Hugh Blackburn

Department of Mechanical and Aerospace Engineering, Monash University

Spectral element methods applied to incompressible Navier–Stokes and linearized Navier–Stokes problems

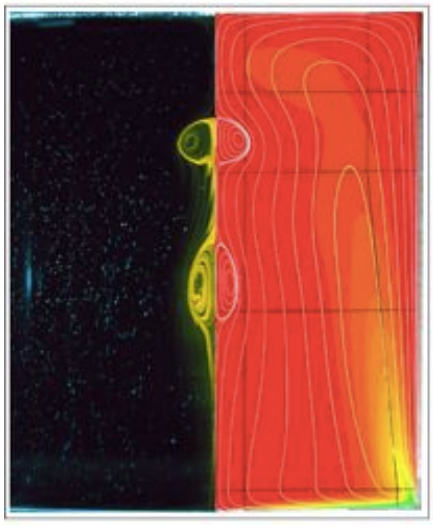
Semtex **Dog**

Source code and user guides (installation instructions) http://www.users.monash.edu.au/~bburn/semtex.html

Semtex **Dog**

Using *Semtex*

Working *Dog*



Design of *Semtex* and *Dog*

General

1. UNIX philosophy (a bunch of command-line tools rather than a big package with a GUI).
2. Mild object-oriented API (high-level extensions are not very hard).
3. Do just a few things well (it's not too big to understand).
4. Use a small number of standard libraries (BLAS, LAPACK, MPI, ARPACK).
5. Public domain (GNU licence).

Specific

1. Equal-order quadrilateral 2D elements, no adaptation or mortar patching;
2. Flat 1D storage for field variables, operators to manipulate them.
3. One variety of timestepping (stiffly stable/backward differencing);
4. Continuous Galerkin for elliptic sub-problems, with direct solution the norm;
5. 2D and 2½D: can do 3D flows in geometries that are extruded 2D assuming Fourier expansions in homogeneous direction;
6. Incompressible flows in Cartesian and cylindrical coordinates;
7. 2D element shape functions are tensor products of Lagrange interpolants through Gauss–Lobatto–Legendre points (makes mass matrices diagonal);
8. Equal-order for velocity and pressure spaces 'P_N–P_N';
9. A simplified XML-like description for session files, inbuilt function parser.

Some practical issues and guidance

Spectral element methods are GREAT for some problems but are not a panacea.

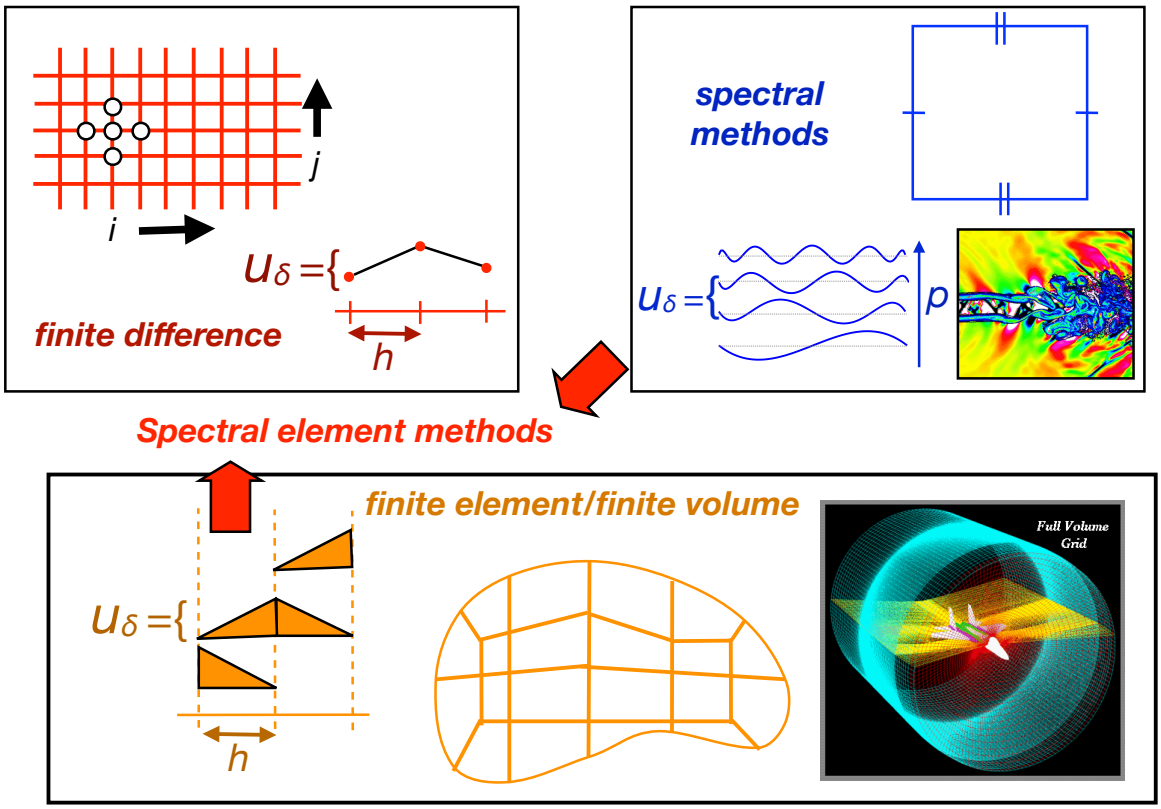
1. It is easy to vary the accuracy of solution by changing spectral element order ('*p*-refinement') BUT this is only a good idea once you have refined the element length scales adequately ('*h*-refinement'). Imagine that element sizes need adjustment until there are a minimum of 5 grid points per wave length. Once this point has been reached, solutions generally start to exhibit exponential convergence.
2. As for finite element methods, one can 'readily' locally refine where needed.
3. As for global spectral methods, continuous Galerkin spectral element methods are generally poor for non-smooth problems (e.g. with shocks). Can be overcome with discontinuous Galerkin methods(?).
4. It is quite easy to tell visually if you do not have enough resolution (especially by computing vorticity or otherwise taking derivatives). This initially seems like a weakness of the method ... but it's very useful.
5. Related: solutions are guaranteed C0 (continuous) at element boundaries but C1 (continuity of derivatives) is only obtained in the limit of resolution. Usually this is no problem for DNS but tends to mean there will be difficulties with methods which are explicitly under-resolved (like LES). It may be possible to overcome this.
6. For problems like stability analysis where accuracy and low dispersion/diffusion is important, spectral element methods are generally excellent.
7. The time-splitting we use gives fast execution and allows equal-order interpolation for velocity and pressure. However, solutions are not divergence-free except in the limit of resolution. Also, it is not possible to apply traction boundary conditions (limited to Dirichlet, Neumann and Robin BCs).
8. Time-split used is semi-implicit so there is usually a CFL timestep restriction.
9. As Reynolds numbers increase it often becomes faster (as well as less memory-hungry) to use iterative solution methods over direct solution methods for viscous Helmholtz parts of the timestep.

Semtex and *Dog*

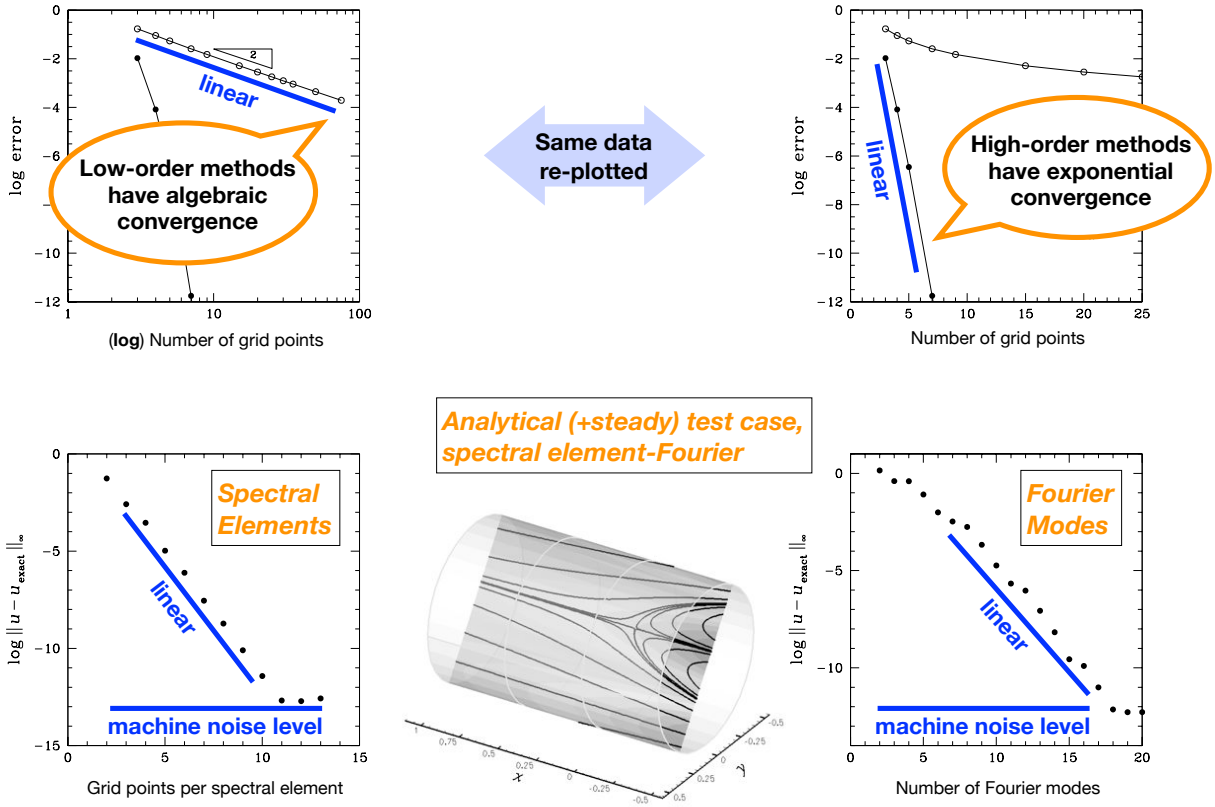
1. Quite an amount of overlap with *Nektar*++ in workflow, concepts, filenames. So there is a *session* file which is 'XML-lite', base flows are called *session*.bse, eigenvalue estimate file is called *session*.evl, eigenmodes are called *session*.eig.X. Etc.
2. One distinction/extra file not used by *Nektar*++ is a global numbering file called *session*.num. If it does not pre-exist, it will be automatically computed with a 'moderate' bandwidth optimisation. However it is good practice to use an enhanced optimisation and run the generation utility 'enumerate' by hand. e.g. `enumerate -O3 session > session.num`. You will see this done in a number of the examples.
3. As well as examples described in the *Dog* userguide, there are a number of other testcases in the `testcases` directory. These are briefly described in `testcases/README` file. Many of the testcases have a `testrun` script file which includes the commands that need to be run to produce the outcomes in the `README` file.
4. As a Floquet examplar, it is quicker to use the `testcases/cylinder3D` case than the `testcases/Floquet/cylinder` case – because a base flow restart file is supplied in the former. Discussion in guidedog.pdf is still relevant (mainly, the TOKENS Re and BETA differ between the two cases.)
5. Good idea to have at least these *Semtex* utilities located thru UNIX `PATH` variable:
   - enumerate
   - compare
   - dns
   - convert (beware of potential conflict with Imagemagick's `convert`)
   - sem2tec and preplot

Part 1. Spatial discretisation

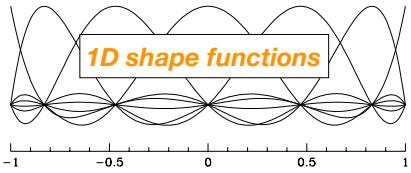
Mesh-based solution techniques for PDEs



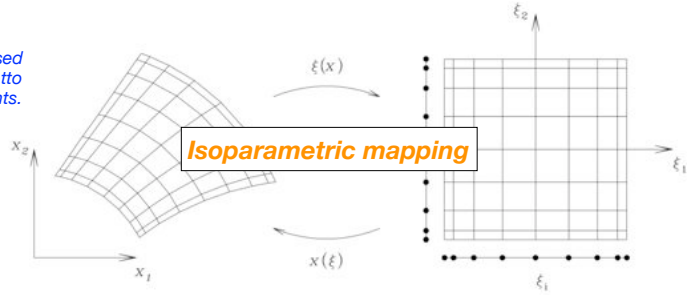
The idea of spectral (exponential) convergence

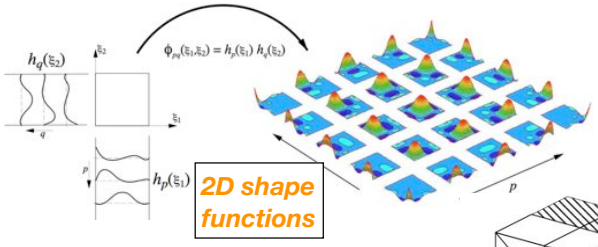


Spectral elements are high-order finite elements

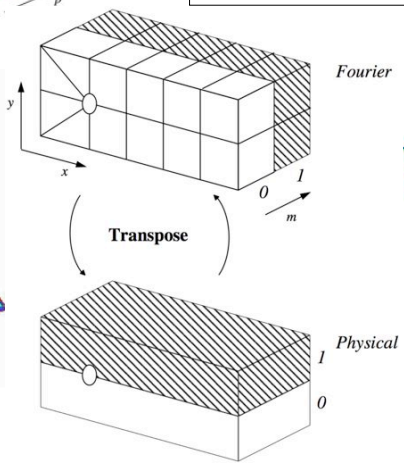


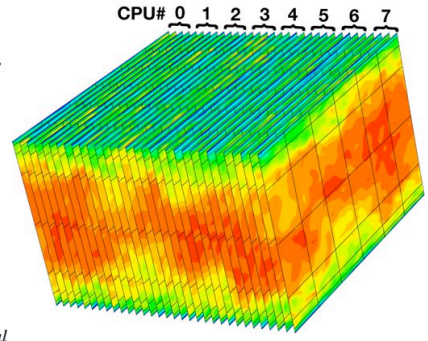
Lagrange interpolants based on Gauss–Lobatto quadrature points.



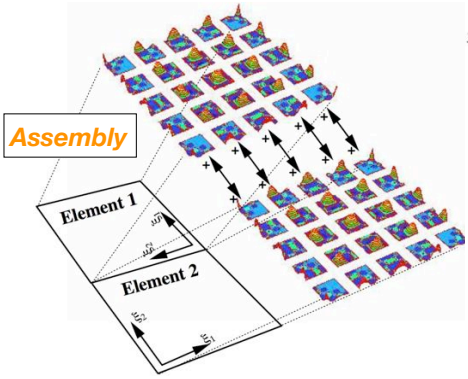


Fourier expansions for 3D capability



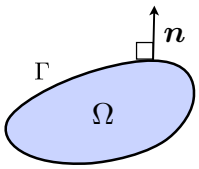


Simple concurrency across 2D Fourier modes



MONASH University Engineering

Galerkin MWR for elliptic problems

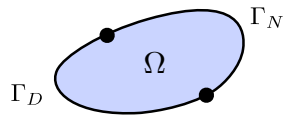


Elliptic scalar equations are all like $\nabla^2 v - \lambda^2 v = f$

Helmholtz problem to be solved inside a bounded domain

Standard Galerkin method involves multiplying by a weight function *w* and using IBP on ∇².

$$\int_\Omega \left(\boldsymbol{\nabla} v \cdot \boldsymbol{\nabla} w + \lambda^2 v w\right) \,\mathrm{d}\Omega = -\int_\Omega f w \,\mathrm{d}\Omega + \int_{\Gamma_N} h w \,\mathrm{d}\Gamma \qquad (h \equiv \partial_n v \equiv \boldsymbol{n} \cdot \boldsymbol{\nabla} \boldsymbol{v})$$



These terms only on Neumann-BC boundaries since *w* = 0 on Dirichlet boundaries.

Partition boundary into

- Γ_D, where **D**irichlet/essential BCs *g* are satisfied (and *w* = 0 is needed since we don't know $\partial_n v$ there)
- Γ_N, where **N**eumann/natural BCs *h* are satisfied.

Start the discretisation with **global** basis functions *N*.

$$u(\boldsymbol{x}) = \sum_{j=1}^{N} u_j \mathcal{N}_j(\boldsymbol{x}) = \sum_{j=1}^{M} u_j \mathcal{N}_j(\boldsymbol{x}) + \sum_{j=M+1}^{N} u_j \mathcal{N}_j(\boldsymbol{x})$$

unknown — given

Zero on Γ_D — Satisfy Dirichlet BCs *g* on Γ_D

$$w(\boldsymbol{x}) = \sum_{i=1}^{N} w_i \mathcal{N}_i(\boldsymbol{x}) \quad \text{WLOG let } w_i = 1.$$

$$\sum_{j=1}^{N} \int_\Omega u_j \left[\boldsymbol{\nabla}\mathcal{N}_j(\boldsymbol{x}) \cdot \boldsymbol{\nabla}\mathcal{N}_i(\boldsymbol{x}) + \lambda^2 \mathcal{N}_j(\boldsymbol{x}) \mathcal{N}_i(\boldsymbol{x})\right] \,\mathrm{d}\Omega = -\int_\Omega f_j \mathcal{N}_j(\boldsymbol{x}) \mathcal{N}_i(\boldsymbol{x}) \,\mathrm{d}\Omega + \int_{\Gamma_N} h_j \mathcal{N}_j(\boldsymbol{x}) \mathcal{N}_i(\boldsymbol{x}) \,\mathrm{d}\Gamma$$

One equation for each **global** weight w_i.

MONASH University Engineering

Galerkin MWR for elliptic problems

$$\sum_{j=1}^{N} u_j \int_\Omega \left[\boldsymbol{\nabla}\mathcal{N}_j(\boldsymbol{x})\cdot\boldsymbol{\nabla}\mathcal{N}_i(\boldsymbol{x}) + \lambda^2\mathcal{N}_j(\boldsymbol{x})\mathcal{N}_i(\boldsymbol{x})\right]\mathrm{d}\Omega \equiv \sum_{j=1}^{N} H_{ij}u_j = -\int_\Omega f_j\mathcal{N}_j(\boldsymbol{x})\mathcal{N}_i(\boldsymbol{x})\,\mathrm{d}\Omega + \int_{\Gamma_N} h_j\mathcal{N}_j(\boldsymbol{x})\mathcal{N}_i(\boldsymbol{x})\,\mathrm{d}\Gamma$$

'Stiffness' 'Mass' 'Helmholtz'

We only need the first *M* rows of this statement, corresponding to the unknown values of *u*.

