

MATHEMATICAL TRIPOS, PART II  
COMPUTATIONAL PROJECT

# Hyperbolic Partial Differential Equations



UNIVERSITY OF CAMBRIDGE

August 10, 2021

## Contents

<b>1</b>	<b>Question 1</b>	<b>2</b>
<b>2</b>	<b>Question 2</b>	<b>3</b>
<b>3</b>	<b>Question 3</b>	<b>6</b>
<b>4</b>	<b>Question 4</b>	<b>7</b>
<b>5</b>	<b>Question 5</b>	<b>8</b>
	<b>References</b>	<b>8</b>
<b>A</b>	<b>Codes</b>	<b>9</b>
A.1	P1 . . . . .	9
A.2	P2 . . . . .	10
A.3	P3 . . . . .	11

## 1 Question 1

$$u_t + f(u)_x = 0, \quad f(u) = \frac{1}{2}u^2 \quad (1)$$

The characteristic curves of 1 are defined as  $C : \frac{dx}{dt} = u$ . Along the characteristic curves,  $\frac{Du}{Dt}\Big|_C = u_t + \frac{dx}{dt}u_x = u_t + uu_x = 0$ . Hence  $u$  is constant along the characteristic curves. Therefore,  $x - u(x_0, 0)t = x_0$  are the characteristics. If we choose initial condition  $u(x, 0) = \sin(x)$ , then we get a shock at  $t = t_s$ , as shown in 1.

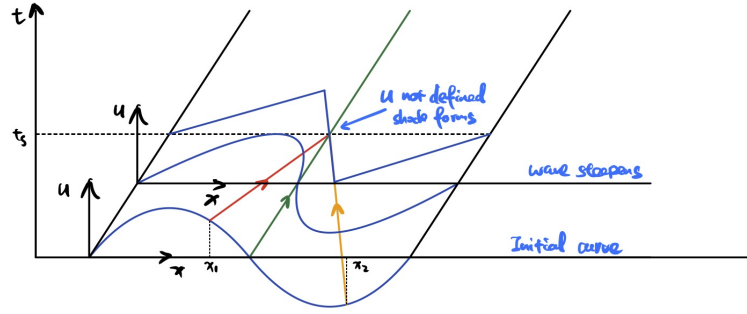


Figure 1.1: Shock formation with initial data  $\sin(x)$

Since the characteristics (yellow and red) cross, at  $t_s$ , the position  $x$  becomes indistinguishable from the initial position  $x_0$ , and we have  $\frac{\partial x}{\partial t} \rightarrow 0$  as  $t \rightarrow t_s$ . From 1 and also by chain rule we can conclude  $\frac{\partial u}{\partial x} = \frac{\frac{du}{dx_0}}{\frac{\partial x}{\partial x_0}} \rightarrow \infty$ .

The condition for a shock to form is that there exists  $t_s$  such that  $\frac{\partial x}{\partial t}\Big|_{t_s} = 0$ . Assume at  $t = t_s$  a shock is formed for the first time, then by the characteristic equation,  $t_s = \frac{1}{\max_{x_0} \left( -\frac{du}{dx_0}(x_0, 0) \right)}$ .

Suppose we have  $u(x, 0) = \text{const.}$ , then there is clearly no shock formation as all characteristics are parallel. Suppose we have  $u(x, 0) = H(x) \ln(x+1)$ , then there is no shock either, as  $t$  is positive and the characteristics have decreasing gradients as  $x$  grows. They don't intersect.

