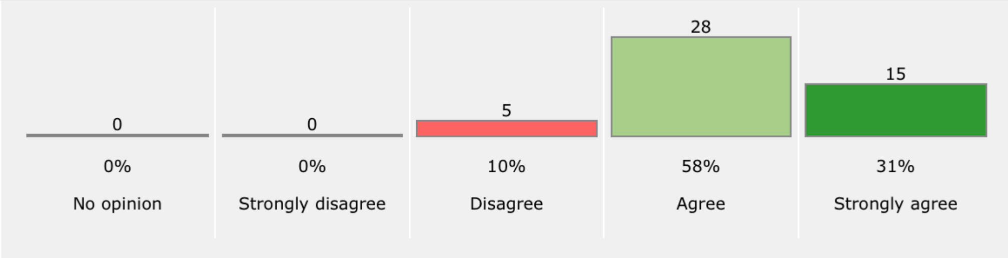
Deadline for Graded Exercise 1 is midnight Wednesday, 16th October (tomorrow!)

Note. If submitting a jupyter notebook sure you run it from the beginning to ensure that all cells work. Sometimes variables remain defined in partial reruns, which can break the notebook.

You can submit 2 (or more) files: *.ipynb and pdf, but use your name as prefix, e.g., julian_shillcock.pdf.

You can also submit a zip file containing your files, e.g., julian_shillcock.zip.

Indicative feedback



Remarks
[10 remarque(s)]

- Les énoncés des séries sont un peu confus et le cours est difficile à suivre car très rapide.
- All good
- Difficult to follow the lectures. Exercises are difficult to do without having the solutions to understand.
- I kind of struggle to associate lectures and exercises
- I like the professor but I find it hard to really understand the material when I come to the lecture. I usually feel like the course is one big exemple and I'm usually pretty much lost at the end of the two hours. However, the exercices help unde better.
- I really like the course and especially the professor! But sometimes the course goes a little too fast especially in the second hour.
- It is a bit fast but if you read the course notes before it works!
- Le cours n'est souvent pas fini pour pouvoir complètement comprendre les séries, ce qui est parfois embêtant. Le cours manque un peu d'intuition pour visualiser de quoi on parle, de plus, le débit est légèrement trop rapide: soit on écrit, soit 2.
- Very Nice class. I still struggle in the exercise series but I think the professor does its best to explain simply and the TAs are great! It would be great to have a microphone so that we can hear better and never lose concentration. ;)
- Very well explained course, the detailed explanations given to questions on the Ed forum are very helpful!

tl;dr: Lectures are too fast.

Possible solution: I’ll leave out algebra in the derivations, but give the results. The full derivation is still in the blackboard notes I put up each week.

Lecture 5 Recapitulation

The 2D non-linear system

$$dx/dt = f(x, y)$$
$$dy/dt = g(x, y)$$

can have FPs anywhere, and their number/type depend on f and g.

The recipe for solving it is:

- Find the nullclines ($f(x, y) = 0$ or $g(x, y) = 0$ independently), note that in general they are curves in 2D, not straight lines
- Find the FPs where $f(x^*, y^*) = g(x^*, y^*) = 0$ simultaneously
- Linearise f and g at each FP to get the Jacobian, then apply the linear 2D recipe in the slides of Lecture 4
- Connect the trajectories between FPs in the whole phase portrait using common sense (trajectories must be continuous, cannot cross)
- Unlike the linear system, the eigenvectors for each FP only give the trajectories close to the FP, not in the whole plane

