ME140A: Numerical Analysis in Engineering

Lecture Notes

Alexander Meiburg

9/22/22 - 11/12/22

Outline

Course Organization

- Goal: **Numerical solution of integrals and differential equations**
- Homework will rely significantly on you programming these methods you learn. MATLAB recommended, alternatives welcome
- Collaborate on the homework! You learn more that way. Just make sure that you are, in fact, learning. ☺
- HW: 10% of grade. Exams: 30%/30%/30%.
- Submit homework via email
- Office hours by appointment or Zoom, but my schedule is very open!
- Full syllabus available here
- These notes will be continually updated here

Numerical Integration

Recall: **Differentiation** systematically lets you take a function $F(x)$ and find its derivative $f(x) = F'(x)$.

$$\frac{d}{dx}\left(e^{\sin(x+\log x)}\right) = e^{\sin(x+\log x)}\cos(x+\log x)\,(1+1/x)$$

Numerical Integration

Recall: **Differentiation** systematically lets you take a function $F(x)$ and find its derivative $f(x) = F'(x)$.

$$\frac{d}{dx}\left(e^{\sin(x+\log x)}\right) = e^{\sin(x+\log x)}\cos(x+\log x)\,(1+1/x)$$

Integration asks for the opposite. You have a handful of rules(!), but they can't cover every case. Often impossible, and we resort to defining new functions or using the computer

$$\int e^{-x^2}\,dx := \Phi(x)$$

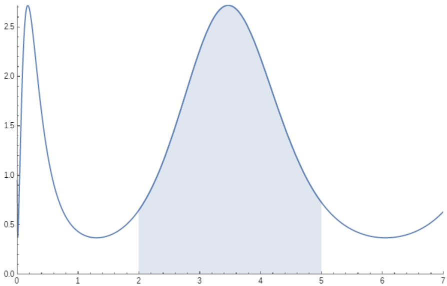
Numerical Integration

Recall: **Differentiation** systematically lets you take a function $F(x)$ and find its derivative $f(x) = F'(x)$.

$$\frac{d}{dx}\left(e^{\sin(x+\log x)}\right) = e^{\sin(x+\log x)}\cos(x+\log x)\,(1+1/x)$$

Integration asks for the opposite. You have a handful of rules(!), but they can't cover every case. Often impossible, and we resort to defining new functions or using the computer

$$\int e^{-x^2}\,dx := \Phi(x)$$



Numerical Integration

Example: Computing the position of an object after some movement.

$$y(t) = \text{Position as a function of time}$$

$$v(t) = \text{Velocity}$$

$$v = \frac{dy(t)}{dt}, \quad y(t) = \int_0^t v(t)\,dt$$

Numerical Integration

Example: Computing the position of an object after some movement.

$$y(t) = \text{Position as a function of time}$$

$$v(t) = \text{Velocity}$$

$$v = \frac{dy(t)}{dt}, \quad y(t) = \int_0^t v(t)\,dt$$

What form are we given $v(t)$? Could be:

- Explicit function of t (from theory, specifications...)

Numerical Integration

Example: Computing the position of an object after some movement.

$$y(t) = \text{Position as a function of time}$$

$$v(t) = \text{Velocity}$$

$$v = \frac{dy(t)}{dt}, \quad y(t) = \int_0^t v(t)\,dt$$

What form are we given $v(t)$? Could be:

- Explicit function of t (from theory, specifications...)
- Data (samples at certain points)

Numerical Integration

Example: Computing the position of an object after some movement.

$$y(t) = \text{Position as a function of time}$$

$$v(t) = \text{Velocity}$$

$$v = \frac{dy(t)}{dt}, \quad y(t) = \int_0^t v(t)\, dt$$

What form are we given $v(t)$? Could be:

- Explicit function of t (from theory, specifications...)
- Data (samples at certain points)
- A function of t **and** of y, or something else

This last one forms a differential equation, and will need different methods. But many ideas will transfer!

Numerical Integration

Example: Computing the position of an object after some movement.

$$y(t) = \text{Position as a function of time}$$

$$v(t) = \text{Velocity}$$

$$v = \frac{dy(t)}{dt}, \quad y(t) = \int_0^t v(t)\, dt$$

What form are we given $v(t)$? Could be:

- Explicit function of t (from theory, specifications...)
- Data (samples at certain points)
- A function of t **and** of y, or something else

This last one forms a differential equation, and will need different methods. But many ideas will transfer!
May also have y as an integral over *several* variables, not just one. *e.g.* Dust accumulating on a surface varies with x, y, and t. Can do three integrals in a row (analytically), or one 3D integral (numerically).

Aside: Numerical Differentiation

Differentiation is easy if we have an exact formula, but what about for data points?

Aside: Numerical Differentiation

Differentiation is easy if we have an exact formula, but what about for data points?

$$v(t) \approx \frac{y(t + 1\mathrm{s}) - y(t)}{1\mathrm{s}}$$

Aside: Numerical Differentiation

Differentiation is easy if we have an exact formula, but what about for data points?

$$v(t) \approx \frac{y(t+1\mathrm{s}) - y(t)}{1\mathrm{s}}$$

But consider:

t	y(t)
0	5
1	6.1
2	7.3
3	8.4
4	9.8
7	15.3
8	17.4
9	59.8
10	138.7
11	138.8

Issues such as irregular data, or gaps in time too large to understand what happened. Big question in its own right, Week 2!

Newton-Cotes

Problem: given $f(t)$, find $F(t) = \int_0^t f(t)\,dt$. If $f(t)$ is too complicated, let's find something simpler we can integrate. What's simple? Polynomials!

Newton-Cotes

Problem: given $f(t)$, find $F(t) = \int_0^t f(t)\, dt$. If $f(t)$ is too complicated, let's find something simpler we can integrate. What's simple? Polynomials!

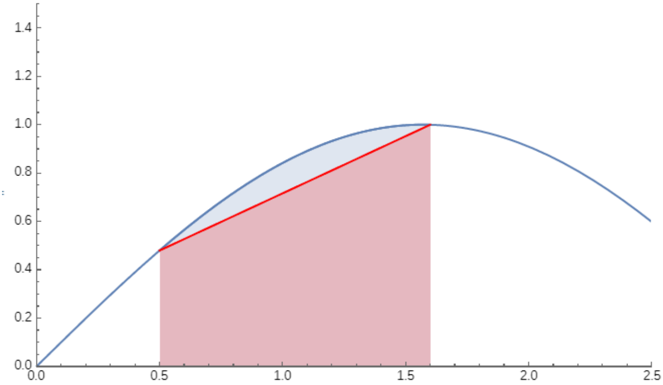
Idea: **Sample** the function at several points, **estimate** the function in between with a simpler formula, **analytically integrate** the estimate.

Newton-Cotes

Problem: given $f(t)$, find $F(t) = \int_0^t f(t)\, dt$. If $f(t)$ is too complicated, let's find something simpler we can integrate. What's simple? Polynomials!

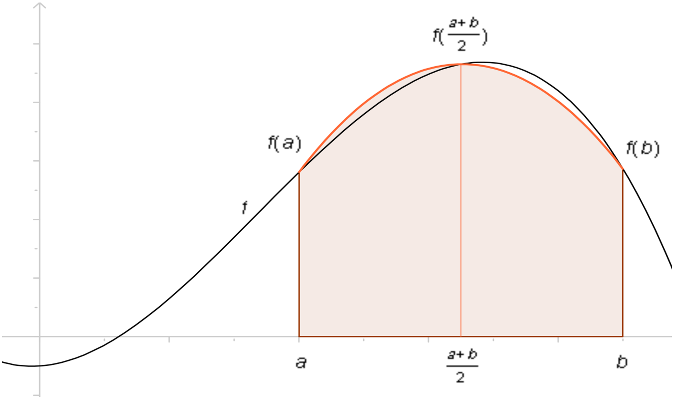
Idea: **Sample** the function at several points, **estimate** the function in between with a simpler formula, **analytically integrate** the estimate.

