Geographic models

Torbjørn Seland

Nov 15, 2014

Contents

1 Introduction

This chapter will introduce a new model for epidemic diseases. By expand the ODE system from previous chapter to consist of a geographic spread of epidemics. The first section *Simple system for spatial spread* will build on the simple SIR model presented in previous chapter and be based on Murray [1]. A couple of tests will be done on the model to verify that the implementation is done correct. The parameters from *Epidemic from the English School* will be used in this model and the result will be compared against the ODE system. The position of the infected student will be compared against the number of infected each day. The last section *Zombiefication* will compare and develop the zombie model presented in the previous chapter.

2 Simple system for spatial spread

A spatial variable, $\mathbf{x}$ will now be introduced to the new model. This means that the position has an effect on the change in each class. The system that will be shown here, will be based on the simple ODE system presented in the previous chapter. The difference will be the diffusion part added to each equation. The system can be seen in eq(1).

$$\begin{aligned}
\frac{\partial S}{\partial t} &= -rIS + D\nabla^2 S \\
\frac{\partial I}{\partial t} &= rIS - aI + D\nabla^2 I \\
\frac{\partial R}{\partial t} &= aI + D\nabla^2 R
\end{aligned} \tag{1}$$

All three classes have the same diffusion coefficient, D. This give the three groups the same diffusion speed. Often will this differ between groups and will be study later in *Zombiefication.*The two parts rIS and aI will work in the same way as in the ODE system. Since this model taking the position into account, the idea is to model a group of infective that moves into a uniform population with susceptible, which is spread around with the density S_0. Then the geotemporal spread can be seen. The problem will first be consider as one-dimensional. The system can be nondimensionalise by writing

$$\begin{aligned}
I^* = \frac{I}{S_0}, \quad I^* &= \frac{I}{S_0}, \qquad R^* = \frac{R}{S_0}, \\
x^* = \left(\frac{rS_0}{D}\right)^{1/2} x, \quad t^* &= rS_0 t, \qquad \lambda = \frac{a}{rS_0},
\end{aligned} \tag{2}$$

S_0 is used as a representative population. Now the model(1) can be expressed as in eq(3). The asterisks have been dropped to make it easier to read.

$$\begin{aligned}
\frac{\partial S}{\partial t} &= -IS + \frac{\partial^2 S}{\partial x^2}, \\
\frac{\partial I}{\partial t} &= IS - \lambda I + \frac{\partial^2 I}{\partial x^2}, \\
\frac{\partial R}{\partial t} &= \lambda I + \frac{\partial^2 R}{\partial x^2},
\end{aligned} \tag{3}$$

The three parameters r, a and D have now been replaced by λ. The *reproduction rate* that was presented for the ODE model can be seen by $1/\lambda$. This has a couple of equivalent meanings. $1/\lambda$ can be seen as the number of secondary infections produced by one primary infected. It can also be used to measure two different time scales. The first one, $1/(rS_0)$, measure the contagious time of the disease. The second one can look at the life expectancy for an infective. This can be described $1/a$.

2.1 Travelling wave 1D

The problem that this section will focus on, is the travelling wave. The travelling wave describe how a disease travel through geographic area of humans. This will be shown here by sending a pulse of infected into a group of susceptible. A travelling wave solution will be described as follows,

$$I(x,t) = I(z), \quad S(x,t) = S(z), \quad R(x,t) = R(z), \quad z = x - ct, \tag{4}$$

Here the value c will be the wave speed. This represents a wave of constant shape that travels in the positive x-direction. These can now be inserted into (3). This gives the ordinary system(5)

$$\begin{aligned} S'' + cS' - IS = 0, \\ I'' + cI' + I(S - \lambda) = 0 \\ R'' + cR + I\lambda = 0 \end{aligned} \tag{5}$$

This gives an eigenvalue problem to find the range of λ so that c always will be positive. The values of S, I and R also has to stay nonnegative. This leads to

$$\begin{aligned} 0 \leq S(-\infty) < S(\infty) = 1 \\ I(-\infty) = I(\infty) = 0, \\ 1 \geq R(-\infty) \geq R(\infty) = 0 \end{aligned} \tag{6}$$

The system(5) is a fourth order phase space system. The lower bound time speed for c can be found. J.D Murray shows this in *kiteMurray*. The *Infective* class in the system(5) can be linearised when $z \to \infty$ $S \to 1$ and $I \to 0$. The result then become

$$I'' + cI' + I(S - \lambda) \approx 0 \tag{7}$$

This can be found by

$$I(z) \exp\left[(-c \pm c^2 - 4(1-\lambda)^{1/2})z/2\right] \tag{8}$$

It is required that $I(z) \to 0$, but not under. This means that the solution cannot oscillate around 0. If a travelling wave exist, it has to satisfy

$$c \geq 2(1-\lambda)^{1/2}, \lambda < 1 \tag{9}$$

If $\lambda > 1$, no travelling wave will exist. Then the disease will die out. The terms defined in (2) will give the threshold conditions,

$$\lambda = \frac{a}{rS_0} < 1 \tag{10}$$

This is the same value that was given for the ODE model in the previous chapter.

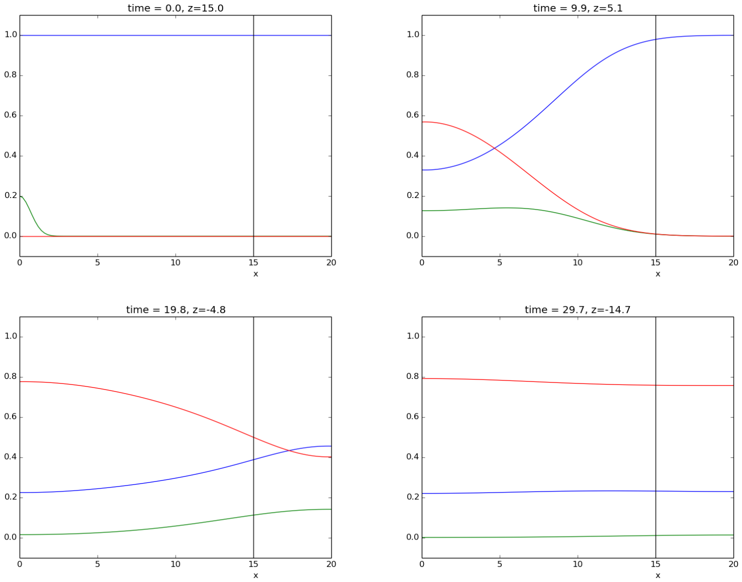


Figure 1: A gauss curve is inserted on the left side. This causes an epidemic wave. The size is measured at $x = 15$ and can be seen in figure (2).

2.2 Verifying the solution

Constant solution. To verify this solution a couple of tests can be done one the system. The first test will be a constant test, where a known constant solution is given. A look at the previous system will give an idea on performing the test.

$$\begin{aligned}\frac{\partial S}{\partial t} &= -IS + \frac{\partial^2 S}{\partial x^2},\\ \frac{\partial I}{\partial t} &= IS - \lambda I + \frac{\partial^2 I}{\partial x^2},\\ \frac{\partial R}{\partial t} &= \lambda I + \frac{\partial^2 R}{\partial x^2},\end{aligned} \tag{11}$$

By setting the solutions to constants, $S = C_s, I = C_i, R = C_r$, the value of C_i has to be 0. This is a poor test and several bugs can escape. The following system can be expanded by adding βR to the system. This give the chance to check the value of R and the constants can be set to a different value than 0.

$$\begin{aligned}C_I C_S &= \beta C_r\\ C_I C_S &= \lambda C_I\\ \lambda C_I &= -\beta C_R\end{aligned} \tag{12}$$

The constant values can freely be chosen. Here they are sat to $C_S = 1.2, C_I = 0.8, C_R = 0.6$. This give $\lambda = 1.2$ and $\beta = 1.6$. A test can be made

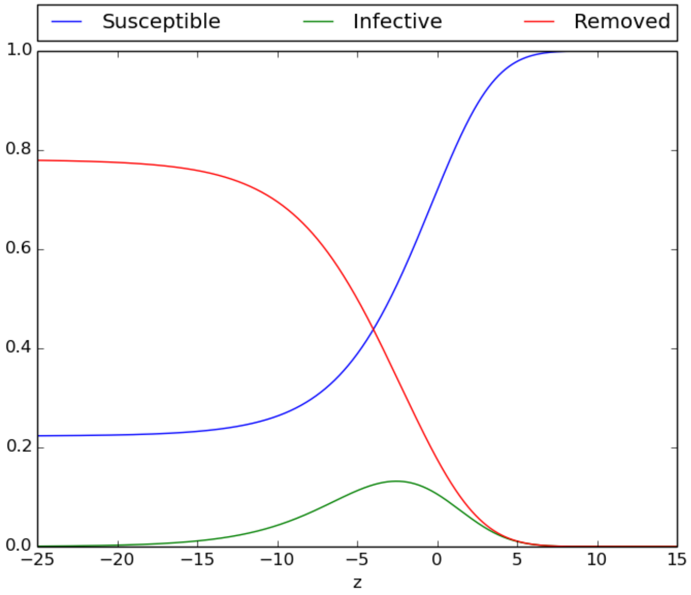


Figure 2: This shows the travelling wave measures at at $x = 15$ in figure(1)

```
def test_constant_solution():
    """
    Test problem where u=u_const is the exact solution, to be
    reproduced (to machine precision) by any relevant method.
    """
    def exact_solution(t):
        return C_s,C_i,C_r

    def lam(t,x):
        return C_s

    def beta(t,x):
        return (C_s*C_i)/float(C_r)

    #Constant values
    C_s = 1.2
    C_i = 0.8
    C_r = 0.6

    #lam = C_s
    #beta = (lam*C_i)/float(C_r)

    T = 2; Nt = 200
    X = 20; Nx = 40
    S_1 = np.ones(Nx+3)*C_s
    I_1 = np.ones(Nx+3)*C_i
    R_1 = np.ones(Nx+3)*C_r

    t,x,S,I,R = simple_PDE(T,Nx,Nt,X,lam,beta,S_1,I_1,R_1)

    S_e,I_e,R_e = exact_solution(t)
```

```
difference = abs(S_e - S).max()  # max deviation
tol = 1E-14
assert difference < tol

difference = abs(I_e - I).max()  # max deviation
tol = 1E-14
assert difference < tol

difference = abs(R_e - R).max()  # max deviation
tol = 1E-14
assert difference < tol
```

This test was fine and next test that can be done on the system is a Manufactured solution

Manufactured solution. By constructing a function to each equation in the system (3), a manufactured solution can be created. Here S,I and R are pre produced. The system will be

$$\begin{aligned}\frac{\partial S}{\partial t} &= -IS + \frac{\partial^2 S}{\partial x^2} + f(x,t),\\ \frac{\partial I}{\partial t} &= IS - \lambda I + \frac{\partial^2 I}{\partial x^2} + g(x,t),\\ \frac{\partial R}{\partial t} &= \lambda I + \frac{\partial^2 R}{\partial x^2} + h(x,t),\end{aligned} \tag{13}$$

where f,g and h are constructed functions to achieve the expected results for S, I and R. In this case the functions will be:

$$\begin{aligned}f(x,t) &= \frac{\partial S}{\partial t} + IS - \frac{\partial^2 S}{\partial x^2}\\ g(x,t) &= \frac{\partial I}{\partial t} - IS + \lambda I - \frac{\partial^2 I}{\partial x^2}\\ h(x,t) &= \frac{\partial R}{\partial t} - \lambda I - \frac{\partial^2 R}{\partial x^2},\end{aligned} \tag{14}$$

