# Estimating the number of COVID-19 victims by using a Monte Carlo algorithm on a generalized logistic equation

Biagio Paparella

10.06.2020

Oberseminar - Institute for Numerical Simulation - Bonn

# Introduction and motivation

Goal of my recent work: influenced by recent events, try predicting the number of COVID-19 victims.

- (Possibly) the most reliable measurement, compare e.g. with infected (test problems, etc...);

# Introduction and motivation

Goal of my recent work: influenced by recent events, try predicting the number of COVID-19 victims.

- (Possibly) the most reliable measurement, compare e.g. with infected (test problems, etc...);
- Studied time span: April to predict May (stable conditions);

## Introduction and motivation

Goal of my recent work: influenced by recent events, try predicting the number of COVID-19 victims.

- (Possibly) the most reliable measurement, compare e.g. with infected (test problems, etc...);
- Studied time span: April to predict May (stable conditions);
- Idea: if it works, repeat now to predict June/July...

# Introduction and motivation

Goal of my recent work: influenced by recent events, try predicting the number of COVID-19 victims.

- (Possibly) the most reliable measurement, compare e.g. with infected (test problems, etc...);
- Studied time span: April to predict May (stable conditions);
- Idea: if it works, repeat now to predict June/July...
- Chosen Countries: Germany, France, Italy.

# The model in theory: ODE description

$X(t)$ : number of deaths in time. Modeled via a simple 1-d ODE with initial conditions $X_0$. Generalized logistic map (S-shaped, exp initially, then flat):

$$X'(t) = \frac{q}{v} X(t) \left(1 - \left(\frac{X(t)}{Q}\right)^{v}\right) \tag{1}$$

- It has a close form solution (here not written);

## The model in theory: ODE description

$X(t)$ : number of deaths in time. Modeled via a simple 1-d ODE with initial conditions $X_0$. Generalized logistic map (S-shaped, exp initially, then flat):

$$X'(t) = \frac{q}{v} X(t) \left( 1 - \left( \frac{X(t)}{Q} \right)^{v} \right) \tag{1}$$

- It has a close form solution (here not written);
- Depends on the parameter $P = \{q, Q, v\}$ (more next slide).

# The model in theory: ODE description

$X(t)$ : number of deaths in time. Modeled via a simple 1-d ODE with initial conditions $X_0$. Generalized logistic map (S-shaped, exp initially, then flat):

$$X'(t) = \frac{q}{v} X(t) \left( 1 - \left( \frac{X(t)}{Q} \right)^{v} \right) \tag{1}$$

- It has a close form solution (here not written);
- Depends on the parameter $P = \{q, Q, v\}$ (more next slide).
- In others words: assume $\exists P \in \mathbb{R}^3$, s.t. $X^P(t)$ correctly describe the (past and future) number of victims. Want to find $P$.

# The model in theory: parameters description

Do not need to remember the form of $X^P(t)$. It's worth understanding the role of the parameter $P$.

- $P = \{q, Q, v\}$ are all *strictly* positive.

## The model in theory: parameters description

Do not need to remember the form of $X^P(t)$. It's worth understanding the role of the parameter $P$.

- $P = \{q, Q, v\}$ are all *strictly* positive.
- $Q$ is the carrying capacity ($\lim_{t\to\infty} X(t) = Q$);

## The model in theory: parameters description

Do not need to remember the form of $X^P(t)$. It's worth understanding the role of the parameter $P$.

- $P = \{q, Q, v\}$ are all *strictly* positive.
- $Q$ is the carrying capacity ($\lim_{t \to \infty} X(t) = Q$);
- $v = 1$ gives the *simple* logistic model, while $v \to 0^+$ the *Gompertz* model. Both very famous and used in the literature;

# The model in theory: parameters description

Do not need to remember the form of $X^P(t)$. It's worth understanding the role of the parameter $P$.

- $P = \{q, Q, v\}$ are all *strictly* positive.
- $Q$ is the carrying capacity ($\lim_{t\to\infty} X(t) = Q$);
- $v = 1$ gives the *simple* logistic model, while $v \to 0^+$ the *Gompertz* model. Both very famous and used in the literature;

# The model in theory: parameters description

Do not need to remember the form of $X^P(t)$. It's worth understanding the role of the parameter $P$.

