This notebook accompanies the article:

Dynamics of COVID-19 pandemic at constant and time-dependent contact rates

by Marek Kochańczyk, Frederic Grabowski, and Tomasz Lipniacki
published in *Mathematical Modeling of Natural Phenomena* (2020)

Last update: **March 28, 2020**. This notebook is compatible with Mathematica **version 12**.

The notebook is divided into three parts: (1) generation of model equations together with computation of a consistent initial value; (2) generation of plots included in the manuscript and its supplementary materials; and (3) interactive simulation. For convenient navigation, most cell groups were collapsed: in order to see their contents, you have to double click on corresponding “square brackets” on the right-hand side of the notebook.

Parameter values and model definition, as well as cells with shared functionalities required to generate plots, are contained within "initialization cells". It is advised to evaluate all initialization cells right after opening the notebook. (If evaluation takes more than a few seconds and seems to be never halting, then quite likely your Mathematica version is older than version 12 and you have to upgrade to be able to properly use this notebook.) If you want to evaluate the whole notebook, please note that it may take about 1 minute -- compute-intensive part is generation of plots that require multiple rounds of ODE integration.

Model

Model parameters

In[∘]:=
```
kExposedSteps = 5;      (* This parameter defines model structure. *)
mInfectiousSteps = 1;  (* This parameter defines model structure. *)
```

In[∘]:=
```
defaultParameterValues = {
   rC → 1 / 2,              (* contact rate [1/day] *)
   rE → 1 / 3,              (* removal rate (a.k.a exclusion rate) [1/day] *)
   τ → 5,                   (* incubation period [day] *)

   PopulationN → 50 * 10^6, (* number of individuals in population *)
   Removed0 → 500,          (* number of Removed individuals at t=0;
                               other initial values are adjusted to this value *)

   selfisolation → 0,       (* 0 in the model with constant rC,
                               1 in the model with "fear effect" *)
   H → 10^4,                (* when selfisolation → 1 and the number of daily new
                               cases begins to suprass H, people reduce rC twice *)

   negligence → 0,          (* 1 in the model with "negligence effect" with H = H(t),
                               0 in the model with H constant in time;
                               note: when negligence → 1, also should be isolation → 1 *)
   Tnegl → 3 * 30           (* H doubling time [day] *)
};
```

Model equations

In[∘]:=
```
SymbolSeries[basename_, num_] := Table[Symbol[basename <> ToString[i]], {i, num}];
modifiedParameterValues[replacements_] := Fold[#1 /. (First@#2 → _) → #2 &, defaultParameterValues, replacements];
```

In[ ]:=
```
exposedNames = SymbolSeries["Exposed", kExposedSteps];
firstExposed = exposedNames[[1]];
lastExposed = exposedNames[[kExposedSteps]];
sumExposed[t] = Total[#[t]& /@ exposedNames];

infectiousNames = SymbolSeries["Infectious", mInfectiousSteps];
firstInfectious = infectiousNames[[1]];
lastInfectious = infectiousNames[[mInfectiousSteps]];
sumInfectious[t] = Total[#[t]& /@ infectiousNames];

variableNames = Flatten@{Susceptible, exposedNames, infectiousNames, Removd};   (* 'Removed' is Wolfram language function name *)
```

In[ ]:=
```
equationsBase = Join[
  { Susceptible'[t] == -fS[t] * rC * sumInfectious[t],
    firstExposed'[t] == fS[t] * rC * sumInfectious[t] - rIpart * firstExposed[t] },  Table[
    exposedNames[[i]]'[t] == rIpart * exposedNames[[i - 1]][t] - rIpart * exposedNames[[i]][t],  {i, 2, kExposedSteps}], {
    firstInfectious'[t] == rIpart * lastExposed[t] - rEpart * firstInfectious[t] },  Table[
    infectiousNames[[i]]'[t] ==  rEpart * infectiousNames[[i - 1]][t] - rEpart * infectiousNames[[i]][t],  {i, 2, mInfectiousSteps}], {
    Removd'[t] == rEpart * lastInfectious[t] }
] /. {
rIpart → kExposedSteps * τ^-1,
rEpart → mInfectiousSteps * rE};

equationsBase // TableForm
```

Out[ ]//TableForm=

$$\text{Susceptible}'[t] == -\text{rC}\,\text{fS}[t] \times \text{Infectious1}[t]$$
$$\text{Exposed1}'[t] == -\frac{5\,\text{Exposed1}[t]}{\tau} + \text{rC}\,\text{fS}[t] \times \text{Infectious1}[t]$$
$$\text{Exposed2}'[t] == \frac{5\,\text{Exposed1}[t]}{\tau} - \frac{5\,\text{Exposed2}[t]}{\tau}$$
$$\text{Exposed3}'[t] == \frac{5\,\text{Exposed2}[t]}{\tau} - \frac{5\,\text{Exposed3}[t]}{\tau}$$
$$\text{Exposed4}'[t] == \frac{5\,\text{Exposed3}[t]}{\tau} - \frac{5\,\text{Exposed4}[t]}{\tau}$$
$$\text{Exposed5}'[t] == \frac{5\,\text{Exposed4}[t]}{\tau} - \frac{5\,\text{Exposed5}[t]}{\tau}$$
$$\text{Infectious1}'[t] == \frac{5\,\text{Exposed5}[t]}{\tau} - \text{rE}\,\text{Infectious1}[t]$$
$$\text{Removd}'[t] == \text{rE}\,\text{Infectious1}[t]$$

Initial condition

Initial conditions are calculated so that they lie on the exponential trajectory, meaning that there is no change of derivative at $t = 0$. Numbers of all individuals at $t = 0$ are adjusted to a given number of Removed individuals.

Technical note: essentially, initial conditions could be prescribed using analytical expressions; however, for m (the number of infectious steps) > 1, they are hard to generalize, so for convenience we rely on numerical calculations.

