SEIR Dynamics

October 14, 2021

[1]:
```
file = open("style.css")
styl = read(file, String)
HTML("$styl")
```

[1]:
```
HTML{String}("<link href='http://fonts.googleapis.com/css?family=Alegreya+Sans:1
00,300,400,500,700,800,900,100italic,300italic,400italic,500italic,700italic,800
italic,900italic' rel='stylesheet' type='text/css'>\r\n<link
href='http://fonts.googleapis.com/css?family=Arvo:400,700,400italic'
rel='stylesheet' type='text/css'>\r\n<link
href='http://fonts.googleapis.com/css?family=PT+Mono' rel='stylesheet'
type='text/css'>\r\n<link
href='http://fonts.googleapis.com/css?family=Shadows+Into+Light'
rel='stylesheet' type='text/css'>\r\n<link href='http://fonts.googleapis.com/css
?family=Philosopher:400,700,400italic,700italic' rel='stylesheet'
type='text/css'>\r\n\r\n<style>\r\n\r\n@font-face {\r\n    font-family:
\"Computer Modern\";\r\n    src: url('http://mirrors.ctan.org/fonts/cm-
unicode/fonts/otf/cmunss.otf');\r\n}\r\n\r\n\r\n\r\n/* Formatting for header
cells */\r\n.text_cell_render h1 {\r\n    font-family: 'Philosopher', sans-
serif;\r\n    font-weight: 400;\r\n    font-size: 2.2em;\r\n    line-height:
100%;\r\n    color: rgb(0, 80, 120);\r\n    margin-bottom: 0.1em;\r\n    margin-
top: 0.1em;\r\n    display: block;\r\n}\t\r\n.text_cell_render h2 {\r\n    font-
family: 'Philosopher', serif;\r\n    font-weight: 400;\r\n    font-size:
1.9em;\r\n    line-height: 100%;\r\n    color: rgb(200,100,0);\r\n    margin-
bottom: 0.1em;\r\n    margin-top: 0.1em;\r\n    display:
block;\r\n}\t\r\n\r\n.text_cell_render h3 {\r\n    font-family: 'Philosopher',
serif;\r\n    margin-top:12px;\r\n    margin-bottom: 3px;\r\n    font-style:
italic;\r\n    color: rgb(94,127,192);\r\n}\r\n\r\n.text_cell_render h4 {\r\n
font-family: 'Philosopher', serif;\r\n}\r\n\r\n.text_cell_render h5 {\r\n
font-family: 'Alegreya Sans', sans-serif;\r\n    font-weight: 300;\r\n    font-
size: 16pt;\r\n    color: grey;\r\n    font-style: italic;\r\n    margin-bottom:
.1em;\r\n    margin-top: 0.1em;\r\n    display:
block;\r\n}\r\n\r\n.text_cell_render h6 {\r\n    font-family: 'PT Mono', sans-
serif;\r\n    font-weight: 300;\r\n    font-size: 10pt;\r\n    color: grey;\r\n
margin-bottom: 1px;\r\n    margin-top: 1px;\r\n}\r\n\r\n.CodeMirror{\r\n
font-family: \"PT Mono\";\r\n        font-size:
100%;\r\n}\r\n\r\n</style>\r\n\r\n")
```

[5]:
```
import Pkg
Pkg.status()
```

```
Status
`C:\Users\JustinNouveau\.julia\environments\v1.5\Project.toml`
  [336ed68f] CSV v0.8.5
  [a93c6f00] DataFrames v0.22.7
  [0c46a032] DifferentialEquations v6.17.1
  [587475ba] Flux v0.12.1
  [7073ff75] IJulia v1.23.2
  [b1bec4e5] LIBSVM v0.6.0
  [eb30cadb] MLDatasets v0.5.6
  [add582a8] MLJ v0.16.0
  [d491faf4] MLJModels v0.14.1
  [872c559c] NNlib v0.7.19
  [1dea7af3] OrdinaryDiffEq v5.55.1
  [65888b18] ParameterizedFunctions v5.10.0
  [d96e819e] Parameters v0.12.2
  [91a5bcdd] Plots v1.15.2
  [c3e4b0f8] Pluto v0.14.8
  [7c2d2b1e] ReservoirComputing v0.5.0
  [44d3d7a6] Weave v0.10.9
  [37e2e46d] LinearAlgebra
  [9a3f8284] Random
  [2f01184e] SparseArrays
  [10745b16] Statistics
```

[4]:
```
#import Pkg; Pkg.add("Parameters")
# import Pkg; Pkg.add("LinearAlgebra")
# import Pkg; Pkg.add("Statistics")
# import Pkg; Pkg.add("Random")
# import Pkg; Pkg.add("SparseArrays")
```

```
  Resolving package versions…
Updating
`C:\Users\JustinNouveau\.julia\environments\v1.5\Project.toml`
  [37e2e46d] + LinearAlgebra
No Changes to
`C:\Users\JustinNouveau\.julia\environments\v1.5\Manifest.toml`
  Resolving package versions…
Updating
`C:\Users\JustinNouveau\.julia\environments\v1.5\Project.toml`
  [10745b16] + Statistics
No Changes to
`C:\Users\JustinNouveau\.julia\environments\v1.5\Manifest.toml`
  Resolving package versions…
Updating
`C:\Users\JustinNouveau\.julia\environments\v1.5\Project.toml`
```

```
  [9a3f8284] + Random
No Changes to
`C:\Users\JustinNouveau\.julia\environments\v1.5\Manifest.toml`
  Resolving package versions…
Updating
`C:\Users\JustinNouveau\.julia\environments\v1.5\Project.toml`
  [2f01184e] + SparseArrays
No Changes to
`C:\Users\JustinNouveau\.julia\environments\v1.5\Manifest.toml`
```

