# Practical considerations for regression methods for stochastic control problems with utility functions

**ICCF24, Amsterdam**

Anthony Britto (KIT, Karlsruhe), Carlos Oliveira (NTNU, Trondheim) | April 2, 2024

# Problem statement



## Goal

Investigate the effects of energy price uncertainty on household consumption.

Let $t = 0, 1, \ldots, N$ be equispaced points in $[0, T]$ with $\Delta T = T/N$. Model the household's wealth dynamic as follows:

$$\Delta W_t = (\rho W_t + \mathcal{J} - x_t - c_t P_t)\Delta T\ , \tag{1}$$

where

- $W_t \geq 0$ is financial wealth;
- $\rho \geq 0$ is portfolio return and $\mathcal{J} \geq 0$ is labour income;
- $x_t \geq 0$ is non heating-energy consumption;
- $c_t \geq 0$ is heating-energy consumption;
- $P_t \geq 0$, the price of energy, follows a discrete geometric Brownian motion

$$\Delta P_t = \mu P_t \Delta T + \sigma P_t \Delta B_t\ , \tag{2}$$

with $\Delta B_t \sim \mathcal{N}(0, \sqrt{\Delta T})$.

# Problem statement



Utility is derived from consumption according to

$$U(x, c) = \frac{x^\gamma}{\gamma} \frac{(\eta c)^\delta}{\delta} \,, \tag{3}$$

where $\eta$ is the efficiency of the fuel to dwelling-warmth conversion.

# Problem statement



Utility is derived from consumption according to

$$U(x, c) = \frac{x^\gamma}{\gamma} \frac{(\eta c)^\delta}{\delta} , \tag{3}$$

where $\eta$ is the efficiency of the fuel to dwelling-warmth conversion.

Define the agent's value function $F_0(w, p)$ via

$$F_n(w, p) = \sup_{x_n, c_n} \mathbb{E}\left[ \sum_{i=n}^{N} \mathrm{e}^{-\beta(t_i - t_n)} U(x_n(W_i, P_i), c_n(W_i, P_i)) \Delta T \mid (W_n, P_n) = (w, p) \right] . \tag{4}$$

# Problem statement





Utility is derived from consumption according to

$$U(x, c) = \frac{x^\gamma}{\gamma} \frac{(\eta c)^\delta}{\delta} \,, \tag{3}$$

where $\eta$ is the efficiency of the fuel to dwelling-warmth conversion.

Define the agent's value function $F_0(w, p)$ via

$$F_n(w, p) = \sup_{x_n, c_n} \mathbb{E}\left[ \sum_{i=n}^{N} \mathrm{e}^{-\beta(t_i - t_n)} U(x_n(W_i, P_i), c_n(W_i, P_i))\Delta T \mid (W_n, P_n) = (w, p) \right] . \tag{4}$$

Solve with a backward iteration of the Bellman equation:

$$F_N(w, p) = \sup_{x_N, c_N} U(x_N(w, p), \eta c_N(w, p))\Delta T \,, \tag{5}$$

$$F_n(w, p) = \sup_{x_n, c_n} \left[ U(x_n, \eta c_n)\Delta T + \mathbb{E}\left[ \mathrm{e}^{-\beta \Delta T} F_{n+1}(W_{n+1}^{x_n, c_n}, P_{n+1}) \mid (W_n, P_n) = (w, p) \right] \right] . \tag{6}$$

# Forward simulation and backwards updating



The *Forward Simulation and Backward Updating* (FSBU) algorithm with random consumption[1] is the standard simulation approach to the above type of problem. Proceed as follows.

- Forward simulate from initial conditions $(w, p)$ assuming random controls.
- Initialise by solving for $F_N$ as per (5).
- For $n = N - 1, N - 2, \dots 0$ do the following:
  - estimate the conditional expectation in the Bellman equation is via a regression, then
  - update the simulated paths from step $n$ to $N$ by successive optimisation, specified again by the Bellman equation.

We illustrate this procedure in the following with a case study, discussing our suggestions along the way.

