Lecture 3: Nested fixed point (NFXP) algorithm for estimation of equilibrium models

Bertel Schjerning, University of Copenhagen

2025 Econometric Society Summer School in Dynamic Structural Econometrics

University College London
June 30th - July 7th , 2025

PART I

The Nested Fixed Point Algorithm (NFXP)

Rust (ECTA, 1987):

OPTIMAL REPLACEMENT OF GMC BUS ENGINES:
AN EMPIRICAL MODEL OF HAROLD ZURCHER

Harold Alois Zuercher June 16, 1926 - June 21, 2020 (age 94)

Who cares about Harold Zurcher?

- Occupational Choice (Keane and Wolpin, JPE 1997)
- Retirement (Rust and Phelan, ECMA 1997)
- Brand choice and advertising (Erdem and Keane, MaScience 1996)
- Choice of college major (Arcidiacono, JoE 2004)
- Individual migration decisions (Kennan and Walker, ECMA 2011)
- High school attendance and work decisions (Eckstein and Wolpin, ECMA 1999)
- Sales and dynamics of consumer inventory behavior (Hendel and Nevo, ECMA 2006)
- Advertising, learning, and consumer choice in experience good markets (Ackerberg, IER 2003)
- Route choice models (Fosgerau et al, Transp. Res. B)
- Fertility and labor supply decisions (Francesconi, JoLE 2002)
- Residential and Work-location choice (Buchinsky et al, ECMA 2015)
- Equilibrium Allocations Under Alternative Waitlist Designs: Evidence From Deceased Donor Kidneys (Argarwal et al, ECMA 2021)
- Equilibrium Trade in Automobiles (Gillingham et al, JPE 2022)
- ...and many more

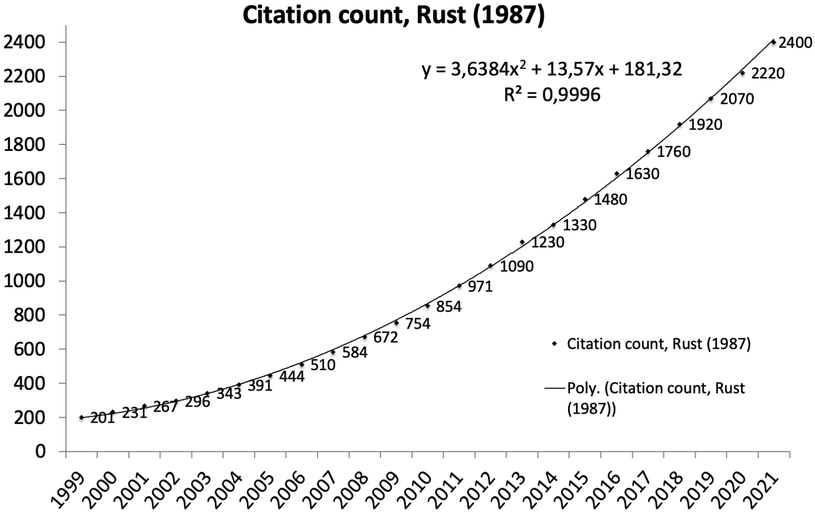
Who cares about Harold Zurcher?

- Occupational Choice (Keane and Wolpin, JPE 1997)
- Retirement (Rust and Phelan, ECMA 1997)
- Brand choice and advertising (Erdem and Keane, MaScience 1996)
- Choice of college major (Arcidiacono, JoE 2004)
- Individual migration decisions (Kennan and Walker, ECMA 2011)
- High school attendance and work decisions (Eckstein and Wolpin, ECMA 1999)
- Sales and dynamics of consumer inventory behavior (Hendel and Nevo, ECMA 2006)
- Advertising, learning, and consumer choice in experience good markets (Ackerberg, IER 2003)
- Route choice models (Fosgerau et al, Transp. Res. B)
- Fertility and labor supply decisions (Francesconi, JoLE 2002)
- Residential and Work-location choice (Buchinsky et al, ECMA 2015)
- Equilibrium Allocations Under Alternative Waitlist Designs: Evidence From Deceased Donor Kidneys (Argarwal et al, ECMA 2021)
- Equilibrium Trade in Automobiles (Gillingham et al, JPE 2022)
- ...and many more

Who cares about Harold Zurcher?

- Occupational Choice (Keane and Wolpin, JPE 1997)
- Retirement (Rust and Phelan, ECMA 1997)
- Brand choice and advertising (Erdem and Keane, MaScience 1996)
- Choice of college major (Arcidiacono, JoE 2004)
- Individual migration decisions (Kennan and Walker, ECMA 2011)
- High school attendance and work decisions (Eckstein and Wolpin, ECMA 1999)
- Sales and dynamics of consumer inventory behavior (Hendel and Nevo, ECMA 2006)
- Advertising, learning, and consumer choice in experience good markets (Ackerberg, IER 2003)
- Route choice models (Fosgerau et al, Transp. Res. B)
- Fertility and labor supply decisions (Francesconi, JoLE 2002)
- Residential and Work-location choice (Buchinsky et al, ECMA 2015)
- Equilibrium Allocations Under Alternative Waitlist Designs: Evidence From Deceased Donor Kidneys (Argarwal et al, ECMA 2021)
- Equilibrium Trade in Automobiles (Gillingham et al, JPE 2022)
- ...and many more

Big Mac Index of Dynamic Structural Econometrics



Zurcher's Bus Engine Replacement Problem

- Choice set: Binary choice set, $C(x_t) = \{0, 1\}$.
  - Engine replacement ($d_t = 1$) or ordinary maintenance ($d_t = 0$)
- State variables: Harold Zurcher observes $s_t = (x_t, \varepsilon_t)$:
  - x_t: mileage at time t since last engine overhaul/replacement
  - $\varepsilon_t = [\varepsilon_t(d_t = 0), \varepsilon_t(d_t = 1)]$: decision specific state variable
- Utility function: $U(x_t, \varepsilon_t, d_t; \theta_1) =$

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

- State variables process
  - ε_t is iid with conditional density $q(\varepsilon_t | x_t, \theta_2)$
  - x_t (mileage since last replacement)

$$p(x_{t+1} | x_t, d_t, \theta_2) = \begin{cases} g(x_{t+1} - 0, \theta_3) & \text{if } d_t = 1 \\ g(x_{t+1} - x_t, \theta_3) & \text{if } d_t = 0 \end{cases} \tag{2}$$

  If engine is replaced, state of bus regenerates to $x_t = 0$.
- Parameters to be estimated $\theta = (RC, \theta_1, \theta_3)$
  (Fixed parameters: (β, θ_2))

Recursive form of the maximization problem

- The Bellman equation

$$V(x,\varepsilon) \equiv T(V)(x,\varepsilon) = \max_{d \in C(x)} \left\{ u(x,\varepsilon,d) + \beta E\left[V(x',\varepsilon') \middle| x,\varepsilon,d \right] \right\}$$

- Expectations are taken over the next period values of state $s' = (x', \varepsilon')$ given it's <u>controlled</u> motion rule, $p(s' \mid s, d)$

$$E\left[V(x',\varepsilon') \middle| x,\varepsilon,d \right] = \int_X \int_\Omega V(x',\varepsilon') p(x',\varepsilon' | x,\varepsilon,d) dx' d\varepsilon'$$

where $\varepsilon = (\varepsilon(1), \ldots, \varepsilon(J)) \in \mathbb{R}^J$

Hard to compute fixed point V such that $T(V) = V$

- x is continuous and ε is continuous and J-dimensional
- $V(x,\varepsilon)$ is high dimensional
- Evaluating E may require high dimensional integration
- Evaluating $V(x',\varepsilon')$ may require high dimensional interpolation/approximation
- $V(x,\varepsilon)$ is non-differentiable

Rust's Assumptions

1. Additive separability in preferences (**AS**):

$$U(s_t, d) = u(x_t, d; \theta_1) + \varepsilon_t(d)$$

2. Conditional independence (**CI**):
   State variables, $s_t = (x_t, \varepsilon_t)$ obeys a (conditional independent) controlled Markov process with probability density

$$p(x_{t+1}, \varepsilon_{t+1} | x_t, \varepsilon_t, d, \theta_2, \theta_3) = q(\varepsilon_{t+1} | x_{t+1}, \theta_2) p(x_{t+1} | x_t, d, \theta_3)$$

3. Extreme value Type I (EV1) distribution of ε (**EV**)
   Each of the choice specific state variables, $\varepsilon_t(d)$ are assumed to be iid. extreme value distributed with CDF

$$F(\varepsilon_t(d); \mu, \lambda) = \exp(-\exp(-(\varepsilon_t(d) - \mu)/\lambda)) \text{ for } \varepsilon_t(d) \in \mathbb{R}$$

   with $\mu = 0$ and $\lambda = 1$

Rust's Assumptions simplifies DP problem

$$V(x,\varepsilon) = \max_{d \in C(x)} \left\{ u(x,d) + \varepsilon(d) + \beta \int_X \int_\Omega V(x',\varepsilon') p(x'|x,d) q(\varepsilon'|x') dx' d\varepsilon' \right\}$$

1. Separate out the deterministic part of choice specific value $v(x, d)$ (assumptions AS and CI)
2. Reformulate Bellman equation on reduced state space (assumption CI)
3. Compute the expectation of maximum using properties of EV1 (assumption EV)

1. DP problem under AS and CI

Separate out the deterministic part of choice specific value $v(x, d)$

$$V(x, \varepsilon) = \max_{d \in C(x)} \left\{ u(x, d) + \beta \int_X \left(\int_\Omega V(x', \varepsilon') q(\varepsilon' | x') d\varepsilon' \right) p(x' | x, d) dx' + \varepsilon(d) \right\}$$

So that

$$V(x', \varepsilon') = \max_{d \in C} \left\{ v(x', d) + \varepsilon'(d) \right\}$$

where

$$v(x, d) = u(x, d) + \beta E\left[V(x', \varepsilon') \middle| x, d \right]$$

2a. Bellman equation in expected value function space

Let $EV(x,d) = E\left[V(x',\varepsilon')\middle|x,d\right]$ denote the expected value function.

Because of CI we can express the Bellman equation in expected value function space

$$EV(x,d) = \Gamma(EV)(x,d) \equiv \int_X \int_\Omega \left[V(x',\varepsilon')q(\varepsilon'|x')d\varepsilon'\right] p(x'|x,d)dx'$$

where

$$V(x',\varepsilon') = \max_{d'\in C(x')} [u(x',d') + \beta EV(x',d') + \varepsilon'(d')]$$

- Γ is a contraction mapping with unique fixed point EV, i.e. $\|\Gamma(EV) - \Gamma(W)\| \leq \beta \|EV - W\|$
- Global convergence of VFI / SA
- $EV(x,d)$ is lower dimensional: does not depend on ε

2b. Bellman equation in integrated value function space

Let $\bar{V}(x) = E\left[V(x,\varepsilon)\middle|x\right]$ denote the integrated value function

Because of CI we can express Bellman equation in integrated value function space

$$\bar{V}(x) = \bar{\Gamma}(\bar{V})(x) \equiv \int_{\Omega} V(x,\varepsilon)q(\varepsilon|x)d\varepsilon$$

where

$$V(x,\varepsilon) = \max_{d\in C(x)}[u(x,d) + \varepsilon(d) + \beta \int_{X} \bar{V}(x')p(x'|x,d)dx']$$

- $\bar{\Gamma}$ is a contraction mapping with unique fixed point $\bar{V}$, i.e. $\left\|\bar{\Gamma}\left(\bar{V}\right) - \bar{\Gamma}\left(W\right)\right\| \leq \beta \left\|\bar{V} - W\right\|$
- Global convergence of VFI / SA
- $\bar{V}(x)$ is lower dimensional: does not depend on ε and d

3. Compute the expectation of maximum under EV

We can express expectation of maximum using properties of EV1 distribution (assumption EV)

Expectation of maximum, $\bar{V}(x)$, can be expressed as "the log-sum"

$$\bar{V}(x) = E\left[\max_{d\in\{1,\dots,J\}}\{v(x,d)+\lambda\varepsilon(d)\} \mid x\right] = \lambda\log\sum_{j=1}^{J}\exp(v(x,d)/\lambda)$$

Conditional choice probability, $P(x,d)$ has closed form logit expression

$$\begin{aligned} P(d \mid x) &= E\left[\mathbb{1}\left\{d = \arg\max_{j\in\{1,\dots,J\}}\{v(x,j)+\lambda\varepsilon(j)\}\right\} \mid x\right] \\ &= \frac{\exp(v(x,d)/\lambda)}{\sum_{j=1}^{J}\exp(v(x,j)/\lambda)} \end{aligned}$$

HUGE benefits

- Avoids J dimensional numerical integration over ε
- $P(d \mid x)$, $\bar{V}(x)$ and $EV(x,d)$ are smooth functions.

