Structural Estimation by Homotopy Continuation

Philipp Müller
Gregor Reich
University of Zurich

September 13, 2019

Introduction

Given Two parameterized models (1, 2) with parameters (θ_A, θ_B) and some observed data.

Estimation Find the maximum likelihood estimates of θ_A and θ_B s.t. the model best explains the data.

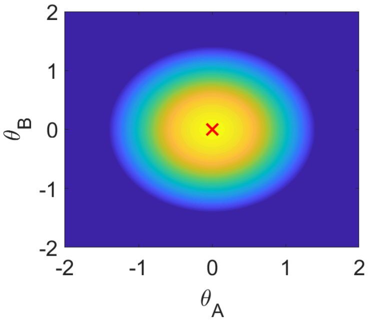


Figure: Model 1: Likelihood.

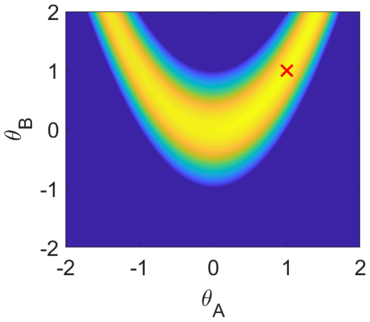


Figure: Model 2: Likelihood.

Introduction

Given Two parameterized models (1, 2) with parameters (θ_A, θ_B) and some observed data.

Estimation Find the maximum likelihood estimates of θ_A and θ_B s.t. the model best explains the data.

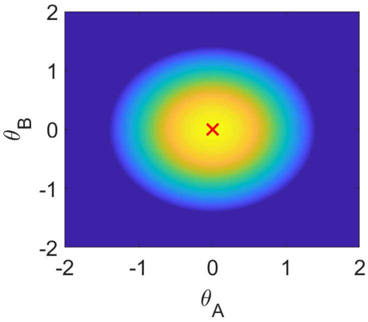


Figure: Model 1: Likelihood.

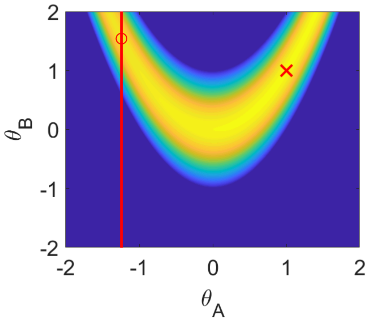


Figure: Model 2: Likelihood.

Introduction

Given Two parameterized models (1, 2) with parameters (θ_A, θ_B) and some observed data.

Estimation Find the maximum likelihood estimates of θ_A and θ_B s.t. the model best explains the data.

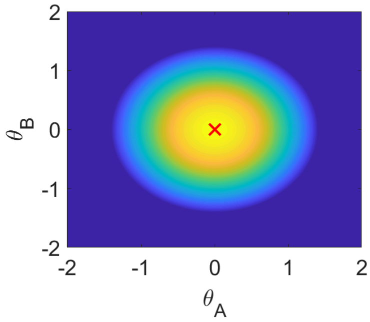


Figure: Model 1: Likelihood.

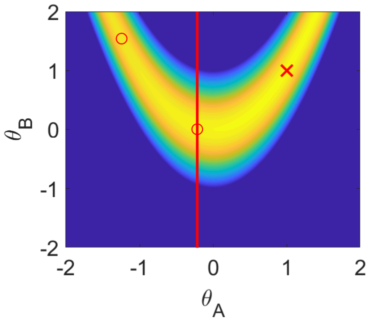


Figure: Model 2: Likelihood.

Introduction

Given Two parameterized models (1, 2) with parameters (θ_A, θ_B) and some observed data.

Estimation Find the maximum likelihood estimates of θ_A and θ_B s.t. the model best explains the data.

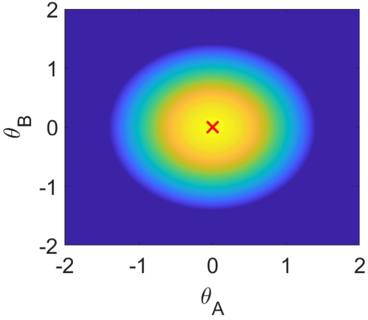


Figure: Model 1: Likelihood.

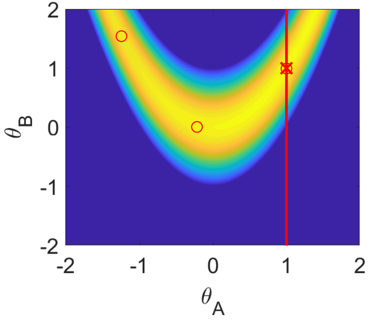


Figure: Model 2: Likelihood.

Introduction

Given Two parameterized models (1, 2) with parameters (θ_A, θ_B) and some observed data.

Estimation Find the maximum likelihood estimates of θ_A and θ_B s.t. the model best explains the data.