Lecture 6 Introduction

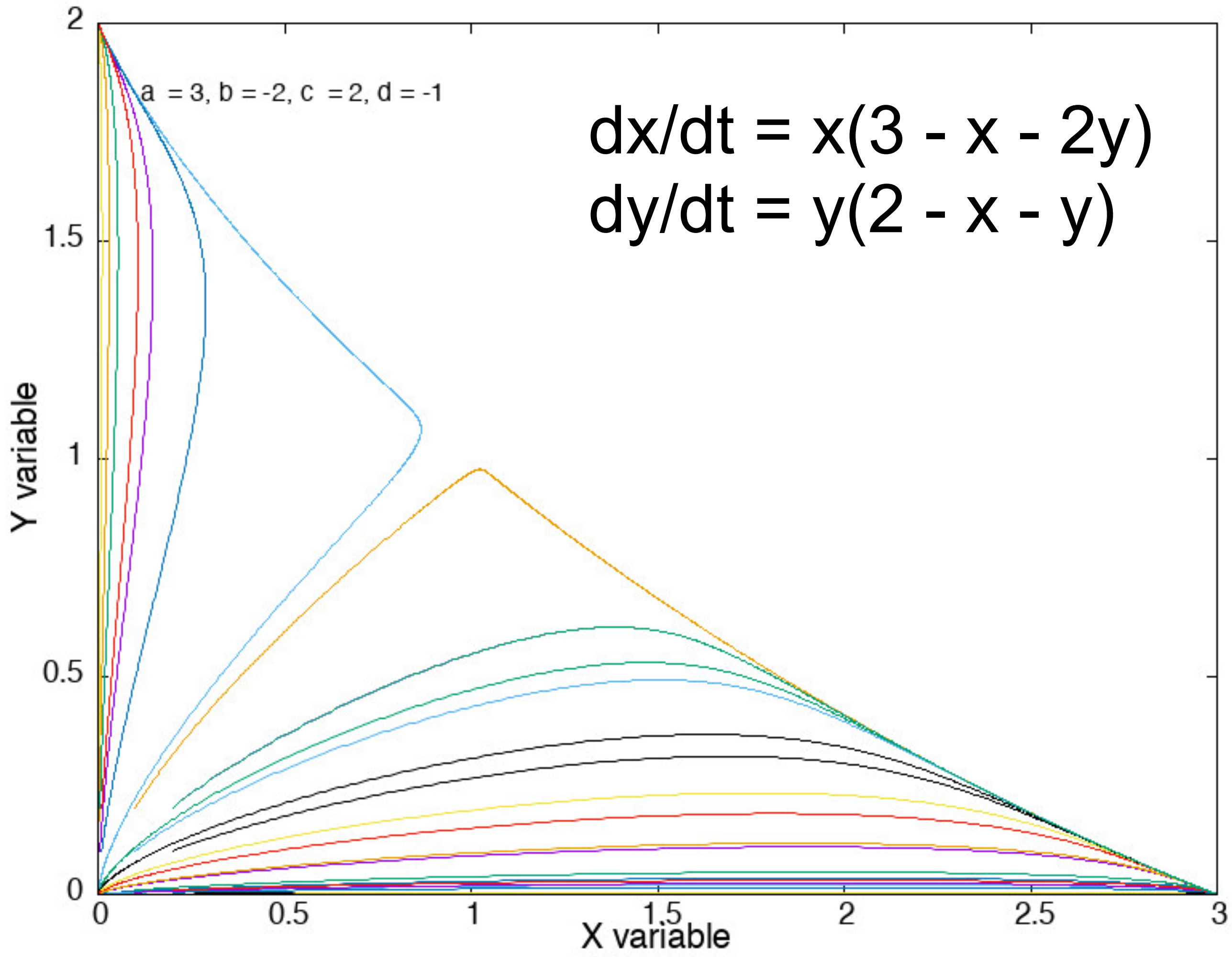
- There are only a small number of possible terms in 2D dynamical systems; once you know them, you can "mix and match" to create new models
- Small changes in parameters/initial conditions may cause huge changes in global behaviour: this is what non-linearity does.
- In 2D, trajectories can go around a FP, which allows the possibility of spirals, where the two variables oscillate in value over time until they approach the FP

There are only a limited number of interesting elements that can be combined in a dynamical system:

- constant production or decay rate $\sim s$ and $-s$
- linear growth / decay $\sim + r\,x$ / $-r\,x$
- logistic growth = $x(1 - x)$
- competition $\sim -xy$
- anti-competition $\sim xy$
- saturating competition $\sim -xy / (1 + x^{**}2)$

The goal is to be able to combine these terms to construct models. These models can apply in many fields, from engineering, neuroscience, psychology, ecology, …

How do trajectories leave the unstable node at (0,0)?