The DP problem under AS, CI and EV

Putting all this together

- Conditional Choice Probabilities (CCPs) are given by

$$P(d|x,\theta) = \frac{\exp\{u(x,d,\theta_1) + \beta EV_\theta(x,d)\}}{\sum_{j\in C(x)} \exp\{u(x,j,\theta_1) + \beta EV_\theta(x,j)\}}$$

- The expected value function can be found as the unique fixed point to the contraction mapping Γ_θ, defined by

$$\begin{aligned} EV_\theta(x,d) &= \Gamma_\theta(EV_\theta)(x,d) \\ &= \int_y \ln\left[\sum_{d'\in C(y)} \exp\left[u(y,d';\theta_1) + \beta EV_\theta(y,d')\right]\right] \\ &\quad p(dy|x,d,\theta_2) \end{aligned}$$

- We have used the subscript θ to emphasize that the Bellman operator,Γ_θ depends on the parameters.
- In turn, the fixed point, EV_θ, and the resulting CCPs, $P(d|x,\theta)$ are implicit functions of the parameters we wish to estimate.

How to deal with continuous mileage state?

Rust discretize the mileage state space x into n grid points

$$X = \{x_1, ..., x_n\} \text{ with } x_1 = 0$$

Mileage transition probability: for $l = 0, \ldots, L$

$$p(x'|\hat{x}_k, d, \theta_2) = \begin{cases} Pr\{x' = x_{k+l}|\theta_3\} = \theta_{3,l} \text{ if } d = 0 \\ Pr\{x' = x_{1+l}|\theta_3\} = \theta_{3,l} \text{ if } d = 1 \end{cases}$$

- where $\theta_3 = [\theta_{3,1}, \ldots, \theta_{3,L}]$, $\theta_{3,0} = 1 - \sum_{l=1}^{L} \theta_{3,l}$, and $\theta_{3,l} \geq 0$
- Mileage in the next period x' can move up at most L grid points.
- L is determined by the empirical distribution of mileage.

State transition matrix for mileage with L=2

$$\Pi(d = \text{keep})_{nxn} = \begin{pmatrix} \theta_{3,0} & \theta_{3,1} & \theta_{3,2} & 0 & \cdot & \cdot & \cdot & 0 \\ 0 & \theta_{3,0} & \theta_{3,1} & \theta_{3,2} & 0 & \cdot & \cdot & 0 \\ 0 & 0 & \theta_{3,0} & \theta_{3,1} & \theta_{3,2} & 0 & \cdot & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \\ 0 & & & & \theta_{3,0} & \theta_{3,1} & \theta_{3,2} & 0 \\ 0 & & & & & \theta_{3,0} & \theta_{3,1} & \theta_{3,2} \\ 0 & & & & & & \theta_{3,0} & 1-\theta_{3,0} \\ 0 & 0 & & & & & & 1 \end{pmatrix}$$

$$\Pi(d = \text{replace})_{nxn} = \begin{pmatrix} \theta_{3,0} & \theta_{3,1} & \theta_{3,2} & 0 & \cdot & \cdot & \cdot & 0 \\ \theta_{3,0} & \theta_{3,1} & \theta_{3,2} & 0 & \cdot & \cdot & \cdot & 0 \\ \theta_{3,0} & \theta_{3,1} & \theta_{3,2} & 0 & \cdot & \cdot & \cdot & 0 \\ \theta_{3,0} & \theta_{3,1} & \theta_{3,2} & 0 & \cdot & \cdot & \cdot & 0 \\ \theta_{3,0} & \theta_{3,1} & \theta_{3,2} & 0 & \cdot & \cdot & \cdot & 0 \\ \theta_{3,0} & \theta_{3,1} & \theta_{3,2} & 0 & \cdot & \cdot & \cdot & 0 \\ \theta_{3,0} & \theta_{3,1} & \theta_{3,2} & 0 & \cdot & \cdot & \cdot & 0 \end{pmatrix}$$

Bellman equation in matrix form

Bellman equation in expected value function space

$$EV(d) = \Gamma(EV) = \Pi(d) \ln \left[\sum_{d'} \exp[u(d') + \beta EV(d')] \right]$$

Bellman equation in integrated value function space

$$\bar{V} = \bar{\Gamma}(\bar{V}) = \ln \left[\sum_{d'} \exp[u(d') + \beta \Pi(d') \bar{V}] \right]$$

where

- $u(d) = [u(x_1, d), .., u(x_n, d)]$
- $EV(d) = [EV(x_1, d), .., EV(x_n, d)]$
- $\bar{V} = [\bar{V}(x_1), .., \bar{V}(x_n)]$
- $\Pi(d)$ is a $n \times n$ state transition matrix conditional on decision d

Structural Estimation under Rust's Assumptions

Data: $(d_{i,t}, x_{i,t})$, $t = 1, ..., T_i$ and $i = 1, ..., N$

Log likelihood function

$$L(\theta, EV_\theta)) = \sum_{i=1}^{N} \ell_i^f(\theta, EV_\theta)$$

$$\ell_i^f(\theta, EV_\theta) = \sum_{t=2}^{T_i} log(P(d_{i,t}|x_{i,t}, \theta)) + \sum_{t=2}^{T_i} log\left(p(x_{i,t}|x_{i,t-1}, d_{i,t-1}, \theta_3)\right)$$

where

$$P(d|x, \theta) = \frac{\exp\{u(x, d, \theta_1) + \beta EV_\theta(x, d)\}}{\sum_{d' \in \{0,1\}}\{u(x, d', \theta_1) + \beta EV_\theta(x, d')\}}$$

and

$$\begin{aligned} EV_\theta(x, d) &= \Gamma_\theta(EV_\theta)(x, d) \\ &= \int_y \ln\left[\sum_{d' \in \{0,1\}} \exp[u(y, d'; \theta_1) + \beta EV_\theta(y, d')]\right] p(dy|x, d, \theta_3) \end{aligned}$$

The Nested Fixed Point Algorithm

Since the contraction mapping Γ_θ always has a unique fixed point, the constraint $EV_\theta = \Gamma(EV_\theta)$ implies that the fixed point EV_θ is an implicit function of θ.

Hence, NFXP solves the unconstrained optimization problem

$$\max_\theta L(\theta, EV_\theta)$$

Outer loop (Hill-climbing algorithm):

- Likelihood function $L(\theta, EV_\theta)$ is maximized w.r.t. θ
- Quasi-Newton algorithm: Usually BHHH, BFGS or a combination.
- Each evaluation of $L(\theta, EV_\theta)$ requires solution of EV_θ

Inner loop (fixed point algorithm):
The implicit function EV_θ defined by $EV_\theta = \Gamma(EV_\theta)$ is solved by:

- Successive Approximations (SA) / Value Function Iterations (VFI)
- Newton-Kantorovich (NK) Iterations

Structural Estimates, n=175

TABLE X

STRUCTURAL ESTIMATES FOR COST FUNCTION $c(x, \theta_1) = .001\theta_{11}x$

FIXED POINT DIMENSION = 175

(Standard errors in parentheses)

Parameter		Data Sample			Heterogeneity Test	
Discount Factor	Estimates Log-Likelihood	Groups 1, 2, 3 3864 Observations	Group 4 4292 Observations	Groups 1, 2, 3, 4 8156 Observations	LR Statistic (df = 6)	Marginal Significance Level
$\beta = .9999$	RC	11.7257 (2.597)	10.896 (1.581)	9.7687 (1.226)	237.53	1.89E−48
	θ_{11}	2.4569 (.9122)	1.1732 (0.327)	1.3428 (0.315)		
	θ_{30}	.0937 (.0047)	.1191 (.0050)	.1071 (.0034)		
	θ_{31}	.4475 (.0080)	.5762 (.0075)	.5152 (.0055)		
	θ_{32}	.4459 (.0080)	.2868 (.0069)	.3621 (.0053)		
	θ_{33}	.0127 (.0018)	.0158 (.0019)	.0143 (.0013)		
	LL	−3993.991	−4495.135	−8607.889		
$\beta = 0$	RC	8.2969 (1.0477)	7.6423 (.7204)	7.3113 (0.5073)	241.78	2.34E−49
	θ_{11}	56.1656 (13.4205)	36.6692 (7.0675)	36.0175 (5.5145)		
	θ_{30}	.0937 (.0047)	.1191 (.0050)	.1070 (.0034)		
	θ_{31}	.4475 (.0080)	.5762 (.0075)	.5152 (.0055)		
	θ_{32}	.4459 (.0080)	.2868 (.0069)	.3622 (.0053)		
	θ_{33}	.0127 (.0018)	.0158 (.0019)	.0143 (.0143)		
	LL	−3996.353	−4496.997	−8614.238		
Myopia tests:	LR Statistic ($df = 1$)	4.724	3.724	12.698		
$\beta = 0$ vs. $\beta = .9999$	Marginal Significance Level	0.0297	0.0536	.00037		

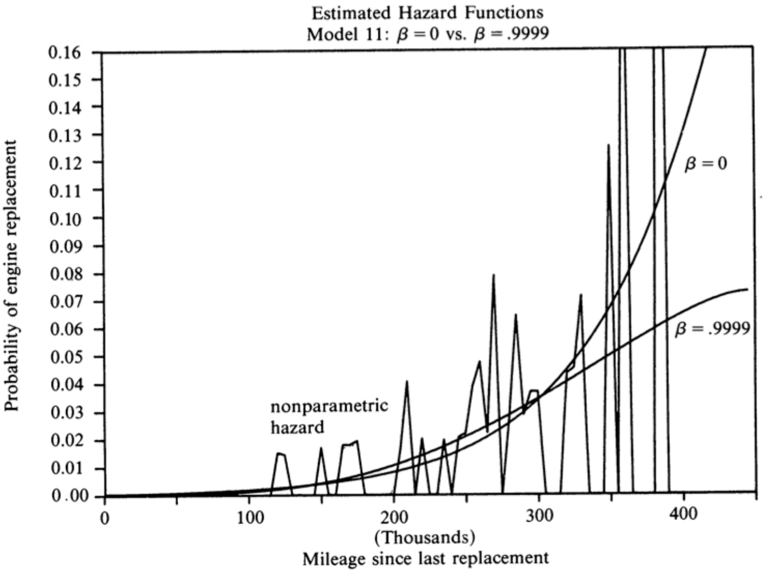
MATLAB implementation, n=175, (replication of Table X)

Output from `run_busdata.m`:

fzp386 — cefhelper (Renderer) ◂ MATLAB_maci64 -nodesktop — 89×35

```
>> run_busdata
Structural Estimation using busdata from Rust(1987)
Bustypes       = [ 1  2  3  4 ]
Beta           =    0.99990
n              =  175.00000
Sample size    = 8156.00000


Method nfxp (pmle)
    Param.                        Estimates          s.e.         t-stat
-----------------------------------------------------------------------------------------
    RC                               9.7910        1.2684         7.7190
    c                                1.3486        0.3458         3.8996
-----------------------------------------------------------------------------------------
log-likelihood     = -300.57017
runtime (seconds) =     0.07795
g'*inv(h)*g       = 2.65552e-09


Method nfxp (mle)
    Param.                        Estimates          s.e.         t-stat
-----------------------------------------------------------------------------------------
    RC                               9.7915        1.2689         7.7168
    c                                1.3488        0.3460         3.8982
    p                  (1)           0.1070        0.0034        31.2111
    p                  (2)           0.5152        0.0055        93.0533
    p                  (3)           0.3622        0.0053        68.0413
    p                  (4)           0.0143        0.0013        10.8947
    p                  (5)           0.0009        0.0003         2.6469
-----------------------------------------------------------------------------------------
log-likelihood     = -8607.88844
runtime (seconds) =     0.07484
g'*inv(h)*g       = 7.26854e-09
>>
```