- $P = \{q, Q, v\}$ are all *strictly* positive.
- $Q$ is the carrying capacity ($\lim_{t\to\infty} X(t) = Q$);
- $v = 1$ gives the *simple* logistic model, while $v \to 0^+$ the *Gompertz* model. Both very famous and used in the literature;
- $q$ is related to the "starting" exponential growth;

## The model in theory: parameters description

Do not need to remember the form of $X^P(t)$. It's worth understanding the role of the parameter $P$.

- $P = \{q, Q, v\}$ are all *strictly* positive.
- $Q$ is the carrying capacity ($\lim_{t\to\infty} X(t) = Q$);
- $v = 1$ gives the *simple* logistic model, while $v \to 0^+$ the *Gompertz* model. Both very famous and used in the literature;
- $q$ is related to the "starting" exponential growth;

- Intuition: $Q$ the "final" number of victims $(10, 30, 50K?)$, we expect $0 < q << 1$ (warning: this comes from the simple logistic) and we prefer $0 < v < 1$ oscillating between the two common models (but no actual math. restriction).

# The model in theory: parameters description

Do not need to remember the form of $X^P(t)$. It's worth understanding the role of the parameter $P$.

- $P = \{q, Q, v\}$ are all *strictly* positive.
- $Q$ is the carrying capacity ($\lim_{t\to\infty} X(t) = Q$);
- $v = 1$ gives the *simple* logistic model, while $v \to 0^+$ the *Gompertz* model. Both very famous and used in the literature;
- $q$ is related to the "starting" exponential growth;

- Intuition: $Q$ the "final" number of victims $(10, 30, 50K?)$, we expect $0 < q << 1$ (warning: this comes from the simple logistic) and we prefer $0 < v < 1$ oscillating between the two common models (but no actual math. restriction).
- Keep in mind: $Q$ large, $> 10.000$, $v$, $q$ small, $\in [0, 1]$.

# The model in practice: observed vector

The available dataset are "numbers of total deaths until day $t_i$": discrete, limited, ODE trajectory of $X^P_{X_0}(t)$. Use them to infer $P$.

Fix $T+1$ times $\{t_i\}_{i=0,\dots,T}$, e.g. 21 days. The observed vector $\mathbf{y} \in \mathbb{R}^{T+1}$ is the random variable defined componentwise as:

$$y_i(\omega) = X^P_{X_0}(t_i) + \eta_i(\omega) \qquad (2)$$

where $\eta_i \sim \mathcal{N}(0, \sigma_i^2)$, $i \in \{0, \dots, T\}$.

## The model in practice: observed vector

The available dataset are "numbers of total deaths until day $t_i$": discrete, limited, ODE trajectory of $X_{X_0}^P(t)$. Use them to infer $P$.

Fix $T+1$ times $\{t_i\}_{i=0,\ldots,T}$, e.g. 21 days. The observed vector $\mathbf{y} \in \mathbb{R}^{T+1}$ is the random variable defined componentwise as:

$$y_i(\omega) = X_{X_0}^P(t_i) + \eta_i(\omega) \tag{2}$$

where $\eta_i \sim \mathcal{N}(0, \sigma_i^2)$, $i \in \{0, \ldots, T\}$.

- In other words, $\mathbf{y}$ is what we measure: "noised truth".

# The model in practice: noise

To assume an error of the form

$$y_i(\omega) = X^P_{X_0}(t_i) + \eta_i(\omega) \tag{3}$$

is a strong assumption (I know). Need to decide the standard variations $\sigma_i$ for each day. Abritrately.

Quantile formula: $\eta_i \sim N(0, \sigma_i^2) \implies \mathbb{P}[-2\sigma_i \leq \eta_i \leq 2\sigma_i] \geq 95\%$

Read like: "trust $0 \pm 2\sigma_i$". Adding the real number $y_i$: "trust $y_i \pm 2\sigma_i$. So we arbitrarily choose:

# The model in practice: noise

To assume an error of the form

$$y_i(\omega) = X^P_{X_0}(t_i) + \eta_i(\omega) \tag{3}$$

is a strong assumption (I know). Need to decide the standard variations $\sigma_i$ for each day. Abritrately.

