Stability of Differential Equations

JMM 2016 in Seattle

Roger Thelwell

James Madison University

January 9, 2016

Outline

The Butterfly

file://Users/faculty/Downloads/Lorenz.ogv.480p.webm

https://commons.wikimedia.org/wiki/File:Lorenz.ogv+

The Butterfly

file://Users/faculty/Downloads/Lorenz.ogv.480p.webm

https://commons.wikimedia.org/wiki/File:Lorenz.ogv+

How do we measure the sensitivity to perturbation in initial condition?

A general ODE system

Consider the ODE system

$$\dot{\mathbf{x}} = \mathbf{F}(\mathbf{x}) \qquad \text{with } \mathbf{x}(0) = \mathbf{x}_0.$$

and let

$$\phi(t; \mathbf{x}_0)$$

represent the flow of this system through the initial point $\mathbf{x}_0$.

A general ODE system

Consider the ODE system

$$\dot{\mathbf{x}} = \mathbf{F}(\mathbf{x}) \qquad \text{with } \mathbf{x}(0) = \mathbf{x}_0.$$

and let

$$\phi(t; \mathbf{x}_0)$$

represent the flow of this system through the initial point $\mathbf{x}_0$.
Then the ratio

$$\frac{||\phi(t; \mathbf{y}_0) - \phi(t; \mathbf{x}_0)||}{||\mathbf{y}_0 - \mathbf{x}_0||}$$

quantifies the rate at which trajectories spread.

Calculating sensitivity...

isn't so easy.

Calculating sensitivity...

isn't so easy. We want to understand the divergence of two nearby trajectories:

$$\phi(t; \mathbf{y}_0) - \phi(t; \mathbf{x}_0) \approx D_x\phi(t; \mathbf{x}_0)(\mathbf{y}_0 - \mathbf{x}_0),$$

but $D_x\phi(t; \mathbf{x}_0)$ is usually hard to compute!

Calculating sensitivity...

isn't so easy. We want to understand the divergence of two nearby trajectories:

$$\phi(t; \mathbf{y}_0) - \phi(t; \mathbf{x}_0) \approx D_x\phi(t; \mathbf{x}_0)(\mathbf{y}_0 - \mathbf{x}_0),$$

but $D_x\phi(t; \mathbf{x}_0)$ is usually hard to compute!

For any curve of initial conditions $\mathbf{x}_s$, define

$$\mathbf{v}(t) = \partial_{\mathbf{s}}\phi(t; \mathbf{x}_s)\Big|_{s=0},$$

then $\mathbf{v}(t)$ satisfies the first variation equation

$$\dot{\mathbf{v}} = D_x\mathbf{F}(\phi(t; \mathbf{x})\Big|_{\mathbf{x}_0}\mathbf{v}_0 \quad \text{with} \quad \mathbf{v}_0 = \partial_{\mathbf{s}}\mathbf{x}_s.$$

Calculating sensitivity...

isn't so easy. We want to understand the divergence of two nearby trajectories:

$$\phi(t; \mathbf{y}_0) - \phi(t; \mathbf{x}_0) \approx D_x\phi(t; \mathbf{x}_0)(\mathbf{y}_0 - \mathbf{x}_0),$$

but $D_x\phi(t; \mathbf{x}_0)$ is usually hard to compute!

For any curve of initial conditions $\mathbf{x}_s$, define

$$\mathbf{v}(t) = \partial_{\mathbf{s}}\phi(t; \mathbf{x}_s)\Big|_{s=0},$$

then $\mathbf{v}(t)$ satisfies the first variation equation

$$\dot{\mathbf{v}} = D_x\mathbf{F}(\phi(t; \mathbf{x})\Big|_{\mathbf{x}_0}\mathbf{v}_0 \quad \text{with} \quad \mathbf{v}_0 = \partial_{\mathbf{s}}\mathbf{x}_s.$$

It is $\mathbf{v}(t)$ that will ultimately give us our growth rate.

