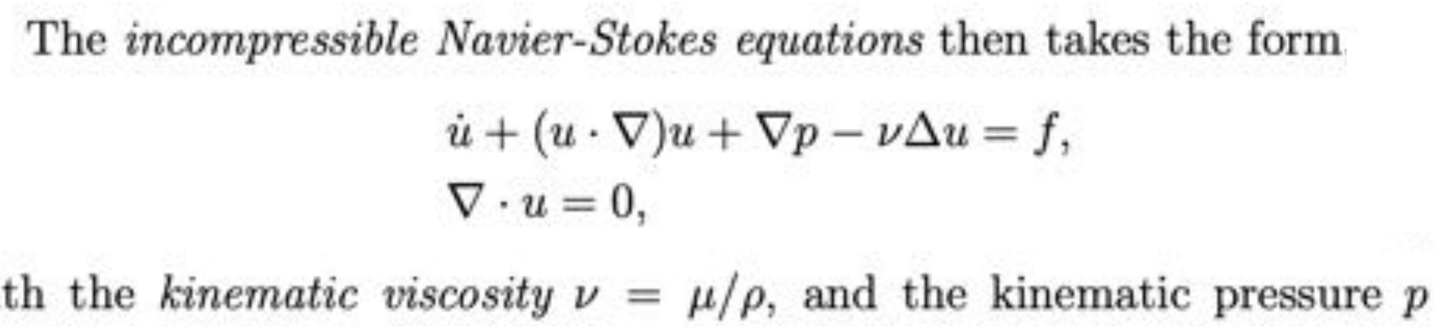
Computational Methods for Fluid Mechanics – lecture 2 The Navier-Stokes equations

Johan Hoffman

Today

- Introduction to nonlinear systems and time stepping
- Semi-discretization of the Navier-Stokes equations by a finite element method with residual-based stabilization
- The Navier-Stokes equations as a dynamical system, stability analysis and the structure of turbulent flow
- Computer simulation of turbulence

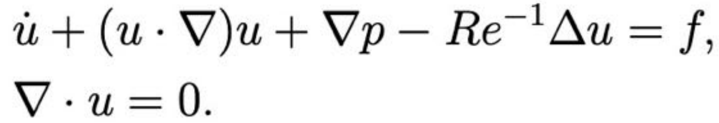
Non-dimensionalization – Reynolds number



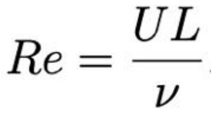
The *incompressible Navier-Stokes equations* then takes the form

$$\begin{aligned}&\dot{u} + (u \cdot \nabla)u + \nabla p - \nu \Delta u = f,\\&\nabla \cdot u = 0,\end{aligned}$$

with the *kinematic viscosity* $\nu = \mu/\rho$, and the kinematic pressure p

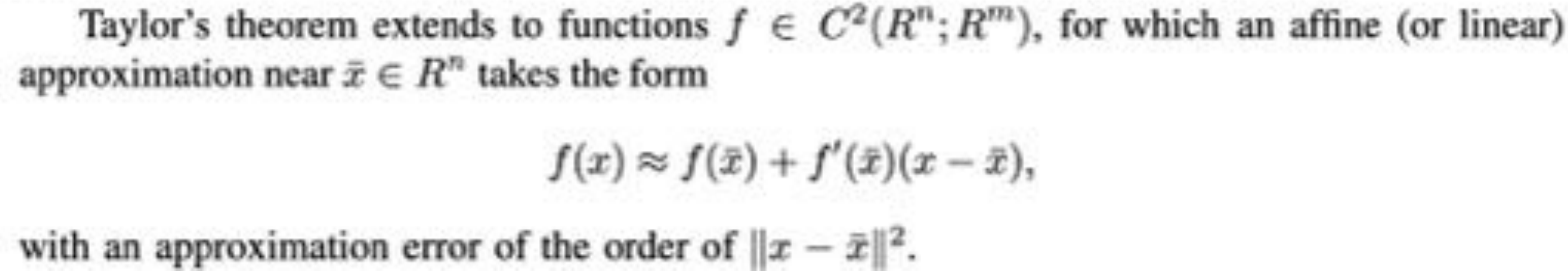


$$\begin{aligned}&\dot{u} + (u \cdot \nabla)u + \nabla p - Re^{-1} \Delta u = f,\\&\nabla \cdot u = 0.\end{aligned}$$



$$Re = \frac{UL}{\nu}$$

Nonlinear functions: linearization

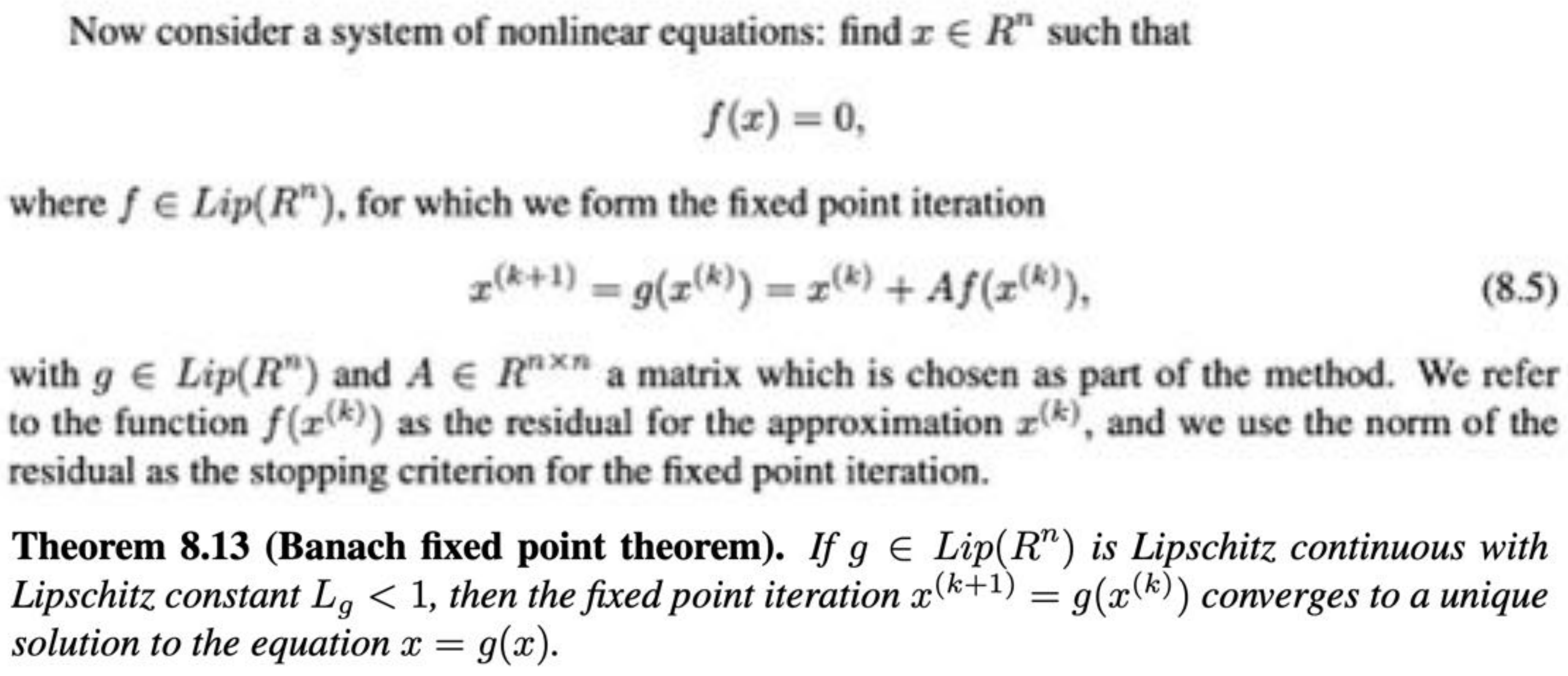


Taylor's theorem extends to functions $f \in C^2(R^n; R^m)$, for which an affine (or linear) approximation near $\bar{x} \in R^n$ takes the form

$$f(x) \approx f(\bar{x}) + f'(\bar{x})(x - \bar{x}),$$

with an approximation error of the order of $\|x - \bar{x}\|^2$.

Nonlinear equations: fixed point iteration



Now consider a system of nonlinear equations: find $x \in R^n$ such that

$$f(x) = 0,$$

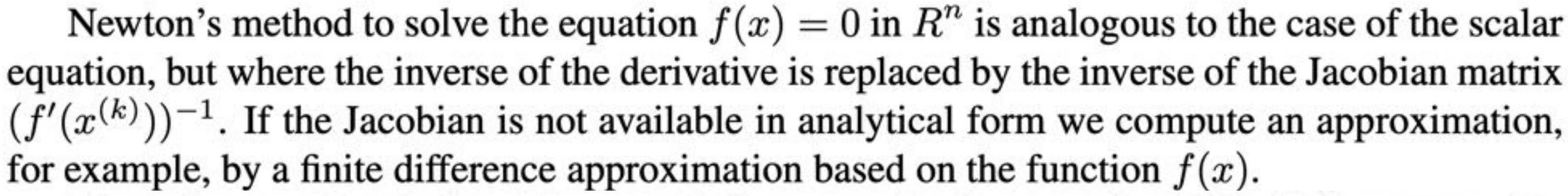
where $f \in Lip(R^n)$, for which we form the fixed point iteration

$$x^{(k+1)} = g(x^{(k)}) = x^{(k)} + Af(x^{(k)}), \tag{8.5}$$

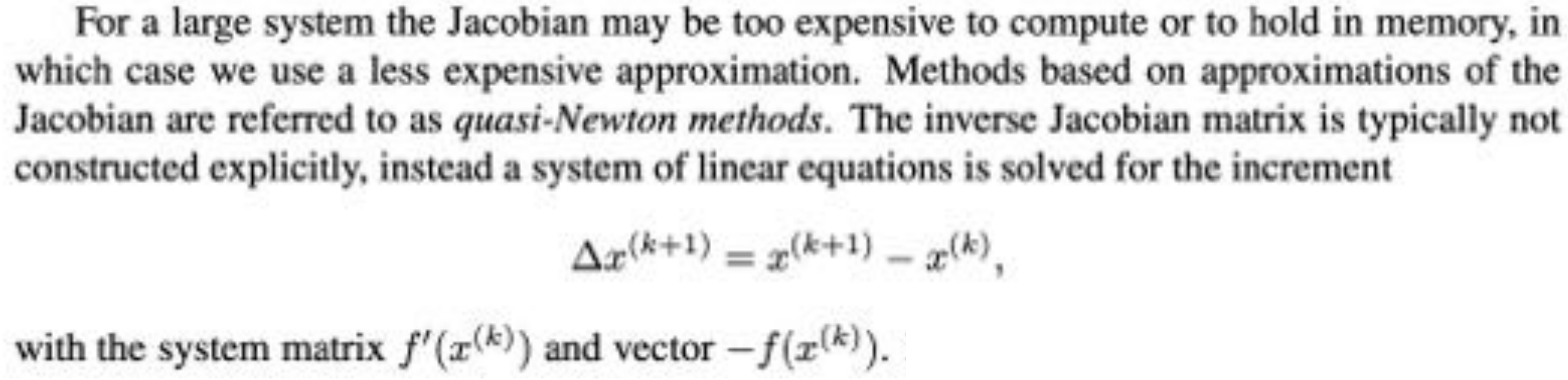
with $g \in Lip(R^n)$ and $A \in R^{n \times n}$ a matrix which is chosen as part of the method. We refer to the function $f(x^{(k)})$ as the residual for the approximation $x^{(k)}$, and we use the norm of the residual as the stopping criterion for the fixed point iteration.

Theorem 8.13 (Banach fixed point theorem). *If $g \in Lip(R^n)$ is Lipschitz continuous with Lipschitz constant $L_g < 1$, then the fixed point iteration $x^{(k+1)} = g(x^{(k)})$ converges to a unique solution to the equation $x = g(x)$.*

Nonlinear equations: Newton’s method



Newton’s method to solve the equation $f(x) = 0$ in R^n is analogous to the case of the scalar equation, but where the inverse of the derivative is replaced by the inverse of the Jacobian matrix $(f'(x^{(k)}))^{-1}$. If the Jacobian is not available in analytical form we compute an approximation, for example, by a finite difference approximation based on the function $f(x)$.



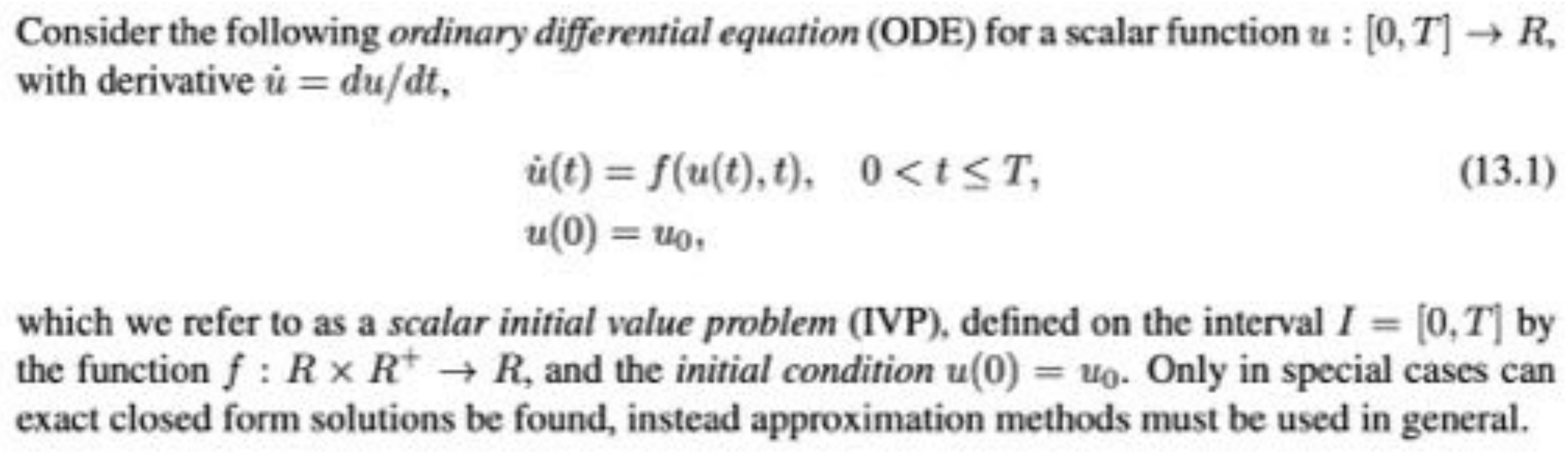
For a large system the Jacobian may be too expensive to compute or to hold in memory, in which case we use a less expensive approximation. Methods based on approximations of the Jacobian are referred to as *quasi-Newton methods*. The inverse Jacobian matrix is typically not constructed explicitly, instead a system of linear equations is solved for the increment

$$\Delta x^{(k+1)} = x^{(k+1)} - x^{(k)},$$

with the system matrix $f'(x^{(k)})$ and vector $-f(x^{(k)})$.

That is, solve the system: $f'\left(x^{(k)}\right) \Delta x^{(k+1)} = -f\left(x^{(k)}\right)$

Time discretization of initial value problem

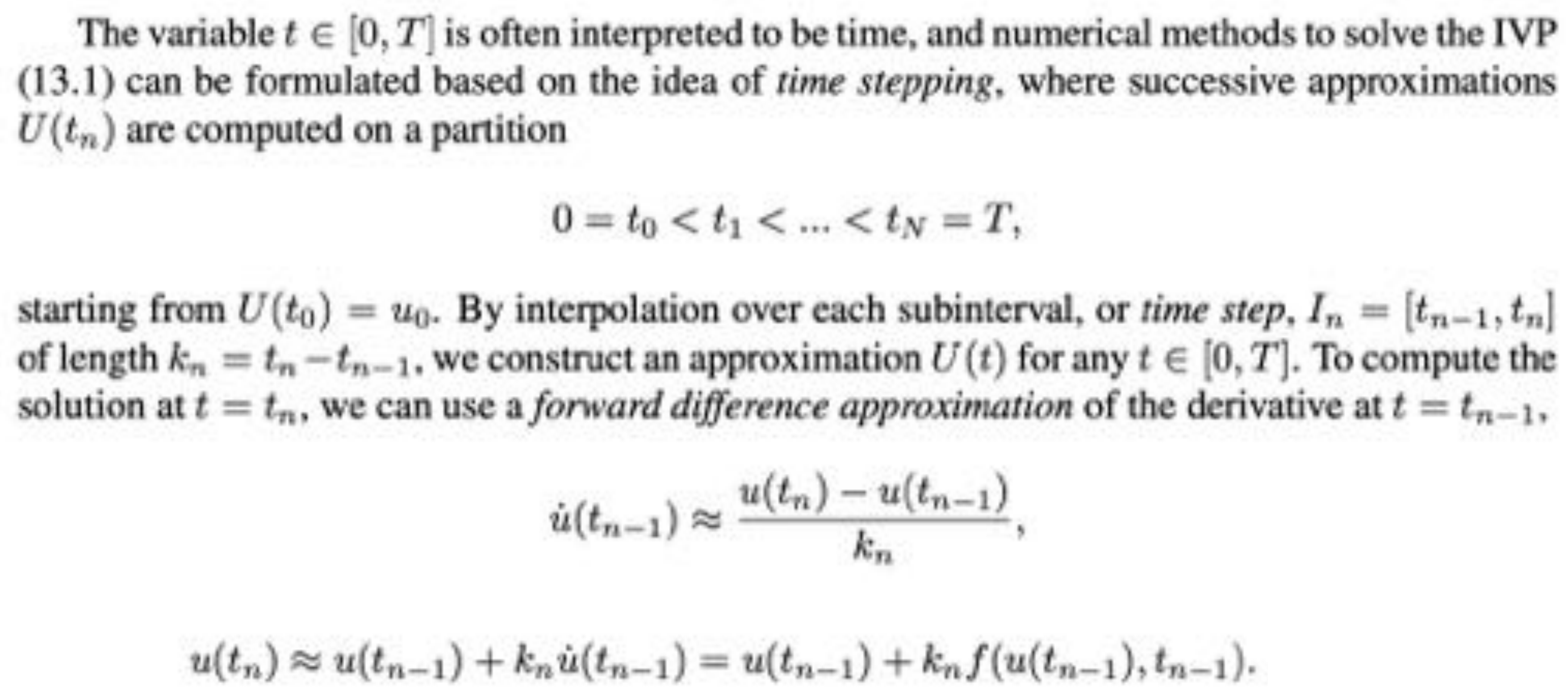


Consider the following *ordinary differential equation* (ODE) for a scalar function $u : [0, T] \rightarrow R$, with derivative $\dot{u} = du/dt$,

$$\begin{aligned} \dot{u}(t) &= f(u(t), t), \quad 0 < t \leq T, \qquad (13.1) \\ u(0) &= u_0, \end{aligned}$$

which we refer to as a *scalar initial value problem* (IVP), defined on the interval $I = [0, T]$ by the function $f : R \times R^+ \rightarrow R$, and the *initial condition* $u(0) = u_0$. Only in special cases can exact closed form solutions be found, instead approximation methods must be used in general.

Time discretization of initial value problem



The variable $t \in [0, T]$ is often interpreted to be time, and numerical methods to solve the IVP (13.1) can be formulated based on the idea of *time stepping*, where successive approximations $U(t_n)$ are computed on a partition

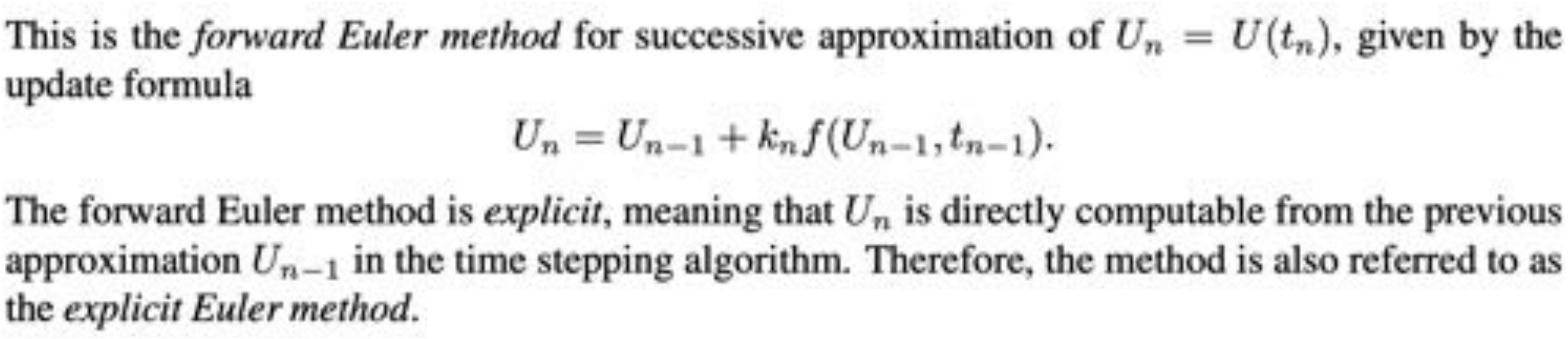
$$0 = t_0 < t_1 < ... < t_N = T,$$

starting from $U(t_0) = u_0$. By interpolation over each subinterval, or *time step*, $I_n = [t_{n-1}, t_n]$ of length $k_n = t_n - t_{n-1}$, we construct an approximation $U(t)$ for any $t \in [0, T]$. To compute the solution at $t = t_n$, we can use a *forward difference approximation* of the derivative at $t = t_{n-1}$,

$$\dot{u}(t_{n-1}) \approx \frac{u(t_n) - u(t_{n-1})}{k_n},$$

$$u(t_n) \approx u(t_{n-1}) + k_n \dot{u}(t_{n-1}) = u(t_{n-1}) + k_n f(u(t_{n-1}), t_{n-1}).$$

Forward Euler method

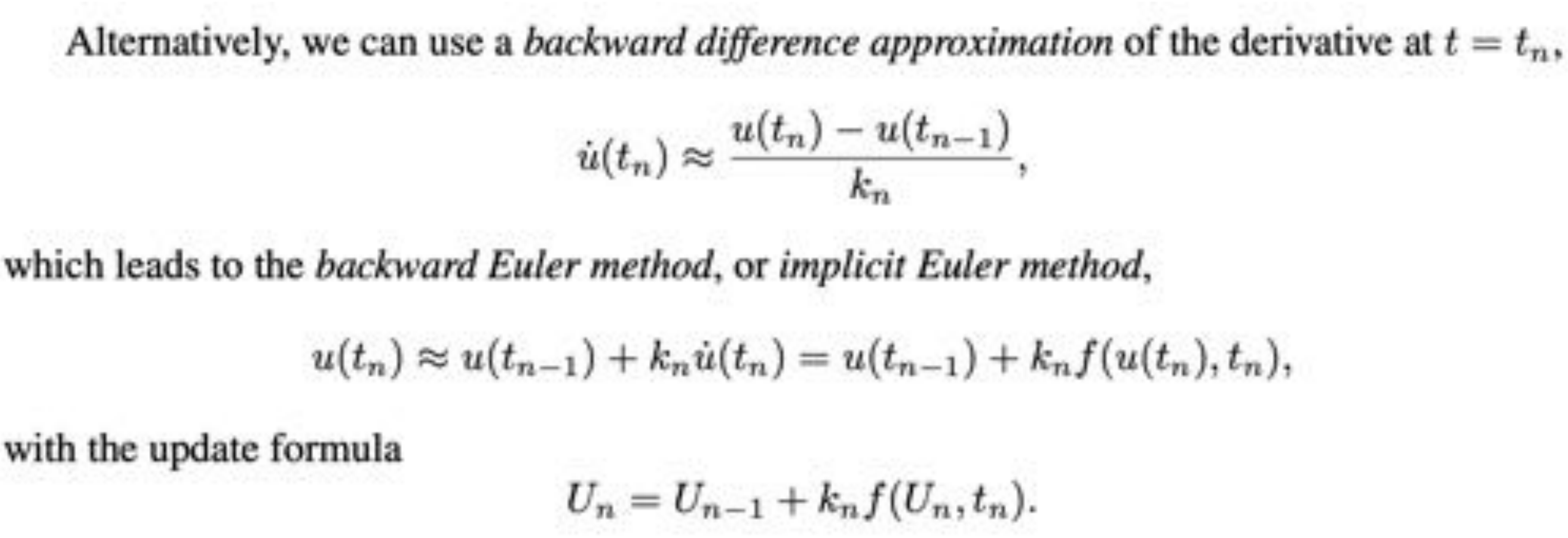


This is the *forward Euler method* for successive approximation of $U_n = U(t_n)$, given by the update formula

$$U_n = U_{n-1} + k_n f(U_{n-1}, t_{n-1}).$$

The forward Euler method is *explicit*, meaning that U_n is directly computable from the previous approximation U_{n-1} in the time stepping algorithm. Therefore, the method is also referred to as the *explicit Euler method*.

Backward Euler method



Alternatively, we can use a *backward difference approximation* of the derivative at $t = t_n$,

$$\dot{u}(t_n) \approx \frac{u(t_n) - u(t_{n-1})}{k_n},$$

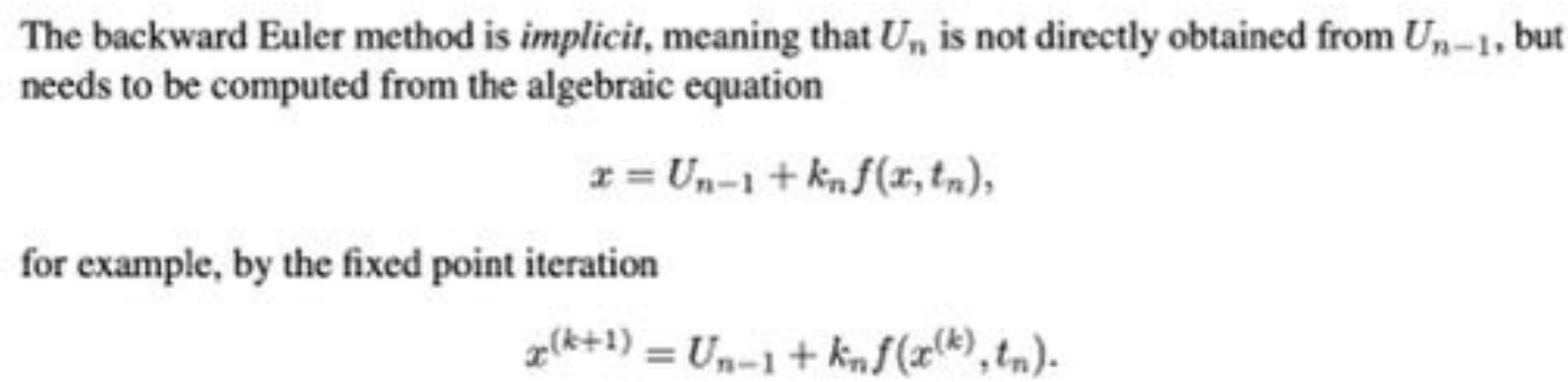
which leads to the *backward Euler method*, or *implicit Euler method*,

$$u(t_n) \approx u(t_{n-1}) + k_n \dot{u}(t_n) = u(t_{n-1}) + k_n f(u(t_n), t_n),$$

with the update formula

$$U_n = U_{n-1} + k_n f(U_n, t_n).$$

Backward Euler method



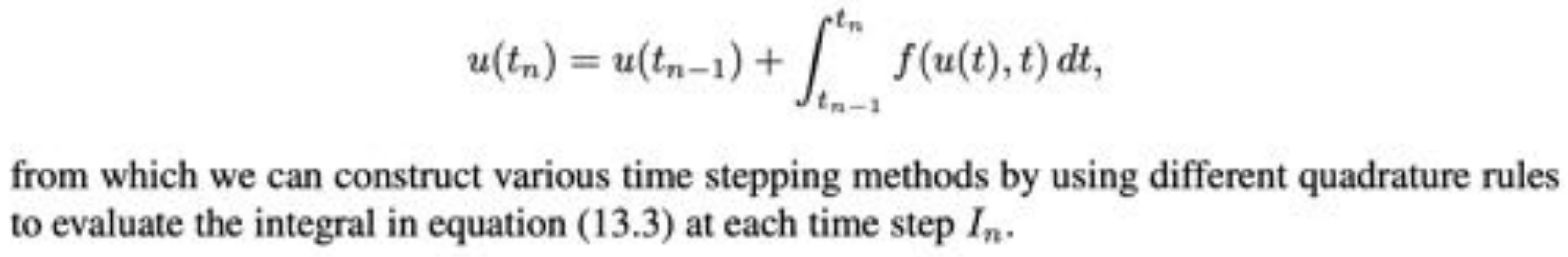
The backward Euler method is *implicit*, meaning that U_n is not directly obtained from U_{n-1}, but needs to be computed from the algebraic equation

$$x = U_{n-1} + k_n f(x, t_n),$$

for example, by the fixed point iteration

$$x^{(k+1)} = U_{n-1} + k_n f(x^{(k)}, t_n).$$

Time stepping methods as quadrature rules



$$u(t_n) = u(t_{n-1}) + \int_{t_{n-1}}^{t_n} f(u(t), t)\, dt,$$

from which we can construct various time stepping methods by using different quadrature rules to evaluate the integral in equation (13.3) at each time step I_n.

Euler methods as quadrature rules

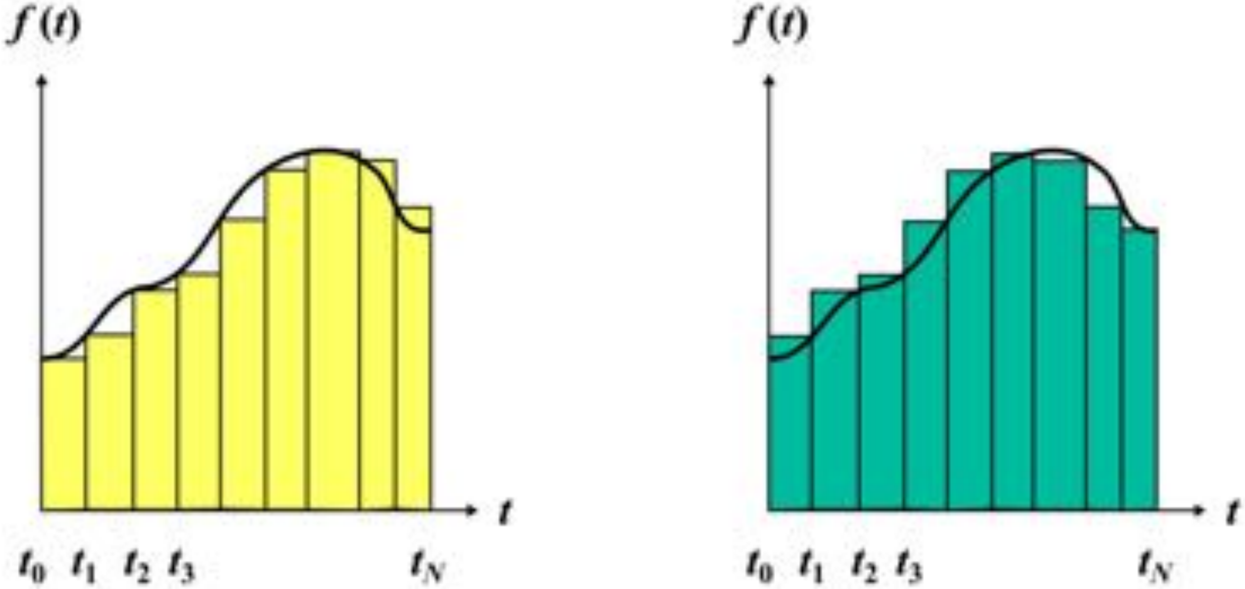


Figure 13.1. *Left (left) and right (right) Riemann sums which approximate the integral of $f(t)$, corresponding to the explicit and implicit Euler time stepping method for approximation of the IVP (13.1) with $f(u(t),t) = f(t)$.*

Midpoint and trapezoidal methods

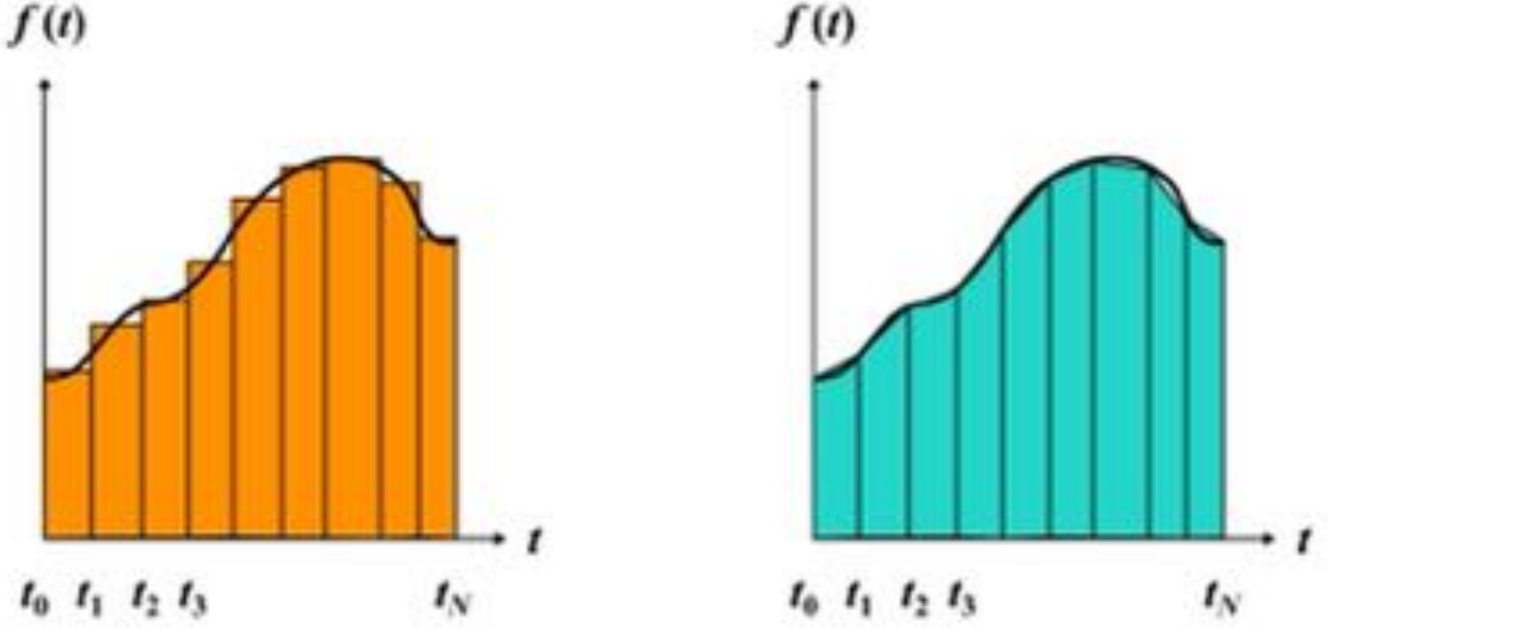
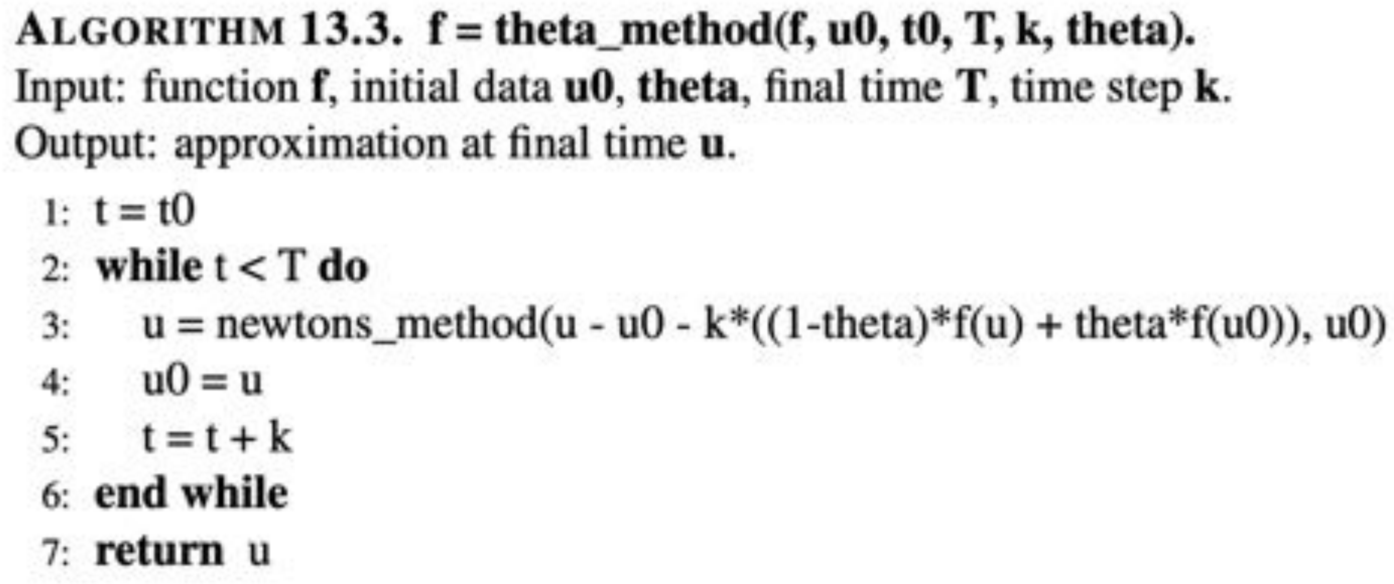


