Although the impact parameter ranges have been referenced to the target atom at the origin [i.e. (0,0) in the surface plane], it is not necessary to start the impact parameter range at this value, because the surface structure is periodic. It would be equally acceptable (for instance) to span a distance (0,0.5) to (1.81,2,31) (the shaded area below) rather than (0,0) to (1.81,1.81) (the shaded area in the preceding figure). The point is to get the total area correct:



To sum up this discussion: you have to be very careful when preparing your IMP file to make sure that it really does span the correct zone of the surface. For normal incidence on a cubic (100) surface, the correct symmetry zone to sample has a triangular shape. It is permissible, but inefficient, to sample a larger area. But in your first testing of the Simulation Kit you might prefer to do things this way because it is quite easy to generate the necessary .IMP file e.g. using a template.

(The preceding remark about inefficiency only holds true for cases when the target used in the simulation does not incorporate thermal vibration effects. The inclusion of (random) thermal vibration effects means that each trajectory samples a unique configuration, regardless of the symmetry of the static lattice.)

One recommended check of any IMP/TRG file combination is to superimpose the IMP file (x,y) coordinates on a diagram of your target lattice using the Target|Display/Orient option.

Impact Specifications

See Also: ~~Impact File~~

Overview

In the Impact Specifications dialog box, the user is required to provide data which will be used to generate an Impact (.IMP) file. For a general discussion on Impact files, or for information about the significance of the **bx**, **by** and **No. of Impacts** fields, click here. The remaining fields used for the Impact file are explained below.

Press the **Refresh** button to get a visual representation of the distribution of impact points that will result from the current dialog box parameters. Use the Impact|Display/Orient option to inspect an existing .IMP file, and modify it (by truncation or reorientation) if necessary. (Use **Copy** to copy the graphic to the Windows clipboard.)

Impact Density

This feature is obsolete and can be ignored.

Specify (x, y) angle

This feature is provided to allow .IMP files to be built up:

(a) for surfaces whose surface mesh vectors are not mutually perpendicular, especially (111) surfaces.
(b) For non-rectangular reduced impact zones.
Note: the minimum bx and by parameters must be 0.0 if you specify a non-perpendicular (x,y) angle. This restriction is imposed by the logic of the generating algorithm. To work around this restriction, you can use the Impact|Display/Orient command, which will let you translate the impact points mesh away from its original location. Alternatively, you can use the Target|Display/Orient command to shift the target coordinates in a similar fashion.

For (100) and (110) surfaces this parameter will always have the value 90 (degrees).

For a worked example of the task of creating an Impact file for a (111) surface, see the Display/Orient Help topic. For impact file templates click here.

Projectile z0

The **z0** field (specified in Angstrom) determines the initial projectile height above the surface at the start of the simulation. (The initial position in the xy plane depends on the current value of **bx** and **by**, as well as the projectile angles of incidence specified in the Run file).

The actual z-coordinate of the projectile is computed as the sum z-coordinate of the first ('anchor') atom in the Target file (which is by default at (0,0,0)) plus z0: **rz[0] = rz[1] + z0** . The recommended value for **z0** is 3.0 A (although 2.0 A is satisfactory for many systems and was used by the first release of).

Note: To provide flexibility, the value of **z0** is allowed to range from -1E6 to +1E6 A. A negative value of z0 will place the 'projectile' inside the target. A zero value of **z0** will, however, cause to crash. A value of z0 < 2.0 A will generate a warning message, in case it was set inadvertantly.

Impact Parameter

The impact parameter (b) is the distance off a cylinder axis (z) which passes through the target and runs parallel to the projectile initial velocity vector, **v0**.

In a cylindrical or Cartesian coordinate system, the net impact parameter, b, can be computed from its (x,y) components as: b = sqrt(bx*bx + by+by), where bx, by are the respective axial displacements. In a two body system, the value of b uniquely specifies the scattering process. However, for the purpose of simulation in solids it is convenient to define the trajectory in terms of bx and by.

