Lecture 6: Time stepping

Summary So Far

- Want to solve Navier-Stokes equations (any PDE).
- Use of simpler equation - Burgers equation

- has same features of N-S equations + has exact solutions.

- Approximated time and space derivatives by considering non-continuous discrete solution and Taylor expansions.
- The order of the terms neglected in the expansions determine the truncation error, and hence the order of accuracy of the scheme.
- Von-Neumann stability analysis used to give amplification factor
- Stability analysis shows that schemes which violate physics of the flow, i.e. have incorrect signal propagation, are unstable.

Also shows that explicit schemes are only stable for $CFL = \frac{c\Delta t}{\Delta x} \leq 1$.

The numerical domain of dependence MUST contain the physical one.
The amplification factor, i.e. the effective damping, then depends on Δt and Δx. The higher order the scheme the smaller the amplification error. Normally $\lambda^2 = 1 - \mathrm{error}(\Delta x^p, \Delta t^q)$.
• Must have an 'upwind' scheme - where sign of the wavespeed determines which points are used in the finite-difference stencil.
However, 'non-upwind' or unstable schemes can be stabilised by adding an 'artificial viscosity' term (in the form $\alpha \frac{\partial^2 u}{\partial x^2}$) to damp the solution.

TODAY

- Further consideration of stability.
- Demonstration of stability.
- Consideration of time-stepping scheme.
- Show non-conservative form gives incorrect signal speed.

Difference Between Schemes

The FTCS has higher order accuracy than the upwind method, but is unstable. Let's look at this in more detail, consider the upwind method, and compare to FTCS method.
FTCS:

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} + c\frac{u_{i+1}^n - u_{i-1}^n}{2\Delta x} = 0$$

Upwind ($c > 0$)

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

Compare the spatial difference approximation, i.e. consider the difference between the spatial terms for each (UPWIND - FTCS)

$$\begin{aligned}\frac{u_i^n - u_{i-1}^n}{\Delta x} - \frac{u_{i+1}^n - u_{i-1}^n}{2\Delta x} &= \frac{1}{2\Delta x}(2u_i^n - 2u_{i-1}^n - u_{i+1}^n + u_{i-1}^n) \\ &= \frac{1}{2\Delta x}(-u_{i+1}^n + 2u_i^n - u_{i-1}^n)\end{aligned}$$

This looks like a finite difference formula - so what gradient does it represent?

What is the RHS term (difference between the two schemes) ?

$$u_{i+1}^n = u_i^n + \Delta x \left.\frac{\partial u}{\partial x}\right|_i^n + \frac{1}{2}(\Delta x)^2 \left.\frac{\partial^2 u}{\partial x^2}\right|_i^n + \frac{1}{6}(\Delta x)^3 \left.\frac{\partial^3 u}{\partial x^3}\right|_i^n + O(\Delta x^4)$$

$$u_{i-1}^n = u_i^n - \Delta x \left.\frac{\partial u}{\partial x}\right|_i^n + \frac{1}{2}(\Delta x)^2 \left.\frac{\partial^2 u}{\partial x^2}\right|_i^n - \frac{1}{6}(\Delta x)^3 \left.\frac{\partial^3 u}{\partial x^3}\right|_i^n + O(\Delta x^4)$$

$$u_{i+1}^n + u_{i-1}^n = 2u_i^n + 0 + (\Delta x)^2 \left.\frac{\partial^2 u}{\partial x^2}\right|_i^n + 0 + O(\Delta x^4)$$

or

$$\left.\frac{\partial^2 u}{\partial x^2}\right|_i^n \simeq \frac{u_{i-1}^n - 2u_i^n + u_{i+1}^n}{(\Delta x)^2} + O(\Delta x^2)$$

so

$$\frac{1}{2\Delta x}(-u_{i+1}^n + 2u_i^n - u_{i-1}^n) = -\frac{\Delta x}{2} \left.\frac{\partial^2 u}{\partial x^2}\right|_i^n - O(\Delta x^2)$$

Hence, clearly the upwind scheme is the FTCS scheme plus

$$\frac{1}{2}\Delta x \frac{\partial^2 u}{\partial x^2}$$

This can be considered an artificial viscosity term.

