Quantifying the uncertainty around the spreading of COVID-19 in Italy and Germany

April 27, 2020

1 Introduction

When modeling the Coronavirus evolution, one aims at finding the function $X(t)$ describing some data running in time, like the number of infected or deceased people, to make then future predictions. One needs to assume the belonging of X to some family of functions described by n real parameters to estimate through the observation of empirical measurements. By collecting them into a vector $P \in \mathbb{R}^n$, we stress the dependence by writing $X(t) = X^P(t)$ when needed.

Instead of looking for precise value of P, we adopt a probabilistic viewpoint when we embrace an *uncertainty* around it, quantified by using Bayesian techniques. Then, by choosing the parameters corresponding to the "worst" and the "best" scenario, we draw curves in time, delimiting the area in which we expect the future data to oscillate.

Dealing with the model's choice for X, we recall that many epidemiological models are given by autonomous ODEs. This is for instance the case of SIS and their related, as well as of general logistic maps. After pointing out a connection between these two common classes, we explain in detail our choice in favor to the latter.

When moving on concrete simulations for Germany and Italy, we clearly point out all the taken assumptions and remark the importance of the right choice for a dataset, the influence of lockdown effects on that, and how the ODE semigroup property might be helpful in that regard.

2 The use of logistic models

2.1 Introduction

One of the simplest idea for epidemiological modeling is to divide the number of total existing people N, into two classes $S(t)$ and $I(t)$, such that their sum $S + I = N$ is constant over time. People switch dynamically from the set S of *susceptible*, to I of *infected*, and conversely. Observe how in this model there

are no fatal cases, but this does not represent a problem since we take it just as a starting point. This simple scheme is known under the name of SIS and governed by a system of two ODEs:

$$\begin{cases} S'(t) = -\beta S(t)I(t) + \alpha I(t) \\ I'(t) = -\alpha I(t) + \beta S(t)I(t) \end{cases} \tag{1}$$

A complete explanation about the interpretation and the ideas beyond can be found in REF, but for us it now important to remark how β and α have specific interpretations. Beta is connected to how many people randomly meet per unit of time, while $\frac{1}{\alpha}$ is the mean time required to heal.

Since the sum $S + I = N$ is constant over time, a substitution into the ODE leads to the simple logistic equation governing the number of actively infected:

$$I'(t) = rI(t)\left(1 - \frac{1}{K}\right) \tag{2}$$

where $r = \beta N - \alpha$ and $k = N - \frac{\alpha}{\beta}$.

The equation above belongs to the family of logistic functions, maps in principle not specifically connected to epidemiology. They are curves obtained as solution of one dimensional autonomous ODEs, with S-shaped trajectories starting with an exponential growth, gradually stopped, until reaching an horizontal constant phase.

We chosen them as a prototype model for our simulations, justified by the intuition that if the simplest compartmental epidemiology model (SIS) is equivalent to the simplest logistic map, then by choosing more general logistic functions we might implicitly work with more realistic epidemiological models. Furthermore, we do not model the number of actively infected, but rather the amount of total infected and number of deaths. We ignore further theoretical analysis for this decision, we kept the choice encouraged by nice concrete results.

2.2 The generalized logistic map: Richard's ODE

The most general logistic map (still expressed by an ODE, we'll see later the reason beyond this requirement) is given by the Richard's equation. It's the following ordinary differential equation governed by three real positive parameters, $P = \{q, Q, \nu\}$, so $n = 3$. We need to fix an initial condition X_0 (not considered as parameter since directly observed from data), and write the model as $X(t) = X^P(t) = X^P_{X_0}(t)$ when we want to stress this dependence. The Richard's law satisfy the equation:

$$X'(t) = qX(t)\left(1 - \left(\frac{X(t)}{Q}\right)^{\nu}\right) \tag{3}$$

with closed form solution:

$$X(t) = \frac{Q}{(1 + A\exp[-q\nu t])^{\frac{1}{\nu}}} \tag{4}$$

Here A is just an abbreviation for $A = -1 + \left(\frac{Q}{X_0}\right)^{\nu}$. The parameter ν is related to the symmetry of the curve, while q represents the initial exponential growth then stopped in time. Intuitively we think of it to be related to the average amount of people's social contacts, as in the simpler logistic case connected to the SIS model.

By taking the limit $\lim_{t\to\infty} X(t) = Q$, we understand Q to represent the asymptotic maximal value.

Setting $\nu = 1$ gives the basic logistic map, described in the next section, while taking the limit $\nu \mapsto 0$ under appropriate gives the Gompertz law, explained later.

