NekMesh: High-order mesh generation (NACA)

Tutorials

August 23, 2022

Department of Aeronautics, Imperial College London, UK
Scientific Computing and Imaging Institute, University of Utah, USA

2

CHAPTER 1

Introduction

This tutorial covers a new feature of NekMesh introduced in the two-dimensional mesh generator. This feature focuses on the creation of a NACA profile surrounded by a rectangular bounding box as well as its meshing.

> **Tip**
>
> This feature is an advanced functionality of NekMesh and the reader is advised to first complete the standard NekMesh tutorial `NekMesh: High-order mesh generation (from STEP)` before proceeding. This chapter however includes a reminder of the steps required to generate a mesh and run the solver.

To achieve the goals of this tutorial you will need an installation of the latest `master` version of Nektar++, available at nektar.info. This is because the features you will be using are relatively new and have yet to be included in the latest release. Standard installation instructions are applicable as described in the user guide, with one addition that CMake should be configured to include `NEKTAR_USE_MESHGEN=ON`. This will activate the installation of the relevant meshing routines.

> **Tip**
>
> When using the mesh generation routines Nektar++ will require an Open-Cascade installation to be present on your machine. Nektar++ is capable of downloading and installing OpenCascade for you, but this is highly not recommended as the installation time is very large. OpenCascade Community Edition (OCE) can be installed through package managers:
>
> - **Homebrew:** `brew install homebrew/science/oce`
>
> - **MacPorts:** `port install oce`
>
> - **Ubuntu/Debian:** `sudo apt-get install liboce-foundation-dev liboce-modeling-dev`
>
> - Other Linux installations have a version of OCE in their package manager, usually called `liboce`.
>
> Nektar++ will be able to pick up on any of these OCE installations. Installation of NekMesh on Windows with mesh generation is not possible due to the OpenCascade requirements at this time.

This tutorial additionally requires:

- Nektar++ IncNavierStokesSolver with pre- and post-processing tools,

- A visualisation tool such as [Paraview](#) or [VisIt](#)

1.1   Goals

After the completion of this tutorial, you will be familiar with:

- the generation of a 2D mesh from the NACA generator;

- the visualisation of the mesh in Paraview or VisIt

- running a simple simulation on this mesh with IncNavierStokesSolver; and

- the post-processing of the data and the visualisation of the results in Paraview or VisIt.

Task 1.1

Prepare for the tutorial. Make sure that you have:

- Installed and tested *Nektar++* v5.2.0 compiled from source. We will refer to the directory where you installed *Nektar++* as `$NEK` for the remainder of the tutorial.

- Make a directory of your choosing, for example `tutorial`, and download the tutorial files from http://doc.nektar.info/tutorials/5.2.0/mesh-generation/2d-naca/mesh-generation-2d-naca.tar.gz into this directory.

- Unpack the tutorial files by using

  ```
  tar -xzvf mesh-generation-2d-naca.tar.gz
  ```

  to produce a directory `2d-naca` with subdirectories called `tutorial` and `complete`.

The tutorial folder contains:

- an .mcf file;

- a session file which will run the NACA mesh.

NACA wing

An additional feature of the 2D capabilities of NekMesh is to generate meshes of NACA wings without the need for a CAD file. By listing the `CADFile` parameter as a 4 digit NACA code in the .mcf, NekMesh will generate the NACA wing. Additional parameters about the domain are required, that define the minimum and maximum x- and y-coordinates for the grid and the angle of attack of the wing:

```
1    <P PARAM="Xmin" VALUE="−1.0" />
2    <P PARAM="Ymin" VALUE="−1.0" />
3    <P PARAM="Xmax" VALUE="3.0" />
4    <P PARAM="Ymax" VALUE="1.0" />
5    <P PARAM="AOA" VALUE="15.0" />
```

The mesh we make is for a simple NACA 6412 profile in a rectangular domain.

Task 2.1

Create a triangular mesh according to the .mcf provided in the tutorial folder. To make this mesh, run the command

```
$NEK/NekMesh naca.mcf naca.xml
```

More details about the generation process can be seen by enabling the *verbose* command line option `-v`:

```
$NEK/NekMesh -v naca.mcf naca.xml
```

This will produce a Nektar++ mesh, which we now need to visualise.

> **Task 2.2**
>
> Visualise the mesh using FieldConvert. For Tecplot output for visualisation in VisIt, run the command
>
> ```
> $NEK/FieldConvert naca.xml naca.dat
> ```
>
> or, for VTK output, for visualisation in ParaView or VisIt, run the command
>
> ```
> $NEK/FieldConvert naca.xml naca.vtu
> ```

As these meshes are converted to a linear mesh, since this is what the visualisation software supports, the curved elements can look quite faceted. To counteract this, we can add an optional `-nX` command line option to FieldConvert, which increases the number of subdivisions used to view the elements through interpolation.

> **Task 2.3**
>
> Run the command
>
> ```
> $NEK/FieldConvert -n 8 naca.xml naca.vtu
> ```
>
> and visualise the output, which should now look far smoother around the aerofoil's leading edge and give a better visualisation of the high-order elements.

The .mcf has already been configured for the generation of a sufficiently fine mesh. Among others, it includes a boundary layer mesh as well as a refinement region in the wake of the aerofoil. We can now try to run some simulations on this mesh. A session file `session_naca.xml` for the aerofoil mesh is provided, which will execute a low Reynolds number incompressible simulation. The resulting flow field should look like in Fig. 2.1.

> **Task 2.4**
>
> Execute the solver with
>
> ```
> $NEK/IncNavierStokesSolver naca.xml session_naca.xml
> ```
>
> Process the final flow field visualisation with
>
> ```
> $NEK/FieldConvert naca.xml naca.fld naca.vtu
> ```
>
> and visualise the results.

It is possible to visualise the unsteady flow by getting the solver to write checkpoint files every certain number of timesteps. The output frequency is controlled by the parameter
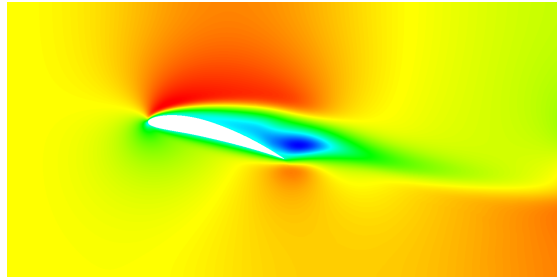
```
1   <P> IO_CheckSteps = 50 </P>
```

Figure 2.1 Visualisation of streamwise velocity field at $t = 2.5$.

Task 2.5

Rerun the solver with the checkpoint I/O turned on, by enabling the parameter in the parameters tag in the session file. Re-run the solver with

```
$NEK/IncNavierStokesSolver naca.xml session_naca.xml
```

Finally, process these checkpoint files for visualisation by using the command

```
for i in {0..100}; do $NEK/FieldConvert naca.xml naca_${i}.chk
naca_${i}.vtu; done
```

Note that this is a `bash` command – if you use an alternative shell, such as `tcsh`, you may need to alter this syntax. Load the files into your visualiser as a group, and use the frame functions to view the flow as an unsteady animation.

Following these first 2 tutorials on two-dimensional high-order mesh generation using NekMesh, the reader should now be able to modify the mesh configuration file according to their needs.

Task 2.6

Change the mesh and solver parameters to produce different meshes and flow results, so that the wake can be captured accurately.