Figure 13.2. *The midpoint (left) and the trapezoidal (right) quadrature rules, corresponding to interpolation by a piecewise constant and piecewise linear function respectively.*

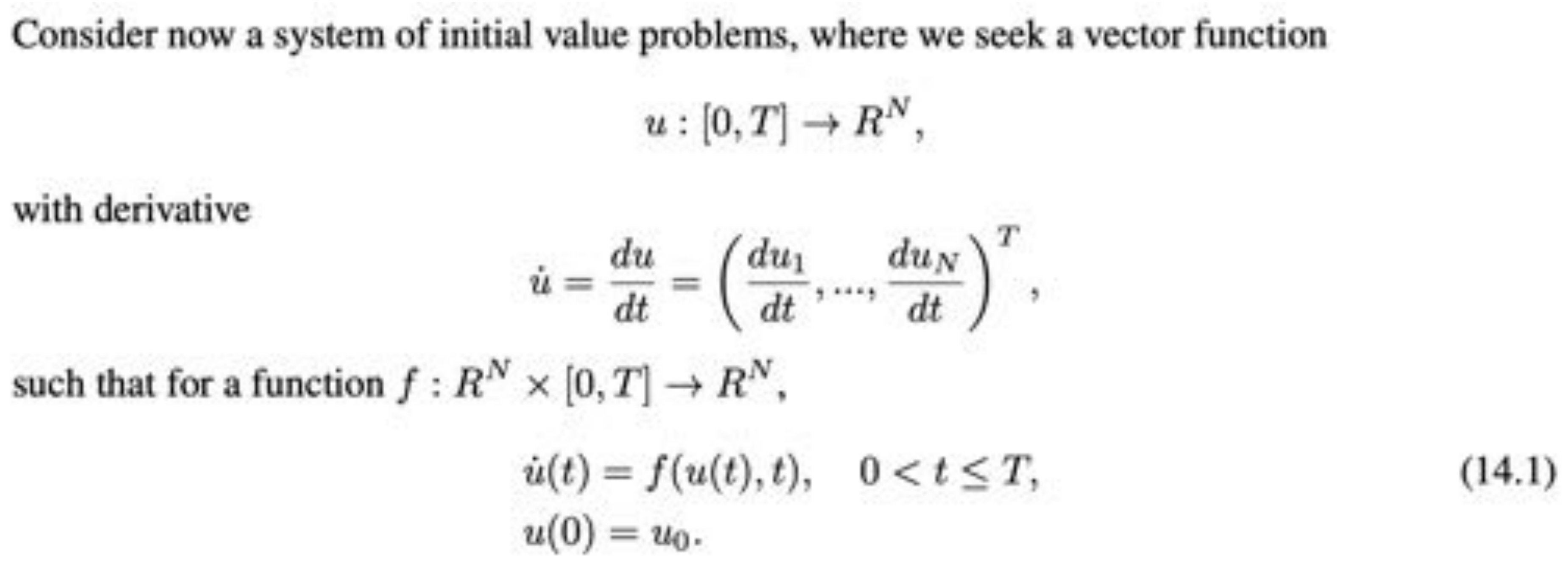
Theta method (F/B Euler, trapezoidal,…)



ALGORITHM 13.3. f = theta_method(f, u0, t0, T, k, theta).
Input: function **f**, initial data **u0**, **theta**, final time **T**, time step **k**.
Output: approximation at final time **u**.

```
t = t0
while t < T do
    u = newtons_method(u - u0 - k*((1-theta)*f(u) + theta*f(u0)), u0)
    u0 = u
    t = t + k
end while
return u
```

Same time stepping methods for IVP systems



Consider now a system of initial value problems, where we seek a vector function

$$u : [0, T] \to R^N,$$

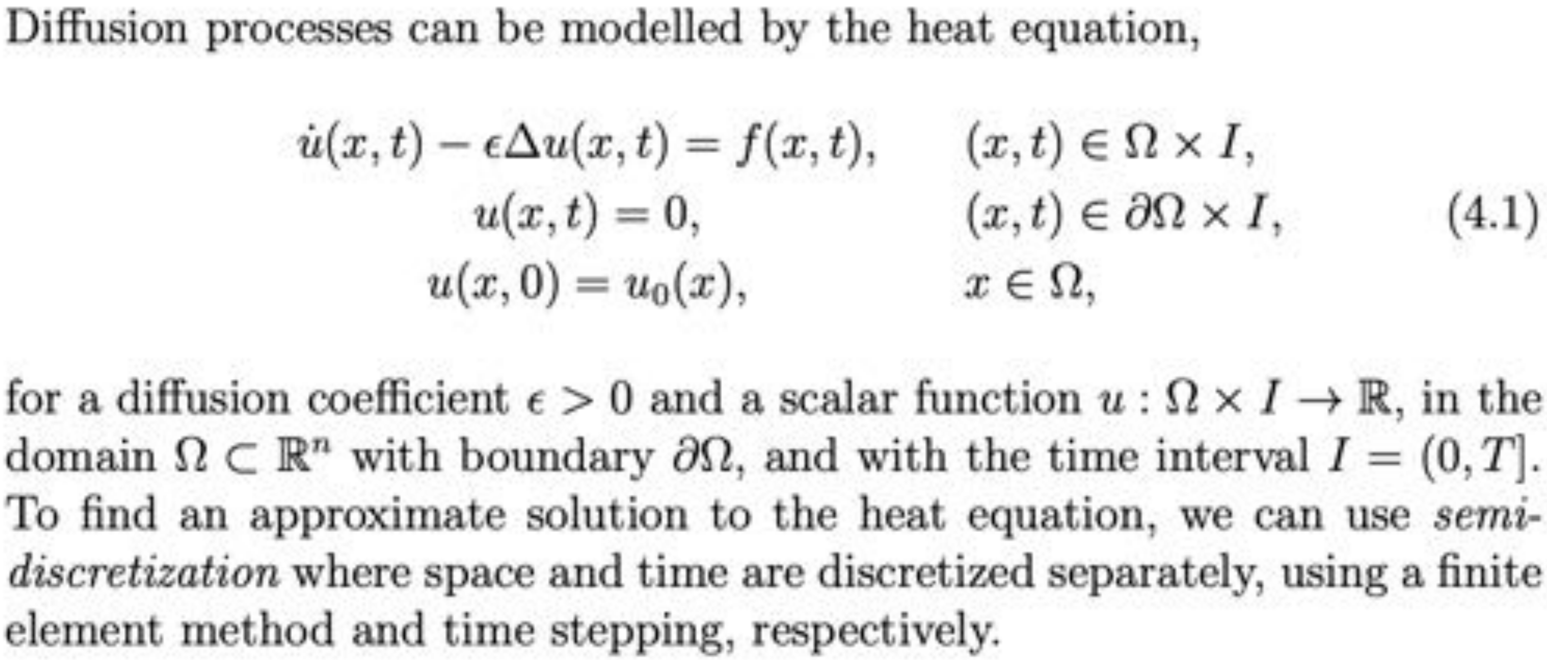
with derivative

$$\dot{u} = \frac{du}{dt} = \left(\frac{du_1}{dt}, \ldots, \frac{du_N}{dt} \right)^T,$$

such that for a function $f : R^N \times [0, T] \to R^N$,

$$\begin{aligned} \dot{u}(t) &= f(u(t), t), \quad 0 < t \leq T, \\ u(0) &= u_0. \end{aligned} \tag{14.1}$$

Partial differential equation: heat equation

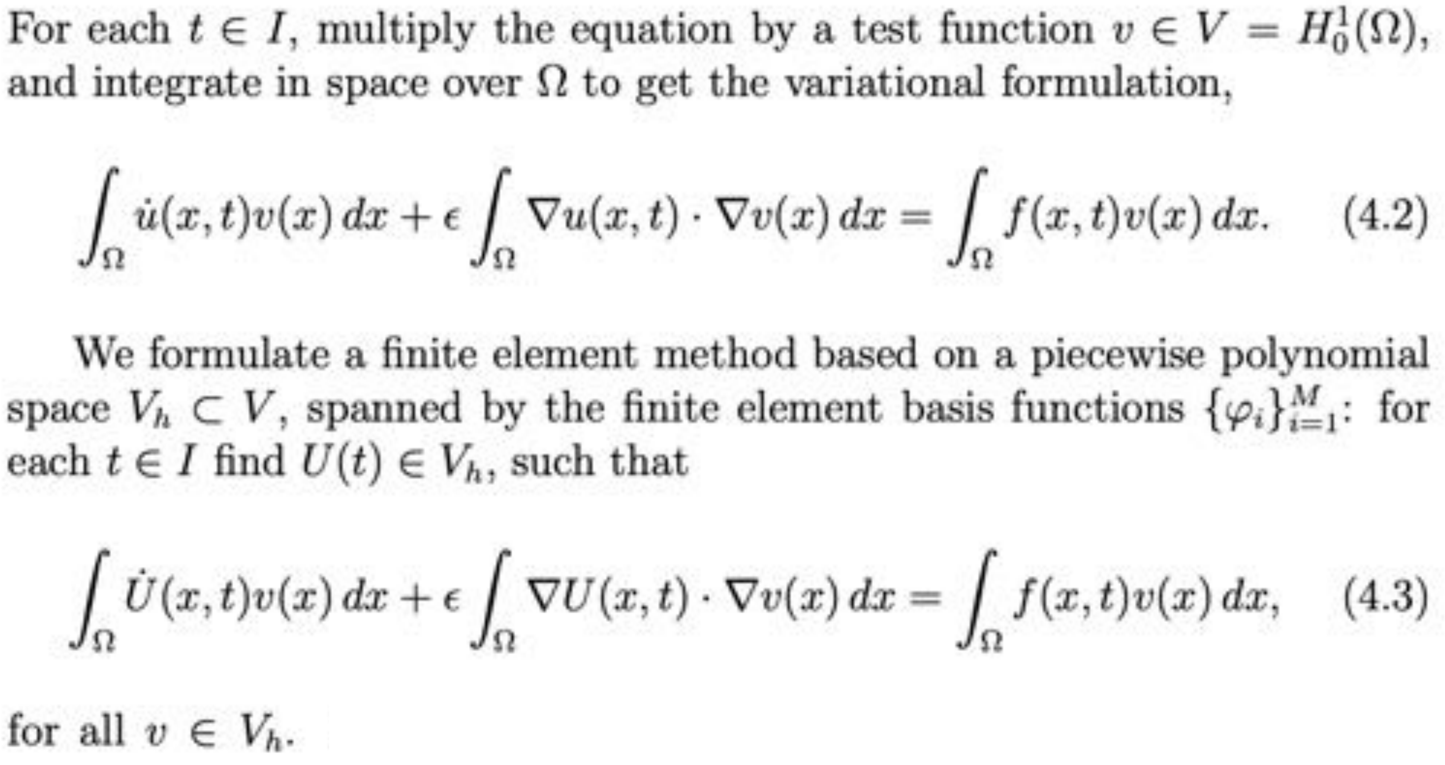


Diffusion processes can be modelled by the heat equation,

$$\begin{aligned} \dot{u}(x,t) - \epsilon\Delta u(x,t) &= f(x,t), && (x,t) \in \Omega \times I, \\ u(x,t) &= 0, && (x,t) \in \partial\Omega \times I, \\ u(x,0) &= u_0(x), && x \in \Omega, \end{aligned} \tag{4.1}$$

for a diffusion coefficient $\epsilon > 0$ and a scalar function $u : \Omega \times I \to \mathbb{R}$, in the domain $\Omega \subset \mathbb{R}^n$ with boundary $\partial\Omega$, and with the time interval $I = (0,T]$. To find an approximate solution to the heat equation, we can use *semi-discretization* where space and time are discretized separately, using a finite element method and time stepping, respectively.

Heat equation: variational form and FEM



For each $t \in I$, multiply the equation by a test function $v \in V = H_0^1(\Omega)$, and integrate in space over Ω to get the variational formulation,

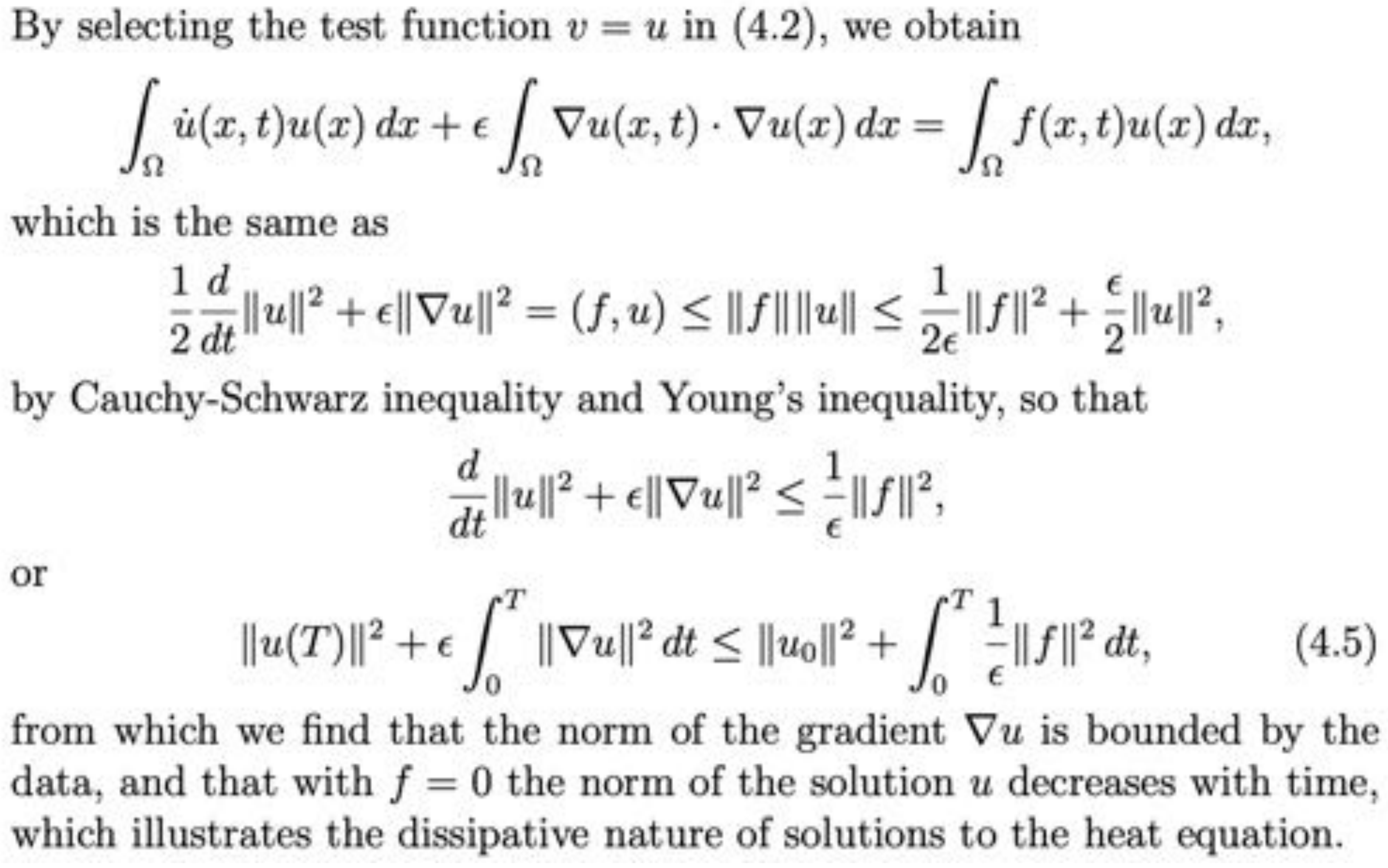
$$\int_\Omega \dot{u}(x,t)v(x)\,dx + \epsilon \int_\Omega \nabla u(x,t) \cdot \nabla v(x)\,dx = \int_\Omega f(x,t)v(x)\,dx. \qquad (4.2)$$

We formulate a finite element method based on a piecewise polynomial space $V_h \subset V$, spanned by the finite element basis functions $\{\varphi_i\}_{i=1}^M$: for each $t \in I$ find $U(t) \in V_h$, such that

$$\int_\Omega \dot{U}(x,t)v(x)\,dx + \epsilon \int_\Omega \nabla U(x,t) \cdot \nabla v(x)\,dx = \int_\Omega f(x,t)v(x)\,dx, \qquad (4.3)$$

for all $v \in V_h$.

Heat equation: stability estimate



By selecting the test function $v = u$ in (4.2), we obtain

$$\int_\Omega \dot{u}(x,t)u(x)\,dx + \epsilon \int_\Omega \nabla u(x,t)\cdot\nabla u(x)\,dx = \int_\Omega f(x,t)u(x)\,dx,$$

which is the same as

$$\frac{1}{2}\frac{d}{dt}\|u\|^2 + \epsilon\|\nabla u\|^2 = (f,u) \le \|f\|\|u\| \le \frac{1}{2\epsilon}\|f\|^2 + \frac{\epsilon}{2}\|u\|^2,$$

by Cauchy-Schwarz inequality and Young's inequality, so that

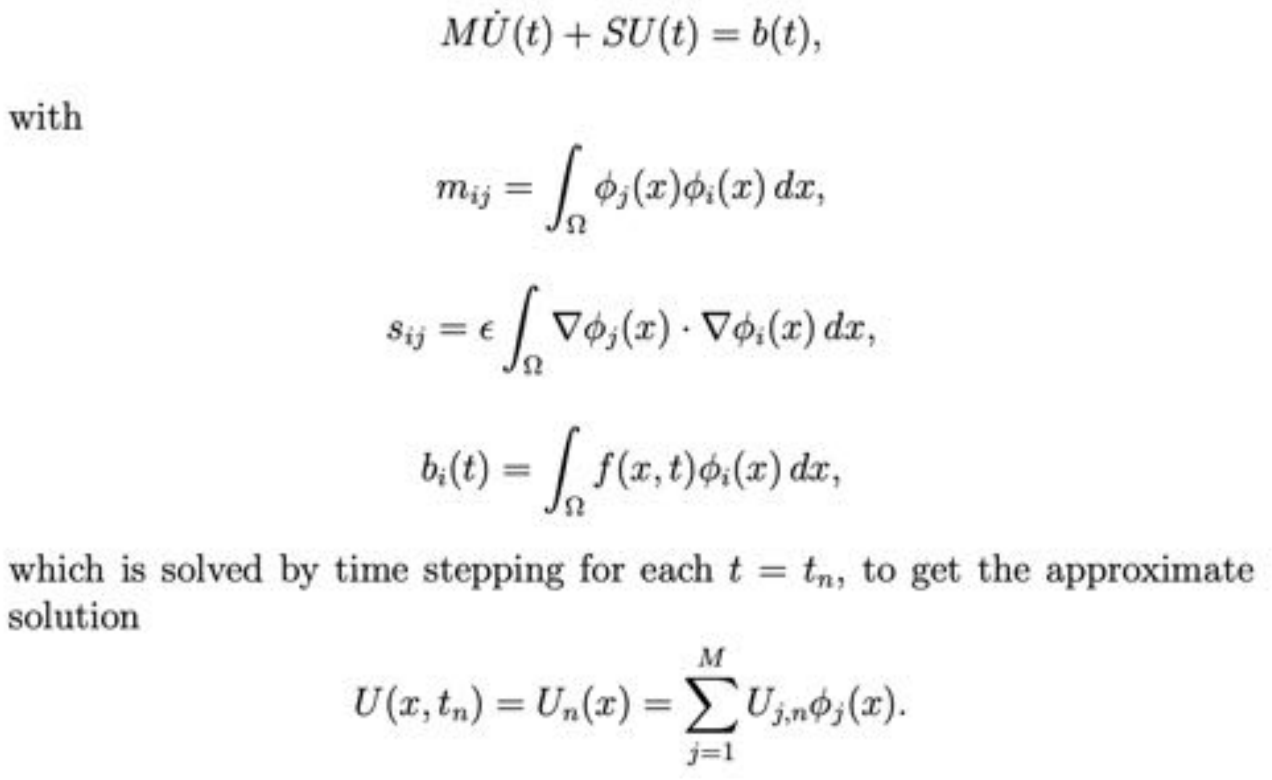
$$\frac{d}{dt}\|u\|^2 + \epsilon\|\nabla u\|^2 \le \frac{1}{\epsilon}\|f\|^2,$$

or

$$\|u(T)\|^2 + \epsilon\int_0^T \|\nabla u\|^2\,dt \le \|u_0\|^2 + \int_0^T \frac{1}{\epsilon}\|f\|^2\,dt, \qquad (4.5)$$

from which we find that the norm of the gradient ∇u is bounded by the data, and that with $f = 0$ the norm of the solution u decreases with time, which illustrates the dissipative nature of solutions to the heat equation.

Heat equation: semi-discrete IVP system



$$M\dot{U}(t) + SU(t) = b(t),$$

with

$$m_{ij} = \int_\Omega \phi_j(x)\phi_i(x)\,dx,$$

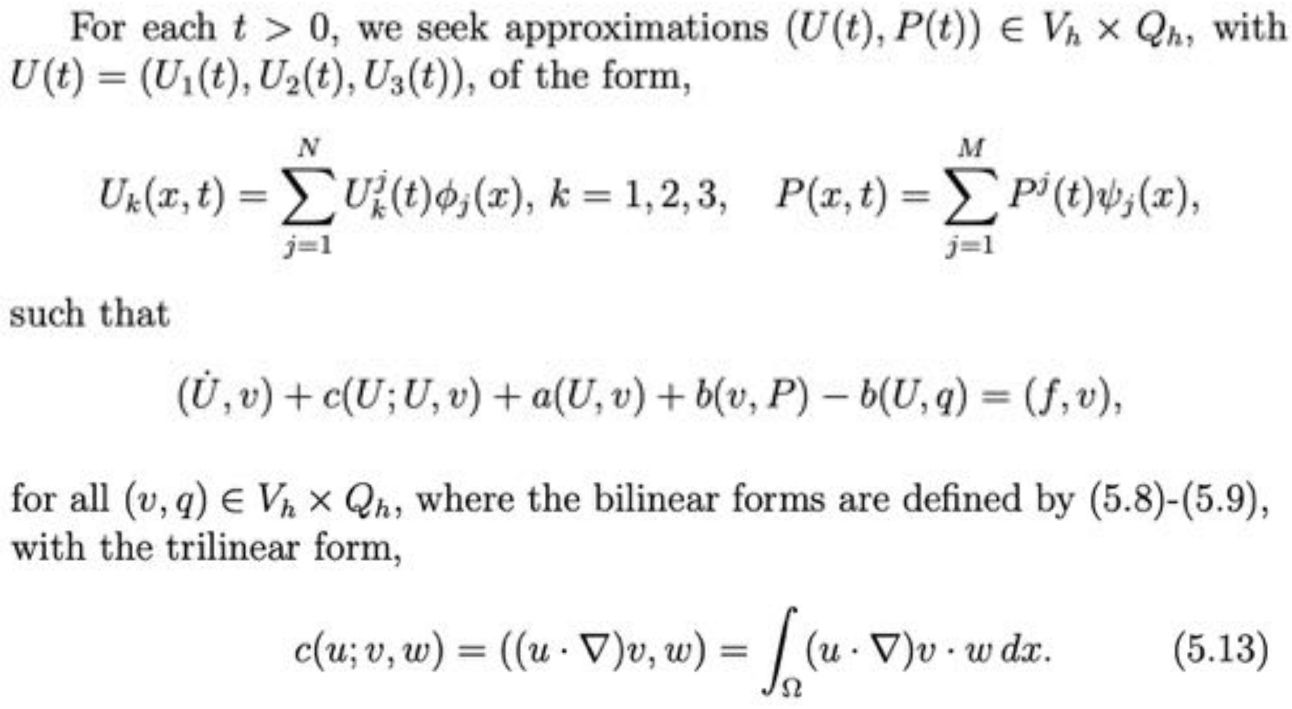
$$s_{ij} = \epsilon \int_\Omega \nabla\phi_j(x) \cdot \nabla\phi_i(x)\,dx,$$

$$b_i(t) = \int_\Omega f(x,t)\phi_i(x)\,dx,$$

which is solved by time stepping for each $t = t_n$, to get the approximate solution

$$U(x,t_n) = U_n(x) = \sum_{j=1}^{M} U_{j,n}\phi_j(x).$$

Navier-Stokes equations: FEM



For each $t > 0$, we seek approximations $(U(t), P(t)) \in V_h \times Q_h$, with $U(t) = (U_1(t), U_2(t), U_3(t))$, of the form,

$$U_k(x,t) = \sum_{j=1}^{N} U_k^j(t)\phi_j(x),\ k = 1,2,3, \quad P(x,t) = \sum_{j=1}^{M} P^j(t)\psi_j(x),$$

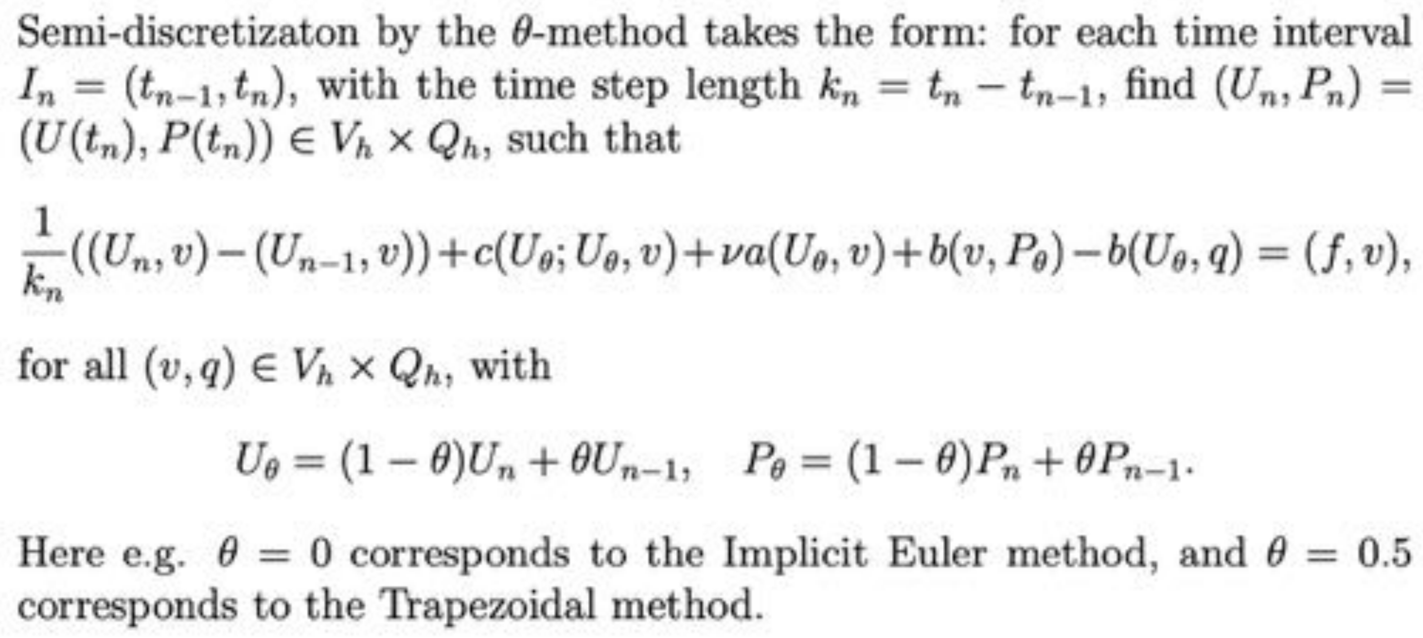
such that

$$(\dot{U}, v) + c(U; U, v) + a(U, v) + b(v, P) - b(U, q) = (f, v),$$

for all $(v, q) \in V_h \times Q_h$, where the bilinear forms are defined by (5.8)-(5.9), with the trilinear form,

$$c(u; v, w) = ((u \cdot \nabla)v, w) = \int_\Omega (u \cdot \nabla)v \cdot w \, dx. \qquad (5.13)$$

FEM semi-discretization of IVP system



Semi-discretizaton by the θ-method takes the form: for each time interval $I_n = (t_{n-1}, t_n)$, with the time step length $k_n = t_n - t_{n-1}$, find $(U_n, P_n) = (U(t_n), P(t_n)) \in V_h \times Q_h$, such that

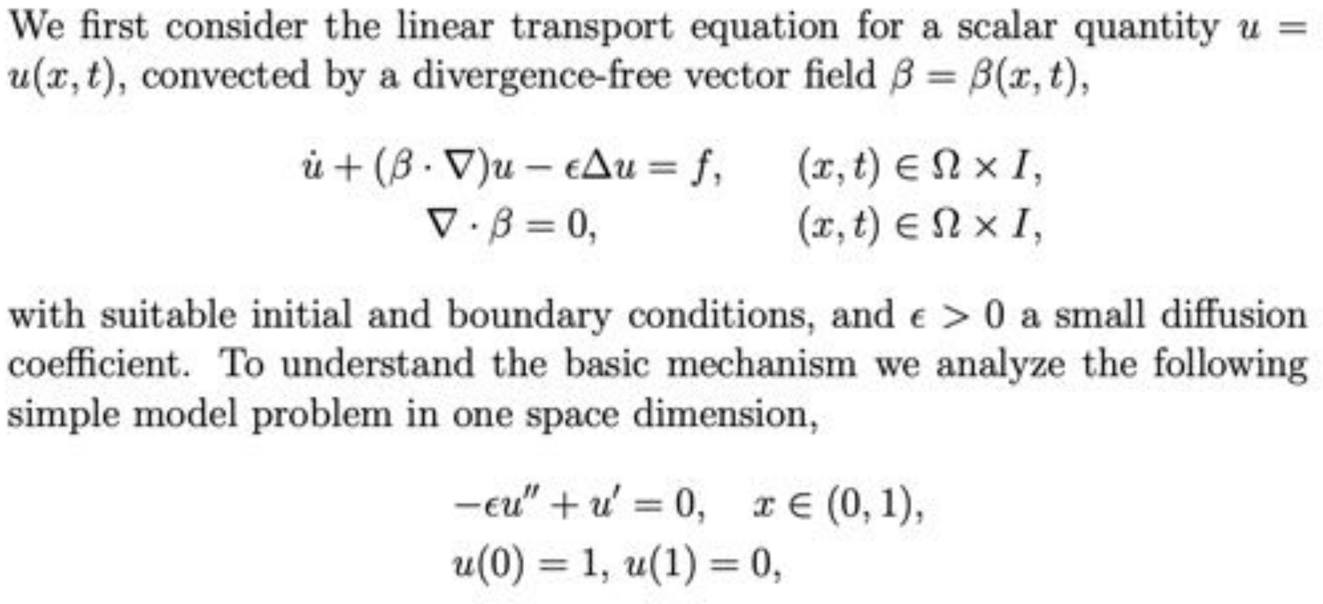
$$\frac{1}{k_n}((U_n, v) - (U_{n-1}, v)) + c(U_\theta; U_\theta, v) + \nu a(U_\theta, v) + b(v, P_\theta) - b(U_\theta, q) = (f, v),$$

for all $(v, q) \in V_h \times Q_h$, with

$$U_\theta = (1-\theta)U_n + \theta U_{n-1}, \quad P_\theta = (1-\theta)P_n + \theta P_{n-1}.$$

Here e.g. $\theta = 0$ corresponds to the Implicit Euler method, and $\theta = 0.5$ corresponds to the Trapezoidal method.

Stabilization: linear transport model problem



We first consider the linear transport equation for a scalar quantity $u = u(x,t)$, convected by a divergence-free vector field $\beta = \beta(x,t)$,

$$\begin{aligned} \dot{u} + (\beta \cdot \nabla)u - \epsilon \Delta u &= f, \qquad (x,t) \in \Omega \times I, \\ \nabla \cdot \beta &= 0, \qquad (x,t) \in \Omega \times I, \end{aligned}$$

with suitable initial and boundary conditions, and $\epsilon > 0$ a small diffusion coefficient. To understand the basic mechanism we analyze the following simple model problem in one space dimension,

$$\begin{aligned} &-\epsilon u'' + u' = 0, \quad x \in (0,1), \\ &u(0) = 1, \; u(1) = 0, \end{aligned}$$

Stabilization: linear transport model problem

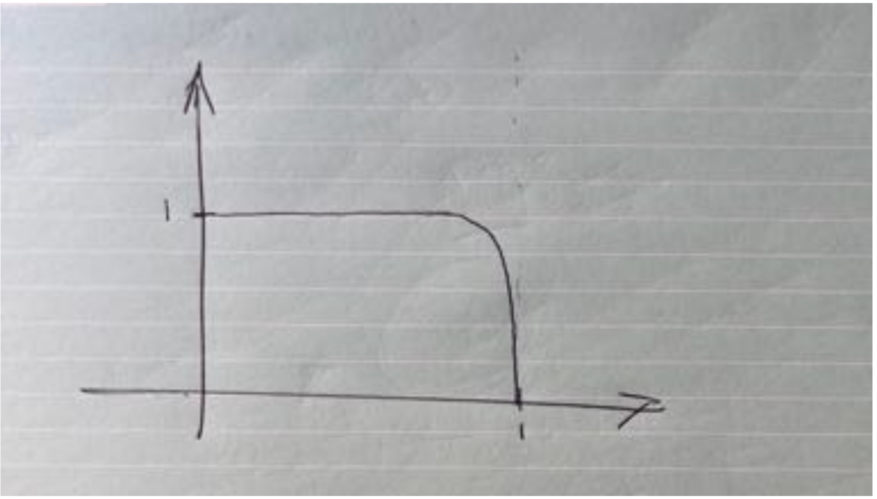
$$-\epsilon u'' + u' = 0, \quad x \in (0,1),$$
$$u(0) = 1,\ u(1) = 0,$$

for which we formulate a standard Galerkin finite element method: find $U \in V_h$ such that,

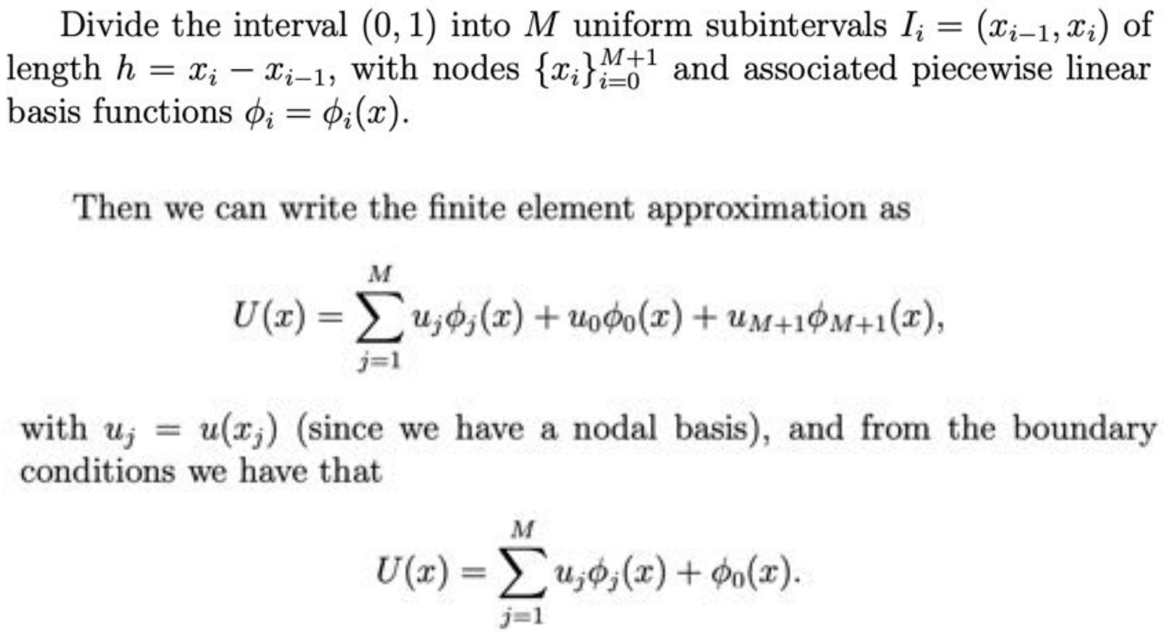
$$\int_0^1 \epsilon u'v' \, dx + \int_0^1 u'v \, dx = 0,$$

for all test functions $v \in V_h^0$, with

$$V_h = \{v \in H^1(0,1) : v(0) = 1, v(1) = 0\},$$
$$V_h^0 = \{v \in H^1(0,1) : v(0) = 0, v(1) = 0\}.$$



Stabilization: linear transport model problem



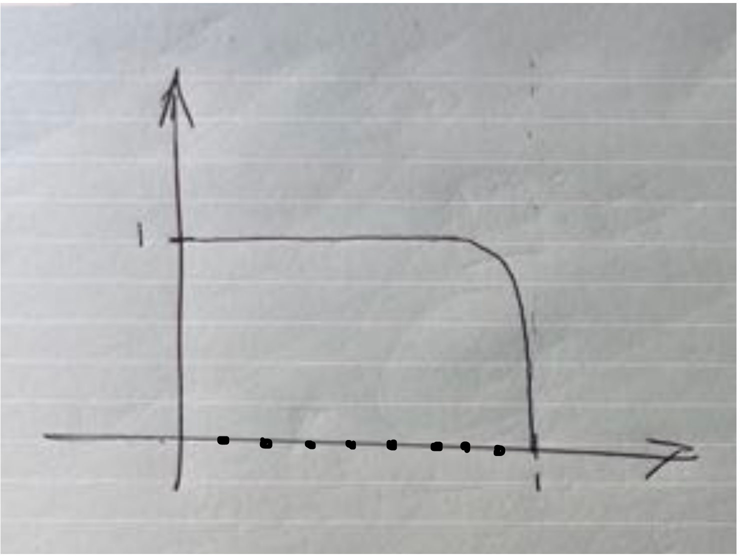
Divide the interval $(0, 1)$ into M uniform subintervals $I_i = (x_{i-1}, x_i)$ of length $h = x_i - x_{i-1}$, with nodes $\{x_i\}_{i=0}^{M+1}$ and associated piecewise linear basis functions $\phi_i = \phi_i(x)$.

