The finite element method for the Navier-Stokes equations

Lucas Payne

October 6, 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

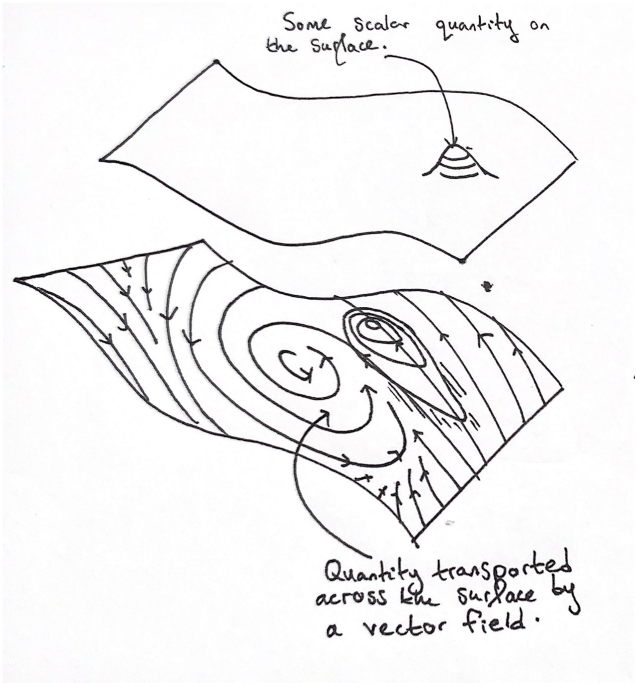
2.1 Introduction

— 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)).

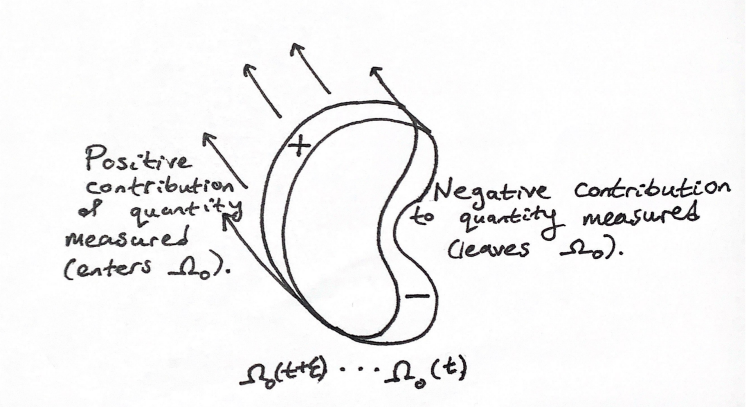
2.2.2 The Reynolds transport theorem

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

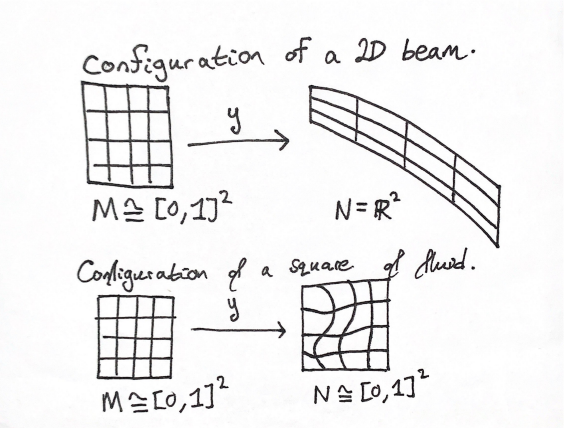
Transport equations are just one notion of "physical motion" in a continuum model. These transport equations, with prescribed flux and source functions, determine a continuous process on a fixed domain M. These conserved quantities are time-varying maps from M to some measurement space of scalars or 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

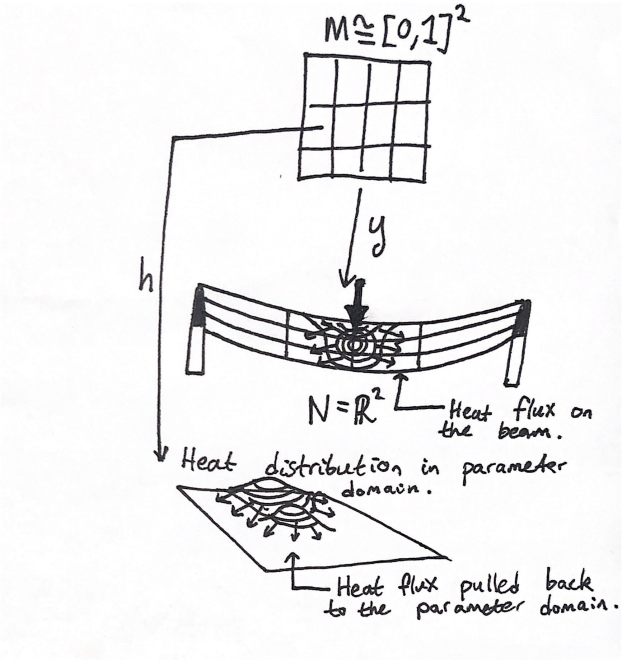
We may consider the manifold M as the parametric domain of some points living in an ambient manifold N. We will call this the "position map"

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

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.

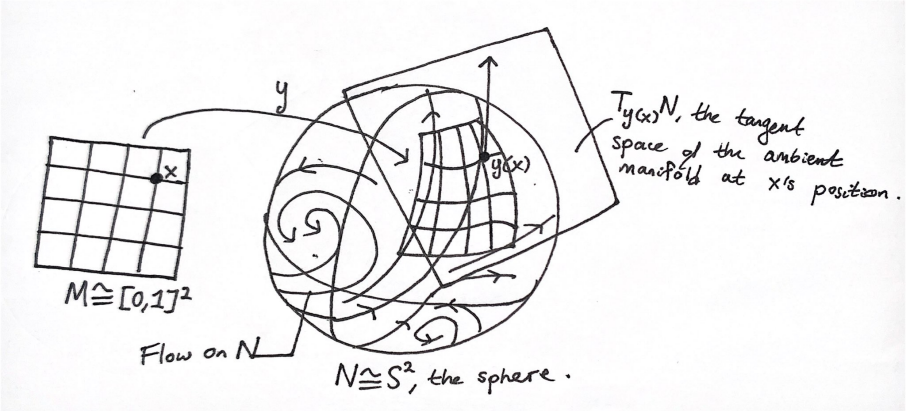


Although this model is so far hopelessly incomplete, we can see that position maps and transport equations are fundamental tools used for modelling the *geometry* of a problem.

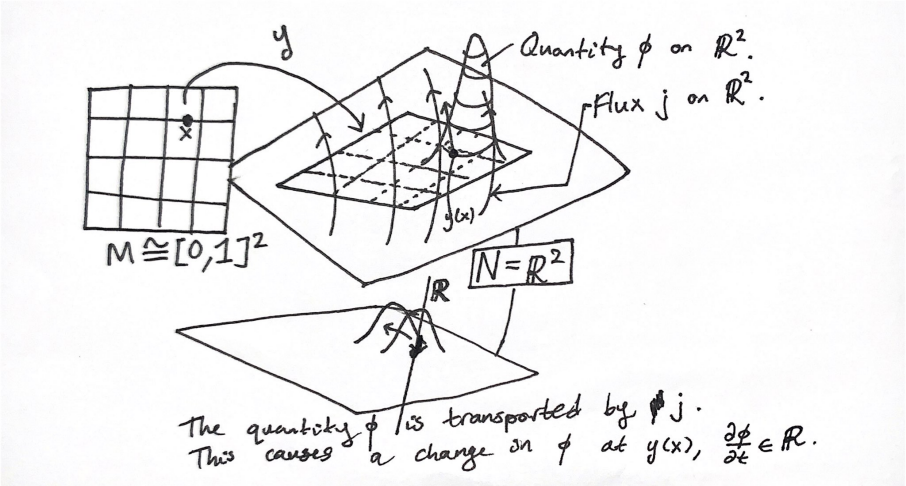
2.3.2 Velocity

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

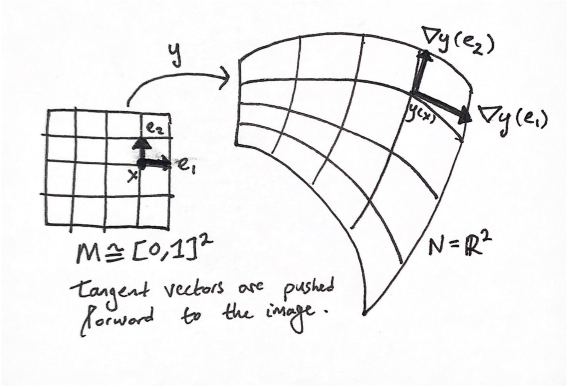
We may think of the mechanics of a particle as a special case of our continuum model, where M is a single point. In this case we have only one $x \in M$, so we cannot vary x. However, for a continuum parameter domain M we can take derivatives with respect to our parameter as well as time. We can extract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