1 MODELE SEIR CF ARTICLE EN LIGNE

Ici nous considérons le modèle SEIR simple utilisé dans l'article sur la modélisation d'une épidémie partie 2

[10]:

```
import Pkg
Pkg.status()
```

```
Status
`C:\Users\JustinNouveau\.julia\environments\v1.5\Project.toml`
  [336ed68f] CSV v0.8.5
  [a93c6f00] DataFrames v0.22.7
  [0c46a032] DifferentialEquations v6.17.1
  [587475ba] Flux v0.12.1
  [7073ff75] IJulia v1.23.2
  [b1bec4e5] LIBSVM v0.6.0
  [eb30cadb] MLDatasets v0.5.6
  [add582a8] MLJ v0.16.0
  [d491faf4] MLJModels v0.14.1
  [872c559c] NNlib v0.7.19
  [1dea7af3] OrdinaryDiffEq v5.55.1
  [65888b18] ParameterizedFunctions v5.10.0
  [d96e819e] Parameters v0.12.2
  [91a5bcdd] Plots v1.15.2
  [c3e4b0f8] Pluto v0.14.8
  [7c2d2b1e] ReservoirComputing v0.5.0
  [44d3d7a6] Weave v0.10.9
  [37e2e46d] LinearAlgebra
  [9a3f8284] Random
  [2f01184e] SparseArrays
  [10745b16] Statistics
```

[10]:

```
using DifferentialEquations, Plots, DataFrames, CSV
```

Tous les paramètres sont là : , , , , , ,

[11]:
```
function SEIR_simple_ode(du, u, p, t)
    S,E,I,R = u
     , , = p
    du[1] = - *S*I
    du[2] =  *S*I -  *E
    du[3] =  *E -  *I
    du[4] =  *I
end
```

[11]:
```
SEIR_simple_ode (generic function with 1 method)
```

[12]:
```
function SEIR_mortal_natal_ode(du, u, p, t)
    S,E,I,R = u
     , , , , = p
    du[1] = - *S*I +  *(S+E+I+R) -  *S
    du[2] =  *S*I -  *E -  *E
    du[3] =  *E -  *I -  *I
    du[4] =  *I -  *R
end
```

[12]:
```
SEIR_mortal_natal_ode (generic function with 1 method)
```

[13]:
```
function SEIR_mortal_natal_immuno_ode(du, u, p, t)
    S,E,I,R = u
     , , , , , = p
    du[1] = - *S*I +  *(S+E+I+R) -  *S +  *R
    du[2] =  *S*I -  *E -  *E
    du[3] =  *E -  *I -  *I
    du[4] =  *I -  *R -  *R
end
```

[13]:
```
SEIR_mortal_natal_immuno_ode (generic function with 1 method)
```

[14]:
```
function SEIR_mortal_natal_vaccin_ode(du, u, p, t)
    S,E,I,R = u
     , , , , , = p
    du[1] = - *S*I +  *(S+E+I+R) -  *S -  *S
    du[2] =  *S*I -  *E -  *E
    du[3] =  *E -  *I -  *I
    du[4] =  *I -  *R +  *S
end
```

[14]:
```
SEIR_mortal_natal_vaccin_ode (generic function with 1 method)
```

[15]:
```
#  Valeur des paramètres pour les différents modèles
parms_seir_simple = [0.8,0.75, 0.05]
parms_seir_mortal_natal = [0.8,0.75, 0.05, 0.009, 0.01]
```

```
parms_seir_mortal_natal_immuno = [0.8,0.75, 0.05, 0.009, 0.01, 0.04]
parms_seir_mortal_natal_vaccin = [0.8,0.75, 0.05, 0.009, 0.01, 0.06]
init = [0.99,0.01,0.00,0.0] #Conditions initiales
tspan = (0.0,50.0) #  100 jours
prob = ODEProblem(SEIR_simple_ode,init,tspan,parms_seir_simple)
# prob_1 = ODEProblem(SEIR_mortal_natal_ode,init,tspan,parms_seir_mortal_natal)
# prob_2 =␣
 ↪ODEProblem(SEIR_mortal_natal_immuno_ode,init,tspan,parms_seir_mortal_natal_immuno)
# prob_3 =␣
 ↪ODEProblem(SEIR_mortal_natal_vaccin_ode,init,tspan,parms_seir_mortal_natal_vaccin)
```

[15]:

```
ODEProblem with uType Array{Float64,1} and tType
Float64. In-place: true
timespan: (0.0, 50.0)
u0: 4-element Array{Float64,1}:
 0.99
 0.01
 0.0
 0.0
```

2 PARAMETRES

Nous utilisons ici les paramètres pris sur le document machine dans le modèle SEIR avec présence de cycle **Ressource**