Impact File Templates

The Impact file templates for a number (100) and (110) cubic surfaces are now found in Tables 6.x of the **User Guide**. The case of fcc (111) is discussed in chapter 6 of the same file.

For other types of target surfaces, you will have to work out the dimensions of the impact zone by yourself.

Note that the shape and size of the reduced impact zone is different for normal and non-normal directions of projectile incidence.

Reduced Impact Zone

The impact file defines a set of statistically representative points on the target surface. These represent a sample of points within a zone of irreducible symmetry, which will be called the reduced impact zone (RIZ) here.

Batch File Template

Function

The batch file template is useful for creating the batch definition files (*.bdf) used by Kalypso when it runs in a "batch" mode (multiple simulation projects executed sequentially).

The user has to input a name for the batch definition file, and a project stem name (i.e. without any file extension). Lastly, the number of simulation projects in the batch job is specified.

This routine will generate a skeleton batch definition file in which all input files have a common stem. You will need to edit this file by hand to customise it to your purposes. Normally every project in a batch job consists of 4 or 5 input files common to all of the projects, while the rest are unique to the individual projects: the editing stage is needed to specify the names of those files which change for different projects in the batch job. Output files normally need to be similarly edited.

For more information on running batch jobs, see the appropriate Help topic for Kalypso.

Velocity Reckoner

The Velocity Reckoner (on the Utilities menu) is a simple gadget that calculates the velocity of an atomic (or molecular projectile). The user is required to enter the atomic mass (in atomic mass units) and the energy (in keV) of the projectile. The corresponding projectile velocity (in A/fs = 10^5 m/s) is updated in the yellow box.

This velocity information is useful for setting a simulation timestep. The timestep should be chosen so that the fastest atom in the system moves ca. 0.1-0.2 A during the timestep.

Scattering Relations

Overview

The Scattering Relations gadget (on the Utilities menu) calculates useful scattering parameters, based on standard binary collision formulae. The scattering system consists of a projectile incident on a target which is initially at rest in the Laboratory coordinate system ('Lab system'). The results generated by this gadget are useful for interpreting ion scattering experiments.

Procedure

The user enters the projectile and target atomic masses (M1, M2) and an asymptotic laboratory scattering angle for the projectile (θ1 Lab).

The symbol E0 represents the projectile incident energy in the Lab system, while E1 and E2 represent the Lab energies of the projectile and target respectively, after the collision.

Any system of units can be used for M1, M2 and E1, E2. All angular variables are expressed in degrees.

On clicking the Evaluate button, the following quantities are calculated:

q2 Lab

This is the scattering (recoil) angle of the target.

q COM (1), q COM (2)

These are the scattering angles in the centre-of-mass coordinate system (by definition, these are identical for both particles). If M1 > M2, the Lab projectile scattering angle must lie between 0 and a maximum value (θ max = arcsin(M1/M2)). In this case there are two possibilities (1) and (2) for the COM angle, which still ranges from 0-180 degrees. (See a textbook on classical dynamics for an explanation.) For M2 > M1, there is only one COM angle per Lab angle, and you will see the message <None> displayed in one box.

E1/E0, E2/E0

These energy ratios reflect the energy transferred during the collision. If M1 > M2, there are two possibilities for energy partition. Otherwise, one box will display the message <None>.

These energy ratios are completely independent of the nature of the interaction potential or primary projectile energy.

Crystal Axes

Summary and Terminology

The Crystal Axes gadget (on the Utilities menu) is a tool which may assist you with crystallographic computations: especially those involved in reorienting the coordinates of a .TRG file. This Help topic assumes that the reader has some knowledge of crystallographic concepts.

The terminology used in this Help topic and elsewhere is as follows. Parentheses indicate Miller indices, e.g. (100). By convention, square brackets indicate directions (vectors); thus, the direction parallel to the x-axis is denoted by [100]. In the **cubic** systems (fcc, bcc, dia) only, the direction normal to an (hkl) Miller plane is equivalent to the vector [hkl]. Directions *parallel* to the (100) surface (which are aligned with surface atomic rows) correspond to [010], [011] etc. Triangular brackets are used to denote sets of directions which are symmetrically equivalent: <100> refers to directions like [100], [010] and so on. Curly brackets {} are used to denote sets of symmetrically equivalent Miller planes e.g. {100} refers to the set (100), (010) and so on. Rotation operations are expressed as R*xy* (meaning rotation in the xy plane) and so on.