Equivalent matrix statement:

$$\begin{matrix} & \xrightarrow{j} & M & N \\ i\downarrow & & & \\ M & & & \end{matrix}\left[\begin{array}{c|c} H_{uu} & H_{ug} \end{array}\right]\left\{\begin{matrix} u \\ u_g \end{matrix}\right\} = -\left[\begin{array}{c} M_{uu} \end{array}\right]\left\{f\right\} + \left\{h\right\}$$

These are the given Dirichlet boundary data → $\{u_g\}$

Forcing from Neumann BCs

Rearrange:

$$\left[H_{uu}\right]\{u\} = -\left[M_{uu}\right]\{f\} + \{h\} - \left[H_{ug}\right]\{u_g\}$$

Banded symmetric *Diagonal*

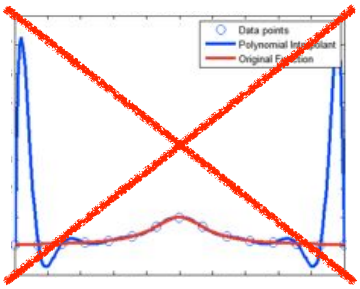
Global Helmholtz matrix **Global Mass matrix**

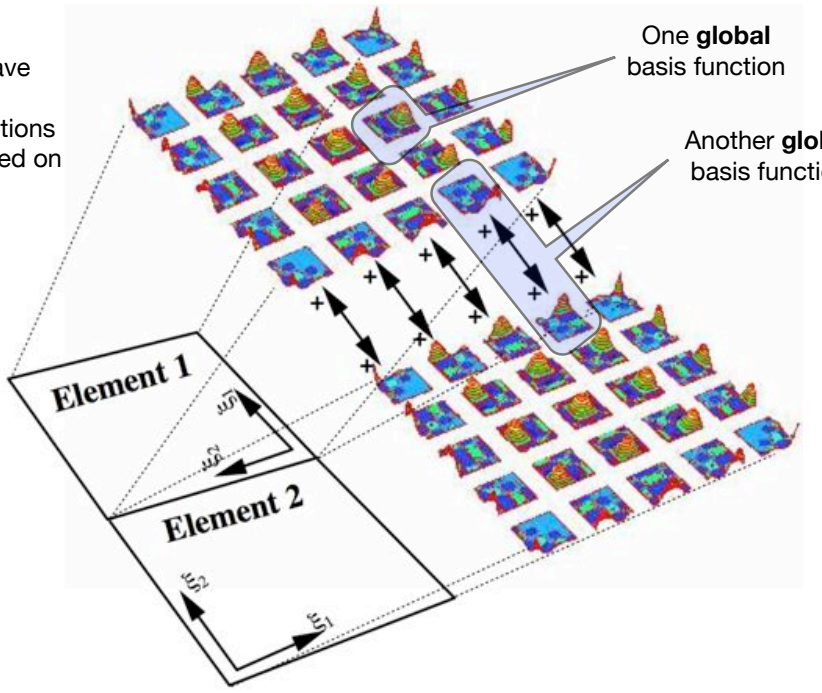
Global ← Local

All statements so far were made in terms of global basis functions (and without considering discretisation).

Finite element idea:
- use global basis functions that have local support
- assemble (sum) global basis functions from shape functions that are defined on sub-domains (elements).

Low-order finite elements typically employ tensor products of first- or second-order shape functions.





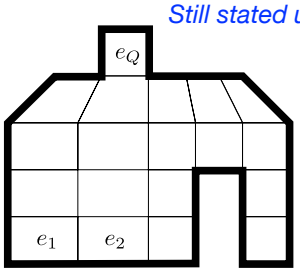
Spectral elements employ higher-order shape functions that do not suffer from Runge's phenomenon and are guaranteed to improve their match to smooth functions as interpolation order increases.

In almost all other respects, spectral elements are identical to finite elements.

Discretisation of elliptic MWR via finite/spectral elements

$$\sum_{j=1}^{N} u_j \int_\Omega \left[\boldsymbol{\nabla}\mathcal{N}_j(\boldsymbol{x})\cdot\boldsymbol{\nabla}\mathcal{N}_i(\boldsymbol{x}) + \lambda^2\mathcal{N}_j(\boldsymbol{x})\mathcal{N}_i(\boldsymbol{x})\right]\,\mathrm{d}\Omega \equiv \sum_{j=1}^{N} H_{ij}u_j = -\int_\Omega f_j\mathcal{N}_j(\boldsymbol{x})\mathcal{N}_i(\boldsymbol{x})\,\mathrm{d}\Omega + \int_{\Gamma_N} h_j\mathcal{N}_j(\boldsymbol{x})\mathcal{N}_i(\boldsymbol{x})\,\mathrm{d}\Gamma$$

Still stated using global basis functions.



Partition domain into sub-domains (elements), use integral(sum) = sum(integrals).

global = sum (**local**)

$$\int_\Omega (\,)\,\mathrm{d}\Omega = \sum_{e=1}^{Q}\int_{\Omega_e}(\,)\,\mathrm{d}\Omega_e$$

Elemental contributions to global Helmholtz matrix

$$H^e_{ab} = \int_\Omega \left[\boldsymbol{\nabla}\mathcal{N}_b(\boldsymbol{x})\cdot\boldsymbol{\nabla}\mathcal{N}_a(\boldsymbol{x}) + \lambda^2\mathcal{N}_b(\boldsymbol{x})\mathcal{N}_a(\boldsymbol{x})\right]\,\mathrm{d}\Omega_e$$

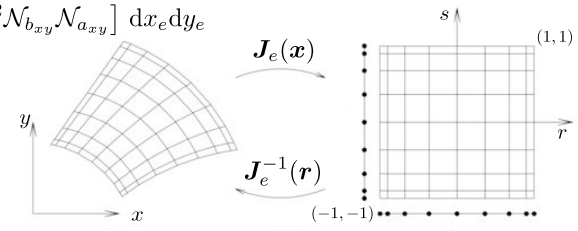
Mesh nodes: Gauss–Lobatto quadrature points.

In 2D we could write this as $\int_y\int_x \left[\boldsymbol{\nabla}\mathcal{N}_{b_{xy}}\cdot\boldsymbol{\nabla}\mathcal{N}_{a_{xy}} + \lambda^2\mathcal{N}_{b_{xy}}\mathcal{N}_{a_{xy}}\right]\,\mathrm{d}x_e\mathrm{d}y_e$

To perform quadratures we map to standard region [-1, 1] × [-1, 1], with

$$\boldsymbol{J}_e(\boldsymbol{x})$$

being the Jacobian of the mapping in each element.



E.g. in 2D, this is $\boldsymbol{J}_e(\boldsymbol{x}) = \begin{bmatrix}\partial_r x & \partial_r y\\ \partial_s x & \partial_s y\end{bmatrix}$. Entries are typically computed isoparametrically.

Elemental Helmholtz matrix

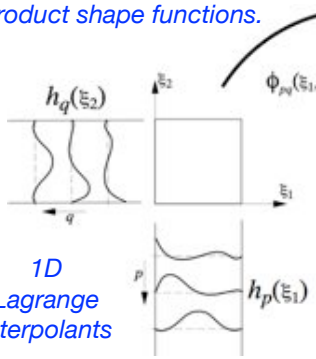
$$H^e_{ab} = \int_{-1}^{1}\int_{-1}^{1}\left[\boldsymbol{J}_e^{-1}\boldsymbol{\nabla}\mathcal{N}_{b_{rs}}\cdot\boldsymbol{J}_e^{-1}\boldsymbol{\nabla}\mathcal{N}_{a_{rs}}|\boldsymbol{J}_e| + \lambda^2\mathcal{N}_{b_{rs}}\mathcal{N}_{a_{rs}}|\boldsymbol{J}_e|\right]\,\mathrm{d}r\mathrm{d}s$$

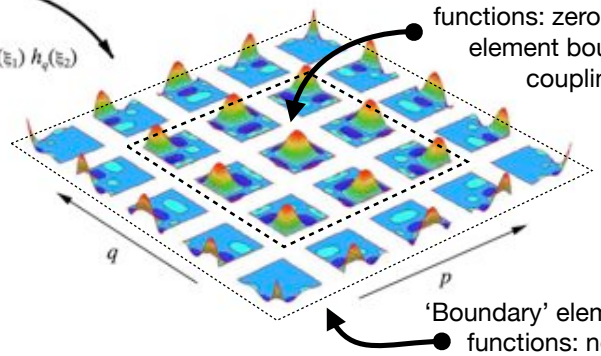
Integrals are approximated via Gauss–Lobatto quadratures.

Exploiting high-order interpolation

Tensor product shape functions.

1D Lagrange interpolants ⊗ *1D Lagrange interpolants*





'Interior' element shape functions: zero all around element boundary. No coupling to other elements.

'Boundary' element shape functions: non-zero on element boundary, sum into global shape functions.

Static condensation (a.k.a. Schur complement decomposition, substructuring)

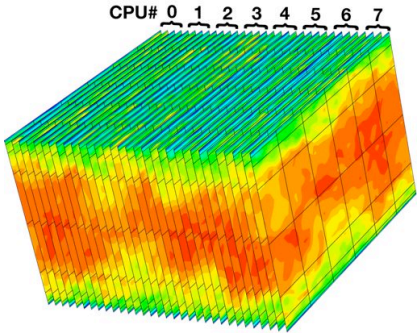
Based on element Hemholtz matrix partitioning; only boundary nodal values need contribute to global Helmholtz matrix. Reduces size of assembled global matrix if direct solution (Cholesky decomposition) is to be used. Can be used recursively.

Then use back-substitution to obtain element-interior nodal values element-by-element.

Tensor product

For iterative (PCG) solutions of global matrix problem, reduce number of operations required for a elemental matrix-vector product (e.g. from N^4 to N^3 in 2D) by exploiting tensor product structure.

3D (or 2½D) using Fourier spanwise



If domain is homogenous in *z* direction we can employ Fourier transforms.

2D complex Fourier modes $$\widehat{\boldsymbol{u}}_k(x,y,t) = L_z^{-1}\int_0^{L_z} \boldsymbol{u}(x,y,z,t)\mathrm{e}^{-\mathrm{i}(2\pi/L_z)kz}\,\mathrm{d}z$$

2D real data $$\boldsymbol{u}(x,y,z,t) = \sum_{k=-N/2}^{(N/2)-1} \widehat{\boldsymbol{u}}_k(x,y,t)\mathrm{e}^{\mathrm{i}(2\pi/L_z)kz}$$

Here *N* is the number of Fourier modes employed –
half the number of 2D *z*-planes of data in physical space.

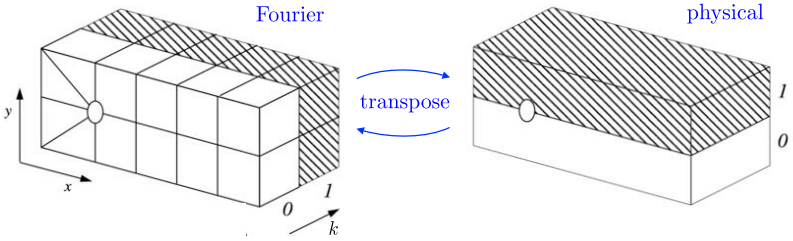
The fundamental wavenumber is $\beta = 2\pi/L_z$

The Fourier transform commutes with all the linear operators in the Navier–Stokes equations which become

$$\partial_t \widehat{\boldsymbol{u}}_k = -\widehat{\boldsymbol{N}(\boldsymbol{u})}_k - \boldsymbol{\nabla}\widehat{P}_k + \nu\nabla^2\widehat{\boldsymbol{u}}_k$$

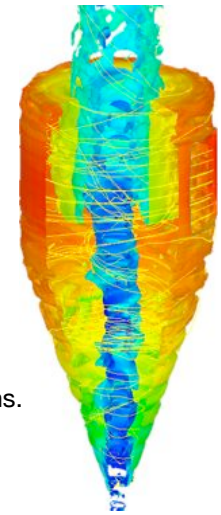
Each of the 2D complex problems can be solved on its own, and assigned to a single process for time evolution.

The exception is the formation of the nonlinear terms which are typically evaluated pseudospectrally, involving iDFT/DFT pairs and transpose across processes by message passing.



The majority of substeps in one timestep are computed in Fourier space.

Cylindrical coordinates – Fourier in azimuth

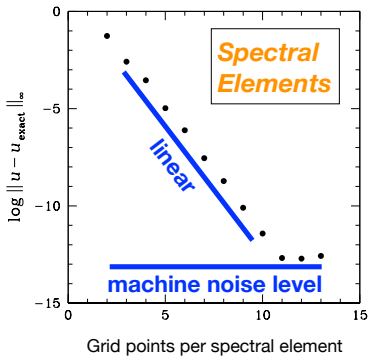


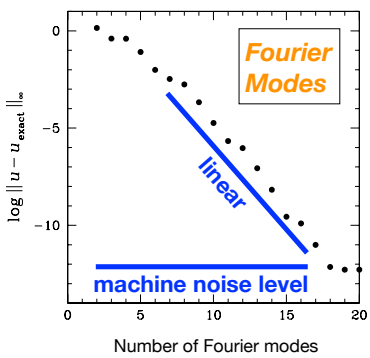
Cylindrical coordinates always present a problem at the axis, which is a singularity.

However, with the introduction of

1. Diagonalization: introduce coupled radial + azimuthal velocities to decouple viscous terms otherwise present;
2. Symmetrization: multiply momentum equations through by *r* ;
3. Galerkin MWR for elliptic problems (as we already have done)
4. Careful choice of (mode-dependent) boundary conditions at the axis;

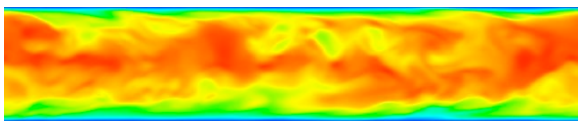
(and with no other changes), it is possible to obtain spectral convergence in all directions.





Blackburn & Sherwin
*JCP **197** (2004)*

This does not necessarily mean that other (CFL-type) problems do not arise.

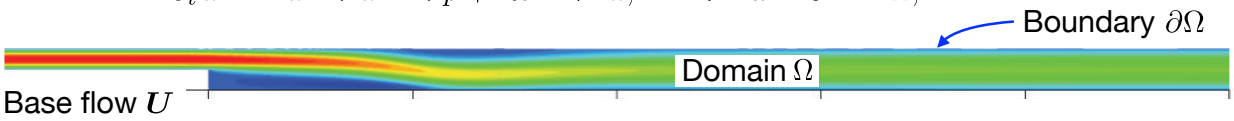


Slice through a pipe flow DNS:
no axis artefacts.

Part 2. Temporal discretisation and timestepper methods applied to eigenvalue (stability) problems

(Linearized) Navier–Stokes

$$\partial_t \boldsymbol{u} = -\boldsymbol{u}\cdot\boldsymbol{\nabla}\boldsymbol{u} - \boldsymbol{\nabla} p + Re^{-1}\nabla^2\boldsymbol{u}, \qquad \boldsymbol{\nabla}\cdot\boldsymbol{u} = 0 \quad \text{in } \Omega,$$



Decompose as $\boldsymbol{u} = \boldsymbol{U} + \boldsymbol{u}'$ where $\boldsymbol{u}'$ is a small/linear perturbation to **base flow** $\boldsymbol{U}$.

Likewise for pressure: $p = P + p'$. $\boldsymbol{U}$ is not necessarily steady in time.

Substitute

$$\partial_t(\boldsymbol{U}+\boldsymbol{u}') = -(\boldsymbol{U}+\boldsymbol{u}')\cdot\boldsymbol{\nabla}(\boldsymbol{U}+\boldsymbol{u}') - \boldsymbol{\nabla}(P+p') + Re^{-1}\nabla^2(\boldsymbol{U}+\boldsymbol{u}'),$$

Expand

$$\partial_t(\boldsymbol{U}+\boldsymbol{u}') = -\boldsymbol{U}\cdot\boldsymbol{\nabla}\boldsymbol{U} - \boldsymbol{U}\cdot\boldsymbol{\nabla}\boldsymbol{u}' - \boldsymbol{u}'\cdot\boldsymbol{\nabla}\boldsymbol{U} - \boldsymbol{u}'\cdot\boldsymbol{\nabla}\boldsymbol{u}' - \boldsymbol{\nabla}(P+p') + Re^{-1}\nabla^2(\boldsymbol{U}+\boldsymbol{u}'),$$

Split into an equation for base flow and an equation for perturbation:

$$\partial_t \boldsymbol{U} = -\boldsymbol{U}\cdot\boldsymbol{\nabla}\boldsymbol{U} - \boldsymbol{\nabla}P + Re^{-1}\nabla^2\boldsymbol{U},$$
$$\partial_t \boldsymbol{u}' = -\boldsymbol{U}\cdot\boldsymbol{\nabla}\boldsymbol{u}' - \boldsymbol{u}'\cdot\boldsymbol{\nabla}\boldsymbol{U} - \boldsymbol{u}'\cdot\boldsymbol{\nabla}\boldsymbol{u}' - \boldsymbol{\nabla}p' + Re^{-1}\nabla^2\boldsymbol{u}'$$

LNSE $\quad \partial_t \boldsymbol{u}' = -\left[\boldsymbol{U}\cdot\boldsymbol{\nabla} + (\boldsymbol{\nabla}\boldsymbol{U})^{\mathrm{T}}\cdot\right]\boldsymbol{u}' - \boldsymbol{\nabla}p' + Re^{-1}\nabla^2\boldsymbol{u}'$

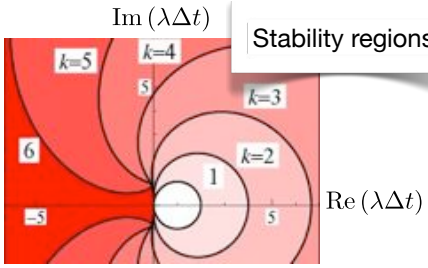
Temporal discretisation in *Semtex*

$$\partial_t \boldsymbol{u} = -\boldsymbol{u}\cdot\boldsymbol{\nabla}\boldsymbol{u} - \boldsymbol{\nabla}P + \nu\nabla^2\boldsymbol{u} \qquad \text{with} \quad \boldsymbol{\nabla}\cdot\boldsymbol{u} = 0 \quad \text{or} \quad \nabla^2 P = -\boldsymbol{\nabla}\cdot(\boldsymbol{u}\cdot\boldsymbol{\nabla}\boldsymbol{u}) \qquad P \equiv p/\rho$$

$$\partial_t \boldsymbol{u} = -\boldsymbol{N}(\boldsymbol{u}) - \boldsymbol{\nabla}P + \nu\nabla^2\boldsymbol{u}$$

Temporal discretisation uses 'stiffly stable' integration.

$$\partial_t u^{(n+1)} \approx \frac{1}{\Delta t}\sum_{q=0}^{k} \alpha_q u^{(n+1-q)}$$ (*k* = 1 is backward Euler.)



