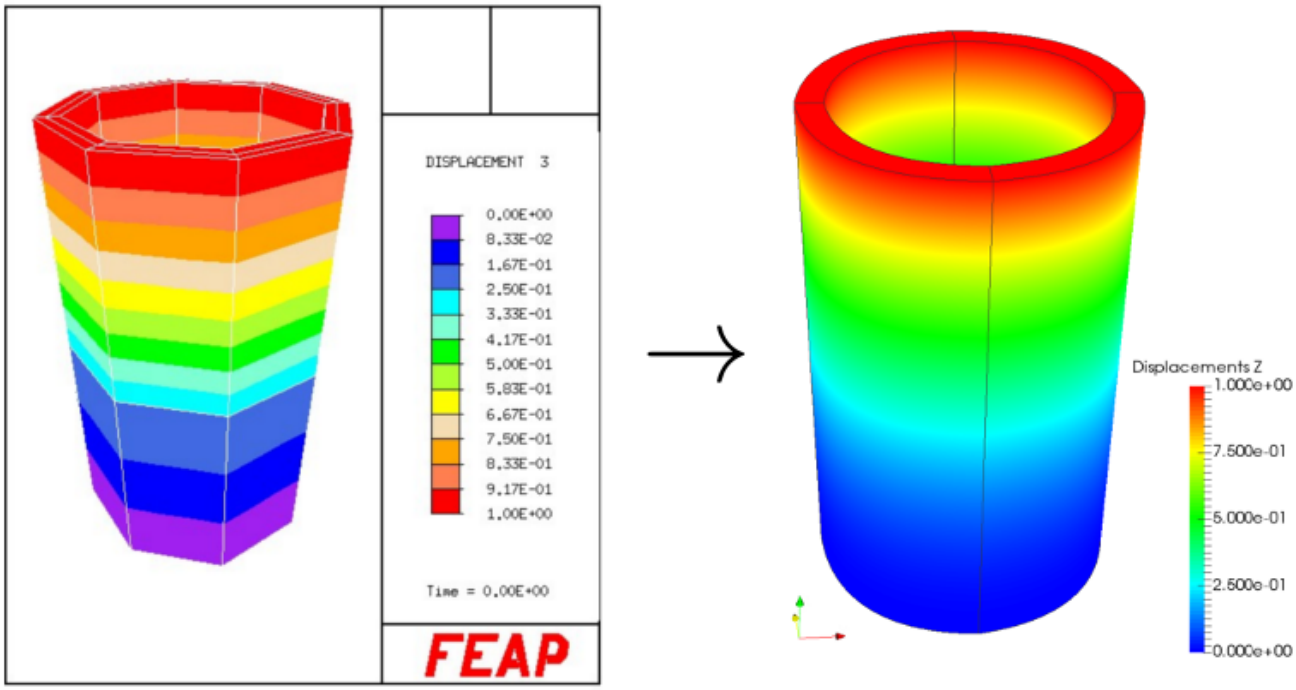
npvi user macro

Last update: July 14, 2020

OTTO VON GUERICKE
UNIVERSITÄT
MAGDEBURG

INSTITUTE OF MECHANICS
COMPUTATIONAL MECHANICS



Kindly report bugs to:
henning.venghaus@st.ovgu.de or
resam.makvandi@ovgu.de

Contents

1 Motivation / Features

It is a common practice in the isogeometric analysis post-processing to use FE-like mesh formats to plot the results. In the first step, an FE-mesh corresponding to the IGA-mesh is created. After that simulation results such as displacements (`DISP`), stresses (`STRE`), principal stresses (`PSTR`), etc. are projected onto the new mesh. The code to perform these actions is already included in the NURBFEAP/igaFEAP for the linear case. We have used the same approach and fitted the algorithms for our purposes. A similar routine to `uparaview.f`[1] is used to write the output file.

The routine includes the following features:

- linear interpolation of the geometry and the solution field (using linear FE elements)
- quadratic interpolation of the geometry and the solution field (using quadratic FE elements)
- uniform and non-uniform subdivisions of the NURBS elements
- standard FEAP output of the IGA mesh (in form of Serendipity and Lagrangian finite elements)
- **NEW:** post-processing of NURBS meshes using the so-called Bézier elements in Paraview (only for ver85 and ver86)
- **NEW:** compress VTU files on-demand (currently by executing an external Python file)

[1] `$FEAPHOME8_X/packages/paraview/uparaview.f`

2 Usage

NURBS **P**ara**vi**ew (NPVI) INPUT COMMAND MANUAL

```
npvi,specifier,n1,n2,n3
```

