The finite element method for the Navier-Stokes equations

Lucas Payne

October 1, 2021

Contents

Chapter 1

Mechanics

> *Corpus omne perseverare in statu suo quiescendi vel movendi uniformiter in directum, nisi quatenus a viribus impressis cogitur statum suum mutare.*
>
> *Mutationem motus proportionalem esse vi motrici impressae, & fieri secundum lineam rectam qua vis illa imprimitur.*
>
> *Actioni contrariam semper & aequalem esse reactionem: sive corporum duorum actiones in se mutuo semper esse aequales & in partes contrarias dirigi.*
>
> Newton [1]

The calculus of variations.

Chapter 2

Continuum mechanics

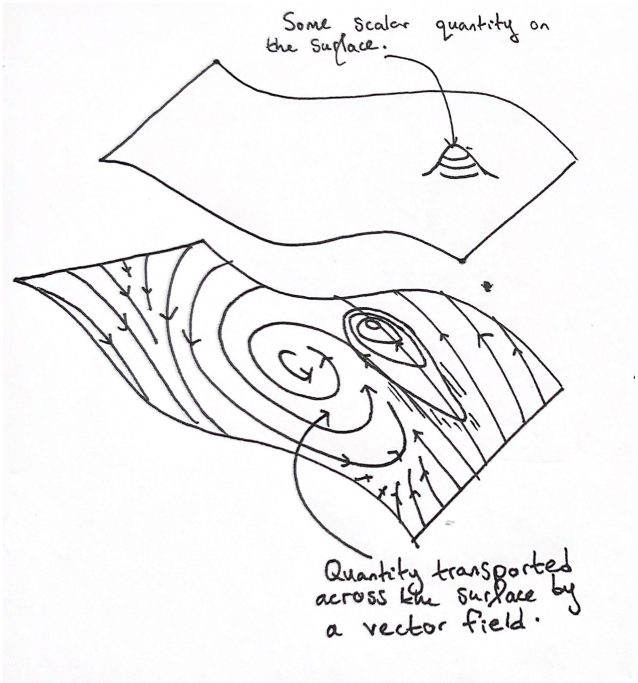
2.1 Introduction

— Introduction.

Although the focus later will be on fluid mechanics, a large set of fundamental concepts of solid and fluid mechanics are the same. The key distinction appears when one focuses on displacements and materials with "shear strength" (solids) versus flows and materials with no shear strength (common fluids).

2.2 Transport

Before considering continuum processes in the framework of Newtonian and Lagrangian mechanics, we will look at a fundamental notion of a "motion" of a point in a function space. Many continuum models in physics, such as the heat equation, Maxwell's equations, and the equations of fluid motion, are formed by *conservation equations*. These laws posit that the evolution of the state (represented by a function) is due to the transport of the quantity that the function measures, which is either pushed around (by some flux either predetermined or dependent on the current state), introduced into the system at sources, or leaves the system at sinks.



We consider here the transport of quantities (scalar, vector, and tensor) on a general finite-dimensional manifold M, colloquially called "the continuum". All transporting vector fields (or flux functions) are considered to be tangent to this manifold M.

2.2.1 Continuity equations and conservation laws

The integral form of a continuity equation

Consider some spatial quantity ϕ on M and a flux function j which by which this quantity flows around M. For clarity, we will begin by specializing ϕ to be a scalar, although later we will find it useful to transport vector quantities such as momentum. By definition we want this flux function to just push quantity around. The entering and exiting of quantity into and out of the system is determined by some arbitrary source function s (of the same kind as ϕ). These variables are related by the conservation condition

$$\frac{d}{dt}\int_{\Omega_0}\phi\,dx = \int_{\Omega_0} s\,dx + \int_{\partial\Omega_0}\phi j\cdot(-\hat{n})\,dx \tag{2.1}$$

for arbitrary control volumes Ω_0 in the continuum. The term $-\hat{n}$ denotes the inward-pointing normal to the boundary of the control volume. This simply says that the change in the total quantity in the fixed control volume is accounted for exactly by that quantity pushed through the boundary by the flux function j, and the internal sources and sinks of quantity s.

The differential form of a continuity equation

A common technique in continuum modelling is the use of Stokes' theorem to simplify integral expressions. Stokes' theorem and its specialisations (such as the divergence theorem and Green's theorem) are really *definitions* of pointwise quantities such as the divergence and curl as limits of these integral expressions for arbitrarily small regions.

$$\nabla\cdot v \coloneqq \lim_{\epsilon\to 0}\frac{1}{\partial\Omega_\epsilon}\int_{\partial\Omega_\epsilon} v\cdot\hat{n}\,dx. \tag{2.2}$$

$$\int_{\partial\Omega} v\cdot\hat{n}\,dx = \sum_{i\in\mathcal{I}}\int_{\partial\Omega_i} v\cdot\hat{n}\,dx. \tag{2.3}$$

$$\int_{\partial\Omega} v\cdot\hat{n}\,dx = \int_{\Omega}\left[\lim_{\epsilon\to 0}\frac{1}{\partial\Omega_{x,\epsilon}}\int_{\partial\Omega_{x,\epsilon}} v\cdot\hat{n}\,dx'\right]dx. \tag{2.4}$$

Equation (2.1) becomes

$$\frac{\partial\phi}{\partial t} = s - \nabla\cdot(\phi j) \tag{2.5}$$

assuming that ϕj is sufficiently differentiable such that the limiting integral exists. Continuity relations are most naturally expressed in form (2.1), while the form (2.5) may be more useful for techniques such as finite differences. Later, when we discuss numerical methods for solving continuum models, we will not take this route. The methods of interest, *Galerkin* methods, work naturally with the integral form (2.1). It will be seen later that some constructions in the presentation of Galerkin methods, such as the "weak form" of a PDE, simply undo the differentialisation (for example (2.5)) of the original integral form of physical PDEs (for example (2.1)).

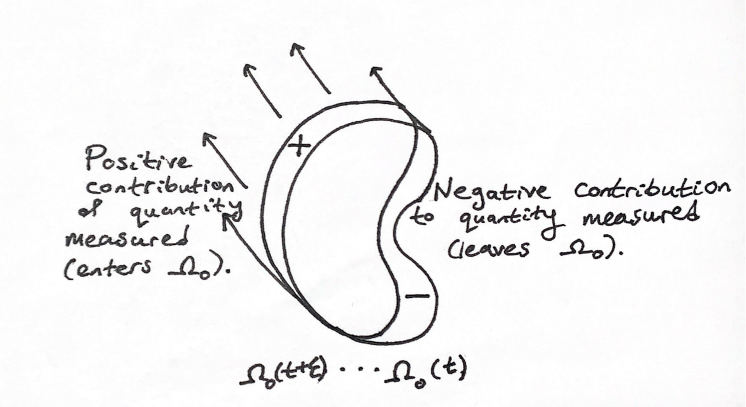
2.2.2 The Reynolds transport theorem

The integral form of Reynolds transport

With our integral formulation of a continuity relation (2.1), the control volume Ω_0 is fixed. We may change our perspective by considering, in addition to the flux function j (which transports quantity ϕ), another vector field $\hat{u}$ which will transport our control volume Ω_0. The rate of change of some time-dependent quantity γ in this *moving* control volume is expressed as

$$\frac{d}{dt}\int_{\Omega_0(t)} \gamma \, dx, \tag{2.6}$$

where $\Omega_0(t)$ implicitly denotes that Ω_0 is being transported under the flow of $\hat{u}$. Clearly, this rate of change of quantity γ is due to the motion of the control volume,



as well as internal changes of γ inside the (fixed) control volume. The formal expression of these contributions to the rate of change (2.6) is

$$\frac{d}{dt}\int_{\Omega_0(t)} \gamma \, dx = \int_{\Omega_0(t)} \frac{\partial \gamma}{\partial t} \, dx + \int_{\partial\Omega_0(t)} \gamma \hat{u} \cdot \hat{n} \, dx. \tag{2.7}$$

This result is called the *Reynolds transport theorem*, a generalisation of Feynman's popularised "differentiation under the integral sign" [10], otherwise named the Leibniz integral rule. See, for example, [8] for a formal derivation of (2.7).

The differential form of Reynolds transport

In the limit, with the routine application of Stokes' theorem, we can differentialise (2.7) to get

$$\frac{d}{dt}\int_{\Omega_0(t)} \gamma \, dx \quad \longrightarrow \quad \frac{\partial \gamma}{\partial t} + \nabla \cdot (\gamma \hat{u}), \tag{2.8}$$

as Ω_0 becomes small. The right-hand-side of (2.8) measures the change in volume of a quantity when a small control volume around the point of evaluation is moved, expanded or contracted by the flow field $\hat{u}$.

Reynolds transport applied to a continuity equation

Letting our quantity γ in (2.7) be the quantity ϕ transported by flux function j, described in continuity equation (2.1), we get a specialised form of the Reynolds transport theorem

for continuity equations. Term $\frac{\partial \gamma}{\partial t}$ in (2.7) becomes $\frac{\partial \phi}{\partial t}$ in the differential form of the continuity equation (2.5), giving

$$\begin{aligned} \frac{d}{dt}\int_{\Omega_0(t)} \phi \, dx &= \int_{\Omega_0(t)} -\nabla \cdot (\phi j) + s \, dx + \int_{\partial\Omega_0(t)} \phi \hat{u} \cdot \hat{n} \, dx \\ &= \int_{\Omega_0(t)} s \, dx + \int_{\partial\Omega_0(t)} \phi(\hat{u} - j) \cdot \hat{n} \, dx \end{aligned} \tag{2.9}$$

by Stokes' theorem. This has a clear interpretation. The $\hat{u} - j$ term is due to us wanting to measure the contributions to the total ϕ due to the moving boundary of Ω_0, where the motion that matters is *relative* to the flux of the quantity j. Specifically, if we move the control volume by the same flux function j (letting $\hat{u} = j$), we get

$$\frac{d}{dt}\int_{\Omega_0(t)} \phi \, dx = \int_{\Omega_0(0)} s \, dx. \tag{2.10}$$

In fact, (2.10) is just another form for the conservation law (2.1), where the "frame of reference" for measurement of ϕ follows the transport of ϕ. This simply means that as we follow some volume of quantity original situated in Ω_0, a conservation law posits that the only change detected is due to the source function s. In differential form (2.10) becomes

$$\frac{\partial \gamma}{\partial t} + \nabla \cdot (\gamma \hat{u}) = s, \tag{2.11}$$

a succint equivalent to (2.5). The idea of following the flow while making measurements is called the *Lagrangian* perspective, in contrast to the *Eulerian*, fixed, perspective.

2.2.3 Incompressible and compressible transport

Analogous to constraints on the motion of a finite mechanical system, we can constrain possible movement of our continuous quantity to *incompressible transport*. Much like how, in the framework of Lagrangian mechanics, constraints on motion are implicitly enforced by strong "virtual forces", constraining transport to be non-compressing will lead to the notion of *pressure*, derived in section 3.4.

Incompressibility

Incompressibility of control volumes gives a constraint on the form of a flux function j. We call this constrained flux function j non-compressing. By incompressibility we mean that a control volume being transported by j will have constant volume. While j may transport other quantities, we express incompressibility by requiring the flux function to transport a constant "volume quantity" with a corresponding null source function,

$$\phi_{\text{vol}} = 1, \quad s_{\text{vol}} = 0.$$

The corresponding conservation law, in differential form (2.5), is

$$\frac{\partial \phi_{\text{vol}}}{\partial t} = -\nabla \cdot (\phi_{\text{vol}} j) + s_{\text{vol}} \quad \Rightarrow \quad \nabla \cdot j = 0. \tag{2.12}$$

This is our non-compressing constraint on j, and has a clear interpretation, as there is a non-zero divergence of j if and only if there is an inward or outward flux which would contract or expand a transported control volume.

2.2.4 Transport of vector and tensor quantities

All previous discussion on the transport of scalar quantities applies trivially to vector and tensor quantities. This will soonest be of use in the discussion of conservation of linear momentum, a vector quantity However, some notational discussion is needed in order to establish differential forms of continuity equations and the Reynolds transport theorem.

