

# Fabrix Dymanic Fabric Simulator v1.0

By  
Reflection Fabrix, Inc.

Fabrix is a dynamic cloth simulator which, when applied to patch or nurbs models, makes them move in a way that approximates the behaviour of various kinds of fabric.

## **Features**

- By adjusting the parameters of the fabric, various different kinds of fabric can be simulated, from flowing “silk-like” cloth to stiffer “cotton-like” cloth or even rubber or sheet metal.
- Support for multiple fabric objects.
- The fabric object will fall under the influence of gravity. Softimage dynamics gravity controls can be used to affect the gravity (or a default “Earth” gravity can be used)
- The fabric object will react to the medium in which it is placed (air resistance and viscosity).
- The fabric object reacts to Softimage wind and fan controls.
- The fabric objects can be made to interact with other Softimage objects (nurbs or patch) through collision detection.
- Certain points on the cloth object can be animated from within Softimage using “clusters”. e.g. a curtain can be fixed at the top, a skirt can be fixed at the waist.
- Fabric objects can collide with themselves or with other fabric objects.
- Multiple fabric objects can be “stitched” together to form more complex objects (e.g. a shirt can be made from sleeves stitched to the body).
- The friction and dissipation coefficients of the fabric object can be adjusted to simulate friction and “bounce” between cloths and between cloths and obstacles.
- The mass (weight) of the fabric object can be adjusted.
- The dynamic simulation automatically calculates the iterations required, but the user can override the minimum and maximum iterations.
- The simulation is calculated once and cached for later playback.
- Open and closed patches can be used for cloths and collision objects.

The fabrix simulator consists of two effects. The first (Effects->Fabrix\_Properties) is a static effect which sets the parameters of the fabric and collision objects (the data is stored as “user data” for the objects). This data remains attached to the object when the scene is saved. The second effect (Effects->Fabrix) is a persistent effect, which performs the dynamic simulation on the selected objects.

## **Procedure**

### **Setting the Cloth and Obstacle Parameters**

- 1.) Select the elements (nurbs or patch models) you wish to designate as fabric or collision objects.
- 2.) Choose **Effects->Fabrix\_Properties**
- 3.) The custom effect dialog box is displayed.
- 4.) Select an model in the list box (you can multi-select by shift-clicking)
- 4.) The model(s) you select are highlighted in the Softimage window(s).
- 5.) To make the selected model(s) into fabric, click the “fabric” radio button and enter the desired parameters.
- 6.) To make the selected model(s) into an obstacle, click the “obstacle” radio button and enter the “push amount” (this is the only parameter for obstacles)
- 7.) When you are satisfied with your parameters for the selected model(s), click **Set**.
- 8.) Continue with this procedure for all the models you wish to define.
- 9.) When you are satisfied with your choices for all the models, click **OK**.

Notes:

- You can set the parameters for several objects at once, but the parameter boxes will not necessarily indicate anything about the selected objects.
- To de-select an item, simply click on it.
- You can set the object-type to “neither”. This will maintain the cloth or obstacle parameters, but the objects will not take part in the simulation.

### **Running the Simulation**

1. First, create clusters on your fabric objects to lock points (any point that is part of a cluster will not be affected by the simulation, but it will affect the simulation by virtue of it’s being attached to other non-locked points).
2. Select the elements (nurbs or patch models or special controls (see below)) you wish to include in the simulation. Of the selected models, *only those objects which have previously had cloth/obstacle properties* assigned with Effect->Fabrix\_Properties will be included in the simulation.
3. Choose **Effects->Fabrix**.
4. Enter the simulation parameters desired.
5. Click **OK**.

6. To run the simulation, click the forward arrow on the time line. The status bar will display the current progress of the simulation. To stop the simulation, hold down the mouse button until playback stops.
7. To remove the simulation, delete the custom effect icon.