Hairer & Wanner (2010)

Generally the region of stability shrinks with increasing *k*.

With *k* = 2 the method is A-stable.

With *k* = 3 the stable region includes almost whole left-half plane.

Generally *k* = 2 provides a good balance between CFL stability and spatio-temporal accuracy.

Operator splitting to solve at time level (n+1) with error $O(\Delta t)^k$

$$\partial_t \boldsymbol{u}^{(n+1)} = -\boldsymbol{N}(\boldsymbol{u})^{(n+1)} - \boldsymbol{\nabla}P^{(n+1)} + \nu\nabla^2\boldsymbol{u}^{(n+1)}$$ via explicit extrapolation: $$\boldsymbol{N}(\boldsymbol{u})^{(n+1)} \approx \sum_{q=0}^{k-1}\beta_q \boldsymbol{N}(\boldsymbol{u}^{(n-q)})$$

1. $\boldsymbol{u}^* = -\sum_{q=1}^{k}\alpha_q \boldsymbol{u}^{(k-q)} - \Delta t\sum_{q=0}^{k-1}\beta_q \boldsymbol{N}(\boldsymbol{u}^{(n-q)})$
2. $\nabla^2 P^{(n+1)} = (\Delta t)^{-1}\boldsymbol{\nabla}\cdot\boldsymbol{u}^*$ + BC: $\partial_n P^{(n+1)} = -\boldsymbol{n}\cdot\sum_{q=0}^{k}\beta_q\left(\boldsymbol{N}(\boldsymbol{u}^{(n-q)}) + \nu\boldsymbol{\nabla}\times\boldsymbol{\nabla}\times\boldsymbol{u}^{(n-q)} + \partial_t\boldsymbol{u}^{(n-q)}\right)$
3. $\boldsymbol{u}^{**} = \boldsymbol{u}^* - \Delta t\boldsymbol{\nabla}P^{(n+1)}$
4. $\nabla^2\boldsymbol{u}^{(n+1)} - \frac{\alpha_0}{\nu\Delta t}\boldsymbol{u}^{(n+1)} = -\frac{\boldsymbol{u}^{**}}{\nu\Delta t}$ +BCs as appropriate

Dominant workload per step
1. Nonlinear terms
2. (Linear) elliptic equations (3 or 4)

A black-box view of time stepping the NSE

Continuum

$$\partial_t \boldsymbol{u} = -\boldsymbol{u}\cdot\boldsymbol{\nabla}\boldsymbol{u} - \boldsymbol{\nabla}p + \nu\nabla^2\boldsymbol{u} \qquad \text{with} \quad \boldsymbol{\nabla}\cdot\boldsymbol{u} = 0$$

Laziness: *p* used in place of *p*/ρ from now on.

Pressure is not an independent variable in incompressible flows $p \equiv \nabla^{-2}\boldsymbol{\nabla}\cdot(\boldsymbol{u}\cdot\boldsymbol{\nabla}\boldsymbol{u})$

$$\partial_t \boldsymbol{u} = -[\boldsymbol{I} - \boldsymbol{\nabla}\nabla^{-2}\boldsymbol{\nabla}\cdot](\boldsymbol{u}\cdot\boldsymbol{\nabla}\boldsymbol{u}) + \nu\nabla^2\boldsymbol{u}$$
$$= \mathcal{N}(\boldsymbol{u}) + \mathcal{L}\boldsymbol{u}$$

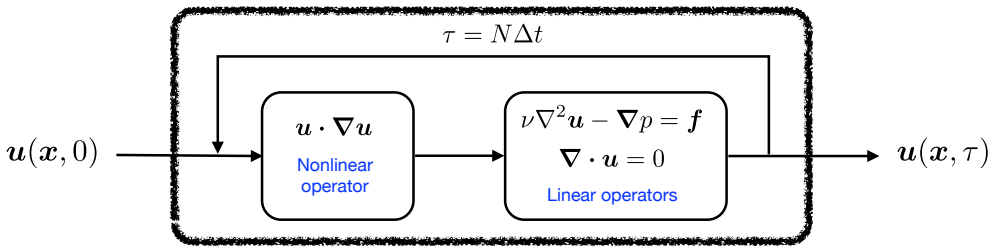
Nonlinear operator — Linear operator

$$\boldsymbol{u}(\boldsymbol{x},0) \longrightarrow \boxed{\partial_t\boldsymbol{u} - \mathcal{N}(\boldsymbol{u}) - \mathcal{L}\boldsymbol{u} = 0} \longrightarrow \boldsymbol{u}(\boldsymbol{x},\tau)$$

Nonlinear state transition operator

Temporal discretisation

Tuckerman & Barkley (2000)



Shorthand notations for LNSE and its EVP

Recall $\partial_t \boldsymbol{u}' = -\left[\boldsymbol{U}\cdot\boldsymbol{\nabla} + (\boldsymbol{\nabla}\boldsymbol{U})^{\mathrm{T}}\cdot\right]\boldsymbol{u}' - \boldsymbol{\nabla}p' + Re^{-1}\nabla^2\boldsymbol{u}'$.

Pressure is a constraint field tied to velocity through $\boldsymbol{\nabla}\cdot\boldsymbol{u}' = 0$.

$$p' \equiv \nabla^{-2}\boldsymbol{\nabla}\cdot\left[\boldsymbol{U}\cdot\boldsymbol{\nabla} + (\boldsymbol{\nabla}\boldsymbol{U})^{\mathrm{T}}\cdot\right]\boldsymbol{u}'$$

so, symbolically we may just deal with evolution of the velocity:

$$\partial_t \boldsymbol{u}' = -\left[\boldsymbol{I} - \boldsymbol{\nabla}\nabla^{-2}\boldsymbol{\nabla}\cdot\right]\left[\boldsymbol{U}\cdot\boldsymbol{\nabla} + (\boldsymbol{\nabla}\boldsymbol{U})^{\mathrm{T}}\cdot\right]\boldsymbol{u}' + Re^{-1}\nabla^2\boldsymbol{u}',$$

and arrive at the linear evolution equation $\partial_t \boldsymbol{u}' = L\boldsymbol{u}'$ or $\partial_t \boldsymbol{u}' - L\boldsymbol{u}' = 0$. **LNSE**

Assuming a separation-of-variables form $\boldsymbol{u}'(\boldsymbol{x},t) = \exp(\lambda_j t)\tilde{\boldsymbol{u}}_j(\boldsymbol{x}) + \text{c.c.}$

leads to the eigenvalue problem $L\tilde{\boldsymbol{u}}_j = \lambda_j \tilde{\boldsymbol{u}}_j$ **EVP**

For evolution over time interval τ we use the state transition operator $\mathcal{M}(\tau)$.

$$\mathcal{M}(\tau) = \exp\int_{t=0}^{t=\tau} L(t)\,\mathrm{d}t \qquad \text{If the flow is steady this is} \quad \mathcal{M}(\tau) = \exp L\tau$$

$$\boldsymbol{u}'(\tau) = \mathcal{M}(\tau)\boldsymbol{u}'(0) \qquad \text{equivalently} \quad \boldsymbol{u}'_\tau = \mathcal{M}(\tau)\boldsymbol{u}'_0$$

This may be applied by integrating (time stepping) the LNSE

$$\boldsymbol{u}'_0 \longrightarrow \boxed{\begin{array}{c}\partial_t \boldsymbol{u}' - L\boldsymbol{u}' = 0\\ \text{integrate forwards}\end{array}} \longrightarrow \boldsymbol{u}'_\tau \quad \text{= apply LNSE =} \quad \boldsymbol{u}'_\tau = \mathcal{M}(\tau)\boldsymbol{u}'_0$$

Large-time (asymptotic) linear stability of steady flow

The eigensystem expansion assumes $\boldsymbol{u}'(\boldsymbol{x},t) = \exp(\lambda_j t)\tilde{\boldsymbol{u}}_j(\boldsymbol{x}) + \text{c.c.}$

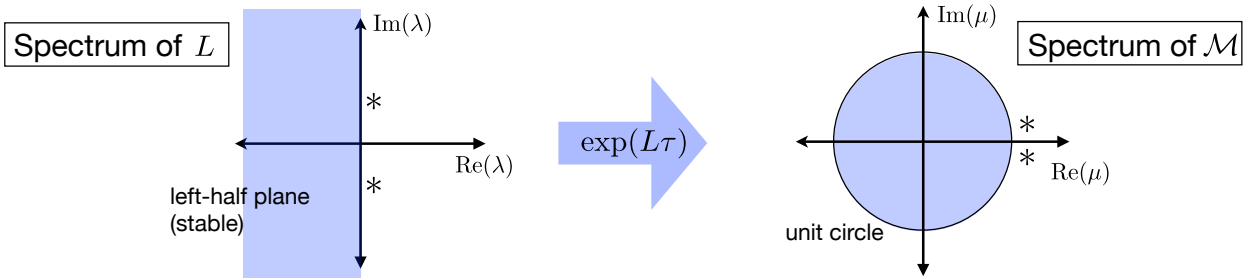
or equivalently $\boldsymbol{u}'(\boldsymbol{x},t+\tau) = \mu_j\tilde{\boldsymbol{u}}_j(\boldsymbol{x}) + \text{c.c.}$

L and $\mathcal{M}(\tau)$ have directly related eigensystems since $\exp L\tau = \lim_{N\to\infty}\sum_{k=0}^{N}\frac{(L\tau)^k}{k!}$

Supposing $\tilde{\boldsymbol{u}}_j$ is an eigenvector of L with corresponding eigenvalue λ_j

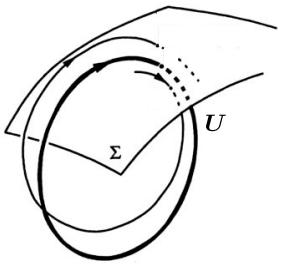
$$\mathcal{M}\tilde{\boldsymbol{u}}_j = \exp L\tau\,\tilde{\boldsymbol{u}}_j = \lim_{N\to\infty}\sum_{k=0}^{N}\frac{(L\tau)^k}{k!}\tilde{\boldsymbol{u}}_j = \lim_{N\to\infty}\sum_{k=0}^{N}\frac{(\lambda_j\tau)^k}{k!}\tilde{\boldsymbol{u}}_j = \exp(\lambda_j\tau)\tilde{\boldsymbol{u}}_j$$

i.e. $\tilde{\boldsymbol{u}}_j$ is also an eigenvector of $\mathcal{M}$ and the corresponding eigenvalue is $\mu_j = \exp\lambda_j\tau$.



It is more convenient numerically to search for dominant eigenvalues of $\mathcal{M}(\tau)$ than the most unstable eigenvalues of L. **Either set gives the large-time behaviour.**

Floquet stability (asymptotic)



If the base flow is time-periodic (with period τ) there are many similarities but some differences.

$$\boldsymbol{U}(t_0+\tau)=\boldsymbol{U}(t_0)$$

We want to know how (linear) perturbations vary from one period to the next.

Again $\boldsymbol{u}'(\tau)=\mathcal{M}(\tau)\boldsymbol{u}'(0)$

but now $\mathcal{M}(\tau)=\exp\int_{t=0}^{t=\tau}L(t)\,\mathrm{d}t$ For periodic analyses, state transition operator a.k.a. monodromy operator.

We still have $\boldsymbol{u}'_0 \longrightarrow$ $\partial_t\boldsymbol{u}'-L\boldsymbol{u}'=0$ integrate forwards $\longrightarrow \boldsymbol{u}'_\tau$ but integration interval is fixed to τ.

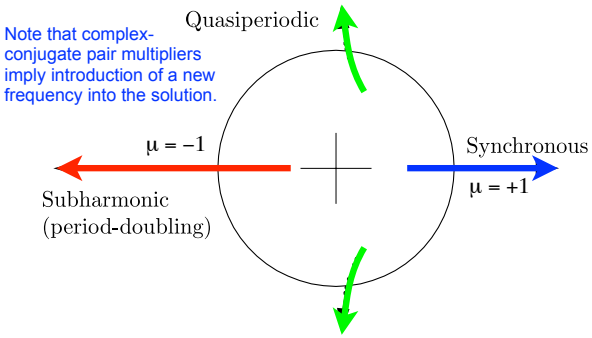
This defines a Floquet problem whose solutions are

$$\boldsymbol{u}'(\boldsymbol{x},t_0+\tau)=\exp(\lambda_j\tau)\,\widetilde{\boldsymbol{u}}_j(\boldsymbol{x},t_0)+\text{c.c.}\equiv\mu_j\,\widetilde{\boldsymbol{u}}_j(\boldsymbol{x},t_0)+\text{c.c.}$$

where μ_j are Floquet multipliers and $\widetilde{\boldsymbol{u}}_j(\boldsymbol{x},t_0)$ are (τ-periodic) Floquet modes.

These are respectively eigenvalues and eigenmodes of $\mathcal{M}(\tau)$.

Note that the problem is now inherently discrete-time in nature, so we are only concerned with how the multipliers relate to (and cross) the unit circle in the complex plane.



Timestepper approach to eigensystems

Outer loop: based on repeated application of operator $\mathcal{M}$ on an initial vector.

1. Generate a Krylov subspace *T* of dimension *N* × *k* (where *N* >> *k*) by repeated application of $\mathcal{M}$ via inner loop:

$$T=\{\boldsymbol{u}'_0,\mathcal{M}\boldsymbol{u}'_0,\mathcal{M}^2\boldsymbol{u}'_0,\ldots,\mathcal{M}^{k-1}\boldsymbol{u}'_0\}$$

$\boldsymbol{u}'_0 \longrightarrow$ $\partial_t\boldsymbol{u}'-L\boldsymbol{u}'=0$ integrate forwards $\longrightarrow \boldsymbol{u}'_\tau = \mathcal{M}\boldsymbol{u}'_0$

2. QR factorize matrix *T*

$$T=QR$$

3. Calculate (*k*×*k*) Hessenberg matrix *H* from *R*

$$h_{i,j}=\frac{1}{r_{j,j}}\left(r_{i,j+1}-\sum_{l=0}^{j-1}h_{i,l}r_{l,j}\right)$$

This ('Barkley') methodology is written up in Barkley, Blackburn & Sherwin IJNMF **57** (2008).

4. Calculate eigensystem of *H* in *k* × *k* subspace (e.g. LAPACK).
5. If converged, stop and project back to full space, else discard oldest vector in *T*, carry out one more integration of $\mathcal{M}$, go to step 2.

Implicitly-restarted Arnoldi method (ARPACK) gives similar performance.

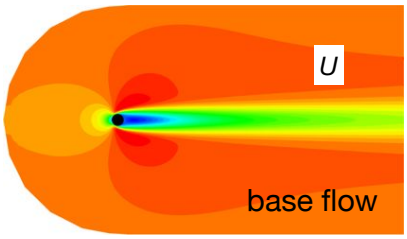
We can find dominant eigenvalues of an operator without constructing it.

(But: not so good if we want a lot of the spectrum).

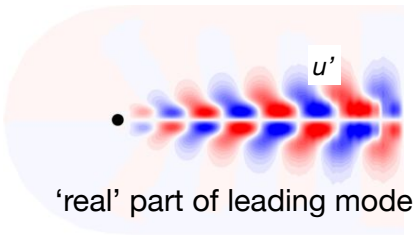
Examples of large-time (eigenmodal) instability

2D instability of steady cylinder wake, *Re*=45

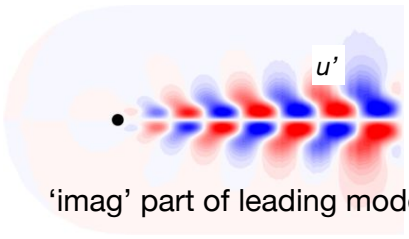
Marginally stable with a complex conjugate-pair eigenvalue $\lambda_{\mathrm{Re}} = 0,\ \lambda_{\mathrm{Im}} = \pm 0.77974$



Jackson *JFM* **182** (1987)



'real' part of leading mode



'imag' part of leading mode

'real' part of leading adjoint mode

'imag' part of leading adjoint mode

Semi-complex modes when base flow is 2D2C

If perturbation flow $\boldsymbol{u}'$ is 3D then using Fourier expansions in *z* it becomes a 2D complex mode.

$$\boldsymbol{u}'(x,y,z,t) = \widehat{\boldsymbol{u}}(x,y,t)\mathrm{e}^{\mathrm{i}\beta z}$$

Since the LNS and Fourier transformation are both linear, they commute and we can solve the eigensystems of each Fourier mode as separate problems.

I.e. timestep each eigensystem using $\partial_t \widehat{\boldsymbol{u}} - L\widehat{\boldsymbol{u}} = 0$ with β as a given parameter.

Now *if* the base flow $\boldsymbol{U}$ is 2D2C, it turns out that the 2D complex modes do not need to be fully populated.

In this case, modes of the form $(u', v', w') = (\hat{u}\cos\beta z, \hat{v}\cos\beta z, -\hat{w}\sin\beta z)$,

$$p' = \hat{p}\cos\beta z$$

will pass through the LNS and retain the same form in *z* (homework).

This means they naturally satisfy the requirements for an eigenmode.

Since in fully populated 2D complex modes, $\hat{u}\cos\beta z \iff \hat{u}.\mathrm{Re},\quad -\hat{u}\sin\beta z \iff \hat{u}.\mathrm{Im}$ the result above means that only half the full complex storage is required when the base flow is 2D2C. This saves half the storage and number of operations.