Simple Example of Stability of Time-Marching

Consider solving the linear wave equation,

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

For a positive wavespeed, we can use the first-order upwind scheme,

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} + \frac{c}{\Delta x}(u_i^n - u_{i-1}^n) + O(\Delta t, \Delta x) = 0$$

which gives the resulting explicit updating scheme,

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

Stability has shown that this is stable for

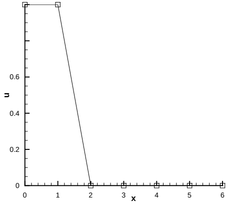
$$\frac{c\Delta t}{\Delta x} \leq 1.$$

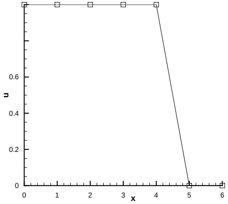
As an example, consider a simple example, $0\text{m} \le x \le 6\text{m}$ with 7 discrete points, and a wavespeed of 1.0 m/s, i.e.

$$\Delta x = \frac{6.0}{7-1} = 1.0\text{m}, \qquad\qquad c = 1.0\text{m/s},$$

with the initial conditions given below on the left.

If we integrate this forward in time by 3.0s we should have the solution on the right, i.e. the solution has propagated right by $c\Delta t$ ($1.0\text{m/s} \times 3.0\text{s} = 3.0\text{m}$).





Now consider integrating forward with three different values of Δt. The CFL condition tells us that the maximum stable value of Δt is 1.0s. Hence, integrate forward using four timesteps of $\Delta t = 0.75$s, three timesteps of $\Delta t = 1.0$s, and two timesteps of $\Delta t = 1.5$s.

For simplicity u_1 will be set constant.

1) Four timesteps of $\Delta t = 0.75s$:

$$
\begin{aligned}
u_1^n &= 1.0\\
u_2^n &= 1.0\\
u_3^n &= 0.0\\
u_4^n &= 0.0\\
u_5^n &= 0.0\\
u_6^n &= 0.0\\
u_7^n &= 0.0
\end{aligned}
$$

$$
\begin{aligned}
u_1^{n+1} &= 1.0\\
u_2^{n+1} &= u_2^n - \tfrac{1.0\times 0.75}{1.0}(u_2^n - u_1^n) = 1.0\\
u_3^{n+1} &= u_3^n - \tfrac{1.0\times 0.75}{1.0}(u_3^n - u_2^n) = 0.75\\
u_4^{n+1} &= u_4^n - \tfrac{1.0\times 0.75}{1.0}(u_4^n - u_3^n) = 0.0\\
u_5^{n+1} &= u_5^n - \tfrac{1.0\times 0.75}{1.0}(u_5^n - u_4^n) = 0.0\\
u_6^{n+1} &= u_6^n - \tfrac{1.0\times 0.75}{1.0}(u_6^n - u_5^n) = 0.0\\
u_7^{n+1} &= u_7^n - \tfrac{1.0\times 0.75}{1.0}(u_7^n - u_6^n) = 0.0
\end{aligned}
$$

$$
\begin{aligned}
u_1^{n+2} &= 1.0\\
u_2^{n+2} &= u_2^{n+1} - \tfrac{1.0\times 0.75}{1.0}(u_2^{n+1} - u_1^{n+1}) = 1.0\\
u_3^{n+2} &= u_3^{n+1} - \tfrac{1.0\times 0.75}{1.0}(u_3^{n+1} - u_2^{n+1}) = 0.9375\\
u_4^{n+2} &= u_4^{n+1} - \tfrac{1.0\times 0.75}{1.0}(u_4^{n+1} - u_3^{n+1}) = 0.5625\\
u_5^{n+2} &= u_5^{n+1} - \tfrac{1.0\times 0.75}{1.0}(u_5^{n+1} - u_4^{n+1}) = 0.0\\
u_6^{n+2} &= u_6^{n+1} - \tfrac{1.0\times 0.75}{1.0}(u_6^{n+1} - u_5^{n+1}) = 0.0\\
u_7^{n+2} &= u_7^{n+1} - \tfrac{1.0\times 0.75}{1.0}(u_7^{n+1} - u_6^{n+1}) = 0.0
\end{aligned}
$$