Simplest: Linear fit through two points. ("Trapezoidal rule")



Newton-Cotes

Fit quadratic ("Simpson's rule"):



Credit: Wikimedia

Newton-Cotes

In general, find

$$f_n(x) = a_0 + a_1 x + a_2 x^2 + \ldots a_n x^n$$

and integrate

$$\int_a^b f_n(x)\, dx$$

Turns out: a_i depend linearly on the $f(x_i)$, so the result is some weighted sum of the $f(x_i)$.

Newton-Cotes

In general, find

$$f_n(x) = a_0 + a_1 x + a_2 x^2 + \ldots a_n x^n$$

and integrate

$$\int_a^b f_n(x)\, dx$$

Turns out: a_i depend linearly on the $f(x_i)$, so the result is some weighted sum of the $f(x_i)$.

Trapezoidal:

$$\int_a^b f(x)\, dx \approx \frac{b-a}{2}(f(b) + f(a))$$

Simpson's:

$$\int_a^b f(x)\, dx \approx \frac{b-a}{6}\left(f(b) + 4f\left(\frac{a+b}{2}\right) + f(a)\right)$$

Newton-Cotes

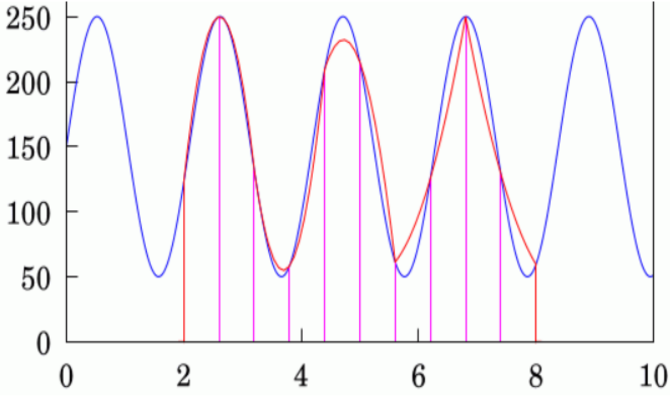
If our function is too complicated over $[a, b]$, then subdivide and do each separately.

$$\int_{x=a}^{b} f(x) = \int_{x=a}^{(a+b)/2} f(x) \quad + \quad \int_{x=(a+b)/2}^{b} f(x)$$

Newton-Cotes

If our function is too complicated over $[a, b]$, then subdivide and do each separately.

$$\int_{x=a}^{b} f(x) = \int_{x=a}^{(a+b)/2} f(x) \quad + \quad \int_{x=(a+b)/2}^{b} f(x)$$



Error Analysis

Write

$$f(x) = f(a) + f'(a)(x - a) + \frac{f''(a)}{2}(x - a)^2 + \dots$$

First and second terms accurate, third isn't. Fitting gives

$$E_{trap} \approx \frac{1}{12}|f''(\xi)|(b - a)^2$$

Error Analysis

Write

$$f(x) = f(a) + f'(a)(x - a) + \frac{f''(a)}{2}(x - a)^2 + \dots$$

First and second terms accurate, third isn't. Fitting gives

$$E_{trap} \approx \frac{1}{12}|f''(\xi)|(b - a)^2$$

Refine this with subintervals

Simple Integration Rules

- Left endpoint rule:

$$F = \int_a^b f(x) \approx \frac{(a-b)}{2} f(a)$$

$$\mathrm{Err} \leq |f'| \frac{(b-a)^2}{2}$$

Simple Integration Rules

- Left endpoint rule:

$$F = \int_a^b f(x) \approx \frac{(a-b)}{2} f(a)$$

$$\mathrm{Err} \le |f'| \frac{(b-a)^2}{2}$$

- Midpoint rule:

$$F = \int_a^b f(x) \approx \frac{(a-b)}{2} f\left(\frac{a+b}{2}\right)$$

$$\mathrm{Err} \le |f''| \frac{(b-a)^3}{24}$$

Simple Integration Rules

- Left endpoint rule:

$$F = \int_a^b f(x) \approx \frac{(a-b)}{2} f(a)$$

$$\text{Err} \leq |f'| \frac{(b-a)^2}{2}$$

- Midpoint rule:

$$F = \int_a^b f(x) \approx \frac{(a-b)}{2} f\left(\frac{a+b}{2}\right)$$

$$\text{Err} \leq |f''| \frac{(b-a)^3}{24}$$

- Trapezoid rule:

$$F = \int_a^b f(x) \approx \frac{(a-b)}{2} \frac{f(a)+f(b)}{2}$$

$$\text{Err} \leq |f''| \frac{(b-a)^3}{12}$$

Simple Integration Rules

- Simpson's "1/3" rule:

$$F = \int_a^b f(x) \approx \frac{(a-b)}{2} \frac{f(a) + 4f((a+b)/2) + f(b)}{3}$$

$$\mathrm{Err} \leq |f^4| \frac{(b-a)^5}{180}$$

Simple Integration Rules

- Simpson's "1/3" rule:

$$F = \int_a^b f(x) \approx \frac{(a-b)}{2} \frac{f(a) + 4f((a+b)/2) + f(b)}{3}$$

$$\mathrm{Err} \le |f^4| \frac{(b-a)^5}{180}$$

- Simpson's "3/8" rule:

$$F = \int_a^b f(x) \approx \frac{(a-b)}{2} \frac{f(a) + 3f((2a+b)/3) + 3f((a+2b)/3) + f(b)}{8}$$

$$\mathrm{Err} \le |f^4| \frac{(b-a)^5}{6480}$$

Composite Integration Rules

Subdivide into intervals of size $h = (b - a)/n$.

Composite Integration Rules

Subdivide into intervals of size $h = (b-a)/n$. Trapezoidal Rule:

$$\int_a^b f(x)\,dx \approx \frac{h}{2}\sum_{j=2}^{n}\left[f(x_{j-1}) + f(x_j)\right] \tag{1}$$

$$= \frac{h}{2}\left[f(x_0) + 2\sum_{j=2}^{n-1} f(x_j) + f(x_n)\right] \tag{2}$$

Composite Integration Rules

Subdivide into intervals of size $h = (b-a)/n$. Trapezoidal Rule:

$$\int_a^b f(x)\,dx \approx \frac{h}{2}\sum_{j=2}^{n}\left[f(x_{j-1}) + f(x_j)\right] \tag{1}$$

$$= \frac{h}{2}\left[f(x_0) + 2\sum_{j=2}^{n-1} f(x_j) + f(x_n)\right] \tag{2}$$

Error changes from

$$\mathrm{Err} \le |f''|\frac{(b-a)^3}{12}$$

into

$$\mathrm{Err} \le n\cdot|f''|\frac{((b-a)/n)^3}{12} = |f''|\frac{(b-a)^3}{12n^2}$$

Scaling like $1/n^2$, so this has a *second order* approximation error. (It is a *first order* rule, because it fits a first order polynomial – a line segment.)

Composite Integration Rules

Simpson's 1/3 Rule:

$$\int_a^b f(x)\,dx \approx \frac{h}{3}\sum_{j=1}^{n/2}\left[f(x_{2j-2}) + 4f(x_{2j-1}) + f(x_{2j})\right] \tag{3}$$

$$= \frac{h}{3}\left[f(x_0) + 4\sum_{j=1}^{n/2} f(x_{2j-1}) + 2\sum_{j=1}^{n/2-1} f(x_{2j}) + f(x_n)\right] \tag{4}$$

Composite Integration Rules

Simpson's 1/3 Rule:

$$\int_a^b f(x)\,dx \approx \frac{h}{3}\sum_{j=1}^{n/2}\left[f(x_{2j-2}) + 4f(x_{2j-1}) + f(x_{2j})\right] \tag{3}$$

$$= \frac{h}{3}\left[f(x_0) + 4\sum_{j=1}^{n/2} f(x_{2j-1}) + 2\sum_{j=1}^{n/2-1} f(x_{2j}) + f(x_n)\right] \tag{4}$$

Error changes from

$$\text{Err} \leq |f^4|\frac{(b-a)^5}{180}$$

into

$$\text{Err} \leq n \cdot |f^4|\frac{((b-a)/n)^5}{180} = |f^4|\frac{(b-a)^5}{180n^4}$$

Scaling like $1/n^4$, so this has a *fourth order* approximation error. (It is a *second order* rule, because it fits a second order polynomial.)