The Lyapunov exponent

The *Lyapunov* exponent, $\lambda(t)$, is the exponential growth rate measuring sensitivity to initial conditions. It is classically computed as:

$$||\mathbf{v}(t)|| \approx exp(\lambda \cdot t)||\mathbf{v_0}||,$$

But what happens if we use power series approach?

Ex 1: $y' = \alpha(t)y$

With MAPLE

```
>> restart:
>> Order := 4:
>> alpha := t -> sum(a[k]*t^k,k=0..Order):
>> GROWTH := diff(y(t),t) = alpha(t)*y(t):
>> Yseries := dsolve({GROWTH,y(0)=y[0]},y(t),type='series');
```

$$y(t) = y_0 + a_0 y_0 t + \left(1/2\, a_0{}^2 y_0 + 1/2\, a_1 y_0\right) t^2 + \left(1/6\, a_0{}^3 y_0 + 1/2\, a_1 a_0 y_0 + 1/3\, a_2 y_0\right) t^3 + O\left(t^4\right)$$

Ex 1: $y' = \alpha(t)y$

With MAPLE

```
>> restart:
>> Order := 4:
>> alpha := t -> sum(a[k]*t^k,k=0..Order):
>> GROWTH := diff(y(t),t) = alpha(t)*y(t):
>> Yseries := dsolve({GROWTH,y(0)=y[0]},y(t),type='series');
```

$$y(t) = y_0 + a_0\, y_0 t + \left(1/2\, a_0{}^2\, y_0 + 1/2\, a_1\, y_0\right) t^2 + \left(1/6\, a_0{}^3\, y_0 + 1/2\, a_1\, a_0\, y_0 + 1/3\, a_2\, y_0\right) t^3 + O\left(t^4\right)$$

which we can check

```
>> SOLN1 := y[0] * exp(int(alpha(tau),tau=0..t));
>> check := taylor(SOLN1,t=0) - Yseries;
```

Why?

From

$$y\left(t\right) = y_0 + a_0\, y_0 t + \left(1/2\, {a_0}^2\, y_0 + 1/2\, a_1\, y_0\right) t^2 + \left(1/6\, {a_0}^3\, y_0 + 1/2\, a_1\, a_0\, y_0 + 1/3\, a_2\, y_0\right) t^3 + O\left(t^4\right)$$

we can find

$$\partial_{y_0} y(t) = 1 + a_0 t + \left(1/2\, {a_0}^2 + 1/2\, a_1\right) t^2 + \left(1/6\, {a_0}^3 + 1/2\, a_1\, a_0 + 1/3\, a_2\right) t^3 + O\left(t^4\right)$$

Sensitivity to initial conditions!

Why?

From

$$y\left(t\right) = y_0 + a_0\,y_0 t + \left(1/2\,{a_0}^2\,y_0 + 1/2\,a_1\,y_0\right)t^2 + \\ \left(1/6\,{a_0}^3\,y_0 + 1/2\,a_1\,a_0\,y_0 + 1/3\,a_2\,y_0\right)t^3 + O\left(t^4\right)$$

we can find

$$\partial_{y_0} y(t) = 1 + a_0 t + \left(1/2\,{a_0}^2 + 1/2\,a_1\right)t^2 + \\ \left(1/6\,{a_0}^3 + 1/2\,a_1\,a_0 + 1/3\,a_2\right)t^3 + O\left(t^4\right)$$

Sensitivity to initial conditions! Which we can easily compute...

```
>> Yy0 := taylor(diff(Yseries,y_0),t=0);
```

The Lyapunov exponent

We have

$$\partial_{y_0} y\,(t) = \sum_{n=0}^{\infty} f_n(y_0, \ldots, y_{n-1}) t^n$$

and so $\lambda(t)$ is easy to compute:

```
>> simplify(taylor(ln(Yy0))/t);
```

$$\lambda(t) = a_0 + 1/2 a_1 t + 1/3 a_2 t^2 + O(t^3)$$

For our problem, a direct calculation verifies this:

$$\frac{1}{t}\int_0^t \alpha(\tau) d\tau).$$

This time average is the mean coefficient on $[0, t]$

Ex: A 2D planar system

Consider the system:

$$\begin{aligned} x' &= -y + x(1 - x^2 - y^2) \\ y' &= \quad x + y(1 - x^2 - y^2) \end{aligned}$$

which can be recast (and decoupled!) as

$$\begin{aligned} r' &= r(1 - r^2) \\ \theta' &= 1 \end{aligned}$$

with a stable orbit at $r = 1$.