$$
\begin{aligned}
u_1^{n+3} &= 1.0\\
u_2^{n+3} &= u_2^{n+2} - \tfrac{1.0\times 0.75}{1.0}(u_2^{n+2} - u_1^{n+2}) = 1.0\\
u_3^{n+3} &= u_3^{n+2} - \tfrac{1.0\times 0.75}{1.0}(u_3^{n+2} - u_2^{n+2}) = 0.9844\\
u_4^{n+3} &= u_4^{n+2} - \tfrac{1.0\times 0.75}{1.0}(u_4^{n+2} - u_3^{n+2}) = 0.8438\\
u_5^{n+3} &= u_5^{n+2} - \tfrac{1.0\times 0.75}{1.0}(u_5^{n+2} - u_4^{n+2}) = 0.4219\\
u_6^{n+3} &= u_6^{n+2} - \tfrac{1.0\times 0.75}{1.0}(u_6^{n+2} - u_5^{n+2}) = 0.0\\
u_7^{n+3} &= u_7^{n+2} - \tfrac{1.0\times 0.75}{1.0}(u_7^{n+2} - u_6^{n+2}) = 0.0
\end{aligned}
$$

$$
\begin{aligned}
u_1^{n+4} &= 1.0\\
u_2^{n+4} &= u_2^{n+3} - \tfrac{1.0\times 0.75}{1.0}(u_2^{n+3} - u_1^{n+3}) = 1.0\\
u_3^{n+4} &= u_3^{n+3} - \tfrac{1.0\times 0.75}{1.0}(u_3^{n+3} - u_2^{n+3}) = 0.9961\\
u_4^{n+4} &= u_4^{n+3} - \tfrac{1.0\times 0.75}{1.0}(u_4^{n+3} - u_3^{n+3}) = 0.9493\\
u_5^{n+4} &= u_5^{n+3} - \tfrac{1.0\times 0.75}{1.0}(u_5^{n+3} - u_4^{n+3}) = 0.7383\\
u_6^{n+4} &= u_6^{n+3} - \tfrac{1.0\times 0.75}{1.0}(u_6^{n+3} - u_5^{n+3}) = 0.3164\\
u_7^{n+4} &= u_7^{n+3} - \tfrac{1.0\times 0.75}{1.0}(u_7^{n+3} - u_6^{n+3}) = 0.0
\end{aligned}
$$

2) Three timesteps of $\Delta t = 1.0s$:

$$u_1^n = 1.0$$
$$u_2^n = 1.0$$
$$u_3^n = 0.0$$
$$u_4^n = 0.0$$
$$u_5^n = 0.0$$
$$u_6^n = 0.0$$
$$u_7^n = 0.0$$

$$u_1^{n+1} = 1.0$$
$$u_2^{n+1} = u_2^n - \frac{1.0\times1.0}{1.0}(u_2^n - u_1^n) = 1.0$$
$$u_3^{n+1} = u_3^n - \frac{1.0\times1.0}{1.0}(u_3^n - u_2^n) = 1.0$$
$$u_4^{n+1} = u_4^n - \frac{1.0\times1.0}{1.0}(u_4^n - u_3^n) = 0.0$$
$$u_5^{n+1} = u_5^n - \frac{1.0\times1.0}{1.0}(u_5^n - u_4^n) = 0.0$$
$$u_6^{n+1} = u_6^n - \frac{1.0\times1.0}{1.0}(u_6^n - u_5^n) = 0.0$$
$$u_7^{n+1} = u_7^n - \frac{1.0\times1.0}{1.0}(u_7^n - u_6^n) = 0.0$$