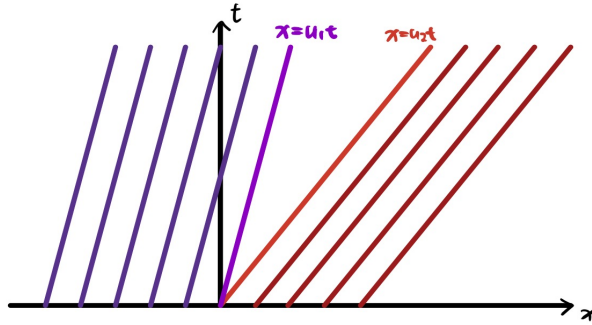


Figure 1.2: Characteristics of step-like Initial data

Given the initial condition:

$$u(x, 0) = \begin{cases} u_1 & x < 0 \\ u_2 & x > 0 \end{cases}$$

with  $u_1 < u_2$ . One weak solution is

$$u(x, t) = \begin{cases} u_1 & x < u_1 t \\ \frac{x}{t} & u_1 t < x < u_2 t \\ u_2 & x > u_2 t \end{cases}$$

The other weak solution is a shock wave, since the characteristics travel away from the shock, it is not casual, hence not possible. From figure 2 there is no possibility of shock formation as the characteristics do not intersect.

## 2 Question 2

The characteristics with the given boundary conditions can be found in 3.

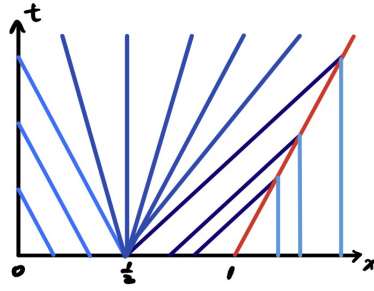


Figure 2.1: characteristics of the initial data (2)

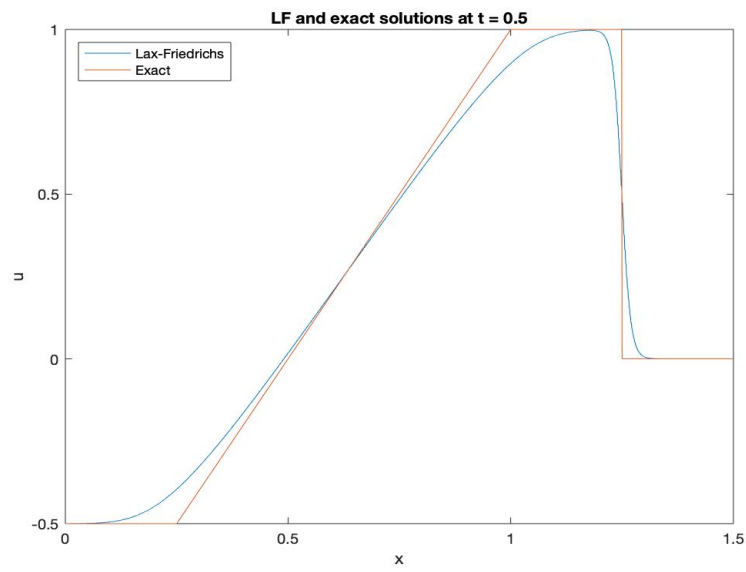


Figure 2.2: Exact solution and Lax-Friedrichs scheme at  $t = 0.5$

At  $x_0 = 1$  there is a shock. By Rankine-Hugoniot:

$$\frac{dx}{dt} = \frac{[u^2]_-^+}{2[u]_-^+} = \frac{u^+ + u^-}{2} = \frac{1}{2}, \quad x_0 = 1$$

Therefore the solution is: when  $t \leq 1$

$$u(x, t) = \begin{cases} -\frac{1}{2} & x < \frac{1-t}{2} \\ \frac{2x-1}{2t} & \frac{1-t}{2} \leq x < t + \frac{1}{2} \\ 1 & t + \frac{1}{2} \leq x < \frac{t}{2} + 1 \\ 0 & x \geq \frac{t}{2} + 1 \end{cases}$$

and when  $t > 1$

$$u(x, t) = \begin{cases} -\frac{1}{2} & x < \frac{1-t}{2} \\ \frac{2x-1}{2t} & \frac{1-t}{2} \leq x < \frac{t}{2} + 1 \\ 0 & x \geq \frac{t}{2} + 1 \end{cases}$$

