The University of British Columbia
MECH 479

Module 2
Numerical Integration of ODEs
&
Classification of PDEs

September 12, 2022

Module 2a: Numerical Integration of ODEs

In this module, we shall review Ordinary Differential Equations (ODEs) and basic numerical operations on them such as time integration and accuracy analysis. We shall also look at a simple Matlab code illustrating these operations. This will build the foundation for extending our study to Partial Differential Equations (PDEs) in the next chapter.

1 Introduction

An ODE is a differential equation consisting of one or more functions of *one* independent variable and the derivatives of those functions. In solving a differential equation, the aim is to determine a differential function that satisfies the underlying differential equation.
Thus if x is our independent variable and $y(x)$ is a function of x, a general implicit ODE in x is of the form

$$F\left(x, y, y', y'', y'', \ldots, y^{(n)}\right) = 0$$

where

$$y' = \frac{dy}{dx}, \quad y'' = \frac{d^2y}{dx^2}, y''' = \frac{d^3y}{dx^3}, y^{(n)} = \frac{d^ny}{dx^n}$$

The highest-order derivative appearing in an ODE determines the order of the ODE. For example, Newton's Second Law is a second-order ODE. In contrast to a PDE, there is only one independent variable e.g. time, then all the functions are differentiated with respect to that independent variable.

2 Time Integration of ODEs

Let us consider a simple first-order ODE of u:

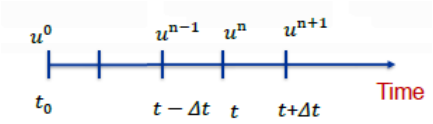
$$\frac{du}{dt} = g(t, u) \tag{1}$$

where u is the solution (unknown) function to be solved and t is the independent variable. Given an initial condition for u at time t_0 as

$$u(t_0) = u^0$$

we can find the values of the solution function u at discrete times t_n. Assuming for now a uniform time discretization, our discrete solutions will be of the form

$$u^0 = u(t_0) \quad u^1 = u(t_0 + \Delta t) \quad u^2 = u(t_0 + 2\Delta t)$$
$$u^{n-1} = u(t - \Delta t) \quad u^n = u(t) \quad u^{n+1} = u(t + \Delta t)$$



Let us see how we can numerically integrate this ODE in time to get our solution function at the discrete times of interest. We rearrange equation 1 to get

$$du = g(\tau, u)d\tau \tag{2}$$

Without loss of generality, let $g(u,t)$ be an arbitrary well-behaved function of the form shown in Figure 1.

Integrating equation 2 in the interval t to $t + \Delta t$ we get

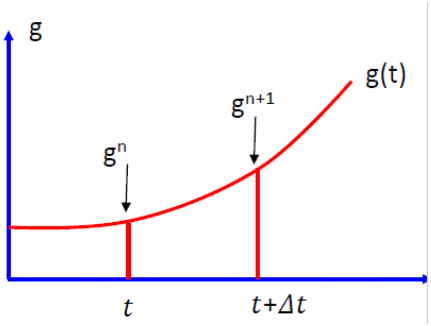


Figure 1: Function $g(u,t)$ evaluated at discrete times t

$$\int_t^{t+\Delta t} du = \int_t^{t+\Delta t} g(\tau, u)d\tau$$

or

$$u^{n+1} - u^n = \int_t^{t+\Delta t} g(\tau, u)d\tau$$

The integration above can be done using various numerical quadrature schemes. We take a look at three simple commonly used quadrature schemes -

2.1 Forward Euler

In the Forward Euler scheme, we approximate the value of the function $g(\tau, u)$ over the discrete time domain of interest $[t, t + \Delta t]$ to be equal to that at time t. Thus, we have

$$\int_t^{t+\Delta t} g(\tau, u) d\tau \approx g(t)\Delta t = g^n \Delta t$$

Giving us

$$u^{n+1} = u^n + g^n \Delta t$$

Figure 2 demonstrates the Forward Euler time integration scheme in the discrete time domain.

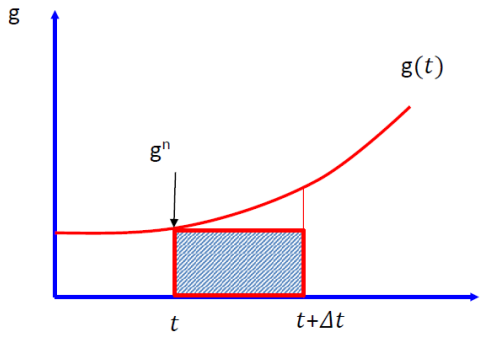


Figure 2: Forward Euler

2.2 Backward Euler

As the name suggests, in Backward Euler we approximate the value of the function $g(\tau, u)$ over the discrete time domain of interest $[t, t + \Delta t]$ to be equal to that at time $t + \Delta t$.

$$\int_t^{t+\Delta t} g(\tau, u) d\tau \approx g(t + \Delta t)\Delta t = g^{n+1} \Delta t$$

Thus, we can evaluate our solution variable (u^{n+1}) at the time $t + \Delta t$ as

$$u^{n+1} = u^n + g^{n+1} \Delta t$$

Figure 3 demonstrates the Backward Euler time integration scheme.

