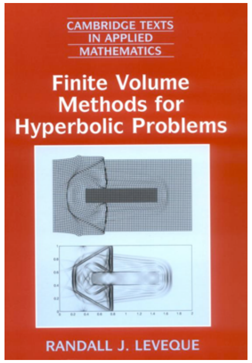
FYS-GEO4500 - Finite volume methods for geophysical fluid dynamics

Galen Gisler, Physics of Geological Processes
University of Oslo

galen.gisler@fys.uio.no
Fysikk byggningen, Rm 411A Vestfløy
+47 99898013

Course Outline


CAMBRIDGE TEXTS IN APPLIED MATHEMATICS
Finite Volume Methods for Hyperbolic Problems
RANDALL J. LEVEQUE

The book:

Finite Volume Methods for Hyperbolic Problems, by Randall J. Leveque, ISBN 0-521-00924-3

should be available at Akademika

The software:

Clawpack, for *C*onservation *LAW PACK*age, by Leveque and his team at the University of Washington, Seattle

available at www.clawpack.org

What we will do in this course

We will explore hyperbolic equations and finite volume methods for solving them:

What are *hyperbolic equations* and why are they important?

Why are *finite volume methods* well suited to these equations?

How do we solve these equations on the computer?

How can we know whether we have solved the equations correctly?

We will apply our solution methods to problems of geophysical interest:

examples: vents, pockmarks, tsunamis, fluidised systems,
explosive volcanism, atmospheric dispersion

Metrics: grades in this course will be based on:

weekly homework assignments	40%
final project	40%
classroom participation	20%

An approximate syllabus - subject to change

	date		Topic	Chapter in LeVeque
1	17 Aug 2009	Monday 13.15-15.00	introduction to conservation laws, Clawpack	1 & 2 & 5
2	24 Aug 2009	Monday 13.15-15.00	the Riemann problem, characteristics	3
3	**28 Aug 2009**	**Thursday 13.15-15.00**	finite volume methods for linear systems	4
4	**8 Sep 2009**	**Tuesday 13.15-15.00**	high resolution methods	6
5	21 Sep 2009	Monday 13.15-15.00	boundary conditions and accuracy	7 & 8
6	**24 Sep 2009**	**Thursday 13.15-15.00**	nonlinear conservation laws, traffic flow	11
7	28 Sep 2009	Monday 13.15-15.00	finite volume methods for nonlinear equations	12
8	5 Oct 2009	Monday 13.15-15.00	nonlinear systems, shallow-water equations	13
9	12 Oct 2009	Monday 13.15-15.00	gas dynamics, Euler equation	14
10	19 Oct 2009	Monday 13.15-15.00	finite volume methods for nonlinear systems	15
11	26 Oct 2009	Monday 13.15-15.00	multidimensional hyperbolic problems & methods	18 & 19
12	2 Nov 2009	Monday 13.15-15.00	multidimensional scalar equations & systems	20 & 21
13	**5 Nov 2009**	**Thursday 13.15-15.00**	applications: tsunamis, pockmarks, venting, impacts	
14	16 Nov 2009	Monday 13.15-15.00	applications: volcanic jets, pyroclastic flows, lahars	
15	23 Nov 2009	Monday 13.15-15.00	review	
16	30 Nov 2009	Monday 13.15-15.00	discuss progress and problems on projects	
17	7 Dec 2009	Monday 13.15-15.00	FINAL PROJECT REPORTS DUE	

There will be problem sets assigned most weeks and due the following class time. **Keep up with the reading, and do the problems!** Things get complicated quickly, and you will flounder if you don't keep up.

Other chapters in the book may be interesting and important for some of you, notably 9, 16, 17, & 22. Feel free to study these, and if popular will demands, we can cover some of that material in class too.

The **Code Debate**:
Build and use, or Procure and use?

Which is better?

Build your own code and use it to solve problems you’re interested in?

Procure an existing code and use it to solve problems you’re interested in?

If you **build**:

You will know exactly what methods it uses, how it works, and its strengths and limitations.

However, you will spend months and perhaps years debugging it; tweaking its performance; adding features to it; porting it to different systems; and you may never get adequate use from it, especially if you have limited time (as a student, for example!).

The **Code Debate**:
Build and use, or Procure and use?

Which is better?

Build your own code and use it to solve problems you’re interested in?

Procure an existing code and use it to solve problems you’re interested in?

If you **procure**:

Once you install it and prepare your input files, you will be able to start solving those problems immediately.

However, you won’t necessarily know the strengths and weaknesses of its methods; you run the risk of generating mountains of meaningless output by attempting to run it on a problem for which it is ill-suited; and you may not be able to defend your results.

The best (**infinite time**) solution

Spend N years developing the **world’s best code** for the problems you’re interested in. Debug, tweak, and add features to your heart’s content.

Then freeze it.

Spend the next M years running your code on those problems, and publish lots of papers.

By this time, numerical techniques have advanced far beyond those you used in your code; the computers you wrote your code for are obsolete; you haven’t kept up with changes in the compilers and operating systems; etc.

Bottom line: it’s not the **world’s best code** any more, and you have new problems you want to solve.

You could do it all again…

OR…

Stand on the shoulders of giants

A better solution may be to find a code that:

Solves the problems you’re interested in with the latest & best techniques

Has a broad user base with experience in solving similar problems

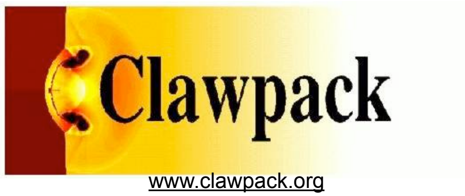
Has a strong development team that are committed to:
helping with porting issues
keeping up with new architectures, compilers, & operating systems
responding to reported difficulties
advising on how to add functionality

Allows you (*or better: **Requires you***) to modify the code yourself, if you need something different or find a better technique. This condition rules out most (all?) commercial codes.

If you do this, you are **obligated** to yourself and to your science to study it carefully so that you clearly **understand** and **can defend** the techniques it uses!

That, in short, is the aim of this course.

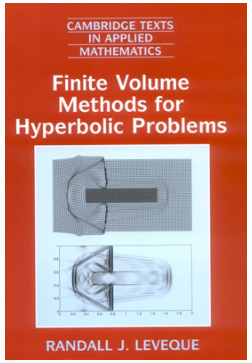
A few codes are documented with textbooks



www.clawpack.org

This isn't the only one, but it's the one we will use.

The techniques you learn using Clawpack and building applications with it will also help you use and build other codes.


CAMBRIDGE TEXTS IN APPLIED MATHEMATICS
Finite Volume Methods for Hyperbolic Problems
RANDALL J. LEVEQUE

ISBN 0-521-00924-3

What you'll need to use Clawpack

A Unix/Linux/MacOS computer

A Fortran compiler (pref. gfortran, free, from http://gcc.gnu.org)

A Python interpreter (free, from www.python.org)

The SciPy extension for Python, which includes NumPy and Matplotlib (free, from www.scipy.org or from www.macinscience.org (for Mac))
This gives you, for free, most of the capability of MatLab, which is expensive!

