Love, War and Zombies - Systems of Differential Equations using Sage

D. Joyner

Department of Mathematics
US Naval Academy
Annapolis, MD 21402

Solving systems of differential equations using Sage

Project Mosaic
M-cast

2011-04-22

Outline

Systems of differential equations can be used to mathematically model the weather, electrical networks, spread of infectious diseases, conventional battles, populations of competing species, and, yes, zombie attacks.

This talk looks at some of these models from the computational perspective using the mathematical software Sage (www.sagemath.org).

Outline

Systems of differential equations can be used to mathematically model the weather, electrical networks, spread of infectious diseases, conventional battles, populations of competing species, and, yes, zombie attacks.

This talk looks at some of these models from the computational perspective using the mathematical software Sage (www.sagemath.org).

Outline

Systems of differential equations can be used to mathematically model the weather, electrical networks, spread of infectious diseases, conventional battles, populations of competing species, and, yes, zombie attacks.

This talk looks at some of these models from the computational perspective using the mathematical software Sage (www.sagemath.org).

Outline

Systems of differential equations can be used to mathematically model the weather, electrical networks, spread of infectious diseases, conventional battles, populations of competing species, and, yes, zombie attacks.

This talk looks at some of these models from the computational perspective using the mathematical software Sage (www.sagemath.org).

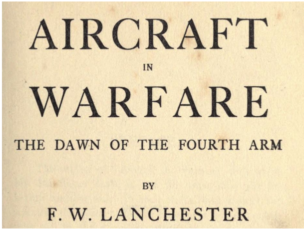
Biography.

The British engineer Frederick William Lanchester was a pioneer automobile manufacturer who invented an early carburetor and manufactured several different types of gasoline engines in the early 1900's (for boats - he built the first British motorboat - cars and airplanes).

Figure: F. W. Lanchester (1868-1946) and the 1935 "Lanchester Ten"

Biography.

During World War I, Lanchester served in the Royal Air Force and discovered a systems of differential equations which predicted the outcome of aerial battles, the so-called **Lanchester Power Laws**. They were published in his book book entitled **Aircraft in Warfare: the Dawn of the Fourth Arm**.



AIRCRAFT
IN
WARFARE
THE DAWN OF THE FOURTH ARM
BY
F. W. LANCHESTER

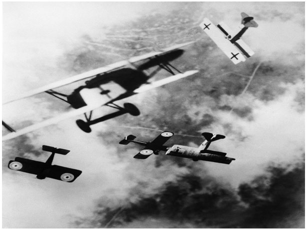


Figure: Lanchester's book Aircraft in Warfare, 1916

The Lanchester model.

Assume

- two armies fight, with $x(t)$ troops on one side (the "X-men") and $y(t)$ on the other (the "Y-men"), and
- the rate at which soldiers in one army are put out of action is proportional to the troop strength of their enemy.

Such a battle is sometimes called **direct fire**.

Examples of such fights include "Cowboys and Indians" hand-to-hand battles, tank battles, and open sea ship battles.

The Lanchester model.

These assumptions give rise to the system of differential equations (the "Lanchester power laws")

$$\begin{cases} x'(t) = -Ay(t), & x(0) = x_0, \\ y'(t) = -Bx(t), & y(0) = y_0, \end{cases}$$

where

- $A > 0$ and $B > 0$ (called their **fighting effectiveness coefficients**) are constants, and
- x_0 and y_0 are the **initial troop strengths**.

(*Note*: Guerrilla warfare, where the enemy could be hiding in a building, is **not** covered by this "square law" model.)

The Lanchester square law.

Lanchester's approach was to solve the separable DE

$$\frac{dy}{dx} = \frac{dy/dt}{dx/dt} = \frac{Bx}{Ay},$$

using separation of variables. This solution can be easily derived:

$$Ay \cdot dy = Bx \cdot dx \implies Ay^2/2 = Bx^2/2 + \mathrm{constant}.$$

The Lanchester square law.