Use of the Gadget

The user inputs the Miller indices of two surfaces (as integers). For cubic crystals (fcc, bcc, diamond), these indices represent crystallographic directions based on a Cartesian coordinate system in which the (x,y,z) axes lie parallel to the unit cell edges (a,b,c).

It is important to realise that this 'crystal' coordinate system only coincides with that in your .TRG file in one special case, namely when your lattice has a (001) surface (surface normal pointing in the z-direction) whose [100] rows have been rotated into the x-direction. We shall refer to the .TRG coordinate system as the 'Lab' system.

When the Evaluate button is clicked, the gadget calculates azimuthal and polar angles of the surface normals (in the crystal coordinate system) represented by the two Miller indices. The polar angle (θ) represents the degree of misalignment with the [001] direction or z-axis. The azimuthal angle (ϕ) defines a direction in the xy-plane, measured as an anticlockwise rotation away from the [100] direction or x-axis, which corresponds to $\phi = 0$. The 'inter-plane angle' (the angle between the 2 surface normal directions) is also calculated for the surfaces in question.

Note: for the [001] and [0,0,-1] directions, the azimuthal angle has an arbitrary value (represented here as 0.0). You should appreciate that no azimuthal rotation is involved in rotating between [001] and any other direction. However, the rotation does take place across a surface of constant ϕ.

Applications

The Crystal Axes gadget has two applications. First, it can be useful in an experimental context, when you have a real crystal mounted on a goniometer. Second, it can help you develop a strategy for reorienting the coordinates in a Target file to create a non-standard surface. The first application is perhaps self-evident. The second will be considered in detail in this Help topic.

The information provided by this gadget does not completely solve the Target file orientation problem, but it helps. It is also necessary to appreciate the idea of a crystallographic zone: a direction defined by the intersection of a particular series of Miller planes. Important planes generally belong to several zones. The planes normal to the directions [001], [011], [021] and [010], for example, lie in the same crystallographic zone. If you draw a number of such vectors (surface normals), they appear like the spokes of a 'wheel'. To reach [012] from [001] involves aligning the crystal's [010] rows with the Lab y-direction (aligning the 'wheel' normal to its axis), then performing an R*yz* rotation (turning the 'wheel' on its axis). To reach [011] from [001] involves the same initial alignment, but a different magnitude of R*yz* rotation.

Applying these principles requires a certain amount of practice (or trial and error). The following remarks are hints rather than prescriptions. (No doubt many readers are familiar with the experimental process of orientation using a goniometer in conjunction with Laue diffraction.)

The general orientation task can be broken down into three steps:

(1) Find the plane which contains your current surface normal (usually [001]) and the axis you wish to rotate to;
(2) Align this plane with the Lab zx (or zy) plane by means of a a suitable Rxy rotation.
(3) Perform the inter-plane angle rotation R*yz* (or R*zx*) as indicated by the Crystal Axes gadget.
Because of symmetry, there is generally more than one way to achieve a particular orientation goal. However, if your inter-plane angle is not equal to either the inter-plane polar angle ($\theta 1 - \theta 2$) or the inter-plane azimuthal angle ($\phi 1 - \phi 2$) you will not be able to use this method, since a suitable rotation for step (2) cannot be found.

Example

Task: Starting from a {100} surface, orient it to a {211} surface.

Procedure #1: Define the first surface as (001) and the second as (211), and calculate angles by clicking Evaluate. We find the angles for the second surface to be (65.9, 26.5). This implies that we have to rotate

the first surface in the xy plane so that its {100} rows are 26.5 degrees away from the (x,y) axes. Then we need to carry out a 65.9 degree polar (R*zx*) rotation. The inter-plane and inter-polar angles are identical in this example because the R*xy* and azimuthal planes coincide.

