Handout 3

Nonlinear convection

The simplest model for nonlinear convection is the

Inviscid Burgers equation

It is given by

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

So the A in the linear convection equation has been replaced by u. This makes the equation nonlinear. Now the convection speed is not a constant.

Using the upwind scheme as before, we get

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} + u_i^n \frac{u_i^n - u_{i-1}^n}{\Delta x} = 0$$

$$u_i^{n+1} = u_i^n - u_i^n \frac{\Delta t}{\Delta x}\left(u_i^n - u_{i-1}^n\right)$$

Now instead of the usual CFL number $A\Delta t/\Delta x$, we have $u\Delta t/\Delta x$. Because the convection speed is a variable and the equations are nonlinear it becomes difficult to perform a stability analysis.

The solution for an initial pulse is shown in Fig. 17. The

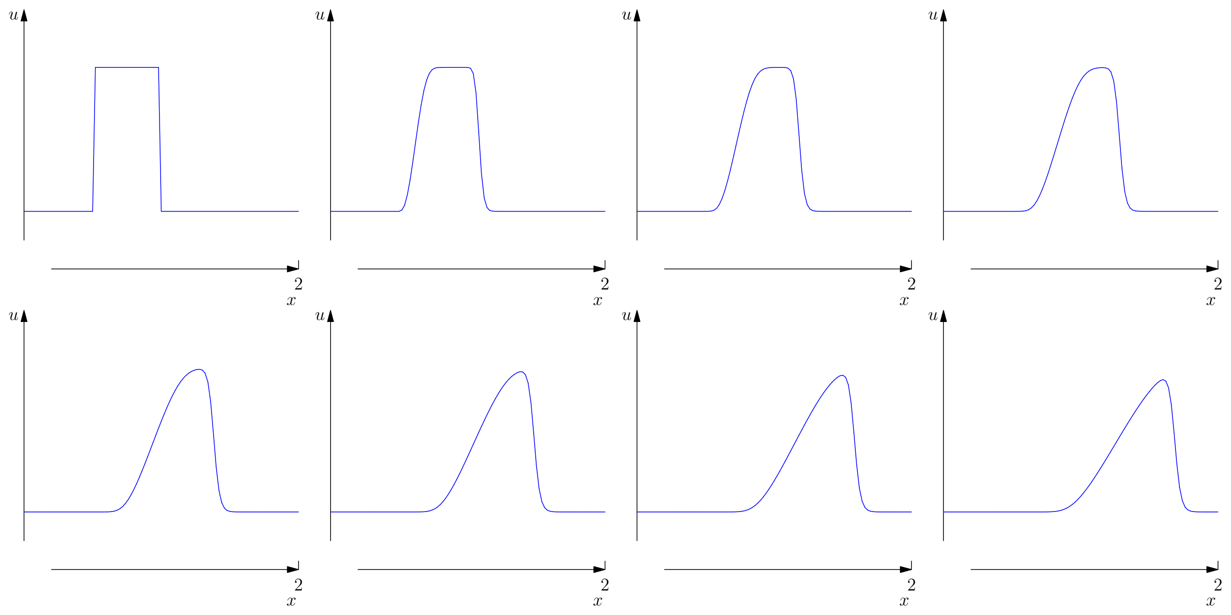


Figure 17: Inviscid Burgers equation solved by the upwind scheme. $t_{\max} = 1$, $dt = 1/140$, $dx = 0.02$. Initial condition: $u(x) = 1.0$ if $(0.5 < x < 1)$ and $O.5$ otherwise.

pulse convects to the right but unlike the linear convection case it also distorts significantly. This is because of nonlinearity. The convection speed u is not a constant, the shift is greatest where the velocity is greatest.