If we define

- Bx^2 = **fighting strength** of X-men,
- Ay^2 = **fighting strength** of Y-men,

then **Lanchester's square law**,

$$Ay^2 - Bx^2 = C,$$

where C is a constant, says that the **relative fighting strength of a "direct fire" battle is constant**.

Example: The Battle of Trafalgar.

In 1805, **twenty-seven** British ships, led by Admiral Nelson, defeated **thirty-three** French and Spanish ships, under French Admiral Pierre-Charles Villeneuve.

British fleet lost: **zero**,
Franco-Spanish fleet lost: **twenty-two** ships.

Figure: The Battle of Trafalgar, by William Clarkson Stanfield (1836)

The Battle of Trafalgar.

Admiral Nelson was shot by a French marksman during battle. As he died, his last words were **'God and my country.'**

Figure: The Fall of Nelson, by Denis Dighton (1825)

The Battle of Trafalgar.

The battle is modeled by the following **system of differential equations**:

$$\begin{cases} x'(t) = -y(t), & x(0) = 27, \\ y'(t) = -25x(t), & y(0) = 33. \end{cases}$$

How can Sage be used to solve this?

The Battle of Trafalgar.

One way is by the **method of eigenvalues and eigenvectors**: if the eigenvalues $\lambda_1 \neq \lambda_2$ of $A = \begin{pmatrix} 0 & -1 \\ -25 & 0 \end{pmatrix}$ are **distinct** then the general solution can be written

$$\begin{pmatrix} x(t) \\ y(t) \end{pmatrix} = c_1 \vec{v_1} e^{\lambda_1 t} + c_2 \vec{v_2} e^{\lambda_2 t},$$

where $\vec{v_1}, \vec{v_2}$ are eigenvectors of A.

The Battle of Trafalgar.

Solving

$$A\vec{v} = \lambda\vec{v}$$

gives us the eigenvalues and eigenvectors. Using Sage, this is easy:

```
sage: A = matrix([[0, -1], [-25, 0]])
sage: A.eigenspaces_right()
[
(5, Vector space of degree 2 and dimension 1 over Rational Field
User basis matrix:
[ 1 -5]),
(-5, Vector space of degree 2 and dimension 1 over Rational Field
User basis matrix:
[1 5])
]
```

The Battle of Trafalgar.

Therefore,

$$\lambda_1 = 5, \quad \vec{v_1} = \begin{pmatrix} 1 \\ -5 \end{pmatrix},$$

$$\lambda_2 = -5, \quad \vec{v_2} = \begin{pmatrix} 1 \\ 5 \end{pmatrix},$$

so

$$\begin{pmatrix} x(t) \\ y(t) \end{pmatrix} = c_1 \begin{pmatrix} 1 \\ -5 \end{pmatrix} e^{5t} + c_2 \begin{pmatrix} 1 \\ 5 \end{pmatrix} e^{-5t}.$$

We must solve for c_1, c_2.

The Battle of Trafalgar.

The initial conditions give

$$\begin{pmatrix} 27 \\ 33 \end{pmatrix} = c_1 \begin{pmatrix} 1 \\ -5 \end{pmatrix} + c_2 \begin{pmatrix} 1 \\ 5 \end{pmatrix} = \begin{pmatrix} c_1 + c_2 \\ -5c_1 + 5c_2 \end{pmatrix}$$

These equations for c_1, c_2 can be solved using Sage:

```
sage: c1,c2 = var("c1,c2")
sage: solve([c1+c2==27, -5*c1+5*c2==33],[c1,c2])
[[c1 == (51/5), c2 == (84/5)]]
```

imply $c_1 = 51/5$ and $c_2 = 84/5$.

The Battle of Trafalgar.