The `specifier` is a string which can contain up to 5 characters. The `specifier`, `n1`, `n2`, and `n3` are used to control the output.

substring of `specifier`	
combinations of:	
`t`	time flag - used when dealing with transient problems
`s`	stress flag - stress and principal stress output when activated
`v`	velocity flag - the velocity field output
`a`	acceleration flag - the acceleration field output
`m`	merge flag - to merge repeated nodes on the destination mesh (currently only for Bézier elements)
or `nusd`	reads values for non-uniform subdivision case (only for non-Bézier elements)
or `help`	displays a quick help
or `feop`	generates FE output
or `comp`	generates a python script to compress the VTU files

The parameters `n1`, `n2` and `n3` control the FE element type, the subdivision factor and the level of debugging information displayed on the screen, respectively. **All the three parameters must be integers.**

`n1`	Post-processing element type
`1`	linear element (2-node line, 4-node surface, 8-node brick) (**default**)
`2`	quadratic element - Serendipity (3-node line, 8-node surface, 20-node brick)
`3`	quadratic element - Lagrangian (3-node line, 9-node surface, 27-node brick) (only for Standard FEAP output)
`4`	Bézier element (line, quadrilateral, and hexahedral elements)

`n2`	uniform subdivision factor (only for non-Bézier elements)
`0`	zero or blank entry indicates non-uniform subdivision
`1`	each NURBS element will be represented by one FE element (**default**)
`2+`	each NURBS element will be split into multiple FE elements

`n3`	screen output
`0`	least output - only name of the `.vtu` file will be displayed (**default**)
`1`	status information will be displayed
`2`	addional information e.g. mesh dimension, number of elements will be displayed

n1 - n3	subdivision factors when `nusd` is active
	subdivision factor for each direction in the order ξ, η, ζ `0` will be treated as `1`

3 Examples

In general, all `specifier` and parameter combinations are possible. We now try to explain the command using some simple examples.

3.1 Basic usage

```
> tang,,1
> npvi
```

After solving the problem, the `npvi` routine runs with no additional specifications. The problem will be treated as static, and stresses will not be included in the output file. A PARAVIEW file will be created with the mesh and the displacements. One NURBS element will be represented by one **linear** FE element and no output, except the name of the written `.vtu` file, will be written on the screen.

3.2 Quadratic elements

```
> tang,,1
> npvi,,2,,1
```

After solving the problem, the `npvi` routine will be initiated. The problem will be treated as static, and stresses will not be included in the output file. A PARAVIEW file will be created with the mesh and the displacements. One NURBS element will be represented by one **quadratic** Serendipity FE element and status information will be written on the screen.

3.3 Stresses and uniform subdivision

```
> tang,,1
> npvi,s,1,2,1
```

After solving the problem, the `npvi` routine will be initiated. The problem will be treated as static, and **stresses** will be included in the output file. A PARAVIEW file will be created with the mesh, displacements and stresses. One NURBS element will be represented by **2** to the power of the dimension of the problem **linear** FE elements and status information will be written on the screen.

3.4 Transient problems and nonuniform subdivision

```
> npvi,nusd,4,2,3
> loop,,100
> time
> tang,,1
> npvi,t,1,0,1
> next
```

In the beginning, the parameters for the non-uniform subdivision is set. This is indicated by the specifier `nusd`. This will subdivide each NURBS element into 4, 2, and 3 elements in the first, the second and the third directions, respectively. In other words, at the end, each NURBS element will be represented using 24 FE elements. After solving the problem, the `npvi` routine will be initiated. The problem be treated

as **transient**, so there will be multiple `.vtu` files which will be named sequentially. Status information will be written on the screen.

3.5 Stresses and uniform subdivision

```
> tang,,1
> npvi,s,2,5,2
```

After solving the problem, the `npvi` routine will be initiated. The `.vtu` file contains the mesh, displacements and **stresses**. One NURBS element will be represented by 5 to the power of the problems dimension quadratic FE elements and status information will be written on the screen.

3.6 Generate new Standard FEAP input

```
> npvi,feop,2,2,2
```

In this case, no solution for the IGA problem needs to be calculated since only the mesh will be turned into a FE mesh. In this example, quadratic Serendipity elements have been used and each NURBS element is uniformly subdivided into two FE elements in each direction. Additional information will be written on the screen.

Boundary conditions and material properties will not be inherited from the original IGA input file.

3.7 Generate new Standard FEAP input with Lagrangian elements

```
> npvi,feop,3,2,2
```

Like the one above, but in this case with quadratic Lagrangian elements instead of Serendipity type.