2.3 Trapezoidal Rule

Yet another simple quadrature scheme is the Trapezoidal Rule. Here our quadrature takes the form of the integrated area of a trapezoid. The approximation we make is that

$$\int_t^{t+\Delta t} g(\tau, u) d\tau \approx \frac{1}{2} \left(g(t) + g(t + \Delta t) \right) \Delta t = \frac{1}{2} \left(g^n + g^{n+1} \right) \Delta t$$

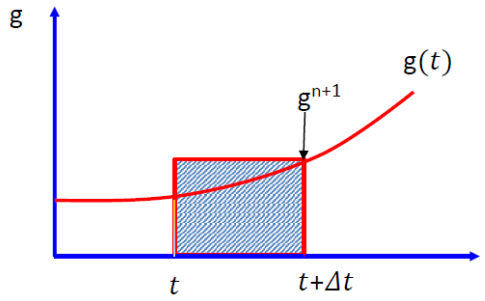


Figure 3: Backward Euler

Thus (u^{n+1}) is obtained to be

$$u^{n+1} = u^n + \frac{1}{2}\left(g^n + g^{n+1}\right)\Delta t$$

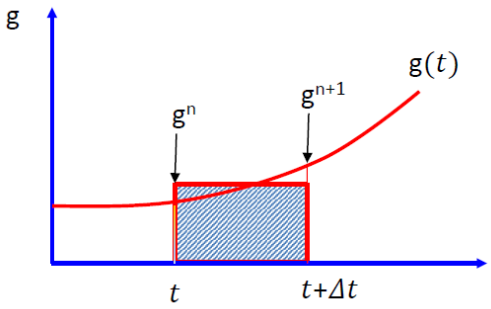


Figure 4: Trapezoidal Rule

3 Accuracy and Error Analysis

3.1 Exact Solution

Let us take a look at how our numerical solution compares to the exact solution of the ODE. In our model equation, let us take (for example) that

$$g(t,u) = -u$$

This gives us

$$du = -udt$$

Using an initial condition of

$$u(0) = 1$$

We get the exact solution of the ODE at time t to be

$$u(t) = e^{-t}$$

This is the known exact solution (ground-truth) that we would like to take as a benchmark to determine how accurate our numerical integration scheme is.

3.2 Accuracy

We can now see how the integration schemes we used perform in comparison to the exact solution. Continuing from the example that we took

$$g(t, u) = -u$$

ie

$$g^n = -u^n \text{ and } g^{n+1} = -u^{n+1}$$

We substitute this into our schemes to get the following approximations for u^{n+1}

Forward Euler

$$u^{n+1} = (1 - \Delta t)u^n$$

Backward Euler

$$u^{n+1} = \frac{u^n}{(1 + \Delta t)}$$

Trapezoidal Rule

$$u^{n+1} = \frac{(1 - \Delta t/2)}{(1 + \Delta t/2)}u^n$$

We use the following simple routine (figure 5) in MATLAB to compute our solution variable using the three schemes and compare the results to the ground-truth.

Figure 6 shows the result of the comparison. We can make the following general observations

- The overall behaviour of the numerical solutions is similar to the exact solution.
- In comparison to the Forward and Backward Euler schemes, the Trapezoidal Rule appears to result in numerical solutions that are closer in value to the exact solution.

How do we improve the accuracy of the Forward and the Backward Euler schemes? Let us try running the Forward Euler scheme with smaller time-steps. Figure 7 shows the result of this exercise. We observe that our error depends on temporal discretization. Let us see why this is so and how the error behaves.

```
% MECH 479 - CFD
% A simple code for three integration methods
nstep=5;
dt=0.5;
u1=zeros(nstep,1);
u2=zeros(nstep,1);
u3=zeros(nstep,1);
uex=zeros(nstep,1);
t=zeros(nstep,1);
t(1)=0;u1(1)=1;u2(1)=1;u3(1)=1;uex(1)=1;
for i=2:nstep
    u1(i)=u1(i-1)-dt*u1(i-1); %Forward Euler
    u2(i)=u2(i-1)/(1.0+dt); %Backward Euler
    u3(i)=u3(i-1)*(1.0-0.5*dt)/(1.0+0.5*dt);
%Trapezoidal Rule
    t(i)=t(i-1)+dt;
    uex(i)=exp(-t(i));
end;
plot(t,u1);hold on;plot(t,u2,'r');plot(t,u3,'k');
plot(t,uex, 'r', 'linewidt',3);
set(gca,'fontsize',24,'linewidt',2);
```

Figure 5: MATLAB Code

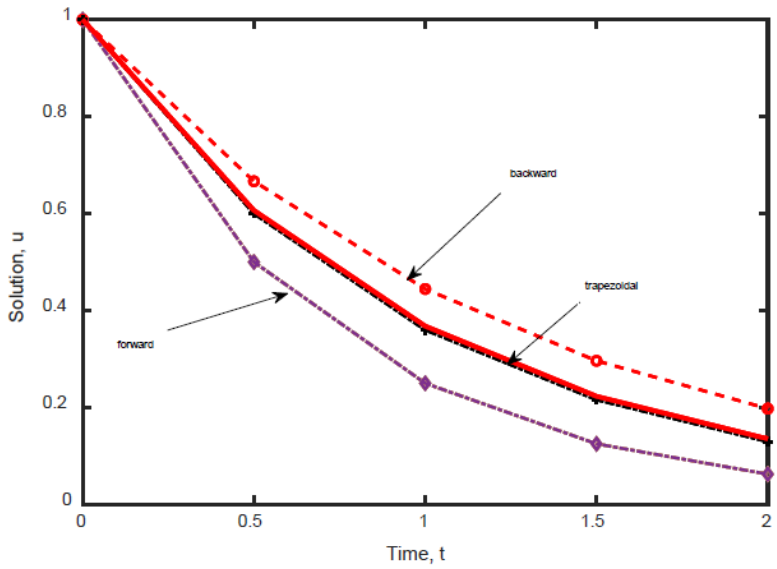


Figure 6: MATLAB Code

3.3 Error Analysis

Let us use a Taylor Series to expand the function $g(t,u)$ about the time t.

