PBCTools Plugin User's Guide

Jerome Henin Olaf Lenz Cameron Mura Jan Saam

Version 2.8

The plugin `pbctools` provides Tcl functions to handle periodic boundary conditions.

1 Basic usage

All of the plugin's functions can be accessed via the Tcl text command

```
pbc subcommand [options]...
```

that you can write in a VMD-Tcl-script or interactively enter in the VMD console window or the VMD TkConsole (accessible via VMD Main Menu → Extensions → Tk Console). When no *subcommand* is provided, a short help message will be printed. The list of available subcommands can be found in table 1.

2 Installation

Since VMD version 1.8.6, the PBCTools plugin is part of the official distribution of VMD[1], and all commands can be used within VMD without further preparation.

In the case that you are using an older version of VMD, or that you want to use a more recent version of PBCTools than what came with the VMD distribution, you can activate the PBCTools plugin as follows:

1. Fetch the PBCTools plugin from Github

   ```
   git clone git://github.com/olenz/pbctools.git
   ```

   This will create a new directory `pbctools` in the current directory.

2. Add the following lines to your VMD startup file (`~/.vmdrc` on Unix or `vmd.rc` on Windows)[2]:

   ```
   set dir pbctools-directory
   source $dir/pkgIndex.tcl
   package require pbctools
   ```

[1] `http://www.ks.uiuc.edu/Research/vmd/`

[2] For more details on the startup files, see chapter "Startup Files" in the VMD User's Guide.

Subcommand	Description	p.
`set` *cell* [*options...*]	Set the VMD unit cell properties (*e.g.* to use VMD's feature that allows to display periodic copies of the system).	3
`get` [*options...*]	Get the VMD unit cell properties.	4
`readxst` *xstfile* [*options...*]	Read the VMD unit cell properties from an XST file.	5
`writexst` *xstfile* [*options...*]	Write the VMD unit cell properties to an XST file.	5
`wrap` [*options...*]	When the atoms of the system are not all in one periodic image, but are distributed over various images, this function wraps all atoms into the chosen image. It is also possible to change between different representations of the unit cell (orthorhombic or triclinic).	6
`unwrap` [*options...*]	When overlong bonds (that stretch the whole system) occur and compounds (residues, segments, chains or fragments) are broken in the course of a simulation trajectory because atoms are wrapped around the periodic boundaries, this function will remove large jumps of atoms between consecutive frames.	8
`join` *compound* [*options...*]	When you have still broken compounds in frames after you have used `unwrap`, this function can be used to join broken compounds. Note, that this function is significantly slower than `unwrap`!	9
`box` [*options...*]	When you want to draw a box around the unit cell of your system, this function can be used. The box will automatically adapt to changes in the unit cell parameters in the course of a trajectory.	11
`box_draw` [*options...*]	When the unit cell parameters do not change in the course of a trajectory, this function draws a static box that will not adapt to changes in the unitcell properties.	12

Table 1: List of the subcommands of the PBCTools plugin.

3 `set` – Setting the unitcell parameters

To be able to work correctly, all other procedures of the PBCTools plugin require the VMD unitcell parameters to be set. Some file formats and their readers provide the necessary information (*e.g.* the DCD, VTF and Amber crdbox formats). When the format does not provide the information, the parameters can either be set with help of the command `pbc set`, or they can be read in from a file in XST format via the procedure `pbc readxst` (see section 5).

Syntax

`pbc set` *cell* [*options...*]

Description

Sets the VMD unit cell properties to *cell* in the specified frames. *cell* must either contain a single set of unit cell parameters that will be used in all frames of the molecule, or it must contain a parameter set for every frame.

Example

```
# set the unit cell side length to 10 in all frames
pbc set {10.0 10.0 10.0} -all
```

Options

`-molid` *molid*|`top`	Which molecule to use (default: `top`).
`-first` *frame*|`first`|`now`	The first frame to use (default: `now`).
`-last` *frame*|`last`|`now`	The last frame to use (default: `now`).
`-all`[`frames`]	Equivalent to `-first first -last last`.
`-now`	Equivalent to `-first now -last now`.
`-namd`|`-vmd`	Format of the unit cell parameters *cell*. When `-vmd` is used, a parameter set must be a list of the VMD unitcell parameters *a*, *b*, *c* (*i.e.* the side lengths of the unit cell) and optionally *alpha*, *beta* and *gamma* (the angles of the unit cell) for non-orthorhombic unitcells. When `-namd` is used, a parameter set must contain the three unit cell vectors A, B and C (the 3D-vectors of the unitcell sides) (default:`-vmd`).
`-`[`no`]`alignx`	If the option `-namd` is used and the unit cell vector A is not parallel to the x-axis, `-alignx` will rotate the system so that it is. If `-noalignx` is used, the function will return with a warning when A ist not aligned with the x-axis.