Clawpack itself (free, from www.clawpack.org, but requires registration)

Curiosity, motivation, patience, and cleverness.

What are hyperbolic problems anyway?

There are three main types of *second-order* partial differential equations. General prototypes of these are:

the *wave equation*:	$q_{tt} - \gamma^2 q_{xx} - f(x) = 0$	"hyperbolic"
the *heat equation*:	$q_t - \alpha^2 q_{xx} - f(x) = 0$	"parabolic"
and *Poisson's equation*:	$q_{xx} - f(x) = 0$	"elliptic"

Why the geometric terminology?

With suitable variable changes, all of these can be squeezed into the form:

$$a\Phi_{\chi\chi} + b\Phi_{\chi\zeta} + c\Phi_{\zeta\zeta} + f(\Phi, \Phi_\chi, \Phi_\zeta, \chi, \zeta) = 0$$

Reminiscent of the equation for conic sections:

$$ax^2 + bxy + cy^2 + d = 0$$

Conic Sections and Partial Differential Equations

The general equation for conic sections $ax^2 + bxy + cy^2 + d = 0$

has the discriminant $b^2 - 4ac$.

$$\text{If } \begin{cases} b^2 - 4ac > 0 & \text{the equation yields a hyperbola,} \\ b^2 - 4ac = 0 & \text{the equation yields a parabola,} \\ b^2 - 4ac < 0 & \text{the equation yields an ellipse.} \end{cases}$$

The general second-order partial differential equation in two variables:

$$a\Phi_{\chi\chi} + b\Phi_{\chi\zeta} + c\Phi_{\zeta\zeta} + f(\Phi, \Phi_\chi, \Phi_\zeta, \chi, \zeta) = 0$$

is easily transformed to the hyperbolic equation $q_{tt} - \gamma^2 q_{xx} - f(x) = 0$

in the case $b^2 - 4ac > 0$.

The other two cases are more difficult to demonstrate.

Examples of Partial Differential Equations in one spatial dimension

elliptic equations

Poisson's equation: $q_{xx} - f(x) = 0$

conditions at boundaries determine the solution everywhere and simultaneously, no time dependence

parabolic equations

heat equation: $q_t - \alpha^2 q_{xx} - f(x) = 0$

inhomogeneities diffuse away irreversibly, leading to a steady state

hyperbolic equations

wave equation: $q_{tt} - \gamma^2 q_{xx} - f(x) = 0$

all physics is local and dynamic; waves that are generated propagate away from the source

Features of the three types of PDEs

	hyperbolic	parabolic	elliptic
example	$q_{tt} - \gamma^2 q_{xx} = 0$	$q_t - \alpha^2 q_{xx} = 0$	$q_{xx} + q_{yy} = 0$
eigenvalues	real	zero	complex
nature of solutions	wave-like, energy-conserving	damping, diffusion, irreversibility	steady-state, no waves
Types of boundary conditions	Cauchy (initial value problem)	Cauchy plus Neumann or Dirichlet	Neumann or Dirichlet (edges)

Features of the three types of PDEs

	hyperbolic	parabolic	elliptic
example	$q_{tt} - \gamma^2 q_{xx} = 0$	$q_t - \alpha^2 q_{xx} = 0$	$q_{xx} + q_{yy} = 0$
eigenvalues	real	zero	complex
nature of solutions	wave-like, energy-conserving	damping, diffusion, irreversibility	steady-state, no waves
Types of boundary conditions	Cauchy (initial value problem)	Cauchy plus Neumann or Dirichlet	Neumann or Dirichlet (edges)

Features of the three types of PDEs

This course

	hyperbolic	parabolic	elliptic
example	$q_{tt} - \gamma^2 q_{xx} = 0$	$q_t - \alpha^2 q_{xx} = 0$	$q_{xx} + q_{yy} = 0$
eigenvalues	real	zero	complex
nature of solutions	wave-like, energy-conserving	damping, diffusion, irreversibility	steady-state, no waves
Types of boundary conditions	Cauchy (initial value problem)	Cauchy plus Neumann or Dirichlet	Neumann or Dirichlet (edges)

Systems of first-order equations

Although the classification of PDEs into hyperbolic, parabolic, and elliptic was designed around the appearance of second-order equations, it is also applied to systems of first-order equations.

For example, the linearised shallow-water (or tsunami) equations:

$$h_t(x,t) + Dv_x(x,t) = 0$$
$$Dv_t(x,t) + gDh_x(x,t) = 0$$

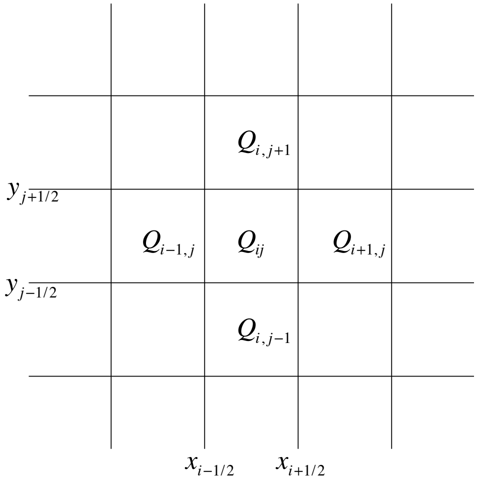
Here h is the wave height, v is the particle speed, D is the ocean depth, and g is the acceleration due to gravity. We'll use different symbols when we derive this later.

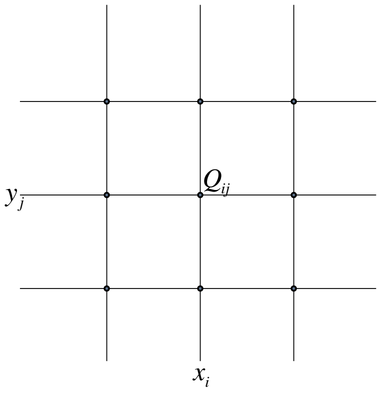
are a hyperbolic set, giving rise to wave-like behaviour as can be seen from the derived wave equation:

$$h_{tt} - gDh_{xx} = 0$$

The methods we speak of in this course are aimed at the solution of systems of first-order equations like these, so we will only occasionally mention second-order equations, which can always be re-cast as systems of first-order equations.

Finite Volume vs Finite Difference





The values are considered to be averages within cells:

$$Q_{ij}^n \approx \frac{1}{\Delta x \Delta y} \int_{y_{j-1/2}}^{y_{j+1/2}} \int_{x_{i-1/2}}^{x_{i+1/2}} q(x,y,t_n)\,dx\,dy$$

The values are considered to be evaluated on a grid:

$$Q_{ij}^n \approx q(x_i, y_j, t_n)$$

Finite Difference, Finite Volume, and Finite Element Methods

Finite difference methods *sample* the solution to form the numerical approximation.

Finite volume methods *average* the solution to form the numerical approximation.

Both of these use the approximation as the primary representation, but they switch to piecewise-polynomial representations to evolve the solution.