When choosing the expected function for the classes, it is important that the fulfill the boundary conditions

$$u_x(0,t) = u_x(X,t) = 0 \tag{15}$$

The quantities have here been sat to:

$$\begin{aligned}S(x,t) &= cos(\frac{\pi}{X}x)t\\ I(x,t) &= cos(\frac{\pi}{X}x)t\\ R(x,t) &= cos(\frac{\pi}{X}x)t\end{aligned} \tag{16}$$

Now `sympy` can be used to do the calculations

```
>>> from sympy import *
>>> x,t,lam = symbols('x t lam')
>>> def s_simple(x,t):
...     return cos(pi*x)*t
...
>>> def i_simple(x,t):
...     return cos(pi*x)*t
...
>>> def r_simple(x,t):
...     return cos(pi*x)*t
...
>>> for x_point in 0,1:
...     print "s_x(%s,t): ", % x_point,
>>> for x_point in 0,1:
...     print "s_x(%s,t): " % x_point,
...     print diff(s_simple(x,t),x).subs(x,x_point).simplify()
...     print "i_x(%s,t): " % x_point,
...     print diff(i_simple(x,t),x).subs(x,x_point).simplify()
...     print "r_x(%s,t): " % x_point,
...     print diff(r_simple(x,t),x).subs(x,x_point).simplify()
...
s_x(0,t):  0
i_x(0,t):  0
r_x(0,t):  0
s_x(1,t):  0
i_x(1,t):  0
r_x(1,t):  0
>>> s = s_simple(x,t)
>>> i = i_simple(x,t)
>>> r = r_simple(x,t)
>>> f = diff(s,t)+i*s-diff(diff(s,x),x)
>>> print f.simplify()
(t**2*cos(pi*x) + pi**2*t + 1)*cos(pi*x)
>>> g = diff(i,t)-i*s+lam*i-diff(diff(i,x),x)
>>> print g.simplify()
(lam*t - t**2*cos(pi*x) + pi**2*t + 1)*cos(pi*x)
>>> h = diff(r,t)-lam*i-diff(diff(r,x),x)
>>> print h.simplify()
(-lam*t + pi**2*t + 1)*cos(pi*x)
```

Which give

$$\begin{aligned} f(x,t) &= (t^2\cos(\pi x) + \pi^2 t + 1)\cos(\pi x) \\ g(x,t) &= (\lambda t - t^2\cos(\pi x) + \pi^2 t + 1)\cos(\pi x) \\ h(x,t) &= (-\lambda t + \pi^2 t + 1)\cos(\pi x) \end{aligned} \tag{17}$$

The following manufactured test will then be

```
def test_manufactured_solution(T,Nt,X,Nx):
    """
    Test problem where u=c*t+I is the exact solution, to be
    reproduced (to machine precision) by any relevant method.
    """

    def exact_solution_S(t,x):
        return np.cos(np.pi*x)*t

    def exact_solution_I(t,x):
        return np.cos(np.pi*x)*t

    def exact_solution_R(t,x):
        return np.cos(np.pi*x)*t
    def beta(t,x):
        return exact_solution_S(t,x)*exact_solution_I(t,x)/exact_solution_R(t,x)
    lam = 1
    def f(t,x):
        return (t**2*np.cos(np.pi*x) + np.pi**2*t + 1)*np.cos(np.pi*x)

    def g(t,x):
        return (lam*t - t**2*np.cos(np.pi*x) + np.pi**2*t + 1)*np.cos(np.pi*x)

    def h(t,x):
        return (-lam*t + np.pi**2*t + 1)*np.cos(np.pi*x)


    dx = X/float(Nx)
    dt = T/float(Nt)
    S_1 = exact_solution_S(0,np.linspace(0-dx,X+dx,Nx+3))
    I_1 = exact_solution_I(0,np.linspace(0-dx,X+dx,Nx+3))
    R_1 = exact_solution_R(0,np.linspace(0-dx,X+dx,Nx+3))

    t,x,S,I,R = simple_PDE(T,Nx,Nt,X,lam,beta,S_1,I_1,R_1,f,g,h)
    S_e = exact_solution_S(t[-1],x)
    I_e = exact_solution_I(t[-1],x)
    R_e = exact_solution_R(t[-1],x)

    difference_S = abs(S_e - S).max()  # max deviation


    #for i in range(4):
    #    print "n",i,"S_e",exact_solution_S(t[i],x)

    #print "S",S
    #t_tot = np.sum(t[:-1])
    #print "t_tot",t_tot
    #difference_exp = t_tot*dt*np.cos(x*np.pi)*((2*(np.cos(np.pi*dx)-1))/dx**2+np.pi**2)
    #print "diff_exp", (abs(difference_exp)).max()
    print "diff",difference_S
    #tol = 1E-14
    #assert difference < tol

    difference_I = abs(I_e - I).max()  # max deviation
    #print "diff",difference_I
    #tol = 1E-14
    #assert difference < tol

    difference_R = abs(R_e - R).max()  # max deviation
    #print "diff",difference_R
    #tol = 1E-14
    #assert difference < tol
    return difference_S,difference_I,difference_R
```

Convergence rate. The program can be controlled by checking the convergence rate. The error term can be described as

$$\epsilon = C_x\Delta x^2 + C_t\Delta t \tag{18}$$

With equation(18, the expected convergence rate can be found for both Δx and Δt. To be able to separate the Δ's, the other value has to be close to eliminated. If the value for Δx wants to be found, $\Delta t \ll \Delta x$ has to be fullfilled. This will lead to $C_t\Delta t \approx 0$, and the error term for Δx can be found. The opposite thing can be done for Δt. A table for the error is produced for different values for $\Delta t = 0.05$ and $\Delta x = 0.1$.

	dx	dx/2	dx/4	dx/8
dt	9.8E-3	-	-	-
dt/4	9.9E-3	2.5E-3	-	-
dt/8	9.9E-3	2.5E-3	6.1E-4	-
dt/16	9.9E-3	2.5E-3	6.1E-4	1.5E-4

The spatial error. The table above gives an insight on how the error develops when reducing Δt and Δx. By studying the row where $\Delta t/16$, the $C_t \Delta t$ can be seen as close to negligible in equation(18. The error can then be expressed

$$\epsilon \propto \Delta x^r \tag{19}$$

The value is expected to be $r = 2$, since Crank Nicolson is used in the spatial discretization. This gives a 2.order error. By comparing the error for different Δx, the convergence rate, r, can be expressed,

$$r_{12} \simeq \frac{\log(\epsilon_1/\epsilon_2)}{\log(\Delta x_1/\Delta x_2)} \tag{20}$$

Since the table above has four different error values, these can be used to give three different convergence rates. $\Delta x_1 = dx, \Delta x_2 = dx/2....$ The same labeling will be done for the different error values, ϵ.

	ϵ_1/ϵ_2	ϵ_2/ϵ_3	ϵ_3/ϵ_4
r	2.0056	2.0014	2.0004

Here the rate goes towards 2, and a 2.order convergence rate seems to be fulfilled.

The temporal error. The temporal error is quite hard to find because the *Stability criteria* which is about

$$2\Delta t \leq \Delta x^2 \tag{21}$$

This results in the case that $\Delta x \ll \Delta t$ is impossible, because this only leads to an unstable solution. By looking at the column for $\frac{\Delta x}{8}$, the only stable solution is for $\Delta t16$. Therefore the technique used for the spatial error cannot be used here. By studying the diagonal numbers in the table, the expected convergence rate is fulfilled for both Δx, which gives $r = 2$ and for Δt that gives $r = 1$

2.3 Travelling wave in 2D

The model can be discretized for a 2D simulation since an epidemic disease will spread geographically. The non dimensional system (3) used in the 1D simulation above can be discretized with Forward Euler in time and Crank Nicolson in space

$$\begin{aligned}
\frac{S_{i,j}^{n+1}-S_{i,j}^{n}}{\Delta t} &= -I_{i,j}^{n}S_{i,j}^{n} + \left(\frac{S_{i-1,j}^{n}-2S_{i,j}^{n}+S_{i+1,j}^{n}}{\Delta x^2} + \frac{S_{i,j-1}^{n}-2S_{i,j}^{n}+S_{i,j+1}^{n}}{\Delta y^2}\right)\\
\frac{I_{i,j}^{n+1}-I_{i,j}^{n}}{\Delta t} &= I_{i,j}^{n}S_{i,j}^{n} - \lambda I_{i,j}^{n} + \left(\frac{I_{i-1,j}^{n}-2I_{i,j}^{n}+I_{i+1,j}^{n}}{\Delta x^2} + \frac{I_{i,j-1}^{n}-2I_{i,j}^{n}+I_{i,j+1}^{n}}{\Delta y^2}\right)\\
\frac{R_{i,j}^{n+1}-R_{i,j}^{n}}{\Delta t} &= \lambda I_{i,j}^{n} + \left(\frac{R_{i-1,j}^{n}-2R_{i,j}^{n}+R_{i+1,j}^{n}}{\Delta x^2} + \frac{R_{i,j-1}^{n}-2R_{i,j}^{n}+R_{i,j+1}^{n}}{\Delta y^2}\right)
\end{aligned} \tag{22}$$

Now the known values can be placed on the right side. The system will then be

$$\begin{aligned}
S_{i,j}^{n+1} &= S_{i,j}^{n} + \Delta t\left(-I_{i,j}^{n}S_{i,j}^{n} + \left(\frac{S_{i-1,j}^{n}-2S_{i,j}^{n}+S_{i+1,j}^{n}}{\Delta x^2} + \frac{S_{i,j-1}^{n}-2S_{i,j}^{n}+S_{i,j+1}^{n}}{\Delta y^2}\right)\right)\\
I_{i,j}^{n+1} &= I_{i,j}^{n} + \Delta t\left(I_{i,j}^{n}S_{i,j}^{n} - \lambda I_{i,j}^{n} + \left(\frac{I_{i-1,j}^{n}-2I_{i,j}^{n}+I_{i+1,j}^{n}}{\Delta x^2} + \frac{I_{i,j-1}^{n}-2I_{i,j}^{n}+I_{i,j+1}^{n}}{\Delta y^2}\right)\right)\\
R_{i,j}^{n+1} &= R_{i,j}^{n} + \Delta t\left(\lambda I_{i,j}^{n} + \left(\frac{R_{i-1,j}^{n}-2R_{i,j}^{n}+R_{i+1,j}^{n}}{\Delta x^2} + \frac{R_{i,j-1}^{n}-2R_{i,j}^{n}+R_{i,j+1}^{n}}{\Delta y^2}\right)\right)
\end{aligned} \tag{23}$$

This results in an explicit system, which is easy to code. It consist of known values on the right side and only one unknown on the left side.