Composite Integration Rules

...and beyond?

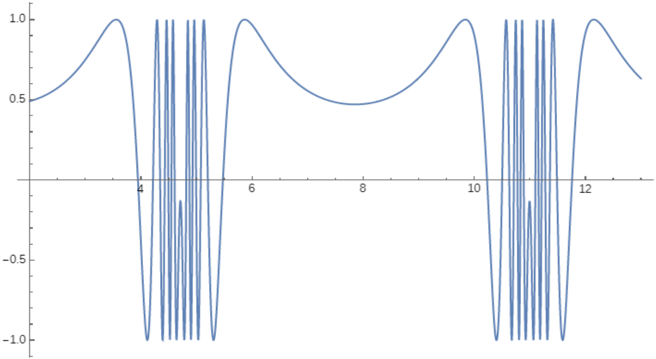
Composite Integration Rules

...and beyond? These have been extended to use 4th, 5th, 6th... order polynomials, and get higher-order methods. In practice, the $1/n^k$ is not the limiting factor if $k > 4$, and the integral will only improve with smaller intervals.

Composite Integration Rules

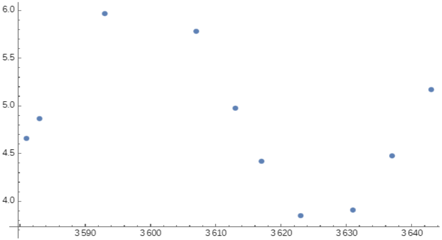
...and beyond? These have been extended to use 4th, 5th, 6th... order polynomials, and get higher-order methods. In practice, the $1/n^k$ is not the limiting factor if $k > 4$, and the integral will only improve with smaller intervals.

```
Plot[Sin[1/(Sin[x]+1.04)], {x, 2, 13}, PlotRange → All]
```

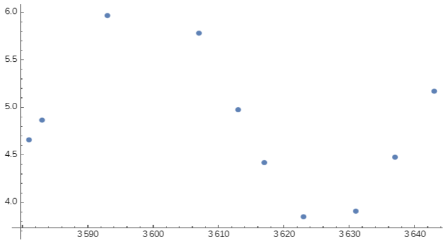


You just can't get this accurate, without having small intervals! And once you get small enough, the function will be roughly quadratic anyway.

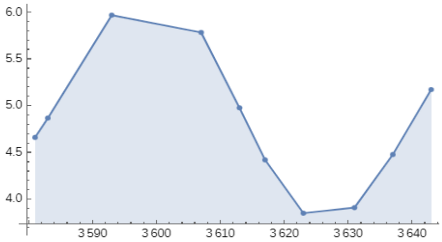
Irregular Integration



Irregular Integration

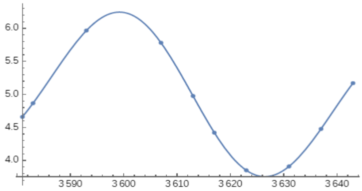


Trapezoidal integration on each part:



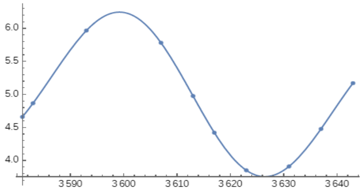
Irregular Integration

In principle, we can fit higher-order polynomials as well

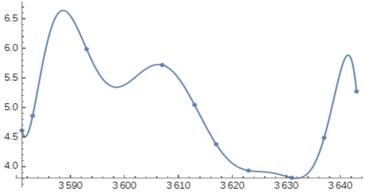


Irregular Integration

In principle, we can fit higher-order polynomials as well



But this can again be very sensitive and unstable:



In practice, also expensive to compute. Need to recompute the "weights" each time, which requires solving a linear system.

Multidimensional Integration

Computing

$$F = \int_{x_1}^{x_2} \int_{y_1}^{y_2} e^{x \sin(y)} + \frac{\ln(y-x)}{\ln(y)} \, dy \, dx$$

Multidimensional Integration

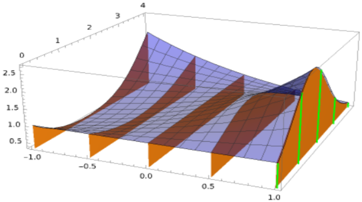
Computing

$$F = \int_{x_1}^{x_2} \int_{y_1}^{y_2} e^{x \sin(y)} + \frac{\ln(y-x)}{\ln(y)}\, dy\, dx$$

One option: do each integral with its own 1D algorithm.

$$F = \int_{x_1}^{x_2} G(x)\, dx$$

$$G(x) = \int_{y_1}^{y_2} e^{x \sin(y)} + \frac{\ln(y-x)}{\ln(y)}\, dy$$

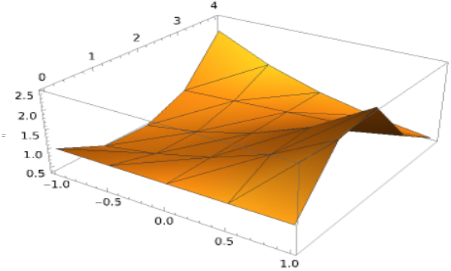


Multidimensional Integration

Computing

$$F = \int_{x_1}^{x_2} \int_{y_1}^{y_2} e^{x \sin(y)} + \frac{\ln(y-x)}{\ln(y)} \, dy \, dx$$

Or, custom multi-dimensional versions of integration rules. Simple functions integrated over squares, triangles, etc.



Richardson Extrapolation

Trapezoidal rule:

$$I = I_n + |f''| \frac{(b-a)^3}{12n^2} + \textit{higher order}$$

As a function of n, can we study the behavior?

Richardson Extrapolation

Trapezoidal rule:

$$I = I_n + |f''| \frac{(b-a)^3}{12n^2} + \textit{higher order}$$

As a function of n, can we study the behavior?

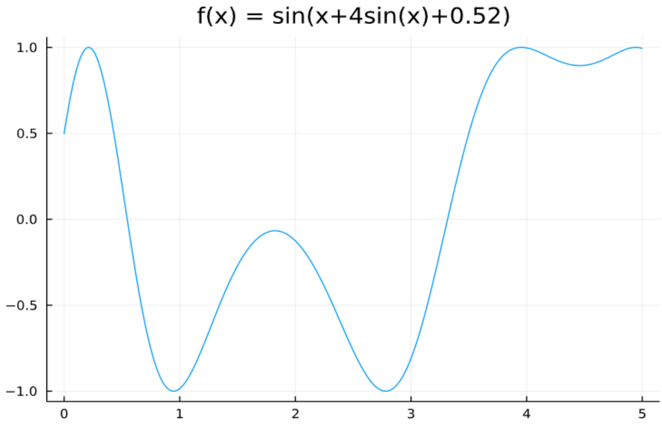
$$I_n \approx c_1 + \frac{c_2}{n^2}$$

Fit some data points, extract the true integral c_1?
With the higher terms,

$$I_n \approx c_1 + \frac{c_2}{n^2} + \frac{c_3}{n^3} + \ldots ?$$

(NB: Actually the next term is only $1/n^4$)

Richardson Extrapolation



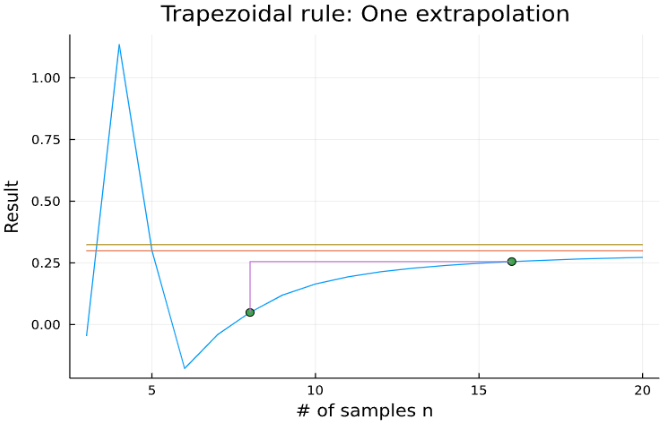
Inspect our function. What kind of accuracy do you think we need?