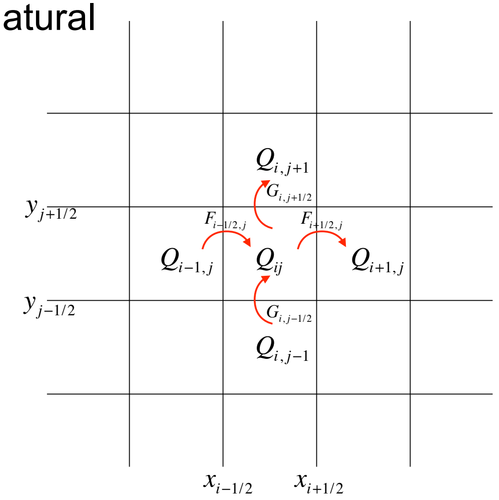
Finite element methods use piecewise-polynomial representations as the primary representation, and evolve the polynomials directly.

They are advantageous for complex geometries, but have difficulty in dealing with shocks and other evolving discontinuities.

They are the subject of FYS-GEO4510, given by Viktoriya Yarushina (usually given by Yuri Podladchikov) in the spring.

For conservation laws, finite volume methods are natural

If the quantity of interest is a *conserved* quantity, we update it by keeping track of the *fluxes* into (and out of) each cell



We can also allow for sources and sinks within the cells

We will use something like this to update Q for the next time step:

$$Q_{ij}^{n+1} \approx Q_{ij}^n - \frac{\Delta t}{\Delta x}\left(F_{i+1/2,j}^n - F_{i-1/2,j}^n\right) - \frac{\Delta t}{\Delta y}\left(G_{i,j+1/2}^n - G_{i,j-1/2}^n\right)$$

Advantages and disadvantages of finite volume methods

Finite volume methods: high resolution, but low order (2nd at best, in general)

Best for cases with discontinuities (sharp material interfaces, shocks)

Best for any kind of waves: acoustic, seismic, water, electromagnetic, etc.

Good for highly compressible media

Good for high-speed flows

In general, best for hyperbolic problems

But…

Not good for slow, viscosity-dominated processes

Poor for parabolic problems (diffusion-dominated processes)

Not applicable to elliptic problems (LaPlace or Poisson type equations)

For these latter cases, linear methods, high-order finite difference methods, or finite element methods are better suited.

Introduction to Conservation Laws (Chapters 1 & 2 in Leveque)

A little on notation, following Leveque

Use subscript notation to refer to partial derivatives:

$$q_x \equiv \frac{\partial q}{\partial x};\;\; q_y \equiv \frac{\partial q}{\partial y};\;\; q_z \equiv \frac{\partial q}{\partial z};\;\; q_t \equiv \frac{\partial q}{\partial t};$$

x, y, and z usually refer to the Cartesian coordinates; t to time.

q is some quantity of interest whose value we need to know: the true solution to the partial differential equation under study. In general, q represents a *vector* of quantities, the components of which are denoted by superscripts as q^p. An $m \times m$ system of equations has eigenvectors r^p and eigenvalues λ^p.

Q_i^n is the numerical approximation to q in the i th grid cell at the nth time step, and the time at the nth time step is denoted t_n.

$F_{i+1/2}^n$ is the numerical approximation to the *flux* of quantity q from cell i to cell $i+1$ at the nth time step.

For two dimensions, we use the additional cell-index subscript j, and the additional flux approximation $G_{j+1/2}^n$.

Numerical solution of differential equations

The equation $\frac{\partial q}{\partial t} = -A\frac{\partial q}{\partial x}$ has a discrete analogue $\frac{\Delta q}{\Delta t} = -A\frac{\Delta q}{\Delta x}$

which can in principle be solved by time stepping: $q_i^{n+1} = q_i^n - A\left(\frac{\Delta q}{\Delta x}\right)_i \Delta t$

The difficulty is in suitably defining the numerical approximation of the spatial derivative, and ensuring that the solution is stable and accurate. A naive implementation of the above is highly unstable!

If the differential equation can be expressed as a *conservation law*, the spatial derivative can be interpreted as a *flux* of the conserved quantity.

Straightforward analysis then produces robust numerical techniques in which it is easier to guarantee stability and accuracy.

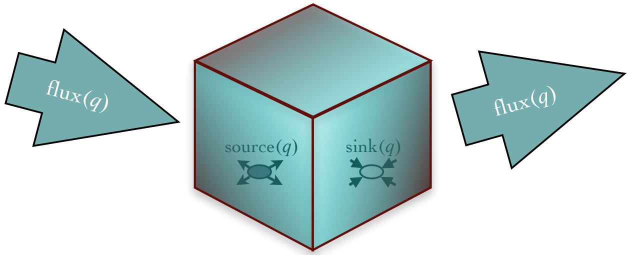
Conservation laws

Many of the fundamental physical laws are conservation laws:

conservation of mass, energy, momentum, entropy (sometimes)…

For any vector of conserved quantities q:

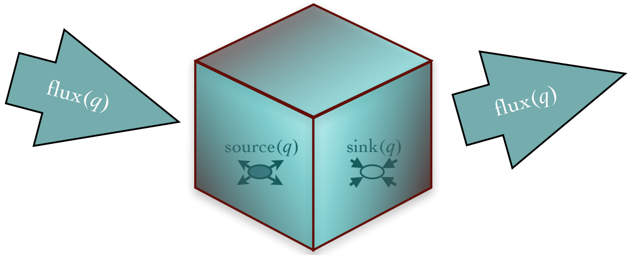
The change with time of q in a volume is due to the net flux of q into or out of the volume and the net amount of q created or destroyed within the volume:



General integral form for a three-dimensional conservation law

$$\frac{d}{dt}\underset{Volume}{\iiint} q(x,y,z;t)\,dV = -\underset{Surface}{\oiint} \vec{F}(q)\,dS - \underset{Volume}{\iiint} R\left(q\right)dV$$

$R(q)$ represents all sources and sinks for the quantity q in the volume V, and $\vec{F}(q)$ represents the net flux into the volume through its surface.



We start with a one-dimensional system

The integral form of the general one-dimensional conservation law over an interval (x_L,x_U), ignoring sources and sinks, is:

$$\frac{d}{dt}\int_{x_L}^{x_U} q(x,t)\,dx = f(q(x_L,t)) - f(q(x_U,t))$$

and the corresponding differential equation is:

$$q_t(x,t) + f(q(x,t))_x = 0$$

The integral form is more general and more fundamental. Finite volume methods are designed to solve the integral form.

The differential form is more compact, but is not valid at discontinuities (shocks or contact surfaces, for example).

We write the equations in differential form for convenience, and for constructing the matrix representation of a system of partial differential equations.

The finite volume method solves the integral form of the equation.

The world has three space dimensions!

True!

But an advantage of *hyperbolic* systems is that it is straightforward to extend the solution method to two, three, and even more dimensions.

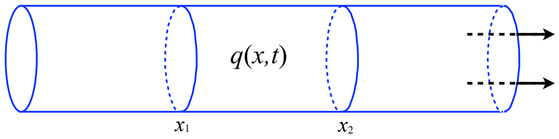
The solutions are wavelike, and waves interact with each other in predictable ways. Added complications include corner transport and boundary conditions.

In the book, two dimensional systems are covered starting in Chapter 18. We'll get there in October.

But first we have a lot of work to do to understand the one-dimensional solution.

