

The SimCellin Manual

For version 2.40.00

By Yves Konigshofer (yvesk@yvessoft.com)

January 5, 2006

Introduction

SimCellin is a JAVA-based three dimensional simulation of cell-to-cell contact. The area where two cells interact is referred to as the **contact area** and there is only one contact area. The two cells themselves are represented as one or more **spheres**. Each sphere can contain one or more different types of **molecules** that diffuse over its surface and that interact with other types molecules when they are found inside of the contact area. Finally, the contact area is itself broken down into smaller **regions** and the amount of molecules found in each region of the contact area determines how the molecules interact with other molecules in the same region.

Each simulation is initially defined using a configuration file that specifies many different parameters for the spheres, the contact area, and the molecules. Each simulation should have a different configuration file and this file can be edited using a text editor. Each line in this file is either a command or a comment and each command must be found on one row.

The location of the spheres is specified using x/y/z coordinates. It is sometimes useful to use only a single sphere to represent two interacting cells. In addition, multiple spheres can be used to represent the same cell. This is useful when there are many different types of molecules being simulated for the same cell since it can be difficult to represent more than three different molecules on a sphere if one wants to be able to separate them based on their red, green, and blue colors.

The radii of the spheres should approximate the radii of the cells being simulated and should be the same for all cells if using a curved contact area. It is possible to use different radii for different spheres; however, if using a curved contact area, the surface areas of opposing regions will differ and will affect the diffusion and binding properties of the molecules.

The location of the contact area on each sphere is specified using an x/y/z vector from the middle of the sphere towards the middle of the contact area.

When using a curved contact area, the angle spanned by the contact area is given as the angle between the middle of the contact area, the center of the sphere, and the edge of the contact area. Similarly to the radii, this angle should also be the same for all spheres.

When using a flat contact area, the location of the contact area on each sphere is determined based on the radius of the contact area. The radius of the sphere needs to be larger or equal to the radius of the contact area.

One can simulate many different types of molecules. Each type of molecule has its own properties and each type of molecule can interact with each other type of molecule. Each type of molecule is found on one sphere.

Once molecules are added to the spheres, the simulation can diffuse the molecules across the surfaces of the spheres and the molecules inside of the contact area can interact with those molecules that are found in the same region.

Diffusion itself is simulated for short periods of time. During this time, molecules should not be able to diffuse through more than one region and, ideally, should diffuse much slower than that. For example, 60 seconds of diffusion could be simulated using 3000 iterations of 20 milliseconds of diffusion. It is possible to bias the direction that molecules are diffusing; however, if this is used, then the outcome of the simulation will depend on the time being simulated in the iterations.

Compiling SimCellin

The following Windows script should compile SimCellin using Sun Microsystems' JDK:

```
CD [SimCellin directory]

del *.class

javac.exe -source 1.4 -target 1.4 -verbose -Xstdout c:\last_compile.txt
-g:none *.java

jar.exe cvmf SimCellinManifest.txt simcellin.jar *.class images license
>>c:\last_compile.txt

notepad c:\last_compile.txt
```

Starting SimCellin

The simulation is started using a JAVA virtual machine. It is best to allocate a large amount of RAM at the start of the simulation. The following line should start SimCellin on systems that have Sun Microsystems' JRE installed:

```
java -Xmx640m -Xms640m -jar simcellin.jar <parameters> <filename>
```

The <parameters> can be as follows:

-t <number>

- if used, this must be the first parameter
- <number> specifies the number of threads to use; the default is 1
- extra threads offer no benefit on a single cpu single core system

-a

- opens <filename> and starts the simulation without user intervention

-s

- similar to -a except that no windows are opened
- is required if the operating system cannot open windows

-h

- displays the version

-gpl

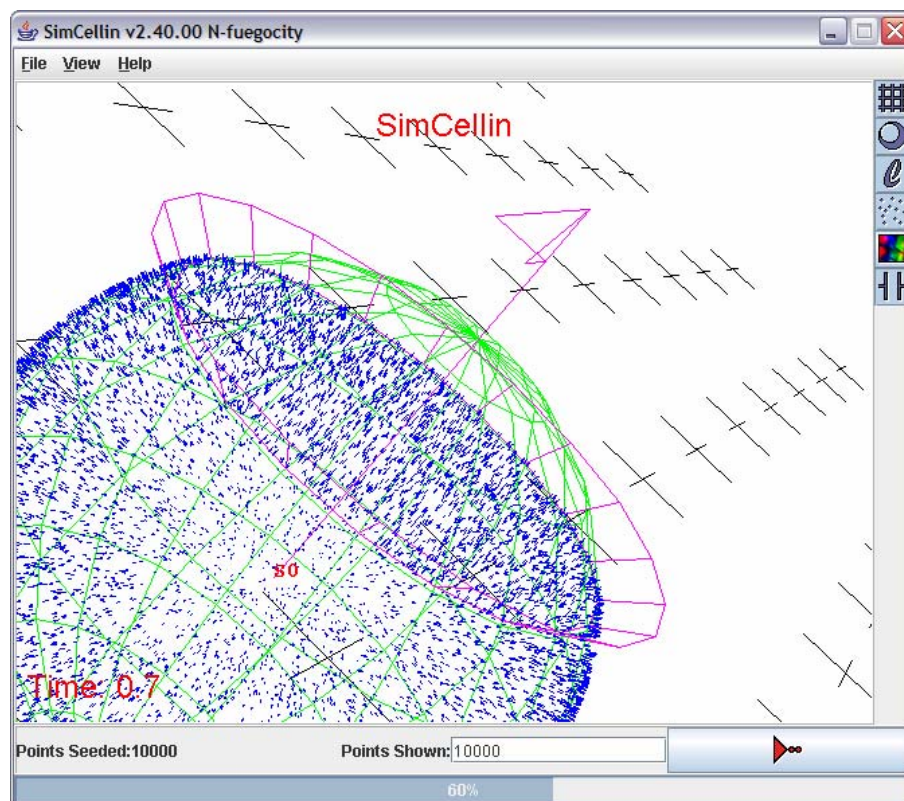
- displays the GNU General Public License

The <filename> is the name of a configuration file and is needed if the -a or -s options are used.

An example is as follows:

```
Java -Xmx640m -Xms640m -jar simcellin.jar -t 2 -s c:\myconfigfile.txt
```

The Main Window



The Main Window allows the user to interact with the simulation.

The **world view** takes up most of the available space and is a three dimensional representation of the simulation. The world view contains a centered title text that is found towards the top and it is possible to specify the text that is displayed. The bottom left corner of the world view contains time that currently being simulated. This time starts at 0 seconds and can be reset to that value. Molecules and spheres are represented inside of the world view. Using the grid and the spheres, it is possible to move around the simulated space and render the results of the simulation from different angles.

The amount of molecules shown in the wireframe representation of the world view can be specified using the **“Points Shown” text box**. The larger the number of molecules that are shown, the longer it takes to redraw the world view. It is possible to show as many molecules as have been seeded (indicated by **“Points Seeded”**). However, showing too many molecules also tends to clutter the image. Different types of molecules cannot be distinguished using the wireframe representation. To distinguish different types of molecules, it is possible to render the image using specified color intensities for each type of molecule. The number of molecules being shown in the text box does not affect what is shown when rendered images are generated.