4 `get` – Getting the unitcell parameters

Syntax

`pbc get` [*options...*]

Description

Gets the VMD unit cell properties from the specified frames. Returns a list of one parameter set for each frame or an empty list when an error occured.

Example

```
# get the unit cell parameters of the current frame
set cell [pbc get -now]
```

Options

`-molid` *molid*|`top`	Which molecule to use (default: `top`)
`-first` *frame*|`first`|`now`	The first frame to use (default: `now`).
`-last` *frame*|`last`|`now`	The last frame to use (default: `now`).
`-all`[`frames`]	Equivalent to `-first first -last last`.
`-now`	Equivalent to `-first now -last now`.
`-namd`|`-vmd`	Format of the unit cell parameters. When `-vmd` is used, a parameter set will contains the VMD unitcell parameters *a*, *b*, *c*, *alpha*, *beta*, *gamma*. When `-namd` is used, a parameter set contains the three 3D unit cell vectors *A*, *B* and *C* (default: `-vmd`).
`-`[`no`]`check`	Check whether the unit cell parameters seem reasonable, *i.e.* whether the side lengths are not too small and the angles are not very small or very large (default: `-nocheck`).

5 `readxst` and `writexst` – Handling XST files

Syntax

`pbc readxst` *xstfile* [*options...*]
`pbc writexst` *xstfile* [*options...*]

Description

Read/write the unit cell information from an XST or XSC file.

Example

```
# read the unit cell parameters from system.xst
pbc readxst system.xst
pbc writexst system.xst
```

Options

`-molid` *molid*|`top`	Which molecule to use (default: `top`).
`-first` *frame*|`first`|`now`	The first frame to use (default: `first`).
`-last` *frame*|`last`|`now`	The last frame to use (default: `last`).
`-all`[`frames`]	Equivalent to `-first first -last last`.
`-now`	Equivalent to `-first now -last now`.
`-stride` *stride*	Read only every *stride*-th timestep from the file (default: 1).
`-`[`no`]`skipfirst`	*(only readxst)* Whether to skip the first line of the file, or not (default: `-skipfirst` for XST files, `-noskipfirst` for XSC files)
`-step2frame` *num*	Conversion factor between step *num* in XST file and frame *num* in DCDs. This is useful when loading multiple XSTs and want to avoid over-writing info of earlier frames by having a unique mapping between step and frame.
`-step0` *num*	*(only writexst)* Timestep number for the first written frame.
`-`[`no`]`alignx`	*(only readxst)* If the unit cell vector A is not parallel to the x-axis, `-alignx` will rotate the system so that it is. If `-noalignx` is used, the function will return with a warning when A ist not aligned with the x-axis.
`-log` *logfile*	*(only readxst)* Log file used for debugging information.

6 wrap – Wrapping atoms

Syntax
pbc wrap [*options...*]

Description
Wrap atoms into a single unitcell.

Example
```
# wrap the system into the orthorhombic box shifted by one box length in X-dir
pbc wrap -orthorhombic -shiftcenterrel 1 0 0
```