$$u_1^{n+2} = 1.0$$
$$u_2^{n+2} = u_2^{n+1} - \frac{1.0\times1.0}{1.0}(u_2^{n+1} - u_1^{n+1}) = 1.0$$
$$u_3^{n+2} = u_3^{n+1} - \frac{1.0\times1.0}{1.0}(u_3^{n+1} - u_2^{n+1}) = 1.0$$
$$u_4^{n+2} = u_4^{n+1} - \frac{1.0\times1.0}{1.0}(u_4^{n+1} - u_3^{n+1}) = 1.0$$
$$u_5^{n+2} = u_5^{n+1} - \frac{1.0\times1.0}{1.0}(u_5^{n+1} - u_4^{n+1}) = 0.0$$
$$u_6^{n+2} = u_6^{n+1} - \frac{1.0\times1.0}{1.0}(u_6^{n+1} - u_5^{n+1}) = 0.0$$
$$u_7^{n+2} = u_7^{n+1} - \frac{1.0\times1.0}{1.0}(u_7^{n+1} - u_6^{n+1}) = 0.0$$

$$u_1^{n+3} = 1.0$$
$$u_2^{n+3} = u_2^{n+2} - \frac{1.0\times1.0}{1.0}(u_2^{n+2} - u_1^{n+2}) = 1.0$$
$$u_3^{n+3} = u_3^{n+2} - \frac{1.0\times1.0}{1.0}(u_3^{n+2} - u_2^{n+2}) = 1.0$$
$$u_4^{n+3} = u_4^{n+2} - \frac{1.0\times1.0}{1.0}(u_4^{n+2} - u_3^{n+2}) = 1.0$$
$$u_5^{n+3} = u_5^{n+2} - \frac{1.0\times1.0}{1.0}(u_5^{n+2} - u_4^{n+2}) = 1.0$$
$$u_6^{n+3} = u_6^{n+2} - \frac{1.0\times1.0}{1.0}(u_6^{n+2} - u_5^{n+2}) = 0.0$$
$$u_7^{n+3} = u_7^{n+2} - \frac{1.0\times1.0}{1.0}(u_7^{n+2} - u_6^{n+2}) = 0.0$$

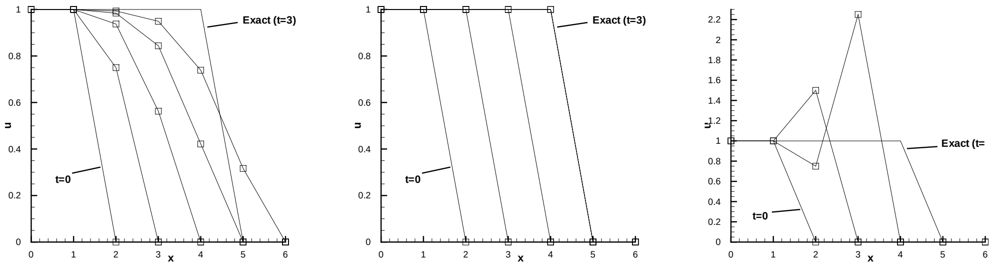
3) Two timesteps of $\Delta t = 1.5s$:

$$u_1^n = 1.0$$
$$u_2^n = 1.0$$
$$u_3^n = 0.0$$
$$u_4^n = 0.0$$
$$u_5^n = 0.0$$
$$u_6^n = 0.0$$
$$u_7^n = 0.0$$

$$u_1^{n+1} = 1.0$$
$$u_2^{n+1} = u_2^n - \frac{1.0\times 1.5}{1.0}(u_2^n - u_1^n) = 1.0$$
$$u_3^{n+1} = u_3^n - \frac{1.0\times 1.5}{1.0}(u_3^n - u_2^n) = 1.5$$
$$u_4^{n+1} = u_4^n - \frac{1.0\times 1.5}{1.0}(u_4^n - u_3^n) = 0.0$$
$$u_5^{n+1} = u_5^n - \frac{1.0\times 1.5}{1.0}(u_5^n - u_4^n) = 0.0$$
$$u_6^{n+1} = u_6^n - \frac{1.0\times 1.5}{1.0}(u_6^n - u_5^n) = 0.0$$
$$u_7^{n+1} = u_7^n - \frac{1.0\times 1.5}{1.0}(u_7^n - u_6^n) = 0.0$$