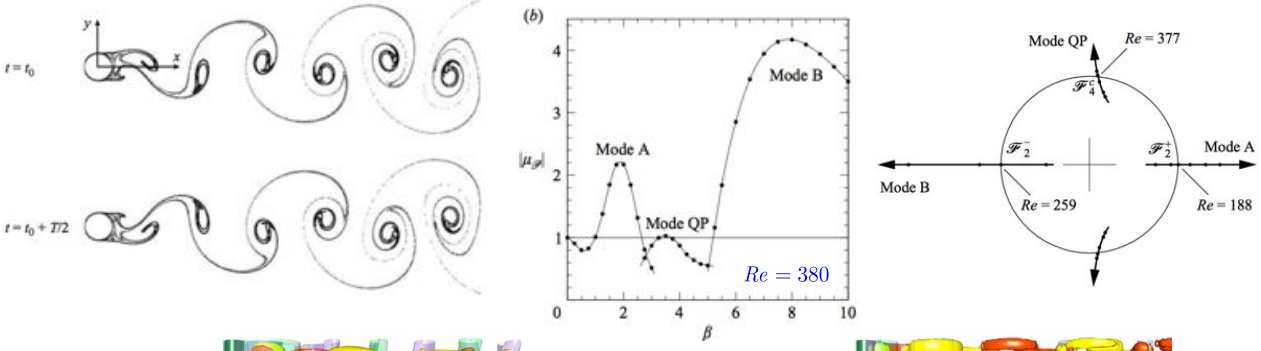
OTOH the mode has a given spatial orientation. We recognise that an arbitrary (but spatially constant) phase shift in *z* gives an equally valid result.

Examples of Floquet instability

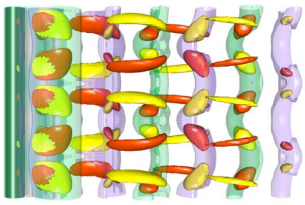
3D instability of 2D time-periodic cylinder wake

Barkley & Henderson *JFM* **322** (1996)

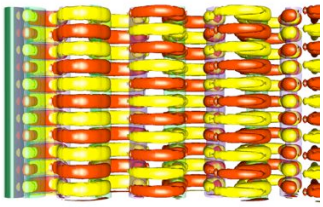
Blackburn, Marques & Lopez *JFM* **522** (2005)



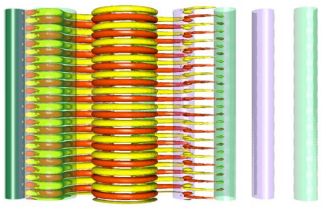
Mode A



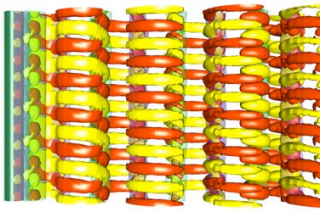
Mode QP (SW)



Mode B



Mode QP (TW)



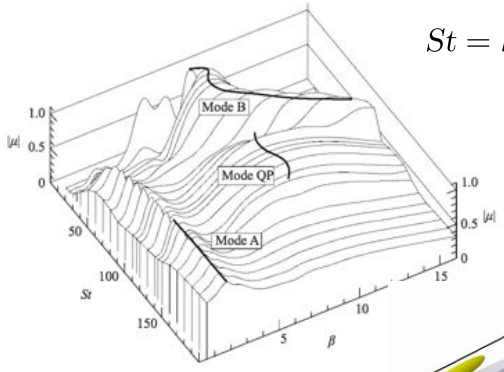
Examples of Floquet instability

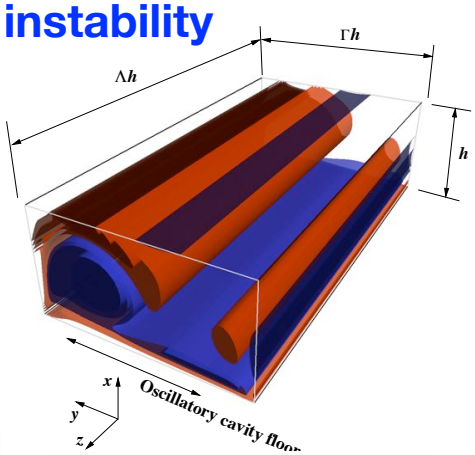
3D instability of 2D time-periodic driven cavity

Blackburn & Lopez *JFM* **497** (2003)

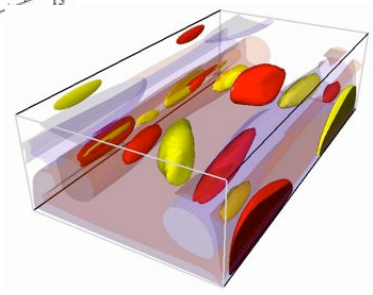
$$Re = V_{\max} h/\nu$$

$$St = h^2 T/\nu$$

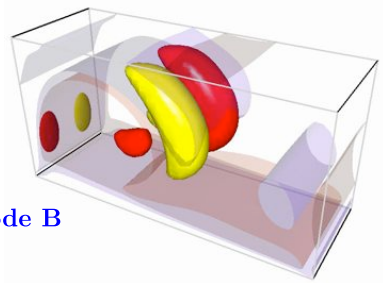




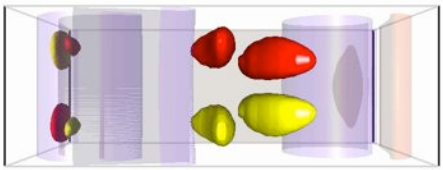
Mode A



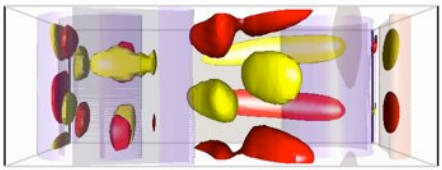
Mode B



Mode QP (SW)



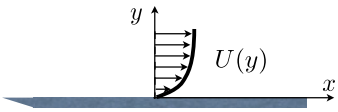
Mode QP (TW)



Part 4: Linear Navier–Stokes and initial value (transient) problems

Transient growth from initial conditions (IVP)

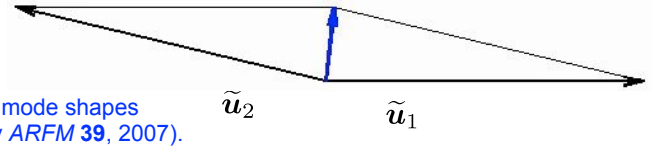
The linearized Navier–Stokes operator is in general non-symmetrical – this is easy to see in the case of parallel shear flows where the base flow $\boldsymbol{U}=(U(y),0,0)$.



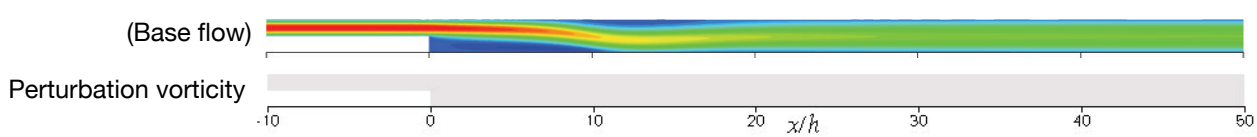
$$\left[\begin{array}{ccc|c} \partial_t + U\partial_x - \nabla^2 & \partial_y U & 0 & \partial_x \\ 0 & \partial_t + U\partial_x - \nabla^2 & 0 & \partial_x \\ 0 & 0 & \partial_t + U\partial_x - \nabla^2 & \partial_x \\ \hline \partial_x & \partial_y & \partial_z & 0 \end{array}\right] \left\{\begin{array}{c} u' \\ v' \\ w' \\ \hline p' \end{array}\right\} = \left\{\begin{array}{c} 0 \\ 0 \\ 0 \\ \hline 0 \end{array}\right\}$$

It follows that the eigenmodes of the problem are non-orthogonal and it turns out that even if all modes are stable a perturbation can produce (perhaps very large) algebraic energy *growth* at short/finite times, as opposed to exponential *decay*.

For mode shapes, non-orthogonality means $\int_\Omega \widetilde{\boldsymbol{u}}_i \cdot \widetilde{\boldsymbol{u}}_j \, \mathrm{d}\Omega \neq 0$



Physically this non-orthogonality manifests as the eigenmode shapes looking rather similar to one another (see Kim & Bewley *ARFM* **39**, 2007).



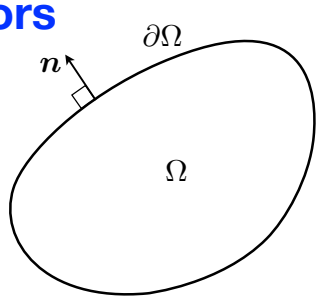
Focus changes from long-time growth to transient growth though ultimately we still expect to see exponential (eigensystem) behaviour as $t \to \infty$.

Adjoint variables and operators

For steady applications, the adjoint variable $\boldsymbol{v}^*$ and operator A^* are defined such that $(\boldsymbol{v}^*, A\boldsymbol{v}) = (A^*\boldsymbol{v}^*, \boldsymbol{v})$

where $(\boldsymbol{a}, \boldsymbol{b}) = \int_\Omega \boldsymbol{a}\cdot\boldsymbol{b}\,\mathrm{d}V$ on domain Ω

and where $\boldsymbol{v}$ and $\boldsymbol{v}^*$ have 'compact support' in Ω.



For unsteady problems (NSE) we have also to consider the temporal domain, say [0, τ] so the overall domain is Ω × [0, τ] and now $\langle \boldsymbol{u}^*, \partial_t\boldsymbol{u}' - L\boldsymbol{u}'\rangle = \langle \boldsymbol{u}', \partial_t\boldsymbol{u}^* + L^*\boldsymbol{u}^*\rangle$

where $\langle \boldsymbol{a}, \boldsymbol{b}\rangle = \int_0^\tau \int_\Omega \boldsymbol{a}\cdot\boldsymbol{b}\,\mathrm{d}V\mathrm{d}t$

Starting from Linearised Navier–Stokes equations (LNSE)

$$\partial_t\boldsymbol{u}' = -\boldsymbol{U}\cdot\boldsymbol{\nabla}\boldsymbol{u}' - (\boldsymbol{\nabla}\boldsymbol{U})^{\mathrm{T}}\cdot\boldsymbol{u}' - \nabla p' + Re^{-1}\nabla^2\boldsymbol{u}' \quad \text{with } \boldsymbol{\nabla}\cdot\boldsymbol{u}' = 0$$

$$\boldsymbol{u}'_\tau = \mathcal{M}(\tau)\boldsymbol{u}'_0$$

Short form $\partial_t\boldsymbol{u}' - L\boldsymbol{u}' = 0$

Integration by parts ⟹ Adjoint NSE (ANSE)

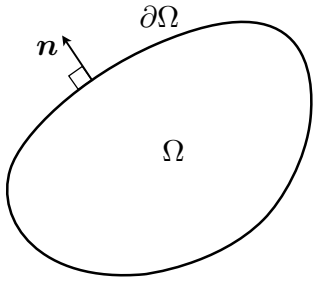
$$-\partial_t\boldsymbol{u}^* = +\boldsymbol{U}\cdot\boldsymbol{\nabla}\boldsymbol{u}^* - \boldsymbol{\nabla}\boldsymbol{U}\cdot\boldsymbol{u}^* - \nabla p^* + Re^{-1}\nabla^2\boldsymbol{u}^* \quad \text{with } \boldsymbol{\nabla}\cdot\boldsymbol{u}^* = 0$$

Short form $\partial_t\boldsymbol{u}^* + L^*\boldsymbol{u}^* = 0$

NB: $\boldsymbol{u}^*_0 \longleftarrow$ $\partial_t\boldsymbol{u}^* + L^*\boldsymbol{u}^* = 0$ integrate backwards $\longleftarrow \boldsymbol{u}^*_\tau$ $\quad \boldsymbol{u}^*_0 = \mathcal{M}^*(\tau)\boldsymbol{u}^*_\tau$

Boundary conditions

Recall $(\boldsymbol{a}, \boldsymbol{b}) = \int_\Omega \boldsymbol{a}\cdot\boldsymbol{b}\,\mathrm{d}V$ $\qquad \langle \boldsymbol{a}, \boldsymbol{b}\rangle = \int_0^T \int_\Omega \boldsymbol{a}\cdot\boldsymbol{b}\,\mathrm{d}V\mathrm{d}t$



Compact support allowed IBP without regard to space-time boundary conditions.

Re-introducing terminal, boundary conditions:

Apply IBP to $-\langle \boldsymbol{u}^*, \partial_t\boldsymbol{u}' - L\boldsymbol{u}'\rangle$

$$-\langle \boldsymbol{u}^*, \partial_t\boldsymbol{u}' - L\boldsymbol{u}'\rangle = \langle \boldsymbol{u}', \partial_t\boldsymbol{u}^* + L^*\boldsymbol{u}^*\rangle + \int_0^\tau\int_\Omega -\partial_t(\boldsymbol{u}'\cdot\boldsymbol{u}^*)\,\mathrm{d}V\mathrm{d}t$$
$$+\int_0^\tau\int_\Omega \boldsymbol{\nabla}\cdot\{-\boldsymbol{U}(\boldsymbol{u}'\cdot\boldsymbol{u}^*) + \boldsymbol{u}'p^* - \boldsymbol{u}^*p' + Re^{-1}[(\boldsymbol{\nabla}\boldsymbol{u}')\cdot\boldsymbol{u}^* - (\boldsymbol{\nabla}\boldsymbol{u}^*)\cdot\boldsymbol{u}']\}\,\mathrm{d}V\mathrm{d}t$$

Exchange order of integration and apply divergence theorem

$$-\langle \boldsymbol{u}^*, \partial_t\boldsymbol{u}' - L\boldsymbol{u}'\rangle = \langle \boldsymbol{u}', \partial_t\boldsymbol{u}^* + L^*\boldsymbol{u}^*\rangle - (\boldsymbol{u}^*_\tau, \boldsymbol{u}'_\tau) + (\boldsymbol{u}^*_0, \boldsymbol{u}'_0)$$

Volume integrals involving terminal conditions

$$+\int_0^\tau \int_{\partial\Omega} \boldsymbol{n}\cdot\{-\boldsymbol{U}(\boldsymbol{u}'\cdot\boldsymbol{u}^*) + \boldsymbol{u}'p^* - \boldsymbol{u}^*p' + Re^{-1}[(\boldsymbol{\nabla}\boldsymbol{u}')\cdot\boldsymbol{u}^* - (\boldsymbol{\nabla}\boldsymbol{u}^*)\cdot\boldsymbol{u}']\}\,\mathrm{d}S\mathrm{d}t$$

Surface integral involving boundary conditions

As far as possible we will choose terminal and boundary conditions to suit us.

Typically means using wall-type BCs everywhere.

Adjoint operator *M**, joint operator *M*M*

Forward LNS operator & variables

$$\mathcal{H} \equiv \begin{cases} \left[\begin{array}{c|c} -\partial_t - \mathrm{DN} + Re^{-1}\nabla^2 & -\nabla \\ \hline \nabla\cdot & 0 \end{array}\right] \quad \boldsymbol{q} = \begin{pmatrix} \boldsymbol{u}' \\ p' \end{pmatrix} \\ \mathrm{DN}\,\boldsymbol{u}' = (\boldsymbol{U}\cdot\nabla)\boldsymbol{u}' + (\nabla\boldsymbol{U})^{\mathrm{T}}\cdot\boldsymbol{u}' \end{cases}$$

Adjoint LNS operator & variables

$$\mathcal{H}^* \equiv \begin{cases} \left[\begin{array}{c|c} \partial_t - \mathrm{DN}^* + Re^{-1}\nabla^2 & -\nabla \\ \hline \nabla\cdot & 0 \end{array}\right] \quad \boldsymbol{q}^* = \begin{pmatrix} \boldsymbol{u}^* \\ p^* \end{pmatrix} \\ \mathrm{DN}^*\boldsymbol{u}^* = -(\boldsymbol{U}\cdot\nabla)\boldsymbol{u}^* + (\nabla\boldsymbol{U})\cdot\boldsymbol{u}^* \end{cases}$$

Forward system

$$\begin{gathered} \mathcal{H}\boldsymbol{q} = 0 \quad (\mathbf{x},t) \in \Gamma = [0,\tau]\times\Omega \\ \boldsymbol{u}(t=0) = \boldsymbol{u}_0 \\ \boldsymbol{u}(\partial\Omega) = 0 \end{gathered}$$

Adjoint system

$$\begin{gathered} \mathcal{H}^*\boldsymbol{q}^* = 0 \quad (\mathbf{x},t) \in \Gamma = [0,\tau]\times\Omega \\ \boldsymbol{u}^*(t=\tau) = \boldsymbol{u}^*_\tau \\ \boldsymbol{u}^*(\partial\Omega) = 0 \end{gathered}$$

Energy, space-time inner products: $(\boldsymbol{u},\boldsymbol{u}^*) = \int_\Omega \boldsymbol{u}\cdot\boldsymbol{u}^*\,\mathrm{d}v\mathrm{d}t \qquad \langle \boldsymbol{q},\boldsymbol{q}^*\rangle = \int_0^\tau\int_\Omega \boldsymbol{q}\cdot\boldsymbol{q}^*\,\mathrm{d}v\mathrm{d}t$

Using IBP and divergence theorem,

$$\langle\mathcal{H}\hat{\boldsymbol{q}},\hat{\boldsymbol{q}}^*\rangle - \langle\hat{\boldsymbol{q}},\mathcal{H}^*\hat{\boldsymbol{q}}^*\rangle = -\int_\Omega [\boldsymbol{u}\cdot\boldsymbol{u}^*]_0^\tau\,\mathrm{d}v +$$

$$\int_0^\tau\oint_{\partial\Omega} \hat{\boldsymbol{n}}\cdot\left\{-\boldsymbol{U}(\boldsymbol{u}\cdot\boldsymbol{u}^*) + \boldsymbol{u}p^* - \boldsymbol{u}^*p + Re^{-1}\left((\nabla\boldsymbol{u})^{\mathrm{T}}\cdot\boldsymbol{u}^* - (\nabla\boldsymbol{u}^*)^{\mathrm{T}}\cdot\boldsymbol{u}\right)\right\}\mathrm{d}S\,\mathrm{d}t,$$

If $\hat{\boldsymbol{q}},\hat{\boldsymbol{q}}^*$ have compact support on $\Gamma = [0,\tau]\times\Omega$ then: $\langle\mathcal{H}\hat{\boldsymbol{q}},\hat{\boldsymbol{q}}^*\rangle - \langle\hat{\boldsymbol{q}},\mathcal{H}^*\hat{\boldsymbol{q}}^*\rangle = 0$

This formally establishes that $\mathcal{H}^*$ is the adjoint of $\mathcal{H}$.