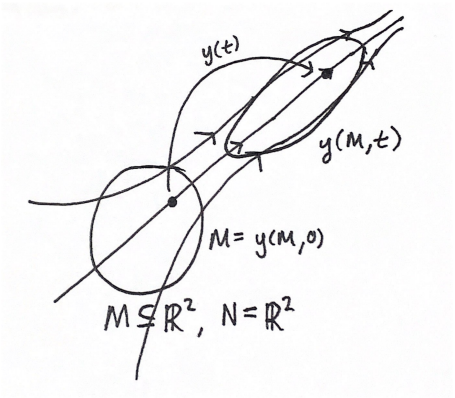
Letting u be our Eulerian velocity as defined above, we may express the position map through an ODE,

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

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

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

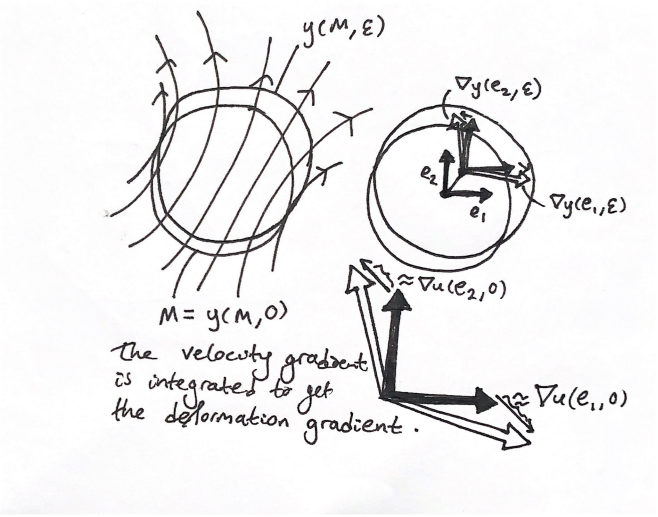
For example, given an initial disc in $\mathbb{R}^2$, we may prescribe a constant Eulerian velocity field u and see how the original disc is "mixed".



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

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

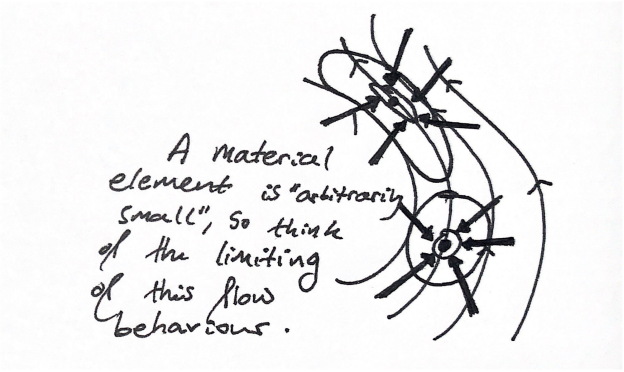
We define the *material derivative* to be

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

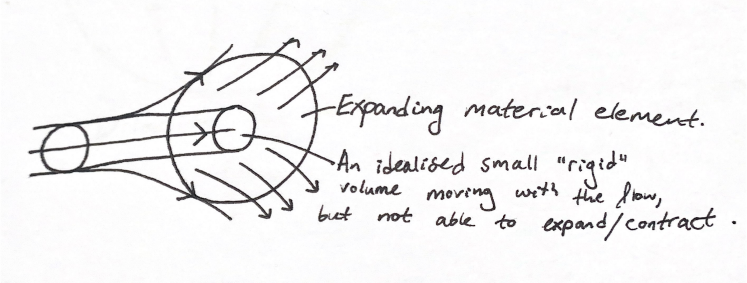
It is a convention to leave the vector field u implicit, as material derivatives are usually taken with respect to the velocity field. This derivative (2.22) will turn out to measure "per-volume" quantities.

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 [23]). 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)

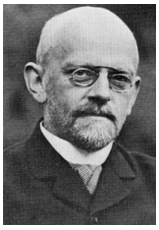
Chapter 4

Galerkin methods

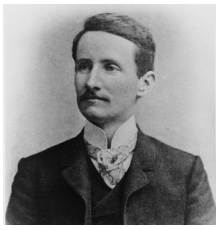
4.1 Introduction

Continuum mechanics provides the foundation for understanding of fluid motion, and the Navier-Stokes equations for a Newtonian fluid are the primary model for the prediction of fluid behaviour in engineering. The domain and initial/boundary conditions can be arbitrarily complex. For example, fluid motion is often simulated through a digital surface model of a real-world object or system created in computer-aided design software. On the other side of the coin, fluid models are very often used in modern film, both live action and animated, and all relevant geometry can be adjusted by artists for a desired visual effect.

Peter Dirichlet
(1805–1859)

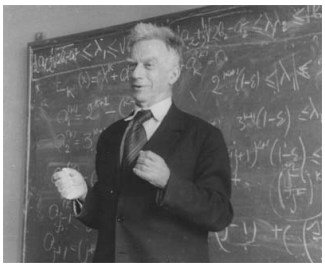


David Hilbert
(1862–1943)

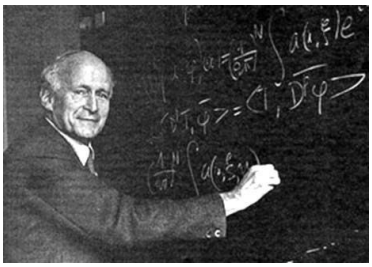


Walther Ritz
(1878–1909)

Boris Galerkin
(1871–1945)



Sergei Sobolev (1908–1989)



Laurent Schwartz (1915–2002)

Figure 4.1: Some key figures whose work precipitated modern developments in PDEs.

In applications, the Navier-Stokes equations are primarily solved on a computer, and a computer works with both finite data and finite precision. We will solve these equations by a *Galerkin method*. Galerkin methods are particularly amenable to computer implementation, so much so that the whole process of geometric modelling, boundary condition handling, residual minimisation, and solution reconstruction, can be automated ([41], [20], [19]). We begin by deriving two instances of Galerkin methods for Poisson's equation, performed on an irregular 2D domain to emphasize that the methods trivially extend to complex geometries. A computer implementation is then presented, which generates visualisations and quality metrics for the results.

The Galerkin method

The Galerkin method was established by engineer and mathematician Boris Galerkin (1871–1945) in a 1915 paper containing an approximation method for the biharmonic equation of classical beam theory [39]. Briefly, the principle of Galerkin is to choose a space of approximations, today called "test functions", and solve for the test function which minimises a residual error with respect to some function space norm. This norm is induced by a choice of what are now called the "trial functions". If the trial functions are the same as the test functions, the residual is minimised with respect to the Euclidean norm.

As well as proving highly useful in numerical applications, the methods of Galerkin (and preceding work by mathematicians such as Walther Ritz) proved important to further developments in the mathematical theory of PDEs. In a 1934 talk named "Generalized Solutions to the Wave Equation", mathematician Sergei L. Sobolev (1908–1989) precipitated the development of generalised functions by Laurent Schwartz (1915-2002), integral to the modern study of PDEs ([40], [38]). These developments are clarifications on the question: What do we mean when we speak of a solution to a PDE from physics? Galerkin's method is much closer to the answer than, for example, the finite difference method.

