

Multi Bodies are More Fun

R. Tyrrell

Director, Advanced Analysis Ltd

Abstract: .

There are three main parts to any overall analysis project, namely definition of the problem in terms of geometry, loads and boundary conditions, mathematical solution, and interpretation of results. This paper concentrates on some methods developed for the generation of loading data within powertrain systems, although the methods described here are applicable to many different problems.

Within a typical powertrain system there are many rotating and reciprocating parts. For a typical reciprocating engine, these include the main crankshaft / connecting rod / piston assemblies, plus multiple geared and otherwise driven (belts, chains) systems. For vehicles, the suspension system exhibits comparable behaviour. Detailed analysis of these components, plus the structures in which these parts are mounted, generally requires a knowledge of the time history of forces acting at interaction points such as bearings. For many applications, the loading data may be determined with the moving parts adequately represented by (assumed) rigid bodies.

This paper is intended to illustrate some of the ways in which Abaqus is used by the author's company as a multi-body dynamics simulation code, in the hope that it may inspire others to explore this under-used facility. The paper outlines the development of load calculation systems, and provides some tips and hints for the building of such models in Abaqus / CAE, plus pitfalls to avoid to ensure rapid converged solutions. It covers many aspects of Abaqus functionality, including substructuring, use of connectors, use of local coordinate systems, and parametric model and loading definitions. It also mentions some of the current methodology development work in progress, including some limitations of the Abaqus solver.

Keywords: Connectors, Mechanisms, Multi-Body Dynamics, Parametric, Powertrain, Substructures, User Element.

1. Introduction

This paper describes some of the work which the author's company performs using Abaqus connector element models. Some hints and work-arounds are presented, primarily based on a typical application.

2. Problem Description

A powertrain crankshaft / connecting rod / piston assembly will be used throughout this paper as an example, to demonstrate the Abaqus connector usage.

The basic mechanics of a crank / slider system are well understood. Much of the force within the system is due to inertial loads from acceleration / deceleration of the piston in the bore. These loads are then modified by the presence of gas pressure of varying magnitude above the piston.

The movement of the piston (slider) is easily predicted using the basic formula:

$$D(t) = r \left[\cos(\check{S}.t) + \left[\left(\frac{l}{r} \right)^2 - (\sin(\check{S}.t))^2 \right]^{0.5} \right],$$

where:

$D(t)$ = Piston displacement (a function of time, t)

r = crank throw (i.e., half stroke)

l = connecting rod length

ω = crank rotational speed

Suitable differentiation of this to yield acceleration, combined with masses of the various parts, gives the forces due to component accelerations. These are then supplemented by the time-varying loads from cylinder pressure, plus the simple rotating inertia loads at the crank pins.

The support of the crankshaft within the bearings at the cylinder block has been the subject of many studies by the author and many other workers over the years (e.g. Garnier, 1999). The "correct", full solution of the problem involves:

- The relative stiffness of the bearing support (crankcase) and the crankshaft.

- The interaction of those parts via oil films. Full solution includes EHD calculations
- Rotation of the crank within the cylinder block, meaning that the solution must be performed in the time domain, to take account of the time-varying relation of crank stiffness and loads to the cylinder block.

There are varying degrees of approximation possible to the solution. The most extreme, which yields surprisingly good agreement with reality, is to assume that loads from one cylinder are carried entirely by the main bearings adjacent to that cylinder only. In this extreme approximation, the crank is essentially modeled as having no bending stiffness between cylinder bays. It then becomes a series of statically determinate problems, one for each bay of the engine, the results for which may be summed. In many cases, this yields a better approximation to reality than intermediate assumptions (e.g. of a statically indeterminate crank supported on multiple bearings, each with a modeled stiffness to ground). This is because of the movement of the crank across any remote bearing, before that bearing takes up load via the oil film.

In a real engine, the crankshaft rotational speed is not constant. Despite the efforts of the flywheel, the crank accelerates when pistons are being slowed towards the end of each stroke, (kinetic energy shifts from the pistons to the crank), and decelerates when energy is needed to accelerate the pistons at the start of each new stroke. This effect is most marked in engines with light flywheels. The effect is complex to include but necessary when studying (for example) NVH aspects.

A further major complication is that a real engine has a multitude of vibrations. A major one is the torsional vibration of the crankshaft itself, which can be sufficient to locally modify the motion and forces at bearings.