2.3 The simple logistic map

In the simple logistic map the model $X^P(t)$ depends on two real positive parameters, $P = \{q, Q\}$, so that $n = 2$, and is the solution of the 1-dimensional ODE:

$$X'(t) = qX(t)\left(1 - \frac{X(t)}{Q}\right) \tag{5}$$

with closed-form formula:

$$X(t) = \frac{X_0 \exp[qt]}{1 - \frac{X_0}{Q}(1 - \exp[qt])} \tag{6}$$

Where again X_0 is the starting ODE condition at time 0. Of course, we have $\lim_{t\to\infty} X(t) = Q$, and since this simple logistic map is equivalent to the simplest SIS model (as explained before), the parameter q can be actually rigorously related to the average number of daily people's social interaction.

2.4 The Gompertz law

The Gompertz law is governed by two real positive parameters, q and Q, therefore we write again $P = \{q, Q\}$ and $n = 2$. Once the starting condition X_0 at time zero is specified, $X(t) = X^P(t) = X^P_{X_0}(t)$ is the solution to:

$$X'(t) = qX(t)\log\left[\frac{Q}{X(t)}\right] \tag{7}$$

expressed by the closed-form formula:

$$X(t) = Q\exp\left[\log\left[\frac{X_0}{Q}\right]\exp\left[-qt\right]\right] \tag{8}$$

As seen for the previous cases, Q corresponds to the limit $\lim_{t\to\infty} X(t)$.

3 The bayesian approach

When using the models above in practice, we can only observe a limited amount of points coming from the ODE trajectory, perturbed by a random noise. Let's fix $T+1$ times $\{t_i\}_{i=0,\dots,T}$.

Definition 1. *The* observed vector $\boldsymbol{y} \in \mathbb{R}^{T+1}$ *is the random variable defined componentwise as:*

$$y_i(\omega) = X^P_{X_0}(t_i) + \eta_i(\omega) \tag{9}$$

We assume the error expression $\eta_i \sim \mathcal{N}(0, \sigma_i^2)$ with a time dependent variance. Sometimes by an abuse of notation we use the symbol η_i to indicate its density function too, so writing $\eta_i(x)$ for $x \in \mathbb{R}$ refers to that.

Recall that the finally goal is to give an estimation of the unknown values for $P\mathbb{R}^n$, *given* the observation $\mathbf{y}$. We choose a Bayesian approach to find a solution. Therefore, we need to define a prior distribution for P, a likelihood function, then we get the solution as the product of them. In other words, during the search for the true parameters, P is seen as a random variable $\Omega \to \mathbb{R}^n$ whose law, *conditioned* on the observations, is what we aim to understand.

As standard in the field, we start by assuming that $\mathbb{P}[P \in A] = \rho(A)$ is described by a specific density function ρ called the prior distribution, representing our blind guess about P *independently* of the observations $\mathbf{y}$.

If we aim at understanding $\mathbb{P}[P|\mathbf{y}]$, then the classical Bayes's law allows to find it in terms of an hypothetical law $\mathbb{P}[\mathbf{y}|P]$: this is where the notion of likelihood comes into play.

Definition 2. *For every fixed choice of the parameters P, the likelihood functions for the observation of $\boldsymbol{y}$, given P, is defined to be:*

$$\mathcal{L}(\boldsymbol{y}|P) \doteq \frac{(2\pi)^{-\frac{T}{2}}}{\sigma_0 \dots \sigma_T} \exp\left(-\frac{1}{2}\sum_{i=0}^{T} \frac{(y_i - X^P_{X_0}(t_i))^2}{\sigma_i^2}\right) \tag{10}$$

By *interpreting* the likelihood as an effective probability conditioning, writing informally $\mathbb{P}[\mathbf{y}|P] = \mathcal{L}(\mathbf{y}|P)$, its formula is then explained when looking at the noise distribution $\eta(\mathbf{y} - X^P)$. By imitating the Bayes's rule:

$$\mathbb{P}[P|\mathbf{y}] \propto \mathbb{P}[\mathbf{y}|P]\mathbb{P}[P] \tag{11}$$

the final result is defined as:

Definition 3. *The (Bayesian) answer to the problem "find the probability density of the parameters P given the observations $\boldsymbol{y}$" is given by the* posterior *distribution on $\mathbb{R}^n$ defined by:*

$$\mu(dx) \propto \mathcal{L}(\boldsymbol{y}|x)\rho(dx) \tag{12}$$

During every use of the Bayesian rule, we constantly omitted the denominator term relying always on the mere proportionality "$\propto$". This because such a value is always the probability normalization constant, a number that can be completely ignored in practice thanks to the use of suitable numerical techniques.

In few words, this section should have transmitted the idea that the problem "estimating the parameters P" has been converted into the problem "understanding the probability distribution μ".

3.1 The pCN Monte Carlo algorithm

In the previous section we explained the Bayesian strategy to convert the problem of parameter estimation to the understating of the posterior probability measure μ. In principle it could be possible to detect various properties by using analytical tools, but due to time constraints we rely on a simple effective strategy: first produce a large amount of samples from μ, then analyze them statistically.