The finite element method

4.2 The equation to solve: The Poisson equation

Among the fundamental PDEs are the heat equation

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

and Poisson's 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 Poisson's 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.

4.2.1 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

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

Thinking about (4.4) leads to the finite difference method, historically the first and still very important in applications. By forming secant approximations of the derivatives over a regular grid, a linear system is formed, and the approximate solution is solved for as a function of this grid. 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 coefficients of, for example, hat basis functions. Yet a finite difference discretization of Poisson equation (4.1) may not take this into account at all. One might feel that limits have been taken too soon.

4.2.2 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),

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

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. To continue with this idea, we must write Poisson's equation (4.1) in integral form.

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 [17]. 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

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

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 Poisson's equation (4.1). In integral form, "undoing" the application of Stokes' theorem above, Poisson's equation is

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

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.

So how do we discretize Poisson's equation?

Even starting with (4.9), there are many routes to take to discretization. Notice that there are two variables in (4.9), although one is bound over universal quantification: the control volume Ω_0 and the function h. The general Galerkin method very naturally appears when the "space" Ω_0 resides in and the space h resides in are reduced such that the equations are both finite dimensional and well-determined. Firstly, let's derive the conceptually simplest discretization method of these spaces, the finite volume method.

4.3 Discretizing Poisson's equation by finite volumes

—NOTE: H^1 May be wrong. —NOTE: Should discuss interpretation in terms of residual minimisation with respect to some norm.

Equation (4.9) is quantified over an infinite number of control volumes Ω_0. These control volumes are in the interior of Ω, as the solution at the boundary Γ is specified by the Dirichlet boundary condition $h|_\Gamma = h_\Gamma$, and there is no flux information across Γ. A simple idea is to choose a finite number of control volumes $\Omega_1, \cdots, \Omega_n$ (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 a finite dimensional space of approximations Φ^*.

We begin by discussing the general construction of a finite volume method, but this will soon be made concrete with a simple, specific example of a finite volume discretization.

4.3.1 The Dirichlet boundary condition and the test space

The Dirichlet boundary condition $h|_\Gamma = h_\Gamma$ specifies the boundary values for any solution. However, the finite dimensional space Φ^* may not be able to exactly represent the boundary function h_Γ, and therefore the first step is to approximate h_Γ (for which there are many options, for example projection in the L^2-norm or interpolation across a finite sampling of boundary points [22]). Choosing some $\phi_\Gamma \in \Phi^*$ with

$$\phi_\Gamma|_\Gamma \approx h_\Gamma,$$

we can define an "interior" function space

$$\Phi = \{h^* \in \Phi^* \mid h|_\Gamma = 0\},$$

and the approximate solution $\tilde{h}$ will then be

$$\tilde{h} = \phi + \phi_\Gamma$$

for some $\phi \in \Phi$. In the finite volume and finite element literature the space Φ is called the test space. Since we selected n control volumes $\Omega_1, \cdots, \Omega_n$, the approximating space Φ^* must be chosen such that Φ is n-dimensional. We can choose a basis for the test space,

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

The problem is now to find a vector of coefficients of these test functions, $\hat{h} = (h_1, \cdots, h_n)^T$, and reconstruct the solution as

$$\tilde{h} = \phi + \phi_\Gamma,$$

where ϕ is the "interior variation"

$$\phi = \Phi \cdot \hat{h} \coloneqq h_1\phi_1 + \cdots + h_n\phi_n.$$

4.3.2 Forming a linear system

The integral-conservation-law form (4.9) of Poisson's equation (4.1) now directly indicates a method of approximation. Letting h be approximated by $\tilde{h} = \phi + \phi_\Gamma$ (where $\phi \in \Phi$ is unknown and $\phi_\Gamma \in \Phi^*$ is known), and restricting the flux integrals to the control volumes $\Omega_1, \cdots, \Omega_n$, the conservation law (4.9) becomes

$$\oint_{\partial\Omega_j} -\nabla(\phi + \phi_\Gamma) \cdot \hat{n}\, dx = \int_{\Omega_j} g\, dx \qquad j = 1, \cdots, n. \tag{4.10}$$

The "interior variation" $\phi = \Phi \cdot \hat{h}$ is the only unknown, so we can move all knowns to the right-hand-side and expand ϕ in terms of the basis test functions $\phi_1, \cdots, \phi_n$:

$$\oint_{\partial\Omega_j} -\nabla\left(\sum_{i=1}^{n} h_i\phi_i\right) \cdot \hat{n}\, dx = \int_{\Omega_j} g\, dx + \oint_{\partial\Omega_j} \nabla\phi_\Gamma \cdot \hat{n}\, dx \qquad j = 1, \cdots, n. \tag{4.11}$$

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

$$\sum_{i=1}^{n} h_i \oint_{\partial\Omega_j} -\nabla\phi_i \cdot \hat{n}\, dx = \int_{\Omega_j} g\, dx + \oint_{\partial\Omega_j} \nabla\phi_\Gamma \cdot \hat{n}\, dx \qquad j = 1, \cdots, n. \tag{4.12}$$

It is seen here that there must be some restrictions on the ϕ_i and ϕ_Γ. Formally, they must be in the Sobolev space $H^1(\Omega)$ (— NOTE: This is probably wrong, as the integrals are over internal cell boundaries and the boundary Γ. Need to read about H^{div}) 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. The system (4.12) can be written in matrix form as

$$\begin{aligned} A\hat{h} &= \begin{bmatrix} \oint_{\partial\Omega_1} -\nabla\phi_1 \cdot \hat{n}\, dx & \cdots & \oint_{\partial\Omega_1} -\nabla\phi_n \cdot \hat{n}\, dx \\ \vdots & & \vdots \\ \oint_{\partial\Omega_n} -\nabla\phi_1 \cdot \hat{n}\, dx & \cdots & \oint_{\partial\Omega_n} -\nabla\phi_n \cdot \hat{n}\, dx \end{bmatrix} \begin{bmatrix} h_1 \\ \vdots \\ h_n \end{bmatrix} \\ &= \begin{bmatrix} \int_{\Omega_1} g\, dx + \oint_{\partial\Omega_1} \nabla\phi_\Gamma \cdot \hat{n}\, dx \\ \vdots \\ \int_{\Omega_n} g\, dx + \oint_{\partial\Omega_n} \nabla\phi_\Gamma \cdot \hat{n}\, dx \end{bmatrix} = \hat{f}. \end{aligned} \tag{4.13}$$

This system is solved for the coefficients of combination $\hat{h} = (h_1 \cdots h_n)^T$, and the approximate solution is constructed as $\tilde{h} = \Phi \cdot \hat{h} + \phi_\Gamma$. If the control volumes $\Omega_1, \cdots, \Omega_n$ and test space $\Phi = \operatorname{span}\{\phi_1, \cdots, \phi_n\}$ are chosen well, this linear system will be nonsingular and hopefully well-conditioned. If the chosen control volumes join together in the interior of Ω like pieces of a puzzle, this gives a conservative system of balanced fluxes, but it is another question whether the approximation $\tilde{h}$ is good.

4.3.3 A piecewise linear triangulation scheme

Possibly the simplest scheme is to triangulate Ω as $\bigcup_i T_i$, such that there are n interior vertices $p_1, \cdots, p_n$, and n_Γ boundary vertices $p_1^\Gamma, \cdots, p_{n_\Gamma}^\Gamma$, and n_T triangles $T_1, \cdots, T_{n_T}$. An example triangulation of an ellipse-shaped domain is shown in figure 4.2.

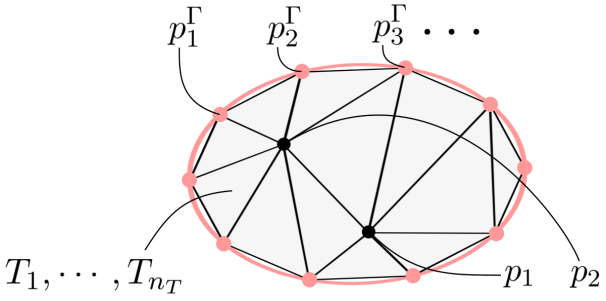


Figure 4.2: The domain Ω is partitioned into $n_T = 12$ triangular cells $T_1 \cdots T_{12}$. The boundary is sampled at $n_\Gamma = 10$ points $p_1^\Gamma, \cdots, p_{10}^\Gamma$. There are $n = 2$ interior points, p_1 and p_2.