$$g = g^n + \left(\frac{dg}{dt}\right)^n \tau + \dots$$

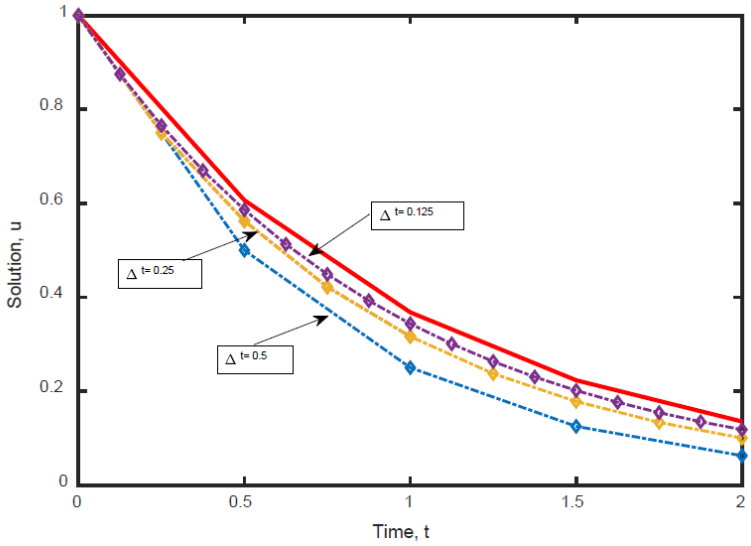


Figure 7: Effect of time-step on solution

Integrating this equation in the discrete time interval from t to $t + \Delta t$

$$\int_t^{t+\Delta t} g d\tau = \int_t^{t+\Delta t} g^n d\tau + \int_t^{t+\Delta t} \left(\frac{dg}{dt}\right)^n \tau d\tau + \dots$$

Giving us

$$\int_t^{t+\Delta t} g d\tau = g^n \Delta t + \left(\frac{dg}{dt}\right)^n \frac{\Delta t^2}{2} + L$$

Where L are the higher order terms. As we saw previously, our solution variable is of the form

$$u^{n+1} = u^n + \int_t^{t+\Delta t} g(\tau, u) d\tau$$

The Taylor Series expansion gives us

$$u^{n+1} = u^n + g^n \Delta t + \left(\frac{dg}{dt}\right)^n \frac{\Delta t^2}{2} + L \tag{3}$$

We can compare this to the Forward Euler scheme which we determined to be

$$u^{n+1} = u^n + g^n \Delta t \tag{4}$$

We can see from equations 3 and 4 that the Forward Euler scheme is missing some terms when compared to the full Taylor Series expansion. This is the source of error in our integration scheme. We observe that the error is largely dictated by the leading truncated term. For the Forward Euler case, this is of the order of Δt^2. Thus we say that our scheme is second-order accurate in time, also represented as

$$u^{n+1} = u^n + g^n \Delta t + O(\Delta t^2)$$

We can relate this to our results in figure 7 where our error reduces quadratically with a decrease in the time-step size.

4 High-order schemes

We consider two popular high-order methods for the ODE integration.

4.1 Runge-Kutta Methods

Let's reconsider the Trapezoidal Rule:

$$u^{n+1} = u^n + \frac{1}{2}(g^n + g^{n+1})\Delta t$$

Comparing with the Forward Euler method:

$$u^{n+1} = u^n + g^n \Delta t$$

We find that the evaluation of derivative at another instant t^{n+1} is involved. In the accuracy analysis, this results in one order higher approximation, thus improving the accuracy:

$$\begin{aligned}
\text{Forward Euler} &: u^{n+1} = u^n + g^n\Delta t + O(\Delta t^2) \\
\text{Trapezoidal} &: u^{n+1} = u^n + g^n\Delta t \quad \underbrace{+\left(\frac{dg}{dt}\right)^n \frac{\Delta t^2}{2}}_{\text{higher-order approximation}} + O(\Delta t^3)
\end{aligned}$$

One question arise naturally from this observation is that can we improve the order of accuracy by involving the evaluation of the derivative or slope at more instants? If so, how to introduce one?

To answer this question, we set up the following explicit scheme based on the forward Euler method:

$$\begin{aligned}
\text{The first instant} &: k_1 = g(t^n, u^n) \\
\text{The second instant} &: k_2 = g(t^n + a_{21}\Delta t, u^n + \Delta t a_{21} k_1) \\
\text{Integration} &: u^{n+1} = u^n + \Delta t(b_1 k_1 + b_2 k_2)
\end{aligned}$$

Compared with the Forward Euler method, we evaluate the derivative k_2 at an additional instant $t^n + a_{21}\Delta t$, and add k_2 in the integration towards the next time step. The scheme reduces to the forward Euler method simply by setting $b_1 = 1$ and $b_2 = 0$. Now we need to see that by selecting a_{21}, b_1 and b_2 properly, can we get higher order of accuracy? In the current case, can we improve the first order accuracy of the forward Euler method to second order accuracy of the current scheme? To analyze the accuracy of the current scheme, we perform a Taylor expansion:

$$\begin{aligned}
u^{n+1} &= u^n + b_1\Delta t k_1 + b_2 \Delta t k_2 \\
&= u^n + b_1 \Delta t g(t^n, u^n) + b_2 \Delta t k_2
\end{aligned} \tag{5}$$