$$u_1^{n+2} = 1.0$$
$$u_2^{n+2} = u_2^{n+1} - \frac{1.0\times 1.5}{1.0}(u_2^{n+1} - u_1^{n+1}) = 1.0$$
$$u_3^{n+2} = u_3^{n+1} - \frac{1.0\times 1.5}{1.0}(u_3^{n+1} - u_2^{n+1}) = 0.75$$
$$u_4^{n+2} = u_4^{n+1} - \frac{1.0\times 1.5}{1.0}(u_4^{n+1} - u_3^{n+1}) = 2.25$$
$$u_5^{n+2} = u_5^{n+1} - \frac{1.0\times 1.5}{1.0}(u_5^{n+1} - u_4^{n+1}) = 0.0$$
$$u_6^{n+2} = u_6^{n+1} - \frac{1.0\times 1.5}{1.0}(u_6^{n+1} - u_5^{n+1}) = 0.0$$
$$u_7^{n+2} = u_7^{n+1} - \frac{1.0\times 1.5}{1.0}(u_7^{n+1} - u_6^{n+1}) = 0.0$$

The resulting solutions are shown below.



Integration over 3 seconds, using four, three, and two timesteps.

Hence, the CFL $= 0.75$ solution shows dissipation, as $\lambda < 1$, the CFL $= 1.0$ solution models the solution exactly, as $\lambda = 1$ always, and the CFL $= 1.5$ solution is unstable. Over a 1.5 second timestep the solution should propagate 1.5 cells, but it is not possible for the numerical solution to propagate more than one cell over each timestep.

Further Aspects of Numerical Algorithms

Before considering a more realistic example, consider the evaluation of the time step, Δt. Clearly we must satisfy the CFL condition, which gives a limit on the time-step. For a time-accurate solution, i.e. we wish to examine the solution evolution in time, the solution at every mesh point must be updated using the same time-step. If the wave speed and mesh spacing is constant we simply compute the GLOBAL time-step using

$$\Delta t \leq \frac{\Delta x}{|c|}$$

But if we have either non-constant mesh spacing ($\Delta x(x)$) and/or non-constant wave speed ($c(x)$) we must compute the time-step to satisfy the CFL limit at every mesh point. Then the only way to use the same time-step at every mesh point and still satisfy the CFL condition at every point is to use the minimum Δt computed over the whole domain.

Time-Accurate Numerical Algorithm

The structure of a numerical time-stepping scheme is as follows

DO 1 TO NT

- APPLY B.C.s
- DO I=1,NI
  - COMPUTE c_i
  - COMPUTE $\Delta t_i \quad (= CFL\frac{\Delta x_i}{c_i})$
- ENDDO
- $\Delta t = \min(\Delta t_i)$
- DO I=1,NI
  - $u_i^{n+1} = u_i^n - \frac{c_i \Delta t}{\Delta x_i}(u_i^n - u_{i-1}^n)$
- ENDDO
- DO I=1,NI
  - $u_i^n = u_i^{n+1}$
- ENDDO

ENDDO

Steady Numerical Algorithm

However, if we simply wish to get to steady state solution, i.e. $t \to \infty$, without worrying about the intermediate solutions we can use a different Δt at each point. There is no need to synchronise the time level. This means we can use the largest time-step allowed at each point. The algorithm is slightly simpler in this case. This approach is called local time-stepping.