To the right of the **“Points Shown” text field** is the **GO button**. The appearance of this button indicates the current state of the program. In some cases, pressing the button starts or continues the simulation.

At the bottom of the main window is the progress bar. This bar is used to indicate how many rounds of diffusion have taken place and how many rounds remain until the next step specified in a configuration file.

At the right side of the main window are various **buttons** that control what is shown inside of the world view. In addition, there are buttons to render the current wireframe view in color and to pause and resume the simulation.

At the top of the main window is a menu bar, which contains options for loading configuration files, saving images, opening up new windows, etc.

Moving in the World View

The following keys can be used to move the eye around in the world view. In order to activate the keys, you have to first click on the world view.

Cursor-UP	Moves the eye up
Cursor-DOWN	Moves the eye down
Cursor-LEFT	Moves the eye to the left
Cursor-RIGHT	Moves the eye to the right
A	Moves the eye forwards
B	Moves the eye backwards
Shift-Cursor-UP	Rotates the eye up/backwards
Shift-Cursor-DOWN	Rotates the eye down/forwards
Shift-Cursor-LEFT	Rotates the eye left
Shift-Cursor-RIGHT	Rotates the eye right
Shift-A	Rotates the eye clockwise
Shift-B	Rotates the eye counterclockwise

Objects in the World View

The world view contains a grid composed of “+”-like markers that are useful in orienting the eye when moving around the world. In addition, the world view contains representations of the spheres being used in the simulation and the individual molecules that are moving around on the spheres.

When viewing objects in the world view, the appearance of the objects is dependent on an angle between the middle of the world view, the eye of the observer, and the top of the world view. This angle defaults to 10 degrees but can take a value between 0 and 90 degrees. Increasing this angle leads to a more fish-eyed, wide-angle view of the world. The ideal angle can be calculated using the distance of the observer to the screen where the world view is being displayed.

The spheres are represented by green wireframe meshes. The sizes of the spheres depend on their radii where one micrometer is represented as one unit in the world. If labels are turned on, each sphere will be designated S0, S1, S2, etc.

Each sphere's sphere vector is represented by a magenta line from the middle of the sphere that extends 1.5 radii away from the center. This line points through the middle of the contact area and the end of this line is represented by two triangles that represent the directions of two normals to the sphere vector. These normals are used to determine which region a molecule is located in.

Each sphere's contact area is represented by a magenta ring that circles the sphere and extends away from it. The sphere vector should point away from the middle of this ring.

Molecules are represented by triangles that rise away from the molecules' current locations on the surfaces of the spheres. The heights of the rectangles represent either the heights of the molecules, if free, or the heights of the complexes, if bound. The widths of the triangles represent the prior and current locations of the molecules or dimers. The colors of the triangles represent whether the molecules are free (blue), bound (yellow).

The Buttons

The right border of the main window contains the buttons that control what is displayed in the world view and allow the user to control the flow of the simulation.



This toggles the display of a grid from -100,-100,0 to 100,100,0.



This toggles the display of wireframe spheres. The spheres themselves are green. The sphere vectors are represented by a magenta lines from the center of the sphere that end in two triangles that represent two normals to the sphere vectors that are used for determining the regions that molecules are found in. Parts of the spheres may also be surrounded by magenta rings that represent the borders of the contact areas.



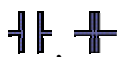
This toggles the display of sphere labels (S0, S1, etc.).



This toggles the display of molecules. The number of molecules that are displayed is specified using the “Points Shown” box. The molecules appear as rectangles on the surfaces of the spheres whose height is determined by the height of the molecule or dimer being represented and whose width is determined by the current and last location. The color of the molecules depends on whether they are free (blue), bound (yellow), or bound having bounced off of the edge of the contact area (red, orange).



This renders all of the molecules. The displayed image will revert back to the wireframe view when the current iteration completes. Therefore, the simulation should be paused before pressing this button. In addition, the selection of menus or the appearance of tool-tips will also cause the displayed image to revert back to the wireframe view. If each molecule is very dim (i.e. has a very low red/green/blue value), one may not be able to see too much in the rendered image.



This indicates whether or not the simulation is paused (left) or running (right). This button is used to pause and resume the simulation as the configuration file is executed and as diffusion takes place. However, if the simulation is currently diffusing molecules, the simulation will only pause once the current iteration is complete.



This is the GO button and indicates whether the simulation is stopped (left), waiting for the user to press the GO button (middle), or currently running (right). The left image is displayed when no configuration file is loaded. The middle image is displayed when a configuration file has been loaded and when the user is asked to press the GO button. The right image is displayed as the configuration file is executed and when molecules are diffusing.

The Menu Options

File menu

Open simcellin.cfg

This opens the default configuration file from the main directory if this file is present. This option is not available if a configuration file has already been loaded. To open a new configuration file, one must first reset the simulation.

Open...

This is used to open a configuration file. This option is not available if a configuration file has already been loaded. To load a new configuration file, one must first reset the simulation.

Reset Simulation

This is used to reset the simulation. If the simulation is busy calculating diffusion, the simulation will be reset at the conclusion of the current iteration. Sometimes, this can take a while depending on the amount of molecules that are being simulated. When the simulation resets, the eye remains in its previous position but it is possible to open a new configuration file.

Save Rendered Image...

This saves a 640x480 RAW image of the current view where each molecule is represented by its color. Each pixel is represented by three bytes with values ranging from 0 to 255 that represent the red, green, and blue values for that pixel. Pixels are stored left to right starting with the top row. If more than one molecule occupies a pixel, the value is the sum of the molecules occupying that pixel.

Print View

This prints the current on-screen wireframe view. This does not print a rendered image – this should be done from a paint program after adjusting the brightness and contrast of the image.

Exit

This exits the program.

View menu

Open Trace View

This opens the Trace View window. This menu is only available if molecules have been seeded.

Adjust Sliders

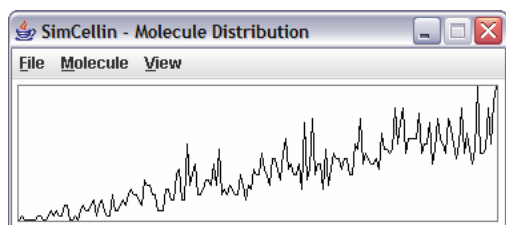
This opens the Adjust Sliders window. Using these sliders, it is possible to specify the angle that the eye rotates and the distance that the eye moves when navigating using the keyboard. It is also possible to specify the angle between the middle of the screen, the eye, and the top of the screen – this affects the appearance of the rendered image.

Help menu

About

This displays information about the current version of the program.

The Trace View



The trace view can be activated once molecules have been seeded onto spheres. It represents the amount of molecules that are located a particular distance from the center of the contact area. When a curved contact area is used, this distance is calculated as an angle of 0 to 180 degrees from the sphere vector and is represented left, 0 degrees, to right, 180 degrees, in 1 degree increments. When a flat contact area is used, this distance is evenly spaced from the middle to the edge of the contact area. Molecules outside the contact area are not represented due to potential differences in cell size.

The **File** menu allows one to print what is currently displayed in the trace view.