Reynolds transport of vector and tensor quantities

For a general tensor quantity Γ, the integral form of Reynolds transport (2.7) is trivially

$$\frac{d}{dt}\int_{\Omega_0(t)} \Gamma \, dx \int_{\Omega_0(t)} \frac{\partial \Gamma}{\partial t} \, dx + \int_{\partial\Omega_0(t)} \Gamma \left(\hat{u} \cdot \hat{n}\right) \, dx. \tag{2.13}$$

The step to the differential form (2.8), however, needs some thought as rearranging

$$\text{“}\Gamma \left(\hat{u} \cdot \hat{n}\right) = \left(\Gamma \hat{u}\right) \cdot \hat{n}\text{”}$$

in order to apply the divergence theorem makes no sense. However, the divergence $\nabla\cdot$ was *defined* to evaluate the limit of this boundary integral for arbitrarily small Ω_0. We therefore have a natural generalisation of the divergence for arbitrary tensors Ψ, as the limit of the boundary integral of the *contraction* of Ψ with the outward normal $\hat{n}$ (which is a contravariant vector). The divergence of a rank n tensor is then a rank $n-1$ tensor,

$$\int_{\Omega_0} \nabla \cdot \Psi \, dx \coloneqq \int_{\partial\Omega_0} \Psi : \hat{n} \, dx. \tag{2.14}$$

We can then rewrite $\Gamma \left(\hat{u} \cdot \hat{n}\right)$ in (2.13) as

$$\Gamma \left(\hat{u} \cdot \hat{n}\right) = \left(\Gamma \otimes \hat{u}\right) : \hat{n},$$

where the tensor product $\otimes$ “defers contraction” of $\hat{u}$ with $\hat{n}$, by storing it as a component of product tensor $\Gamma \otimes \hat{u}$. This leads to a differentialisation of (2.13),

$$\frac{d_{\hat{u}}\Gamma}{d_{\hat{u}}t} = \frac{\partial \Gamma}{\partial t} + \nabla \cdot \left(\Gamma \otimes \hat{u}\right). \tag{2.15}$$

Conservation equations for vector and tensor quantities

With the previous ideas from tensor algebra, it will be easy to describe continuity relations for transport of tensors. The integral form of the scalar continuity equation (2.1), generalised to transported tensor Φ, trivially becomes

$$\frac{d}{dt}\int_{\Omega_0} \Phi \, dx = \int_{\partial\Omega_0} \Phi \left(j \cdot \left(-\hat{n}\right)\right) \, dx + \int_{\Omega_0} s \, dx. \tag{2.16}$$

By the same tensor algebra as above we have

$$\Phi \left(j \cdot \left(-\hat{n}\right)\right) = -\left(\Phi \otimes j\right) : \hat{n},$$

giving (2.16) in differential form as

$$\frac{\partial \Phi}{\partial t} = -\nabla \cdot \left(\Phi \otimes j\right) + s. \tag{2.17}$$

2.3 The kinematics of the continuum

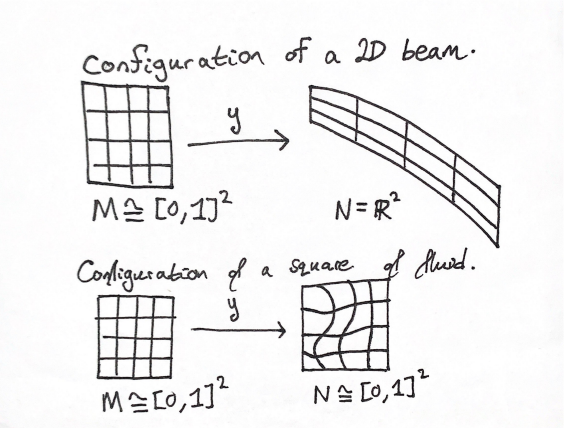
Transport equations are just one notion of "physical motion" in a continuum model. These transport equations, with prescribed flux and source functions, determine a continuous process on a fixed domain M. These conserved quantities are time-varying maps from M to some measurement space of scalars or tensors. Each map is a component of the total configuration space C, which clearly must be infinite-dimensional. We now consider another component of C which will let us model a physical domain with alterable shape. In our discussion we will consider a fixed time interval $[t_1, t_2]$ in which our physical motions will take place.

2.3.1 Position maps

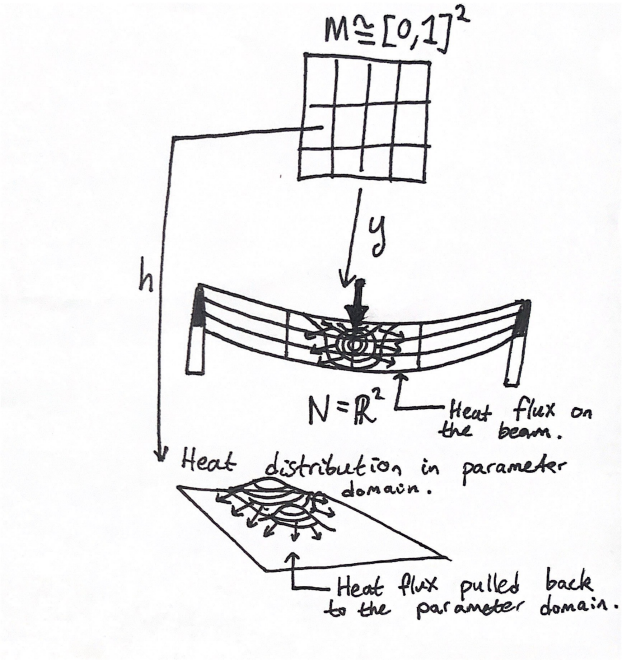
We may consider the manifold M as the parametric domain of some points living in an ambient manifold N. We will call this the "position map"

$$y : M \times [t_1, t_2] \to N.$$

In general, y needs not be continuous, differentiable, or invertible. These restrictions are only introduced in accord to the physical meaning of the position map. For example, models of small beam deflections may require continuity, and invertibility to prevent self-intersections.



As an example, suppose we are modelling the heat distribution of a 2D beam supporting a point load which is also a heat source. We could model the beam geometry as a smooth invertible map $y : [0, 1]^2 \to \mathbb{R}^2$, letting $M = [0, 1]^2$ and $N = \mathbb{R}^2$. The heat distribution on the beam could be represented by a function $h : M \to \mathbb{R}$, and a heat flux function j could be pulled back to M from N.

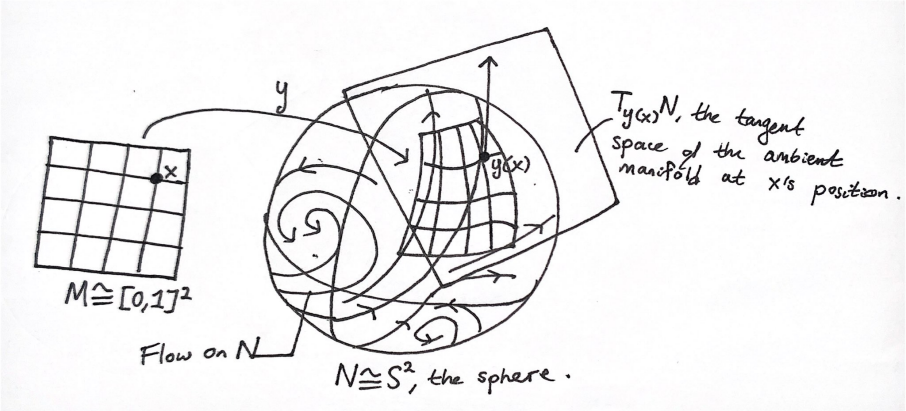


Although this model is so far hopelessly incomplete, we can see that position maps and transport equations are fundamental tools used for modelling the *geometry* of a problem.

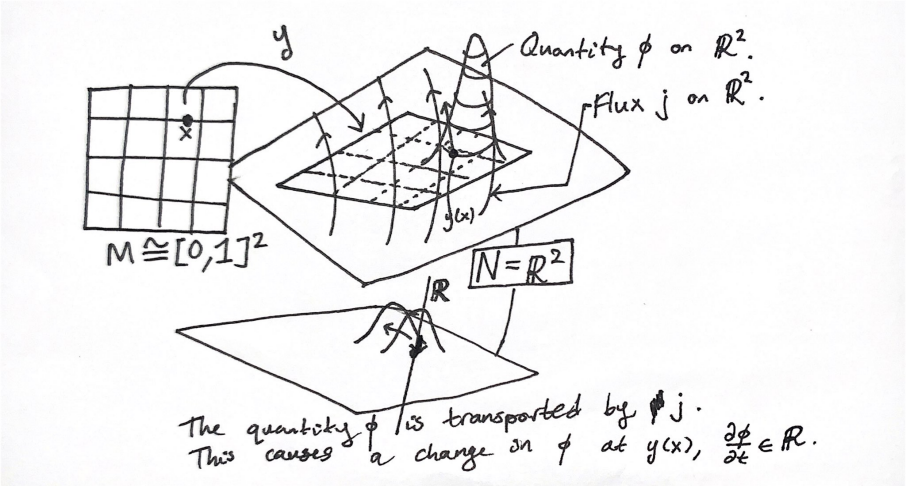
2.3.2 Velocity

As in the mechanics of a particle, each component of our state $q \in C$ will have a corresponding velocity which "generates" a physical motion of that component. In the case of the position map $y : M \times [t_1, t_2] \to N$, the velocity will be given by a vector in the tangent space of N at $y(x)$ for each parameter $x \in M$. This vector field is denoted $\dot{y}$. This is the *Lagrangian* description of motion. We will find it useful to instead use the *Eulerian* description, where we measure the velocity of the position map in the position domain N. Formally we denote this Eulerian velocity by the letter u, ubiquitous in fluid mechanics, and let

$$u(y,t) \coloneqq \dot{y}(y^{-1}(y,t),t) \quad \text{for all valid } y \in N.$$



For some transported scalar quantity $\phi : M \times [t_1, t_2] \to \mathbb{R}$, the tangent space at each point of $\mathbb{R}$ is $\mathbb{R}$, and therefore our velocity is represented by a scalar function $\frac{\partial \phi}{\partial t}$ giving local change in time of $\phi(x)$ for each $x \in M$.



We may denote our total velocities as a state variable $\dot{q}$. When we have state q, the corresponding velocity $\dot{q}$ will be in the tangent space of C at q, denoted T_qC. We can then define the space of velocities as the *tangent bundle* of the configuration space,

$$TC = \bigcup_{q \in C} T_qC.$$

2.3.3 The deformation and velocity gradients

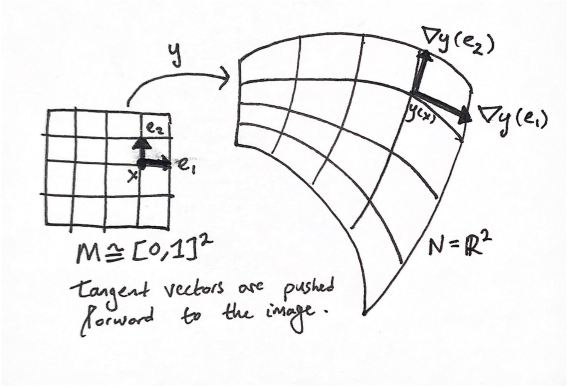
We may think of the mechanics of a particle as a special case of our continuum model, where M is a single point. In this case we have only one $x \in M$, so we cannot vary x. However, for a continuum parameter domain M we can take derivatives with respect to our parameter as well as time. We can extract important geometric/kinematic information from the spatial derivatives of our position map y.

The deformation gradient

The gradient of the position map y with respect to parameter $x \in M$ is called the *deformation gradient*

$$\nabla y. \tag{2.18}$$

The deformation gradient is equivalent to the *Jacobian matrix*, used to compute the *pushforward* of tangent vectors under the displacement map.



The determinant of the Jacobian matrix is usually denoted $J = \det(\nabla y)$, and is called the Jacobian.

