Elmer SIF file

Content of ElmerSolver Input File explained

ElmerTeam

CSC, 11/2020

Basic Structure of SIF file

- Basic idea: **Sections** + **Keywords**
- Each section starts with **SectionName** and ends with with "**End**"
  - Alternative for one keyword
    SectionName :: Keyword
- In each section we may have an arbitrary number of keywords
- Keywords are of type
  - **Real** : real valued number
  - **Integer** : integer number
  - **Logical** : True or False
  - **String**: not case-sensitive text
  - **File**: case-sensitive text

- Sections are
  - **Header**
  - **Constants**
  - **Simulation**
  - **Solver i**
  - **Body i**
  - **Equation i**
  - **Body Force i**
  - **Material i**
  - **Initial Condition i**
  - **Boundary Condition i**
  - **Run Control**
  - **Component i**
- Not all sections are needed

Example of minimal sif file

```
! Minimal sif file example
Check Keywords "Warn"

Header :: Mesh DB "." "square"

Simulation
  Max Output Level = 5
  Coordinate System = Cartesian
  Simulation Type = Steady
  Output Intervals(1) = 1
  Steady State Max Iterations = 1
  Post File = "case.vtu"
End

Body 1
  Equation = 1
  Material = 1
End

Equation 1
  Active Solvers(1) = 1
End

Solver 1
  Equation = "ModelPDE"
  Variable = "Field"
  Procedure = "ModelPDE" "AdvDiffSolver"
  Linear System Solver = Direct
End

Material 1
  diffusion coefficient = 1.0
End

Boundary Condition 1
  Name = ”Fixed”
  Target Boundaries(1) = 1
  Field = 0.0
End

Boundary Condition 2
  Name = ”Flux”
  Target Boundaries(1) = 2
  Field Flux = 1.0
End
```

Further details of SIF file

- Comments start with **!**
- Avoid non-printable characters
  - Indent with spaces not tabulators
- Many keywords defined in **SOLVER.KEYWORDS** database
  - for others keyword type must be given
- Keyword(n,m) indicates a n × m array
  - Applicable for Integer and Real
- Correct unit's are the users responsibility
- Order of sections is mainly arbitrary
  - Except header
- For sections with indexing use continuous numbering starting from 1.
- include statement may be used to read other files within the SIF file
  - include material.sif
- Scripting by **MATC**
  - Preprocessor: $dens=1.013 — Evaluated once
  - Run-time: MATC "..." — Evaluated every time
- Scripting by **LUA**
  - Preprocessor: #dens=1.013 — Evaluated once
  - Run-time: LUA "..." — Evaluated every time

Real valued keyword functions

1) Tables can be use to define a piecewise linear (or cubic) dependency of a variable

```
Density = Variable Temperature
    Real cubic
       0  900
     273 1000
     300 1020
     400 1000
    End
```

Inside range: Interpolation

Outside range: Extrapolation!

2) MATC: a library for numerical evaluation of mathematical expressions

```
Density = Variable Temperature
  MATC "1000*(1 - 1.0e-4*(tx(0)-273.0))"
```

or as constant expressions

3) LUA: external library, faster than MATC

```
Density = Variable Temperature
  LUA "1000*(1 - 1.0e-4*(tx[0]-273.0))"
```

4) User defined function

```
Density = Variable Temperature
  Procedure "mymodule" "myproc"
```

Example of F90 User Function

File mymodule.F90:

```
FUNCTION myproc( Model, n, T ) RESULT(dens)
USE DefUtils
IMPLICIT None
TYPE(Model_t) :: Model
INTEGER :: n
REAL(KIND=dp) :: T, dens

    dens = 1000*(1-1.0d-4 *(T-273.0_dp))

END FUNCTION myproc
```

Compilation script comes with installation: **elmerf90**

Linux

```
$ elmerf90 mymodule.F90 -o mymodule.so
```

Windows

```
$ elmerf90 mymodule.F90 -o mymodule.dll
```

Keyword vectors and tensors

- Real valued keyword may be a vector or tensor
- Integer valued keyword may be a vector

```
Heat Conductivity(3,3) = 1.0 0.0 0.0 \
                         0.0 1.0 0.0 \
                         0.0 0.0 100.0

Gravity(4) = 0.0 0.0 -1.0 9.81


Target Nodes(4) = Integer 1 3 7 12
```

Sif file: Header

```
Header
  Check keywords "warn"
  Mesh DB "." "mymesh"
  Include Path "mylib"
  Results Directory "results"
End
```

- Header section does not follow the "Keyword = Value" syntax!
  - Read before the keyword database
- When checking keywords what to do
  - Warn, Abort, Echo
- Optionally we may define include and results directory
  - Working directory used by default