[1] Kharroubi, I., Langrené, N. & Pham, H. (2014). *A numerical algorithm for fully nonlinear HJB equations: An approach by control randomization.* Monte Carlo Methods and Applications, 20(2), 145-165.

# Forward simulation: random consumption



Firstly, reasonable bounds must be imposed on the "random" consumption.

This is not just due numerical restrictions, but because these paths are the only information that the regression models will be trained on ("garbage in, garbage out.")

The more structure is imposed on the simulation to begin with, the more convincing the results (of course a balance must be struck between *a priori* structure and optimisation).

# Forward simulation: random consumption



Firstly, reasonable bounds must be imposed on the "random" consumption.

This is not just due numerical restrictions, but because these paths are the only information that the regression models will be trained on ("garbage in, garbage out.")

The more structure is imposed on the simulation to begin with, the more convincing the results (of course a balance must be struck between *a priori* structure and optimisation).

For our problem, consider the *share of consumption*, defined as follows:

$$s(w,p) = \frac{(x(w,p) + c(w,p)p)\Delta T}{w} . \tag{7}$$

Since all wealth is consumed at the end, $s_N = \Delta T$. It is reasonable to postulate $s_n \leq s_{n+1}$.

We simulate accordingly, drawing from uniform distributions subject to this restriction.

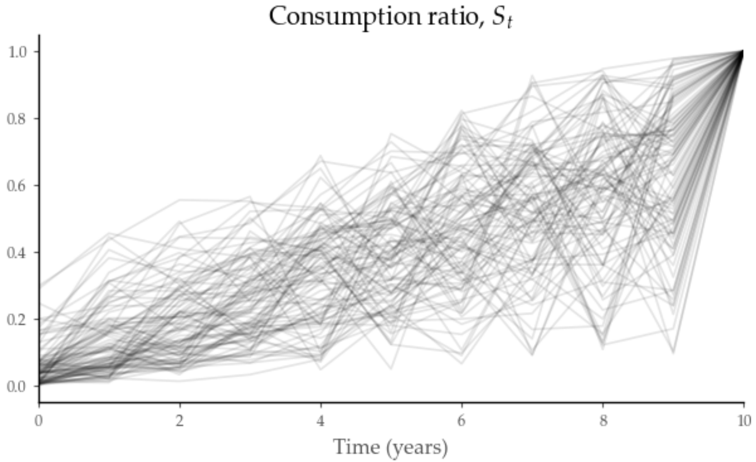
$$s_n \sim \mathrm{Unif}(0, s_{n+1}) , \tag{8}$$