Then we can write the finite element approximation as

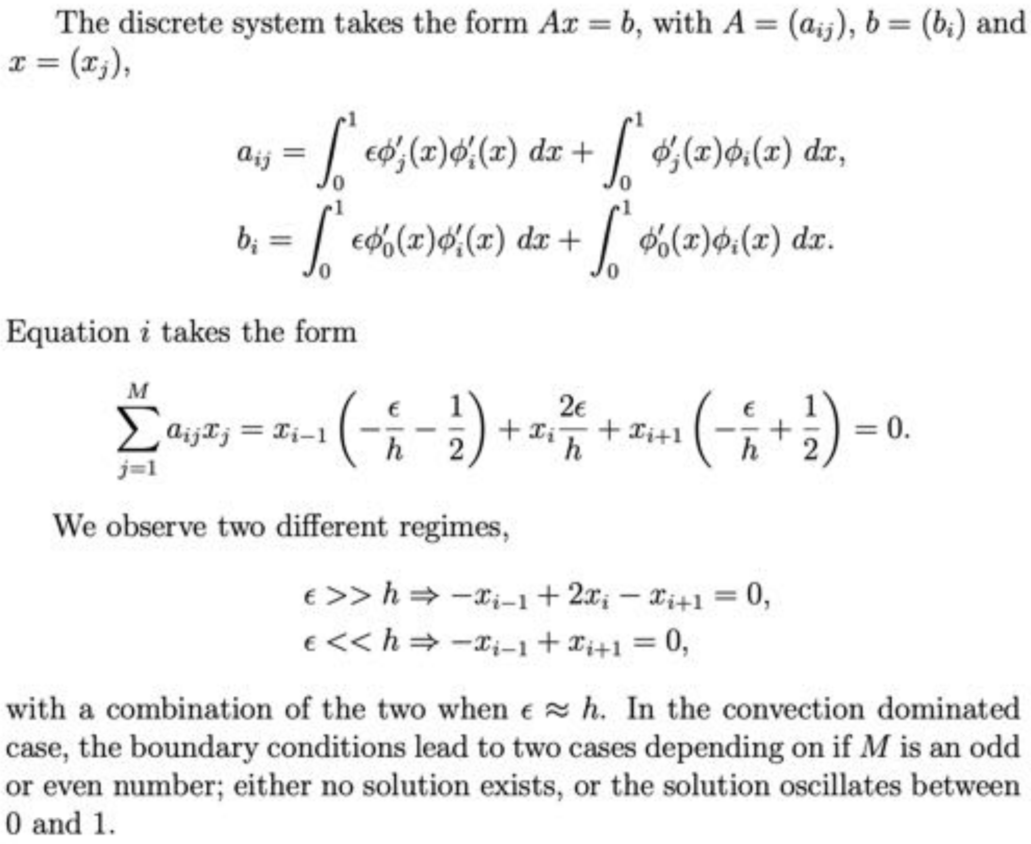
$$U(x) = \sum_{j=1}^{M} u_j \phi_j(x) + u_0 \phi_0(x) + u_{M+1} \phi_{M+1}(x),$$

with $u_j = u(x_j)$ (since we have a nodal basis), and from the boundary conditions we have that

$$U(x) = \sum_{j=1}^{M} u_j \phi_j(x) + \phi_0(x).$$



Stabilization: linear transport model problem



The discrete system takes the form $Ax = b$, with $A = (a_{ij})$, $b = (b_i)$ and $x = (x_j)$,

$$a_{ij} = \int_0^1 \epsilon \phi_j'(x)\phi_i'(x)\ dx + \int_0^1 \phi_j'(x)\phi_i(x)\ dx,$$
$$b_i = \int_0^1 \epsilon \phi_0'(x)\phi_i'(x)\ dx + \int_0^1 \phi_0'(x)\phi_i(x)\ dx.$$

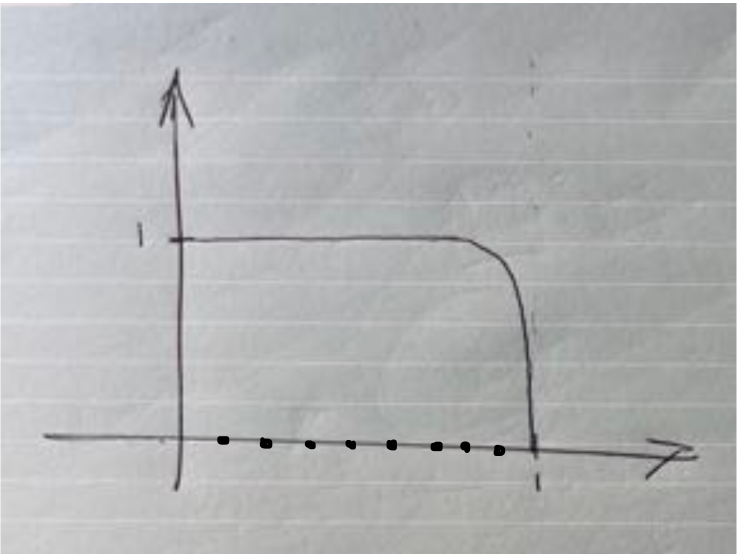
Equation i takes the form

$$\sum_{j=1}^{M} a_{ij}x_j = x_{i-1}\left(-\frac{\epsilon}{h} - \frac{1}{2}\right) + x_i\frac{2\epsilon}{h} + x_{i+1}\left(-\frac{\epsilon}{h} + \frac{1}{2}\right) = 0.$$

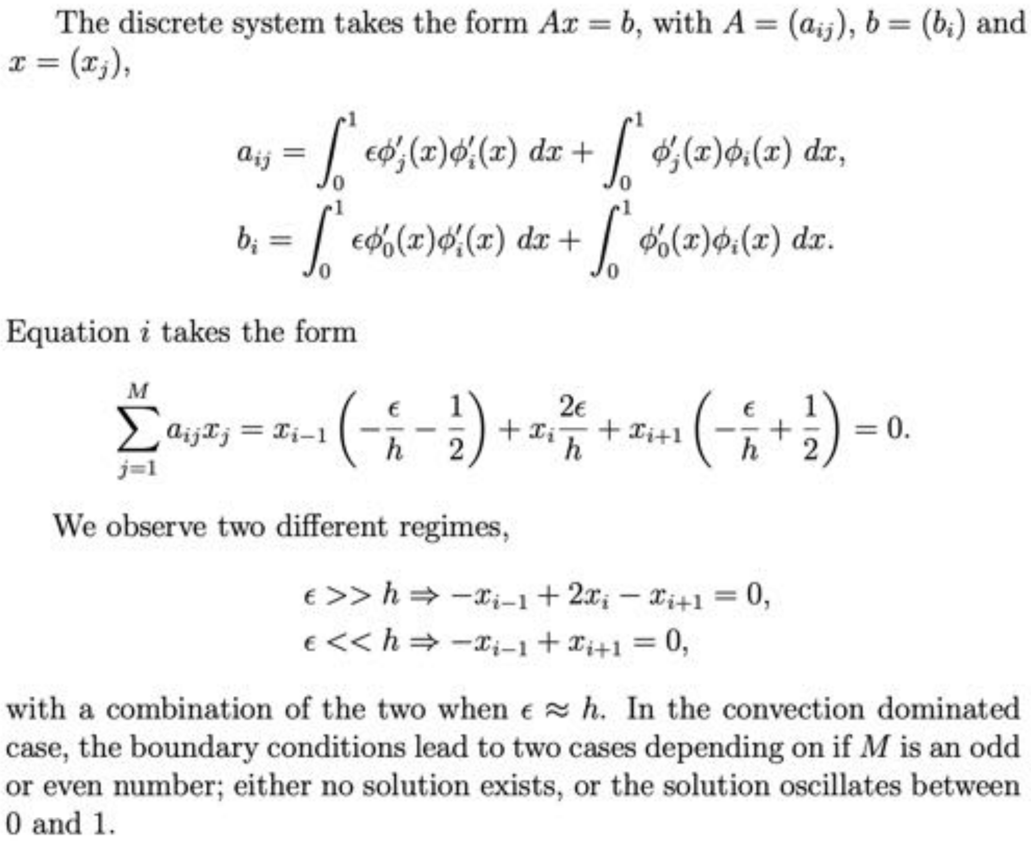
We observe two different regimes,

$$\epsilon >> h \Rightarrow -x_{i-1} + 2x_i - x_{i+1} = 0,$$
$$\epsilon << h \Rightarrow -x_{i-1} + x_{i+1} = 0,$$

with a combination of the two when $\epsilon \approx h$. In the convection dominated case, the boundary conditions lead to two cases depending on if M is an odd or even number; either no solution exists, or the solution oscillates between 0 and 1.



Stabilization: linear transport model problem



The discrete system takes the form $Ax = b$, with $A = (a_{ij})$, $b = (b_i)$ and $x = (x_j)$,

$$a_{ij} = \int_0^1 \epsilon \phi_j'(x)\phi_i'(x)\ dx + \int_0^1 \phi_j'(x)\phi_i(x)\ dx,$$

$$b_i = \int_0^1 \epsilon \phi_0'(x)\phi_i'(x)\ dx + \int_0^1 \phi_0'(x)\phi_i(x)\ dx.$$

Equation i takes the form

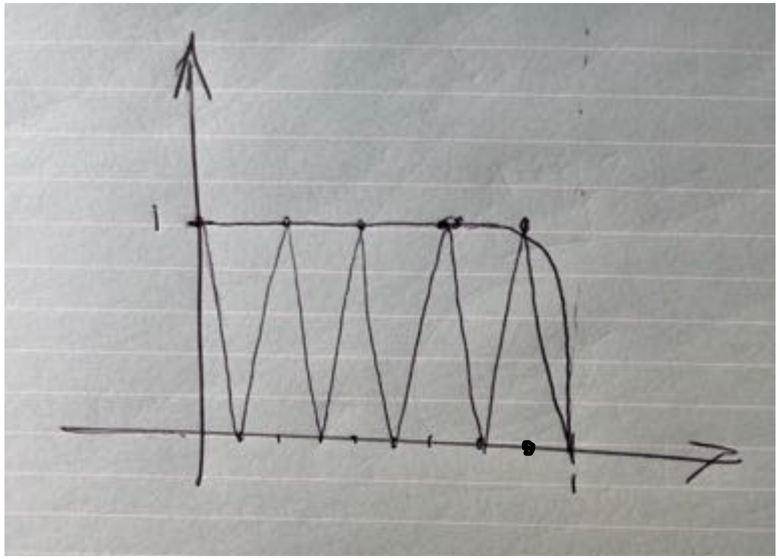
$$\sum_{j=1}^{M} a_{ij}x_j = x_{i-1}\left(-\frac{\epsilon}{h} - \frac{1}{2}\right) + x_i\frac{2\epsilon}{h} + x_{i+1}\left(-\frac{\epsilon}{h} + \frac{1}{2}\right) = 0.$$

We observe two different regimes,

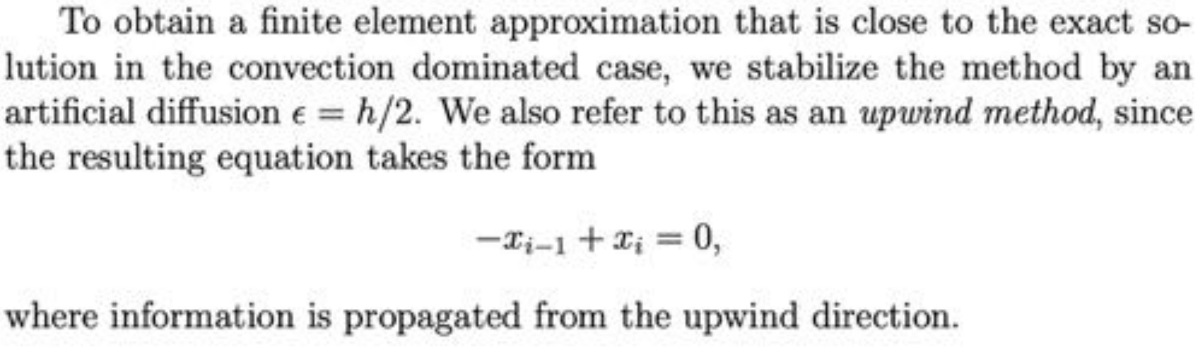
$$\epsilon >> h \Rightarrow -x_{i-1} + 2x_i - x_{i+1} = 0,$$

$$\epsilon << h \Rightarrow -x_{i-1} + x_{i+1} = 0,$$

with a combination of the two when $\epsilon \approx h$. In the convection dominated case, the boundary conditions lead to two cases depending on if M is an odd or even number; either no solution exists, or the solution oscillates between 0 and 1.



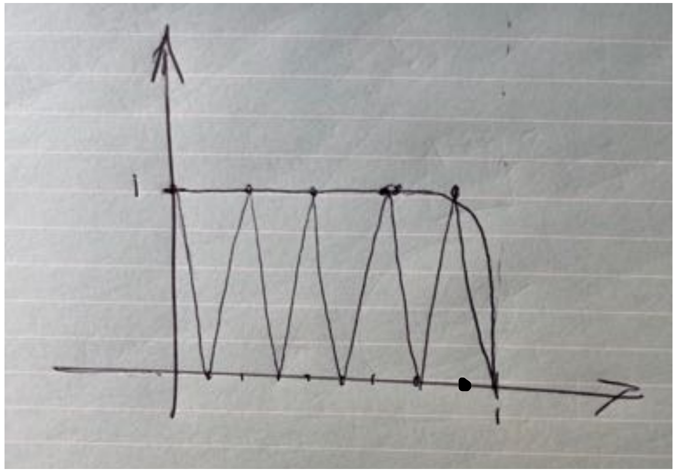
Stabilization: linear transport model problem

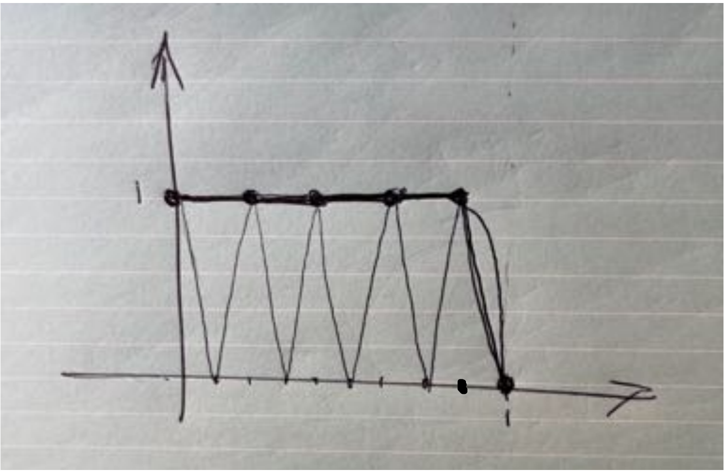


To obtain a finite element approximation that is close to the exact solution in the convection dominated case, we stabilize the method by an artificial diffusion $\epsilon = h/2$. We also refer to this as an *upwind method*, since the resulting equation takes the form

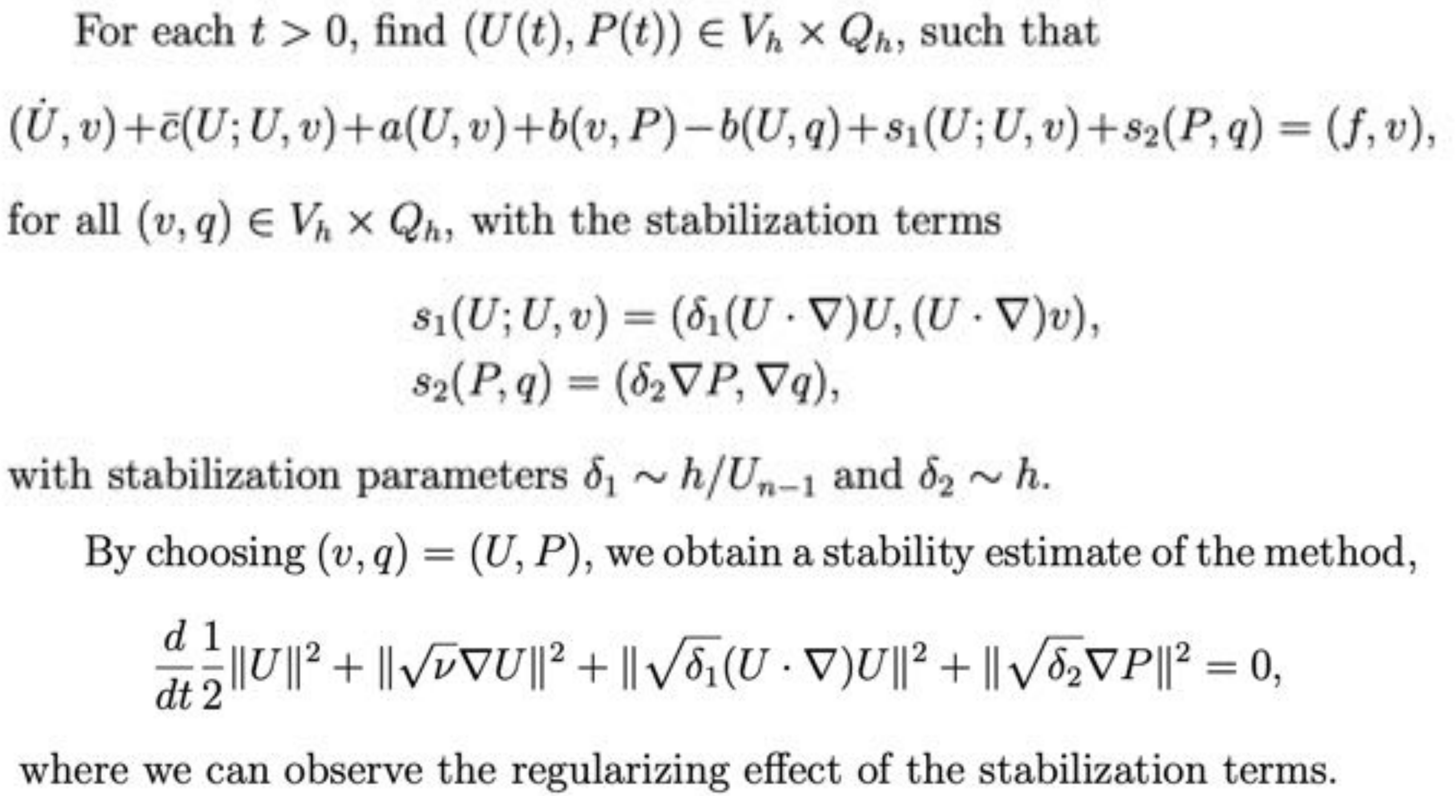
$$-x_{i-1} + x_i = 0,$$

where information is propagated from the upwind direction.





NSE: Streamline diffusion stabilization



For each $t > 0$, find $(U(t), P(t)) \in V_h \times Q_h$, such that

$$(\dot{U}, v) + \bar{c}(U; U, v) + a(U, v) + b(v, P) - b(U, q) + s_1(U; U, v) + s_2(P, q) = (f, v),$$

for all $(v, q) \in V_h \times Q_h$, with the stabilization terms

$$\begin{aligned} s_1(U; U, v) &= (\delta_1 (U \cdot \nabla) U, (U \cdot \nabla) v), \\ s_2(P, q) &= (\delta_2 \nabla P, \nabla q), \end{aligned}$$

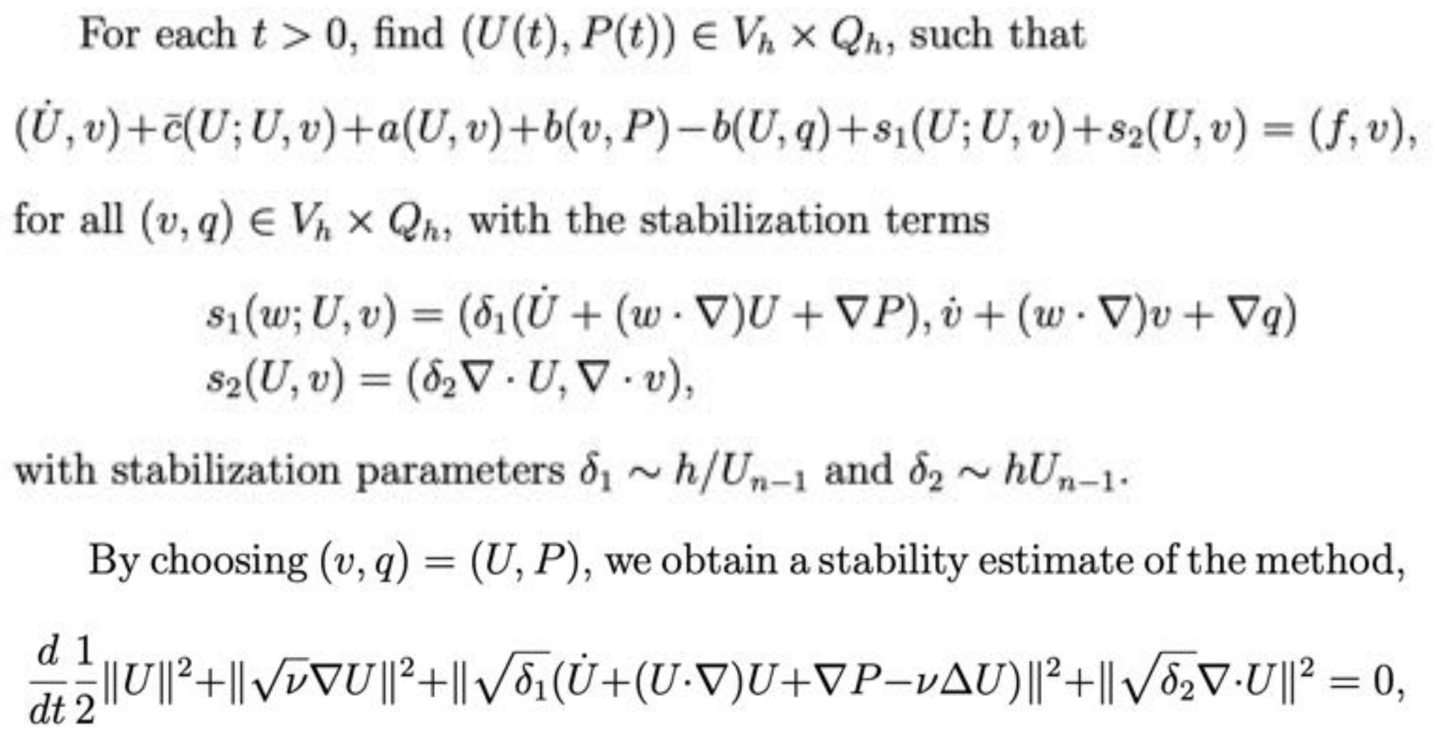
with stabilization parameters $\delta_1 \sim h/U_{n-1}$ and $\delta_2 \sim h$.

By choosing $(v, q) = (U, P)$, we obtain a stability estimate of the method,

$$\frac{d}{dt}\frac{1}{2}\|U\|^2 + \|\sqrt{\nu}\nabla U\|^2 + \|\sqrt{\delta_1}(U \cdot \nabla)U\|^2 + \|\sqrt{\delta_2}\nabla P\|^2 = 0,$$

where we can observe the regularizing effect of the stabilization terms.

NSE: Least squares stabilization



For each $t > 0$, find $(U(t), P(t)) \in V_h \times Q_h$, such that

$$(\dot{U}, v) + \bar{c}(U; U, v) + a(U, v) + b(v, P) - b(U, q) + s_1(U; U, v) + s_2(U, v) = (f, v),$$

for all $(v, q) \in V_h \times Q_h$, with the stabilization terms

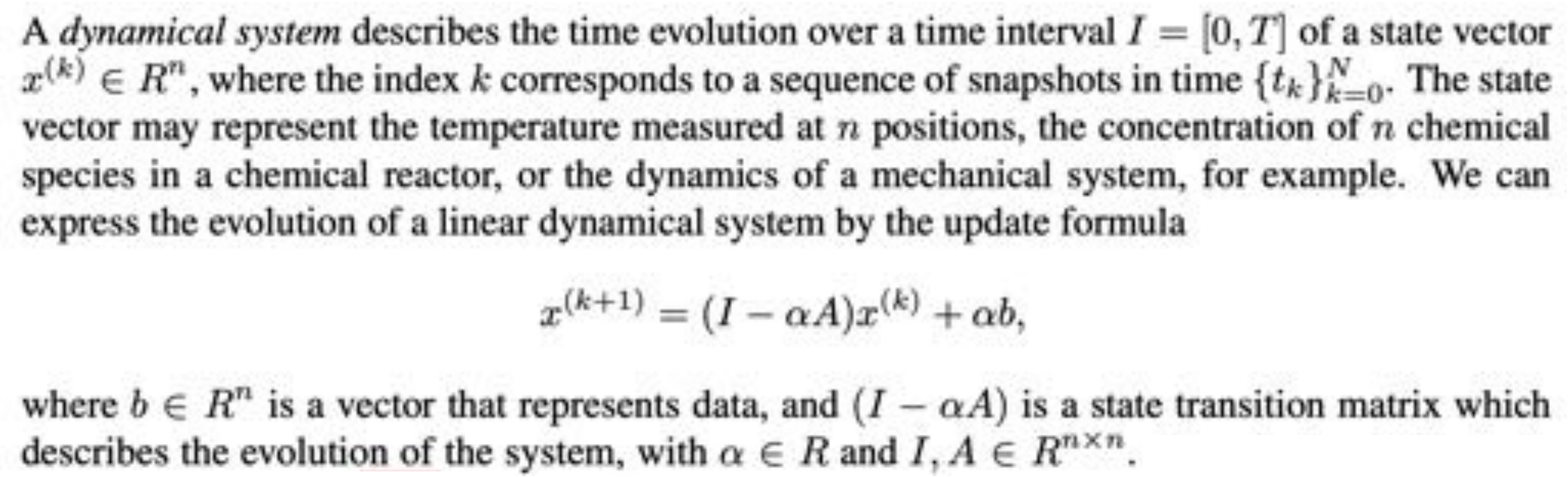
$$\begin{aligned} s_1(w; U, v) &= (\delta_1(\dot{U} + (w \cdot \nabla)U + \nabla P), \dot{v} + (w \cdot \nabla)v + \nabla q) \\ s_2(U, v) &= (\delta_2 \nabla \cdot U, \nabla \cdot v), \end{aligned}$$

with stabilization parameters $\delta_1 \sim h/U_{n-1}$ and $\delta_2 \sim hU_{n-1}$.

By choosing $(v, q) = (U, P)$, we obtain a stability estimate of the method,

$$\frac{d}{dt}\frac{1}{2}\|U\|^2 + \|\sqrt{\nu}\nabla U\|^2 + \|\sqrt{\delta_1}(\dot{U} + (U \cdot \nabla)U + \nabla P - \nu \Delta U)\|^2 + \|\sqrt{\delta_2}\nabla \cdot U\|^2 = 0,$$

Linear dynamical system

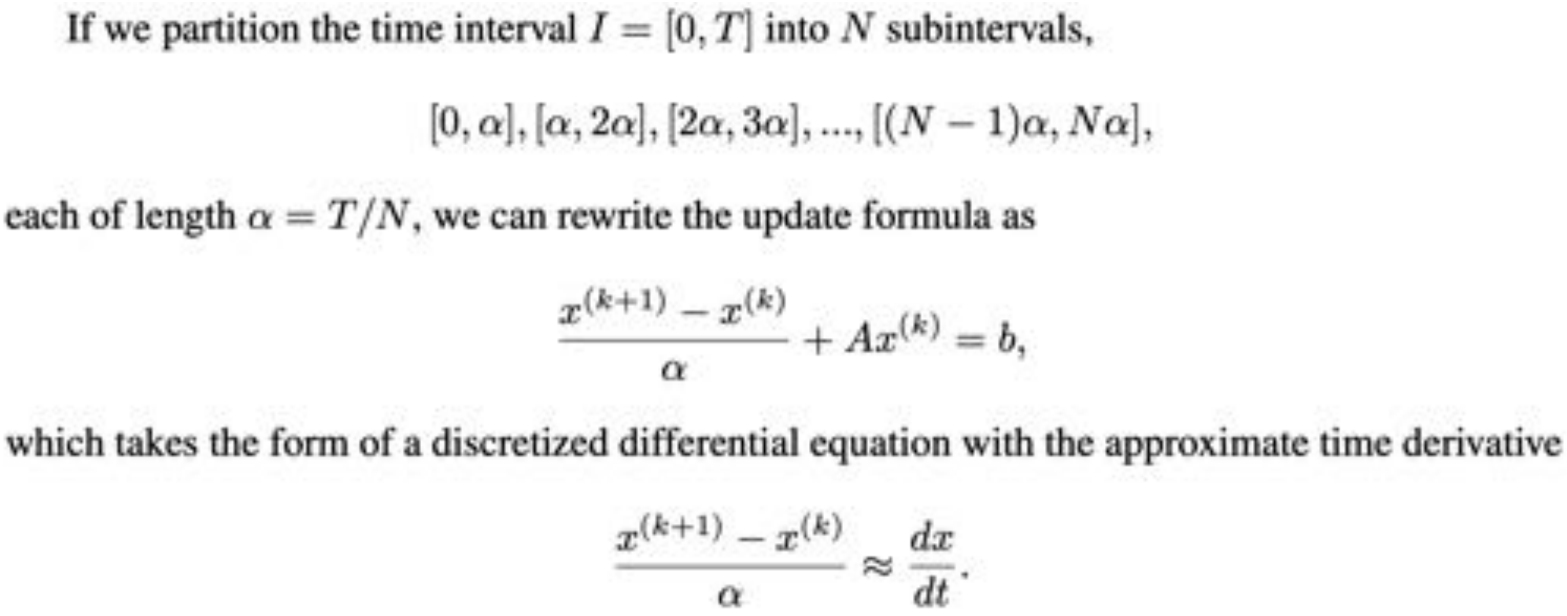


A *dynamical system* describes the time evolution over a time interval $I = [0, T]$ of a state vector $x^{(k)} \in R^n$, where the index k corresponds to a sequence of snapshots in time $\{t_k\}_{k=0}^{N}$. The state vector may represent the temperature measured at n positions, the concentration of n chemical species in a chemical reactor, or the dynamics of a mechanical system, for example. We can express the evolution of a linear dynamical system by the update formula

$$x^{(k+1)} = (I - \alpha A)x^{(k)} + \alpha b,$$

where $b \in R^n$ is a vector that represents data, and $(I - \alpha A)$ is a state transition matrix which describes the evolution of the system, with $\alpha \in R$ and $I, A \in R^{n \times n}$.