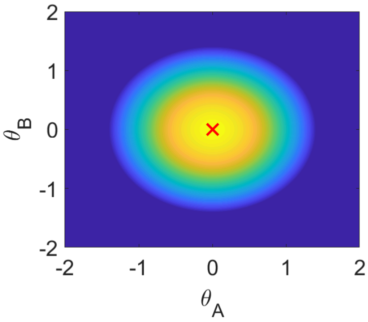


Figure: Model 1: Likelihood.

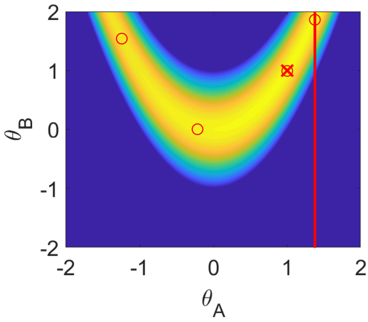


Figure: Model 2: Likelihood.

Introduction

Given Two parameterized models (1, 2) with parameters (θ_A, θ_B) and some observed data.

Estimation Find the maximum likelihood estimates of θ_A and θ_B s.t. the model best explains the data.

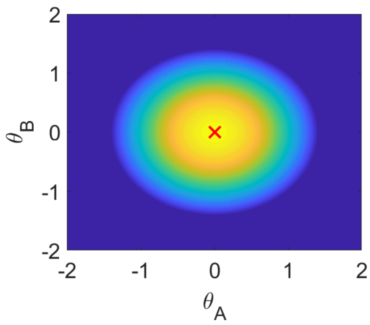


Figure: Model 1: Likelihood.

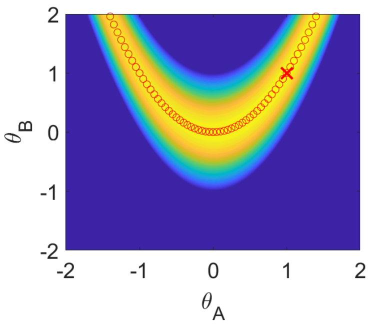


Figure: Model 2: Likelihood.

Introduction

- We focus on dynamic discrete choice models where the discount parameter β is generally considered to be poorly identified.
- We propose to formulate
  - the structural estimation as **parameterized** constrained optimization, i.e., parameterized version of Su and Judd [2012]
  - and solve this efficiently by **homotopy parameter continuation**.
- This novel approach enables the econometrician to computationally efficiently
  - estimate the structural parameters **even in models with one poorly identified parameter** and
  - perform inference based on the full (profile) likelihood function (not only point estimates).

The Bus Engine Replacement Model (Rust, 1987)

John Rust: *Optimal replacement of GMC bus engines: An empirical model of Harold Zurcher.* Econometrica, 1987.

Utility Function

- Agent's utility + shock for the single period payoff

$$u(x_t, d_t; \theta_1, \mathrm{RC}) + \epsilon_t(d_t) = \begin{cases} -c(x_t, \theta_1) + \epsilon_t(0) & \text{if } d_t = 0 \\ -RC + \epsilon_t(1) & \text{if } d_t = 1 \end{cases}$$

  - $d_t = 0$: performing regular maintenance
  - $d_t = 1$: replacing the engine
- State variables
  - x_t mileage state
  - ϵ i.i.d. gumbel utility shock (only observed by agent)
- **Parameters**
  - θ_1 regular maintenance cost parameter
  - RC replacement cost parameter

Value Function - Regenerative Optimal Stopping

Objective The agent wants to maximize his expected discounted utility over an **infinite horizon**.

$$V_{\theta,\beta}(x_t,\epsilon_t) = \max_{D(x_t)\in\mathcal{D}} \mathbb{E}\left[\sum_{j=t}^{\infty} \beta^{j-t}\left(u(x_j, D(x_j);\theta_1,\mathrm{RC}) + \epsilon(D(x_j))\right)\Big| x_t\right]$$

where $\theta \equiv (\mathrm{RC},\theta_1)$ and $D(\cdot)$ denotes the policy function.

Bellman $V_{\theta,\beta}$ is the unique solution to the Bellman equation

$$V_{\theta,\beta}(x,\epsilon) = \max_{d\in\{0,1\}} [u(x,d,\theta_1) + \epsilon(d) + \beta\mathbb{E}[V_{\theta,\beta}(x',\epsilon')|x,d]],$$

where x' and ϵ' denote the next period state variables.

Preference $\beta = 1$: Maximize the long-run average utility [Bertsekas, 2012].

$\beta > 1$: Maximize today's and future utility - "future-bias" [Blom Västberg and Karlström, 2017].

Relative Value Iteration

- After discretizing the mileage x_t into 90 states, we solve for the expected value vector $\overline{V} \in \mathbb{R}^{90}$.
- The classic value iteration solves for the (expected) value by

$$\overline{V} = T_{\theta,\beta}(\overline{V}),$$

  with $T_{\theta,\beta}(\cdot)$ denoting the Bellman operator.
- However, $\overline{V} \to \infty$ for $\beta \to 1$.
- To mitigate this, we use the relative value fixed-point equation Bertsekas [2012] for normalization as

$$h = T_{\theta,\beta}(h) - T_{\theta,\beta}(h)_1 \tag{1}$$

  with $h = \overline{V} - \overline{V}_1$ and $h \in \mathbb{R}^{90}$.