Quantile formula: $\eta_i \sim N(0, \sigma_i^2) \implies \mathbb{P}[-2\sigma_i \leq \eta_i \leq 2\sigma_i] \geq 95\%$

Read like: "trust $0 \pm 2\sigma_i$". Adding the real number $y_i$: "trust $y_i \pm 2\sigma_i$. So we arbitrarily choose:

- Case 1: $\sigma_i$ s.t. we trust $y_i \pm \frac{y_i}{10}$, i.e. "10% of error";

# The model in practice: noise

To assume an error of the form

$$y_i(\omega) = X^P_{X_0}(t_i) + \eta_i(\omega) \tag{3}$$

is a strong assumption (I know). Need to decide the standard variations $\sigma_i$ for each day. Abritrately.

Quantile formula: $\eta_i \sim N(0, \sigma_i^2) \implies \mathbb{P}[-2\sigma_i \leq \eta_i \leq 2\sigma_i] \geq 95\%$

Read like: "trust $0 \pm 2\sigma_i$". Adding the real number $y_i$: "trust $y_i \pm 2\sigma_i$. So we arbitrarily choose:

- Case 1: $\sigma_i$ s.t. we trust $y_i \pm \frac{y_i}{10}$, i.e. "10% of error";
- Case 2: $\sigma_i$ s.t. we trust $y_i \pm y_i$, i.e. "100% of error";

# The model in practice: noise

To assume an error of the form

$$y_i(\omega) = X^P_{X_0}(t_i) + \eta_i(\omega) \tag{3}$$

is a strong assumption (I know). Need to decide the standard variations $\sigma_i$ for each day. Abritrately.

Quantile formula: $\eta_i \sim N(0, \sigma_i^2) \implies \mathbb{P}[-2\sigma_i \leq \eta_i \leq 2\sigma_i] \geq 95\%$

Read like: "trust $0 \pm 2\sigma_i$". Adding the real number $y_i$: "trust $y_i \pm 2\sigma_i$. So we arbitrarily choose:

- Case 1: $\sigma_i$ s.t. we trust $y_i \pm \frac{y_i}{10}$, i.e. "10% of error";
- Case 2: $\sigma_i$ s.t. we trust $y_i \pm y_i$, i.e. "100% of error";
- Warning: the inference algorithm [not yet described] is destroyed for higher errors, but not with our cases - checked with toy model data.

# Inference technique: MCMC

- We have the model $X^P(t)$, the noise structure and the data. Last step: choose how to infer $P \in \mathbb{R}^3$.

## Inference technique: MCMC

- We have the model $X^P(t)$, the noise structure and the data. Last step: choose how to infer $P \in \mathbb{R}^3$.
- in general, I am mainly into the topic "Bayesian Inverse Problems", where a common technique is the "Preconditioned Monte Carlo" algorithm (pCN);

# Inference technique: MCMC

- We have the model $X^P(t)$, the noise structure and the data. Last step: choose how to infer $P \in \mathbb{R}^3$.
- in general, I am mainly into the topic "Bayesian Inverse Problems", where a common technique is the "Preconditioned Monte Carlo" algorithm (pCN);
- most suitable for infinite-dimensional cases, large data, inversion of PDEs...

# Inference technique: MCMC

- We have the model $X^P(t)$, the noise structure and the data. Last step: choose how to infer $P \in \mathbb{R}^3$.
- in general, I am mainly into the topic "Bayesian Inverse Problems", where a common technique is the "Preconditioned Monte Carlo" algorithm (pCN);
- most suitable for infinite-dimensional cases, large data, inversion of PDEs...
- here: only 20 days, 3 real parameters, a simple 1-dim ODE with close form solution.

## Inference technique: MCMC

- We have the model $X^P(t)$, the noise structure and the data. Last step: choose how to infer $P \in \mathbb{R}^3$.
- in general, I am mainly into the topic "Bayesian Inverse Problems", where a common technique is the "Preconditioned Monte Carlo" algorithm (pCN);
- most suitable for infinite-dimensional cases, large data, inversion of PDEs...
- here: only 20 days, 3 real parameters, a simple 1-dim ODE with close form solution.
- But I was curious and insisted: let's use the pCN algorithm anyway!

## Inference technique: MCMC

- We have the model $X^P(t)$, the noise structure and the data. Last step: choose how to infer $P \in \mathbb{R}^3$.
- in general, I am mainly into the topic "Bayesian Inverse Problems", where a common technique is the "Preconditioned Monte Carlo" algorithm (pCN);
- most suitable for infinite-dimensional cases, large data, inversion of PDEs...
- here: only 20 days, 3 real parameters, a simple 1-dim ODE with close form solution.
- But I was curious and insisted: let's use the pCN algorithm anyway!
- In brief: we infer $P$ by using a Monte Carlo algorithm.