Estimated Hazard Functions



Equilibrium bus mileage and demand for enigines

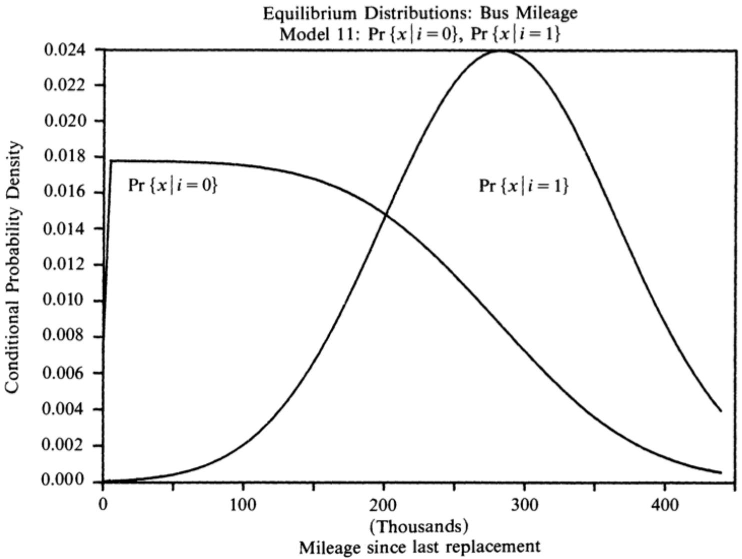
- Let π be the long run stationary (or equilibrium) distribution of the controlled process $\{i_t, x_t\}$
- π is then given by the unique solution to the functional equation

$$\pi(x, i) = \int_y \int_j P(i|x, \theta)p(x|y, j, \theta_3)\pi(dy, dj)$$

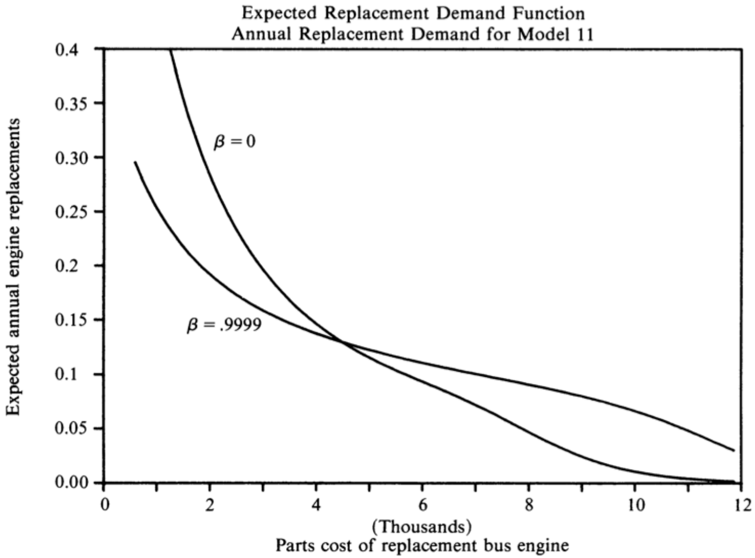
- π is the ergodic distribution of the controlled state transition matrix
- Carly the equilibrium distribution of π is an implicit function of the structural parameters θ, which we emphasize by the notation π_θ
- Given π_θ, we can also obtain the following simple formula for annual equilibrium demand for engines as a function of RC

$$d(RC) = 12M \int_0^\infty \pi_\theta(dx, 1)$$

Equilibrium Bus mileage, bus group 4



Demand Function, bus group 4



Mathematical Programming with Equilibrium Constraints

MPEC solves the constrained optimization problem

$$\max_{\theta, EV} L(\theta, EV) \text{ subject to } EV = \Gamma_\theta(EV)$$

using general-purpose constrained optimization solvers such as KNITRO

Su and Judd (ECMA, 2012) considers two such implementations:

MPEC/AMPL:

- AMPL formulates problems and pass it to KNITRO.
- Automatic differentiation (Jacobian and Hessian)
- Sparsity patterns for Jacobian and Hessian

MPEC/MATLAB:

- User need to supply Jacobians, Hessian, and Sparsity Patterns
- Su and Judd do not supply analytical derivatives.
- ktrlink provides link between MATLAB and KNITRO solvers.

Su, Che-Lin , Kenneth L. Judd (2015): "Constrained Optimization Approaches to Estimation of Structural Models." Econometrica 80-5, pp. 2213-2230.

Monte Carlo: Rust's Table X - Group 1,2, 3

Su and Judd (ECMA, 2012) did a head to head comparison between NFXP and MPEC

Experimental Design

- Fixed point dimension: $n = 175$
- Maintenance cost function: $c(x, \theta_1) = 0.001 * \theta_1 * x$
- Mileage transition: stay or move up at most $L = 3$ grid points
- True parameter values:
  - $\theta_1 = 2.457$
  - $RC = 11.726$
  - $\theta_3 = (\theta_{3,0}, \theta_{3,1}, \theta_{3,2}, \theta_{3.3}) = (0.0937, 0.4475, 0.4459, 0.0127)$
- Solve for EV at the true parameter values
- Simulate 250 datasets of monthly data for 10 years and 50 buses

Is NFXP a dinosaur method?

TABLE II

NUMERICAL PERFORMANCE OF NFXP AND MPEC IN THE MONTE CARLO EXPERIMENTS[a]

β	Implementation	Runs Converged (out of 1250 runs)	CPU Time (in sec.)	# of Major Iter.	# of Func. Eval.	# of Contraction Mapping Iter.
0.975	MPEC/AMPL	1240	0.13	12.8	17.6	–
	MPEC/MATLAB	1247	7.90	53.0	62.0	–
	NFXP	998	24.60	55.9	189.4	134,748
0.980	MPEC/AMPL	1236	0.15	14.5	21.8	–
	MPEC/MATLAB	1241	8.10	57.4	70.6	–
	NFXP	1000	27.90	55.0	183.8	162,505
0.985	MPEC/AMPL	1235	0.13	13.2	19.7	–
	MPEC/MATLAB	1250	7.50	55.0	62.3	–
	NFXP	952	43.20	61.7	227.3	265,827
0.990	MPEC/AMPL	1161	0.19	18.3	42.2	–
	MPEC/MATLAB	1248	7.50	56.5	65.8	–
	NFXP	935	70.10	66.9	253.8	452,347
0.995	MPEC/AMPL	965	0.14	13.4	21.3	–
	MPEC/MATLAB	1246	7.90	59.6	70.7	–
	NFXP	950	111.60	58.8	214.7	748,487

[a]For each β, we use five starting points for each of the 250 replications. CPU time, number of major iterations, number of function evaluations and number of contraction mapping iterations are the averages for each run.

CPR for NFXP

Step 1: Read NFXP manual and print out NFXP pocket guide
Step 2: Recenter logit and logsum formulas
Step 3: Use Fixed Point Poly-Algorithm (SA+NK)
Step 4: Provide analytical gradients of Bellman operator
Step 5: Provide analytical gradients of likelihood
Step 6: Use BHHH (outer product of gradients as hessian approx.)

If NFXP heartbeat is still weak:
Read NFXP pocket guide until help arrives!

CPR for NFXP

Step 1: Read NFXP manual and print out NFXP pocket guide
Step 2: Recenter logit and logsum formulas
Step 3: Use Fixed Point Poly-Algorithm (SA+NK)
Step 4: Provide analytical gradients of Bellman operator
Step 5: Provide analytical gradients of likelihood
Step 6: Use BHHH (outer product of gradients as hessian approx.)

If NFXP heartbeat is still weak:
Read NFXP pocket guide until help arrives!

STEP 1: NFXP documentation

References

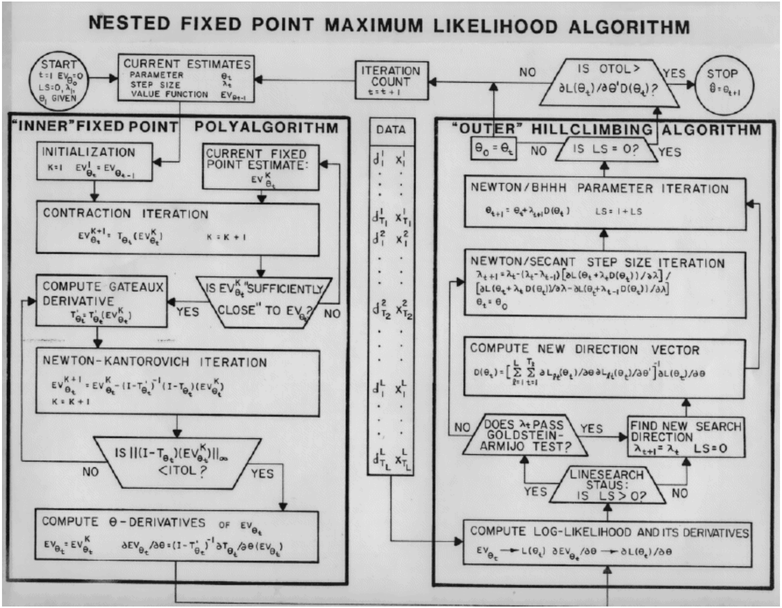
- Rust (1987): "Optimal Replacement of GMC Bus Engines: An Empirical Model of Harold Zurcher" Econometrica 55-5, pp 999-1033.
- Rust (2000): “Nested Fixed Point Algorithm Documentation Manual: Version 6” `https://editorialexpress.com/jrust/nfxp.html`
- Iskhakov, F. , J. Rust, B. Schjerning, L. Jinhyuk, and K. Seo (2015): "Constrained Optimization Approaches to Estimation of Structural Models : Comment." Econometrica 84-1, pp. 365-370.

Nested Fixed Point Algorithm

NFXP Documentation Manual version 6, (Rust 2000, page 18):

> *Formally, one can view the nested fixed point algorithm as solving the following constrained optimization problem:*
>
> $$\max_{\theta, EV} L(\theta, EV) \text{ subject to } EV = \Gamma_\theta(EV) \tag{3}$$
>
> *Since the contraction mapping Γ always has a unique fixed point, the constraint $EV = \Gamma_\theta(EV)$ implies that the fixed point EV_θ is an implicit function of θ. Thus, the constrained optimization problem (3) reduces to the unconstrained optimization problem*
>
> $$\max_{\theta} L(\theta, EV_\theta) \tag{4}$$
>
> *where EV_θ is the implicit function defined by $EV_\theta = \Gamma(EV_\theta)$.*

NFXP pocket guide



STEP 2: Recenter to ensure numerical stability

Logit formulas must be reentered.

$$P_i = \frac{\exp(v_i)}{\sum_j \exp(v_j)} = \frac{\exp(v_i - v_0)}{\sum_j \exp(v_j - v_0)}$$

and "log-sum" must be recentered too

$$\ln \sum_j \exp(v_j) = v_0 + \ln \sum_j \exp(v_j - v_0)$$

If v_0 is chosen to be $v_0 = \max_j v_j$ we can avoid numerical instability due to overflow/underflow

STEP 3: Use Fixed Point Poly-Algorithm (SA+NK)

Problem: Find fixed point of the contraction mapping, Γ_θ

$$EV_\theta = \Gamma(EV_\theta)$$

Fixed Point Poly-Algorithm:

1. Successive Approximations (SA) by contraction iteration:

$$EV_{k+1} = \Gamma_\theta(EV_k)$$

   - Error bound: $||EV_{k+1} - EV|| \leq \beta ||EV_k - EV||$
     $\rightarrow$ Linear convergence $\rightarrow$ slow when β close to 1

2. Newton-Kantorovich (NK) iteration:
   - Solve $F = [I - \Gamma](EV_\theta) = 0$ using Newtons method

$$EV_{k+1} = EV_k - (I - \Gamma')^{-1}(I - \Gamma)(EV_k)$$

     Γ'_θ is the Fréchet derivative of Γ_θ
     I is the identity operator on B
     0 is the zero element of B
   - Error bound: $||EV_{k+1} - EV|| \leq A||EV_k - EV||^2$
     $\rightarrow$ Quadratic convergence around fixed point, EV

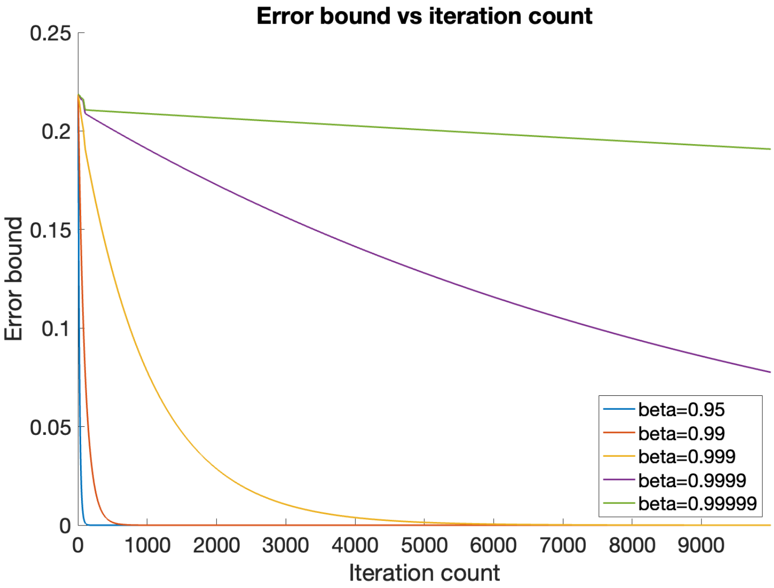
Successive Approximations, $\beta = 0.9999$

```
>> run_fxp
iter            tol          tol(j)/tol(j-1)
   1         0.21854635        1.00000000
   2         0.21852208        0.99988895
   3         0.21849729        0.99988654
   :              :                :
 49998       0.00142119        0.99990000
 49999       0.00142105        0.99990000
 50000       0.00142090        0.99990000
Maximum number of iterations exceeded without convergence!
Elapsed time: 1.44489 (seconds)
```