We know that the upwind scheme introduces artificial dissipation. As with the linear convection case let us try to resolve this problem by using the MacCormack scheme. It can be derived by first writing the inviscid Burgers equation in the following form

$$\frac{\partial u}{\partial t} + \frac{\partial}{\partial x}\left(\frac{u^2}{2}\right) = 0$$

The MacCormack differencing is now given by

Predictor:
$$\overline{u_i^{n+1}} = u_i^n - \frac{\Delta t}{\Delta x}\left\{\left(\frac{u^2}{2}\right)_{i+1}^n - \left(\frac{u^2}{2}\right)_i^n\right\}$$

Corrector:
$$u_i^{n+1} = \frac{1}{2}\left[u_i^n + \overline{u_i^{n+1}} - \frac{\Delta t}{\Delta x}\left\{\overline{\left(\frac{u^2}{2}\right)_i^{n+1}} - \overline{\left(\frac{u^2}{2}\right)_{i-1}^{n+1}}\right\}\right]$$

Starting from the same initial condition as before the solution using the MacCormack scheme is shown in Fig. 18. As before we find that although the scheme is less dissipative (the shocks are sharper than the upwind case), the scheme introduces a dispersive error. Note that for a smoother initial condition the dispersion error will be small. We will quantify the effect of numerical diffusion and dispersion on the solution in the next section.

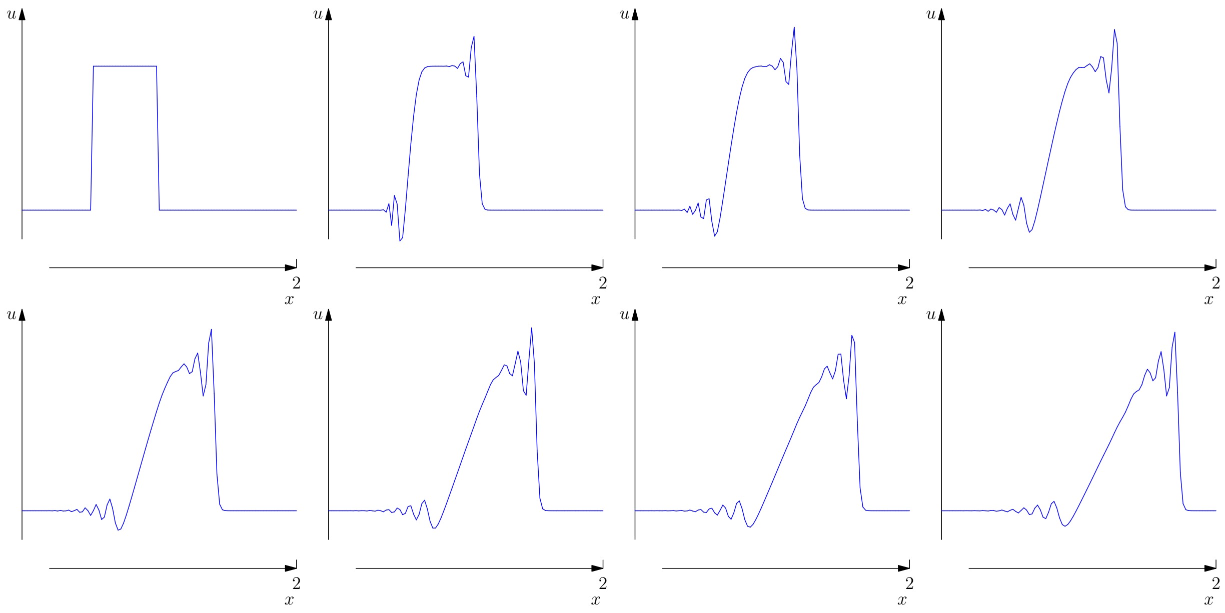


Figure 18: Inviscid Burgers equation solved by the MacCormack scheme. $t_{\max} = 1$, $dt = 1/140$, $dx = 0.02$.

Nonlinear convection and diffusion

If we add the diffusion term we had used in the heat equation to the inviscid Burgers equation, we obtain the

Viscous Burgers equation

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = \nu\frac{\partial^2 u}{\partial x^2}$$

Note that this is very similar to the 1-D incompressible momentum conservation equation (Navier-Stokes):

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = -\frac{1}{\rho}\frac{\partial p}{\partial x} + \nu\frac{\partial^2 u}{\partial x^2}$$

The only difference is that the pressure gradient term is absent. So we can relate what we have been doing with our model problems to the fluid flow equations.

