

Frequently Asked Questions

The frequently asked questions about OpenSim are broken up into several categories:

- [Installation and Compatibility](#)
- [Visualization and Graphics](#)
- [Importing and Exporting Data](#)
- [Muscle Models](#)
- [Plotting](#)
- [Scaling and Customizing Models](#)
- [Generating Simulations](#)
- [Applying OpenSim](#)
- [Scripting](#)
- [OpenSense](#)
- [OpenSim vs SIMM](#)

Installation and Compatibility

Q. Can I run OpenSim application on 32 bit Windows machines?

A. As of version 4.0 we're only supporting 64 bit windows machines as these are the most commonly used Windows platform. If you have 32 bit windows machine you can build the API from sources or use an earlier version (e.g. OpenSim 3.3)

Q. Does OpenSim work on a Mac?

A. Starting with OpenSim 4.0, we provide a version of the graphical user interface compatible with Mac.

Q. Do you need Matlab to run OpenSim?

A. No. You only need Matlab to use the Matlab scripting. If you don't have Matlab, you can still use the application or GUI, the GUI or python scripting shell, and the API.

Q: Do I have to uninstall the previous version of OpenSim before installing a new version?

A. No. You can keep old installations of OpenSim to compare your results.

Q: Where can I download previous releases of OpenSim?

A. You can access all previous releases through the [OpenSim Downloads Page](#). Directly under the current version is the drop down menu 'List all previous releases'.

Q: Can I run OpenSim from a USB drive?

A. If you would like to run OpenSim on a Windows machine, but you don't have install privileges, you can try running OpenSim from a USB drive. This may be useful for instructors of biomechanics courses, for example, allowing students to run OpenSim on library computers. To set up the drive, insert the USB drive and install OpenSim, selecting a path the USB drive as your install directory. Note that you must do this on a Windows machine. After installing you can then run OpenSim from another Windows machine. You will likely need to update the Geometry folder under **Edit > Preferences...**, to the appropriate path on the new machine.

Visualization and Graphics

Q. Why can't I launch the GUI's Visualizer Window?

A: The OpenSim GUI has been unable to launch the visualizer on some Windows and Mac machines. The most common reasons are:

- No WebGL support: some older machines do not support Hardware Accelerated Graphics rendering used by the new OpenSim Visualization. Unfortunately, no workaround is available for this, you'll either need to upgrade your graphics card, update drivers, or revert to earlier version.
- Collision with other applications on Mac: OpenSim internally uses a third party library that happens to collide with some rarely used applications on Mac (e.g., RescueTime and Contexts). We are investigating workarounds to resolve this issue. If you have a similar issue, please report it.

Q. How do I tell OpenSim where to locate the bone/geometry files?

A. Open Edit/Preferences and edit the entry for "Paths: Geometry Search Path". This entry is a semicolon-separated list of directories. See [User Preferences](#) for more information.

Q. What are .vtp, .stl, .obj files?

A. These are the geometry files corresponding to the bones and other objects used by your model. The "vtp" format is a format used by VTK which was used by earlier versions of OpenSim for visualization. vtp files come in two flavors (ASCII, Binary) we only support ASCII format. Both .stl, .obj are standard mesh file formats as well.

Q. My meshes look different in 4.0 than they did in earlier versions of OpenSim

A. In version 4.0 we switched from using vtk (for parsing mesh files) to native parsers built into the SimTK libraries. Some STL, OBJ files are not handled by the SimTK parsers, which try to build closed meshes from these files for contact purposes. For example, files with multiple solids, or no solids may not be handled the same in 4.0.

Q. OpenSim loads many models in the same graphics window, how do I know which one I'm working on now?

A. Although many models are loaded, only a single model can be marked as "Current". This is indicated in bold in the navigator.

Q. I can't pick some objects in the Graphics window, is this a bug?

A. Picking is restricted to "Current model" (by default non-current models are dimmed out). If you want to pick on a non-current model you'll have to make it current first.

Q. I can't pick some Muscle Points in the Visualizer window why?

A. Only objects that are pickable in the Visualizer window are Path Points and Markers. Moving Muscle Points are special Path Points that have Splines describing their trajectories so they can't be picked and dragged from the GUI.