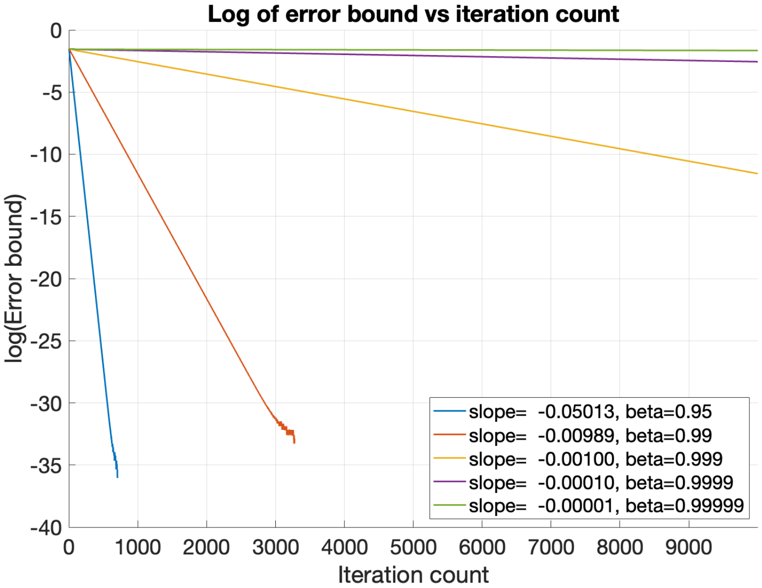
Observations:

- $tol_k = \|EV_{k+1} - EV_k\| < \beta\|EV_k - EV\|$
- Tolerance always improves due to contraction property
- tol_k quickly slow down and declines very slowly for β close to 1
- Relative tolerance tol_{k+1}/tol_k approach β

Successive Approximations - VERY slow when β close to 1



Successive Approximations - linear convergence



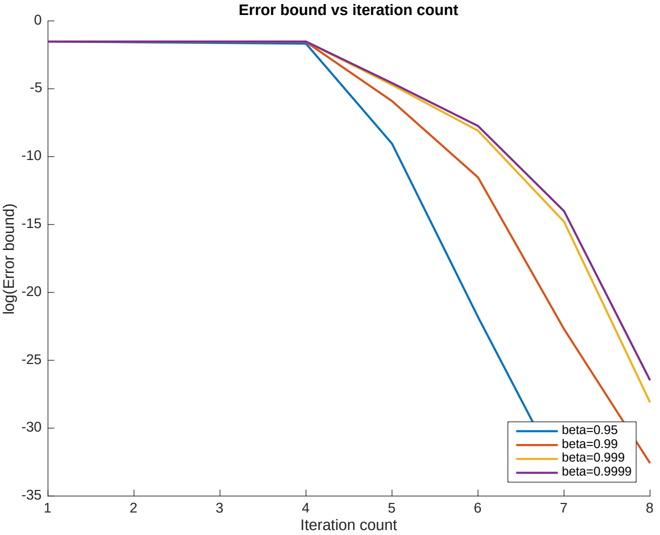
Newton-Kantorovich Iterations, $\beta = 0.9999$

```
>> run_fxp
Begin contraction iterations (for the 1. time)
iter             tol          tol(j)/tol(j-1)
   1         0.21854635       1.00000000
   2         0.21852208       0.99988895
SA stopped prematurely due to rel. tolerance. Begin NK iterations
Elapsed time: 0.00147 (seconds)

Begin Newton-Kantorovich iterations (for the 1. time)
iter             tol          tol(j)/tol(j-1)
   1         0.01037444              NaN
   2         0.00041127              NaN
   3         0.00000069              NaN
   4         0.00000000              NaN
N-K converged after 4 iterations, tolerance: 2.27374e-12
Elapsed time: 0.00331 (seconds)

Convergence achieved!
Total elapsed time: 0.00300 (seconds)
```

Observations:

- Quadratic convergence!
- Very fast, once in domain of attraction

Newton-Kantorovich Iterations - quadratic convergence!



When to switch to Newton-Kantorovich?

When to switch to Newton-Kantorovich?

- Suppose that $EV_0 = EV + k$.
  (Initial EV_0 equals fixed point EV plus an arbitrary constant)
- Another successive approximation does not solve this:

$$
\begin{aligned}
tol_0 &= \|EV_0 - \Gamma(EV_0)\| = \|EV + k - \Gamma(EV + k)\| \\
&= \|EV + k - (EV + \beta k)\| = (1-\beta)k \\
tol_1 &= \|EV_1 - \Gamma(EV_1)\| = \|EV + \beta k - \Gamma(EV + \beta k)\| \\
&= \|EV + \beta k - (EV + \beta^2 k)\| = \beta(1-\beta)k \\
tol_1/tol_0 &= \beta
\end{aligned}
$$

- Newton will immediately "strip away" the irrelevant constant k
- Switch to Newton whenever tol_1/tol_0 is sufficiently close to β

The Fixed Point (poly) Algorithm

Fixed Point poly Algorithm

1. Successive contraction iterations

$$EV_{k+1} = \Gamma_\theta(EV_k)$$

   until EV_k is in the domain of attraction
   (i.e. when tol_{k+1}/tol_k is close to β)

2. Newton-Kantorovich (quadratic convergence)

$$EV_{k+1} = EV_k - (I - \Gamma')^{-1}(I - \Gamma)(EV_k)$$

   until convergence
   (i.e. when $\|EV_{k+1} - EV_k\|$ is close to machine precision)

STEP 4: Analytical derivative of Bellman operator

Bellman equation in integrated value function space

$$\bar{\Gamma} = \bar{\Gamma}(\bar{V}) = \ln \left[\sum_{d'} \exp[u(d') + \beta \Pi(d') \bar{V}] \right]$$

Derivative of Bellman operator, $\bar{\Gamma}'$

- Needed for the NK iteration

$$\bar{V}_{k+1} = \bar{V}_k - (I - \bar{\Gamma}')^{-1}(I - \bar{\Gamma})(\bar{V}_k)$$

- In discretized approximation, $\bar{\Gamma}'$ is a $n \times n$ matrix with partial derivatives of the $n \times 1$ vector function $\bar{\Gamma}(V_\theta)$ with respect to the $n \times 1$ vector $\bar{V}_\theta$
- $\bar{\Gamma}'_\theta$ is simply β times the choice probability weighted state transition probability matrix

$$\bar{\Gamma}'_\theta = \beta \sum_j \Pi(j) .* P(j)$$

- One line of code in MATLAB
- A similar matrix can be derived for Γ'

STEP 1-4: MATLAB implementation of $\bar{\Gamma}_\theta$ and $\bar{\Gamma}'_\theta$

```
function [V1, pk, dBellman_dV]=bellman_iv(V0, mp, u, P)
  vK= u(:,1) + mp.beta*P{1}*V0;     % Value of keeping
  vR= u(:,2) + mp.beta*P{2}*V0;     % Value of replacing

  % Recenter logsum
  maxV=max(vK, vR);
  V1=(maxV + log(exp(vK-maxV)  +  exp(vR-maxV)));

  % If requested, compute keep probability
  if nargout>1
    pk=1./(1+exp((vR-vK)));
  end

  % If requested, compute derivative of Bellman operator
  if nargout>2
    dBellman_dV=mp.beta*(P{1}.*pk + P{2}.*(1-pk));
  end
end
```

STEP 1-4: MATLAB implementation of Γ_θ and Γ'_θ

```
function [ev, pk, dbellman_dev]=bellman_ev(ev0, mp, u, P)
  vK= u(:,1) + mp.beta*ev0;       % Value off keep
  vR= u(:,2) + mp.beta*ev0(1);    % Value of replacing

  % Need to recenter logsum by subtracting max(vK, vR)
  maxV=max(vK, vR);
  V=(maxV + log(exp(vK-maxV)  +  exp(vR-maxV)));
  ev=P{1}*V; % compute expected value of keeping
             % ev(1) is the expected value of replacing

  % If requested, also compute choice probability
  if nargout>1
    pk=1./(1+exp((vR-vK)));
  end

  % If requested, compute derivative of Bellman operator
  if nargout>2
    dbellman_dev=mp.beta*(P{1}.*pk');
    % Add additional term for derivative wrt ev(1),
    % since ev(1) enter logsum for all states
    dbellman_dev(:,1)=dbellman_dev(:,1)+mp.beta*P{1}*(1-pk);
  end
end
```

STEP 5: Provide analytical gradients of likelihood

Simple use of chain rule:

3. Gradients (wrt utility parameters) - similar to standard logit

$$\partial \ell_i^1(\theta)/\partial \theta_1 = \sum_t \sum_j [y_{j(it)} - P(j|x_{it}, \theta)] \partial v(x_{it}, j)/\partial \theta_1$$

2. Derivative of the choice specific value function

$$\partial v(j)/\partial \theta_1 = \partial u(j)/\partial \theta_1 + \beta \Pi(j) \partial \bar{V}/\partial \theta_1$$

- $\partial u(j)/\partial \theta_1$, is trivial to compute
- $\partial \bar{V}_\theta/\partial \theta$ can be obtained by the implicit function theorem

$$\partial \bar{V}_\theta/\partial \theta = [I - \bar{\Gamma}'_\theta]^{-1} \partial \bar{\Gamma}/\partial \theta$$

where $[I - \bar{\Gamma}'_\theta]^{-1}$ is a by-product of the N-K algorithm!!!.

1. Derivative of Bellman operator wrt. θ_1

$$\partial \bar{\Gamma}/\partial \theta_1 = \beta \sum_j P(j) \cdot \partial u(j)/\partial \theta_1$$

where $\cdot$ is the element by element product

STEP 5: MATLAB implementation of scores

```
function score = score(data, mp, P, pk, px_j, V0, du, dBellman_dV);
  y_j=[(1-data.d) data.d];  % choice dummies [keep replace]

  % Compute scores (use chain rule - three steps)

  % STEP 1: derivative of bellman operator wrt. utility parameters
  dbellman=pk.*du(:,:,1) + (1-pk).*du(:,:,2);
  if strcmp(mp.bellman_type, 'ev');
    dbellman=P{1}*dbellman;
  end

  % STEP 2: derivative of fixed point, V, wrt. utility parameters
  dV=(speye(size(dBellman_dV)) - dBellman_dV)\dbellman;

  % STEP 3: derivative of log-likelihood wrt. utility parameters
  score=0;
  for j=1:size(y_j, 2);
    dv= du(:,:,j) + mp.beta*P{j}*dV;
    score = score+ (y_j(:,j)-px_j(:,j)).*dv(data.x,:);
  end
end
```

STEP 6: BHHH

- Recall Newton-Raphson

$$\theta^{g+1} = \theta^{g} - \lambda \left(\Sigma_i H_i\left(\theta^{g}\right)\right)^{-1} \Sigma_i s_i\left(\theta^{g}\right)$$

- Berndt, Hall, Hall, and Hausman, (1974):
  Use outer product of scores as approx. to Hessian

$$\theta^{g+1} = \theta^{g} + \lambda \left(\Sigma_i s_i s_i'\right)^{-1} \Sigma_i s_i$$

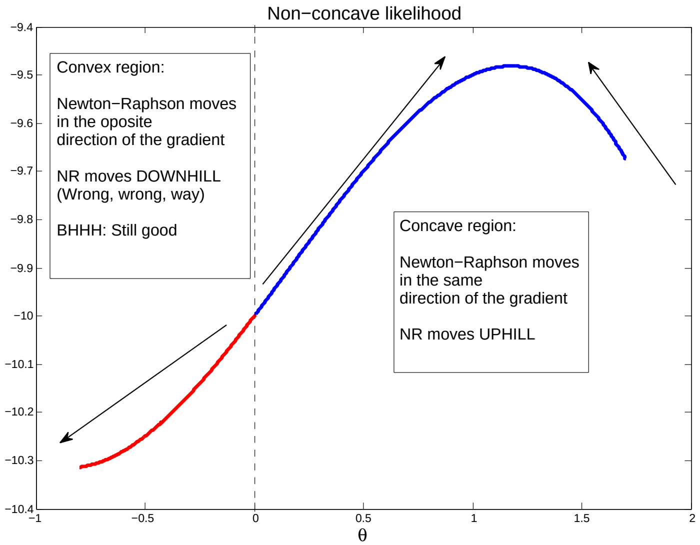
- Why is this valid? Information identity:

$$-E\left[H_i(\theta)\right] \quad = \quad E\left[s_i(\theta)\, s_i(\theta)'\right]$$

(valid for MLE if model is well specified)

STEP 6: BHHH

Some times linesearch may not help Newtons Method



STEP 6: BHHH

Advantages

- $\sum_i s_i s_i'$ is always positive definite
  I.e. it always moves uphill for λ small enough
- Does not rely on second derivatives

Disadvantages

- Only a good approximation
  - At the true parameters
  - for large N
  - for well specified models (in principle only valid for MLE)
- Only superlinear convergent - not quadratic

We can always use BHHH for first iterations and the switch to BFGS to update to get an even more accurate approximation to the hessian matrix as the iterations start to converge.