# The Bayesian setting: how to infer $P$?

- The idea is: *given* the data $\mathbf{y}$, find $P$. Equivalent: find $\mathbb{P}[P|\mathbf{y}]$.

## The Bayesian setting: how to infer $P$?

- The idea is: *given* the data $\mathbf{y}$, find $P$. Equivalent: find $\mathbb{P}[P|\mathbf{y}]$.
- Step 1: assume to have a probability density on $\mathbb{R}^3$, called the prior. Blind guess about $P$, i.e. $\mathbb{P}[P \in dx]$. (more later);

## The Bayesian setting: how to infer $P$?

- The idea is: *given* the data $\mathbf{y}$, find $P$. Equivalent: find $\mathbb{P}[P|\mathbf{y}]$.
- Step 1: assume to have a probability density on $\mathbb{R}^3$, called the prior. Blind guess about $P$, i.e. $\mathbb{P}[P \in dx]$. (more later);
- Step 2: interpret the read data as $\mathbb{P}[\mathbf{y}|P]$ (likelihood, obtain a complete formula by using the Gaussian noise, etc...)

## The Bayesian setting: how to infer $P$?

- The idea is: *given* the data $\mathbf{y}$, find $P$. Equivalent: find $\mathbb{P}[P|\mathbf{y}]$.
- Step 1: assume to have a probability density on $\mathbb{R}^3$, called the prior. Blind guess about $P$, i.e. $\mathbb{P}[P \in dx]$. (more later);
- Step 2: interpret the read data as $\mathbb{P}[\mathbf{y}|P]$ (likelihood, obtain a complete formula by using the Gaussian noise, etc...)
- Step 3: conclude by using the Bayes formula:

$$\mathbb{P}[P|\mathbf{y}] \propto \mathbb{P}[\mathbf{y}|P] \times \mathbb{P}[P] \tag{4}$$

where $\mathbb{P}[P|\mathbf{y}]$ is the posterior measure.

## The Bayesian setting: how to infer $P$?

- The idea is: *given* the data $\mathbf{y}$, find $P$. Equivalent: find $\mathbb{P}[P|\mathbf{y}]$.
- Step 1: assume to have a probability density on $\mathbb{R}^3$, called the prior. Blind guess about $P$, i.e. $\mathbb{P}[P \in dx]$. (more later);
- Step 2: interpret the read data as $\mathbb{P}[\mathbf{y}|P]$ (likelihood, obtain a complete formula by using the Gaussian noise, etc...)
- Step 3: conclude by using the Bayes formula:

$$\mathbb{P}[P|\mathbf{y}] \propto \mathbb{P}[\mathbf{y}|P] \times \mathbb{P}[P] \tag{4}$$

  where $\mathbb{P}[P|\mathbf{y}]$ is the posterior measure.
- In brief: fix a belief about $P$, i.e. the prior. Read the data, obtain the likelihood. Multiply them to have the posterior probability, containing information about the "true" $P$ given the data (details omitted on purpose).

## The Bayesian setting: how to infer $P$?

- The idea is: *given* the data $\mathbf{y}$, find $P$. Equivalent: find $\mathbb{P}[P|\mathbf{y}]$.
- Step 1: assume to have a probability density on $\mathbb{R}^3$, called the prior. Blind guess about $P$, i.e. $\mathbb{P}[P \in dx]$. (more later);
- Step 2: interpret the read data as $\mathbb{P}[\mathbf{y}|P]$ (likelihood, obtain a complete formula by using the Gaussian noise, etc...)
- Step 3: conclude by using the Bayes formula:

$$\mathbb{P}[P|\mathbf{y}] \propto \mathbb{P}[\mathbf{y}|P] \times \mathbb{P}[P] \tag{4}$$

  where $\mathbb{P}[P|\mathbf{y}]$ is the posterior measure.
- In brief: fix a belief about $P$, i.e. the prior. Read the data, obtain the likelihood. Multiply them to have the posterior probability, containing information about the "true" $P$ given the data (details omitted on purpose).
- **Infer $P$ = sample from the posterior**