The velocity gradient

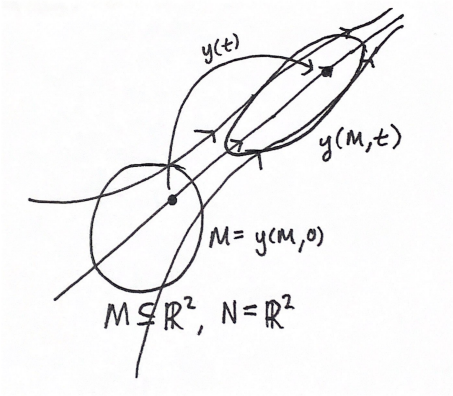
Letting u be our Eulerian velocity as defined above, we may express the position map through an ODE,

$$\frac{d}{dt}y(x,t) = u(y(x,t),t), \quad y(x,0) = y_0(x). \tag{2.19}$$

It is common, especially in flow problems, to let M be a subset of N which is the "initial geometry". In this case we could let the initial position map be the identity map

$$y_0(x) = x \in M \subset N.$$

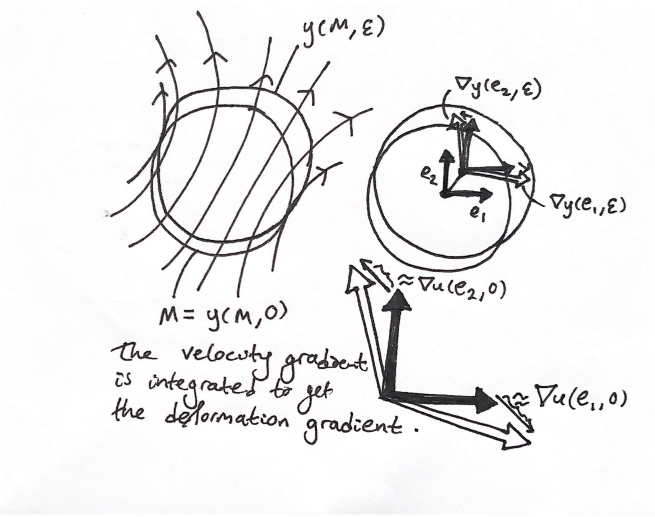
For example, given an initial disc in $\mathbb{R}^2$, we may prescribe a constant Eulerian velocity field u and see how the original disc is "mixed".



We can see that, in this case, it is meaningful to take spatial gradients in (2.19) to derive an ODE for the deformation gradient ∇y:

$$\nabla\left[\frac{d}{dt}y(x,t)\right] = \nabla u(y(x,t),t), \quad \nabla y(x,0) = I, \tag{2.20}$$

where I is the identity tensor. This is easily visualised.



We can see that the term ∇u is a "differential generator" of the deformation gradient. ∇u is called the *velocity gradient*.

2.3.4 Material points and material derivatives

The material derivative

Assuming an non-compressing flux function u which transports quantity ϕ, the differential form of the Reynolds transport theorem (2.11) becomes

$$\frac{\partial \phi}{\partial t} + \nabla \cdot (\phi u) = \frac{\partial \phi}{\partial t} + u \cdot \nabla \phi + \cancel{\phi \nabla \cdot u}. \tag{2.21}$$

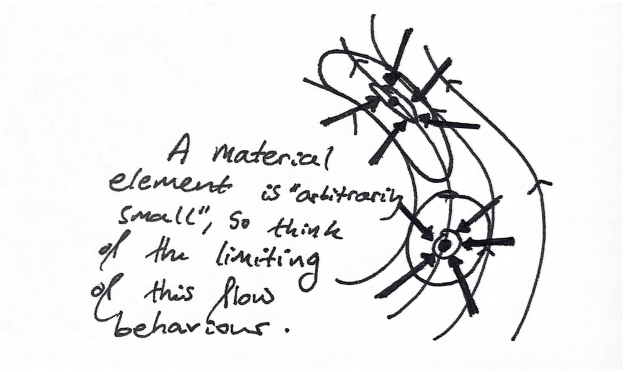
We define the *material derivative* to be

$$\frac{D}{Dt} \coloneqq \frac{\partial}{\partial t} + u \cdot \nabla. \tag{2.22}$$

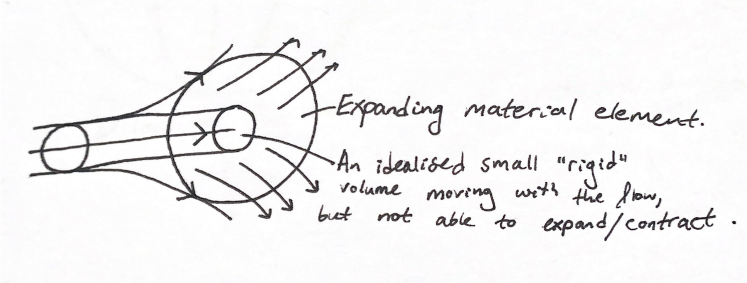
It is a convention to leave the vector field u implicit, as material derivatives are usually taken with respect to the velocity field. This derivative (2.22) will turn out to measure "per-volume" quantities.

Pieces of the continuum

A *material element* is a small piece of a continuum model which evolves with the displacement and flow. In fluid dynamics this is also called a fluid parcel, although it should not be thought of as an object placed in the fluid, but rather some tracer such as a non-diffusing dye.



In continuum mechanics a *material point* is the ideal point of the continuum, corresponding to some macroscopic averaging in physical models. At a certain point, we can imagine an arbitrarily small fluid parcel being transported by the flow. No matter how small this is, the parcel will still start to shear, expand, and contract due to the flow. The material derivative was defined as $\frac{D}{Dt} \coloneqq \frac{\partial}{\partial t} + u \cdot \nabla$. Notably this derivative contains no divergence term, and so does not measure the change of a quantity due to expansion and contraction of the arbitrarily small fluid parcel. Therefore we can think of the material derivative as acting on *per-volume* quantities.



2.4 The dynamics of the continuum

2.4.1 Conservation of mass

We will take for granted that there is some initial mass density function $\rho_0 : M \to \mathbb{R}^+$ which we would like to conserve.

The Lagrangian form of mass conservation

As the position map y evolves, for example under the action of Eulerian velocity field u in the ODE (2.19), we would like a mass density function $\rho : N \to \mathbb{R}^+$ to be greater if the position map "compresses" the material, and smaller if it "stretches" the material. Our aim is that the total mass measured in a control volume of N is the same as in its initial configuration. We can express this by the integral equation

$$\int_{\Omega_0} \rho_0(x)\, dx = \int_{y(\Omega_0,t)} \rho(y,t)\, dy, \tag{2.23}$$

where $y(\Omega_0, t)$ denotes the domain pushed forward by the position map. By the usual change of variables formula we have

$$\int_{\Omega_0} \rho_0(x)\, dx = \int_{\Omega_0} \det(\nabla y)\rho(y(x,t),t)\, dx, \tag{2.24}$$

where $J = \det(\nabla y)$ is the Jacobian, measuring the local change in the volume element due to the change of variables.

(draw this)

As (2.24) must hold identically for all Ω_0, we have the localised conservation law

$$\rho_0 = \det(\nabla y)\rho. \tag{2.25}$$

The Eulerian form of mass conservation

Conservation law (2.25) is simple. However, it is given in terms of absolute deformation gradient ∇y. Instead of asking how mass is distributed in comparison to its initial configuration, we can ask how mass is transported by u. This gives an instance of the continuity equation (2.1), where we have no source:

$$\frac{d}{dt}\int_{\Omega_0} \rho\, dx + \int_{\partial\Omega_0} \rho u \cdot \hat{n}\, dx = 0. \tag{2.26}$$

With application of Stokes' theorem we have

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho u) = 0. \tag{2.27}$$

Following [8], there is a more geometrical statement of this equation, which will be useful when we discuss "material points" By a divergence product rule (2.27) is equivalent to

$$\frac{\partial \rho}{\partial t} + u \cdot \nabla\rho + \rho\nabla \cdot u = 0 \quad \Rightarrow \quad \frac{D\rho}{Dt} = -\rho\nabla \cdot u, \tag{2.28}$$

where $\frac{D}{Dt}$ is the material derivative as defined in (2.22).

2.4.2 Conservation of linear momentum

If we conserve the linear momentum ρu, a "quantity of motion", under the flow of u, then we get a continuity equation

$$\frac{d}{dt}\int_{\Omega_0(t)} \rho u \, dx = \int_{\Omega_0(t)} \rho g \, dx + \int_{\partial\Omega_0(t)} \hat{t} \, dx, \tag{2.29}$$

a specific realisation of the Lagrangian continuity equation (2.10). The Lagrangian perspective is convenient as it allows us to separate interior and boundary forces on a moving piece of material. The term g is a regular body force per unit mass, where ρg corresponds to the source term s in (2.10). The boundary term involving $\hat{t}$, however, has no analogue in the scalar continuity equation (2.10). This vector term $\hat{t}$ is called the *traction* in continuum mechanics, and measures a local force exerted across the boundary of the control volume due to the immediately adjacent material.

The Euler-Cauchy stress principle

Clearly, in accord with Newton, we would like that two Ω_0 and Ω_0' which share a boundary element should have equal and opposite tractions across that boundary element. Since the normal $\hat{n}$ represents a boundary element, and is negative for the opposite element, if $\hat{t}$ is linear in $\hat{n}$ we have this required property. We can then let (2.29) become

$$\frac{d}{dt}\int_{\Omega_0(t)} \rho u \, dx = \int_{\Omega_0(t)} \rho g \, dx + \int_{\partial\Omega_0(t)} \sigma : \hat{n} \, dx \tag{2.30}$$

where σ is termed the *Cauchy stress tensor*. This is the integral form of the *Cauchy momentum equation*, the standard form of $F = ma$ in continuum mechanics.

Differentializing the Cauchy momentum equation

By application of the Reynolds transport theorem (2.13) to (2.30) we get

$$\int_{\Omega_0(0)} \frac{\partial(\rho u)}{\partial t} \, dx + \int_{\partial\Omega_0(0)} \rho u (u \cdot \hat{n}) \, dx = \int_{\Omega_0(0)} \rho g \, dx + \int_{\partial\Omega_0(0)} \sigma : \hat{n} \, dx. \tag{2.31}$$

Differentializing (2.31), by our previously derived tensor identities, gives

$$\frac{\partial(\rho u)}{\partial t} + \nabla \cdot (\rho u \otimes u) = \rho g + \nabla \cdot \sigma. \tag{2.32}$$

This is called the *conservative form* of the Cauchy momentum equation. We can derive another, possibly more convenient form of (2.32) using the fact that ρ is conserved and has no source. Here, this will be derived purely algebraically, although the final form of the equation has a useful interpretation. Expanding the partial derivative

$$\frac{\partial(\rho u)}{\partial t} = \rho \frac{\partial u}{\partial t} + u \frac{\partial \rho}{\partial t}$$

is simple. The tensor divergence $\nabla \cdot (\rho u \otimes u)$ is defined such that

$$\int_{\Omega_0} \nabla \cdot (\rho u \otimes u) \, dx = \int_{\partial\Omega_0} (\rho u \otimes u) : \hat{n} \, dx = \int_{\partial\Omega_0} \rho u (u \cdot \hat{n}) \, dx$$

for arbitrary control volumes Ω_0. As Ω_0 becomes small, we can separately assume u and ρu are constant to derive

$$\int_{\partial\Omega_0} \rho u (u \cdot \hat{n}) \, dx = u \int_{\partial\Omega_0} (\rho u) \cdot \hat{n} \, dx + \rho u \cdot \int_{\partial\Omega_0} u \hat{n} \, dx \quad + \quad \cdots$$

where a trailing term becomes neglible for a small control volume. This gives a "tensor product rule" for the divergence,

$$\nabla \cdot (\rho u \otimes u) = u \nabla \cdot (\rho u) + \rho u \cdot \nabla u. \tag{2.33}$$

Equation (2.32) then becomes

$$\rho \frac{\partial u}{\partial t} + u \frac{\partial \rho}{\partial t} + u \nabla \cdot (\rho u) + \rho u \cdot \nabla u = \rho g + \nabla \cdot \sigma.$$

Noting that $\frac{\partial \rho}{\partial t}$ is already given by continuity equation (2.27)

$$\frac{\partial \rho}{\partial t} = -\nabla \cdot (\rho u),$$

as mass is transported by u and has no source, we get

$$\rho \frac{\partial u}{\partial t} - \cancel{u \nabla \cdot (\rho u)} + \cancel{u \nabla \cdot (\rho u)} + \rho u \cdot \nabla u = \rho g + \nabla \cdot \sigma.$$

Finally, the material derivative as defined in section (ref) is helpful in simplifying the above to

$$\rho \frac{Du}{Dt} = \rho g + \nabla \cdot \sigma. \tag{2.34}$$

This form of (2.32) is called the *convective form* of the Cauchy momentum equation, and is more obviously a form of $F = ma$. Recall that the material derivative is defined as

$$\frac{D}{Dt} \coloneqq \frac{\partial}{\partial t} + u \cdot \nabla,$$

which measures the rate of change of a pointwise quantity from the perspective of a particle moving with the flow field u. The equation (2.34) then says that, if the continuum consists of idealised points each with a certain linear momentum (in the particle sense), deflection of their inertial path is due only to the application of a body force ρg at this point, and a total traction force exerted by the surrounding material.