Importing and Exporting Data

Q. Would it be possible to open a C3D file directly in OpenSim?

A. Currently, the OpenSim Plotter can load and plot the forces from a C3D file.

Q. Can one extract the muscle force magnitudes, points of origin and direction of action?

A. OpenSim muscles provide access to their force and their GeometryPath, which includes their point of origin. Using this information, you can compute the direction of action. Use the OpenSim scripting interfaces to access this information. You can also use OpenSim Analyses or Reporters to obtain some of this information. Depending on your needs and programming experience you may write your own code to export this data. Alternatively there is a plugin https://simtk.org/projects/force_direction that allows exporting of these forces/line-of-actions to files for interface to FEA. Please check the website for the proper version that is compatible with your OpenSim installation.

Q: Is it possible to get position over time data from a model? For example can I track a point on the shank of the leg as the model is moving?

A. Yes, one can obtain the position of a point over time using the "location" Output on Point objects (e.g., Markers and Stations).

Q: Can we import a Solidworks model into OpenSim? For example, a prosthetic socket.

A. There are multiple steps to importing a CAD model into OpenSim. Getting the Geometry is straightforward, since you can export individual parts to stl/obj files that are read directly by OpenSim (when attached to an Body in OpenSim model .osim file). Joints/Constraints/masses are not imported; however, and you need to reimplement them on your own using OpenSim objects (Joints, Constraints, etc.).

Muscle Models

Q. What muscle models are included in OpenSim and what do you have to develop yourself? Are there some generic muscle models available I can use to get going?

A. There are many muscle models implemented in OpenSim.

One of the most commonly used muscles is the Thelen2003Muscle, based on the model described in Thelen, 2003.

Thelen DG. Adjustment of muscle mechanics model parameters to simulate dynamic contractions in older adults. J Biomech Eng. 2003; 125(1):70-7.

We've also added some new muscle models. Watch [Matt Millard's webinar](#) to find out more.

Q. Do OpenSim muscle models account for the delay within activation dynamics in your model? If yes, how?

A. Yes, the muscle models in OpenSim model the electro-mechanical delay between muscle excitation (or the electrical neural signal sent to the motor-neuron junction) and activation (i.e., the level of force production). From more details on the theory behind our muscle models, check out this great review paper by Felix Zajac: Zajac FE. Muscle and tendon: properties, models, scaling, and application to biomechanics and motor control. Crit Rev Biomed Eng. 1989; 17(4):359-411.

Q. Is there any way that you can incorporate stretch activation of muscle to try and get the model to predict the activation of muscle before foot strike?

A. Sure! You could develop a custom controller in the OpenSim API that is based upon the stretch of muscle or tendons in the model and map that back to predicted activations.

Q. Is the orientation (i.e., pennation angle) or constitution of fibers (e.g., slow vs. fast) taken into account in the muscle models?

A. Yes, we do model the orientation, or pennation angle, of muscle fibers, but no, we do not currently model slow versus fast twitch fibers. For more details on the theory behind these muscle models, check out this great review paper by Felix Zajac: Zajac FE. Muscle and tendon: properties, models, scaling, and application to biomechanics and motor control. Crit Rev Biomed Eng. 1989;17(4):359-411.

Q. Can the geometry of the muscles be changed? Like wrapping around bones/other muscles?

A. When defining a muscle in OpenSim you specify a geometry path. This path can wrap only around "wrap objects" (simplified geometry to speed up path computation). Currently supported types are: Sphere, Cylinder, Ellipsoid, Torus or parts of these objects. Arbitrary meshes can not be specified as wrap objects.

Plotting

Q. How can I import data into OpenSim Plotter?

A. Under Y-Quantity, select "Load file..." and then you can read data files in the storage or motion format.

Q. Can I export data from a plot into a ASCII file for processing by Matlab or use in Excel?

A. Under the "curves list", right click on the top level node and select "Export Data...". OpenSim will prompt you for a file name to save the data into. This file can be later re-imported into OpenSim or used by other applications.

Scaling and Customizing Models

Q. What anthropometric database is used for model scaling?

A. OpenSim does not use a database for scaling. The model is scaled based on measurements of experimental markers. See [How Scaling Works](#) for more information.

Q. When you are scaling a generic model to match a particular subject, how do you account for the difference in bone length, like, if someone has a proportionally shorter tibia?

A. Scaling in OpenSim scales each body segment independently based on measurements of markers placed on that segment. See the section on [Scaling](#) for more details.

Q. Why have some studies with OpenSim scaled the model's muscle strength? Does this reflect the model's development from cadaver data?