The chosen algorithm is the preconditioned Crank-Nicolson Monte Carlo (pCN), in general particularly strong on high dimensional spaces and capable of managing well discretizations coming from infinite dimensional Gaussians. Here we briefly revise and adapt it for our specific and much simpler situation. The pCN algorithm *requires* the choice of a Gaussian prior distribution ρ.

Recall that n is the carnality of P (so, 2 or 3 according to the chosen model). For two points in $\mathbb{R}^n$, define the choice probability:

$$a(u, v) \doteq \min\{1, \frac{\mathcal{L}(\mathbf{y}|u)}{\mathcal{L}(\mathbf{y}|v)}\} \tag{13}$$

and set the *exploratory* parameter $0 < \beta_{pcn} < 1$ (see step 3).

To produce a **single sample** from μ, construct a chain $\{x_i\}_{i\in\mathbb{N}}$ as follows:

1. set $x_0 \in \mathbb{R}^n$ arbitrarily. Then, for each $k > 0$:
2. sample a point R from the gaussian prior distribution ρ;
3. propose a candidate as $\hat{x}_k = \sqrt{(1-\beta^2)}x_{k-1} + \beta R$;
4. accept it (i.e. set $x_k = \hat{x}_k$) with probability $a(x_{k-1}, \hat{x}_k)$;
5. (accepted or not) repeat from 2;

We define N_{pcn} the integer at which we always stop the chain, producing therefore a single approximated posterior's sample. Of course repeating the algorithm allows to collect a large amount of sample (define this number as S_{pcn}), and we remark how every instance is independent and therefore suitable to parallelization (the user must be warned about the use of a proper seed).

4 Numerical results

4.1 Assumptions and methodology

We apply the previous theory in order to formulate hypothesis for Italy and Germany, concerning the future number of deceased people, whose data are considered to be more "reliable", and the amount of total infected, information more controversial due to test limitations and hidden asymptomatic cases. We need to completely list and clarify the hypothesis into play, and to explain how the results must be read.

4.1.1 Parameters for the Bayesian technique

In both Countries the available data start around the mid of February, so for drawing a model one might be tempted to use these *entire* datasets (until today 19th of April) relying on the intuition "more data, more accuracy". But in the case of logistic growths, some parameters are connected to the average number of people's daily contacts. Since lockdown measures have been adopted, their values surely changed in time, resulting possibly in *two* ODE instances: a first trajectory with the pre-lockdown coefficients (or just possibly another model), followed then by a new independent one with other coefficients, which are the one to discover.

By considering that the countermeasures started their effects more or less around the 25th of March, to increase the chance of being in the right phase one might start the interpolation some days later.

But this strategy hides a pitfall: it's necessary to ensure that the skipped days can actually be "forgotten" according to model under usage! When choosing autonomous ODEs, this issue is completely solved. Thanks to the semigroup property of the associated flow, no matter if we start observing a value, say V_n, at day n, or V_{n+1} at $n+1$: the trajectory produced from day $n+1$ (with initial conditions V_{n+1}) is precisely the same of beginning at day n with initial condition V_n, therefore they are ruled by the same values of P.

As a consequence of the reasoning above we took the safe choice of focusing on datasets spanning from the 31th of March to the 19th of April, setting so $T = 19$, and $t_i = i$, with t_0 interpreted as the 31.03 and t_{19} as the 19.04.

To complete the definition of the observed vector $\mathbf{y}$, we need to specify the variances σ_i for the noises η_i. Remarking how this is definitely an arbitrarily choice for which we do not possess any better hint, we finally opted for setting σ_i^2 precisely as the empirical value read at day i. By experimenting with data, it seemed to model a possibly "reasonable" perturbation.

Finally there is the need to fix a precise prior distribution ρ. Recall that it must be a centered Gaussian in order to use the pCN algorithm. For the general logistic case we set the covariance matrix as

$$\begin{pmatrix} 0.1^2 & 0 & 0 \\ 0 & 5000^2 & 0 \\ 0 & 0 & 0.25^2 \end{pmatrix} \tag{14}$$

when working with the total number of infected,

$$\begin{pmatrix} 0.1^2 & 0 & 0 \\ 0 & 3000^2 & 0 \\ 0 & 0 & 0.25^2 \end{pmatrix} \tag{15}$$

when with the number of deaths. For both the simple logistic and Gompertz laws we used the same covariance matrix, namely the upper left 2x2 submatrix of the general logistic above.