A gaussian wave. In the PDE system for the 1D equation, a Gaussian quantity of infected humans was placed on the left side in the initial value. This resulted in a wave of infected spread along the x-axis. A similar thing can be done for the 2D simulation. A couple of simulations have been done for the 2D system. The first simulation is run for a Gaussian function along $x = 0$ and the second simulation is run for a Gaussian function from $x = 0, y = 0$ for the infected humans as initial value. Both simulations can be seen in the Appendix.

By studying the travelling wave at a certain point, the size of the epidemic wave can be measured and compared. In these two 2D simulations, the wave will be measured in the point (15,15) while the travelling wave in the 1D simulation was measured in point(15). The two travelling waves can be seen in the figure(3).

The shape of these two travelling waves are similar. The only difference is the time when the wave occur. The plot for 1D wave in fig(2) has the same shape. With a closer study, the area under the function can be measured for all three cases. The result can be seen in the table below.

1.43352971688

1D wave	2D wave line	2D wave point
1.43	1.43	1.43

The area in all three cases are similar. The size and shape will not be affected based on the expansion from 1D to 2D. But by studying the figure(3), the waves occur at different time in the two 2D simulations. This is caused by the distance

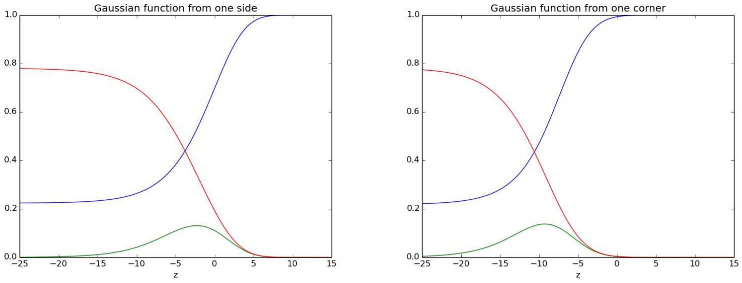


Figure 3: Travelling wave measured at point (15,15) with two different initial values for the Infected class. The initial value is sat as a Gaussian line along (0,y) in the left plot and as a Gaussian point (0,0) in the right plot

from the start position for the Gaussian wave. The first subplot that starts with a Gaussian function along the $x = 0$ axis, gets a wave of infected wash along the x axis. This can be seen as a wave on the beach. Everyone that have the same distance from the ocean will be hit simultaneously. The travelling wave for the 1D simulation and the first subplot occur at the same time, because they are measured at the same distance from the starting point. The last plot is also measured at (15,15), but occur later. Since the wave starts at point (0,0), the distance to (15,15) is 21.21. This means that the wave will start about 6 time steps later. This is also reasonable by looking at the plot.

Change in initial flow. Another test that can be done to the travelling wave, is to check if the size of the initial wave will affect the travelling wave. The simulation is run with the same parameter as for the three simulations above and the only difference is the initial value for the infected group. The Gaussian wave of infected are placed a point (0,0) as for right subplot in figure(3).The simulation can be seen in figure(4).

By measuring the travelling wave at (15,15), the size and shape can be compared with the second subplot in figure(3). This is done in figure(5).

The figure(5) shows a similar size and shape as for the other simulations run earlier. By increasing the initial value, the size of the travelling wave cannot be affected. But there is a difference in the time when the wave occur. In this simulation where the initial value is higher, the travelling wave reaches the measuring point (15,15) earlier. This can be explained by the idea of a ball dropped from height. If the ball is released or thrown to the ground, will only affect the acceleration of the ball, but not the terminal velocity. After a certain time the released ball and the thrown ball will reach the same maximum speed. This is also the case for the speed of the travelling wave.

Change in lambda. The one thing that affects the speed and size, is the λ variable in the PDE system(3). This λ is a combination of a, which controls deaths among infected, r, which control the number of infected in a meeting

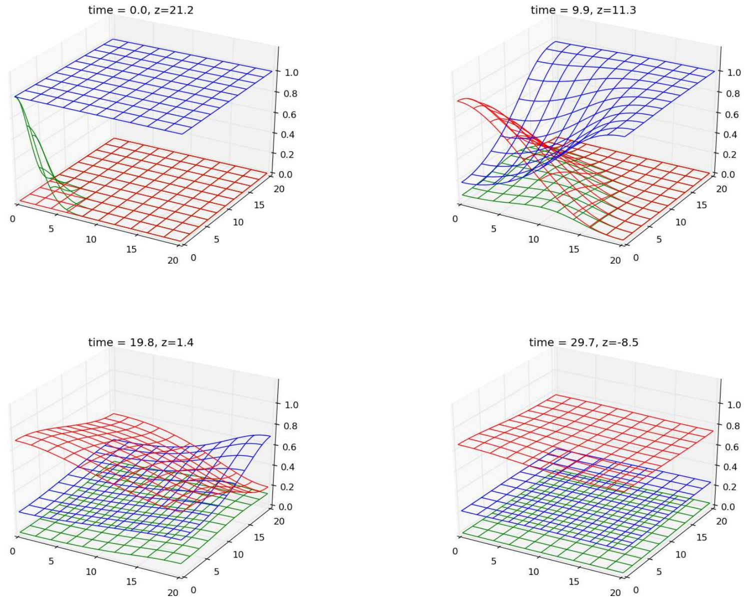


Figure 4: A major flow of infected spread outwards the field. After a certain time, the wave has past the area and the number in each class stabilize.

between an infected and a susceptible. The last parameter in λ is the concentration of Susceptible. By changing this parameter, the travelling wave will change in both size and shape. In figure(6), the simulation is run with four different values of λ.

To understand the result in figure(6), the λ function can be study,

$$\lambda = \frac{a}{rS_0}, \tag{24}$$

When $\lambda \to 0$, this cause a major and aggressive travelling wave. In figure(6), λ is run with the value 0.01 in the first subplot. This result in a travelling wave of infected that eradicates the number of Susceptible in a short time. The wave stars decreasing when all Susceptible are infected. By looking at the equation(24), a small value is caused by a small a compared to r and S_0. If a is low, this result in few deaths/immune in the infected class. This means that this class will grow and be able to infect even more humans from the Susceptible class. The same thing will happen if r is large. A result of this will cause an aggressive disease that will infect major part of the population. The same result will happen if S_0 is large. Then there are several possible humans to infect. Therefore a outburst of a disease is more critically in a city than in the wilderness far from other humans.

If λ increases above 1, the disease will not be able to spread. Then the number of infected will decrease, since the number of dead/immune infected are higher

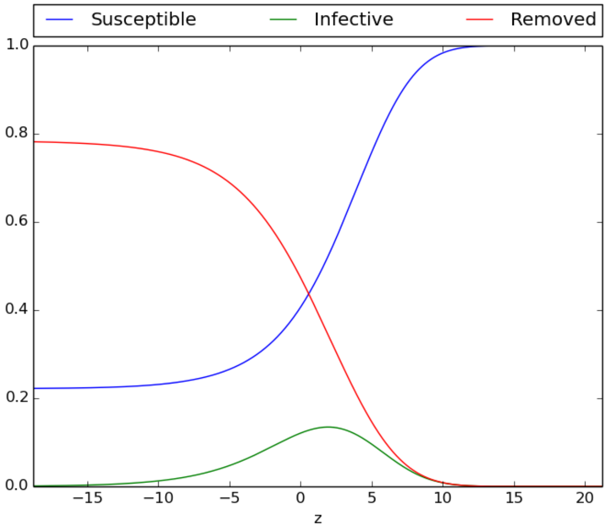


Figure 5: The travelling wave with a major increase of infected at the initial time.

than the infected susceptible. After a certain time, the number of infected will die out. If λ stays at 1, the number of infected will be equal the whole time.

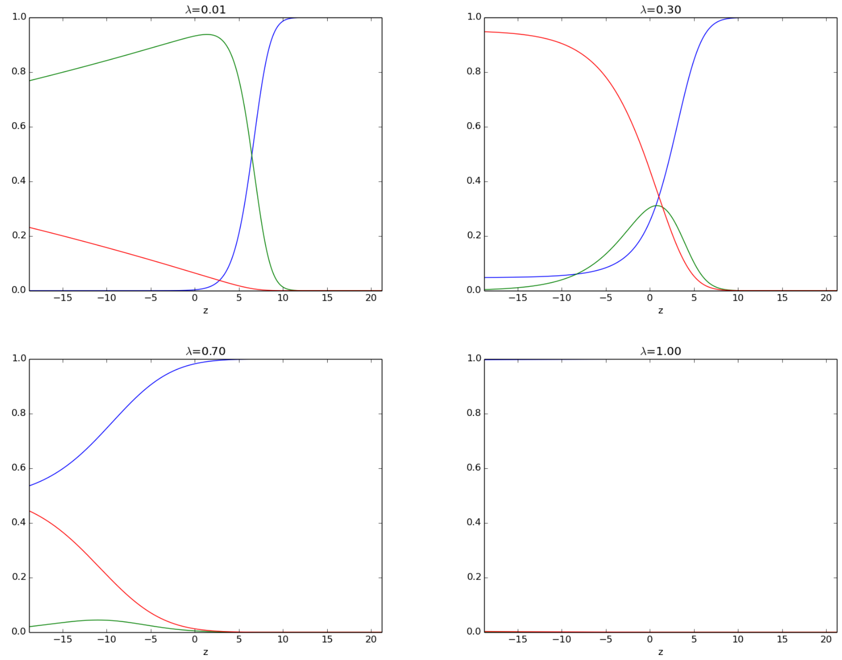


Figure 6: The travelling wave simulated with λ values in the range of 0.01 to 1

3 Epidemic in an English Boarding School 1978

I the previous chapter an example from the English boarding school was presented. This example was based on the book from J.D Murray and was modeled for a ODE system. By using the parameters a similar result should appear for the PDE system. The school had 763 students where one of the students brought with him a disease back to the school. The following number was used for the ODE system in chapter one. $N = 763, S_0 = 762, I_0 = 1, R_0 = 0, \rho = 202$ and $r = 2.18 \cdot 10^{-3}$.

The first simulation that is done on the PDE system is produced with an uniform distributed concentration of the three classes all over the area. A person has a given volume of one cubic. The total volume of the whole group is spread over the area. The area is sat to be 10 x 10, which means that the height of one person will be 1/100. Since the Infected group only consist of one person, the total height will be 0.01 for the whole area. The Susceptible group consist of 762 students and the total height in each point will be 7.62. The simulation can be seen in the Appendix.

The results from figure(7) are equal with the result from the ODE system modeled in the previous chapter.This can be seen in (3) This is as expected since the diffusion term i negligible in this system and that it in reality is group of separate ODE systems modeled over an area.