Linear dynamical system



If we partition the time interval $I = [0, T]$ into N subintervals,

$$[0, \alpha], [\alpha, 2\alpha], [2\alpha, 3\alpha], ..., [(N-1)\alpha, N\alpha],$$

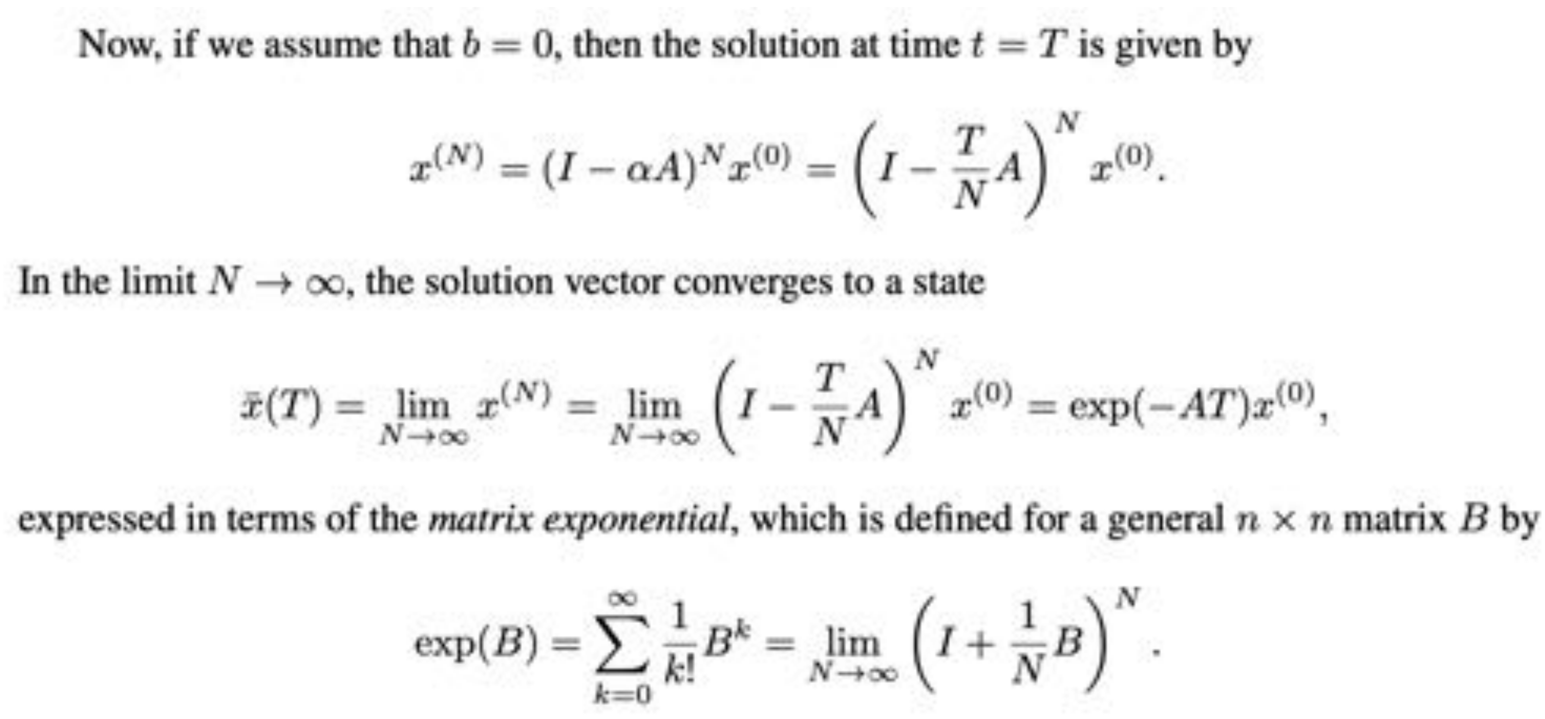
each of length $\alpha = T/N$, we can rewrite the update formula as

$$\frac{x^{(k+1)} - x^{(k)}}{\alpha} + Ax^{(k)} = b,$$

which takes the form of a discretized differential equation with the approximate time derivative

$$\frac{x^{(k+1)} - x^{(k)}}{\alpha} \approx \frac{dx}{dt}.$$

Linear dynamical system



Now, if we assume that $b = 0$, then the solution at time $t = T$ is given by

$$x^{(N)} = (I - \alpha A)^N x^{(0)} = \left(I - \frac{T}{N} A\right)^N x^{(0)}.$$

In the limit $N \to \infty$, the solution vector converges to a state

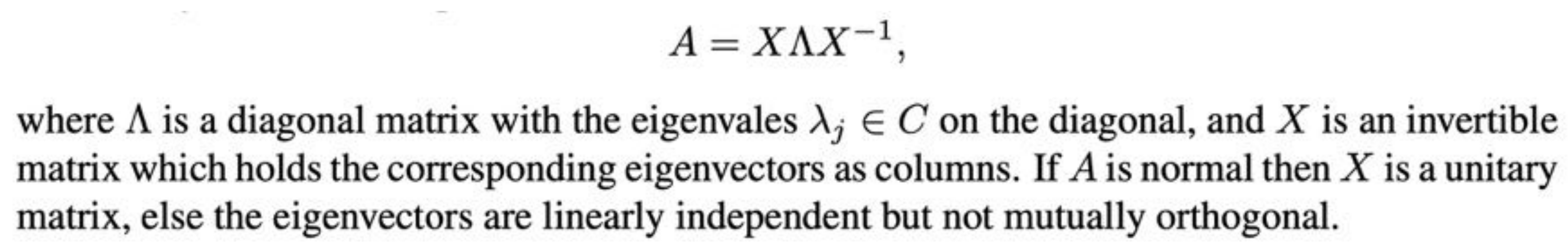
$$\bar{x}(T) = \lim_{N \to \infty} x^{(N)} = \lim_{N \to \infty} \left(I - \frac{T}{N} A\right)^N x^{(0)} = \exp(-AT) x^{(0)},$$

expressed in terms of the *matrix exponential*, which is defined for a general $n \times n$ matrix B by

$$\exp(B) = \sum_{k=0}^{\infty} \frac{1}{k!} B^k = \lim_{N \to \infty} \left(I + \frac{1}{N} B\right)^N.$$

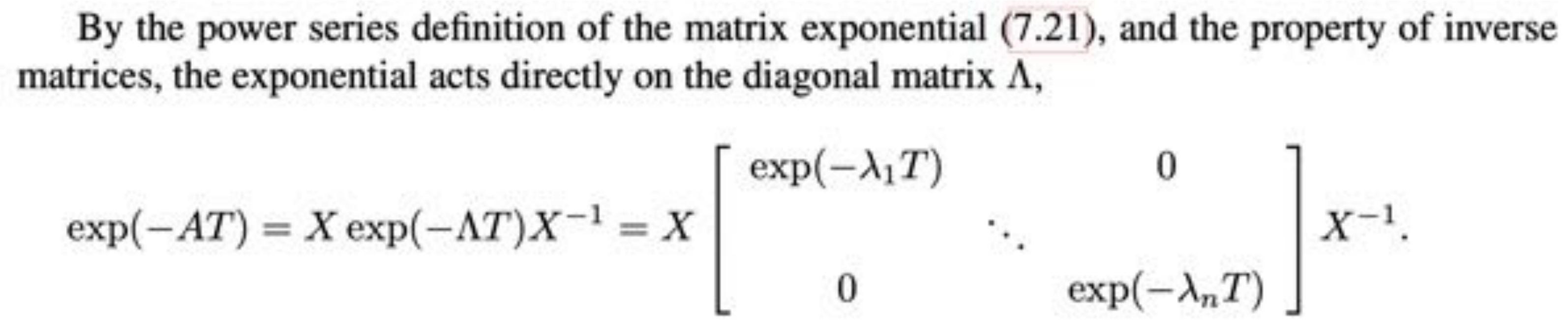
Stability of linear dynamical system

If the matrix A is non-defective, then



$$A = X \Lambda X^{-1},$$

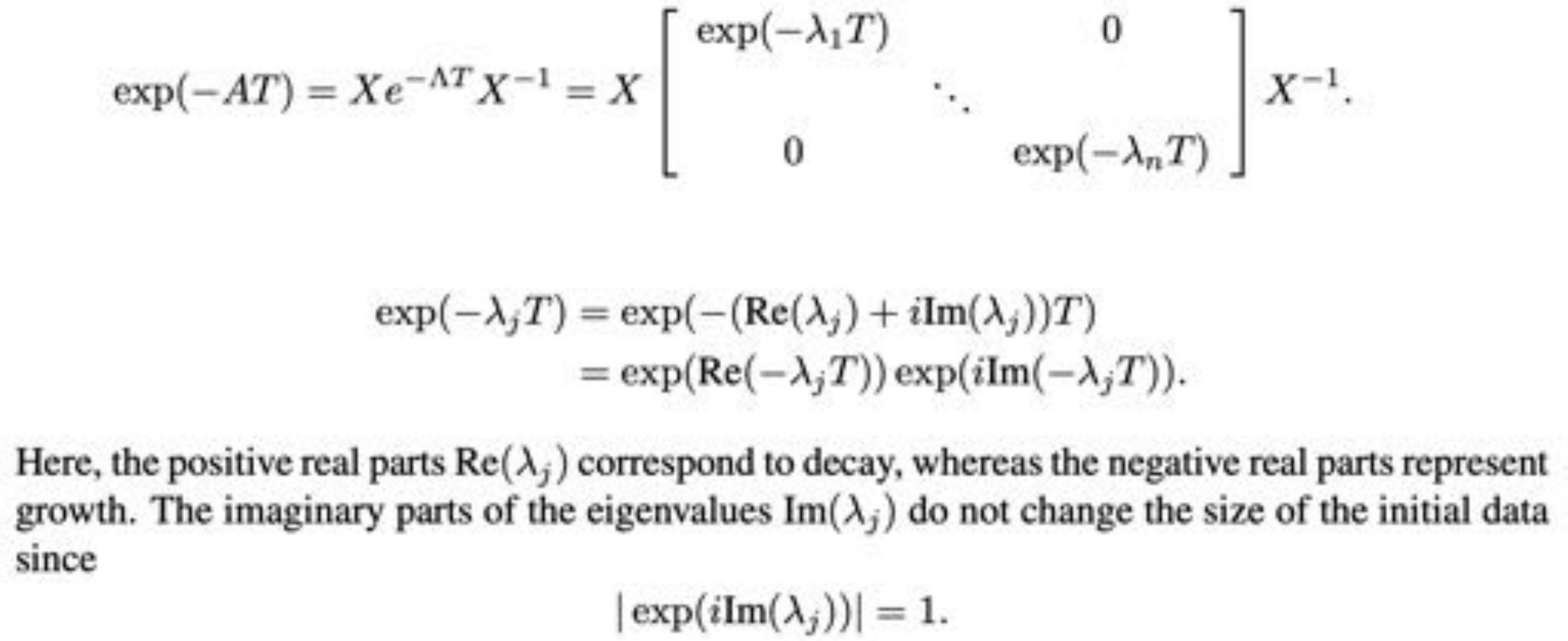
where Λ is a diagonal matrix with the eigenvales $\lambda_j \in C$ on the diagonal, and X is an invertible matrix which holds the corresponding eigenvectors as columns. If A is normal then X is a unitary matrix, else the eigenvectors are linearly independent but not mutually orthogonal.



By the power series definition of the matrix exponential (7.21), and the property of inverse matrices, the exponential acts directly on the diagonal matrix Λ,

$$\exp(-AT) = X \exp(-\Lambda T) X^{-1} = X \begin{bmatrix} \exp(-\lambda_1 T) & & 0 \\ & \ddots & \\ 0 & & \exp(-\lambda_n T) \end{bmatrix} X^{-1}.$$

Stability of linear dynamical system



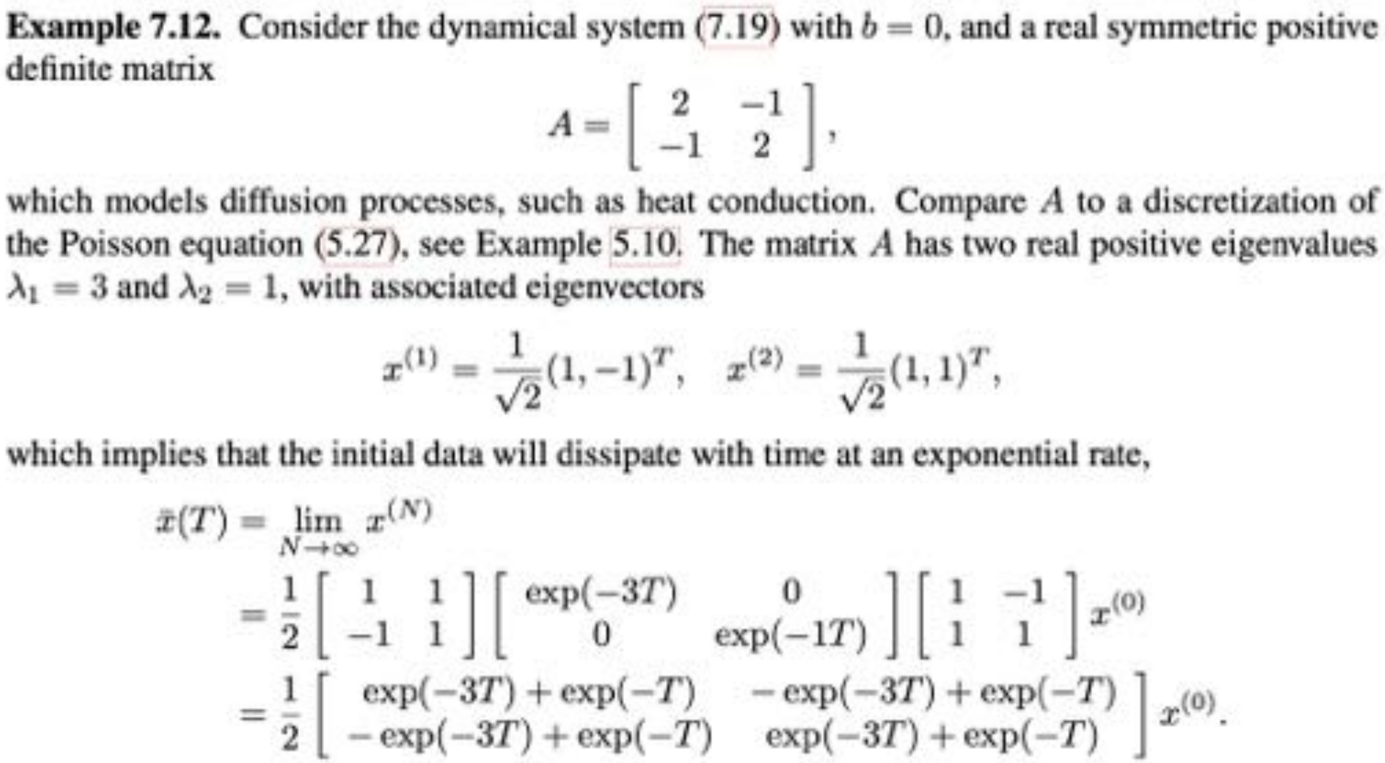
$$\exp(-AT) = Xe^{-\Lambda T}X^{-1} = X \begin{bmatrix} \exp(-\lambda_1 T) & & 0 \\ & \ddots & \\ 0 & & \exp(-\lambda_n T) \end{bmatrix} X^{-1}.$$

$$\begin{aligned} \exp(-\lambda_j T) &= \exp(-(\mathrm{Re}(\lambda_j) + i\mathrm{Im}(\lambda_j))T) \\ &= \exp(\mathrm{Re}(-\lambda_j T)) \exp(i\mathrm{Im}(-\lambda_j T)). \end{aligned}$$

Here, the positive real parts $\mathrm{Re}(\lambda_j)$ correspond to decay, whereas the negative real parts represent growth. The imaginary parts of the eigenvalues $\mathrm{Im}(\lambda_j)$ do not change the size of the initial data since

$$|\exp(i\mathrm{Im}(\lambda_j))| = 1.$$

Symmetric positive definite matrix



Example 7.12. Consider the dynamical system (7.19) with $b = 0$, and a real symmetric positive definite matrix

$$A = \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix},$$

which models diffusion processes, such as heat conduction. Compare A to a discretization of the Poisson equation (5.27), see Example 5.10. The matrix A has two real positive eigenvalues $\lambda_1 = 3$ and $\lambda_2 = 1$, with associated eigenvectors

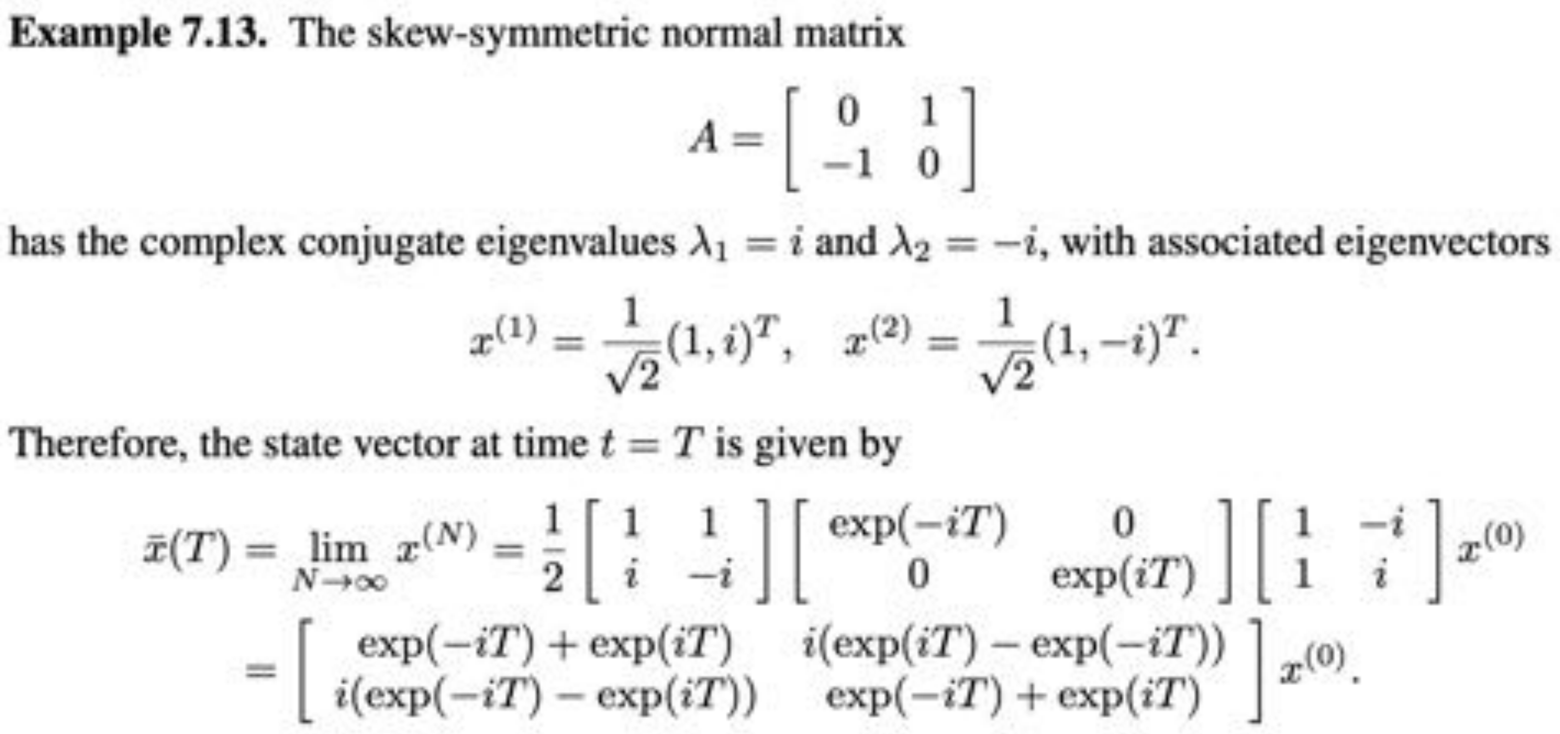
$$x^{(1)} = \frac{1}{\sqrt{2}}(1, -1)^T, \quad x^{(2)} = \frac{1}{\sqrt{2}}(1, 1)^T,$$

which implies that the initial data will dissipate with time at an exponential rate,

$$\begin{aligned} \bar{x}(T) &= \lim_{N \to \infty} x^{(N)} \\ &= \frac{1}{2} \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} \exp(-3T) & 0 \\ 0 & \exp(-1T) \end{bmatrix} \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix} x^{(0)} \\ &= \frac{1}{2} \begin{bmatrix} \exp(-3T) + \exp(-T) & -\exp(-3T) + \exp(-T) \\ -\exp(-3T) + \exp(-T) & \exp(-3T) + \exp(-T) \end{bmatrix} x^{(0)}. \end{aligned}$$

Diffusion and heat conduction

Skew-symmetric normal matrix



Example 7.13. The skew-symmetric normal matrix

$$A = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

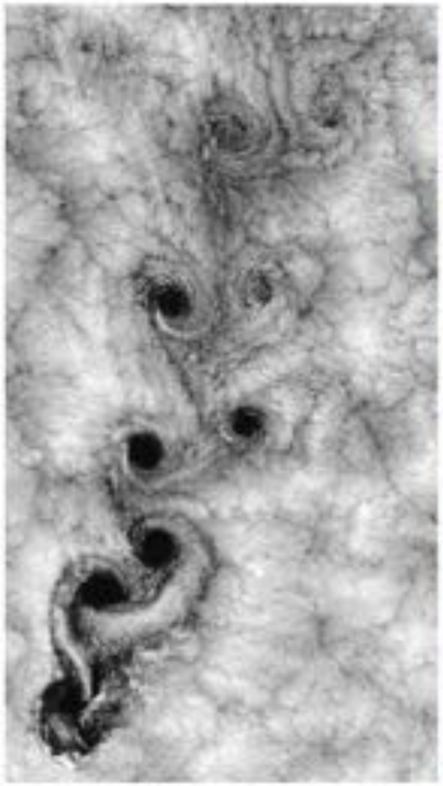
has the complex conjugate eigenvalues $\lambda_1 = i$ and $\lambda_2 = -i$, with associated eigenvectors

$$x^{(1)} = \frac{1}{\sqrt{2}}(1, i)^T, \quad x^{(2)} = \frac{1}{\sqrt{2}}(1, -i)^T.$$

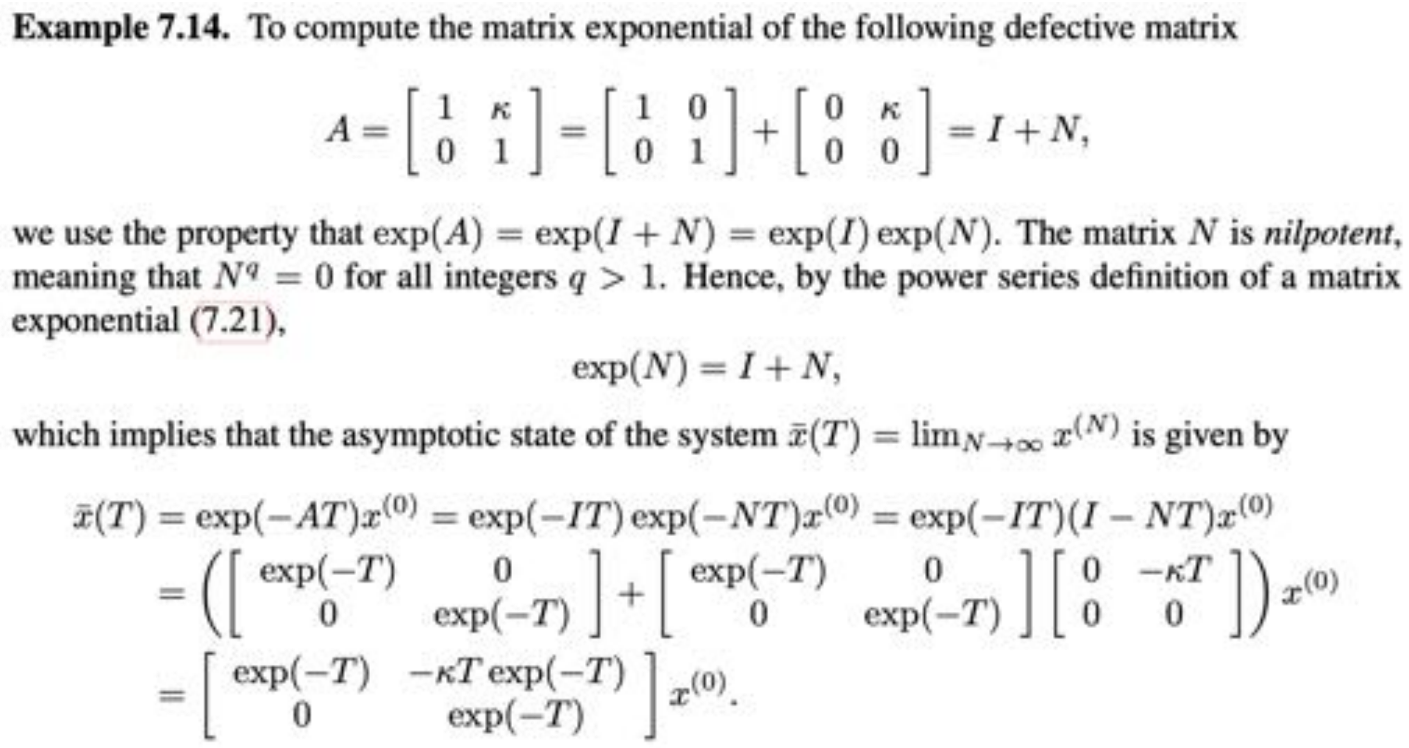
Therefore, the state vector at time $t = T$ is given by

$$\begin{aligned} \bar{x}(T) = \lim_{N\to\infty} x^{(N)} &= \frac{1}{2}\begin{bmatrix} 1 & 1 \\ i & -i \end{bmatrix}\begin{bmatrix} \exp(-iT) & 0 \\ 0 & \exp(iT) \end{bmatrix}\begin{bmatrix} 1 & -i \\ 1 & i \end{bmatrix} x^{(0)} \\ &= \begin{bmatrix} \exp(-iT) + \exp(iT) & i(\exp(iT) - \exp(-iT)) \\ i(\exp(-iT) - \exp(iT)) & \exp(-iT) + \exp(iT) \end{bmatrix} x^{(0)}. \end{aligned}$$

Vortex and wave propagation



Defective matrix



Example 7.14. To compute the matrix exponential of the following defective matrix

$$A = \begin{bmatrix} 1 & \kappa \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \begin{bmatrix} 0 & \kappa \\ 0 & 0 \end{bmatrix} = I + N,$$

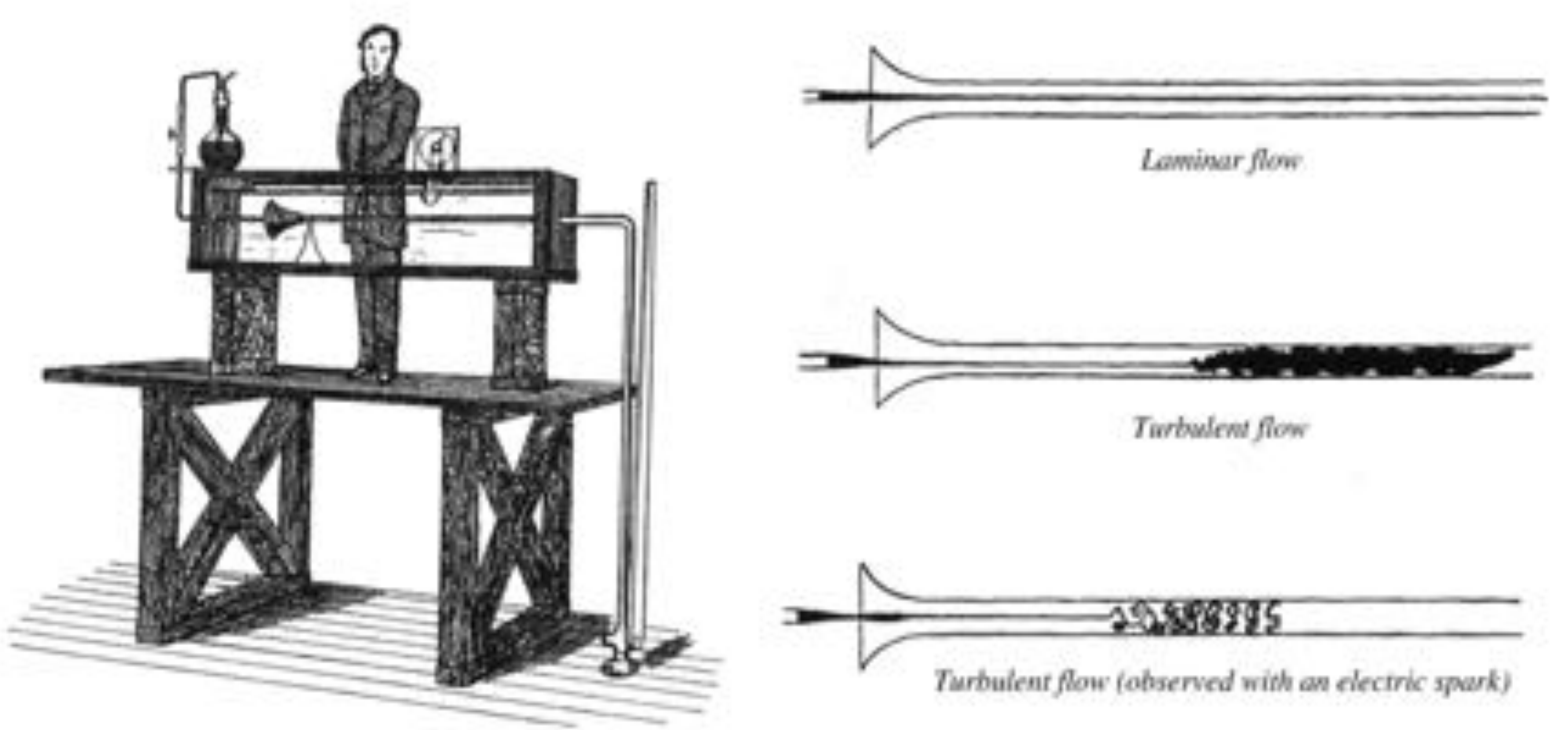
we use the property that $\exp(A) = \exp(I + N) = \exp(I)\exp(N)$. The matrix N is *nilpotent*, meaning that $N^q = 0$ for all integers $q > 1$. Hence, by the power series definition of a matrix exponential (7.21),

$$\exp(N) = I + N,$$

which implies that the asymptotic state of the system $\bar{x}(T) = \lim_{N\to\infty} x^{(N)}$ is given by

$$\begin{aligned} \bar{x}(T) &= \exp(-AT)x^{(0)} = \exp(-IT)\exp(-NT)x^{(0)} = \exp(-IT)(I - NT)x^{(0)} \\ &= \left(\begin{bmatrix} \exp(-T) & 0 \\ 0 & \exp(-T) \end{bmatrix} + \begin{bmatrix} \exp(-T) & 0 \\ 0 & \exp(-T) \end{bmatrix} \begin{bmatrix} 0 & -\kappa T \\ 0 & 0 \end{bmatrix} \right) x^{(0)} \\ &= \begin{bmatrix} \exp(-T) & -\kappa T \exp(-T) \\ 0 & \exp(-T) \end{bmatrix} x^{(0)}. \end{aligned}$$

Transition to turbulence

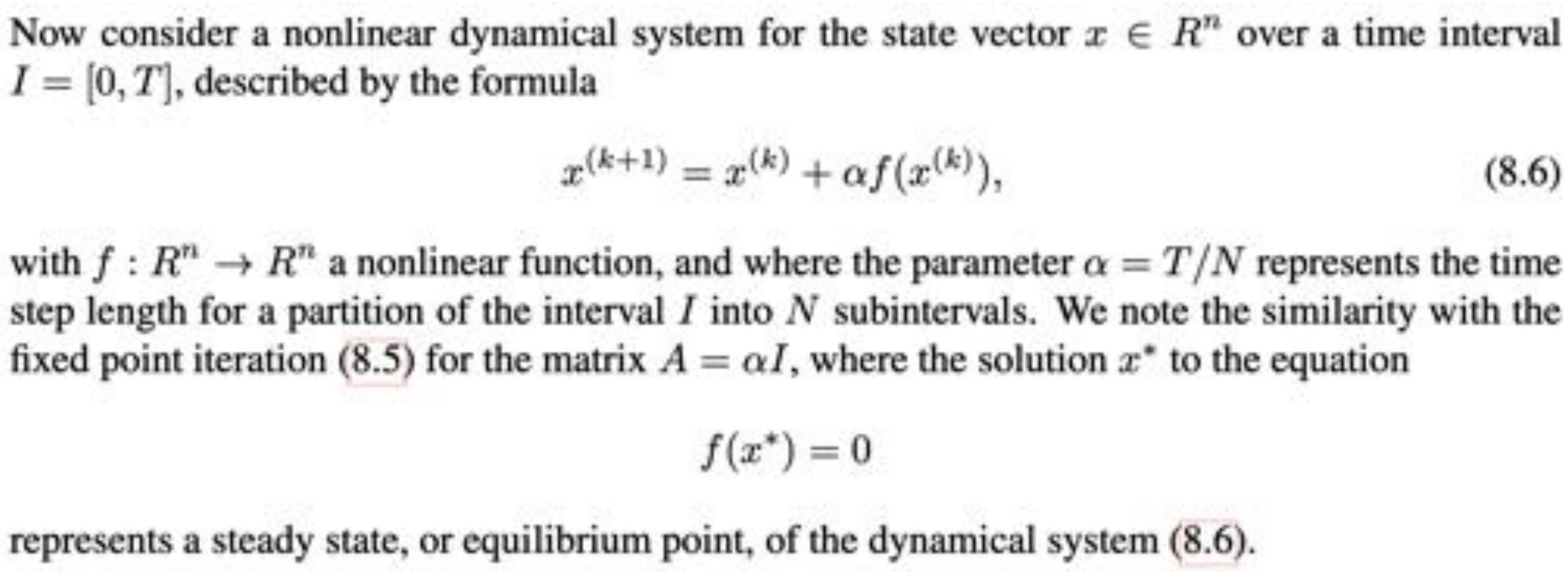


Laminar flow

Turbulent flow

Turbulent flow (observed with an electric spark)

Nonlinear dynamical system



Now consider a nonlinear dynamical system for the state vector $x \in R^n$ over a time interval $I = [0, T]$, described by the formula

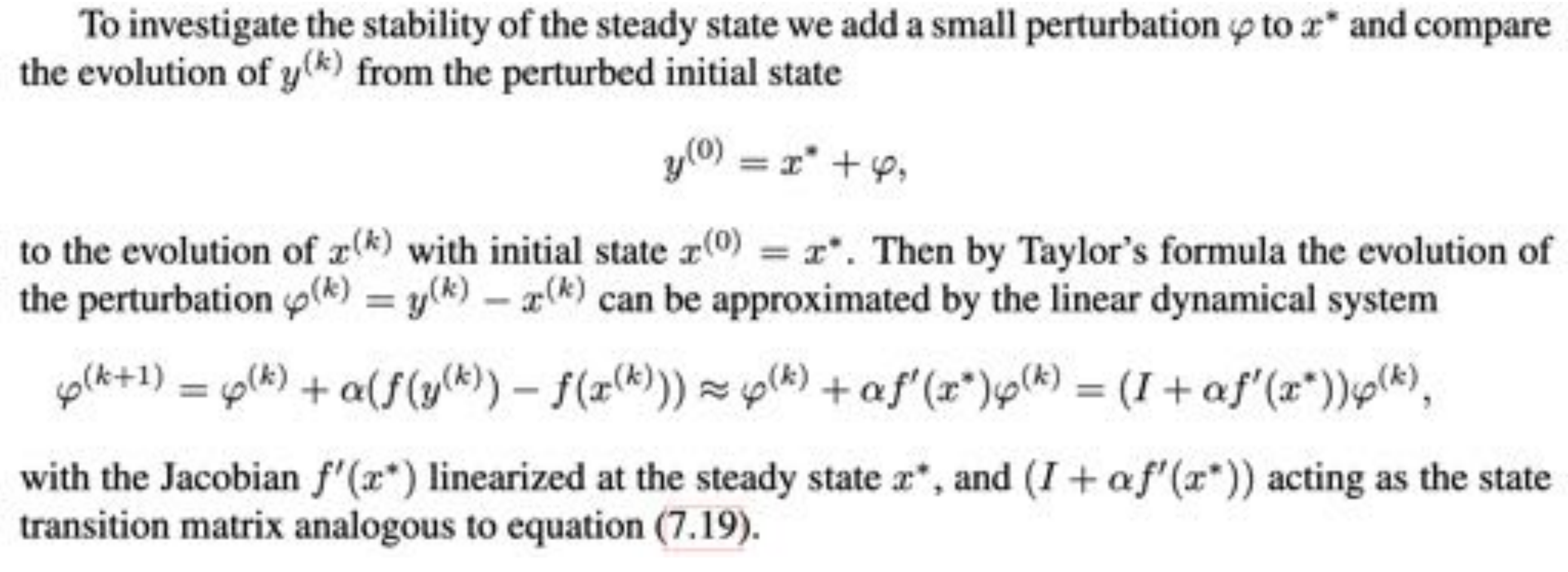
$$x^{(k+1)} = x^{(k)} + \alpha f(x^{(k)}), \tag{8.6}$$

with $f : R^n \to R^n$ a nonlinear function, and where the parameter $\alpha = T/N$ represents the time step length for a partition of the interval I into N subintervals. We note the similarity with the fixed point iteration (8.5) for the matrix $A = \alpha I$, where the solution x^* to the equation

$$f(x^*) = 0$$

represents a steady state, or equilibrium point, of the dynamical system (8.6).

Linear stability analysis



To investigate the stability of the steady state we add a small perturbation φ to x^* and compare the evolution of $y^{(k)}$ from the perturbed initial state

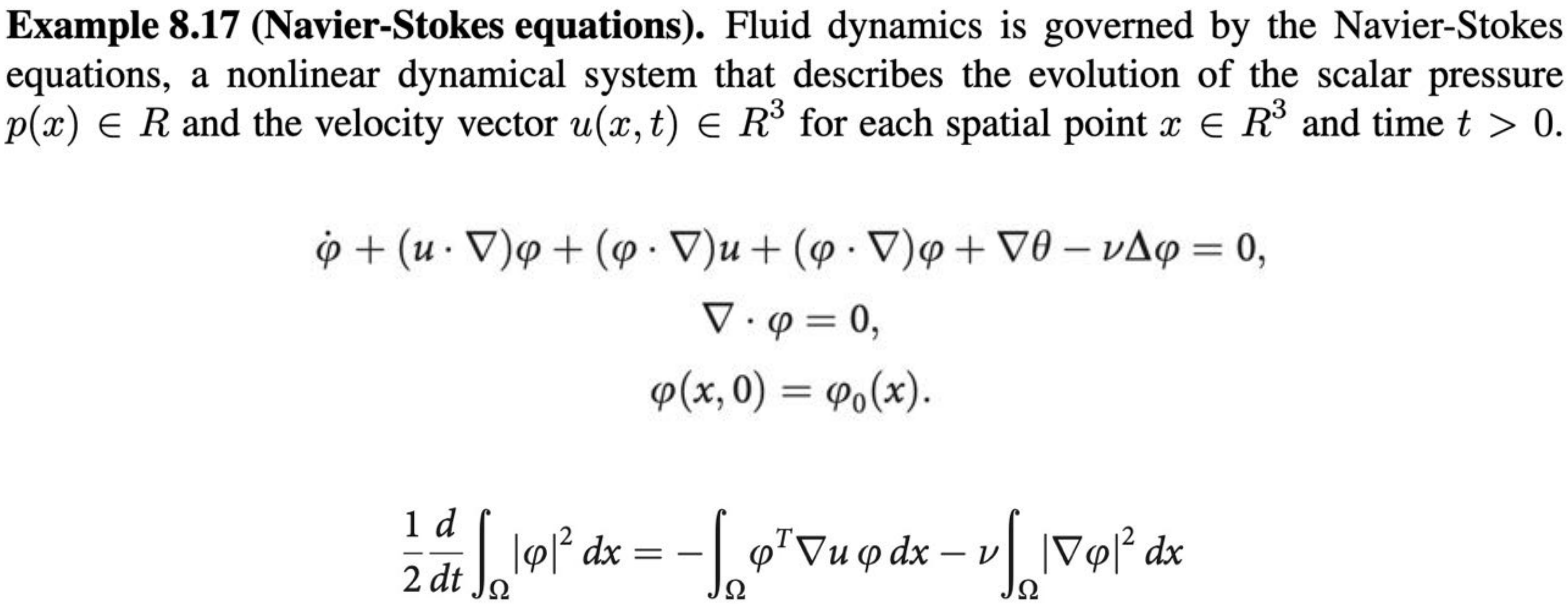
$$y^{(0)} = x^* + \varphi,$$

to the evolution of $x^{(k)}$ with initial state $x^{(0)} = x^*$. Then by Taylor's formula the evolution of the perturbation $\varphi^{(k)} = y^{(k)} - x^{(k)}$ can be approximated by the linear dynamical system

$$\varphi^{(k+1)} = \varphi^{(k)} + \alpha(f(y^{(k)}) - f(x^{(k)})) \approx \varphi^{(k)} + \alpha f'(x^*)\varphi^{(k)} = (I + \alpha f'(x^*))\varphi^{(k)},$$

with the Jacobian $f'(x^*)$ linearized at the steady state x^*, and $(I + \alpha f'(x^*))$ acting as the state transition matrix analogous to equation (7.19).

