Global Stability Analysis: Flow over a Backward-Facing Step

Tutorials

January 20, 2021

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

2

Introduction

This tutorial further explores the use of the spectral/hp element framework *Nektar++* to perform global stability computations. Information on how to install the libraries, solvers, and utilities on your own computer is available on the webpage www.nektar.info.

This tutorial assumes the reader has already completed the previous tutorials in the Flow Stability series on the channel and cylinder and therefore already has the necessary software installed.

> **Task 1.1**
>
> Prepare for the tutorial. Make sure that you have:
>
> - Downloaded the tutorial files: `http://doc.nektar.info/tutorials/5.`
>   `0.1/flow-stability/bfs/flow-stability-bfs.tar.gz`
>   Unpack it using `unzip flow-stability-bfs.tar.gz` to produce a directory `flow-stability-bfs` with subdirectories called `tutorial` and `complete` We will refer to the `tutorial` directory as `$NEKTUTORIAL`.

In this tutorial we will perform a transient growth analysis of the flow over a backward-facing step. This is an important case which allows us to understand the effects of separation due to abrupt changes of geometry in an open flow. The transient growth analysis consists of computing the maximum energy growth, $G(\tau)$, attainable over all possible initial conditions $\mathbf{u}'(0)$ for a specified time horizon τ. It can be demonstrated that it is equivalent to calculating the largest eigenvalue of $\mathcal{A}^*(\tau)\mathcal{A}(\tau)$, with $\mathcal{A}$ and $\mathcal{A}^*$ being the direct and the adjoint operators, respectively. Also note that the eigenvalue must necessarily be real since $\mathcal{A}^*(\tau)\mathcal{A}(\tau)$ is self-adjoint in this case.

- Folder `geometry`

   - `bfs.geo` - *Gmsh* file that contains the geometry of the problem

- – `bfs.msh` - *Gmsh* generated mesh data listing mesh vertices and elements.

- Folder `base`

  - – `bfs-Base.xml` - *Nektar++* session file, generated with the `$NEK/NekMesh` utility, for computing the base flow.

  - – `bfs-Base.fld` - *Nektar++* field file that contains the base flow, generated using
    `bfs-Base.xml`.

- Folder `stability`

  - – `bfs_tg.xml` - *Nektar++* session file, generated with `$NEK/NekMesh`, for performing the transient growth analysis.

  - – `bfs_tg.bse` - *Nektar++* field file that contains the base flow. It is the same as the `.fld` file present in the folder `Base`.

Figure 2.1 shows the mesh, along with a detailed view of the step edge, that we will use for the computation. The geometry is non-dimensionalised by the step height. The domain has an inflow length of 10 upstream of the step edge and a downstream channel of length 50. The mesh consist of $N = 430$ elements. Note that in this case the mesh is composed of both triangular and quadrilateral elements. A refined triangular unstructured mesh is used near the step to capture the separation effects, whereas the inflow/outflow channels have a structure similar to the previous example. Therefore in the `EXPANSION` section of the `bfs-Base.xml` file, two composites (`C[0]` and `C[1]`) are present. For this example, we will use the modal basis with 7th-order polynomials.

We will perform simulations at $Re = 500$, since it is well-known that for this value the flow presents a strong convective instability.

