Maximum Likelihood Estimation of Discrete Markov Decision Models using NFXP

Bertel Schjerning

University of Copenhagen

ZICE 2014

Road Map

1. Dynamic Discrete Choice Problems, Infinite Horizon Case
   - General Behavioral framework
   - Structural Estimation by NFXP
   - Example: Rusts model
   - Death to good old NFXP?
2. Maximum Likelihood Estimation of Discrete Markov Decision Models by Sieve Approximations
   - Approximation of Expected Value Function
   - DP Mixed Logit
   - Approximation errors: Implications for statistical inference
3. A model of Vehicle Ownership, Type Choice and Usage
   - Finite horizon
   - Combine discrete continuous states
   - Reformulate model as dynamic discrete choice model

PART I

Dynamic Discrete Choice Problems, Infinite Horizon Case

The General Problem

Bellman equation

$$V_\theta(z) = \max_{d \in \mathcal{D}(z)} \{u_{\theta_u}(z,d) + \beta \int V_\theta(z') p_{\theta_p}(z'|z,d) dz'\}$$

u_{θ_u} and p_{θ_p}: known up to a set of parameters, θ_u and θ_p

- **The agent's problem:** Maximize expected sum of current and future discounted utilities
  - d : Discrete control variable, $d \in \mathcal{D}(z) = \{1, 2, .., J\}$.
  - z : Current state, fully observed by agent
  - $z\prime$: Future state; possibly continuous and subject to uncertainty
- **The agents beliefs about $z\prime$:**
  - Obeys a (controlled) Markov transition probability $p_{\theta_p}(z_{t+1}|z_t, d_t)$
- **Model solution, $V_\theta(z)$**
  - Find the fixed point for the Bellman equation
  - $V_\theta(z)$ can be a very high dimensional function

MLE of Markov Decision Models

- **Econometric problem:** Given observations of n individual agents over T time periods:

$$(d_{i,t}, x_{i,t}),\ t = 1, ..., T \text{ and } i = 1, ..., n,$$

we wish to estimate the underlying Markov decision model.

  - $d_{i,t}$: individual i's discrete choice at time t.
  - $x_{i,t}$: sub-component of individual i's state vector $z_{i,t}$.
  - $z_t = (x_t, \varepsilon_t)$ where ε_t is a set of unobservables.

- **MLE:**

$$\hat{\theta} = \arg\max_{\theta} \ell_n(\theta), \quad \ell_n(\theta) = \sum_{i=1}^{n} \log p(\underline{x}_i, \underline{d}_i; \theta),$$

where $\underline{x}_i = (x_{i,1}, ..., , x_{i,T_i})$, $\underline{d}_i = (d_{i,1}, ..., , d_{i,T_i})$ and $\log p(\underline{x}_i, \underline{d}_i; \theta)$ is the log-likelihood of individual i.

- **Numerical evaluation of model and MLE:** Requires (approximate) computation of value function and likelihood.

Rust's Assumptions

Assumption (CI)

State variables, $s_t = (x_t, \varepsilon_t)$ obeys the *Conditional Independence* (CI) assumption, i.e. the probability density of the controlled Markov process factors as

$$p(x_{t+1}, \varepsilon_{t+1} | x_t, \varepsilon_t, d, \theta_x, \theta_\varepsilon) = q(\varepsilon_{t+1} | x_{t+1}, \theta_\varepsilon) p(x_{t+1} | x_t, i, \theta_x)$$

Assumption (AS (Additive separability))

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

Assumption (XV)

The unobserved state variables, ε_t are assumed to be multivariate iid. *extreme value distributed*

The Dynamic Programming Problem

- Additivity: Simplified dynamic programming problem

$$V_\theta(x_t, \varepsilon_t) = \max_{d \in D(x_t)} [u(x_t, d, \theta_1) + \varepsilon_t(d) + \beta E V_\theta(x_{t+1}, \varepsilon_{t+1} | x_t, \varepsilon_t, d)]$$

- Under (CI) and (XV) we can integrate out the unobserved state variables

$$EV_\theta(x, d) = \Gamma_\theta(EV_\theta)(x, d)$$

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

- CI significantly reduces the dimension of the state space
- XV allows us to integrate out the unobserved state variables, ε_t, and thereby circumvent the cost of multiple integrations over unobserved state variables.
- Γ_θ is a *contraction mapping* with unique fixed point EV_θ, i.e.

$$\|\Gamma(EV) - \Gamma(W)\| \leq \beta \|EV - W\|$$

Likelihood Function

Likelihood

- Under assumption (CI) the log-likelihood function contribution ℓ^f has the particular simple form

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

where $P(d_{i,t}|x_{i,t},\theta)$ is the choice probability given the observable state variable, $x_{i,t}$.

Two step estimator

1. Estimate $p(x_{i,t}|x_{i,t-1},d_{i,t-1},\theta_x)$ non-parametrically or specify parametric model and estimate using e.g. NLS or MLE (i.e. by maximizing $\ell^2(\theta_x)$)
2. Maximize partial log likelihood $\sum_i^{T_i} \ell_i^1(\theta_u,\hat{\theta}_x)$ wrt. θ_u, where $\ell_i^1(\theta_u,\hat{\theta}_x) = \sum_{t=2}^{T} log\left(P(d_{i,t}|x_{i,t};\theta_u,\hat{\theta}_x)\right)$

Choice Probabilities

- Under the extreme value (XV) assumption choice probabilities are multinomial logistic

$$P(d|x,\theta) = \frac{\exp\{u(x,d,\theta_u) + \beta EV_\theta(x,d)\}}{\sum_{j\in D(y)}\{u(x,j,\theta_1) + \beta EV_\theta(x,j)\}} \tag{1}$$

- The expected value function is given by the unique fixed point to the contraction mapping Γ_θ, defined by

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

Engine Replacement Model

- Choice set: binary state dependent choice set, $\mathcal{D}(x_t) = \{0, 1\}$, where 1 denotes the replacement decision.
- Utility function:

$$u(x_t, d, \theta_u) + \varepsilon_t(d) = \begin{cases} -RC - c(0, \theta_u) + \varepsilon_t(1) & \text{if } d = 1 \\ -c(x_t, \theta_u) + \varepsilon_t(0) & \text{if } d = 0 \end{cases} \tag{2}$$

- State variables process x_t (mileage since last replacement)

$$p(x_{t+1}|x_t, d_t, \theta_x) = \begin{cases} g(x_{t+1} - 0, \theta_x) & \text{if } d = 1 \\ g(x_{t+1} - x_t, \theta_x) & \text{if } d = 0 \end{cases} \tag{3}$$

- If engine is replaced, state of bus regenerates to $x = 0$.
- In Rust's terminology, x_t follows a *regenerative random walk*.