Navier-Stokes equations



Example 8.17 (Navier-Stokes equations). Fluid dynamics is governed by the Navier-Stokes equations, a nonlinear dynamical system that describes the evolution of the scalar pressure $p(x) \in R$ and the velocity vector $u(x,t) \in R^3$ for each spatial point $x \in R^3$ and time $t > 0$.

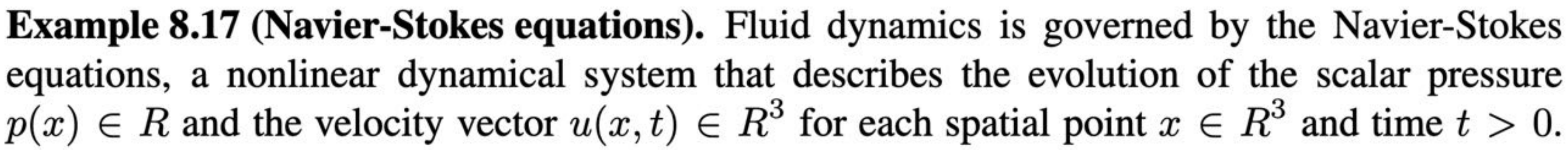
$$\dot{\varphi} + (u \cdot \nabla)\varphi + (\varphi \cdot \nabla)u + (\varphi \cdot \nabla)\varphi + \nabla\theta - \nu\Delta\varphi = 0,$$

$$\nabla \cdot \varphi = 0,$$

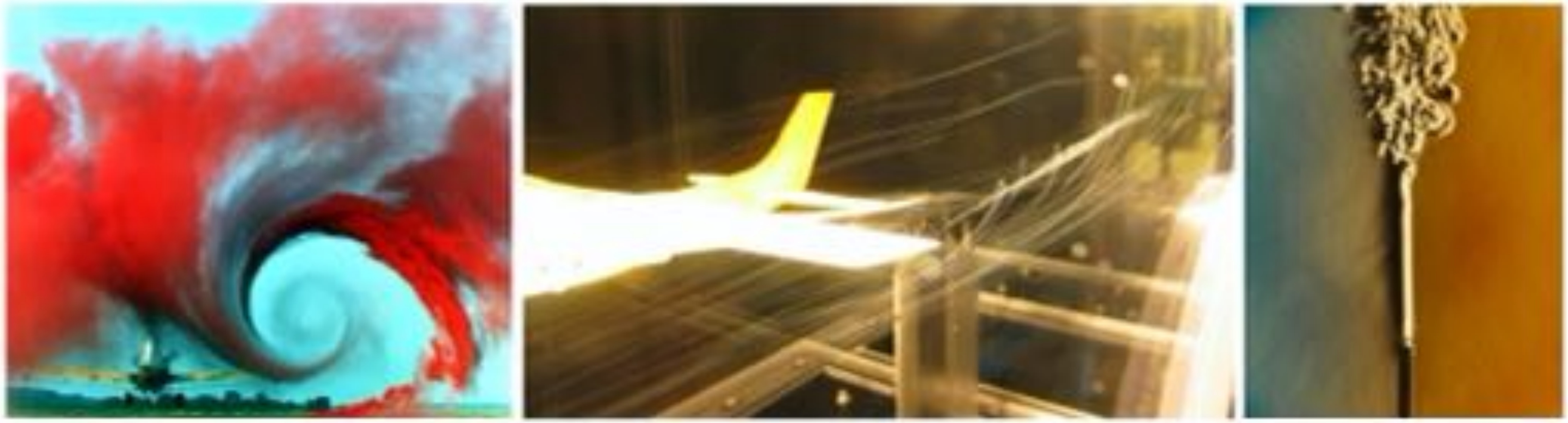
$$\varphi(x,0) = \varphi_0(x).$$

$$\frac{1}{2}\frac{d}{dt}\int_\Omega |\varphi|^2\,dx = -\int_\Omega \varphi^T \nabla u\,\varphi\,dx - \nu\int_\Omega |\nabla\varphi|^2\,dx$$

Navier-Stokes equations

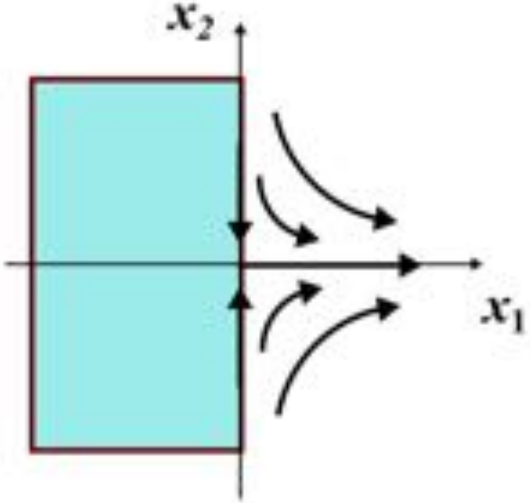


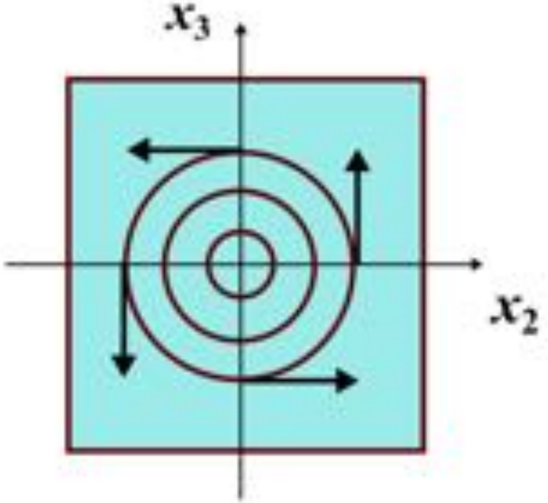
Example 8.17 (Navier-Stokes equations). Fluid dynamics is governed by the Navier-Stokes equations, a nonlinear dynamical system that describes the evolution of the scalar pressure $p(x) \in R$ and the velocity vector $u(x, t) \in R^3$ for each spatial point $x \in R^3$ and time $t > 0$.

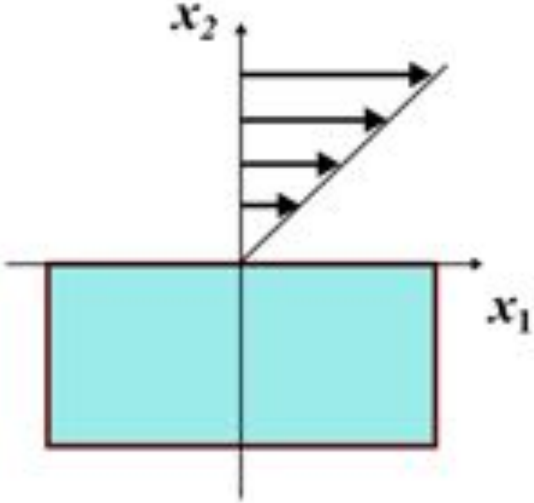


Triple decomposition of velocity gradient

$$\nabla u = Q^T((\nabla u)_{EL} + (\nabla u)_{RR} + (\nabla u)_{SH})Q$$



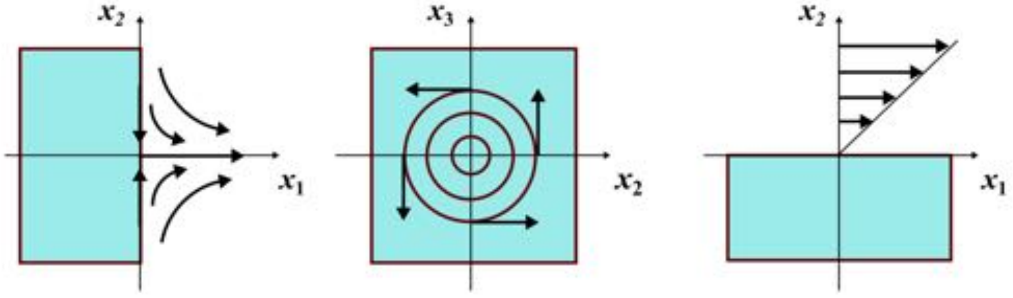




Triple decomposition of velocity gradient

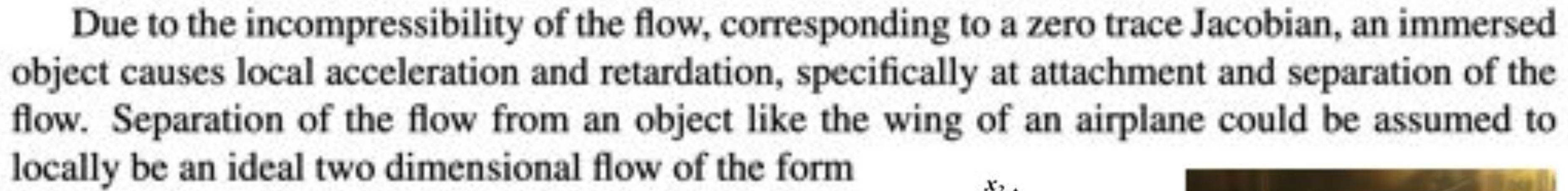
$$\nabla u = Q^T((\nabla u)_{EL} + (\nabla u)_{RR} + (\nabla u)_{SH})Q$$

$$\begin{aligned} u(x) &\approx u(x_0) + \nabla u(x_0)(x - x_0) \\ &= u_C + u_{EL}(x) + u_{RR}(x) + u_{SH}(x) \end{aligned}$$

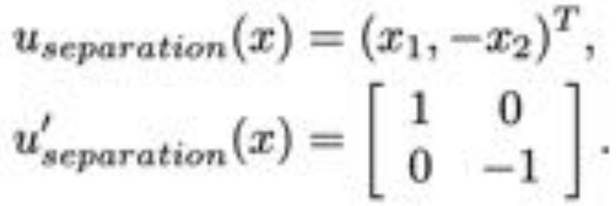


$$\begin{aligned} u_C &= u(x_0), \\ u_{EL}(x) &= (Q^T(\nabla u)_{EL}Q)(x_0)(x - x_0) \\ u_{RR}(x) &= (Q^T(\nabla u)_{RR}Q)(x_0)(x - x_0) \\ u_{SH}(x) &= (Q^T(\nabla u)_{SH}Q)(x_0)(x - x_0) \end{aligned}$$

Linear stability analysis

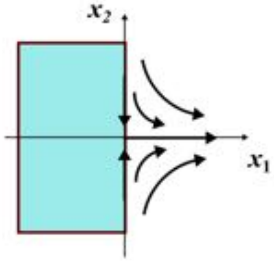


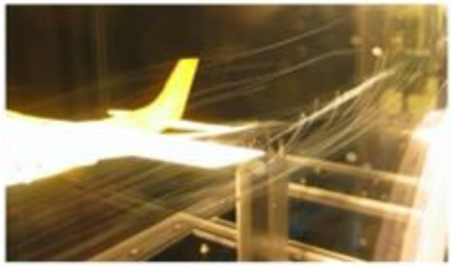
Due to the incompressibility of the flow, corresponding to a zero trace Jacobian, an immersed object causes local acceleration and retardation, specifically at attachment and separation of the flow. Separation of the flow from an object like the wing of an airplane could be assumed to locally be an ideal two dimensional flow of the form

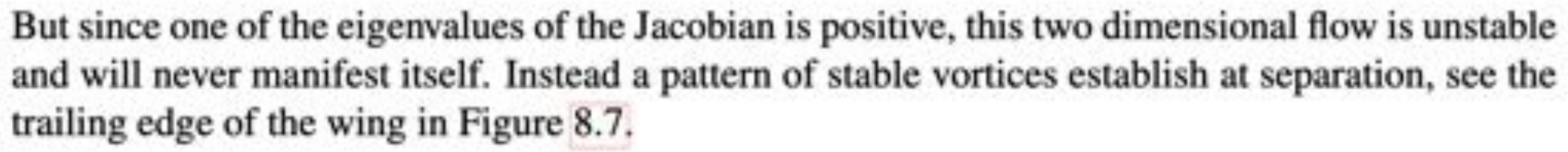


$$u_{separation}(x) = (x_1, -x_2)^T,$$

$$u'_{separation}(x) = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}.$$

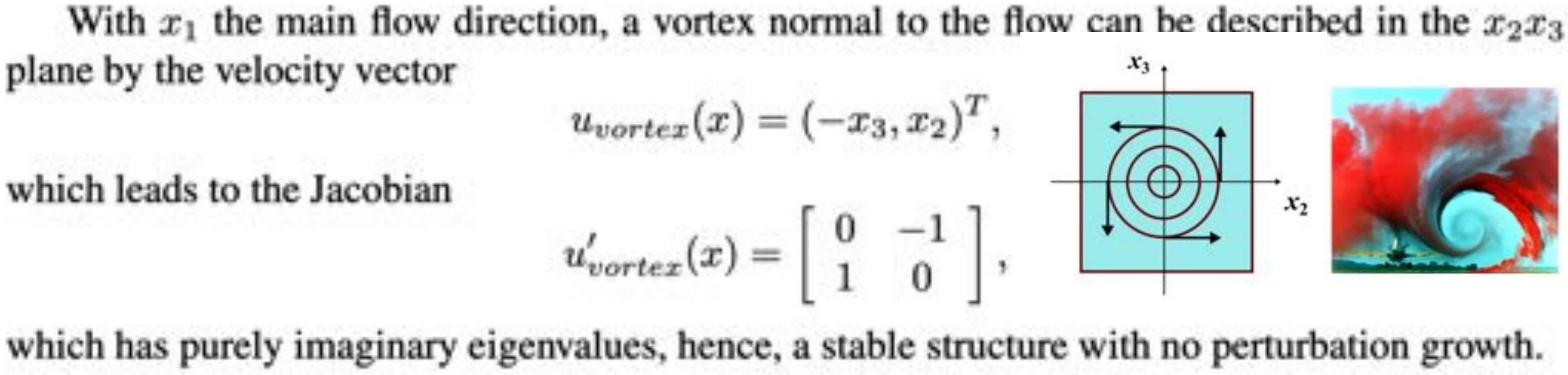






But since one of the eigenvalues of the Jacobian is positive, this two dimensional flow is unstable and will never manifest itself. Instead a pattern of stable vortices establish at separation, see the trailing edge of the wing in Figure 8.7.

Linear stability analysis



With x_1 the main flow direction, a vortex normal to the flow can be described in the x_2x_3 plane by the velocity vector

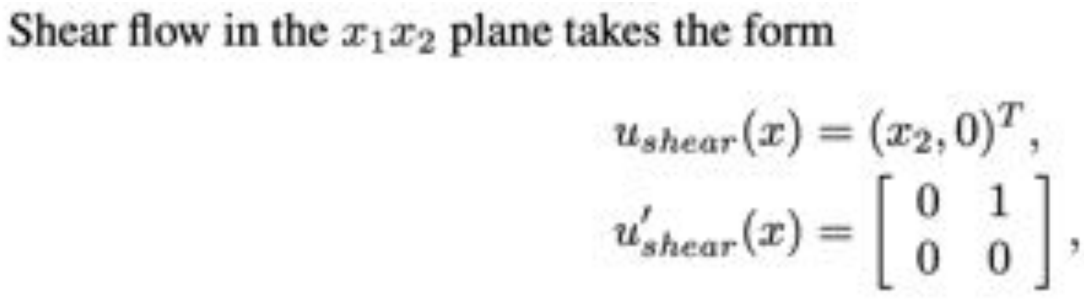
$$u_{vortex}(x) = (-x_3, x_2)^T,$$

which leads to the Jacobian

$$u'_{vortex}(x) = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix},$$

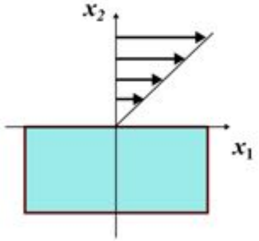
which has purely imaginary eigenvalues, hence, a stable structure with no perturbation growth.

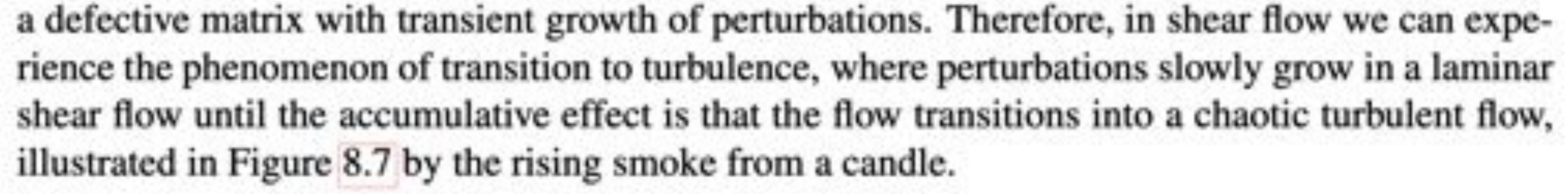
Linear stability analysis



Shear flow in the x_1x_2 plane takes the form

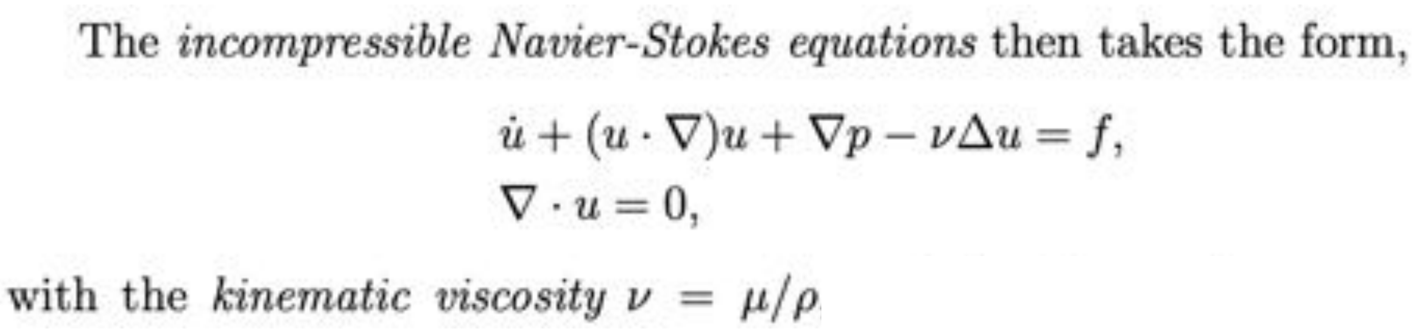
$$u_{shear}(x) = (x_2, 0)^T,$$
$$u'_{shear}(x) = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix},$$





a defective matrix with transient growth of perturbations. Therefore, in shear flow we can experience the phenomenon of transition to turbulence, where perturbations slowly grow in a laminar shear flow until the accumulative effect is that the flow transitions into a chaotic turbulent flow, illustrated in Figure 8.7 by the rising smoke from a candle.

Navier-Stokes equations



The *incompressible Navier-Stokes equations* then takes the form,

$$\begin{aligned}&\dot{u} + (u \cdot \nabla)u + \nabla p - \nu \Delta u = f,\\&\nabla \cdot u = 0,\end{aligned}$$

with the *kinematic viscosity* $\nu = \mu/\rho$

No slip boundary condition: $u = 0$

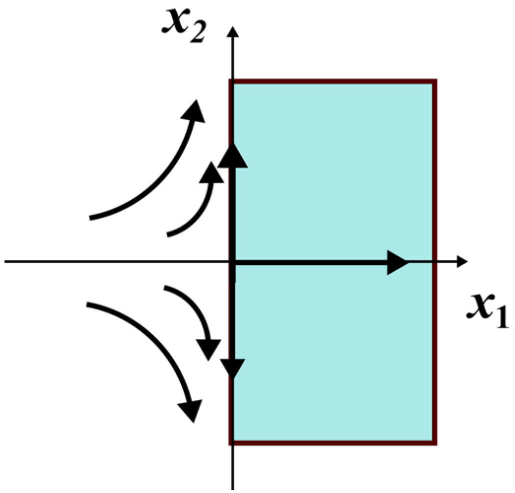
Slip boundary conditions: $u \cdot n = 0$

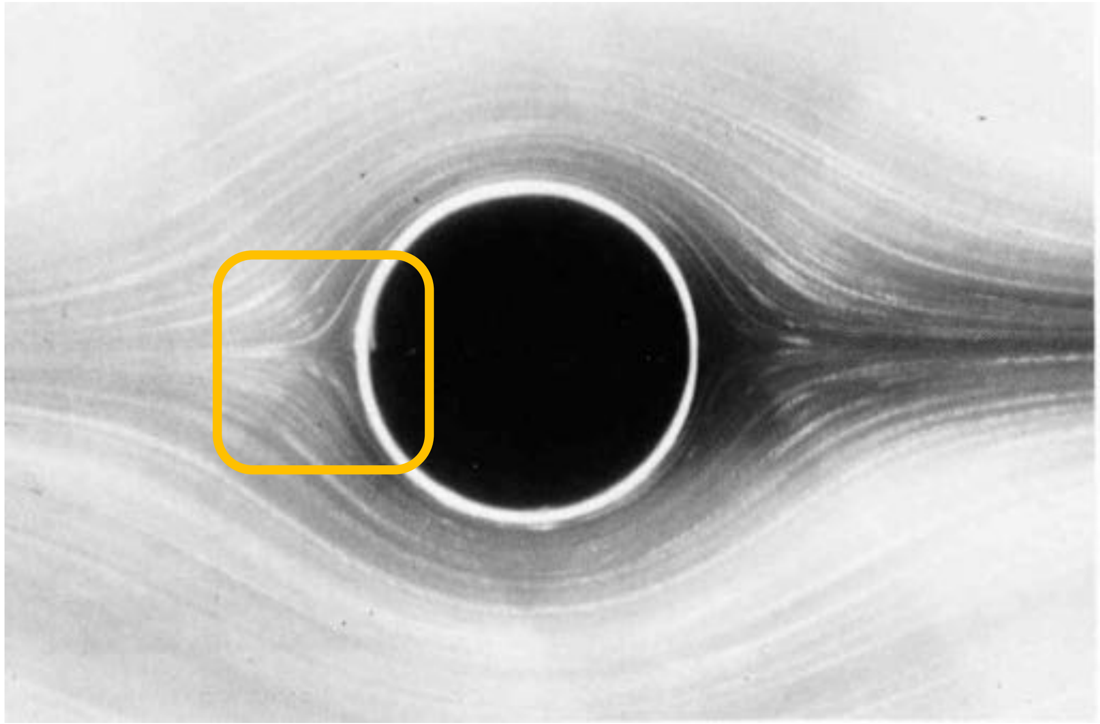
Friction boundary conditions: $n^T \sigma t_i = \beta u \cdot t_i$

Outflow boundary conditions: $n^T \sigma = 0$

Incompressible flow – attachment point

- $\nabla \cdot u = 0$
- $\dfrac{\partial u_2}{\partial x_2} = -\dfrac{\partial u_1}{\partial x_1}$

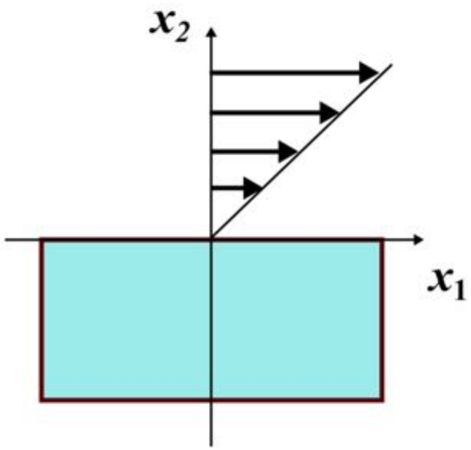


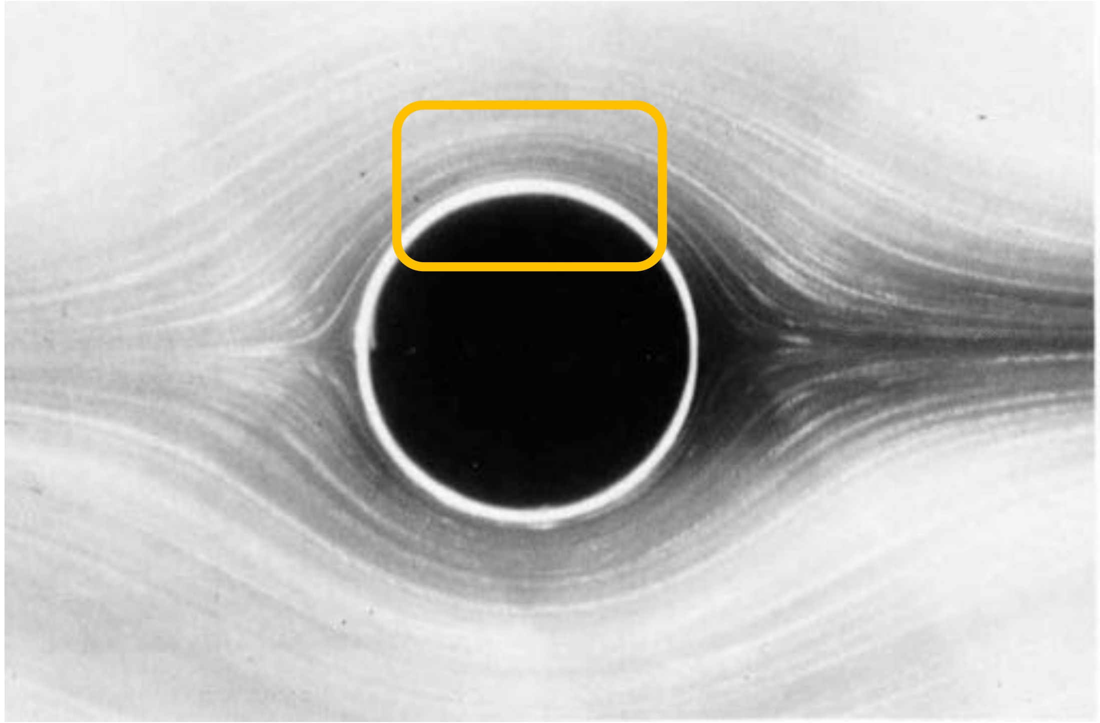


[Water and aluminum dust.]

Incompressible flow – boundary layer

- $\nabla \cdot u = 0$
- $u_1 = f(x_2)$
- $u_2 = 0$

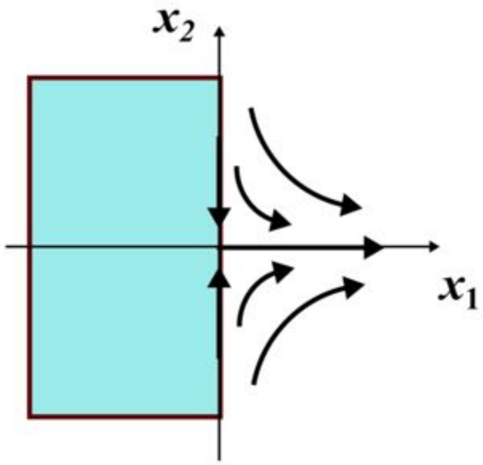


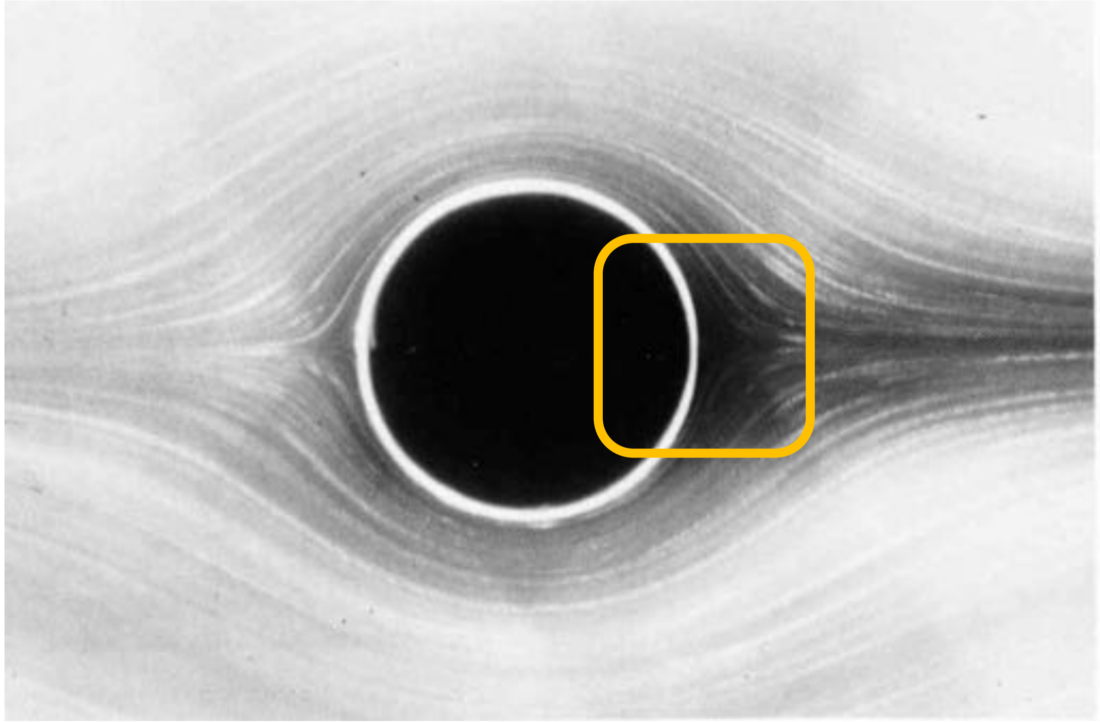


[Water and aluminum dust.]

Cylinder (Re = 0.16) – separation point

- $\nabla \cdot u = 0$
- $\dfrac{\partial u_1}{\partial x_1} = -\dfrac{\partial u_2}{\partial x_2}$

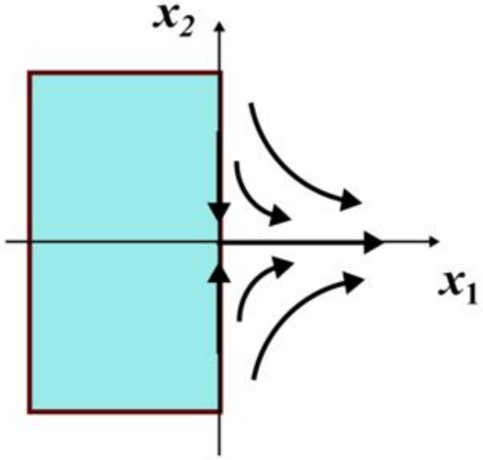


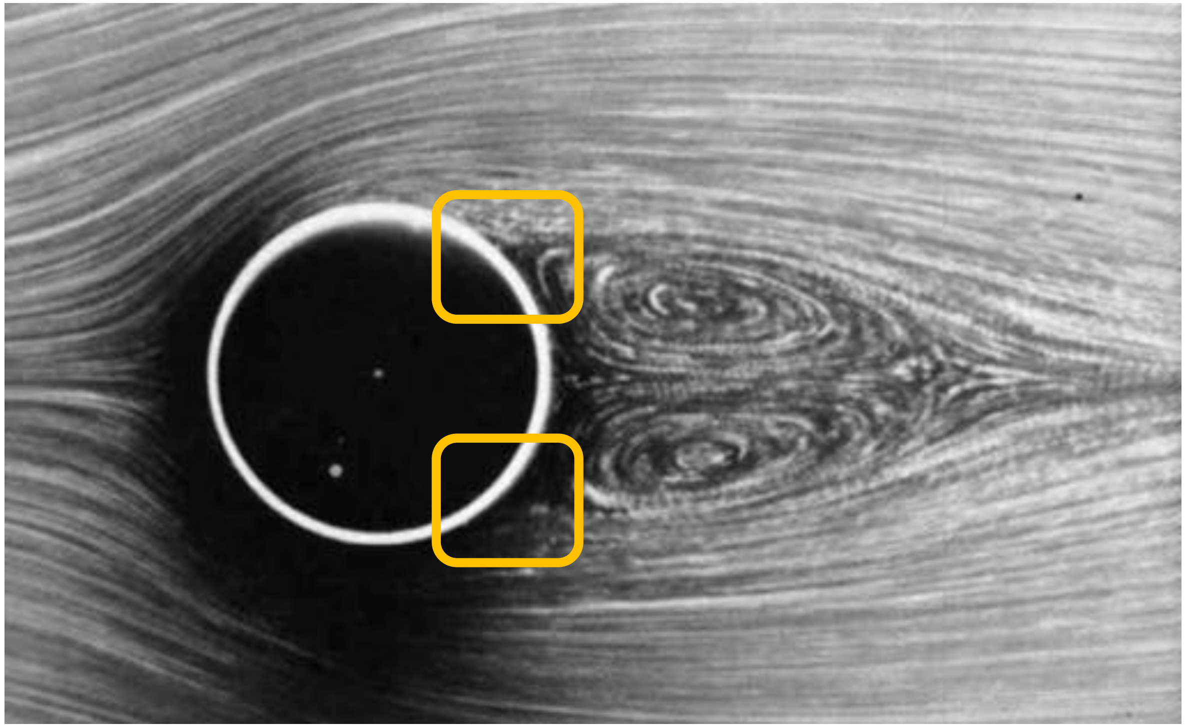


[Water and aluminum dust.]

Cylinder (Re = 26) – 2 separation points

- $\nabla \cdot u = 0$
- $\frac{\partial u_1}{\partial x_1} = -\frac{\partial u_2}{\partial x_2}$

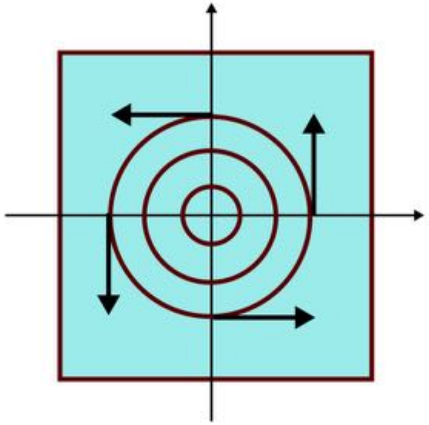


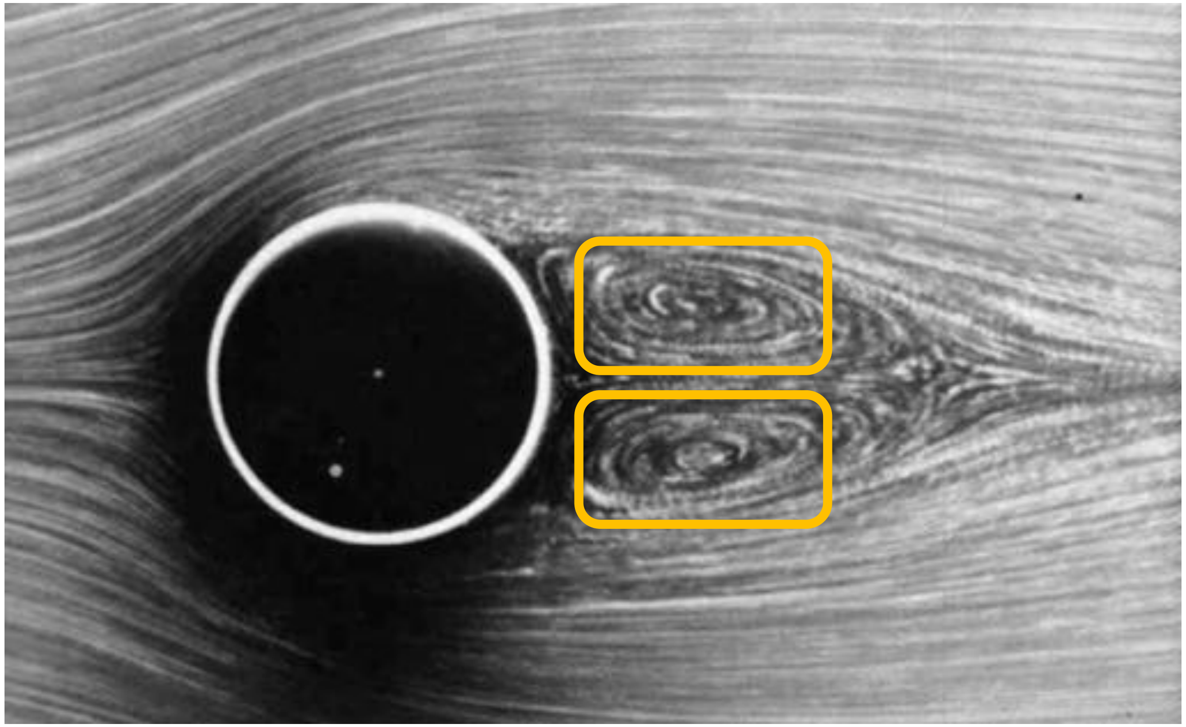


[Oil and magnesium.]