Adjoint operator *M**, joint operator *M*M*

By appropriate choice of *spatial* BCs (e.g. zero Dirichlet everywhere) we eliminate terms

$$\langle\mathcal{H}\hat{\boldsymbol{q}},\hat{\boldsymbol{q}}^*\rangle - \langle\hat{\boldsymbol{q}},\mathcal{H}^*\hat{\boldsymbol{q}}^*\rangle = -\int_\Omega [\boldsymbol{u}\cdot\boldsymbol{u}^*]_0^\tau\,\mathrm{d}v +$$

$$\int_0^\tau\oint_{\partial\Omega} \hat{\boldsymbol{n}}\cdot\left\{-\boldsymbol{U}(\boldsymbol{u}\cdot\boldsymbol{u}^*) + \boldsymbol{u}p^* - \boldsymbol{u}^*p + Re^{-1}\left((\nabla\boldsymbol{u})^{\mathrm{T}}\cdot\boldsymbol{u}^* - (\nabla\boldsymbol{u}^*)^{\mathrm{T}}\cdot\boldsymbol{u}\right)\right\}\mathrm{d}S\,\mathrm{d}t,$$

and require a constraint between terminal solutions of forward and adjoint systems:

$$\int_\Omega [\boldsymbol{u}'\cdot\boldsymbol{u}^*]_0^\tau\,\mathrm{d}v = 0 \implies (\boldsymbol{u}'(\tau),\boldsymbol{u}^*(\tau)) = (\boldsymbol{u}'(0),\boldsymbol{u}^*(0)) \qquad \boldsymbol{u}'(\tau) = \mathcal{M}(\tau)\boldsymbol{u}'(0)$$

$$(\mathcal{M}(\tau)\boldsymbol{u}'(0),\boldsymbol{u}^*(\tau)) = (\boldsymbol{u}'(0),\mathcal{M}^*(\tau)\boldsymbol{u}^*(\tau)) \qquad \boldsymbol{u}^*(0) = \mathcal{M}^*(\tau)\boldsymbol{u}^*(\tau)$$

Link the solutions together by setting $\boldsymbol{u}^*(\tau) = \boldsymbol{u}'(\tau) = \mathcal{M}(\tau)\boldsymbol{u}'(0)$ and then we have

$$(\mathcal{M}(\tau)\boldsymbol{u}'(0),\mathcal{M}(\tau)\boldsymbol{u}'(0)) = (\boldsymbol{u}'(0),\mathcal{M}^*(\tau)\mathcal{M}(\tau)\boldsymbol{u}'(0))$$

which is used to transform

$$G(\tau) = \frac{E(\tau)}{E(0)} = \frac{(\boldsymbol{u}'(\tau),\boldsymbol{u}'(\tau))}{(\boldsymbol{u}'(0),\boldsymbol{u}'(0))} = \frac{(\mathcal{M}\boldsymbol{u}'(0),\mathcal{M}\boldsymbol{u}'(0))}{(\boldsymbol{u}'(0),\boldsymbol{u}'(0))} = \frac{(\boldsymbol{u}'(0),\mathcal{M}^*\mathcal{M}\boldsymbol{u}'(0))}{(\boldsymbol{u}'(0),\boldsymbol{u}'(0))}$$

into the eigenproblem $\mathcal{M}^*\mathcal{M}\boldsymbol{u}'(0) = G(\tau)\boldsymbol{u}'(0)$

I.e: solutions to this eigenproblem fulfil the required constraint.

Transient growth and the SVD

Look for initial condition $\boldsymbol{u}'(0)$ that gives maximum energy growth over finite interval τ — optimal perturbation for interval τ.

If $(\boldsymbol{u},\boldsymbol{v}) = \int_\Omega \boldsymbol{u}\cdot\boldsymbol{v}\,\mathrm{d}v$ then $G(\tau) = \dfrac{E(\tau)}{E(0)} = \dfrac{(\boldsymbol{u}'(\tau),\boldsymbol{u}'(\tau))}{(\boldsymbol{u}'(0),\boldsymbol{u}'(0))} = \dfrac{(\mathcal{M}\boldsymbol{u}'(0),\mathcal{M}\boldsymbol{u}'(0))}{(\boldsymbol{u}'(0),\boldsymbol{u}'(0))}$

$$= \frac{(\boldsymbol{u}'(0),\mathcal{M}^*\mathcal{M}\boldsymbol{u}'(0))}{(\boldsymbol{u}'(0),\boldsymbol{u}'(0))}$$

where $\mathcal{M}^*$ is the adjoint of $\mathcal{M}$.

Considering an analog in linear algebra (with $\mathcal{M}\equiv A$ and $\mathcal{M}^*\equiv A^{\mathrm{T}}$) this is like looking for *x* of unit norm that maximizes ‖*Ax*‖². $\|Ax\|^2 = \|x^{\mathrm{T}}A^{\mathrm{T}}Ax\|$

$A^{\mathrm{T}}A$ is symmetric positive definite and so has orthogonal eigenvectors, and positive real eigenvalues.

The vector *x* that is most amplified by *A* is the eigenvector of operator $A^{\mathrm{T}}A$ corresponding to its maximum eigenvalue σ_{max}^2.

The eigensystem of $A^{\mathrm{T}}A$ is related to the singular value decomposition (SVD) of *A*.

$$AV = U\Sigma \quad \text{or} \quad A = U\Sigma V^{\mathrm{T}} \quad \text{or} \quad U^{\mathrm{T}}AV = \Sigma$$

$$\Sigma = \mathrm{diag}\,(\sigma_1,\sigma_2,\ldots,\sigma_N)$$

where *U* & *V* are orthogonal matrices and Σ (singular values) is diagonal.

(For simplicity we take *A* to be *N*×*N*).

Transient growth and the SVD

SVD of *A*

$$AV = U\Sigma \quad \text{or} \quad A = U\Sigma V^{\mathrm{T}} \quad \text{or} \quad U^{\mathrm{T}}AV = \Sigma$$

$$\Sigma = \mathrm{diag}\,(\sigma_1,\sigma_2,\ldots,\sigma_N)$$

The eigenvectors of $A^{\mathrm{T}}A$ are *V*, the right singular vectors of *A*, while the eigenvectors of AA^{T} are *U*, the left singular vectors of *A*. All these vectors have unit norm. For each singular value σ_i there is a corresponding pair of vectors u_i and v_i.

The most amplified vector (which achieves optimal growth under the action of *A*) is then v_{opt} corresponding to σ_{max} (i.e. the leading eigenvector), which is mapped to its partner left singular vector u_{opt}, scaled by σ_{max}

$$Av_{\mathrm{opt}} = \sigma_{\max}u_{\mathrm{opt}}$$

recall $\mathcal{M}\equiv A$

The sets of input and output vectors are orthogonal: $V^{\mathrm{T}}V$=I; $U^{\mathrm{T}}U$=I.

This means that we can rank the input vectors *V* (in the sense of contribution to output energy) according to the (squared) singular values Σ^2, and each one will map to a single output vector in *U*. This is the property that the eigenvectors had lost.

In our problems, the right singular vectors *v* are initial perturbation flow fields, and under the action of *M*(τ) (LNS state transition) the outcomes are the matching left singular vectors *u*. The amount of kinetic energy growth is σ^2.

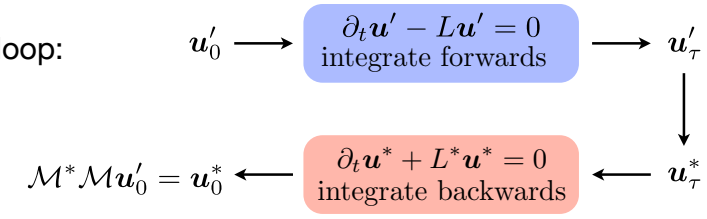
Compute eigensystem by timestepping

$$\mathcal{M}^*\mathcal{M}\boldsymbol{u}_0' = G(\tau)\boldsymbol{u}_0'$$

To be able to solve the eigensystem of $\mathcal{M}^*\mathcal{M}$ we need only to be able to apply the operator to a vector.

Solve by Krylov method with inner loop:

Same outer loop as for instability solution, different inner (operator) loop.



Generates Krylov sequence $T = \{\boldsymbol{u}_0', (\mathcal{M}^*\mathcal{M})\boldsymbol{u}_0', (\mathcal{M}^*\mathcal{M})^2\boldsymbol{u}_0', \ldots, (\mathcal{M}^*\mathcal{M})^{k-1}\boldsymbol{u}_0'\}$

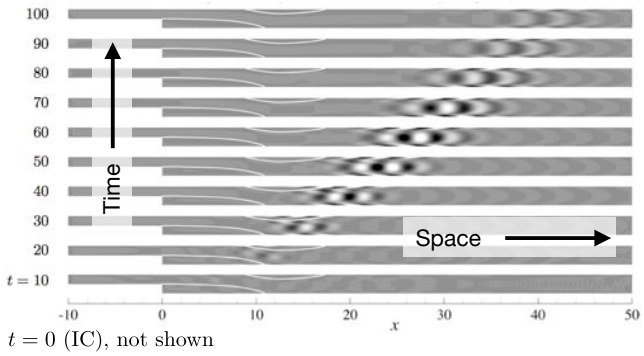
We can compute optimal initial conditions via either optimisation-based or eigensystem solvers – great for bootstrapping new codes!

The only issue then is relative performance - with most of the work being in time-integration as the inner loop is basically identical in each case.

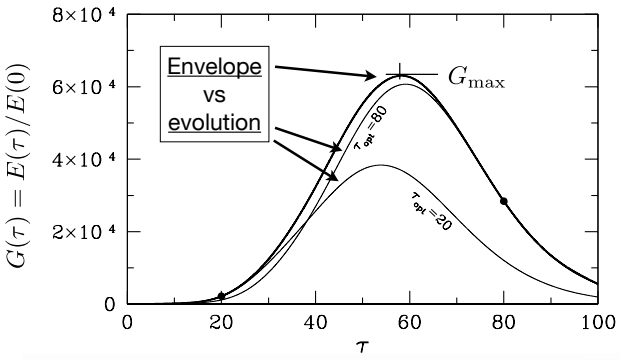
One useful feature of the eigensystem approach is that one can obtain `suboptimal' initial perturbations – no obvious way to do this with the optimisation approach.

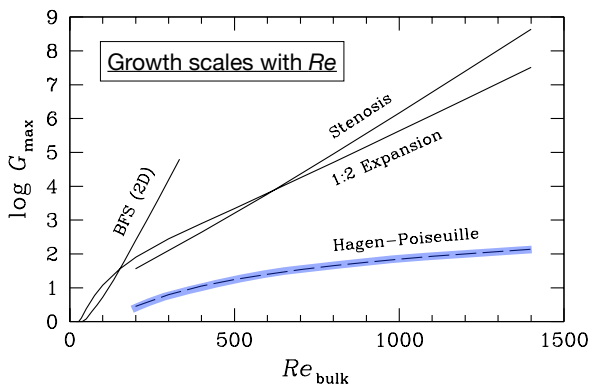
Transient growth (from ICs) 101

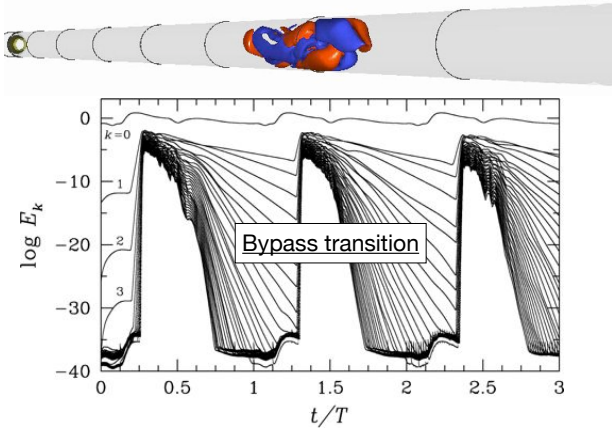
Perturbation vorticity evolution, optimal IC



$t = 0$ (IC), not shown







Part 5: Optimisation toolkit

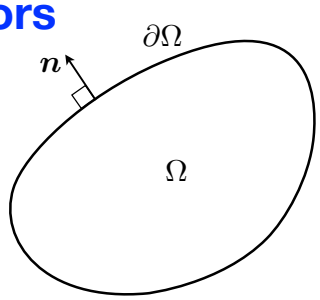
Adjoint variables and operators

For steady applications, the adjoint variable $\boldsymbol{v}^*$ and operator A^* are defined such that

$$(\boldsymbol{v}^*, A\boldsymbol{v}) = (A^*\boldsymbol{v}^*, \boldsymbol{v})$$

where $(\boldsymbol{a}, \boldsymbol{b}) = \int_\Omega \boldsymbol{a} \cdot \boldsymbol{b}\, \mathrm{d}V$ on domain Ω

and where $\boldsymbol{v}$ and $\boldsymbol{v}^*$ have 'compact support' in Ω



For unsteady problems (NSE) we have also to consider the temporal domain, say [0, τ] so the overall domain is Ω × [0, τ] and now

$$\langle \boldsymbol{u}^*, \partial_t \boldsymbol{u}' - L\boldsymbol{u}' \rangle = \langle \boldsymbol{u}', \partial_t \boldsymbol{u}^* + L^*\boldsymbol{u}^* \rangle$$

where $\langle \boldsymbol{a}, \boldsymbol{b} \rangle = \int_0^\tau \int_\Omega \boldsymbol{a} \cdot \boldsymbol{b}\, \mathrm{d}V\, \mathrm{d}t$

Starting from Linearised Navier–Stokes equations (LNSE)

$$\partial_t \boldsymbol{u}' = -\boldsymbol{U} \cdot \boldsymbol{\nabla}\boldsymbol{u}' - (\boldsymbol{\nabla}\boldsymbol{U})^{\mathrm{T}} \cdot \boldsymbol{u}' - \nabla p' + Re^{-1}\nabla^2 \boldsymbol{u}' \text{ with } \boldsymbol{\nabla} \cdot \boldsymbol{u}' = 0$$

$$\boldsymbol{u}'_\tau = \mathcal{M}(\tau)\boldsymbol{u}'_0$$

Short form $\partial_t \boldsymbol{u}' - L\boldsymbol{u}' = 0$

Integration by parts ⟹ Adjoint NSE (ANSE)

$$-\partial_t \boldsymbol{u}^* = +\boldsymbol{U} \cdot \boldsymbol{\nabla}\boldsymbol{u}^* - \boldsymbol{\nabla}\boldsymbol{U} \cdot \boldsymbol{u}^* - \nabla p^* + Re^{-1}\nabla^2 \boldsymbol{u}^* \text{ with } \boldsymbol{\nabla} \cdot \boldsymbol{u}^* = 0$$

Short form $\partial_t \boldsymbol{u}^* + L^*\boldsymbol{u}^* = 0$

NB: $\boldsymbol{u}^*_0 \longleftarrow$ $\partial_t \boldsymbol{u}^* + L^*\boldsymbol{u}^* = 0$ integrate backwards $\longleftarrow \boldsymbol{u}^*_\tau$

$$\boldsymbol{u}^*_0 = \mathcal{M}^*(\tau)\boldsymbol{u}^*_\tau$$

Two optimal energy functionals

The two kinds of optimisations we consider:

1. Initial flow perturbation $\boldsymbol{u}'_0$ that produces maximum kinetic energy growth at time τ.

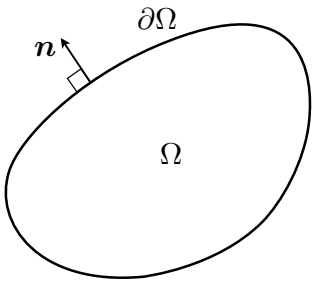
Growth $$G = \max_{\boldsymbol{u}'_0} \frac{(\boldsymbol{u}'_\tau, \boldsymbol{u}'_\tau)}{(\boldsymbol{u}'_0, \boldsymbol{u}'_0)} = \max_{\boldsymbol{u}'_0} \frac{E(\tau)}{E(0)}$$ Final energy / Initial energy

2. Boundary flow perturbation $\boldsymbol{u}'_c$ that produces maximum kinetic energy gain at time τ.

We need definitions for boundary integrals:

$$[\boldsymbol{a}, \boldsymbol{b}] = D \int_{\partial\Omega} \boldsymbol{a} \cdot \boldsymbol{b} \, \mathrm{d}S \qquad \{\boldsymbol{a}, \boldsymbol{b}\} = D \int_0^\tau \int_{\partial\Omega} \boldsymbol{a} \cdot \boldsymbol{b} \, \mathrm{d}S \mathrm{d}t$$

where *D* is a representative length.



Gain $$K = \max_{\boldsymbol{u}'_c} \frac{(\boldsymbol{u}'_\tau, \boldsymbol{u}'_\tau)}{\{\boldsymbol{u}'_c, \boldsymbol{u}'_c\}}$$ Final energy / Boundary energy

Optimisation in both cases is constrained: solutions have to obey LNSE.

Constrained optimisation

Generalising the kinetic energy functionals to be optimised as $\mathcal{J}(\boldsymbol{u}'_{\mathrm{opt}})$

$$\mathcal{J}(\boldsymbol{u}'_{\mathrm{opt}}) \equiv G = \max_{\boldsymbol{u}'_0} \frac{(\boldsymbol{u}'_\tau, \boldsymbol{u}'_\tau)}{(\boldsymbol{u}'_0, \boldsymbol{u}'_0)} \quad \text{or} \quad K = \max_{\boldsymbol{u}_c} \frac{(\boldsymbol{u}'_\tau, \boldsymbol{u}'_\tau)}{\{\boldsymbol{u}'_c, \boldsymbol{u}'_c\}}$$

we will construct an augmented/Lagrangian functional

$$\mathcal{L} = \mathcal{J}(\boldsymbol{u}'_{\mathrm{opt}}) - \langle \boldsymbol{u}^*, \partial_t \boldsymbol{u}' - L\boldsymbol{u}' \rangle$$

Constrains solutions to satisfy LNSE

for which we will find extrema.

$\boldsymbol{u}^*$ plays the role of a Lagrange multiplier.

We have converted a constrained optimisation problem into an unconstrained optimisation problem – but with more variables.