$$x_n \sim \mathrm{Unif}(0, s_n(\Delta T)^{-1} w_n) , \tag{9}$$

$$c_n \sim \mathrm{Unif}(0, (s_n(\Delta T)^{-1} w_n - x_n)/p_n) . \tag{10}$$

# Forward simulation: illustration

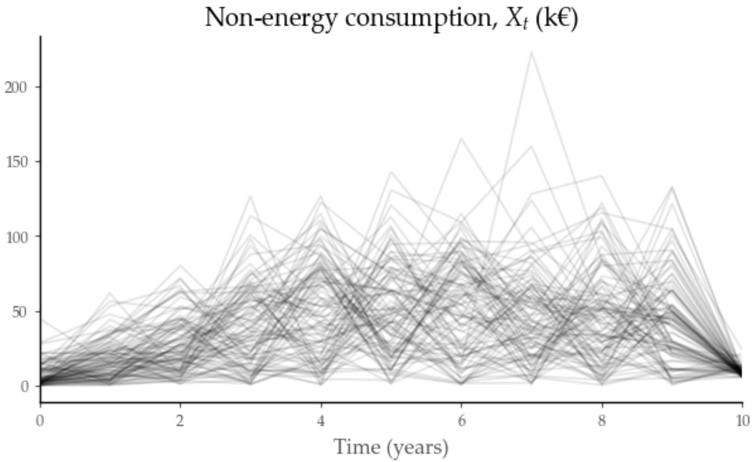





*Random consumption share, subject to* $s_n \leq s_{n+1}$ *for each n.*

# Forward simulation: illustration

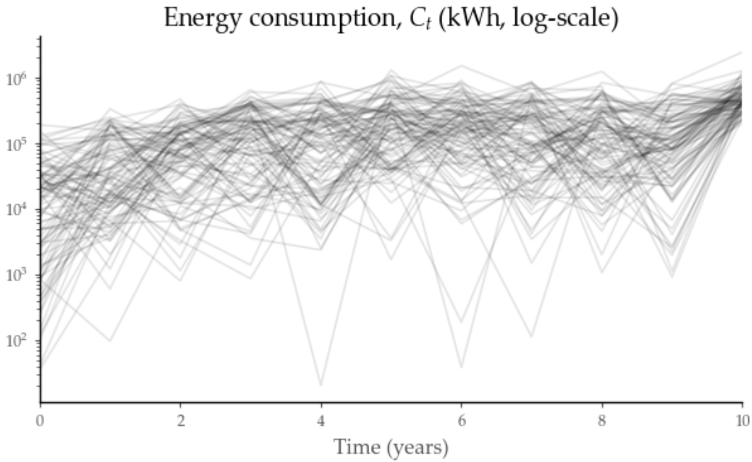





*Non-energy consumption, subject to* $x_n \leq s_n (\Delta T)^{-1} w_n$

# Forward simulation: illustration

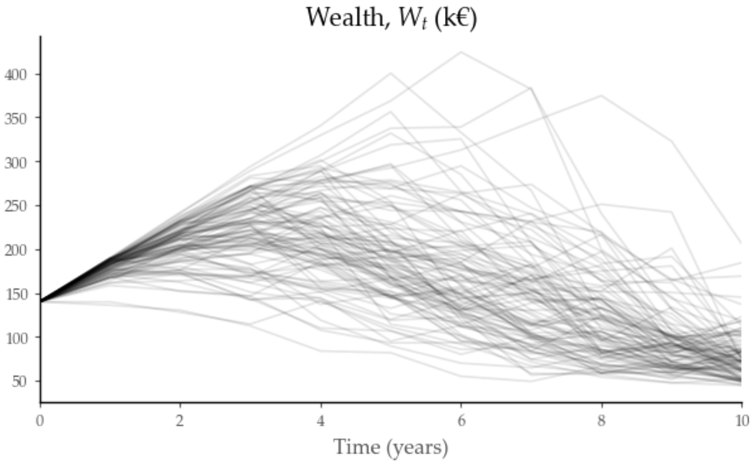





*Non-energy consumption, subject to* $c_n \leq (s_n(\Delta T)^{-1} w_n - x_n)/p_n$

# Forward simulation: illustration






*Resulting wealth evolution.*

# Backward updating: Regression and optimisation



Regarding the regression

$$\mathbb{E}\left[\mathrm{e}^{-\beta\Delta T}F_{n+1}\right] = \widehat{\phi}_n\left(W_n, P_n, X_n, C_n\right) + \epsilon_n \tag{11}$$

we make two suggestions.

The first concerns the choice of regression model and basis. Typically, (orthogonal) polynomials in the state variables are employed.

- These models are inherently susceptible to overfitting.
- The extreme multicollinearity strongly affects the stability of the estimated model.

Whereas this may not present difficulties in optimal stopping problems, in optimal control problems, a high-degree polynomial is likely to cause trouble in the optimisation.

# Backward updating: Regression and optimisation



Regarding the regression

$$\mathbb{E}\left[\mathrm{e}^{-\beta\Delta T}F_{n+1}\right] = \widehat{\phi}_n\left(W_n, P_n, X_n, C_n\right) + \epsilon_n \tag{11}$$

we make two suggestions.

The first concerns the choice of regression model and basis. Typically, (orthogonal) polynomials in the state variables are employed.

- These models are inherently susceptible to overfitting.
- The extreme multicollinearity strongly affects the stability of the estimated model.

Whereas this may not present difficulties in optimal stopping problems, in optimal control problems, a high-degree polynomial is likely to cause trouble in the optimisation.