Cylinder (Re = 26) – 2 vortices

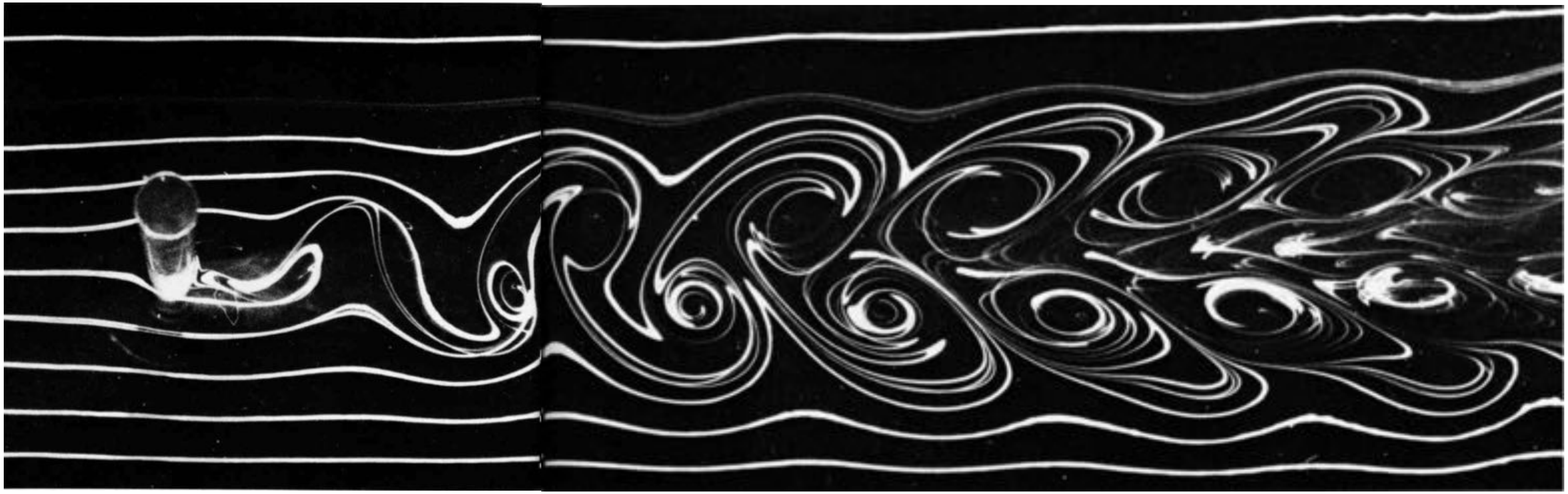
- $\nabla \cdot u = 0$
- $u_1 = f(x_2)$
- $u_2 = g(x_1)$





[Oil and magnesium.]

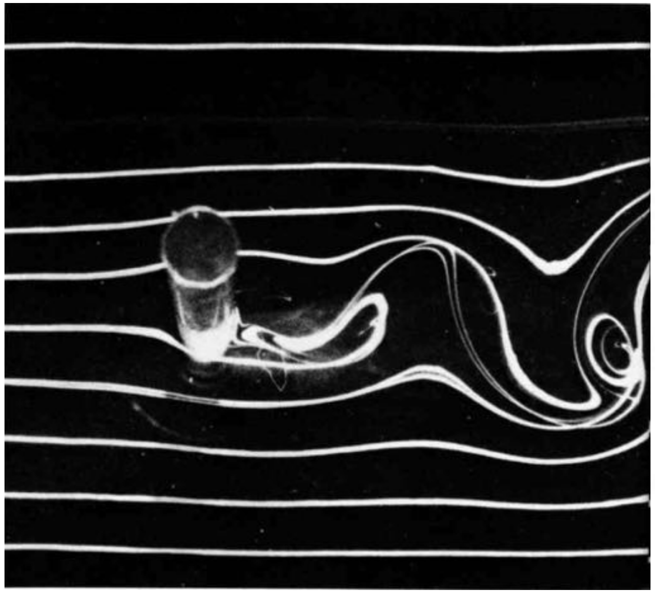
Cylinder (Re = 300) – Karman vortex street



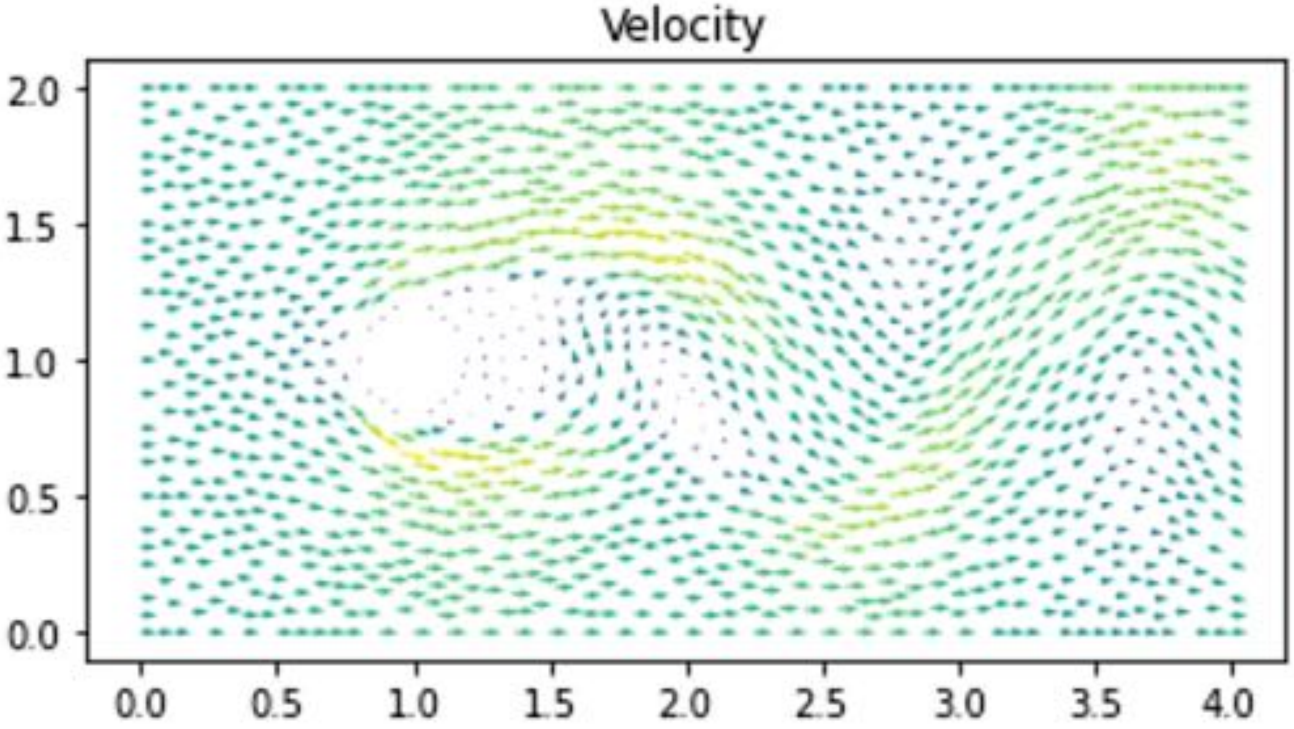
[Wind and smoke.]

Lab 2 – The Navier-Stokes equations

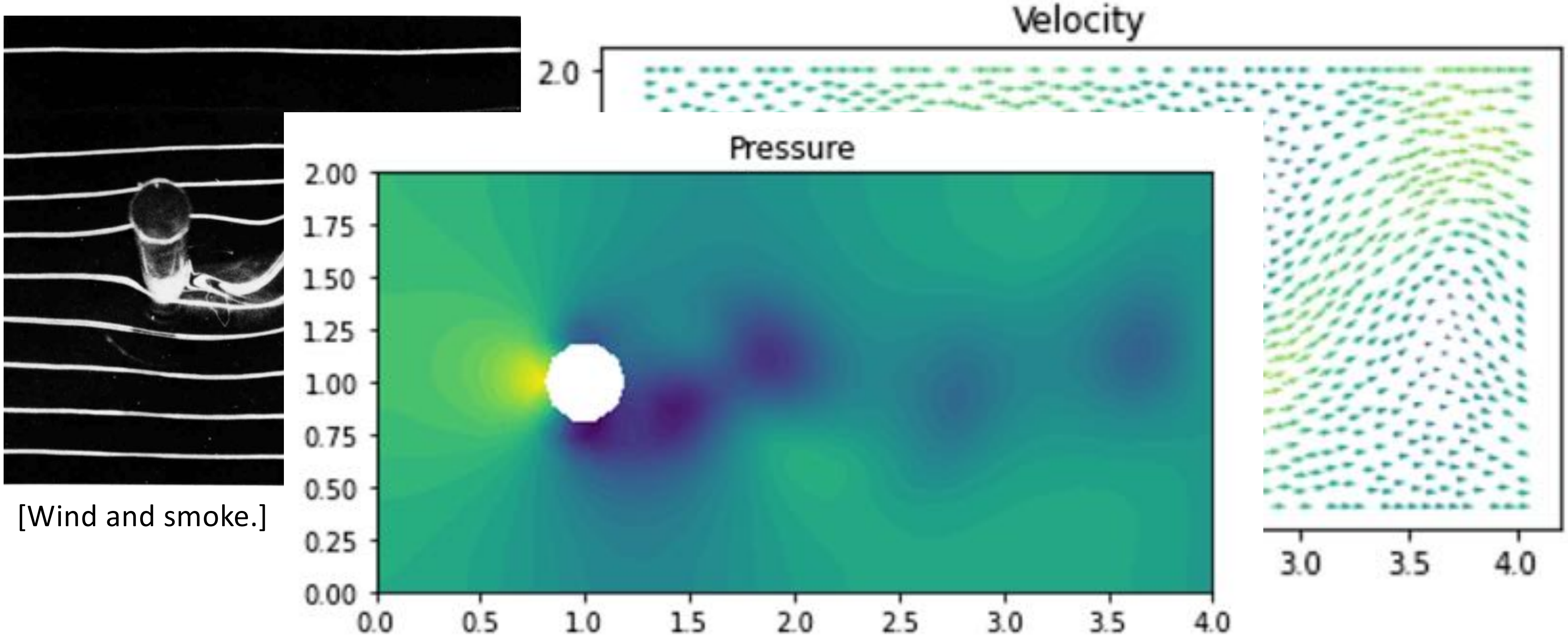
Lab 2 – velocity vector field



[Wind and smoke.]

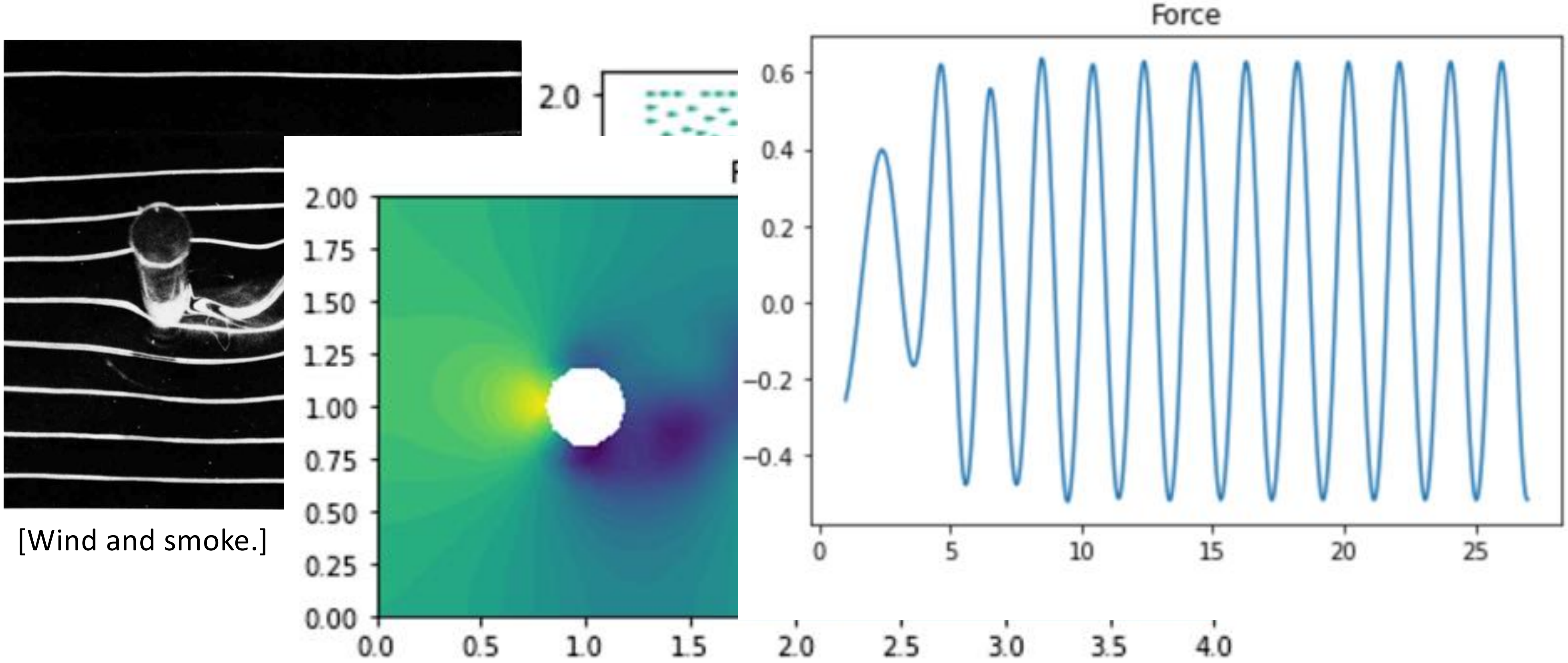


Lab 2 – scalar pressure field



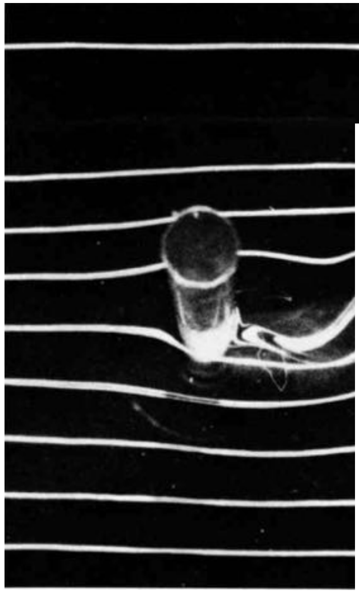
[Wind and smoke.]

Lab 2 – lift force on cylinder

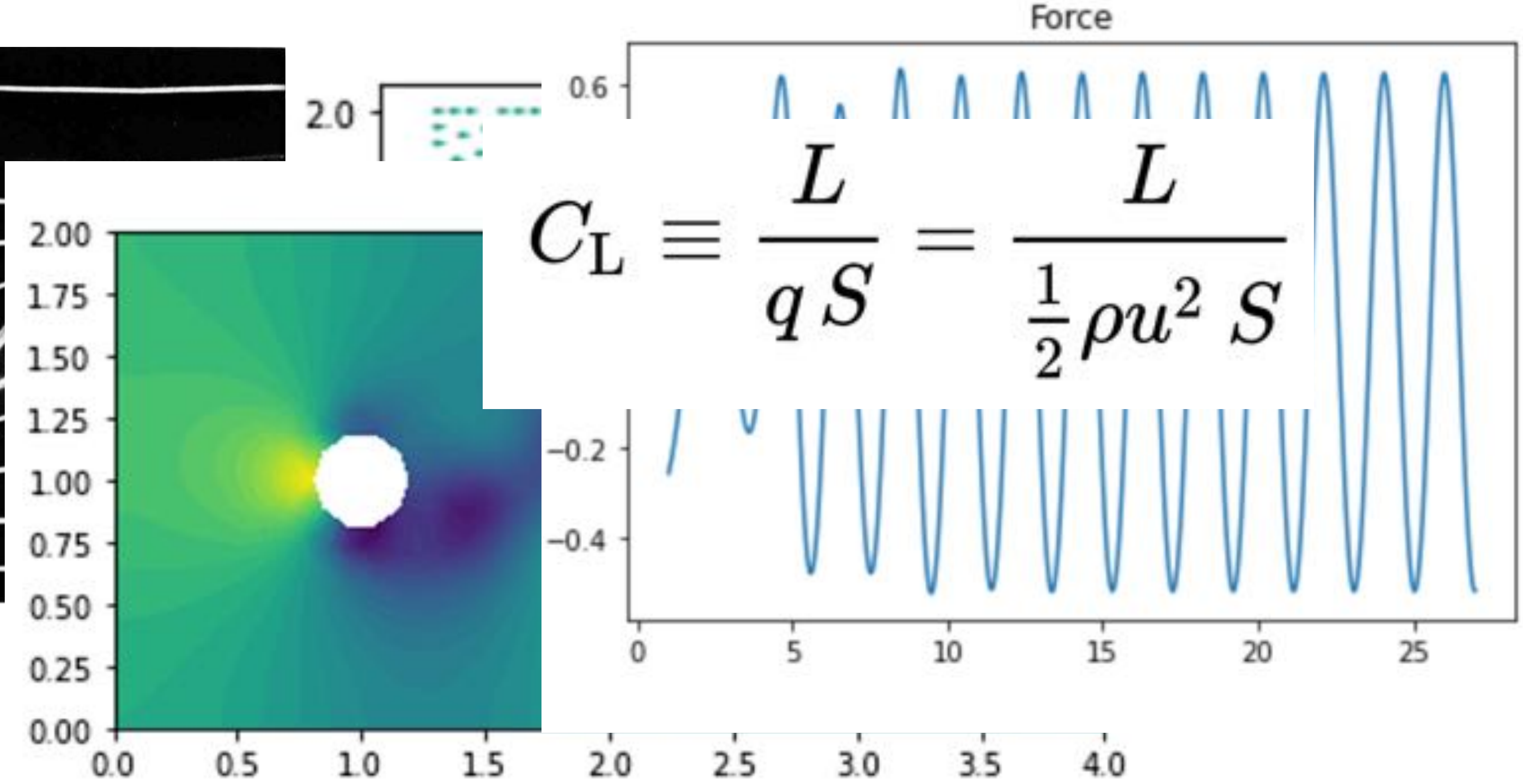


[Wind and smoke.]

Lab 2 – lift force on cylinder

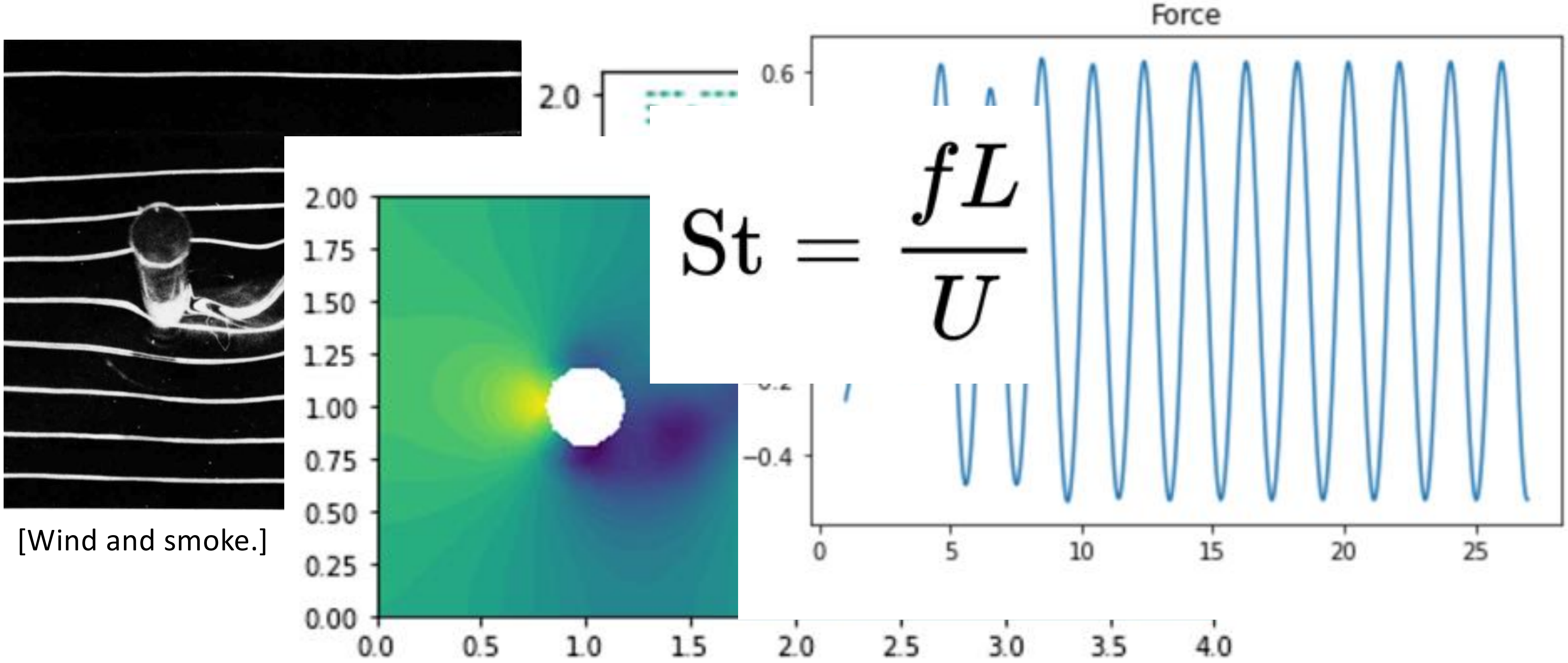


[Wind and smoke.]



$$C_{\mathrm{L}} \equiv \frac{L}{q\,S} = \frac{L}{\frac{1}{2}\rho u^2\ S}$$

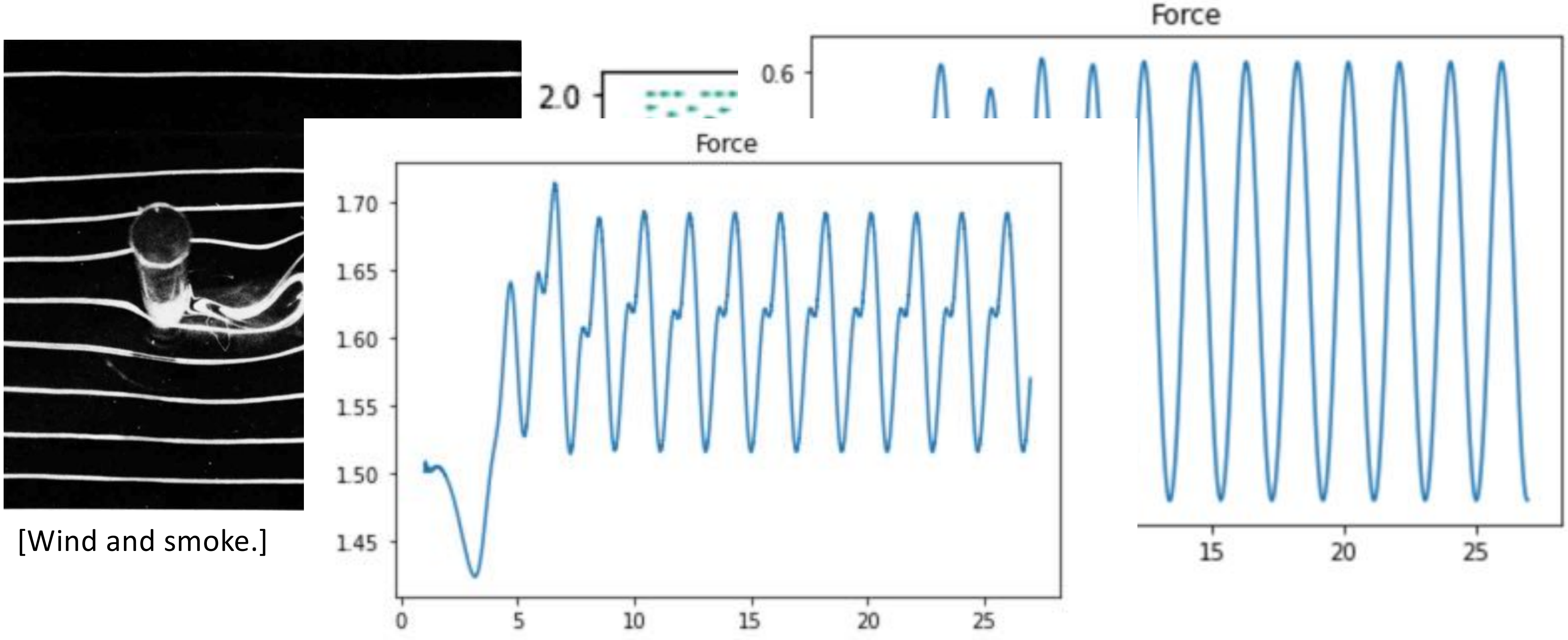
Lab 2 – Strouhal number



[Wind and smoke.]

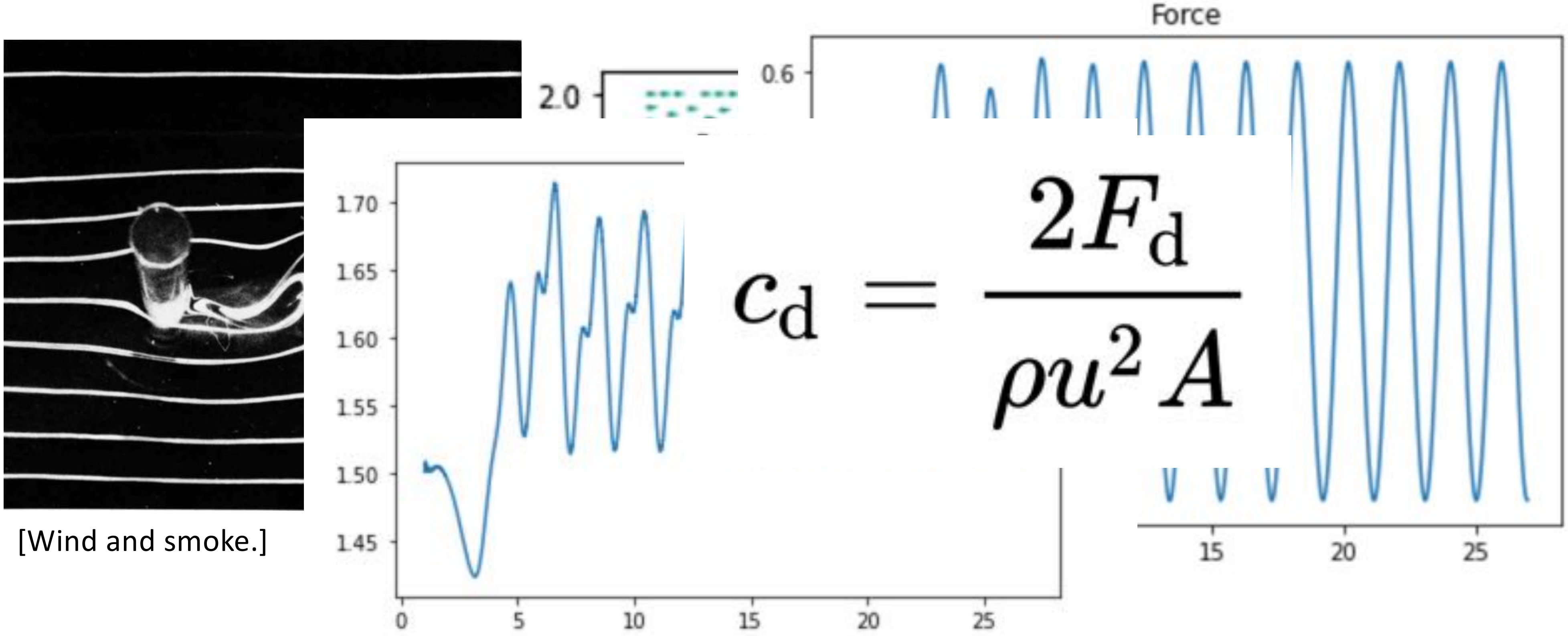
$$\mathrm{St} = \frac{fL}{U}$$

Lab 2 – drag force on cylinder



[Wind and smoke.]

Lab 2 – drag force on cylinder

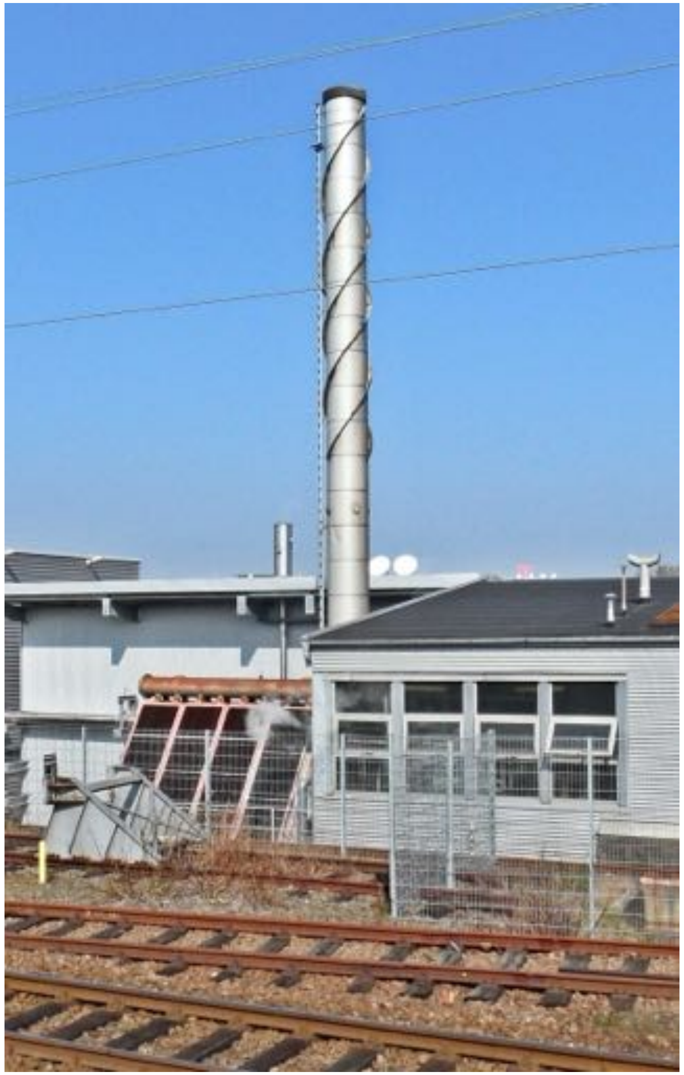


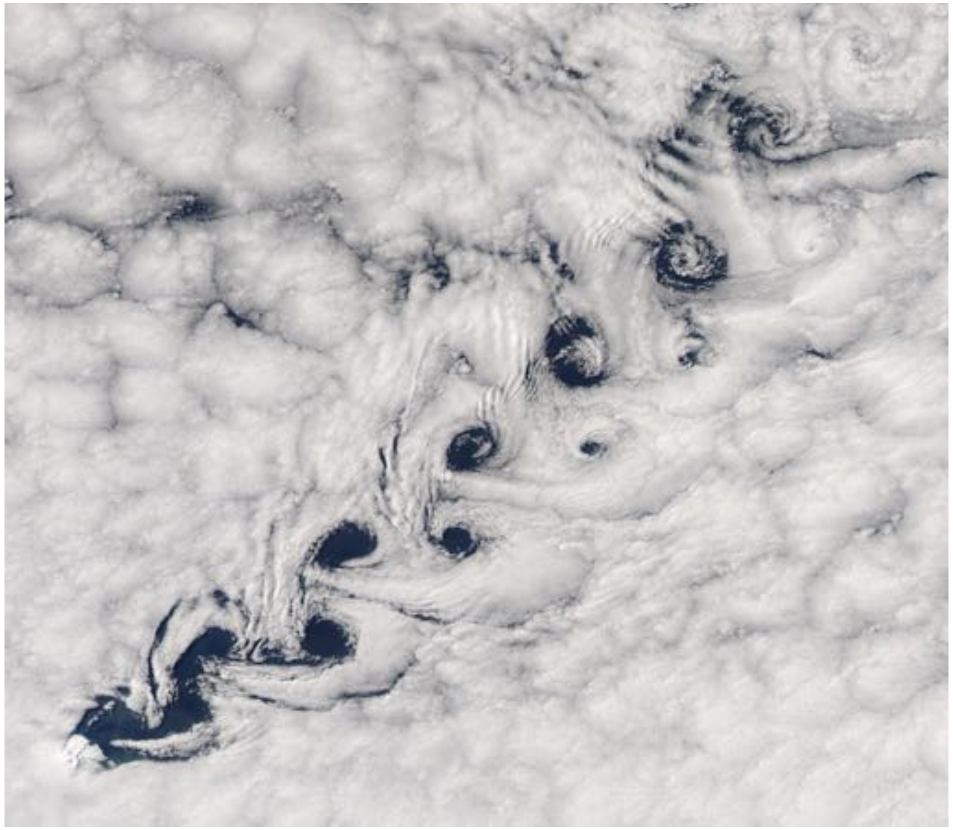
[Wind and smoke.]

$$c_\mathrm{d} = \frac{2F_\mathrm{d}}{\rho u^2 A}$$

[https://en.wikipedia.org/wiki/Vortex_shedding#/media/File:SchornsteinwendelSKL.jpg]

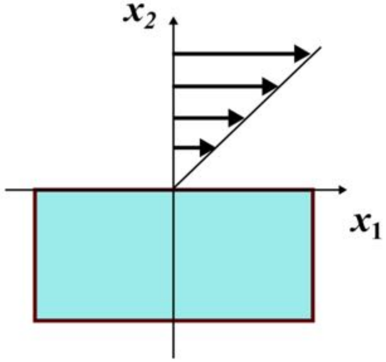
Karman vortex streets

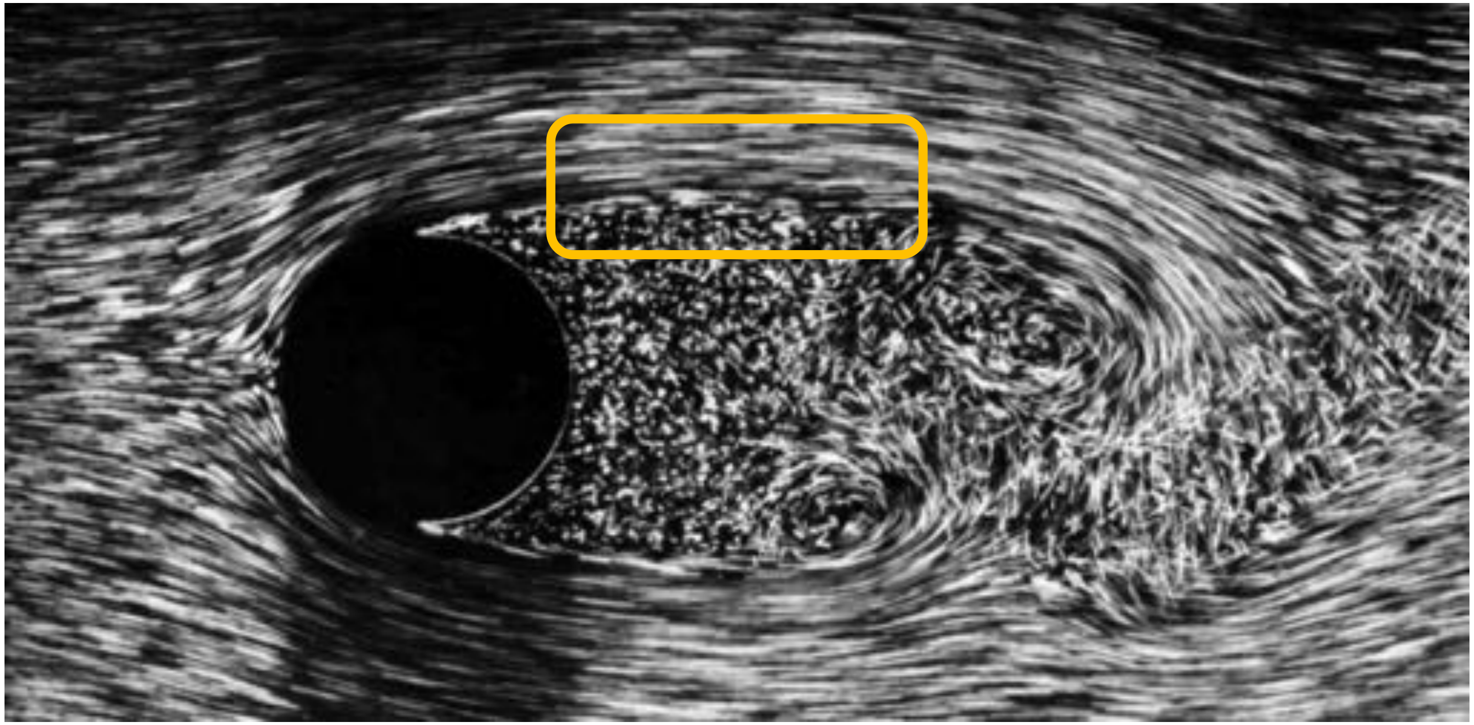




[https://en.wikipedia.org/wiki/Vortex_shedding#/media/File:Heard_Island_Karman_vortex_street.jpg]