Specifically, let's consider the Taylor expansion for k_2:

$$\begin{aligned} k_2 &= g(t^n + a_{21}\Delta t, u^n + \Delta t a_{21} g(t^n, u^n)) \\ &= g(t^n, u^n) + \frac{\partial g}{\partial t}(t^n, u^n) a_{21}\Delta t + \frac{\partial g}{\partial u}(t^n, u^n)\Delta t a_{21} g(t^n, u^n) \end{aligned}$$

Substituting k_2 into Eq. (5), we have the Taylor expansion of the current scheme:

$$u^{n+1} = u^n + \Delta t(b_1 + b_2)g(t^n, u^n) + 2a_{21}b_2\frac{\Delta t^2}{2}\left(\frac{\partial g}{\partial t} + \frac{\partial g}{\partial u}g\right)(t^n, u^n) \quad (6)$$

Since

$$\frac{dg}{dt} = \frac{\partial g}{\partial t} + \frac{\partial g}{\partial u}\frac{\partial u}{\partial t},$$

we can rewrite Eq. (6) as:

$$u^{n+1} = u^n + \Delta t(b_1 + b_2)g(t^n) + 2a_{21}b_2\frac{\Delta t^2}{2}\frac{dg}{dt}(t^n, u^n) \quad (7)$$

Now, let's compare the Taylor expansion of the current scheme with the Taylor expansion of the analytical solution given by Eq. (3):

$$u^{n+1} = u^n + g^n\Delta t + \frac{dg}{dt}(t^n, u^n)\frac{\Delta t^2}{2} + L \quad (8)$$

By comparing Eq. (7) with Eq. (8), we find that the condition for the current scheme to be second order accurate is:

$$b_1 + b_2 = 1, \quad 2a_{21}b_2 = 1$$

This is referred to as the order condition. Given a_{21} as a constant selected as α, a general second order explicit Runge-Kutta scheme is given by:

$$a_{21} = \alpha, b_1 = 1 - \frac{1}{2\alpha}, b_2 = \frac{1}{2\alpha}$$

There are many choices of selecting the constant parameters. By setting $a_{21} = \frac{1}{2}$, we get a specific second order explicit Runge-Kutta scheme which is refered to as the midpoint method:

$$\begin{aligned} \text{The first instant} &: k_1 = g(t^n, u^n) \\ \text{The second instant} &: k_2 = g(t^n + \frac{\Delta t}{2}, u^n + \frac{\Delta t}{2}k_1) \\ \text{Integration} &: u^{n+1} = u^n + \Delta t k_2 \end{aligned}$$

Following similar procedures, we can get the fourth order Runge-Kutta scheme, which is the commonly used one among the family of fourth order Runge-Kutta schemes:

$$\begin{aligned} k_1 &= g(t^n, u^n) \\ k_2 &= g(t^n + \frac{\Delta t}{2}, u_n + \Delta t\frac{k_1}{2}) \\ k_3 &= g(t^n + \frac{\Delta t}{2}, u_n + \Delta t\frac{k_2}{2}) \\ k_4 &= g(t^n + \Delta t, u_n + \Delta t k_3) \\ u^{n+1} &= u^n + \Delta t(k_1 + 2_{k2} + 2k_3 + k_4)/6 \end{aligned}$$

4.2 Multistep methods

In the last subsection, the Runge-Kutta methods utilize multiple intermediate time step. Similarly, we can use the information from multiple time steps instead of intermediate time steps. The general form of multistep method is given by:

$$\sum_{j=1-s}^{1} a_j u^{n+j} = \Delta t \sum_{j=1-s}^{1} b_j g(t^{n+j}, u^{n+j})$$

Among the families of multistep methods, Adams-Bashforth methods are commonly used explicit schemes, which takes the form of:

$$u^{n+1} = u^n + \int_{t^n}^{t^{n+1}} p(t)dt$$

where $p(t)$ is a polynomial which approximates $g(t, u)$ at previous multiple time steps. For example, the first order Adams-Bashforth method approximates $g(t, u)$ with $p(t)$ at t^n, which leads to $p(t) = g(t^n, u^n)$:

$$u^{n+1} = u^n + \int_{t^n}^{t^{n+1}} g(t^n, u^n)dt \tag{9}$$

This is essentially the Forward Euler method.

For the second order Adams-Bashforth (AB2) method, $p(t)$ should approximate $g(t, u)$ at t^n and t^{n-1}:

$$p(t^n) = g(t^n, u^n)$$
$$p(t^{n-1}) = g(t^{n-1}, u^{n-1})$$

This can be achieved through the Lagrange formula:

$$p(t) = \frac{(t - t^{n-1})}{(t^n - t^{n-1})} g(t^n, u^n) + \frac{(t - t^n)}{(t^{n-1} - t^n)} g(t^{n-1}, u^{n-1})$$

Substituting this polynomial approximation into Eq. (9):

$$u^{n+1} = u^n + \int_{t^n}^{t^{n+1}} \left(\frac{(t - t^n)}{(t^{n-1} - t^n)} g(t^{n-1}, u^{n-1}) + \frac{(t - t^{n-1})}{(t^n - t^{n-1})} g(t^n, u^n) \right) dt$$

After integration, we get the second order Adams-Bashforth method:

$$u^{n+1} = u^n + \Delta t \left(\frac{3}{2} g(t^n, u^n) - \frac{1}{2} g(t^{n-1}, u^{n-1}) \right)$$

Note that the calculation for t^{n+1} needs derivatives at two previous time steps. To start the scheme, we usually predict t^1 from t^0 with forward Euler scheme, then switch to the second order Adams-Bashforth scheme.