Parmeters

Rust (Econometrica, 1987)

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		

Death to NFXP?

Judd and Su (Econometrica, 2012)

TABLE II

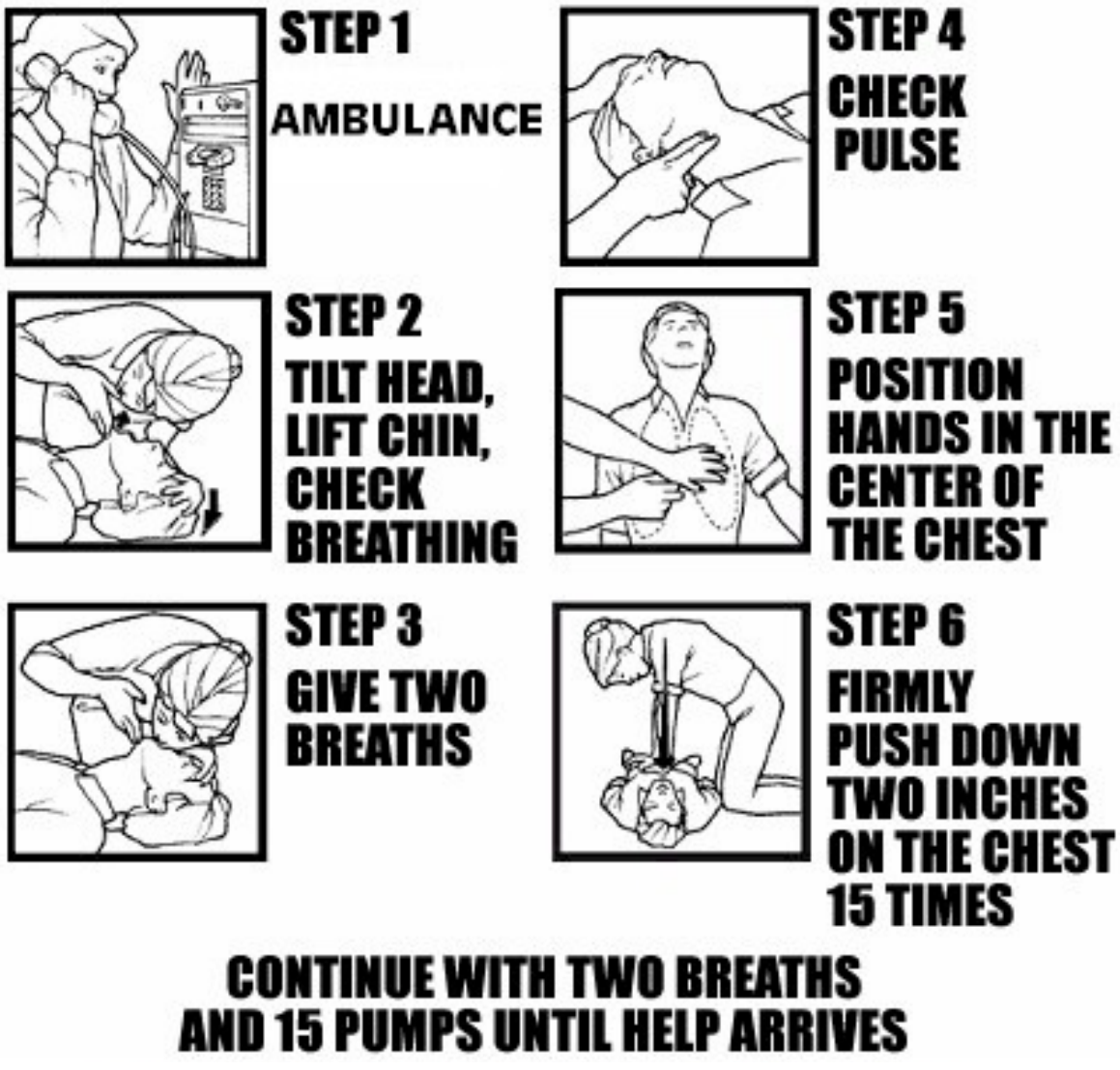
NUMERICAL PERFORMANCE OF NFXP AND MPEC IN THE MONTE CARLO EXPERIMENTS[a]

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.

Monte Carlo study demonstrates the uses of parametric bootstrap to compute

How to do CPR?


STEP 1
AMBULANCE
STEP 2
TILT HEAD, LIFT CHIN, CHECK BREATHING
STEP 3
GIVE TWO BREATHS
STEP 4
CHECK PULSE
STEP 5
POSITION HANDS IN THE CENTER OF THE CHEST
STEP 6
FIRMLY PUSH DOWN TWO INCHES ON THE CHEST 15 TIMES
CONTINUE WITH TWO BREATHS AND 15 PUMPS UNTIL HELP ARRIVES

NFXP survival kit

Step 1: Read NFXP manual and print out NFXP pocket guide
Step 2: Solve for fixed point using Newton Iterations
Step 3: Provide analytical gradients of Bellman operator
Step 4: Provide analytical gradients of likelihood
Step 5: Use BHHH (outer product of gradients as hessian approx.)
Step 6: Use sieves to approximate expected value functions
(e.g. Chebyshev polynomials)

If NFXP heartbeat is still weak:
Read NFXP pocket guide and Judd's book again until help arrives!