2.4.3 Constitutive relations

The conservation equations derived are conservation of mass (2.27) and conservation of linear momentum (2.34). Here the equations are given in their typical differential form for flow problems, with the Eulerian perspective of mass conservation, and the convective form of linear momentum conservation:

$$\begin{aligned} \frac{D\rho}{Dt} + \rho \nabla \cdot u = 0 \quad &\text{(Conservation of mass)}, \\ \rho \frac{Du}{Dt} = \rho g + \nabla \cdot \sigma \quad &\text{(Conservation of linear momentum)}. \end{aligned} \tag{2.35}$$

The unknowns include mass density ρ and velocity u, described by $1 + d$ scalar functions where d is the dimension of the domain. We can see there are $1 + d$ equations by expanding u in terms of components in some basis.

$$\begin{aligned} \rho \frac{Du_i}{Dt} = \rho g_i + (\nabla \cdot \sigma)_i \quad &\text{(Conservation of linear momentum components)}, \\ i = 1, \cdots, d. \end{aligned}$$

When g and σ are known, this system is well-formed, but not very interesting. This effectively models a continuum of non-interacting material points, with a linear-momentum-introducing source function $\rho g + \nabla \cdot \sigma$. However, we usually let g be known (as this is a per-mass external body force, for example gravity), and the Cauchy stress tensor σ be unknown. This gives d^2 more unknowns, so the system is heavily underdetermined. In effect, the system (2.35) is underdetermined because we don't have a specific material in mind.

Constitutive relations

A specification of the Cauchy stress tensor σ is called a *constitutive relation*, as σ depends on the "material constitution". A constitutive relation specifies how the material configuration induces forces on the material, or rather, how kinematics is related to dynamics. With σ specified, the Cauchy momentum equation (2.34) becomes well-posed (however, in general, σ may depend on new variables such as temperature, requiring further equations).

Completing the equations

We have so far been working quite generally with the forms of continuum mechanics models and their constraints. Although we have made some assumptions (for example, we require σ to be a purely local function, an idealisation due to Cauchy [ref]), we must so far leave our systems underdetermined. In the next chapter, we will investigate an important constitutive relation for the stress σ, forming what is called a *Navier-Stokes fluid*, giving a complete system of equations called the *Navier-Stokes equations*.

Chapter 3

The Navier-Stokes equations

3.1 Introduction

The incompressible Navier-Stokes equations model the motion of a common kind of viscous fluid called a *Newtonian fluid*. They are:

- The Cauchy momentum equation (2.34) for constant mass density ρ and velocity u,
- an incompressibility constraint $\nabla \cdot u = 0$ and unknown pressure p,
- and a concrete constitutive relation for the deviatoric stress τ.

In anticipation, their common form is

$$\rho \frac{Du}{Dt} = -\nabla p + \nabla \cdot \tau + \rho g, \quad \nabla \cdot u = 0, \tag{3.1}$$

where τ is defined by Stokes' constitutive relation

$$\tau = \mu \left(\nabla u + \nabla u^T \right), \tag{3.2}$$

where μ is called the *viscosity*, and ∇u is the velocity gradient defined in section 2.3.3, measuring the local deformation of a small control volume under the flow of u. We will assume their domain is a subset of $\mathbb{R}^d$, where typically $d = 2$ or 3, although the Navier-Stokes equations can be solved in curved domains (see [21]). Alongside a domain and appropriate initial and boundary conditions, the Navier-Stokes equations (3.1) form a concrete flow problem which can be solved numerically, or in special situations analytically.

3.2 The equations of fluid motion

We begin with the standard conservation equations of mass and linear momentum, (2.35):

$$\begin{aligned} \frac{D\rho}{Dt} + \rho \nabla \cdot u = 0 \quad &\text{(Conservation of mass)}, \\ \rho \frac{Du}{Dt} = \rho g + \nabla \cdot \sigma \quad &\text{(Conservation of linear momentum)}. \end{aligned} \tag{3.3}$$

As discussed in section 2.4.3, we need to specify the Cauchy stress tensor σ such that the system is well-formed. It would be helpful to restrict the possible form of σ. In the next section we will show that σ's antisymmetric part describes a couple force, a force which induces an angular momentum (a "spin") in a small control volume, but which doesn't contribute to linear momentum. Therefore, if we do not want couple forces, we want σ to be symmetric.

3.2.1 Conservation of angular momentum

Angular momentum is traditionally presented in terms of rigid bodies, bodies subject to a distance-preserving constraint between material points. It is the moment of linear momentum.

The discussion in section 2.3.4 indicates that we can think of a very small rigid body at a material point, subject to the flow. This will be subject to a "spin force".

Let the material point be $c \in \mathbb{R}^d$, which will act as a "centre of mass', and let $c \in \Omega_0$. Define $\bar{x} \coloneqq x - c$. Define the moment of linear momentum as

$$\int_{\Omega_0} \bar{x} \wedge (\rho u)\, dx. \tag{3.4}$$

The symbol $\wedge$ indicates the cross product, whose value should be thought of as a pseudo-vector or "plane with magnitude". We will call this the angular momentum of the control volume Ω_0. We repeat here an integral form of linear momentum conservation, which already must hold:

$$\int_{\Omega_0} \frac{\partial(\rho u)}{\partial t}\, dx + \int_{\partial\Omega_0} \rho u (u \cdot \hat{n})\, dx = \int_{\Omega_0} \rho g\, dx + \int_{\partial\Omega_0} \sigma \hat{n}\, dx. \tag{3.5}$$

It is simple to derive an angular momentum conservation equation, just by taking moments of each vector quantity:

$$\int_{\Omega_0} \bar{x} \wedge \frac{\partial(\rho u)}{\partial t}\, dx + \int_{\partial\Omega_0} \bar{x} \wedge (\rho u)\,(u \cdot \hat{n})\, dx = \int_{\Omega_0} \bar{x} \wedge (\rho g)\, dx + \int_{\partial\Omega_0} \bar{x} \wedge (\sigma \hat{n})\, dx. \tag{3.6}$$

(This precludes the introduction of surface and body couples which induce no linear momentum but do induce angular momentum [8]. We ignore these torques.)
If (3.5) holds, should (3.6) hold automatically?

By Stokes' theorem, the final term in (3.5) can be written as

$$\int_{\partial\Omega_0} \sigma \hat{n}\, dx = \int_{\Omega_0} \nabla \cdot \sigma\, dx.$$

(draw the differentialization happening at each point, contracting a small control volume).

With the help of the above picture, we can try to derive a per-point form for the final term in (3.6),

$$\int_{\partial\Omega_0} \bar{x} \wedge (\sigma \hat{n})\, dx = \int_{\Omega_0} \cdots ? \cdots\, dx.$$

At a point c in Ω_0, we contract an even smaller control volume Ω_c around that point, in order to express the boundary integral over $\partial\Omega_0$ in terms of smaller boundary integrals on the interior. As this takes a limit, we can separately assume that $\bar{x} = x - c$ and σ are constant in Ω_c, giving the "product rule"

$$\int_{\partial\Omega_c} \bar{x} \wedge (\sigma \hat{n})\, dx \quad \to \quad \bar{x} \wedge \nabla \cdot \sigma + (\text{some term keeping } \sigma \text{ constant}).$$

We can reason geometrically to find the final term. We can split σ into its symmetric part S and antisymmetric part N:

$$\sigma = \frac{1}{2}\left(\sigma + \sigma^T\right) + \frac{1}{2}\left(\sigma - \sigma^T\right) = S + N.$$

Antisymmetric matrices have a lot to do with rotations: A special property of antisymmetric matrices is

$$\langle x, Nx\rangle = \left\langle x, N^T x\right\rangle = \langle x, -Nx\rangle \quad \Rightarrow \quad \langle x, Nx\rangle = 0.$$

In fact the antisymmetric matrices are exactly those which generate rotations (formally, $\exp(N)$ is orthogonal, where exp is the matrix exponential). If σ is kept constant over $\partial\Omega_c$, we can visualise the tractions contributed by the symmetric and antisymmetric parts of σ:

(draw this)

By the above diagram we can conclude that, letting $\sigma = S + N$ be constant,

$$\int_{\partial\Omega_c} \bar{x} \wedge ((S+N)\,\hat{n})\; dx = \int_{\partial\Omega_c} \bar{x} \wedge (N\hat{n})\; dx \quad \rightarrow \quad \widehat{N}$$

where $\widehat{N}$ is defined in $\mathbb{R}^3$ as the axis-angle vector representation of the differential rotation corresponding to N:

$$\begin{gathered} Nv = \widehat{N} \wedge v, \quad v \in \mathbb{R}^3 \\ N = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \Rightarrow \widehat{N} = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}. \end{gathered} \tag{3.7}$$

We now have

$$\int_{\partial\Omega_c} \bar{x} \wedge (\sigma\hat{n})\; dx \quad \rightarrow \quad \bar{x} \wedge \nabla\cdot\sigma + \widehat{N}, \tag{3.8}$$

which gives

$$\int_{\partial\Omega_0} \bar{x} \wedge (\sigma\hat{n})\; dx = \int_{\Omega_0} \bar{x} \wedge \nabla\cdot\sigma + \widehat{N}\, dx. \tag{3.9}$$

This shows that for the tractions measured across the boundary, although their contributions to the linear momentum of the control volume conserve it, there is another contribution to the angular momentum, which is the "spin part" of σ. To show this more directly, localise the linear conservation equation (3.5):

$$\int_{\Omega_0} \frac{\partial(\rho u)}{\partial t} + \nabla\cdot(\rho u \otimes u) - \rho g - \nabla\cdot\sigma\, dx = 0. \tag{3.10}$$

The corresponding differential form of (3.6) is

$$\int_{\Omega_0} \bar{x} \wedge \left[\frac{\partial(\rho u)}{\partial t} + \nabla\cdot(\rho u \otimes u) - \rho g - \nabla\cdot\sigma\right] dx = \int_{\Omega_0} \widehat{N}\, dx. \tag{3.11}$$

As the conservation law (3.10) must hold for all Ω_0, we can see that for (3.10) to imply (3.11) (for linear momentum conservation to imply angular momentum conservation) we need

$$\widehat{N} = 0 \quad \Rightarrow \quad \sigma = \sigma^T. \tag{3.12}$$

So, without the introduction of any explicit angular momentum sources (body and surface couples), the Cauchy stress tensor σ is required to be symmetric, reducing the d^2 unknowns to $d(d+1)/2$ unknowns.

(May be $\widehat{N}/2$, maybe a sign error. Check Leal p68.)