Another family of multistep methods are Adams-Moulton methods, which are implicit methods. This family of schemes takes the same form with the Adams-Bashforth methods:

$$u^{n+1} = u^n + \int_{t^n}^{t^{n+1}} p(t)dt$$

while $p(t)$ needs to approximate $g(t, u)$ at both previous time steps and current time step t^{n+1}, which requires

$$p(t^{n+1}) = g(t^{n+1}, u^{n+1})$$

The derivation of Adams-Moulton methods are identical with that of Adams-Bashforth methods. Here, we give the second order Adams-Moulton scheme:

$$u^{n+1} = u^n + \frac{\Delta t}{2}\left(g(t^{n+1}, u^{n+1}) + g(t^n, u^n)\right)$$

Module 2b: PDEs and their Classification

The equations of fluid mechanics have many properties in common with equations arising in other fields such as solid mechanics and electromagnetism. These properties are usually best investigated by considering a simplified set of equations. Considerations of the simpler equations that are the "building blocks" for the full fluid equations help develop an understanding of the nature of those equations and in the development of suitable numerical schemes. Furthermore, it is usually only for these simpler equations that a numerical method can be fully analyzed. In this module, we will consider the basic elements of the theory of partial differential equations that are relevant to the subsequent development.

5 Introduction

A PDE is a differential equation consisting of unknown multivariable functions and their partial derivatives. PDEs are distinguished from ODEs by the fact that they contain derivatives of more than one independent variable. A general implicit PDE is of the form

$$F\left(x, y, \ldots, u, \frac{\partial u}{\partial x}, \frac{\partial u}{\partial y} \ldots, \frac{\partial^2 u}{\partial x^2}, \frac{\partial^2 u}{\partial xy}, \frac{\partial^2 u}{\partial y^2}, \ldots\right) = 0$$

where $x, y, \ldots$ are the independent variables, and $u = u(x, y, \ldots)$ is the dependent variable. Some simple examples of PDEs that we shall encounter in fluid mechanics are

- The Advection equation; $\frac{\partial \phi}{\partial t} + c\frac{\partial \phi}{\partial x} = 0$
- The Diffusion equation; $\frac{\partial \phi}{\partial t} - v\frac{\partial^2 \phi}{\partial x^2} = 0$
- The Wave propagation equation; $\frac{\partial^2 \phi}{\partial t^2} - v\frac{\partial^2 \phi}{\partial x^2} = 0$
- The Laplace equation; $\frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} = 0$

The order of a PDE is the order of its highest derivative. A PDE is linear if the solution (dependent variable) and all its partial derives occur to the first power only and there are no products involving more than one of these terms. The

dimension of a PDE is the number of independent spatial variables it contains. The essence of solving a PDE implies finding the unknown (dependent) function. Most PDEs of practical interest do not have analytical solutions; hence a numerical procedure must be adopted to find an approximate solution.

6 Classification

Partial Differential Equations are classified according to the highest order derivative that appears in the equation. Therefore, first-order equations contain only first derivatives, second order contain derivatives up to the second and so on.

Further, PDEs are classified based on the mathematical concept of characteristics that are lines (2D) or surfaces (3D) along which certain properties remain constant. This helps classify them broadly into three groups - hyperbolic, parabolic and elliptic. Each of these classes represents different physical processes and requires a different approach for a solution. The number of initial and boundary conditions to be specified also depends on the type of PDE.

1. **Hyperbolic PDEs**

   In processes defined by hyperbolic PDEs, information in the domain propagates at a finite speed. The solution variables take on a wave-like behaviour and can be represented as a superposition of multiple simple waves.

2. **Parabolic PDEs**

   In processes defined by parabolic PDEs, information travels forward in time. A time-marching approach can be taken to obtain the solutions, which take on a damped wave-like behaviour.

3. **Elliptic PDEs**

   Elliptic PDEs describe equilibrium phenomena. The information travels through the domain at infinite speeds, and the solutions do not demonstrate wave-like behaviour.

We will first consider first order equations, with particular emphasis on the concept of characteristics.

6.1 First order equations

The general first order quasi-linear partial differential equation in two-dimensions is

$$a\frac{\partial u}{\partial x} + b\frac{\partial u}{\partial y} = c \tag{10}$$

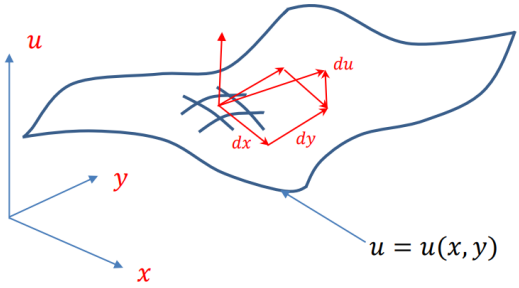


Figure 8: The solution of Eq. (10) as a surface in 3D space

where x and y are the independent variables, $a = a(x, y, u), b = b(x, y, u)$, and $c = c(x, y, u)$ are known functions, and the term quasi-linear refers to that the derivatives of u (dependent variable) appear in a linear combination.
The solution $u = u(x, y)$ is now a surface in 3D space. See Figure 8. We can write small variation in u as

$$du = \frac{\partial u}{\partial x}dx + \frac{\partial u}{\partial y}dy \tag{11}$$