[16]:

```
# parametres obtenus sur le document machine
parms_seir_mortal_natal = [3/14,1/7, 1/14, 1/(365*76), 1/(365*76)]
parms_seir_mortal_natal_immuno = [3/14,1/7, 1/14, 1/(365*76), 1/(365*76), 1/365]
init = [0.9999995,4.0e-7 ,1.0e-7,0.00] #Conditions initiales
tspan = (0.0,1095.0) #  1095 jours /365 jours = 3 années
prob_2 =␣
 ↪ODEProblem(SEIR_mortal_natal_immuno_ode,init,tspan,parms_seir_mortal_natal_immuno)
prob_1 = ODEProblem(SEIR_mortal_natal_ode,init,tspan,parms_seir_mortal_natal)
sir_sol_2 = solve(prob_2,saveat = 0.1)
sir_sol_1 = solve(prob_1,saveat = 0.1)
```

[16]:

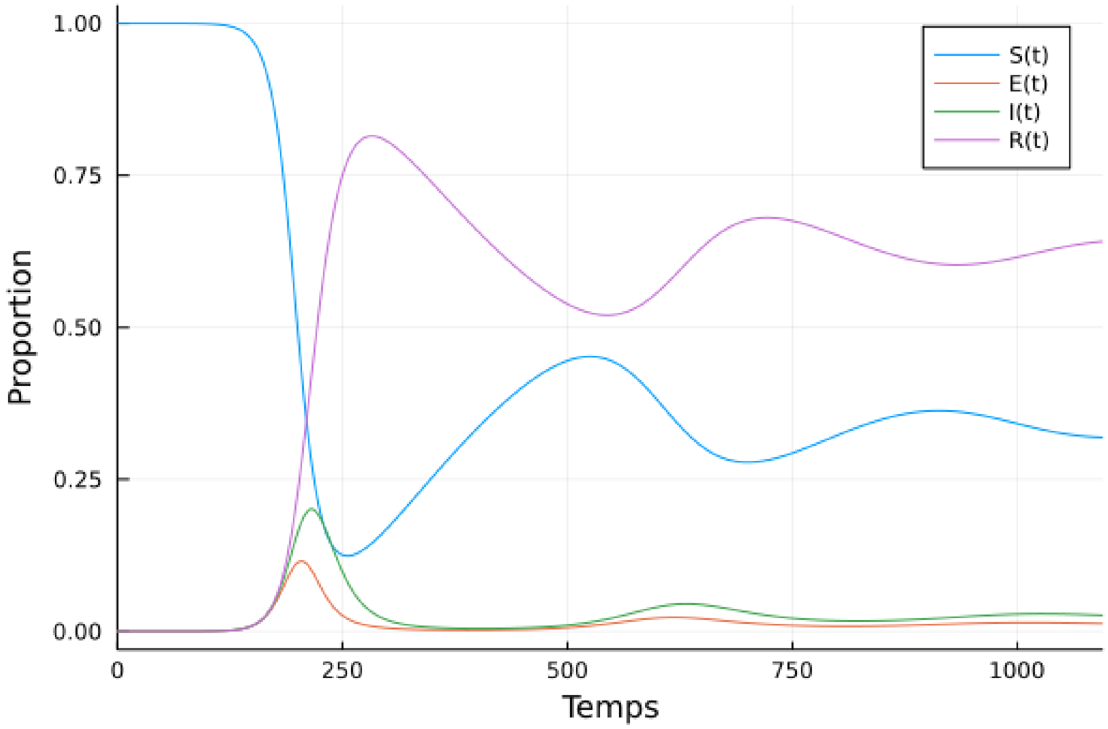
```
retcode: Success
Interpolation: 1st order linear
t: 10951-element Array{Float64,1}:
    0.0
    0.1
    0.2
    0.3
    0.4
    0.5
    0.6
    0.7
    0.8
    0.9
    1.0
    1.1
    1.2

 1093.9
 1094.0
 1094.1
 1094.2
 1094.3
 1094.4
 1094.5
 1094.6
 1094.7
 1094.8
 1094.9
 1095.0
u: 10951-element Array{Array{Float64,1},1}:
 [0.9999995, 4.0e-7, 1.0e-7, 0.0]
 [0.9999994978056939, 3.965055337003434e-7, 1.0495673587519686e-7,
7.320365620440317e-10]
 [0.9999994955060862, 3.9316520281999436e-7, 1.0982953734706415e-7,
1.4991736577969513e-9]
 [0.9999994931029415, 3.8997507395926293e-7, 1.146211615274221e-7,
2.3008231432409915e-9]
 [0.9999994905979696, 3.869313195639944e-7, 1.193342959456981e-7,
3.136415005950164e-9]
 [0.9999994879928262, 3.840302179255692e-7, 1.2397155854892693e-7,
4.005397365090322e-9]
 [0.9999994852891126, 3.812681531809031e-7, 1.285354977017504e-7,
4.907236471419085e-9]
 [0.9999994824883759, 3.7864161531244726e-7, 1.3302859218641734e-7,
5.841416707285821e-9]
 [0.9999994795921086, 3.76147199206118e-7, 1.3745325183164804e-7,
6.807440430028674e-9]
 [0.9999994766027644, 3.737795703439389e-7, 1.4181317541628345e-7,
7.8044897698415e-9]
 [0.9999994735224059, 3.7153417629649573e-7, 1.4611154050662997e-7,
8.831877295047464e-9]
 [0.9999994703520164, 3.6940862426038376e-7, 1.503500841569356e-7,
9.88927513124597e-9]
 [0.9999994670925644, 3.6740054874706913e-7, 1.5453052636830273e-7,
1.097636053490577e-8]
```

```
 [0.08780876322251749, -1.7701824391708504e-11, 2.7613623832147496e-11,
0.9121912367675701]
 [0.08781205157728399, -1.7452356486674652e-11, 2.7252767468033923e-11,
0.9121879484129151]
 [0.08781533992018505, -1.7201650012777096e-11, 2.6889999428515594e-11,
0.9121846600701261]
 [0.08781862825122067, -1.6949704970015253e-11, 2.6525319713591683e-11,
0.9121813717392032]
 [0.08782191657039083, -1.6696521358390236e-11, 2.6158728323263818e-11,
0.9121780834201465]
 [0.08782520487769557, -1.644209917790092e-11, 2.5790225257530318e-11,
0.912174795112956]
 [0.08782849317313485, -1.618643842854847e-11, 2.5419810516392915e-11,
0.912171506817631 5]
 [0.0878317814567087, -1.5929539110332247e-11, 2.504748409985068e-11,
0.9121682185341731]
 [0.08783506972841709, -1.567140122325174e-11, 2.4673246007902835e-11,
0.9121649302625808]
 [0.08783835798826005, -1.5412024767308072e-11, 2.4297096240551063e-11,
0.9121616420028548]
 [0.08784164623623757, -1.515140974250009e-11, 2.391903479779368e-11,
0.9121583537549947]
 [0.08784493447234963, -1.4889556148828987e-11, 2.3539061679632398e-11,
0.9121550655190008]
```