The **Molecule** menu allows one to turn on and off the display of the distributions for particular molecules. The names of the molecules that are shown can be specified in a configuration file.

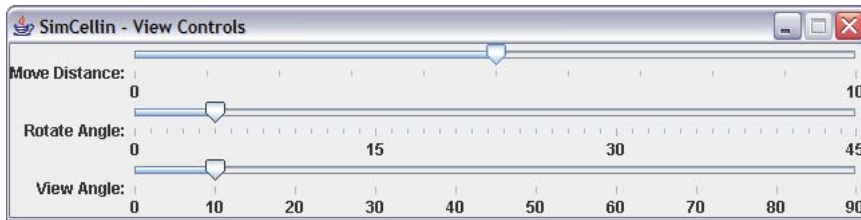
The **View** menu allows one to change the way the traces are drawn. The traces are shown in a way that the maximal and minimal values for each molecule are found on the top and bottom of the window, respectively. There are three different ways of drawing the traces: normalized, standard, and logarithmic.

The **normalized** view assigns values based on the number of molecules at a particular angle in relation to the area of the sphere that they are found in. If all molecules are randomly distributed, this value should be about the same for all angles. When a curved contact area is used, the high and low angles will show greater deviation from the average value since the surface areas of the spheres at these angles are very small and thus the presence or absence of a few molecules can lead to large changes in the calculated value. When a flat contact area is used, the values on the left will show greater deviation than the values on the right since the surface area is small close to the middle of the contact area compared to the edge of the contact area.

The **standard** view assigns values based on the number of molecules at a particular angle or distance. If all molecules are randomly distributed over a sphere, this should lead to a hill-shaped curve. If all molecules are randomly distributed over a flat contact area, this should lead to a straight line that slopes upwards to the right.

The **logarithmic** view assigns values based on the logarithm of molecules at a particular angle and is useful for looking at angles where there are only very few molecules present. If all molecules are randomly distributed over a sphere, this should lead to a very wide hill-shaped curve.

The Sliders



The sliders allow one to adjust the distance that the eye moves (Move Distance) and the angle that the eye rotates (Rotate Angle) when moving around the world view using the keyboard. In addition, the view angle can also be adjusted. When adjusting the move distance, the slider will automatically rescale between 0 and 10 and 0 and 100 units depending on the currently selected value.

The Configuration File

Simulations are run by first putting together a configuration file. The default file is “simcellin.cfg” and, if found in the main directory of the program, can be loaded by selecting the **File->Open simcellin.cfg** option from the menu. Alternatively, the **File->Open...** option can be used to open up any other configuration file.

Each configuration file should start by first defining the amount of regions in the contact area. Next, the amount of spheres should be defined followed by the amount of molecules.

When the configuration file is loaded, it is parsed from the beginning to the end and any syntax errors will be pointed out. Each command is found on a separate line and each command must appear on a single line. If a command is not recognized, an error will be reported and one can either ignore that line or abort the loading of the configuration file.

Comments can be added to the configuration file by starting a line with a semicolon.

Once the configuration file is loaded, you can start the simulation by pressing the GO button. The configuration is then executed in a linear fashion from the first to the last command. You can load another configuration file by first selecting the **File->Reset Simulation** option from the menu. This option can also be used during the simulation to abort the simulation.

There are many commands that can be used in the configuration file. These are listed below and are separated into four categories: general, molecule, region, and sphere-related.

General Commands

ASK

This temporarily stops the configuration file from being parsed until the user hits the GO button.

ASKGO

This starts the diffusion process for the specified number of iterations using the specified time interval for the iterations after the user hit the GO button.

CONTACTRADIUS <double>

This sets the radius of the contact area to <double> micrometers. This command is required if a flat contact area is used. The radius must be smaller than the radius of each sphere. The location of the contact area is determined for each sphere using the SPHERETALL, SPHERESHORT, and SPHEREVECTOR commands.

GO

This starts the diffusion process for the specified number of iterations using the specified time interval for the iterations.

ITERATIONS

This specifies the number of iterations that is performed with the ASKGO and GO commands.

LOOKAT <double_x> <double_y> <double_z>

This points the eye to the specified x/y/z location in the simulation. Each micrometer is simulated as one unit. When looking at a target, eye is initially righted so that it looks parallel to the x/y plane and rotated towards the target. The eye is then rotated up or down at the target.

LOOKEAST

This points the eye parallel to the x/y plane towards increasing x-axis values.

LOOKNORTH

This points the eye parallel to the x/y plane towards increasing y-axis values.

LOOKSOUTH

This points the eye parallel to the x/y plane towards decreasing y-axis values.

LOOKWEST

This points the eye parallel to the x/y plane towards decreasing x-axis values.

MAXPERREGION <integer>

This specifies the maximum amount of molecules that may be found in a region once molecules are allowed to move. A separate count is kept for each of two cells. This prevents excessive amounts of molecules from accumulating in any one region – especially, if there is directed movement towards the middle of the contact area. Note that USEBRICKGRID doubles the area of each region and that the areas of regions of a curved contact area are different in each ring.

MOLECULES <integer>

This specifies the number of different types of molecules that are being simulated.

MOLECULESEED <integer_1> <integer_2>

This adds <integer_2> molecules of type <integer_1> to the simulation. Molecules are stored as a large linked list of molecules and new molecules are inserted evenly into an existing list. It is important to have specified the sphere each type of molecule is found on before using this command.

MOVEBACKWARD <double>

This moves the eye backwards by <double> micrometers.

MOVEDOWN <double>

This moves the eye down by <double> micrometers.

MOVEFORWARD <double>

This moves the eye forwards by <double> micrometers.

MOVELEFT <double>

This moves the eye to the left by <double> micrometers.

MOVERIGHT <double>

This moves the eye to the right by <double> micrometers.

MOVEUP <double>

This moves the eye up by <double> micrometers.

MOVETO <double_x> <double_y> <double_z>

This moves the eye to the specified x/y/z location in the simulation. Each micrometer is simulated as one unit.

OUTPUTFILE <filename>

If specified, this redirects all output regarding molecule distributions and region heights to the given filename. This file can be quite large. If <filename> is not specified, no file is generated and the output is suppressed. If this command is not specified, output is sent to the window that started the program.

REGIONS <integer_1> <integer_2>

This divides the contact area into smaller regions and should be one of the first commands in the configuration file. When using a curved contact area, <integer_1> specifies the number of rings that represent the contact area while <integer_2> specifies the number of sectors that subdivide each ring.

When using a flat contact area, <integer_1> represents the number of regions in each row and column. <integer_2> must not be specified when using a flat contact area. When USESHIFTEDGRID is set, each second column contains an additional region. When USEBRICKGRID is set, regions are twice as long as they are wide in each column.

RESETTIME

This resets the time of the simulation to zero.

ROTATEBACKWARD <double>

This rotates the eye backwards (looking up) by <double> radians.

ROTATECLOCKWISE <double>

This rotates the eye clockwise by <double> radians.

ROTATECOUNTERCLOCKWISE <double>

This rotates the eye counterclockwise by <double> radians.

ROTATEFORWARD <double>

This rotates the eye forwards (looking down) by <double> radians.

ROTATELEFT <double>

This rotates the eye left by <double> radians.