Structural Estimation

Objective Identify the most likely values for the parameters $\theta = (\theta_1, \mathrm{RC})$ and β given the observed data.

Data $\sim$ 8000 observations of the state and control variables ($=$ mileage states and replacement decisions).

Approach Simultaneously solve the **likelihood** and **fixed-point** problem

$$\theta^*, \beta^* = \arg\max_{\theta,\beta} L(h, \theta, \beta; \{x_t, d_t\})$$

$$h = T_{\theta,\beta}(h) - T_{\theta,\beta}(h)_1|_{\theta=\theta^*,\beta=\beta^*}$$

- Two popular solution methods are the nested fixed-point algorithm (NFXP) Rust [1987] and the mathematical programming with equilibrium constraints by Su and Judd [2012]

MPEC

- Su and Judd [2012] formulate the structural estimation as constrained optimization

$$\max_{(h,\theta,\beta)} L(\theta, \beta, h; \{x_t, d_t\}),$$
$$\text{s.t. } h = T_{\theta,\beta}(h) - T_{\theta,\beta}(h)_1,$$

with $\theta \in \mathbb{R}^2$, $\beta \in \mathbb{R}_+$, and $h \in \mathbb{R}^{90}$.

MPEC

- Su and Judd [2012] formulate the structural estimation as constrained optimization

$$\max_{(h,\theta,\beta)} L(\theta, \beta, h; \{x_t, d_t\}),$$
$$\text{s.t. } h = T_{\theta,\beta}(h) - T_{\theta,\beta}(h)_1,$$

with $\theta \in \mathbb{R}^2$, $\beta \in \mathbb{R}_+$, and $h \in \mathbb{R}^{90}$.

- If β is poorly identified, its Hessian becomes nearly singular and the maximum likelihood estimation numerically hard.

MPEC

- Su and Judd [2012] formulate the structural estimation as constrained optimization

$$\max_{(h,\theta,\cancel{\beta})} L(\theta, \beta, h; \{x_t, d_t\}),$$
$$\text{s.t. } h = T_{\theta,\beta}(h) - T_{\theta,\beta}(h)_1,$$

with $\theta \in \mathbb{R}^2$, $\beta \in \mathbb{R}_+$, and $h \in \mathbb{R}^{90}$.

- If β is poorly identified, its Hessian becomes nearly singular and the maximum likelihood estimation numerically hard.
- Thus, β is often **calibrated** to some value.

Profile Likelihood

- The profile likelihood expresses the maximum likelihood estimates as parametric maximum likelihood estimates w.r.t. a **controlled parameter**
- By setting β as controlled parameter we define

$$L_p(\beta) = \max_{\theta,h} L(\theta, h; \{x_t, d_t\}, \beta)$$
$$\text{s.t. } h = T_{\theta,\beta}(h) - T_{\theta,\beta}(h)_1,$$

i.e., we optimize w.r.t. all parameters **but** the controlled parameter β

First-Order Necessary Optimality Conditions

- Its Lagrangian $\mathcal{L}$ is defined as

$$\mathcal{L}(\theta, h, \boldsymbol{\mu}) = L(\theta, h; \beta) - \sum_i \mu_i \left(h - T_{\theta,\beta}(h) + T_{\theta,\beta}(h)_1\right)$$

- If (θ^*, h^*) is a local optimal solution to $L_p(\beta)$, where the LICQ holds, then there exists a unique μ^* s.t.

$$\nabla_{(\theta,h,\mu)}\mathcal{L}(\theta^*, h^*, \mu^*; \beta) = 0. \tag{2}$$

- **We solve** (2) **parametrically by homotopy parameter continuation for** $\beta \in [a, b]$**.**

PMPEC - Summary

- Consider the structural estimation as constrained optimization

$$\max_{(h,\theta)} L(\theta, \beta, h; \{x_t, d_t\}),$$
$$\text{s.t. } h = T_{\theta,\beta}(h) - T_{\theta,\beta}(h)_1,$$

- First-order necessary conditions (Lagrange) form a SE, parametrized by β

$$\nabla_{(\theta,h,\mu)}\mathcal{L}(\theta^*, h^*, \mu^*; \beta) = 0.$$

- We are interested in the solution manifold of the parametrized FOC (profile likelihood as implicit function)

$$c \equiv \{\beta, \theta, h, \mu : \nabla_{\theta,h,\mu}\mathcal{L}(\beta, \theta, h, \mu) = 0\}$$

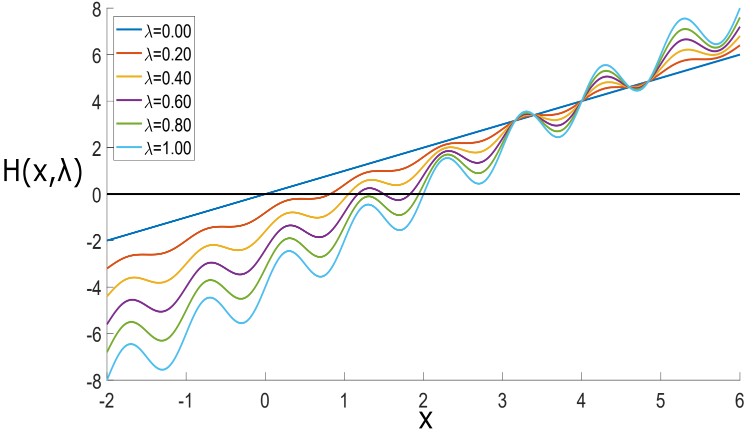
Homotopy Parameter Continuation

Objective Suppose we want to solve $H(x, \lambda) = 0$ for $\lambda \in [0, 1]$.