3.2.2 Conservation of energy

$$\begin{aligned} \frac{D\rho}{Dt} + \rho\nabla\cdot u = 0 \quad &\text{(Conservation of mass)},\\ \rho\frac{Du}{Dt} = \rho g + \nabla\cdot\sigma \quad &\text{(Conservation of linear momentum)},\\ \sigma = \sigma^T \quad &\text{(Conservation of angular momentum)}. \end{aligned}$$

3.3 Scaling and dimension

3.3.1 The Reynolds number

3.4 Stokes flow and the meaning of pressure

If we assume that the advective term $u\cdot\nabla u$ in the incompressible Navier-Stokes equations is "small", we can ignore it and derive the linear *unsteady Stokes equations*:

$$\rho\frac{\partial u}{\partial t} = \mu\Delta u + \rho g - \nabla p, \quad \nabla\cdot u = 0. \tag{3.13}$$

We are assuming validity for low Reynolds number $Re \ll 1$, where convective behaviour is neglible compared to the viscous forces, which for a Navier-Stokes fluid "diffuse" the linear momentum. Setting the left-hand-side of (3.13) to zero results in the *steady Stokes equations*

$$\mu\Delta u + \rho g - \nabla p = 0, \quad \nabla\cdot u = 0. \tag{3.14}$$

Time-dependent equation (3.13) can be thought of as a "gradient descent" to find the steady Stokes flow (3.14). The steady Stokes equation is a constrained vector Poisson equation, where we have introduced pressure p explicitly. It is well-known, by Dirichlet's principle, that we can think of a weak solution to the unconstrained vector Poisson equation as a minimiser of the Dirichlet energy,

$$\underset{u}{\text{minimize}} \quad E(u) = \frac{\mu}{2}\left\langle \nabla u, \nabla u\right\rangle - \left\langle u, \rho g\right\rangle. \tag{3.15}$$

We can validate this by computing the Euler-Lagrange equations:

$$\frac{\delta E}{\delta u} = \frac{\partial \mathcal{L}}{\partial u} - \frac{d}{dx}\frac{\partial \mathcal{L}}{\partial u_x} = -\rho g - \mu\Delta u = 0.$$

We now introduce the incompressibility constraint $\nabla\cdot u = 0$, giving the constrained minimization

$$\begin{aligned} \underset{u}{\text{minimize}} \quad & E(u) = \frac{\mu}{2}\left\langle \nabla u, \nabla u\right\rangle - \left\langle u, \rho g\right\rangle\\ \text{subject to} \quad & \nabla\cdot u = 0. \end{aligned} \tag{3.16}$$

It is not immediately obvious how to form the constrained Euler-Lagrange equations here, as $\nabla\cdot$ is a differential operator. We cannot just write

$$\text{“}\frac{\delta E}{\delta u} = \lambda\nabla\,\text{.”}$$

for scalar function λ, as we can with a pointwise linear constraint such as $u\cdot v = 0$ for some vector field v. However, this is just a problem of notation. The evaluation of energy change with perturbations is defined as

$$\left\langle \frac{\delta E}{\delta u}, \delta u\right\rangle = \int_\Omega \frac{\delta E}{\delta u}\cdot\delta u\, dx.$$

We want this measure of energy change to be purely a divergence measure, up to a scalar multiplier λ:

$$\int_\Omega \frac{\delta E}{\delta u} \cdot \delta u \, dx = \int_\Omega \lambda \nabla \cdot \delta u \, dx. \tag{3.17}$$

This means that virtual displacements with $\nabla \cdot \delta u = 0$ will not cause an energy change, which is the condition that we want for a stationary point. We can now apply integration by parts to (3.17), assuming that δu vanishes on the boundary of the domain, to get

$$\int_\Omega \frac{\delta E}{\delta u} \cdot \delta u \, dx = -\int_\Omega \nabla \lambda \cdot \delta u \, dx. \tag{3.18}$$

We can now reasonably apply the localisation step to get the constrained Euler-Lagrange equations

$$\frac{\delta E}{\delta u} = -\nabla \lambda \quad \equiv \quad \mu \Delta u + \rho g - \nabla \lambda = 0. \tag{3.19}$$

Along with the constraint $\nabla \cdot u = 0$, this is just the steady Stokes equations (3.14), where $\lambda = p$! We can see that the pressure p is actually a Lagrange multiplier, which measures a virtual force that responds to virtual displacements which would break the constraint of incompressibility. In fact, we may think of this as a derivation of the pressure.

Alternative direct derivation in terms of a modified energy

Previously, we emphasized the meaning of the Lagrange multiplier. One utility of Lagrange's methods is their automated calculational power. It is standard to express that a solution to the optimization problem (3.16), with a differentiable equality constraint, is a stationary point of the modified energy

$$L(u, \lambda) \coloneqq \frac{\mu}{2} \langle \nabla u, \nabla u \rangle - \langle u, \rho g \rangle - \langle \lambda, \nabla \cdot u \rangle . \tag{3.20}$$

We can take an evaluated first variation with respect to u to get

$$\left\langle \frac{\delta L}{\delta u}, \delta u \right\rangle = \langle -\rho g - \mu \Delta u, \delta u \rangle - \langle \lambda, \nabla \cdot \delta u \rangle ,$$

which by integration by parts becomes

$$\left\langle \frac{\delta L}{\delta u}, \delta u \right\rangle = \langle -\rho g - \mu \Delta u + \nabla \lambda, \delta u \rangle . \tag{3.21}$$

We then get

$$\begin{aligned} \frac{\delta L}{\delta u} &= -\rho g - \mu \Delta u + \nabla \lambda = 0, \\ \frac{\delta L}{\delta \lambda} &= -\nabla \cdot u = 0, \end{aligned} \tag{3.22}$$

which are the steady Stokes equations (3.14) with pressure $p = \lambda$.

3.4.1 Application to hydrostatics

For example, we may imagine the steady Stokes equations modelling a calm sea with a flat seabed. We can let the body force be gravity described by a potential ϕ:

$$\rho g = -\nabla \phi.$$

If we make a perturbed displacement of the velocity field at the bottom of the ocean, supposing that some volume of water is beginning to expand, we are working against gravity as well as our virtual force, pressure.

(draw this)

Chapter 4

The finite element method

4.1 Introduction: Solving Poisson's equation

— Finite element method. — Galerkin methods.

—NOTE: Mistake in n numbering. Need to talk about Dirichlet condition earlier. —NOTE: Mistake in the form of the linear systems. They need to incorporate the boundary conditions.

Among the fundamental PDEs are the heat equation

$$\frac{\partial h}{\partial t} = \Delta h + g$$

and the Poisson equation

$$-\Delta h = g. \tag{4.1}$$

The latter can be thought of as the steady-state version of the former. The solution of a Poisson problem will be a crucial component of the solution of the Stokes equations. It seems ideal to start by discussing discretization methods for the Poisson equation (4.1) in particular, as it is likely the simplest non-trivial PDE. The focus here is on the Dirichlet problem

$$-\Delta h = g, \quad h|_\Gamma = h_\Gamma, \tag{4.2}$$

where Γ is the boundary of the domain, and the domain is 2D.

4.1.1 Discretizing the differential versus integral form

Discretizing the differential form of the PDE

It is a theorem of Gauss that in Euclidean space $\mathbb{R}^3$ we have

$$\nabla \cdot v = \frac{\partial v_x}{\partial x} + \frac{\partial v_y}{\partial y} + \frac{\partial v_z}{\partial z}, \tag{4.3}$$

and we get (4.1) in the form

$$-\left(\frac{\partial^2 h}{\partial x^2} + \frac{\partial^2 h}{\partial y^2} + \frac{\partial^2 h}{\partial z^2}\right) = g. \tag{4.4}$$

This form immediately indicates an effective method of discretization, by forming secant approximations of the derivatives over a regular grid. The approximate solution is then represented as a function of this grid. Thinking about (4.4) leads to the finite difference method, historically the first and still very important in applications. However, it may be hard to give a real interpretation of what the solution samples at grid points say about the solution everywhere. They could be a coefficients of, for example, hat basis functions. Yet a finite difference discretization of the Poisson equation (4.1) may not take this into account at all. One might feel that limits have been taken too soon.

Discretizing the integral form of the PDE

In physics, Poisson's equation is an equilibrium conservation law, and therefore has an integral form. This form will give clearer routes to discretizations which have geometric meaning. For example, the general integral conservation law (2.1),

$$\frac{d}{dt}\int_{\Omega_0} \phi \, dx = \int_{\Omega_0} s \, dx + \int_{\partial\Omega_0} \phi j \cdot (-\hat{n}) \, dx,$$

is a geometric statement about fluxes of quantity ϕ by j, quantified over arbitrary control volumes Ω_0. There are an infinite number of control volumes, and therefore an infinite number of equations which must hold, and there is an infinite-dimensional space of solutions to choose from. A key idea, then, is to choose a finite number of equations and a finite-dimensional subspace of possible approximate solutions. A supposed solution will be represented by the coefficients of some choice of basis functions for the subspace. Each equation will be checked exactly against this supposed solution.

(figure)

Finite element methods, and more generally Galerkin methods, live comfortably in the language of weak solutions, integral forms of partial differential equations, and the calculus of variations. To continue with this idea, we must write the Poisson equation (4.1) in integral form.

4.1.2 Deriving the heat and Poisson equation through diffusion processes

The Poisson equation (4.1) can be thought of as the steady state of some diffusion process with a source term g, although this need not be its literal physical interpretation. A diffusion process "levels out" some quantity, such as temperature or some chemical concentration. A diffusion could intuitively be thought of as a progressive "blurring", such as in a camera defocus, and in fact many common image processing techniques use diffusion PDEs from physics [16]. We will stick with the notion of "temperature" h as the diffused quantity. *Fick's law of diffusion* is a constitutive relation giving the bulk flux of temperature h as proportional to the negative gradient:

$$hj = -\mu \nabla h,$$

where μ is called the diffusion coefficient. This is one way of saying that the temperature tends to level out. If we form a continuity equation (2.1) for temperature, with source s, we get

$$\frac{d}{dt}\int_{\Omega_0} h \, dx = \int_{\Omega_0} s \, dx + \int_{\partial\Omega_0} \mu \nabla h \cdot \hat{n} \, dx, \tag{4.5}$$

which by application of Stokes' theorem becomes

$$\frac{dh}{dt} = s + \nabla \cdot (\mu \nabla h) \, . \tag{4.6}$$

If we further assume that the diffusion coefficient μ is constant, we get

$$\frac{dh}{dt} = s + \mu \nabla \cdot \nabla h = s + \mu \Delta h, \tag{4.7}$$

which is the standard heat equation. The steady-state heat equation is then

$$-\Delta h = g, \tag{4.8}$$

where we let $g = s/\mu$ in the above. This completes the derivation of the Poisson equation (4.1). In integral form, "undoing" the application of Stokes' theorem above, the Poisson equation is

$$\int_{\partial\Omega_0} -\nabla h \cdot \hat{n}\, dx = \int_{\Omega_0} g\, dx \quad \text{for all control volumes } \Omega_0. \tag{4.9}$$

Form (4.9) clearly shows that we are calculating a steady state, as we are solving for h such that the amount of heat that leaves Ω_0 is the amount introduced into Ω_0 by the source function. The form (4.9), rather than (4.1), will be the starting point for deriving Galerkin methods.

4.1.3 Discretizing the Poisson equation by finite volumes

Equation (4.9) is quantified over arbitrary control volumes Ω_0. A simple idea is to break the domain Ω up into small cells $\Omega_1, \cdots, \Omega_n$ (hence "finite volumes"), and check that the flux integral holds over each of these. We will then have n equations on h. As this system will be underdetermined (h has infinite degrees of freedom), we must restrict h to what we will call a "test space",

$$\Phi = \operatorname{span}\{\phi_1, \cdots, \phi_n\}.$$

The basis functions ϕ_i should generate a "good" space of approximations, such that the resulting linear system is non-singular. We then have a discrete system of equations

$$\int_{\partial\Omega_j} -\nabla \left(\sum_{i=1}^{n} h_i \phi_i \right) \cdot \hat{n}\, dx = \int_{\Omega_j} g\, dx, \quad j = 1, \cdots, n.$$

By linearity, to emphasize the separate integrals that need to be computed, the above equation can be written as

$$\sum_{i=1}^{n} h_i \int_{\partial\Omega_j} -\nabla \phi_i \cdot \hat{n}\, dx = \int_{\Omega_j} g\, dx, \quad j = 1, \cdots, n. \tag{4.10}$$

We see here that there must be some restrictions on the ϕ_i. Formally, the basis functions must be in the Sobolev space $H^1(\Omega)$. This simply means that they must have a gradient defined "almost everywhere". It does not matter if the gradient is not defined at isolated lower-dimensional subsets, as these make no contribution to the integral. Since the source g, the domain partition Ω_j, and the basis functions ϕ_i are known, we can pre-compute the majority of (4.10) to give a matrix system

$$A\hat{h} = \begin{bmatrix} \int_{\partial\Omega_1} -\nabla\phi_1 \cdot \hat{n}\, dx & \cdots & \int_{\partial\Omega_1} -\nabla\phi_n \cdot \hat{n}\, dx \\ \vdots & & \vdots \\ \int_{\partial\Omega_n} -\nabla\phi_1 \cdot \hat{n}\, dx & \cdots & \int_{\partial\Omega_n} -\nabla\phi_n \cdot \hat{n}\, dx \end{bmatrix} \begin{bmatrix} h_1 \\ h_2 \\ \vdots \\ h_{n-1} \\ h_n \end{bmatrix} = \begin{bmatrix} \int_{\Omega_1} g\, dx \\ \int_{\Omega_2} g\, dx \\ \vdots \\ \int_{\Omega_{n-1}} g\, dx \\ \int_{\Omega_n} g\, dx \end{bmatrix} = \hat{g}. \tag{4.11}$$

This system is solved for the coefficients of combination for the test functions, $\hat{h} = (h_1 \cdots h_n)^T$. The solution is then denoted $\Phi \cdot \hat{h} = h_1\Phi_1 + \cdots + h_n\Phi_n$. If the domain partition $\Omega_1, \cdots, \Omega_n$ and test space $\Phi = \operatorname{span}\{\phi_1, \cdots, \phi_n\}$ are chosen well, this linear system will be non-singular and hopefully well-conditioned. In all cases we have a conservative system of balanced fluxes, but it is another question whether our approximation $\Phi \cdot \hat{h}$ is good.

