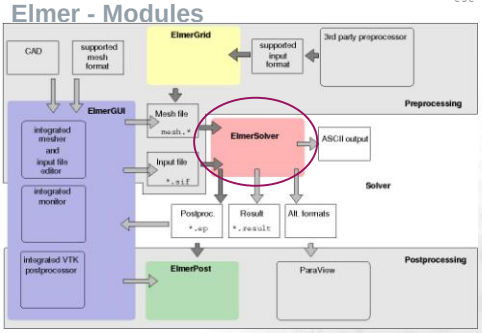
Solution Strategies with Elmer

Elmer Team
CSC – IT Center for Science Ltd. Elmer

Contents

Elmer - Modules



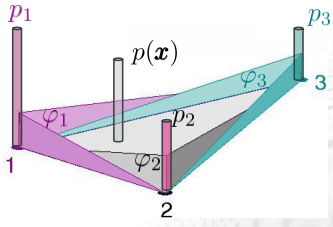
From the PDE to the SIF

- General advection-diffusion problem

$$c\varrho\left(\frac{\partial\Psi}{\partial t}+\mathbf{u}\nabla\Psi\right)=\nabla\cdot\underbrace{(-\kappa\nabla\Psi)}_{\mathbf{q}}+\varrho\sigma$$

  - For instance, heat transfer problem: $\Psi = T$
  - Coupled to (Navier-)Stokes via velocity: **u**
  - Non-linearities via material parameters, e.g., $c = c(\Psi)$

From the PDE to the SIF



$$p(\boldsymbol{x}) = p_1\varphi_1|_{\boldsymbol{x}} + p_2\varphi_2|_{\boldsymbol{x}} + p_3\varphi_3|_{\boldsymbol{x}}$$

From the PDE to the SIF

- Weak formulation:

$$\underbrace{\Psi_\beta\, a(\Delta t)\int_\Omega \varrho\varphi_\beta\varphi_\alpha d\Omega}_{\mathbf{M}} + \underbrace{\Psi_\beta\int_\Omega [\varrho\mathbf{u}\cdot\nabla\varphi_\beta\varphi_\alpha + \kappa\nabla\varphi_\beta\cdot\nabla\varphi_\alpha]\, d\Omega}_{\mathbf{S}} = \underbrace{\oint_{\partial\Omega}(\mathbf{q}\varphi_\alpha)\cdot\mathbf{n}d\Omega}_{\text{nat. BC}} + \underbrace{\int_\Omega \varrho\sigma\varphi_\alpha d\Omega}_{\mathbf{f}}$$

  - Time integration
  - Steady State:dependence on other variables
  - Non-linear iteration: internal dependence

From the PDE to the SIF

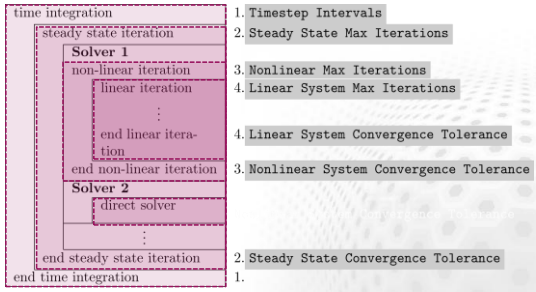
- Weak formulation:

$$\underbrace{\Psi_\beta\, a(\Delta t)\int_\Omega \varrho c\varphi_\beta\varphi_\alpha d\Omega}_{\mathbf{M}} + \underbrace{\Psi_\beta\int_\Omega [\varrho c\mathbf{u}\cdot\nabla\varphi_\beta\varphi_\alpha + \kappa\nabla\varphi_\beta\cdot\nabla\varphi_\alpha]\, d\Omega}_{\mathbf{S}} =$$

$$\underbrace{\oint_{\partial\Omega}(\mathbf{q}\varphi_\alpha)\cdot\mathbf{n}d\Omega}_{\text{nat. BC}} + \underbrace{\int_\Omega \varrho\sigma\varphi_\alpha d\Omega}_{\mathbf{f}}$$

  - System matrix: $\mathbf{M}+\mathbf{S}=\mathbf{A} \quad\Rightarrow\quad \mathbf{A}\,\Psi=\mathbf{f}$
  - Non-linearities: e.g. $c(\Psi)\Psi|^{(i)} \Rightarrow c(\Psi^{(i-1)})\Psi^{(i)}$
  - Natural BC: either to **A** or **f**