STEP 6: BHHH

Advantages

- $\Sigma_i s_i s_i'$ is always positive definite
  I.e. it always moves uphill for λ small enough
- Does not rely on second derivatives

Disadvantages

- Only a good approximation
  - At the true parameters
  - for large N
  - for well specified models (in principle only valid for MLE)
- Only superlinear convergent - not quadratic

We can always use BHHH for first iterations and the switch to BFGS to update to get an even more accurate approximation to the hessian matrix as the iterations start to converge.

STEP 6: BHHH

Use BHHH!

Convergence!

```
>> run_busdata
Structural Estimation using busdata from Rust(1987)
Bustypes       = [ 1  2  3  4 ]
Beta           =     0.99990
n              =   175.00000
Sample size    =  8156.00000

Method nfxp (mle)
    Param.                   Estimates         s.e.          t-stat
----------------------------------------------------------------------
    RC                          9.7915        1.2689          7.7168
    c                           1.3488        0.3460          3.8982
    p                  (1)      0.1070        0.0034         31.2111
    p                  (2)      0.5152        0.0055         93.0533
    p                  (3)      0.3622        0.0053         68.0413
    p                  (4)      0.0143        0.0013         10.8947
    p                  (5)      0.0009        0.0003          2.6469
----------------------------------------------------------------------
log-likelihood     = -8607.88844
runtime (seconds) =     0.07882
g'*inv(h)*g       = 7.26689e-09
```

MPEC versus NFXP-NK: sample size 6,000

β	Converged (out of 1250)	CPU Time (in sec.)	# of Major Iter.	# of Func. Eval.	# of Bellm. Iter.	# of N-K Iter.
			MPEC-Matlab			
0.975	1247	1.677	60.9	69.9		
0.985	1249	1.648	62.9	70.1		
0.995	1249	1.783	67.4	74.0		
0.999	1249	1.849	72.2	78.4		
0.9995	1250	1.967	74.8	81.5		
0.9999	1248	2.117	79.7	87.5		
			MPEC-AMPL			
0.975	1246	0.054	9.3	12.1		
0.985	1217	0.078	16.1	44.1		
0.995	1206	0.080	17.4	49.3		
0.999	1248	0.055	9.9	12.6		
0.9995	1250	0.056	9.9	11.2		
0.9999	1249	0.060	11.1	13.1		
			NFXP-NK			
0.975	1250	0.068	11.4	13.9	155.7	51.3
0.985	1250	0.066	10.5	12.9	146.7	50.9
0.995	1250	0.069	9.9	12.6	145.5	55.1
0.999	1250	0.069	9.4	12.5	141.9	57.1
0.9995	1250	0.078	9.4	12.5	142.6	57.5
0.9999	1250	0.070	9.4	12.6	142.4	57.7

Summary remarks

Su and Judd (Econometrica, 2012) used an inefficient version of NFXP

- that solely relies on the method of successive approximations to solve the fixed point problem.

Using the efficient version of NFXP proposed by Rust (1987) we find:

- MPEC and NFXP-NK are similar in performance when the sample size is relatively small.
- NFXP does not slow down as $\beta \to 1$

Desirable features of MPEC

- Ease of use by people who are not interested in devoting time to the special-purpose programming necessary to implement NFXP-NK.
- Can easily be implemented in the intuitive AMPL language.

Inference

- NFXP: Trivial to compute standard errors by inverting the Hessian from the unstrained likelihood (which is a by-product of NFXP).
- MPEC: Standard errors can be computed inverting the bordered Hessian Reich and Judd (2019): Develop simple and efficient approach to compute confidence intervals.

MPEC does not seem appropriate when estimating life cycle models

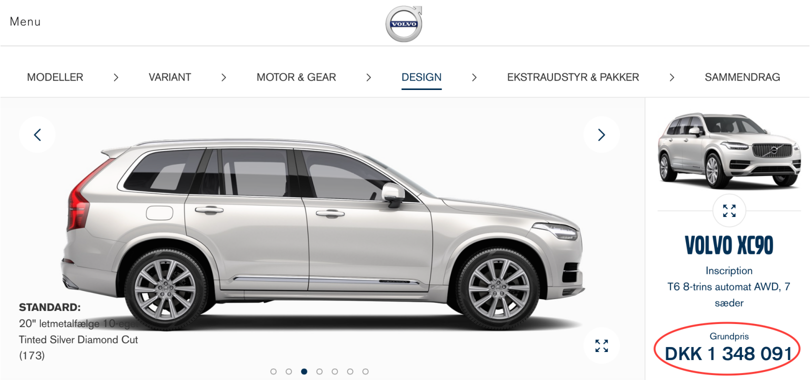
PART II
Equilibrium Trade in Automobiles
(JPE, 2022)

The Doubly Nested Fixed Point Algorithm (DNFXP)

Kenneth Gillingham, Yale University,
Fedor Iskhakov, Australian National University,
Anders Munk-Nielsen, University of Copenhagen,
John Rust, Georgetown University, and
Bertel Schjerning, University of Copenhagen

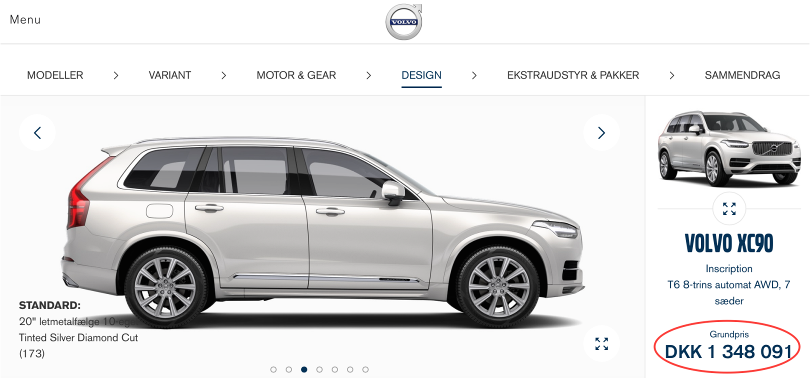
How much is a Volvo in Denmark?

About $200,000!



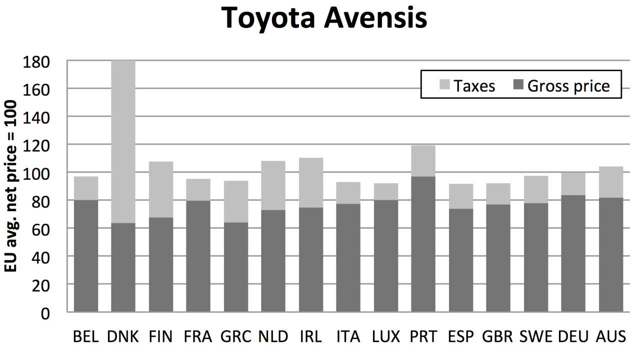
MSRP in US: $62,350

About $200,000!



MSRP in US: $62,350

Danish car registration tax: 180%! (plus 25% VAT)



Car taxes in Denmark

- **Annual Revenue:** 30–50 billion DKK
- $\cong$ 2–3 pct. of GDP
- $\cong$ 4-7 pct. of total tax revenue
- Most revenue originates from taxation of ownership and registration of new cars.
- Car *usage* tax (fuel tax) is not unusually high

Policy question: What are the effects of switching car taxes from purchase to usage?

Our main empirical finding: The Danish new car tax is “over the top” of the Laffer curve. We identify welfare improving tax policies that reduce the new car tax and raise the gas tax, generating higher consumer welfare, government tax revenue, and reducing CO_2 emissions.

Equilibrium trade in automobiles

A tractable dynamic equilibrium model of trading in new and used cars

- **Cars**
  - initially sold as new in the primary market and then traded in (secondary) used-car markets until they are eventually scrapped
  - $j \in \{1, ..., J\}$ indexes car types (e.g. makes/models)
  - $a \in \{1, ..., \bar{a}\}$ indexes the ages of the traded cars.
- **Consumers**
  - Driving decision: how much to drive
  - Ownership decisions: purchase, replacement, trading, and scrapping decisions
  - infinitely lived and maximize expected discounted utility
  - heterogeneous: choice specific IID EV/GEV taste shocks and discrete types τ
- **Equilibrium**
  - Endogenous used car prices, P, and ownership distribution, q

DNFXP algorithm (roadmap)

- **Outer optimization:** Maximum likelihood search over θ
- **Inner equilibrium solver:** Find prices, $P^\star$, so $ED(P^\star, q(P^\star)) = 0$
- **Excess demand:** Each trial value of P requires
  1. Solve single agent DP/fixed point given P
  2. Compute transition matrices $\Omega(P)$ and Q
  3. Find stationary holdings distribution $q(P) : q = q\Omega(P)Q$
  4. Evaluate excess demand $ED(P, q(P))$

DNFXP algorithm (roadmap)

- **Outer optimization:** Maximum likelihood search over θ
- **Inner equilibrium solver:** Find prices, $P^{\star}$, so $ED(P^{\star}, q(P^{\star})) = 0$
- **Excess demand:** Each trial value of P requires
  1. Solve single agent DP/fixed point given P
  2. Compute transition matrices $\Omega(P)$ and Q
  3. Find stationary holdings distribution $q(P) : q = q\Omega(P)Q$
  4. Evaluate excess demand $ED(P, q(P))$

DNFXP algorithm (roadmap)

- **Outer optimization:** Maximum likelihood search over θ
- **Inner equilibrium solver:** Find prices, $P^\star$, so $ED(P^\star, q(P^\star)) = 0$
- **Excess demand:** Each trial value of P requires
  1. Solve single agent DP/fixed point given P
  2. Compute transition matrices $\Omega(P)$ and Q
  3. Find stationary holdings distribution $q(P) : q = q\Omega(P)Q$
  4. Evaluate excess demand $ED(P, q(P))$

Utility of car ownership and consumer heterogeneity

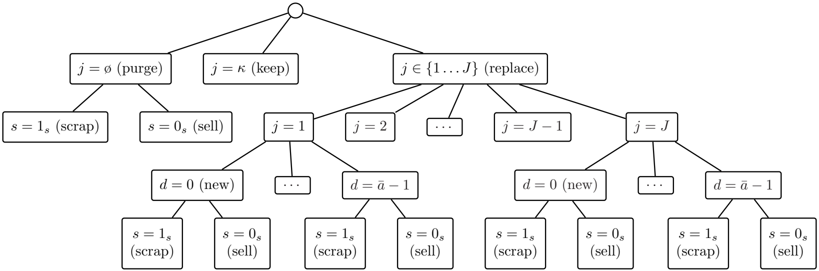
$$\text{Utility} = u(i, a) - \mu\big[\text{operating costs} + \text{trade and transaction costs}\big] + \epsilon$$

- Car utility $u(i, a)$ is a decreasing function of car age a that reflects
  - decreasing utility of car services
  - increasing cost of maintenance
- Marginal utility of money μ

Idiosyncratically heterogeneous consumers

- **Extreme value** consumer types (taste shifters)
- GEV specification for $\epsilon \rightarrow$ nested choices to allow correlation between alternatives
- Logit choice probabilities and analytic expectations

Consumer choice tree



Zurcher on Steroids: Car owners' trading problem

$$V(i,a,\epsilon) = \max \left\{ \begin{array}{l} v(i,a,\kappa) + \epsilon(\kappa); \\ \max_{s\in\{1_s,0_s\}} \left[v(i,a,\emptyset,s) + \epsilon(\emptyset,s) \right]; \\ \max_{\substack{j\in\{1,\dots,J\},\\ d\in\{0,1,\dots,\bar{a}-1\},\\ s\in\{1_s,0_s\}}} \left[v(i,a,j,d,s) + \epsilon(j,d,s) \right] \end{array} \right\}$$