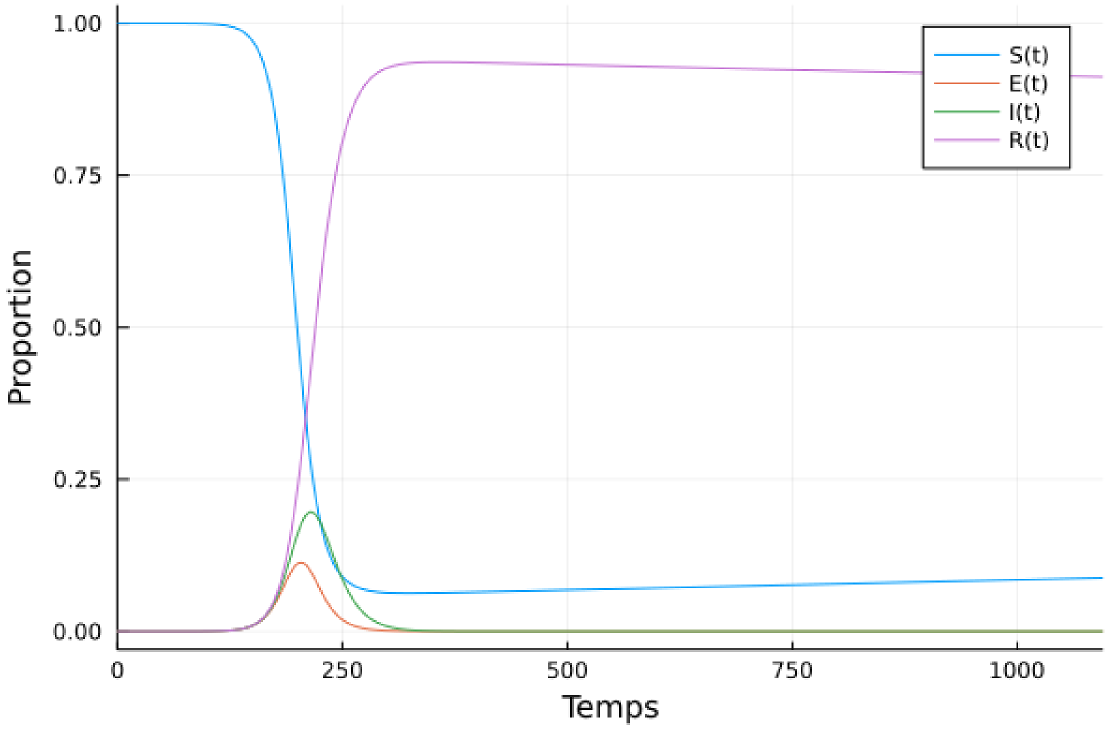
[17]:
```
plot(sir_sol_2,xlabel="Temps",ylabel="Proportion", label = ["S(t)" "E(t)"␣
 ↪"I(t)" "R(t)"])
# savefig("CYCLE_seir_mortal_natal_immuno.png")
```

[17]:



[18]:
```
plot(sir_sol_1,xlabel="Temps",ylabel="Proportion", label = ["S(t)" "E(t)"␣
 ↪"I(t)" "R(t)"])
# savefig("CYCLE_seir_mortal_natal.png")
```

[18]:



[19]:

```
begin
        Saint=[]
    Expo = []
        Infe=[]
        Rec=[]
        for i in 1:length(sir_sol_2)
                push!(Saint,sir_sol_2[i][1])
                push!(Expo,sir_sol_2[i][2])
                push!(Infe,sir_sol_2[i][3])
        push!(Rec,sir_sol_2[i][4])
        end
end
```