ROTATERIGHT <double>

This rotates the eye right by <double> radians.

SPHERES <integer>

This specifies the number of spheres used by the simulation. You can have one or more spheres, but there is only one contact area.

TIMEINTERVAL <double>

This specifies the length of time, in <double> seconds, that is simulated during the iterations.

TITLETEXT <text>

This displays the specified text at the top of the simulation window.

USEBRICKGRID

When a flat contact area is used, this instructs the simulation to combine two adjacent square regions in each column. This results in rectangular regions that are twice as long as they are wide. In order to have a uniformly long border with each adjacent region, this command should be combined with the USESHIFTEDGRID command.

USECURVEDCONTACT

This instructs the simulation to use a round contact area that is divided into rings and sectors of regions and is the default. This command needs to appear before the REGIONS command.

USEDEMOSHADING

This instructs the simulation to color molecules based on the region they are in.

USEFLATCONTACT

This instructs the simulation to use a flat contact area that is divided into rows and columns of regions. This command needs to appear before the REGIONS command.

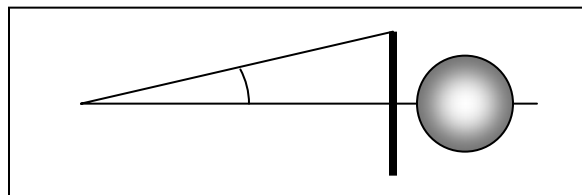
USESHIFTEDGRID

When a flat contact area is used, this instructs the simulation to shift every second column of regions up by a half region. Thus, instead of touching four different regions with each side and another four regions with each corner, regions would now touch one region up, one region down, two regions left, and two regions right. In order to have a uniformly long border with each adjacent region, this command should be combined with the USEBRICKGRID command in order to combine adjacent regions in each column.

When a curved contact area is used, this instructs the simulation to rotate every second ring by a half sector. Thus, instead of touching four different regions with each side and another four regions with each corner, regions would now touch two regions in the same ring and two regions in each adjacent ring.

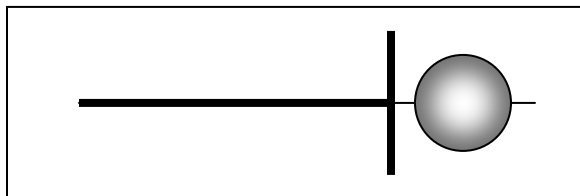
VIEWANGLE <double>

This specifies the angle, in <double> radians, between the middle of the world view window, the eye of the observer, and the top of the world view window. This defaults to 0.1745 radians (10 degrees) and can range between 0 and 1.5708 radians (0 and 90 degrees). Larger values give a more fish-eyed, wide-angle view of the world.



VIEWDISTANCE <double_1> <double_2>

This is similar to VIEWANGLE but <double_1> specifies the distance of the eye of the observer from the screen and <double_2> specifies the height of the screen. The units are in micrometers. Thus, it is possible to look at scenes e.g. as if one is 20 micrometers away from a 10 micrometer screen and 25 micrometers away from a sphere.



Molecule Commands

MOLECULEACTIVECOLOR <integer_1> <integer_r> <integer_g> <integer_b>

This specifies the color contribution of a molecule of type <integer_1> and is used for rendering the view when saving a file of activated molecules.

MOLECULEACTIVEDEGRADE <integer> <double>

This specifies the first order rate constant in <double> per second for a molecule of type <integer> to become degraded if it is currently activated and internal.

MOLECULEADDPERSECOND <integer> <double>

This specifies the <double> amount of molecules of type <integer> that move from being degraded to being internal per second. This can be used to represent molecule synthesis. As long as no seeded molecules of type <integer> are degraded, no molecules are synthesized and, thus, the number of seeded molecules represents the maximum number of molecules of type <integer> that are present at any given time.

MOLECULEBINDDISTANCE <integer> <double_1> <double_2> <double...>

This specifies the heights, in <double> meters, of molecules of type <integer> when they are present as complexes with other types of molecules. This value is used to calculate the distances between the opposing membranes in the various regions. The amount of defined molecule types determines the amount of <double> values that need to be specified. If certain complexes cannot form, values of 0.0 should be entered.

MOLECULEBINDSTDEV <integer> <double_1> <double_2> <double...>

This specifies the standard deviation, in <double> meters, of the normal curve used to adjust the value of the on-rate when a molecule of type <integer> attempts to bind one of the sequentially listed molecule types. The separation of the opposing membranes in the applicable region is subtracted from the height of the complex (specified by MOLECULEBINDDISTANCE), and the height of the normal curve at this value is divided by the maximum height of the normal curve to obtain the value with which the on-rate is multiplied. The amount of defined molecule types determines the amount of <double> values that need to be specified. If certain dimers cannot form, values of 0.0 should be entered.

MOLECULEBOUNDACTIVEINTERNALIZE <integer> <double_1> <double...>

This specifies the first order rate constant in <double> per second for a molecule of type <integer> to become internalized if it is currently activated and part of a complex. The amount of defined molecule types determines the amount of <double> values that need to be specified.

MOLECULEBOUNDCOLOR <integer_1> <integer_r> <integer_g> <integer_b>

This specifies the color contribution of a molecule of type <integer_1> and is used for rendering the view when saving a file of bound molecules.

MOLECULEBOUNDDIFFUSION <integer> <double_1> <double_2> <double...>

This specifies the diffusion coefficients, in <double> m^2s^{-1} , of molecules of type <integer> when they are bound as complexes to the other types of molecules. The amount of defined molecule types determines the amount of <double> values that need to be specified. If certain complexes cannot form, values of 0.0 can be entered.

MOLECULEBOUNDENTRY <integer> <double_1> <double_2> <double...>

This specifies the standard deviation, in <double> meters, of the normal curve used to determine if a bound complex, composed of a molecule of type <integer> and one of the sequentially listed molecule types, can enter a region where the separation of the opposing membranes differs from the height of the complex. The height of the region is subtracted from the height of the complex and height of the normal curve at this value is divided by the maximum height of the normal curve to obtain the chance that the complex can enter the region. The amount of defined molecule types determines the amount of <double> values that need to be specified. If certain complexes cannot form, values of 0.0 can be entered.

MOLECULEBOUNDMOVEBIAS <integer> <double_1> <double_2> <double...>

This specifies the velocity in <double> $\text{m}\cdot\text{s}^{-1}$ that a complex involving a molecule of type <integer> moves towards the targets chosen using SPHEREVECTOR, SPHEREVECTORSHIFT, and SPHEREVECTORSTDEV. The amount of defined molecule types determines the amount of <double> values that need to be specified.

MOLECULEBOUNDRESTINGINTERNALIZE <integer> <double_1> <double...>

This specifies the first order rate constant in <double> per second for a molecule of type <integer> to become internalized if it is currently resting and part of a complex. The amount of defined molecule types determines the amount of <double> values that need to be specified.

MOLECULECELL <integer_1> <integer_2>

This specifies the <integer_2> cell that the molecule of type <integer_1> is found on. This is used in conjunction with the number specified with MAXPERREGION. The number <integer_2> can be either 0 or 1.