# The pCN Monte Carlo method

Let $\mathbb{P}[P|\mathbf{y}]$ be the posterior on $\mathbb{R}^3$. Let $0 < \beta < 1$ the speed parameter, and the acceptance probability, for $v, u \in \mathbb{R}^3$:

$$a(u, v) \doteq \min\{1, \frac{\mathbb{P}(\mathbf{y}|v)}{\mathbb{P}(\mathbf{y}|u)}\} \tag{5}$$

To produce a **single sample**, 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 $R \in \mathbb{R}^3$ from the **gaussian** prior $\mathbb{P}(P \in dx)$;
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;

Run 120.000 multiple chains in parallel, different starting point, each stopped after 250.000 iterations. Beta around 0.05, s.t. acceptance rate $\sim 25\%$. Last missing info: which prior?

# Prior measure on $P$: part 1

Before moving to the numerical results, we need to clarify the structure of the prior measure. It influences the results.

- The pCN Monte Carlo algorithm *requires* a centered Gaussian prior (on $\mathbb{R}^3$).

## Prior measure on $P$: part 1

Before moving to the numerical results, we need to clarify the structure of the prior measure. It influences the results.

- The pCN Monte Carlo algorithm *requires* a centered Gaussian prior (on $\mathbb{R}^3$).
- Problematic: $P \in \mathbb{R}^3$ is strictly positive...so we just rejected the negative proposals...(pitfall?)...

## Prior measure on $P$: part 1

Before moving to the numerical results, we need to clarify the structure of the prior measure. It influences the results.

- The pCN Monte Carlo algorithm *requires* a centered Gaussian prior (on $\mathbb{R}^3$).
- Problematic: $P \in \mathbb{R}^3$ is strictly positive...so we just rejected the negative proposals...(pitfall?)...
- and we assumed a prior structure with covariance matrix

$$\begin{pmatrix} \sigma_q^2 & 0 & 0 \\ 0 & \sigma_Q^2 & 0 \\ 0 & 0 & \sigma_v^2 \end{pmatrix} \tag{6}$$

# Prior measure on $P$: part 1

Before moving to the numerical results, we need to clarify the structure of the prior measure. It influences the results.

- The pCN Monte Carlo algorithm *requires* a centered Gaussian prior (on $\mathbb{R}^3$).
- Problematic: $P \in \mathbb{R}^3$ is strictly positive...so we just rejected the negative proposals...(pitfall?)...
- and we assumed a prior structure with covariance matrix

$$\begin{pmatrix} \sigma_q^2 & 0 & 0 \\ 0 & \sigma_Q^2 & 0 \\ 0 & 0 & \sigma_v^2 \end{pmatrix} \tag{6}$$

- Therefore: we only need to choose the $\sigma$, and then we can move on the simulations!

# Prior measure on $P$: part 2

Goal of this slide: explain how the values for $\sigma_q$, $\sigma_Q$ and $\sigma_v$ have been chosen.

- The idea is: the prior represents the "guess", the "default" choice. But if "nothing is done", the phenomenon propagates exponentially. Therefore we run an exponential interpolation, and register the number of victims in 1 Month: $K$. Then choose $\sigma_Q$ in order to have the Gaussian 95% quantile just below this quantity;

# Prior measure on $P$: part 2

Goal of this slide: explain how the values for $\sigma_q$, $\sigma_Q$ and $\sigma_v$ have been chosen.

- The idea is: the prior represents the "guess", the "default" choice. But if "nothing is done", the phenomenon propagates exponentially. Therefore we run an exponential interpolation, and register the number of victims in 1 Month: $K$. Then choose $\sigma_Q$ in order to have the Gaussian 95% quantile just below this quantity;
- Dealing with $\sigma_q$ and $\sigma_v$, they have been chosen in order to have that quantile between 0 and 1.

# Prior measure on $P$: part 2

Goal of this slide: explain how the values for $\sigma_q$, $\sigma_Q$ and $\sigma_v$ have been chosen.

- The idea is: the prior represents the "guess", the "default" choice. But if "nothing is done", the phenomenon propagates exponentially. Therefore we run an exponential interpolation, and register the number of victims in 1 Month: $K$. Then choose $\sigma_Q$ in order to have the Gaussian 95% quantile just below this quantity;
- Dealing with $\sigma_q$ and $\sigma_v$, they have been chosen in order to have that quantile between 0 and 1.
- These idea formalize the intuition that we expect parameters in these range, without *forcing* any constraint.

