

# ElmerPost

*Post-processing your results*

Thomas Zwinger

`thomas.zwinger[at]csc.fi`

Computational Environment & Application

CSC—Scientific Computing Ltd.

The Finnish IT center for science

Espoo, Finland



# Contents

ElmerPost Program

The Interface

The Interface - Menus

How to Read Files

Command Window

Output for Other Post-processors

ASCII Based Output



# ElmerPost Program

- written by J. Ruokolainen



# ElmerPost Program

- written by J. Ruokolainen
- utilizes OpenGL (e.g. Mesa) and TCL/TK

# ElmerPost Program

- written by J. Ruokolainen
- utilizes OpenGL (e.g. Mesa) and TCL/TK
- nice add-ons as modules (JPG and MPEG output)



# ElmerPost Program

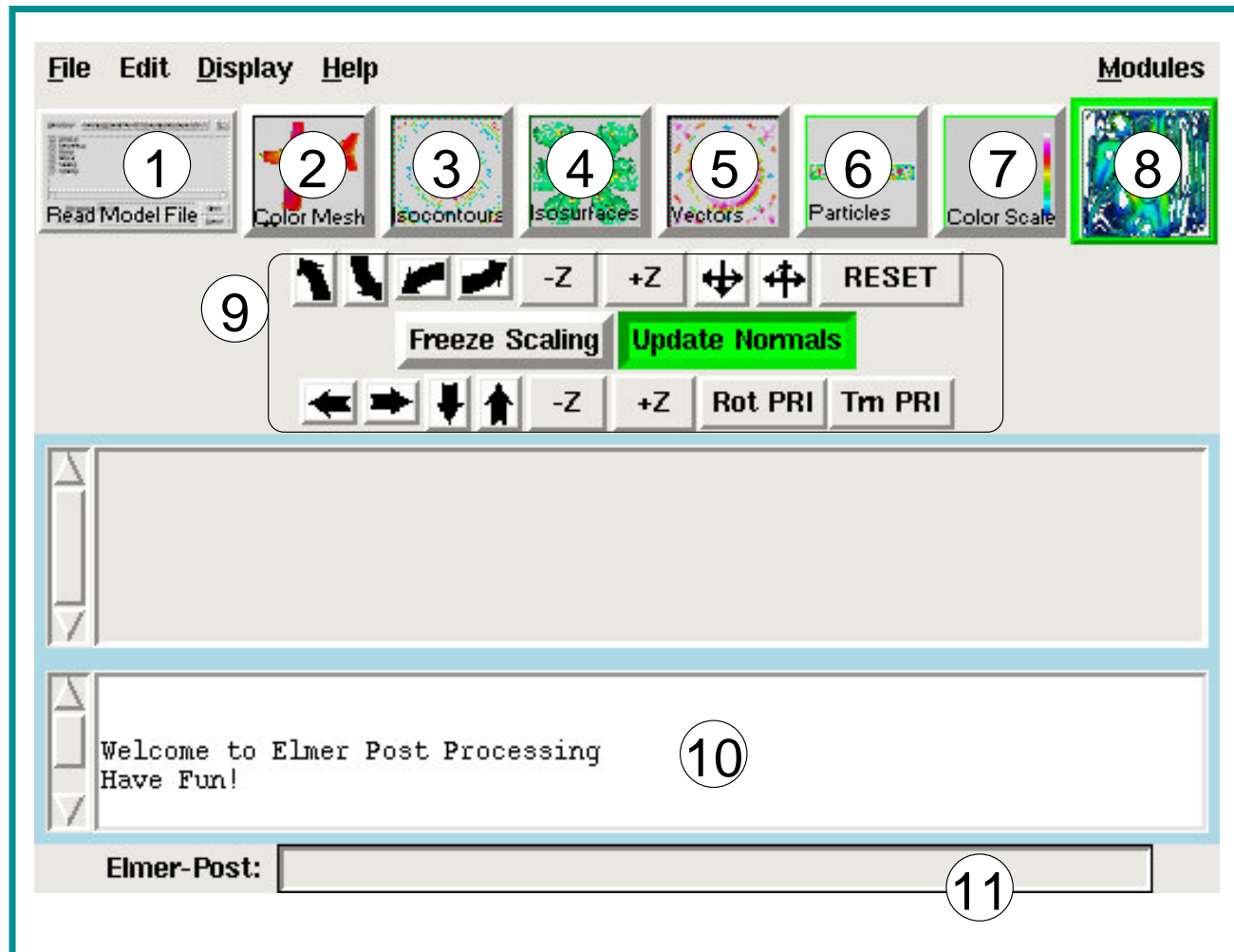
- written by J. Ruokolainen
- utilizes OpenGL (e.g. Mesa) and TCL/TK
- nice add-ons as modules (JPG and MPEG output)
- fast and easy way to post-process