The numerical solution is as in 4. By calculating the truncation error analysis. By [1], the order of accuracy of the scheme is defined as the error caused by approximating the time derivative. We have

$$kT_h^k = k(u_t + (f(u))_x) + k^2 u_t - \frac{h^2}{2} u_{xx} + \mathcal{O}(kh^2),$$

$$i.e. \quad T_h^k = k u_t - \frac{h^2}{2k} u_{xx} + \mathcal{O}(h^2 + k^2).$$

Therefore, the order of accuracy is  $\mathcal{O}(k + \frac{h^2}{k})$  and hence Lax-Friedrichs scheme is of first order. It can be shown similarly that the Richtmyer scheme is of second order.

### 3 Question 3

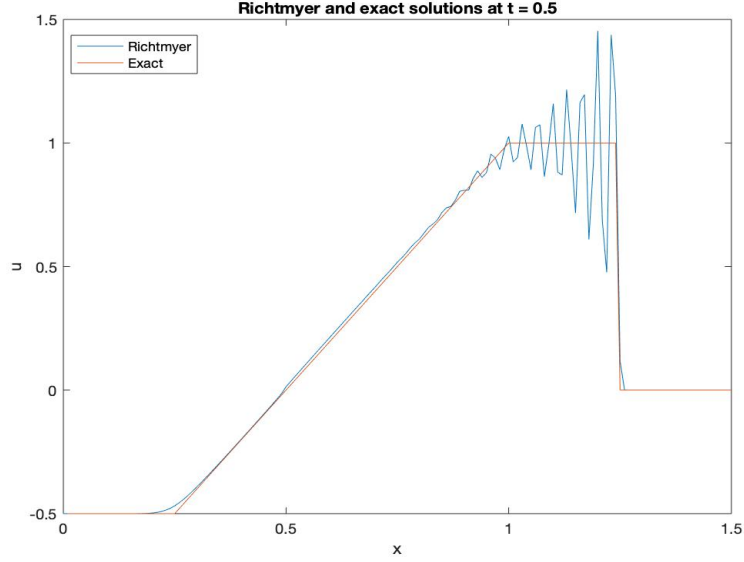


Figure 3.1: Exact solution and Richtmyer scheme at  $t = 0.5$

From Figure 3.1 it is clear that there are spurious oscillations near the point of discontinuity, which causes the inaccuracy of the numerical method near points of high gradient. At the same time, the Richtmyer scheme is more accurate at places where the solution is smoother, as a result of its higher order of accuracy.

From p.437 of [1], a scheme

$$u_i^{n+1} = H(u_{i-k_L+1}^n, \dots, u_{i+k_R}^n)$$

with  $k_L$  and  $k_R$  non-negative integers is said to be monotone if

$$\frac{\partial H}{\partial u_j^n} \geq 0 \quad \forall j$$

. For conservation laws, this is just a discrete version of the property that the order of two different initial data is preserved, which is the case for the exact solution. Also from [1], on p.439, it is shown that the Lax-Friedrichs method is monotone, and it can also be shown that Richtmyer is not monotone<sup>1</sup>, which is the cause of the spurious oscillations near the discontinuity.

---

<sup>1</sup>One can argue that because of the Godunov's Theorem (p.444), and the fact that the Richtmyer scheme is second order, it is not monotone.

