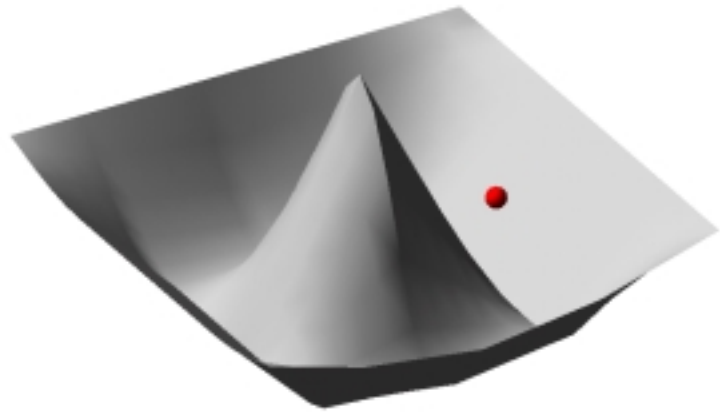


3D Contact Tips, Examples, and some Implications.

Sam McDonald, Virtual Model Shop, Inc.

International ADAMS Users Conference
Orlando, Florida
June , 2000



ABSTRACT

ADAMS is finally moving towards an era where modeling will no longer be the bottleneck it has been. Various new techniques and approaches are now (or soon will be) in place, shifting the bottleneck towards the runtime environment. This presentation will show some of the new techniques, give a few tips on 'rolling your own' contacts, and discuss the implications of how the new advances will effect the use of ADAMS in a typical production environment.

There are three major items that need to be addressed when implementing a modeling paradigm into ADAMS. These are:

1. System Tips for Ease of Use
2. Modeling Technique
3. Impact on the Integrator

Tips are given for each of these areas and a final section including a few questions and implications regarding what future developments of ADAMS may bring.

System Tips for Ease of Use:

If the techniques are desired to be used in all modeling, and are involved in setting up the elements, it is imperative that the data entry be made as simple as possible. When entering the data is too complex, data entry errors can be taken

for modeling errors inherent in the technique, leading a responsible modeler to ignore the technique and use whatever had worked in the past! So, to assist in getting the contact techniques used, the following design points were followed:

1.A. Use graphics as much as possible to define contact.

A simple example is a sphere-block contact. The block information automatically tells where the edge of the contact is, and shows when to expect contact for the sphere. A sphere-plane contact (represented graphically with a marker icon) with an edge cutoff would not be as clear.

A second example of this principle involves using a 3D spline as one surface, and a sphere as the second surface. The spline is defined in ADAMS with a simple statement that

does not have a 'natural' graphic in View. So, a short VARSUB was written to create a shell, which was then used to represent the surface. An ideal solution would be to be able to extract the data directly from the shell to represent the spline.

1.B. Use Macros to aid in contact definition.

This makes the data entry less onerous. A good example is the sphere-block contact. The data entry could consist of a GFORCE statement with a parameter list X parameters long that would need to exactly match the block parameters (including the correct reference marker) in order for the contact model to match the graphics. With the Macro, it is easy to program the parent of the block to be the j part for the GFORCE, the reference marker for the GFORCE to be the reference marker of the block, and the block parameters directly by getting just the one mouse pick on the block.

Also, as in the previous example for sphere-spline, the macro can help define the steps needed to get a graphic from a non-graphical element.

1.C. Customize menus to reduce mouse clicks.

In the standard version of view, you must go to the command navigator, make several picks, until finally the dialog box appears. You can copy the dialog box, and inset a menu pick to allow it to come up without the hassle of having to find it.

1.D. Make the modified executable and binary files default.

After saving a copy of the original aview.bin, aview.exe, and solver.exe, place the versions with the new routines in with the new names. These directories should be in the installation directories as follows:

```
aview.bin -> d:\installation directory\
aview\aview.bin
aview.exe -> d:\installation directory\
aview\aview.exe
solver.exe -> d:\installation directory\
solver\solver.exe
```

Modeling Tips

Once you have a system for surface contacts, the major difficulty comes in determining where the contact lies. The system used here applies a GFORCE on the I marker (for a sphere the I marker would be at the center of the geometry) with the GFORCE's RM at the GEOMETRY reference marker. A force and torque are then applied, with the torque arising from where the contact is located, and the appropriate torque resulting from friction force acting perpendicular to the moment arm.

2.a. Use GFOSUB with 'false bottom'.

ADAMS can only have one GFOSUB. However, that GFOSUB can call any number of other subroutines. When you write a GFOSUB, often times you will have the first parameter represent a branching parameter, to tell your GFOSUB which algorithm to execute. Use this branching parameter to point to your subroutine of specialized contacts, which can then use the second parameter as a branching id. This allows you to use any number of independently written GFOSUBS, without running into difficulties with the order of the parameters.

2.b. $N_1 = -N_2$

When defining your contact geometry section, remember that at the point of contact, the normal for one surface will be the exact opposite vector of the normal at the contact point for the second surface. If you are having trouble visualizing this, imagine two spheres contacting at a point. The contact point intuitively lies along a line connecting the centers of the two spheres, with the normals being reversed in direction from one another.

This equation will hold true for finding the maximum penetration between two surfaces, for ANY surface.

2.c. Make surface as smooth as possible. If not possible, make the path the I marker follows for persistent contact as smooth as possible.

This usually has the most impact when trying to stitch two surfaces together, as in the sphere-block example mentioned earlier.

2.d. Use Flexible/rigid over flexible/flexible.