Derivation of the conservation law

Assume we have a fluid of density $q(x,t)$ flowing through a pipe.



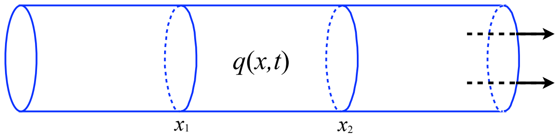
The total mass of the fluid between the positions x_1 and x_2 is:

$$\int_{x_1}^{x_2} q(x,t)\,dx$$

($q(x,t)$ is measured in units of mass per unit length).

This mass changes only because of the flux of the fluid through the left or right ends of the interval.

Conservation law, continued



This mass in the interval changes only because of the flux of the fluid through the left or right ends of the interval:

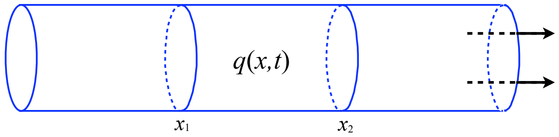
$$\frac{d}{dt}\int_{x_1}^{x_2} q(x,t)\,dx = f(q(x_1,t)) - f(q(x_2,t))$$

where $f(q)$ is the flux function. For a fluid of density q flowing at a velocity $u(x,t)$, the flux function is

$$f(q(x,t)) = u(x,t)q(x,t)$$

This is the **integral** form of the one-dimensional conservation law.

Differential form of conservation law



If q is **smooth** enough, we can rewrite:

$$\frac{d}{dt}\int_{x_1}^{x_2} q(x,t)\,dx = f(q(x_1,t)) - f(q(x_2,t))$$

as $\int_{x_1}^{x_2} q_t\,dx = -\int_{x_1}^{x_2} f(q)_x\,dx$ or $\int_{x_1}^{x_2} \left(q_t + f(q)_x\right)dx = 0$

since this must be true for all x_1 and x_2, then: $q_t + f(q)_x = 0$

This is the **differential** form of the one-dimensional conservation law.

So we have two representations

The integral form of the general one-dimensional conservation law, *valid everywhere*:

$$\frac{d}{dt}\int_{x_L}^{x_U} q(x,t)\,dx = f(q(x_L,t)) - f(q(x_U,t)),$$

and the differential form, valid where the function is *smooth*:

$$q_t(x,t) + f(q(x,t))_x = 0.$$

Because the differential form is not valid at discontinuities, it must be supplemented there by the Rankine-Hugoniot *jump conditions:*

$$f_r - f_l = s(q_r - q_l),$$

where the subscripts r and l refer to right and left states of the solution and the flux function, and s is the speed with which the discontinuity moves. We will derive this later.

Differential vs. Integral forms

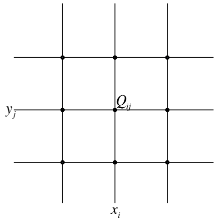
With **finite difference** methods:

approximate pointwise values, i.e. $Q_i^n \approx q(x_i, t_n)$

derivatives are approximated by differences

use **differential form** of conservation law, assume smoothness

$$q_t + f(q)_x = 0$$

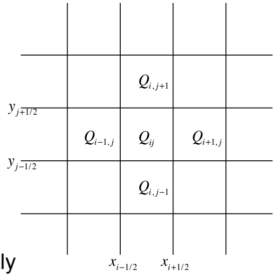


With **finite volume** methods:

approximate cell averages, i.e. $Q_i^n \approx \dfrac{1}{\Delta x}\displaystyle\int_{x_{i-1/2}}^{x_{i+1/2}} q(x, t_n)dx$

use **integral form** of the conservation law

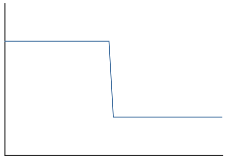
$$\frac{d}{dt}\int_{x_{i-1/2}}^{x_{i+1/2}} q(x,t)\,dx = f(q(x_{i-1/2},t)) - f(q(x_{i+1/2},t))$$



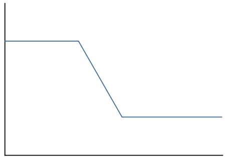
This formulation does not assume smoothness, and leads directly to the numerical method.

Discontinuities occur in real problems

Discontinuities in function values:
shocks, contacts, edges, etc.



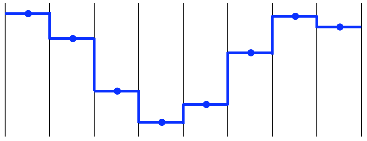
or discontinuities in function derivatives:
rarefactions, reaction fronts, etc.



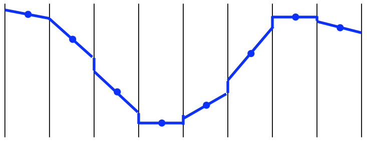
In finite volume methods the *Riemann technique* is used to calculate the propagation of waves away from each discontinuity using the *eigenvalues* of the system of differential equations.

The waves that propagate away from a discontinuity are called *characteristics.*

Artificial discontinuities occur in the numerical solution of equations

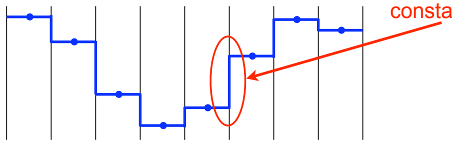


Discretising a problem for numerical solution involves the creation of many small discontinuities; therefore understanding what *characteristics* are and how to solve *Riemann problems* is essential to accurate numerical work.

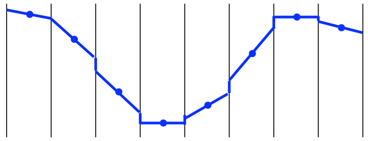


Artificial discontinuities occur in the numerical solution of equations

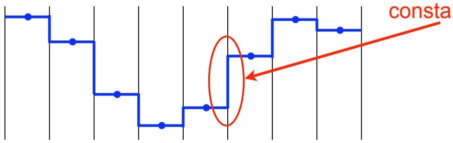
discontinuity in a piecewise constant approximation



Discretising a problem for numerical solution involves the creation of many small discontinuities; therefore understanding what *characteristics* are and how to solve *Riemann problems* is essential to accurate numerical work.

