ADORE Update Version 5.50

Release Date: July 9, 2010

ADORE 5.50 is an enhancement to earlier version 5.41. The following specific enhancements have been incorporated:

Code Enhancements

Predictor-Corrector Integration Algorithm

The predictor-corrector algorithm, which was first introduced in very early version of ADORE and taken out during code conversion to FORTRAN-90, has been incorporated in ADORE once again. The associated codes for this algorithm have been basically rewritten to meet the FORTRAN-90 and FORTRAN-95 standards. Although these procedures may have limited utility for most rolling bearings, they do provide a means to perform a relatively fast integration in absence of large discontinuities and cyclic variations.

Rolling Element to Rolling Element Interaction

In all ADORE version thus far there was no provision of rolling element to rolling element interaction in a cageless bearing. In version 5.50 a new procedure is added to model ball-to-ball contact in cageless ball bearings. The additional inputs required for such modeling is the ball-to-ball traction or friction. Since the slip velocity between the interacting balls is generally very high, a constant friction coefficient may suffice in many instances. Just for the purpose of maintaining some generality a hypothetical traction curve is input to model the frictional interaction. This, of course, resulted in new inputs in input data files as discussed below.

It is anticipated that in future version of ADORE these procedures will be further generalized to permit interaction between all types of rolling elements.

Cage Geometrical imperfections

When cage pockets are non uniformly distributed the initial angular position of rolling elements was set in the user programmable routine in earlier ADORE versions. In version 5.50 the rolling elements are automatically set in the center of cage pocket upon startup.

Spherical Cage Pockets for Ball Bearings

In ball riding cages with spherical pockets, the pocket holes on the cage OD and ID now have fictitious conical surfaces. This prevents excessive transient cage motion which was resulting from no cage contact when the minimum geometrical interaction point is in the pocket hole area. Such a simulated contact is tagged in the output; a flag "<<??>>" is printed next to the pocket interaction solution. This serves as a warning when the ball cage interactions are not within the permitted area of cage pockets. Thus, the purpose of this enhancement is simply to eliminate any numerical problems during transient cage motion, and in steady-state all cage pocket interaction outputs should be free of the above flags. In the events these flags are seen for some pockets and the contact angle ϕ appears to be constant at a relatively large value then the cage pocket geometry must be modified such that the cage pocket contacts are within the permissible pocket surface. It is anticipated that this enhancement will be useful in development of improved designs for spherical cage pockets.

Tapered Roller Bearings

The default initial race position was modified for improved numerical convergence while obtaining an equilibrium solution with tapered roller bearings, particularly with bearings which have a guide flange on the outer race. In addition, the reference race position is now included in the print output. This is useful when running ADORE with prescribed race displacement. The reference race displacement is simply the displacement between the races for a contact with no load. Thus this value helps in determining the input displacement in a loaded configuration. As always, the purpose of prescribed displacements is to provide assistance in obtaining a quasi-static solution when the iterative procedure for race equilibrium does not seem to converge. With prescribed race displacements the race equilibrium equations are eliminated and the resulting loads from the prescribed displacements are shown in the print output. Thus several solutions with varying displacements may be obtained and convergence to the actual applied loads may be accomplished manually.

Improved Diagnostics

In a continued effort to improve identification of numerical problems, divergence of equilibrium equations due to marginally singular Jacobian matrices is more precisely checked and the associated error messages are now printed before the execution blows out due to system generated messages such as zero divide etc.

Also based on user interactions some additional checks of bearing input data are incorporated in the data input routine. This eliminates unnecessary effort spent in trying to resolve problems which are simply caused by unrealistic input data.

Code Corrections

A few corrections to code, identified during both the development of above enhancements and user interactions have been applied in the current version.

1. While accessing the material data base the shaft temperature in the material array in corrected in the input routine Adra1 for a call to materials data base subprogram Adra7.

2. The radius of clearance circle in cage orbit plot is corrected when the plot axes have an axis multiplier ($x10^{x}$).

3.Computation of cage pocket contact angle, ϕ, for spherical cage pockets, is corrected in cage interaction routine Adre1.

4. A transformation matrix error, again influencing spherical cage pockets in ball bearings, in subroutine Adre9 is corrected.

5. Scale factor while printing roller load distribution at the outer race contact is corrected in the print output subroutine Adra3.

ADORE User Manual

The user manual is updated to include the additional input data required to model the frictional interaction at the newly introduced ball-to-ball contact.

ADORE Input, Plot and Animation Facilities

The Java based input facility, adrInput is updated to include the new input data required for the ball-to-ball interaction. In the input data this simply resulted in an additional flag on record

10.0 to turn on rolling element to rolling element interaction and the associated frictional data on a new data record 10.5.3. Thus data files generated earlier for bearings with cage can be made compatible with ADORE 5.50 by simply appending a variable with a value of zero on record 10.0.

The plot facility adrPlot is enhanced to add an additional plot in the rolling element section to display rolling element to rolling element interaction.

There are no changes to the graphic animation facility, AGORE.

Test Cases

The normal ball and cylindrical roller bearing test cases are included with the program files; in addition to the input data, print output files and all plot data sets are included in the program media. These examples must be run and checked after installation of the program. All outputs, at least at step 0, must match against the supplied output.

While comparing the results with those produced by earlier versions some differences in the transient solutions and time step sizes may be observed. These difference are primarily due to code corrections and enhancements outlined above.

Program File Contents:

As usual program updates are distributed on a CD in normal data format. The files may be easily extracted from this disk on any computer system and then transferred to appropriate system for which ADORE is licensed for.

The media contains the following three subdirectories:

Disk1

Update550.pdf:

A pdf file containing notes of the latest updates (this file).

adoreInput.txt:

A text file containing details of ADORE input data.

adoreManual.pdf:

ADORE user's manual containing detailed instructions for program installation and use.

Ball:

Subdirectory containing ball bearing test case

Roller:

Subdirectory containing roller bearing test case

AdrxExamples

Subdirectory containing few of the user program able examples via subroutine ADRX1.

Disk2

***.f files:**

ADORE FORTRAN-90 source files

Disk3

setup.bat:

Setup batch file to compile adrInput, adrPlot and AGORE on Windows system.

adrInput.bat:

Batch file to execute adrInput.

adrPlot.bat:

Batch file to execute adrPlot.

agore.bat:

Batch file to execute the graphics animation facility, AGORE.

Java:

Subdirectory containing all Java source.

Program Installation

On the Windows system, if the Microsoft Developer Studio is used to create the executable, the following suggested procedure may be helpful.

1. Start Microsoft Developer Studio and select the File option to create a new project.

2. For type of application, select “Console Application” and name the application as adore55 or other desired name.

3. Once the project space is created, use the inert option to add source files. After navigating to the appropriate source directory, first add the file m_parameters.f only. In the second step add all the m_*.f module files. In the final step all the other source file. The file to be added is simply selected by a mouse click on the file in the selection widow.

4. Now use the Build option to create the executable.

Java facilities adrInput, adrPlot and Agore

Edit the setup.bat file in Disk3 subdirectory to correct the paths to all source files and the Java Development Kit. Execute the updated setup file to compile and install these facilities.

The setup files for the three applications may then be edited to update the paths and installed in appropriate directory compatible with the environmental variables which provide access to all executables.

Contact Information

In the event of any questions and/or technical support please contact:

Dr. Pradeep K. Gupta
PKG Inc.
117 Southbury Road
Clifton Park, NY 12065-7714 USA
Phone: 518-383-1167
Fax: 518-371-3833
Web: www.PradeepKGuptaInc.com
Email: guptap@PradeepKGuptaInc.com