Choosing a domain partition and test space

Possibly the simplest scheme is to triangulate Ω as $\bigcup_i T_i$, such that we have n nodal points and n_T triangles. The boundary of Ω is approximated as piecewise linear. Each nodal point p_i will be associated with a piecewise "hat" basis function ϕ_i which is 1 at p_i and 0 at its neighbours. An example decomposition for an ellipse-shaped domain is shown in figure 4.1.

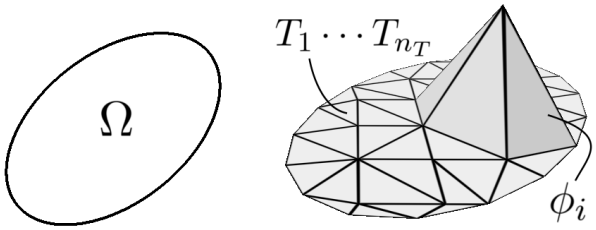


Figure 4.1: The domain Ω is partitioned into triangular cells $T_1 \cdots T_{n_T}$. One example of a hat basis function ϕ_i is shown, where the vertical axis is the basis function value.

As there are typically more triangles than vertices, we cannot use the T_i as the domain partition. However, if we choose some characteristic "triangle centre" for each triangle, then we may associate to each p_i a domain Ω_i defined by the polygon which joins the centres of the adjacent triangles. Two common choices for the triangle centre are the barycentre, which is the average position of the three vertices, and the circumcentre, which is the centre of the unique circle passing through the three vertices. The circumcentre scheme gives what are called "Voronoi cells" due to their relation to Voronoi diagrams in computational geometry [29]. The resulting cell decompositions are displayed in figure 4.2.

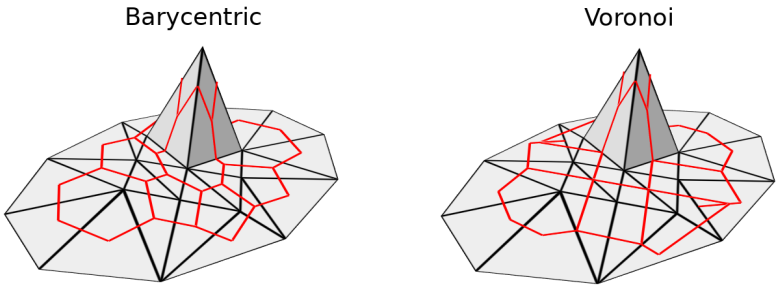


Figure 4.2: The domain Ω is partitioned into triangular cells $T_1 \cdots T_{n_T}$. Flux integrals are taken over n polygonal cells, one for each internal vertex p_i, for example using triangle barycenters or circumcenters.

This scheme has found some success, especially in the domain of geometry processing [14]. By Stokes' theorem, the matrix A in (4.11) can be thought of as a negative discrete Laplacian. If we compute these integrals, we will find a very simple closed form for the entries of A:

cotan formula.

In geometry processing this matrix is called the "cotangent Laplacian" [14], and it is typically applied to surface meshes in $\mathbb{R}^3$, which can be thought of as triangulations of a smooth surface.

Results and visualisation

(results and visualisation)

We have worked through an instance of a *finite volume method* [4]. Finite volume methods are characterised by an exact domain partition and computation of flux integrals. Finite volume methods are typically *conservative*, due to the "flux network" nature of the discretisation.

4.1.4 From finite volumes to finite elements

The alternative "finite element" approach is directly related to the "finite volumes" described above. While each finite volume had a direct geometric meaning (as a small cell in which a certain test flux integral is taken), the geometric meaning of a "finite element" requires slightly more thought, although it will be seen to be the same idea in disguise. The matrix equation (4.11) consists of linear equations

$$\int_{\partial\Omega_1} -\nabla\left(\sum_{i=1}^{n} h_i\phi_i\right)\cdot\hat{n}\,dx = \int_{\Omega_1} g\,dx \quad (\text{for } j=1),$$

and so on. We cannot compute flux integrals over all arbitrary control volumes, but we can take a number of "trial" flux integrals over the finite number of cells Ω_i. We can take linear combinations of these equations to get more equations which must hold on a solution. For example,

$$\int_{\partial\Omega_1} -\nabla\left(\sum_{i=1}^{n} h_i\phi_i\right)\cdot\hat{n}\,dx + \int_{\partial\Omega_2} -\nabla\left(\sum_{i=1}^{n} h_i\phi_i\right)\cdot\hat{n}\,dx = \int_{\Omega_1} g\,dx + \int_{\Omega_2} g\,dx \qquad (4.12)$$

must hold. At first sight, (4.12) cannot directly be interpreted as a statement about a "flux integral", but rather about a sum of flux integrals. However, a key idea is to regard (4.12) as a flux integral over a *formal sum* of domains,

$$\Omega_1 + \Omega_2.$$

We now have the equation

$$\int_{\partial\Omega_1+\partial\Omega_2} -\nabla\left(\sum_{i=1}^{n} h_i\phi_i\right)\cdot\hat{n}\,dx = \int_{\Omega_1+\Omega_2} g\,dx. \qquad (4.13)$$

In differential geometry $\Omega_1 + \Omega_2$ is called a *chain*. For example, we may visualise $\Omega_1 + 2\Omega_2 + 0.5\Omega_4$ as:

(draw this)

We define the boundary operator ∂ to be linear in formal sums e.g.,

$$\partial(\Omega_1 + \Omega_2) = \partial\Omega_1 + \partial\Omega_2.$$

If Ω_1 and Ω_2 share a boundary, we would like $\Omega_1 + \Omega_2$ to represent their union, such that a flux integral over $\partial(\Omega_1 + \Omega_2)$ evaluates to zero on the shared boundary. This can be done by thinking of the boundary as *oriented*, as in, consisting of oriented "surface elements" over which flux integrals can be taken. For example, the $\hat{n}$ in a flux integral denotes the outward-pointing normal, which represents an "outward-flux-measuring surface element".

The opposite $-\hat{n}$ then represents the "inward-flux-measuring surface element", which is outward from the perspective of an adjacent cell.

(draw this)

We may now define

$$\Psi = \operatorname{span}\{\Omega_1, \cdots, \Omega_n\}$$

to be the *trial space*, where the span is taken with respect to formal sums. As with a typical linear space, we may choose from many possible bases. For example,

$$\Psi = \operatorname{span}\{\Omega_1, \Omega_2, \Omega_3\} = \operatorname{span}\{\Omega_1 + \Omega_2, 2\Omega_2, \Omega_3\}.$$

A key idea, leading to Galerkin methods, is to allow freedom in the choice of our trial space Ψ. Notably, we do not need Ψ to be a space of formal sums of domains. The Poisson equation is discretised over flux integrals around cell boundaries in the linear system (4.10), which we repeat here:

$$\sum_{i=1}^{n} h_i \int_{\partial\Omega_j} -\nabla\phi_i \cdot \hat{n}\, dx = \int_{\Omega_j} g\, dx, \quad j = 1, \cdots, n.$$

Applying Stokes' theorem, we get

$$\sum_{i=1}^{n} h_i \int_{\Omega_j} -\Delta\phi_i\, dx = \int_{\Omega_j} g\, dx, \quad j = 1, \cdots, n.$$

We can think of these integrals as over the *entire domain* Ω, giving the form

$$\sum_{i=1}^{n} h_i \int_{\Omega} -\Delta\phi_i \cdot \chi(\Omega_j)\, dx = \int_{\Omega} g \cdot \chi(\Omega_j)\, dx, \quad j = 1, \cdots, n.$$

where $\chi(\Omega_j)$ is the indicator function of Ω_j,

$$\chi(\Omega_j)(x) \coloneqq \begin{cases} 0 & \text{if } x \in \Omega_j \\ 1 & \text{if } x \notin \Omega_j. \end{cases}$$

We can now think of our trial space Ψ as a span of functions, instead of a span of domains:

$$\Psi = \operatorname{span}\{\chi(\Omega_1), \cdots, \chi(\Omega_n)\}.$$

Now, as the trial space is just a regular function space, we could instead let

$$\Psi = \operatorname{span}\{\psi_1, \cdots, \psi_n\}$$

where the ψ_j need not be the indicator functions of a domain partition. We now have the system of equations

$$\sum_{i=1}^{n} h_i \int_{\Omega} -\Delta\phi_i\psi_j\, dx = \int_{\Omega} g\psi_j\, dx, \quad j = 1, \cdots, n.$$

By integration by parts we have the system of equations

$$\sum_{i=1}^{n} h_i \int_{\Omega} \nabla\phi_i \cdot \nabla\psi_j\, dx = \int_{\Omega} g\psi_j\, dx, \quad j = 1, \cdots, n, \tag{4.14}$$

and we see that we still only require the ϕ_i to be in $H^1(\Omega)$. We can see that equation (4.14) is very similar to the exact fluxes in (4.10), and indeed (4.14) gives a discrete linear system in much the same way. There is a real geometric sense in which (4.14) is a "blurred convolution" of flux integrals.

— Mention the weak form, the above is an alternative motivation using the discrete equations rather than variational methods with the original PDE.

4.1.5 Discretizing Poisson's equation by finite elements

Equation (4.14) gives a discrete linear system

$$A\hat{h} = \begin{bmatrix} \int_\Omega \nabla\phi_1 \cdot \nabla\psi_1 \, dx & \cdots & \int_\Omega \nabla\phi_n \cdot \nabla\psi_1 \, dx \\ \vdots & & \vdots \\ \int_\Omega \nabla\phi_1 \cdot \nabla\psi_n \, dx & \cdots & \int_\Omega \nabla\phi_n \cdot \nabla\psi_n \, dx \end{bmatrix} \begin{bmatrix} h_1 \\ h_2 \\ \vdots \\ h_{n-1} \\ h_n \end{bmatrix} = \begin{bmatrix} \int_\Omega g\psi_1 \, dx \\ \int_\Omega g\psi_2 \, dx \\ \vdots \\ \int_\Omega g\psi_{n-1} \, dx \\ \int_\Omega g\psi_n \, dx \end{bmatrix} = \hat{g}. \quad (4.15)$$

This has quite a similar form to (4.11), but with boundary and cell integrals replaced by integrals over the entire domain. Again, if the domain partition $\Omega_1, \cdots, \Omega_n$ and test space $\Phi = \operatorname{span}\{\phi_1, \cdots, \phi_n\}$ are chosen well, this linear system will be non-singular and hopefully well-conditioned.

Choosing a test and a trial space

Notice that the integrals in (4.15), in contrast to (4.11), are over the *entire domain*. It may be very costly in general to construct such a matrix, requiring the computation of many large integrals. The finite element method, in particular, solves this problem by "localising" basis functions of the test and trial spaces. Each basis test function ϕ_i has *compact support*, meaning that there is some compact subdomain D_i^ϕ such that

$$\phi_i(x) = \begin{cases} \phi_i(x) & \text{if } x \in D_i^\phi \\ 0 & \text{otherwise,} \end{cases}$$

and similarly each basis trial function ψ_i has some corresponding compact subdomain D_i^ψ such that

$$\psi_i(x) = \begin{cases} \psi_i(x) & \text{if } x \in D_i^\psi \\ 0 & \text{otherwise.} \end{cases}$$

This has the effect of reducing the domain of each integral in (4.15), as

$$\int_\Omega \nabla\phi_i \cdot \nabla\psi_j \, dx = \int_{D_i^\phi \cap D_j^\psi} \nabla\phi_i \cdot \nabla\psi_j \, dx.$$

With well-localised basis functions, the intersection will be

$$D_i^\phi \cap D_j^\psi = \emptyset$$

for most indices i, j, implying the matrix (4.15) will be *sparse*. In practice this allows the use of iterative or graph-based sparse matrix algorithms which can be hugely more efficient than dense matrix computations of the same size. The size of the matrix will increase quadratically with the number of nodes in the discretization, while the number of nonzeros typically increases linearly. A typical sparsity pattern for a finite element problem is shown in figure 4.3.

