Compilation: see readme.txt

StressRefine Architecture

Stressrefine consists of several executables. The user interface, stressRefine.exe, is the "conductor". It is supported by a translator, bdfTranslate.exe, and an analysis engine, SRwithMkl.exe. stressRefine.exe opens an Fea file such as a Nastran file. It calls bdfTranslate.exe to convert to the internal stressRefine format. To use an input file other than Nastran format requires a different translator. The analysis executable is then called to adaptively solve the problem. Results are post-processed using the free executable cgx from Calculiix (`http://www.dhondt.de/)`. If it is necessary to translate Nastran binary output, the free executable pyNastran is used (https://github.com/SteveDoyle2/pyNastran).

The source for the Ui, stressRefine, the translator, bdfTranslate, and the analysis engine, SRwithMkl, have all been provided in the form of Visual Studio 13 projects. The UI is written in C#, everything else is in C++.

StressRefine Library

The stressRefine library provides a lot of support routines for heirarchical basis functions, mapping, etc. A Visual Studio project has been provided to build the library as well. The analysis engine makes use of the stressRefine library. You can optionally download the compiled Windows x64 version of the static library and link it to the analysis engine executable.

If you'd like to use the library to do p-adaptivity in your own code, then the UI, translator, and analysis engine are all not necessary. Source from the analysis engine may provide useful sample code, for example for controlling adaptive analysis.

Using the Source

Currently the source is available is Window Visual Studio projects. I am working on a Linux version. You are also free to create a version compatible with Linux or any other operating system,

1. Executable, Complete Build:
   a. The library SrlibSimple needs to be compiled
   b. The executables bdftranslate, srwithmkl and Srui need to be built. This will create the executables bdftranslate.exe, srwithmkl.exe, and stressRefine.exe. These executables should all be placed in the same folder
   c. Run the program by starting stressRefine.exe. It will look for Nastran input files as explained in the user's guide. The samplefiles may be downloaded from then stressRefineShare folder.
   d. The UI will require the opensource executable cgx.exe to plot stress fringes. This is available in the stressRefineshare folder.
2. Executable, analysis engine only
   a. The library SrlibSimple needs to be compiled

   b. The executable srwithmkl needs to be built. Run this executable directly to solve input files in stressRefine format. Samples of these are in srwithmklsamplefiles.zip
   c. To run srwithmkl.exe, a file modelfilename.txt must exist in the same folder as srwithmkl.exe. This file contains a single line, which is the full path of the input folder
   d. The input folder must contain a single file “filename”.msh. For example,I f the path to the folder is mydrive\myfolder\myinputfile, then the input file in the folder is myinputfile.msh. All output from srwithmkl.exe is directed to this folder. A summary is in report.txt. Fringes for plotting in cgx.exe are in myinputfile.frd. Output for plotting with a postprocessor in ascii Nastran format is in myinputfile.f06. Detailed text output is in myinputfile.out.
   e. an optional settings file myinputfile.srs (for “stress refine settings”) can also be included in the input file. These settings are explained in the file stressRefineSettings.txt

3. Library only
   a. The library SrlibSimple needs to be compiled
   b. link the library to your own executable, which must provide its own equation solver. Feel free to use source from SRwithmkl as desired in your own executable, for example to use for the adaptivity loop.

Note on Intel MKL pardiso solver (Copyright (c) 2018 Intel Corporation https://software.intel.com/en-us/license/intel-simplified-software-license):

The intel pardiso solver is used in the executable srwithmkl.exe. This is available for free download from Intel at https://software.intel.com/en-us/mkl/choose-download. While free, this library is proprietary to Intel. If you redistribute srwithmkl the Intel copyright must be included along with a link to the intel simplified software license.

I am working on replacing pardiso with an open-source sparse solver. You are also free to do so.