Twenty years ago, the loads acting throughout the crank mechanism were calculated using varying degrees of simplifying assumption, as encoded in a Fortran routine. In more recent years, the author's company has used a relatively sophisticated MathCad sheet to perform this Task.

A major disadvantage of purely mathematical calculations (Fortran, MathCad, Matlab) is that there is no visual assessment of the system behaviour – it is necessary to concentrate when looking at program numerical or graphical output to check that the various loads are applied at the right time and right place, and that things are (mathematically) connected together and behaving as intended.

A mechanism model has the advantage of visual assembly of the system, plus visual assessment of the system behaviour. This is, of course, available via commercial mechanism software (e.g. Schittenhelm, 2006) – but at their additional costs in both lease and training.

Similar arguments to the above hold for studies of other systems, such as engine balancer arrangements or suspension systems. The visual advantages are particularly important for “one-off” analysis tasks – the author will confess that several errors have been found (and corrected!) in the Fortran, and the subsequent MathCad analyses over the years.

This paper will concentrate on modeling the simplest crankshaft approximation, i.e. operation assuming constant rotational speed, with no inclusion of torsional vibration effects. Pointers for development of the system to include more sophisticated system representation are included.

3. Input File

Extracts from an example input file appear:

```

*HEADING
Job description
*PREPRINT, CONTACT=NO ,ECHO=NO ,HISTORY=NO ,MODEL=NO
*NODE
nodelist
*ELEMENT, TYPE=MASS, ELSET=DUMMASS
    1,      1
    etc
*ELEMENT, TYPE=ROTARYI, ELSET=DUMROT
    501,    1,
Dummy, small masses and rotary inertia values should be put onto nodes to prevent singularity errors
*MASS, ELSET=DUMMASS
0.0001E-03,
*ROTARY INERTIA, ELSET=DUMROT
1.0E-12,
*ELEMENT,TYPE=MASS, ELSET=PISMASS
    18,    18
    etc
*MASS, ELSET=PISMASS
value
*MASS, ELSET=WEB1
    etc
*ELEMENT, TYPE=CONN3D2, ELSET=CBMS
** Crank
    31,    3,    4
    etc
** rods
    etc
** ccase
    etc
**
*CONNECTOR SECTION, ELSET=CBMS
Beam, See text
*ORIENTATION, NAME=CRNKAXIS
1.0, 0.0, 0.0, 0.0, 1.0, 0.0
1, 0.0
*ELEMENT, TYPE=CONN3D2, ELSET=MAINS
    201, 31, 30
    etc
*ELEMENT, TYPE=CONN3D2, ELSET=THRUST
    202, 4, 25
*CONNECTOR SECTION, ELSET=MAINS
CARTESIAN, CARDAN
CRNKAXIS,
*CONNECTOR MOTION
MAINS, 2
MAINS, 3

```

```

*CONNECTOR SECTION, ELSET=THRUST
CARTESIAN, ROTATION
CRNKAXIS,
*CONNECTOR MOTION
THRUST, 1
THRUST, 2
THRUST, 3
*****
** Pistons slide so that 1,3 are as master, 2 (Y) is free
*ELEMENT, TYPE=CONN3D2, ELSET=PISTS
    221, 26, 18,

etc
*ORIENTATION, NAME=CYLAXIS
0.0, 1.0, 0.0, 1.0, 0.0, 0.0
1, 0.0
*CONNECTOR SECTION, ELSET=PISTS
CARTESIAN, ROTATION
CRNKAXIS,
*CONNECTOR MOTION
PISTS, 1
PISTS, 3
PISTS, 5
PISTS, 6
*ELSET, ELSET=CRANK

etc
*PARAMETER
PI=3.141592
rpm=9000.0
** radians / sec
angvel=2.0*PI*rpm/60.0
timelrev= 60.0 / rpm
timelcyc= timelrev * 2.0
timedeg = timelrev / 360.0
** This 2-cylinder model is built with pistons at BDC, hence cyl 1 fires at 180 deg rot etc. See text
fire1 = (180.0-720.0) * timedeg
fire2 = (540.0-720.0) * timedeg
*AMPLITUDE, NAME=CPR1, SHIFTX=<fire1>, DEFINITION=EQUALLY SPACED, SHIFTY=0.0,
SCALEY=0.83, FIXED INTERVAL=<timedeg>, BEGIN=0.0, INPUT=generic_cp_petr.asc
*AMPLITUDE, NAME=CPR2, SHIFTX=<fire2>, DEFINITION=EQUALLY SPACED, SHIFTY=0.0,
SCALEY=0.83, FIXED INTERVAL=<timedeg>, BEGIN=0.0, INPUT=generic_cp_petr.asc
*constraint controls, print=yes
*****
** Run 1 cycle to settle
*STEP, NLGEOM=YES, INC=730
*DYNAMIC, DIRECT, ALPHA=-0.25
<timedeg>, <timelcyc>
*BOUNDARY, TYPE=VELOCITY, OP=NEW
1, 4, 4, <angvel>
*BOUNDARY, OP=NEW
P5      , 1, 6
** Pistons:
*CLOAD, AMPLITUDE=CPR1
18, 2, -103.7

etc
** Cylinder heads (to ensure zero reactions at ground)
*CLOAD, AMPLITUDE=CPR1
27, 2, 103.7
*OUTPUT, FIELD, FREQ=10