The test space

Firstly, there is a clear choice of function space Φ^*, which satisfies the H^1 condition (NOTE: Is H^1 necessary? Need to check conditions for finite volumes), consisting of piecewise linear "hat" basis functions at each interior or boundary vertex,

$$\Phi^* = \text{span}\left\{\text{Hat}(p_1), \cdots, \text{Hat}(p_n), \text{Hat}(p_1^\Gamma), \cdots, \text{Hat}(p_{n_\Gamma}^\Gamma)\right\}.$$

The hat function at a vertex has value 1 at that vertex and value 0 at its neighbours. The test space (which is zero on the boundary Γ) is then

$$\Phi = \text{span}\left\{\text{Hat}(p_1), \cdots, \text{Hat}(p_n)\right\} = \text{span}\left\{\phi_1, \cdots, \phi_n\right\},$$

where $\phi_i \coloneqq \text{Hat}(p_i)$. One of these basis functions for the ellipse-shaped domain is shown in figure 4.3.

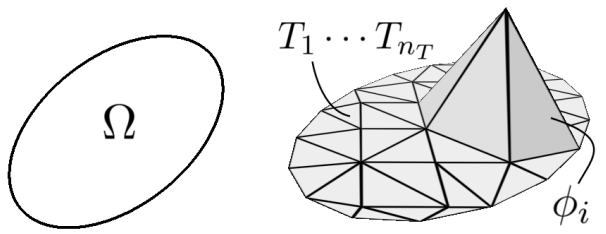


Figure 4.3: The domain Ω is partitioned into triangular cells $T_1 \cdots T_{n_T}$. One example of a hat basis function ϕ_i is shown, where the vertical axis is the basis function value.

Approximating the boundary values

One simple approximation of $h|_\Gamma = h_\Gamma$ is

$$\phi_\Gamma = h_\Gamma(p_1^\Gamma)\mathrm{Hat}(p_1^\Gamma) + \cdots + h_\Gamma(p_{n_\Gamma}^\Gamma)\mathrm{Hat}(p_{n_\Gamma}^\Gamma), \tag{4.14}$$

which is a piecewise linear interpolation when restricted to the boundary Γ. The approximation ϕ_Γ for some boundary function h_Γ is shown in figure 4.4.

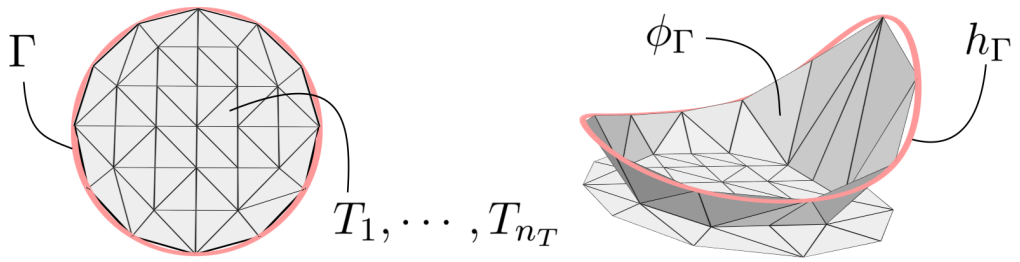


Figure 4.4: A circle domain Ω with boundary Γ is tessellated by triangles $T_1, \cdots, T_{n_T}$. The boundary function h_Γ is approximated by ϕ_Γ, a linear combination of hat functions centred at the boundary vertices.

This approximation ϕ_Γ is added to the "interior variation" $\phi = \Phi \cdot \hat{h}$ to construct a possible solution. When a linear solver is given the linear system (4.13), it will find the interior variation ϕ which minimises some residual norm. An iterative solver, for example, can then be thought of as progressively varying a thin membrane attached to the boundary curve, until the residual error is sufficiently small. This construction is shown in figure 4.5.

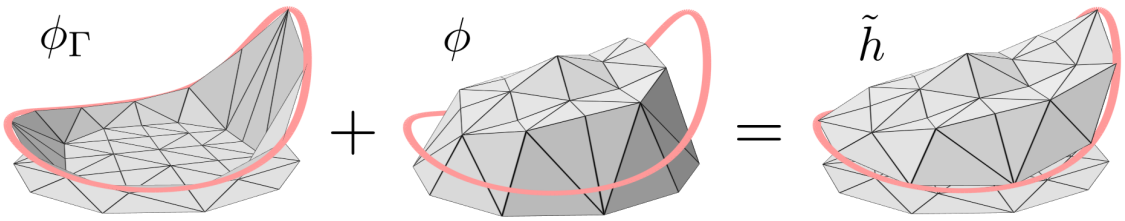


Figure 4.5: The fixed boundary approximation ϕ_Γ is added to a variable interior variation ϕ to give a possible approximate solution $\tilde{h}$. This possible solution can be checked by integration against the trial control volumes $\Omega_1, \cdots, \Omega_n$.

Choosing a finite number of control volumes

If we choose some characteristic "triangle centre" for each triangle, then we can associate to the interior vertices $p_1, \cdots, p_n$ a set of domains $\Omega_1, \cdots, \Omega_n$. Each Ω_i is determined by the polygon which joins the centres of the triangles incident to p_i. 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 [31]. The resulting cell decompositions are displayed in figure 4.6.

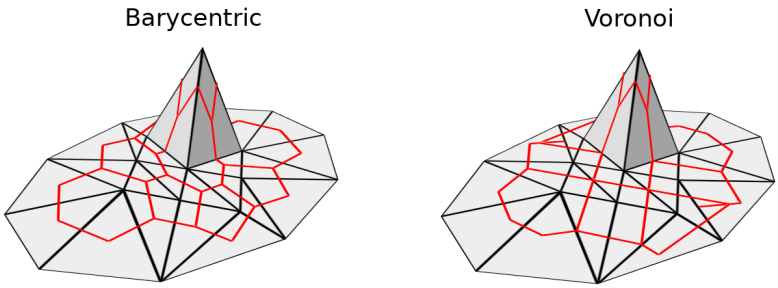


Figure 4.6: The domain Ω is partitioned into triangular cells $T_1 \cdots T_{n_T}$. Flux integrals are taken over n polygonal cells, one for each internal vertex p_i, for example using triangle barycentres or circumcentres.

Notice that, in the circumcentre case, the thin triangles do not contain their circumcentre, and therefore each cell boundary is not necessarily contained in a single fan of triangles. Notice also that the Voronoi polygon passes through the midpoints of each edge incident to p_i. This midpoint property will soon prove useful in computing integrals, but the resulting exposition will be clearer if we can also assume that the cell boundary *is* contained in a single fan of triangles. Therefore we define the *mixed Voronoi cell* [14], where a wayward circumcentre is pulled back to the opposite edge.

(draw this)

The scheme as described so far, consisting of a triangle mesh, hat test functions, and barycentric, Voronoi, or mixed Voronoi control volumes, has found some success, especially in the domain of geometry processing ([14], [15]). We will work with the mixed Voronoi cells.

4.3.4 Computing the linear system by closed-form integration

Now that we have chosen a specific finite volume discretization, given a domain triangulation we can compute the coefficients of the linear system (4.13) and solve for $\hat{h}$, giving the approximate solution $\tilde{h} = \phi_\Gamma + \Phi \cdot \hat{h}$. By fortuitous circumstances, these integrals actually have simple closed forms. (This is with the exception of the heat source integrals $\int_{\Omega_j} g\, dx$, which may require numerical integration if the heat source g is not restricted to Φ^*.)