A. Yes, the optimal force or max strength of the muscles in the model used in many studies are based on Delp, 1990 (see reference below), which determined muscle parameters from cadavers. While the strength of the model was not increased to study running (Hamner et al., 2010), other researchers have noticed that the "strength" (i.e., optimal force) of the muscles has to be increased for movements with higher forces and accelerations, like running and cutting, or for pathological gaits, such as crouch gait. Delp SL et al., An interactive graphics-based model of the lower extremity to study orthopedic surgical procedures. IEEE Trans Biomed Eng. 1990; 37(8):757-67.

Q. Are muscle forces or strength scaled based on the subject's weight (compared to the generic model weight)?

A. No, the scale tool in OpenSim does not automatically scale muscle forces based on the geometric based scaling (i.e., scaling based on distance measurements between markers) or the mass scaling. However, you can scale muscle forces in the model based manually if you have a criterion for doing this. See the [Scripting](#) chapter for info about how to do this with a script.

Q. How can you incorporate BIODEX data into your simulations to scale models?

A. BIODEX data can be used to determine the strength, or torque generating capacity of your subject. Researchers have used this information to modify the optimal muscle forces used in their model. For an example, check out the paper by Anderson & Pandy from 2001. They explain, "Values of peak isometric muscle force and tendon slack length were adjusted so that the maximum isometric torque angle curves for each joint in the model matched average torque-angle curves measured for the five subjects." Anderson FC, Pandy MG. Dynamic optimization of human walking. J Biomech Eng. 2001;123(5):381-90.

Generating Simulations

Q. What are typical settings that are modified during RRA?

A. RRA is an optimization algorithm whose goal is to minimize the non-physical "residual" forces acting on a dynamic simulation. The settings modified for RRA are the inputs to the optimization algorithm, which include the excitation limits (i.e., min/max controls) for the residual actuators along with the tracking weights for each of the coordinates (i.e., joint angles). See [Getting Started with RRA](#) for more information.

Q. Can you briefly describe available ways to compute muscle forces in OpenSim besides CMC?

A. Yes, OpenSim has another tool called [Static Optimization](#), which has been used by numerous researchers to estimate muscle forces. Check out this paper describing static optimization: Anderson FC, Pandy MG. Static and dynamic optimization solutions for gait are practically equivalent. J Biomech. 2001;34(2):153-61.

Q. Does CMC take synergistic muscles into account?

A. No, we do not explicitly model synergistic muscle groups. Computed muscle control treats each muscle as an independently controlled actuator. However, our results show that muscles are still activated in the observed synergistic groups.

Applying OpenSim

Q. How can I find a good model for my research study?

See our [Musculoskeletal Models](#) page for a list of OpenSim models that we've developed at Stanford, as well as a link to the library of models from our team and other researchers on Simtk.org. Please also see our page on [Evaluating a Musculoskeletal Model](#). It has a list of questions to review when deciding what model to use in your research.

Q. Has anyone used Opensim to simulate pathological gait, like stroke patients or OA patients? What is the difficulty of simulating pathological gait?

A. Yes, students and researchers in the Neuromuscular Biomechanics Lab at Stanford have created muscle-actuated simulations of individuals with cerebral palsy with a crouch gait. Check out a recent study led by Kat Steele. Steele, K.M., Seth, A., Hicks, J.L., Schwartz, M.S., Delp, S.L. Muscle contributions to support and progression during single-limb stance in crouch gait. Journal of Biomechanics, vol 43, pp 2099-2105 and other papers on the [Neuromuscular Biomechanics Lab Page](#). See the [OpenSim Website](#) for more research applications.

Q. I am new to these kinds of simulations and would like to make a new model of a simple experimental set up (in this case an inverted pendulum). What software could I use to generate this for dynamic analysis?

A. For dynamic simulations of rigid body systems, like a double pendulum, check out Simbody(TM), which is the rigid-body dynamics engine used by OpenSim. It is open-source and freely available with a well-tested and stable API. The project page is <https://simtk.org/home/simbody>.

Q: How useful do you feel Opensim currently and in the future will be at smoothly importing, calculating, and reporting clinical gait data?

A. Importing data can sometimes be time consuming and require troubleshooting. The most difficult part is setting up a custom workflow that fits our lab and needs. See [Preparing Your Data](#) for more information about this process. We've added [Scripting](#) to make it easier to customize your workflows. Other users have also contributed their tools in the [OpenSim Utilities](#) library. We also plan to continue to improve our usability going forward, as one of our top priorities.

Q: Can you simulate contact forces (e.g. between the foot and a platform) in OpenSim?