```

```

*NODE OUTPUT
U,
*END STEP
*****
*STEP, NLGEOM=YES, INC=730
*DYNAMIC, DIRECT, ALPHA=-0.25, INITIAL=NO
<timedeg>, <time1cyc>
****
Entries as for step 1
*OUTPUT, FIELD, FREQ=10
*NODE OUTPUT
U,
*OUTPUT, HISTORY, FREQ=5
*ELEMENT OUTPUT, ELSET=BRGS
CTF,
*NODE OUTPUT, NSET=FLNG
RF,
Repeats for other interesting points / connectors
*END STEP

```

This file will be used in the following sections to illustrate various aspects of the model. It is necessary to perform some of the file generation manually, as Abaqus / CAE is not able to handle all of the features used.

A general comment is made that the author believes it is essential for the analyst to fully understand what is happening within all parts of the model. To this end, some of the more advanced connector types are not used. For example, “SLIDER” type connectors might be used for the pistons within the bores, but it is perhaps better (especially when starting this type of project) to generate the equivalent of the slider mechanism using the basic connector types (CARTESIAN and ROTATION).

4. Model generation

4.1 Geometry Build

Two typical example, “bottom level” connector models are shown in Figure 1. The first is for the main running gear of a conventional four cylinder engine, the second for a horizontally opposed twin. In these models, an assumed rigid crankcase is also modeled, with CoG at the correct location. This “crankcase” model is then restrained to ground via connectors (which may be assigned mount stiffness values) at the engine mount points. Such a model is useful to assess basic excitations at the mount locations. The crankcase model is highlighted.

Generation of these models within Abaqus / CAE is rather different from generation of conventional models. There need be no separate “Parts”, for example: all model building is done in the assembly module.

Some general comments include:

- It is very useful to build up a template python script to build the model. A few edits of basic points within that script then permit very rapid model build

- Duplicate points are needed in many locations. Care is needed to ensure the correct reference point is selected when making the various wires for connector element application.

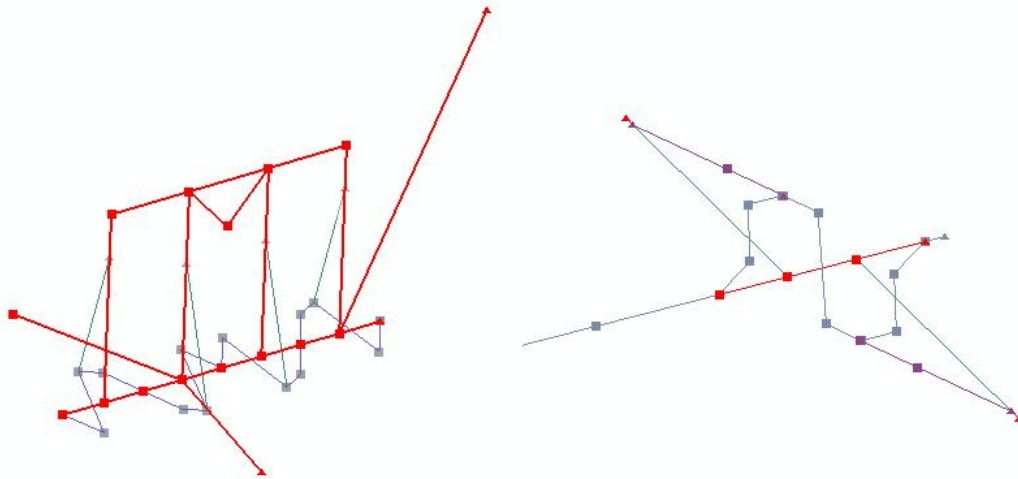


Figure 1: Example mechanism models

4.2 Connector and Joint Types

The main “structural” parts of models as developed are often built using BEAM type connectors. This is the one exception to the general rule that basic connector types are employed.