Procedure #2: Define the first surface as (001) and the second as (112), and calculate angles by clicking Evaluate. We now find the angles for the second surface to be (35.2,45). This implies that we have to rotate the first surface in the xy plane so that its {100} rows are 45 degrees away from the (x,y) axes. Then we carry out a 35.2 degree polar (R*zx*) rotation. The inter-plane and inter-polar angles are again identical in this example because the R*xy* and azimuthal planes coincide. Since Spider generates {011} surfaces in the configuration required (with {100} rows at 45 degrees to x,y axes) this procedure turns out to be simpler, and also involves minimal tilting of the original lattice (by only 35.2 degrees).

The author finds it helpful to visualise a ray or beam emanating from the crystal in a direction normal to the plane of interest. The orientation task is then to align this ray, through appropriate crystal rotations, with the Lab z-axis.

To locate the crystallographic zones which you can exploit for your own orientation tasks, you may need to consult a book on crystallography which includes stereographic projections. From such a diagram it will be apparent that any crystallographic orientation can be achieved by starting from a (001) surface, which has the advantage that its key crystallographic rows, namely [010] and [110] (and their equivalents by symmetry), are relatively easy to identify.

References

A.M. Glazer, **The Structures of Crystals**, (IOP Publishing, Bristol, 1987). [Good introductory text.]

M.W. Roberts and C.S. McKee, **Chemistry of the Metal-Gas Interface**, (Clarendon Press, Oxford, 1978). [Chapter 3 is an excellent review of the crystallography of metals, with other key references.]

Pair Potential and Apsis

Summary

The Pair Potential gadget calculates:

(a) Screened Coulombic potentials (in eV) according to the specifications of the user. The potentials supported are the (Thomas-Fermi-Dirac-) Moliere potential (with either Lindhard or Firsov definitions of the screening length), and the ZBL potential. The user has to supply the atomic numbers of the two interacting atoms, their radial separation (in A) and a screening length correction (a dimensionless parameter, usually in the range 0.8-1.0, which is used to adjust the strength of the potential).

(b) The apsidal distance (distance of closest approach) for an elastic binary collision specified by the parameters in (a), in addition to the masses of the projectile (m1) and target (m2) atoms respectively, the collision impact parameter (in Angstrom), and lastly the incident energy (in eV).

A graph of the various screened Coulombic potentials is also displayed. (Note: you can zoom in on the display by selecting an area using the mouse.)

Definitions

A screened Coulombic potential (expressed in eV) takes the general form:

$V = Z1Z2e/(4\pi\varepsilon\ r)*\Phi\ (r/a)$

The screening function, $\Phi\ (r/a)$, and the screening length (a) are defined as follows:

ZBL Potential

Φ (r) = 0.028171exp(-0.20162r/a) +

0.28022exp(-0.40290r/a)+
0.50986exp(-0.94229r/a)+
0.18175exp(-3.1998r/a).

a = 0.46850 / (Z1^0.23 + Z2^0.23) (Angstrom)

Moliere Potential

Φ (r) = 0.35exp(-0.3r/a) +

0.55exp(-1.2r/a)+
0.1exp(-6.0r/a).

a = 0.46850*(Z1**2/3) + Z2**2/3)**(-0.5) Lindhardt screening length (Angstrom)

a = 0.46850*(Z1**0.5 + Z2**0.5)**(-2/3) Firsov screening length (Angstrom)

Apsidal Distance

The Apsidal distance, Ra, is defined by the equation of motion of the scattering system at the turning point of the collision:

1 - V(Ra)/Er - sqr(p/Ra) = 0,

where V(Ra) is the potential at a distance Ra, Er is the centre of mass (COM) energy of the collision (equal to the incident Lab energy scaled by a factor m2/(m1+m2)), p is the impact parameter. The above equation can be solved using Newton's method. An upper limit of 5 A is assumed by Spider - you won't be able to get a solution if the solution is located outside this range.

The inputs required by the dialog box are the atomic weights of the projectile (m1) and target (m2) respectively, the incident projectile energy in the lab frame (the target is assumed to be initially stationary), and the impact parameter (in A). The outputs are the COM energy (in eV) and the apsidal distance (in A).