We allow arbitrary variations of the Lagrangian with respect to all the variables and ensure that all gradients are simultaneously zero.

The standard tool for this job is the Gateaux differential

$$\frac{\delta \mathcal{L}}{\delta \boldsymbol{q}} = \lim_{\epsilon \to 0} \frac{\mathcal{L}(\boldsymbol{q} + \epsilon \delta \boldsymbol{q}) - \mathcal{L}(\boldsymbol{q})}{\epsilon} \equiv \langle \nabla_{\boldsymbol{q}} \mathcal{L}, \delta \boldsymbol{q} \rangle,$$

which identifies the directional derivative of $\mathcal{L}$ with respect to arbitrary variation in variable $\boldsymbol{q}$.

Calculus of variations for optimal ICs

Set boundary perturbations to zero, and seek the initial perturbation $\boldsymbol{u}'_0$ that provides maximum energy growth for a given time horizon τ.

$$\mathcal{L}_0 = \frac{(\boldsymbol{u}'_\tau, \boldsymbol{u}'_\tau)}{(\boldsymbol{u}'_0, \boldsymbol{u}'_0)} - \langle \boldsymbol{u}^*, \partial_t \boldsymbol{u}' + L(\boldsymbol{u}') \rangle$$

$$\equiv \frac{(\boldsymbol{u}'_\tau, \boldsymbol{u}'_\tau)}{(\boldsymbol{u}'_0, \boldsymbol{u}'_0)} + \langle \boldsymbol{u}', \partial_t \boldsymbol{u}^* + L^*(\boldsymbol{u}^*) \rangle - (\boldsymbol{u}^*_\tau, \boldsymbol{u}'_\tau) + (\boldsymbol{u}^*_0, \boldsymbol{u}'_0)$$

removed using zero BCs

$$+ \int_0^T \int_{\partial\Omega} \boldsymbol{n} \cdot \{ -\boldsymbol{U}(\boldsymbol{u}' \cdot \boldsymbol{u}^*) + \boldsymbol{u}' p^* - \boldsymbol{u}^* p' + Re^{-1} [(\nabla \boldsymbol{u}') \cdot \boldsymbol{u}^* - (\nabla \boldsymbol{u}^*) \cdot \boldsymbol{u}'] \} \, \mathrm{d}S \mathrm{d}t$$

$$\frac{\delta \mathcal{L}_0}{\delta \boldsymbol{u}^*} = 0 \implies \partial_t \boldsymbol{u}' - L\boldsymbol{u}' = 0$$

$$\frac{\delta \mathcal{L}_0}{\delta \boldsymbol{u}'} = 0 \implies \partial_t \boldsymbol{u}^* + L^* \boldsymbol{u}^* = 0$$

$$\frac{\delta \mathcal{L}_0}{\delta \boldsymbol{u}'_\tau} = 0 \implies \boldsymbol{u}^*_\tau = 2 \frac{\boldsymbol{u}'_\tau}{(\boldsymbol{u}'_0, \boldsymbol{u}'_0)}$$

$$\frac{\delta \mathcal{L}_0}{\delta \boldsymbol{u}'_0} = 0 \implies \boldsymbol{u}^*_0 = 2 \frac{(\boldsymbol{u}'_\tau, \boldsymbol{u}'_\tau)}{(\boldsymbol{u}'_0, \boldsymbol{u}'_0)^2} \boldsymbol{u}'_0$$

In each case we use

$$\frac{\delta \mathcal{L}}{\delta \boldsymbol{q}} = \lim_{\epsilon \to 0} \frac{\mathcal{L}(\boldsymbol{q} + \epsilon \delta \boldsymbol{q}) - \mathcal{L}(\boldsymbol{q})}{\epsilon} \equiv \langle \nabla_{\boldsymbol{q}} \mathcal{L}, \delta \boldsymbol{q} \rangle$$

i.e. $$\nabla_{\boldsymbol{u}'_0} \mathcal{L}_0 = \boldsymbol{u}^*_0 - 2 \frac{(\boldsymbol{u}'_\tau, \boldsymbol{u}'_\tau)}{(\boldsymbol{u}'_0, \boldsymbol{u}'_0)^2} \boldsymbol{u}'_0$$

Optimisation approach for initial perturbation

Calculus of variations gave four outcomes:

$$\partial_t \boldsymbol{u}' - L\boldsymbol{u}' = 0$$

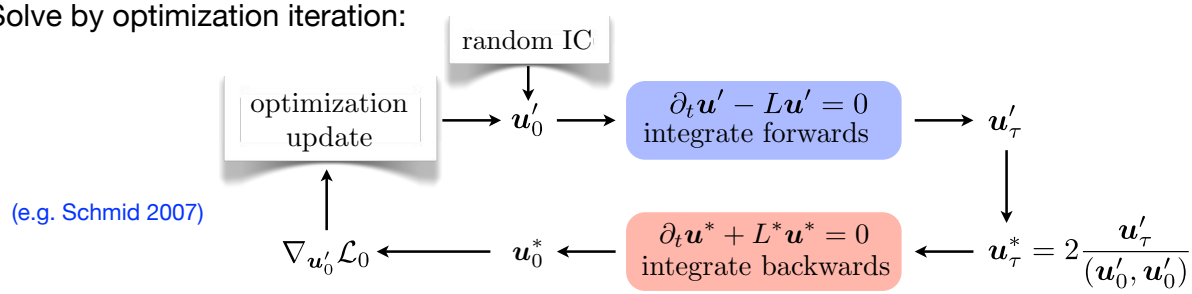
Evolution equations

$$\partial_t \boldsymbol{u}^* + L^* \boldsymbol{u}^* = 0$$

$$\boldsymbol{u}^*_\tau = 2 \frac{\boldsymbol{u}'_\tau}{(\boldsymbol{u}'_0, \boldsymbol{u}'_0)}$$ Terminal condition

$$\nabla_{\boldsymbol{u}'_0} \mathcal{L}_0 = \boldsymbol{u}^*_0 - 2 \frac{(\boldsymbol{u}'_\tau, \boldsymbol{u}'_\tau)}{(\boldsymbol{u}'_0, \boldsymbol{u}'_0)^2} \boldsymbol{u}'_0 = 0$$ Optimality condition

Solve by optimization iteration:



(e.g. Schmid 2007)

The 'optimisation update' could be steepest descent or other appropriate method.

At convergence we have IC $\boldsymbol{u}'_0$ that maximizes $(\boldsymbol{u}'_\tau, \boldsymbol{u}'_\tau) / (\boldsymbol{u}'_0, \boldsymbol{u}'_0)$ subject to constraints.

Eigenvalue approach for initial perturbation

Recall $\boldsymbol{u}'_\tau = \mathcal{M}\boldsymbol{u}'_0$ obtained by forward integration of LNSE

$\boldsymbol{u}^*_0 = \mathcal{M}^*\boldsymbol{u}^*_\tau$ obtained by backward integration of ANSE

Now compare

$$\mathcal{L}_0 = \frac{(\boldsymbol{u}'_\tau, \boldsymbol{u}'_\tau)}{(\boldsymbol{u}'_0, \boldsymbol{u}'_0)} - \langle \boldsymbol{u}^*, \partial_t \boldsymbol{u}' + L(\boldsymbol{u}') \rangle$$

$$\equiv \frac{(\boldsymbol{u}'_\tau, \boldsymbol{u}'_\tau)}{(\boldsymbol{u}'_0, \boldsymbol{u}'_0)} + \langle \boldsymbol{u}', \partial_t \boldsymbol{u}^* + L^*(\boldsymbol{u}^*) \rangle - (\boldsymbol{u}^*_\tau, \boldsymbol{u}'_\tau) + (\boldsymbol{u}^*_0, \boldsymbol{u}'_0)$$

If LNSE and ANSE are always satisfied, $(\boldsymbol{u}^*_\tau, \boldsymbol{u}'_\tau) = (\boldsymbol{u}^*_0, \boldsymbol{u}'_0)$

equivalently $(\boldsymbol{u}^*_\tau, \mathcal{M}\boldsymbol{u}'_0) = (\mathcal{M}^*\boldsymbol{u}^*_\tau, \boldsymbol{u}'_0)$, meaning $\mathcal{M}^*$ is the operator adjoint to $\mathcal{M}$ with respect to the inner product (,).

So now $$\mathcal{L}_0 = \frac{(\boldsymbol{u}'_\tau, \boldsymbol{u}'_\tau)}{(\boldsymbol{u}'_0, \boldsymbol{u}'_0)} = \frac{(\mathcal{M}\boldsymbol{u}'_0, \mathcal{M}\boldsymbol{u}'_0)}{(\boldsymbol{u}'_0, \boldsymbol{u}'_0)} = \frac{(\boldsymbol{u}'_0, \mathcal{M}^*\mathcal{M}\boldsymbol{u}'_0)}{(\boldsymbol{u}'_0, \boldsymbol{u}'_0)}$$

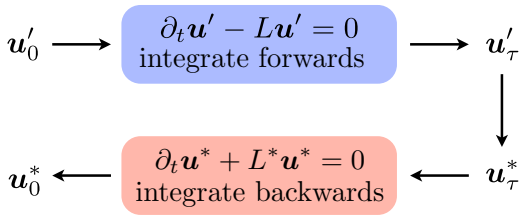
which is maximised when $\boldsymbol{u}'_0$ is the eigenvector of joint symmetric operator $\mathcal{M}^*\mathcal{M}$ corresponding to the largest eigenvalue. This eigenvalue is *G*.

Compute eigensystem by timestepping

$$\mathcal{M}^*\mathcal{M}\boldsymbol{u}'_0 = G(\tau)\boldsymbol{u}'_0$$

To be able to solve the eigensystem of $\mathcal{M}^*\mathcal{M}$ we need only to be able to apply the operator to a vector.

Solve by Krylov method with inner loop:



Same outer loop as for instability solution, different inner (operator) loop.

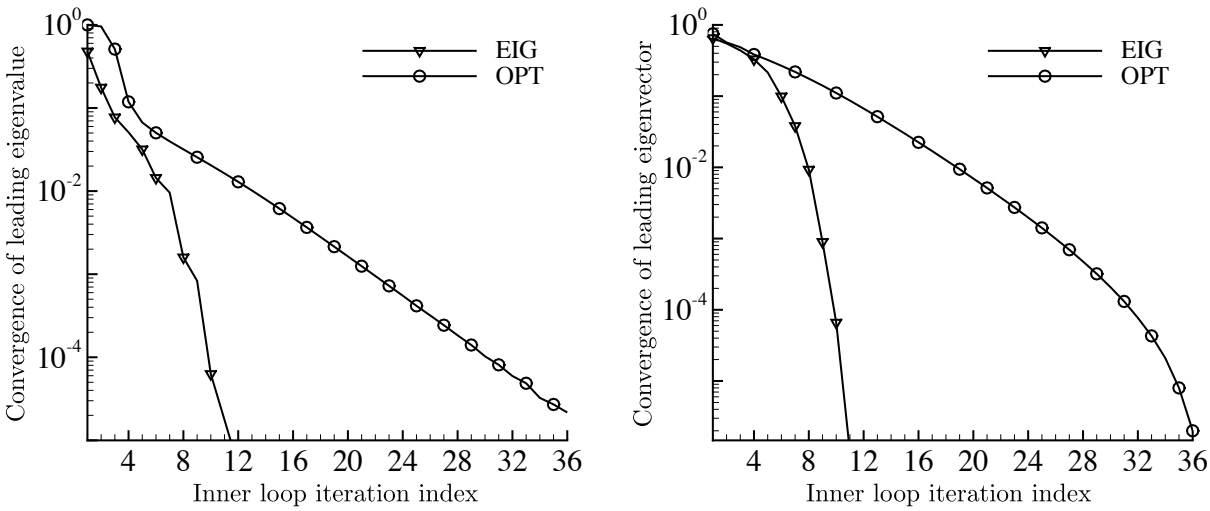
Generates Krylov sequence $T = \{\boldsymbol{u}'_0, (\mathcal{M}^*\mathcal{M})\boldsymbol{u}'_0, (\mathcal{M}^*\mathcal{M})^2\boldsymbol{u}'_0, \ldots, (\mathcal{M}^*\mathcal{M})^{k-1}\boldsymbol{u}'_0\}$

This means we can compute optimal initial conditions via either optimisation-based or eigensystem solvers – great for bootstrapping new codes!

The only issue then is relative performance - with most of the work being in time-integration as the inner loop is basically identical in each case.

But first we'll look at some outcomes.

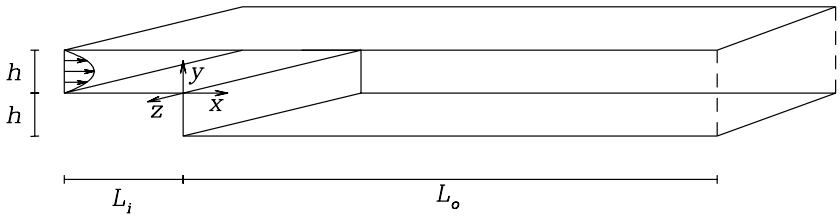
Relative performance



This suggests that the eigensystem approach is generally preferable when computing optimal initial conditions.

Part 6. Example applications of optimal IVP analyses

Backward-facing step, Re_{max} = 500

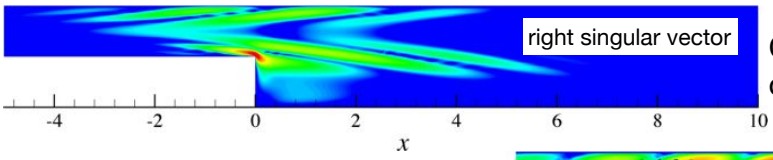


Asymptotic/global stability:
Barkley, Gomez, Henderson (2002)

$Re_c = 748$ ⇒ steady 3D.

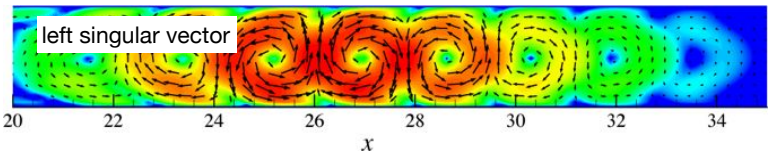
Flow is notorious for its convective instability, even (2D) for *Re* < 800.

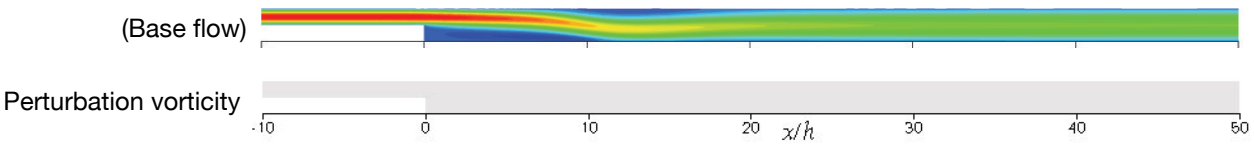
2D global optimum, *Re*=500: G_{max}=63,000, τ=58.



Contours of (log) energy in global optimum initial condition.

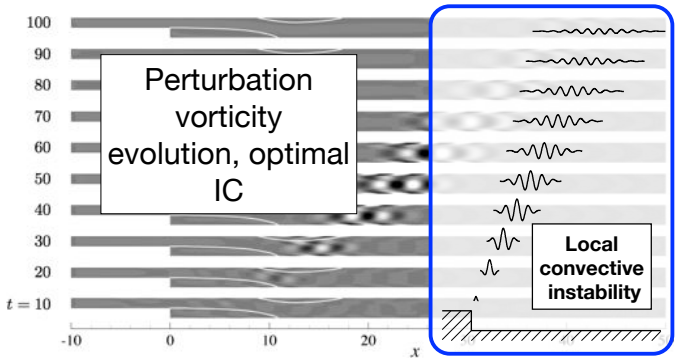
Contours of (log) energy in the outcome, at τ=58.0, with velocity vector field.