MOLECULECOLOR <integer_1> <integer_r> <integer_g> <integer_b>

This specifies the color contribution of a molecule of type <integer_1> and is used for rendering the view of all molecules and when saving a file. If more than one molecule occupies a pixel, the colors add up at that pixel. Using low r/g/b values leads to a larger range of intensities since 255 is the maximum value for a given pixel. However, using low r/g/b values also frequently leads to dim images that require contrast and brightness adjustments.

MOLECULEENTRYHEIGHT <integer> <double>

This specifies the height, in <double> meters, of the molecule <integer>. If the separation of the membranes is smaller than this value in a particular region, then the MOLECULEENTRYSTDEV comes into play.

MOLECULEENTRYSTDEV <integer> <double>

This specifies the standard deviation, in <double> meters, of the normal curve used to determine if an unbound molecule of type <integer> can enter a region where the membranes are spaced closer together than the height of the molecule. The height of the region is subtracted from the height of the molecule and height of the normal curve at this value is divided by the maximum height of the normal curve to obtain the chance that the molecule can enter the region.

MOLECULEEXTERNALACTIVATE <integer> <double_1> <double_2> <double...>

This specifies the amount of time in <double> seconds that a molecule of type <integer> must be bound to another molecule before it becomes activated. The amount of defined molecule types determines the amount of <double> values that need to be specified.

MOLECULEEXTERNALDEACTIVATE <integer> <double>

This specifies the first order rate constant in <double> per second for an external molecule of type <integer> to deactivate once it is no longer part of a complex.

MOLECULEEXTERNALIZE <integer> <double>

This specifies the first order rate constant in <double> per second for a molecule of type <integer> to become externalized if it is currently internal.

MOLECULEFOUNDON <integer_1> <integer_2>

This specifies the sphere <integer_2> that molecules of type <integer_1> are found on. Once molecules are seeded, this cannot be changed.

MOLECULEFRACTIONINTERNAL <integer> <double>

This specifies the <double> fraction (between 0.0 and 1.0; 1.0 = 100%) of molecules of type <integer> that are initially found inside the sphere as molecules are seeded. Internal molecules are not rendered.

MOLECULEFRACTIONDEGRADED <integer> <double>

This specifies the <double> fraction (between 0.0 and 1.0; 1.0 = 100%) of molecules of type <integer> that are initially represented as being degraded as molecules are seeded. Degraded molecules are not rendered and must be synthesized before they can be externalized.

MOLECULEFREEACTIVEINTERNALIZE <integer> <double>

This specifies the first order rate constant in <double> per second for a molecule of type <integer> to become internalized if it is currently activated and not part of a complex.

MOLECULEFREEDIFFUSION <integer> <double>

This specifies the diffusion coefficient, in <double> m^2s^{-1} , of molecules of type <integer> when they are not bound.

MOLECULEFREEMOVEBIAS <integer> <double>

This specifies the velocity in <double> m·s⁻¹ of a molecule of type <integer> as it moves towards the targets chosen using SPHEREVECTOR, SPHEREVECTORSHIFT, and SPHEREVECTORSTDEV.

MOLECULEFREERESTINGINTERNALIZE <integer> <double>

This specifies the first order rate constant in <double> per second for a molecule of type <integer> to become internalized if it is currently resting and not part of a complex.

MOLECULEINTERNALDEACTIVATE <integer> <double>

This specifies the first order rate constant in <double> per second for an internal molecule of type <integer> to deactivate.

MOLECULEKOFF <integer> <double_1> <double_2> <double_...>

This specifies the values of the off-rate, in <double> s⁻¹, of molecules of type <integer> when they try to dissociate from complexes with the other types of molecules. The amount of defined molecule types determines the amount of <double> values that need to be specified. If certain complexes cannot form, values of 0.0 should be entered.

MOLECULEKON <integer> <double_1> <double_2> <double_...>

This specifies the values of the on-rate, in <double> M⁻¹s⁻¹, of molecules of type <integer> when they try to form complexes with the other types of molecules. The amount of defined molecule types determines the amount of <double> values that need to be specified. If certain complexes cannot form, values of 0.0 should be entered.

MOLECULENAME <integer> <text>

This specifies the name of the molecule <integer>. This is used by the trace view to facilitate the representation of different types of molecules and when saving the distribution of molecules.

MOLECULERELEASEANGLE <integer> <double>

When using a round contact area, this specifies the angle, in <double> radians, from the sphere vector that internalized molecules of type <integer> are externalized on the sphere surface. A value of 3.1416 indicates that molecules are reintroduced at the other end of the sphere from where the sphere vector points. A value of 1.5708 indicates that molecules are reintroduced along the equator. Not all molecules are externalized at the same location, and the spread of this area is represented by a normal distribution.

When using a flat contact area, this specifies the distance, in <double> micrometers, from the middle of the contact area that internalized molecules of type <integer> are reintroduced on the surface.

If the specified distance is 0.0 and the value specified by MOLECULERELEASESTDEV is also 0.0, then molecules are evenly externalized at random locations on the sphere surface.

MOLECULERELEASESTDEV <integer> <double>

When using a round contact area, this specifies the spread of the area, in <double> radians, that corresponds to one standard deviation of a normal curve. This is used in conjunction with MOLECULERELEASEANGLE when internalized molecules of type <integer> are externalized on the sphere surface.

When using a flat contact area, this specifies the spread of the distance, in <double> micrometers, that corresponds to one standard deviation of a normal curve. This is used in conjunction with MOLECULERELEASEANGLE when internalized molecules of type <integer> are externalized on the sphere surface.

If the specified distance is 0.0 and the value specified by MOLECULERELEASEANGLE is also 0.0, then molecules are evenly externalized at random locations on the sphere surface.

MOLECULERESTINGDEGRADE <integer> <double>

This specifies the first order rate constant in <double> per second for a molecule of type <integer> to become degraded if it is currently inactive and internal.

MOLECULEVOLUMEHEIGHT <integer> <double>

This specifies the height, in <double> meters, used to convert the two dimensional area of a region into a volume when calculating the concentration of two types of molecules that are trying to bind to each other. The molecule type is specified by <integer>. This value is also known as the confinement distance.

Region Commands

REGIONHEIGHTS <double_1> <double_2> <double_...>

This is used to specify the membrane separations in particular regions that are caused by virtual molecules. The <double> values should be on the order of nanometers (e.g. 5.0E-9).

If using a curved contact area, the <double> values specify the separation of opposing membranes inside of a particular ring of the contact area. The first <double> value is for the innermost ring. If no additional values are provided, this <double> value is used for all other rings as well. Otherwise, a <double> value must be provided for each ring.

If using a flat contact area, only one <double> value is specified and this value applies to all regions.

REGIONWEIGHTS <integer_1> <integer_2> <integer_...>

This is used to specify the number of virtual molecules.

If using a curved contact area, the <integer> values specify how many virtual bound molecules **per cell** give rise to the separation specified by the REGIONHEIGHTS command. The first value is for this innermost ring of the contact area. If no additional values are provided, this value is used for other rings as well. Otherwise a value must be provided for each ring. The simulation then adjusts the per cell number to represent the amount of molecules found in the region being calculated. In conjunction with the REGIONHEIGHTS command, the simulation then calculates the separation of membranes in a region by multiplying the region-adjusted value by the specified height, adding the heights of the complexes inside of the region, and dividing the result by the sum of the region-adjusted value and the amount of complexes inside of the region.