Computation of the base flow

The file `bfs_tg.bse` is the output of the base-flow computation that should be run for a non-dimensional time of $t \geq 300$ to ensure that the solution is steady.
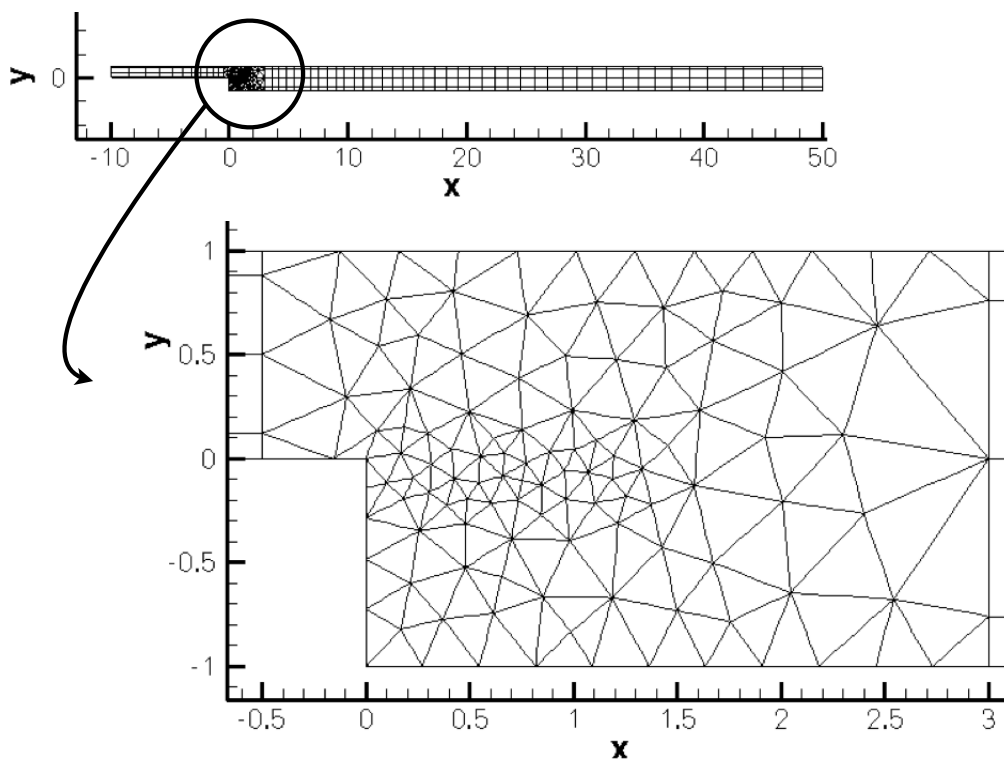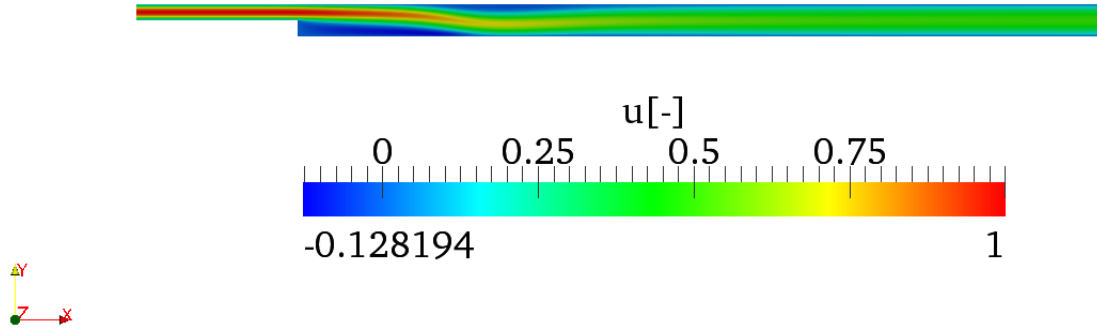
Figure 2.1 Mesh used for the backward-facing step

> **Task 2.1**
>
> Convert the base flow field file `bfs_tg.bse` into VTK format to look at the profile of the base flow. Note the separation at the step-edge and the reattachment downstream.

The streamwise component of the velocity, u, should look like in figure 2.2.

Figure 2.2 Streamwise component of the velocity of the backward-facing step base flow.

Stability analysis

We will now perform transient growth analysis with a Krylov subspace of `kdim=4`. The parameters and properties needed for this are present in the file `bfs_tg.xml` in `$NEKTUTORIAL/stability`. In this case the `Arpack` library was used to compute the largest eigenvalue of the system and the corresponding eigenmode. We will compute the maximum growth for a time horizon of $\tau = 1$, usually denoted $G(1)$.

> **Task 3.1**
>
> Configure the `bfs_tg.xml` session for performing transient growth analysis:
>
> - Set the `EvolutionOperator` to `TransientGrowth`.
>
> - Define a parameter `FinalTime` that is equal to 1 (this is the time horizon τ).
>
> - Set the number of steps (`NumSteps`) to be the ratio between the final time and the time step.
>
> - Since the simulations take several iterations to converge, use the restart file `bfs_tg.rst` for the initial condition. This file contains an eigenmode of the system.
>
> Now run the simulation
>
> ```
>  IncNavierStokesSolver bfs_tg.xml
> ```