It will be more convenient to start again with the trial flux integral

$$\oint_{\partial\Omega_j} -\nabla\tilde{h} \cdot \hat{n}\, dx, \tag{4.15}$$

rather than with the expanded linear system (4.13). By linearity, the ∇ in $\nabla\tilde{h}$ "falls through" the expansion of $\tilde{h}$:

$$\nabla\tilde{h} = \nabla\phi_\Gamma + \nabla\phi = \sum_{i=1}^{n_\Gamma} h_\Gamma(p_i^\Gamma)\nabla\phi_i^\Gamma + \sum_{i=1}^{n} h_i \nabla\phi_i.$$

By linearity the integral (4.15) becomes

$$\sum_{i=1}^{n_\Gamma} \left[h_\Gamma(p_i^\Gamma) \oint_{\partial\Omega_j} -\nabla\phi_i^\Gamma \cdot \hat{n}\, dx \right] + \sum_{i=1}^{n} \left[h_i \oint_{\partial\Omega_j} -\nabla\phi_i \cdot \hat{n}\, dx \right]. \tag{4.16}$$

Now consider a certain flux integral (4.15) over control volume Ω_j. We will reexpress the above integral in "local" terms. Let $v = p_j$ be the interior vertex corresponding to the

control volume Ω_j, and denote by $v_1, \cdots, v_k$ its adjacent vertices (either in the interior or on the boundary), and by $T_1^v \cdots T_k^v$ its adjacent triangles. Triangle T_l^v joins v, v_l, v_{l+1} where addition wraps around from k to 1. As it will be necessary to relate these local indices (denoted l) to the global vertex indices (denoted i), define an index map by

$$I(l) = \text{the global index } i \text{ of the } p_i \text{ corresponding to adjacent vertex } v_l,$$

which is valid only when v_l is an interior vertex. The vertex v corresponds to a single trial flux integral over the surrounding Voronoi cell Ω_j. As the hat functions have compact support, Ω_j only intersects the domains of the hat function ϕ_j and those corresponding to the adjacent vertices. Name these basis functions $\phi_j = \phi_v$ and $\phi_{v_1}, \cdots \phi_{v_k}$. Then the local form of the flux integral (4.15) is

$$\oint_{\partial\Omega_j} -\nabla\tilde{h} \cdot \hat{n}\, dx = h_j \oint_{\partial\Omega_j} -\nabla\phi_v \cdot \hat{n}\, dx + \sum_{l=1}^{k} h_{v_l} \oint_{\partial\Omega_j} -\nabla\phi_l^v \cdot \hat{n}\, dx, \tag{4.17}$$

where h_{v_i} is defined as

$$h_{v_i} = \begin{cases} h_\Gamma(p_i^\Gamma) & \text{if } v_i \text{ is a boundary vertex,} \\ h_{I(v_i)} & \text{if } v_i \text{ is an interior vertex.} \end{cases}$$

The ϕ_v and $\phi_{v_1}, \cdots, \phi_{v_k}$ are hat functions which are linear when restricted to each triangle $T_1^v, \cdots, T_k^v$. Therefore the gradients $\nabla\phi_v$ and $\nabla\phi_{v_1}, \cdots, \nabla\phi_{v_k}$ are constant on those triangles. We can break the closed flux integrals in (4.17) into flux integrals restricted to each triangle adjacent to v, giving

$$\oint_{\partial\Omega_j} -\nabla\tilde{h} \cdot \hat{n}\, dx = \sum_{t=1}^{k} \left[\int_{\Omega_j \cap T_t^v} -\nabla\phi_v \cdot \hat{n}\, dx + \sum_{l=1}^{k} h_{v_l} \int_{\Omega_j \cap T_t^v} -\nabla\phi_l^v \cdot \hat{n}\, dx \right]. \tag{4.18}$$

We can simplify matters by focusing on just one adjacent triangle $T_t^v = T$, between vertices v, v_t, v_{t+1}. This triangle T only overlaps with the domains of the hat functions centred at v, v_t, v_{t+1}, so we now only need to compute

$$\int_{\Omega_j \cap T} -\nabla\phi_v \cdot \hat{n}\, dx \quad \text{and} \quad \int_{\Omega_j \cap T} -\nabla\phi_{v_t} \cdot \hat{n}\, dx \quad \text{and} \quad \int_{\Omega_j \cap T} -\nabla\phi_{v_{t+1}} \cdot \hat{n}\, dx.$$

This is simple because the gradients are constant on $\Omega_j \cap T_t^v$.

(figure)

The Voronoi polygon joins the midpoints $m' = \frac{1}{2}(v + v_t)$ and $m'' = \frac{1}{2}(v + v_{t+1})$, taking a detour to a point within T.

(figure)

However, since the gradient vector fields are constant, they are conservative, and we can simplify the domain $\Omega_j \cap T$ to the line segment L between the midpoints m' and m''. Since the gradients are constant on T, these integrals are the length of L multiplied by one "sample" flux across L. This can be done easily, by noticing L is parallel with and half the length of the line segment between v_t and v_{t+1}. We can then compute the integrals as:

$$\frac{1}{2}\nabla\phi_v \cdot (v_{t+1} - v_t)^\perp, \quad \text{and} \quad \frac{1}{2}\nabla\phi_{v_t} \cdot (v_{t+1} - v_t)^\perp, \quad \text{and} \quad \frac{1}{2}\nabla\phi_{v_{t+1}} \cdot (v_{t+1} - v_t)^\perp, \tag{4.19}$$

where $(v_{t+1} - v_t)^\perp$ is the vector $v_{t+1} - v_t$ rotated 90 degrees anticlockwise. We now only need the constant gradients on T. This requires some simple geometry. As this construction is the same for each vertex, consider an anticlockwise ordering of v, v_t, v_{t+1}, named v', v'', v'''.

(figure of a rectangle drawn around the triangle, and the perpendicular direction, with width and height labels)

We must normalize this vector and then divide by the height of this rectangle, as the slope is inversely proportional to the rectangle height. This gives the gradient

$$\nabla\phi_{v'} = \frac{(v''' - v'')^\perp}{\|v''' - v''\|} \Big/ h = \frac{(v''' - v'')^\perp}{wh}.$$

Finally, $\text{Area}(T) = wh/2$, so we can write the gradients as

$$\nabla\phi_v = \frac{(v_{t+1} - v_t)^\perp}{2\text{Area}(T)}, \quad \text{and} \quad \nabla\phi_{v_t} = \frac{(v - v_{t+1})^\perp}{2\text{Area}(T)}, \quad \text{and} \quad \nabla\phi_{v_{t+1}} = \frac{(v_t - v)^\perp}{2\text{Area}(T)}. \tag{4.20}$$

Plugging the gradients (4.20) into the closed form integrals (4.19), we get

$$\begin{aligned}
\int_L -\nabla\phi_v \cdot \hat{n}\, dx &= \frac{1}{4\text{Area}(T)}\|v_{t+1} - v_t\|^2, \\
\int_L -\nabla\phi_{v_t} \cdot \hat{n}\, dx &= \frac{1}{4\text{Area}(T)}(v - v_{t+1}) \cdot (v_{t+1} - v_t), \\
\int_L -\nabla\phi_{v_{t+1}} \cdot \hat{n}\, dx &= \frac{1}{4\text{Area}(T)}(v_t - v) \cdot (v_{t+1} - v_t),
\end{aligned}$$

where the 90-degree rotation has been removed, as it preserves the dot product. With these closed forms, we can express the original integral as

$$\begin{aligned}
&\oint_{\partial\Omega_j} -\nabla\tilde{h} \cdot \hat{n}\, dx \\
&= \sum_{t=1}^{k} \frac{1}{4\text{Area}(T_t^v)} \left[h_j\|v_{t+1} - v_t\|^2 + h_{v_t}(v - v_{t+1}) \cdot (v_{t+1} - v_t) + h_{v_{t+1}}(v_t - v) \cdot (v_{t+1} - v_t) \right].
\end{aligned} \tag{4.21}$$

This directly indicates an effective algorithm for constructing the linear system.

Integrating the source function

One more thing is needed, however. The right-hand-side vector of (4.13) contains source integrals

$$\int_{\Omega_j} g\, dx,$$

so we require some numerical integration.