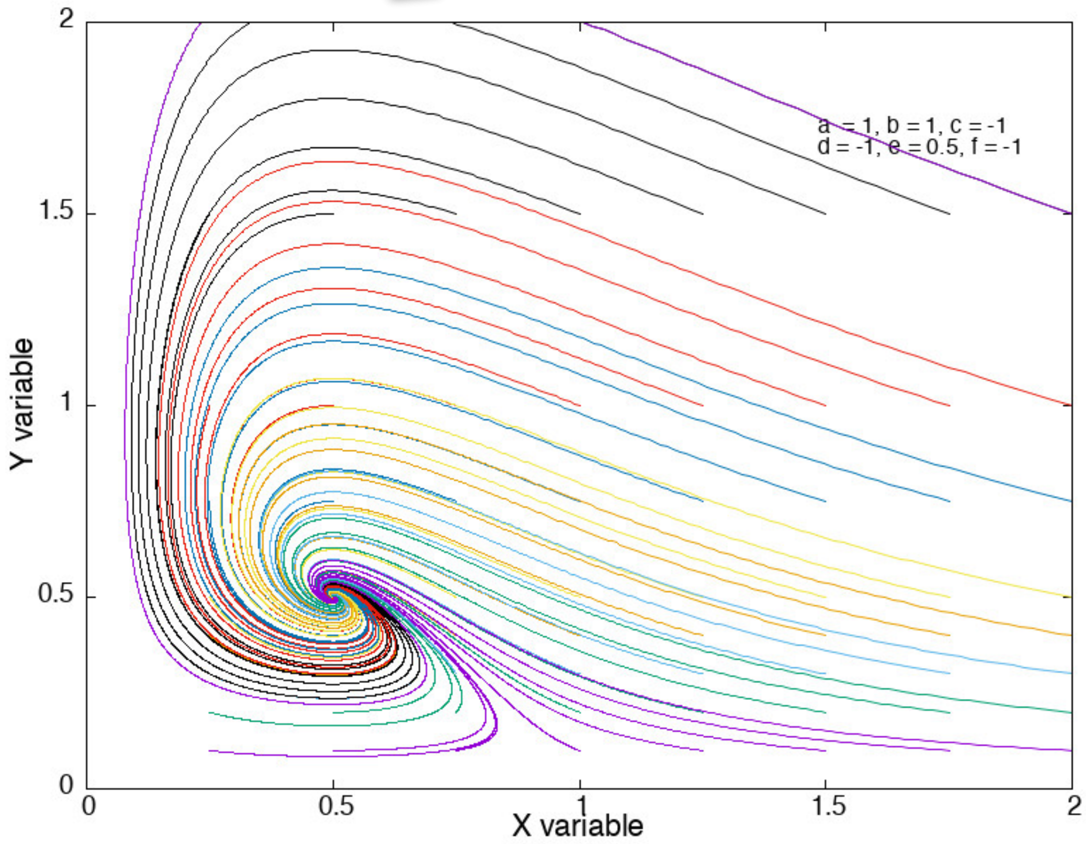
The extreme non-linearity means you cannot easily tell to which fixed point the system will move, cp. climate change, politics, economic crashes …

In the real world, however, the coefficients are **not** constants. We can change them, and small changes can make a difference.

The purpose of this course is to be able to …

convert these … into this

dN/dt = N(1 - N - P)
dP/dt = -P(1/2 - N)



Where's the spiral in the equations?