## 4 Question 4

For a more careful treatment please refer to 13.7 of [1]

- (i) If there is sharp change in the gradient at a point  $x$ , we choose a **small** neighbourhood of it. If the sign is preserved, for some  $i$  such that  $ih \leq x < (i+1)h$ , we have  $|u_{i+1}^n - u_i^n| \gg |u_i^n - u_{i-1}^n|$  and  $|u_{i+1}^n - u_i^n| \gg |u_{i+2}^n - u_{i+1}^n|$ . Then we have  $r_i^n \approx 0$ , and thus  $F \approx F^{LO}$ . If the sign is reversed, then  $r_i^n = 0$ , and  $F = F^{LO}$ . Therefore, the overall flux is predominantly low order accurate.
- (ii) If there is slight change in gradient at a point  $x$ , then in a **small** neighbourhood <sup>2</sup> of it, for some  $i$  such that  $ih \leq x < (i+1)h$ , we have  $|u_{i+1}^n - u_i^n| \approx |u_i^n - u_{i-1}^n|$  and  $|u_{i+1}^n - u_i^n| \approx |u_{i+2}^n - u_{i+1}^n|$ . Then we have  $r_i^n \approx 1$ , and thus the overall flux  $F \approx F^{HI}$ , which is predominantly high order accurate.

It is desirable because we can choose the low order flux to be a monotone one, and the high order flux to be one with order of accuracy  $\geq 2$ . Therefore, in places where there are sharp changes in gradient, we have a monotone method and hence no spurious oscillations; in places where there are slight changes in gradient, we have a high order method hence higher resolution.

---

<sup>2</sup>small in the sense that any swap of sign is excluded



## 5 Question 5

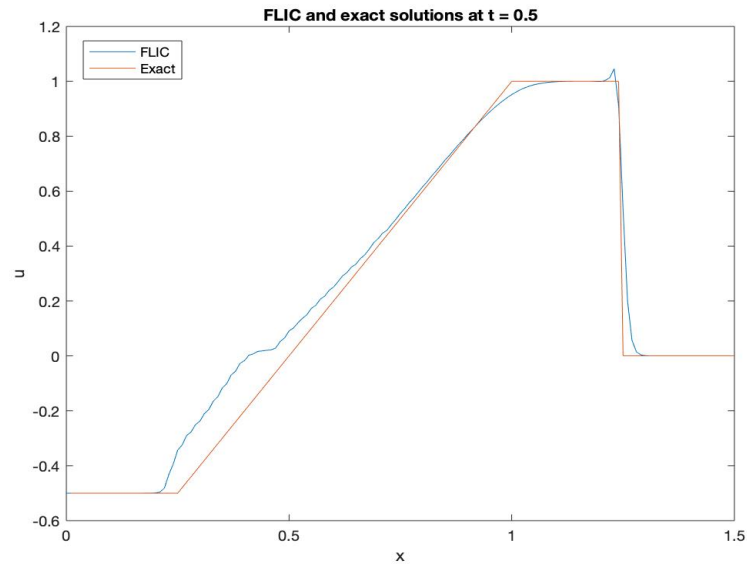


Figure 5.1: Exact solution and FLIC scheme at  $t = 0.5$

By comparing with 2 and 4, though it does not fit better than the previous schemes at all points, the scheme has worked better at the discontinuity. Clearly the error is smaller there and there has been no spurious oscillations. For a better scheme we may try other limiter functions.

## References

- [1] Toro, E.F. *Riemann Solvers and Numerical Methods for Fluid Dynamics: a practical introduction* Springer-Verlag, 1999.