Approach Starting from the initial solution $(x_0, \lambda = 0)$, we follow the solution manifold.
EXPLAIN d H / d s

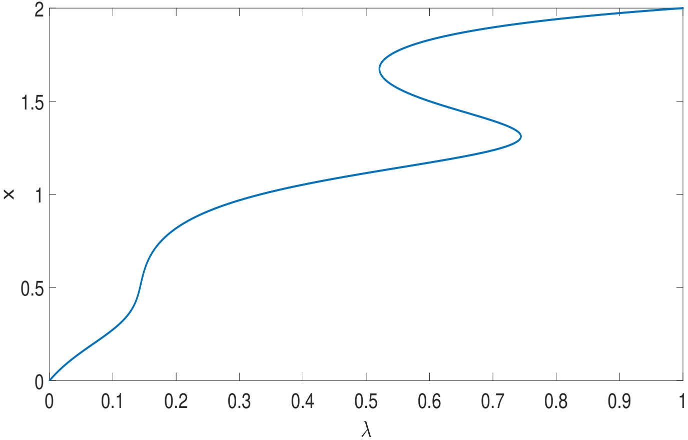
An Example

$$H(x, \lambda) = \lambda\,(x - 4 + \sin(2\pi x)) + x$$



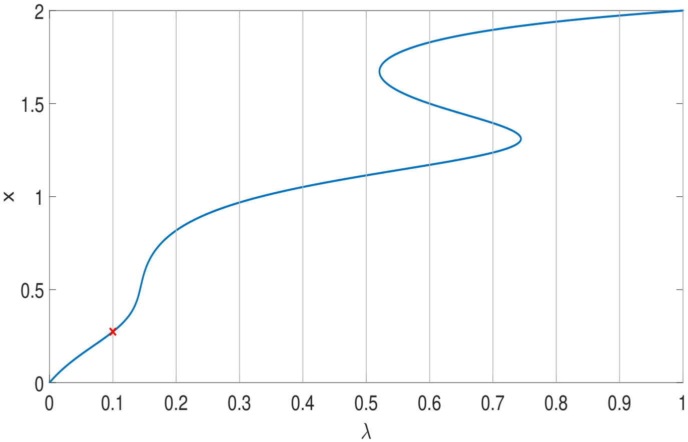
Simple Continuation

$$c := \{(x, \lambda) : H(x, \lambda) = 0\}$$



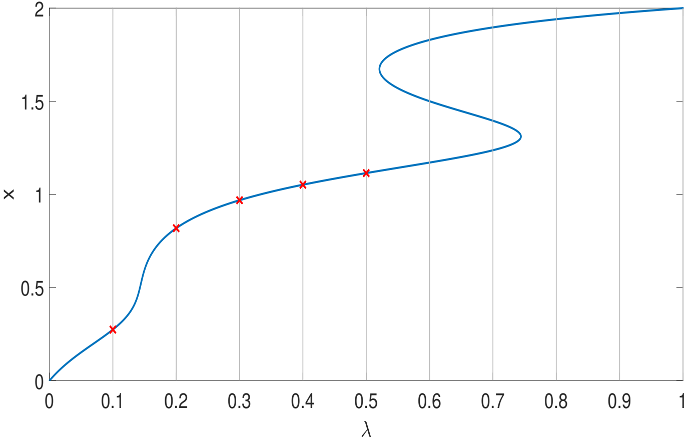
Simple Continuation

$$H(x,\lambda) = \textcolor{red}{\lambda}\,(x - 4 + \sin(2\pi x)) + x = 0$$



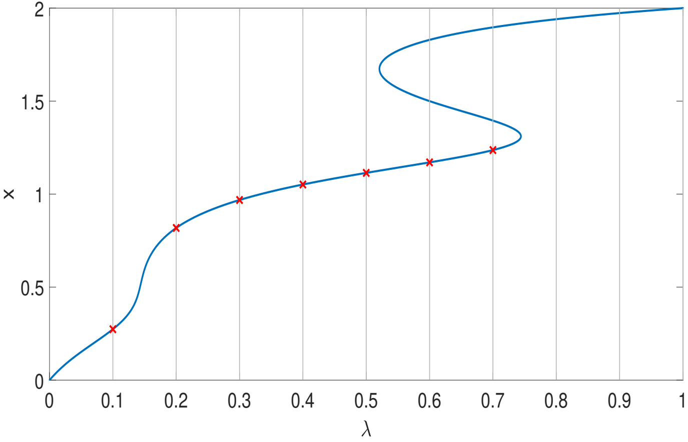
Simple Continuation

$$H(x, \lambda) = \lambda\,(x - 4 + \sin(2\pi x)) + x = 0$$



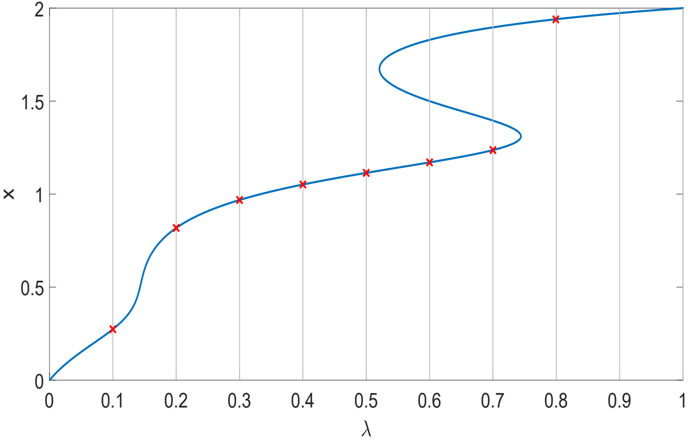
Simple Continuation

$$H(x,\lambda) = \lambda\,(x - 4 + \sin(2\pi x)) + x = 0$$



Simple Continuation

$$H(x,\lambda) = \lambda\,(x - 4 + \sin(2\pi x)) + x = 0$$



Towards Predictor Corrector Methods

Objective Find all solutions to $H(x, \lambda) = 0$, by tracing the curve $c := \{(x, \lambda) : H(x, \lambda) = 0\}$.

Approach Use the **arclength** s as parameterisation for the curve c.

$\Rightarrow$ The homotopy map changes to $H(x(s), \lambda(s)) = 0$!

Towards Predictor Corrector Methods: ODE-Theory

Objective Find the solution to $H(x, \lambda) = 0$ *for all* λ by tracing the curve $c := \{(x, \lambda) : H(x(s), \lambda(s)) = 0\}$.

- Differentiating $H(x(s), \lambda(s))$ w.r.t. s, yields the initial and boundary value problem (IBVP)

$$x(0) = x_0, \quad \lambda(0) = 0, \quad ||(x'(s), \lambda'(s))||_2^2 = 1, \tag{3}$$