If using a flat contact area, the single <integer> value specifies how many virtual bound molecules **per contact area** give rise to the separation specified by the REGIONHEIGHTS command in each region.

Sphere Commands

SPHERE2DTARGET <integer>

When using a curved contact area, this specifies that the target a molecule will diffuse towards will be found on a plane that contains both the molecule, the center of the contact area of sphere <integer>, and the center of sphere <integer>.

SPHERE3DTARGET <integer>

When using a curved contact area, this specifies that the target a molecule will diffuse towards can be found anywhere on sphere <integer>.

SPHEREANGLE <integer> <double>

When a curved contact area is used, this specifies the angle, in <double> radians, of the contact area of sphere <integer> from the middle of the contact area using a vector given by SPHEREVECTOR. This command has no function when using a flat contact area.

SPHERELOCATION <integer> <double_x> <double_y> <double_z>

This specifies the x/y/z location of the center of sphere <integer>. This location cannot be changed once molecules have been seeded. One unit is a micrometer.

SPHERERADIUS <integer> <double>

This specifies the radius, in <double> micrometers, of sphere <integer>. If a flat contact area is used, this must be larger than the radius of the contact area.

SPHERESHORT <integer>

When a flat contact area is used, there are two locations where a contact area with a given radius can reside on the sphere. This moves the contact area such that at most half of sphere <integer> is present. SPHERETALL is the opposite command. SPHEREVECTOR points up from the contact area.

SPHERETALL <integer>

When a flat contact area is used, there are two locations where a contact area with a given radius can reside on the sphere. This moves the contact area such that at least half of sphere <integer> is present and is the default. SPHERESHORT is the opposite command. SPHEREVECTOR points up from the contact area.

SPHEREVECTOR <integer> <double_x> <double_y> <double_z>

This specifies the x/y/z direction of a vector from the center of sphere <integer> to the contact area. The length of the vector does not matter but you cannot use a null vector. Using the combination “0.0 0.0 1.0” puts the contact area at the top of the sphere.

SPHEREVECTORSHIFT <integer> <double>

When using a round contact area, this specifies the **angle**, in <double> radians, that the normal curve that is usually centered over the SPHEREVECTOR of sphere <integer> is shifted away from that point and is used to choose a target for diffusion when diffusion is biased for one or more molecules. Whether or not the target is in the same plane as the molecule, the center of the sphere, and the center of the contact area is determined by SPHERE2DTARGET and SPHERE3DTARGET.

When using a flat contact area, this specifies the **distance**, in <double> micrometers from the middle of the contact area, that the normal curve that is usually centered over the SPHEREVECTOR of sphere <integer> is shifted away from that point and is used to choose a target for diffusion when diffusion is biased for one or more molecules. The chosen target will reside inside the contact area.

SPHEREVECTORSTDEV <integer> <double>

When using a round contact area, this specifies the **angle**, in <double> radians, that represents one standard deviation of the normal curve that is centered over the SPHEREVECTOR of sphere <integer> and is used to choose target for diffusion when diffusion is biased for one or more molecules. Whether or not the target is in the same plane as the molecule, the center of the sphere, and the center of the contact area is determined by SPHERE2DTARGET and SPHERE3DTARGET.

When using a flat contact area, this specifies the **distance**, in <double> micrometers, that represents one standard deviation of the normal curve that is centered over the SPHEREVECTOR of sphere <integer>, is shifted away from it by SPHEREVECTORSHIFT, and is used to choose a target for diffusion when diffusion is biased for one or more molecules. The chosen target will reside inside the contact area.

Photobleaching Commands

PHOTBLEACH <integer> <double_start_x> <double_start_y> <double_start_z> <double_end_x> <double_end_y> <double_end_z> <double_radius>

This command makes all molecules of type <integer> invisible that are found within <double_radius> micrometers of a line that goes through start_x/y/z and end_x/y/z.

PHOTBLEACHCONE <integer> <double_start_x> <double_start_y> <double_start_z> <double_end_x> <double_end_y> <double_end_z> <double_angle>

This command makes all molecules of type <integer> invisible that are found within set of cones that that converge on end_x/y/z and include start_x/y/z. The angle between the middle of the cones to the side is given by <double_angle> radians.

UNBLEACH

This command makes all molecules of type <integer> visible if they have previously been photobleached.

One-time Writing Commands

These commands are for writing to new files, or appending to existing files, on a one-time basis. It is possible to specify the filename each time.

APPENDACTIVEIMAGE <filename>

This is similar to APPENDIMAGE but only renders molecules that are currently activated using their activated colors.

APPENDBOUNDIMAGE <filename>

This is similar to APPENDIMAGE but only renders molecules that are currently bound using their bound colors.

APPENDCONTACTAREAIMAGE <filename>

This is similar to APPENDIMAGE but only renders molecules that are currently inside the contact area.

APPENDDISTRIBUTION <filename>

This is similar to SAVEDISTRIBUTION but appends the data to the file specified by <filename> if the file already exists.

APPENDIMAGE <filename>

This saves a 640x480 RAW image of the current view where each molecule is represented by its color. Each pixel is represented by three bytes with values ranging from 0 to 255 that represent the red, green, and blue values for that pixel. Pixels are stored left to right starting with the top row. If more than one molecule occupies a pixel, the value is the sum of the molecules occupying that pixel. Unlike SAVEIMAGE, this appends the image to the file specified by <filename> if the file already exists. This can be used to generate a large file that can later be broken down into smaller files so that color, contrast, and other parameters can be adjusted identically for all images.

SAVEACTIVEIMAGE <filename>

This is similar to SAVEIMAGE but only renders molecules that are currently activated using their activated colors.

This command will overwrite files with the name <filename>

SAVEBOUNDIMAGE <filename>

This is similar to SAVEIMAGE but only renders molecules that are currently bound using their bound colors.

This command will overwrite files with the name <filename>

SAVECONTACTAREAIMAGE <filename>

This is similar to SAVEIMAGE but only renders molecules that are currently inside the contact area.

This command will overwrite files with the name <filename>

SAVEDISTRIBUTION <filename>

This saves a file containing the molecule distribution data for each type of molecule.

When using a curved contact area, each line contains the simulation time, the angle of the molecules from the sphere vector (0-179 degrees), the number of molecules of each type at that angle, and the area of a sphere for that angle that has a radius of 1. Thus, the observed/expected value for molecules at a given angle can be calculated by dividing the observed molecules at that angle by the product of the total molecules (of a given type) and the area of the angle divided by 4π .

When using a flat contact area, the output is similar except that the angles refer to a distance from the center of the contact area (with 180 even divisions of this distance; 0 is for the middle of the contact area; 179 is for the edge of the contact area) and the area values are based on a circle that has a radius of 1. Thus, the observed/expected value for molecules at a given "angle" can be calculated by dividing the observed molecules at that "angle" by the product of the expected molecules inside the contact area and the area of the "angle" divided by π .

This data is represented visually in the trace view during the simulation.