States Choices

- Existing car (i,a), traded car (j,d)
- When existing car (i,a) is replaced, there is additional scrappage choice $s \in \{0_s, 1_s\}$: to sell or to scrap the replaced car.
- Similar recursive maximization problems for consumers with no car and owner of car of terminal age $\bar{a}$

Choice specific value functions

$$
\begin{aligned}
v(i, a, \emptyset, 1_s) &= u(\emptyset) + \mu \underline{P}_i + \beta EV(\emptyset) \\
v(i, a, \emptyset, 0_s) &= u(\emptyset) + \mu[P_{ia} - T_s(P, i, a)] + \beta EV(\emptyset) \\
v(i, a, \kappa) &= u(i, a) + \beta\big(1-\alpha\big) EV(i, a+1) + \beta\alpha EV(i, \bar{a}) \\
v(i, a, j, d, 1_s) &= u(j, d) - \mu[P_{jd} - \underline{P}_i + T_b(P, j, d)] + \\
&\quad + \beta\big(1-\alpha\big) EV(j, d+1) + \beta\alpha EV(j, \bar{a}) \\
v(i, a, j, d, 0_s) &= u(j, d) - \mu[P_{jd} - P_{ia} + T_s(P, i, a) + T_b(P, j, d)] + \\
&\quad + \beta\big(1-\alpha\big) EV(j, d+1) + \beta\alpha EV(j, \bar{a})
\end{aligned}
$$

States Choices $\rightarrow$ Current period utility Future value

- Similar expressions for consumers with no car and owner of car of terminal age $\bar{a}$

Solving the consumers' problem

$$EV(i,a) = \sigma \log \left\{ \sum_{j,d,s} \exp \left[\frac{v(i,a,j,d,s)}{\sigma} \right] \right\}$$

- **Fixed point of Bellman operator in *EV* space**

$$EV(P) = \Gamma\big(EV(P), P\big)$$

- Conditional choice probabilities are then analytical, similar to

$$\Pi(j,d,s|i,a) = \frac{\exp\left[v(i,a,j,d,s)/\sigma\right]}{\sum_{j'} \exp\left[v(i,a,j',d',s')/\sigma\right]}.$$

  - *Note:* CCPs implicitly depend on car prices, P
  - The sell/scrap decision s is separable (see paper for details)
  - Fixed point solved using gradient-based Newton method with very precise starting values

DNFXP algorithm (roadmap)

- **Outer optimization:** Maximum likelihood search over θ
- **Inner equilibrium solver:** Find prices, $P^{\star}$, so $ED(P^{\star}, q(P^{\star})) = 0$
- **Excess demand:** Each trial value of P requires
  1. Solve single agent DP/fixed point given P
  2. Compute transition matrices $\Omega(P)$ and Q
  3. Find stationary holdings distribution $q(P) : q = q\Omega(P)Q$
  4. Evaluate excess demand $ED(P, q(P))$

DNFXP algorithm (roadmap)

- **Outer optimization:** Maximum likelihood search over θ
- **Inner equilibrium solver:** Find prices, $P^\star$, so $ED(P^\star, q(P^\star)) = 0$
- **Excess demand:** Each trial value of P requires
  1. Solve single agent DP/fixed point given P
  2. Compute transition matrices $\Omega(P)$ and Q
  3. Find stationary holdings distribution $q(P) : q = q\Omega(P)Q$
  4. Evaluate excess demand $ED(P, q(P))$

DNFXP algorithm (roadmap)

- **Outer optimization:** Maximum likelihood search over θ
- **Inner equilibrium solver:** Find prices, $P^\star$, so $ED(P^\star, q(P^\star)) = 0$
- **Excess demand:** Each trial value of P requires
  1. Solve single agent DP/fixed point given P
  2. Compute transition matrices $\Omega(P)$ and Q
  3. Find stationary holdings distribution $q(P) : q = q\Omega(P)Q$
  4. Evaluate excess demand $ED(P, q(P))$

Stationary flow market equilibrium framework

Assumptions

1. Infinitely inelastic supply of new cars $\overline{P}_j$
2. Infinitely elastic demand for scrapped cars $\underline{P}_j$
3. $J(\bar{a}-1)$ endogenously determined used car prices P_{jd}

Definition: Ownership Distribution

$$q = \Big(\underbrace{(q_{11}, \dots, q_{1\bar{a}})}_{\text{car 1}}, \dots, \underbrace{(q_{J1}, \dots, q_{J\bar{a}})}_{\text{car } J}, \underbrace{q_{\emptyset}}_{\text{no car}}\Big) \in \mathbb{R}^{J\bar{a}+1}$$

- q_{ia} is the fraction of consumers holding car i of age a
- By our timing assumption new cars purchased in any time period are accounted for as one-years-old cars in the next time period (so q_{j0} is undefined)

Stationary flow market equilibrium framework

Assumptions

1. Infinitely inelastic supply of new cars $\overline{P}_j$
2. Infinitely elastic demand for scrapped cars $\underline{P}_j$
3. $J(\bar{a}-1)$ endogenously determined used car prices P_{jd}

Definition: Ownership Distribution

$$q = \Big(\underbrace{(q_{11}, \ldots, q_{1\bar{a}})}_{\text{car 1}}, \ldots, \underbrace{(q_{J1}, \ldots, q_{J\bar{a}})}_{\text{car } J}, \underbrace{q_{\emptyset}}_{\text{no car}}\Big) \in \mathbb{R}^{J\bar{a}+1}$$

- q_{ia} is the fraction of consumers holding car i of age a
- By our timing assumption new cars purchased in any time period are accounted for as one-years-old cars in the next time period (so q_{j0} is undefined)

Equilibrium

Definition: Stationary Equilibrium

A pair $q^\star \in \mathbb{R}^{J\bar{a}+1}$ and $P^\star \in \mathbb{R}^{J(\bar{a}-1)}$ such that

1. Consumers maximize expected discounted utility,
2. Secondary market clears for all tradeable cars,
3. Ownership distribution is time-invariant.

The dynamics of the ownership distribution q are described by

- *Trade* transision probability matrix $\Omega(P)$ composed of conditional choice probabilities of trading decisions
- *Physical* transition probability matrix Q: ageing of cars + stocastic transitions to terminal age $\bar{a}$ (involuntary scrappage)

In the paper we also prove the *flow property* of this stationary equilibrium: all cars scrapped in each period are replenished by the exact amount of new cars bought in the same period

Trade transition probability matrix

$\Omega(P) =$ $J\bar{a} + 1 \times J\bar{a} + 1$ matrix

$$\begin{bmatrix} \Delta_{11}(P) + \Lambda_1(P) & \Delta_{12}(P) & \dots & \Delta_{1J}(P) & \Delta_{1\o}(P) \\ \Delta_{21}(P) & \Delta_{22}(P) + \Lambda_2(P) & \dots & \Delta_{2J}(P) & \Delta_{2\o}(P) \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \Delta_{J1}(P) & \Delta_{J2}(P) & \dots & \Delta_{JJ}(P) + \Lambda_J(P) & \Delta_{J\o}(P) \\ \Delta_{\o 1}(P) & \Delta_{\o 2}(P) & \dots & \Delta_{\o J}(P) & \Pi(\o|\o, P) \end{bmatrix}$$

Then $q \cdot \Omega(P)$ is distribution of cars after the trading phase

- $\Delta_{ij}(P)$ composed of choice proabbilities of trading car i to car j
- $\Lambda_i(P)$ composed of keeping probabilities for car i
- $\Pi(\o|\o, P)$ is probability to remain in the no car state

Physical transition probability matrix

$Q =$ $J\bar{a}+1 \times J\bar{a}+1$ matrix

$$\begin{bmatrix} Q_1 & 0 & 0 & \dots & 0 & 0 \\ 0 & Q_2 & 0 & \dots & 0 & 0 \\ 0 & 0 & Q_3 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \\ 0 & 0 & 0 & \dots & Q_J & 0 \\ 0 & 0 & 0 & \dots & 0 & 1 \end{bmatrix} \quad Q_j = \begin{bmatrix} 0 & 1-\alpha_j & \dots & 0 & \alpha_j \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 1-\alpha_j & \alpha_j \\ 0 & 0 & \dots & 0 & 1 \\ 1-\alpha_j & 0 & \dots & 0 & \alpha_j \end{bmatrix}$$

- Aging of cars with probability $1-\alpha_j$
- Total loss accidents with probability α_j
- Last row in each Q_j block is applies to the purchased new cars

$q \cdot \Omega(P) Q$ is ownership distribution in the next period

The stationary holdings distribution

$$\underbrace{q}_{t} \quad \rightarrow \quad \underbrace{q\Omega(P)}_{\text{after trading}} \quad \rightarrow \quad \underbrace{q\Omega(P)Q}_{t+1}$$

Condition for time invariance of the ownership distribution:

$$q = q\Omega(P)Q$$

Theorem (Uniqueness of stationary ownership distribution)

If scale of GEV shocks disctribution is positive then stationary ownership distribution is unique.

DNFXP algorithm (roadmap)

- **Outer optimization:** Maximum likelihood search over θ
- **Inner equilibrium solver:** Find prices, $P^\star$, so $ED(P^\star, q(P^\star)) = 0$
- **Excess demand:** Each trial value of P requires
  1. Solve single agent DP/fixed point given P
  2. Compute transition matrices $\Omega(P)$ and Q
  3. Find stationary holdings distribution $q(P) : q = q\Omega(P)Q$
  4. Evaluate excess demand $ED(P, q(P))$

DNFXP algorithm (roadmap)

- **Outer optimization:** Maximum likelihood search over θ
- **Inner equilibrium solver:** Find prices, $P^\star$, so $ED(P^\star, q(P^\star)) = 0$
- **Excess demand:** Each trial value of P requires
  1. Solve single agent DP/fixed point given P
  2. Compute transition matrices $\Omega(P)$ and Q
  3. Find stationary holdings distribution $q(P) : q = q\Omega(P)Q$
  4. Evaluate excess demand $ED(P, q(P))$

DNFXP algorithm (roadmap)

- **Outer optimization:** Maximum likelihood search over θ
- **Inner equilibrium solver:** Find prices, $P^\star$, so $ED(P^\star, q(P^\star)) = 0$
- **Excess demand:** Each trial value of P requires
  1. Solve single agent DP/fixed point given P
  2. Compute transition matrices $\Omega(P)$ and Q
  3. Find stationary holdings distribution $q(P) : q = q\Omega(P)Q$
  4. Evaluate excess demand $ED(P, q(P))$

Excess demand functions

- **Demand:** Fraction of consumers buying a given car (j, d):

$$D_{jd}(P, q) = \Pi(j, d|\o, P)q_{\o} + \sum_{i=1}^{J}\sum_{a=1}^{\bar{a}} \Pi(j, d|i, a, P)q_{ia}$$

- **Supply:** Fraction of owners that sell (not scrap) their car (j, d)

$$S_{jd}(P, q) = \big(1 - \Pi(\kappa|j, d, P)\big)\big(1 - \Pi(1_s|j, d, P)\big)q_{jd}$$

- **Market clearing condition** is the non-linear system of equations in ownership shares q and prices P

$$ED\big(P, q\big) \equiv D(P, q) - S(P, q) = 0$$

  - Given the stationarity condition $q = q(P)$
  - $J(\bar{a} - 1)$ equations with $J(\bar{a} - 1)$ unknowns

Existence of stationary equilibrium

Theorem (Equilibrium existence)

The stationary equilibrium for the automobile economy with the idiosyncratically heterogeneous consumers $(q^\star, P^\star)$ *exists, and in equilibrium it holds:*

$$\begin{aligned} q^\star &= q^\star \Omega(P^\star) Q, \\ 0 &= ED(P^\star, q^\star). \end{aligned}$$

- Only existence: $q^\star$ is unique, but unclear about $P^\star$
- However, have not seen any signs of multiplicity in computations

DNFXP algorithm (roadmap)

- **Outer optimization:** Maximum likelihood search over θ
- **Inner equilibrium solver:** Find prices, $P^{\star}$, so $ED(P^{\star}, q(P^{\star})) = 0$
- **Excess demand:** Each trial value of P requires
  1. Solve single agent DP/fixed point given P
  2. Compute transition matrices $\Omega(P)$ and Q
  3. Find stationary holdings distribution $q(P) : q = q\Omega(P)Q$
  4. Evaluate excess demand $ED(P, q(P))$

How to compute stationary flow equilibrium quickly?