## A Codes

### A.1 P1

```
xmin = 0;
xmax = 1.5;
tmax = 0.5;
h = 0.001;
k = 0.1*h;
N = round((xmax-xmin)/h)+1;
xgrid = xmin:h:xmax;
M = round(tmax/k);

syms u0(x)
u0(x) = piecewise(x<0, 0, ...
    0<=x & x<0.5, -0.5, 0.5<=x & x<1, 1, x>=1 & x<=1.5, 0, ...
    x>1.5, 0);
uinit = double(u0(xgrid));
f = @(x) 0.5*x^2;

a1 = @(t) (1-t)/2;
a2 = @(t) t+1/2;
a3 = @(t) t/2+1;

b1 = a1(1/2);
b2 = a2(1/2);
b3 = a3(1/2);

syms uexact(x)
v(x) = piecewise(x>0 & x<b1, -1/2, ...
    b1<=x & x<b2, (2*x-1)/(2*1/2), ...
    b2<=x & x<b3, 1, ...
    x>=b3, 0);

uexact = double(v(xgrid));

u1 = uinit;
u2 = u1;

Flux = zeros(1,N);
```

```

for m = 1:M
    for i = 2:N-1
        Flux(i) = 0.5*(f(u1(i-1))+f(u1(i))) ...
            +0.5*(h/k)*(u1(i-1)-u1(i));
        Flux(i+1) = 0.5*(f(u1(i))+f(u1(i+1))) ...
            +0.5*(h/k)*(u1(i)-u1(i+1));
        u2(i) = u1(i) + (k/h)*(Flux(i) - Flux(i+1));
    end
    u2(1) = u2(2);
    u2(N) = u2(N-1);
    u1 = u2;
    plot(xgrid,u2)
    pause(0.001)
end
hold on
plot(xgrid,uexact)
title('LF and exact solutions at t=0.5')
xlabel x
ylabel u

```

## A.2 P2

```

xmin = 0;
xmax = 1.5;
tmax = 0.5;
h = 0.01;
k = 0.1*h;
N = round((xmax-xmin)/h)+1;
xgrid = xmin:h:xmax;
M = round(tmax/k);

syms u0(x)
u0(x) = piecewise(x<0, 0, ...
    0<=x & x<0.5, -0.5, 0.5<=x & x<1, ...
    1, x>=1 & x<=1.5, 0, ...
    x>1.5, 0);
uinit = double(u0(xgrid));
f = @(x) 0.5*x^2;

a1 = @(t) (1-t)/2;
a2 = @(t) t+1/2;
a3 = @(t) t/2+1;

```

```

b1 = a1(1/2);
b2 = a2(1/2);
b3 = a3(1/2);

syms uexact(x)
v(x) = piecewise(x>0 & x<b1, -1/2, ...
    b1<=x & x<b2, (2*x-1)/(2*1/2), ...
    b2<=x & x<b3, 1, ...
    x>=b3, 0);

uexact = double(v(xgrid));

u1 = uinit;
u2 = u1;

Flux = zeros(1,N);
URi = Flux;
for m = 1:M
    for i = 2:N-1
        URi(i) = 0.5*(u1(i-1)+u1(i)) ...
            +0.5*(k/h)*(f(u1(i-1))-f(u1(i)));
        URi(i+1) = 0.5*(u1(i)+u1(i+1)) ...
            +0.5*(k/h)*(f(u1(i))-f(u1(i+1)));
        Flux(i) = f(URi(i));
        Flux(i+1) = f(URi(i+1));
        u2(i) = u1(i) + (k/h)*(Flux(i) - Flux(i+1));
    end
    u2(1) = u2(2);
    u2(N) = u2(N-1);
    u1 = u2;
    plot(xgrid,u2)
    pause(0.001)
end
hold on
plot(xgrid,uexact)
title('Richtmyer and exact solutions at t=0.5')
xlabel x
ylabel u
legend

A.3 P3

xmin = 0;

```

```

xmax = 1.5;
tmax = 0.5;
h = 0.01;
k = 0.5*h;
N = round((xmax-xmin)/h)+1;
xgrid = xmin:h:xmax;
M = round(tmax/k);

syms u0(x)
u0(x) = piecewise(x<0, 0, ...
    0<=x & x<0.5, -0.5, 0.5<=x & x<1, ...
    1, x>=1 & x<=1.5, 0, ...
    x>1.5, 0);
uinit = double(u0(xgrid));
f = @(x) 0.5*x^2;

a1 = @(t) (1-t)/2;
a2 = @(t) t+1/2;
a3 = @(t) t/2+1;

b1 = a1(1/2);
b2 = a2(1/2);
b3 = a3(1/2);

syms uexact(x)
v(x) = piecewise(x>0 & x<b1, -1/2, ...
    b1<=x & x<b2, (2*x-1)/(2*1/2), ...
    b2<=x & x<b3, 1, ...
    x>=b3, 0);

uexact = double(v(xgrid));

syms limiter(r)
limiter(r) = piecewise(r<=0, 0, r>=0 & r<=1, r, r>=1, 1);

u1 = uinit;
u2 = u1;

F = zeros(1,N);
FRi = F;
FLF = F;
URi = F;