If we discretise the left hand side using the upwind scheme as we did for the inviscid Burgers equation and the right by using a second-order central difference scheme we get:

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} + u_i^n\frac{u_i^n - u_{i-1}^n}{\Delta x} = \nu\frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{\Delta x^2}$$

As before once the initial conditions are known we can march forward in time to obtain the solution at subsequent times

$$u_i^{n+1} = u_i^n - u_i^n\frac{\Delta t}{\Delta x}\left(u_i^n - u_{i-1}^n\right) + \nu\frac{\Delta t}{\Delta x^2}\left(u_{i+1}^n - 2u_i^n + u_{i-1}^n\right)$$

As we have introduced physical dissipation in the problem it is difficult to distinguish between the physical and numerical dissipation when investigating the accuracy of the solution. To help with this task, we will choose an initial condition for which the analytical solution to the viscous Burgers equation is known. One such initial condition is

$$\begin{aligned} u(x,0) &= -\frac{2\nu}{\phi}\frac{\partial\phi}{\partial x} + 4, \\ \phi(x) &= \exp\left(\frac{-x^2}{4\nu}\right) + \exp\left(\frac{-(x-2\pi)^2}{4\nu}\right) \end{aligned}$$

The analytical solution is given by

$$\begin{aligned} u(x,t) &= -\frac{2\nu}{\phi}\frac{\partial\phi}{\partial x} + 4 \\ \phi(x,t) &= \exp\left(\frac{-(x-4t)^2}{4\nu(t+1)}\right) + \exp\left(\frac{-(x-4t-2\pi)^2}{4\nu(t+1)}\right) \end{aligned}$$

We will solve the problem in the domain $0 \le x \le 2\pi$ with the boundary conditions

$$u(0) = u(2\pi)$$

This represents a *periodic boundary condition*.

The solution using upwind differencing is shown in Fig. 19. As expected the numerical solution introduces artificial dissipation.

As in the previous section, we could use the MacCormack scheme to reduce the artificial dissipation. The MacCormack scheme for the viscous Burgers equation can be obtained by adding a second order central difference formula for the viscous term, $\nu\partial^2 u/\partial x^2$, to the MacCormack scheme for the inviscid Burgers

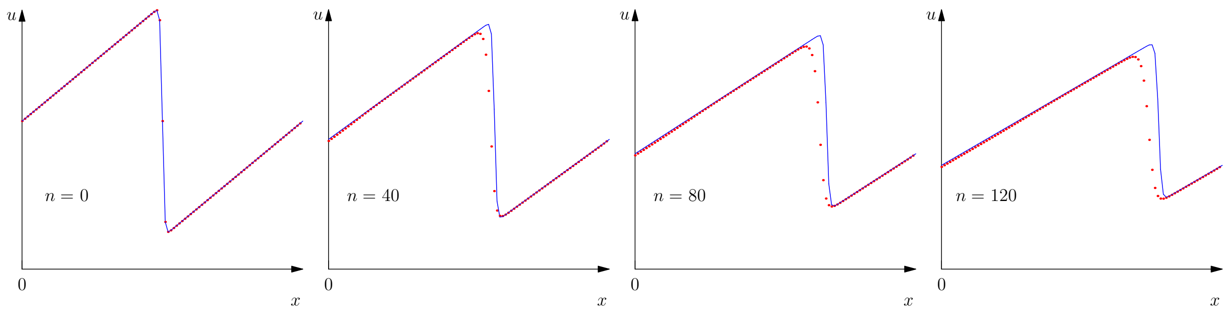


Figure 19: Burgers equation solution using upwind scheme. Solid line: Analytical solution, dots: Numerical solution. Parameters: $nu = 0.07$, $dx = 0.02\pi$, $dt = 0.0036$.

equation:

Predictor:
$$u_i^{\overline{n+1}} = u_i^n - \frac{\Delta t}{\Delta x}\left\{\left(\frac{u^2}{2}\right)_{i+1}^n - \left(\frac{u^2}{2}\right)_i^n\right\} + \frac{\nu\Delta t}{\Delta x^2}\left(u_{i+1}^n - 2u_i^n + u_{i-1}^n\right)$$

Corrector:
$$u_i^{n+1} = \frac{1}{2}\left[u_i^n + u_i^{\overline{n+1}} - \frac{\Delta t}{\Delta x}\left\{\left(\frac{u^2}{2}\right)_i^{\overline{n+1}} - \left(\frac{u^2}{2}\right)_{i-1}^{\overline{n+1}}\right\} + \frac{\nu\Delta t}{\Delta x^2}\left(u_{i+1}^{\overline{n+1}} - 2u_i^{\overline{n+1}} + u_{i-1}^{\overline{n+1}}\right)\right]$$

The solution is shown in Fig. 20. There is very little dissipation as expected. The dispersion error is very low as well. This is because the initial condition is smoother here (has continuous derivatives) compared to the pulse used in the previous section.