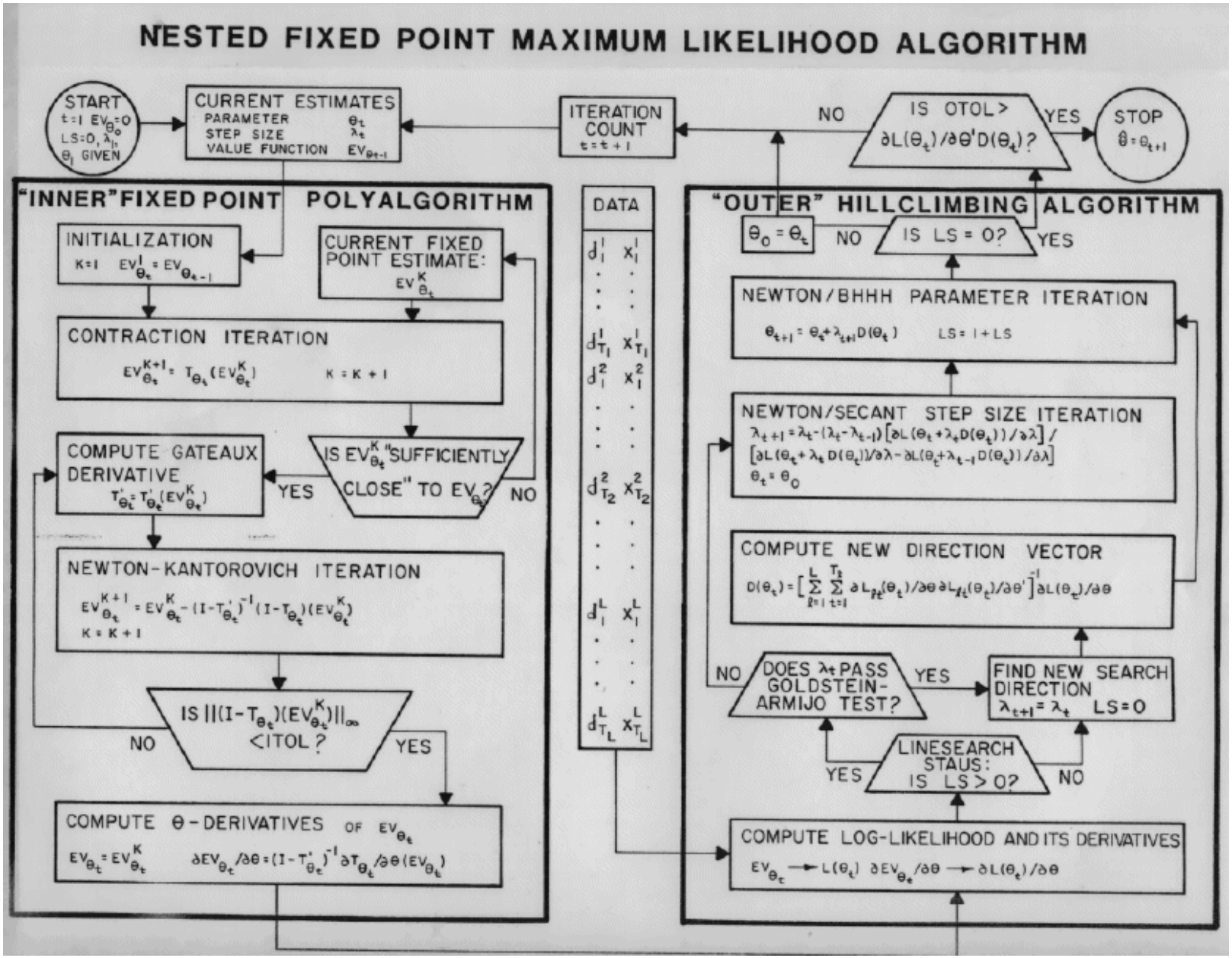
STEP 1: NFXP documentation

Main references

Rust (1987): "Optimal Replacement of GMC Bus Engines: An Empirical Model of Harold Zurcher" *Econometrica* 55-5 999-1033.

Rust (2000): "Nested Fixed Point Algorithm Documentation Manual: Version 6"
`http://gemini.econ.umd.edu/jrust/nfxp.html`

NFXP pocket guide



STEP 2: Newton-Kantorovich Iterations

- Problem: Find fixed point of the contraction mapping

$$EV = \Gamma(EV)$$

- Error bound on successive contraction iterations:
  $||EV_{k+1} - EV|| \leq \beta ||EV_k - EV||$
  linear convergence $\rightarrow$ slow when β close to 1
- Newton-Kantorovich:
  Solve $[I - \Gamma](EV_\theta) = 0$ using Newtons method
  $||EV_{k+1} - EV|| \leq A ||EV_k - EV||^2$
  quadratic convergence around fixed point, EV

STEP 2: Newton-Kantorovich Iterations

Newton-Kantorovich iteration:

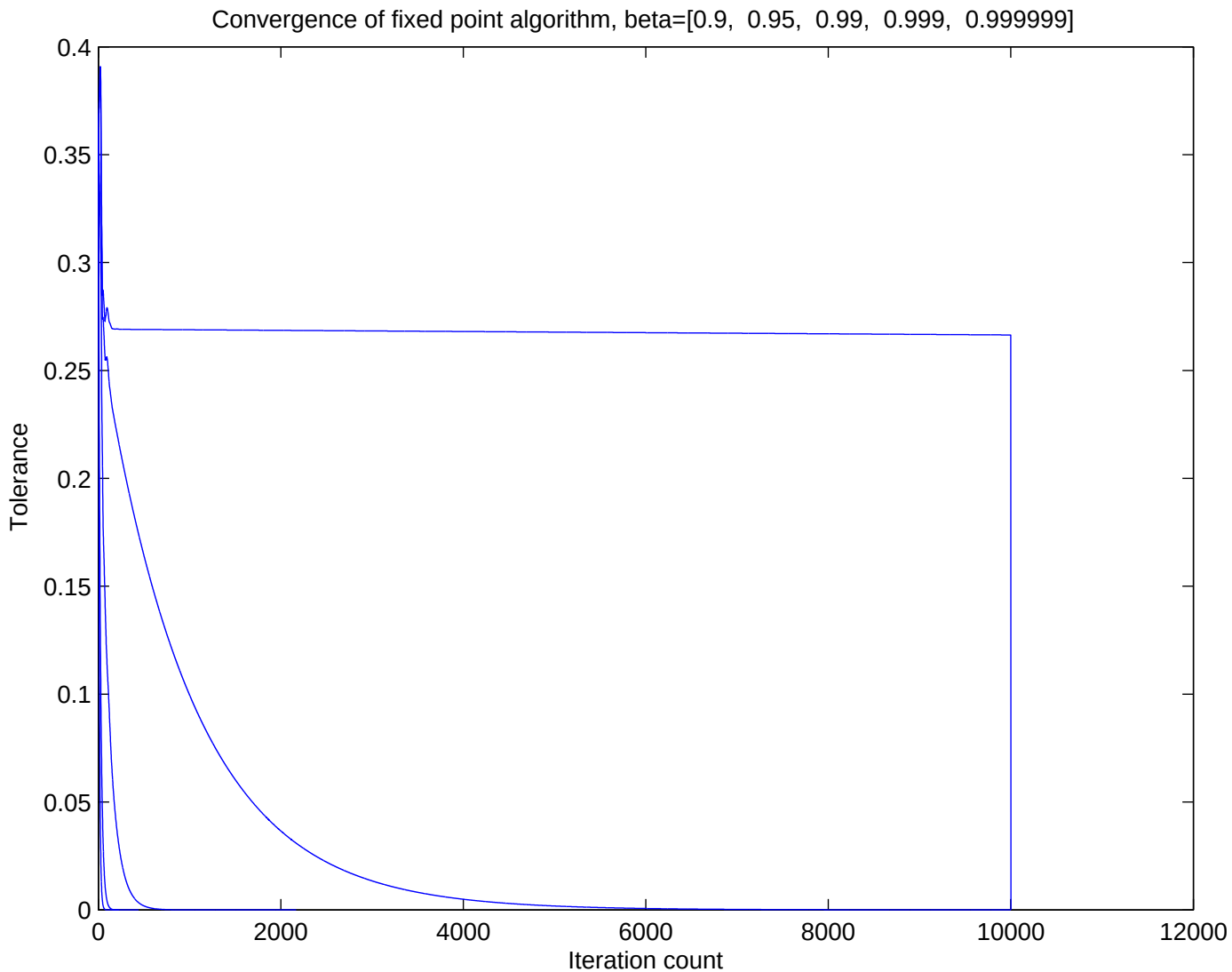
$$EV_{k+1} \quad = \quad EV_k - (I - \Gamma')^{-1}(I - \Gamma)(EV_k)$$

where I is the identity operator on B, and 0 is the zero element of B (i.e. the zero function). The nonlinear operator $I - \Gamma$ has a Fréchet derivative $I - \Gamma'$ which is a bounded linear operator on B with a bounded inverse.

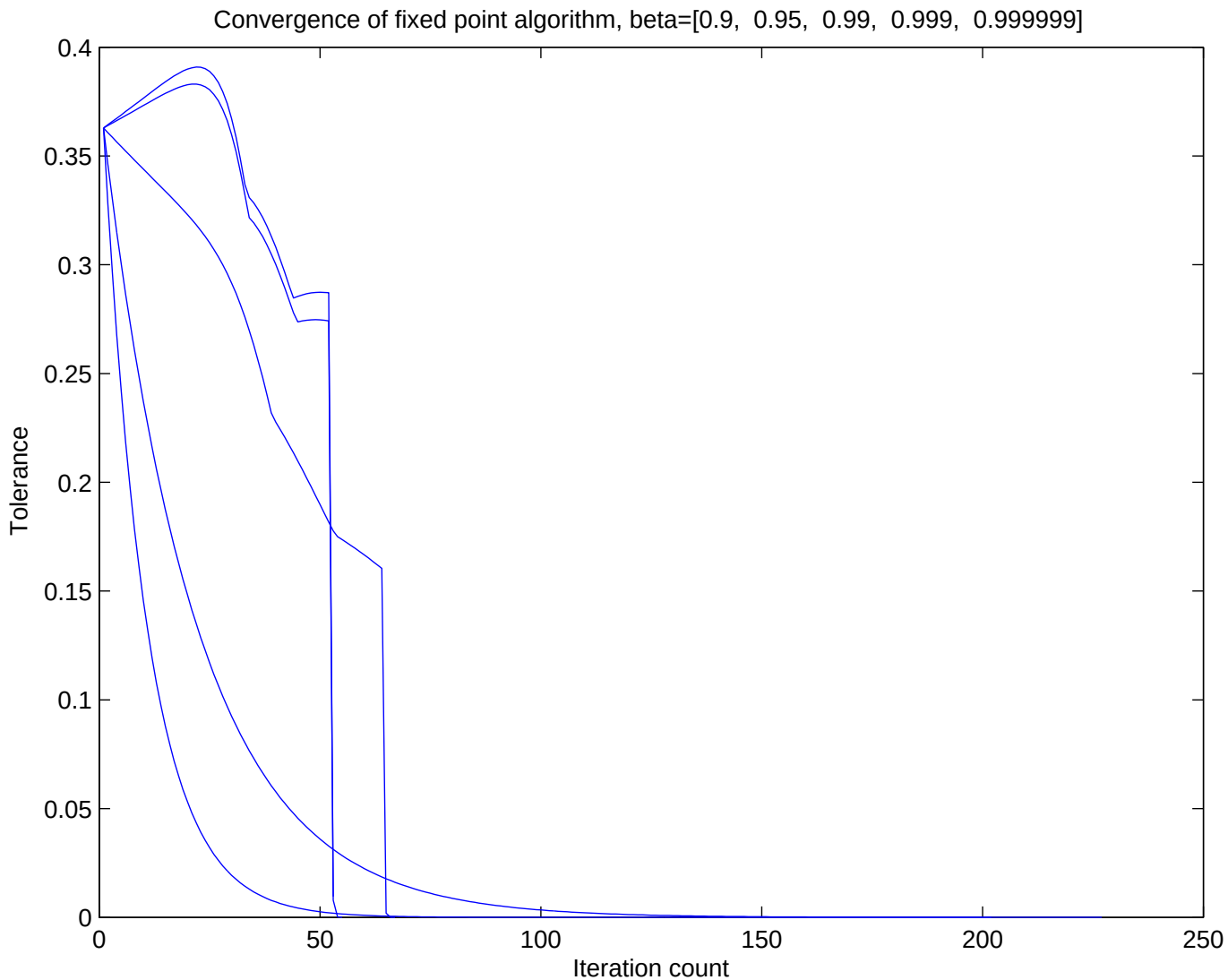
The Fixed Point (poly) Algorithm

1. Successive contraction iterations
   (until EV is in domain of attraction)
2. Newton-Kantorovich (until convergence)

STEP 2: Newton-Kantorovich Iterations



STEP 2: Newton-Kantorovich Iterations



STEP 2: Newton-Kantoriwich Iterations, $\beta = 0.99999$

Successive Approximations, VERY Slow

```
Begin contraction iterations
   j            tol           tol(j)/tol(j-1)
   2          0.32792388         0.32792388
   4          0.26780082         0.81665544
   6          0.21871383         0.81670338
   :                :                    :
 9988         0.26645258         0.99999800
 9990         0.26645205         0.99999800
 9992         0.26645151         0.99999800
 9994         0.26645098         0.99999800
 9996         0.26645045         0.99999800
 9998         0.26644992         0.99999800
 10000        0.26644938         0.99999800

Begin Newton-Kantorovich iterations at time: 2.77421
  nwt            tol
   0       1.74622983e-10

Time to convergence is 2.77585645 seconds
```

