Phase Planes
&
Numerical Integration

- Lotke Volterra model with limited capacity
- Phase plane, nullclines, fixed points
- Numerical integration
- Linearization of Nonlinear Dynamical System.

Lotka-Volterra (with limited capacity) a predator-prey model

- We will use this model as an example of non-linear system analysis
- The unlimited capacity version was suggested by Alfred J. Lotka and Vito Volterra in the 1920s as a predator-prey (or parasite-host) model.
- $x(t)$ = size of the **prey** population.
- $w(t)$ = size of the **predator** population.

  (as percentage from full capacity) at time t.
- Prey population is limited by the size of the environment and predated upon.
- Predator population is balanced by the percentage and a natural death rate.

Lotka-Volterra Equations

- System equations:

$$\begin{aligned} f_1 = \dot{x} &= (1 - x)x - bxw \\ f_2 = \dot{w} &= cxw - dw \end{aligned}$$

  $b,c,d > 0$

- $(1 - x)x$ means prey population grows in the absence of predators up to 1. Fastest growth at 1/2.

- -*bxw* means prey population diminishes do to predation. *b* has to do with the chance of a prey dying in an encounter.

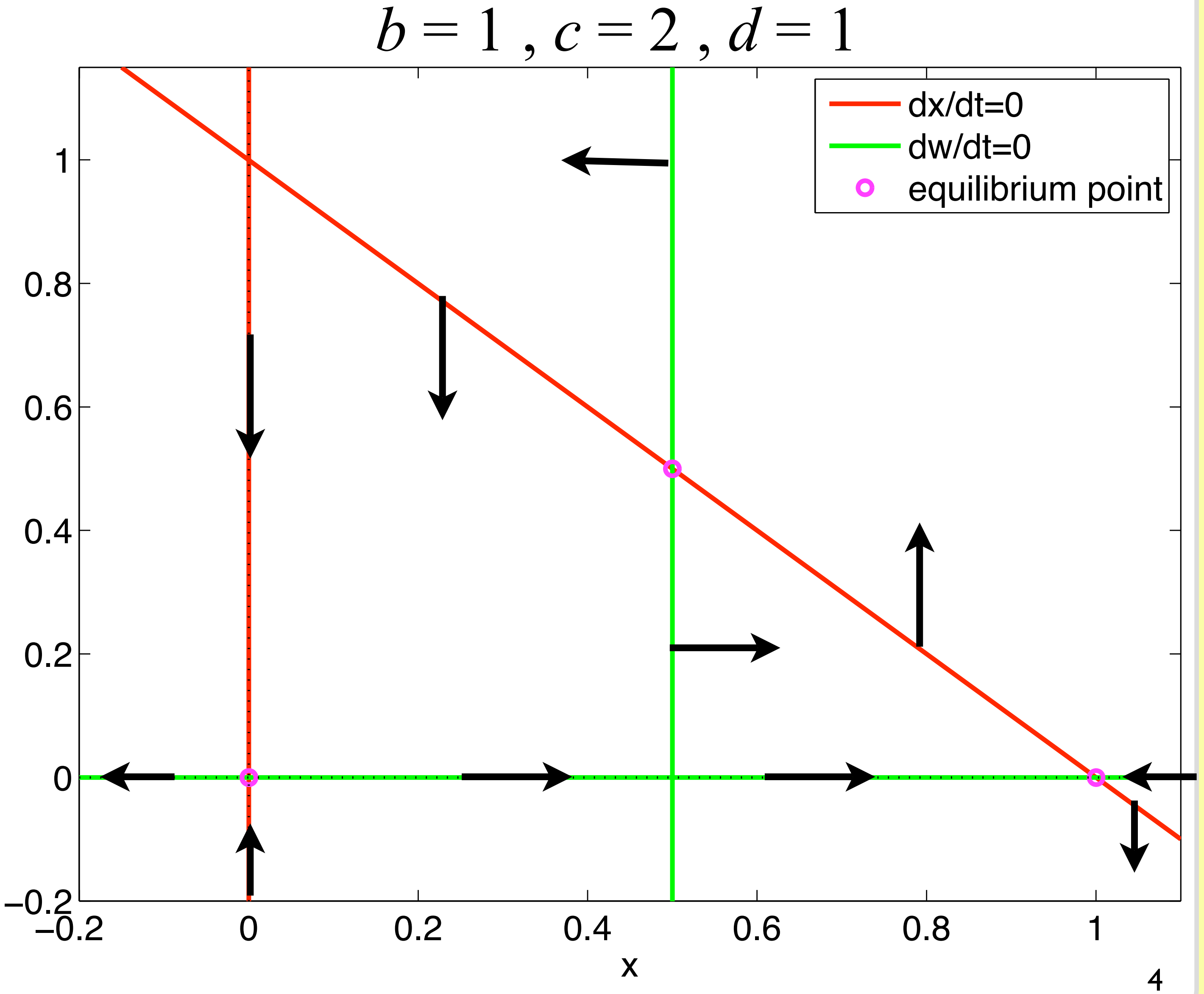
- *cxw* means the predator population grows do to predation. *c* has to do with how much an encounter is productive for the predator.