Richardson Extrapolation

Accuracy has initial "bad" period, then for $n \geq 6$ we see smooth $1/n^2$ decay in error. Estimate the asymptote

Richardson Extrapolation

Richardson Extrapolation



Two point estimate of asymptote: $\frac{4}{3}I_{16} - \frac{1}{3}I_8$

Richardson Extrapolation

This result is a *new* integration rule: take any previous rule with error $1/n^k$, and cancel out the errors. This gives you a new rule with error $1/n^{k+2}$.

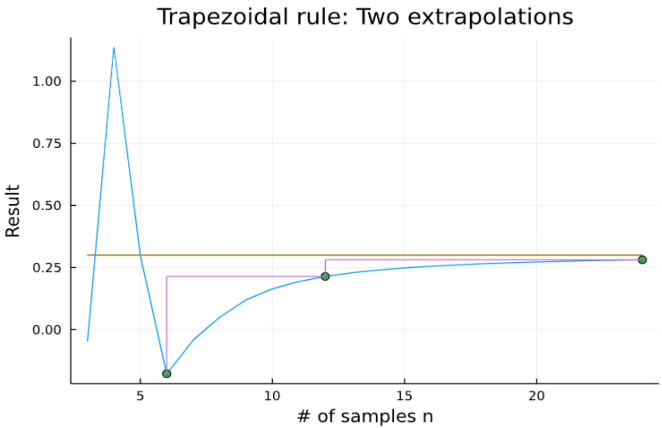
If we one integral uses a subset of the other integral's points, we don't need any new samples.

Richardson Extrapolation

This result is a *new* integration rule: take any previous rule with error $1/n^k$, and cancel out the errors. This gives you a new rule with error $1/n^{k+2}$.

If we one integral uses a subset of the other integral's points, we don't need any new samples. ...and we can apply this process to itself, to keep raising the order!

Richardson Extrapolation



Four point estimate of asymptote: $\frac{64}{45} I_{24} - \frac{20}{45} I_{12} + \frac{1}{45} I_6$

Richardson Extrapolation

$n = 24$ trapezoidal error: 0.0183
$n = 24$ one extrapolation: 0.00402
$n = 24$ two extrapolations: 0.00127
We got 10x the accuracy with no extra samples!

Romberg Algorithm

$$I_{j,k} = \frac{4^{k-1} I_{j+1,k-1} - I_{j,k-1}}{4^{k-1} - 1}$$

Here, j is doubling the number of samples: e.g. $j = 4$ has 1000 samples means $j = 5$ has 2000 samples. k is the order of the method: original trapezoid rule starts at $k = 1$.

Romberg Algorithm

$$I_{j,k} = \frac{4^{k-1} I_{j+1,k-1} - I_{j,k-1}}{4^{k-1} - 1}$$

Here, j is doubling the number of samples: e.g. $j = 4$ has 1000 samples means $j = 5$ has 2000 samples. k is the order of the method: original trapezoid rule starts at $k = 1$. Lets us estimate error:

$$|\epsilon| \approx \left| \frac{I_{1,k} - I_{2,k-1}}{I_{1,k}} \right|$$

Gaussian Quadrature

Choosing our integration points wisely: reducing error.

$$x_0 = \frac{a+b}{2} - \frac{b-a}{\sqrt{3}}, \quad x_1 = \frac{a+b}{2} + \frac{b-a}{\sqrt{3}}$$

yields a *fourth* order method (depends on $|f^4|$) with only two samples.
[Picture]

1D Stability

Consider a one-variable ODE,

$$\frac{dx}{dt} = f(x, t)$$

What can we say about long-term behavior of $x(t)$?

1D Stability

Consider a one-variable ODE,

$$\frac{dx}{dt} = f(x, t)$$

What can we say about long-term behavior of $x(t)$?

Not much, since f can depend on t arbitrarily. But if f only depends on x, then we can only have *monotonic* behavior.

1D Stability

Consider a one-variable ODE,

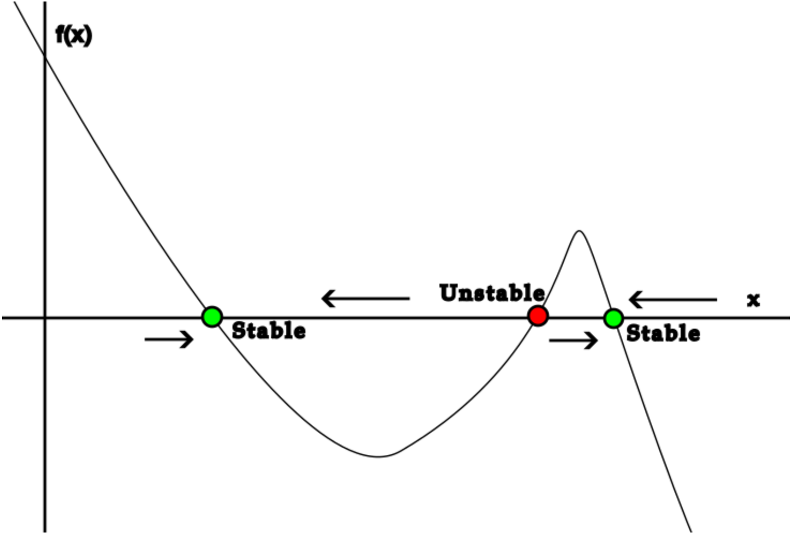
$$\frac{dx}{dt} = f(x, t)$$

What can we say about long-term behavior of $x(t)$?

Not much, since f can depend on t arbitrarily. But if f only depends on x, then we can only have *monotonic* behavior.

Can't have $x(t) = \sin(t)$, because at the same value of $x = 0$ we have both $\frac{dx}{dt} = 1$ (when $t = 0$) and $\frac{dx}{dt} = -1$ (when $t = \pi$).

1D Stability



1D Stability

The *fixed points* will occur where

$$\frac{dx}{dt} = 0 \implies f(x) = 0$$

1D Stability

The *fixed points* will occur where

$$\frac{dx}{dt} = 0 \implies f(x) = 0$$

Suppose x_0 is a fixed point, and we're at $x(0) = x_0 + \epsilon$. What happens?

1D Stability

The *fixed points* will occur where

$$\frac{dx}{dt} = 0 \implies f(x) = 0$$

Suppose x_0 is a fixed point, and we're at $x(0) = x_0 + \epsilon$. What happens?

If $f(x)$ is positive, then we'll keep increasing, and move away from the fixed point, and ϵ grows. If $f(x)$ is negative, we'll decrease, and head back towards x_0.

1D Stability

The *fixed points* will occur where

$$\frac{dx}{dt} = 0 \implies f(x) = 0$$

Suppose x_0 is a fixed point, and we're at $x(0) = x_0 + \epsilon$. What happens?

If $f(x)$ is positive, then we'll keep increasing, and move away from the fixed point, and ϵ grows. If $f(x)$ is negative, we'll decrease, and head back towards x_0.

If we're at $x(0) = x_0 - \epsilon$, opposite occurs. Can tell which will happen using the *derivative of f*.

1D Stability

The *fixed points* will occur where

$$\frac{dx}{dt} = 0 \implies f(x) = 0$$

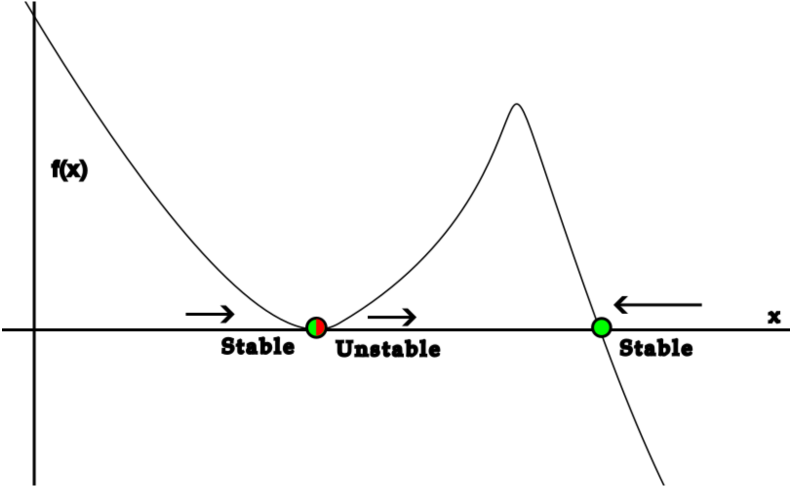
Suppose x_0 is a fixed point, and we're at $x(0) = x_0 + \epsilon$. What happens?

