Upwind Derivative vs Downwind Derivative

Zhouzhou Gu\textsuperscript{†}

November 26, 2021
Upwind and downwind method

Let us start from convective equation (square root of wave Eq):

\[
\frac{\partial u(x, t)}{\partial t} + a \frac{\partial u(x, t)}{\partial x} = 0, \quad (0.1)
\]

we assume \(a > 0\), it has a solution:

\[u(x, t + \Delta) = u(x - a\Delta, t)\]

Upwind method utilizes upstream’s information.

- Upwind \((a > 0)\):

\[
\frac{u_j^{n+1} - u_j^n}{\Delta t} + a \frac{u_j^n - u_{j-1}^n}{\Delta x} = 0
\]
Upwind and downwind method Courant et al. (1952)\(^1\)

Why upwind/downwind?

\[ u(x, t) \]

\[ \begin{align*}
  u^n_j & \quad u^{n+1}_j \\
  u^n_{j-1} & \quad u^{n+1}_j \\
  x &
\end{align*} \]

\[ \begin{align*}
  u^n_j & \quad u^{n+1}_j \\
  u^n_{j-1} & \quad u^{n+1}_j \\
  x &
\end{align*} \]

\(^1\)Courant, Isaacson, and Rees proposed the CIR method
Upwind and downwind method

Why upwind/downwind?

We can also consider an extreme case that $u_{n_1}^0 = 1, u_n^0 = 0, n \neq n_1$. The wave behave morbidly.

- Intuitively, the information for the part uninfluenced cannot be utilized, they may be wrong.
- Quantitatively, it will lead to an error: $|(u_{j+1}^n - u_j^n) - (u_j^n - u_{j-1}^n)|$, for uninfluenced region, first part is generally zero while the second part is a finite one $O(1)$, or at least $O(\Delta x)$.

**Remark**: there’s another error comes from pseudo diffusion in convective Eq, it can be eliminated by choosing $\Delta t = \frac{\Delta x}{a}$.

$$
\frac{\partial u(x, t)}{\partial t} + a\frac{\partial u(x, t)}{\partial x} = \frac{1}{2} a\Delta x \left(1 - a\frac{\Delta t}{\Delta x}\right) \frac{\partial^2 u(x, t)}{\partial x^2}
$$

$u(x, t)$
In practice, we consider a PDE:

\[ \frac{\partial u}{\partial t} + f(u) \frac{\partial}{\partial x} = 0 \]

can be decomposed into:

\[ \frac{\partial u}{\partial t} + f^+ \frac{\partial}{\partial x} + f^- \frac{\partial}{\partial x} = 0 \]

We introduce the upwind to avoid information from upstream getting stucked.
Recall Ben’s HJB:

\[
\rho v(z) - v'(z)\mu(z) - \frac{1}{2}v''(z)\sigma^2(z) - \pi(z)
\]

Standard Algorithm with upwind finite difference:

- **Forward approximation** (downwind): \( v'(z) \approx \frac{v^{n+1}_j - v^n_j}{\Delta z} \)
- **Backward approximation** (upwind): \( v'(z) \approx \frac{v^n_j - v^{n-1}_j}{\Delta z} \)
- Central difference diverges in convective Eq.
The HJB in difference method

Explicit:

\[ \frac{v_j^{n+1} - v_j^n}{\Delta} + \rho v_j^n = \pi(z) + \mu(z)1_{\mu(z)>0} \partial_F v_j^n + \mu(z)1_{\mu(z)<0} \partial_B v_j^n + \frac{1}{2} \sigma_j^2 \partial^2 v_j^n \]

Or equivalently:

\[ \frac{v_j^{n+1} - v_j^n}{\Delta} + \rho v_j^n = \pi(z) + \mu(z)1_{\mu(z)>0} \frac{v_{j+1}^n - v_j^n}{\Delta z} + \mu(z)1_{\mu(z)<0} \frac{v_j^n - v_{j-1}^n}{\Delta z} + \frac{1}{2} \sigma_j^2 \frac{v_{j+1}^n - 2v_j^n + v_{j-1}^n}{\Delta z^2} \]

%Construct matrix
X = -min(mu,0)/dz + sig2/(2*dz2);
Y = -max(mu,0)/dz + min(mu,0)/dz - sig2/dz2;
Z = max(mu,0)/dz + sig2/(2*dz2);
A = spdiags(Y,0,I,I) + spdiags(X(2:I),-1,I,I) +
    spdiags([0;Z(1:I-1)],1,I,I);
Matrix Formulation

We reconstruct the matrix formulation of HJB in finite difference:

\[
\frac{1}{\Delta}(v^{n+1} - v^n) + \rho v^n = \pi_n + A^n v^n
\]

where:

\[
A^n = \begin{bmatrix}
  y_1 & z_1 & 0 & \cdots & 0 \\
  x_2 & y_2 & z_2 & \cdots & 0 \\
  0 & x_3 & y_3 & z_3 & 0 \\
  \cdots & \cdots & \cdots & \cdots & \cdots \\
  0 & \cdots & \cdots & x_N & y_N
\end{bmatrix}
\]

and

\[
x_j = -\frac{\mu(z)1_{\mu<0}}{\Delta a}, \quad y_j = -\frac{\sigma_j^2}{\Delta z^2}, \quad z_j = \frac{\mu(z)1_{\mu>0}}{\Delta a}
\]

If we have multi variables to deal with, i.e., \(z, s\), we should consider \(Z \otimes S\) and \(A_Z \otimes A_s\), it is not a triangular matrix any more.