```

```

FHI = F;
FLO = F;
r = F;
phi = ones(1,N);

for m = 1:M
    for i = 2:N-3

        URi(i) = 0.5*(u1(i-1)+u1(i)) ...
            +0.5*(k/h)*(f(u1(i-1))-f(u1(i)));
        URi(i+1) = 0.5*(u1(i)+u1(i+1)) ...
            +0.5*(k/h)*(f(u1(i))-f(u1(i+1)));
        FRi(i) = f(URi(i));
        FRi(i+1) = f(URi(i+1));

        FLF(i) = 0.5*(f(u1(i-1))+f(u1(i))) ...
            +0.5*(h/k)*(u1(i-1)-u1(i));
        FLF(i+1) = 0.5*(f(u1(i))+f(u1(i+1))) ...
            +0.5*(h/k)*(u1(i)-u1(i+1));

        FHI(i) = FRi(i);
        FHI(i+1) = FRi(i+1);

        FLO(i) = 0.5*(FLF(i)+FRi(i));
        FLO(i+1) = 0.5*(FLF(i+1)+FRi(i+1));

        if (u1(i)-u1(i-1) == 0 && u1(i+1)-u1(i) == 0)
            delta1 = 1;
        elseif (u1(i+1)-u1(i) == 0 && u1(i)-u1(i-1) > 0)
            delta1 = 2;
        elseif (u1(i+1)-u1(i) == 0 && u1(i)-u1(i-1) < 0)
            delta1 = -2;
        else
            delta1 = (u1(i)-u1(i-1))/(u1(i+1)-u1(i));
        end

        if (u1(i+1)-u1(i) == 0 && u1(i+2)-u1(i+1) == 0)
            delta2 = 1;
        elseif (u1(i+2)-u1(i+1) > 0 && u1(i+1)-u1(i) == 0)
            delta2 = 2;
        elseif (u1(i+2)-u1(i+1) < 0 && u1(i+1)-u1(i) == 0)
            delta2 = -2;
        else
            delta2 = (u1(i+2)-u1(i+1))/(u1(i+1)-u1(i));
    end
end

```

```

end

if (u1(i+3)-u1(i+2) == 0 && u1(i+2)-u1(i+1) == 0)
    delta3 = 1;
elseif (u1(i+3)-u1(i+2) > 0 && u1(i+2)-u1(i+1) == 0)
    delta3 = 2;
elseif (u1(i+3)-u1(i+2) < 0 && u1(i+2)-u1(i+1) == 0)
    delta3 = -2;
else
    delta3 = (u1(i+3)-u1(i+2))/(u1(i+2)-u1(i+1));
end

r(i) = min(delta1 , delta2) ;
r(i+1) = min(delta2 , delta3) ;

phi(i) = double(limiter(r(i)));
phi(i+1) = double(limiter(r(i+1)));

F(i) = FLO(i)+phi(i)*(FHI(i)-FLO(i));
F(i+1) = FLO(i+1)+phi(i+1)*(FHI(i+1)-FLO(i+1));

u2(i) = u1(i) + (k/h)*(F(i)-F(i+1));
end
u2(1) = u2(2);
u2(N-2) = u2(N-3);
u2(N-1) = u2(N-2);
u2(N) = u2(N-1);
u1 = u2;
plot(xgrid,u2)
pause(0.001)
end
hold on
plot(xgrid,uexact)
title('FLIC_and_exact_solutions_at_t=0.5')
xlabel x
ylabel u

```