Nektar++ for stability analysis

August 11, 2011

The aim of this tutorial is to introduce the user to the spectral/*hp* element framework *Nektar++* and its use for simple stability analyses of flows. This guide assumes the user has successfully compiled the libraries, the solvers and the utilities, as explained on the website[1]. A series of regression tests are included to check that the software is producing the expected results. Please ensure these all pass before continuing.

Note: The example commands given assume the *Nektar++* executables can be found in your shell path. If you have not set your path accordingly you will need to specify the full path to the *Nektar++* executable.

In the first section it will be presented the stability analysis of the two dimensional channel flow, through both the splitting scheme and the Stokes algorithm.It will be then studied the transient growth of the flow past a backfacing step and the direct/adjoint stability analysis of a flow past a cylinder.

1 Two-dimensional Channel flow

The linear stability analysis is a technique that allows to determine the asymptotic stability of a given flow. First of all, let us consider the non-dimensional, viscous linearised Navier-Stokes equations. These equations describe the evolution of an infinitesimal perturbation of the fields for a given base flow.

$$\frac{\partial \mathbf{u}'}{\partial t} + \mathbf{U}\cdot\nabla\mathbf{u}' + \mathbf{u}'\cdot\nabla\mathbf{U} = -\nabla p' + \frac{1}{Re}\nabla^2\mathbf{u}' + \mathbf{f}' \qquad (1)$$

with continuity equation

$$\nabla\cdot\mathbf{u}' = 0 \qquad (2)$$

$\mathbf{u}'$ and p' represent the perturbations of the velocity and pressure, while $\mathbf{U}$ the base flow. We will consider as parallel base flow through a 2-D channel (known as Poiseuille flow) at Reynolds number $Re = 7500$:

$$\mathbf{U} = y(1-y)\mathbf{e_x} \qquad (3)$$

The domain is $\Omega = [-\pi,\pi]\times[-1,1]$ and it is composed by 48 quadrilaterals as shown in figure (**??**)

[1] www.nektar.info

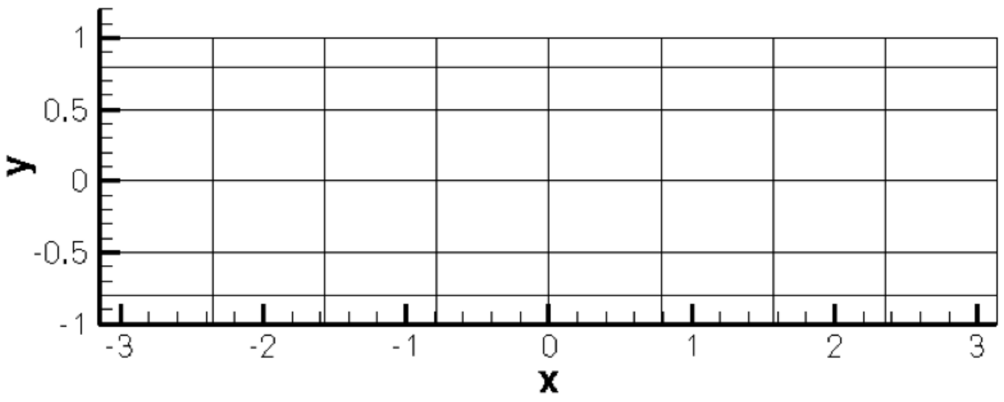

Figure 1: 48 quadrilaterals mesh

This simple mesh was created using the software *Gmsh* and the first step is to convert it into a suitable input format so that it can be processed by *Nektar* + + libraries.

In the tutorial folder `NekTutorial/Tutorial/Channel/Geometry` you will find the following files:

- `Channel.msh`- *Gmsh* generated mesh data listing mesh vertices and elements. This is the mesh file used by the *Nektar* + + pre-processing utilities.
- `Channel.xml` - *Nektar++* session file generated from `Channel.msh` using a *Nektar++* utility.