From the PDE to the SIF



Time Integration

- Two different methods:
  - Crank – Nicholson: `Crank - Nicholson`
  - Backward Difference Formula: `BDF`
    - `BDF Order` (if 1, then backward Euler – only choice for adaptive time-stepping)
- Additional settings:
  - `Time Derivative Order` (if 2 then Bossak)
  - `Timestep Intervals`
  - `Timestep Sizes`

Steady State Iteration

- Mutual dependence between Solvers
  - (e.g., Flow solution and convected temperature)
  - `Steady State Convergence Tolerance`

$$\|\Psi^{(j)}-\Psi^{(j-1)}\|/\|\Psi^{(j)}\| < \epsilon_{\text{st}}$$

  - `Steady State Max Iterations` $\quad j < j_{\max}.$
  - `Steady State Relaxation Factor`

$$\lambda_{\text{st}}: \quad \Psi^{(j)} \to \lambda_{\text{st}}\,\Psi^{(j)} + (1-\lambda_{\text{st}})\,\Psi^{(j-1)}$$

Nonlinear Solver Iteration

- **Nonlinear Problem**:

$$\underbrace{\mathbf{A}(\Psi)}_{\mathbf{M}+\mathbf{S}}\Psi=\mathbf{f}(\Psi) \quad\Rightarrow\quad \mathbf{A}(\Psi^{(i-1)})\,\Psi^{(i)}=\mathbf{f}(\Psi^{(i-1)})$$

- `Nonlinear System Convergence Tolerance`

$$\|\Psi^{(i)}-\Psi^{(i-1)}\|/\|\Psi^{(i)}\| < \epsilon_{\text{nl}}$$

- `Nonlinear System Max Iterations` $\quad i \le i_{\max}$
- `Nonlinear System Relaxation Factor`

$$\lambda_{\text{nl}}: \quad \Psi^{(i)} \to \lambda_{\text{nl}}\,\Psi^{(i)} + (1-\lambda_{\text{nl}})\,\Psi^{(i-1)}$$

Pre-conditioner

- But, before we solve, we usually apply a **pre-conditioner**
  - Find $\mathbf{P}\sim\mathbf{A}$, but much easier to invert
  - $\mathbf{P}^{-1}\mathbf{A}\sim\mathbf{I}$ has favourable condition number

$$\mathbf{P}^{-1}\mathbf{A}\,\Psi=\mathbf{P}^{-1}\mathbf{f}$$

`Linear System Preconditioning`
- `None`
- `Diagonal`
- `ILUn` (*n*=0,1,2,...) and `ILUT`

- Large potential for improving scalability!

Linear(ized) Problem

- Solving the **Linear(ized) Problem**

$$\mathbf{A}\,\Psi = \mathbf{f}$$

  - Keyword:
    `Linear System Solver`
- 3 ways to do that in Elmer:
  1. `Direct` methods (= inversion of **A**)
  2. `Iterative` methods (=working with approximations to **A**)
  3. Multi-grid methods (built-in or via externla libraries)

Linear(ized) Problem

- **Direct linear system solver**
- Keyword:
  `Linear System Direct Method`
  - `Banded` (default) LAPack
  - `UMFPACK` Unsymmetric MultiFrontal method (only serial)
  - `MUMPS` Unsymmetric MultiFrontal method (only parallel)
  - Sometimes the only way to go (if bad conditioned)
  - Costly: Elimination takes ~N^3 operations and needs to store N^2 unknows in memory

Linear(ized) Problem

- **Iterative solvers**:
  - Krylov subspace: $\mathbf{x}_n \in \mathrm{span}(K_n)$

$$K_n = \left[\mathbf{f}, \mathbf{A}\mathbf{f}, \mathbf{A}^2\mathbf{f}, \mathbf{A}^3\mathbf{f}, \ldots, \mathbf{A}^{n-1}\mathbf{f}\right]$$

$$\mathbf{R} = (\mathbf{f} - \mathbf{A}\,\mathbf{x}_n) \rightarrow \min.$$

  - `Linear System Iterative Method`
    - `GMRES` Generalized Minimal Residual Method
    - `CG`, `CGS`, `BiCGStab` Conjugate Gradient
    - `TFQMR` Transpose-free quasi-minimal residual
    - `GCR` Generalized Conjugate Residual