Draw phase portrait using Runge-Kutta integration scheme (see moodle, Lecture 4)

```
double dt = 0.01;

double xn, yn;
double xn1, yn1;

xn = x0;
yn = y0;

for(long i=0; i<NSTEPS; ++i)
{
    m_outStream << xn << " " << yn << zEndl;

    Runge_Kutta(params, dt, xn, yn, &xn1, &yn1);

    xn = xn1;
    yn = yn1;
}
```

Just define two functions f1, f2 that return dx/dt and dy/dt

```
double f1(const double a[6], double x, double y)
{
    return a[0]*x*(a[1] + a[2]*x - y);
}

double f2(const double a[6], double x, double y)
{
    return a[3]*y*(a[4] + a[5]*x);
}
```

```
// ****************************************


void Runge_Kutta(const double a[6], const double dt, const double xn, const double yn, double *pxn1, double *pyn1)
{
    const double k11 = f1(a, xn,yn)*dt;
    const double k12 = f2(a, xn,yn)*dt;

    const double k21 = f1(a, xn + 0.5*k11,yn + 0.5*k12)*dt;
    const double k22 = f2(a, xn + 0.5*k11,yn + 0.5*k12)*dt;

    const double k31 = f1(a, xn + 0.5*k21,yn + 0.5*k22)*dt;
    const double k32 = f2(a, xn + 0.5*k21,yn + 0.5*k22)*dt;

    const double k41 = f1(a, xn + k31,yn + k32)*dt;
    const double k42 = f2(a, xn + k31,yn + k32)*dt;

    *pxn1 = xn + 0.1666666*(k11 + 2.0*k21 + 2.0*k31 + k41);
    *pyn1 = yn + 0.1666666*(k12 + 2.0*k22 + 2.0*k32 + k42);
}
```