### **Adding Gravity Controls**

By default, Fabrix assumes an Earth gravity of  $9.8\text{ms}^{-2}$  in the negative y direction. To change this, you need to add a gravity control (this also enables you to animate the gravity). From the Dynamics Module choose **Control->Get Special Control->Gravity**. The direction of the gravity control determines the direction of the gravitational force, and the y-scaling of the control determines the strength of the force. The force is the y-scaling in  $\text{cms}^{-2}$ . The default gravity control has a y-scaling of 910, which translates to  $910\text{cms}^{-2}$ , which is approximately Earth gravity (Really, the default Softimage gravity control should have a scaling of 980 not 910, but we can't change that ☺). You can animate the gravity control any way you like.

Note: adding a gravity control removes the default gravity from the simulation. To have zero-gravity, you must include a gravity control with y-scaling of zero.

Note: you can add as many gravity controls as you wish.

### **Adding Winds and Fans**

To add a fan or wind to the simulation, simply add the wind or fan special controls, with **Control->Get Special Control->Wind** or **Control->Get Special Control->Fan**. The direction and scaling of these controls affects their properties in the same way as they do in Softimage Dynamics. A Wind has a velocity and direction. The direction is determined by the direction of the icon and the velocity is determined by the y-scaling. The velocity is the y-scaling expressed in  $\text{ms}^{-1}$ . The default Wind has a y-scaling of 4.0 which translates to a velocity of  $4.0\text{ms}^{-1}$ . (note: **not**  $4\text{cms}^{-1}$ .) 4.0 meters per second is approximately 9 mph. A Fan is simply a Wind with a limited volume of influence. The volume of influence is a cylinder whose length is the x-scaling (in Softimage units), and whose radius is the z-scaling (in Softimage units). The influence of a Fan diminishes linearly as the distance from the source. The y-scaling of a Fan determines the air velocity at the source (the velocity at any given point is determined by it's distance from the center of the icon).

Note: you can add as many Fans or Winds as you wish.

### **Including Special Controls in the Simulation**

To include a special control (Gravity, Wind or Fan) in the simulation, include it in the selection before you choose **Effects->Fabrix**.

## **The Fabrix Properties Parameters**

The Fabrix simulator is based on a so-called mass-spring network, in which points are connected to neighboring points with “springs”. The stiffness of these springs determines the properties of the fabric. Different springs are used to control Stretching, Shearing and Bending. In additions to the stiffness of these springs, there are limits placed on how much these springs can extend or compress. For example, many fabrics will stretch

relatively easily to say 5% over their natural length, but then additional stretching becomes extremely difficult.

**Stretching Stiffness:** The stiffness of the “Stretching” springs.

**Shearing Stiffness:** The stiffness of the “Shearing” springs.

**Bending Stiffness:** The stiffness of the “Bending” springs.

**Stretch Limit:** The extension/compression limit on the “Stretching” springs.

**Shear Limit:** The extension/compression limit on the “Shearing” springs.

**Bend Limit:** The extension/compression limit on the “Bending” springs.

**Damping (0..1):** Without damping, springs tend to oscillate without end.

With too much damping, the springs move slowly towards equilibrium. Use this parameter to adjust the damping.

**Bounciness (0..1):** when a fabric object collides with another fabric object or obstacle, it will rebound with a certain force. The bounciness parameter determines this force. A bounciness of zero means that there will be no bounce and the fabric will remain touching the other object after collision. A bounciness of 1 means that the fabric object will rebound like a perfectly elastic rubber ball (not very realistic for cloth).

**Friction (0..inf):** When a fabric object is in contact with another fabric or obstacle, it will have a tendency to resist “sliding” over the other object. This resistance is controlled by the Friction parameter. A value of zero means no friction and a very high value will cause the fabric to “stick” to the other object. There is no upper limit to this parameter.

**Mass (0..inf):** the mass of the fabric object in Kg. Note that a mass of 0 is not permitted, as this would result in infinite acceleration.

### **Obstacle Parameters**

Collision detection between fabric objects and obstacles is calculated between the control points on the fabric object and the obstacle surface. This may result in the surface of the fabric appearing to penetrate an obstacle between control points. One approach to this limitation is to expand the obstacle slightly before calculating collisions. This is the purpose of the push\_amount parameter.