# ElmerPost Program

- written by J. Ruokolainen
- utilizes OpenGL (e.g. Mesa) and TCL/TK
- nice add-ons as modules (JPG and MPEG output)
- fast and easy way to post-process
- launch simply with `ElmerPost "winsize 1000 800"`



# The Interface



1. Read result
2. Mesh display
3. Iso-contours
4. Iso-surfaces
5. Vector-field
6. Particles
7. Color-bar
8. Refresh
9. View settings
10. Output
11. Command

# The Interface - Menus

## File

- Open same as button 1 on the previous slide
  - Save Image save ppm or PostScript (not too large!!) image
  - Load Sicopolis load result file from SICOPOLIS (R. Greve's SIA code)
  - Quit exit ElmerPost
- ## Display
- buttons 2 - 7 from previous slide in a menu
- ## Help
- short help screen (quite useful)
- ## Modules
- if installed: *jpeg* and *mpeg-export*



# The Interface - Menus

## File

- Open same as button 1 on the previous slide
- Save Image save ppm or PostScript (not too large!!) image
- Load Sicopolis load result file from SICOPOLIS (R. Greve's SIA code)
- Quit exit ElmerPost

## Display

buttons 2 - 7 from previous slide in a menu

## Help

short help screen (quite useful)

## Modules

if installed: *jpeg* and *mpeg-export*

## Edit

- Colormap change the colormap
- Grouping toggle entities for display/hide
- Material define some optical effects, like opacity
- Background background color
- Timestep Control displays a series of time steps/iteration steps
- Math Module Window MATC command interface
- Camera Settings change your point of view (which is sometimes necessary)
- Object Clip Planes how far your model can reach



# How to Read Files

- Steady simulation: by default contains results of all iteration steps

The image shows a software dialog box with a title bar containing the text "Status: Not Done". The dialog is organized into several sections:

- Options:** Contains two checkboxes:  **Generate Surface Element Sides** and  **Generate Volume Element Sides**.
- File Information:** A section with four labels and corresponding input fields: **Nodes:**, **Elements:**, **Timesteps:**, and **DOFS:**.
- Select timesteps:** A section with three input fields labeled **First:** (containing '1'), **Last:** (containing '1'), and **Increment:** (containing '1'), followed by an **All** button.
- Select file:** A section with a label **Model file:** and an empty text input field.
- Buttons:** A row of four buttons at the bottom: **Read header**, **Read file**, **OK**, and **Close**.

# How to Read Files

- Steady simulation: by default contains results of all iteration steps  
⇒ the last one contains the correct result

The image shows a software dialog box with a title bar containing the text "Status: Not Done". The dialog is organized into several sections:

- Options:** Contains two checkboxes: "Generate Surface Element Sides" (checked) and "Generate Volume Element Sides" (unchecked).
- File Information:** Contains four labels with corresponding input fields: "Nodes:", "Elements:", "Timesteps:", and "DOFS:".
- Select timesteps:** Contains a row of controls: "First:" followed by an input field with "1", "Last:" followed by an input field with "1", "Increment" followed by an input field with "1", and an "All" button.
- Select file:** Contains a label "Model file:" followed by a text input field.
- Buttons:** At the bottom, there are four buttons: "Read header", "Read file", "OK", and "Close".

# How to Read Files

- Steady simulation: by default contains results of all iteration steps
  - ⇒ the last one contains the correct result
  - ⇒ use the highest possible value (obtained by pressing **All**) in both fields (**First** and **Last**)

The screenshot shows a dialog box with a title bar containing 'Status: Not Done'. The dialog is divided into several sections:

- Options:** Contains two checkboxes:  **Generate Surface Element Sides** and  **Generate Volume Element Sides**.
- File Information:** Contains four input fields: **Nodes:**, **Elements:**, **Timesteps:**, and **DOFS:**.
- Select timesteps:** Contains four input fields: **First:** (with value 1), **Last:** (with value 1), **Increment:** (with value 1), and an **All** button.
- Select file:** Contains a text input field labeled **Model file:**.
- Buttons:** At the bottom, there are four buttons: **Read header**, **Read file**, **OK**, and **Close**.