Options

-molid *molid*|top	Which molecule to use (default: top)
-first *frame*|first|now	The first frame to use (default: now).
-last *frame*|last|now	The last frame to use (default: now).
-all[frames]	Equivalent to -first first -last last.
-now	Equivalent to -first now -last now.
-parallelepiped |-orthorhombic	Wrap the atoms into the unitcell parallelepiped or the corresponding orthorhombic box with the same volume and center as the (non-orthrhombic) unitcell. The unitcell displacement vectors are not changed (default: -parallelepiped).
-sel *sel*	The selection of atoms to be wrapped (default: "all"). Use this if you don't want to wrap all atoms.
-nocompound |-compound res[id[ue]]|seg[id]|chain|fragment	Defines, which atom compounds should be kept together, *i.e.* which atoms will not be wrapped if a compound would be split by the wrapping: residues, segments or chains (default: -nocompound).
-nocompoundref |-compoundref *refsel*	When compounds have been defined via the -compound option, this defines a reference selection of atoms. After the wrapping, at least one of the atoms in this selection will be in the central image. This can be useful, for example, when water molecules should be wrapped such that the oxygen atom ends up in the central image (default: -nocompoundref).
-center origin|unitcell |com|centerofmass |bb|boundingbox	Specify the center of the wrapping cell. The center can be set to the origin (origin), to the center of the unit cell (unitcell), to the center of mass (com or centerofmass) of the selection specified by the option -centersel, or to the center of the bounding box (bb or boundingbox) of the selection specified by the option -centersel (default: unitcell).
-centersel *sel*	Specify the selection *sel* that defines the center of the wrapping cell in the option -center (default: "all"). Note that this option only has an effect if used together with the arguments com, centerofmass, bb or boundingbox to the option -center.

...

`-shiftcenter` *shift*	Shift the center of the box by *shift*. *shift* has to be a list of three numerical values. (default: `{0 0 0}`)
`-shiftcenterrel` *shift*	Shift the center of the box by *shift* (in units of the unit cell vectors). *shift* has to be a list of three numerical values. (default: `{0 0 0}`)
`-`[`no`]`verbose`	Turn on/off verbosity of the function (for debugging) (default: `-noverbose`).
`-`[`no`]`draw`	Draw some test vectors (for debugging) (default: `-nodraw`).

7 `unwrap` – Unwrapping atoms

Syntax

`pbc unwrap` [*options...*]

Description

If a simulation only saves the central image coordinates of a system, atoms are wrapped around when they reach the boundaries. This leads to big jumps in the coordinates of the atoms, and to bonds that stretch the whole box length. This procedure will reverse these jumps and make the movement of the atoms continuous over a series of frames. This process is not necessarily unique, so this procedure can *not* exactly reverse the effects of the command `pbc wrap`.

In the case of a simulation trajectory, the following process most probably gives the best result:

1. Go to the first frame.
2. Shape the unitcell of the frame for the best visualization by using the commands `pbc join -now` and `pbc wrap -now` with appropriate options.
3. Unwrap the trajectory, starting from the current frame, by using `pbc unwrap -first now`.
4. Visually check the result. If the system gets smeared out too fast because the diffusion is too high, repeat the procedure with successively later frames.

Example

```
# unwrap all protein atoms
pbc unwrap -sel "protein"
```

Options

`-molid` *molid*|`top`	Which molecule to use (default: `top`)
`-first` *frame*|`first`|`now`	The first frame to use (default: `now`).
`-last` *frame*|`last`|`now`	The last frame to use (default: `now`).
`-all`[`frames`]	Equivalent to `-first first -last last`.
`-now`	Equivalent to `-first now -last now`.
`-sel` *sel*	The selection of atoms to be unwrapped (default: `"all"`). Use this if you don't want to unwrap all atoms.
`-`[`no`]`verbose`	Turn on/off verbosity of the function (for debugging) (default: `-noverbose`).

8 `join` – Joining residues, chains, segments, fragments, and connected/bonded groups

Syntax

`pbc join` *compound* [*options*...]

Description

Joins compounds of type *compound* of atoms that have been split due to wrapping around the unit cell boundaries, so that they are not split anymore. *compound* must be one of the values `res[id[ue]]`, `chain`, `seg[id]`, `fragment` or `connected`.

This procedure can help to remove bonds that stretch the whole box. Note, however, that `join` is relatively slow and is required only in few cases. If you have a simulation trajectory that contains frames with overstretched bonds, it is usually enough to apply `join` only to the first frame and then the much faster procedure `unwrap` to all of the frames:

```
pbc join compound -first 0 -last 0
pbc unwrap
```

Note also, that the (faster) default algorithm only works when none of the compounds stretches more than half the periodic box in any direction. With the option `-bondlist` you can select an alternate algorithm that joins compounds based on direct bonds and does not suffer from this limitation, but can be significantly slower.