Ex: A 2D planar system: exponents?

With $\dot{r} = r(1 - r^2)$ and $\dot{\theta} = 1$, the variational problem is simple...

$$\dot{\mathbf{v}} = \begin{bmatrix} 1 - 3r^2 & 0 \\ 0 & 0 \end{bmatrix}_{r=1} \mathbf{v_0}$$

Remember that it's **v** that we need to find, and it provides an (approximate) basis of $D_x \phi$.

Ex: A 2D planar system: exponents?

With $\dot{r} = r(1 - r^2)$ and $\dot{\theta} = 1$, the variational problem is simple...

$$\dot{\mathbf{v}} = \begin{bmatrix} 1 - 3r^2 & 0 \\ 0 & 0 \end{bmatrix}_{r=1} \mathbf{v_0}$$

Remember that it's $\mathbf{v}$ that we need to find, and it provides an (approximate) basis of $D_x\phi$.

So $v_r = \exp(-2t)v_0$, and the lyapunov exponent is -2 in the radial direction (and 0 in the tangential direction).

Ex: A 2D planar system: series

We can use power series to explore the original cartesian system.

```
>> restart:
>> Order := 2:
>> F1 := (x,y) -> -y+x*(1-x^2-y^2);
>> F2 := (x,y) -> x+y*(1-x^2-y^2);
>> Planar := [diff(x(t),t) = F1(x(t),y(t)) , ...
               diff(y(t),t) = F2(x(t),y(t))];
>> ART := dsolve(Planar,[x(t),y(t)],type = 'series);
>> BOB := convert(subs({x(0)=x,y(0)=y},ART),polynom);
>> JacP := {x,y} -> Jacobian(BOB,[x,y]);
>> SingularValues(JacP(1,0));
```

Ex: A 2D planar system: series

We were able to compute singular values as a time series,

```
>> check := SingularValues(JacP(1,0));
>> plot( ln(check[1](t))/t , t=0..0.0001);
```

$$exp(\lambda_1 t) \approx \sqrt{3 * t^2 + 1 - 2 * t + 2 * t * \sqrt{2 * t^2 - 2 * t + 1}}$$
$$exp(\lambda_2 t) \approx \sqrt{3 * t^2 + 1 - 2 * t - 2 * t * \sqrt{2 * t^2 - 2 * t + 1}}$$

Ex: A 2D planar system: series

We were able to compute singular values as a time series,

```
>> check := SingularValues(JacP(1,0));
>> plot( ln(check[1](t))/t , t=0..0.0001);
```

$$exp(\lambda_1 t) \approx \sqrt{3*t^2+1-2*t+2*t*\sqrt{2*t^2-2*t+1}}$$
$$exp(\lambda_2 t) \approx \sqrt{3*t^2+1-2*t-2*t*\sqrt{2*t^2-2*t+1}}$$

or evaluate the Jacobian at a specific time and then compute the singular values. Either way:
at `t = 1e-7`, $\lambda_1 \approx$ `0.00000005` and $\lambda_2 \approx$ `-2.000000015`

3D: Lorenz

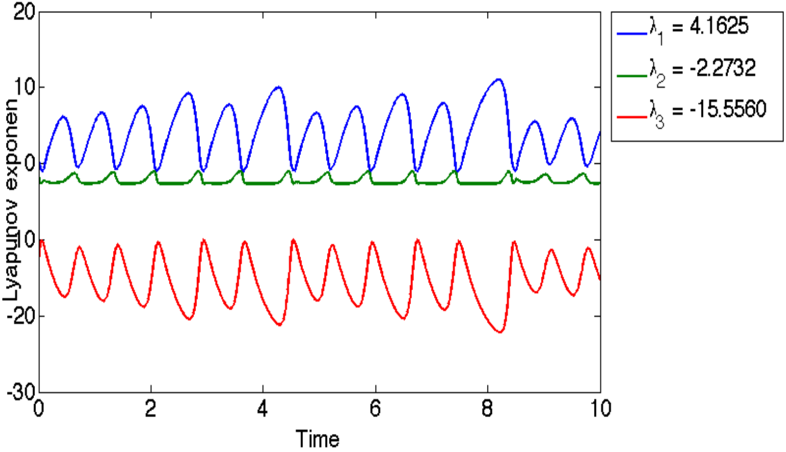
And now for the real work...