Yes. You can model Hunt Crossley Contact Forces and add contact geometry. See the [OpenSim Doxygen](#) for more information. Check out the XML Browser in the GUI (under Help) to find the xml tags to add the force to your model.

Scripting

Q. Does OpenSim support python scripting? (or only MATLAB?)

A. Yes, OpenSim version 3.2 and later supports scripting in Python, Jython (an implementation of Python in Java), and MATLAB. In the OpenSim application, the scripting shell is a Jython shell. All scripting capabilities use a technique (called wrapping) to invoke the OpenSim C++ API from Python, Java, or MATLAB.

Q. Is there full documentation on how the API is exposed to Scripting?

A. You should be able to leverage the extensive [Doxygen documentation](#) of the OpenSim API to use it in Scripts. You'll also find the examples included with the distribution as well as the help page on Scripting ([Common Scripting Commands](#)) helpful.

Q. Is the MATLAB interface only for 32 bit MATLAB or will it work for 64 bit as well?

A. Starting with OpenSim 4.0, we only provide 64-bit versions of the software. You can use OpenSim with 64-bit Matlab by downloading and installing the OpenSim GUI. This assumes you are using Windows. If necessary, you can compile OpenSim from the source code as 32-bit. Read more in [Scripting with Matlab](#).

Q. Should OpenSim be running when doing scripting from another language like MATLAB/python?

A. No, the OpenSim application and the scripting interface both utilize the same libraries, but are unrelated otherwise. While scripting in the GUI, you have access to all the API and some GUI functionality (e.g. control of view, plotter, loading models, motions, etc.). While in MATLAB you have access to the API and you can use the MATLAB tools for plotting and the API Visualizer. You can also use the scripting interface from a standalone Python or Jython shell but you have access to neither the GUI functionality nor MATLAB's. See [Common Scripting Commands](#) for more information.

Q. Can OpenSim scripting control all of the GUI's functionality?

A. You can invoke almost any menu command and many other functions (e.g. Plotting) from the GUI scripting layer, but not all dialogs/tools/options are exposed. Please consult the page [Scripting in the GUI](#) for specific examples.

Q. Do you provide a debug mode when scripting?

A. You can turn on more verbose messages so you get more feedback while running scripts by calling `modeling.setDebugLevel(3)`, but for syntax errors you usually get the error message displayed in the scripting shell directly.

Q. How should I do batch processing for the OpenSim tools (e.g. IK, ID, CMC) with a script?

In general, we strongly discourage doing raw parsing of XML files in a Matlab script. This approach is not robust and does not take advantage of the OpenSim API in Matlab. Instead you should instantiate tools/objects from setup files and let the code do the parsing internally. The XML format changes with different versions of the software and the version number of the documents written by newer versions of the software may even be different from the one you read due to backward compatibility built into the code and assumptions about default values, etc. You'll keep changing your code endlessly rather than rely on a robust API that makes calls to set/get values (rather than parse XML files). In essence the assumptions about the contents of the XML files and the XML file header are all built into the code that is exercised by the API but ignored by any XML parsing code. Violating these assumptions will break the functionality downstream.

An example using parsing (not recommended):

```
% get initial and inal time from first and last entry in time column
initial_time = trcData(1,1);
final_time=trcData(end,1);

xmlDoc.getElementsByTagName('IKTool').item(0).getAttributes.item(0).setValue(name);
xmlDoc.getElementsByTagName('marker_file').item(0).getFirstChild.setNodeValue([trialsForIKPath markerFile{1}]);
xmlDoc.getElementsByTagName('output_motion_file').item(0).getFirstChild.setNodeValue([results_folder '\ ' name
'_ik.mot']);
xmlDoc.getElementsByTagName('time_range').item(0).getFirstChild.setNodeValue([num2str(initial_time) ' ' num2str
(final_time)]);
```

An example of calling the API (recommended):

```
% Get trc data to determine time range
markerData = MarkerData(markerDataFileName);

% Get initial and initial time
initial_time = markerData.getStartFrameTime();
final_time = markerData.getLastFrameTime();

% Setup the ikTool for this trial
ikTool.setName(name);
ikTool.setMarkerDataFileName(fullpath);
ikTool.setStartTime(initial_time);
ikTool.setEndTime(final_time);
```

The code snippet above is taken from the example "setupAndRunIKBatchExample.m".