- -*dw* means the predator population has a natural death rate. *d* is the percentage of deaths per time unit.

Phase Plane Analysis

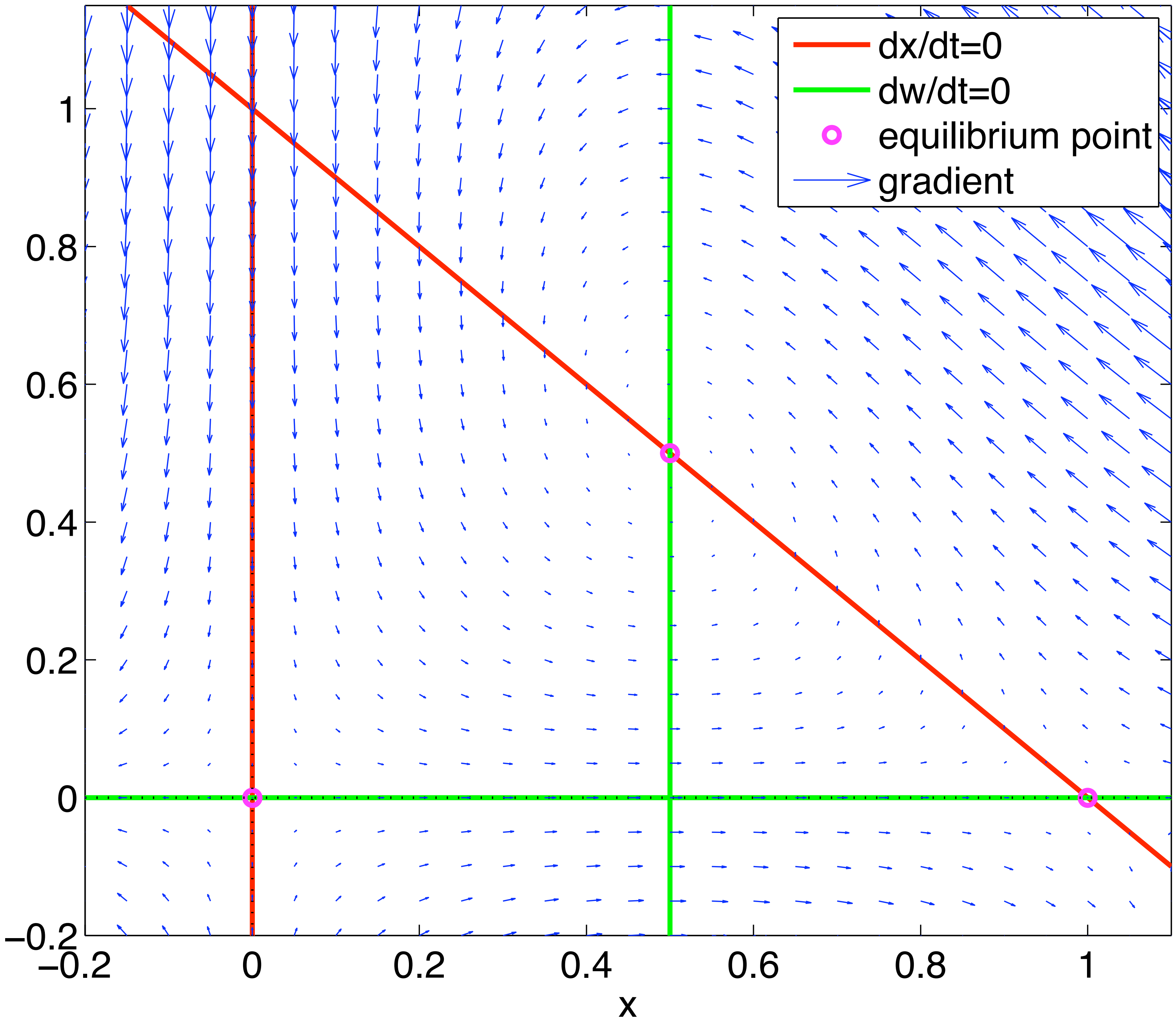
- A 2D method ☹
- *nullclines*: places where one of the derivatives is zero
  - for $f_1 = 0$: $x = 0$ or x = 1-bw
  - for $f_2 = 0$: $w = 0$ or x = d/c
- *Fixed* (aka *equilibrium*) *points*: places where all derivatives are 0.
  - (x , w) = (0 , 0) , (1 , 0) or (d/c , 1/b-d/(bc))
  - Clearly (0,0) and (1,0) are saddle points.
  - Less clearly, The third is stable