Examples

```
# join all residues such that the Carbon alpha atom
# is in the central image
pbc join res -ref "name CA"
# join all bonds of long polymer chain molecules
pbc join fragment -bondlist
```

Options

`-molid` *molid*|`top`	Which molecule to use (default: `top`)
`-first` *frame*|`first`|`now`	The first frame to use (default: `now`).
`-last` *frame*|`last`|`now`	The last frame to use (default: `now`).
`-all`[`frames`]	Equivalent to `-first first -last last`.
`-now`	Equivalent to `-first now -last now`.
`-sel` *sel*	The selection of atoms to be joined (default: `"all"`). Use this if you don't want to join all atoms.
`-noborder`|`-border` *depth*	When only atoms close to the boundaries of the unit cell have overstretched bonds, this option can be used to specify the maximal depth inside the system. Using this option will significantly speed up join (default: `-noborder`).

...

`-noref`|`-ref` *refsel*	This defines a reference selection of atoms. When joining compounds, the first atom matching the selection in each compound will be chosen, and all atoms will be wrapped into a unit cell around this atom. If `noref` is used, the first atom in the compound is the reference atom (default: `-noref`).
`-`[`no`]`bondlist`	Turn on/off alternate, bond topology based joining algorithm (default: `-nobondlist`).
`-`[`no`]`verbose`	Turn on/off verbosity of the function (for debugging) (default: `-noverbose`).

9 box and box_draw – Drawing the unit cell boundaries

9.1 box – Automatically updateing box

Syntax

pbc box [*options...*]

Description

(Re)Draws a box that shows the boundaries of the unit cell. The box will automatically adapt to changes in the unit cell parameters in the course of a trajectory, as for example for simulations at constant pressure. Only a single automatically updated box can exist at a time.

Example

```
# draw a box, centered on the origin
pbc box -center origin
```

Options

-molid *molid*|top	Which molecule to use (default: top)
-on|-off|-toggle	Turn the box on, off, or toggle whether it is on or off. (default: -on)
-parallelepiped |-orthorhombic	Draw the box as a parallelpiped, or as the corresponding orthorhombic box. (default: -parallelepiped).
-color *color*	Draw the box in color *color*. (default: blue)
-material *Material*	Draw the box using the material *Material*. (default: Opaque)
-style lines|dashed|arrows|tubes	Choose the style of the box (default: lines).
-width *width*	Define the *width* of the lines/arrows/tubes (default: 3).
-resolution *res*	Use *resolution* faces for the tube style (default: 8).
-center origin|unitcell |com|centerofmass |bb|boundingbox	Specify the center of the box. The center can be set to the origin (origin), to the center of the unit cell (unitcell), to the center of mass (com or centerofmass) of the selection specified by the option -centersel, or to the center of the bounding box (bb or boundingbox) of the selection specified by the option -centersel (default: unitcell).
-centersel *sel*	Specify the selection *sel* that defines the center of the wrapping cell in the option -center (default: "all").
-shiftcenter *shift*	Shift the center of the box by *shift*. *shift* has to be a list of three numerical values. (default: {0 0 0})
-shiftcenterrel *shift*	Shift the center of the box by *shift* (in units of the unit cell vectors). *shift* has to be a list of three umerical values. (default: {0 0 0})

9.2 box_draw – Drawing a static box

Syntax
pbc box_draw [*options...*]

Description
Draws a static box that shows the boundaries of the unit cell, but will not adapt to changes in the unitcell properties. This might be useful when you want to draw more than one box at a time (*e.g.* to show periodic images of a box), or to show the initial box in a simulation with fluctuating box unit cell geometry.

Options
pbc box_draw uses the same options as the command pbc box, with the exception of the options -on|-off|-toggle and -color, which can not be used. To set the color of the box, use the graphics color command.

Example
```
# draw a box around the central image
set box0 [pbc box_draw -shiftcenterrel  0 0 0 ]
# draw a box around the central image shifted by
# the unit cell vector C
set box1 [pbc box_draw -shiftcenterrel  0 0 1 ]
```

10 Credits

The PBCTools plugin has been written by (in alphabetical order)

- Toni Giorgino <toni.giorgino _at_ isib.cnr.it>
- Jerome Henin <jhenin _at_ cmm.upenn.edu>
- Olaf Lenz <olenz _at_ icp.uni-stuttgart.de> (maintainer)
- Cameron Mura <cmura _at_ mccammon.ucsd.edu>
- Jan Saam <saam _at_ charite.de>

The pbcbox procedure copies a lot of the ideas of Axel Kohlmeyer's script vmd_draw_unitcell. Please submit your bug reports, comments and feature requests on the PBCTools homepage[3].

[3] http://github.com/olenz/pbctools/