This gives the solution to the system of DEs as

$$x(t) = \frac{51}{5}e^{5t} + \frac{84}{5}e^{-5t}, \quad y(t) = -51e^{5t} + 84e^{-5t}.$$

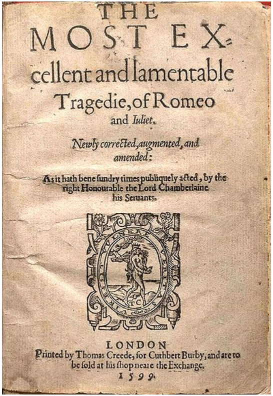
The solution satisfies

$$x(0.033) = 26.27... \quad (\text{“}\mathbf{0}\text{ losses”}),$$
$$y(0.033) = 11.07... \quad (\text{“}\mathbf{22}\text{ losses”}),$$

consistent with the losses in the actual battle.

Romeo and Juliet

From **war and death**, we turn to **love and romance**.



THE
MOST EX-
cellent and lamentable
Tragedie, of Romeo
and Iuliet.

Newly corrected, augmented, and amended:

As it hath bene sundry times publiquely acted, by the right Honourable the Lord Chamberlaine his Seruants.

LONDON
Printed by Thomas Creede, for Cuthbert Burby, and are to be sold at his shop neare the Exchange.
1599.

Title: **Romeo and Juliet: Act One, Scene Five**.
Engraver: Facius, Georg Sigmund Facius, Johann Gottlieb
Designer: Miller, William

Date: 1789

Romeo and Juliet

Romeo:

If I profane with my unworthiest hand

This holy shrine, the gentle sin is this:

My lips, two blushing pilgrims, ready stand

To smooth that rough touch with a tender kiss.

Rome and Juliet, by Frank Bernard Dicksee (1884)

Romeo and Juliet

Juliet:

Good pilgrim, you do wrong your hand too much,

Which mannerly devotion shows in this;

For saints have hands that pilgrims' hands do touch,

And palm to palm is holy palmers' kiss.

- **Romeo and Juliet**, Act I, Scene V

Romeo and Juliet

Romeo is madly in love with Juliet. The more she loves him, the more he loves her.

Juliet's emotions are more complicated. Her love for Romeo makes her feel good about herself, which makes her love him even more. However, if Romeo's love for her is too much, she reacts negatively.

Can we model **romance** using differential equations??

Romeo and Juliet

Let

- $r = r(t)$ denote the love Romeo has for Juliet at time t,
- $j = j(t)$ denote the love Juliet has for Romeo at time t.

The **Romeo and Juliet equations** are

$$\begin{cases} r' &= & Aj, & r(0) = r_0, \\ j' &= & -Br + Cj, & j(0) = j_0, \end{cases}$$

where $A > 0$, $B > 0$, $C > 0$, r_0, j_0 are given constants.

Romeo and Juliet

A few examples help display the type of behavior. We will solve

$$\begin{cases} r' = 5j, & r(0) = 4, \\ j' = -r + 2j, & j(0) = 6, \end{cases}$$

with the help of Sage. We will use the method of **Laplace transforms**,

$$f(t) \longmapsto F(s) = \mathcal{L}[f(t)](s) = \int_0^\infty f(t)e^{-st}\,dt.$$

Basically: take LTs of both sides, solve, then take inverse LTs.

Romeo and Juliet

Let

$$R(s) = \mathcal{L}[r(t)](s), \quad J(s) = \mathcal{L}[j(t)](s).$$

Taking Laplace Transforms gives

$$sR(s) - r(0) = 5J(s), \quad sJ(s) - j(0) = -R(s) + 2J(s).$$

To **solve this**, you can compute the row-reduced echelon form of the corresponding augmented matrix

$$A = \begin{pmatrix} s & -5 & 4 \\ 1 & s-2 & 6 \end{pmatrix}.$$

Romeo and Juliet

Solving the system

```
sage: s,t = var("s,t")
sage: A = matrix(SR, [[s,-5,4],[1,s-2,6]])
sage: B = A.echelon_form()
sage: Rs = B[0][2]; Js = B[1][2]
sage: rt = Rs.inverse_laplace(s,t); rt
(13*sin(2*t) + 4*cos(2*t))*e^t
sage: jt = Js.inverse_laplace(s,t); jt
(sin(2*t) + 6*cos(2*t))*e^t
sage: parametric_plot((rt,jt),(t,0,1.5))
```

and plotting the solution:

Romeo and Juliet

The `parametric_plot((rt,jt), (t,0,1.5))` command results in the following plot, of $\{(r(t), j(t)) \mid t \in \mathbb{R}\}$:

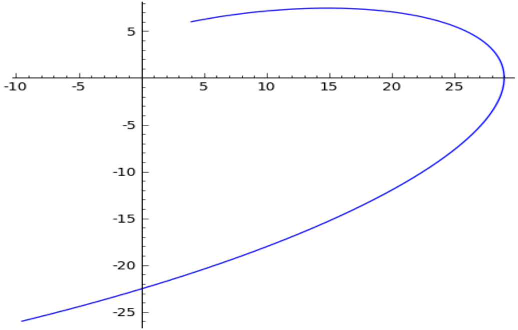


Figure: Romeo and Juliet plots.

Romeo and Juliet

The plot shows Juliet reacts **negatively** to Romeo's increasing love for her.

Let us try some **new coefficients** in an effort to better balance their emotions.

As **another example**, we use Sage to compute the parametric plot of the solution to

$$\begin{cases} r' = 5j, & r(0) = 4, \\ j' = -r + j/5, & j(0) = 6. \end{cases}$$

Romeo and Juliet

Using Sage's `desolve_system` command, we can solve this and plot the solution.

```
sage: t = var("t")
sage: r = function("r",t)
sage: j = function("j",t)
sage: de1 = diff(r,t) == 5*j
sage: de2 = diff(j,t) == -r+(1/5)*j
sage: soln = desolve_system([de1, de2], [r,j], ics=[0,4,6])
sage: rt = soln[0].rhs(); jt = soln[1].rhs()
sage:  parametric_plot((rt,jt),(t,0,5))
```

Romeo and Juliet

Now, the plot indicates less extreme behavior for both Romeo and Juliet:

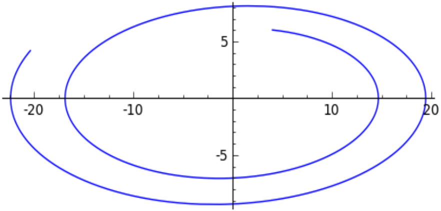


Figure: New Romeo and Juliet plot.

Can you find "better" coefficients?

Zombies

All this talk of romance is fine and good. **However**, what do you do if there is a **Zombie attack**?

Figure: Zombies in George Romero's 1968 **Night of the Living Dead**

Zombies

Fortunately for us all, a 2009 paper called "**When zombies attack! Mathematical modelling of an outbreak of zombie infection**" helps us solve this problem as well!

Let

- S represents **people** (the "susceptibles"),
- Z is the number of **zombies**, and
- R (the "removed") represents (a) **deceased zombies**, (b) **bitten people** (who are sometimes turned into zombies), or (c) **dead people**.

Zombies

The simplest system of ODEs developed in that paper is:

$$
\begin{aligned}
S' &= B - \beta SZ - \delta S, \\
Z' &= \beta SZ + \zeta R - \alpha SZ, \\
R' &= \delta S + \alpha SZ - \zeta R.
\end{aligned}
$$

Zombies

What do these terms mean and how do they arise?

We **assume** that

- people (counted by S) have a **constant birth rate** B.
- a proportion δ of people **die a non-zombie related death**. (This accounts for the $-\delta S$ term in the S' line and the $+\delta S$ term in the R' line.)
- a proportion ζ of dead humans (counted by R) can **resurrect** and become a zombie.
  (This accounts for the $+\zeta R$ term in the Z' line and the $-\zeta R$ term in the R' line.)

Zombies

There are "non-linear" terms as well. These correspond to when a **person interacts with a zombie** - the "SZ terms." We **assume** that

- some of these interactions results in a **person killing a zombie** by destroying its brain
  (This accounts for the $-\alpha SZ$ term in the Z' line and the $+\alpha SZ$ term in the R' line.)
- some of these interactions results in a **zombie infecting a person** and turning that person into a zombie
  (This accounts for the $-\beta SZ$ term in the S' line and the $+\beta SZ$ term in the Z' line.)