Connector models are extremely fussy about over-constraints. Unlike conventional models, any overconstraint is liable to result in a failure to get the model to run. This means that all the motions and restrictions on motions of individual parts must be fully understood by the analyst, and each possible degree of freedom only restrained once. The overall message from this section may be summarized “The analyst must think, and clearly understand exactly what is meant to be happening at each modeled connection”.

The simple model assumption is that there is no bending moment transfer from one bay of the engine to the next: only rotational loads about the crank axis should be transferred. Thus, each main bearing is represented by two separate nodes for the crankshaft, and one for the crankcase. As with all modeling, there are many possible ways to model the desired system. Figure 2 presents possible ways of connection of the required degrees of freedom at the main bearings for a typical crankshaft. In this Figure, each ellipse represents an additional connector element (as well as the BEAM type connectors to represent the crank and crankcase itself).

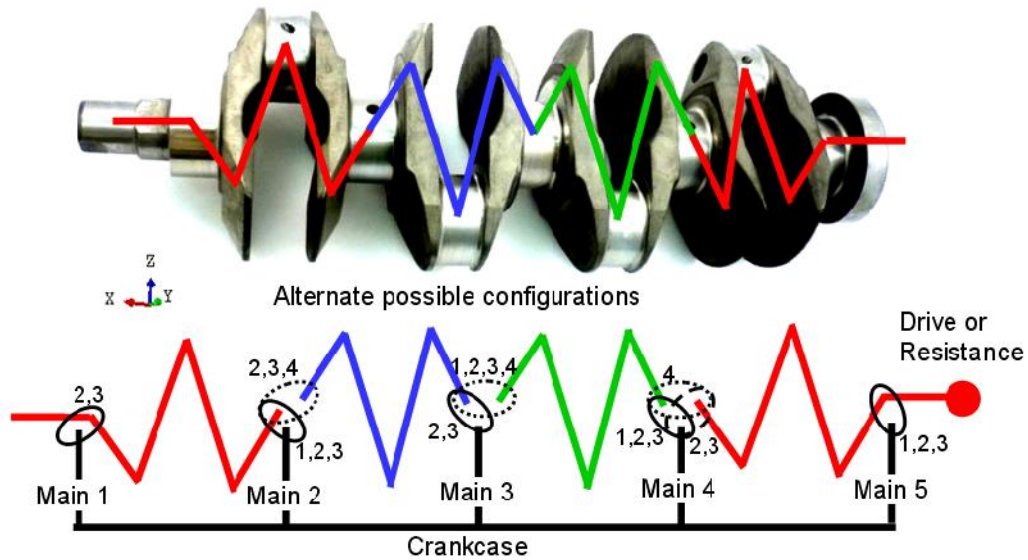


Figure 2: Schematic representation of 4 cylinder crankshaft

In order to meet the requirements of the approximation that the crank transmits no bending moment between sections, only the crank-axis component of rotation is connected between sections.

In order to prevent unwanted axial motion of each represented crank bay or major crank section, the main bearing at one end of the section is connected to the remaining structure in all 3 translatory DoF, whereas that at the other is only connected in the vertical and transverse directions. Additional restraints in the axial direction will result in overconstraint.

The various DoF connections at each bearing are perhaps most easily achieved using ***CONNECTOR MOTION** specifications. Thus, for example, the bay: bay connections of crank parts as illustrated in Figure 2 may be achieved using:

```
*ELEMENT, TYPE=CONN3D2, ELSET=MAINS
      201, 31, 30
```

etc

```
*CONNECTOR SECTION, ELSET=MAINS
CARTESIAN, ROTATION
CRNKAXIS,
*CONNECTOR MOTION
MAINS, 2
MAINS, 3
```

Whereas the connection between one crank part and the crankcase may be achieved using:

```
*ELEMENT, TYPE=CONN3D2, ELSET=THRUST
      202, 4, 25
```

...

```
*CONNECTOR SECTION, ELSET=THRUST
```



```

CARTESIAN, ROTATION
CRNKAXIS,
*CONNECTOR MOTION
THRUST, 1
THRUST, 2
THRUST, 3

```

Note the use of local axis set “CRNKAXIS” here. This will be further discussed in relation to output, below.