STEP 2: Newton-Kantoriwich Iterations, $\beta = 0.95$

Quadratic convergence!

```
Begin contraction iterations
   j              tol          tol(j)/tol(j-1)
   2          0.34614187          0.34614187
   4          0.31495936          0.90991407
   6          0.28660077          0.90996111

Begin Newton-Kantorovich iterations at time: 0.00178
  nwt             tol
   0       2.80699221e-03
   2       1.08179346e-05
   4       2.40107489e-10
   6       5.32907052e-15

Time to convergence is 0.00724992 seconds
```

STEP 3: Fréchet derivative of Bellman operator

Fréchet derivative

- $\hat{\Gamma}'$: $J \cdot m \times J \cdot m$ matrix of partial derivatives
- l, k'th block equals a $m \times m$ matrix of derivatives of contraction mapping

$$\hat{\Gamma}'_{l,k} = \frac{\partial \tilde{\Gamma}\left(\widehat{EV}\right)(x,l)}{\partial \widehat{EV}(x,k)'} = \beta \frac{1}{R} \sum_{r=1}^{R} \hat{P}\left(k|x'_r\right) T\left(x'_r\right) P_T$$

where $\hat{P}\left(k|x'_r\right)$ is the approximated choice probability conditional on a draw x'_r from $f\left(x'|x,l\right)$ and based on the value $\widehat{EV}$ evaluated at the Chebyshev nodes, x.

STEP 4: Provide analytical gradients of likelihood

Analytical derivatives using

1. Chain rule
2. Implicit function theorem $\rightarrow$ derivative of EV wrt. θ

STEP 4: Analytical gradients of likelihood

Log likelihood (discrete decision part)

$$\ell_i^1(\theta_u, \hat{\theta}_x) = \sum_{t=2}^{T} log P(d_{i,t}|x_{i,t}; \theta_u, \hat{\theta}_x)$$

Using Leibniz' rule for products of differentiable functions we obtain

$$\begin{aligned} \nabla_\theta \ell_i^1(\theta) &= \sum_{t=1}^{T_n} \frac{\nabla_\theta P(d_{i,t})}{P(d_{i,t})} \\ &= \sum_{t=1}^{T_n} \left(\nabla_\theta \hat{v}(d_{i,t}) - \sum_{j \in \mathcal{D}} P(j) \nabla_\theta \hat{v}(j) \right) \end{aligned}$$

where

- $P(j) \equiv \hat{P}(j|x_{i,t}, \theta)$
- $\hat{v}(d) \equiv [u(x_{i,t}, d; \theta) + \beta T(x_{i,t}) p_T \widehat{EV}(x, d; \theta)]/\lambda$

denote value functions and choice probabilities associated with alternative j evaluated at the data

STEP 4: Analytical gradients of likelihood

Gradient similar to the gradient for the conventional logit

- Only thing that differs is the derivative of the choice specific value function:

$$\nabla_\theta \hat{v}(d) = [\nabla_\theta u(x_{i,t}, d; \theta_u) + \beta T(x_{i,t}) p_T \nabla_\theta \widehat{EV}(x, d; \theta)]/\lambda$$

- Trivial to derive $\nabla_\theta u(x_{nt} : d_{nt}; \theta_u)$
  .... but contrary to a static logit, $\nabla_\theta \hat{v}(d)$ also includes $\nabla_\theta \widehat{EV}(x, d; \theta)$

How to compute derivative of $\widehat{EV} = \Gamma(\widehat{EV})$?

- By the implicit function theorem we obtain

$$\nabla_\theta \widehat{EV}(x, \mu, d; \theta) = [I - \Gamma'_\theta]^{-1} \partial\Gamma/\partial\theta'$$

  where $\partial\Gamma/\partial\theta_u = \nabla_{\theta_u} u(x_{i,t}, d_{i,t}; \theta_u)$

By-product of the Newton-Kantorovich algorithm: $[I - \Gamma'_\theta]^{-1}$

STEP 5: MATLAB implementation of score function

```
function score=gu(ev, d, p, u, du, TpT, dev, S, nP, mp);
    % STEP 1: Partial derivative of EV wrt. parameters
    pdu=nan(S.m*S.nD,nP);
    for iP=1:nP; % for each parameter in pnames.u
        for i=1:S.nD; % for each alternative
            p.s1=fxp.p(u.s1, TpT.s1(:,:,i), ev, mp);
            pdu_r=0;
            for j=1:S.nD;  % sum over alternatives
                pdu_r=pdu_r+p.s1(:,j).*du.s1(:,iP,j);
            end;
            % Average over the R random draws:
            pdu((i-1)*S.m+1:i*S.m, iP)=mean(reshape(pdu_r, S.m, S.R),2);
        end
    end

    % STEP 2: Derivative of EV wrt. parameters
    devdmp=(eye(S.nD*S.m)-dev)\pdu;  % (S.nD*S.m x nP matrix)

    % STEP 3: Derivative of log-likelihood wrt. parameters
    score=zeros(size(d,1), nP);  % Nobs by nP matrix of scores
    for iP=1:nP; % loop over parameters
        for i=1:S.nD; % sum over alternatives
            score(:,iP)=score(:,iP)+((d==i)-p.obs(:,i)).*(du.obs(:,iP,i)
            + mp.beta*TpT.obs*devdmp((i-1)*S.m+1:i*S.m, iP));
        end
    end
end  % End of function gu
```

STEP 5: Analytical gradients of likelihood

Complicated?