4.1.2 Parameters in the pCN Algorithm

When coming to the pCN algorithm, we set the parameter $\beta_{pcn} = 0.4$ is order to have a "compromise" between a conservative and an exploratory Markov Chain, while the starting vector in the step (1) has always chosen uniformly random. The first component, i.e. the one referring to q, in $[0.01, 0.6]$, while the second (Q) in $[200000, 900000]$ for the infected case, $[20000, 80000]$ for the deaths in Italy and $[1000, 7000]$ for the same in Germany. When using the general logistic growth, the third component (ν) started randomly in $[0.1, 3.0]$. We set the number of samples $S_{pcn} = 5000$, and the stopping time of the Markov Chain at $N_{pcn} = 10000$. The experiments run under (Slackware) Linux using a C library for general numerical purposes developed by the authors, and each complete simulations required between one and two minutes on a Dell XPS 13 with a i3-8145U CPU @ 2.10GHz processor.

4.2 Known weak points

When using the bayesian algorithm one must carefully verify the dependence of the results, i.e. posterior distribution μ, on the choice of the prior ρ. The general theoretical framework in described in REF, but we didn't manage to check it carefully due to time constraints. Repeating plenty of simulations on synthetic data with comparable numbers as in the real cases, suggested how the prior does not remarkably influence the posterior, rather only the performance of the Monte Carlo algorithm (e.g. more points were needed).

We are aware that structuring the noise as described is for sure a strong assumption, so we tested again against toy-simulations. For each run the true value of P was therefore known, giving the possibility of computing not only the interpolation but also the true errors. When setting the noise to very small values (e.g. $\sigma_i = \sigma = 0.1$) we always obtained extremely concentrated posterior measures, with both true and interpolation errors inferior to 1%. When giving the noise larger values, the results generally kept an interpolation error less that 3%, but the true errors were oscillating more, especially in the case of the general logistic growth. This is not an error in the algorithm, rather the fact that different parameters can give very similar output results. When working on real data, we underline how it does not represent a problem at all: we always consider the *whole* probabilistic distribution of parameters, and by selecting the

"most" and the "least" convenient we are able to draw the area of uncertainty as described in the upcoming section.

Finally, we remark how the parameters for the pCN Monte Carlo algorithm didn't seem to afflict the *results*, rather only the *performance*, and we tuned them using numerous synthetic datasets.

The convergence of the Monte Carlo algorithm was empirically checked by re-running with the same conditions but larger and larger stopping times, seeing no differences in the results.

4.3 Results

When working with logistic models in the case of Coronavirus, we can exploit a simple fact: instead of the full composition of P, what *truly* matters is the value of Q, representing the asymptotic reached value (e.g. the maximum number of deaths, or the maximum number of infected). By merely plotting the posterior distribution we are able to read it.

In the case of the general logistic growth, once Q is selected, the other two parameters have generally some degree of freedom. Conversely, when using the simple logistic or the Gompertz law, choosing Q forces automatically the parameter q, being the posterior 2-dimensional. It implies that when we plot the posterior distributions, for these two simpler models we are able to directly get two couple of parameters P_{worst} and P_{best} corresponding to the "worst" (i.e. highest Q) and the "best" (lowest Q) ODE prediction. Their are pictured two times: the first following the format "day, number of cases", and a second taking into account their derivative so showing the daily's *new* cases.

Note that sometimes the posterior includes values inferior to the current registered in real life, producing a not acceptable result. This is not an error, rather the effect of inserting a gaussian noise (an algorithm's mathematical requirement) so that the measurements are equally considered underestimated (realistic) and overestimated (not realistic). There is no problem with that, and it's enough to ignore these points.

We set as day zero the 31th of March, with 774 cases of deceased people in Germany and 12468 in Italy. On the same day, the number of total infected is 71690 in Germany and 105792 in Italy (these are consequently the starting conditions for all the ODEs). The time span for the predictions last 61 days, reaching the end of May when the asymptotic values are always touched for all the simulated cases.

5 Conclusions - TO BE COMPLETELY UPDATED

Logistic : very bad, as expected... Generalied Logistic: overfitting? Too many parameters? Gompertz...promising...

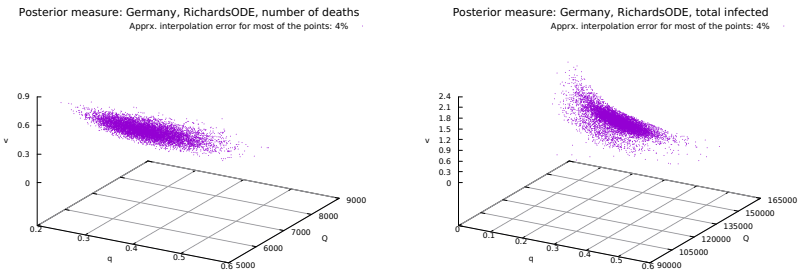

Figure 1: Posterior distribution obtained by using the **general logistic** growth models on **Germany**. The number of **deaths** and the number of total **infected** are taken into analysis. The former is bounded between 5500 and 9000, while the latter surely below 170000. The *same* results are then obtained with the Gompertz and simple logistic map, but there it's easier to also understand the growth's rate.