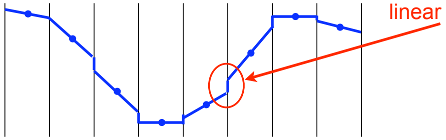


Artificial discontinuities occur in the numerical solution of equations



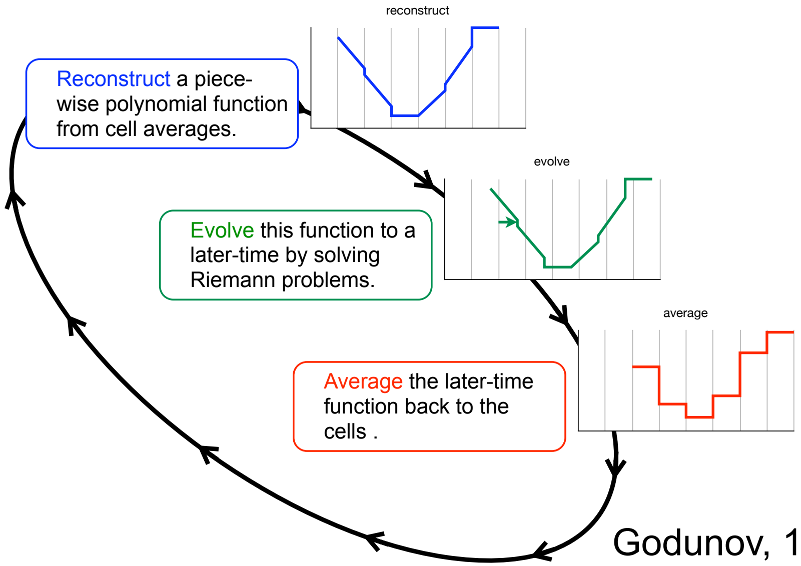
discontinuity in a piecewise constant approximation

Discretising a problem for numerical solution involves the creation of many small discontinuities; therefore understanding what *characteristics* are and how to solve *Riemann problems* is essential to accurate numerical work.



discontinuity in a piecewise linear approximation

Reconstruct - Evolve - Average -(REA)



Reconstruct a piece-wise polynomial function from cell averages.

Evolve this function to a later-time by solving Riemann problems.

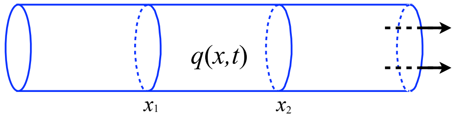
Average the later-time function back to the cells .

Godunov, 1959

The advection equation — the simplest hyperbolic differential equation

With $q(x,t)$ a tracer concentration, and flow velocity

$$u(x,t) = u = \text{constant},$$



The flux function is:

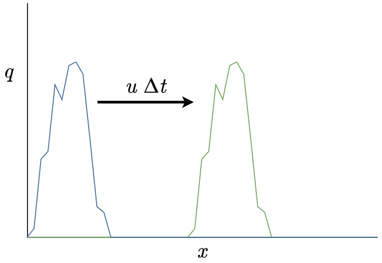
$$f(x,t) = uq(x,t)$$

The conservation law is:

$$q_t(x,t) + uq_x(x,t) = 0$$

And the solution is:

$$q(x,t) = q(x-ut,0)$$



The advection equation is also called the *scalar wave equation*, because the solution is a one-way wave, propagating to the right with velocity u.

The advection equation — the simplest hyperbolic differential equation

With $q(x,t)$ a tracer concentration, and flow velocity

$$u(x,t) = u = \text{constant},$$

The flux function is:

$$f(x,t) = uq(x,t)$$

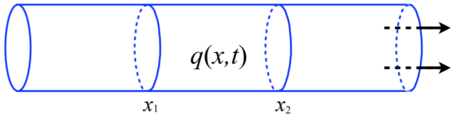
The conservation law is:

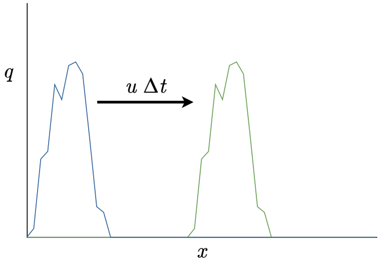
$$\boxed{q_t(x,t) + uq_x(x,t) = 0}$$

And the solution is:

$$q(x,t) = q(x-ut,0)$$

The advection equation is also called the *scalar wave equation*, because the solution is a one-way wave, propagating to the right with velocity u.





Advection is easy; why bother?

Advection is always involved in more complex problems:
material is advected through shocks, for example: and
advection *always* occurs when fluids move, and fluids move in complex ways.

We will find that the solution to more general problems is a superposition of advection solutions.

Advection, by itself, is **notoriously difficult** to do correctly in a numerical problem:
fluid is discretised into finite volumes
fluid velocities are different from the *discretisation ratio*: $\left(\frac{\text{spatial grid size}}{\text{time step}}\right)$
(so every time step advance requires interpolation)

In fact the fluid velocity must be less than this ratio for stability!

Finally, advection is an easy problem to start with, so:
we will develop techniques using this problem
then use the same techniques to solve more difficult ones.

Problems we will be able to solve: acoustic, seismic, and electromagnetic wave propagation, gas dynamics (airplanes, wind turbines, hydrothermal vents, volcanic jets), water waves, tsunamis, traffic flow.

Demonstrate advection difficulty

Using Clawpack -

first-order method

second-order method

modified version of $CLAW/clawpack/1d/example1

A full wave equation is derived from the linearised acoustic equations (for example):

$$p_t(x,t) + K u_x(x,t) = 0$$
$$u_t(x,t) + \frac{1}{\rho} p_x(x,t) = 0$$

The unknown functions $p(x,t)$ and $u(x,t)$ are pressure and velocity; K and ρ are the material constants (bulk modulus and density). Differentiate the first with respect to t and the second with respect to x:

$$p_{tt} - K u_{xt} = 0$$
$$u_{xt} - \frac{1}{\rho} p_{xx} = 0$$

Then add K times the second to the first:

$$p_{tt} - \frac{K}{\rho} p_{xx} = 0$$

This is the familiar **2nd order** wave equation, and it is hyperbolic. The solution gives waves travelling in both directions at velocity $c = \sqrt{\frac{K}{\rho}}$.

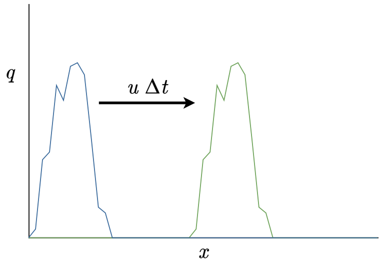
Characteristics for the advection equation

The one-dimensional advection equation

$$q_t(x,t) + u q_x(x,t) = 0$$

has the solution

$$q(x,t) = \eta(x - ut)$$



The function $q(x,t)$ is constant along any space-time ray for which

x–ut = constant

Characteristics for the advection equation

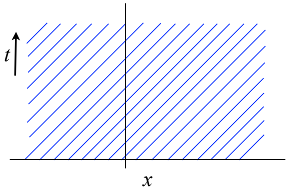
The solution $q(x,t)=\eta(x-ut)$ implies that $q(x,t)$ is constant along lines

$$X(t)=x_0+ut,\ \ t\geq 0$$

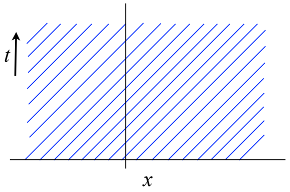
Proof:

$$\frac{d}{dt}q\big(X(t),t\big)=q_x\big(X(t),t\big)X'(t)+q_t\big(X(t),t\big)$$
$$=q_x\big(X(t),t\big)u+q_t\big(X(t),t\big)$$
$$=0$$

$X(t)$ are the wave fronts, or *characteristics*. In the x–t plane they are lines of constant slope:



Initial conditions



If we solve the advection equation on an infinite one-dimensional domain:

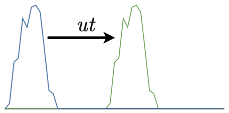
$$q_t(x,t)+uq_x(x,t)=0 \qquad -\infty<x<\infty,\ \ t\geq 0$$