[20]:

```
df = DataFrame(Saints=Saint,
               Exposes=Expo,
               Infectieux = Infe,
               Gueris=Rec
               )
```

[20]:

	Saints	Exposes	Infectieux	Gueris
	Any	Any	Any	Any
1	1.0	4.0e-7	1.0e-7	0.0
2	0.999999	3.96506e-7	1.04957e-7	7.31937e-10
3	0.999999	3.93165e-7	1.0983e-7	1.49877e-9
4	0.999999	3.89975e-7	1.14621e-7	2.2999e-9
5	0.999999	3.86931e-7	1.19334e-7	3.13475e-9
6	0.999999	3.8403e-7	1.23972e-7	4.00275e-9
7	0.999999	3.81268e-7	1.28535e-7	4.90337e-9
8	0.999999	3.78642e-7	1.33029e-7	5.83608e-9
9	0.999999	3.76147e-7	1.37453e-7	6.80038e-9
10	0.999999	3.7378e-7	1.41813e-7	7.79542e-9
11	0.999999	3.71534e-7	1.46112e-7	8.82053e-9
12	0.999999	3.69409e-7	1.5035e-7	9.87535e-9
13	0.999999	3.67401e-7	1.54531e-7	1.09596e-8
14	0.999999	3.65508e-7	1.58655e-7	1.20728e-8
15	0.999999	3.63728e-7	1.62724e-7	1.32149e-8
16	0.999999	3.62058e-7	1.6674e-7	1.43853e-8
17	0.999999	3.60497e-7	1.70705e-7	1.55839e-8
18	0.999999	3.59041e-7	1.7462e-7	1.68103e-8
19	0.999999	3.5769e-7	1.78487e-7	1.80642e-8
20	0.999999	3.5644e-7	1.82308e-7	1.93454e-8
21	0.999999	3.5529e-7	1.86083e-7	2.06535e-8
22	0.999999	3.54238e-7	1.89815e-7	2.19882e-8
23	0.999999	3.5328e-7	1.93506e-7	2.33493e-8
24	0.999999	3.52416e-7	1.97156e-7	2.47365e-8
25	0.999999	3.51643e-7	2.00767e-7	2.61496e-8
26	0.999999	3.50959e-7	2.04341e-7	2.75882e-8
27	0.999999	3.50363e-7	2.07879e-7	2.90521e-8
28	0.999999	3.49852e-7	2.11382e-7	3.05411e-8
29	0.999999	3.49424e-7	2.14853e-7	3.2055e-8
30	0.999999	3.49077e-7	2.18293e-7	3.35933e-8
…	…	…	…	…

[104]:
```
CSV.write("data_seirs.csv", df)
```

[104]:
```
"data_seirs.csv"
```

3 FIN

[21]:
```
sir_sol = solve(prob,saveat = 1)
sir_sol_1 = solve(prob_1,saveat = 0.001)
sir_sol_2 = solve(prob_2,saveat = 1)
sir_sol_3 = solve(prob_3,saveat = 1)
```

```
[21]: retcode: Success
      Interpolation: 1st order linear
      t: 51-element Array{Float64,1}:
         0.0
         1.0
         2.0
         3.0
         4.0
         5.0
         6.0
         7.0
         8.0
         9.0
        10.0
        11.0
        12.0

        39.0
        40.0
        41.0
        42.0
        43.0
        44.0
        45.0
        46.0
        47.0
        48.0
        49.0
        50.0
      u: 51-element Array{Array{Float64,1},1}:
       [0.99, 0.01, 0.0, 0.0]
       [0.9295101931409542, 0.0064675700582942745, 0.005577262241243603,
      0.05744547439288293]
       [0.8699349936208816, 0.0070630746699959475, 0.01003760402142444,
      0.11096632635503113]
       [0.8116232293749709, 0.009346845924283084, 0.015359043915367195,
      0.16067537628875186]
       [0.7540926040183019, 0.012724817348904755, 0.022449573429140897,
      0.206740994547644]
       [0.6966570285937379, 0.017078829274710626, 0.031946283297506206,
      0.2493303380267277]
       [0.6386651611192825, 0.02235156666462771, 0.04440405823114653,
      0.28859717803887863]
       [0.5796569379263959, 0.028406651976965295, 0.06027532185564471,
      0.3246855311742292]
       [0.5195221463088686, 0.03494257435861995, 0.07982853648673141,
      0.3577386576828407]
```