*Non-parametric regression models* are a promising alternative, and performed well in our tests: we mention in particular *Gaussian processes*, and *gradient-boosted trees*.

The former allows for an infinite-dimensional basis via the kernel trick. The latter are quite robust against overfitting, and proved stable in optimisation.

Both cope well with the non-linearity of the utility function.

# Backward updating: Regression and optimisation



The second suggestion is somewhat specific to problems involving utility functions: instead of estimating (11), estimate instead

$$\mathbb{E}\left[\mathrm{e}^{-\beta\Delta T}F_{n+1}\right] = \widehat{\phi}_n\left(\rho W_n + \mathcal{J} - X_n - C_n P_n, P_n\right) + \epsilon_n \, . \tag{12}$$

With this natural choice of basis (income and price) the first-order condition of the optimisation problem in the Bellman equation reduces to

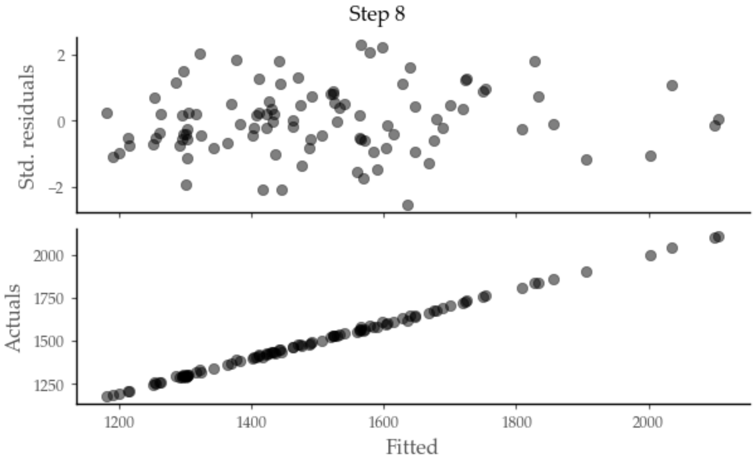
$$\partial_x U(x, \eta c) - \frac{\partial_c U(x, \eta c)}{p} = 0 \, , \tag{13}$$

i.e. the agent allocates consumption so that marginal utilities per unit expenditure coincide.

Equation (13) can often be solved explicitly for classical utility functions, reducing the dimension of the optimisation problem by one.