OpenSense

Q. Can OpenSense be used for both lower and upper extremity movements?

A. Yes, OpenSense can be used for both lower and upper extremity movements, although for any population, movement, or musculoskeletal model of interest, you should be sure that you or others have validated the pipeline for the measures you are interested in. To use OpenSense for your use case, make sure to use a model that has the bodies and joints relevant for your dataset and research question.

Q. OpenSense gives joint angles as outputs. Can it also be used to estimate other quantities, such as spatiotemporal (distance and time) metrics or kinetics (including ground reaction forces)?

A. Currently, the OpenSense tools provide only joint angles as outputs; however, the results from the IMU Inverse Kinematics tool can be used with other OpenSim tools to estimate other quantities. For instance, the inverse kinematics results can be used with the Body Kinematics and Point Kinematics tools to calculate the position, velocities, and accelerations of bodies or points in the model throughout the motion. Careful validation should be done when doing this since, unlike markers, positions are not directly measured by IMUs.

Estimating kinetics, such as ground reaction forces or joint moments, from IMU data is a very active area of research, and OpenSense does not currently provide tools to estimate kinetics directly. Workflows that use physics-based simulations to estimate kinetics have been developed to address this. OpenSim Moco is our software package that generates simulations using an advanced optimization technique called direct collocation, and a [Moco example](#) was developed to generate simulations using accelerometer and gyroscope data. Other approaches being researched include using statistical and machine learning techniques to estimate kinetics, without physics-based simulation.

Q. Can OpenSense be used in real-time?

A. Yes, OpenSense can be used in real-time. We provide one such workflow with OpenSenseRT, which uses open-source software and hardware to compute the motions of body segments based on IMU data using a microcontroller worn on the body. Instructions and other details can be found on the [OpenSenseRT documentation page](#).

Q. What import options are there for OpenSense? Which IMU devices can be used for OpenSense?

A. OpenSense requires IMU orientations expressed as quaternions as input into the tools, so sensor fusion must be performed before importing data. We provide import tools for Xsens and APDM systems through our API (C++, MATLAB, and Python) to create an OpenSim storage file, and we have an [OpenSense MATLAB example](#) that shows this step with Xsens data. For other IMUs or if you want to use different sensor fusion algorithms, users can create custom import tools by matching the format in any example files provided with OpenSense.

For users comfortable with using the OpenSim API, if you can obtain IMU quaternions from an API with your IMU system, you can instead populate an OpenSim TimeSeriesTableQuaternion ([TimeSeriesTableQuaternion API entry](#) and [TimeSeriesTable API documentation](#)) and use the Table with the OpenSense tools directly, without using the import tools discussed above to generate a file.

As user needs evolve, we may add other import options in the future.

Q. How does the IMU Placer work? Does it only consider orientation of the IMU or also the position? When should the Heading Correction option be used?

A. The IMU Placer tool uses the IMU calibration data to find the orientations of the IMU frames relative to an OpenSim model's body segments. It does not consider the positions of the IMUs.

The Heading Correction option within the IMU Placer tool allows for the user to indicate a specific axis of an IMU that should be aligned with the model's forward direction. This is helpful to ensure the IMU orientations on the OpenSim model are correct in cases when the participant was not facing in the same direction as the model's forward direction (usually the positive X direction in OpenSim). More details can be found on the [IMU Placer documentation page](#).

Q. Can I scale my OpenSim model using OpenSense?

A. OpenSense does not provide any tools to scale OpenSim models. Models can be scaled either by using marker data or using manual scaling factors. Find out more about scaling at the [Scaling documentation page](#). For some applications where you are only studying kinematics, it may not be necessary to scale your musculoskeletal model. If taking this approach, validation for your particular population and simulation pipeline should be conducted.

Q. For the IMU Placer and IMU Inverse Kinematics tools, how do you determine the rotations between IMU space to OpenSim space?

A. Transforming data from IMU to OpenSim space depends on the individual lab setup, and there are a couple of places to start that can be helpful. First, check the documentation for your IMU system as it can have helpful information about the coordinate system it uses. Second, visualization tools in the OpenSim GUI can help as well by visualizing the IMU orientations and comparing them to the OpenSim ground frame. IMU data can be visualized in the GUI by going to File -> "Preview Sensor Data...". Using this feature with a calibration trial, or any other trial where the IMU orientations in the lab space are straightforward, can help figure out what rotations are needed.

OpenSim vs SIMM

Q. What is the relationship between SIMM and OpenSim

A. Please [review this PDF](#) (updated March 2010)