Solving non-linear system of equations:

- **Gradient-based** solver! (in a series of lemmas show that all major objects in the model are smooth functions of prices, see paper)
- **Analytical derivatives**
- **Precise starting values** from the solution of the similar problem without transaction costs (linear system of equations, see appendix)

Newton method is therefore applied:

1. When solving the DP problem (Newton-Kantorovich)
2. When solving for equilibrium prices
3. When maximizing likelihood

- Chain rule of calculus used everywhere to build up gradients from already computed parts
- Run time in seconds for reasonable size problems on a laptop using simple Matlab implementation

Adding persistent consumer heterogeneity

We extend the model to allow for sevel *types* of consumer heterogeneity:
• time-invariant • time-variant • combination of the two

- Existence theorems
- Computational algorithm is linear in the number of types
- Allows for sorting of consumers into the ages and types of cars
  - Rich hold newer better cars, poor hold older worse cars
  - Gains from trade and longer surviving cars
- The equilibrium conditions change only slightly

$$\text{Stationarity by type:} \quad \forall \tau \; q^\star_\tau = q^\star_\tau \Omega_\tau(P^\star) Q$$

$$\text{Market clearing in a sum:} \quad 0 = \sum_{\tau=1}^{N} f_\tau ED(P^\star, q^\star_\tau).$$

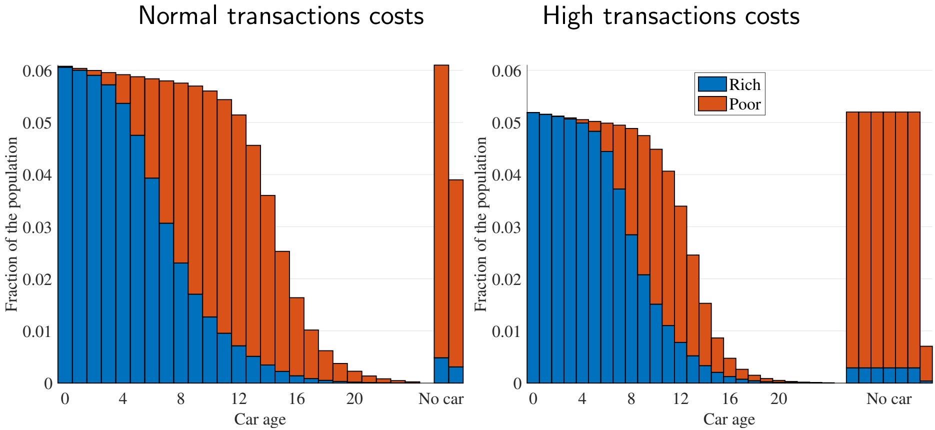
- Market clearing condition integrated over types

Gains from trade between rich and poor consumers

Rich mans Volvo

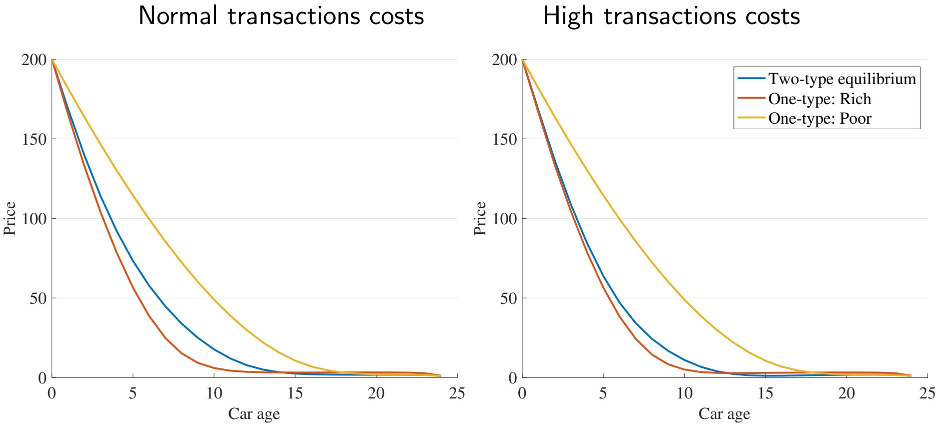
Poor mans Volvo

Illustrative example: ownership by two consumer types



- Sorting of consumers in each regime
- Heterogeneous effects of transaction costs
- Replication code (see `run_illustrations.m`) at `https://github.com/fediskhakov/EQB`

Illustrative example: equilibrium prices - two consumer types



High transactions costs:

- Equilibrium prices similar to economy where all consumers are rich (many poor consumers are now driven out of the market)
- Transactions costs limits gains from trade
  $\rightarrow$ partially "kills off" the market for used cars.
  (cars are scrapped earlier)

DNFXP algorithm (roadmap)

- **Outer optimization:** Maximum likelihood search over θ
- **Inner equilibrium solver:** Find prices, $P^\star$, so $ED(P^\star, q(P^\star)) = 0$
- **Excess demand:** Each trial value of P requires
  1. Solve single agent DP/fixed point given P
  2. Compute transition matrices $\Omega(P)$ and Q
  3. Find stationary holdings distribution $q(P) : q = q\Omega(P)Q$
  4. Evaluate excess demand $ED(P, q(P))$

Doubly Nested Fixed Point MLE estimator

- **Data:** counts $N_{x'x\tau}$ of transitions of household from state x (combining ownership and observable characteristics) to x' by the observed types τ
- Let θ denote the vector of structural parameters
- Transition probability $\Pi(x'|x,\tau,\theta)$ of the observed household state x composed of choice and transition probabilities at θ
  (see paper for details)
- Likelihood function

$$L(\theta) = \sum_{\tau} \sum_{x'} \sum_{x} N_{x'x\tau} \log \Pi(x'|x,\tau,\theta)$$

- Analytic gradient of the likelihood function again relies on using chain rule of calculus and already computed derivatives
- BHHH algorithm for approximation of Hessian

Simulating the effects of a hypothetical tax reform

Proposed Danish IRUC reform:

- lowers registration taxes, and
- raises usage taxes (road charging or gas tax).

Outcomes of interest:

- Equilibrium dynamics of car ownership and type choice:
  - new car sales and trade in secondary markets
  - fleet age and scrappage
  - value of the car stock
- Driving, fuel demand, and emissions
- Redistribution and welfare
- Need to capture these effects simultaneously

To implement the counterfactual simulation:

1. Estimate the model using Danish register data
2. Cut the registration tax rates for new vehicles by half
3. Increase the fuel tax rate such that revenue is unchanged
4. Compute economic/welfare/environmental implications

Utility specification with driving

Consider a utility function (indexes i and τ dropped)

$$u(a, x) = u_{\text{car}}(a) + u_{\text{drive}}(a, x) + \mu[\text{trade} + \text{transaction cost}]$$

$$\text{Onwership utility: } u_{\text{car}}(a) = \alpha_0 + \alpha_1 a + \alpha_2 a^2$$

$$\text{Utility from driving: } u_{\text{drive}}(a, x) = (\gamma_0 + \gamma_1 a)x - \mu p x + \frac{\phi}{2} x^2$$

- x is kilometers driven, p is cost per kilometer inclusive of tax
- parameters may be specific to car type i and consumer type τ
- See paper and online appendix for the estimated values of parameters

Assumption

The probability of an accident and other physical deterioration in an automobile is independent of the amount of driving x.

$\Rightarrow$ driving is a static subproblem of the overall DP problem that can be solved independently

Optimal amount of driving

Structural driving equation implied by the F.O.C.

$$x^*(p,a) = -\frac{1}{\phi}\left[\gamma_0 + \gamma_1 a - \mu p\right] = d_0 + d_1 a + d_2 p$$

Plugging optimal driving back into $u(a,x)$

$$u\big(a, x^*(p,a)\big) - \mu[\text{trade + transaction cost}] =$$
$$\alpha_0 + \alpha_1 a + \alpha_2 a^2 - \frac{1}{2\phi}\left[\gamma_0 + \gamma_1 a - \mu p\right]^2 = u_0 + u_1 a + u_2 a^2$$

- (d_0, d_1, d_2) are reduced form parameters that can be estimated separately from data on driving
- (u_0, u_1, u_2) are indirect utility parameters that can be identified from the dynamic model model together with μ
- Then structural parameters $(\alpha_0, \alpha_1, \alpha_2, \gamma_0, \gamma_1, \mu, \phi)$ are identified

Coefficient specification and estimations stages

1. Least squares estimation of reduced form parameters of driving equation (d_0, d_1, d_2)
   - intercept $d_0 = d_0^\tau + d_0^i$ with car and consumer fixed effects
   - common coefficient with age d_1
   - coefficient with driving cost $d_2 = d_2^\tau$ by consumer types

2. DNFXP MLE estimation of indirect utility coefficients (u_0, u_1, u_2) and the remaining parameters (μ, ϕ), transaction costs and accident probabilities
   - intercept u_0^τ and age coefficient u_1^τ by consumer type
   - marginal utility of money μ_τ by consumer type
   - Accident probabilities by car type i
   - Buyer and seller transaction costs (see paper for details)

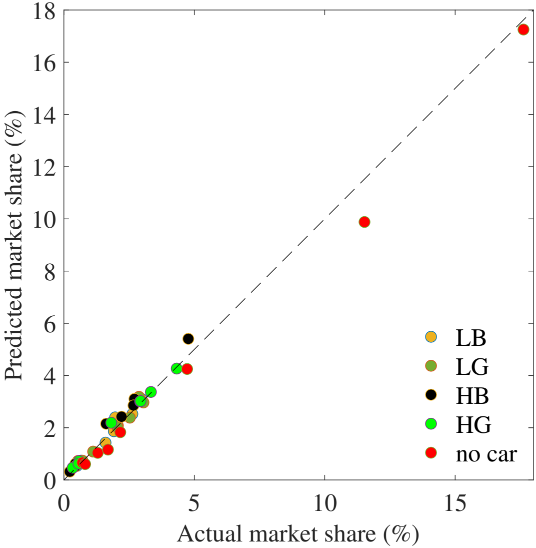
3. Back out structural parameters in order to run counterfactual simulations

See paper and online appendix for the estimated values of parameters

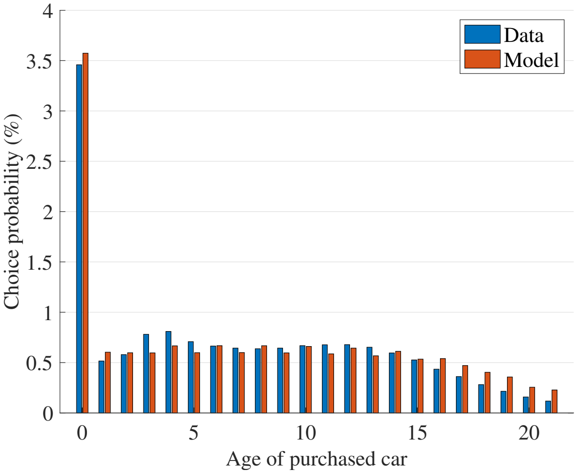
Model captures key features of Danish households

- Poor households are significantly more likely not to own a car than rich ones which are also willing to pay more for any type of car
- Couples are more likely to own cars and generally have higher willingness to pay for cars than singles
- High work distance households are relatively more likely to own cars and have higher willingness to pay for cars than those with low work distance
- All households preferred the heavy cars to the light ones and brown cars to green ones:
  heavy brown $\succ$ heavy green $\succ$ light brown $\succ$ light green
- Households with high work distance drive much more than those with low, and more so for the rich
- Model implies fuel price elasticities between -0.10 and -0.60 across households, similar to Gillingham and Munk-Nielsen (2015)