`Channel.xml` can be generated using the `MeshConvert` pre-processing tool in the `utilities/builds/PreProcessing/` directory. To generate the *.xml* file from the `.msh` file, run the command

```
MeshConvert Channel.msh Channel.xml
```

2 Computation of the base flow

The first step is creating an appropriate base flow that is necessary for solving the the linearised equations. Since in the hydrodynamic instability theory, it is assumed that the base flow is a incompressible, it can be easily computed using the Incompressible Navier-Stokes solver.
Moreover, the boundary condition will be no-slipping conditions on the walls and periodic ones for the inflow/outflow. In this case, since it is not a constant pressure gradient that drives the flow, it is required to set up a constant body-force in the axial direction that it is easy to verify is equal to 8ν.

In the folder `NekTutorial/Tutorial/Channel/Base` it is present the file `Channel-Base.xml` that contains the geometry above described and the necessary parameters to solve the problem.
The `GEOMETRY` section defines the mesh of the problem and it is generated automatically through the pre-processing utilities. The expansion type and order is specified in the `EXPANSIONS` section. An expansion basis is applied to a geometry composite specified in the `GEOMETRY` section. A default entry is always included by the `MeshConverter`. In this case `C[0]` refers to the set of all elements. The `TYPE` tag specifies the choice of the polynomial functions to use in the expansion. It can be set to `FOURIER`

or CHEBYSHEV. In this case it was chosen to use the GLL_LAGRANGE that refers to Lagrange polynomials through the Gauss, Lobatto As an alternative, it is possible to use MODIFIED that refers to a basis of Legendre polynomials modified to enable the boundary/interior decomposition.The section FIELDS specifies the fields that are considered in the problem, for example the two components of the velocity and the pressure.

```
<EXPANSIONS>
    <E COMPOSITE="C[0]" NUMMODES="8"  FIELDS="u,v,p" TYPE="MODIFIED"/>
</EXPANSIONS>
```

If we examine Channel-Base.xml, we can see how to define the conditions of the particular problem to solve. These are all enclosed in a CONDITIONS section. This section contains a number of things:

1. **Solver information** (SOLVERINFO) such as the equation type, the projection type (Continuous or Discontinuous Galerkin), the kind of advection term (Convective, Linearised, Adjoint) and problem to solve (Driver property), along with other properties. The solver properties are specified as quoted attributes and have the form

   ```
   <I PROPERTY="[STRING]" VALUE="[STRING]" />
   ```

   To Do: In the SOLVERINFO section set the property EQTYPE to UnsteadyNavierStokes to select the non-linear incompressible Navier-Stokes equations., the ADVECTIONFORM to Convective and the Projection property to Continuous. The property Driver selects the type of problem to solve (stability analysis with different methods to compute the eigenvalues or computation of the solution of the equations). In this case, set its value to STANDARD to compute and print the solution of the non-linear Navier-Stokes equations.

2. The **parameters** are specified as name-value pairs:

   ```
   <P> [KEY] = [VALUE] </P>
   ```

   Parameters may be freely used either from within the solver , or within other expressions, such as function definitions or other parameters defined subsequently.

   To Do: Declare a parameter Re that sets the Reynolds to 7500 and Kinvis equal to $1/Re$ that represents the the kinematic viscosity ν.

3. The declaration of the variable(s).

   ```
   <VARIABLES>
       <V ID="0"> u </V>
   </VARIABLES>
   ```

4. The specification of boundary regions in terms of composites defined in the geometry and the conditions applied on those boundaries. Boundary regions have the form

   ```
   <B ID="[INDEX]"> [COMPOSITE-ID] </B>
   ```

   The boundary conditions enforced on a region take the following format for one or more variable names specified in the VARIABLES section. The REF attribute for a boundary condition region should correspond to the ID of the desired BOUNDARYREGION.

```
<REGION REF="[B-REGION-INDEX]">
    <[TYPE] VAR="[VARIABLE]" VALUE="[EXPRESSION]"/>
    ...
</REGION>
```

5. The definition of the forcing term, f, and the exact solution. A body-forcing term has the form

   ```
   <FUNCTION NAME="BodyForce">
   <E VAR="[VARIABLE]" VALUE="[EXPRESSION]"/>
   </FUNCTION>
   ```

   To Do: Define a body forcing $f_x = 8\nu$ in the axial direction.