In[ ]:=

```
E0s = SymbolSeries["E0", kExposedSteps];
I0s = SymbolSeries["I0", mInfectiousSteps];
exposedAnsatzen[i_] := Function[t, E0s[[i]] * Exp[α t]];
infectiousAnsatzen[i_] := Function[t, I0s[[i]] * Exp[α t]];
ansatzRepl = Flatten@Join[{
   fS[t] → 1 (* assumed constant => no depletion of susceptibles *),
   Susceptible → Function[t, S0 Exp[α t]],
   Removd → Function[t, R0 Exp[α t]]},
  Table[exposedNames[[i]] → exposedAnsatzen[i], {i, kExposedSteps}],
  Table[infectiousNames[[i]] → infectiousAnsatzen[i], {i, mInfectiousSteps}]
];

ansatzEqs = equationsBase /. ansatzRepl;
ansatzAsmpts = {rC > 0, rE > 0, τ > 0, rC > rE, α > 0, R0 == Removed0, Removed0 > 0};
ansatzToSolve = ansatzEqs ⋃ ansatzAsmpts;

Off[NSolve::ratnz]
initValuesWithParams[pars_] := Block[
  { initValues = First@NSolve[ansatzToSolve /. pars, E0s ⋃ I0s ⋃ {α, R0, S0}, ℝ]
  , adjustedSusceptible = (S0 → s_) → (S0 → (s + PopulationN /. pars))
  , initVariables = #[0]& /@ variableNames },
  initVariables /. ansatzRepl /. (initValues /. adjustedSusceptible)
];

initCondsWithParams[pars_] := Thread[#[0]& /@ variableNames == initValuesWithParams[pars]]
initCondsWithParamsAsRules[pars_] := Thread[(#[0]& /@ variableNames) → initValuesWithParams[pars]]

(* initCondsWithParams(*AsRules*)[defaultParameterValues] // TableForm *)
```

Final system of equations

At this point, we amend the system with depletion of susceptibles and glue it with initial conditions.

In[ ]:=
```
systemWithParams[pars_, repls_:{
  fS[t] → (Susceptible[t] / PopulationN / (1 + selfisolation*rE*sumInfectious[t]/H[t])),
  H[t] → ((1 - negligence)*H + negligence*H*(Tnegl + t)/Tnegl)
}] := (equationsBase //. repls /. pars) ~Join~ initCondsWithParams[pars]

systemWithParams[modifiedParameterValues[{selfisolation → 1, negligence → 1}]] // TableForm
```

Out[ ]//TableForm=

$$\text{Susceptible}'[t] == -\frac{\text{Infectious1}[t]\times\text{Susceptible}[t]}{100000000\left(1+\frac{3\,\text{Infectious1}[t]}{1000\,(90+t)}\right)}$$

$$\text{Exposed1}'[t] == -\text{Exposed1}[t] + \frac{\text{Infectious1}[t]\times\text{Susceptible}[t]}{100000000\left(1+\frac{3\,\text{Infectious1}[t]}{1000\,(90+t)}\right)}$$

$$\text{Exposed2}'[t] == \text{Exposed1}[t] - \text{Exposed2}[t]$$

$$\text{Exposed3}'[t] == \text{Exposed2}[t] - \text{Exposed3}[t]$$

$$\text{Exposed4}'[t] == \text{Exposed3}[t] - \text{Exposed4}[t]$$

$$\text{Exposed5}'[t] == \text{Exposed4}[t] - \text{Exposed5}[t]$$

$$\text{Infectious1}'[t] == \text{Exposed5}[t] - \frac{\text{Infectious1}[t]}{3}$$

$$\text{Removd}'[t] == \frac{\text{Infectious1}[t]}{3}$$

$$\text{Susceptible}[0] == 4.99993\times 10^{7}$$

$$\text{Exposed1}[0] == 37.7208$$

$$\text{Exposed2}[0] == 35.8237$$

$$\text{Exposed3}[0] == 34.022$$

$$\text{Exposed4}[0] == 32.3108$$

$$\text{Exposed5}[0] == 30.6858$$

$$\text{Infectious1}[0] == 79.4369$$

$$\text{Removd}[0] == 500.$$

Analytical asymptotic infection rate

Derivation: solve for contact rate, r_C , as a function of α, τ, and r_E:

In[ ]:=
```
rcExpr = Block[{
  rcExprPrelim = rC /. First @ Solve[Union[DivideSides[ansatzEqs, Exp[α t]], ansatzAsmpts], {rC, S0, R0} ⋃ E0s ⋃ I0s, ℝ]},
  Refine[FullSimplify[rcExprPrelim], ansatzAsmpts]]
```

Out[ ]=
$$\frac{(rE + \alpha)\ (5 + \alpha\,\tau)^5}{3125}$$

We can use this equation to calculate β as a function of α, τ, r_E, and r_C:

In[ ]:=
```
alpha[pars_] := Max[α /. Solve[rC == rcExpr /. pars, α, ℝ]]
beta[pars_] := Exp[alpha[pars]]
betaFromAlpha[alpha_] := Exp[alpha]
alphaFromBeta[beta_] := Log[beta]

quarantineR[alpha_, pars_] := rcExpr/rE /. pars /. {α → alpha}
```

Figures

Common plotting and data exporting utilities

In[ ]:=
```
plotDynamics[plotfun_, plotvars_, pars_, until_] := Block[{
  sol = Flatten@NDSolve[systemWithParams[pars], variableNames, {t, 0, until}]},
  deltaRemoved[t_ /; t > 1] := Removd[t] - Removd[t - 1] /. sol;

  worstDay = NMaximize[{deltaRemoved[t], 1 ≤ t, t ≤ until}, {t}];
  Print["Daily new cases in the worst day: fraction: ", worstDay[[1]] / (PopulationN /. pars)];
  Print["Daily new cases in the worst day: time to: ", t /. worstDay[[2]], " days"];
  Print["Final fraction removed: ", Removd[until] / (PopulationN /. pars) /. sol ];

  plotfun[
   Evaluate[plotvars /. sol],
   {t, 0, until},
   PlotLabels → plotvars,
   AxesLabel → {"Time [day]", "Number of cases"},
```

```
    ImageSize → Large
  ]
]

plotDailyNewCases[pars_, until_:6*12*30, ymaxfrac_:0.002] := Block[{
  sol = Flatten@NDSolve[systemWithParams[pars], variableNames, {t, 0, until}]},
  Plot[
    Evaluate[(sumInfectious[t]/. sol)*(rE /. pars)], {t, 0, until},
     PlotRange → {{0, until}, {0, ymaxfrac*(PopulationN /. pars)}},
    AxesLabel → {"Time [day]", "Daily new cases"},
    PlotStyle → Directive[RGBColor[0.4,0,.4]],
    ImageSize → Large
  ]
]

SwappedAxes = # /. {
 x_Line | x_GraphicsComplex ⧴ MapAt[# ~Reverse~ 2 &, x, 1],
  x: ((AxesLabel→ _) | (PlotRange -> _)) ⧴ x ~Reverse~ 2
} &;

plotToCsv[plot_, basename_] := Block[{
  data = Cases[plot, Line[data_] :> data, Infinity],
  savedir = "/tmp/"   (* mind the final '/' *) },
  Table[
    Export[savedir <> basename <> "_part_" <> ToString@i <> ".csv", data〚i〛],
    {i, 1, Length@data}
  ]
]
```