$$\frac{\partial H(x(s), \lambda(s))}{\partial x} x'(s) + \frac{\partial H(x(s), \lambda(s))}{\partial \lambda} \lambda'(s) = 0. \tag{4}$$

- ODE-theory algorithms can solve the IBVP (3) - (4) to follow the curve c closely.

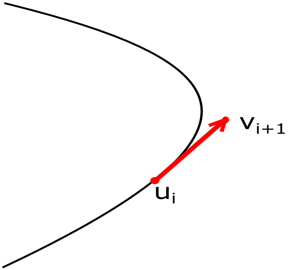
Predictor Corrector Methods: Algorithm

Approach Trace c by *alternating* **prediction** and **correction** steps.

Predictor Use e.g., Euler's explicit step to predict

$$v_{i+1} = u_i + h \cdot H'(x(s_i), \lambda(s_i)).$$

Corrector Use the predicted point v_{i+1} and improve prediction by e.g., Newton-type methods.



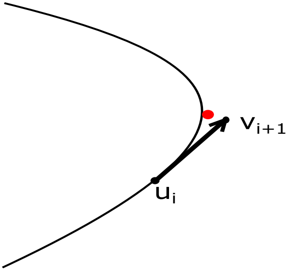
Predictor Corrector Methods: Algorithm

Approach Trace c by *alternating* **prediction** and **correction** steps.

Predictor Use e.g., Euler's explicit step to predict

$$v_{i+1} = u_i + h \cdot H'(x(s_i), \lambda(s_i)).$$

Corrector Use the predicted point v_{i+1} and improve prediction by e.g., Newton-type methods.



Finite Differences

- We know that we can approximate any analytic function $f \in C^\omega : \mathbb{R} \to \mathbb{R}$ around a as

$$f(a+h) = f(a) + h \frac{\partial f(x)}{\partial x}_{|x=a} + \frac{h^2}{2} \frac{\partial f(x)}{\partial x}_{|x=a} + \dots \qquad (5)$$

- Truncating and rearanging yields the well-known forward differences equation

$$\frac{\partial f(x)}{\partial x}_{|x=a} = \frac{f(a+h) - f(a)}{h} + O(h) \qquad (6)$$

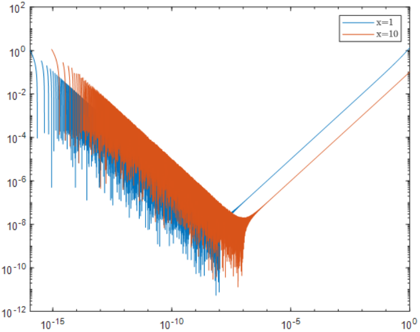
- Problem solved by using $\lim_{h \to 0}$?