If $f(x)$ is positive, then we'll keep increasing, and move away from the fixed point, and ϵ grows. If $f(x)$ is negative, we'll decrease, and head back towards x_0.

If we're at $x(0) = x_0 - \epsilon$, opposite occurs. Can tell which will happen using the *derivative of f*.

$$\begin{aligned} f'(x_0) > 0 &\implies \epsilon \text{ grows} \implies \text{Unstable} \\ f'(x_0) < 0 &\implies \epsilon \text{ shrinks} \implies \text{Stable} \end{aligned}$$

1D Stability



1D Stability

When $f(x_0) = 0$ and $f'(x_0) = 0$, we can have *half stability*: stable from one side, not from the other. Which side is which depends on $f''(x_0)$. Not all such points are half-stable. For example, consider $f(x) = x^3$. Also keep in mind $f(x) = |x|$ or $f(x) = -\sqrt[3]{x}$. Now f isn't differentiable at zero, but we can still see that $|x|$ is half-stable and $-\sqrt[3]{x}$ is stable.

Multivariate Stability

What happens with the two-variable system

$$\frac{dx}{dt} = f(x, y)$$

$$\frac{dy}{dt} = g(x, y)$$

Multivariate Stability

What happens with the two-variable system

$$\frac{dx}{dt} = f(x, y)$$

$$\frac{dy}{dt} = g(x, y)$$

Remember that we can always right 2nd-order ODEs as 1st-order ODEs in new variables.

$$\frac{d_2 x}{dx^2} = -x \implies x(t) = \sin(t)$$

becomes

$$\frac{dx}{dt} = v$$

$$\frac{dv}{dt} = -x.$$

Multivariate Stability

What happens with the two-variable system

$$\frac{dx}{dt} = f(x, y)$$

$$\frac{dy}{dt} = g(x, y)$$

Remember that we can always right 2nd-order ODEs as 1st-order ODEs in new variables.

$$\frac{d_2 x}{dx^2} = -x \implies x(t) = \sin(t)$$

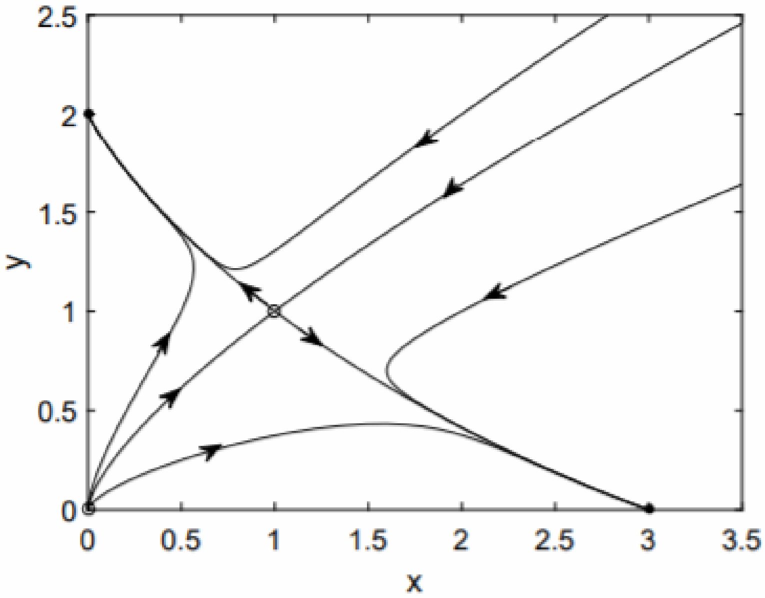
becomes

$$\frac{dx}{dt} = v$$

$$\frac{dv}{dt} = -x.$$

More than just unstable and stable! Can have indefinite oscillations.

Multivariate Stability



Can still discuss fixed points, at least. Points where

$$\frac{dx}{dt} = f(x, y) = 0, \quad \frac{dy}{dt} = g(x, y) = 0$$

Can still discuss fixed points, at least. Points where

$$\frac{dx}{dt} = f(x, y) = 0, \quad \frac{dy}{dt} = g(x, y) = 0$$

Fixed points can be:

- Stable: small perturbation shrinks in size

Can still discuss fixed points, at least. Points where

$$\frac{dx}{dt} = f(x, y) = 0, \quad \frac{dy}{dt} = g(x, y) = 0$$

Fixed points can be:

- Stable: small perturbation shrinks in size
- Unstable: small perturbations grow without bound

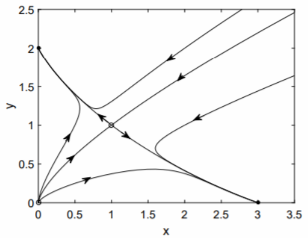
Can still discuss fixed points, at least. Points where

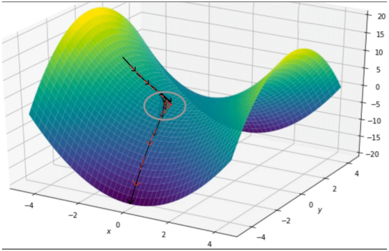
$$\frac{dx}{dt} = f(x, y) = 0, \quad \frac{dy}{dt} = g(x, y) = 0$$

Fixed points can be:

- Stable: small perturbation shrinks in size
- Unstable: small perturbations grow without bound
- Saddle nodes / saddle points: approach from two directions, but ultimately unstable

Multivariate Stability





credit

Multivariate Stability

Cordura Trail

credit

Multivariate Stability

Often the fixed points don't paint a complete story, and what we care about is oscillations, or cycles.

Multivariate Stability

Often the fixed points don't paint a complete story, and what we care about is oscillations, or cycles.

Cycles could:

- Gradually peter out – actually a stable fixed point! Damped harmonic oscillator.

Multivariate Stability

Often the fixed points don't paint a complete story, and what we care about is oscillations, or cycles.

Cycles could:

- Gradually peter out – actually a stable fixed point! Damped harmonic oscillator.
- Conserve some quantity. Infinitely many cycles, coexisting. Undamped oscillator

Multivariate Stability

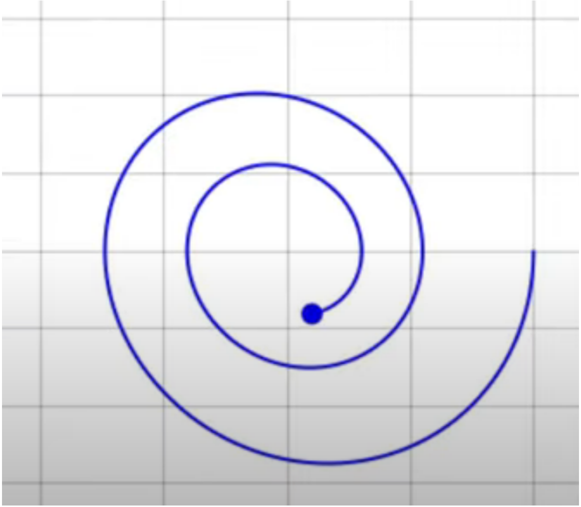
Often the fixed points don't paint a complete story, and what we care about is oscillations, or cycles.