Then we need the initial condition (initial data):

$$q(x,0)=\eta(x), \qquad -\infty<x<\infty$$

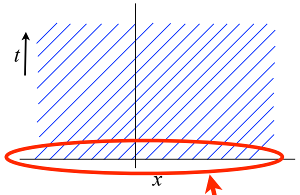
And we have the solution:

$$q(x,t)=\eta(x-ut) \qquad -\infty<x<\infty,\ \ t\geq 0$$



This is an Initial Value Problem (IVP) or Cauchy problem

Initial conditions



If we solve the advection equation on an infinite one-dimensional domain:

$$q_t(x,t)+uq_x(x,t)=0 \qquad -\infty<x<\infty,\ \ t\geq 0$$

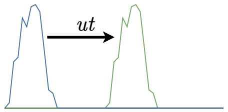
Then we need the initial condition (initial data):

$$q(x,0)=\eta(x), \qquad -\infty<x<\infty$$

Cauchy problem:
all characteristics
originate at t=0

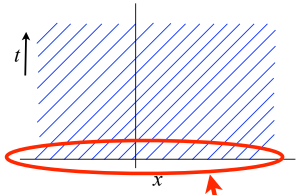
And we have the solution:

$$q(x,t)=\eta(x-ut) \qquad -\infty<x<\infty,\ \ t\geq 0$$



This is an Initial Value Problem (IVP) or Cauchy problem

Initial conditions



If we solve the advection equation on an infinite one-dimensional domain:

$$q_t(x,t)+uq_x(x,t)=0 \qquad -\infty<x<\infty,\ \ t\geq 0$$

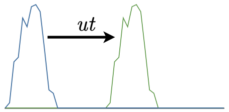
Then we need the initial condition (initial data):

$$q(x,0)=\eta(x), \qquad -\infty<x<\infty$$

Cauchy problem:
all characteristics
originate at t=0

And we have the solution:

$$q(x,t)=\eta(x-ut) \qquad -\infty<x<\infty,\ \ t\geq 0$$



This is an Initial Value Problem (IVP) or Cauchy problem

Boundary Conditions

If we need to solve the same equation on a finite domain:

$$q_t(x,t)+uq_x(x,t)=0 \qquad a<x<b,\ \ t\geq 0$$

Then we need initial data:

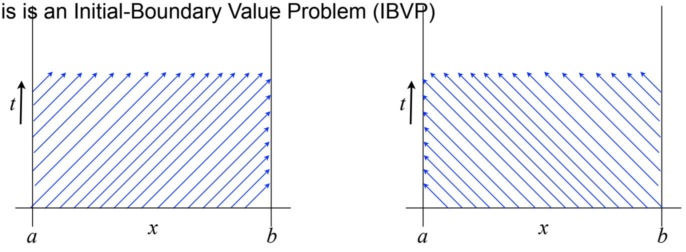
$$q(x,0)=\eta(x), \qquad a<x<b$$

And boundary data at the *inflow* boundary (no boundary condition on *outflow*):

$$q(a,t)=g(t), \qquad t\geq 0 \qquad \underline{\textbf{or}} \qquad q(b,t)=g(t), \qquad t\geq 0$$

$$\text{if } u>0 \qquad\qquad\qquad \text{if } u<0$$

This is an Initial-Boundary Value Problem (IBVP)



Boundary Conditions

If we need to solve the same equation on a finite domain:

$$q_t(x,t)+uq_x(x,t)=0 \qquad a<x<b,\ \ t\geq 0$$

Then we need initial data:

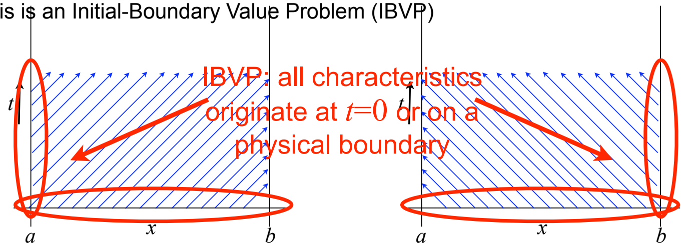
$$q(x,0)=\eta(x), \qquad a<x<b$$

And boundary data at the *inflow* boundary (no boundary condition on *outflow*):

$$q(a,t)=g(t), \qquad t\geq 0 \qquad \underline{\textbf{or}} \qquad q(b,t)=g(t), \qquad t\geq 0$$

$$\text{if } u>0 \qquad\qquad\qquad \text{if } u<0$$

This is an Initial-Boundary Value Problem (IBVP)



IBVP: all characteristics originate at t=0 or on a physical boundary

Periodic Boundary Conditions

The easiest boundary conditions to implement numerically are periodic boundary conditions. Under certain conditions this mimics an infinite domain, but beware the inherent periodicity. For periodic boundary conditions:

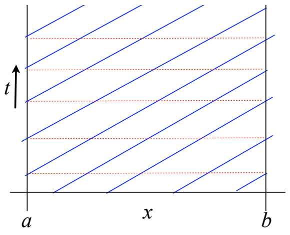
$$q(a,t)=q(b,t), \qquad t\geq 0$$

In this case, the solution is:

$$q(x,0)=\eta\big(X_0(x,t)\big)$$

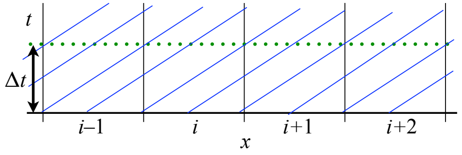
And the characteristics are:

$$X_0(x,t)=a+\mathrm{mod}(x-ut-a,\ b-a)$$



Characteristics lead to the concept of "domain of influence"

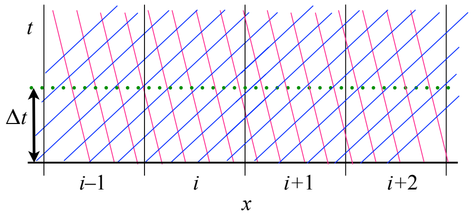
The *domain of influence* is the region of space that can be physically affected by a source.



In this example, cell i can influence **all** of cell i+1 in the time Δt, and cell i is in turn influenced by cell i–1.

For stability, a cell must not attempt to influence more than its immediate neighbours during a single time step of a numerical method.

Characteristic lines for systems of equations



Systems of equations can have as many characteristics as there are equations. These characteristics can propagate in both directions.

Every point in the accessible space-time is crossed by all of the characteristics.

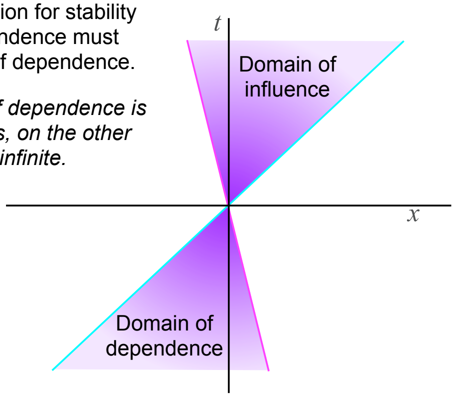
In this example, following the blue characteristics, cell i can influence most of cell $i+1$ in the time Δt, and cell i is in turn influenced by cell $i-1$. Following the red characteristics, cell $i+1$ influences part of cell i, and cell i influences part of cell $i-1$.

