writeMesh (write finite element mesh to inp, bdf, and msh files)

writeMesh is a MATLAB project for writing MATLAB finite element mesh to inp, bdf, and msh files. writeMesh is a part of lm2mesh package. Tested in MATLAB R2021b.

writeMesh supports mesh with single or multiple phases. Note that phase is also known as part, domain, or physical surface. In finite element modeling of composite materials, each phase in the mesh represents a distinct material component.

writeMesh woks for 2D mesh (triangular and quadrilateral); works for 3D mesh (tetrahedral and hexahedral).

Mesh file formats

- `inp` file (Abaqus)
- `bdf` file (Nastran bulk data, compatible with COMSOL)
- `msh` file (Gmsh mesh file format)
- For other formats, you can import the generated `msh` file into Gmsh and then export.

How to start

9 examples are provided. Examples are live script `mlx` files (`demo01.mlx` ~ `demo09.mlx`). If you find some text in the `mlx` file is missing, read the `html` file instead. Examples are also available as `html` files in the folder "demo_html".

Examples:

- demo01 - Triangular mesh with single phase
- demo02 - Triangular mesh with multiple phases
- demo03 - Quadrilateral mesh
- demo04 - 2D Mesh with quadratic elements
- demo05 - Node sets at the boundary and interface when writing `inp` files
- demo06 - Select formats when writing `inp` files
- demo07 - Tetrahedral mesh
- demo08 - Hexahedral mesh
- demo09 - Set `tolerance` when searching boundary node set
- inp_example1 - Format #1 of `inp` file
- inp_example2 - Format #2 of `inp` file
- inp_example3 - Format #3 of `inp` file

Write 2D mesh - Functions and input arguments

Functions:

```
printInp2d( vert, ele, tnum, ele_type, precision, file_name, opt );
printBdf2d( vert, ele, tnum, ele_type, precision, file_name );
printMsh( vert, ele, tnum, conn, precision, file_name );
```

In the above functions, the 1st and 2nd input arguments are required. The other input arguments are optional. The typical usage of function `printInp2d` is shown below.

```
printInp2d( vert, ele );
printInp2d( vert, ele, [], [], [], file_name );
printInp2d( vert, ele, [], [], precision, file_name );
printInp2d( vert, ele, [], [], precision, file_name, opt );
printInp2d( vert, ele, tnum, [], precision, file_name );
printInp2d( vert, ele, tnum, ele_type, precision, file_name, opt );
```

The usage of other functions is similar.

Input arguments:

- `vert`: Mesh nodes. It's a Nn-by-2 matrix, where Nn is the number of nodes in the mesh. Each row of `vert` contains the x, y coordinates for that mesh node.
- `ele`: Mesh elements. For linear triangular mesh, it s a Ne-by-3 matrix. For quadratic triangular mesh, it s a Ne-by-6 matrix. For linear quadrilateral mesh, it s a Ne-by-4 matrix. For quadratic quadrilateral mesh, it s a Ne-by-8 matrix. Ne is the number of elements in the mesh. Each row in `ele` contains the indices of the nodes for that mesh element.
- `tnum`: Label of phase, which corresponds to physical surface tag in Gmsh. `tnum` is a Ne-by-1 array, where Ne is the number of elements. If your mesh has multiple part or multiple phase, you can use `tnum` for labelling. `tnum(j,1) = k;` means the j-th element belongs to the k-th phase. If you mesh has single phase, you can set `tnum` as an empty array.
- `conn`: C-by-2 array of constraining edges, where each row defines an edge. You can set `conn` as an empty array.
- `ele_type`: element type. When omitted, `ele_typ` will be determined based on the size of variable `ele`.
- `precision`: number of digits to the right of the decimal point when writing node coordinates. When omitted, `precision=8;`
- `file_name`: file name of exported file, such as `'aaa.inp'`, `'D:\aaa.inp'`. When omitted, `file_name='test.suffix';`
- `opt` - a structure array. It is the extra options for `printInp2d`. It stores extra parameter settings for `printInp2d`. See demo05, demo06, and comments in `printInp2d.m`.

Notes:

- Function `printInp2d` will automatically search and export node sets at the boundary and interface. We can also define customized node set (see demo05).
- Function `printInp2d` works for triangular and quadrilateral mesh; works for linear and quadratic element.
- Function `printBdf2d` works for linear triangular and quadrilateral element; does not work for quadratic element.
- Function `printMsh` does not work for quadrilateral mesh.
- Function `printMsh` can be modified to support quadratic triangular element. I'm not sure whether this is necessary. If you prefer `msh` file with quadratic triangular element, feel free to let me know.

- I have tested the `msh` file generated by function `printMsh` in Gmsh 4.13.1. I didn't see any errors. The `msh` file generated by function `printMsh` is MSH file format version 4.1. Please refer to the Gmsh manual about details.

Write 3D mesh - Functions and input arguments

Functions:

```
printInp3d( vert, ele, tnum, ele_type, precision, file_name, opt );
printBdf3d( vert, ele, tnum, ele_type, precision, file_name );
```

In the above functions, the 1st and 2nd input arguments are required. The other input arguments are optional. The input arguments and usage of these functions are similar to their 2D counterparts.

Input arguments:

- `vert`: Mesh nodes. It's a Nn-by-3 matrix, where Nn is the number of nodes in the mesh. Each row of `vert` contains the x, y, z coordinates for that mesh node.
- `ele`: Mesh elements. For linear tetrahedral mesh, it s a Ne-by-4 matrix. For quadratic tetrahedral mesh, it s a Ne-by-10 matrix. For linear hexahedral mesh, it s a Ne-by-8 matrix. For quadratic hexahedral mesh, it s a Ne-by-20 matrix. Ne is the number of elements in the mesh. Each row in `ele` contains the indices of the nodes for that mesh element.
- `tnum`: Label of phase. It is a Ne-by-1 array. If you mesh has single phase, you can set `tnum` as an empty array.

Notes:

- Function `printInp3d` will automatically search and export node sets at the boundary and interface. We can also define customized node set (see demo05).
- Function `printInp3d` works for tetrahedral and hexahedral mesh; works for linear and quadratic element.
- Function `printBdf3d` works for linear element; does not work for quadratic element.
- Function `printMsh3d` is under development.

Copyright notice

Cite as

The development of writeMesh is nontrivial. If you use writeMesh for your project, please cite Im2mesh package as follows.

Ma, J., & Li, Y. (2025). Im2mesh: A MATLAB/Octave package for generating finite element mesh based on 2D multi-phase image (2.1.5). Zenodo. https://doi.org/10.5281/zenodo.14847059

Acknowledgments

Great thanks Dr. Yang Lu for providing valuable suggestions.