Rewriting Eq. (10) and Eq. (11), we have

$$\left(\frac{\partial u}{\partial x}, \frac{\partial u}{\partial y}, -1\right) \cdot (a, b, c) = 0 \tag{12}$$

and

$$\left(\frac{\partial u}{\partial x}, \frac{\partial u}{\partial y}, -1\right) \cdot (dx, dy, du) = 0 \tag{13}$$

Notice that (dx, dy, du) is a vector tangential to the surface $u = u(x, y)$, Equation (13) shows that the vector $(\partial u/\partial x, \partial u/\partial y, -1)$ is a normal to the surface $u = u(x, y)$ and Eq. (12) shows that the vector (a, b, c) is tangent to the surface at every point. (a, b, c) must therefore lie on the surface. In other words, for a given point (x_0, y_0, z_0), move in the tangential direction (a, b, c) for a infinitesimal distance ds, the end point $(x_0 + ads, y_0 + bds, z_0 + cds)$ must lie on the surface, which is the solution of the PDE. Following this process, the problem can be reformulated as:

Given $x = x_0$, $y = y_0$, $z = z_0$, find the curve $dx = ads$, $dy = bds$ on which the function $z = z(x, y)$ can be solved by $dz = c(x, y, z)ds$. Eliminating ds from $dx = ads$, $dy = bds$ gives:

$$\frac{dx}{dy} = \frac{a}{b} \tag{14}$$

which is an equation for a curve in the $x - y$ plane. The curves are called the *characteristics* of the partial differential equation. Along the curve, the solution

of the PDE can be found as $z(x,y) - z_0 = \int c(x,y,z)ds$, which reduces to a ODE on the characteristics.

The characteristics have considerable significance in the theory of differential equations. Notice, in particular, that the solution does not depend on the derivative of the function perpendicular to the characteristic line and is governed by ordinary differential equations along the characteristics, whose position is also governed by ordinary differential equations. Therefore, we could solve the partial differential equation by selecting a few initial points and solving ordinary differential equations to predict the position and function value of these curves. Given $x = x(s_o), y = y(s_o)$ and $u = u(s_o)$ we can solve for $x = x(s_o, s)$, $y = y(s_o, s)$ and $u = u(s_o, s)$. Selecting more initial points gives us the function at more points but, since no information are transferred between the characteristics, the accuracy is not affected. A couple of examples may help to develop a better intuitive feel for the importance of characteristics. Consider the linear advection equation

$$\frac{\partial u}{\partial t} + U\frac{\partial u}{\partial x} = 0 \tag{15}$$

Here we use t and x to denote the coordinates (independent variables) of the two-dimensional space. The characteristics are given by

$$\frac{dx}{dt} = U; \quad \text{and} \quad du = 0 \tag{16}$$

since U is a constant, the characteristics are simply lines of constant slope. Furthermore, along each line the value of the function does not change. The solution to this equation is therefore that the initial distribution of u is simply translated by the velocity U. Notice, in particular, that since there is no communication between the solution on different characteristics, that the solution can jump discontinuously from one characteristic to an other. To see an example of a more complicated behavior, consider the nonlinear advection equation

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = 0 \tag{17}$$

Again we use t and x to denote the coordinates of the two-dimensional space. The characteristics (level curves) are given by

$$\frac{dx}{dt} = u; \quad \text{and} \quad du = 0 \tag{18}$$

This example differs from the previous one in that the slope of the characteristics depends on the value of u. since u does not change along the characteristics, they must be lines of constant slope, with the slope determined by the initial value of u. Points with large u propagate therefore faster than points with a smaller value of x. This can lead to some interesting behavior. See Figure 3. In both of the above examples, the value of the function remained constant on the characteristics. This does not have to be the case, and adding, for example, a constant on the right-hand side produces solutions that grow (or decay) linearly. Adding $\pm u$ on the right-hand side leads to exponentially growing or decaying solutions.

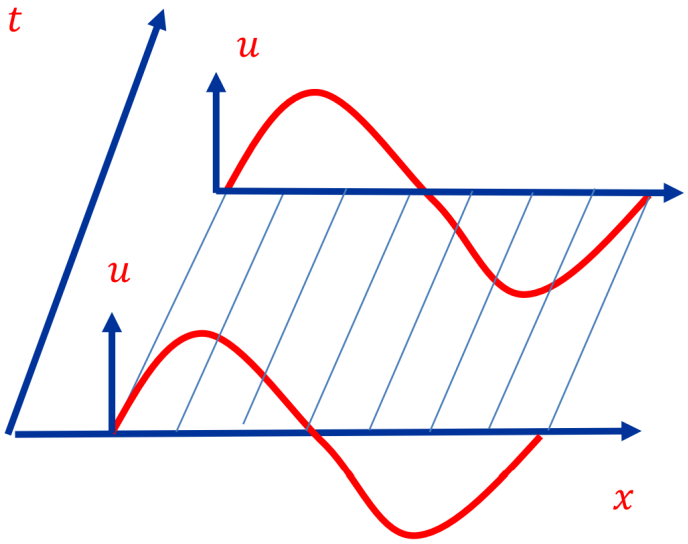


Figure 9: The solution of the linear advection equation

Generally, first-order equations can produce unphysical solutions. Look again at the second example above. While each characteristic has a constant slope, the slope varies from place to place. In our example, it has had the effect of steepening the initial sine wave. The points with higher value of u move faster than points with a lower value and the characteristics crowd around the point where u changes sign. If we continue the characteristics, it is obvious that they will cross and since the value of u is constant along each characteristic, the solution is now double valued at the point of crossing. Generally, this is unacceptable behavior. (If u represented the depth of a water channel, then one might argue that the crossing of characteristics corresponding to breaking and overturning waves, but if u represents temperature or density, then such interpretation is clearly impossible.) The reason for this unphysical behavior is usually that the physical modeling is incomplete. Terms of a higher order that prevent the characteristics from crossing have been assumed to be so small that they could be neglected. The obvious solution adds in the missing terms - has several disadvantages. The first is that these terms are always of higher order and the equations are no longer first-order ones. The second disadvantage is that these terms may represent physics that require finer resolution that we want to devote to the problem.