Zombies attack example

These non-linear terms mean that the system is **too complicated** to solve by simple methods, such as the method of eigenvalues or Laplace transforms.

The method of eigenvalues or Laplace transforms only work for **linear** systems, such as (in the 2×2 case)

$$\begin{cases} x'(t) = ax + by + f(t), & x(0) = x_0, \\ y'(t) = cx + dy + g(t), & y(0) = y_0. \end{cases}$$

Zombie attack example

However, solutions to the Zombies Attack model can be **numerically approximated** in Sage.

```
sage: from sage.calculus.desolvers import desolve_system_rk4
sage: t,s,z,r = var('t,s,z,r')
sage: a,b,zeta,d,B = 0.005,0.0095,0.0001,0.0001,0.0
sage: P = desolve_system_rk4([B-b*s*z-d*s,b*s*z-zeta*r-a*s*z,d*s+a*s*z-zeta*r],
 [s,z,r],ics=[0,11,10,5] ,ivar=t,end_points=30)
sage: Ps = list_plot([[t,s] for t,s,z,r in P],plotjoined=True,
 legend_label='People')
sage: Pz = list_plot([[t,z] for t,s,z,r in P],plotjoined=True,rgbcolor='red',
 legend_label='Zombies')
sage: Pr = list_plot([[t,r] for t,s,z,r in P],plotjoined=True,rgbcolor='black',
 legend_label='Deceased')
sage: show(Ps+Pz+Pr)
```

Zombies attack example

In other words, take

$$\alpha = 0.005, \quad \beta = 0.0095, \quad \zeta = 0.0001,$$
$$\delta = 0.0001, \quad B = 0,$$

and solve the above system numerically.

What do we get?

Zombies attack example

The plot of the solutions is given below. **The Zombies win!**

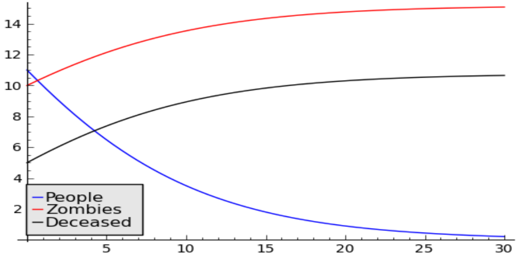


Figure: Suseptibles, Zombies and Removed plot

Not good.

Zombies attack example

Too many zombies are **infecting people** (this is the constant β, which is relatively large).

Let's make β smaller

$$\alpha = 0.005, \ \ \beta = 0.004, \ \ \zeta = 0.0001,$$
$$\delta = 0.0001, \ \ B = 0,$$

and solve the above system numerically. Now, what do we get?

Zombie attack example

Plug the new parameters into Sage:

```
sage: from sage.calculus.desolvers import desolve_system_rk4
sage: t,s,z,r = var('t,s,z,r')
sage: a,b,zeta,d,B = 0.005,0.004,0.0001,0.0001,0.0
sage: P = desolve_system_rk4([B-b*s*z-d*s,b*s*z-zeta*r-a*s*z,d*s+a*s*z-zeta*r],
 [s,z,r],ics=[0,11,10,5] ,ivar=t,end_points=30)
sage: Ps = list_plot([[t,s] for t,s,z,r in P],plotjoined=True,
 legend_label='People')
sage: Pz = list_plot([[t,z] for t,s,z,r in P],plotjoined=True,rgbcolor='red',
 legend_label='Zombies')
sage: Pr = list_plot([[t,r] for t,s,z,r in P],plotjoined=True,rgbcolor='black',
 legend_label='Deceased')
sage: show(Ps+Pz+Pr)
```