$$
\begin{aligned}
f_1 = \dot{x} &= (1-x)x - bxw \\
f_2 = \dot{w} &= cxw - dw
\end{aligned}
$$



Computerized Methods

- Gradient field lets us visualize possible paths
- Numerical integration lets us estimate specific paths either from given starting points or end points (by reversing the gradients)



Numerical Integration

- Given **f**(**x**) = d**x**/d**t** we would like to find **x**(*t*) starting from a given **x**(0).
- i.e. we would like to solve

$$\mathbf{x}(t) = \mathbf{x}_0 + \int_0^t \mathbf{f}\left(\mathbf{x}(\tau)\right) d\tau$$

- Problem: while above formula is correct we can not solve it without knowing its solution in advance (we need to know the path in order to know what the gradient along the path is..).
- Solution: advance in small steps using the gradients at and around the previous point in order to estimate the next point on the path:

$$\mathbf{x}(t+h) = \mathbf{x}(t) + \int_t^{t+h} \mathbf{f}\left(\mathbf{x}(\tau)\right) d\tau$$

- If *h* is small we can approximate $\mathbf{f}\left(\mathbf{x}(\tau)\right)$ in $\int_t^{t+h} \mathbf{f}\left(\mathbf{x}(\tau)\right) d\tau$ as constant.
- Tradeoff between smaller *h* and complexity of each step..

Euler Method

(Or 1st order Runge Kutta)

- This method uses a 1st degree Taylor expansion (i.e. a linearizarion) to propagate the system state:

$$\mathbf{x}(t+h) \approx \mathbf{x}(t) + \dot{\mathbf{x}}(t)h = \mathbf{x}(t) + \mathbf{f}(\mathbf{x}(t))h$$

- We therefore assume

$$\mathbf{f}\left(\mathbf{x}(\tau)\right) = \mathbf{f}\left(\mathbf{x}(t)\right) \; \tau \in [t, t+h]$$

- As **f(x(**t+h**))** becomes very different than **f(x(**t**))** this will result in a larger error.
- O(h^2) at every step. One calculation of **f** at every step.
- Note that errors add up..