Although a single first order partial differential equation is rarely encountered in CFD (advection of a passive scalar in a prescribed velocity field is one example), we will make frequent reference to the above two equations as model equations for the advection equations.

6.2 PDE Classification using Eigenvalue Method

Systems of first-order PDEs with more than two independent variables can be recast in matrix form. Their classification involves finding eigenvalues of the resulting matrix.

Let us look at how we can classify these first order PDEs. Taking a general system of PDEs in two-dimensions as

$$a_1\frac{\partial u}{\partial x} + b_1\frac{\partial u}{\partial y} + c_1\frac{\partial v}{\partial x} + d_1\frac{\partial v}{\partial y} = 0$$
$$a_2\frac{\partial u}{\partial x} + b_2\frac{\partial u}{\partial y} + c_2\frac{\partial v}{\partial x} + d_2\frac{\partial v}{\partial y} = 0$$

This can be re-written as

$$\begin{bmatrix} a_1 & c_1 \\ a_2 & c_2 \end{bmatrix}\frac{\partial \mathbf{W}}{\partial x} + \begin{bmatrix} b_1 & d_1 \\ b_2 & d_2 \end{bmatrix}\frac{\partial \mathbf{W}}{\partial y} = 0 \quad \text{where} \quad \mathbf{W} = \begin{pmatrix} u \\ v \end{pmatrix}$$

Which can be simplified as

$$\frac{\partial \mathbf{W}}{\partial x} + \begin{bmatrix} a_1 & c_1 \\ a_2 & c_2 \end{bmatrix}^{-1}\begin{bmatrix} b_1 & d_1 \\ b_2 & d_2 \end{bmatrix}\frac{\partial \mathbf{W}}{\partial y} = 0$$

The eigenvalues of the matrix

$$\begin{bmatrix} a_1 & c_1 \\ a_2 & c_2 \end{bmatrix}^{-1}\begin{bmatrix} b_1 & d_1 \\ b_2 & d_2 \end{bmatrix}$$

can then be used to classify the PDE system.

1. If all eigenvalues are *real*, then the system is *hyperbolic*.
2. If all eigenvalues are *complex*, then the system is *elliptic*.
3. If the eigenvalues are mixed real and complex values, then the system is mixed *elliptic-hyperbolic*.

6.3 Second order equations

We now turn to second order equations. The basic tool in our analysis is the fact that any high order equation can be rewritten as a system of first order equations. Consider the general, quasi-linear, second order equation in two dimensions.

$$a\frac{\partial^2 u}{\partial x^2} + b\frac{\partial^2 u}{\partial x \partial y} + c\frac{\partial^2 u}{\partial y^2} = d \tag{19}$$

where a, b, c, d are all functions of $\partial u/\partial x, \partial u/\partial y, u, x, y$. By defining $\phi = \partial u/\partial x$ and $\psi = \partial u/\partial y$ our equation becomes

$$a\frac{\partial \phi}{\partial x} + b\frac{\partial \phi}{\partial y} + c\frac{\partial \psi}{\partial y} = d \tag{20}$$

and

$$\frac{\partial \psi}{\partial x} - \frac{\partial \phi}{\partial y} = 0 \tag{21}$$

where the second equation follows from that $\partial^2 u/\partial x \partial y = \partial^2 u/\partial y \partial x$. If $a \neq 0$ these equations may be written as

$$\begin{pmatrix} \partial\phi/\partial x \\ \partial\psi/\partial x \end{pmatrix} + \begin{pmatrix} b/a & c/a \\ -1 & 0 \end{pmatrix} \begin{pmatrix} \partial\phi/\partial y \\ \partial\psi/\partial y \end{pmatrix} = \begin{pmatrix} d/a \\ 0 \end{pmatrix} \tag{22}$$

By defining the appropriate vectors, this equation can be written as:

$$\mathbf{u}_x + \mathbf{A}\mathbf{u}_y = \mathbf{s} \tag{23}$$

We now ask whether we can repeat the development for a single first order equation for this system. Specifically: Are there lines in the $x - y$ plane such that the solution (ϕ, ψ) is governed by ordinary differential equations along these lines? (The lines themselves are, of course, also given by ordinary differential equations.)
Generally the solution is a function of both x and y, such that $\phi = \phi(x, y)$, for example. Along each line, if such lines exist, y is a function of x or $y = y(x)$, so that on each line we can write $\phi = \phi(x, y(x))$. The rate of change of ϕ in the x -direction, but still along the lines, is now the total derivative of ϕ,

$$\frac{d\phi}{dx} = \frac{\partial \phi}{\partial x} + \frac{\partial \phi}{\partial y}\frac{\partial y}{\partial x} = \frac{\partial \phi}{\partial x} + \alpha \frac{\partial \phi}{\partial y} \tag{24}$$