The terminal screen should look like this:

```
=========================================================================
             EquationType: UnsteadyNavierStokes
             Session Name: bfs_tg
             Spatial Dim.: 2
        Max SEM Exp. Order: 7
```

```
              Expansion Dim.: 2
            Projection Type: Continuous Galerkin
                  Advection: explicit
                  Diffusion: explicit
                  Time Step: 0.002
              No. of Steps: 500
        Checkpoints (steps): 500
            Integration Type: IMEXOrder2
========================================================================
      Arnoldi solver type    : Arpack
      Arpack problem type    : LM
      Single Fourier mode    : false
      Beta set to Zero       : false
      Evolution operator     : TransientGrowth
      Krylov-space dimension : 4
      Number of vectors      : 1
      Max iterations         : 500
      Eigenvalue tolerance   : 1e-06
=======================================================
Initial Conditions:
Field p not found.
Field p not found.
  - Field u: from file bfs_tg.rst
  - Field v: from file bfs_tg.rst
  - Field p: from file bfs_tg.rst
Writing: "bfs_tg_0.chk"
        Inital vector       : input file
Iteration 0, output: 0, ido=1 Steps: 500     Time: 1     CPU Time: 10.4384s
Writing: "bfs_tg_1.chk"
Time-integration  : 10.4384s
Steps: 500      Time: 29            CPU Time: 8.96463s
Writing: "bfs_tg_1.chk"
Time-integration  : 8.96463s
Writing: "bfs_tg.fld"
Iteration 1, output: 0, ido=1 Steps: 500     Time: 2     CPU Time: 8.90168s
Writing: "bfs_tg_1.chk"
Time-integration  : 8.90168s
Steps: 500      Time: 30            CPU Time: 8.90607s
Writing: "bfs_tg_1.chk"
Time-integration  : 8.90607s
Iteration 2, output: 0, ido=1 Steps: 500     Time: 3     CPU Time: 8.96875s
Writing: "bfs_tg_1.chk"
Time-integration  : 8.96875s
Steps: 500      Time: 31            CPU Time: 8.92276s
Writing: "bfs_tg_1.chk"
Time-integration  : 8.92276s
Iteration 3, output: 0, ido=1 Steps: 500     Time: 4     CPU Time: 8.92597s
Writing: "bfs_tg_1.chk"
Time-integration  : 8.92597s
Steps: 500      Time: 32            CPU Time: 8.96103s
```

```
Writing: "bfs_tg_1.chk"
Time-integration  : 8.96103s
Iteration 4, output: 0, ido=99
Converged in 4 iterations
Converged Eigenvalues: 1
        Magnitude   Angle      Growth      Frequency
EV: 0 3.23586     0           1.1743        0
Writing: "bfs_tg_eig_0.fld"
L 2 error (variable u) : 0.0118694
L inf error (variable u) : 0.0118647
L 2 error (variable v) : 0.0174185
L inf error (variable v) : 0.0244285
L 2 error (variable p) : 0.0109063
L inf error (variable p) : 0.0138423
```

Initially, the solution will be evolved forward in time using the operator $\mathcal{A}$, then backward in time through the adjoint operator $\mathcal{A}^*$.

> **Task 3.2**
> Verify that the leading eigenvalue is equal to $\lambda = 3.23586$.

The leading eigenvalue corresponds to the largest possible transient growth at the time horizon $\tau = 1$. The leading eigenmode is shown in figures 3.1 and 3.2. This is the optimal initial condition which will lead to the greatest growth when evolved under the linearised Navier-Stokes equations. We can visualise graphically the optimal growth, recalling
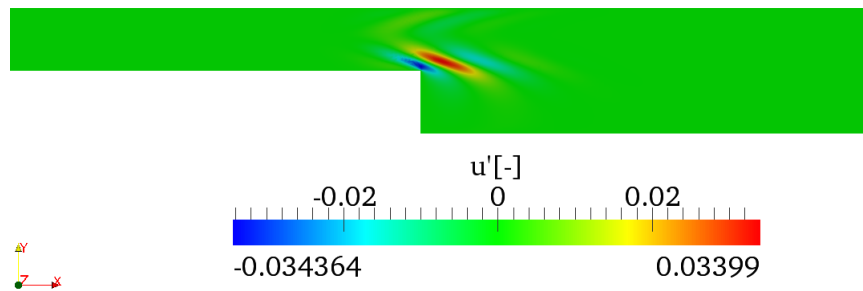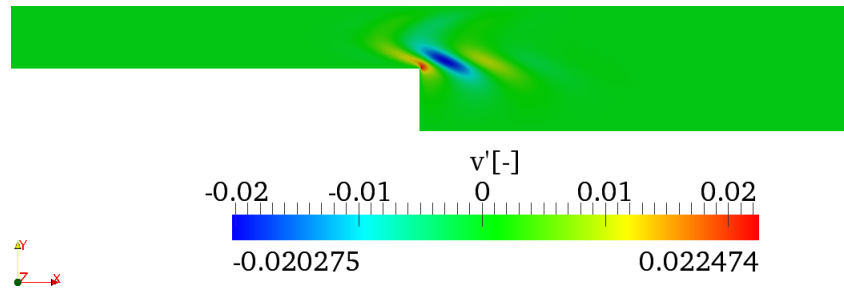