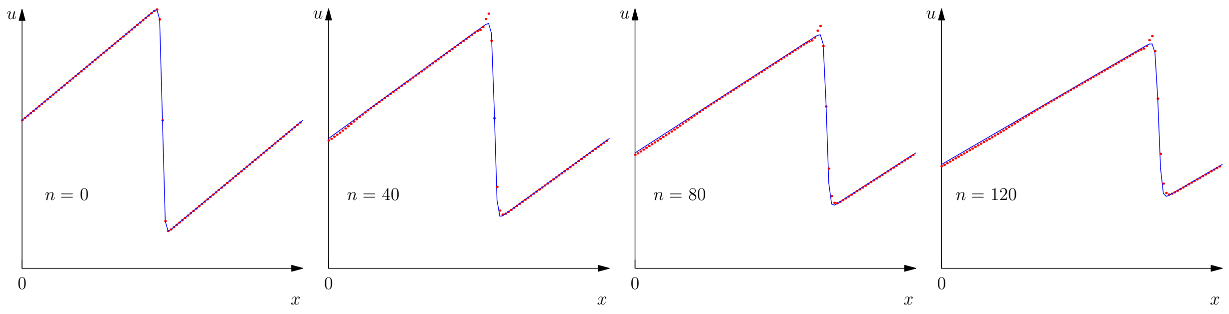


Figure 20: Burgers equation solution using MacCormack scheme.

Equations of fluid flow

For simplicity, let us consider inviscid and one-dimensional *compressible* flow. The equation for conservation for mass is

$$\frac{\partial\rho}{\partial t} + \frac{\partial}{\partial x}(\rho u) = 0$$

The equation for conservation of momentum is given by

$$\frac{\partial(\rho u)}{\partial t} + \frac{\partial}{\partial x}\left(\rho u^2 + p\right) = 0$$

and conservation of energy is given by

$$\frac{\partial E}{\partial t} + \frac{\partial}{\partial x}\left((E + p)u\right) = 0$$

where E is the total energy density given by

$$E = \frac{p}{\gamma - 1} + \frac{\rho}{2} U^2$$

These three conservation equations together are called *Euler equations*. They can be written in the following vector form:

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}}{\partial x} = 0$$

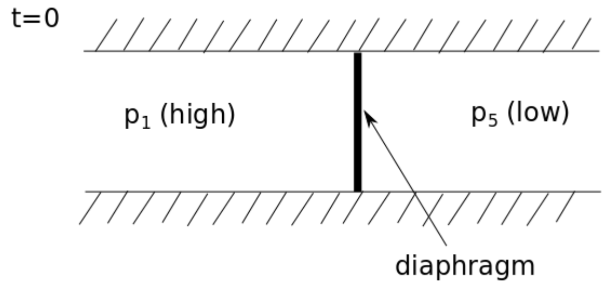
where

$$\mathbf{U} = \begin{Bmatrix} \rho \\ \rho u \\ E \end{Bmatrix}, \qquad \mathbf{F} = \begin{Bmatrix} \rho u \\ \rho u^2 + p \\ (E + p)u \end{Bmatrix}$$

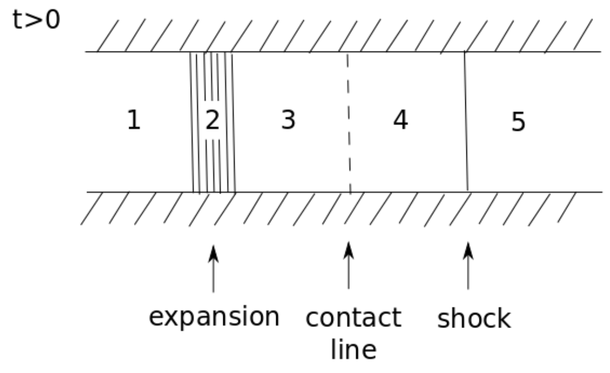
The shock tube problem

Consider an infinitely long tube with two gasses separated by a thin diaphragm. The gas on the left has a high pressure and density whereas the one on the right has lower values for pressure and density. At $t = 0$ the diaphragm bursts and the gas is allowed to flow. We want to calculate what happens to the flow at subsequent times.