In[∘]:=

```
fractionRemoved0[pars_ /; ((rC > rE) /. pars)] := Removd[t] / (Removd[t] + sumExposed[t] + sumInfectious[t]) /. t → 0 /. initCondsWithParamsAsRules[pars]
```

Worst day/week/month

In[ ]:=
```
maximumRemoved[pars_, days_] := Block[
  { T = {t, 0, ∞}
  , verySlowRate = 10^-12* (PopulationN /. pars)
  , end = None
  , stop = WhenEvent[(Removd'[t] < verySlowRate) && (t > 60), end = t; "StopIntegration"]
  , systemWithStop = systemWithParams[pars] ⋃ {stop}
  , solution = Quiet@NDSolve[systemWithStop, variableNames, Evaluate@T]
  , removed = Function[t, First[Removd[t] /. solution]]
  , removedDeltaDay = Function[t, removed[t] - removed[t - days]]
  },
  NMaximize[{removedDeltaDay[s], days ≤ s, s ≤ end}, s]
]
maxRemovedCount[pars_, days_] := First @ maximumRemoved[pars, days]
maxRemovedTime[pars_, days_] := Nest[Last, maximumRemoved[pars, days], 3]
```

In[ ]:=
```
infectiousMaxValueHalfmaxTime[pars_] := Quiet @ Block[
  { T = {t, 0, ∞}
  , verySlowRate = 10^-12* (PopulationN /. pars)
  , end = None
  , stop = WhenEvent[(Removd'[t] < verySlowRate) && (t > 60), end = t; "StopIntegration"]
  , systemWithStop = systemWithParams[pars] ⋃ {stop}
  , solution = NDSolve[systemWithStop, variableNames, Evaluate@T]
  , infectious = Total[#[t] /. First[solution] & /@ infectiousNames]
  , m = NMaximize[{infectious, t > 0}, {t}]
  , maxValue = First @ m, (* max Infectious *)
  , maxTime = t /. Last[m]
  , halfMaxValue = maxValue/2
  , hm = NMinimize[{(halfMaxValue - infectious)^2, t > maxTime}, {t}]
  , halfmaxTime = t /. Last[hm]},
  { maxValue, halfmaxTime }
]
removedMaxCountAlt[pars_] := First[infectiousMaxValueHalfmaxTime[pars]] * (rE /. pars)
removedHalfmaxTime[pars_] := Last[infectiousMaxValueHalfmaxTime[pars]]
```

Figure 2

Fig. 2A

In[ ]:=
```
plot = Plot[
  { quarantineR[alphaFromBeta[varbeta], modifiedParameterValues[{rE → 1/5}]]
  , quarantineR[alphaFromBeta[varbeta], modifiedParameterValues[{rE → 1/3}]]
  , quarantineR[alphaFromBeta[varbeta], modifiedParameterValues[{rE → 10^12}]] },
  {varbeta, 0.8, 1.6},
  PlotStyle → Directive /@ (
    { RGBColor/@{{0, 0.7, 0.7}, {0, 0.5, 0.5}, {0, 0.3, 0.3}}
    , Dashing /@ {0.01, 0, 0.02} }ᵀ),
  PlotLegends → {"rE = 1/5", "rE = 1/3", "rE → ∞"},
  AxesLabel → {"β", "R = rC/rE"},
  PlotRange → {{0.8, 1.6}, {0, 21}},
  AxesOrigin → {1, 1}, AspectRatio → 1
] // SwappedAxes
(*plotToCsv[plot, "figure_2A"]*)
```