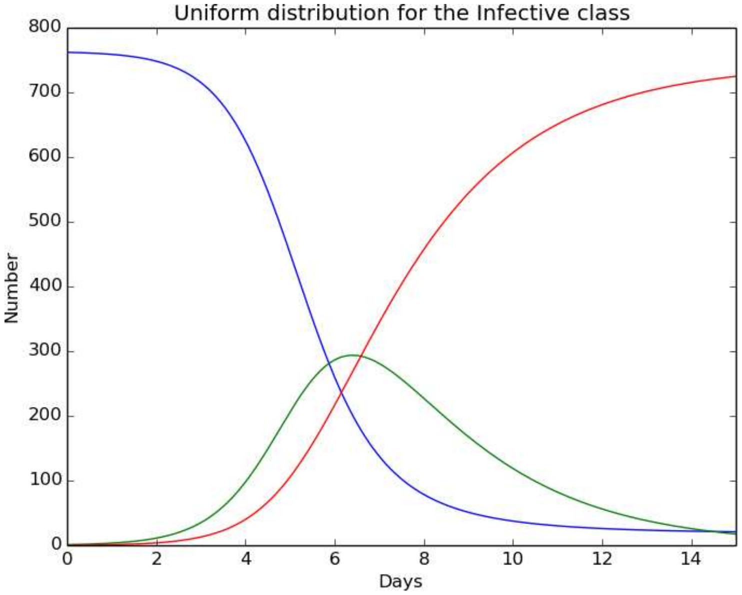
	ODE system	PDE uniform dist	PDE center	PDE corner
5 Days	——	———-	——	——
Susceptible	444.62	444.62	748.03	757.33
Infective	209.56	209.56	7.36	2.35
Removed	108.82	108.82	7.60	3.32
10 Days	——	———-	——	——
Susceptible	37.59	37.59	697.71	743.58
Infective	117.59	117.59	24.43	6.66
Removed	607.82	607.82	40.86	12.76
15 Days	——	———-	——	——
Susceptible	21.09	21.09	597.01	717.02
Infective	17.30	17.30	46.96	12.37
Removed	724.62	724.62	119.03	33.61

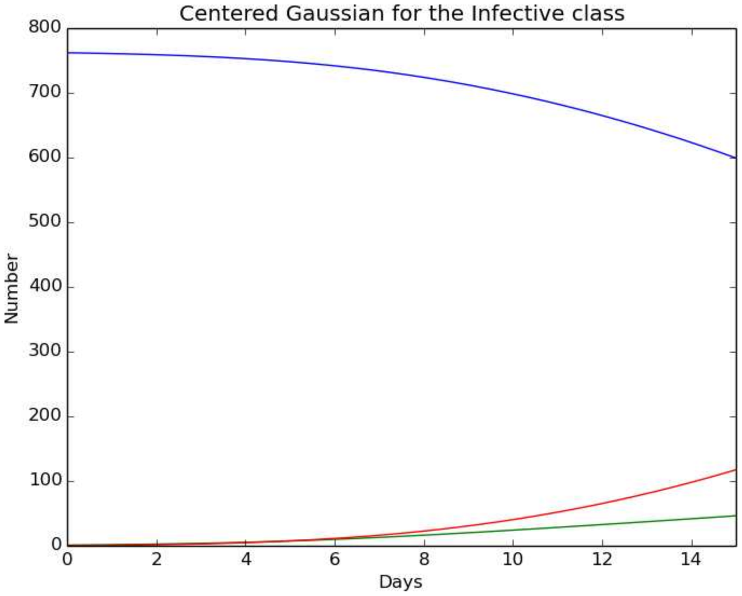
Introducing a Gaussian distribution of infective. A reasonable assumption to make is that a person is not able to be evenly distributed over an area. In this example with only one infected student, the chance of being infected increase as closer you get to the infected student. To check in this affect the result, the volume of student is placed as a gauss function in the center of the schoolyard at initial time. The height is sat to 1 and the volume of the Gauss function is sat to 1 cubic. The simulation can be seen in figure(8) and the total total amount of students can be be seen in figure(7).

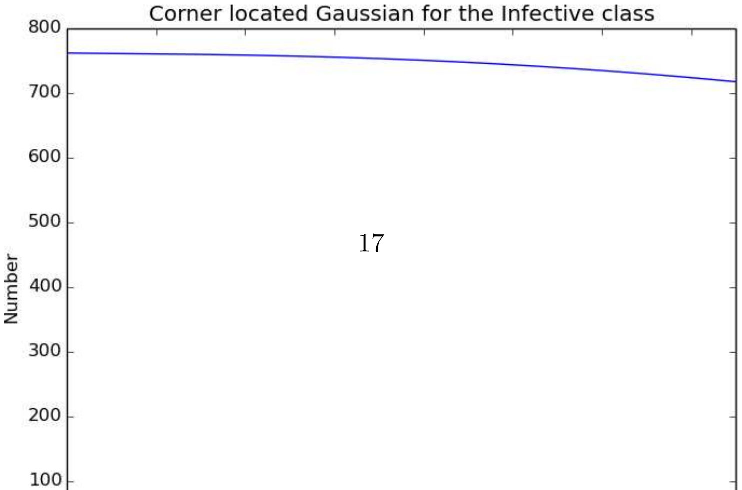
By comparing the result from the uniform distributed PDE system and this simulation there are some interesting results. While the only difference between the two simulations is the placement of the infective. And this change has a major impact. Since the only one that can be infected are the students close to the infective, it restrict the spread of the epidemic. But the chance of getting infected in this area is major. The figure(8) shows that the amount of infective quickly grow in the center, where the infected one was placed. The last subplot also show that the amount of removed in the center are closed to the maximum of the initial value of Susceptible. While the students along the boundary of the school seems to be unaffected after 15 days. This simulation show that the placement of the infective has a major role in the simulation.

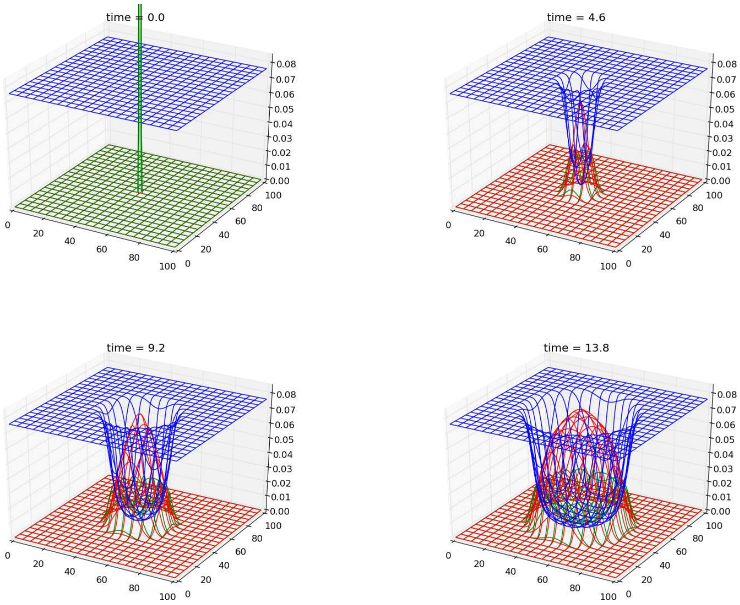
The placement of the Infective class also has an effect on the spread of the disease. The last column in (3) describe a simulation where the Gaussian function is placed in the corner(0,0). The total volume of the function is increased to 4 since only a quarter of the function is placed in the area. The table shows that the total number of infected is lower than for the centered placed Gaussian. This infected student is only able to spread the disease to a quarter of the population compared with the infected one in the center. The simulation can be seen in the Appendix.

If the simulations are run for a longer time, the difference in each class will decrease. After 100 days there will be about 18 Susceptible students left in the uniform distributed group, while there will be about 25 students in both of the Gaussian groups. The simulations can be seen in the Appendix.


Uniform distribution for the Infective class
Number
Days
0
100
200
300
400
500
600
700
800
0
2
4
6
8
10
12
14


Centered Gaussian for the Infective class
Number
Days
0
100
200
300
400
500
600
700
800
0
2
4
6
8
10
12
14


Corner located Gaussian for the Infective class
Number
100
200
300
400
500
600
700
800


time = 0.0

time = 4.6

time = 9.2

time = 13.8

Figure 8:

4 Zombiefication

The previous chapter studied a ODE system which was used to calculate the results of the TV series *Walking Dead*. This model was based on the system from LMR and was developed with an extra term in the phase *counter attack*. The following system that was used in chapter *Epidemic models* can be expand with a diffusion term in each class. The PDE system can be seen in eq(**??**)

$$
\begin{aligned}
\frac{\partial S}{\partial t} =& \Sigma - (\beta + \mu\omega(t))SZ - \delta_S S + D_s \nabla^2 S \\
\frac{\partial I}{\partial t} =& (\beta + \mu\omega(t))SZ - \varrho I - D_i \delta_I I + \nabla^2 I \\
\frac{\partial Z}{\partial t} =& \varrho I - (\alpha + \omega(t))SZ + \zeta R + D_z \nabla^2 Z \\
\frac{\partial R}{\partial t} =& \delta_S S + \delta_I I - \zeta R + (\alpha + \omega(t))SZ + D_r \nabla^2 R
\end{aligned}
$$

By discretize this with Forward Euler in time and Crank Nicolson in space as for the SIR model(22), the system can be expressed

$$
\begin{aligned}
\frac{S^{n+1}_{i,j} - S^n_{i,j}}{\Delta t} &= \Sigma - (\beta + \mu\omega(t))S^n_{i,j}Z^n_{i,j} - \delta_S S^n_{i,j} + \ D_s \left(\frac{S^n_{i-1,j} - 2S^n_{i,j} + S^n_{i+1,j}}{\Delta x^2} + \frac{S^n_{i,j-1} - 2S^n_{i,j} + S^n_{i,j+1}}{\Delta y^2} \right) \\
\frac{I^{n+1}_{i,j} - I^n_{i,j}}{\Delta t} &= (\beta + \mu\omega(t))S^n_{i,j}Z^n_{i,j} - \varrho I^n_{i,j} - \delta_I I^n_{i,j} + \ D_i \left(\frac{I^n_{i-1,j} - 2I^n_{i,j} + I^n_{i+1,j}}{\Delta x^2} + \frac{I^n_{i,j-1} - 2I^n_{i,j} + I^n_{i,j+1}}{\Delta y^2} \right) \\
\frac{Z^{n+1}_{i,j} - Z^n_{i,j}}{\Delta t} &= \varrho I^n_{i,j} - (\alpha + \omega(t))S^n_{i,j}Z^n_{i,j} + \zeta R^n_{i,j} + \ D_z \left(\frac{Z^n_{i-1,j} - 2Z^n_{i,j} + Z^n_{i+1,j}}{\Delta x^2} + \frac{Z^n_{i,j-1} - 2Z^n_{i,j} + Z^n_{i,j+1}}{\Delta y^2} \right) \\
\frac{R^{n+1}_{i,j} - R^n_{i,j}}{\Delta t} &= \delta_S S^n_{i,j} + \delta_I I^n_{i,j} - \zeta R^n_{i,j} + (\alpha + \omega(t))S^n_{i,j}Z^n_{i,j} D_r \left(\frac{R^n_{i-1,j} - 2R^n_{i,j} + R^n_{i+1,j}}{\Delta x^2} + \frac{R^n_{i,j-1} - 2R^n_{i,j} + R^n_{i,j+1}}{\Delta y^2} \right)
\end{aligned}
\tag{25}
$$