The connectors between the pistons and the bores follow similar rules.

A useful enhancement of the model is to include estimated friction and / or damping coefficients at the bearings and at the piston / bore representation. An example of such a specification would be:

```

*CONNECTOR BEHAVIOR, NAME=BP1ST
*CONNECTOR DAMPING, COMPONENT=2
0.1,

```

4.3 Loading

Having accepted that some manual editing of input decks will be necessary for these models, it becomes highly desirable to introduce the use of parameters for loading definition. Examples of their usage are presented in the input file listing above.

The application of cylinder pressure illustrates two useful Abaqus features: the SHIFT option in amplitude tables, and the SCALE factor on CLOAD cards using those tables.

Use of the SHIFT factor also demands some clear thinking. The file “generic_cp_petr.asc” referred to in the file listing represents a typical cylinder pressure trace as illustrated in Figure3. Note that the data has been repeated, such that two complete engine cycles (4 revolutions) are represented.

The SHIFT value is effectively *subtracted* from the time specified in the table, i.e. it represents a delay. Thus, a SHIFT value (delay) of +5 would mean that, at step time 0, Abaqus looks for the value at a time of [0 - 5] = -5 from the table, at which there is no data. Extension of the table to two cycles means that we DO have data for 0 -> 1440 deg in table so we may “pick up” the required value from (near) the second half of table. We require the data for a table time of -5, which is the same data as at a time of [720-5]. As Abaqus negates this, we may specify a shift value of [shift-720]. E.g. [5-720] will mean Abaqus uses the table time of +715, which is what we want.

The SCALE parameters are more easily understood: the SCALEY on the amplitude card modifies the amplitude in the generic diagram to give the correct peak cylinder pressure, and the entries on the CLOAD cards are used to modify the cylinder pressure from bar (10^5N/mm^2) to the required force in Newtons, taking account of the piston top area.

In this simple model, the crank speed variation with angle is ignored. Instead, the crank is rotated (at the drive end) at the required steady angular velocity <angvel>. A useful check on power output is obtained by integrating the resultant torque reaction at this node.

More advanced versions of the model allow for the crankshaft angular velocity variation. This is achieved by the introduction of a torsional spring and damper, modeled between the flywheel and the (constant velocity) model drive point. Experience has built up an ability to specify suitable spring and damper rates, so yielding angular speed variations very close to those measured or predicted by more sophisticated, dedicated software.

4.4 Output

One huge advantage of connector elements is the ability to output results (e.g. forces at bearings, as output using the CTF parameter) in local axes. These are output as History values, enabling ready access.

The general rule is that forces are output according to the axis set active at the first node in the connector definition. The force reported is effectively “the force acting *onto* the first node from the connector”. Thus, for example, the same model may be used to output forces acting on the main bearings (in global coordinates) and also the forces resolved into the crankshaft local axis set. For the main bearing version, the connectors at those bearings are defined with the crankcase node specified first, and global axes. For the same values, but resolved into forces suitable for analysis of a crankshaft, those elements may be defined using a “CRNKAXIS” set and specification of the connectors with the crankshaft axis first.

5. Solution

The first thing to ensure with all models which include connector elements is that there are no reported overconstraints or singularities. These may be very difficult to track down, hence the reliance of the author’s company on basic connector types and clear thinking throughout. A major advantage of these models is that they run – when they are set up correctly – in embarrassingly short times.