- NO: Derivative have a quite general form
- Usually we only need to specify $\nabla_\theta u(x_{nt} : d_{nt}; \theta_u)$ when model specification change

STEP 5: 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

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

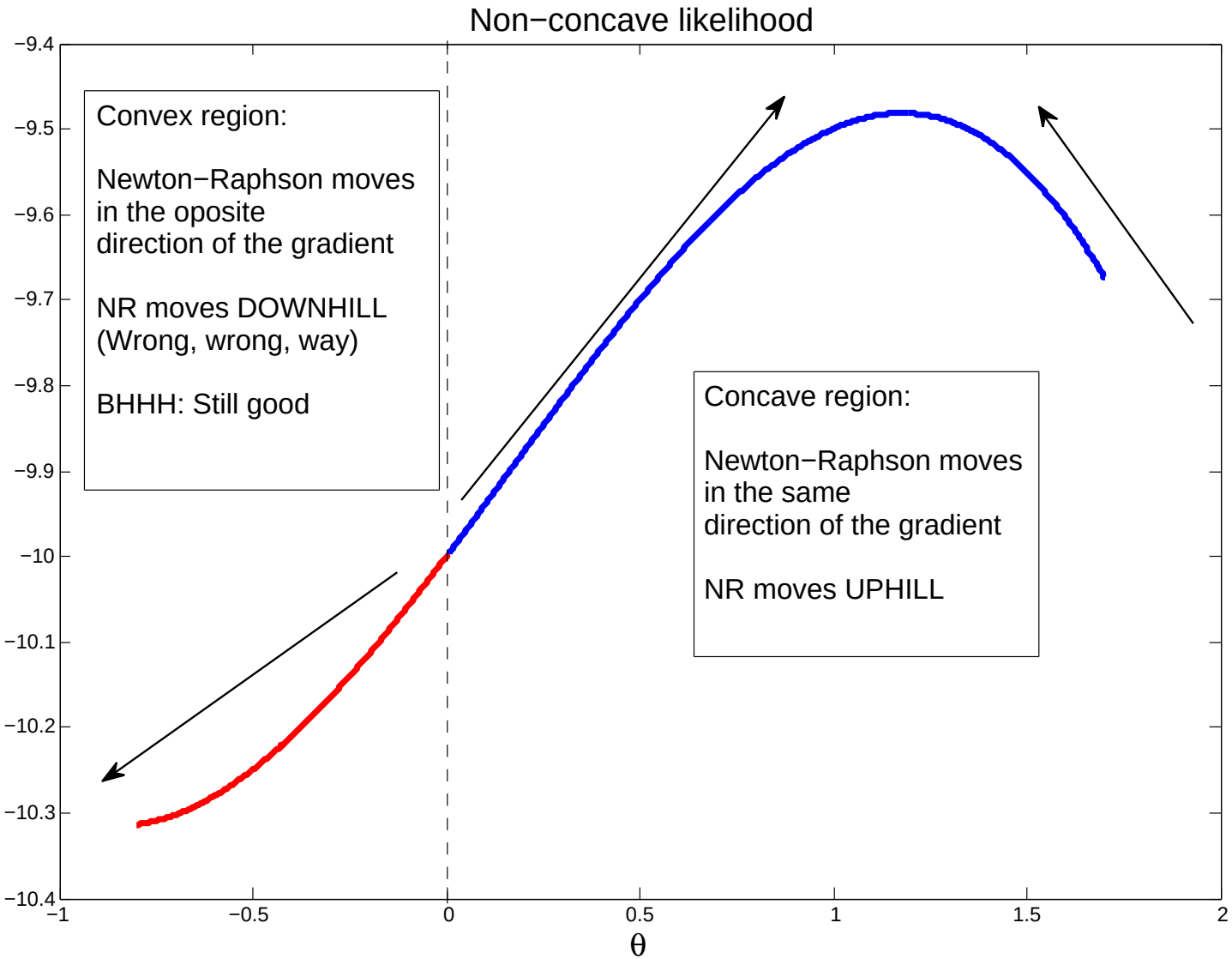
- Why is this valid? Information identity:

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

(only valid for MLE and CMLE)

STEP 5: BHHH

Some times linesearch may not help Newtons Method



STEP 5: 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 5: BHHH

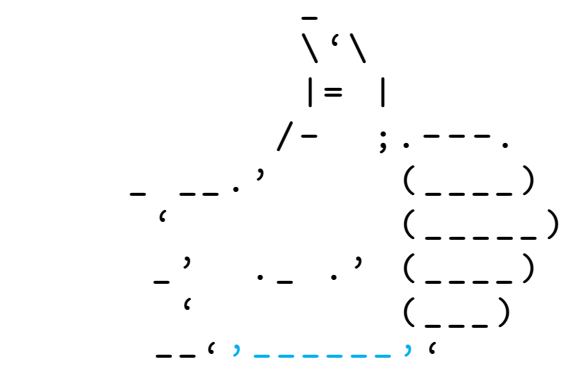
"The road ahead will be long. Our climb will be steep. We may not get there in one year or even in one term. But, America, I have never been more hopeful than I am tonight that we will get there. I promise you, we as a people will get there." (Barack Obama, Nov. 2008)

STEP 6: Approximate estimator

- I approximate expectation operator in Bellman equation by simulation (somewhat inefficient)
- I approximate expected value function with Chebyshev polynomials

Convergence!