Out[ ]=

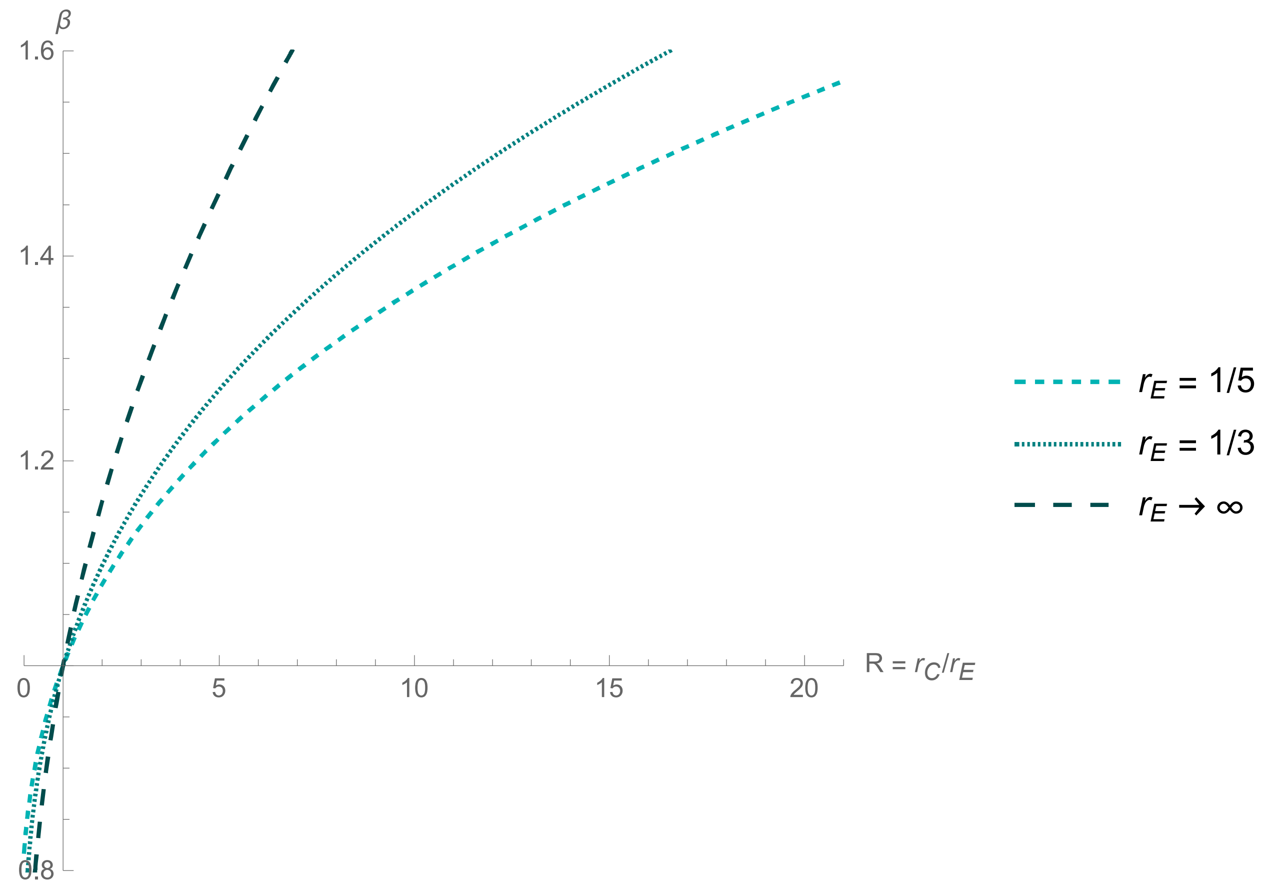


Figure 3

Fig. 3A

Note that the model variant with self-isolation requires longer simulation times than the model without self-isolation simulations.

In[•]:=
```
sixYears = 6*12*30;
plot = plotDynamics[
  LogPlot, {sumInfectious[t]*(rE /. defaultParameterValues), Susceptible[t]},
  modifiedParameterValues[{rC → 0.79, selfisolation → 1, H → 10^4}], sixYears]
(* plotToCsv[plot(*should NOT be LogPlot*), "figure_3A"]    *)
```

```
Daily new cases in the worst day: fraction: 0.000266601
Daily new cases in the worst day: time to: 101.677 days
Final fraction removed: 0.366645
```

Out[•]=

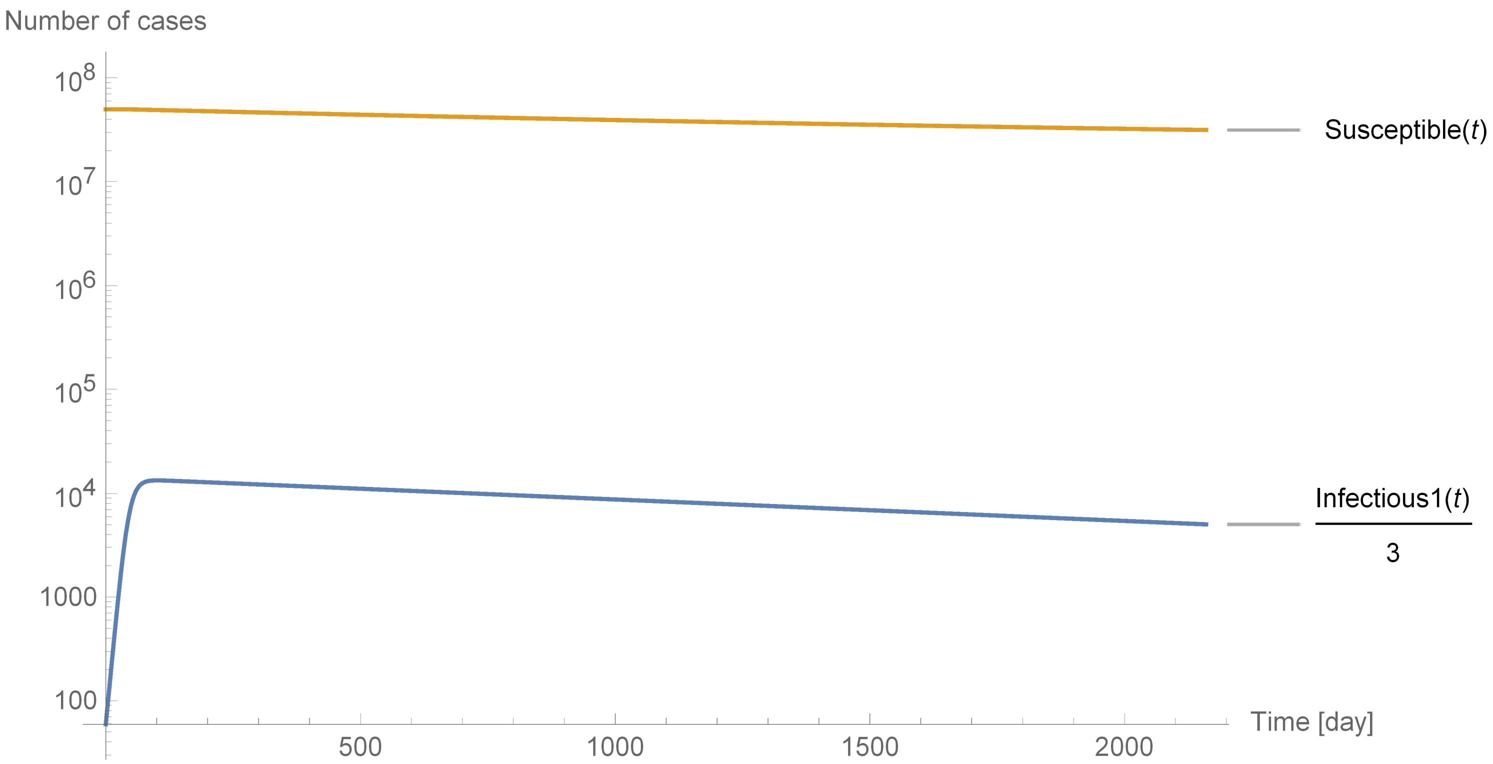


Fig. 3B, right axis

In[•]:=
```
rCs = {0.5, 1, 2};
colors = {Blue, Darker@Green, Orange, Red, Darker@Gray};
defaultH = H /. defaultParameterValues;
```