Causal domains in space-time

The domains of influence and dependence depend on the characteristics of the equations.

The Courant-Friedrich-Lewy (CFL) condition for stability states that the *numerical* domain of dependence must completely contain the *physical* domain of dependence.

For any hyperbolic system, the domain of dependence is bounded. For elliptic or parabolic systems, on the other hand, the domain of dependence can be infinite.

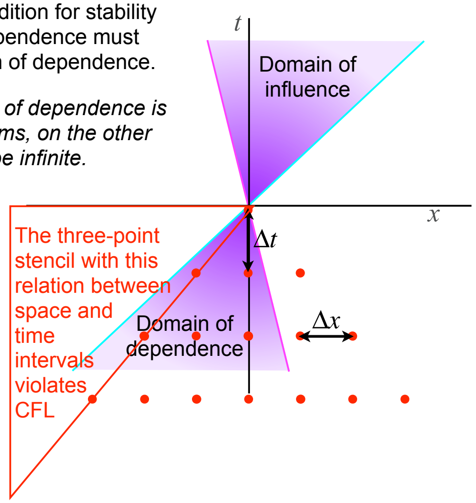


Causal domains in space-time

The domains of influence and dependence depend on the characteristics of the equations.

The Courant-Friedrich-Lewy (CFL) condition for stability states that the *numerical* domain of dependence must completely contain the *physical* domain of dependence.

For any hyperbolic system, the domain of dependence is bounded. For elliptic or parabolic systems, on the other hand, the domain of dependence can be infinite.



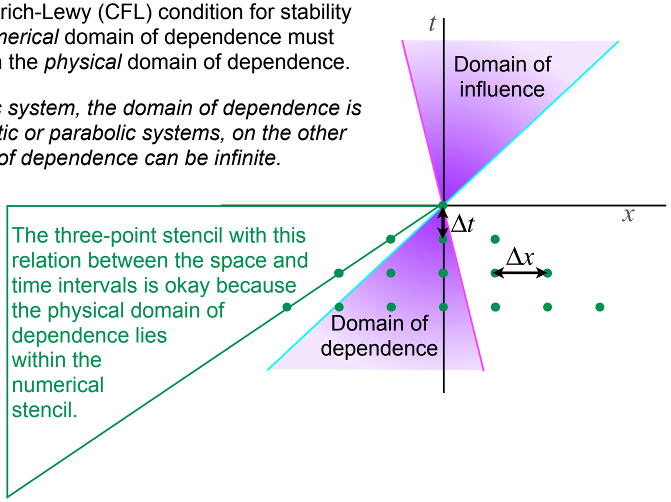
The three-point stencil with this relation between space and time intervals violates CFL

Causal domains in space-time

The domains of influence and dependence depend on the characteristics of the equations.

The Courant-Friedrich-Lewy (CFL) condition for stability states that the *numerical* domain of dependence must completely contain the *physical* domain of dependence.

For any hyperbolic system, the domain of dependence is bounded. For elliptic or parabolic systems, on the other hand, the domain of dependence can be infinite.



The three-point stencil with this relation between the space and time intervals is okay because the physical domain of dependence lies within the numerical stencil.

Variable coefficients

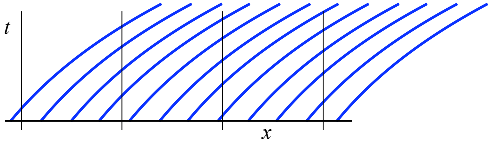
If the fluid velocity varies with x, then the conservation law is:

$$q_t + \big(u(x)q\big)_x = 0$$

And then the characteristics are not straight lines, but *curves* which are found by solving the ODE:

$$X' = u\big(X(t)\big)$$

In this case, q is not constant along the curves, but the curves still track material particles.



The *material derivative* $\dfrac{\partial}{\partial t} + u\dfrac{\partial}{\partial x}$ tracks changes observed by someone

moving along with the fluid (along a characteristic curve).

A little gas dynamics: the continuity equation

A special kind of variable-coefficient equation arises from considering the flow of a gas. Since gasses are compressible, the density and velocity can both vary during the flow. Then the conservation equation derived before, expressed in terms of density ρ is:

$$\rho_t + f(\rho)_x = 0$$

With the flux function $\quad f(x,t) = \rho(x,t)u(x,t)$

This becomes the equation for the conservation of mass, called the continuity equation:

$$\rho_t + (\rho u)_x = 0$$

Conservation of momentum

The continuity equation by itself isn't sufficient to solve the flow of a gas, so we need additional conservation laws.

The product $\rho(x,t)u(x,t)$ is the density of momentum, and we can derive a conservation law equation from it, remembering that pressure p contributes to change of momentum:

$$\big(\rho u\big)_t + \big(\rho u^2 + p\big)_x = 0$$

And then we would need another conservation law for energy and an *equation of state* relating energy to both p and ρ. As a short-cut, we could simply use an equation relating p to ρ. This is sometimes sufficient.

One example is the polytropic equation $p = K\rho^\gamma$, which is the isothermal condition for $\gamma = 1$. More generally we may use the barotropic relation $p = P(\rho)$.

Then we have a closed system of two equations. If $P'(\rho) > 0$ for positive ρ, the system is hyperbolic.

The basic system of equations for a barotropic fluid (simplified gas dynamics)

Our system is

$$\rho_t + (\rho u)_x = 0$$
$$\big(\rho u\big)_t + \big(\rho u^2 + P(\rho)\big)_x = 0$$

Which we can write as $q_t + f(q)_x = 0$, if we define

$$q = \begin{bmatrix} \rho \\ \rho u \end{bmatrix}, \quad f(q) = \begin{bmatrix} \rho u \\ \rho u^2 + P(\rho) \end{bmatrix}$$

If q is sufficiently smooth, we can write $q_t + f'(q)q_x = 0$ (quasilinear form)

Where $f'(q) = \begin{bmatrix} \dfrac{\partial f^1}{\partial q^1} & \dfrac{\partial f^1}{\partial q^2} \\ \dfrac{\partial f^2}{\partial q^1} & \dfrac{\partial f^2}{\partial q^2} \end{bmatrix}$ is the *Jacobian* matrix.

Linear acoustics

This will be a useful example that we'll return to often. We linearise the barotropic system by examining only a perturbation to the (constant) background state. Let

$$q(x,t)=q_0+\tilde{q}(x,t),\ \tilde{q}=\begin{bmatrix}\tilde{\rho}\\ \widetilde{\rho u}\end{bmatrix}$$

The conservation law $q_t+f(q)_x=0$ becomes the constant-coefficient linear system $\tilde{q}_t+f'(q_0)\tilde{q}_x=0$ since we discard powers of the perturbed quantity.