Cycles could:

- Gradually peter out – actually a stable fixed point! Damped harmonic oscillator.
- Conserve some quantity. Infinitely many cycles, coexisting. Undamped oscillator
- Have a natural radius – stable cycles! Driven, damped oscillator (e.g. vibrating, resonating machinery)

Multivariate Stability

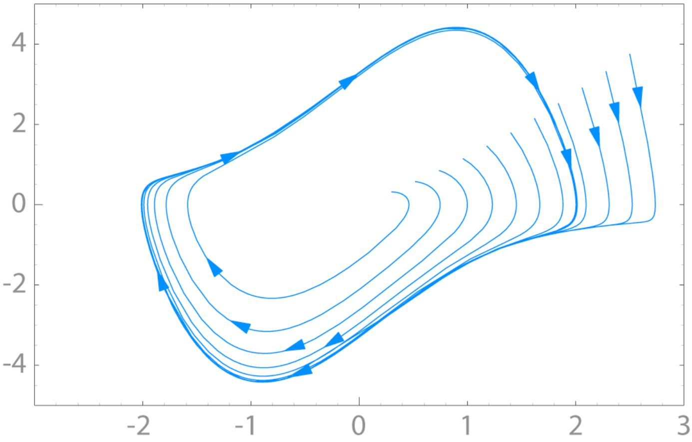
Damped harmonic oscillator:



Fixed point at $(0, 0)$. The derivatives there can tell us whether it falls "straight" in or spirals around.

Multivariate Stability

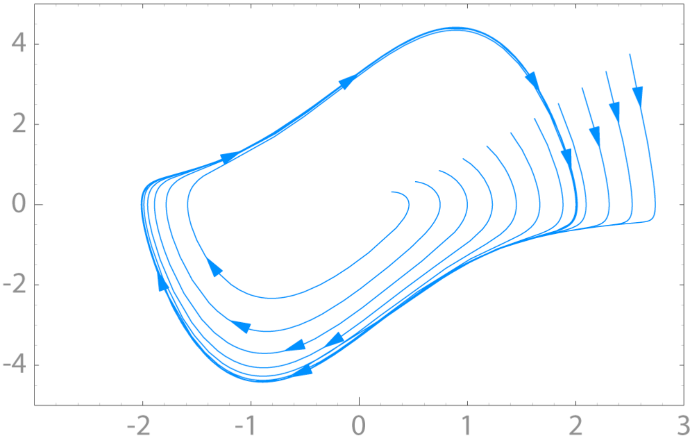
Stable, driven cycle – or *limit cycle*. e.g. Van der Pol equation:



credit

Multivariate Stability

Stable, driven cycle – or *limit cycle*. e.g. Van der Pol equation:



credit Run it backwards, and we have an *unstable limit cycle*.

Jacobian

How do we determine the stability of a fixed point in more variables?

Jacobian

How do we determine the stability of a fixed point in more variables?

$$J = \begin{bmatrix} \frac{\partial f}{\partial x} & \frac{\partial f}{\partial y} \\ \frac{\partial g}{\partial x} & \frac{\partial g}{\partial y} \end{bmatrix}$$

Intuitively, J tells us how

$$x(0) = x_0 + \epsilon_x, \quad y(0) = y_0 + \epsilon_y$$

evolves. Approximately,

$$\begin{bmatrix} \frac{d\epsilon_x}{dt} \\ \frac{d\epsilon_y}{dt} \end{bmatrix} = J \begin{bmatrix} \epsilon_x \\ \epsilon_y \end{bmatrix}$$

Jacobian

How do we determine the stability of a fixed point in more variables?

$$J = \begin{bmatrix} \frac{\partial f}{\partial x} & \frac{\partial f}{\partial y} \\ \frac{\partial g}{\partial x} & \frac{\partial g}{\partial y} \end{bmatrix}$$

Intuitively, J tells us how

$$x(0) = x_0 + \epsilon_x, \quad y(0) = y_0 + \epsilon_y$$

evolves. Approximately,

$$\begin{bmatrix} \frac{d\epsilon_x}{dt} \\ \frac{d\epsilon_y}{dt} \end{bmatrix} = J \begin{bmatrix} \epsilon_x \\ \epsilon_y \end{bmatrix}$$

This *linear* ODE has a simple solution in terms of the eigenvalues of J.

Jacobian

In terms of the eigenvalues of J:

- All eigenvalues of J have positive real part: unstable
- All negative: stable fixed point
- Mixture of positive and negative: saddle node
- Zero real part: like 1D, needs further analysis.
- Nonzero imaginary parts means they spiral

Jacobian

In terms of the eigenvalues of J:

- All eigenvalues of J have positive real part: unstable
- All negative: stable fixed point
- Mixture of positive and negative: saddle node
- Zero real part: like 1D, needs further analysis.
- Nonzero imaginary parts means they spiral

This all holds in any number of variables. In 2D, easy tests:

$$J = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

$$\det(J) = ad - bc, \quad \mathrm{Tr}(J) = a + d$$

The $\det(J)$ is the product of the two eigenvalues, $\mathrm{Tr}(J)$ is the sum.

Jacobian

In terms of the eigenvalues of J:

- All eigenvalues of J have positive real part: unstable
- All negative: stable fixed point
- Mixture of positive and negative: saddle node
- Zero real part: like 1D, needs further analysis.
- Nonzero imaginary parts means they spiral

This all holds in any number of variables. In 2D, easy tests:

$$J = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

$$\det(J) = ad - bc, \quad \mathrm{Tr}(J) = a + d$$

The $\det(J)$ is the product of the two eigenvalues, $\mathrm{Tr}(J)$ is the sum. Stable if and only if $\det(J) > 0$ and $\mathrm{Tr}(J) < 0$. Saddle node if $\det(J) < 0$. Spirals if

$$\det(J) > \frac{(\mathrm{Tr}(J))^2}{4}$$

Jacobian: Damped Harmonic Oscillator

Example: a damped harmonic oscillator.

$$x''(t) = -kx - \gamma x'(t)$$

A stiffness $k > 0$ and damping friction $\gamma \geq 0$.

Jacobian: Damped Harmonic Oscillator

Example: a damped harmonic oscillator.

$$x''(t) = -kx - \gamma x'(t)$$

A stiffness $k > 0$ and damping friction $\gamma \geq 0$. Rewrite as two first-order equations:

$$x'(t) = v, \quad v'(t) = -kx - \gamma v$$

Jacobian: Damped Harmonic Oscillator

Example: a damped harmonic oscillator.

$$x''(t) = -kx - \gamma x'(t)$$

A stiffness $k > 0$ and damping friction $\gamma \geq 0$. Rewrite as two first-order equations:

$$x'(t) = v, \quad v'(t) = -kx - \gamma v$$

Jacobian is

$$J = \begin{bmatrix} 0 & 1 \\ -k & -\gamma \end{bmatrix}$$

Jacobian: Damped Harmonic Oscillator

Example: a damped harmonic oscillator.

$$x''(t) = -kx - \gamma x'(t)$$

A stiffness $k > 0$ and damping friction $\gamma \geq 0$. Rewrite as two first-order equations:

$$x'(t) = v, \quad v'(t) = -kx - \gamma v$$

Jacobian is

$$J = \begin{bmatrix} 0 & 1 \\ -k & -\gamma \end{bmatrix}$$

Determinant and trace:

$$\det(J) = k, \quad \mathrm{Tr}(J) = -\gamma$$

Jacobian: Damped Harmonic Oscillator

Example: a damped harmonic oscillator.

$$x''(t) = -kx - \gamma x'(t)$$

A stiffness $k > 0$ and damping friction $\gamma \geq 0$. Rewrite as two first-order equations:

$$x'(t) = v, \quad v'(t) = -kx - \gamma v$$

Jacobian is

$$J = \begin{bmatrix} 0 & 1 \\ -k & -\gamma \end{bmatrix}$$

Determinant and trace:

$$\det(J) = k, \quad \mathrm{Tr}(J) = -\gamma$$

See that the determinant is always positive, trace is negative when $\gamma > 0$. So it's stable! Has spirals as long as $k > \frac{\gamma^2}{4}$, the "underdamped" regime.