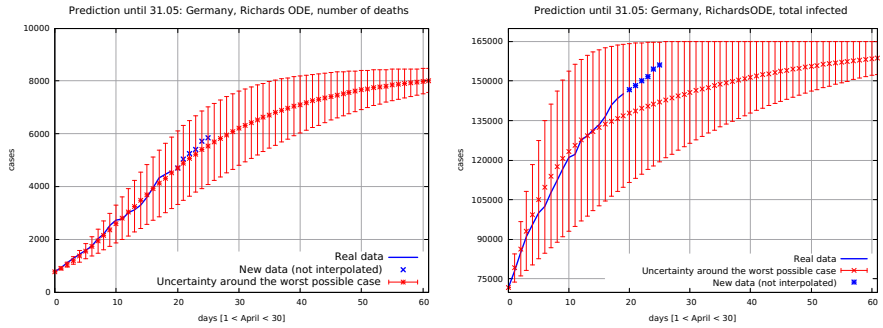

Figure 2: to write

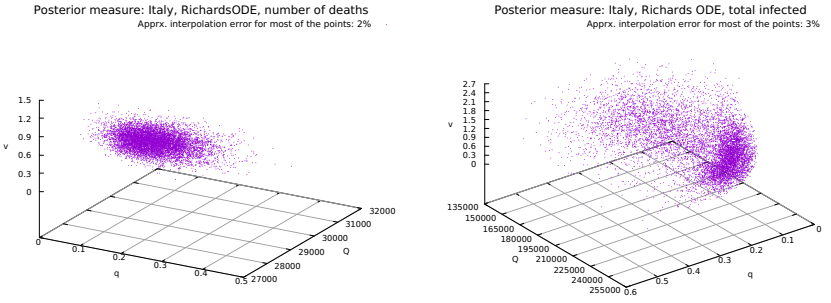

Figure 3: Posterior distribution obtained by using the **general logistic** growth models on **Italy**. The number of **deaths** and the number of total **infected** are taken into analysis. The former is bounded between 27000 and 34000, the latter stays below 250000. The *same* results will be obtained with the Gompertz and simple logistic map, where it's easier to estimate the growth's rate.

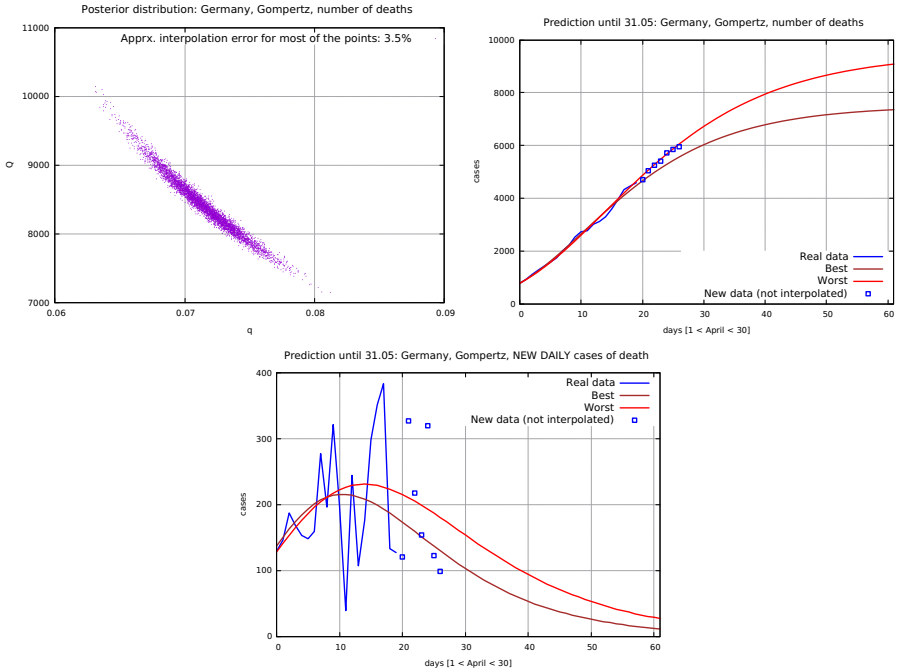

Figure 4: Predicting the number of **deceased** people in **Germany** until the end of May. Result according to the **Gompertz** model, which produced an interpolation error of around 3%.