In[ ]:=
```
plot = Show @@ Table [
   ParametricPlot[
    { rCs[[i]] * varH, removedMaxCountAlt[ modifiedParameterValues[{rC → rCs[[i]], selfisolation → 1, H → varH}]]},
    {varH, 0.1 * defaultH, 10 * defaultH},
    PlotLegends → {"rC = " <> ToString[rCs[[i]]]},
    AxesLabel → {"rC*H", "Max Excluded(t)"},
    PlotRange → {{0.3 * defaultH, 3 * defaultH}, {0, 100 000}},
    PlotStyle → colors[[i]], AspectRatio → 1,
    PlotPoints → 3, MaxRecursion → 0
   ], {i, 1, Length[rCs]}
  ]
(*plotToCsv[plot, "figure_3B_maxvalue"]*)
```

Out[ ]=

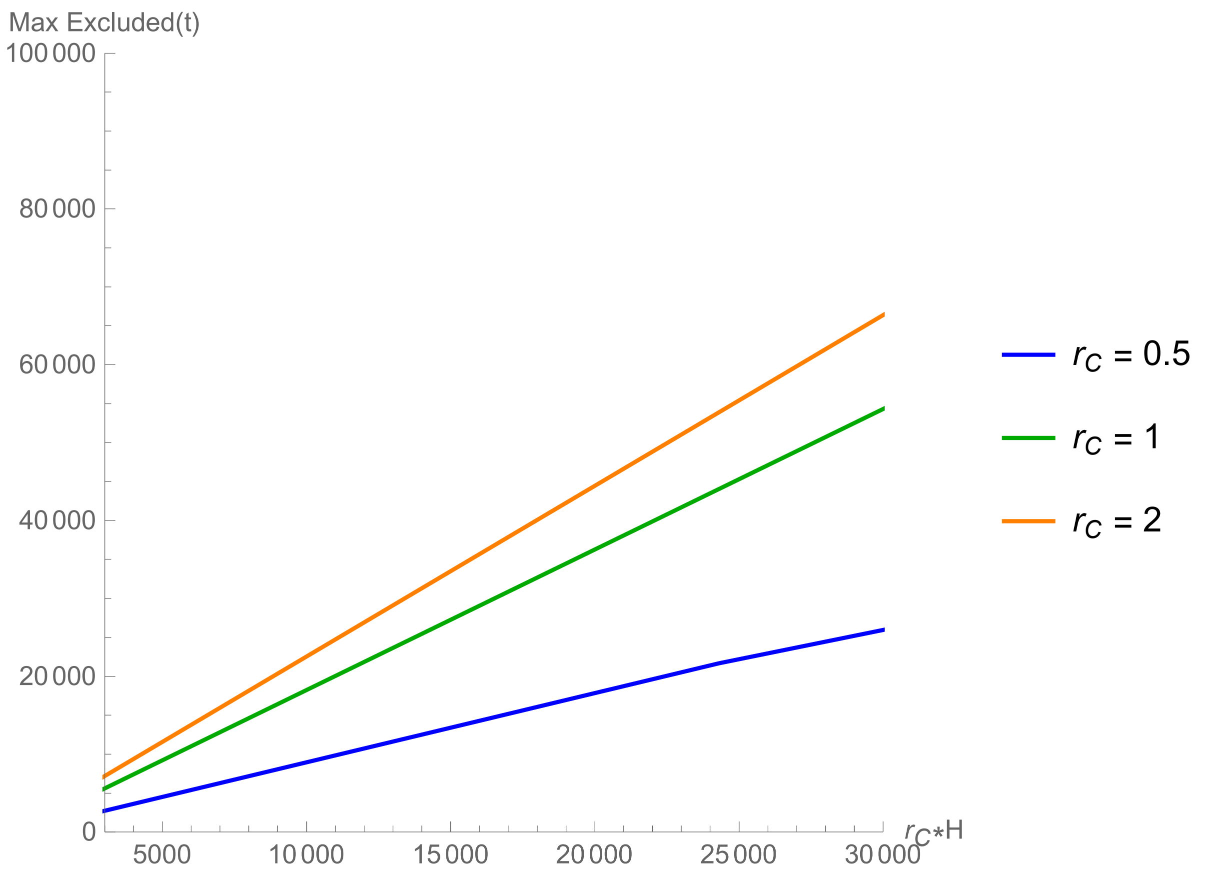


Fig. 3B, left axis

In[ ]:=
```
plot = Show @@ Table [
   ParametricPlot[
    { rCs[[i]] * varH, removedHalfmaxTime[ modifiedParameterValues[{rC → rCs[[i]], selfisolation → 1, H → varH}]]},
    {varH, 0.3 * defaultH / rCs[[i]], 3 * defaultH / rCs[[i]]},
    PlotLegends → {"rC = " <> ToString[rCs[[i]]]},
    AxesLabel → {"rC*H", "Time of Removed(t) half-max\n in decay phase (in days)"},
    PlotRange → {{0.3 * defaultH, 3 * defaultH}, {0, 2500}},
    PlotStyle → colors[[i]], AspectRatio → 1,
    PlotPoints → 8, MaxRecursion → 1
   ], {i, 1, Length[rCs]}
  ]
(*plotToCsv[plot, "figure_3B_halfmaxtime"]*)
```