The Jacobian of the perturbed barotropic system then becomes

$$A=f'(q)=\begin{bmatrix}\frac{\partial f^1}{\partial q^1} & \frac{\partial f^1}{\partial q^2}\\ \frac{\partial f^2}{\partial q^1} & \frac{\partial f^2}{\partial q^2}\end{bmatrix}=\begin{bmatrix}0 & 1\\ -u^2+P'(\rho) & 2u\end{bmatrix}$$

Linear acoustics

The system just obtained can be written out as:

$$\tilde{\rho}_t+(\widetilde{\rho u})_x=0$$

$$(\widetilde{\rho u})_t+\left(-u_0^2+P'(\rho_0)\right)\tilde{\rho}_x+2u_0(\widetilde{\rho u})_x=0$$

Then writing $P'(\rho_0)\tilde{\rho}=\tilde{p}$, $\widetilde{\rho u}=u_0\tilde{\rho}+\rho_0\tilde{u}$, and $K=\rho_0P'(\rho_0)$

We obtain the linear acoustics equations

$$\tilde{p}_t+u_0\tilde{p}_x+K\tilde{u}_x=0$$

$$\rho_0\tilde{u}_t+\tilde{p}_x+\rho_0u_0\tilde{u}_x=0$$

With $u_0=0$, this becomes the system we showed earlier (dropping ~s):

$$q_t(x,t)+Aq_x(x,t)=0$$

$$q=\begin{bmatrix}p\\ u\end{bmatrix},\ A=\begin{bmatrix}0 & K\\ \frac{1}{\rho_0} & 0\end{bmatrix},\ \text{so we have}\ \begin{matrix}p_t+Ku_x=0\\ \rho_0u_t+p_x=0\end{matrix}$$

Sound waves

$$q_t(x,t)+Aq_x(x,t)=0 \qquad q=\begin{bmatrix}p\\ u\end{bmatrix},\ A=\begin{bmatrix}0 & K\\ \frac{1}{\rho_0} & 0\end{bmatrix}$$

The acoustics equations must produce a solution with sound waves travelling in both directions.

We try a solution of the form $q(x,t)=\eta(x-st)$ and then compute the derivatives:

$$q_t(x,t)=-s\eta'(x-st),\qquad q_x(x,t)=\eta'(x-st)$$

So from $q_t(x,t)+Aq_x(x,t)=0$ we get

$$A\eta'(x-st)=s\eta'(x-st)$$

implying that s is an *eigenvalue* and η' the corresponding *eigenvector* of the matrix A.

This is an important key to the methods we will develop in this course.

Reminder: eigenvalues of a 2x2 matrix

Eigenvalues λ and eigenvectors r for a matrix A are found from $Ar=\lambda r$

For the matrix $A=\begin{bmatrix}a & b\\ c & d\end{bmatrix}$, the eigenvalues are:

$$\lambda^{1,2}=\frac{a+d}{2}\pm\frac{\sqrt{4bc+(a-d)^2}}{2}$$

Hence the acoustic equation matrix $A=\begin{bmatrix}0 & K\\ \frac{1}{\rho} & 0\end{bmatrix}$

has eigenvalues: $\lambda^{1,2}=\pm c=\pm\sqrt{\frac{K}{\rho}}$ and eigenvectors $r^{1,2}=\begin{bmatrix}\pm\sqrt{K\rho}\\ 1\end{bmatrix}$

Important result: ***the eigenvalues are the wave speeds***. The eigenvectors express the relation between the components of the solution vector.

Hyperbolicity Definition

The Jacobian matrix of a linear $m \times m$ system of partial differential equations:

$$A = f'(q) = \begin{bmatrix} \frac{\partial f^1}{\partial q^1} & \cdots & \frac{\partial f^1}{\partial q^m} \\ \vdots & \ddots & \vdots \\ \frac{\partial f^m}{\partial q^1} & \cdots & \frac{\partial f^m}{\partial q^m} \end{bmatrix}$$

A is diagonalisable if it has a complete set of eigenvectors/eigenvalues, *i.e.*

$$Ar^p = \lambda^p r^p \text{ for } p = 1,2,\ldots,m$$

(the vectors r^p must be nonzero)

Strongly hyperbolic:
the matrix A is diagonalisable and has real eigenvalues

Strictly hyperbolic:
the matrix A is diagonalisable and has *distinct* real eigenvalues

Weakly hyperbolic:
the matrix A is not diagonalisable but has real eigenvalues (not a complete set)

Review: conservation law and advection

The fundamental conservation law in one spatial dimension, expressed in differential form, is:

$$q_t(x,t) + f(q(x,t))_x = 0.$$

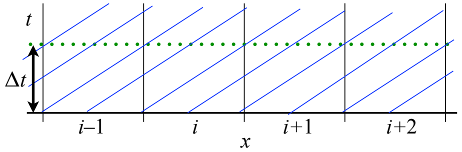
The advection equation, the simplest hyperbolic differential equation,

$$q_t(x,t) + uq_x(x,t) = 0,$$

is a conservation law with the flux function $f(x,t) = uq(x,t)$. Its solution is

$$q(x,t) = q(x - ut, 0),$$

and this function is constant along rays in space-time (*characteristics*) with x–ut = constant.



Review: Linear acoustics in a stationary gas

The acoustic equations are:

$$p_t(x,t) + Ku_x(x,t) = 0$$
$$u_t(x,t) + \tfrac{1}{\rho} p_x(x,t) = 0.$$

Expressed in linear form, with matrix notation:

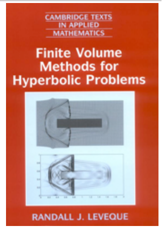
$$q_t(x,t) + Aq_x(x,t) = 0 \qquad q = \begin{bmatrix} p \\ u \end{bmatrix}, \quad A = \begin{bmatrix} 0 & K \\ \frac{1}{\rho_0} & 0 \end{bmatrix}.$$

This can be resolved into the eigensystem $Ar = \lambda r$,

with eigenvalues $\lambda^{1,2} = \pm c = \pm\sqrt{\dfrac{K}{\rho}}$ and eigenvectors $r^{1,2} = \begin{bmatrix} \pm\sqrt{K\rho} \\ 1 \end{bmatrix}.$

The eigenvalues are the wave speeds, and the eigenvectors express relations between the components of the solution q.

Assignment (due Monday 24 August):



CAMBRIDGE TEXTS IN APPLIED MATHEMATICS
Finite Volume Methods for Hyperbolic Problems
RANDALL J. LEVEQUE

Get the book! It's available at Akademika now.

Download the *Errata* from `http://www.amath.washington.edu/~claw/` and **apply the corrections**. Your copy may not need all of them since it's a newer printing.

Read all of Chapter 1, and Chapter 2 at least through section 2.11. The remainder of Chapter 2 may interest you, but it won't be necessary for this course.

Work Exercises 2.2 (a and b parts) and 2.8 (a part only) and hand them in to me by next Monday (the 24th). There is a file of sample solutions available at the above website, but please try them first on your own before consulting it.

Read the handout with instructions for downloading Clawpack. Make sure you have access to a Unix/Linux machine (Mac OS X is fine) with a good development environment, including at a minimum Fortran 90/95 (gfortran, for example), and Python 2.5 or 2.6.

Download Clawpack and install it on your computer. Run some tests to make sure it is installed correctly. Come see me if you get stuck, and we will discuss installation difficulties on Monday.

Next: Riemann Problem (Ch 3)