# How to Read Files

- Steady simulation: by default contains results of all iteration steps
  - ⇒ the last one contains the correct result
  - ⇒ use the highest possible value (obtained by pressing **All**) in both fields (**First** and **Last**)
- Transient simulation: load all (by pressing **All + Read File**)

Status: Not Done

Options:

Generate Surface Element Sides  
 Generate Volume Element Sides

File Information:

Nodes:   
Elements:   
Timesteps:   
DOFS:

Select timesteps:

First:  Last:  Increment

Select file:

Model file:

# How to Read Files

- Steady simulation: by default contains results of all iteration steps
  - ⇒ the last one contains the correct result
  - ⇒ use the highest possible value (obtained by pressing **All**) in both fields (**First** and **Last**)
- Transient simulation: load all (by pressing **All + Read File**)
  - ⇒ different time steps can be accessed using the **Timestep Control** entry in menu **Edit**

Status: Not Done

Options:

Generate Surface Element Sides

Generate Volume Element Sides

File Information:

Nodes:

Elements:

Timesteps:

DOFS:

Select timesteps:

First:  Last:  Increment

Select file:

Model file:

# Command Window

- either for geometry manipulation:

```
translate 1 0 1
```

```
rotate 90 0 0
```

```
scale 1 10 1
```



# Command Window

- either for geometry manipulation:

```
translate 1 0 1
```

```
rotate 90 0 0
```

```
scale 1 10 1
```

- or for MATC expressions: `math on = nodes`

```
math nodes(2,time(0)) = 100*(on(2,time(0)) + Mesh.Update(2,time(0)))
```

# Command Window

- either for geometry manipulation:

```
translate 1 0 1
```

```
rotate 90 0 0
```

```
scale 1 10 1
```

- or for MATC expressions: `math on = nodes`

```
math nodes(2,time(0)) = 100*(on(2,time(0)) + Mesh.Update(2,time(0)))
```

- similar window also in **Timestep Control**: the current time level is addressed by `time($t)`

# Output for Other Post-processors

- ResultOutputSolve offers during runtime:

|                 |           |            |
|-----------------|-----------|------------|
|                 | GID       | GID        |
|                 | Gmsh      | Gmsh       |
| Output Format = | Vtk       | VTK legacy |
|                 | Dx Format | Open DX    |
|                 | vtu       | ParaView   |

# Output for Other Post-processors

- ResultOutputSolve offers during runtime:

|                 |           |            |
|-----------------|-----------|------------|
|                 | GID       | GID        |
|                 | Gmsh      | Gmsh       |
| Output Format = | Vtk       | VTK legacy |
|                 | Dx Format | Open DX    |
|                 | vtu       | ParaView   |

```
Solver 1
  Equation = "ResultOutput"
  Procedure = "ResultOutputSolve" "ResultOutputSolver"
  Output File Name = "test"
  Output Format = string "vtu"
  Scalar Field 1 = String "Temperature"
  Vector Field 1 = String "Velocity"
End
```

# Output for Other Post-processors

- ResultOutputSolve offers during runtime:

|                 |           |            |
|-----------------|-----------|------------|
|                 | GID       | GID        |
|                 | Gmsh      | Gmsh       |
| Output Format = | Vtk       | VTK legacy |
|                 | Dx Format | Open DX    |
|                 | vtu       | ParaView   |

```
Solver 1
  Equation = "ResultOutput"
  Procedure = "ResultOutputSolve" "ResultOutputSolver"
  Output File Name = "test"
  Output Format = string "vtu"
  Scalar Field 1 = String "Temperature"
  Vector Field 1 = String "Velocity"
End
```

# ASCII Based Output

- ResultOutputSolve offers during runtime:

```
Procedure = File "SaveData" ...
```

- ... SaveScalars            bulk output
- ... SaveLine             line output
- ... SaveMaterials        material parameters added as variables
- ... SaveBoundaryValues   boundary parameters added as variable