   Todo: Define the section for the `EXACTSOLUTION` along with an expression for u_{ex}. The format is identical to that for specifying the forcing function.

This completes the specification of the problem on the rectangualar mesh. It can then be solved using the `IncNavierStokesSolver`. The executable is located in the folder `solver/builds/dist/bin/`[2]

```
IncNavierStokesSolver Channel-Base.xml
```

To view the output in *Gmsh* use the post-processing tools in the `utilities` directory, as we have done for the pre-processing. You can produce outputs for *Gmsh* , *TecPlot* and *Paraview* using the corresponding converter from the terminal. For example, to convert the `Channel-Base.fld` to *Gmsh* format, use

```
FldToGmsh Channel-Base.xml Channel-Base.fld
```

[2]If you compiled the library in Debug mode, the executables will have the suffix `-g`.

3 Stability analysis

After having computed the base flow it is now possible to calculate the eigenvalues and the eigenmodes of the linearised Navier-Stokes equations. Two different algorithm can be used to perform this analysis: the splitting scheme (`VelocityCorrectionScheme`) and the Stokes algorithm (`CoupledLinearisedNS`). We will consider both cases, highlighting the similarities and differences of these two methods. In this tutorial it will be used the Implicitly Arnoldi Method (I.R.A.M.), that is implemented in the open-source *Arpack* library.

3.1 Splitting Scheme Method

The first analysis to run is the computation of the leading eigenvalues and eigenvectors for using the splitting scheme method. It is necessary that the base flow file, computed in the previous section, is copied in the folder `NekTutorial/Tutorial/Channel/VelocityCorrectionScheme` and renamed `Channel_VCS.bse`, that is the format used by the code to upload an external flow. In this folder it is possible to find the file `Channel_VCS.xml` that contains all the necessary specifications to perform the direct stability analysis. The format is similar to the one described in the previous section, therefore only the main differences are going to be discussed.

The `GEOMETRY`, the `TYPE` and number of modes (`NUMMODES`) must be the same used for the base flow, so they are not going to be described again. However, it is important to set some important parameters that are characteristic of the method.

To Do:

1. the `AdvectionForm` in this case should be set to `Linearised` in order to select the linearised Navier-Stokes advection term.
2. `Driver` must be set to `Arpack` to use *Arpack* library .
3. To start Arnoldi method with a random initial vector it is necessary to add the Solver Info `InitialVector` to `Random`.
4. It can be specified what kind of eigenvalues to compute through the solver property `ArpackProblemType`, in particular the ones with the largest magnitude (`LargestMag`), largest real part (`LargestReal)` or largest imaginary part (`LargestIm`). In our case we are interested in computed the eigenvalues with the largest magnitude that determinate the stability of the flow.
5. A series of parameters to make I.R.A.M. work are then specified.
   - `kdim`: it is the dimension of Krylov-subspace.
   - `nvec`: it is the number of requested eigenvalues.
   - `nits`: it is the number of maximum allowed iterations.
   - `evtol`: it is the accepted tolerance on the eigenvalues and it determines the stopping criterion.

It is now possible to run the direct stability analysis for the channel flow, typing:

```
IncNavierStokesSolver Channel_VCS.xml
```

The code will converge after 264 iterations printing the requested eigenvalues and eigenmodes both on screen on the file `Channel_VCS.evl`.