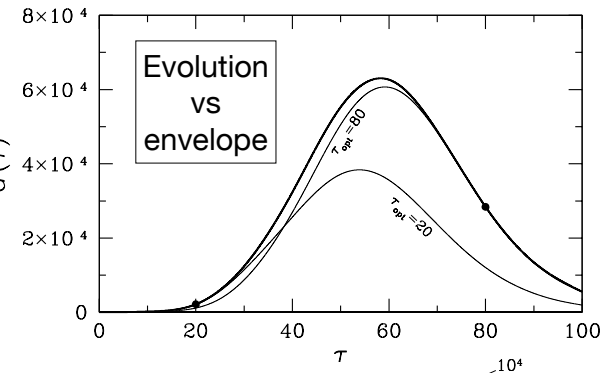


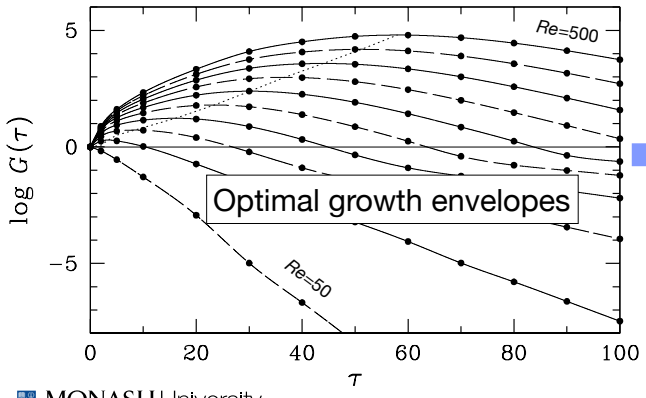


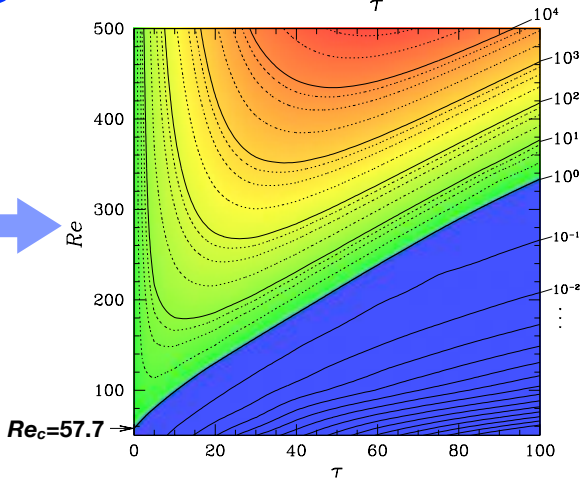
MONASH University Engineering

Backward-facing step, Re_{max} = 500 (2D)





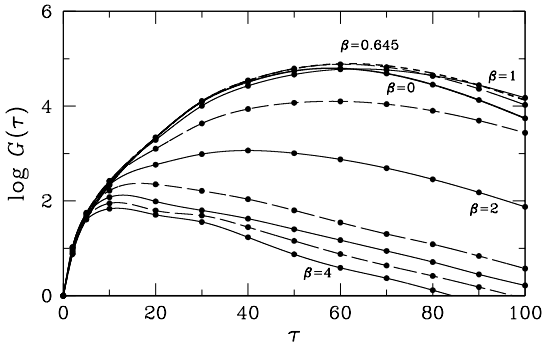


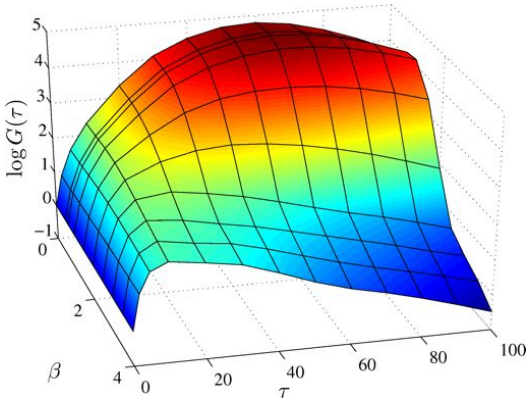


MONASH University Engineering

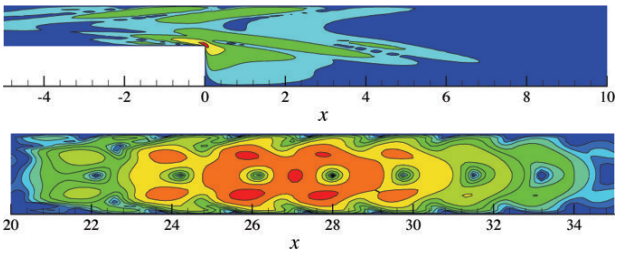
Backward-facing step, *Re* = 500 (3D)

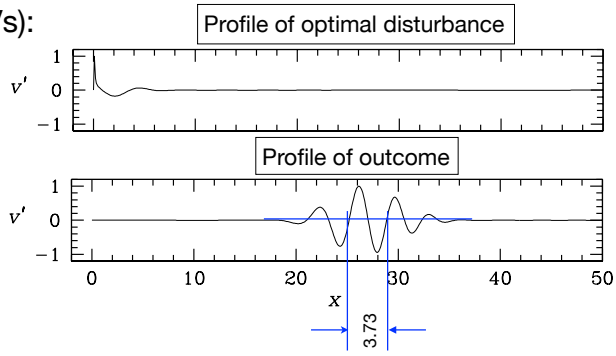
3D global optimum, *Re*=500: G_{max}=78,100, τ=62, spanwise wavenumber β=0.645.





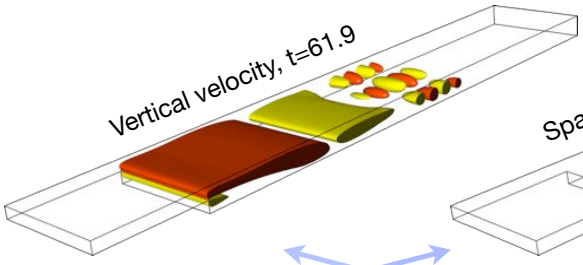
Contours of (log) energy in the global optimum initial disturbance and outcome (right & left SVs):

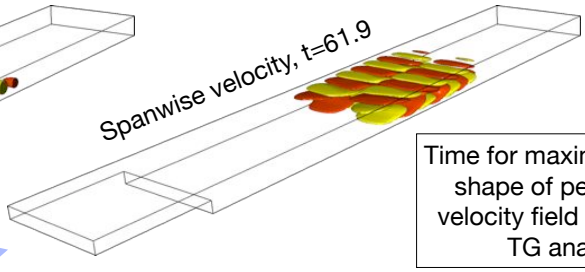




MONASH University Engineering

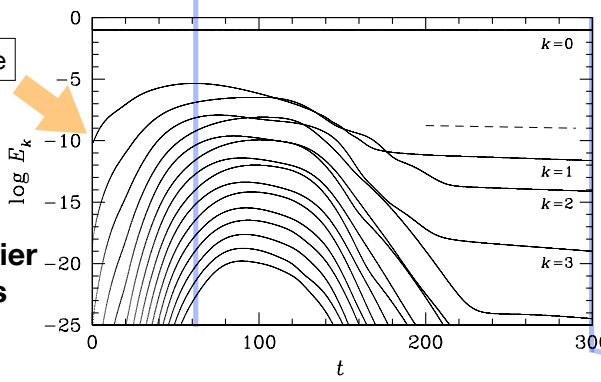
DNS with optimal initial 3D perturbation, *Re*=500





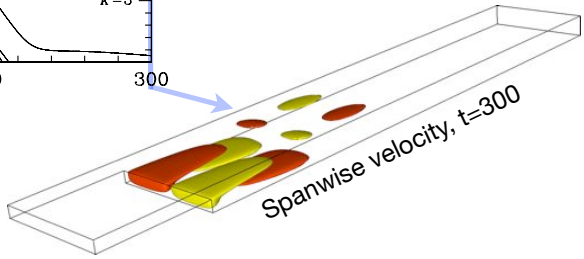
Time for maximum growth, shape of perturbation velocity field agrees with TG analysis.

Perturbed *k*=1 mode



Linear asymptotic decay agrees with leading eigenvalue of *A*.

Evolution of Fourier mode energies



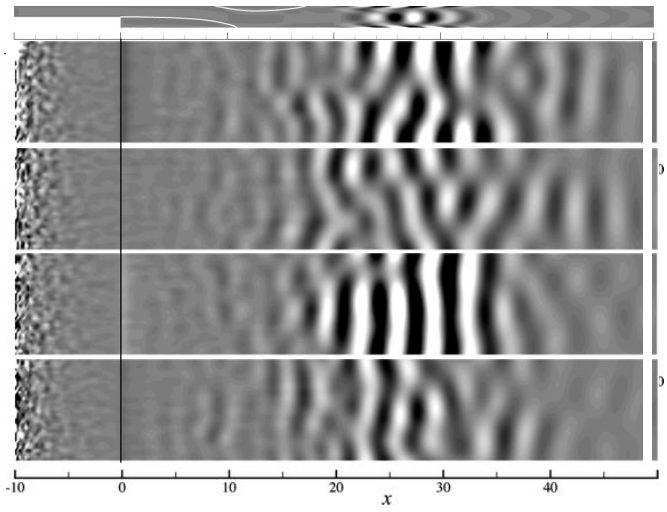
Asymptotic decay velocity field agrees with leading eigenvector of *A*.

MONASH University Engineering

DNS with white noise inflow perturbation, *Re*=500

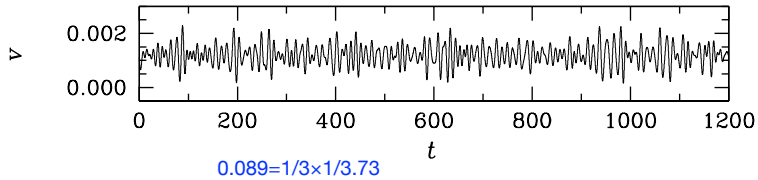
2D features

Contours of vertical velocity on plane y=0.25, 4 instants

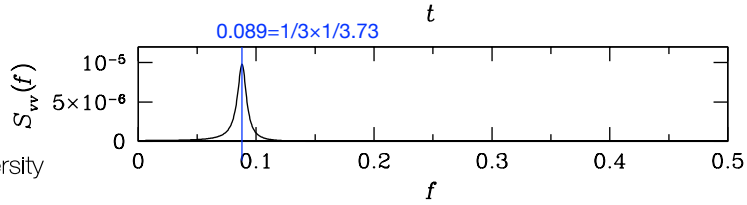


Recall wavelength≈3.73

Post-step bulk velocity=1/3



Time series of vertical velocity at x=25, y=0, z=0



Energy spectrum

Stenotic flow morphology

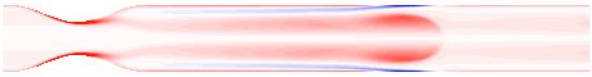
Generic features of stenotic flow:

1. Shear layers
2. Vortex rings

Parameters: Re and U_{red}

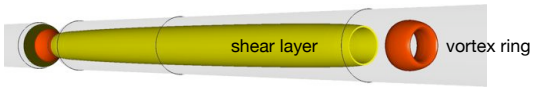
Steady base flow vorticity (shear layer)

Simple pulsatile base flow vorticity, $U_{red} = 2$



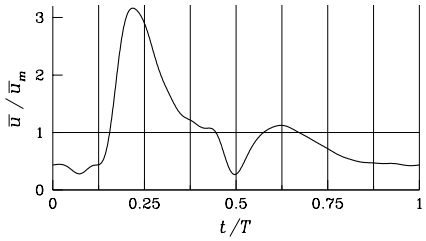
Simple pulsatile base flow vorticity, $U_{red} = 10$

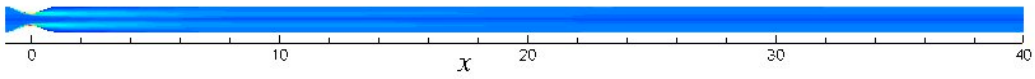
In general, as reduced velocity increases, a smaller proportion of shed vorticity is incorporated into the leading vortex ring.



Physiological type pulsatile base flow vorticity, carotid artery, $U_{red} = 27.7$

Both vortex ring(s) *and* shear layer(s) have significant vorticity.

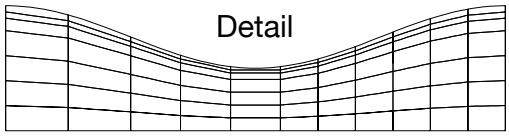




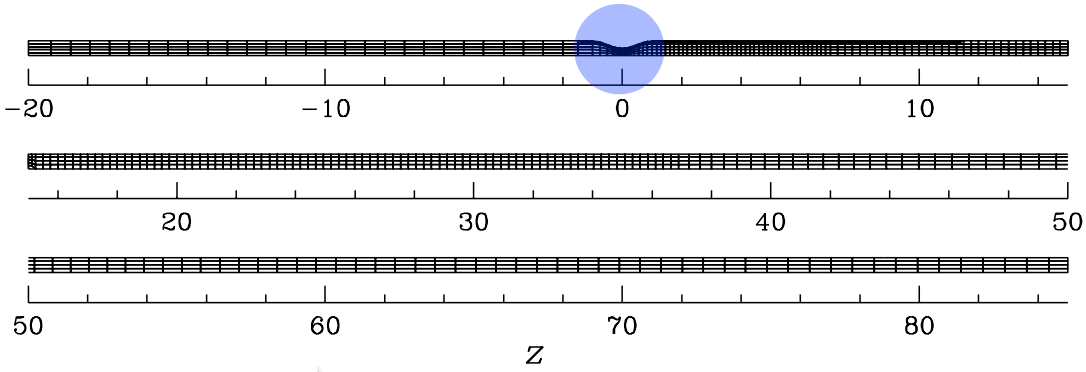
Nodal spectral element–Fourier discretization (*Semtex*)

Velocity correction scheme, cylindrical coordinates

*Blackburn & Sherwin, J Comp Phys **179**, (2004)*

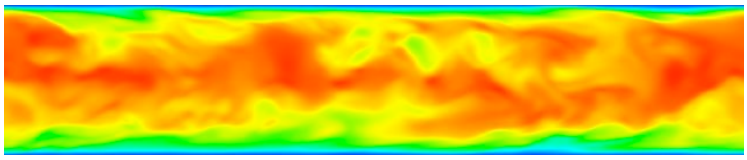


E.g.: stenosis mesh with 1336 elements, 6th-order tensor-product shape functions



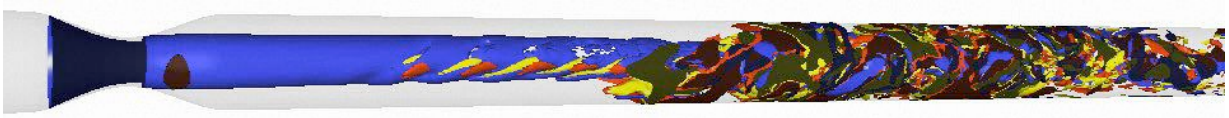
Approx 200,000 DoF – operator A has approx 40,000 million entries.

Method gives exponential convergence in all coordinates.



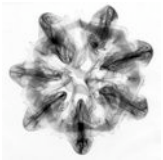
Summary of long-time/asymptotic instability results

Instability of steady flow: Re_c=722, k=1: **associated with jet/shear layer**

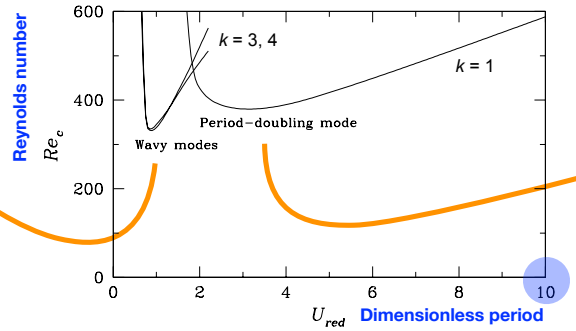


Floquet instabilities of pulsatile flows: **associated with vortex rings**

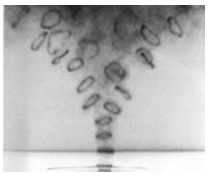
short periods



Widnall mode of isolated ring

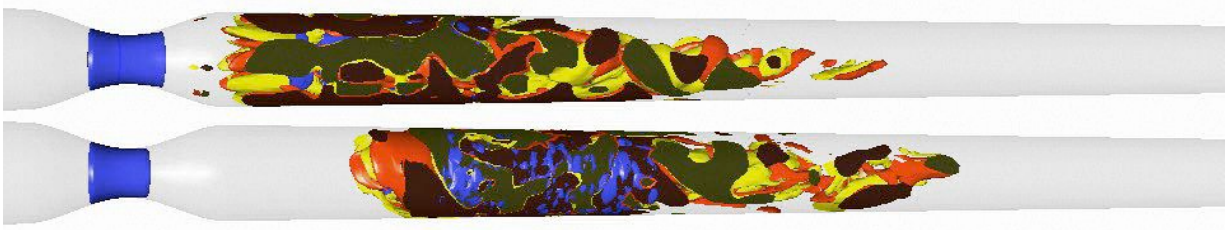


long periods

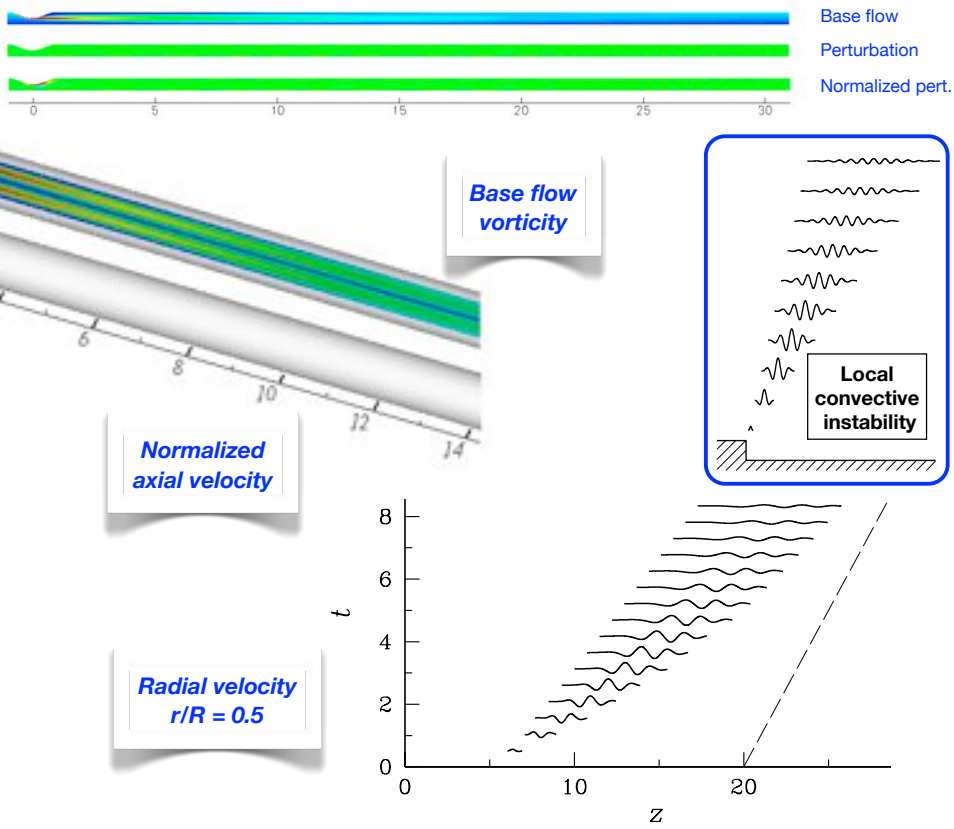


Blooming/ bifurcating jet

(period-doubling)