Out[ ]=

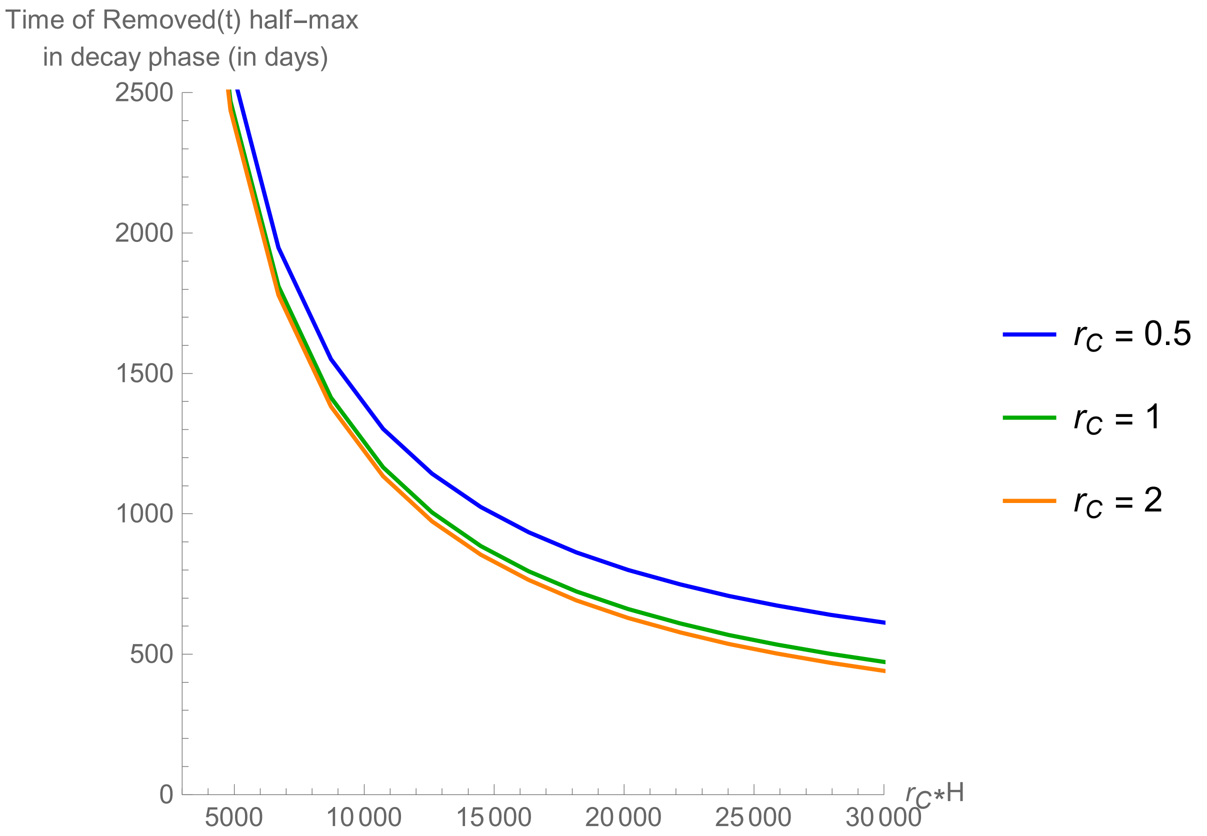


Fig. 3C

In[ ]:=
```
plot1 = plotDynamics[
   Plot, lastInfectious[t] * (rE /. defaultParameterValues),
   modifiedParameterValues[{rC → 0.79, selfisolation → 1, H → 10^4, negligence → 1, Tnegl → 3 * 30}], sixYears];
(*plotToCsv[plot, "figure_3C_3mo"]*)
```

```
Daily new cases in the worst day: fraction: 0.000848179

Daily new cases in the worst day: time to: 363.623 days

Final fraction removed: 0.597387
```

In[ ]:=
```
plot2 = plotDynamics[
   Plot, lastInfectious[t] * (rE /. defaultParameterValues),
   modifiedParameterValues[{rC → 0.79, selfisolation → 1, H → 10^4, negligence → 1, Tnegl → 6 * 30}], sixYears];
(*plotToCsv[plot, "figure_3C_6mo"]*)
```

```
Daily new cases in the worst day: fraction: 0.000609173

Daily new cases in the worst day: time to: 453.896 days

Final fraction removed: 0.591017
```

In[ ]:=
```
plot3 = plotDynamics[
   Plot, lastInfectious[t] * (rE /. defaultParameterValues),
   modifiedParameterValues[{rC → 0.79, selfisolation → 1, H → 10^4, negligence → 1, Tnegl → 9 * 30}], sixYears];
(*plotToCsv[plot, "figure_3C_9mo"]*)
```