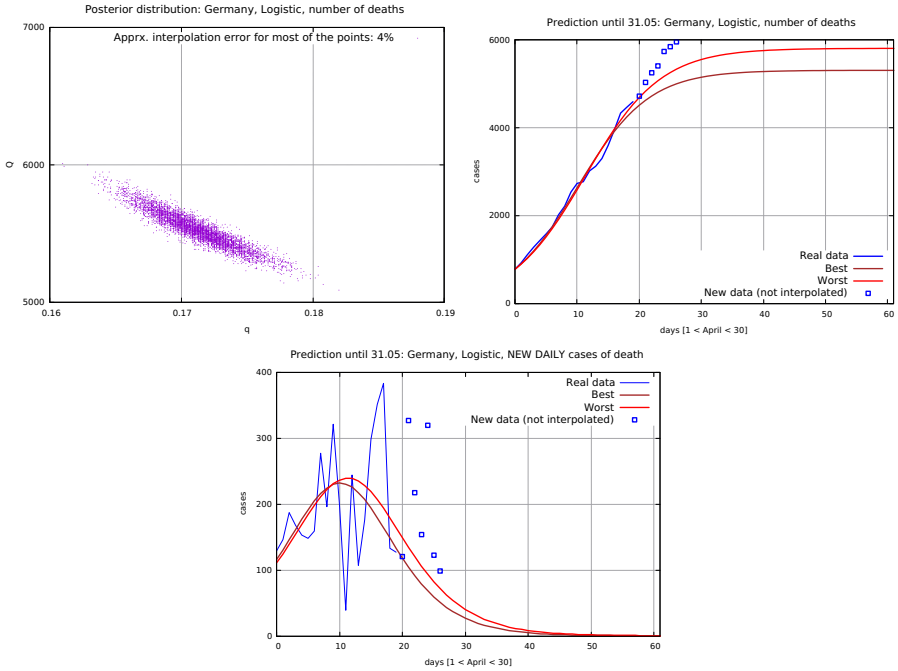

Figure 5: Predicting the number of **deceased** people in **Germany** until the end of May. Result according to the **simple logistic** map, interpolation error around 4%.

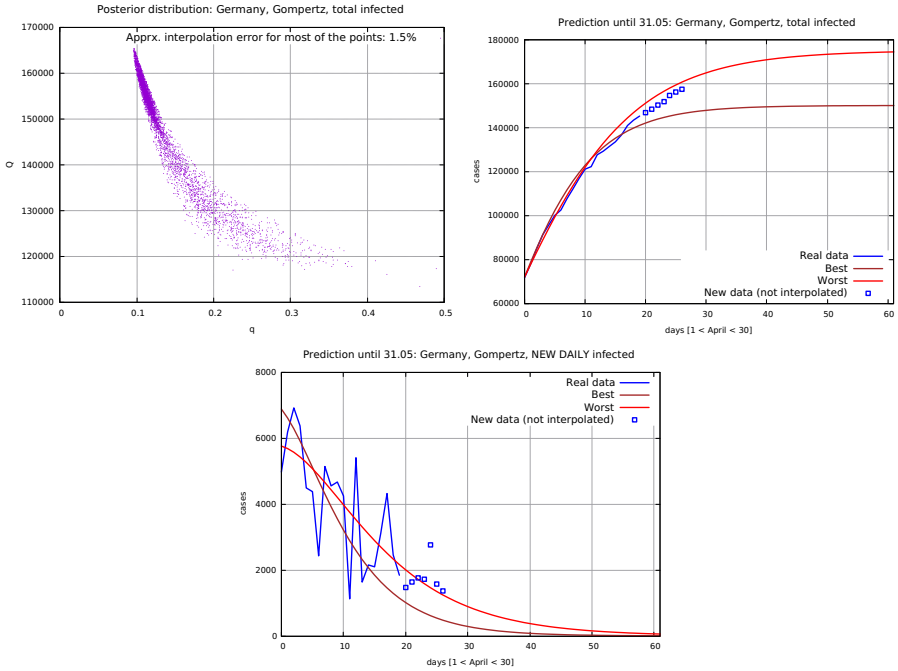

Figure 6: Predicting the number of the total **infected** people in **Germany** before the end of May. Results according to the **Gompertz** law; interpolation error of 1.5%.