Improved Euler (Or 2nd order Runge Kutta)

- In Euler we took $\int_t^{t+h} \mathbf{f}\left(\mathbf{x}(\tau)\right) d\tau$ = $h\mathbf{f}(\mathbf{x}(t))$. i.e. the gradient at the beginning of the step. We would probably do better to take the gradient halfway through the step:

$$\int_t^{t+h} \mathbf{f}\left(\mathbf{x}(\tau)\right) d\tau \approx h\mathbf{f}\left(\mathbf{x}\left(t+\frac{h}{2}\right)\right)$$

- Problem: we don't know $\mathbf{x}\left(t+\frac{h}{2}\right)$
- Solution: approximate is using Euler (see a recursion here?)

$$h\mathbf{f}\left(\mathbf{x}\left(t+\frac{h}{2}\right)\right) \approx h\mathbf{f}\left(\mathbf{x}(t)+\frac{h}{2}\mathbf{f}(\mathbf{x}(t))\right)$$

- $\mathbf{x}(t+h) = \mathbf{x}(t) + h\mathbf{f}(\mathbf{x} + \Delta\mathbf{x}_1)$ $\Delta\mathbf{x}_1 = {}^{h}/_{2}\mathbf{f}(\mathbf{x}(t))$
- Can prove that error is $O(h^3)$
- Note: we calculate **f** at two points to make one step..

4th Order Runge Kutta

- Find $\mathbf{\Delta x}_1$ through $\mathbf{\Delta x}_4$ incrementally (sampling **f** 4 times).

$$\begin{aligned}
\Delta\mathbf{x}_1 &= h\mathbf{f}(\mathbf{x}(t)) \\
\Delta\mathbf{x}_2 &= h\mathbf{f}\left(\mathbf{x}(t) + \frac{1}{2}\Delta\mathbf{x}_1\right) \\
\Delta\mathbf{x}_3 &= h\mathbf{f}\left(\mathbf{x}(t) + \frac{1}{2}\Delta\mathbf{x}_2\right) \\
\Delta\mathbf{x}_4 &= h\mathbf{f}\left(\mathbf{x}(t) + \Delta\mathbf{x}_3\right)
\end{aligned}$$

- Estimate the next state as:

$$\mathbf{x}(t+h) \approx \mathbf{x}(t) + \frac{1}{6}(\Delta\mathbf{x}_1 + 2\Delta\mathbf{x}_2 + 2\Delta\mathbf{x}_3 + \Delta\mathbf{x}_4)$$

- Yields error O(h^5). Considered a good balance between step complexity and step accuracy.

- Note: in all three methods, *h* can be chosen differently at each step by considering the quickness of changes in previously calculated gradients.

Linearization (esp. near equilibrium point)

- We will see in future lectures that linear systems of the form

$$\mathbf{f}(\mathbf{x}, \mathbf{u}) = \dot{\mathbf{x}}(\mathbf{x}, \mathbf{u}) = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}$$

  are easy to analyze (and control).

- Many nonlinear systems are approximately linear when observed at very short time intervals (or in very small areas of state space).

- Therefore, *linearization* (approximating a nonlinear system with a linear one) is very useful, especially near equilibrium points.

- We'll leave out the control signal, *u* for a while (easy to infer on your own)

- In scalar case, we wish to find an approximation of $f(x_0 + x)$ that has the form:

$$\hat{f}(x_0 + x) \approx f(x_0) + ax$$

- We do so by using a Taylor expansion of order 1.

$$\hat{f}(x_0 + x) = f(x_0) + \frac{df}{dx}x + O(x^2)$$

i.e. $a = \frac{df}{dx}$

- In the multidimensional case we wish to find an approximation of **f**(**x**) of the form

$$\hat{\mathbf{f}}(\mathbf{x}_0 + \mathbf{x}) \approx \mathbf{f}(\mathbf{x}_0) + \mathbf{A}\mathbf{x}$$

- Again, we can do so using the multidimensional 1st order Taylor expansion

$$\mathbf{f}(\mathbf{x}_0 + \mathbf{x}) = \mathbf{f}(\mathbf{x}_0) + \mathbf{J}\mathbf{x} + O(||\mathbf{x}||^2)$$

- What is **J**?