## How to read the results

- Now all the parameters, assumptions, hypothesis should be clear. In particular, we are using a possibly-not-so-good Monte Carlo algorithm with strong assumptions on the noise and the prior;

# How to read the results

- Now all the parameters, assumptions, hypothesis should be clear. In particular, we are using a possibly-not-so-good Monte Carlo algorithm with strong assumptions on the noise and the prior;
- We remark: by reading the data we infer a 3-dimensional probability measure for $P$...

# How to read the results

- Now all the parameters, assumptions, hypothesis should be clear. In particular, we are using a possibly-not-so-good Monte Carlo algorithm with strong assumptions on the noise and the prior;
- We remark: by reading the data we infer a 3-dimensional probability measure for $P$...
- we clustered it by the k-means algorithm and selected three set of parameters:

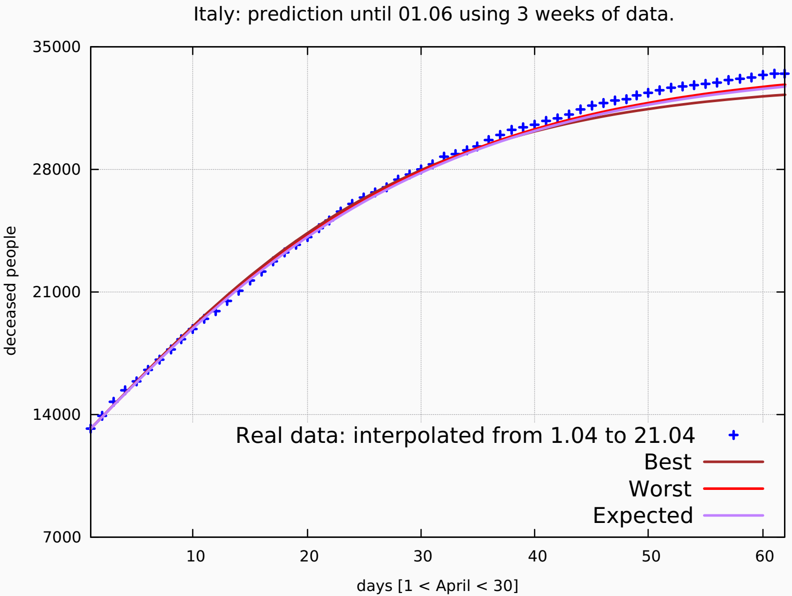
## How to read the results

- Now all the parameters, assumptions, hypothesis should be clear. In particular, we are using a possibly-not-so-good Monte Carlo algorithm with strong assumptions on the noise and the prior;
- We remark: by reading the data we infer a 3-dimensional probability measure for $P$...
- we clustered it by the k-means algorithm and selected three set of parameters:
- the worst case, i.e.the triple with the highest value for $Q$; the best (converse) and the expectation.

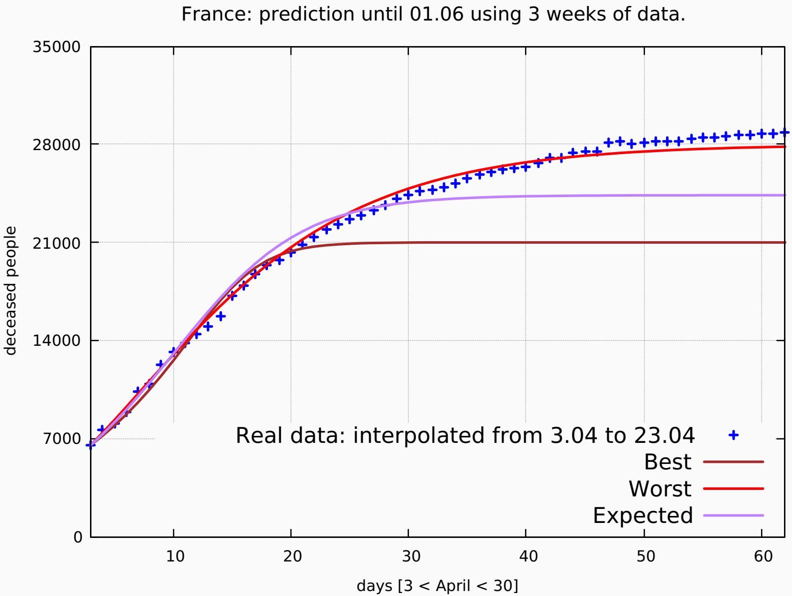
## How to read the results

- Now all the parameters, assumptions, hypothesis should be clear. In particular, we are using a possibly-not-so-good Monte Carlo algorithm with strong assumptions on the noise and the prior;
- We remark: by reading the data we infer a 3-dimensional probability measure for $P$...
- we clustered it by the k-means algorithm and selected three set of parameters:
- the worst case, i.e.the triple with the highest value for $Q$; the best (converse) and the expectation.
- We have plots for the "10%" error scenario, as well as for the "100%" (*meaning explained before*)