Finite Differences

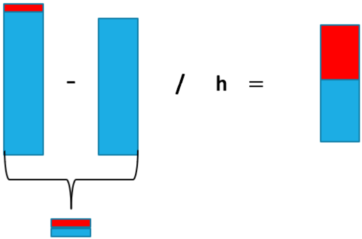
Apply forward differences to

$$f(x) = x^3$$

and decrease step size from 1 to 10^{-16}



Cancellation Error



$$h = 1E - 12$$
$$a = 1$$

$$b = a + h$$
$$c = b - a$$
$$d = c/h$$

$$d = 0.999201$$

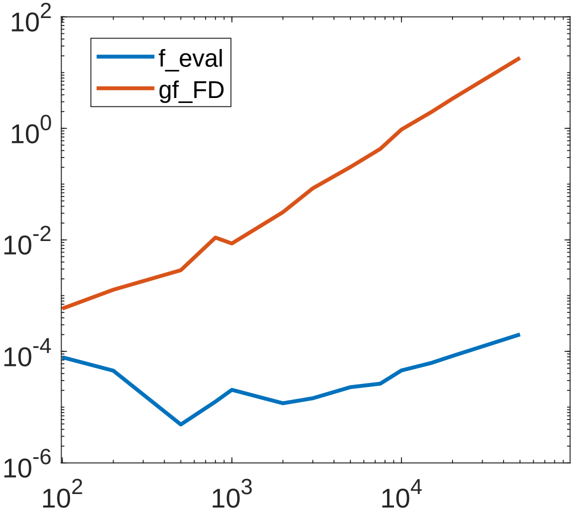
Scaling

Let's consider the Rosenbrock function $f : \mathbb{R}^n \to \mathbb{R}$ as benchmark.

$$f(x) = \sum_{i=1}^{n-1} 10(x_{i+1} - x_i^2)^2 + (1 - x_i)^2.$$

Finite differences for $f : \mathbb{R}^n \to \mathbb{R}$ become **directional derivates**.

Scaling



Automatic Differentiation

- Every function implementation is a composition of
  - basic arithmetic operations as, e.g., $+$, - etc.
  - and basic functions as, e.g., sin, cos and tan.
- Automatic differentiation (AD) **transforms the source code** of our functions into the gradient by applying the chain rule of differentiation to the function code until it is only faced with derivatives of basic functions and operations!

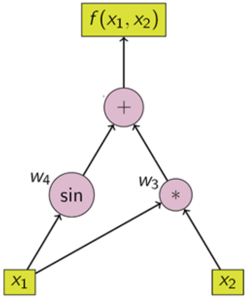
Toy Example

As a simple toy example, we consider the function

$$f(x_1, x_2) = x_1 x_2 + sin(x_1)$$

We might have implemented it as

$$w_1 = x_1$$
$$w_2 = x_2$$
$$w_3 = w_1 w_2$$
$$w_4 = sin(w_1)$$
$$w_5 = w_3 + w_4$$



Source: Wikipedia on Automatic Differentiation

Reverse Mode (Adjoint Mode)

Compute the adjoint $\bar{w}_i = \frac{\partial f}{\partial w_i}$ for all intermediate values.

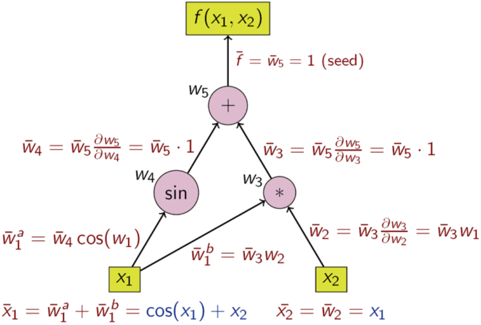
$$\bar{w}_5 = \bar{f} = 1(seed)$$

$$\bar{w}_4 = \frac{\partial f}{\partial w_4} = \frac{\partial f}{\partial w_5}\frac{\partial w_5}{\partial w_4} = 1 \cdot 1$$

$$\bar{w}_3 = \frac{\partial f}{\partial w_3} = \frac{\partial f}{\partial w_5}\frac{\partial w_5}{\partial w_3} = 1 \cdot 1$$

$$\bar{w}_2 = \frac{\partial f}{\partial w_3}\frac{\partial w_3}{\partial w_2} = x_1$$

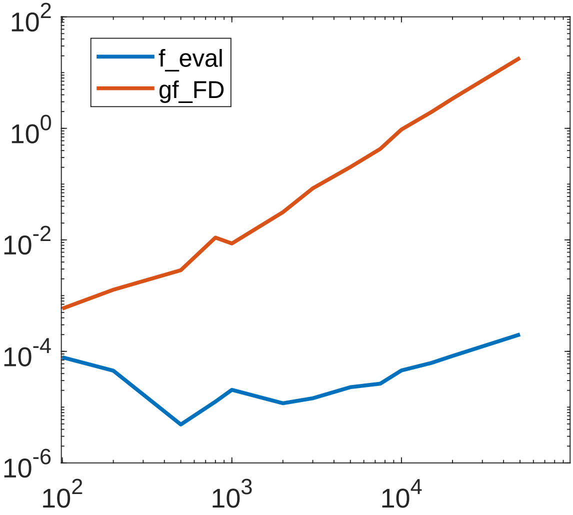
$$\bar{w}_2 = \frac{\partial f}{\partial w_4}\frac{\partial w_4}{\partial w_1} + \frac{\partial f}{\partial w_3}\frac{\partial w_3}{\partial w_1} = cos(x_1) + x_2$$