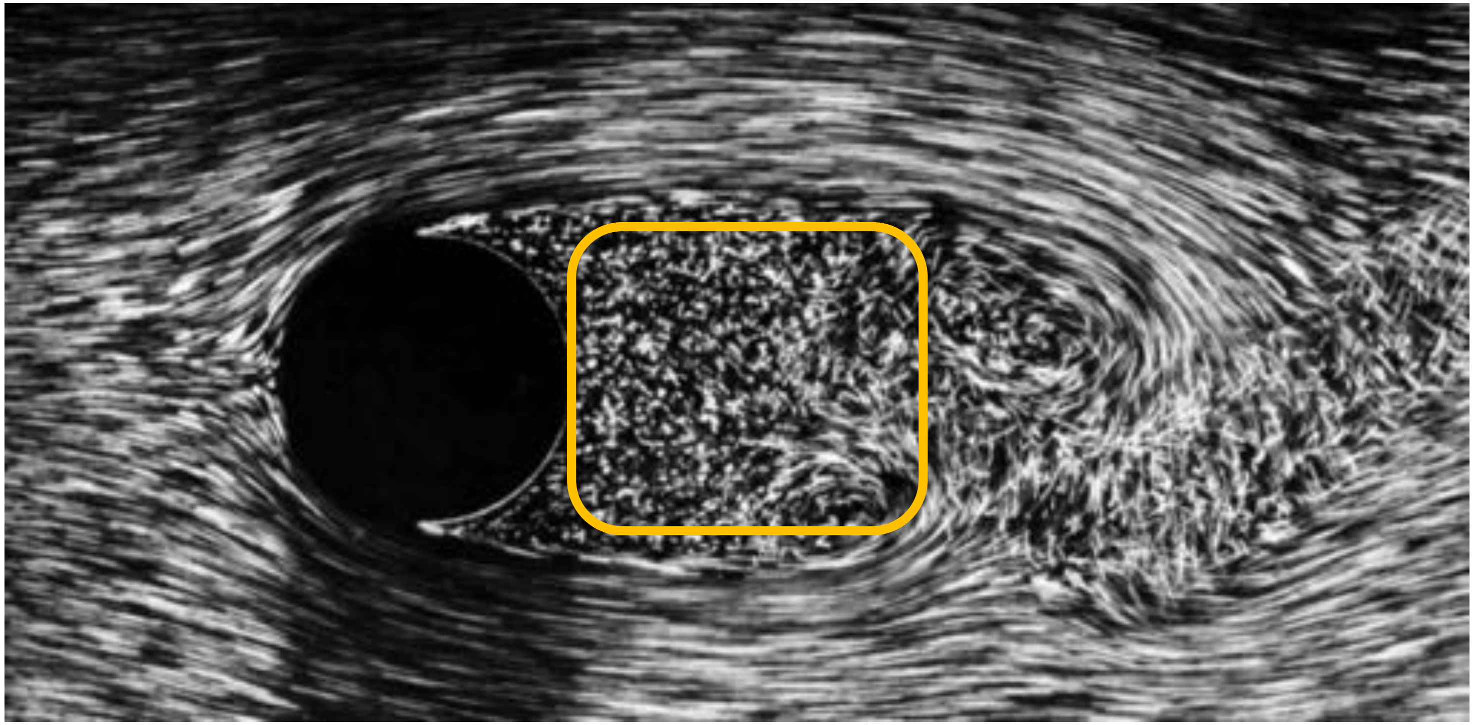
Cylinder (Re = 2000) – shear layer





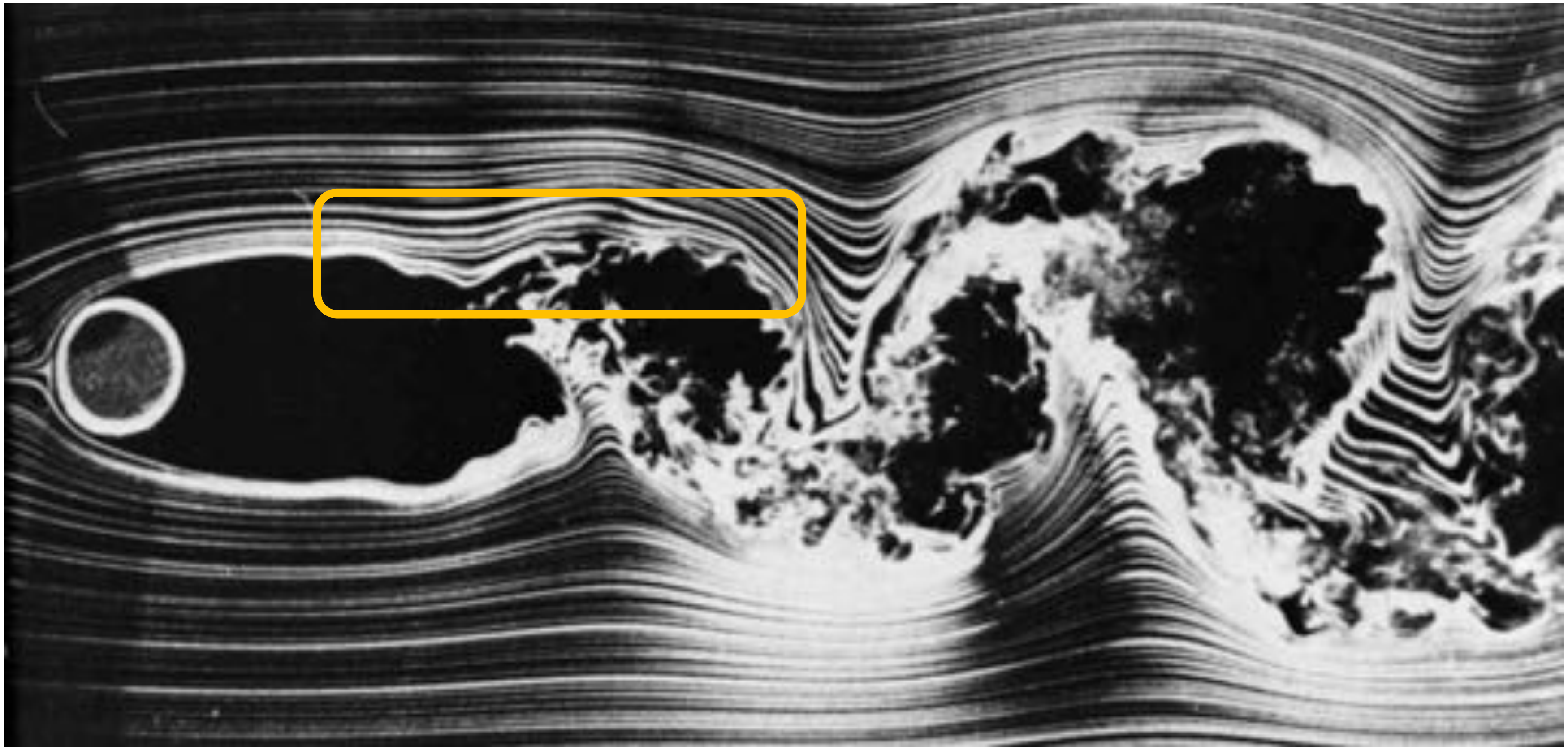
[Water and air bubbles.]

Cylinder (Re = 2000) – 3D turbulent wake



[Water and air bubbles.]

Re = 10 000 – turbulent shear layers

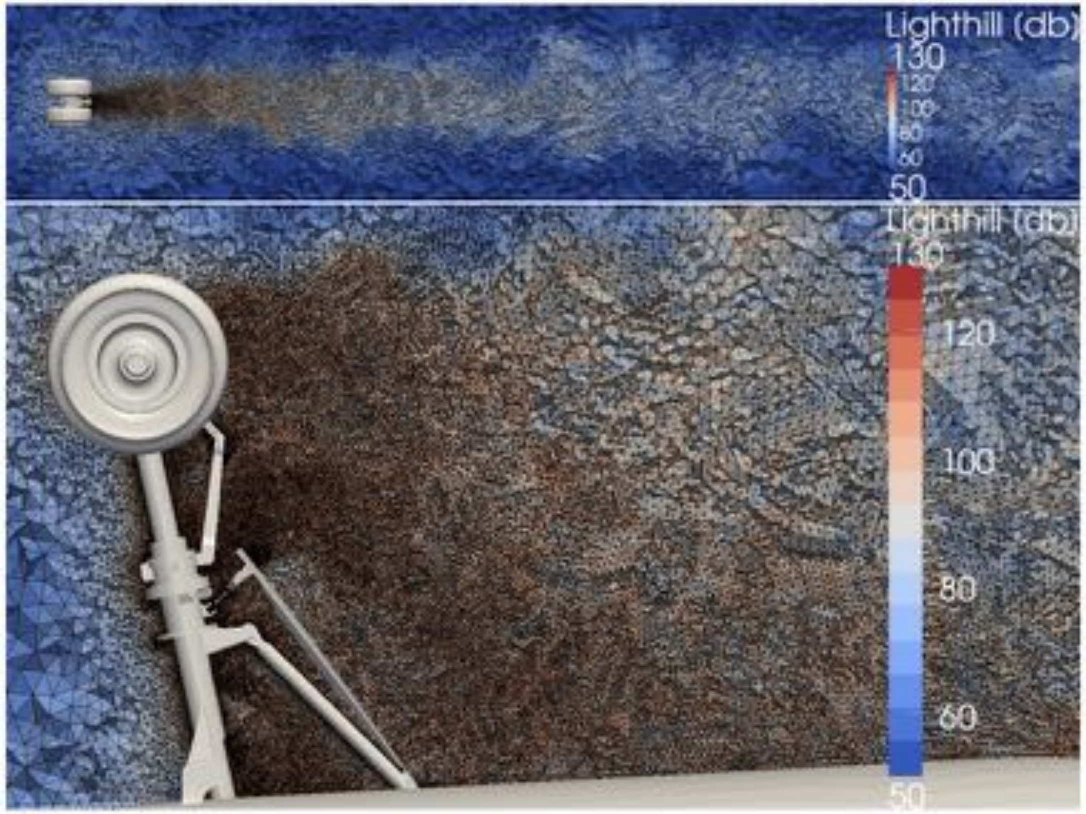


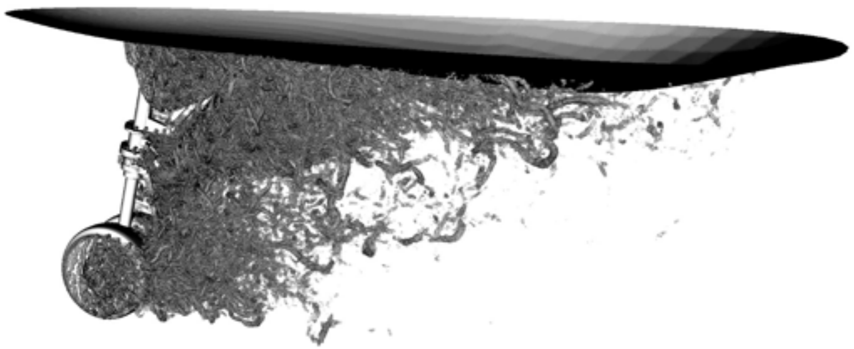
[Water and air bubbles.]

Simulation of airflow past landing gear

[De Abreu et al., Computers and Fluids, 2016]

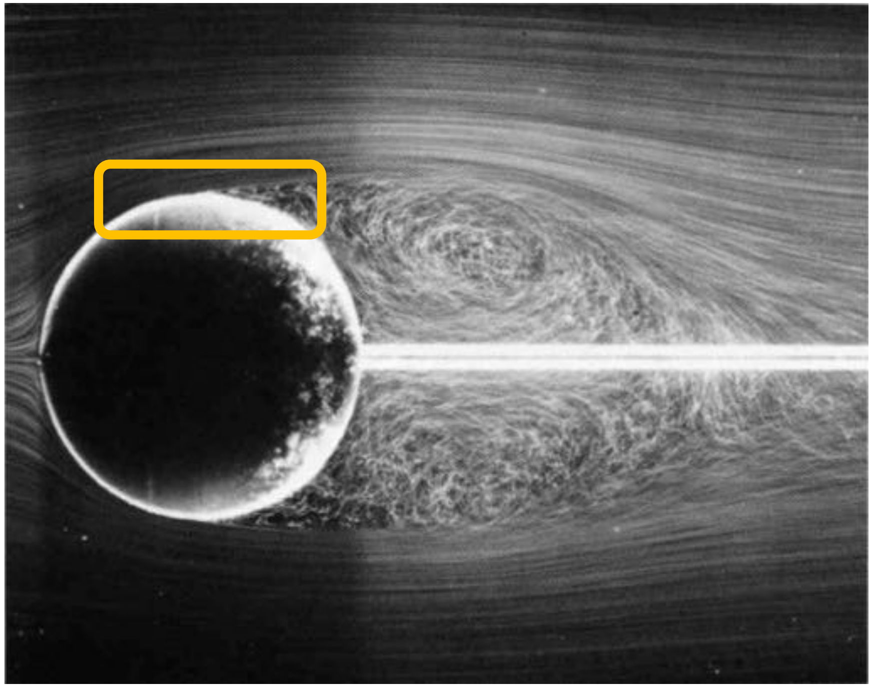
Acoustic sources and turbulent vortices

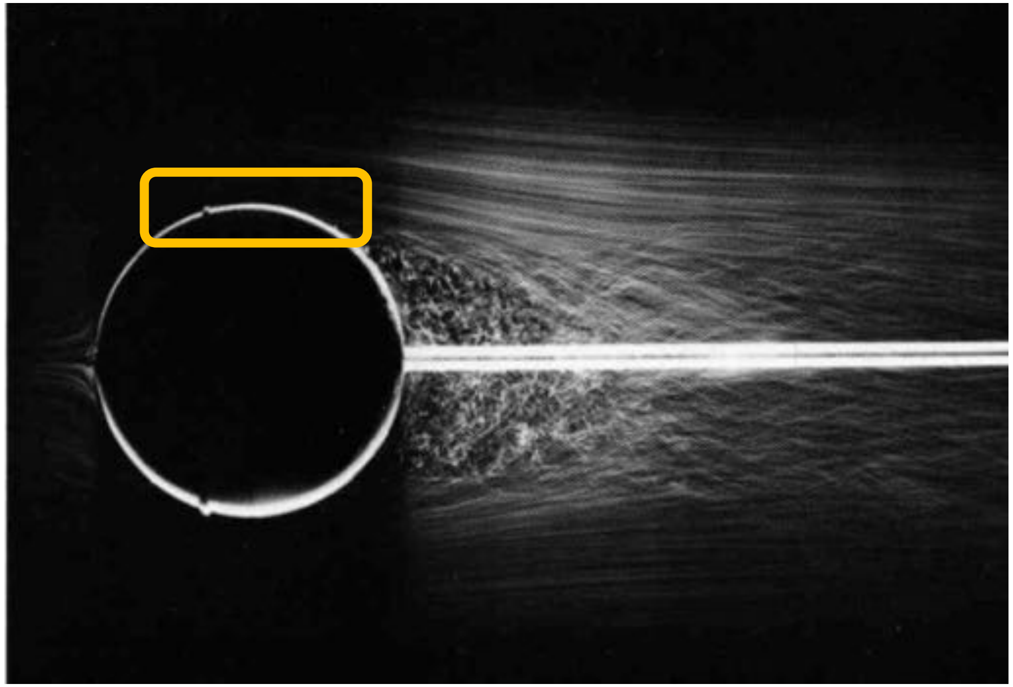




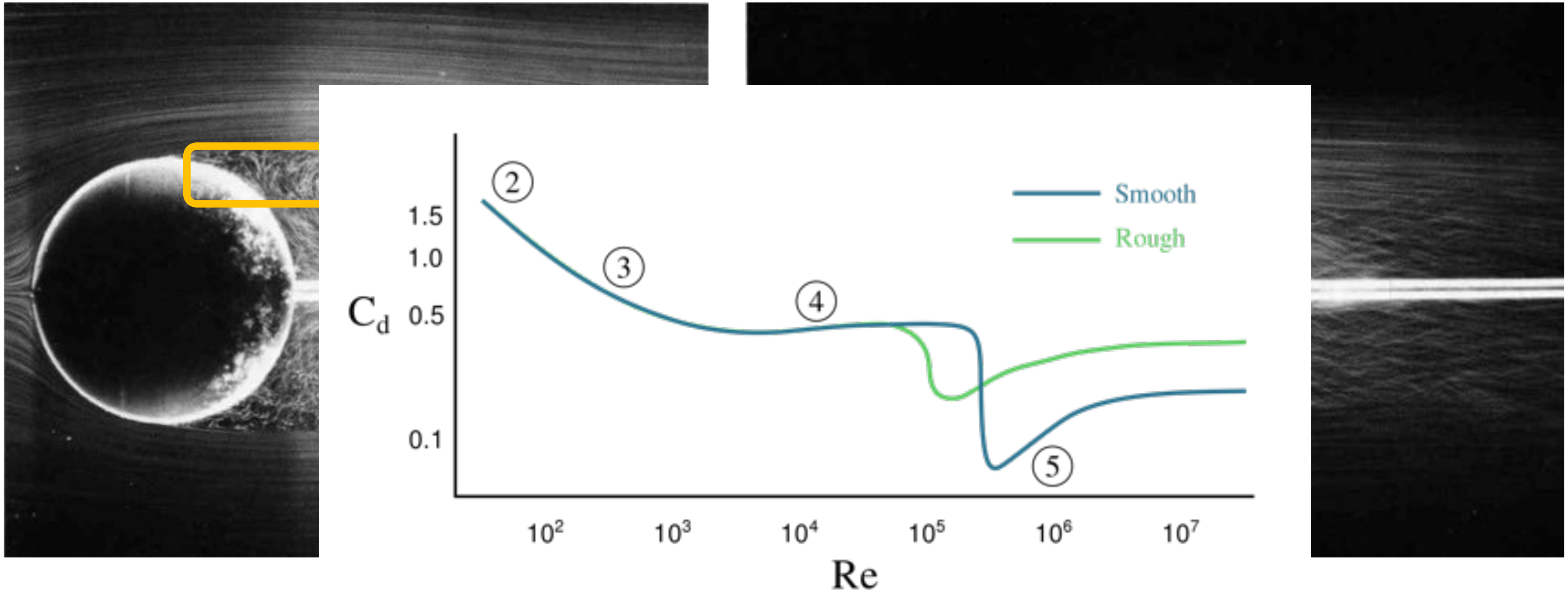
[De Abreu et al., Computers and Fluids, 2016]

Sphere: Re = 15 000 vs 30 000
turbulent boundary layers (drag crisis)



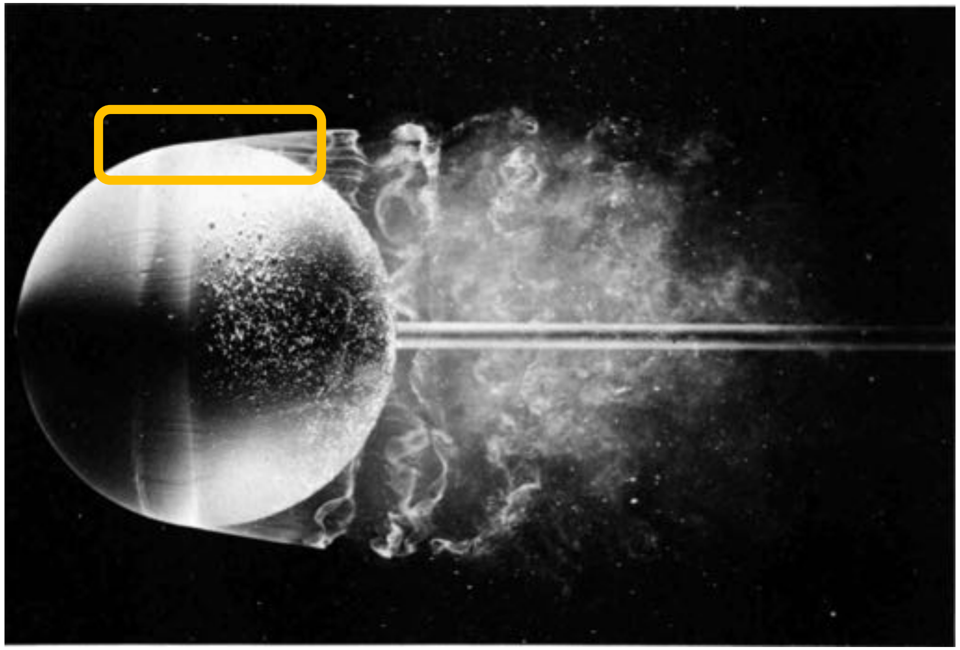


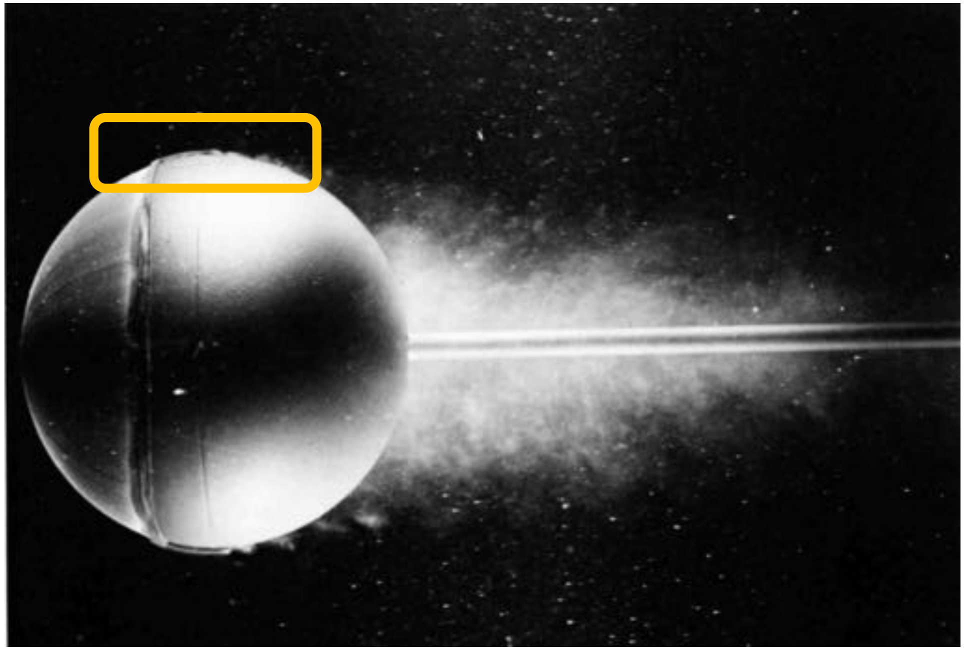
Sphere: Re = 15 000 vs 30 000
turbulent boundary layers (drag crisis)



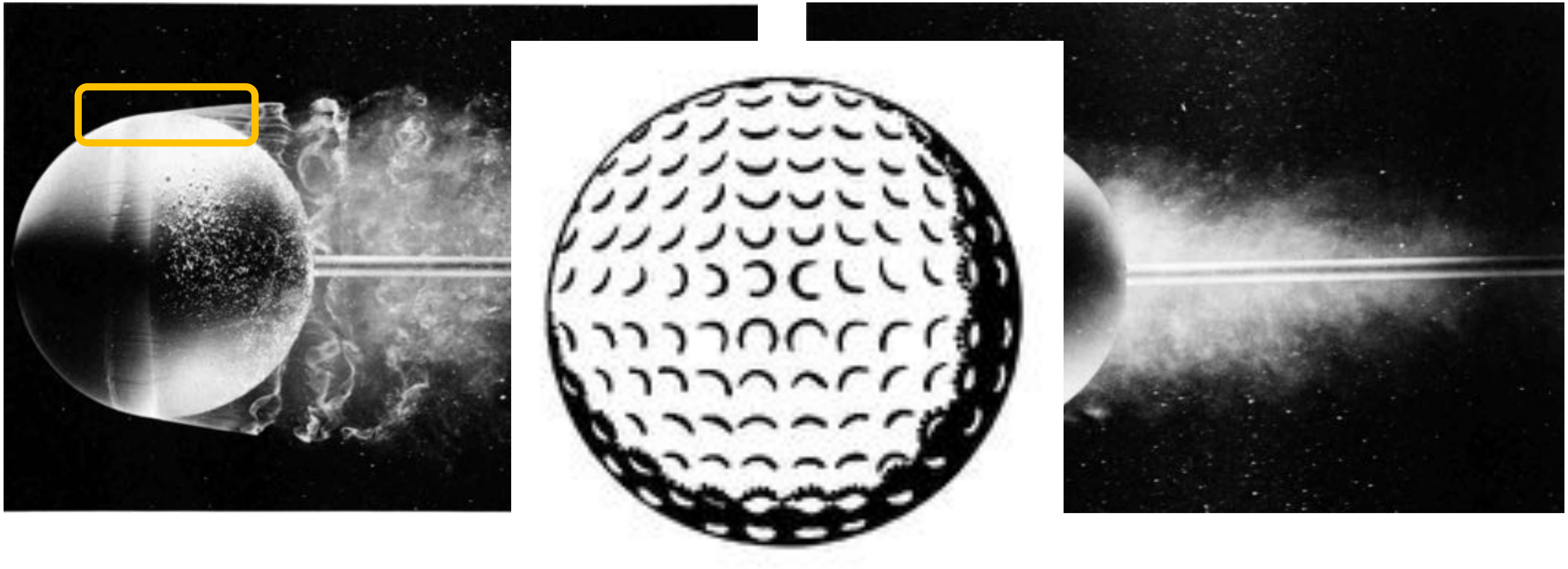
Sphere: Re = 15 000 vs 30 000

trip wire – to trigger turbulent boundary layer



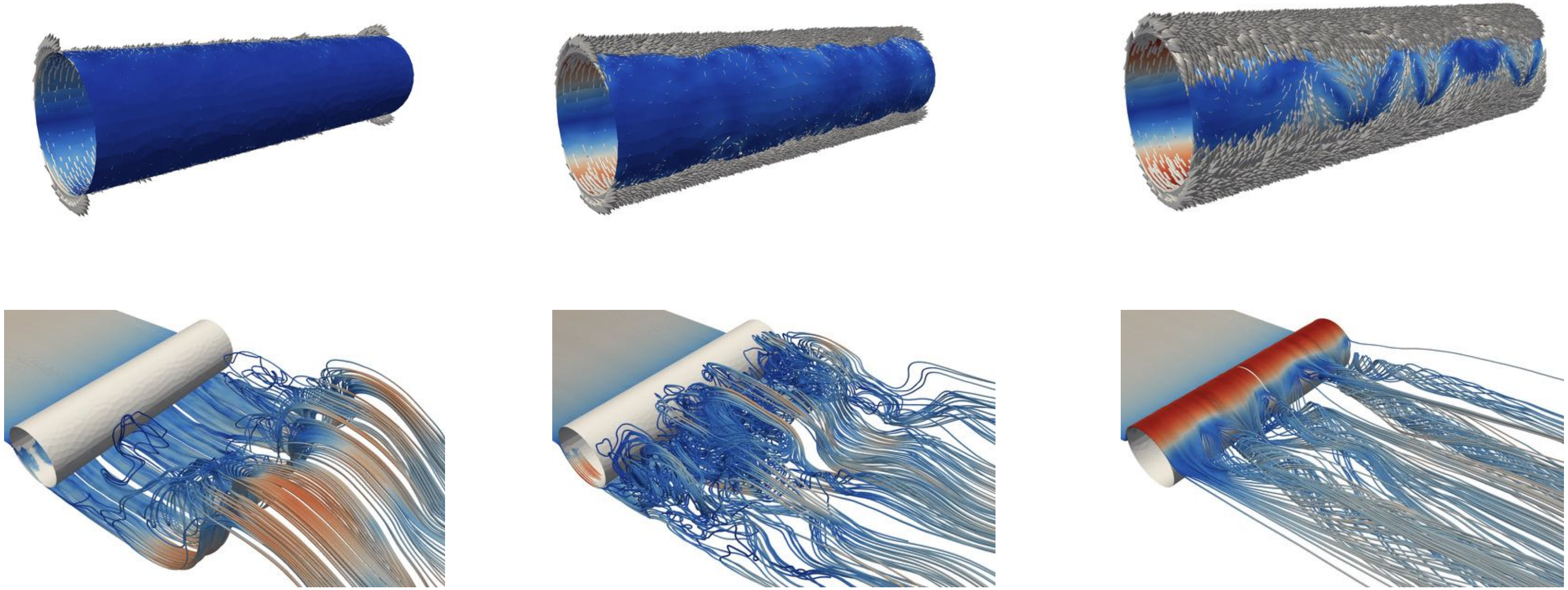


Sphere: Re = 15 000 vs 30 000
trip wire – to trigger turbulent boundary layer

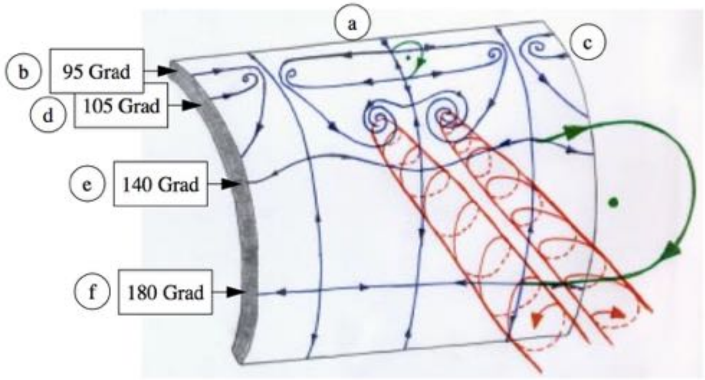


[https://en.wikipedia.org/wiki/Golf_Ball#/media/File:Golf_Ball.jpg]

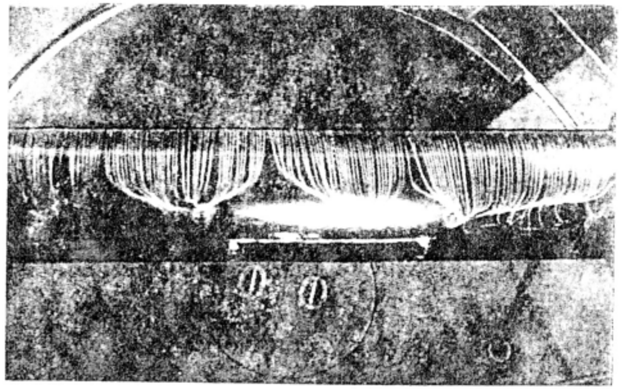
Simulation of drag crisis – slip/friction bc



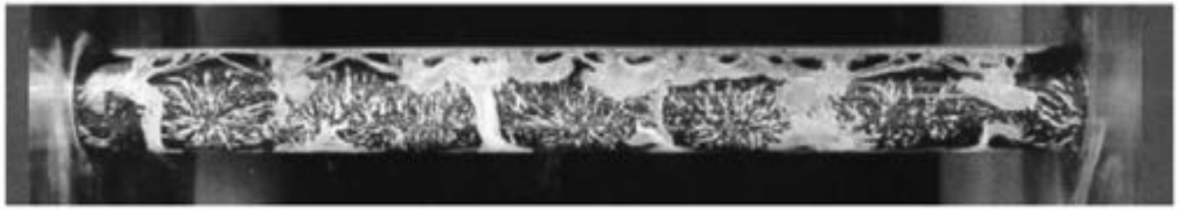
Streamwise vortex structures in the wake

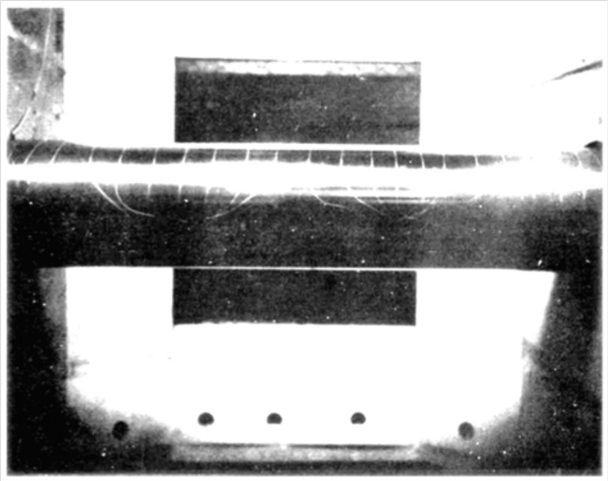


[Gölling DLR 2001]



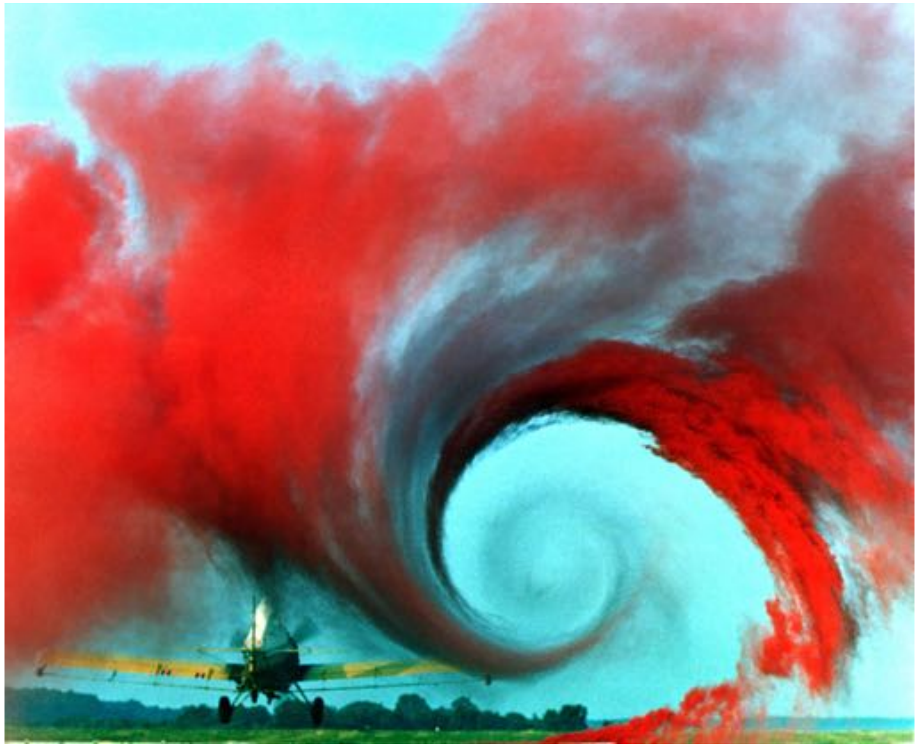
[Korotkin 1976]