Sif file: Simulation

```
Simulation
  Max Output Level = 5
  Coordinate System = Cartesian
  Simulation Type = Transient ! Steady
  Timestep Intervals = 100
  Timestep Sizes = 0.1
  Timestepping Method = implicit euler
  Output Intervals(1) = 1
  Steady State Max Iterations = 1
  Post File = "case.vtu"
End
```

- Type of coordinate system
- Steady or Transient
- If transient: time stepping parameters
- Output files (to restart a run) and VTU file
- Output level : how verbose is the code?
- Restart information (optional)

Sif file: Constants

```
Constants
  Gas Constant = Real 8.314
  Stefan Boltzmann = Real 6.78e-08
End
```

- Natural constants etc.
  - As needed by the solver modules

Sif file: Body

```
Body i
  Name = "MyBody"
  Target Bodies(1) = 1
  Equation = 1
  Body Force = 2
  Initial Condition = 2
  Material = 4
End
```

- In Body are assigned the Equation, Body Force, Material and Initial Condition

Sif file: Equation

```
Equation i
  Name = "MySolvers"
  Active Solvers(2) = 1 2

  Convection = "computed"
End
```

- Lists the active solvers for the body.

- Some rare solver specific keywords also

Sif file: Solver

```
Solver i
  Equation = "HeatSolver"
  Exec Solver = "always"
  Variable = Temperature
  Procedure = "HeatSolve" "HeatSolver"
  Stabilize = True
  Steady State Convergence Tolerance = 1.0e-5
  Nonlinear System Max Iterations = 1
  Linear System Solver = Iterative
  Linear System Iterative Method = BiCGstab
  Linear System Max Iterations = 1000
  Linear System Convergence Tolerance = 1.0e-8
  Linear System Preconditioning = ILU1
End
```

- Specifies the numerical treatment for these equations (methods, criteria of convergence,...)
  - Name of variable to be solved
  - Element definitions
  - Stabilization strategies
  - Nonlinear system strategies
  - Linear system strategies
- Keywords treated both by library and solver module

Sif file: Material

```
Material i
  Name = "MyMaterial"
  Density = 1.0e3
  Heat Conductivity = 10.0
  Heat Capacity = 4.19e3
  Viscosity Model = "power law"
  Viscosity = 1.0
  Viscosity Exponent = $1.0/3.0
  Critical Shear Rate = 1.0e-10
End
```

- Sets material properties for the body
  - Most real values keywords can be dependent functions
  - Some can also be scalars and tensors

Sif file: Initial Condition

```
Initial Condition i
  Name = "MyGuess"
  Temperature = 293.0
  Velocity 1 = 1.0e-3
End
```

- Initial condition sets initial values
- Essential for time-dependent systems
- For steady-state problems provides the initial guess that may affect the iteration

Sif file: Body Force

```
Body Force i
  Name = "MySource"
  Heat Source = 1.0
  Flow Bodyforce 2 = -1.0
  Current Density = 1.23

  Varname Load = Real …
End
```

- Typically specifies the right-hand-side source term of the partial differential equation to be solved

- Discrete loads may be given that are directly associated with the matrix equation.

Sif file: Boundary Condition

```
Boundary Condition i
  Name = "Inlet"
  Target Boundaries(2) = 1 2
  Temperature = 293.0
  Velocity 1 = Variable "Coordinate 2"
    Real MATC "4*tx*(1-tx)"

  Body Id = 4

  Periodic BC = 5
End
```

- Different types of boundary conditions
  - Dirichlet: Variablename = Value (library routine)
  - Neumann: special keyword depending on the solver

- Boundary may be given a body id so that it can have an Equation, Material, Body Force etc. associated to it.
- Boundaries may be periodic, mortar boundaries, contact boundaries etc.

Sif file: Run Control

```
Run Control
  Run Control Iterations = Integer 100
  Parameter Count = Integer 4
  Optimization Method = String "simplex"
  Cost Function = Variable Time
     Real Procedure "CostFunction" "CostFunction"
End
```

- New section since 2020
- Provides parametric looping and internal optimization also in transient cases
- If used, this section should be the 1st one

Sif file: Component

```
Component i
  Name = string "gap_down"
  Master Bodies(1) = integer 5
  Calculate Magnetic Force = True
End
```

- Rarely used new section
- May define a collection of bodies or boundaries to be used for model lumping etc.
- Main usage currently in electromagnetics

Some remarks about the sif file

- The structure of sif file has almost one-to-one mapping with type Model_t in ElmerSolver code
  - Each keyword is an entry in list structure, e.g.

```
R = ListGetConstReal( Model % Constants,"Gas Constant")
```

- For many tasks there exists a separate solver a.k.a. module
  - Don't be afraid to add new addition solvers
  - Elmer modules + Elmer/Ice solvers
- Copy-paste works is often a good way to start
  - Hundreds of consistency tests under elmerfem/fem/test and elmerice/Tests
- Elmer Models Manual and ElmerSolver Manual have a keyword index
- Documentation is never complete – ask!