Worksheet n°4: Model reduction

1/ Globally stable flows: the case of cavity flow at Re=5000.

A/ Input-Output setup

We consider a flow over an open cavity of width and depth equal to 1. The mesh is defined in Mesh:

```
FreeFem++ mesh.edp
```

The Reynolds number based on the cavity length and upstream velocity is equal to $Re = 5000$. The base-flow, solution of $\frac{1}{2}\mathcal{N}(w_0, w_0) + \mathcal{L}w_0 = 0$, may be obtained in BF:

```
FreeFem++ newton.edp
```

For a closed-loop setup, we define an actuator and a sensor. The actuator is a forcing $f(t) = B_c u(t)$ which acts on the momentum equation. It is defined in folder Actuator:

```
FreeFem++ B.edp
```

The measurement $y = C_c w$ corresponds to shear-stress measurement near the downstream edge of the cavity.

B/ Impulse response

The dynamics of small-amplitude perturbations around the base-flow is governed by the linearized Navier-Stokes equations forced by the actuator, which read after spatial discretization:

$$d_t w = A_c w + B_c u(t)$$

$$y(t) = C_c w$$

Integrating over a sampling time Δt, we obtain the following discrete in time equations:

$$w_{k+1} = A_d w_k + B_d u_k$$

$$y_k = C_d w_k$$

(A_d, B_d, C_d) are three large-scale matrices. The dynamics from the input $u(t)$ to the output $y(t)$ is characterized by the impulse response. It is obtained in folder ROM_5000:

```
FreeFem++ impulsedns.edp

Octave plotlinlin('cavite.txt',1,3,1)    //    plot chosen input law u

Octave plotlinlin('cavite.txt',1,2,2)    //    plot resulting output y(t)
```

What is the temporal law u that is specified in this code?

C/ ERA algorithm

a/ In folder ROM_5000, complete program era.m to obtain the reduced-order model:

$$w_{k+1} = A_r w_k + B_r u_k$$

$$y_k = C_r w_k$$

Check the quality of the approximation by typing in ROM_5000:

```
Octave check.m          // plot large-scale and reduced impulse functions.
```

c/ Analyse the reduced-order model by typing in Octave:

```
sysrom=ss(Ar,Br,Cr,0,Ts);  // create state-space model with matrices Ar, Br and Cr with
                           sampling time Ts.

get(sysrom)     // List all components of structure sysrom

sysrom.a        // Matrix A of sysrom

sysrom.b        // Matrix B of sysrom

sysrom.c        // Matrix C of sysrom

sysrom.ts       // Time sample

ee=eig(Ar)      // Eigenvalues of Ar (should all be below 1 to insure stability of model)

figure(2)

plot(real(ee),imag(ee),'x')         // plot eigenvalues in z space

theta=linspace(0,2*pi,200);         // plot unit circle

hold on                             // plot unit circle

plot(cos(theta),sin(theta),'-');    // plot unit circle

[mag,phase,om]=bode(sysrom);             // create transfer function. How could you generate
                                         this data with an Octave code?

figure(3)

plot(om,mag);  // plot magnitude of transfer function. Why are there three peaks?

figure(4)

plot(om,phase); // plot phase of transfer function. Why is this curve decreasing monotically?

sysromc=d2c(sysrom);  // generate continuous-time state-space model which has is the
                      same as sysrom.

eee= eig(sysromc.a)   // eigenvalues of continuous-time model

figure(5)
```

```
plot(real(eee),imag(eee),'x')      // plot eigenvalues in z space
```

b/ Complete check.m to evaluate the error in 2-norm. Vary the order p of the reduced-order model (4th input in ROM_5000/data.txt) and plot these errors as a function of p

c/ Compute the lower and upper error-bounds for the error in $\infty -$ norm. Complete check.m to evaluate this error for a given p. Show that this curve lies within the theoretical error bounds.

d/ For $p = 10$, check that the infinite controllability and observability Gramians are diagonal and equal, with diagonal terms equal to those computed in the SVD performed in era.m. You may use the Octave function dlyap (Discrete Lyapunov equation solver) to compute the Gramians.

e/ Choose a sampling time which is ten times larger. Build the new reduced-order model and compare its transfer function to the previous one.

2/ Globally unstable flow: the case of cavity flow at Re=6250.

The flow at Re=6250 is globally unstable. The unstable direct and adjoint eigenmodes verify:

$$A_c\widehat{w} = \lambda\widehat{w}$$
$$\tilde{A}_c\widetilde{w} = \lambda^*\widetilde{w}$$
$$< \widetilde{w}, \widehat{w} > = 1$$

Where $<>$ is the usual energy-based inner-product and $\tilde{A}_c$ the adjoint of A_c.

These modes are computed in folder ROM_6250, by typing:

```
FreeFem++ eigen.edp
FreeFem++ eigenadj.edp
FreeFem++ norm.edp
```

To obtain a reduced-order model, the unstable and stable subspaces are treated separately: the stable subspace is reduced by ERA while the unstable subspace is modeled exactly using the direct and adjoint eigenmodes.

A/ Stable subspace

The impulse in the stable subspace is generated by a linearized Navier-Stokes simulation with an initial condition consisting of the actuator matrix B projected on the stable subspace:

$$B_c^p = B_c - (< \widetilde{w}, B_c > \widehat{w} + < \overline{\widetilde{w}}, B_c > \overline{\widehat{w}})$$

Here, we actually project B_c on the unstable subspace and remove this part from B_c.