Background quiz

Background quiz: go.epfl.ch/turningpoint

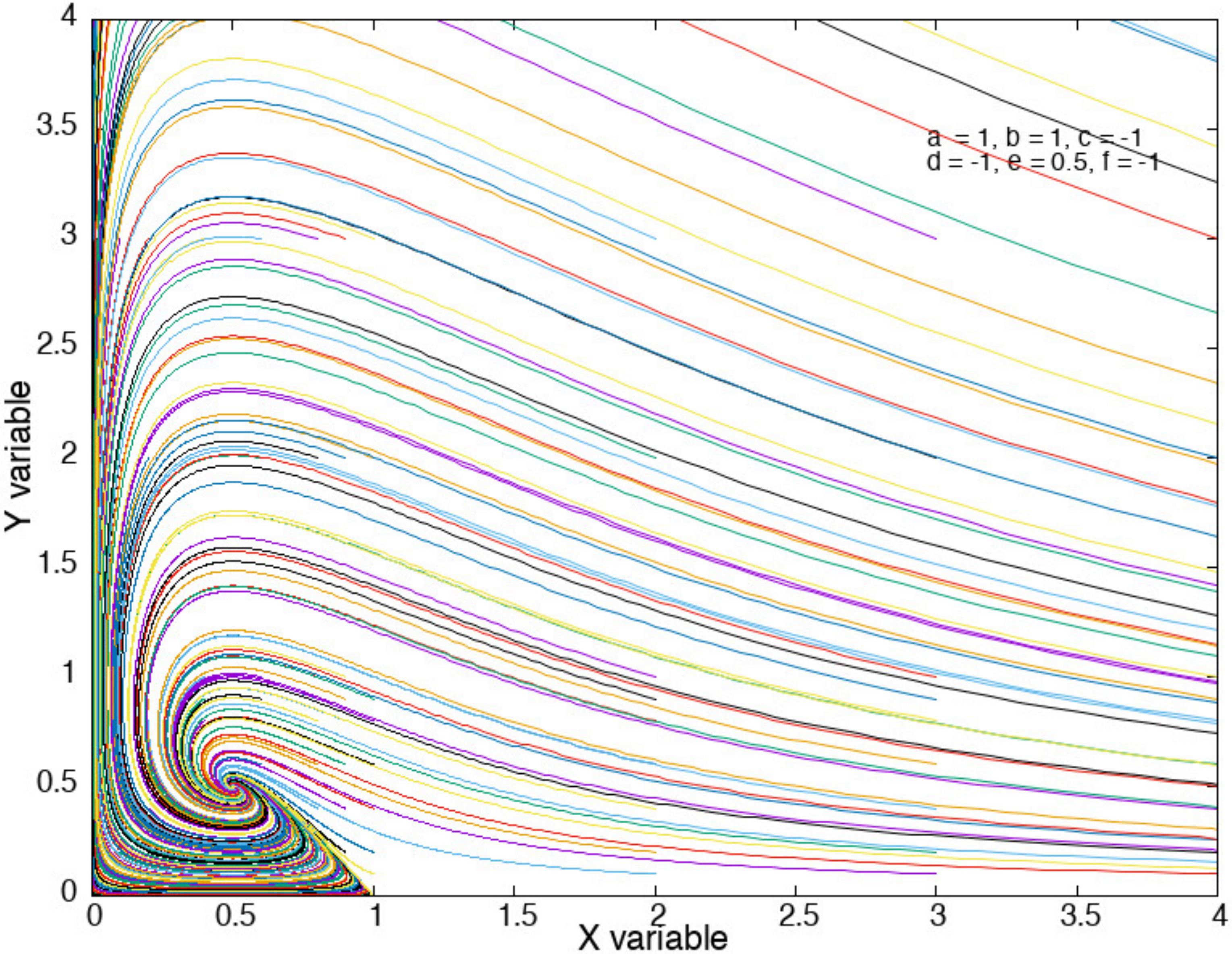