Model fit: Household-specific market shares

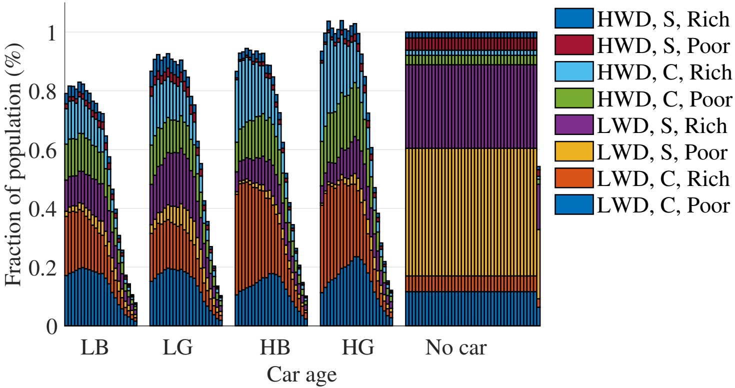


Model fit: Actual and predicted probability purchase

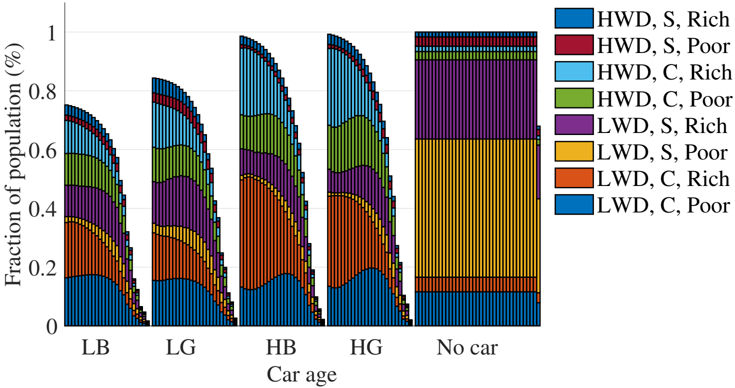


(a) Probability of purchase

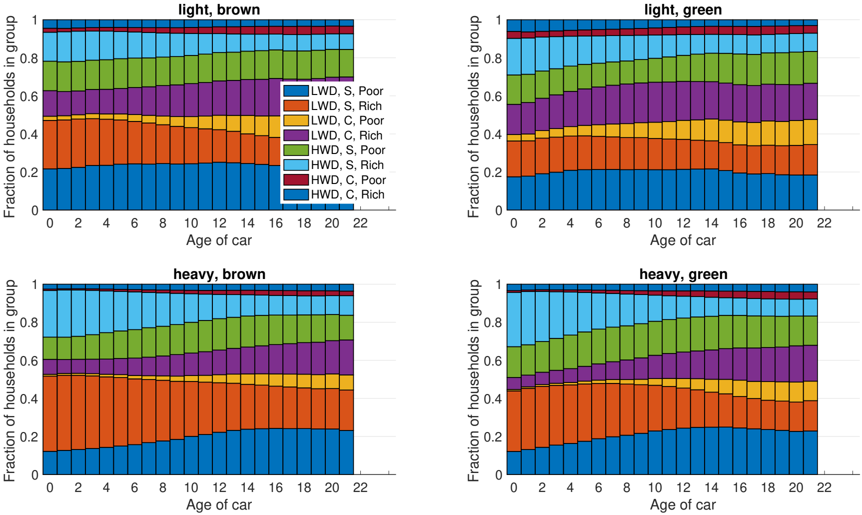
Model fit: Observed ownership distribution



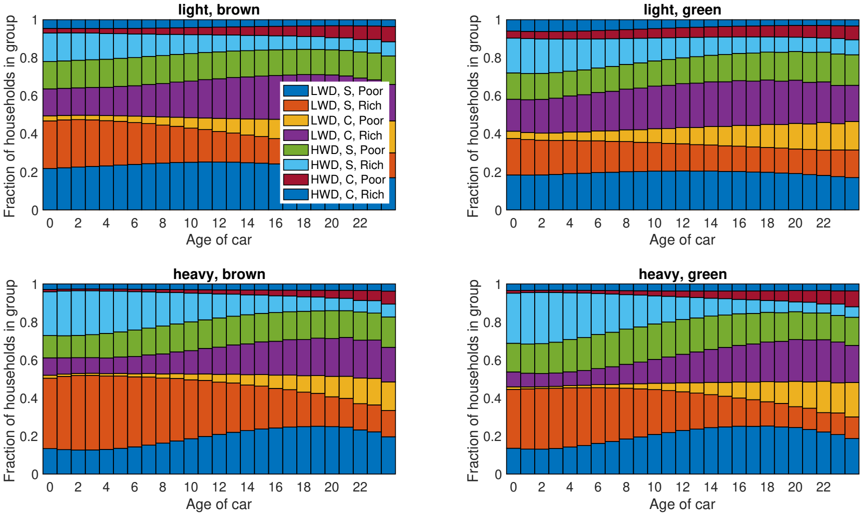
Model fit: Predicted ownership distribution



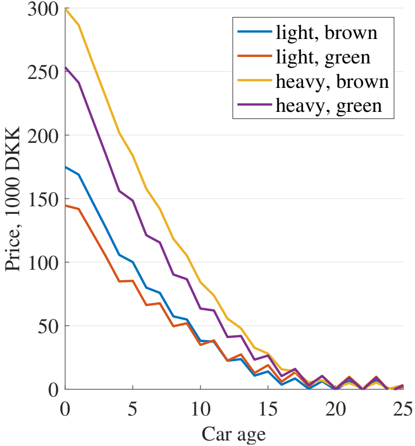
Model fit: Observed sorting



Model fit: Predicted sorting



Predicted equilibrium prices at secondary market



- Predicted prices similar to used car prices recommended by DAF.

Counterfactual simulation

Halving registration tax: Reduction in new car price between 25.6% (cheapest car), 28,6% (most expensive car)

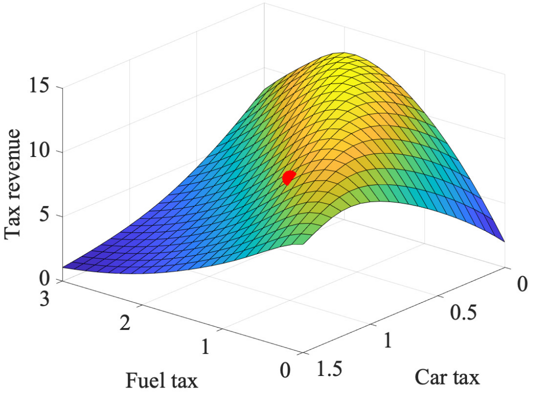
We consider the following four scenarios:

1. **Baseline:** Calibration under Danish tax rates from 2008.
2. **Naive, expected:** Non-equilibrium simulation:
   - Assume new and used car prices drops proportionally
   - Increase fuel taxes to keep total tax revenue neutral (Fuel price increase from 56% to 76% of the price at the pump)
3. **Naive, realized:** Equilibrium simulation:
   - Policy as above + market equilibrium imposed
   - Not revenue neutral in equilibrium (20% lower revenue than expected!)
4. **Sophisticated policy maker:**
   - Policy revenue-neutral in equilibrium
   - Fuel tax is lower, but leads to higher total tax revenue (compared to realized tax revenue for naive policymaker)

Policy Simulation Results

	Baseline	Naive, expected	Naive, realized	Sophisticated
Policy choice variables				
Registration tax (bottom rate)	1.050	0.525	0.525	0.525
Registration tax (top rate)	1.800	0.900	0.900	0.900
Fuel tax (share of pump price)	0.573	0.761	0.761	0.732
Prices				
Price, light, brown (1000 DKK)	174.902	129.532	129.532	129.532
Price, light, green (1000 DKK)	144.551	107.532	107.532	107.532
Price, heavy, brown (1000 DKK)	299.452	214.048	214.048	214.048
Price, heavy, green (1000 DKK)	253.397	182.796	182.796	182.796
Fuel price (DKK/l)	8.322	14.885	14.885	13.243
Outcomes				
Social surplus (1000 DKK)	9.382	11.281	8.439	10.203
Total tax revenue (1000 DKK)	9.391	9.391	7.452	9.391
Fuel tax revenue (1000 DKK)	4.282	5.184	4.983	6.224
Car tax revenue (1000 DKK)	5.110	4.207	2.468	3.167
Non-CO2 externalities (1000 DKK)	6.751	3.385	3.281	4.711
Externalities (1000 DKK)	7.374	3.702	3.586	5.157
Consumer surplus (1000 DKK)	7.364	5.592	4.573	5.969
CO2 (ton)	2.148	1.094	1.052	1.537
Driving (1000 km)	10.861	5.446	5.279	7.580
E(car age)	6.507	3.080	4.336	5.417
Pr(no car)	0.367	0.535	0.534	0.418

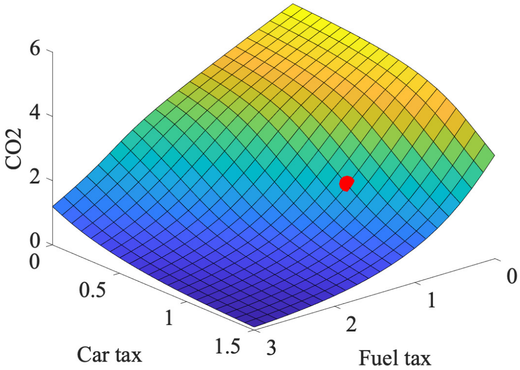
Laffer curves for new car registration tax and fuel tax



New car registration and the fuel tax relative to the baseline level of 1.

Tax revenue from new car sales tax and fuel tax.

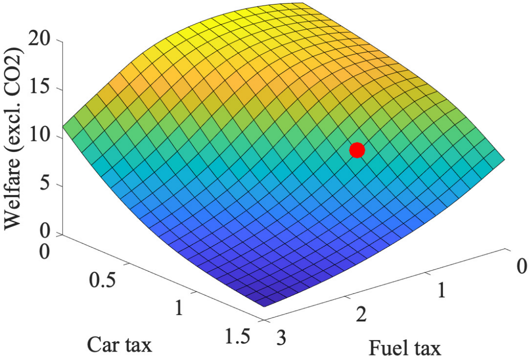
CO_2 emissions vs. new car registration and fuel taxes



New car registration and the fuel tax relative to the baseline level of 1.

Tax revenue from new car sales tax and fuel tax.

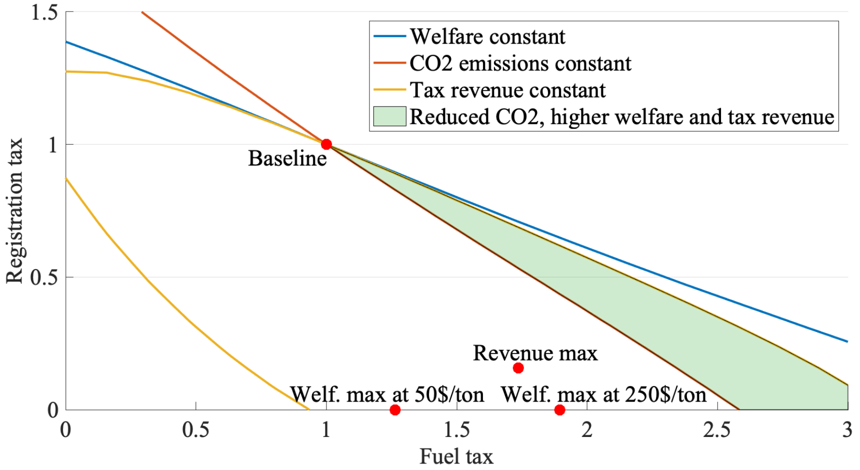
Social welfare (ex CO_2) vs. registration and fuel taxes



New car registration and the fuel tax relative to the baseline level of 1.

Tax revenue from new car sales tax and fuel tax.

Trade-off between CO_2 emissions and social welfare



Conclusion

- **Theory contribution:** characterize and prove existence of equilibrium in a tractable model of primary and secondary markets.
- **Applied contribution:** tractable model with
  - Transactions, scrappage, consumer/car heterogeneity,
  - Flexible utility: estimating 131 parameters with $39 \cdot 10^6$ observations in under 30 min on a laptop
- **Conclusion:** High Danish taxes above the Laffer curve's top point
  - "naive" model overestimates the strength of this effect,
  - possibly leading to detrimental policies for tax revenues and the environment
  - Opportunity to reduce CO_2, and increase tax revenues and social welfare

Conclusion

- **Theory contribution:** characterize and prove existence of equilibrium in a tractable model of primary and secondary markets.
- **Applied contribution:** tractable model with
  - Transactions, scrappage, consumer/car heterogeneity,
  - Flexible utility: estimating 131 parameters with $39 \cdot 10^6$ observations in under 30 min on a laptop

- **Conclusion:** High Danish taxes above the Laffer curve's top point
  - "naive" model overestimates the strength of this effect,
  - possibly leading to detrimental policies for tax revenues and the environment
  - Opportunity to reduce CO_2, and increase tax revenues and social welfare

Conclusion

- **Theory contribution:** characterize and prove existence of equilibrium in a tractable model of primary and secondary markets.
- **Applied contribution:** tractable model with
  - Transactions, scrappage, consumer/car heterogeneity,
  - Flexible utility: estimating 131 parameters with $39 \cdot 10^6$ observations in under 30 min on a laptop

- **Conclusion:** High Danish taxes above the Laffer curve's top point
  - "naive" model overestimates the strength of this effect,
  - possibly leading to detrimental policies for tax revenues and the environment
  - Opportunity to reduce CO_2, and increase tax revenues and social welfare