refine. Analysis Interface Module (AIM)

Marshall Galbriath
MIT

May 22, 2024

0.1 Introduction

0.1.1 refine AIM Overview

A module in the Computational Aircraft Prototype Syntheses (CAPS) has been developed to interact with the unstructured mesh adaptation software `refine`.

refine is designed primarily to interact with Fun3D, but provides more a generic interface as well. As of Fun3D 14, a set of scripts are provided to drive Fun3D solutions with refine, which are outlined in the Fun3D user manual. Using this AIM in lieu of the Fun3D scripts enables data transfer as well as parametric shape sensitivities in the CAPS framework.

An outline of the AIM's inputs and outputs are provided in AIM Inputs and AIM Outputs, respectively.

The refine AIM can automatically execute ref, with details provided in AIM Execution. The specific executable can be changed with the 'ref' input string.

0.2 AIM Inputs

The following list outlines the refine inputs along with their default value available through the AIM interface.

- **ref = "ref"**
  refine executable
- **Passes = 30**
  Number of refine internal adaptation iterations
- **Fun3D = false**
  Use refine "loop" operation, i.e. in conjunction with Fun3D solver
- **Mesh = NULL**
  An Area_Mesh or Volume_Mesh link for mesh adaptation
- **Mesh_Format = NULL**
  Optional list of string mesh formats to generate meshes not linked to analysis.
  Available format names include: "exodus", "fast", "libMeshb", "stl", "bstl", "su2", "tecplot", "ugrid", "vtk", and "bvtk".
  where the "b" prefix indicates binary version.
- **Complexity = NULL**
  Must be specified in combination with a ScalarFieldFile or HessianFieldFile.
  Cannot be specified in combination with MetricFieldFile.
  Complexity is approximately half the target number of vertices.
- **ScalarFieldFile = NULL**
  Scalar field file for constructing the multi-scale metric.
  Only one of ScalarFieldFile, HessianFieldFile, or MetricFieldFile may be specified
- **HessianFieldFile = NULL**
  Hessian field file for constructing the multi-scale metric.
  Only one of ScalarFieldFile, HessianFieldFile, or MetricFieldFile may be specified
- **MetricFieldFile = NULL**
  Metric field file.
  Only one of ScalarFieldFile, HessianFieldFile, or MetricFieldFile may be specified
- **InterpolateFile = NULL**
  libMeshb .sol or .solb file to interpolate

0.3 AIM Execution

If auto execution is enabled when creating an refine AIM, the AIM will execute refine just-in-time with the command line:

```
ref $(cat refInput.txt) > refOutput.txt
```

where preAnalysis generated the file "refInput.txt" which contains commandline arguments for ref.

The refine analysis directory is assumed to contain a metric.meshb file. This file will be generated automatically with preAnalysis using ScalarFieldFile or HessianFieldFile inputs, or can be generated manually via system calls to refine and setting MetricFieldFile.

The analysis can be also be explicitly executed with caps_execute in the C-API or via Analysis.runAnalysis in the pyCAPS API.

Calling preAnalysis and postAnalysis is NOT allowed when auto execution is enabled.

Auto execution can also be disabled when creating an refine AIM object. In this mode, caps_execute and Analysis.↩ runAnalysis can be used to run the analysis, or refine can be executed by calling preAnalysis, system call, and posAnalysis as demonstrated below with a pyCAPS example:

```
print ("\n\preAnalysis......")
refine.preAnalysis()
print ("\n\nRunning......")
refine.system("ref $(cat refInput.txt) > refOutput.txt"); # Run via system call
print ("\n\postAnalysis......")
refine.postAnalysis()
```

0.4 AIM Outputs

List of available outputs from the refine AIM

- **Mesh**
  The output Area_Mesh or Volume_Mesh for a link
- **InterpolateFile**
  Interpolated solb file if input InterpolateFile is set. NULL otherwise.
- **xyz**
  Grid coordinates. Useful for constructing scalar, hessian, or metric fields