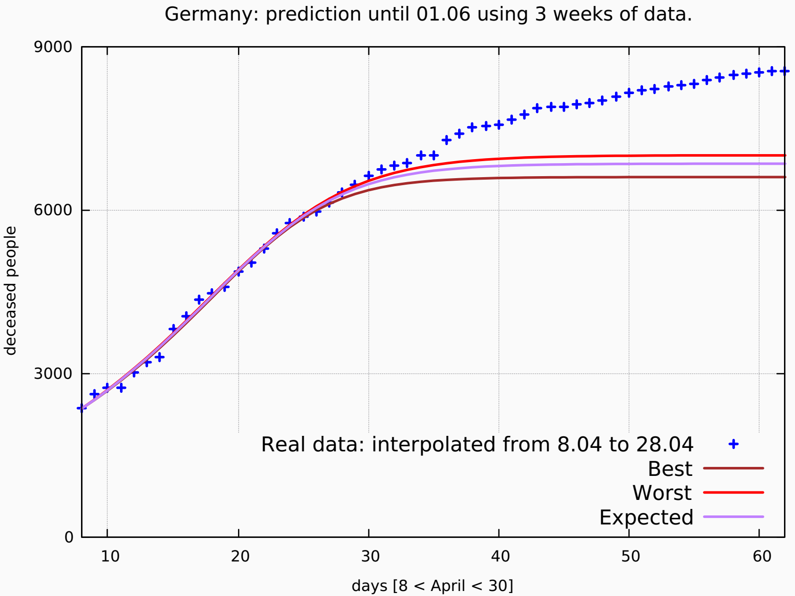
# Results: Italy, assuming $10\%$ of "error"



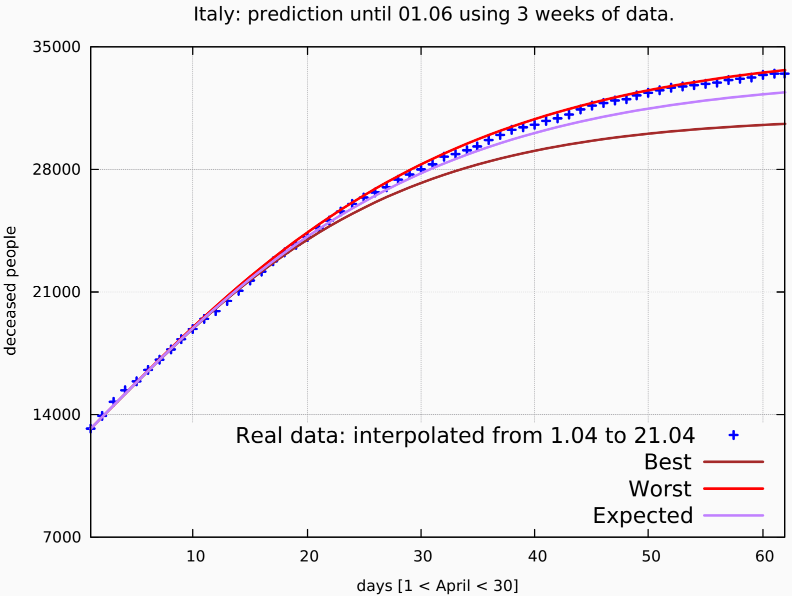
# Results: France, assuming $10\%$ of "error"