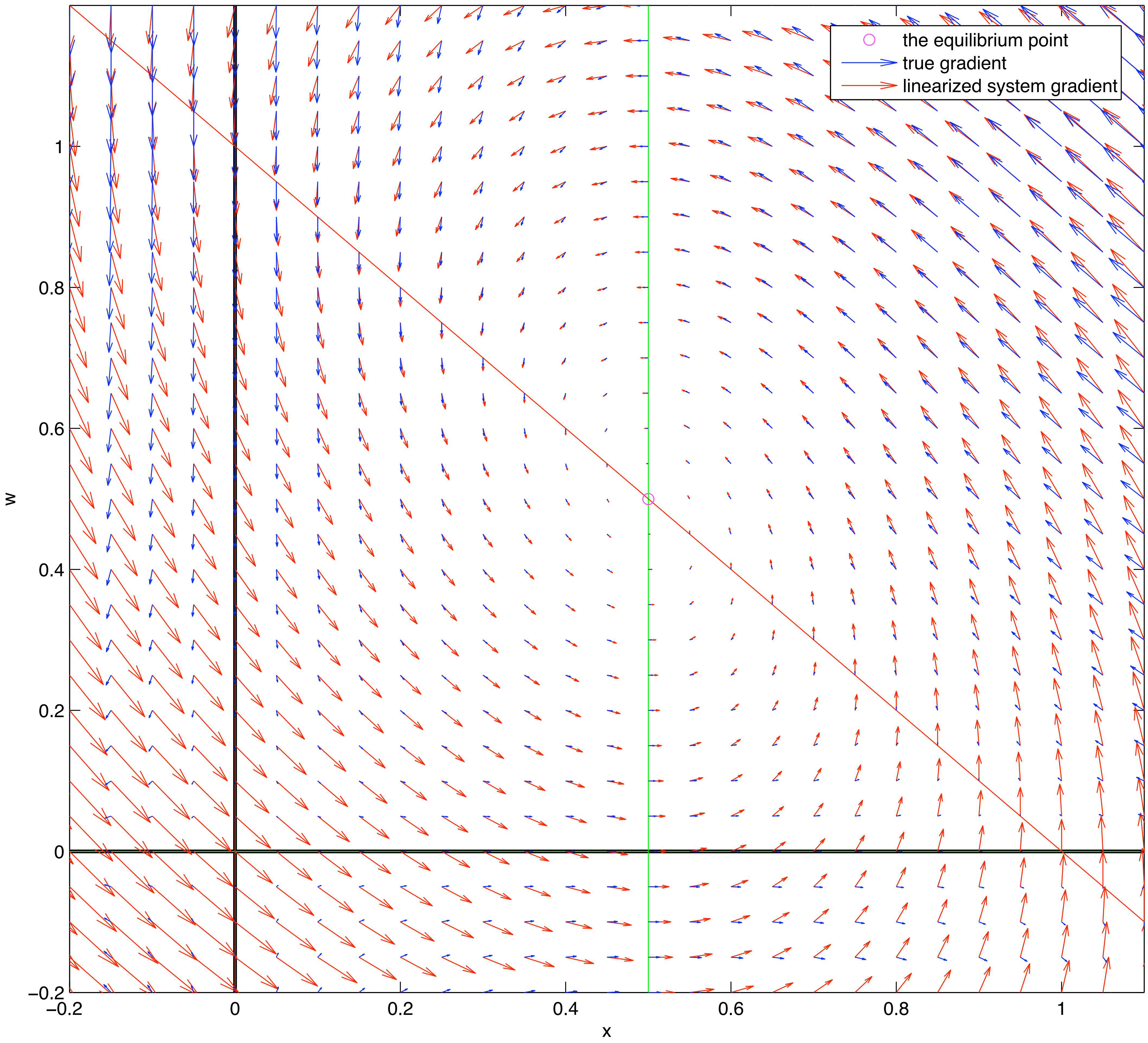
$$\mathbf{f}(\mathbf{x}_0 + \mathbf{x}) = \mathbf{f}(\mathbf{x}_0) + \mathbf{J}\mathbf{x} + O(||\mathbf{x}||^2)$$

- **J** is the *Jacobian* of **f**

$$\mathbf{J} = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_n} & \cdots & \frac{\partial f_n}{\partial x_n} \end{pmatrix}$$

  so **A=J**

- When done around equilibrium point first term vanishes



○ the equilibrium point
true gradient
linearized system gradient
w
x
−0.2
0
0.2
0.4
0.6
0.8
1