Using the I part as the deformable body and the j part as the rigid body allows a cleaner conceptualization for the continuity of the surface than does having the I part and the j part be deformable.

Imagine a sphere-block contact and imagine the sphere (the I part) rolling over the edge of the block (the J part). When using a flexible/rigid idealization, it is easy to see that the contact is like a sphere-point contact. With a flexible/flexible idealization,

It is harder to characterize the contact as clearly. Perhaps sphere-cylinder, with a variable radius cylinder would work. Thus, using flexible/flexible would add to the complexity, and not really add much to the analysis.

2.e. Include friction as a switchable option, to allow using the crawl, walk, run approach.

2.f. Edge effects are important.

Model the edges, they are important, sometimes the most important part of the contact. Using a step or other smoothing function to switch off the normal force will make it much more difficult to accurately define the friction force at that location.

Analysis Tips

3.a. Use the crawl, walk, run approach.

Start with friction off, activate dynamic friction, and then activate 'static' friction.

3.b. If the contact is persistent (e.g. rolling not impacting), and NVH is not an issue, do not worry excessively over stiffness and damping.

In cases of persistent contact, the penetration depth will usually be less than around $1e-4$. Increasing the stiffness by an order of magnitude will decrease the penetration from $1e-4$ to $1e-5$ (depending on your chosen exponent). Usually, this difference in penetration depth is of no consequence to the model.

K and C do need to be kept from getting excessively large. If you notice the integrator laboring, with many corrector

failures, consider reducing the contact stiffness and damping.

3.c. Excessive corrector failures which drive down the integrator step size can indicate the need to modify the values for static friction 'velocity.

If the static friction velocity is too small, then the corrector can spend too much time trying to decide what kind of micro torque to apply to the rolling sphere. Relaxing the static value a little allows a little more error in the relative velocities at the contact point.

3.d. Integrator/si2, pattern = t, hmin = ?

The pattern set to true helps the Jacobian catch when the contact impacts in the middle of the corrector sequence, and when the contact surfaces separate.

HMIN is the minimum step size the integrator will use, and the default value is usually too high for using with contacts, leading to the analysis prematurely failing. Usually the integrator will reduce the step size and only need a few steps at around HMIN, and then climb back to the output step size.

Caution does need to be used while setting HMIN, because this could possibly lead to LONGER integration times if the contacts are not well formulated, or the accelerations arising from the contact are changing consistently and rapidly.

3.e. If impacting, use SENSUB to catch the contact penetration from GFOSUB to switch integration step size down for impending contact.

This can be important if you are using particularly large step sizes or if the integrator does not seem to be catching all the energy from the contact.

Implications of Current/Future MDI Development

Assumptions:

1. MDI will eventually develop a general contact feature, one that will probably use the parasolids information from AVIEW.
2. MDI will not move to a minimum coordinate set of equations.

Implications:

1. This contact will be great for Modeling systems.
2. This contact will be hell for simulation times.
3. MDII will break 7 for the first time in years.
4. There will still be a market for analytic contacts.

Reasoning:

1. This contact will be great for modeling systems.

Many that have noted most real world systems include surfaces contacting one another. There has been no consistent, easy way for the regular or casual ADAMS user to include these to date. Thus, they were reduced to finding a kinematic workaround in their model, ignoring the contacts, or simply not doing the analysis.

These analyses will now be easily modeled.

2. This contact will be hell for the integrator.

The integrators in ADAMS are very robust, compared to integrators in some other packages. However robust, because the contacts are assumed to be using a general approach and accessing the parasolids data kernel, they will likely be slow. Also slowing the analyses will be the temptation to now model every possible contact in the model. This will also slow the analysis.

3. MDII stock will rise above 7.

Because contacts have been the stumbling point for many users to make full use of ADAMS and a barrier to some, the addition of the general contacts will lead to more ADAMS sales, boosting the price.

4. There will still be a market for analytic contacts.

The speed of the generalized contact will always be somewhat slower than that of an analytic contact, as the general contact will have much more code it will need to evaluate. For some runs, especially where DOE's are desired, or multiple analytic contacts are needed (such as chains), the analytic contact will likely keep a significant speed advantage. As long as the analytic contacts are easy to model, they should still be quite useful.

In Summary

Ease of Use tips:

- 1.a. Use Graphics to define contact surfaces.

- 1.b. Use ADAMS/View macros, (or executables) to define the elements needed for the contacts.

- 1.c. Customize menus to reduce mouse clicks.

- 1.d. Move the macros and executables into the production environment.

Modeling Tips:

- 2.a. Use GFOSUB with "false bottom".

- 2.b. Normals at surfaces must satisfy $N_1 = -N_2$

- 2.c. Make surface as smooth as possible. If not possible, make path of I marker as smooth as possible.

- 2.d. Two types of contacts: 'Flexible/Rigid' and 'flexible/flexible'. Use Flexible/Rigid.

- 2.e. Include friction as option, for crawl, walk, run approach.

- 2.f. edge effects are important. Try to model based on contact point, and not 'stepping it off'.

Analysis Tips:

- 3.a. Crawl, walk, run approach. Start with friction off, activate dynamic friction, and then activate 'static' friction.

- 3.b. If the contact is persistent, and NVH is not an issue, do not worry excessively over K or C.

- 3.c. Excessive corrector failures which drive down the integrator step size can

indicate the need to modify the values
for static friction 'velocity.

3.d. Integrator/si2, pattern = t, hmin = ?

3.e. If impacting, use a SENSUB to
catch d from GFOSUB to switch
integration step size down for impending
contact.