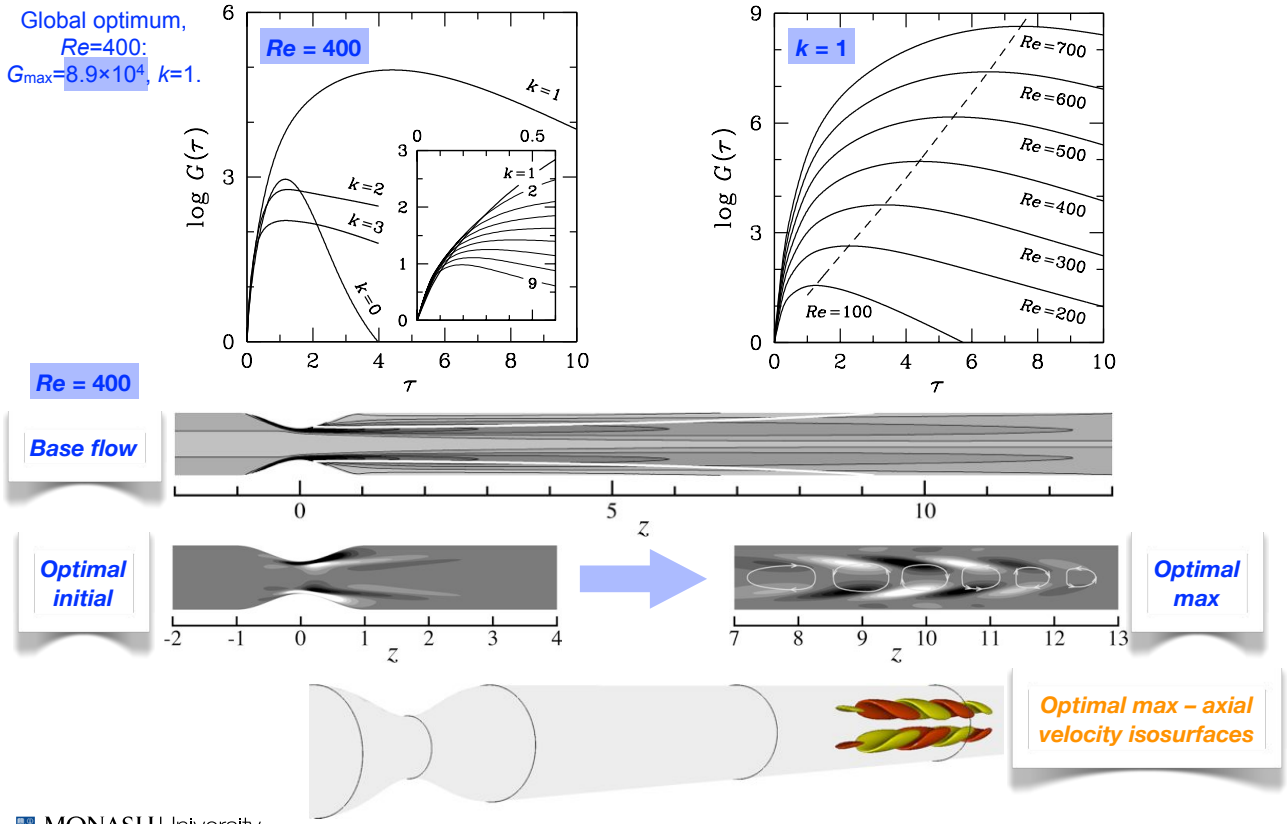


Transient growth for steady flow

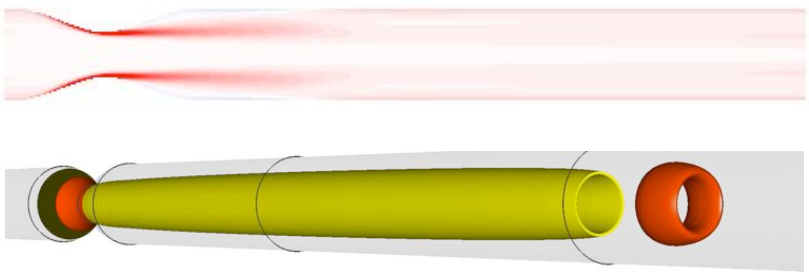


Transient growth for steady flow (Re_c = 722)

Global optimum, Re=400: G_{max}=8.9×10⁴, k=1.



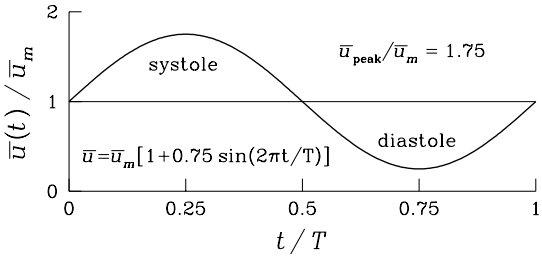
Pulsatile stenotic flow — transient growth



Pulsatile flow, *Re*=400, U_{red}=10: Global optimum G_{max}=1.1×10^{10}, *k*=1.

Extra parameter when base flow varies in time — phase t_0 at which the disturbance is initiated relative to the base flow (period *T*).

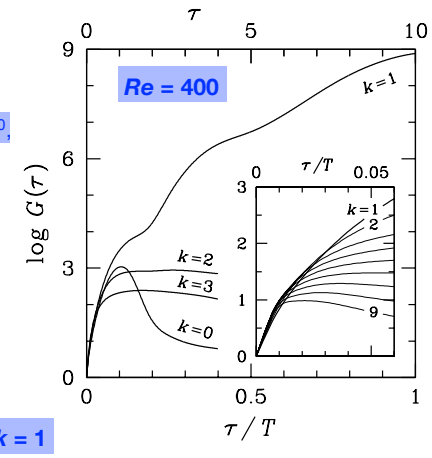
$$\bar{u}(t) = \bar{u}_m \{1 + 0.75 \sin [2\pi(t + t_0)/T]\}$$

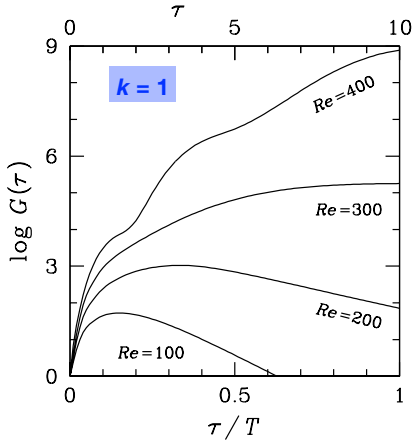


MONASH University Engineering

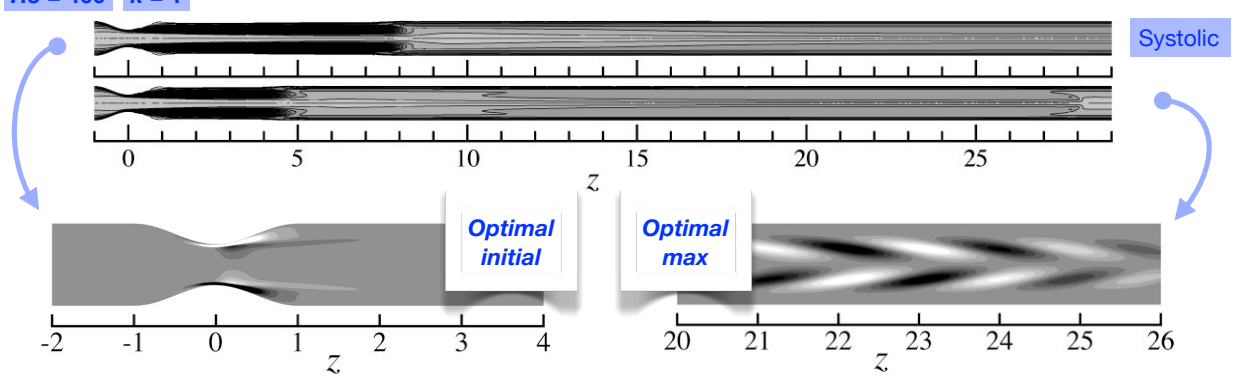
Transient growth in simple pulsatile flow, t_0 = 0

Global optimum, *Re*=400: G_{max}=1.1×10^{10}, *k*=1.



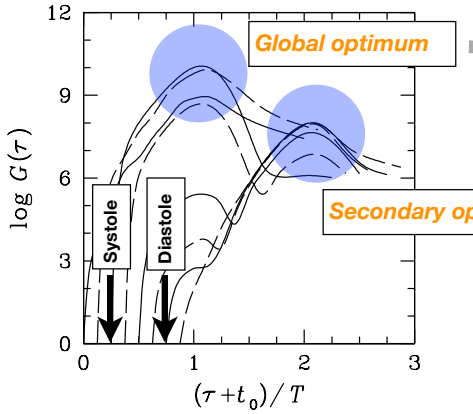


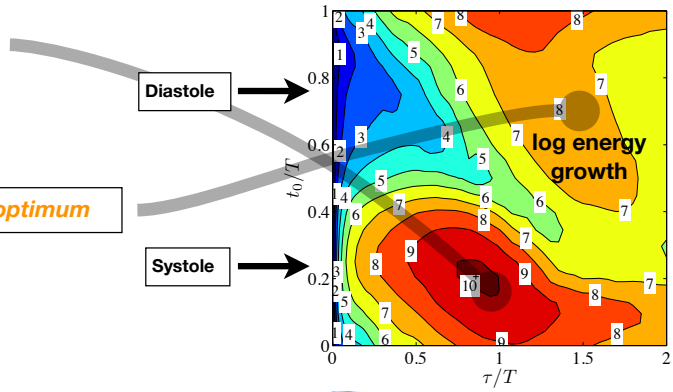
Re = 400 **k = 1**



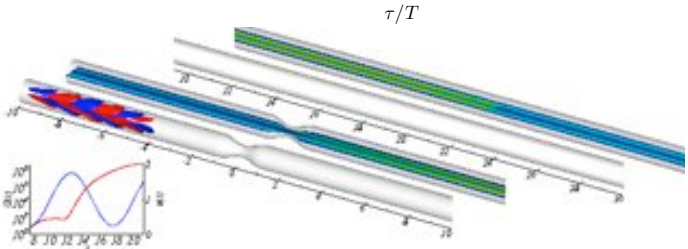
MONASH University Engineering

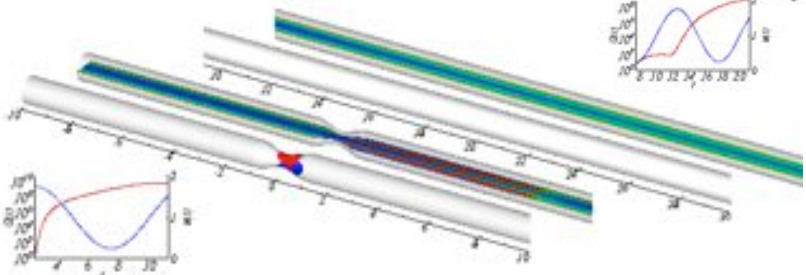
Stenotic flow — transient growth, t_0 variable





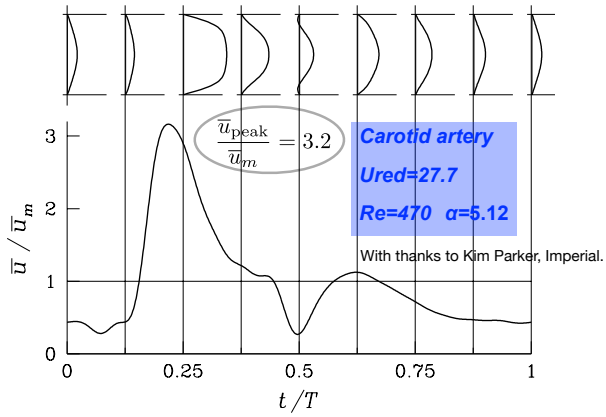
Evolution of global optimum

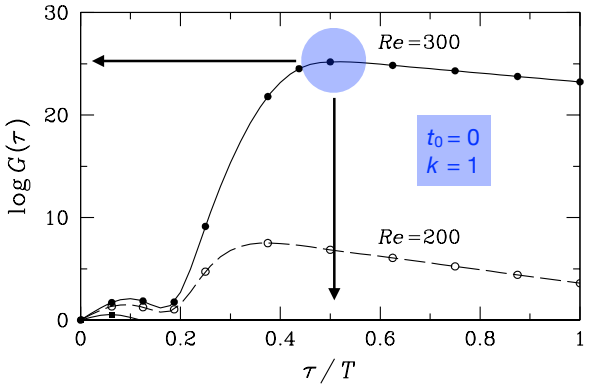


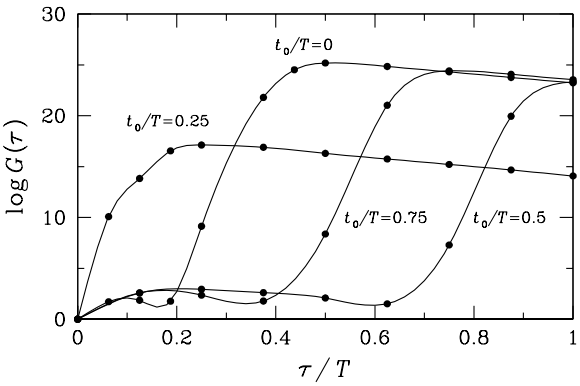


Evolution of secondary optimum

Physiological flow transient growth, (Re_c > 400)





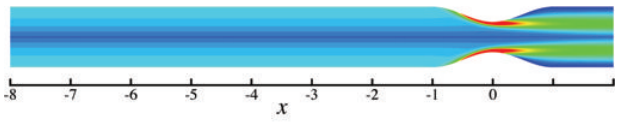


Re=300, k=1, G_{max} ≈ 1×10²⁵, i.e. velocity perturbations can grow by O(10¹²) in approx. T/2.

Linear transient growth, *Re* = 300

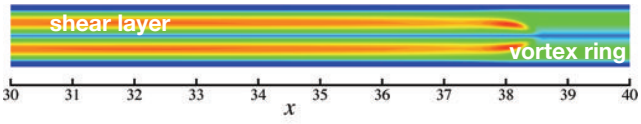
Initial condition, t = 0

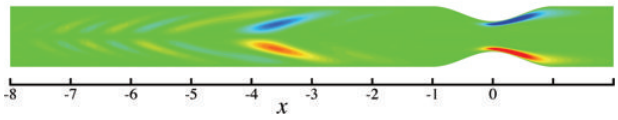
Base flow vorticity



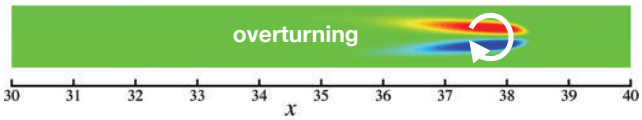
At maximum, t = τ

Base flow vorticity





Global optimum disturbance axial velocity

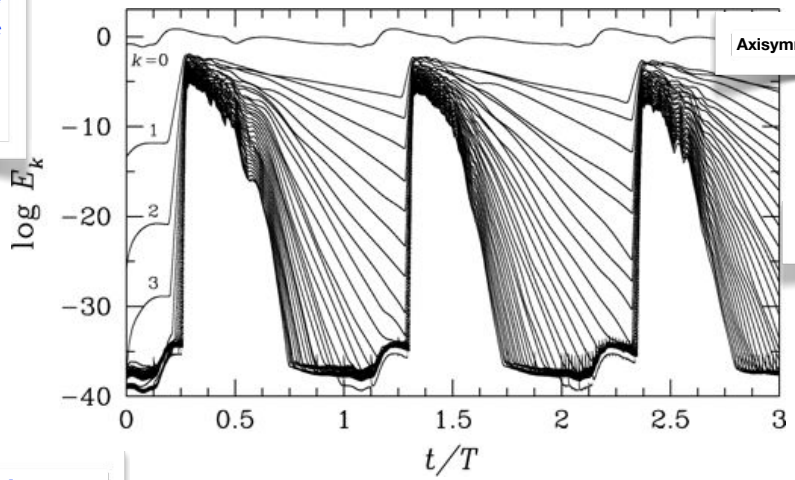


Global optimum disturbance axial velocity

Normalized isosurfaces of axial velocity

Nonlinear transient growth (DNS), *Re* = 300

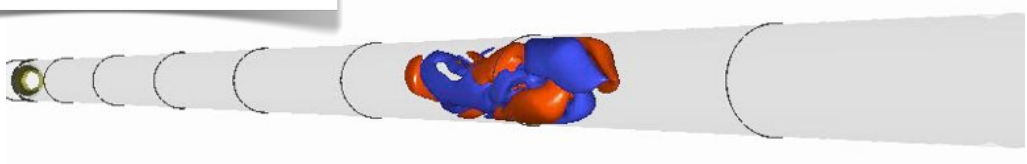
Evolution of Fourier mode energies for axisymmetric IC seeded with optimal disturbance at 10^{-12} relative energy level.



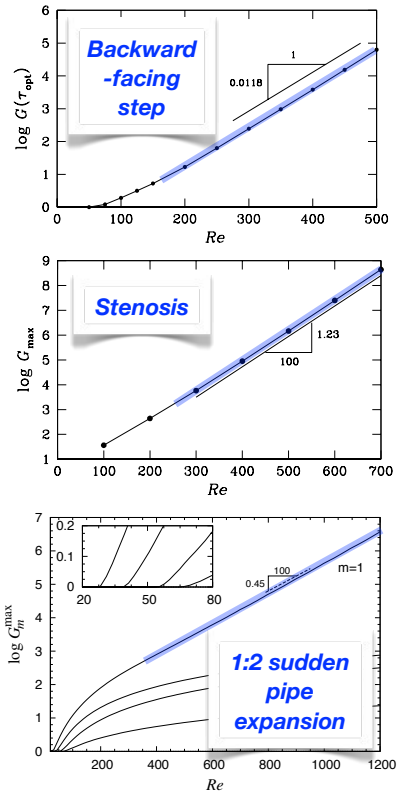
Axisymmetric component

3D disturbance

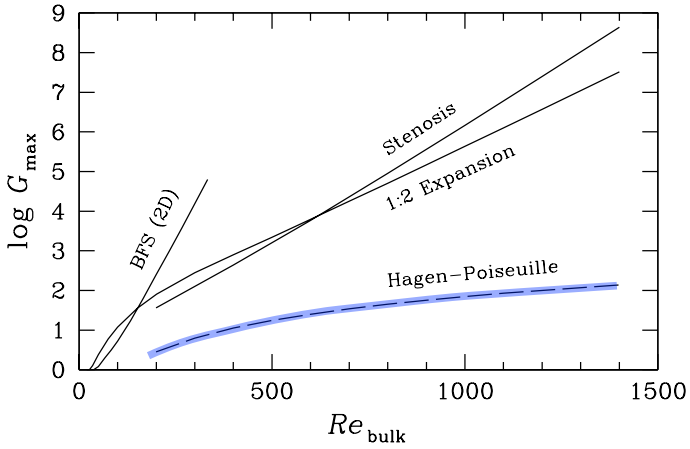
Corresponding animation: Isosurfaces of azimuthal vorticity, ± swirl velocity



Dependence of G_{max} on Re for separated flows



Steady separated shear layer flows investigated so far share common behaviour: maximum transient energy growth increases exponentially with Re (i.e. faster than any power of Re).



In parallel shear flows, maximum transient energy growth typically increases only with Re^2.

MONASH University
Engineering