By setting the unknown on the left side and the known on the right, the following system can be computed

$$
\begin{aligned}
S^{n+1}_{i,j} &= S^n_{i,j} + \Delta t \Bigg(\Sigma - (\beta + \mu\omega(t))S^n_{i,j}Z^n_{i,j} - \delta_S S^n_{i,j} + \ D_s \Bigg(\frac{S^n_{i-1,j} - 2S^n_{i,j} + S^n_{i+1,j}}{\Delta x^2} + \frac{S^n_{i,j-1} - 2S^n_{i,j} + S^n_{i,j+1}}{\Delta y^2} \Bigg) \Bigg) \\
I^{n+1}_{i,j} &= I^n_{i,j} + \Delta t \Bigg((\beta + \mu\omega(t))S^n_{i,j}Z^n_{i,j} - \varrho I^n_{i,j} - \delta_I I^n_{i,j} + \ D_i \Bigg(\frac{I^n_{i-1,j} - 2I^n_{i,j} + I^n_{i+1,j}}{\Delta x^2} + \frac{I^n_{i,j-1} - 2I^n_{i,j} + I^n_{i,j+1}}{\Delta y^2} \Bigg) \Bigg) \\
Z^{n+1}_{i,j} &= Z^n_{i,j} + \Delta t \Bigg(\varrho I^n_{i,j} - (\alpha + \omega(t))S^n_{i,j}Z^n_{i,j} + \zeta R^n_{i,j} + \ D_z \Bigg(\frac{Z^n_{i-1,j} - 2Z^n_{i,j} + Z^n_{i+1,j}}{\Delta x^2} + \frac{Z^n_{i,j-1} - 2Z^n_{i,j} + Z^n_{i,j+1}}{\Delta y^2} \Bigg) \Bigg) \\
R^{n+1}_{i,j} &= R^n_{i,j} + \Delta t \Bigg(\delta_S S^n_{i,j} + \delta_I I^n_{i,j} - \zeta R^n_{i,j} + (\alpha + \omega(t))S^n_{i,j}Z^n_{i,j} D_r \Bigg(\frac{R^n_{i-1,j} - 2R^n_{i,j} + R^n_{i+1,j}}{\Delta x^2} + \frac{R^n_{i,j-1} - 2R^n_{i,j} + R^n_{i,j+}}{\Delta y^2}
\end{aligned}
\tag{26}
$$

To verify that the discretization has been done correct, a simulation with uniform distributed classes can be done. The result is the expected to be similar as for the ODE system in the previous chapter. A zombie attack can be separate into three different phases, explained in detail in the previous chapter. The first phase is short and is called the *initial phase*. This is a condition where the humans are unfamiliar with the disease and behave quite naive to the disease. This result in a high chance of getting infected. The next phase is called *Hysterical phase*. The humans are now more familiar with the situation and tries to avoid the infected ones. This result in a lower chance of getting infected. The last phase, which happens at the same time as the hysterical phase, is the counter attack. This phase is often started if the humans are attacked by zombies. The following parameter that was used for simulating the first episodes in *Walking Dead* will also be used here. These can be seen in (4). By compute the system for all three phases, the value in each phase can be compared with the ones from the ODE system in the previous chapter. This will give an indication if the discretization is done correct.

parameter	Initial phase	hysterical phase	counter attack
β	0.01155	0.000011	0.00011
ϱ	1.37	1.5	1.5
α	0.00044	0.000208	0.000208
a	0	0	0.0073
σ	0	0	0.005
μ	0	0	0.14

The simulation in figure(9) seems to match the result from the ODE system. A closer check can be done by comparing the groups in each phase. This result can be seen in (4)

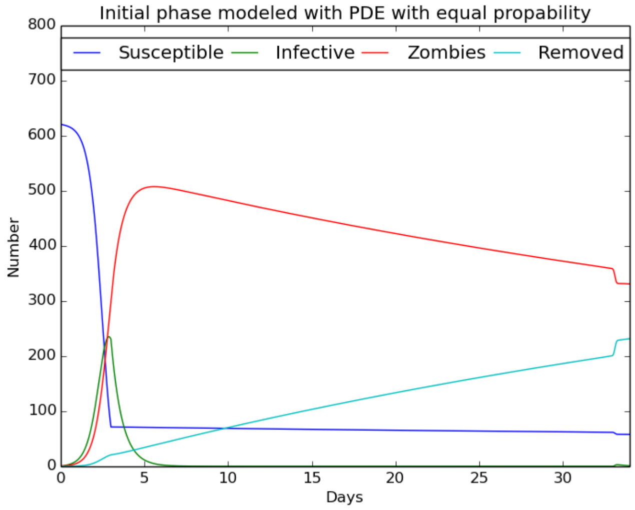


Figure 9: Same results as for the ODE system in the previous chapter

	ODE system	PDE uniform dist	PDE gauss center
Initial phase	———	—————-	—————-
Susceptible	71.3	71.5	81.12
Infected	230.8	230.6	210.94
Zombie	298.9	298.9	310.11
Removed	21.0	21.0	<:20.60
Hysterical phase	———	—————-	—————-
Susceptible	61.6	61.8	70.55
Infected	0.3	0.3	0.34
Zombie	358.6	357.9	334.33
Removed	201.5	202.0	217.56
Counter attack	———	—————-	—————-
Susceptible	57.8	58.2	66.50
Infected	1.2	1.2	1.23
Zombie	331.8	331.1	305.86
Removed	231.3	231.7	249.19

These results shows that the PDE system gives the same result as for the ODE. The error can be controlled by decreasing the Δt and $Deltax$ for the system.

Variation in diffusion.

Where does the infected one arises. In the previous section *English boarding school*, the placement of the infected was proven to have a major influence on the result. But here the *Susceptible* was uniformed distributed over the schoolyard. Based on the data from *Walking Dead*, the amount in each class was based on three different places where Rick went by. Based on the TV series, it is hard to decide the geograpic distance between these three places. Therefore they have been placed with a certain distance from each other. The following simulation is done on a grid with size(20 x 20) with the following placement on the towns. Small(4,4) with size 21, middle(9,15) with size 200 and large(15,5) with size 400. Since these values was based on the humans and zombies seen in the series, these can be scaled up. The total population can be scaled up with 1000, so the large town can be seen as an area with a population of 400 000. The length can be measured in kilometre. Then the distance between the middle and large town will be 11.66 km. Compared to the distance between Oslo and Bærum(Sandvika) which is 15 km, the simulation can be seen as rough estimate of the area around Oslo.

The diffusion term describe the diffusion for each class. This can be seen as the speed of each group. If the diffusion constant is large, the flow towards equilibrium will go faster. The values have been sat as follows.

- $D_s = 1$, The *Susceptible* has a basic moving speed. The other classes are based on the speed of a healthy human.

- $D_i = 0.5$, The *Infected* are often injured caused by lately fights. This affect their moving speed
- $D_z = 0.9$, The average moving speed for a zombie. There are zombies with the moving speed of a human, but there are also cases where zombies edge them self forward with only one arm. This result in a major difference in speed and therefore a lower average speed for the zombies.
- $D_r = 0$, The removed are here seen as dead, therefore there are no moving speed.

The parameters from (4) will be used here and the three phases will be modeled as shown for the uniformed distributed PDE system. This will be done for three different simulations with similar initial value for the susceptible class. The initial value can be seen in figure(10) and are based on the data given above. The difference in the three different simulations will be the placement of the zombie at initial time. The zombie will be placed in center of the small, middle and large town.

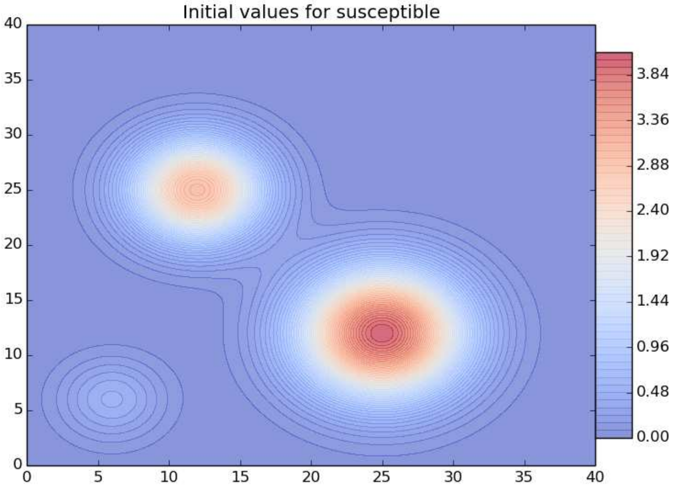
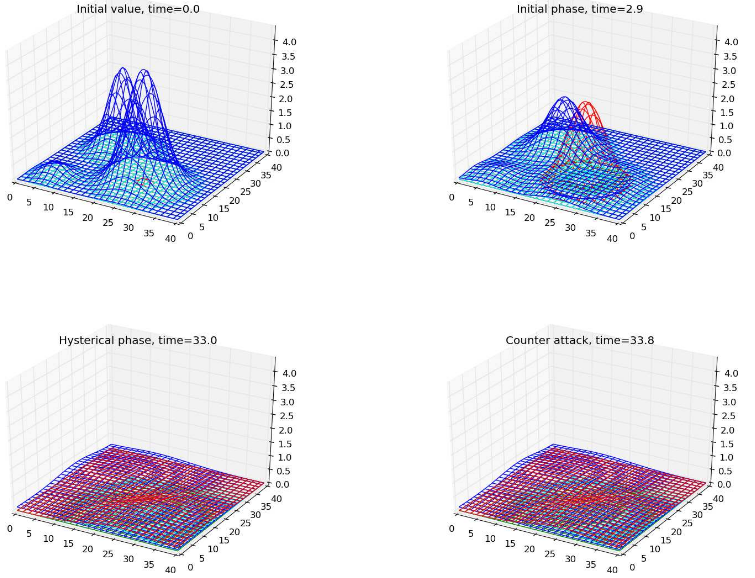


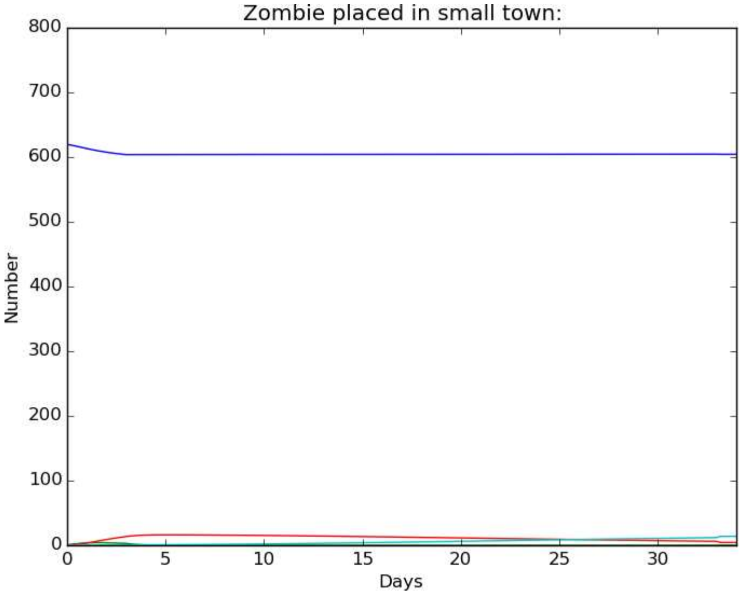
Figure 10:

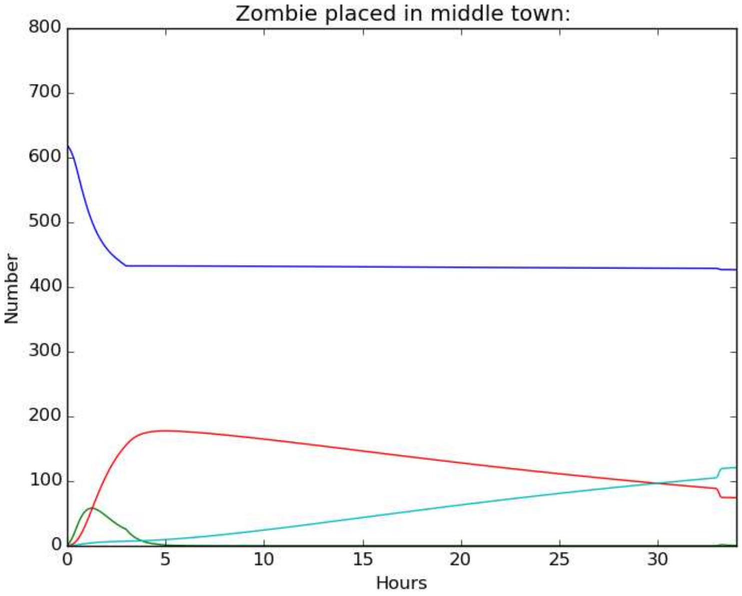
The figure() shows the simulation where the zombie is placed in the small town. The four subplots are from the different phases that arise under a zombie attack. The different classes have the same color as introduced in figure(9). It is a challenge to separate the three groups *Infected*,*Zombie* and *Removed*, since they all have a low value at initial time. The development of the amount can easier be seen in the figure(), which also shows the result from the placement in the middle and large group. Since the amount of *Susceptible* is quite low in the small town where the Zombie arise, the disease is not able to infect to many before the society has moved to the next phase. Assuming that the broadcasting works okay for the first days. This result in a eradication of the disease in about a month. The table() shows that the number of zombies decrease towards zero after a month.

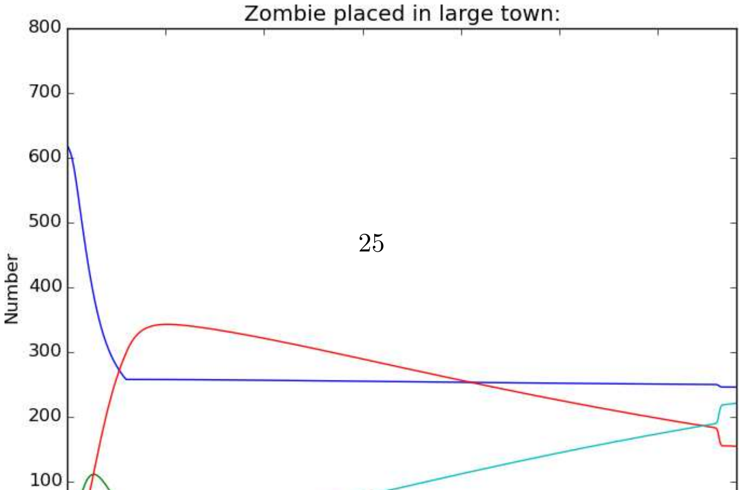


By placing the Zombie in the middle town, the amount of zombies increase to a much higher level. The amount can be seen as the second sub figure in figure(). The damages are higher and after a month the total population of *Susceptible* will be reduced to around (300). The simulation can be seen in the Appendix. The last calculation done in the large town shows the major damages. Here the amount of zombies increase above the number of *Susceptible*. The *Infected* class also increase towards 100 around in the infected phase. This can be explained by the high number of meeting between susceptible and zombies. A plot that shows each class is produced for this simulation. This give a better overview of the concentration of the four groups during this simulation. Since the diffusion term for the *Removed* group is sat to zero, the *Removed* subplots gives an indication on the placement of the deaths. This seems to be grouped in the city. The amount of zombies also seems to be higher here. The major amount of *Susceptible* are in the middle town. This area was able to avoid the damages from the initial phase.

The result from the uniformed distributed simulation still is much higher for the *Zombie* class then for the large town. This shows that by using the parameters from the ODE system in a geographic area give no sense. A realistic assumption is that a zombie is restricted to a given area. Therefore the parameters will not be equal for all. The chance of getting infected is much higher if a susceptible is closed to an infected. There is also a greater chance of getting infected if the susceptible stay in a crowded area.


Zombie placed in small town:
Number
800
700
600
500
400
300
200
100
0
0
5
10
15
20
25
30
Days


Zombie placed in middle town:
Number
800
700
600
500
400
300
200
100
0
0
5
10
15
20
25
30
Hours


Zombie placed in large town:
Number
800
700
600
500
400
300
200
100

	Small town	Middle town	Large
Initial phase	———	————-	————-
Susceptible	603.74	433.22	259.20
Infected	2.96	25.51	50.94
Zombie	13.79	155.27	297.24
Removed	0.66	7.16	13.78
Hysterical phase	———	————-	————-
Susceptible	604.42	429.35	251.14
Infected	0.03	0.18	0.35
Zombie	6.25	87.31	178.45
Removed	12.14	106.00	192.90
Counter attack	———	————-	————-
Susceptible	604.21	427.45	247.33
Infected	0.08	0.59	1.17
Zombie	4.49	73.70	151.44
Removed	14.11	121.16	222.96

4.1 Lock in different areas

To model a realistic zombie attack, humans ability to think logic is crucial in the fight. The moving speed was presented as a factor in the previous section. Another important skill that the *Susceptible* class hold, is the ability to decide if an area is safe. In the TV series *Walking Dead*, the humans build barricades to keep the zombies outside. This give the *Susceptible* class free areas where they can stay. This idea can be transfer to the PDE system by rewrite the system (**??**) with spatial dependent diffusion term. The diffusion constant D_u is now replaced with a diffusion function $\gamma_u(x)$ for $u = S, I, Z, R$ which is spatial discretized.

$$
\begin{aligned}
\frac{\partial S}{\partial t} =& \Sigma - (\beta + \mu\omega(t))SZ - \delta_S S + \nabla(\gamma_S(x)\nabla S) \\
\frac{\partial I}{\partial t} =& (\beta + \mu\omega(t))SZ - \varrho I - D_i\delta_I I + \nabla(\gamma_I(x)\nabla I) \\
\frac{\partial Z}{\partial t} =& \varrho I - (\alpha + \omega(t))SZ + \zeta R + \nabla(\gamma_Z(x)\nabla Z) \\
\frac{\partial R}{\partial t} =& \delta_S S + \delta_I I - \zeta R + (\alpha + \omega(t))SZ + \nabla(\gamma_R(x)\nabla R)
\end{aligned}
\tag{27}
$$

The difference for this system compared with (**??**) is the diffusion term in each equation. The discretization can be shown for for a general γ. This will be similar for all classes. A Crank nicolson discretization is used in space.

$$
\begin{aligned}
&= \nabla(\gamma(x)\nabla S) \\
&= (\gamma(x)S_x)_x + (\gamma(x)S_y)_y \\
&= \left(\gamma(x)\frac{S^n_{i+1/2,j} - S^n_{i-1/2,j}}{\Delta x}\right)_x + \left(\gamma(x)\frac{S^n_{i,j+1/2} - S^n_{i,j-1/2}}{\Delta y}\right)_y \\
&= \left(\frac{\gamma(x_{i+1/2,j})(S^n_{i+1,j} - S^n_{i,j}) - \gamma(x_{i-1/2,j})(S^n_{i,j} - S^n_{i-1,j})}{\Delta x^2}\right) + \left(\frac{\gamma(x_{i,j+1/2})(S^n_{i,j+1} - S^n_{i,j}) - \gamma(x_{i,j-1/2})(S^n_{i,j} - S^n_{i,j-})}{\Delta y^2}\right.
\end{aligned}
\tag{28}
$$

Since the calculation is based on spatial points, the values inside the function of γ need to be adjusted. This can be done by an aritmetic mean, which can be seen in eq(29). The writing $q_{i+1/2}$ is used for the function $q(x_{i+1/2})$ with $x_{i+1/2} = x_i + 1/2\Delta x$

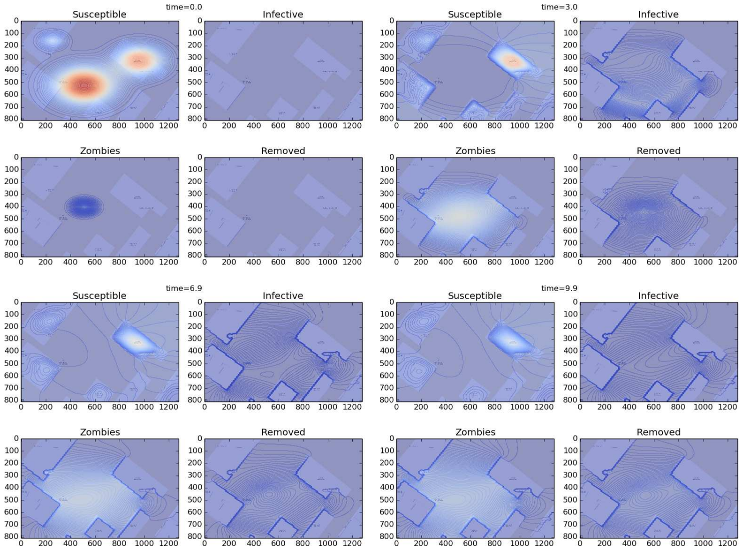
$$
q_{i+1/2} \approx \frac{1}{2}(q_i + q_{i+1}) \tag{29}
$$

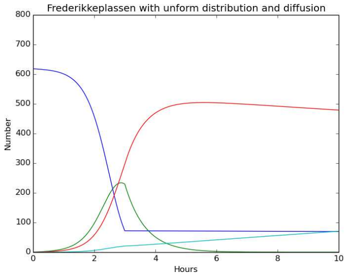
This arithmetic mean can be inserted for all γ's in the system.By cleaning up, the system can be expressed.

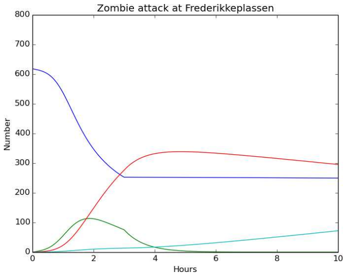
$$
\begin{aligned}
S^{n+1}_{i,j} &= S^n_{i,j} + \Delta t\left(\Sigma - (\beta + \mu\omega(t))S^n_{i,j}Z^n_{i,j} - \delta_S S^n_{i,j} + \frac{1}{2\Delta x^2}\left(\gamma_S(x_{i-1,j})(S^n_{i-1,j} - S^n_{i,j}) + \gamma_S(x_{i,j})(S^n_{i-1,j} - 2S^n_{i,j} + S\right.\right. \\
I^{n+1}_{i,j} &= I^n_{i,j} + \Delta t\left((\beta + \mu\omega(t))S^n_{i,j}Z^n_{i,j} - \varrho I^n_{i,j} - \delta_I I^n_{i,j} + \frac{1}{2\Delta x^2}\left(\gamma_I(x_{i-1,j})(I^n_{i-1,j} - I^n_{i,j}) + \gamma_I(x_{i,j})(I^n_{i-1,j} - 2I^n_{i,j} + I^n\right.\right. \\
Z^{n+1}_{i,j} &= Z^n_{i,j} + \Delta t\left(\varrho I^n_{i,j} - (\alpha + \omega(t))S^n_{i,j}Z^n_{i,j} + \zeta R^n_{i,j} + \frac{1}{2\Delta x^2}\left(\gamma_Z(x_{i-1,j})(Z^n_{i-1,j} - Z^n_{i,j}) + \gamma_Z(x_{i,j})(Z^n_{i-1,j} - 2Z^n_{i,j} +\right.\right. \\
R^{n+1}_{i,j} &= R^n_{i,j} + \Delta t\left(\delta_S S^n_{i,j} + \delta_I I^n_{i,j} - \zeta R^n_{i,j} + (\alpha + \omega(t))S^n_{i,j}Z^n_{i,j}\right)
\end{aligned}
\tag{30}
$$

The diffusion term for the *Removed* class is take away, since dead people are not able to move. This system looks quite messy, but it is straight forward to calculate. All values on the right side are known values and the system is easy to solve. Now every point will has its own diffusion constant. This makes it easier to control the flow in each class. With a high diffusion constant, the diffusion will spread fast. When the diffusion constant goes towards zero, the flow will go towards a steady state. This will result in a set of ODE systems modeled for each point.