3D: Lorenz

With MAPLE and MATLAB

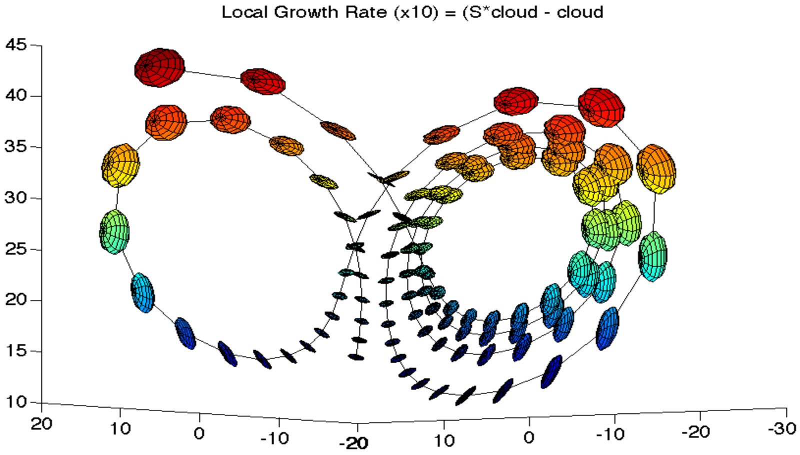
Ex: Lorenz

Uhoh....



Sprott gets: $0.966, 0, -14.6$. At least our trace, 13.6667, is right.

Ex: Lorenz



Local Growth Rate (x10) = (S*cloud - cloud
45
40
35
30
25
20
15
10
20
10
0
-10
-20
10
0
-10
-20
-30

The PROCESS

Generate a polynomial (quadratic) system.(Reverse Polish)

Compute analytic series approximation $Y(t; ic)$

Differentiate to find sensitivity to IC $S(t; ic) := \partial_{ic} Y(t; ic)$

Build IC series $y_0(T)$ 'on' attractor.

Choose tau for local continuation length

20 WHILE T$<$ Tmax

Evaluate $D = svd(S(tau; y_0(T))$

Compute $\lambda(T)$ (or mean) for local time interval

Advance: $T = T + r$

GOTO 20

Questions?

The community typically evolves trajectory and linearized variational problem for many time steps to accumulate growth, then re-orthogonalizes.

Instead of answering questions, we now have many more:

Questions?

The community typically evolves trajectory and linearized variational problem for many time steps to accumulate growth, then re-orthogonalizes.

Instead of answering questions, we now have many more:

A direct TNB frame calculation seems clunky, and the resulting power series computation messy.

Is there a trick we can play to rotate our local SVD frame onto the physical frame? It seems so, but ...

Questions?

The community typically evolves trajectory and linearized variational problem for many time steps to accumulate growth, then re-orthogonalizes.

Instead of answering questions, we now have many more:

A direct TNB frame calculation seems clunky, and the resulting power series computation messy.

Is there a trick we can play to rotate our local SVD frame onto the physical frame? It seems so, but ...

Another series approach might be to compute

$$||\phi(t;\mathbf{x}_0) - \phi(t;\mathbf{y}_0)||$$

directly using the analytic (approximate) integrator.

Conclusions

- Easy to find approximate solution operator as a function of IV.
- Seems to provide information about local growth of variations.
- No need to evolve tangent space (a la Wolf)?
- Non-autonomous? No fear!
- Non-linear? No problem!

Thanks!

Questions?

Thanks!

Questions?
(I sure have a lot!)

thelwerj@jmu.edu