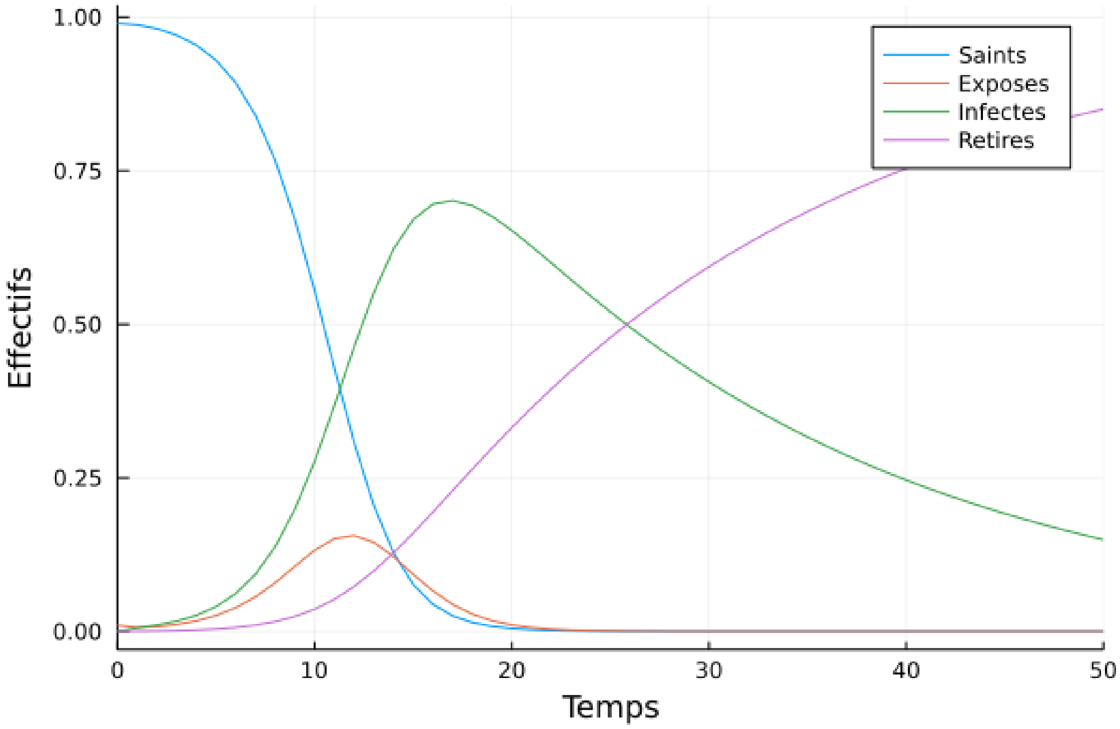
```
 [0.4586400850944519, 0.04146032466336983, 0.10302803509200256,
0.38791193392305934]
 [0.3979513602202938, 0.04729238485548522, 0.1294200115528834,
0.4153860771205057]
 [0.33889399957423116, 0.05174181992743028, 0.15804599843300357,
0.4403784608407037]
 [0.28320335343070707, 0.05419121467108709, 0.18753666515107914,
0.4631404796090573]

 [0.03746466152291716, 0.007180087249216008, 0.1789291694277908,
0.7381767909464428]
 [0.03813896623093289, 0.007088555850392772, 0.17370274136273162,
0.741859175708266]
 [0.038818931376128296, 0.007001049037869848, 0.1687179016443341,
0.7452912478894668]
 [0.03950146320459351, 0.006923191650425587, 0.16395721583826475,
0.7484879098792007]
 [0.040184921516089045, 0.006845487061728514, 0.1594188153077653,
0.7514621661814478]
 [0.04086753221589238, 0.006766965424083719, 0.15509263623139724,
0.7542268236016734]
 [0.04154724960620684, 0.006696175798415463 5, 0.15095946524233422,
0.7567945911861435]
 [0.04222312023688604, 0.006627869692546016, 0.1470144595642224,
0.7591765126970602]
 [0.04289442548423481, 0.006556517490937356, 0.14325350690677632,
0.7613829477084227]
 [0.043559875718201475, 0.006488381876879929, 0.13966108484904016,
0.7634244446333832]
 [0.044218629432094536, 0.00642454534934579, 0.13622735812082912,
0.7653105967962355]
 [0.04487025313820768, 0.006361361122791482, 0.1329476920951202,
0.7670501181445947]
```

[22]:

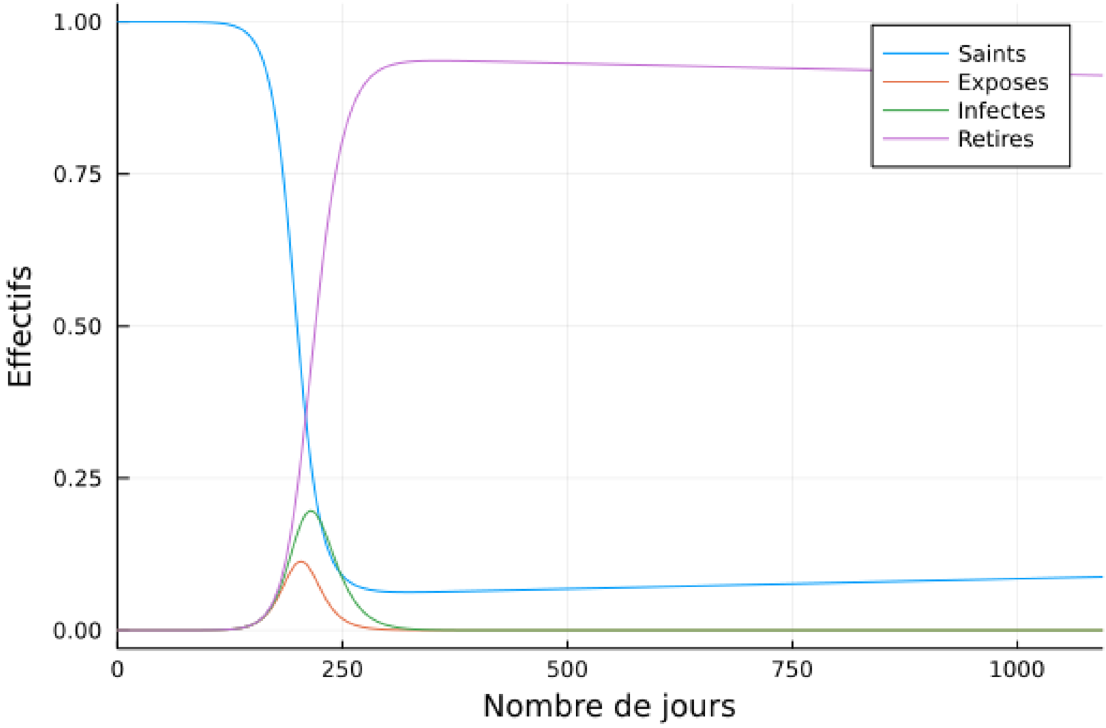
```
plot(sir_sol,xlabel="Temps",ylabel="Effectifs", label = ["Saints" "Exposes"␣
↪"Infectes" "Retires"])
# savefig("seir_simple.png")
```