We expect the gas to flow from the region of high pressure to



low pressure but the structure of the flow turns out to be really interesting with five distinct regions



1. Left side of the initial state

2. Expansion wave
3. x-independent state
4. x-independent state
5. Right side of the initial state

The contact line between 3 and 4 separates fluids of different entropy (but they have the same pressure and velocity) i.e. it represents an invisible line – e.g. two fluids one side with water and the other with dye – that is moving.

To obtain the solution we will solve the Euler equations with the following initial conditions:

$$\mathbf{V}(x,0) = \begin{cases} \mathbf{V_L} & x < 0 \\ \mathbf{V_R} & x \geq 0 \end{cases}$$

where

$$\mathbf{V_L} = \begin{Bmatrix} \rho_L \\ u_L \\ p_L \end{Bmatrix} = \begin{Bmatrix} 1 \\ 0 \\ 1 \end{Bmatrix}, \qquad \mathbf{V_R} = \begin{Bmatrix} \rho_R \\ u_R \\ p_R \end{Bmatrix} = \begin{Bmatrix} 0.125 \\ 0 \\ 0.1 \end{Bmatrix}$$

We will use the MacCormack scheme to obtain the finite difference equation to solve this equation. It can be applied easily when the Euler equations are in vector form

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}}{\partial x} = 0$$

Predictor:
$$\overline{\mathbf{U}_i^{n+1}} = \mathbf{U}_i^n - \frac{\Delta t}{\Delta x}(\mathbf{F}_{i+1}^n - \mathbf{F}_i^n)$$

Corrector:
$$\mathbf{U}_i^{n+1} = \frac{1}{2}\left[\mathbf{U}_i^n + \overline{\mathbf{U}_i^{n+1}} - \frac{\Delta t}{\Delta x}(\overline{\mathbf{F}_i^{n+1}} - \overline{\mathbf{F}_{i-1}^{n+1}})\right]$$

We use this finite-difference scheme along with the initial condition

$$\mathbf{U}(x,0) = \begin{cases} \mathbf{U_L} & x < 0 \\ \mathbf{U_R} & x \geq 0 \end{cases}$$

where $\mathbf{U_L}$ and $\mathbf{U_R}$ are derived from $\mathbf{V_L}$ and $\mathbf{V_R}$ respectively. The solution at $t = 0.2$ is shown in Fig. 21. The jump in the in ital condition leads to dispersion errors but the solution is otherwise of good quality. This is an extreme example to test the robustness and accuracy of numerical schemes. Smoother in ital conditions will yield very low dispersive errors.

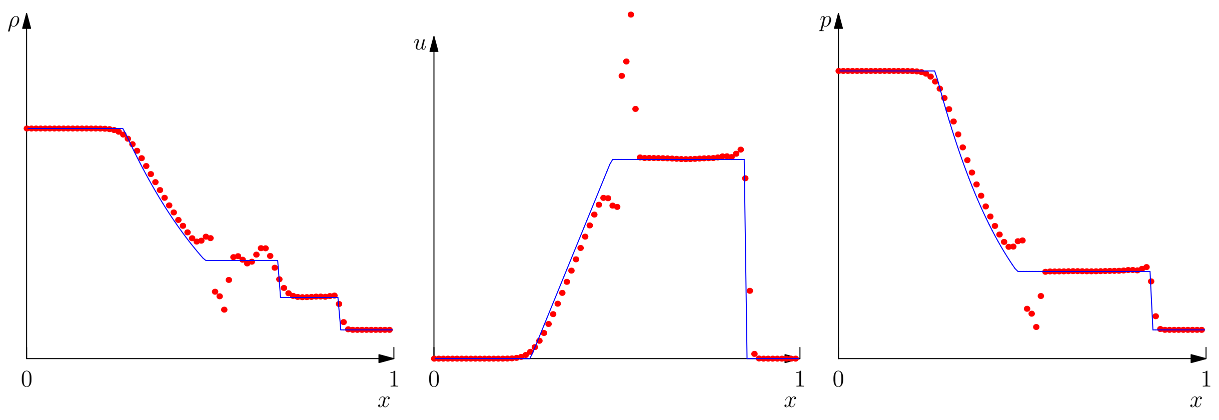


Figure 21: Solution to the shock-tube problem at $t = 0.2$. Solid line: analytical, dots: numerical solution. ($dx = 1/80$)