Click on the "Evaluate" button to update these outputs after you edit the input parameters.

Inelastic File

Overview

The Inelastic file is an **optional** component of a simulation project. New users of Kalypso should probably ignore it. The Inelastic file specifies the parameters which are used for modelling inelastic energy losses (responsible for so-called "electronic stopping") in a collision process. Three different inelastic loss models are available, which may be used singly or in any arbitrary combination. The models are:

(a) Lindhard-Scharff-Schiott (LSS) model
(b) Oen-Robinson (OR) model
(c) Shapiro-Tombrello (ST) model

The physical background to these models is discussed in the Simulation Primer (in the `\docs` directory). The Compute K(LSS), K(OR) gadget helps you compute the parameters used for the first two models.

Click on the 'Add Pair' or 'Add Atom' buttons on the LSS, OR or ST tabs to add another set of parameters, or the corresponding 'Clear' buttons to delete them.

Implementation of the models

The implementation of each of the inelastic loss models needs to take into account the fact that different individual or pairs of atoms are associated with different inelastic loss parameters. This is achieved in Kalypso by "tagging" each set of parameters with a number, which is used like an index or hashing function at runtime to look up the appropriate set of parameters.

For the LSS model, which represents electronic stopping of an atom by a continuum target material, the tag is simply the atomic number of the atom which is being stopped.

For the OR and ST models, which predict the inelastic energy loss in a binary collision event, the tag is the product of the atomic numbers of the participating atoms, i.e. Z1*Z2. The tag for an Ar-Ar collision event is thus 324, while for an Ar-Cu event it is 522. Up to 10 different tags may be used. (This scheme permits the tracking of inelastic events between the projectile and the different components of a target with 1-3 different kinds of atoms in it.)

For all models, tags of zero ('0') and their associated parameters will be ignored by the simulation at run-time. You can use this feature to disable parameter sets without deleting them. Duplicate tags are allowed but will also be ignored (only the first match is used). Invalid tags (i.e. tags that never occur in the simulation) are allowed, but have no effect.

The "scale" parameters for the LSS and OR models, and the "dE" parameter for the ST model, can be modified to partition the inelastic energy loss over 2 or more models. For example, an equipartition of the energy loss between the LSS and OR models can be achieved by setting both LSS and OR scale parameters to 0.5. A scale factor of 0.0 is equivalent to a null tag parameter (see preceding).

The electronic stopping effects are applied to the system in the following order: ST, OR, LSS. For example, the OR energy loss is calculated after the ST energy loss (if any) has been removed from the system, but before the LSS loss (if any) has been removed. (There is no particular significance in this order.)

See also Chapter 7 of the User Guide.

Important

In order to incorporate inelastic effects into your simulation model, you must also select the model(s) (LSS, OR or ST) you wish to use in 's Simulation Options dialog box. All inelastic models are disabled by default. Only those you select will be used. You don't need to supply parameters, realistic or otherwise, for inelastic models which you don't want use in your simulations.

LSS Model

Lindhard-Scharff-Schiott Parameters

See the Inelastic File overview for general remarks about "tags" and the implementation of inelastic loss models in Kalypso.

Z1: the atomic number of the moving projectile or target atom which is being stopped by your target (this value is used as a tag only)

K(LSS): a coefficient (units: eV fs / A2) which reflects the stopping power of the target matter for an atom with atomic number Z1. K depends on several other parameters, and can be computed most easily by clicking on the "Compute K(LSS), K(OR)" tab of the input dialog box.

Scale: this is a value by which K(LSS) is scaled at runtime in the simulation (a value of 1.0 thus has no effect).

Velocity threshold: this parameter sets a velocity threshold (in m/s) for the stopped particle below which LSS energy losses are ignored, i.e. losses are not calculated for slowly moving or static lattice atoms.

OR Model

Oen-Robinson Parameters

See the Inelastic File overview for general remarks about "tags" and the implementation of inelastic loss models in Kalypso.