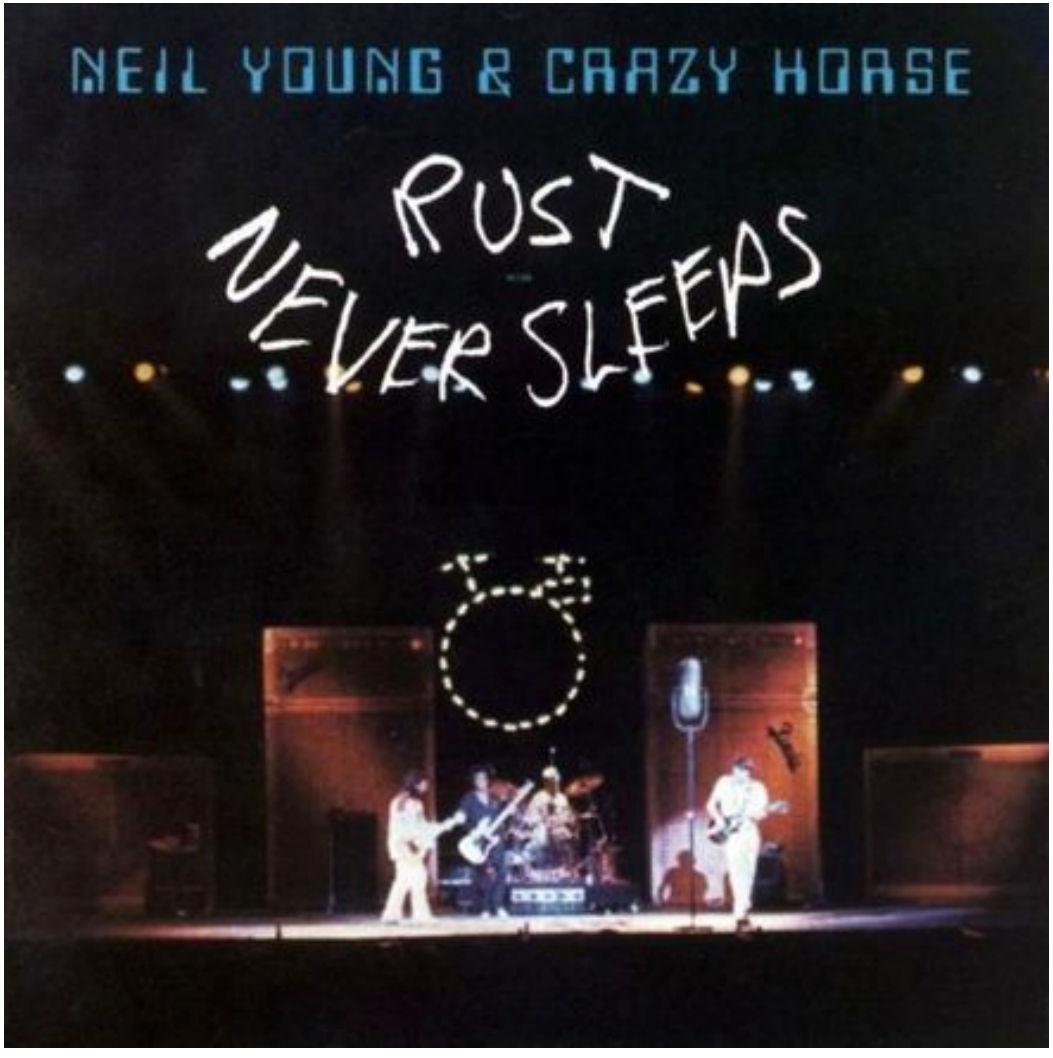
```
---------------------------------------------------------------------
***                    Convergence Achieved                       ***
---------------------------------------------------------------------
```



```
Structural parameters in Rust's Engine replacement model
Estimation Method: Partial MLE
Number of iterations: 11
grad*direc            0.00275
Log-likelihood     -131.46681
Step length           0.60000

    Param.         Estimates          s.e.              t-stat
---------------------------------------------------------------------
    RC              11.7231          2.4660             4.7538
    c                1.0999          0.3756             2.9280
---------------------------------------------------------------------
Time to convergence is   0 min and 0.63 seconds
```

Patient still allive



PART II

Maximum Likelihood Estimation of Discrete Markov Decision Models by Sieve Approximations

with Dennis Kristensen (UCL)

Approximation

- Most available solution algorithms and estimation procedures make use of numerical approximations in many dimensions:
  1. Value/Policy function
  2. Expectation operator in Bellman equation
  3. Integrals in choice probabilities and likelihood function
- Various approximations are employed such as
  1. Discretization (uniform grids, random grids, low discrepancy grids, etc.)
  2. Parametric approximations (polynomials, splines, wavelets, neural networks, etc.)
  3. Quadrature/Simulation (MCMC, importance sampling, particle filtering, etc.)
- **References:** Judd and Solnick (1994), Judd and Su (2012), Rust (1987, 1988, 1997), Keane and Wolpin (1994), Imai, Jain and Ching (2009), Norets (2009,2011). Aguirregabiria and Mira (2002), Hotz and Miller (1993) .

Approximation error?

What are the implications for statistical inference when approximating the value functions, Bellman operator, conditional choice probabilities and the likelihood function?

Outline part II

1. Estimation and Solution Method
   - Augmentation of model
   - Approximation of value function
   - Approximation of likelihood
2. Theory:
   - Error bounds on value function and MLE
3. Numerical performance
4. Conclusion

The General Problem

Bellman equation

$$V_\theta(z) = \max_{d \in \mathcal{D}(z)} \{u_{\theta_u}(z,d) + \beta \int V_\theta(z') p_{\theta_p}(z'|z,d) dz'\}$$

u_{θ_u} and p_{θ_p}: known up to a set of parameters, θ_u and θ_p

- **The agent's problem:** Maximize expected sum of current and future discounted utilities
  - d : Discrete control variable, $d \in \mathcal{D}(z) = \{1, 2, .., J\}$.
  - z : Current state, fully observed by agent
  - $z\prime$: Future state; possibly continuous and subject to uncertainty
- **The agents beliefs about z':**
  - Obeys a (controlled) Markov transition probability $p_{\theta_p}(z_{t+1}|z_t, d_t)$
- **Model solution, $v_\theta(z)$**
  - Find the fixed point for the Bellman equation
  - $V_\theta(z)$ can be a very high dimensional function

MLE of Markov Decision Models

- **Econometric problem:** Given observations of n individual agents over T time periods:

$$(d_{i,t}, x_{i,t}),\ t = 1, ..., T \text{ and } i = 1, ..., n,$$

we wish to estimate the underlying Markov decision model.

  - $d_{i,t}$: individual i's discrete choice at time t.
  - $x_{i,t}$: sub-component of individual i's state vector $z_{i,t}$.
  - $z_t = (x_t, \varepsilon_t)$ where ε_t is a set of unobservables.

- **MLE:**

$$\hat{\theta} = \arg\max_{\theta} \ell_n(\theta), \quad \ell_n(\theta) = \sum_{i=1}^{n} \log p(\underline{x}_i, \underline{d}_i; \theta),$$

where $\underline{x}_i = (x_{i,1}, ..., , x_{i,T_i})$, $\underline{d}_i = (d_{i,1}, ..., , d_{i,T_i})$ and $\log p(\underline{x}_i, \underline{d}_i; \theta)$ is the log-likelihood of individual i.