[22]:



```
[23]: plot(sir_sol_1,xlabel="Nombre de jours",ylabel="Effectifs", label = ["Saints"␣
       ↪"Exposes" "Infectes" "Retires"])
      #savefig("seir_mortal_natal.png")
```

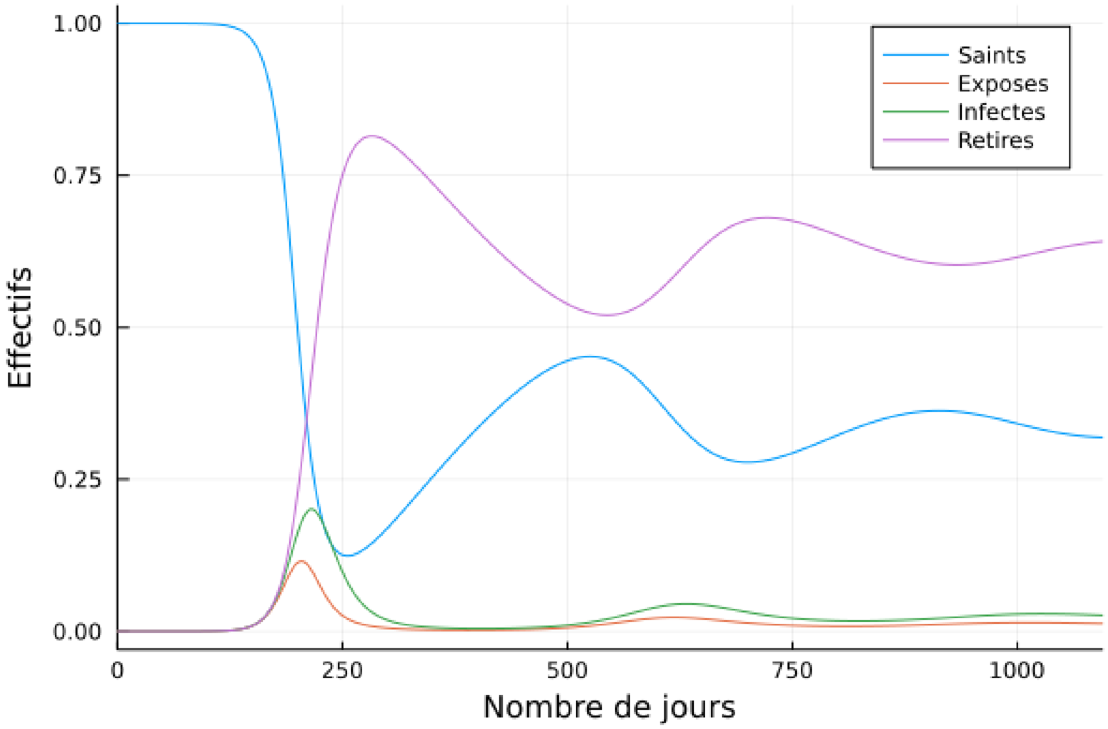
```
[23]:
```



[24]:

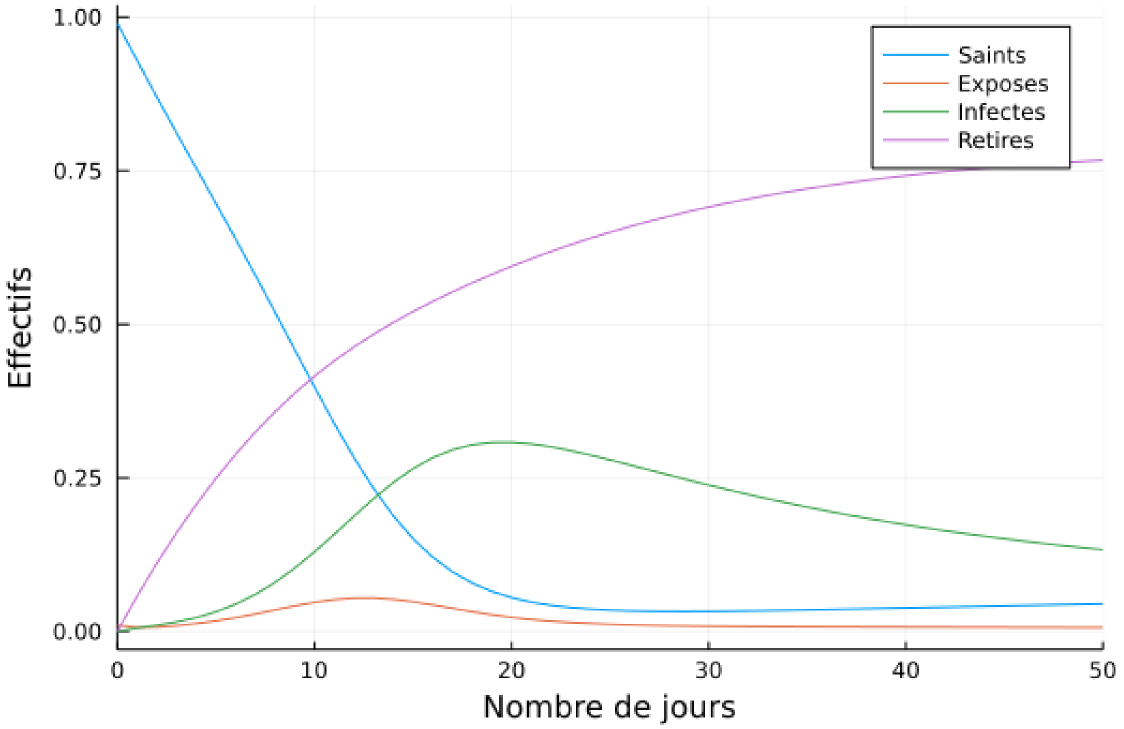
```
plot(sir_sol_2,xlabel="Nombre de jours",ylabel="Effectifs", label = ["Saints"␣
 ↪"Exposes" "Infectes" "Retires"])
#savefig("seir_mortal_natal_immuno.png")
```

[24]:



```
[25]: plot(sir_sol_3,xlabel="Nombre de jours",ylabel="Effectifs", label = ["Saints"␣
       ↪"Exposes" "Infectes" "Retires"])
      #savefig("seir_mortal_natal_vaccin.png")
```

```
[25]:
```



4 Dataframe

[32]:
```
begin
        Saint=[]
    Expo = []
        Infe=[]
        Rec=[]
        for i in 1:length(sir_sol_1)
                push!(Saint,sir_sol_1[i][1])
                push!(Expo,sir_sol_1[i][2])
                push!(Infe,sir_sol_1[i][3])
        push!(Rec,sir_sol_1[i][4])
        end
end
```

[33]:
```
df = DataFrame(Saints=Saint,
               Exposes=Expo,
               Infectieux = Infe,
               Gueris=Rec
               )
```

[33]:

	Saints	Exposes	Infectieux	Gueris
	Any	Any	Any	Any
1	0.99	0.01	0.0	0.0
2	0.989999	0.0099924	7.49693e-6	1.87447e-10
3	0.989998	0.00998482	1.49877e-5	7.49582e-10
4	0.989997	0.00997724	2.24724e-5	1.68609e-9
5	0.989996	0.00996967	2.99509e-5	2.99667e-9
6	0.989995	0.00996211	3.74233e-5	4.68101e-9
7	0.989995	0.00995456	4.48895e-5	6.73879e-9
8	0.989994	0.00994701	5.23497e-5	9.16971e-9
9	0.989993	0.00993948	5.98038e-5	1.19735e-8
10	0.989992	0.00993195	6.72518e-5	1.51497e-8
11	0.989991	0.00992444	7.46937e-5	1.86982e-8
12	0.98999	0.00991693	8.21295e-5	2.26186e-8
13	0.989989	0.00990943	8.95592e-5	2.69106e-8
14	0.989988	0.00990194	9.69829e-5	3.15739e-8
15	0.989987	0.00989445	0.0001044	3.66081e-8
16	0.989986	0.00988698	0.000111812	4.20131e-8
17	0.989985	0.00987951	0.000119218	4.77884e-8
18	0.989984	0.00987206	0.000126617	5.39337e-8
19	0.989983	0.00986461	0.00013401	6.04489e-8
20	0.989982	0.00985717	0.000141398	6.73335e-8
21	0.989981	0.00984974	0.000148779	7.45872e-8
22	0.98998	0.00984231	0.000156155	8.22098e-8
23	0.989979	0.0098349	0.000163524	9.02009e-8
24	0.989979	0.00982749	0.000170887	9.85602e-8
25	0.989978	0.0098201	0.000178245	1.07288e-7
26	0.989977	0.00981271	0.000185596	1.16382e-7
27	0.989976	0.00980533	0.000192941	1.25845e-7
28	0.989975	0.00979796	0.000200281	1.35674e-7
29	0.989974	0.0097906	0.000207614	1.4587e-7
30	0.989973	0.00978324	0.000214942	1.56432e-7
...	...	...	...	...

[35]:

```
# CSV.write("data_simulated_seir_mortal_natal_0.001_v2.csv", df)
```

[35]:

```
"data_simulated_seir_mortal_natal_0.001_v2.csv"
```