# Backward updating: Regression plot






*Exemplary regression diagnostic plot for a gradient-boosted regression model.*

# FSBU: Exemplary results

Problem statement

FSBU algorithm

Forward simulation

Regression / optimisation

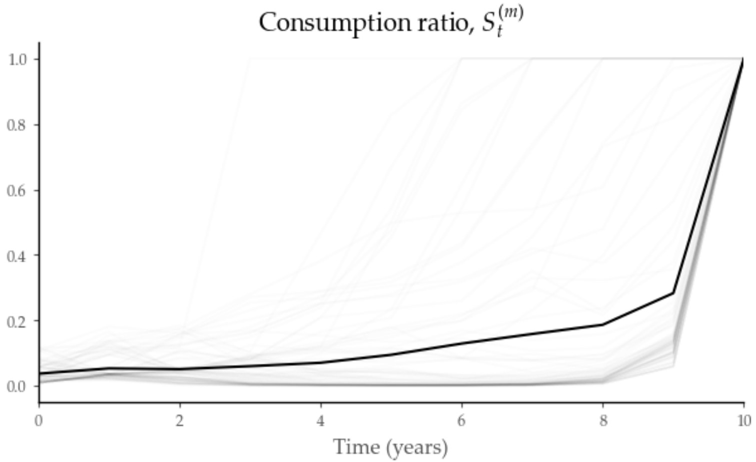
Results

BSBU algorithm

Description

Illustration

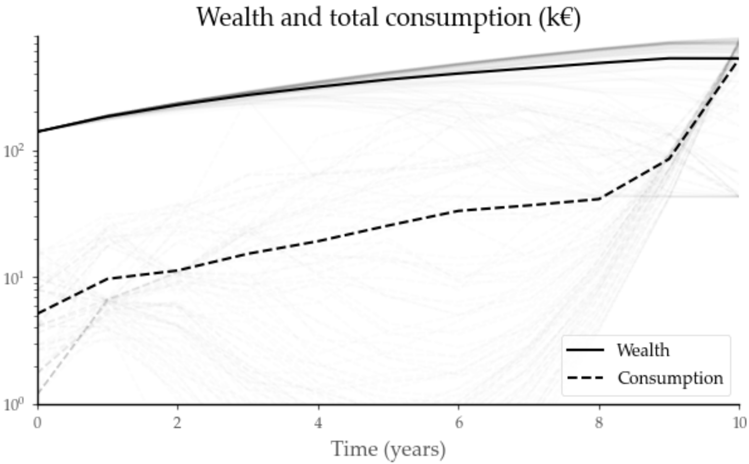
Conclusions




*Optimised consumption ratio $\widehat{s}_n$. The consumption rules $\widehat{x}_n$ and $\widehat{c}_n$ can now easily be derived.*

# FSBU: Exemplary results






*Optimised wealth and total consumption.*

# From FSBU to BSBU



The FSBU algorithm is an established method with a proven track record.

However, two limitations in particular promoted a search for alternatives:

- Due to the repeated forward simulation, the FSBU algorithm is computationally expensive and prohibitively slow.
- By design, the functions $x_n, c_n, F_n$ are estimated over smaller and smaller regions of the $(w, p)$ domain, with $x_0, c_0, F_0$ finally being (noisily) estimated at a *single point.*

The *Backwards Simulation and Updating* (BSBU) algorithm is a recent proposal that attempts to address these and other limitations of the FSBU algorithm.[2]

[2] Zhiyi Shen (2019). *Numerical Solutions to Stochastic Control Problems: When Monte Carlo Simulation Meets Nonparametric Regression* [Doctoral dissertation, University of Waterloo]. UWSpace.

# BSBU: Description

Karlsruhe Institute of Technology



*Idea.* Since the Bellman equation is solved backwards, exploit time symmetry and simulate and update *backwards* on-the-go.

This avoids the costly forward-updating of FSBU.

# BSBU: Description





*Idea.* Since the Bellman equation is solved backwards, exploit time symmetry and simulate and update *backwards* on-the-go.

This avoids the costly forward-updating of FSBU.

*Implementation.* The main difference to FSBU is the following: the wealth and price dynamics are simulated backwards by solving the implicit equations (superscript $(m)$ denotes a given scenario):

$$P_{n+1}^{(m)} = P_n^{(m)} + \mu P_n^{(m)} \Delta T + \sigma P_n^{(m)} \Delta B_n^{(m)} \tag{14}$$

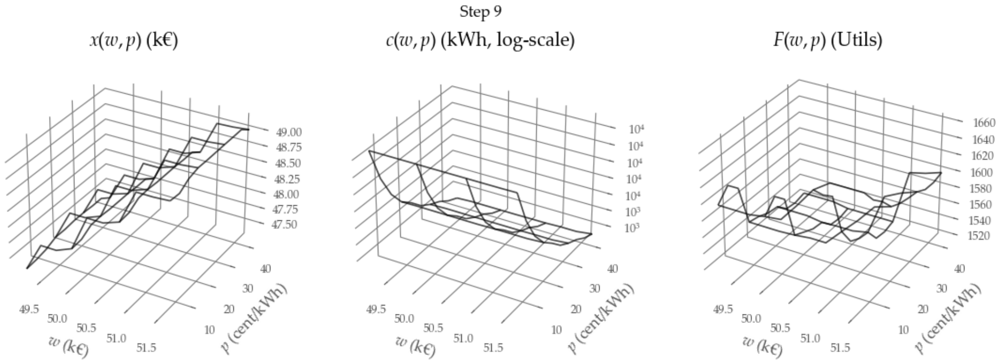
$$W_{n+1}^{(m)} = W_n^{(m)} + (\rho W_n^{(m)} + \mathcal{J} - \widehat{x}_{n+1}^{(m)}(W_n^{(m)}, P_n^{(m)}) - \widehat{c}_{n+1}^{(m)}(W_n^{(m)}, P_n^{(m)}))\Delta T\,. \tag{15}$$

The regression and optimisation are then performed exactly as in FSBU to estimate $\widehat{x}_n$ and $\widehat{c}_n$.