Algorithm 1 Algorithm to solve the shock-tube problem

```
1:  $nx \leftarrow 81$                                  ▷ Number of grid points in $x$
2:  $nt \leftarrow 42$                                  ▷ Number of grid points in $t$
3:  $t_{\max} \leftarrow 0.2$                           ▷ Value of $t$ to stop the code
4:  $x_{\min} \leftarrow 0$
5:  $x_{\max} \leftarrow 1$
6:  $\gamma \leftarrow 1.4$
7:  $dx \leftarrow (x_{\max} - x_{\min})/(nx - 1)$
                                  ▷ Define 1D array $x[]$ to store $x$ cooridnates
8:                           ▷ Define 2D arrays $U[][]$, $Un[][]$, $Ubar[][]$, $F[][]$
9:
10: for $i \leftarrow 1, nx$ do
11:     $x[i] \leftarrow x_{\min} + i * dx$
12: end for
13: for $i \leftarrow 1, nx$ do                         ▷ Define initial conditions
14:     if $x[i] \leq 0.5$ then
15:         $\rho \leftarrow 1.0$
16:         $v \leftarrow 0.0$
17:         $p \leftarrow 1.0$
18:     else
19:         $\rho \leftarrow 0.125$
20:         $v \leftarrow 0.0$
21:         $p \leftarrow 0.1$
22:     end if
23:     $U[i][1] \leftarrow \rho$
24:     $U[i][2] \leftarrow \rho * v$
25:     $U[i][3] \leftarrow p/(\gamma - 1) + 0.5 * \rho * v * v$
26: end for
27: $t \leftarrow 0$                                     ▷ Initial value for $t$
```

Algorithm 2 Shock tube, part2

```
for $n \leftarrow 1, nt$ do
    for $i \leftarrow 1, nx$ do
        $F[i][1] \leftarrow U[i][2]$
        $rv2 \leftarrow U[i][2] * U[i][2] / U[i][1]$    ▷ Calculate $\rho u^2$
        $p \leftarrow (\gamma - 1) * (U[i][3] - 0.5 * rv2)$
        $v \leftarrow U[i][2] / U[i][1]$
        $vel[i] \leftarrow v$    ▷ Save $u$ at each grid point
        $F[i][2] \leftarrow rv2 + p$
        $F[i][3] \leftarrow v * (U[i][3] + p)$
        $c[i] \leftarrow \mathrm{sqrt}(\gamma * p / U[i][1])$
    end for
    $dt \leftarrow 0.9 * dx / \max(\mathrm{abs}(vel) + c)$    ▷ compute $dt$ from a CFL number of 0.9
    $t \leftarrow t + dt$
    for $j \leftarrow 1, 3$ do    ▷ Predictor
        for $i \leftarrow 2, nx - 1$ do
            $Ubar[i][j] \leftarrow U[i][j] - dt/dx * (F[i+1][j] - F[i][j])$
        end for
        $Ubar[1][j] \leftarrow U[1][j]$    ▷ Boundary conditions
        $Ubar[nx][j] \leftarrow U[nx][j]$
    end for
    for $i \leftarrow 1, nx$ do
        $F[i][1] \leftarrow Ubar[i][2]$
        $rv2 \leftarrow Ubar[i][2] * *2 / Ubar[i][1]$
        $p \leftarrow (\gamma - 1) * (Ubar[i][3] - 0.5 * rv2)$
        $v \leftarrow Ubar[i][2] / Ubar[i][1]$
        $F[i][2] \leftarrow rv2 + p$
        $F[i][3] \leftarrow v * (Ubar[i][3] + p)$
    end for
    for $j \leftarrow 1, 3$ do    ▷ Corrector
        for $i \leftarrow 2, nx - 1$ do
            $Un[i][j] \leftarrow 0.5 * (Ubar[i][j] + U[i][j] - dt/dx * (F[i][j] - F[i-1][j]))$
        end for
    end for
    for $j \leftarrow 1, 3$ do
        for $1 \leftarrow 1, nx$ do
            $U[i][j] \leftarrow Un[i][j]$
        end for
    end for
end for
```