The Lagrangian elements are only available for the Standard FEAP input, not for `.vtu`-files.

3.8 Generate a Python script to compress the created .VTU files

```
> npvi,comp
```

This creates a Python script (`npvi_vtkcomp.py`) which can be executed using `pvpython` or `pvbatch` (comes with the Paraview installation). The script goes through all the VTU files in the current path and compress the non-compressed ones. To run `pvpython` or `pvbatch`, the `bin` folder of Paraview must be included in the PATH environmental variable. The Python script can be ran multiple times during a simulation.

4 Code flowchart

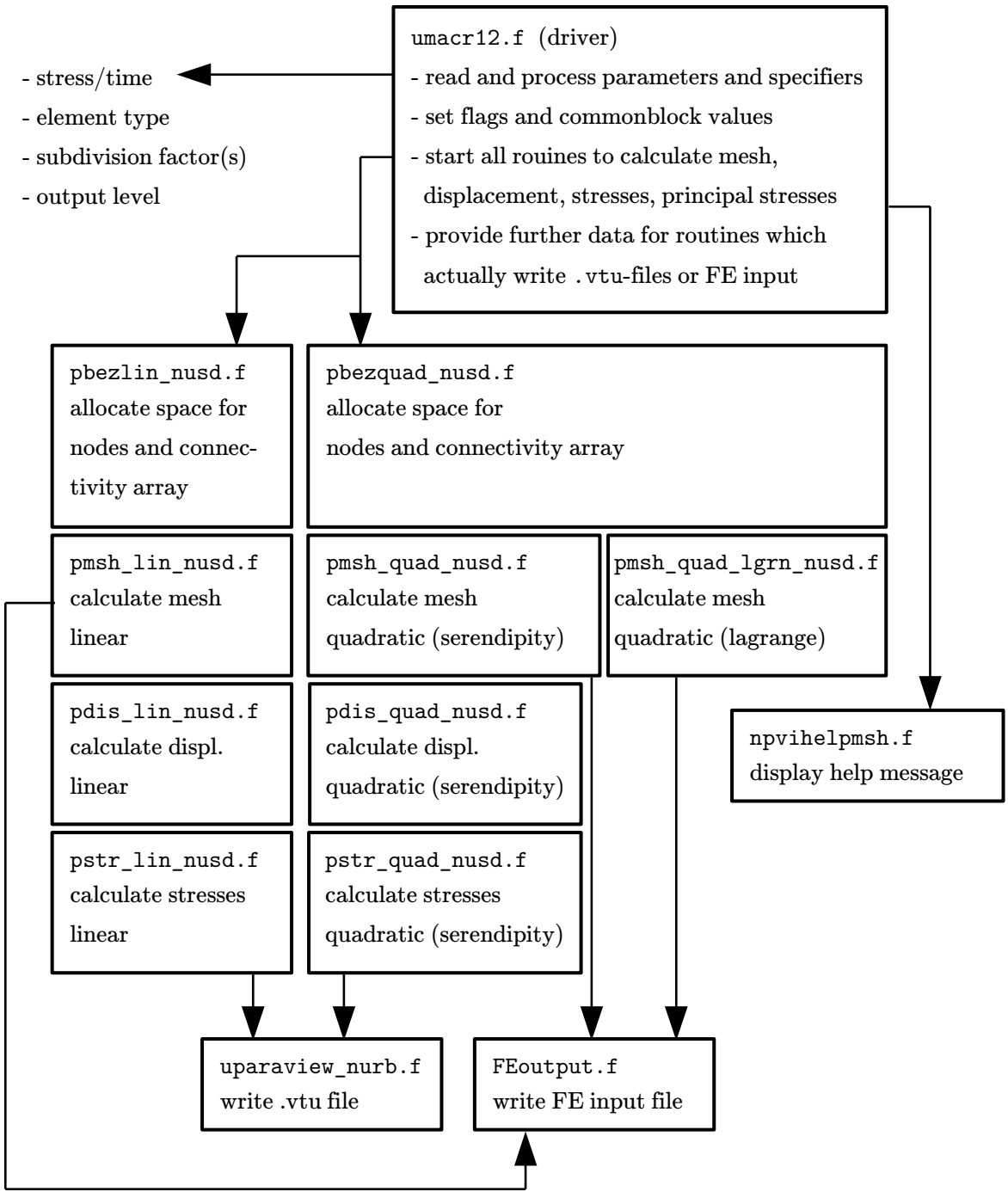


Figure 1: flowchart of `npvi` routine