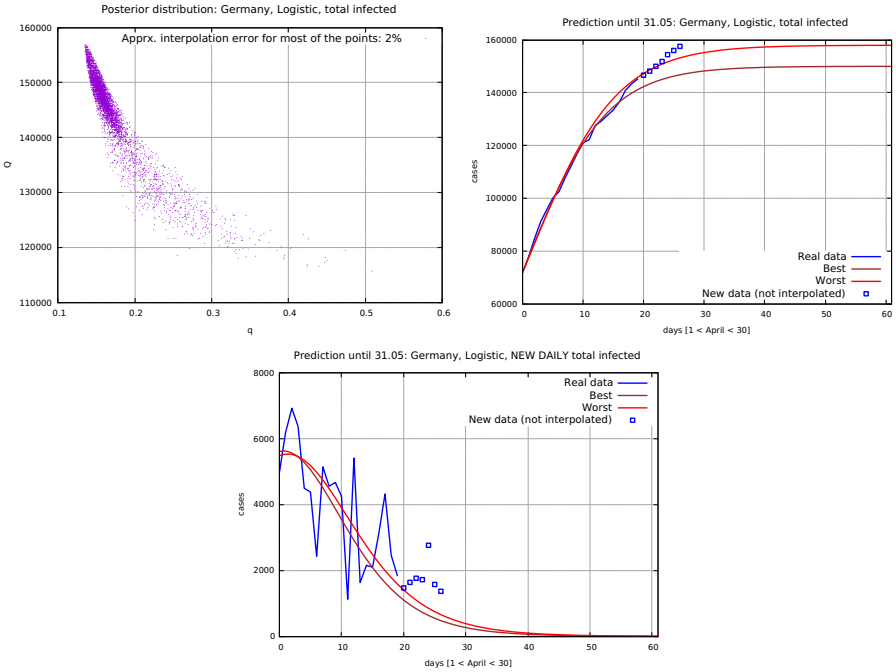

Figure 7: Predicting the number of the total **infected** people in **Germany** before the end of May. Results according to the **simple logistic** map; interpolation error of 2%.

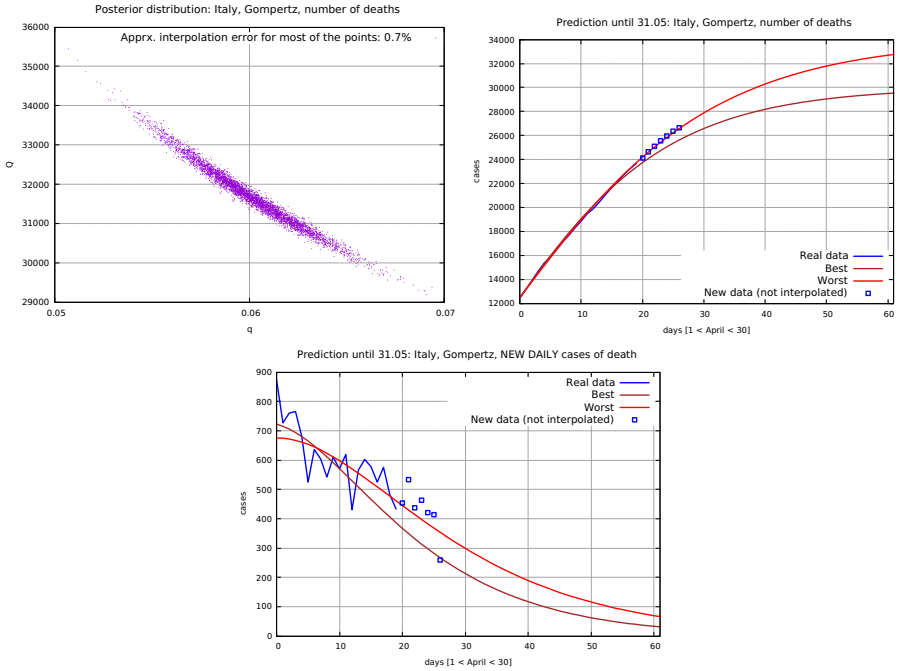

Figure 8: Predicting the number of **deceased** people in **Italy** until the end of May. Result according to the **Gompertz** law, interpolation error around 0.7%.

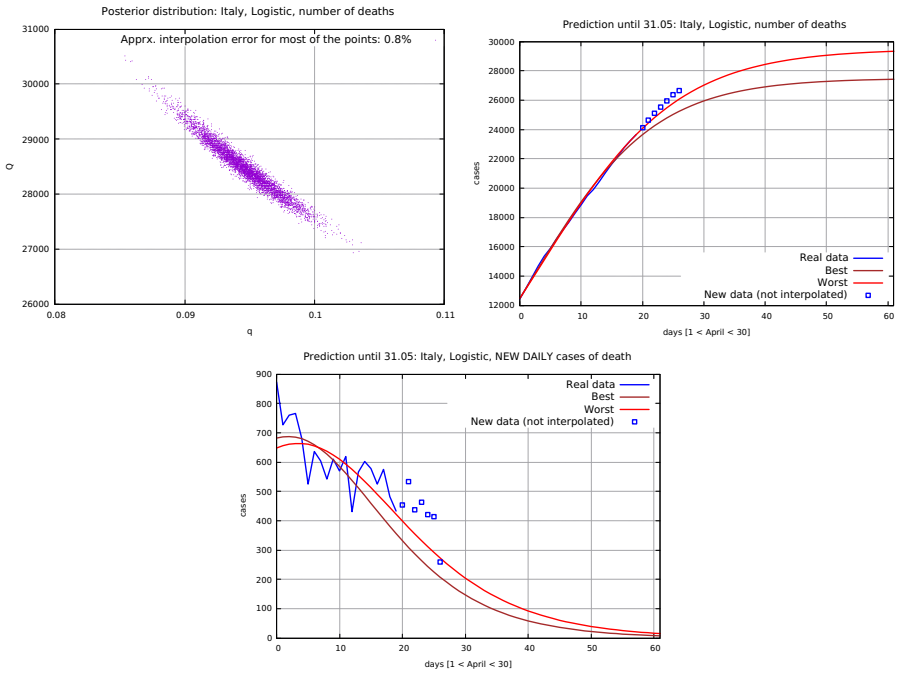

Figure 9: Predicting the number of **deceased** people in **Italy** until the end of May. Result according to the **simple logistic** map interpolation error around 0.8%.