Jacobian: Damped Harmonic Oscillator

Example: a damped harmonic oscillator.

$$x''(t) = -kx - \gamma x'(t)$$

A stiffness $k > 0$ and damping friction $\gamma \geq 0$. Rewrite as two first-order equations:

$$x'(t) = v, \quad v'(t) = -kx - \gamma v$$

Jacobian is

$$J = \begin{bmatrix} 0 & 1 \\ -k & -\gamma \end{bmatrix}$$

Determinant and trace:

$$\det(J) = k, \quad \mathrm{Tr}(J) = -\gamma$$

See that the determinant is always positive, trace is negative when $\gamma > 0$. So it's stable! Has spirals as long as $k > \frac{\gamma^2}{4}$, the "underdamped" regime.

If $\gamma = 0$, the eigenvalues are $\pm i\sqrt{k}$ – purely imaginary, and it's neither stable nor unstable (just oscillates forever).

Poincaré Maps

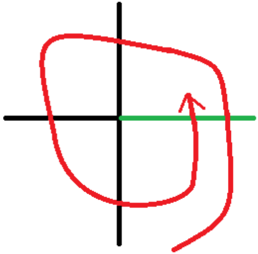
Studying cyclic or periodic behavior is, as we commented earlier, generally hard. How can we try to address it?
Let's first discuss a time-independent (i.e. homogeneous) system. If we have an (approximate?) cycle, then we should expect it to return to a similar point at a later time. We can formalize this by picking some condition, and asking what it takes to return to that condition. For example, if we think our system cycles in (x, y) around the origin, we could pick:

$$y = 0, \quad x \geq 0$$

Poincaré Maps

$$y = 0, \quad x \geq 0$$

In the picture below, this is asking for moments when we cross the green segment:



This gives a map function $M(x)$. Given a point x_1 where we cross the line, what is the *next* point $x_2 = M(x_1)$ where we'll cross it again?

Poincaré Maps

This discrete map $M(x)$ is called a Poincaré map of the system. Usually we can't compute it in closed form, but we can solve of our system from many points and build up a picture of what's going on.

Poincaré Maps

This discrete map $M(x)$ is called a Poincaré map of the system. Usually we can't compute it in closed form, but we can solve of our system from many points and build up a picture of what's going on.

In more variables, this condition would probably a plane (e.g. $x = 0$, $\frac{dx}{dt} > 0$, and any values for y and z). Then it would be a vector function $\vec{M}(y, z)$ that gives the new values of both y and z.

Poincaré Maps

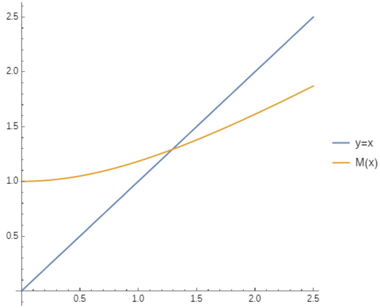
This discrete map $M(x)$ is called a Poincaré map of the system. Usually we can't compute it in closed form, but we can solve of our system from many points and build up a picture of what's going on.

In more variables, this condition would probably a plane (e.g. $x = 0$, $\frac{dx}{dt} > 0$, and any values for y and z). Then it would be a vector function $\vec{M}(y, z)$ that gives the new values of both y and z.

This means that now understanding the behavior of our system is a question about *discrete* dynamics, instead of *continuous* time. This is, in general, more complicated and messy, because we can't reason about "flows" in our space! But we can still get somewhere.

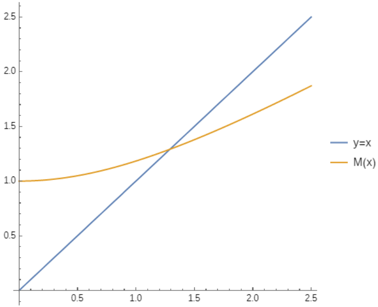
Discrete Dynamics

A useful picture (in the univariate case, $M(x)$) is to plot $M(X)$ against x:



Discrete Dynamics

A useful picture (in the univariate case, $M(x)$) is to plot $M(X)$ against x:



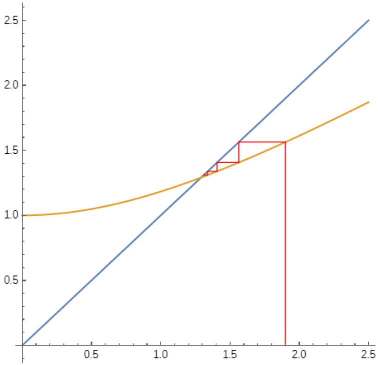
The point where the two curves intersect is an exact cycle. The *cycle* of the original system is a *fixed point* of the map M. If we start at that point $x \approx 1.6$, then the map M will bring us back to that same point. If we start at a point $x = 1.9$ away, we can plot how each new iteration changes our position:

Discrete Dynamics

The point where the two curves intersect is an exact cycle. If we start at that point $x \approx 1.6$, then the map M will bring us back to that same point. If we start at a point $x = 1.9$ away, we can plot how each new iteration changes our position:

Discrete Dynamics

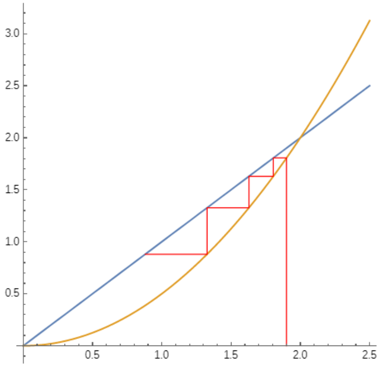
The point where the two curves intersect is an exact cycle. If we start at that point $x \approx 1.6$, then the map M will bring us back to that same point. If we start at a point $x = 1.9$ away, we can plot how each new iteration changes our position:



And each bounce brings us closer to the cycle at ≈ 1.6. The cycle is, therefore, stable.

Discrete Dynamics

Here's another possible map. This one has two fixed points, one at $x = 0$ and one at $x = 2$. If we start at $x = 1.9$:



We see that the perturbation grows with each iteration, moving us away from $x = 2$ and towards $x = 0$. We conclude that $x = 2$ is an unstable cycle, and $x = 0$ is a stable cycle (or, maybe, stable fixed point, depending on the original system).

Discrete Stability

Given a fixed point x_0 of a map M, we can categorize the stability similarly to continuous systems. The condition is:

- If $|M'(x_0)| > 1$, then the magnitude of a small perturbation will grow each time by a factor $\approx |M'(x_0)|$, so it's unstable.
- If $|M'(x_0)| < 1$, then the magnitude of a small perturbation will shrink each time by a factor $\approx |M'(x_0)|$, so it's stable.
- and again, the intermediate case $|M'(x_0)| = 1$ is indeterminate, and can be stable, unstable, or half-stable.

Discrete Stability

Consider the following map:

$$M(x) = 3.3(x - x^2)$$

defined on the interval $[0, 1]$. It's easy to check that the only two fixed points are $x = 0$ and $x \approx 0.69697$, and that it maps the interval $[0, 1]$ to itself (it never produces negative values or values larger than 1). But:

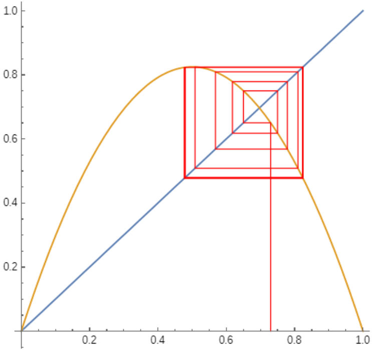
$$|M'(0)| = 3.3(1 - 2 \cdot 0) = 3.3 \implies \text{unstable}$$

$$|M'(0.69697)| = 3.3(1 - 2 \cdot 0.69697) = -1.3 \implies \text{unstable}$$

So this is a map with only two unstable points, no stable fixed points, that maps a fixed interval to itself! This can never happen in a one-variable *continuous* system, where either we fly off to infinity (and there's no fixed interval), or have a stable fixed point.

Discrete Stability

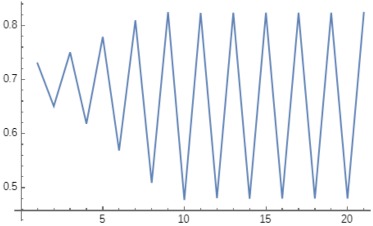
Let's plot it:



This is from an initial point of 0.73, for 500 iterations. As predicted, we move away from the unstable fixed point at 0.69697. But then it approaches a loop, alternating between two points.