The simplest choice is a "rectangle rule", where we sample the source function g at the interior vertex p_j, then assume that g is constant over Ω_j. This gives the approximation

$$\int_{\Omega_j} g\, dx \approx g(p_j)\text{Area}(\Omega_j). \tag{4.22}$$

— However, a linear quadrature will give much better convergence results.

(figure)

4.3.5 The resulting matrix assembly algorithm

Using the expression (4.21), the linear system (4.13) can be constructed by iterating over each interior vertex, then each adjacent triangle, and computing these closed form integrals. Each adjacent triangle corresponds to two vertices, and for each of these vertices, a value is placed in either the system matrix or adjusting the right-hand-side, depending on whether the triangle vertex in the interior or on the boundary. Also, while iterating over the interior vertices, the corresponding source integral is approximated by (4.22), and the result is added to the right-hand-side vector. Pseudocode for this algorithm is given below.

Routine *Poisson_FVM_Matrix_Assembly*:
  Data: Domain Ω with boundary Γ.
    Heat source function $g : \Omega \to \mathbb{R}$.
    Dirichlet boundary function $h_\Gamma : \Gamma \to \mathbb{R}$.
    Triangulation of Ω consisting of:
      Interior vertices $p_1, \cdots, p_n$.
      Boundary vertices $p_1^\Gamma, \cdots, p_{n_\Gamma}^\Gamma$.
      Triangles $T_1, \cdots, T_{n_T}$.
  Result: The coefficients $(A, \hat{f})$ of the system $A\hat{h} = \hat{f}$ (4.13), which can be solved by the caller for $\hat{h}$ in order to reconstruct an approximate solution $\tilde{h} : \Omega \to \mathbb{R}$ of the boundary value problem (4.2),
    $$-\Delta h = g,\ h|_\Gamma = h_\Gamma.$$
  $A \leftarrow n \times n$ zero matrix;
  $\hat{f} \leftarrow$ length n zero vector;
  for $j = 1, \cdots, n$ **do**
    Let v denote the interior vertex p_j.
    Let $v_1, \cdots, v_k$ denote the vertices adjacent to v.
    Let $T_1^v, \cdots, T_k^v$ denote the triangles adjacent to v.
    (Triangle T_t joins vertices v, v_t, v_{t+1}.)
    for $t = 1, \cdots, k$ **do**
      $C \leftarrow \frac{1}{4\text{Area}(T)}$;
      $A[j,j] \leftarrow A[j,j] + C\|v_{t+1} - v_t\|^2$;
      if v_t *is on the boundary* **then**
        $\hat{f}[I(t)] \leftarrow \hat{f}[I(t)] - h_\Gamma(v_t)C(v - v_{t+1}) \cdot (v_{t+1} - v_t)$;
      else
        $A[j, I(t)] \leftarrow A[j, I(t)] + C(v - v_{t+1}) \cdot (v_{t+1} - v_t)$;
      end
      if v_{t+1} *is on the boundary* **then**
        $\hat{f}[I(t+1)] \leftarrow \hat{f}[I(t+1)] - h_\Gamma(v_{t+1})C(v_t - v) \cdot (v_{t+1} - v_t)$;
      else
        $A[j, I(t+1)] \leftarrow A[j, I(t+1)] + C(v_t - v) \cdot (v_{t+1} - v_t)$;
      end
    end
    $\hat{f}[j] \leftarrow \hat{f}[j] + \text{Control\_Volume\_Area}(\Omega_j) \cdot g(v)$;
  end
  return $(A, \hat{f})$;

Algorithm 1: Pseudocode for the described finite volume matrix assembly process for the Poisson boundary value problem (4.2).

4.3.6 The resulting finite volume algorithm

We now have an effective algorithm for approximating Poisson's equation:

1. Partition the 2D domain Ω into triangles $T_1, \cdots, T_{n_T}$. This determines the hat basis functions $\phi_1, \cdots, \phi_n$ and the mixed Voronoi cells $\Omega_1, \cdots, \Omega_n$.

2. Form the matrix and right-hand-side in (4.13) with the algorithm **??**.

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

Each of these steps is conceptually well-separated into the domains of mesh generation, matrix assembly, and numerical linear algebra. The above pseudocode **??** is a matrix assembly algorithm. It assumes that there is already a domain triangulation, and constructs the linear system (4.13). Although we have addressed the problem of matrix assembly, specific to this particular finite volume method, mesh generation and numerical linear algebra can be delegated to external libraries. The full solve algorithm is given below.

Routine *Solve_Poisson_FVM*:
Data: Domain Ω with boundary Γ.
Heat source function $g : \Omega \to \mathbb{R}$.
Dirichlet boundary function $h_\Gamma : \Gamma \to \mathbb{R}$.
Result: Approximate solution $\tilde{h} : \Omega \to \mathbb{R}$ of the boundary value problem (4.2),
$$-\Delta h = g,\ h|_\Gamma = h_\Gamma.$$
// Mesh generation
triangulation $\leftarrow$ Generate_Triangulation(Ω);
// Matrix assembly
$(A, \hat{f}) \leftarrow$ Poisson_FVM_Matrix_Assembly(Ω, g, h_Γ, triangulation);
// Linear solve
Solve $A\hat{h} = \hat{f}$ for $\hat{h}$.
// Solution reconstruction
$\tilde{h} \leftarrow \sum_{j=1}^{n} h_j \mathrm{Hat}(p_j) + \sum_{j=1}^{n_\Gamma} h_\Gamma(p_j^\Gamma)\mathrm{Hat}(p_j^\Gamma)$;
return $\tilde{h}$;

The routine **Generate_Triangulation** is provided by a mesh generator, and the **Solve** routine is provided by a numerical linear algebra library.

Mesh generation

Mesh generation is a huge field. However, there are only two essentials for the handling of 2D domains and piecewise linear triangulations — 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 [35], the Polygon Mesh Processing library [36], and OpenMesh [37].

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 volume mesh. The Triangle [29] library, for example, contains a single very efficient and robust C routine (consisting of 16k lines of highly optimized code) for performing Delaunay triangulations [31] on 2D domains, with the specific goal of creating robust finite element meshes.

Numerical linear algebra

The solution of large linear systems is a vast topic. Finite element solvers are typically clients of standard, robust linear and non-linear solver libraries, such as Argonne National Laboratory's PETSc libraries [32] and the smaller-scale C++ libraries Armadillo [33] and Eigen [34]. 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 solver access to the system coefficients when they are needed.

— Discuss sparsity here.

4.3.7 An implementation in C++

I have implemented the above algorithm in C++. This code uses a simple mesh generator based on the C library Triangle [29]. Triangle is given a list of points and boundary points, and generates a Delaunay triangulation [31]. This triangulation is then converted to a halfedge mesh data structure based on the ideas in the Geometry Central library [35]. The system is solved with a sparse LU factorisation method provided by the C++ Eigen library [34]. With these interfaces, the resulting C++ was almost a direct transliteration of the pseudocodes **??** and **??**.

4.3.8 Validating the method

In order to validate this algorithm, and in particular its C++ implementation described above, some simple test cases can be generated, using the "method of manufactured solutions" [21]. Firstly, suppose the exact solution is

$$h = x^2 - y^2.$$

From the chosen solution h, we can derive the boundary condition and heat source that would give that solution. Of course, the boundary condition must be $h_\Gamma = h|_\Gamma$.We can solve for the source term g by plugging h into Poisson's equation (4.1), giving

$$-\Delta\left(x^2 - y^2\right) = g \quad \Rightarrow \quad g = -(2 - 2) = 0.$$

This test case has been chosen as it degenerates to an instance of the simpler Laplace's equation, where there is no heat source:

$$-\Delta h = 0, \quad h|_\Gamma = x^2 - y^2 \quad \text{(The test problem)}. \tag{4.23}$$

For clarity, we use a simple square domain $\Omega = [-1, 1]^2$, and a uniform grid parameterized by value r [1]. Each square in this grid corresponds to two right-angled triangles in the triangulation [2]. To measure convergence, we use the L^2 norm (although others could be used), defined by

$$\left\|\tilde{h} - h\right\|_{L^2(\Omega)} = \sqrt{\int_{-1}^{1}\int_{-1}^{1} (\tilde{h}(x, y) - h(x, y))^2 \, dx \, dy}. \tag{4.24}$$

To validate the convergence of the method, simply increase h, generate the mesh, solve the test equation, then compute the L^2 error (4.24), and repeat. A log-log plot of the error for the simple test problem is shown in figure 4.7.