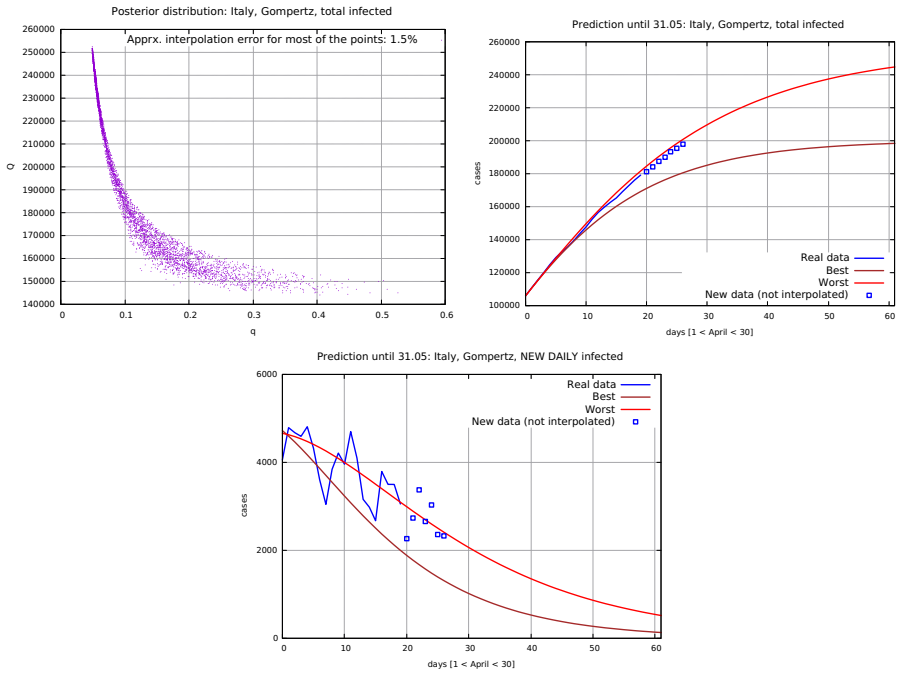

Figure 10: Predicting the number of total **infected** people in **Italy** until the end of May. Result according to the **Gompertz** law, interpolation error around 1%.

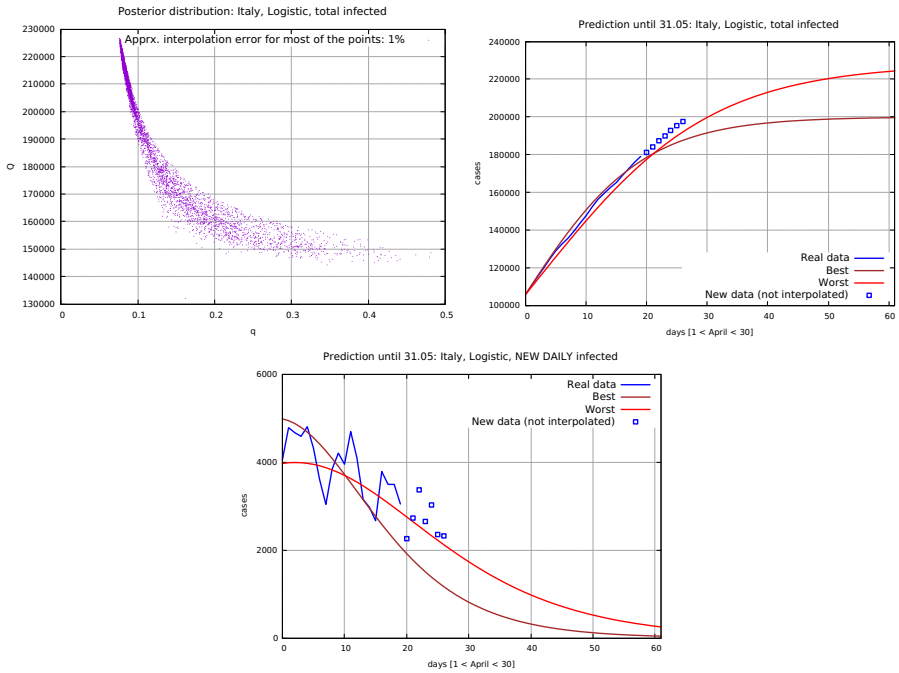

Figure 11: Predicting the number of total **infected** people in **Italy** until the end of May. Result according to the **simple logistic** map, interpolation error around 1%.