Periodic Behavior

We can also plot the behavior over time in a more traditional form:



Yup, it's alternating. Why?

Periodic Behavior

Well, we can look at the *second iterate* map:

$$M_2(x) = M(M(x)) = 3.3(M(x) - M(x)^2)$$

$$= 10.89x - 46.83x^2 + 71.87x^3 - 35.94x^4$$

Then we can (numerically) solve for the fixed points:

$$M_2(x) = x \implies x \in \{0, 0.6970, 0.4794, 0.8236\}$$

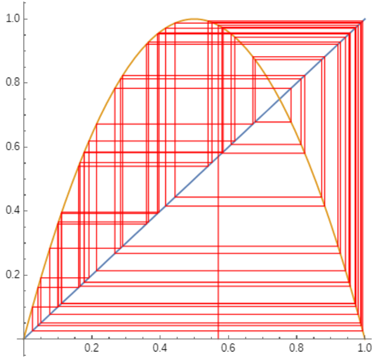
Of course we have the two fixed points from before, 0 and 0.6970, but we have two more fixed points of the second iterate – points that return to themselves after two mappings. And those two switch with each other, and (as one can check) are stable! It's a stable cycle.

Chaos

One more example. What happens if we instead use

$$M(x) = 4(x - x^2)$$

which also maps the interval $[0, 1]$ to itself.



This time, it never settles into any behavior at all!

Chaos

Actually as we adjust the parameter a in

$$M(x) = a(x - x^2)$$

we can get any length period: something that repeats every three, or four, or fifty three. There are also values that give chaos, where no iterate has any fixed points, and any "randomly chosen" initial condition will effectively cover the whole interval $[0, 1]$ as it jumps around. Of particular note is an extreme sensitivity to initial conditions: each iteration of the map *magnifies* small changes in the initial conditions, so that by the twelfth iteration a tiny change of 0.001 in the initial point completely changes the result.
Chaos is defined by this:

Periodic Inhomogeneous Systems

Before we depart the topic of Poincaré maps, it's worth mentioning the other major point of applicability. If we have a *time dependent* system, but that time dependence is *periodic*, then we can make a Poincaré map for the state of the system after each cycle. For instance, a simplified model of the climate will have changes over the course of a year, and we can't readily compare June to October because the seasons are different. But we can compare the climate in June 2021 to June 2022, and ask how the integrated dynamics of 1 year map the system forward. This is the type of map you will deal with in your homework.

Boundary Value Problems

Solving differential equations like

$$x'' = f(x, x', t), \quad x(0) = 5, \quad x(10) = 6$$

... so instead of having one fully determined point to go off of, we have partial information at the boundaries. (In the above example, we aren't given x' at the boundaries).

Boundary Value Problems

Solving differential equations like

$$x'' = f(x, x', t), \quad x(0) = 5, \quad x(10) = 6$$

... so instead of having one fully determined point to go off of, we have partial information at the boundaries. (In the above example, we aren't given x' at the boundaries). Two methods to solve:

- Shooting methods
- Iterative methods

Shooting methods

Want to solve

$$x'' = -\sin(x), \quad x(0) = 5, \quad x(10) = 6$$

Idea: we have $x(0)$ and $x(10)$. We don't know $x'(0)$. What if we just guess? Say, $x'(0) = 0.5$.

Shooting methods

Want to solve

$$x'' = -\sin(x), \quad x(0) = 5, \quad x(10) = 6$$

Idea: we have $x(0)$ and $x(10)$. We don't know $x'(0)$. What if we just guess? Say, $x'(0) = 0.5$. Then we use our IVP solver (like Predictor-Corrector) to solve the equation out to $x(10)$. Then we try to refine our guess to find the correct value for $x'(0)$.

Shooting methods

Want to solve

$$x'' = -\sin(x), \quad x(0) = 5, \quad x(10) = 6$$

Idea: we have $x(0)$ and $x(10)$. We don't know $x'(0)$. What if we just guess? Say, $x'(0) = 0.5$. Then we use our IVP solver (like Predictor-Corrector) to solve the equation out to $x(10)$. Then we try to refine our guess to find the correct value for $x'(0)$. Really we have a function $F(a)$ that solves the diffeq with $x'(0) = a$ and returns $x(10)$. We solve $F(a) = 6$ with any solver (like bisection method, secant method, Newton's method).

Shooting methods

In general, we can have several different pieces of data, pick any point, and guess the values there. For example with

$$x' = y - z, y'' = z + x - \frac{y'}{1+y}, z'' = -z - 0.1z' + 0.1x$$

$$x(-1) = 0,\ y(-1) = 1\ z(1) = 0,\ z'(2) = \pi$$

the full data is (x, y, y', z, z'). We could choose to go from $t = -1$ and guess (y', z, z'), or from $t = 1$ and guess (x, y, y', z'), or from $t = 2$ and guess (x, y, y', z, z'). We could also pick any other point and guess all five values.

Usually we just have two points and not many values to guess. Guessing more than one values means solving several simultaneous equations, so bisection doesn't work; need fancy methods like Newton's method or Simplex.

Iterative Methods

Iterative methods are built on guessing the solution as a whole, and then refining it to simultaneouly fit the differential equation and boundary conditions. For the example

$$x'' = -\sin(x) - 0.1x', \quad x(0) = 5, \quad x(10) = 6$$

I could discretize time into 100 points with gap Δt, and create arrays $x[100]$ and $x'[100]$. I initialize these to some guess (all zeros, say, or a linear interpolation between endpoints). Then I try to 'fix them up' to fit.

Iterative Methods

$$x'' = -\sin(x) - 0.1x', \quad x(0) = 5, \quad x(10) = 6$$

At each point the derivative should obey

$$x'[i] = \left.\frac{dx}{dt}\right|_{t=t_i} \approx \frac{x(t_{i+1}) - x(t_{i-1})}{2\,\Delta t} = \frac{x[i+1] - x[i-1]}{2\,\Delta t}$$

which is a simple finite difference rule for derivatives. The second derivative should obey

$$x'' = -\sin(x) - 0.1x'$$

$$x''(t_i) = \left.\frac{d_2 x}{dt^2}\right|_{t=t_i} \approx \frac{x(t_{i+1}) + x(t_{i-1}) - 2x(t_i)}{\Delta t^2} = \frac{x[i+1] + x[i-1] - 2x[i]}{\Delta t^2}$$

Iterative Methods

From

$$x'' = -\sin(x) - 0.1x'$$

$$x''(t_i) = \left.\frac{d_2 x}{dt^2}\right|_{t=t_i} \approx \frac{x(t_{i+1}) + x(t_{i-1}) - 2x(t_i)}{\Delta t^2} = \frac{x[i+1] + x[i-1] - 2x[i]}{\Delta t^2}$$

Rearranging,

$$x[i] = \frac{1}{2}\left(x[i+1] + x[i-1] - \Delta t^2(-\sin(x[i]) - 0.1x'[i])\right)$$

Which is an expression for $x[i]$. (Note that $x[i]$ also appears on the right, but, let's try not to worry about that...)

Iterative Methods

Two rules:

$$x'[i] = \left.\frac{dx}{dt}\right|_{t=t_i} \approx \frac{x(t_{i+1}) - x(t_{i-1})}{2\,\Delta t} = \frac{x[i+1] - x[i-1]}{2\,\Delta t}$$

$$x[i] = \frac{1}{2}\left(x[i+1] + x[i-1] - \Delta t^2(-\sin(x[i]) - 0.1x'[i])\right)$$

Repeatedly apply these at each point i, updating the x and x' arrays until they converge. Hopefully! Careful analysis is required to show that they do converge, and a lot of work goes into making them converge *quickly*. Accuracy improvements can be attained by:

1. Letting it converge longer
2. Increasing the number of points
3. Using better differentiation rules (the finite difference rules from week 2)

An important thing to note: sweep i up from 1 to N, then back down from N to 1, and repeat. Much better than repeating 1 to N over and over.