Session Id: julian23

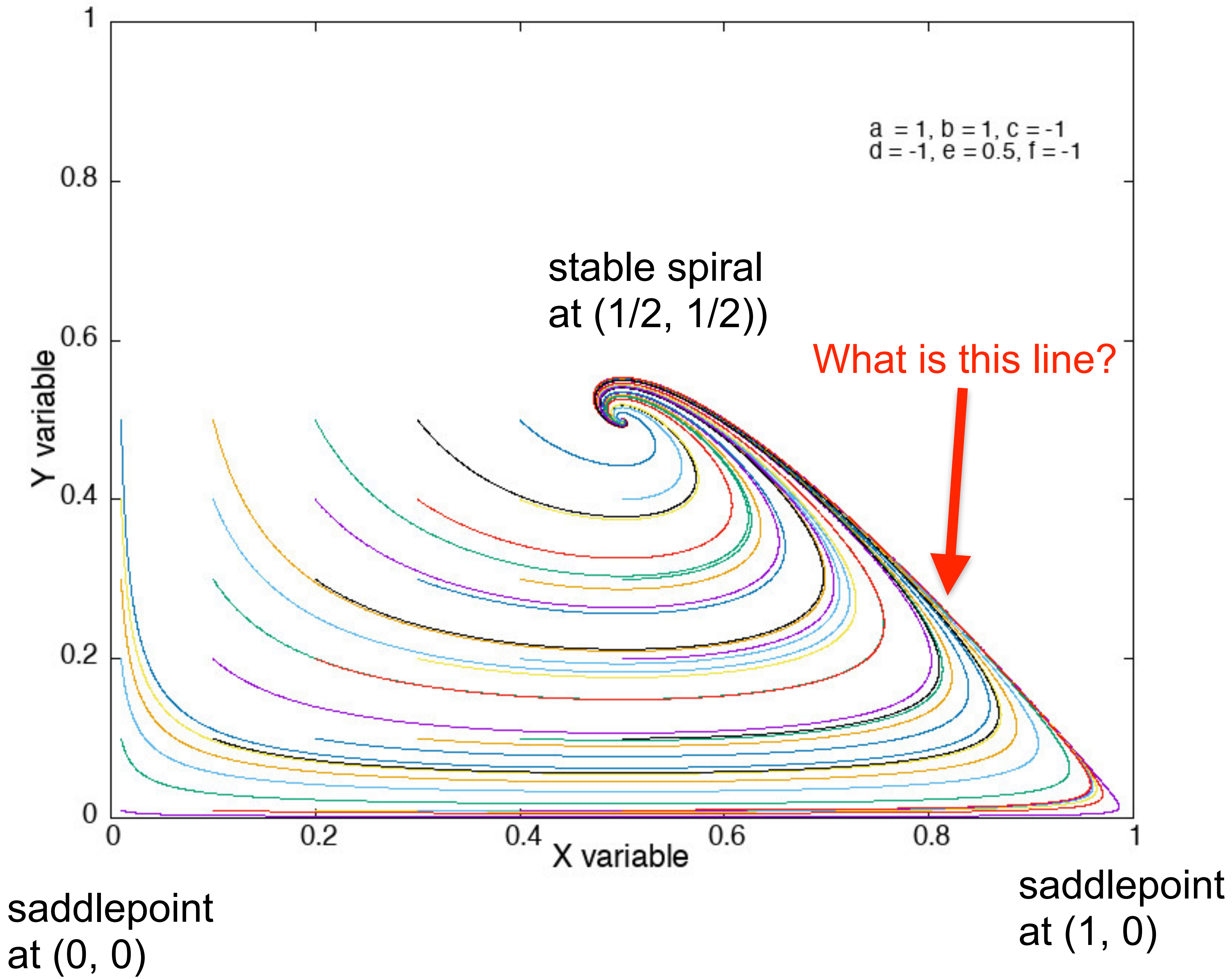
All input is anonymous; data are stored outside CH