The impulse response in the stable subspace may be obtained by typing in ROM_6250:

```
FreeFem++ impulsedns.edp
```

Then, the reduced-order model of the stable subspace is obtained in ROM_6250:

```
Octave era.m
Octave check.m          // check quality of model in stable subspace
```

B/ Unstable subspace

a/ In the subspace spanned by one unstable eigenvalue λ and its complex conjugate $\bar{\lambda}$, the continuous time dynamics is governed by:

$$d_t\begin{pmatrix} w \\ \overline{w} \end{pmatrix} = \begin{pmatrix} \lambda & 0 \\ 0 & \bar{\lambda} \end{pmatrix}\begin{pmatrix} w \\ \overline{w} \end{pmatrix} + \begin{pmatrix} < \widetilde{w}, B_c > \\ \overline{< \widetilde{w}, B_c >} \end{pmatrix} u$$
$$y = \begin{pmatrix} C_c\widehat{w} & \overline{C_c\widehat{w}} \end{pmatrix}\begin{pmatrix} w \\ \overline{w} \end{pmatrix}$$

Assuming that $u(t)$ is constant within one sampling time (zero-order hold), we may integrate this equation over one time-step:

$$\begin{pmatrix} w_{k+1} \\ \overline{w}_{k+1} \end{pmatrix} = \underbrace{e^{\begin{pmatrix} \lambda & 0 \\ 0 & \overline{\lambda} \end{pmatrix}\Delta t}}_{A_d} \begin{pmatrix} w_k \\ \overline{w}_k \end{pmatrix} + \underbrace{\left[\int_0^{\Delta t} e^{\begin{pmatrix} \lambda & 0 \\ 0 & \overline{\lambda} \end{pmatrix}(\Delta t-\tau)} d\tau\right] \begin{pmatrix} < \widetilde{w}, B_c > \\ \overline{< \widetilde{w}, B_c >} \end{pmatrix}}_{B_d} u_k$$

$$y_k = \underbrace{\begin{pmatrix} C_c\widehat{w} & \overline{C_c\widehat{w}} \end{pmatrix}}_{C_d} \begin{pmatrix} w_k \\ \overline{w}_k \end{pmatrix}$$

$$A_d = \begin{pmatrix} e^{\lambda\Delta t} & 0 \\ 0 & e^{\overline{\lambda}\Delta t} \end{pmatrix}, B_d = \begin{pmatrix} \dfrac{(e^{\lambda\Delta t}-1)}{\lambda} < \widetilde{w}, B_c > \\ \overline{\dfrac{(e^{\lambda\Delta t}-1)}{\lambda} < \widetilde{w}, B_c >} \end{pmatrix}, C_d = \begin{pmatrix} C_c\widehat{w} & \overline{C_c\widehat{w}} \end{pmatrix}$$

The second equation is the complex conjugate of the first equation to remain real. This equation involves complex scalar quantities in the various matrices. It may be rewritten only with real quantities by considering $\begin{pmatrix} w_{k+1}^{(r)} \\ w_{k+1}^{(i)} \end{pmatrix}$ as the state instead of $\begin{pmatrix} w_{k+1} \\ \overline{w}_{k+1} \end{pmatrix}$, where $w_{k+1} = w_{k+1}^{(r)} + iw_{k+1}^{(i)}$. Show that:

$$\begin{pmatrix} w_{k+1}^{(r)} \\ w_{k+1}^{(i)} \end{pmatrix} = A_r \begin{pmatrix} w_k^{(r)} \\ w_k^{(i)} \end{pmatrix} + B_r u_k$$

$$y_k = C_r \begin{pmatrix} w_k^{(r)} \\ w_k^{(i)} \end{pmatrix}$$

Where:

$$A_r = e^{\sigma\Delta t} \begin{pmatrix} \cos\omega\Delta t & -\sin\omega\Delta t \\ \sin\omega\Delta t & \cos\omega\Delta t \end{pmatrix}, B_r = \begin{pmatrix} \mathrm{Re}\left(\dfrac{(e^{\lambda\Delta t}-1)}{\lambda} < \widetilde{w}, B_c >\right) \\ \mathrm{Im}\left(\dfrac{(e^{\lambda\Delta t}-1)}{\lambda} < \widetilde{w}, B_c >\right) \end{pmatrix}$$

$$C_r = \begin{pmatrix} 2\mathrm{Re}(C\widehat{w}) & -2\mathrm{Im}(C\widehat{w}) \end{pmatrix}$$

In folder ROM_6250, the complete reduced-order model is obtained by:

```
FreeFem++ addunstabledyn.edp          // New model is stored in A.txt, B.txt, C.txt
```

b/ Compare the impulse response of the linearized DNS without projection to the impulse generated by the full reduced-order model. In folder ROM_6250,

```
FreeFem++ impulsedns_unstable.edp
Octave checku.m
```

c/ Analyse the transfer function of the full model using the bode function.

d/ Using the d2c Octave function (discrete to continuous), build a continuous-time equivalent system and analyse eigenvalues.

e/ Using the d2d Octave function (discrete to discrete), change the sampling time of the model.