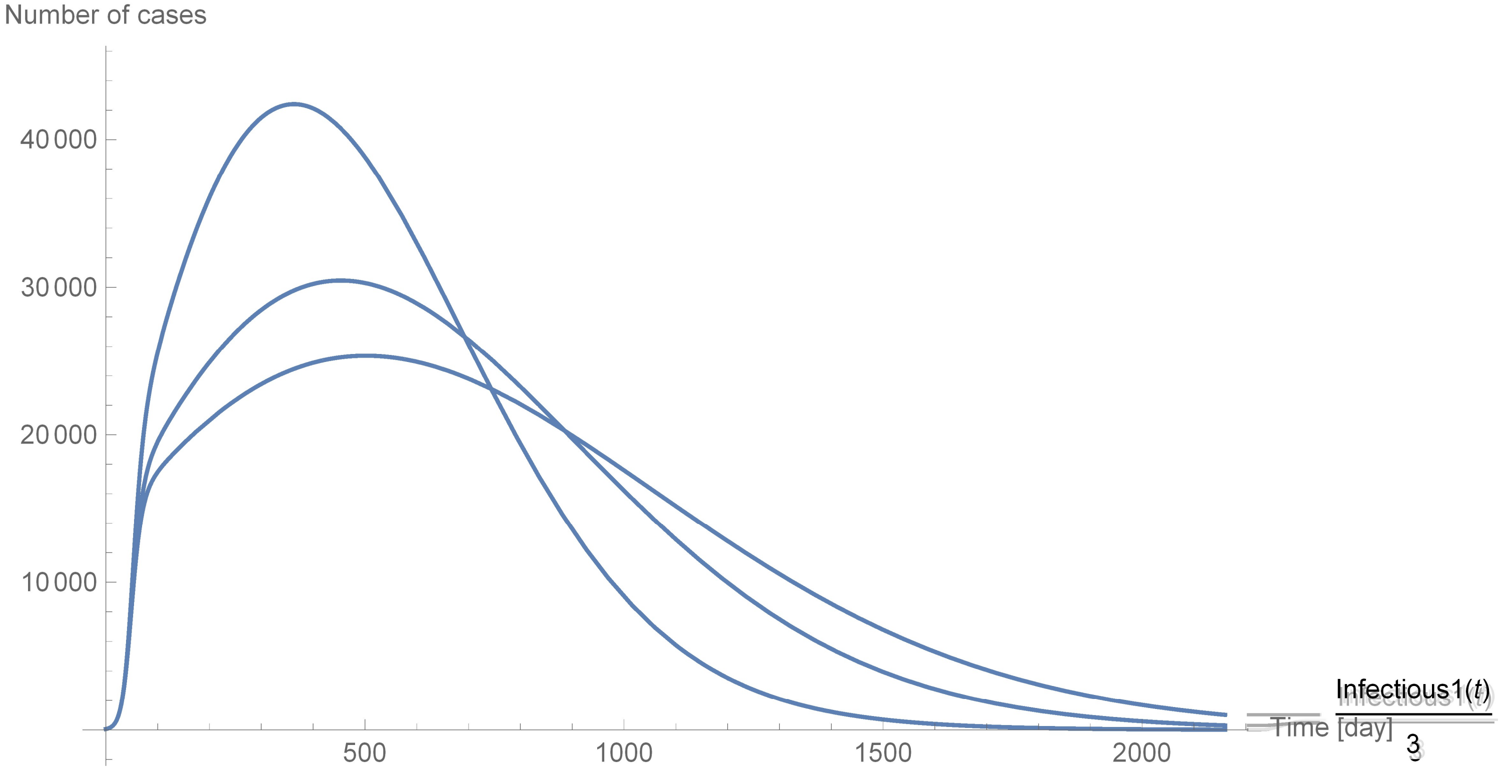
```
Daily new cases in the worst day: fraction: 0.000507015

Daily new cases in the worst day: time to: 502.177 days

Final fraction removed: 0.584169
```

```
In[ ]:= Show[plot1, plot2, plot3]
```

Out[ ]=



Supplementary Figure 1

Fig. S1A

Fig. S1B

Fig. S1C

Supplementary Figure 2

Fig. S2A

Please note that log-scale is used.

In[ ]:=
```
beta /@ modifiedParameterValues /@ {{rC → 0.678}, {rC → 0.485}}

plotBeta110 = plotDynamics[
   LogPlot, {sumInfectious[t], Removd[t], deltaRemoved[t]},
   modifiedParameterValues[{rC → 0.678}], 8 * 30];
(*plotToCsv[plotBeta110, "figure_S2A_beta110"]  (*should NOT be LogPlot*) *)

plotBeta105 = plotDynamics[
   LogPlot, {sumInfectious[t], Removd[t], deltaRemoved[t]},
   modifiedParameterValues[{rC → 0.485}], 14 * 30];
(* plotToCsv[plotBeta105, "figure_S2A_beta105"] (*should NOT be LogPlot*) *)

Show[plotBeta105, plotBeta110]
```

Out[ ]= {1.10053, 1.05003}

```
Daily new cases in the worst day: fraction: 0.0199164
Daily new cases in the worst day: time to: 118.453 days
Final fraction removed: 0.805839
Daily new cases in the worst day: fraction: 0.00687294
Daily new cases in the worst day: time to: 224.095 days
Final fraction removed: 0.55223
```