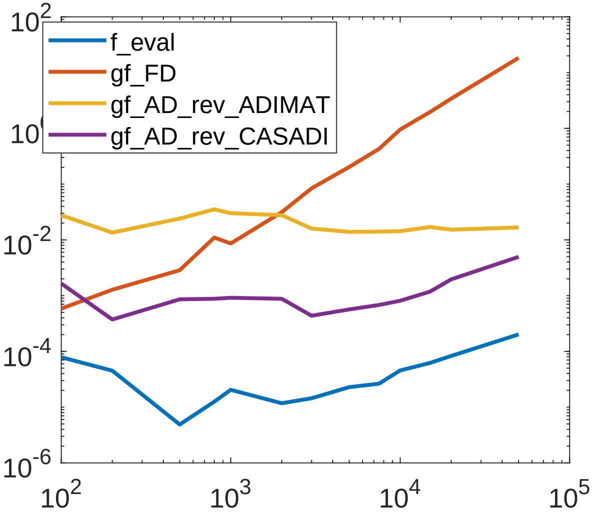
Source: Wikipedia on AD

Accurate up to machine precision; does not scale in n; high memory.

Scaling



Automatic Differentiation - Scaling

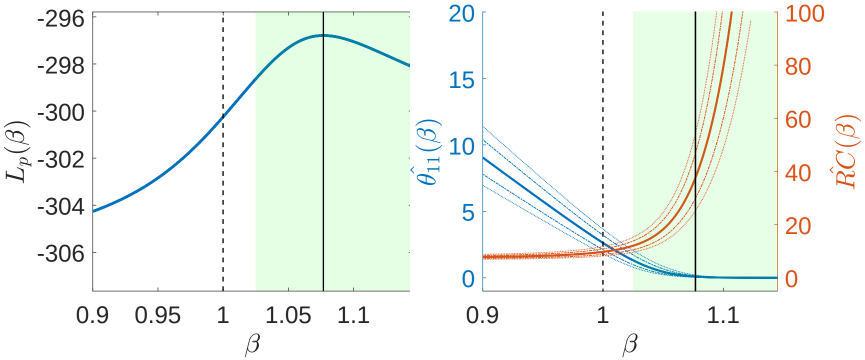


Software Tools

- Homotopy parameter continuation
  - HOMPACK90 by Watson et al. [1997]: a Fortran 90 collection of homotopy solution methods
  - M-Hompack by Müller and Reich [2018]: an interface between Matlab and HOMPACK90 to easily access and employ the efficient homotopy solution methods
- Automatic Differentiation
  - Especially for the homotopy continuation, fast and accurate derivatives are mandatory
  - AD provides analytic derivatives by source code transformation via successively utilizing the chain rule. We use CasADi by Andersson et al. [2018].

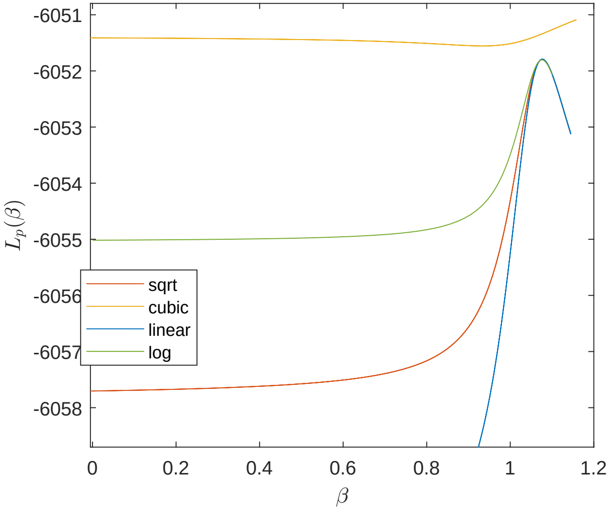
Rust [1987]'s Assumed Value for β and Likelihood

"not able to precisely estimate the discount factor β [...] Changing β to .98 or .9999 produced negligible changes in the likelihood function and parameter estimates [...]
I did note a systematic tendency for the estimated value of
to be driven to 1." Rust [1987]



	β	RC	θ_{11}	L
Rust(1987)	0.9999	9.7558	2.6275	-6055.250
	-	[8.200, 11.76]	[1.810, 3.669]	
MR	1.0768	37.7109	0.0905	-6051.792
	[1.025, ∞]	[13.00, 354.9]	[0.001, 1.029]	

Robustness


-6051
-6052
-6053
-6054
-6055
-6056
-6057
-6058
$L_p(\beta)$
0
0.2
0.4
0.6
0.8
1
1.2
β
sqrt
cubic
linear
log

Conclusion

- Given the original data set and model we can reject that β is unidentified.
- The estimate for β is unexpectedly even **statistically significantly larger than 1** with $\beta = 1.078$ ($p = 0.0086$).
- Capable of systematic and efficient structural estimation, even for models with a poorly identified parameter and in the presence of multiple equilibria.
- We enable further inference on the full (profile) likelihood function.