**Push\_amount:** How much (in Softimage units) to “push” out the surface of the obstacle for collision detection.

**Note:** the obstacle model’s geometry is **not** affected by this parameter.

Note: the push\_amount acts in the direction of the normals of the model surface.

Note: collision detection is “one-way”, that is, it only acts when the obstacle’s normal is pointing toward the cloth. Be careful to ensure that the normals of your obstacles are oriented correctly (for example, if you want to confine a cloth **inside** a sphere, make sure the sphere normals are pointing **inwards**.)

## **The Fabrix Parameters**

**Start Frame:** The frame at which the simulation is to begin.

**End Frame:** The frame at which the simulation is to end.

**Release Frame:** The frame at which “clustered” points are released.

**Min. Iterations:** The minimum number of calculation iterations per frame. (see section “notes on iterations” for details).

**Max. Iterations:** The maximum number of calculation iterations per frame.

**Air drag:** the viscosity of the medium in which the fabric objects are moving (default value approximates air).

**S.I. Units/meter:** How many Softimage Units represent one meter.

**Stitch Distance:** How close two points must be (at the beginning of the simulation) to get “stitched” together by the simulation.

**Info:** Information on the effect.

**Cancel:** Cancel the effect

**OK:** accept the parameters and complete the effect.

**Obstacle Collision:** This check box, when checked, activates collision detection with obstacles.

**Ignore Cloth Collision:** This check box, when checked, disables cloth-cloth collision detection. This will speed up the simulation at the expense of allowing the cloth to pass through itself and other cloths (this may not be problem for certain simulations).

**Simple Cloth Collision:** This check box, when checked, activates a simple form of self-collision detection and collision detection with other cloth objects. This will perform a reasonably good job without excessive cpu time.

**Thorough Cloth Collision:** Activates more thorough form of self-collision detection and collision detection with other cloth objects. This option requires the most cpu time.

### **Limitations**

- Only patch and nurbs objects can be used for both fabric and collision objects.
- Cannot continue a simulation after scene is saved and re-loaded.
- Parameters cannot be animated.

- Effects->Custom->Edit Parameters has no effect. (to change parameters, delete the icon and re-apply the effect)

## Notes on iterations

The Fabrix simulator works on numerical integration. From one frame to the next, the simulator takes small steps forward in time. The number of steps it is taking is referred to as the “iterations”. For example, if it is taking steps of 1/300sec and the frame rate is 30 frames per second (i.e. each frame = 1/30sec), there are 10 “iterations” per frame.

The cloth parameters and the simulation conditions determine the number of iterations that the simulator uses. Stiffer “springs” require a smaller timestep (more iterations).

Most cloth is not very “stretchy”, which means that the stretching springs are very strong, and a larger iterations value is needed. If you don’t set the stretching springs strong enough, the cloth will look elastic-like. Similarly, most cloth resists in-plane shearing, which means strong shearing springs. In order to simplify the operation of the simulator, the stretching and shearing springs have a limit (set in Effects->Fabrix\_Properties).

Above this limit, the simulator will make the springs as strong as possible, give the iterations being used. This means that you can get “stiffer” springs simply by increasing the value of the “min iterations” parameter in Effects->Fabrix. Note that this does not apply to bending springs. The default limit on stretching and shearing springs is zero, meaning that the simulator will always try to make those springs as strong as possible for the iterations being used. In summary: **if you want less “stretchy” cloth, increase the value of “min iterations”**.

Note that the simulator calculates its own minimum iterations, which may be greater than the value specified by the user. However, the simulator will never use an iterations value that is less than the min iterations specified.

As the simulation conditions vary, the simulator will vary the iterations used to keep the simulation “stable”. If the cloth is being move around very quickly, the iterations is increased. When the cloth is not very active, the iterations are again decreased (but never below “min iterations”). “Max iterations” is an imposed ceiling on the number of iterations. Sometimes the simulator may get “stuck” and may try to increase the iterations to compensate. To prevent the number of iterations being astronomical, it will never increase the iterations above the value specified in “max iterations”.

## More Notes

- 1.) Collision nurbs objects should be uniformly parameterized (use Effect->Reparameterization).