DO 1 TO NT

  APPLY B.C.s

  DO I=1,NI

    COMPUTE c_i

    COMPUTE $\Delta t_i \quad (= CFL\frac{\Delta x_i}{c_i})$

    $u_i^{n+1} = u_i^n - \frac{c_i \Delta t_i}{\Delta x_i}(u_i^n - u_{i-1}^n)$

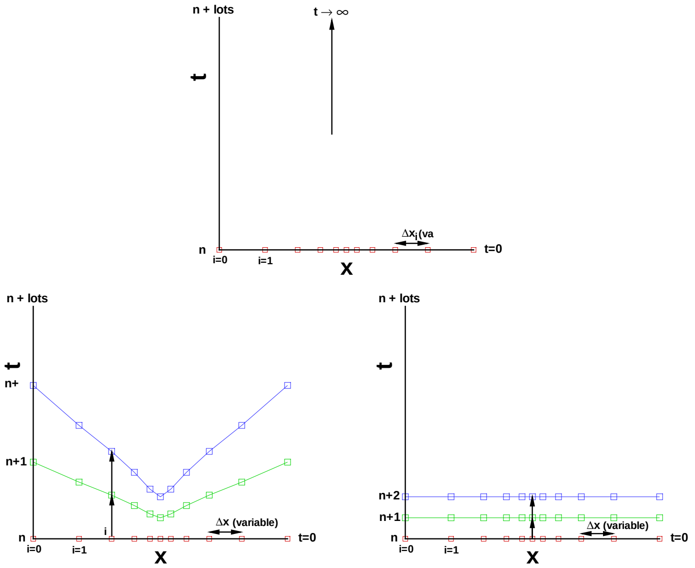
  ENDDO

  DO I=1,NI

    $u_i^n = u_i^{n+1}$

  ENDDO

ENDDO



Local time steps. Global time step.

Numerical Example

Consider square pulse initial conditions for non-linear wave equation (wave speed not constant).

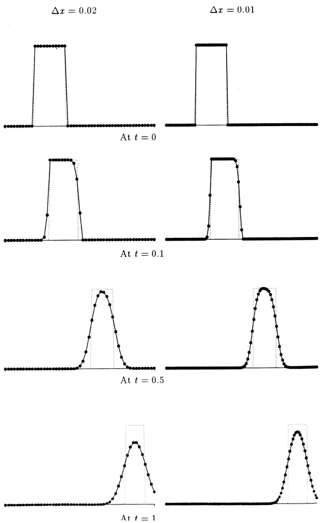
$$u_t + c(x,t)u_x = 0$$

$$c(x,t) = \frac{1+x^2}{1+2xt+2x^2+x^4}$$

$$u(x,0) = \begin{cases} 1.0 & 0.2 \leq x \leq 0.4 \\ 0.0 & \text{otherwise} \end{cases}$$

For all points $\Delta t = \Delta x$, and so $\nu = \frac{c\Delta t}{\Delta x}$ is variable.

Computed using first-oder upwind scheme.



Conservative v Non-Conservative Form

Consider the non-linear Burgers' equation:

$$\frac{\partial u}{\partial t} + \frac{\partial \frac{1}{2}u^2}{\partial x} = 0 = \frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x}$$

These two forms are mathematically the same, but are not the same numerically. An upwind scheme to discretise the non-conservative form could be written as

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} + u_i^n \frac{(u_i^n - u_{i-1}^n)}{\Delta x} = 0 \quad \text{if} \quad u_i^n \geq 0$$

$$u_i^{n+1} = u_i^n - u_i^n \frac{\Delta t(u_i^n - u_{i-1}^n)}{\Delta x} \quad \text{if} \quad u_i^n \geq 0$$

This is the finite-difference analogue of the non-conservative equation.

Now consider the FDA on the conservative equation. Again assume $u_i^n \geq 0$, then we have

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} + \frac{\frac{1}{2}(u_i^n)^2 - \frac{1}{2}(u_{i-1}^n)^2}{\Delta x} = 0$$

or

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} + \frac{\frac{1}{2}(u_i^n + u_{i-1}^n)(u_i^n - u_{i-1}^n)}{\Delta x} = 0$$

$$u_i^{n+1} = u_i^n - \frac{1}{2}(u_i^n + u_{i-1}^n)\frac{\Delta t(u_i^n - u_{i-1}^n)}{\Delta x} = 0$$

Hence, the two schemes have resulted in different wave speeds. The non-conservative scheme gives the correct form of the wavespeed (u), but does not tell us how to evaluate it. This can lead to serious problems if used. Must use the conservative scheme.

Summary

• For explicit schemes, we must satisfy the CFL condition. This results because the numerical domain of dependence MUST contain the physical one. A simple demonstration, using a first-order spatial scheme, showed what happens if we try to send a signal further than numerical spacing, i.e. if $c\Delta t > \Delta x$.

• Shown for $CFL < 1$ first-order scheme is very difussive.

• Considered stability; demonstrated that upwind scheme is the FTCS scheme plus an effective dissipation term.

• Considered time stepping. For a steady solution, don't care about solution progression from $t = 0$ to $t \to \infty$. Can use local time stepping, wherein largest allowable time step in every cell used.

• If we use the non-conservative form of an equation, the effective wavespeed is not clear.

NEXT LECTURE: Look at higher order methods to represent spatial gradients.