Ten minutes at Frederikkeplassen. Frederikkeplassen at the university is an area for an upcoming zombie attack. This simulation will try to model a ten minutes sequence with the diffusion parameter added in this section. Since students often learn and interact fast, they will only use three minutes before they realise the danger and goes into the *Hysterical phase*. A map of Frederikkeplassen is used to define the safe and non save areas. The buildings are set as areas where only the *Susceptible* are allowed to move. This is done by setting the diffusion constant to zero for the *Zombie* and *Infected* class. Since the buildings will be safe spots for the *Susceptible*, an idea would be to express this in the diffusion term. This more difficult, since the concentrations in each class wants to go towards equilibrium. A way to delay this process is by setting the diffusion constant to be low in the buildings and high outside. This will result in a fast diffusion in the open areas and



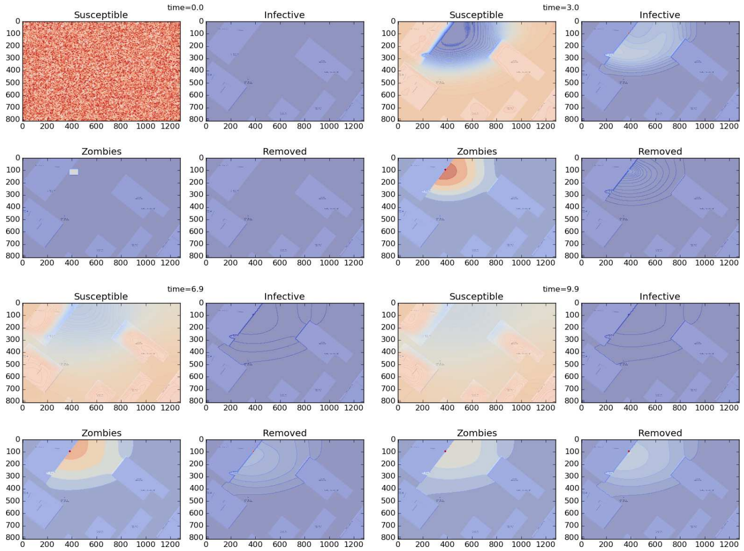




-

	Uniform distribution	Free areas	Random placement
3 Minutes	———	———	———-
Susceptible	72.23	252.72	524.77
Infected	229.65	75.69	26.07
Zombie	296.67	276.55	66.51
Removed	20.84	13.94	3.66
7 Minutes	———	———	———-
Susceptible	70.78	251.35	524.23
Infected	0.83	0.51	0.20
Zombie	498.72	325.54	81.88
Removed	49.12	41.26	14.80
10 Minutes	———	———	———-
Susceptible	69.69	249.84	523.61
Infected	0.25	0.38	0.16
Zombie	479.00	295.71	69.67
Removed	70.55	72.36	27.77

random distribution.



5 Appendix

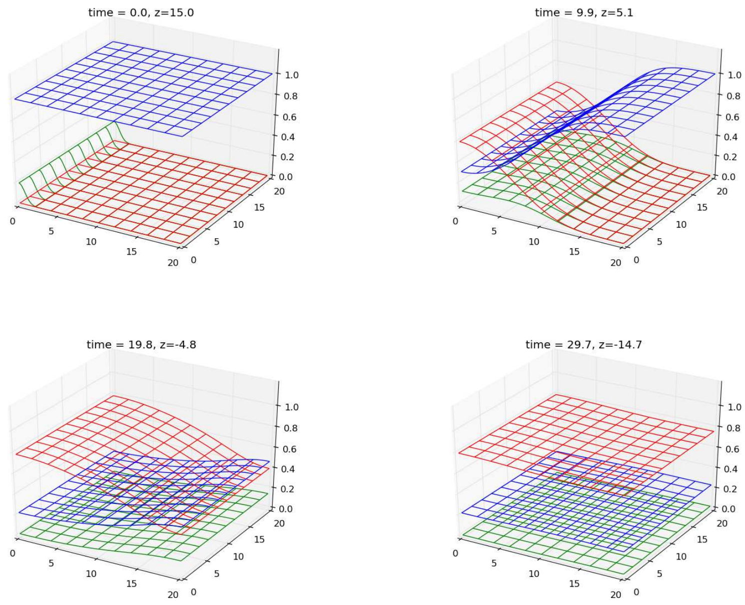
5.1 Discretization

$$
\begin{aligned}
\frac{dS}{dt} =& \Sigma - \beta SZ - \delta_S S + \nabla(\gamma_s(x)\nabla S) \\
\frac{dI}{dt} =& \beta SZ - \varrho I - \delta_I I + \nabla(\gamma_I(x)\nabla I) \\
\frac{dZ}{dt} =& \varrho I - (\alpha + \omega(t))SZ + \zeta R + \nabla(\gamma_Z(x)\nabla Z) \\
\frac{dR}{dt} =& \delta_S S + \delta_I I - \zeta R + (\alpha + \omega(t))SZ
\end{aligned}
\tag{31}
$$

The calculations will be shown for the diffusion part in the first equation. This idea will be used for the whole system

$$
\begin{aligned}
\frac{dS}{dt} =& \nabla(\gamma_s(x)\nabla S) \\
\frac{S^{n+1}_{i,j} - S^{n}_{i,j}}{\Delta t} =& \left(\gamma(x_{i+1/2,j})\frac{S^{n}_{i-1,j} - 2S^{n}_{i,j} + S^{n}_{i+1,j}}{\Delta x} + \frac{S^{n}_{i,j-1} - 2S^{n}_{i,j} + S^{n}_{i,j+1}}{\Delta y}\right)
\end{aligned}
\tag{32}
$$

5.2 2D Gaussian function from x=0



time = 0.0, z=15.0

time = 9.9, z=5.1

time = 19.8, z=-4.8

time = 29.7, z=-14.7

Figure 11: The PDE system (3) simulated for a 2D system with $\lambda = 0.5$. A gauss wave from $x = 0$.

5.3 2D Gaussian function from x=0,y=0

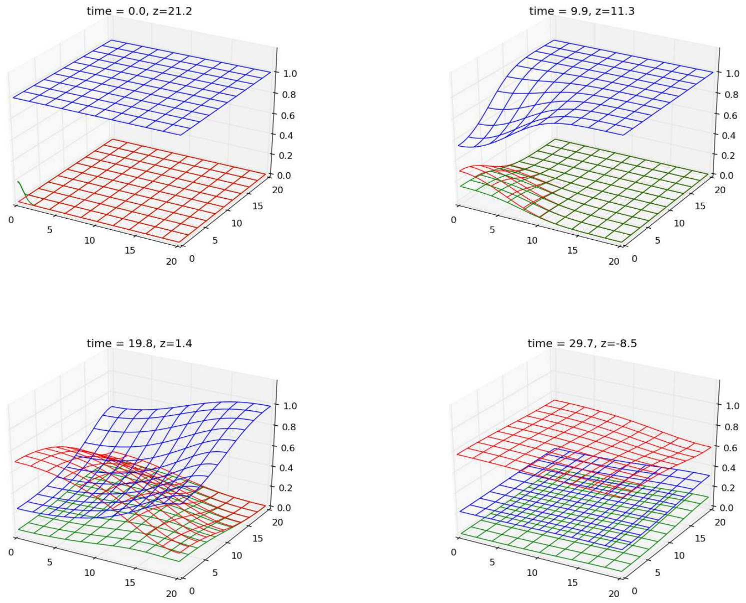
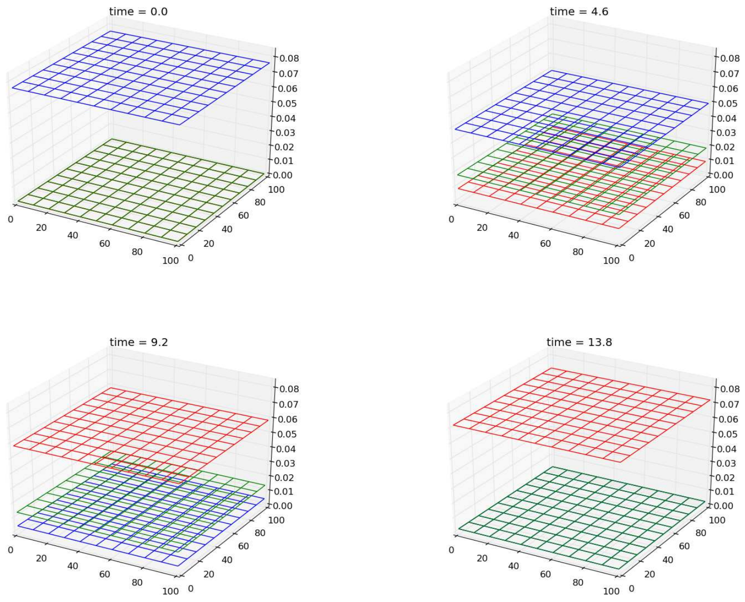


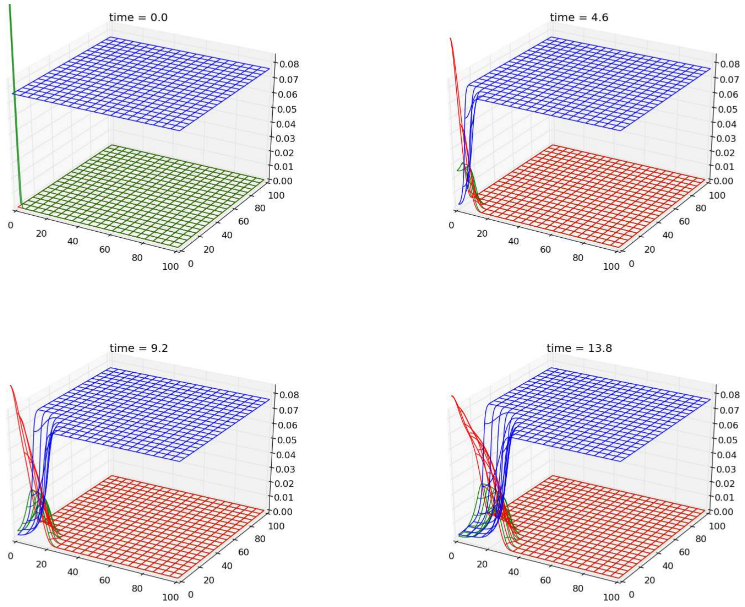
Figure 12: A gaussian function from $x = 0, y = 0$ based on the PDE system (3) with $\lambda = 0.5$