The following parameters are entered for each atomic collision pair:

Z1*Z2: this number is the product of the atomic numbers of the pair of atoms which are involved in a binary collision (this value is used as a tag only)

K(OR): a coefficient (units: eV A2) which reflects the energy loss or electronic stopping effect of the binary collision. K depends on several other parameters, and can be computed most easily by clicking on the "Compute K(LSS), K(OR)" tab of the input dialog box. K(OR) is closely related to K(LSS) from the LSS model. But since the stopping cross-section in the OR model explicitly depends on the collision impact parameter (p), the latter model includes a multiplicative term, the "Oen-Robinson" factor, F(OR):

F(OR) = sqr(c/a)*exp(-cRmin/a),

where a is the unadjusted Lindhard screening length, Rmin is the apsidal distance (of closest approach) and c is the "**Exponential parameter**". (See the Simulation Primer for a detailed discussion.) In the usual OR model, c = 0.3, but the Simulation Kit allows any user-selected value.

Maximum apsidal distance: this parameter (maximum value 3 A) sets a maximum apsidal distance (distance of closest approach) above which OR energy losses are ignored, i.e. losses will only be calculated for 'collisions' which involve encounters closer than this value. The parameter should be chosen such that F(OR), evaluated when Rmin = Maximum apsis, makes a negligible contribution to the total stopping electronic stopping. The best way to set the parameter is by trial and error (observing the reported OR loss from trial simulations). For 5 keV Ar-Cu, a = 0.116 A, Rmin at zero impact parameter is 0.36 A, so with c = 0.3, the exponential term in F(OR) is 0.39. For Rmin = 1.0 A, this term decreases to 0.08, and for Rmin = 1.3, it has a value of 0.03. The latter is not negligible, because the elementary losses have to be weighted by cross-section factors which vary roughly as sqr(Rmin). Bearing in mind these considerations, the parameter should be chosen to be as small as possible, because large values may impact on the speed of the calculations.

ST Model

Shapiro-Tombrello Parameters

See the Inelastic File overview for general remarks about the implementation of inelastic loss models in Kalypso. The ST model is discussed in the Simulation Primer. The ST model describes the inelastic loss event as a transition involving the promotion of one or more electrons, which is initiated by a close binary encounter, in which the distance of closest approach is less than some critical value.

The following parameters are entered for each atomic collision pair:

Z1*Z2: this number is the product of the atomic numbers of the pair of atoms which is involved in a binary collision (this value is used as a tag only)

Rcrit: the critical distance (in A) for excitation of the level-crossing transition.

dE: this is the inelastic energy loss (in eV) resulting from the transition.

Prob: the probability of the transition.

Nmax: the maximum number of electrons which can be promoted in the transition.

Compute K(LSS), K(OR)

Overview

This gadget computes the "K" parameters which are needed as inputs to the LSS and OR inelastic loss models. The two models require different input parameters, with very different physical dimensions. In particular, the LSS model treats the target as an homogenous medium, so you may have to define the target atomic number, Z2, as some non-integral average value if your target is inhomogenous (e.g. an alloy).

The formulae used to calculate the LSS and OR parameters respectively are shown on the tab sheet inset, as are the formulae relating these parameters to the inelastic loss associated with an elementary collision. The formulae are explained in more detail in the Simulation Primer. The term *F*(OR) represents the "Oen-Robinson factor", which is a decaying exponential function of Rmin, the collision distance of closest approach.

The LSS assymetry between (moving) projectile and (stationary) target atom is also reflected in the original formulation of the OR model, where the Z1 and Z2 terms appear raised to different powers. The implementation used here allows you to optionally "permute" Z1 and Z2, and thereby remove the asymmetry (an average value of K is computed by swapping the roles of Z1 and Z2).

Select an energy loss model (LSS or OR), then click on the "Evaluate" buttons to compute a K(LSS) or K(OR) value for your inputs. This value is shown in the yellow edit box. It can be copied (Ctrl-C) and pasted (Ctrl-V) directly into one of the input cells on the relevant (LSS or OR) tab sheet.