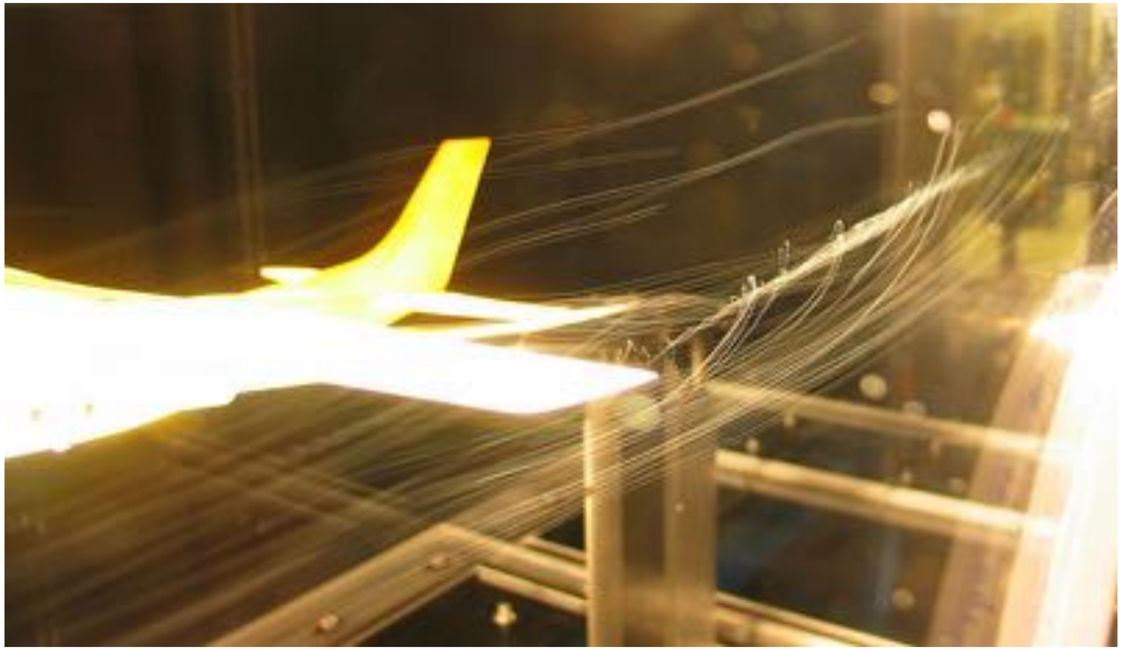


[Humphreys JFM 1960]

Wing – streamwise vortex structures



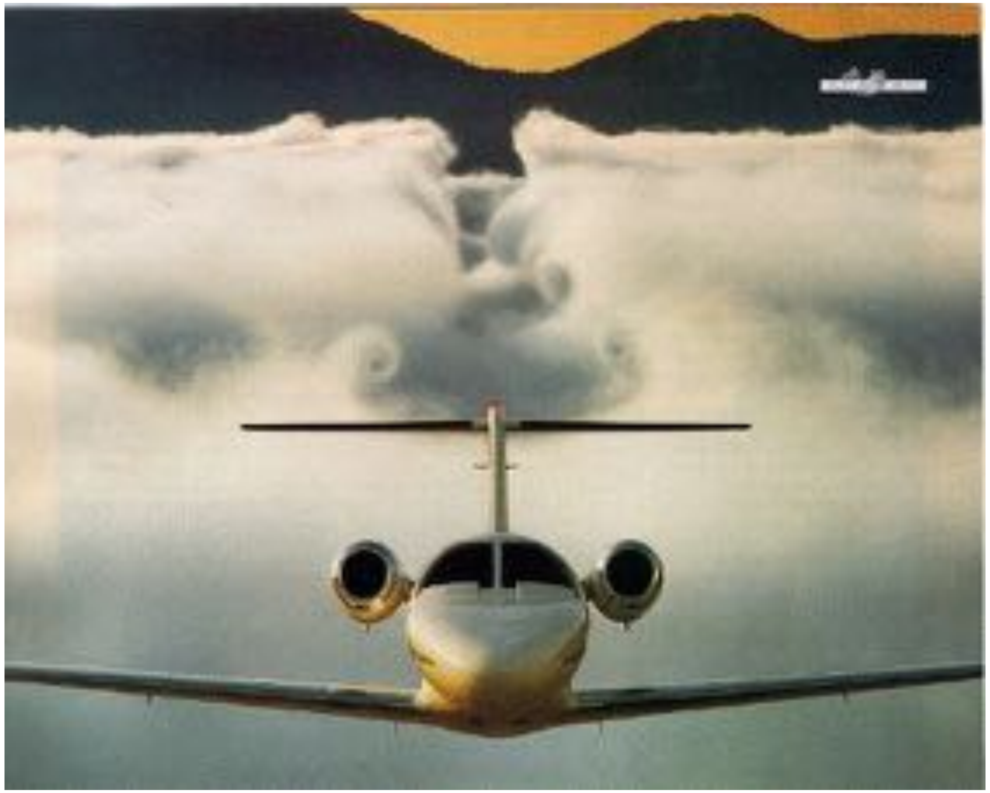
[https://en.wikipedia.org/wiki/File:Airplane_vortex_edit.jpg]



[https://sv.m.wikipedia.org/wiki/Fil:Cessna_182_model-wingtip-vortex.jpg]

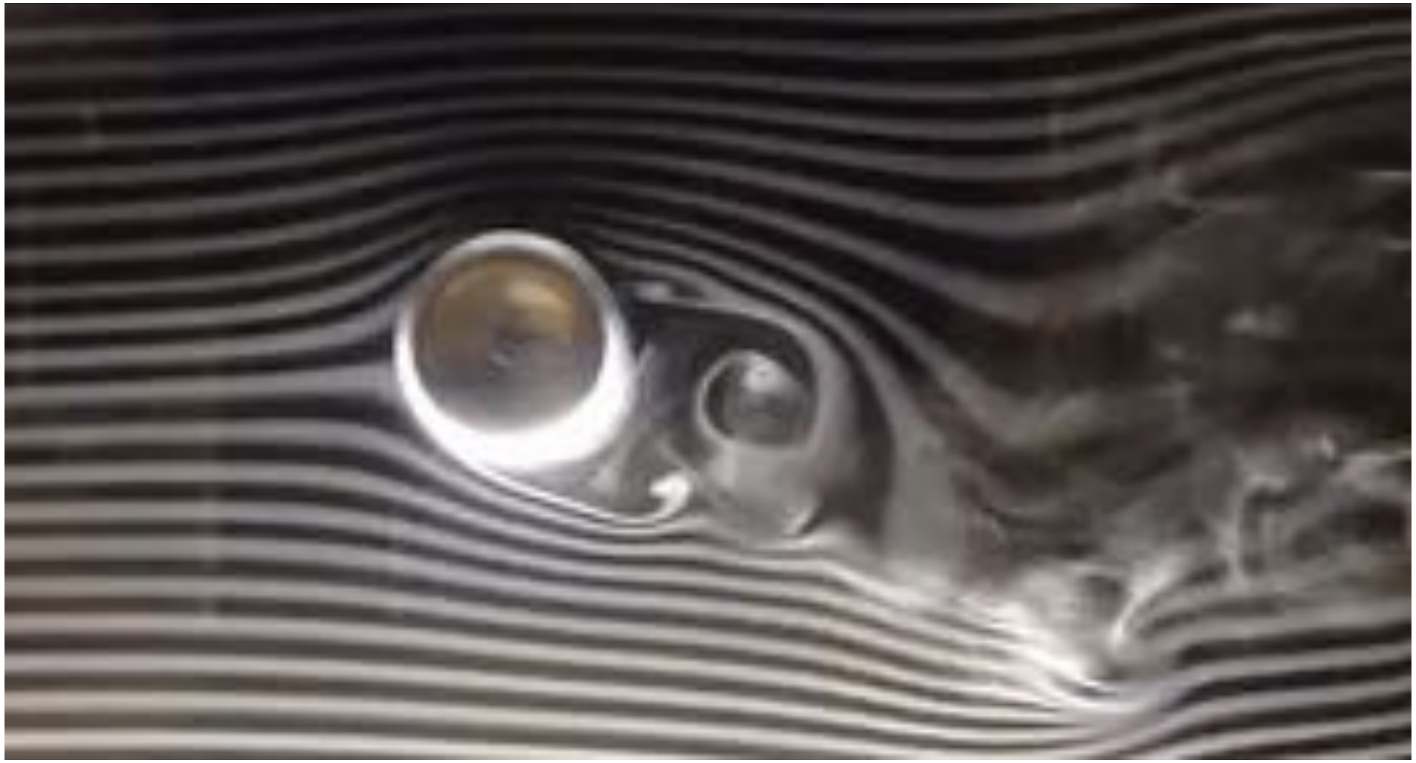
Wing – lift by delayed separation - downwash

[https://i.redd.it/7yj9h0x2h9f61.jpg]



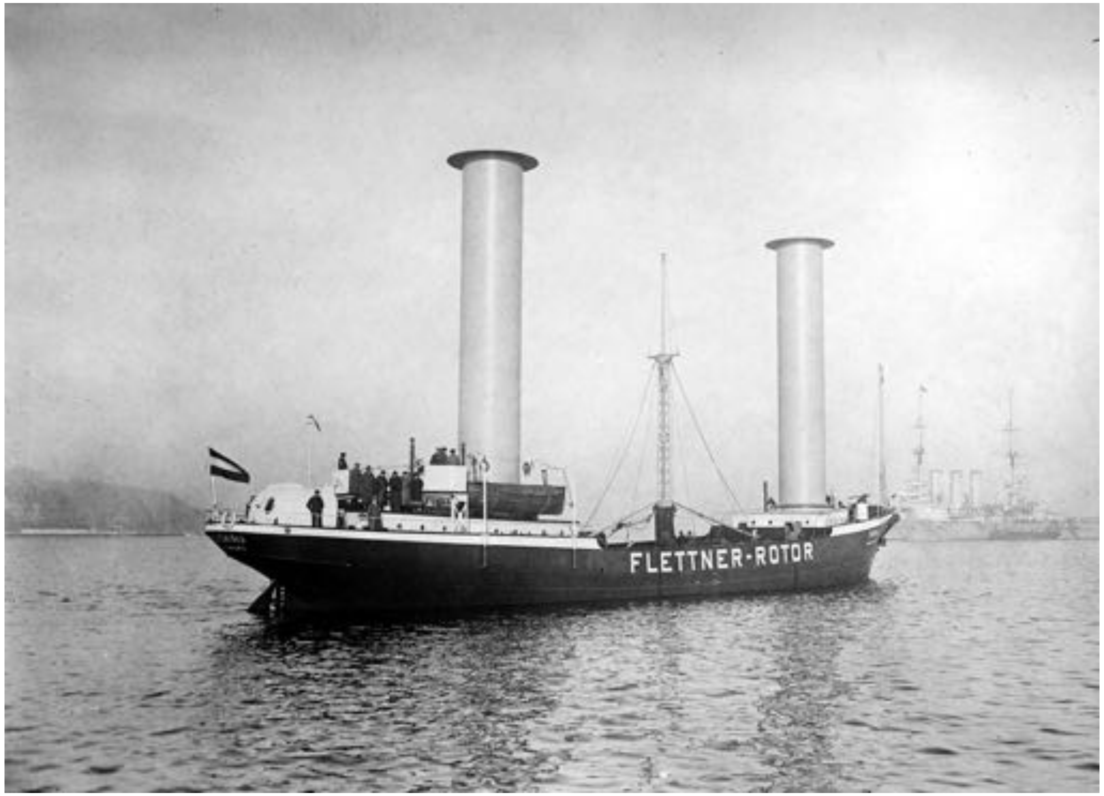
[https://www.grc.nasa.gov/www/k-12/airplane/downwash.html]

Magnus effect – downwash through rotation



[https://en.wikipedia.org/wiki/Magnus_effect#/media/File:Magnus-anim-canette.gif]

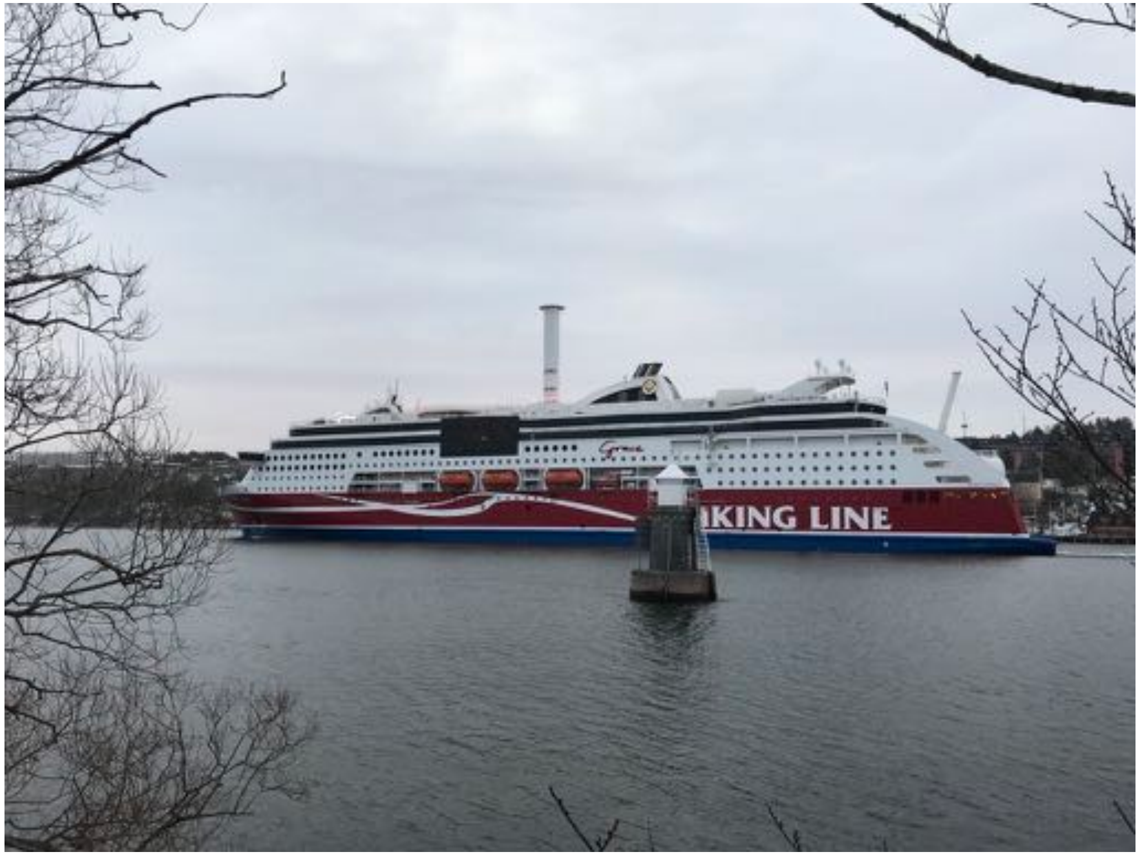
Buckau - Flettner rotor ship 1924



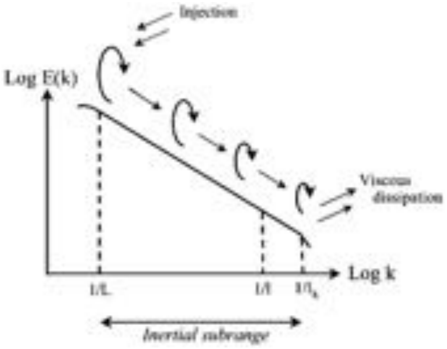
[https://en.wikipedia.org/wiki/Flettner_rotor#/media/File:Buckau_Flettner_Rotor_Ship_LOC_37764u.jpg]

Grace - Flettner rotor ship 2019

Grace - Flettner rotor ship 2019



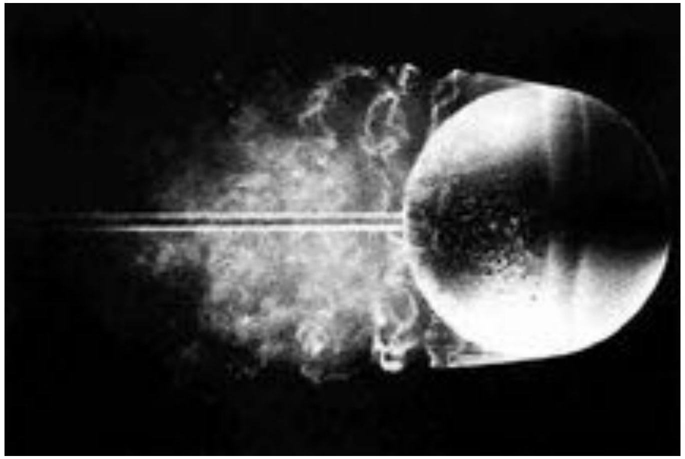
Direct Numerical Simulation (DNS)

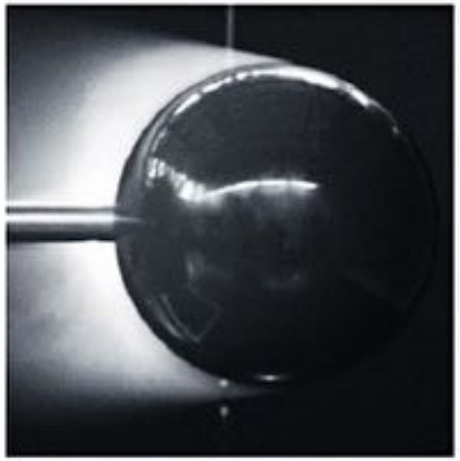


- Energy dissipation to heat at smallest scale in turbulent flow (Kolmogorov scale) ~ $Re^{-3/4}$
- Cost of Direct Numerical Simulation (DNS) ~ $Re^{9/4}$
- $Re > 10^6$: DNS impossible! Need cheaper models!

Reynolds Averaged NSE (RANS)

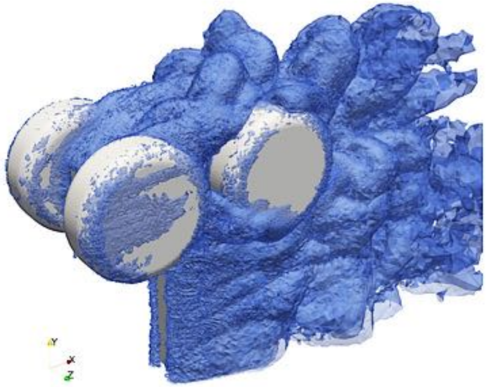
- Compute statistical ensemble average of NSE solutions
- Introduces Reynolds stresses that need to be modeled (turbulent eddy viscosity) : the closure problem
- Calibration of RANS model parameters is a challenge

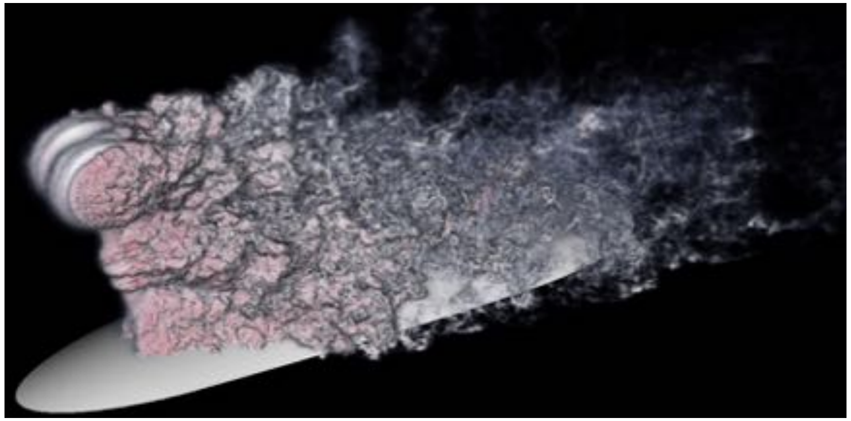




Large Eddy Simulation (LES)

- Compute only the largest scales of the flow
- Filter NSE (spatial average): model subgrid scales
- Near walls “Resolved LES” impractical for high Re
- Wall-layer models: e.g., assume RANS near the wall (DES)





Direct FEM Simulation of Turbulence (DFS)

- Direct FEM approximation of the Navier-Stokes equations
  - No RANS/LES averaging/filtering (only the mesh scale)
  - No explicit turbulence/subgrid model (no closure problem)
  - Automatic turbulence model based on NSE residual
    (cf. Implicit LES, MILES [Fureby/Grinstein AIAA 99],
    VMM-LES [Bazilevs et.al. 07, Principe/Codina/Henke 10,…])

- Automatic mesh resolution : adaptive algorithm with error control in output of interest (drag, lift,…)
- Cheap wall-layer model : slip (friction) boundary condition
  (no boundary layer resolution : no boundary layer mesh)

Direct FEM Simulation of Turbulence (DFS)

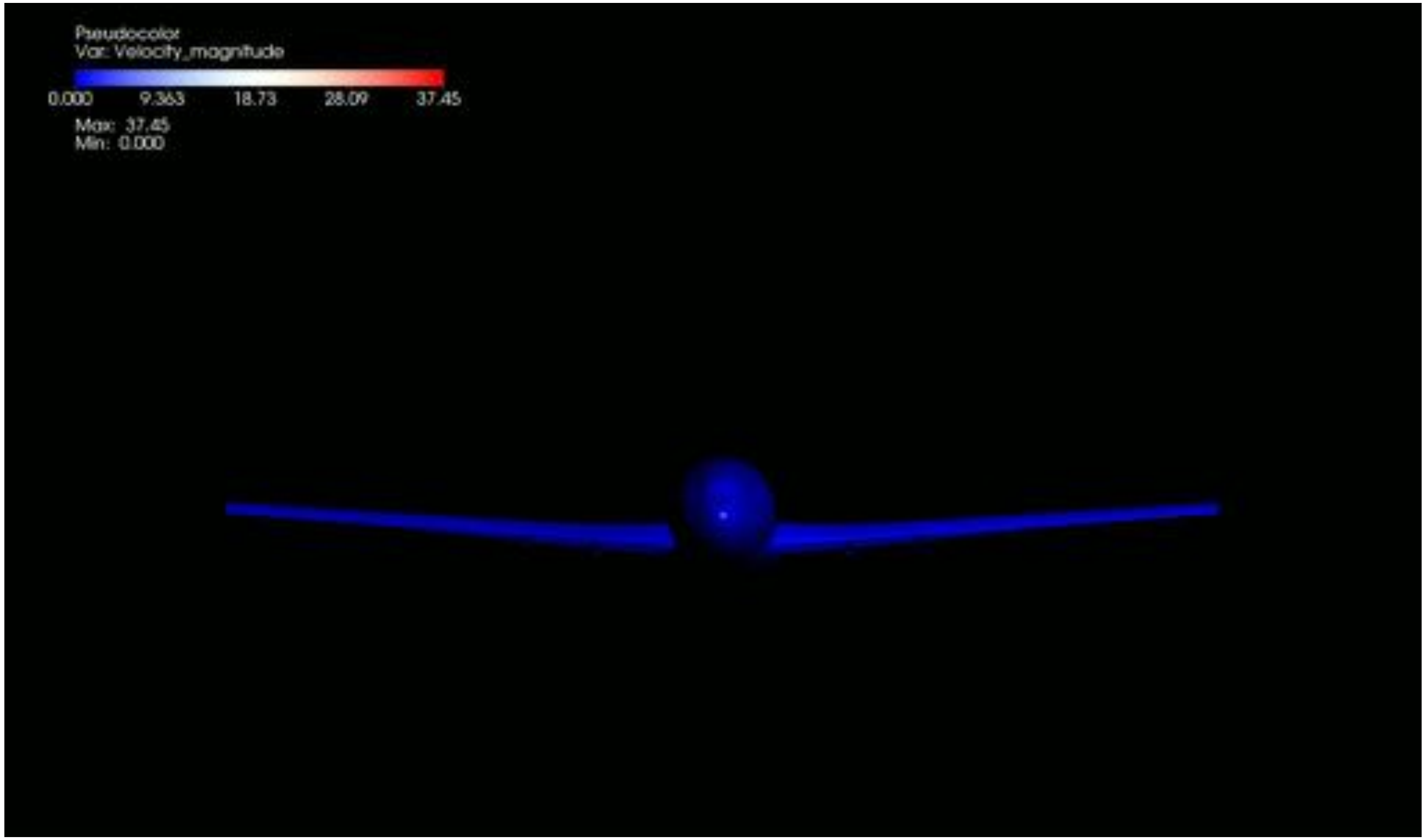
- For (v,q) in W_h : find (U,P) in V_h such that

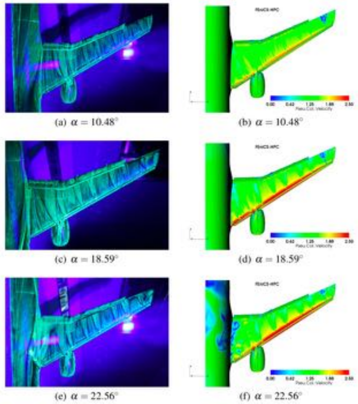
$$(D_t U + (U\cdot\nabla)U,v) + (\nu\nabla U,\nabla v) - (P,\nabla\cdot v) + (q,\nabla\cdot U) \\ + (\delta R(U,P),R(v,q)) = (f,v)$$

- Slip (no penetration) velocity: $u\cdot n = 0$
- Wall shear stress: $\tau = n^T\sigma t = \beta(u\cdot t)$ (β skin friction coefficient)

- Least squares stabilization of residual: $R(U,P)$, with $\delta \sim h$
- $R(v,q) = [D_t v + (U\cdot\nabla)v + \nabla q - \nu\Delta v - f, \nabla\cdot v]^T$
- No explicit subgrid model of unresolved scales
- Dissipation: $- dK/dt = \|\beta^{1/2}u\cdot t\|^2 + \|\nu^{1/2}\nabla U\|^2 + \|\delta^{1/2}R(U,P)\|^2$

Simulation of airflow past airplane

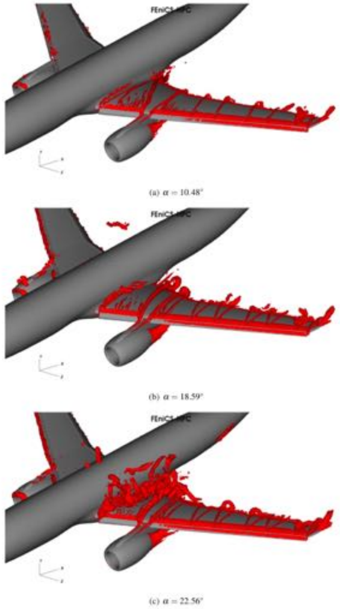




(a) $\alpha = 10.48^\circ$ (b) $\alpha = 10.48^\circ$

(c) $\alpha = 18.59^\circ$ (d) $\alpha = 18.59^\circ$

(e) $\alpha = 22.56^\circ$ (f) $\alpha = 22.56^\circ$



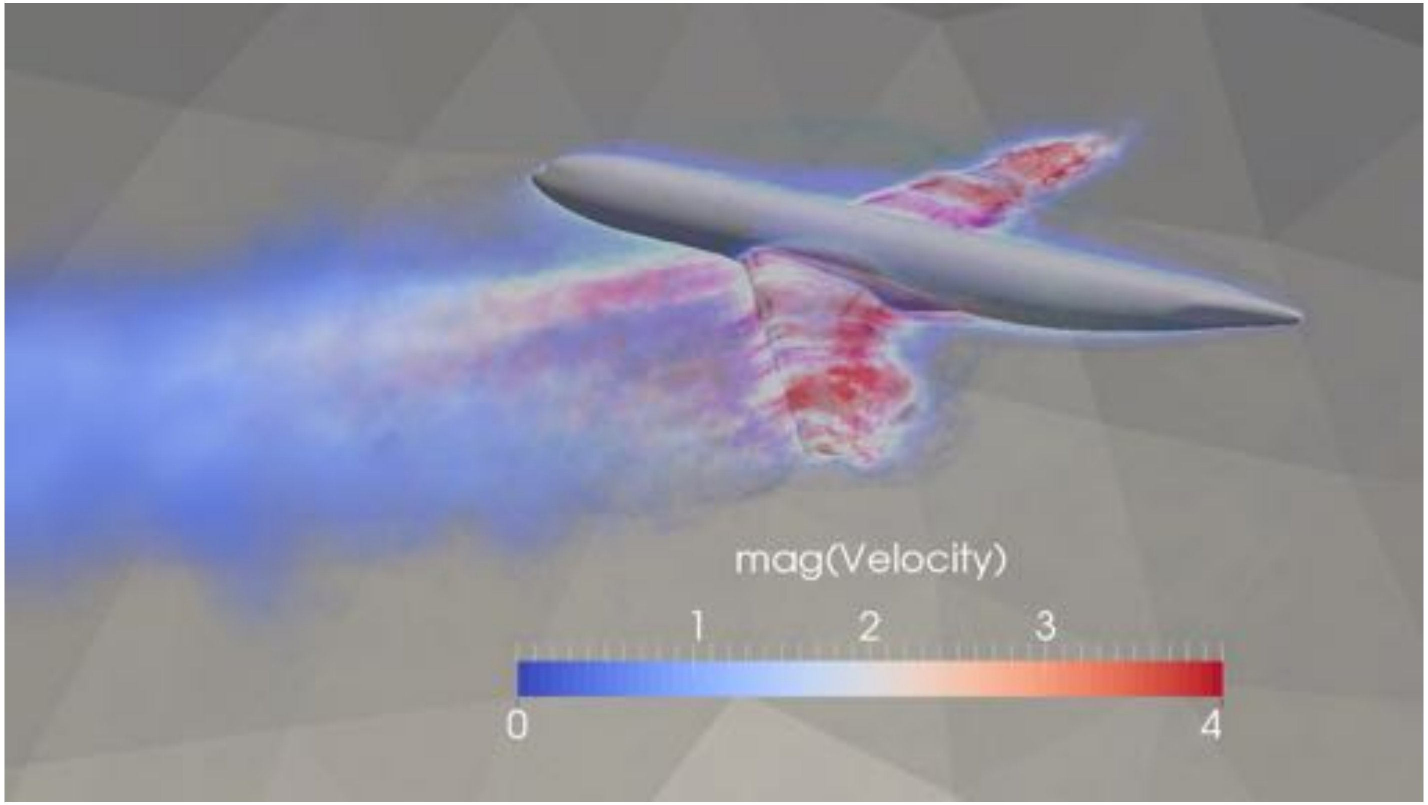
(a) $\alpha = 10.48^\circ$

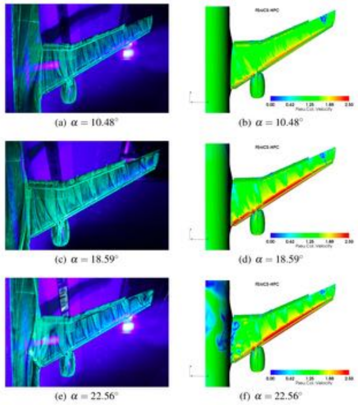
(b) $\alpha = 18.59^\circ$

(c) $\alpha = 22.56^\circ$

[Jansson et al., Springer, 2018]

Simulation of airflow past airplane





(a) $\alpha = 10.48°$

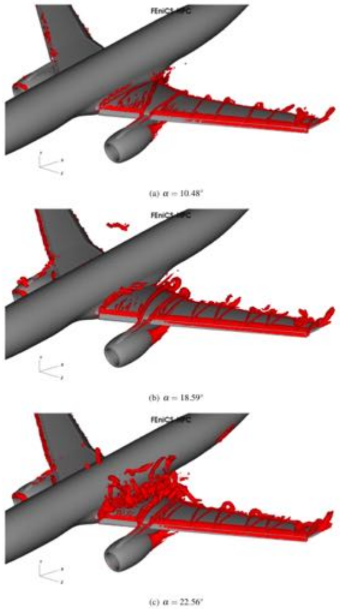
(b) $\alpha = 10.48°$

(c) $\alpha = 18.59°$

(d) $\alpha = 18.59°$

(e) $\alpha = 22.56°$

(f) $\alpha = 22.56°$



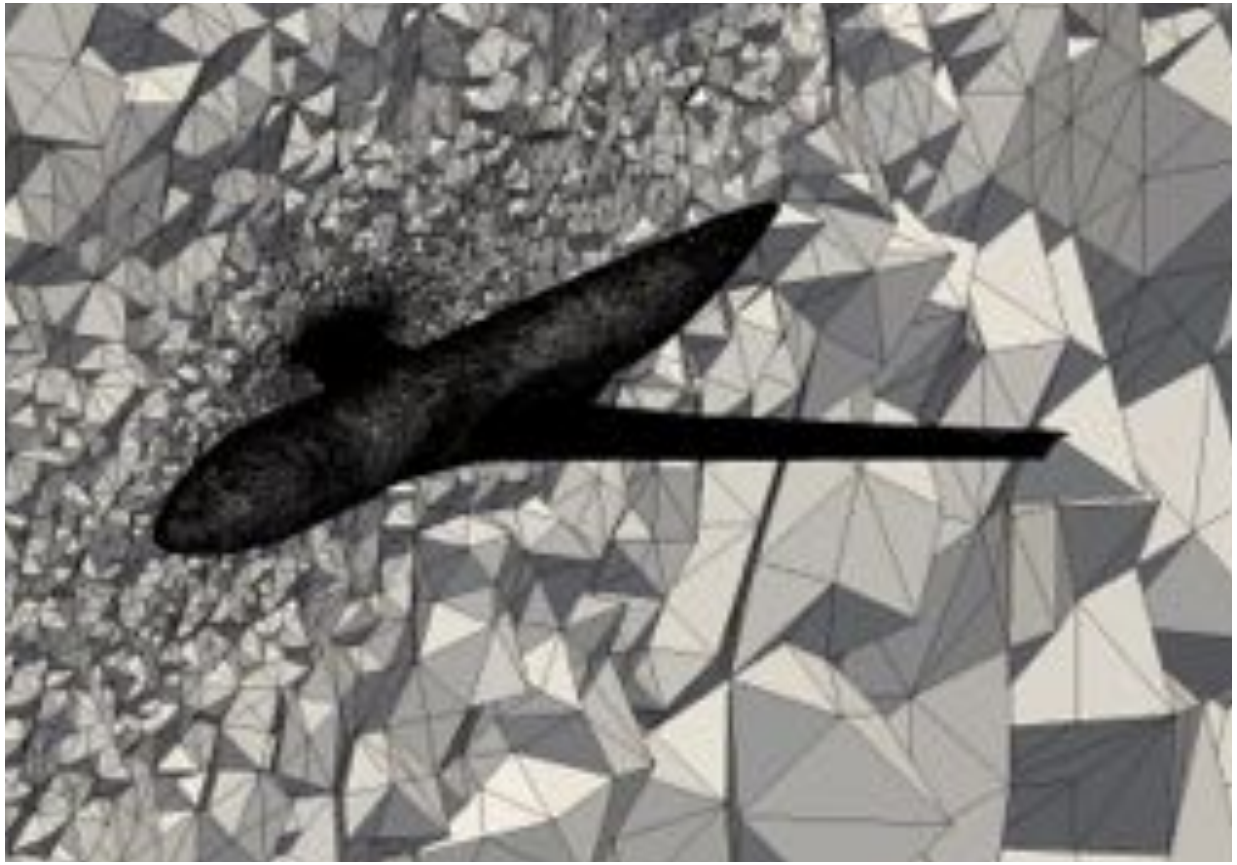
(a) $\alpha = 10.48°$

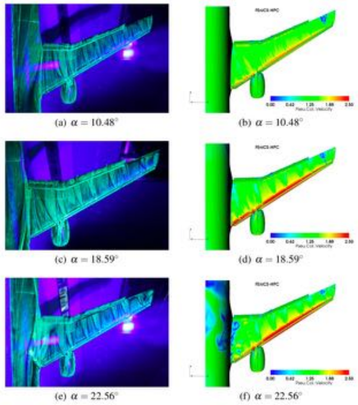
(b) $\alpha = 18.59°$

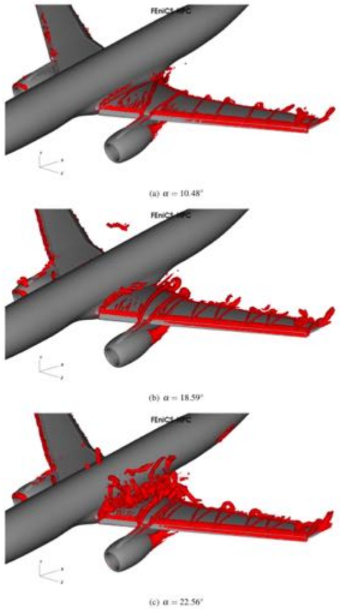
(c) $\alpha = 22.56°$

[Jansson et al., Springer, 2018]

Discretization by a mesh







[Jansson et al., Springer, 2018]

Credits

Album of fluid flow (Milton Van Dyke)

- https://en.wikipedia.org/wiki/An_Album_of_Fluid_Motion
- https://www.abebooks.com/9780915760022/Album-Fluid-Motion-Milton-Dyke-0915760029/plp