5.4 2D Gaussian function from x=0,y=0 with higher initial value

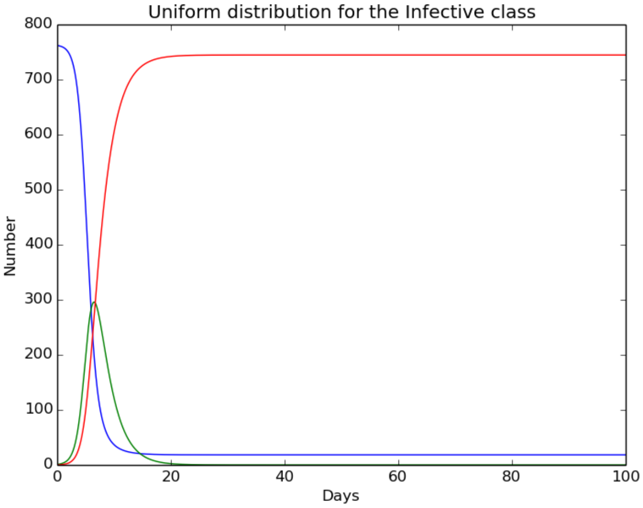
5.5 English Boarding School

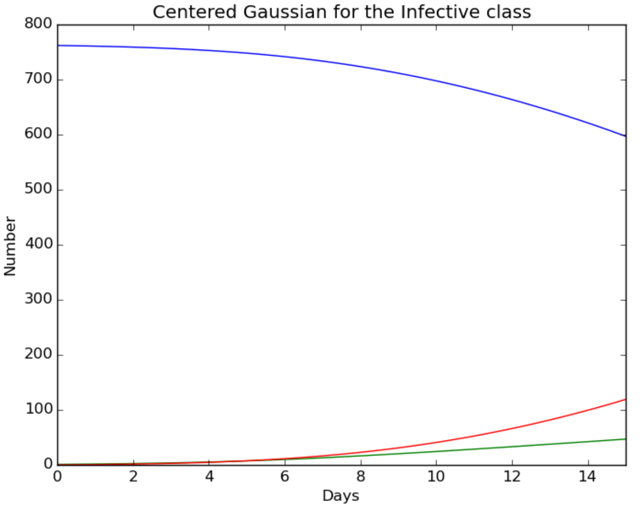


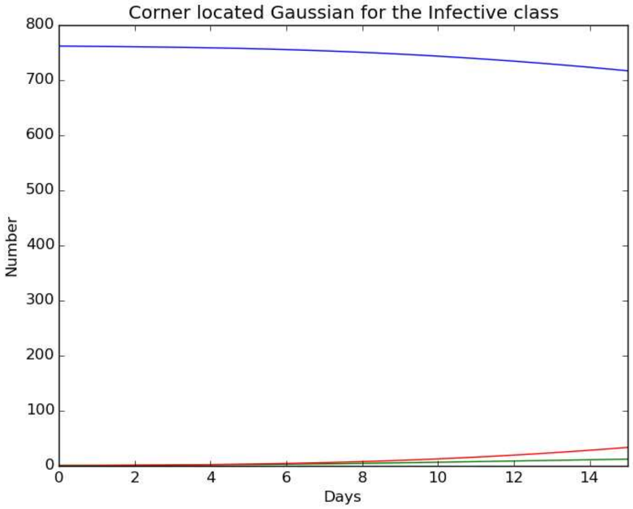
Gaussian from the corner.



A long simulation on 100 Days.

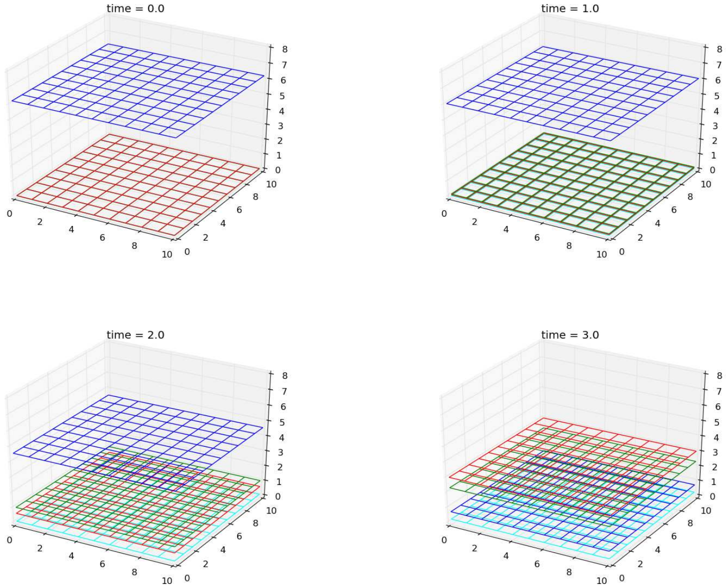






5.6 Zombiefication

Verify the initial phase.



Three phases.

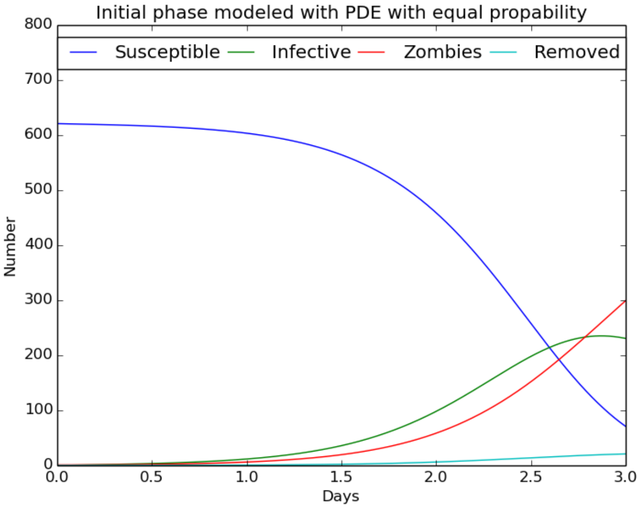
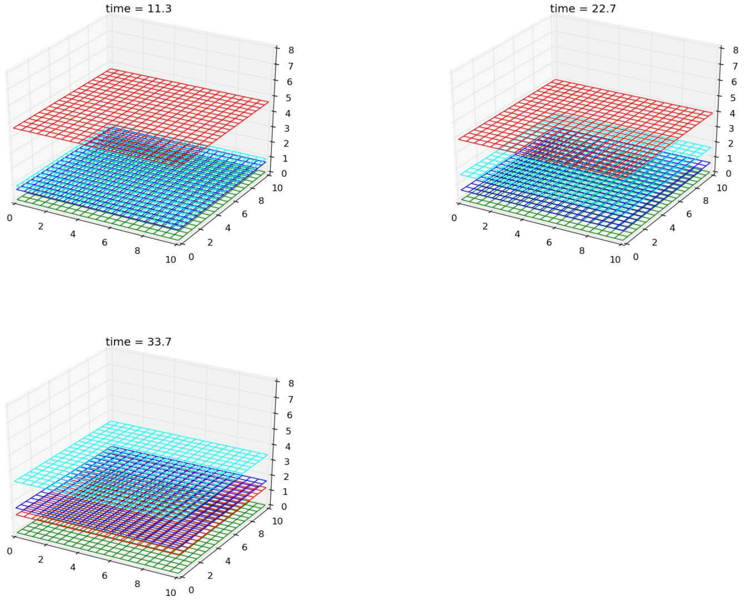
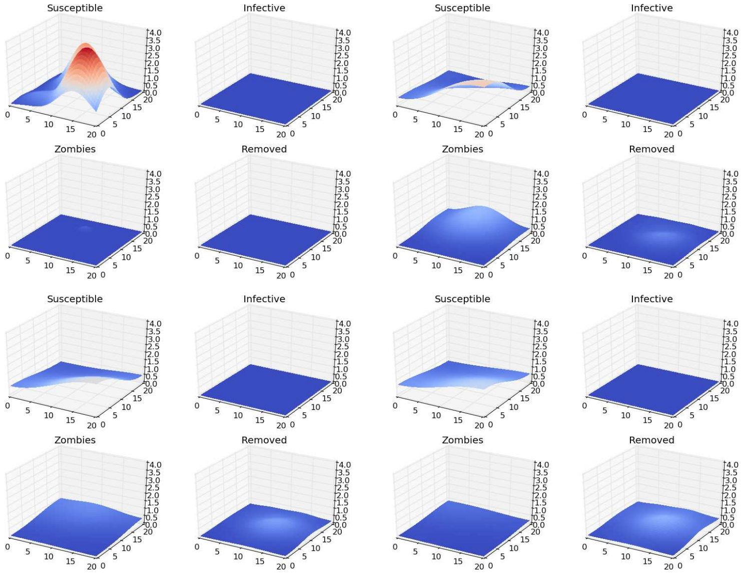


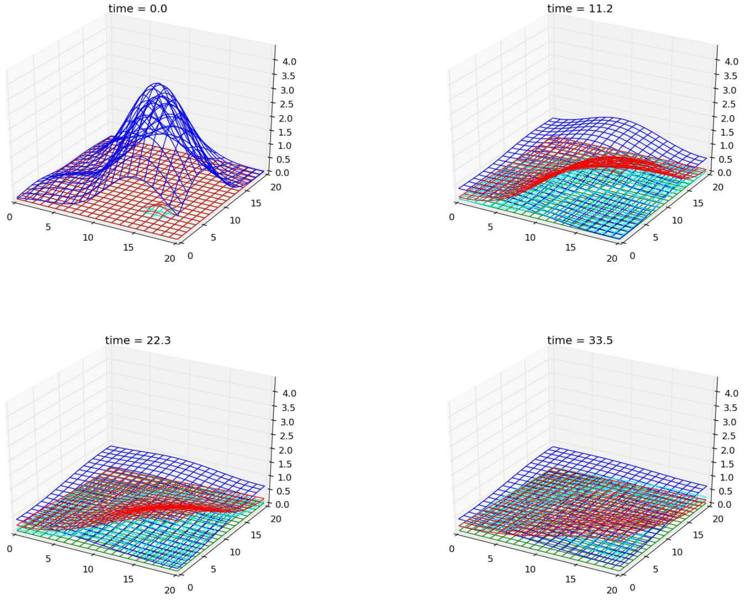
Figure 13: Creates the same results as for the ODE system



middle town.


Susceptible
Infective
Susceptible
Infective
Zombies
Removed
Zombies
Removed
Susceptible
Infective
Susceptible
Infective
Zombies
Removed
Zombies
Removed
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
0
5
10
15
20

large town.


time = 0.0
time = 11.2
time = 22.3
time = 33.5
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
0
5
10
15
20

References

[1] J.D. Murray. *Mathematical Biology II: Spatial Models and Biomedical Applications*. Intercisciplinary Applied Mathematics: Mathematical Biology. Springer, 2003.