The paths are updated by repeating (15), but now with $\widehat{x}_n$ and $\widehat{c}_n$.

# BSBU: Illustration

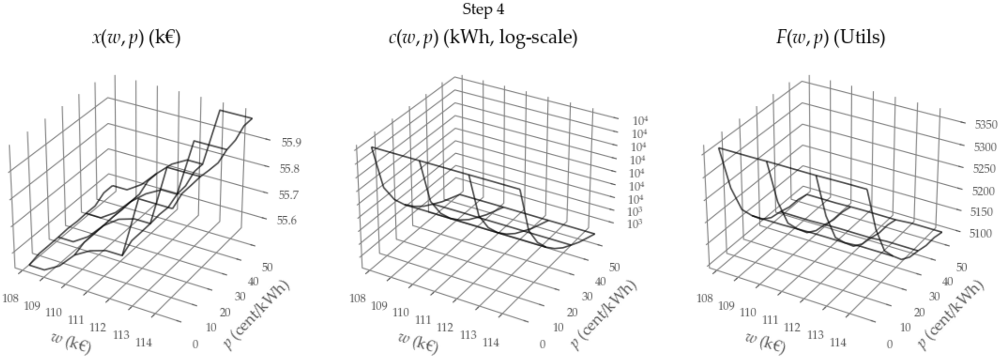





*Initial estimation. Domain quite small, value function estimate rather noisy.*

# BSBU: Illustration

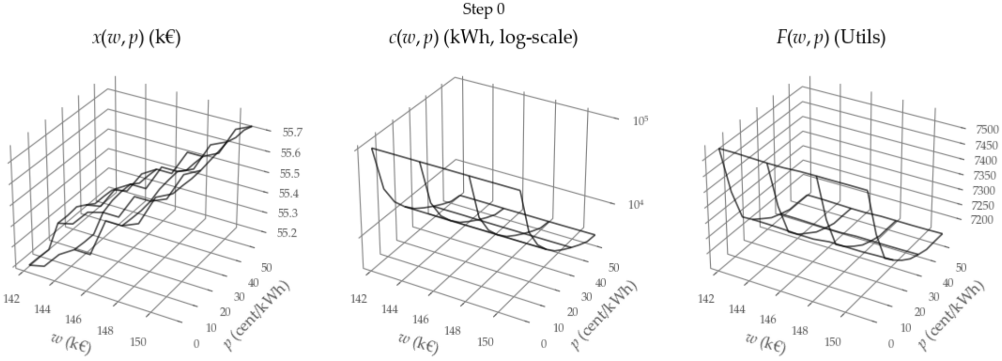





*The sub-domain moves as we iterate backwards.*

# BSBU: Illustration






*Final step. Obtain functions* $\widehat{x}_0, \widehat{c}_0, \widehat{F}_0$ *instead of point estimates.*

# Conclusions



In our experience, the following has proved helpful for FSBU.

- Impose meaningful constraints on the "random" consumption.
- Use non-parametric regression.
- If natural combinations of variables occur in the problem, these are likely good candidates for basis vectors.

# Conclusions

Karlsruhe Institute of Technology



In our experience, the following has proved helpful for FSBU.

- Impose meaningful constraints on the "random" consumption.
- Use non-parametric regression.
- If natural combinations of variables occur in the problem, these are likely good candidates for basis vectors.

We also discussed BSBU as an alternative to FSBU. Its main advantages are speed, and the ability to cover larger subsets of the domain.

On the other hand, BSBU is much more suseptible to numerical instability ("blow up") due to the repeated implicit-equation–solving.

Care is also required to ensure that the backwards iteration eventually ends up in the subset of the domain that we care about.