References I

J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl. CasADi – A software framework for nonlinear optimization and optimal control. *forthcoming in: Mathematical Programming Computation*, 2018.

D. P. Bertsekas. *Dynamic Programming and Optimal Control* , volume 2 of *Approximate dynamic programming*. Athena Scientific, Belmont, MA, 4th edition, 2012.

O. Blom Västberg and A. Karlström. Discount factors greater than or equal to one in infinite horizon dynamic discrete choice models. *unpublished; available on request from the authors*, 2017.

J. Rust. Optimal Replacement of GMC Bus Engines: An Empirical Model of Harold Zurcher. *Econometrica: Journal of the Econometric Society*, 55(5):999–1033, 1987.

References II

C.-L. Su and K. L. Judd. Constrained Optimization Approaches to Estimation of Structural Models. *Econometrica: Journal of the Econometric Society*, 80(5):2213–2230, 2012.

L. T. Watson, M. Sosonkina, R. C. Melville, A. P. Morgan, and H. F. Walker. Algorithm 777: HOMPACK90: a suite of Fortran 90 codes for globally convergent homotopy algorithms. *ACM Transactions on Mathematical Software (TOMS)*, 23(4): 514–549, Dec. 1997.

Software Tools

- Homotopy parameter continuation
  - HOMPACK90 by Watson et al. [1997]: a Fortran 90 collection of homotopy solution methods
  - M-Hompack by Müller and Reich [2018]: an interface between Matlab and HOMPACK90 to easily access and employ the efficient homotopy solution methods
- Automatic Differentiation
  - Especially for the homotopy continuation, fast and accurate derivatives are mandatory
  - AD provides analytic derivatives by source code transformation via successively utilizing the chain rule. We use CasADi by Andersson et al. [2018].

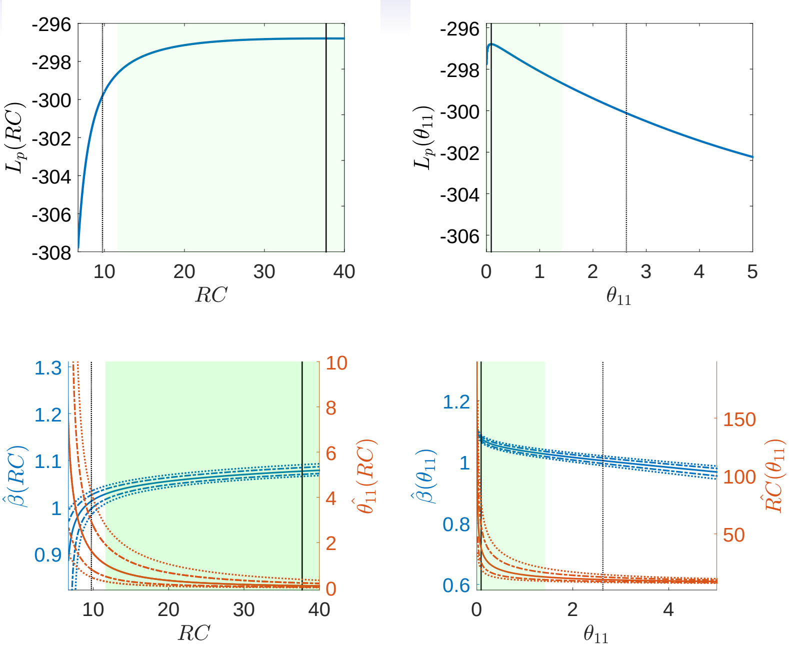
Confidence Intervals

The γ-*likelihood ratio confidence interval* of parameter θ_j as function of β reads

$$\left\{\theta_j : \max_{\theta_{-j}} L(\theta;\beta) - \Big(L(\hat{\theta}(\beta);\beta) - 0.5\chi_1^2(\gamma)\Big) \geq 0\right\}, \qquad (7)$$

$\hat{\theta}(\beta)$ denotes the maximum likelihood estimate in dependence of β. This naturally integrates into our tracing approach

$$\begin{pmatrix} L(\theta;\beta) - (L(\hat{\theta}(\beta);\beta) - 0.5\chi_1^2(\gamma)) \\ \nabla_{\mu,\theta_{-j},\sigma}\mathcal{L}(\mu,\theta,\sigma;\beta) \end{pmatrix} = 0. \qquad (8)$$


-296
-298
-300
-302
-304
-306
-308
10
20
30
40
RC
$L_p(RC)$
-296
-298
-300
-302
-304
-306
0
1
2
3
4
5
θ_{11}
$L_p(\theta_{11})$
1.3
1.2
1.1
1
0.9
10
8
6
4
2
0
$\hat{\beta}(RC)$
$\hat{\theta_{11}}(RC)$
10
20
30
40
RC
1.2
1
0.8
0.6
150
100
50
$\hat{\beta}(\theta_{11})$
$\hat{RC}(\theta_{11})$
0
2
4
θ_{11}