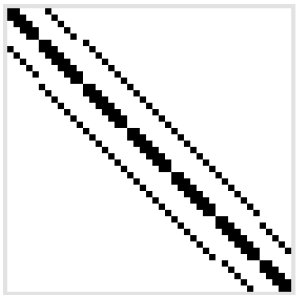


Figure 4.3: An example sparsity pattern for the finite element matrix of a 2D Poisson problem. White is zero and black is non-zero. The mesh has 61 vertices (16 on the boundary), and 104 triangles, resulting in a 45x45 system with 2025 entries, 197 non-zeros, and fill of 0.0973.

Similar to the hat-functions-and-Voronoi-cells decomposition described for finite volumes, the simplest scheme for the finite element Poisson equation is to triangulate Ω as $\bigcup_i T_i$ such that we have n nodal points and n_T triangles. Each nodal point p_i will be associated with a piecewise "hat" basis function ϕ_i which is 1 at p_i and 0 at its neighbours. Now, diverging from the previous finite volume method, the simplest thing we could do is let the trial functions be the same as the test functions, $\psi_i = \phi_i$. This trivially gives a one-to-one correspondence between the ϕ_i and the ψ_i, which will lead to a well-formed linear system.

(draw this)

We have so far worked up to an instance of a *finite element method*. Finite element methods are characterised by a test space Φ and trial space Ψ with basis functions of *compact support*, where Φ and Ψ typically consist of continuous functions in some Sobolev space, constructed over a domain tessellation.

4.1.6 Implementing the finite element method for Poisson's equation

We now have an effective algorithm for approximating the Poisson equation:

1. Partition the 2D domain Ω into triangles T_i. This implicitly defines the hat basis functions $\phi_i = \psi_i$.

2. Form the matrix and right-hand-side in (4.15), by analytical or numerical integration.

3. Solve the resulting linear system for $\hat{h}$, and construct the solution as $\Phi \cdot \hat{h}$.

Each of these steps is conceptually well-separated into the domains of mesh generation, matrix assembly, and numerical linear algebra.

Mesh generation

Mesh generation is itself a huge field. However, there are only two essentials for the handling of 2D meshes and piecewise-linear finite elements — a mesh data structure and a triangulator. Typically a mesh data structure will be provided by a separate library, and the data organization (e.g., vertex, triangle, and adjacency information) will be designed to facilitate the kinds of mesh traversals performed during matrix assembly. This is typically some variant of a "halfedge" data structure [14], implemented in libraries such as Geometry Central [33], the Polygon Mesh Processing library [34], and OpenMesh [35]. A mesh data structure is a foundational component of *mesh generation* systems. For 2D domains, a somewhat regular sampling of points on the boundary and interior, followed by

triangulation of these points, can suffice for a piecewise-linear finite element mesh. The Triangle [27] library, for example, contains a single very efficient and robust C routine (consisting of 16k lines of highly optimized code) for performing Delaunay triangulations [29] on 2D domains, with the specific goal of creating robust finite element meshes.

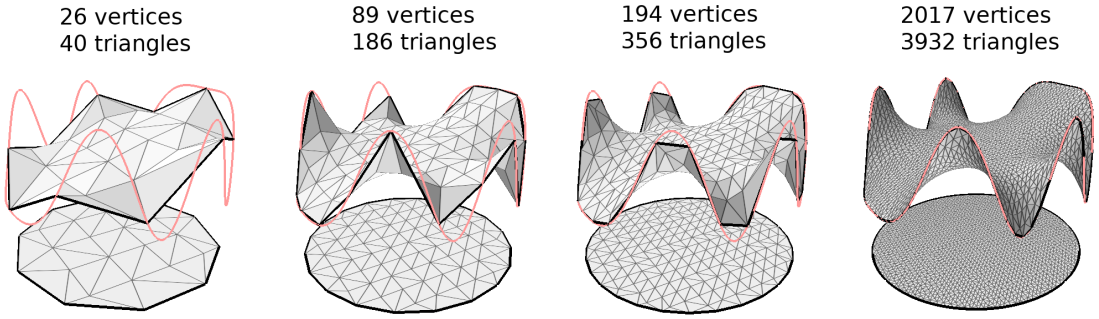
Matrix assembly

The matrix assembly stage is the core of a finite element implementation. The simplest idea is to iterate over all pairs ϕ_i and ψ_j, letting $i = 1, \cdots, n$, $j = 1, \cdots, n$, and approximate the integral $\int_\Omega \nabla\phi_i \cdot \nabla \cdot \psi_j \, dx$, and store the value as entry (i, j) of the matrix.

$M \leftarrow$ zero matrix;
for $i = 1, \cdots, n$ **do**
  for $j = 1, \cdots, n$ **do**
    $M[i, j] \leftarrow \int_\Omega \nabla\phi_i \cdot \nabla \cdot \psi_j \, dx$.
  end
end

Numerical linear algebra

The solution of large linear systems is a vast topic in itself. Finite element solvers are typically clients of standard, robust linear and non-linear solver libraries, such as Argonne National Laboratory's PETSc libraries [30] and the smaller-scale C++ libraries Armadillo [31] and Eigen [32]. A finite element solver will typically pass either a full matrix to the linear solver (dense or sparse), or provide the linear solver with callback routines that give the linear solver access to the linear system coefficients when they are needed.



4.2 Solving the Stokes equations

— Lid-driven cavity flow.

The Stokes equations (3.13), which are solved for a stable incompressible Navier-Stokes flow, assume the Reynolds number is $Re \ll 1$ and thus convective processes are neglible in comparison to viscous processes. We will begin with the steady-state form (3.14). Due to this simplification, the Steady stokes equations (3.14), repeated here:

$$\mu\Delta u + \rho g - \nabla p = 0, \quad \nabla \cdot u = 0,$$

form a constrained linear equation. As we saw in section 3.4, the pressure term p is a Lagrange multiplier introduced with the constraint $\nabla \cdot u = 0$. We will begin by discretizing the *unconstrained* steady Stokes equations, which are a vector Poisson equation:

$$-\mu\Delta u = \rho g. \tag{4.16}$$

4.2.1 Discretizing the vector Poisson equation

In principle we should keep the Stokes equation in integral form (using the conservative-form Cauchy momentum equation (2.31)), and continue as we did in section 4.1.4. However, we will take a formal step to skip the reasoning of section 4.1.4, typical of finite element method derivations. As we start with the *differential* equation (4.16), we can introduce a trial space Ψ and then "weaken" the equation by integrating against $v \in \Psi$, and removing the Laplacian by integration by parts:

$$\int_\Omega -\mu\Delta u \cdot v \, dx = \int_\Omega \rho g \cdot v \, dx \quad \equiv \quad \int_\Omega -\mu \nabla u : \nabla v \, dx = \int_\Omega \rho g \cdot v \, dx. \tag{4.17}$$

Noting that the left-hand-side of (4.17) is a bilinear form in u and v, and the right-hand-side is a linear functional in ψ, it is standard practice (ref) to write this kind of equation as

$$a(u, v) = f(v). \tag{4.18}$$

Our subsequent derivations are much the same as in 4.1.3, simplified by our new notation. We can now approximate u in the test space Φ as $\hat{u} = \sum_{i=1}^n u_i \phi_i$. By linearity we only need to compute trials over the basis trial functions ψ_j. We then have the linear system of equations

$$\sum_{i=1}^n u_i a\left(\phi_i, \psi_j\right) = f(\psi_j), \quad j = 1, \cdots, n, \tag{4.19}$$

which can be written in matrix form as

$$A\hat{u} = \begin{bmatrix} a(\phi_1, \psi_1) & \cdots & a(\phi_1, \psi_n) \\ \vdots & & \vdots \\ a(\phi_n, \psi_1) & \cdots & a(\phi_n, \psi_n) \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_{n-1} \\ u_n \end{bmatrix} = \begin{bmatrix} f(\psi_1) \\ f(\psi_2) \\ \vdots \\ f(\psi_{n-1}) \\ f(\psi_n) \end{bmatrix} = \hat{f}. \tag{4.20}$$

We can solve (4.20) to get a velocity field $\sum_{i=1}^n u_i \phi_i$, although in general this will not satisfy $\nabla \cdot u = 0$. As some preliminary analysis, if $\Phi = \Psi$ and have the same basis functions, we have a symmetric-positive-definite system. This form of linear system is known to be stably solvable, for example by the conjugate gradient method.

4.2.2 Discretizing the steady Stokes equations

As described in section 3.4, the pressure p is a Lagrange multiplier that appears when solving the optimization problem (3.16):

$$\begin{aligned} \underset{u}{\text{minimize}} \quad & E(u) = \frac{\mu}{2} \langle \nabla u, \nabla u \rangle - \langle u, \rho g \rangle \\ \text{subject to} \quad & \nabla \cdot u = 0. \end{aligned}$$

As a first idea, we can introduce p as a variable to solve for. Solving for the pressure (the "dual variable") simultaneously with the velocity (the "primal variable") is called a primal-dual method for the optimization (3.16), and the resulting finite element method is called *mixed*. Pressure then needs to be discretized, so we introduce another test space Φ_{pressure}. To get a weak form of the steady Stokes equations (3.14), which are two equations including the constraint $\nabla \cdot u = 0$, we introduce another trial space $\Psi_{\text{constraint}}$, whose functions will be integrated against $\nabla \cdot u$. The weak form is then

$$\begin{aligned} &\int_\Omega (\mu\Delta u + \rho g - \nabla p) \cdot v \, dx = 0, \\ &\int_\Omega (\nabla \cdot u) \, q \, dx = 0, \quad \text{where } v \in \Psi, q \in \Psi_{\text{constraint}}, \end{aligned}$$

which by integration by parts can be written as

$$\begin{aligned} &\int_\Omega -\mu\nabla u : \nabla v - (\nabla\cdot v)\, p\, dx = \int_\Omega \rho g\cdot v\, dx, \\ &\int_\Omega (\nabla\cdot u)\, q\, dx = 0, \quad \text{where } v\in\Psi, q\in\Psi_{\text{constraint}}. \end{aligned} \tag{4.21}$$

As in section 4.2.1, we introduce notation for the bilinear and linear forms in (4.21):

$$\begin{aligned} a(u,v) &\coloneqq \int_\Omega -\mu\nabla u : \nabla v\, dx, \quad \text{for } u\in\Phi, v\in\Psi, \\ \hat{b}(p,v) &\coloneqq \int_\Omega -(\nabla\cdot v)\, p\, dx, \quad \text{for } p\in\Phi_{\text{pressure}}, v\in\Psi, \\ b(u,q) &\coloneqq \int_\Omega -(\nabla\cdot u)\, q\, dx, \quad \text{for } u\in\Phi, q\in\Psi_{\text{constraint}}, \\ f(v) &\coloneqq \int_\Omega \rho g\cdot v\, dx \quad \text{for } v\in\Psi. \end{aligned} \tag{4.22}$$

Although they have the same form, b and $\hat{b}$ are distinguished as they take inputs in different function spaces. We now have a simplified notation for the weak form (4.21),

$$\begin{aligned} &a(u,v)+\hat{b}(p,v) = f(v), \\ &b(u,q) = 0, \quad \text{where } v\in\Psi, q\in\Psi_{\text{constraint}}. \end{aligned} \tag{4.23}$$

Working with discrete function spaces, we get a $2n\times 2n$ linear system in the unknowns $u_1,\cdots,u_n$ and $p_1,\cdots,p_n$,

$$\begin{aligned} &\sum_{i=1}^n u_i a\left(\phi_i,\psi_j\right) + \sum_{i=1}^n p_i \hat{b}\left(\phi_i^C,\psi_j\right) = f(\psi_j), \\ &\sum_{i=1}^n u_i b\left(\phi_i,\psi_j^C\right) = 0, \quad j=1,\cdots.n. \end{aligned} \tag{4.24}$$

To emphasize the linear system structure of (4.23), the block matrix form is:

$$\begin{aligned} M\hat{x} &= \begin{bmatrix} A & \hat{B} \\ B & 0 \end{bmatrix} \hat{x} \\ &= \left[\begin{array}{ccc|ccc} a(\phi_1,\psi_1) & \cdots & a(\phi_1,\psi_n) & \hat{b}(\phi_1^C,\psi_1) & \cdots & \hat{b}(\phi_1^C,\psi_n) \\ \vdots & & \vdots & \vdots & & \vdots \\ a(\phi_n,\psi_1) & \cdots & a(\phi_n,\psi_n) & \hat{b}(\phi^C n,\psi_1) & \cdots & \hat{b}(\phi_n^C,\psi_n) \\ \hline b(\phi_1,\psi_1^C) & \cdots & b(\phi_1,\psi_n^C) & 0 & \cdots & 0 \\ \vdots & & \vdots & & & \\ & \vdots & & \vdots & & \\ b(\phi_n,\psi_1^C) & \cdots & b(\phi_n,\psi_n^C) & 0 & \cdots & 0 \end{array}\right] \left[\begin{array}{c} u_1 \\ \vdots \\ u_n \\ \hline p_1 \\ \vdots \\ p_n \end{array}\right] = \left[\begin{array}{c} f(\psi_1) \\ \vdots \\ f(\psi_n) \\ \hline 0 \\ \vdots \\ 0 \end{array}\right] = \hat{b}. \end{aligned} \tag{4.25}$$

Is this method reasonable?