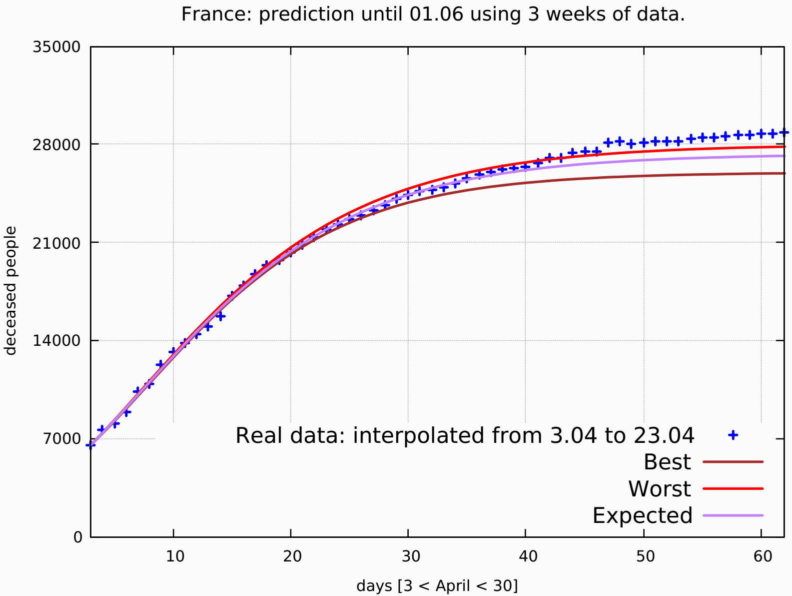
# Results: Germany, assuming 10% of "error"



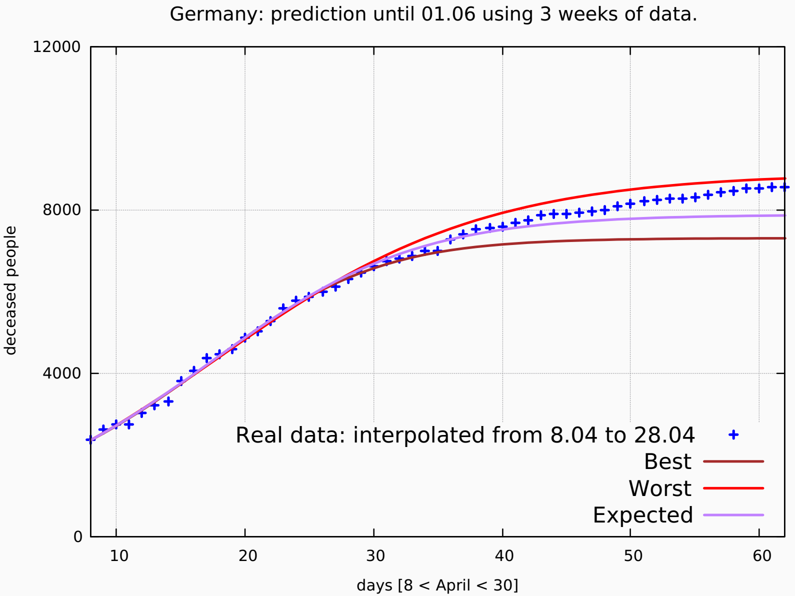
# Results: Italy, assuming $100\%$ of "error"



# Results: France, assuming $100\%$ of "error"



# Results: Germany, assuming $100\%$ of "error"



# Conclusions

Despite various theoretical issues, the results give some degree of satisfaction. But they *do not seem* to work with Spain, UK (I quickly tested).

- A next step can be to use the last 20 days of data to predict for June/July;

# Conclusions

Despite various theoretical issues, the results give some degree of satisfaction. But they *do not seem* to work with Spain, UK (I quickly tested).

- A next step can be to use the last 20 days of data to predict for June/July;
- Or try with another more appropriate Monte Carlo methods with better prior measures;

## Conclusions

Despite various theoretical issues, the results give some degree of satisfaction. But they *do not seem* to work with Spain, UK (I quickly tested).

- A next step can be to use the last 20 days of data to predict for June/July;
- Or try with another more appropriate Monte Carlo methods with better prior measures;
- Or instead of the Bayesian approach, try maximizing the Likelihood (i.e. probabilistic least square);

## Conclusions

Despite various theoretical issues, the results give some degree of satisfaction. But they *do not seem* to work with Spain, UK (I quickly tested).

- A next step can be to use the last 20 days of data to predict for June/July;
- Or try with another more appropriate Monte Carlo methods with better prior measures;
- Or instead of the Bayesian approach, try maximizing the Likelihood (i.e. probabilistic least square);
- Or simply to consider this example as a nice application of Monte Carlo Bayesian techniques, before moving back to the main PhD topic.

Thanks for the attention!