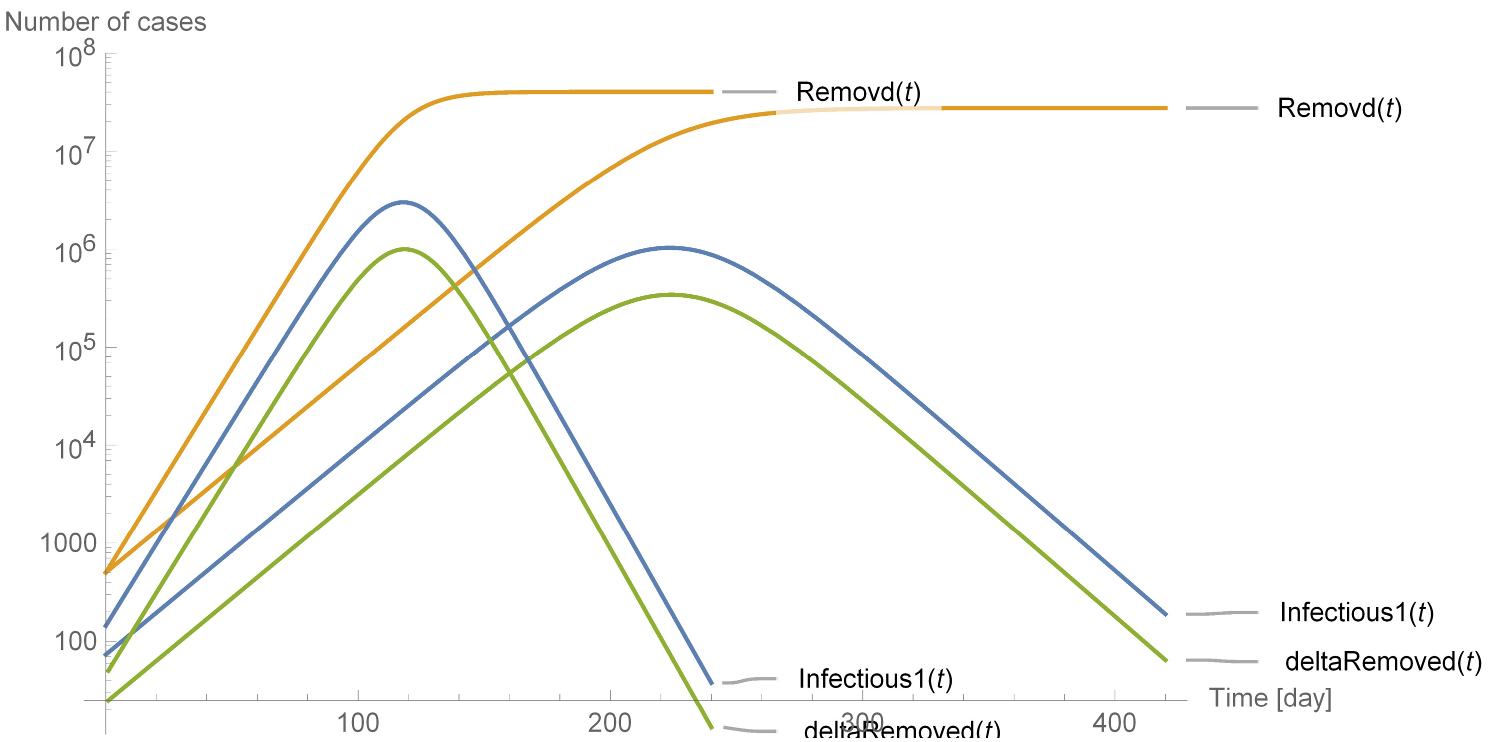
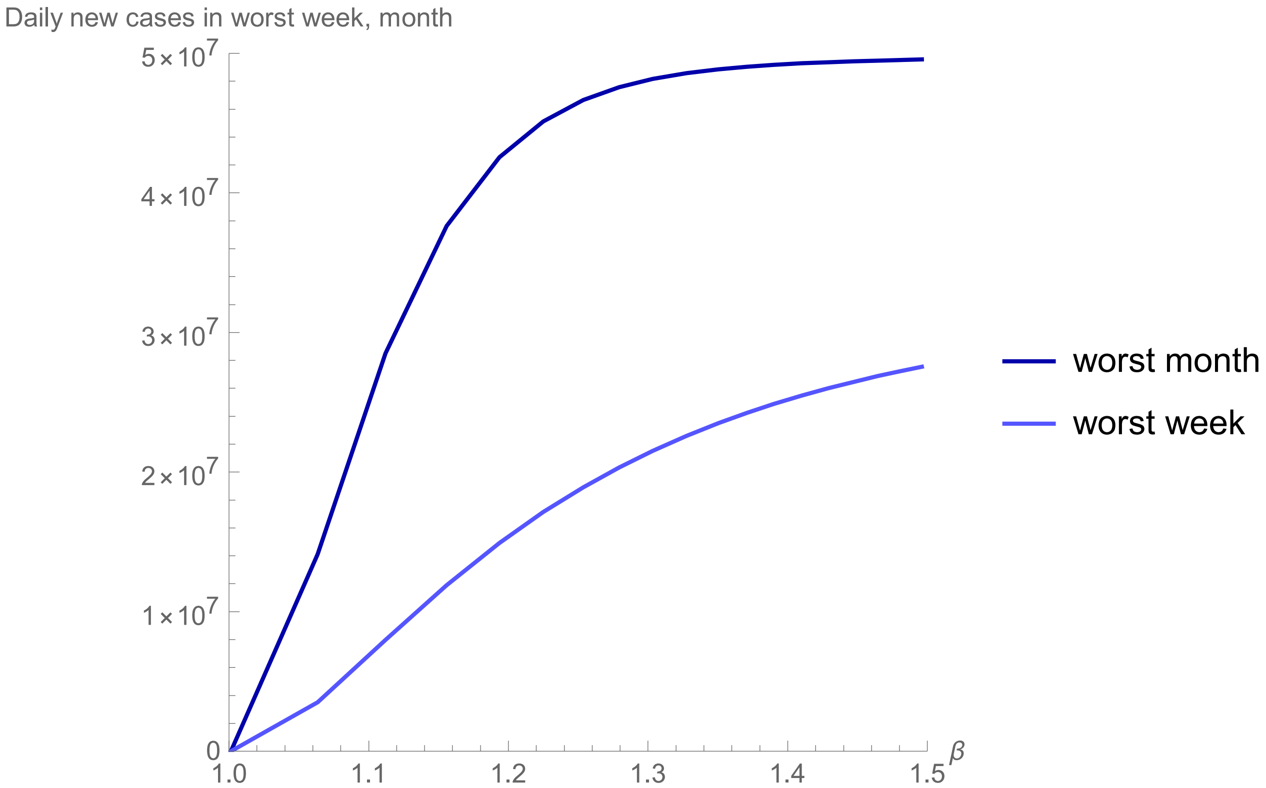
Out[ ]=


Fig. S2B: left axis

Fig. S2B: right axis

In[ ]:=
```
plot = ParametricPlot[
 { {beta[#], maxRemovedCount[#, 30]} &@modifiedParameterValues[{rC → varrC}]
 , {beta[#], maxRemovedCount[#,  7]} &@modifiedParameterValues[{rC → varrC}] },
   {varrC, 1.01*rE /. defaultParameterValues, 4},
  PlotStyle → Directive /@ {Darker@Blue, Lighter@Blue},
  PlotLegends → {"worst month", "worst week"},
  AxesLabel → {"β", "Daily new cases in worst week, month"},
   PlotRange → {{1, 1.5}, {0, PopulationN /. defaultParameterValues}},
  AxesOrigin → {1, 0}, AspectRatio → 1,
  PlotPoints → 10, MaxRecursion → 1
]
(*plotToCsv[plot, "figure_S2B_right"]*)
```

Out[ ]=



Simulations

The model assumes initial exponential growth of the epidemic, which is the case only when $r_C > r_E$.

In[ ]:=
```
Manipulate[Module[{p},
  p =
   plotDailyNewCases[modifiedParameterValues[{rC → varrC, rE → varrE, selfisolation → 1, H → varH, negligence → 1,
      Tnegl → varTnegl }]]],
 {{varrC, rC /. defaultParameterValues, "rC"}, 0.1, 2},
 {{varrE, rE /. defaultParameterValues, "rE"}, 0.1, 2},
 {{varH, H /. defaultParameterValues, "H"}, 10^3, 10^5},
 {{varTnegl, Tnegl /. defaultParameterValues, "Tnegl"}, 1*30, 12*30}]
```

Out[ ]=

r_C

r_E

H

T_{negl}

```
plotDailyNewCases[modifiedParameterValues[{rC → 1/2, rE → 1/3, selfisolation → 1, H → 10000, negligence → 1, Tnegl → 90}]]
```