Figure 3.1 u'-component of the eigenmode

that the energy of the perturbation field at any given time t is defined by means of the inner product:

$$E(\tau) = \frac{1}{2}(\mathbf{u}'(t), \mathbf{u}'(t)) = \frac{1}{2} \int_\Omega \mathbf{u}' \cdot \mathbf{u}' dv \tag{3.1}$$

The solver can output the evolution of the energy of the perturbation in time by using the `ModalEnergy` filter (defined in the `FILTERS` section of the XML file):

Figure 3.2 v'-component of the eigenmode

```
1 <FILTER TYPE="ModalEnergy">
2     <PARAM NAME="OutputFile">energy</PARAM>
3     <PARAM NAME="OutputFrequency">10</PARAM>
4 </FILTER>
```

This will write the energy of the perturbation every 10 time steps to the file `energy.mld`. Repeating these simulations for different τ with the optimal initial perturbation as the initial condition, it is possible to create a plot similar to figure 3.3. Each curve necessarily meets the optimal growth envelope (denoted by the circles) at its corresponding value of τ, and never exceeds it.

The `$NEKTUTORIAL/energy` folder contains the files `bfs_energy_tau01.xml` and `bfs_energy_tau20.xml`, as well as the pre-computed optimal initial condition for $\tau = 20$ (`bfs_energy_tau20.rst`), with corresponding optimal growth of 2172.9.

Task 3.3

(Advanced/Optional) Generate energy curves for the optimal initial condition (leading eigenmode) computed in the previous task for $\tau = 1$, and for $\tau = 20$. Use your favourite plotting program (e.g. MATLAB or GNUPlot) to read in the files produced by the energy filter and plot the normalised energy growth curves.

Tip

You will need to switch to using the `Standard` driver. You should also use the `Direct` evolution operator for this task, similar to the channel example.

Examine your plot. Verify the energy at time $t = \tau$ matches the optimal growth in each case. Now examine the plot at time $t = 1$. Note that although the overall energy growth for the $\tau = 20$ curve is far greater than the corresponding $\tau = 1$ curve, the $\tau = 1$ curve has greater growth at $t = \tau = 1$.
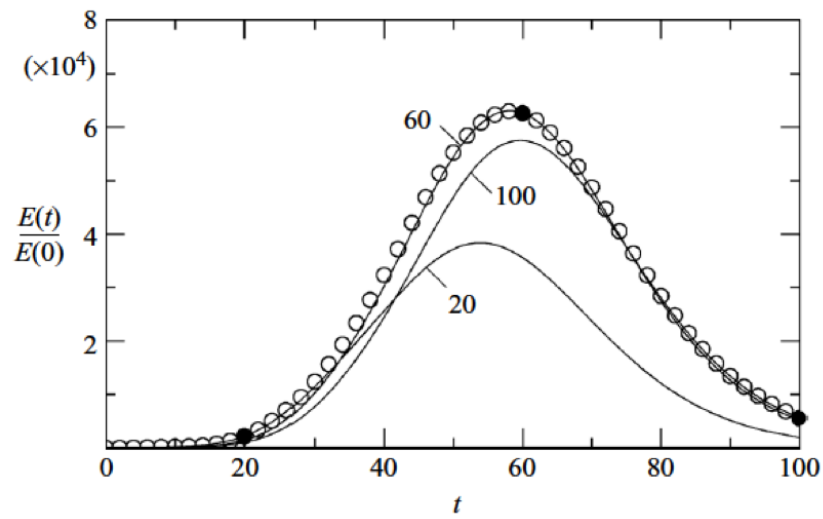
Figure 3.3 Envelope of two-dimensional optimal at $Re = 500$ together with curves of linear energy evolution starting from the three optimal initial conditions for specific values of τ 20, 60 and 100. Figure reproduced from J. Fluid. Mech. (2008), vol 603, *pp.* 271-304.