This command will overwrite files with the name <filename>

SAVEIMAGE <filename>

This saves a 640x480 RAW image of the current view where each molecule is represented by its color. Each pixel is represented by three bytes with values ranging from 0 to 255 that represent the red, green, and blue values for that pixel. Pixels are stored left to right starting with the top row. If more than one molecule occupies a pixel, the value is the sum of the molecules occupying that pixel.

This command will overwrite files with the name <filename>

SAVEPOINTS <filename>

This saves a comma-delimited file containing the locations of all molecules being used by the simulation. This can be a rather large file.

This command will overwrite files with the name <filename>

Continuous Writing Commands

These commands are for opening, writing to, and closing files. The names of the files are specified when they are opened and the files remain open for writing until they are closed. These files must be closed at the end of the simulation.

CLOSEACTIVEFILE

This closes the file that was opened with OPENACTIVEFILE.

CLOSEBOUNDFILE

This closes the file that was opened with OPENBOUNDFILE.

CLOSECONTACTAREAFILE

This closes the file that was opened with OPENCONTACTAREAFILE.

CLOSEDISTRIBUTION

This closes the file that was opened with OPENDISTRIBUTION.

CLOSEFILES

This closes any files that were opened with OPENACTIVEFILE, OPENBOUNDFILE, OPENCONTACTAREAFILE, OPENDISTRIBUTION, and OPENIMAGEFILE

CLOSEIMAGEFILE

This closes the file that was opened with OPENIMAGEFILE.

OPENACTIVEFILE <filename>

This opens a file for saving 640x460 RAW images of the activated molecules based on the current view where each activated molecule is represented by its activated color. Each pixel is represented by three bytes with values ranging from 0 to 255 that represent the red, green, and blue values for that pixel. Pixels are stored left to right starting with the top row. If more than one activated molecule occupies a pixel, the value is the sum of the activated molecules occupying that pixel.

This file must be closed using either the CLOSEACTIVEFILE or CLOSEFILES command.

This file can be written to using the WRITEACTIVEFILE command.

This command will overwrite files with the name <filename>

OPENBOUNDFILE <filename>

This opens a file for saving 640x460 RAW images of the bound molecules based on the current view where each bound molecule is represented by its bound color. Each pixel is represented by three bytes with values ranging from 0 to 255 that represent the red, green, and blue values for that pixel. Pixels are stored left to right starting with the top row. If more than one bound molecule occupies a pixel, the value is the sum of the bound molecules occupying that pixel.

This file must be closed using either the CLOSEBOUNDFILE or CLOSEFILES command.

This file can be written to using the WRITEBOUNDFILE command.

This command will overwrite files with the name <filename>

OPENCONTACTAREAFILE <filename>

This opens a file for saving 640x460 RAW images of the molecules inside the contact area based on the current view where each molecule is represented by its color. Each pixel is represented by three bytes with values ranging from 0 to 255 that represent the red, green, and blue values for that pixel. Pixels are stored left to right starting with the top row. If more than one molecule occupies a pixel, the value is the sum of the molecules occupying that pixel.

This file must be closed using either the CLOSECONTACTAREAFILE or CLOSEFILES command.

This file can be written to using the WRITECONTACTAREAFILE command.

This command will overwrite files with the name <filename>

OPENDISTRIBUTION <filename>

This opens a file containing the molecule distribution data for each type of molecule.

When using a curved contact area, each line contains the simulation time, the angle of the molecules from the sphere vector (0-179 degrees), the number of molecules of each type at that angle, and the area of a sphere for that angle that has a radius of 1. Thus, the observed/expected value for molecules at a given angle can be calculated by dividing the observed molecules at that angle by the product of the total molecules (of a given type) and the area of the angle divided by 4π .

When using a flat contact area, the output is similar except that the angles refer to a distance from the center of the contact area (with 180 even divisions of this distance; 0 is for the middle of the contact area; 179 is for the edge of the contact area) and the area values are based on a circle that has a radius of 1. Thus, the observed/expected value for molecules at a given “angle” can be calculated by dividing the observed molecules at that “angle” by the product of the expected molecules inside the contact area and the area of the “angle” divided by π .

This data is represented visually in the trace view during the simulation.

This file must be closed using either the CLOSEDISTRIBUTION or CLOSEFILES command.

This file can be written to using the WRITEDISTRIBUTION command.

This command will overwrite files with the name <filename>

OPENIMAGEFILE <filename>

This opens a file for saving 640x460 RAW images of the current view where each molecule is represented by its color. Each pixel is represented by three bytes with values ranging from 0 to 255 that represent the red, green, and blue values for that pixel. Pixels are stored left to

right starting with the top row. If more than one molecule occupies a pixel, the value is the sum of the molecules occupying that pixel.

This file must be closed using either the CLOSEIMAGEFILE or CLOSEFILES command.

This file can be written to using the WRITEIMAGEFILE command.

This command will overwrite files with the name <filename>

WRITEACTIVEFILE

This adds a 640x480 RAW image of the current view to the file that was opened with OPENACTIVEFILE where each activated molecule is represented by its activated color. Each pixel is represented by three bytes with values ranging from 0 to 255 that represent the red, green, and blue values for that pixel. Pixels are stored left to right starting with the top row. If more than one activated molecule occupies a pixel, the value is the sum of the activated molecules occupying that pixel.

WRITEBOUNDFILE

This adds a 640x480 RAW image of the current view to the file that was opened with OPENBOUNDFILE where each bound molecule is represented by its bound color. Each pixel is represented by three bytes with values ranging from 0 to 255 that represent the red, green, and blue values for that pixel. Pixels are stored left to right starting with the top row. If more than one bound molecule occupies a pixel, the value is the sum of the bound molecules occupying that pixel.

WRITECONTACTAREAFILE

This adds a 640x480 RAW image of the molecules inside the contact area based on the current view to the file that was opened with OPENCONTACTAREAFILE where each molecule is represented by its color. Each pixel is represented by three bytes with values ranging from 0 to 255 that represent the red, green, and blue values for that pixel. Pixels are stored left to right starting with the top row. If more than one molecule occupies a pixel, the value is the sum of the molecules occupying that pixel.

WRITEDISTRIBUTION

This adds data to the file that was opened with OPENDISTRIBUTION.

When using a curved contact area, each line contains the simulation time, the angle of the molecules from the sphere vector (0-179 degrees), the number of molecules of each type at that angle, and the area of a sphere for that angle that has a radius of 1. Thus, the observed/expected value for molecules at a given angle can be calculated by dividing the observed molecules at that angle by the product of the total molecules (of a given type) and the area of the angle divided by 4π .

When using a flat contact area, the output is similar except that the angles refer to a distance from the center of the contact area (with 180 even divisions of this distance; 0 is for the middle of the contact area; 179 is for the edge of the contact area) and the area values are based on a circle that has a radius of 1. Thus, the observed/expected value for molecules at a given "angle" can be calculated by dividing the observed molecules at that "angle" by the product of the expected molecules inside the contact area and the area of the "angle" divided by π .

WRITEIMAGEFILE

This adds a 640x480 RAW image of the current view to the file that was opened with OPENIMAGEFILE where each molecule is represented by its color. Each pixel is represented

by three bytes with values ranging from 0 to 255 that represent the red, green, and blue values for that pixel. Pixels are stored left to right starting with the top row. If more than one molecule occupies a pixel, the value is the sum of the molecules occupying that pixel.