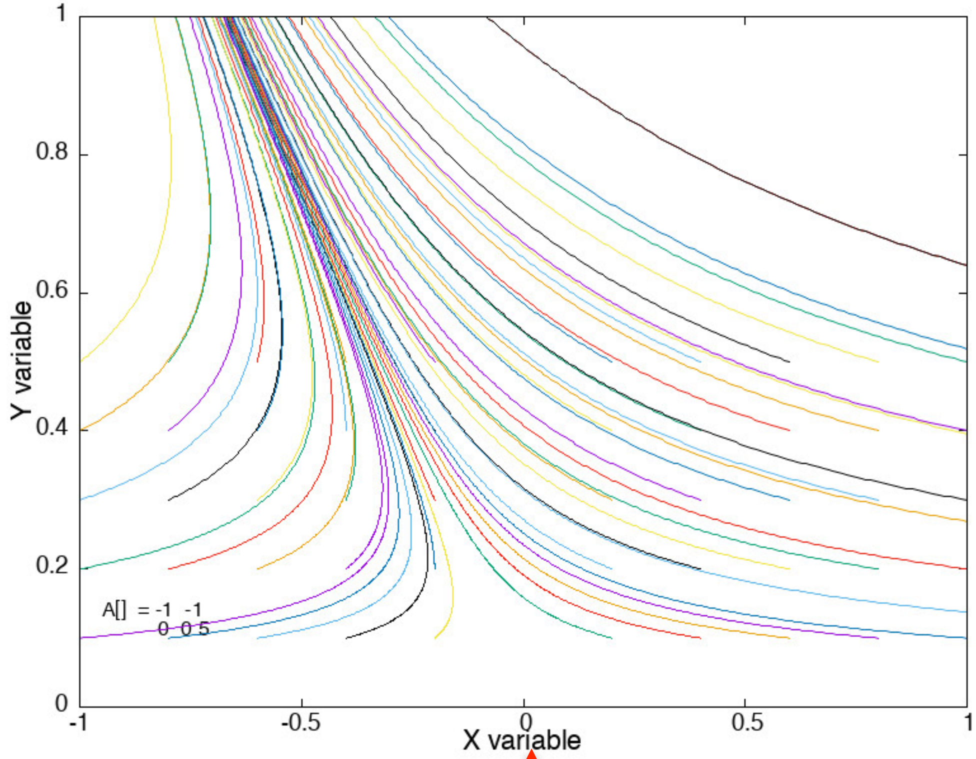
Break

Original model

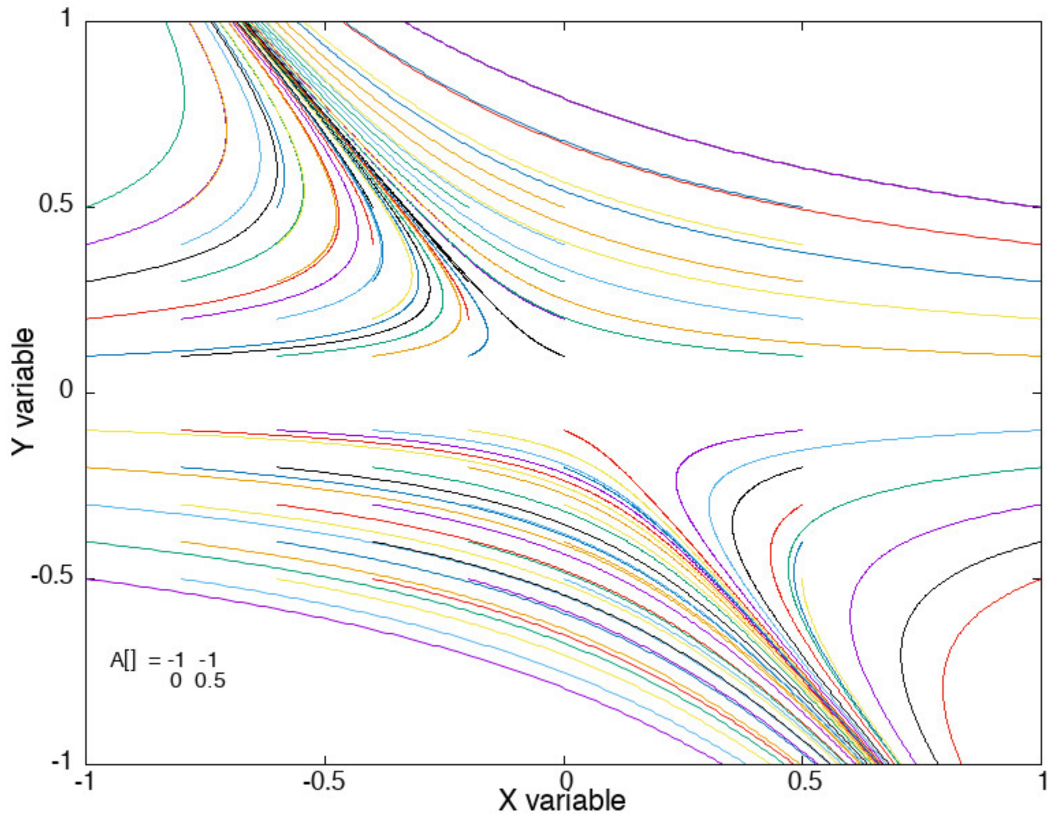
$$dN/dt = N(1 - N - P)$$
$$dP/dt = -P(1/2 - N)$$




a = 1, b = 1, c = -1
d = -1, e = 0.5, f = -1
stable spiral
at (1/2, 1/2))
What is this line?
Y variable
X variable
0
0.2
0.4
0.6
0.8
1
saddlepoint
at (0, 0)
saddlepoint
at (1, 0)