The values printed allow to write the eigenvalues in an exponential form $Me^{i\theta}$ where $M = |\lambda|$ is the magnitude, while $\theta = arctan(\lambda_i/\lambda_r)$.

$$\lambda_{1,2} = 1.0024e^{\pm 0.24984i} \tag{4}$$

It is interesting to consider more general quantities that do not depend on the time length of each iteration T. For this purpose it was considered the growth rate $\sigma = ln(M)/T$ and the frequency $\omega = \theta/T$. Verify that for this case:

$$\sigma = 2.23711 \cdot 10^{-3} \tag{5}$$

$$\omega = \pm 2.498413 \cdot 10^{-1} \tag{6}$$

This values are in accordance with the literature, in fact in Canuto et al., 1988 suggests $2.23497 \cdot 10^{-3}$ and $2.4989154 \cdot 10^{-1}$ for growth and frequency respectively. The eigenmodes associated to the computed eigenvalues are stored in the vectors `Channel_VCS_eig_1` and `Channel_VCS_eig_2`. It is then possible to convert this file into a *Gmsh* output using *Nektar* + + post-processing routines. For the first eigenvector, for example:

```
FldToGmsh Channel_VCS.xml Channel_VCS_eig_1
```

3.2 Stokes Algorithm

It is now possible to perform the same stability analysis using a different method base on the Stokes algorithm. This method requires the solution of the full velocity-pressure system, meaning that the velocity matrix system and the pressure system are coupled, differently from the splitting/projection schemes. It is easy to extend this method to solve the unsteady Navier-Stokes equations introducing into the Stokes problem an unsteady term $\mathbf{u_f}$ that modifies the weak Laplacian operator into a weak Helmoholtz operator. Furthermore, the non-linear terms are explicitly advanced in time and treated as a forcing function to the Stokes solver.

Inside the folder `NekTutorial/Tutorial/Channel/CoupledSolver` it is present the file `Channel_Coupled.xml` that contains all the necessary parameters that should be defined. Similarly to the previous case, it is possible to specify the base flow putting the `Channel_Coupled.bse` in the working directory. However, it is possible to specify the base flow directly from session file if it has an analytical expression like in our example. This second method will the be presented in this section. Even in this case, the geometry, the type and number of modes are the the same of the previous simulations.

To Do:

- Set up the `SolverType` tag to `CoupledLinearisedNS` in order to solve the linearised Navier-Stokes equations using $Nektar++$'s coupled solver.
- the `EQTYPE` must be set to `SteadyLinearisedNS` and the `Driver` to `Arpack`.
- To compute the eigenvalues with the largest magnitude, specify `LargestMag` in the property `ArpackProblemType`.

It is important to note that the usage of the coupled solver requires that **only the velocity components** must be specified, while the pressure can be directly computed through their values.

To Do: In this case it is a body force that provides the driving of the flow. This is done by the setting the previously introduced function `BodyForce` to $cos(y)$ for the u component and $sin(y)$ for the v component. At last it is necessary to set up the base flow. In this case, since we have an analytical expression we can define it through the function:

```
<FUNCTION NAME="AdvectionVelocity">
<E VAR="[VARIABLE]" VALUE="[EXPRESSION]"/>
</FUNCTION>
```

The u component must be set up to $1 - y^2$, while the v-component to zero.
It is now possible to run the simulation typing:

```
./IncNavierStokesSolver Channel_Coupled.xml
```

Using the Stokes algorithm, we are computing the leading eigenvalue of the inverse of the evolution operator $\mathcal{L}^{-1}$[3]. Therefore the eigenvalues of $\mathcal{L}$ are the inverse of the found values found. However,

[3] $\mathcal{L}$ is the evolution operator $d\mathbf{u}/dt = \mathcal{L}\mathbf{u}$

it is interesting to note that the value is different from the one calculated with the Splitting Scheme Method, producing an apparent inconsistency in the results. However, this can be explained considering that the largest eigenvalues associated to the operator $\mathcal{L}$ correspond the ones that are clustered near the origin of the complex plane if it is considered the spectrum of $\mathcal{L}^{-1}$. Therefore eigenvalues with a smaller magnitude may be present but they are not associated to the largest magnitude eigenvalue of operator $\mathcal{L}$. In order to verify this issue, it is possible to search the eigenvalues with the largest imaginary part.

To Do: Set up the Solver Info tag `ArpackProblemType` to `LargestImag` and run the simulation again.

In this case, it is easy to to see that the eigenvalues of the evolution operator $\mathcal{L}$ are the same ones computed in the previous chapter with the time-stepping approach.

(7)