While the equations as written are clearly not in this form (unless $b = 0$ and $c/a = -1$) it may be possible that we can add and subtract the equations until they consist of combinations of terms of the above form. In particular, we write the sum

$$l_1 \left(\frac{\partial \phi}{\partial x} + \frac{b}{a}\frac{\partial \phi}{\partial y} + \frac{c}{a}\frac{\partial \psi}{\partial y} \right) + l_2 \left(\frac{\partial \psi}{\partial x} - \frac{\partial \phi}{\partial y} \right) = l_1 \frac{d}{a} \tag{25}$$

and ask whether this is ever equal to

$$l_1 \left(\frac{\partial \phi}{\partial x} + \alpha \frac{\partial \phi}{\partial y} \right) + l_2 \left(\frac{\partial \psi}{\partial x} + \alpha \frac{\partial \psi}{\partial y} \right) = l_1 \frac{d}{a} \tag{26}$$

for some l 's and α 's. Clearly, this is only true if

$$\begin{array}{c} l_1 \frac{b}{a} - l_2 = l_1 \alpha \\ l_1 \frac{c}{a} = l_2 \alpha \end{array} \tag{27}$$

Or, written in matrix form

$$\begin{pmatrix} b/a - \alpha & -1 \\ c/a & -\alpha \end{pmatrix} \begin{pmatrix} l_1 \\ l_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \tag{28}$$

A solution for this system exist only if the determinant of the matrix vanishes, thus giving an eigenvalue problem for the characteristics α's. This results in

$$-\alpha \left(\frac{b}{a} - \alpha \right) + \frac{c}{a} = 0, \quad \text{or} \quad \alpha = \frac{1}{2a} \left(b \pm \sqrt{b^2 - 4ac} \right) \tag{29}$$

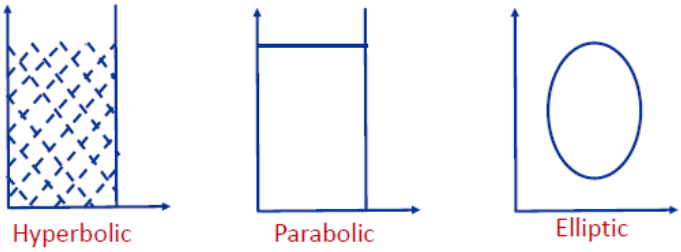


Figure 10: Schematic aspect of the three types of second order equations

So, only if the quantity under the square root is positive can we write the second order equation (or the system of first order equations) in a characteristic form where the evolution of the function is governed by an ordinary differential equations along each characteristic.
The results of our investigation is therefore that in some cases there are lines in the $x-y$ plane such that the position of these lines and the evolution of the solution along these lines are governed by ordinary differential equations and sometimes there are no such lines. This leads to a general classification of second order equations, depending on the sign of discriminant b^2-4ac:
$b^2-4ac>0$ there are two real characteristics; *hyperbolic*
$b^2-4ac=0$ there is one real characteristics; *parabolic*
$b^2-4ac<0$ there are no real characteristics; *elliptic*

The behavior of the solution of the equations depends strongly on what type they are. These considerations are not limited to only two first order equations. The system

$$\frac{\partial u_i}{\partial t}+A_{ij}\frac{\partial u_j}{\partial x}=d_i \tag{30}$$

can be written in the form

$$l_i\left(\frac{\partial u_i}{\partial t}+\alpha\frac{\partial u_j}{\partial x}\right)=l_i d_i \tag{31}$$

if $l_i A_{ij}=l_j\alpha$, which is true if $\det\left(A_{ji}-\alpha\delta_{ij}\right)=0$

6.4 Classification of Navier-Stokes equation

Incompressible Navier-Stokes equation (NSE) contains physical processes that are

1. Elliptic type
2. Convective type
3. Diffusive type

It is important for us to have this classification because their mathematical behaviour is very different.

In practice, only certain forms of the NSEs have exact solution. The general solution can be obtained *numerically*. To analyze the solution structure of NSEs, we consider the linear subset of NSEs. We summarize their characteristics of the model equations below.

Mathematical	**Corresponding Physics**	**Example**
Hyperbolic (wave-like)	Convection/Advection	$\frac{\partial u}{\partial t} + c\frac{\partial u}{\partial x} = 0$
Parabolic	Dissipation/Diffusion	$\frac{\partial u}{\partial t} = v\frac{\partial^2 u}{\partial x^2}$
Elliptic	Equilibrium	$\frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} = \mathrm{f}(x, y)$

The convection and diffusion equations will be mainly considered in the following lectures. Note also that some flows exhibit mixed behavior, for example, elliptic and hyperbolic. The complicated mixture of elliptic, parabolic and hyperbolic behaviors has implications for the way in which boundary conditions enter into a CFD problem, in particular at locations where flows are bounded by fluid boundaries. There are typically three types of boundary conditions that can be enforced:

- Dirichlet boundary conditions, sometimes called essential boundary conditions, in which the solution u is specified
- Neumann boundary conditions, sometimes called natural boundary conditions, in which one of the derivatives u_x or u_y is specified
- Robin boundary conditions, or mixed boundary conditions, in which a combination of solution values and derivative values is specified.

Numerical Approaches

We seek to approximate the exact solution of Navier-Stokes equations (NSE) by using numerical methods. Basically, the process involves 3 stages

1. Spatial Discretization
2. Temporal Discretization
3. Solutions of resultant ordinary differential equations (ODE) or algebraic equations

In the following, we will consider model equations rather than the full NSE.