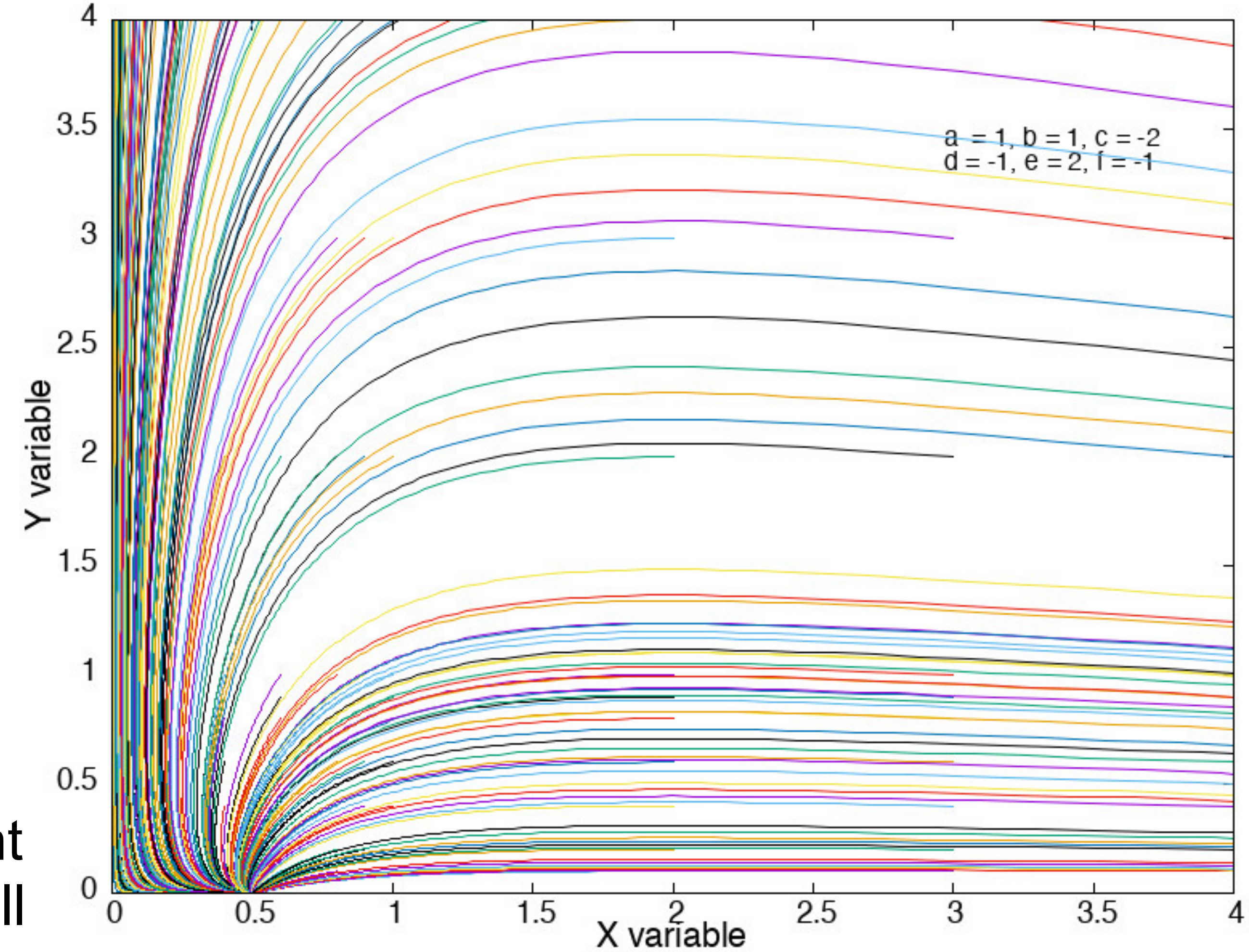
It’s the unstable manifold of the saddlepoint at (1, 0), and is in the direction (1, -3/2)



Linearised version

Modified model

$dN/dt = N(1 - 2N - P)$
$dP/dt = -P(2 - N)$



predators increase here and prey drop, but later …

saddlepoint at (0, 0) still

Stable spiral at (1/2, 1/2) disappears, and a stable node appears at (1/2, 0).

Tricky points

- To complete phase portrait, examine the behaviour of dY/dX = (dy/dt / dx/dt) for large values of (x,y), i.e., far from the fixed point(s)
- Be able to interpret the meaning of a change in the coefficients of a population model, e.g., faster loss of predators and fewer prey leads to predator extinction (spiral changes to a stable node on the x axis)

- The “shape” of an equation is a fancy way of saying that different types of term in the ODEs describing a dynamical system have specific qualitative effects; so one can assemble the terms into equations to describe different systems