The second aspect is that the solution must appear smooth. Some interesting aspects of the connector elements have come to light when trying to obtain a solution avoiding erroneous high frequency oscillations. These include the fact that specification of a CONNECTOR ELASTICITY as RIGID does not actually mean rigid, but rather a specification of high stiffness (1.0E06 in these models, as there are no adjacent conventional elements). This is insufficiently “rigid” for many applications used by the author, thus explicit specification of component rigidity is preferred, often with values as high as 1.0E10 required to obtain stable solutions. The use of the BEAM type connector is often replaced with an explicit definition, e.g.:

```
*Connector Behavior, name=CBMS
*Connector Elasticity, component=1
1e+08,
*Connector Elasticity, component=2
1e+08,
*Connector Elasticity, component=3
1e+08,
*Connector Elasticity, component=4
1e+08,
*Connector Elasticity, component=5
1e+08,
```

```
*Connector Elasticity, component=6  
1e+08,
```

It is usually necessary to include a preliminary “settling” analysis step prior to the real step from which results are output. This is far easier than trying to specify all the otherwise necessary initial values of local velocity and displacement. The second (“useful”) step is then run using control of the form:

```
*STEP, NLGEOM=YES, INC=730  
*DYNAMIC, DIRECT, ALPHA=-0.25, INITIAL=NO  
<timedeg>, <time1cyc>
```

Wherever possible, checks are built into the models. These include simple things like watching animations of the time history, to ensure everything is moving as expected, through checks on overall reaction forces, including the power output check as previously mentioned.

6. Results

Results from these models compare well against those obtained by other means. Figure3, for example, presents a comparison of predicted loads at a main bearing against those derived by the sophisticated software in use by a major engine bearing supplier.

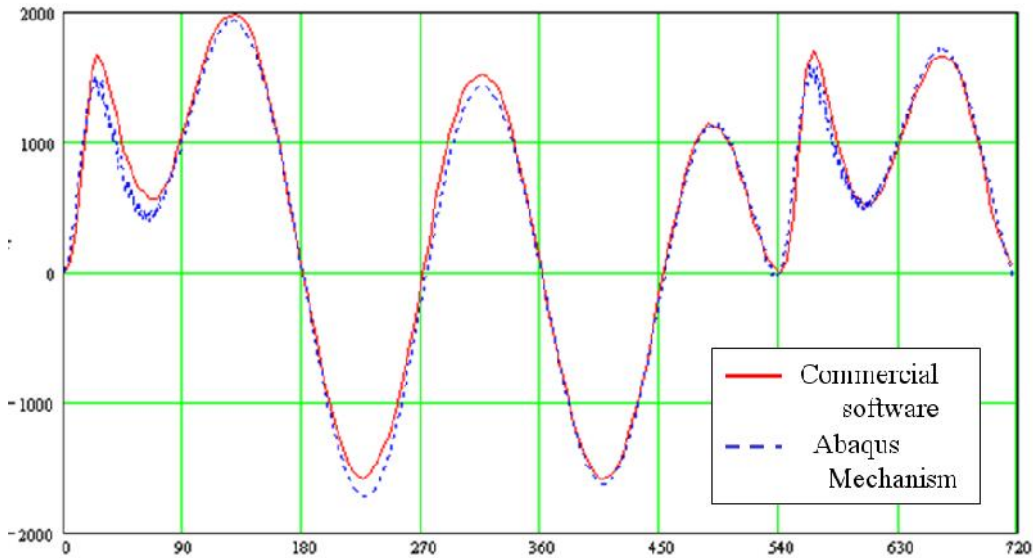


Figure 3: Mechanism code vs. commercial analysis

7. Parametric Models

A high proportion of the mechanism models used by the author relate to either 3- or 4-cylinder engines, with conventional layout. Thus, a completely parametric version of the simple mechanism model has been created. Usage of this model version does not involve Abaqus / CAE in any way: instead, a few geometric parameters are manually entered into the top of the input deck, and the calculations within that deck sort out all the necessary geometry and model build. Input parameters thus include such features as bore, stroke, cylinder spacing, piston pin offset and more. The use of parameter values is surprisingly flexible, enabling all normal input file values to be parametric, e.g.:

```
*NODE, NSET=CPINS
    11, <CPlx>, <CPly>, <CPlz>
*TRANSFORM, NSET=PISTS
    1, 0, 0, <Xaxis>, <Yaxis>, <Zaxis>
```

8. More Advanced Developments

A relatively simple extension of the model type is useful when studying suspension systems. Suspension at full compression may impart very high loads into a vehicle structure. A single connector may be used to model the combined effect of the spring and bump stop, for example:

```
*Connector Behavior, name=Spring_BumpStop
*Connector Elasticity, nonlinear, component=1
    1000., 10.
    2500., 25.
    5000., 50.
    5600., 55.
    20000., 58.
    200000., 60.
```

Note here the rapid increase in load with compressions of greater than 55mm. This represents the effect of the bump stop at the end of the system travel. One problem with this analysis is that it appears impossible (at present) to use Abaqus/CAE to specify linear connector properties for one connector component, and non-linear ones for another. This thus also demands some manual edit of the input deck.