[1] Traditionally this kind of parameter is denoted h, which we are already using for the heat function.

[2] This is a rather uninteresting mesh which doesn't take advantage of Galerkin methods' ability to model complex geometries, but it is a difficult problem to parameterise a complex mesh by some r such that the error metric acts "nicely".

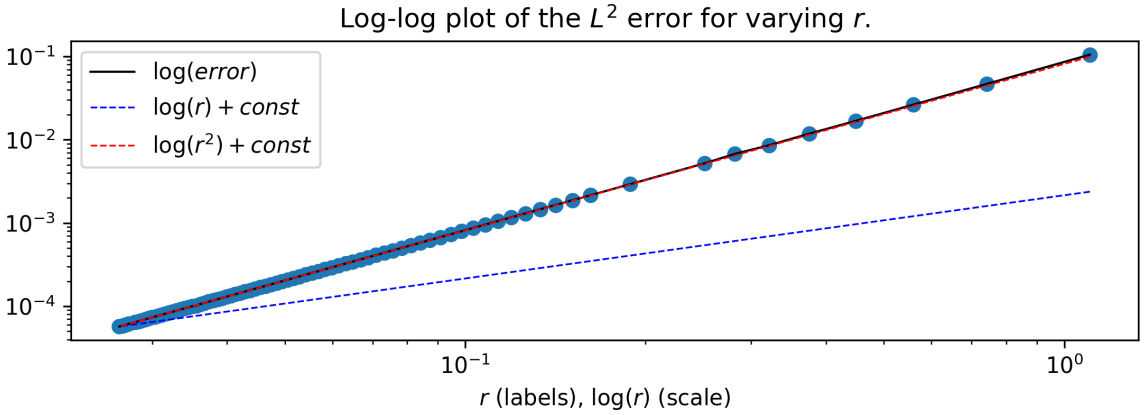


Figure 4.7: The L^2-norm error for the test problem (4.23), as parameter r varies.

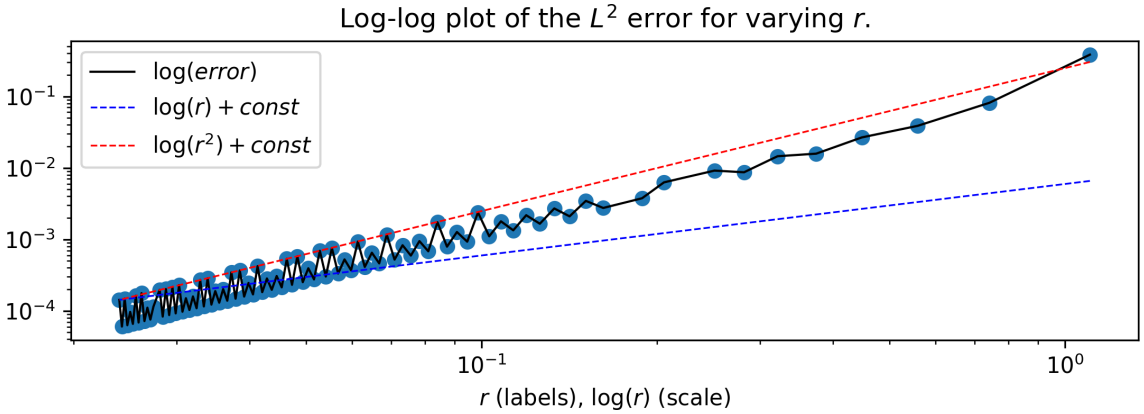


Figure 4.8: The L^2-norm error for the test problem (4.25), as parameter r varies.

It can be seen clearly that the convergence for test problem (4.23) is quadratic.

We should also try a test problem which does not degenerate to Laplace's equation. Suppose instead that the exact solution is

$$h = e^{-2(x^2+y^2)}.$$

The source term g is solved for as

$$-\Delta\left(e^{-2(x^2+y^2)}\right) = g \quad \Rightarrow \quad g = \left(8 - 16x^2 - 16y^2\right) e^{-2(x^2+y^2)},$$

giving the new test problem,

$$-\Delta h = \left(8 - 16x^2 - 16y^2\right) e^{-2(x^2+y^2)}, \quad h|_\Gamma = e^{-2(x^2+y^2)} \quad \text{(The test problem)}. \quad (4.25)$$

The log-log error plot is displayed in figure (4.8). In this case, there is still a clear quadratic convergence, but with some oscillation in quality as r decreases. This is likely due to the coarse numerical integration of the heat source (which is using the rectangle rule for simplicity). A visualization of these convergence results for problem (4.25) is given in figure 4.9. To emphasize the generality of the algorithm, figure 4.10 displays convergence for problem (4.23) on an irregularly-meshed disk.