Zombies attack example

The plot of the new solution is given below, but **the Zombies still win!**

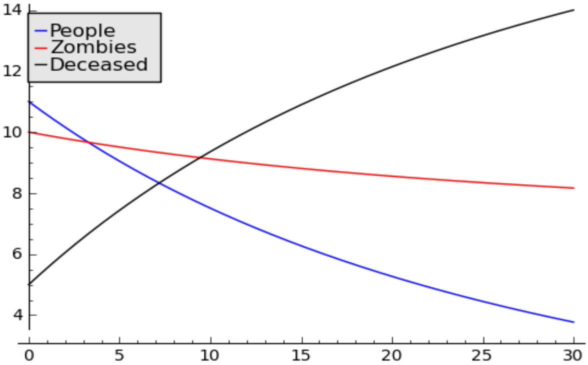


Figure: Another Suseptibles, Zombies and Removed plot

Zombies attack example

Still too many zombies are infecting people!

Let's make β even smaller

$$\alpha = 0.005, \quad \beta = 0.002, \quad \zeta = 0.0001,$$
$$\delta = 0.0001, \quad B = 0,$$

and solve the above system numerically. Now, what do we get?

Zombies attack example

Using these new parameters, the plot of the solutions is given below. Finally, **the Zombies lose** but the people are dying off as well.

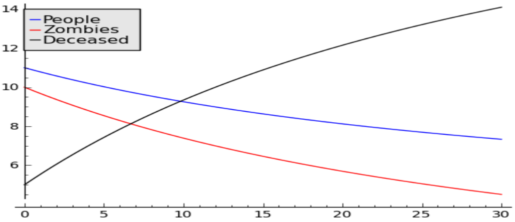


Figure: Yet another Suseptibles, Zombies and Removed plot

This is **bad**.

Zombies attack example

Is the reason why the people die off because they have no birth rate (i.e., because $B = 0$)?

We have a battle *a la* **Lanchester** (people vs. zombies).

Maybe, to conquer the zombies, we need **more love** (*a la* **Romeo and Juliet**)??

Zombie attack example

New B parameter value, $B = 2$ (higher birth rate):

```
sage: from sage.calculus.desolvers import desolve_system_rk4
sage: t,s,z,r = var('t,s,z,r')
sage: a,b,zeta,d,B = 0.005,0.002,0.0001,0.0001,2.0
sage: P = desolve_system_rk4([B-b*s*z-d*s,b*s*z-zeta*r-a*s*z,d*s+a*s*z-zeta*r],
 [s,z,r],ics=[0,11,10,5] ,ivar=t,end_points=30)
sage: Ps = list_plot([[t,s] for t,s,z,r in P],plotjoined=True,
 legend_label='People')
sage: Pz = list_plot([[t,z] for t,s,z,r in P],plotjoined=True,rgbcolor='red',
 legend_label='Zombies')
sage: Pr = list_plot([[t,r] for t,s,z,r in P],plotjoined=True,rgbcolor='black',
 legend_label='Deceased')
sage: show(Ps+Pz+Pr)
```

Zombies attack example

The plot of the solutions is given below. Finally, **the Zombies die off** and the people are **surviving**!

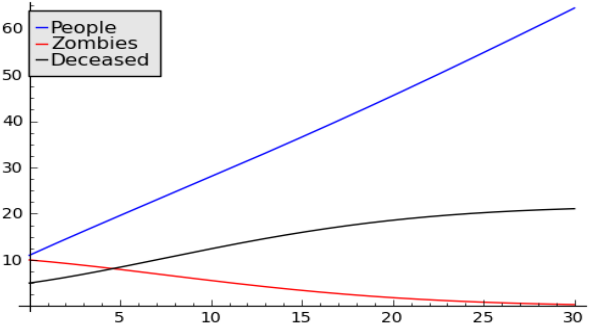


Figure: Suseptibles, Zombies and Removed plot with $B = 2$

Zombies attack example

The moral of the story: to survive, you must **both attack and protect yourself from zombies**!

Figure: Zombies!

- photo by sookie http://www.flickr.com/photos/sookie/1490740447/

The end

$\boxed{\text{The End}}$

Get Sage for free at
http://www.sagemath.org