For the vector Poisson equation, letting $\Phi=\Psi$, we ended up with a symmetric-positive-definite system (4.20), which is known to be stably solvable. We can ask how reasonable it

is to solve (4.25), and what trial and test spaces we should use. In fact, in the problem (4.23), and more generally in a "saddle point problem", arising in Lagrange-multiplier methods for constrained PDEs, we should not choose just any test and trial spaces. The Ladyzhenskaya–Babuška–Brezzi condition, discussed later, enforces restrictions on choices that result in a stable method. We will until then continue with computations.

Results and visualisation

(results and visualisation)

4.2.3 Discretizing the unsteady Stokes equations

The steady Stokes above are the stable state of the time-dependent Stokes flow, after the transient flow behaviour settles down. The unsteady Stokes equations (3.13) are

$$\rho\frac{\partial u}{\partial t} = \mu\Delta u + \rho g - \nabla p, \quad \nabla\cdot u = 0.$$

These form an initial-boundary-value problem, and this will be our first attempt at discretizing a PDE in time. We could think of solving with the test and trial spaces over the domain $\Omega\times[0,T)$, but this is typically not done due to the memory costs, and different qualitative meaning of the time variable [22]. Instead, we will use an implicit-Euler finite difference in time:

$$\frac{\rho}{\Delta t}\left(u^{(n)} - u^{(n-1)}\right) = \mu\Delta u^{(n)} + \rho g - \nabla p^{(n)}, \quad \nabla\cdot u^{(n)} = 0, \tag{4.26}$$

where Δt is a fixed time step, and $u^{(n)}$ and $p^{(n)}$ is the solution at time $t_n = n\Delta t$. We can weaken each step (4.26) by integrating against trial functions $v\in\Psi$ and $q\in\Psi_{\text{constraint}}$, performing integration by parts as in section 4.2.2, and rearranging the knowns and unknowns. We can also let u be $u^{(n)}$, p be $p^{(n)}$, and u_{prev} be $u^{(n-1)}$ in the above to simplify subsequent notation. The weak form of (4.26) is then:

$$\begin{gathered}\int_\Omega \frac{\rho}{\Delta t}u\cdot v - \mu\nabla u : \nabla v - (\nabla\cdot v)\,p\,dx = \int_\Omega \frac{\rho}{\Delta t}u_{\text{prev}}\cdot v + \rho g\cdot v\,dx,\\ \int_\Omega (\nabla\cdot u)\,q\,dx = 0,\end{gathered} \tag{4.27}$$

We can define the linear forms as

$$\begin{aligned} a(u,v) &\coloneqq \int_\Omega \frac{\rho}{\Delta t}u\cdot v - \mu\nabla u : \nabla v\,dx, \quad \text{for } u\in\Phi, v\in\Psi,\\ \hat{b}(p,v) &\coloneqq \int_\Omega -(\nabla\cdot v)\,p\,dx, \quad \text{for } p\in\Phi_{\text{pressure}}, v\in\Psi,\\ b(u,q) &\coloneqq \int_\Omega -(\nabla\cdot u)\,q\,dx, \quad \text{for } u\in\Phi, q\in\Psi_{\text{constraint}},\\ f(v) &\coloneqq \int_\Omega \frac{\rho}{\Delta t}u_{\text{prev}}\cdot v + g\cdot v\,dx \quad \text{for } v\in\Psi \end{aligned} \tag{4.28}$$

to reexpress (4.27) in the notation

$$\begin{aligned} &a(u,v) + \hat{b}(p,v) = f(v),\\ &b(u,q) = 0, \quad \text{where } v\in\Psi, q\in\Psi_{\text{constraint}}. \end{aligned} \tag{4.29}$$

This is the same structure as in the steady Stokes system (4.23), and so the matrix block structure is the same as in (4.25). Therefore, every step we need to solve a linear system that is very similar to the steady Stokes problem. In fact this step can be thought of as successively introducing the momentum source ρg (ignoring convection), while solving for the updated pressure needed to keep the fluid non-compressed.

Discretizing the initial condition

We may have some analytically determined, or otherwise, initial velocity field u with $\nabla \cdot u = 0$. We would like to form $u^{(0)}$ in order to start the iteration. The velocity u should be projected in some way into the test space Ψ. Enforcing $u^{(0)}$ to give the same "blurred average" value when evaluated against a trial function,

$$\int_\Omega u^{(0)} \cdot v \, dx = \int_\Omega u \cdot v \, dx, \quad \forall v \in \Psi, \tag{4.30}$$

gives a linear system

$$\sum_{i=1}^{n} u_i^{(0)} \int_\Omega \phi_i \cdot \psi_j \, dx = \int_\Omega u \cdot \psi_j \, dx, \quad j = 1, \cdots, n. \tag{4.31}$$

Solving this linear system for the $u_i^{(0)}$ gives $u^{(0)} = \sum_{i=1}^{n} u_i^{(0)} \phi_i$ as a projection of u onto Φ. This projection is orthogonal if $\Phi = \Psi$, and therefore could be considered the "best" such projection in the Euclidean norm. This is the standard Gramian matrix construction for projection in approximation theory [20].

4.3 Solving non-linear equations

4.3.1 A non-linear Poisson equation

4.3.2 A non-linear heat equation

4.3.3 The Burgers equation

4.4 Implementing finite element methods

4.4.1 The Ciarlet definition of a finite element space

[15], [22], [19]

One great utility of the finite element method is that it is compatible with complex geometric domains and domain partitions. Some greater effort is needed to apply finite differences correctly across complex boundaries, and it is non-trivial to implement a varying resolution of the discretisation. In the finite element method, however, specifying the test and trial functions over a square grid is much the same as specifying them over, for example, a surface mesh of arbitrary topology. For modelling, for example, heat transfer in a complex solid, one can construct basis functions over a grid on the interior, and over cut-off boundary cells near the exterior. This is all in principle, of course, as one needs to

1. Break the domain up into small pieces.
2. Construct the basis and trial functions over this domain partition (e.g. by finding polynomial coefficients).
3. Compute all inner products of test and trial functions, and their relevant derivatives, either numerically or analytically.
4. Solve possibly many huge sparse linear systems, while possibly changing the structure of the domain partition (requiring changes to the inner products).

Step (1) is already a field in itself, as evidenced by the open source tool TetGen [28]. TetGen is a small tool whose primary purpose is to perform *constrained Delaunay tetrahedralization*, given a solid boundary and a point cloud on the interior. This constructs

a valid tetrahedral partition intended for finite element solvers. The partition is efficiently created in a very robust manner in over 36k lines of C code. TetGen is part of the geometric backbone of many FEA tools (references).

Chapter 5

Solving the Navier-Stokes equations

Chapter 6

Some functional analysis

(Appendix)

Bibliography

[1] Isaac Newton, *Philosophiae Naturalis Principia Mathematica (Third edition)*, 1726.

[2] Johann Bernoulli, *"Problema novum ad cujus solutionem Mathematici invitantur." (A new problem to whose solution mathematicians are invited.)*, 1696. (retrieved from wikipedia/brachistochrone_curve)

[3] A. F. Monna, *Dirichlet's principle: A mathematical comedy of errors and its influence on the development of analysis*, 1975.

[4] Stig Larsson, *Partial differential equations with numerical methods*, 2003.

[5] Peter Lax, *Hyperbolic Systems of Conservation Laws and the Mathematical Theory of Shock Waves*, 1973.

[6] Cornelius Lanczos, *The Variational Principles of Mechanics*, 1952.

[7] G. K. Batchelor, *Introduction to Fluid Dynamics*, 1967.

[8] L. Gary Leal, *Advanced Transport Phenomena: Fluid Mechanics and Convective Transport Processes*, 2007.

[9] Vivette Girault, Pierre-Arnaud Raviart, *Finite Element Methods for Navier-Stokes equations*, 1986.

[10] Richard Feynman, *Surely You're Joking, Mr. Feynman!*, 1985.

[11] V.I. Arnol'd, *Mathematical Methods of Classical Mechanics*, 1978.

[12] Alan Turing, *The Chemical Basis of Morphogenesis*, 1952.

[13] *The Princeton Companion to Applied Mathematics*, 2015.

[14] Mario Botsch, Leif Kobbelt, Mark Pauly, Pierre Alliez, Bruno Lévy, *Polygon Mesh Processing*, 2010.

[15] P. G. Ciarlet, *The Finite Element Method for Elliptic Problems*, 1978.

[16] Daniel Cremers, *Variational Methods in Computer Vision*, (https://vision.in.tum.de/teaching/online/cvvm)

[17] Lawrence Evans, *Partial Differential Equations*, 2010.

[18] Documentation for DOLFIN-1.5.0 (Python), (https://fenicsproject.org/olddocs/dolfin/1.5.0/python/index.html)

[19] Anders Logg, Kent-Andre Mardal, Garth N. Wells (editors), *The FEniCS book*, 2012.

[20] E. Ward Cheney, *Introduction to Approximation Theory*, 1966.

[21] Jos Stam, *Flows on surfaces of arbitrary topology*, 2003.

[22] David Ham, Finite Element Course (Imperial College London, 2013-2014), http://wp.doc.ic.ac.uk/spo/finite-element/

[23] Howard Elman, David Silvester, Andy Wathen, *Finite Elements and Fast Iterative Solvers, with Applications in Incompressible Fluid Dynamics, 2nd edition*, 2014.

[24] Gilbert Strang, *A Framework for Equilibrium Equations*, 1988.

[25] Susanne C. Brenner, L. Ridgway Scott, *The Mathematical Theory of Finite Element Methods*, 2008.

[26] Gene H. Golub, Charles F. Van Loan, *Matrix Computations, third edition*, 1996.

[27] Jonathan Richard Shewchuk, *Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator*, 1996.

[28] Hang Si, *TetGen: A quality tetrahedral mesh generator and a 3D Delaunay triangulator*, 2015.

[29] Joseph O'Rourke, *Computational Geometry in C*, 1998.

[30] Balay et al., PETSc web page, https://petsc.org/, 2021.

[31] Conrad Sanderson, Ryan Curtin, *Armadillo: a template-based C++ library for linear algebra*, 2016.

[32] Gaël Guennebaud, *Eigen: A C++ linear algebra library*, 2013.

[33] Nicholas Sharp, Keenan Crane, and others, Geometry Central (surface mesh library), www.geometry-central.net, 2019.

[34] Daniel Sieger, Mario Botsch, The Polygon Mesh Processing Library, http://www.pmp-library.org, 2020.

[35] Leif Kobbelt, OpenMesh (surface mesh library), https://www.graphics.rwth-aachen.de/software/openmesh/, 2021.