4.4 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

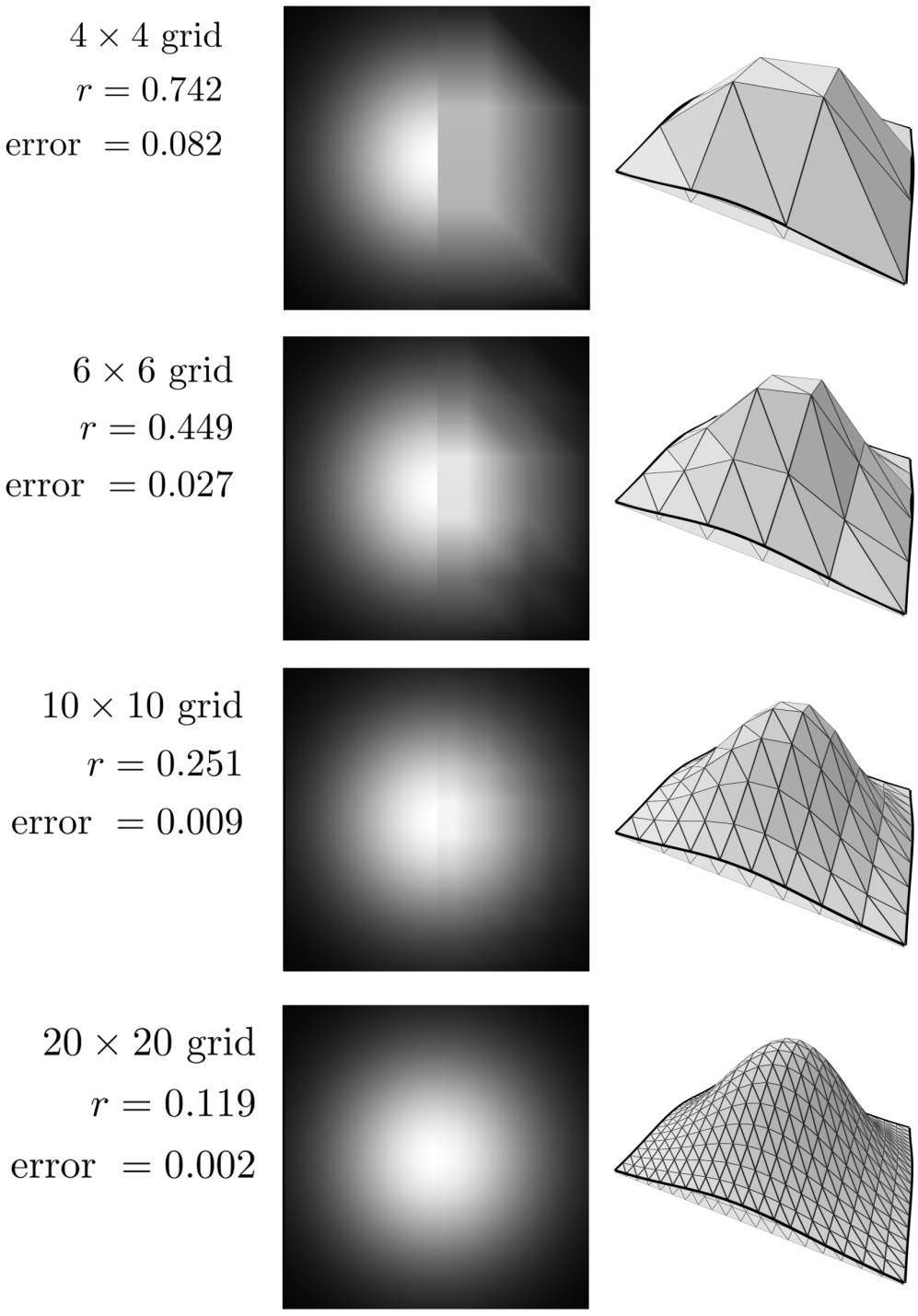


Figure 4.9: Approximate solutions to the test problem (4.25) with varying r. The middle squares contain plots of the exact solution (on the left rectangle), and the approximate solution (on the right rectangle). Black is $h = 0$, and white is $h = 1$. On the right, a 3D view of the solution and mesh is shown.

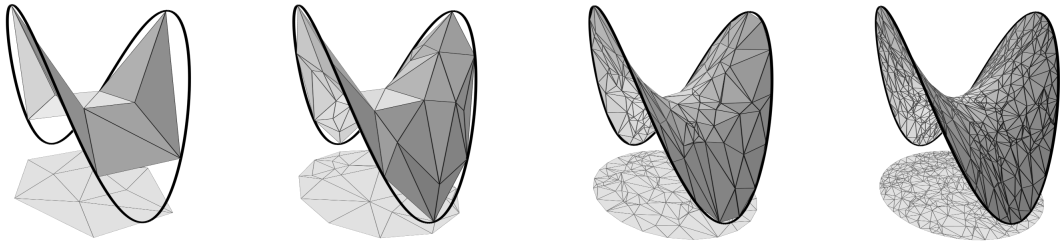


Figure 4.10: Approximate solutions to the test problem (4.23) with varying r. The domain is a unit disk, and r is reduced by introducing random points. The finite volume algorithm **??** works trivially with curved domains and irregular meshes.

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 (4.13) 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,

$$\int_{\partial\Omega_1} -\nabla \left(\sum_{i=1}^{n} h_i \phi_i \right) \cdot \hat{n} \, dx + \int_{\partial\Omega_2} -\nabla \left(\sum_{i=1}^{n} h_i \phi_i \right) \cdot \hat{n} \, dx = \int_{\Omega_1} g \, dx + \int_{\Omega_2} g \, dx \quad (4.26)$$

must hold. At first sight, (4.26) 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 (4.26) as a flux integral over a *formal sum* of domains,

$$\Omega_1 + \Omega_2.$$

We now have the equation

$$\int_{\partial\Omega_1 + \partial\Omega_2} -\nabla \left(\sum_{i=1}^{n} h_i \phi_i \right) \cdot \hat{n} \, dx = \int_{\Omega_1 + \Omega_2} g \, dx. \quad (4.27)$$

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 the 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 (4.12), 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 selection of control volumes. 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

$$\sum_{i=1}^{n} h_i \int_{\Omega} \nabla\phi_i \cdot \nabla\psi_j\, dx = \int_{\Omega} g\psi_j\, dx, \quad j = 1, \cdots, n, \tag{4.28}$$

and we see that we still only require the ϕ_i to be in $H^1(\Omega)$. We can see that equation (4.28) is very similar to the exact fluxes in (4.12), and indeed (4.28) gives a discrete linear system in much the same way. There is a real geometric sense in which (4.28) 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.

4.5 Discretizing Poisson's equation by finite elements

Equation (4.28) gives a discrete linear system

$$A\hat{h} = \begin{bmatrix} \int_\Omega \nabla\phi_1 \cdot \nabla\psi_1 \, dx & \cdots & \int_\Omega \nabla\phi_n \cdot \nabla\psi_1 \, dx \\ \vdots & & \vdots \\ \int_\Omega \nabla\phi_1 \cdot \nabla\psi_n \, dx & \cdots & \int_\Omega \nabla\phi_n \cdot \nabla\psi_n \, dx \end{bmatrix} \begin{bmatrix} h_1 \\ h_2 \\ \vdots \\ h_{n-1} \\ h_n \end{bmatrix} = \begin{bmatrix} \int_\Omega g\psi_1 \, dx \\ \int_\Omega g\psi_2 \, dx \\ \vdots \\ \int_\Omega g\psi_{n-1} \, dx \\ \int_\Omega g\psi_n \, dx \end{bmatrix} = \hat{g}. \quad (4.29)$$

This has quite a similar form to (4.13), but with boundary and cell integrals replaced by integrals over the entire domain. Again, if the control volumes $\Omega_1, \cdots, \Omega_n$ and test space $\Phi = \text{span}\{\phi_1, \cdots, \phi_n\}$ are chosen well, this linear system will be non-singular and hopefully well-conditioned.

4.5.1 Choosing a test and a trial space

Notice that the integrals in (4.29), in contrast to (4.13), 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 (4.29), 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 (4.29) 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 4.11.

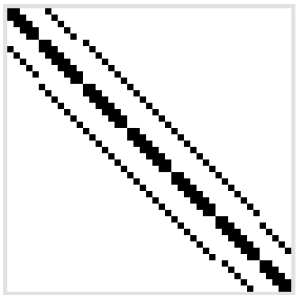


Figure 4.11: An example sparsity pattern for the finite element matrix of a 2D Poisson problem. White is zero and black is non-zero. The mesh has 61 vertices (16 on the boundary), and 104 triangles, resulting in a 45x45 system with 2025 entries, 197 non-zeros, and fill of 0.0973.

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 there are n nodal points and n_T triangles. Each nodal point p_i will be associated with a piecewise linear "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.

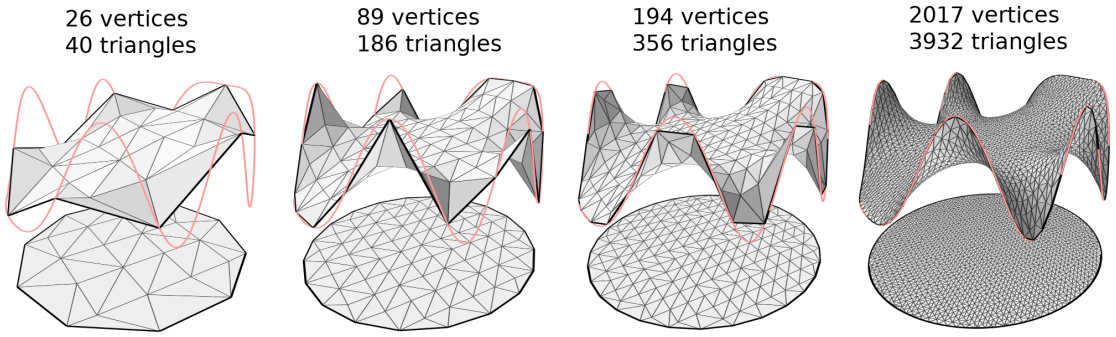


Figure 4.12: Text

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] M. Meyer, M. Desbrun, P. Schroder and A.H. Barr, *Discrete Differential-Geometry Operators for Triangulated 2-Manifolds*, 2003.

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

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

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

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

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

[21] — todo, *Solving PDEs in FEniCS with Python*, —.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

[38] Sergey Repin, *One hundred years of the Galerkin method*, 2017.

[39] Boris Galerkin, *Beams and plates. Series in some questions of elastic equilibrium of beams and plates (In Russian)*, 1915.

[40] Sobolev Institute of Mathematics, Sergei Sobolev, biographical web page, http://www.math.nsc.ru/conference/sobolev/english/About_Sobolev_SL.htm

[41] Florian Rathgeber, David A. Ham, and others, *Firedrake: automating the finite element method by composing abstractions*, 2016.