An obvious extension of the whole system is to include actual component stiffness values. This entails the replacement of selected connector elements with substructure representations. Such work has been reported by others (e.g. Stephenson, 2010), and is used by the author on studies such as the torsional vibration of the crankshaft.

8.1 Gears

Thus far, this paper has presented a rather glowing report on the use of connector elements. However, there are some areas where the author has so far failed to overcome limitations imposed by Abaqus software.

One example is in the modeling of gears. Most modern motorcycle engines have a geared primary drive: that is, the crankshaft has a direct gear connection to the gearbox input. Likewise, balancer

shafts are often gear driven. It is thus desirable to be able to model gear pairs within these mechanism models.

The simplistic representation of a gear pair as a constraint on angular rotations is easily implemented using a scaled equation relating the angle of one gear to another. However, such a representation misses the vital additional loads imparted on the gears. For a simple straight-cut spur gear pair, this is a separating force between the gears, caused by the pressure angle of the gear teeth. This separating force is a function of the torque being transmitted through the gear pair, and is (usually) independent of the direction of rotation. More advanced gear types (helical, hypoid) impose additional forces, including axial loads.

The author has developed a USER ELEMENT to represent such gear pairs. This code includes the necessary mathematics to generate all the forces acting on the gears due to torque transmission, and it works extremely well when built into mechanism models.

That is, it works extremely well in 95% of cases. In the remaining 5%, a major problem occurs when the total rotation of either gear exceeds 2π . Abaqus / standard only “understands” rotations which lie in the range 0 to $+2\pi$. This means that angles outside this range are internally adjusted by $\pm 2\pi$.

Checks are built into the software to eliminate the basic problem of the 2π limitation within Abaqus / standard, namely a tracker is built in to correctly modify the angle used in calculations to the total angular displacement. However, the iteration algorithm in use within Abaqus occasionally returns a value 2π greater or less than expected, and - to date - no coding method has been identified to overcome the resultant instability.

Work is ongoing to overcome this rather major current limitation to mechanism (and other) model usage.

9. Conclusion

Mechanism models represent a major application area for the Abaqus software. However, their use demands clear, logical thinking on the part of the analyst. There are many subtle limitations as to their use, most of which may be overcome by the adoption of a logical, self-testing approach.

And the animations are fun!

10. References

Do not cite references by number. Use the name of the lead author, followed by a comma and the date of publication, enclosed in parentheses; for example, “(Crisfield, 1982).”

1. Liang Chen, Xigeng Song, Dongxin Xue, Zhangjie Ming “Elastohydrodynamic Lubrication and Asperity Contact Simulation of Engine Main Bearing with Flexible Rotating Crankshaft and Flexible Engine Block”, ADVANCED TRIBOLOGY 2010, Part 3, VII.,

2010 SIMULIA UK Regional User Meeting

2. H. Dannbauer, C. Gaier and M. Steinbatz “A Statistical Measure Of The Non-Proportionality Of Stresses – Investigations And Applications”
www.femfat.com/ftp/papers/2004/02_7th_Biaxial.pdf
3. Ebrat, O; Mourelatos , Z; Hu, K “An Elastohydrodynamic Coupling of a Rotating Crankshaft and a Flexible Engine Block”, J. Tribol. April 2004 Volume 126, Issue 2
4. Garnier T.;Student R.;Bonneau D.;Grente C. “Three-dimensional EHD behavior of the engine block/crankshaft assembly for a four cylinder inline automotive engine” Journal of Tribology, 1999, vol.121
5. Schittenhelm, M; Mulski, S “SIMPACK Engine”, www.engine-expo.com/06engine_conf/pres/.../mulski.pdf
6. Stephenson, M “The Up Side of Engine Downsizing”, Abaqus Insights, Vol1 2010
7. Xingguo, M; Xiaomei, Y; Bangchun W; “Multi-body Dynamics Simulation on Flexible Crankshaft System”, 12th IFToMM World Congress, Besançon (France), June18-21, 2007