Sample Configuration File

```
; suppress most output by including OUTPUTFILE without a filename
OUTPUTFILE
; select a contact area model
; either USEFLATCONTACT or USECURVEDCONTACT
USEFLATCONTACT
; shift the grid of regions up a half region for every second column
UESHIFTEDGRID
; use brick-shaped regions
USEBRICKGRID
; specify the radius of the contact area
; we will be using cells with radii of 5 micrometers
CONTACTRADIUS 4.0
; divide the contact area into a 20x10/11 grid
; without USEBRICKGRID, this would be a 20x20/21 grid
; without UESHIFTEDGRID, this would be a 20x20 grid
REGIONS 20
; specify the maximum molecules per region per cell
MAXPERREGION 5000
; specify the number of virtual molecules inside the contact area
REGIONWEIGHTS 1000
; specify the heights of the virtual molecules (40 nanometers)
REGIONHEIGHTS 40.0E-9
; specify the number of spheres that molecules diffuse/move on
; this is at least one and can be very high, if needed
; each cell can be represented by many spheres
; a single sphere can be used to represent both cells
SPHERES 1
; specify the number of different molecule types that will be used
MOLECULES 2

; specify the parameters for the single sphere we will be using
SPHERELOCATION      0 0.0 0.0 5.0
SPHERETALL         0
SPHEREVECTOR       0 0.0 0.0 1.0
SPHERERADIUS       0 5.0
SPHEREVECTORSTDEV 0 0.5
SPHEREVECTORSHIFT 0 0.0

; move the eye to look at the middle of the sphere
; assume that the 640x480 view is 10 micrometers tall
; assume that we are looking at the view from 30 micrometers away
MOVETO 0.0 60.0 60.0
LOOKAT 0.0 0.0 5.0
VIEWDISTANCE 30.0 10.0

; define the molecules

; while both molecules move and diffuse on the same sphere,
; they are found on separate cells

; molecule 0
TITLETEXT Defining Molecule 0
```

MOLECULENAME	0	Molecule 0	
MOLECULEFOUNDON	0	0	
MOLECULECELL	0	0	
MOLECULECOLOR	0	5 0 0	
MOLECULEBOUNDCOLOR	0	5 0 0	
MOLECULEACTIVECOLOR	0	5 0 0	
MOLECULEENTRYHEIGHT	0	21.0E-9	
MOLECULEENTRYSTDEV	0	5.25E-9	
MOLECULEBOUNDENTRY	0	0.0	10.0E-9
MOLECULEFREEDIFFUSION	0	2.9E-14	
MOLECULEBOUNDDIFFUSION	0	0.0	1.2E-14
MOLECULEKON	0	0.0	2.0E+5
MOLECULEKOFF	0	0.0	0.1
MOLECULEBINDDISTANCE	0	0.0	40.0E-9
MOLECULEBINDSTDEV	0	0.0	20.0E-9
MOLECULEVOLUMEHEIGHT	0	2.00E-9	
MOLECULEFREEMOVEBIAS	0	0.0	
MOLECULEBOUNDMOVEBIAS	0	0.0	0.0
MOLECULERELEASEANGLE	0	0.0	
MOLECULERELEASESTDEV	0	0.0	
MOLECULEFRACTIONINTERNAL	0	0.0	
MOLECULEFRACTIONDEGRADED	0	0.0	
MOLECULEADDPERSECOND	0	0.0	
MOLECULEEXTERNALIZE	0	-1.0	
MOLECULEEXTERNALACTIVATE	0	-1.0	-1.0
MOLECULEEXTERNALDEACTIVATE	0	-1.0	
MOLECULEINTERNALDEACTIVATE	0	-1.0	
MOLECULEACTIVEDGRADE	0	-1.0	
MOLECULEREESTINGDEGRADE	0	-1.0	
MOLECULEBOUNDACTIVEINTERNALIZE	0	-1.0	-1.0
MOLECULEBOUNDRESTINGINTERNALIZE	0	-1.0	-1.0
MOLECULEFREEACTIVEINTERNALIZE	0	-1.0	
MOLECULEFREERESTINGINTERNALIZE	0	-1.0	

; molecule 1

TITLETEXT Defining Molecule 1

MOLECULENAME	1	Molecule 1	
MOLECULEFOUNDON	1	0	
MOLECULECELL	1	1	
MOLECULECOLOR	1	0 5 0	
MOLECULEBOUNDCOLOR	1	0 5 0	
MOLECULEACTIVECOLOR	1	0 5 0	
MOLECULEENTRYHEIGHT	1	19.0E-9	
MOLECULEENTRYSTDEV	1	4.75E-9	
MOLECULEBOUNDENTRY	1	10.0E-9	0.0
MOLECULEFREEDIFFUSION	1	2.0E-14	
MOLECULEBOUNDDIFFUSION	1	1.2E-14	0.0
MOLECULEKON	1	2.0E+5	0.0
MOLECULEKOFF	1	0.1	0.0
MOLECULEBINDDISTANCE	1	40.0E-9	0.0
MOLECULEBINDSTDEV	1	20.0E-9	0.0
MOLECULEVOLUMEHEIGHT	1	2.00E-9	
MOLECULEFREEMOVEBIAS	1	0.0	
MOLECULEBOUNDMOVEBIAS	1	0.0	0.0
MOLECULERELEASEANGLE	1	0.0	
MOLECULERELEASESTDEV	1	0.0	
MOLECULEFRACTIONINTERNAL	1	0.0	
MOLECULEFRACTIONDEGRADED	1	0.0	
MOLECULEADDPERSECOND	1	0.0	
MOLECULEEXTERNALIZE	1	-1.0	

MOLECULEEXTERNALACTIVATE	1	-1.0	-1.0
MOLECULEEXTERNALDEACTIVATE	1	-1.0	
MOLECULEINTERNALDEACTIVATE	1	-1.0	
MOLECULEACTIVEDEGRADE	1	-1.0	
MOLECULERESTINGDEGRADE	1	-1.0	
MOLECULEBOUNDACTIVEINTERNALIZE	1	-1.0	-1.0
MOLECULEBOUNDRESTINGINTERNALIZE	1	-1.0	-1.0
MOLECULEFREEACTIVEINTERNALIZE	1	-1.0	
MOLECULEFREERESTINGINTERNALIZE	1	-1.0	

; now seed 100,000 molecules of each type

TITLETEXT Seeding Molecules

MOLECULESEED 0 100000

MOLECULESEED 1 100000

; set time parameters for the simulation

RESETTIME

; simulate in 0.1 second increments

TIMEINTERVAL 0.1

; each time there is a GO, simulate 100 of the 0.1 second increments

ITERATIONS 100

; wait before we continue

ASK

; this is commented out so that we do not open any files

;TITLETEXT Opening files

; change the destination of the file for your OS

;OPENIMAGEFILE /tmp/imagefile.raw

; save the current view

;WRITEIMAGEFILE

TITLETEXT Simulating 10 seconds

; ask the user to press the GO button

ASK

GO

; save the view now that 10 seconds have elapsed

;WRITEIMAGEFILE

; the file now contains two 640x480 images

; close the files

;CLOSEFILES

TITLETEXT Done!

ASK