- **Numerical evaluation of model and MLE:** Requires (approximate) computation of value function and likelihood.

Augmentation of Model

To facilitate implementation, we augment the model.

- Let $\eta_t = (\eta_t(1), ..., \eta_t(J))$ be a extreme value shock which is i.i.d. over alternatives and time, and independent of $\{z_t\}$.
- Transition density in augmented model:

$$p_{\theta_p,\lambda}(z_{t+1}, \eta_{t+1}|z_t, \eta_t, d_t) = p_{\theta_p}(z_{t+1}|z_t, d_t) f_\lambda(\eta_{t+1}),$$

  where $f_\lambda(\eta) := f(\eta/\lambda)/\lambda$ with $f(\eta)$ being extreme value density and $\lambda > 0$ scale parameter.
- Value function in augmented model:

$$v_{\theta,\lambda}(z_t) = \max_{d_t \in \mathcal{D}} \{u_{\theta_u}(z_t) + \lambda\eta(d_t) + \beta V_{\theta,\lambda}(z_t, d_t)\},$$

  where $V_{\theta,\lambda}(z_t, d_t)$ is the expected value function,

$$V_{\theta,\lambda}(z_t, d_t) := \int_{\mathcal{Z}} \int_{\mathbb{R}^J} v_{\theta,\lambda}(z_{t+1}) p_{\theta_p}(z_{t+1}|z_t, d_t) f_\lambda(\eta_{t+1}) d\eta_{t+1} dz_{t+1}.$$

Augmentation of model with extreme value errors

Augmentation of Model

- The addition of η_t to the model works as a smoothing device. It facilitates computation of the (expected) value function and likelihood.
- A similar idea have been used in the estimation of static discrete choice models; see e.g. McFadden (1989) and McFadden and Train (2003).
- One can think of the extreme value density $f_\lambda(\eta)$ as a kernel smoother with $\lambda > 0$ playing the role of a bandwidth.
- We fix $\lambda = \lambda_n$ at a (small) value for a given sample size n. As $\lambda \to 0$ as $\to \infty$, augmented model and MLE is asymptotically equivalent to the original ones.

Sieve Approximation of Value Function

- **Bellman operator:** The augmented model falls within the framework of Rust (1988). Thus, the expected value function solves a fixed point problem:

$$V_{\theta,\lambda}(z,d) = \Gamma_{\theta,\lambda}(V_{\theta,\lambda})(z,d), \tag{4}$$

  where $\Gamma_{\theta,\lambda} : \mathcal{V} \mapsto \mathcal{V}$ is the so-called Bellman operator.
- Since $f_\lambda(\eta)$ is an extreme value density $\Gamma_{\theta,\lambda}$ can be written as:

$$\begin{aligned}\Gamma_{\theta,\lambda}(V)(z,d) \\ = \int_{\mathcal{Z}} \log\left[\sum_{j\in\mathcal{D}} \exp\left[\frac{u_{\theta_u}(z',j)+\beta V(z',j)}{\lambda}\right]\right] p_{\theta_p}(z'|z,d)dz'.\end{aligned}$$

- The fixed-point problem is an infinite-dimensional problem and so in general numerically infeasible.

Sieve Approximation of Value Function

- Suppose that the expected value function $V_{\theta,\lambda}(z,d)$ can be approximated by by a set of basis functions,

$$V_{\theta,\lambda}(z,d) \simeq B_K(z)'\gamma(d), \quad \gamma(d) \in \mathbb{R}^K.$$

- Here, $B_K(z) = (b_1(z), .., b_K(z))$ is a set of K basis functions chosen by the researcher and $\gamma(d)$ is set a coefficients that uniquely characterizes the expected value function
- For example, we can choose $B_K(z)$ as polynomials, such that $B_K(z) = (1,\ z, z^2, .., z^{K-1})$ or Chebyshev polynomials
- As K increases the approximation gets more flexible.

Sieve Approximation of Value Function

Approximate $V_{\theta,\lambda}$ by combining simulations and sieve methods.

Simulate Bellman operator: With $Z_\theta^{(r)}(z,d) \sim p_{\theta_p}(z'|z,d)$,

$$\hat{\Gamma}_{\theta,\lambda}(V)(z,d) = \frac{1}{R_1}\sum_{r=1}^{R_1} \log\left[\sum_{j\in\mathcal{D}} \exp\left[\frac{u_{\theta_u}(Z_\theta^{(r)}(z,d),j) + \beta V(Z_\theta^{(r)}(z,d),j)}{\lambda}\right]\right]$$

Approximate $V_{\theta,\lambda}$ by $\hat{V}_{\theta,\lambda}(z,d) = B_K(z)'\hat{\gamma}_{\theta,\lambda}(d)$ where $\hat{\gamma}_{\theta,\lambda}$ solves the approximate fixed-point problem:

$$\hat{\gamma}_{\theta,\lambda} = \arg\min_{\gamma\in\mathbb{R}^{JK}} \sum_{d=1}^{J}\sum_{r=1}^{R_2} [\hat{\Gamma}_{\theta,\lambda}(B'\gamma)(\tilde{Z}^{(r)},d) - B(\tilde{Z}^{(r)})'\gamma(d)]^2,$$

for some (random) grid $\tilde{Z}^{(r)}$, $r = 1,...,R_2$.

Sieve Approximation of Value Function

- The above least-squares problem can be solved iteratively:

$$\begin{aligned}\hat{\gamma}_{\theta,\lambda}^{[i]}(d) &= [\sum_{r=1}^{R_2} B_K(\tilde{Z}^{(r)})B_K(\tilde{Z}^{(r)})']^{-1} \\ &\quad \times \sum_{r=1}^{R_2} B_K(\tilde{Z}^{(r)})\hat{\Gamma}_{\theta,\lambda}(B'\hat{\gamma}_{\theta,\lambda}^{[i-1]})(\tilde{Z}^{(r)}, d)\end{aligned}$$

- This is a standard series regression estimator as used in nonparametric econometrics (Newey, 1997).
- Can be combined with Newton-Kantorivich iterations as mentioned above.

Sieve Approximation of Value Function

In matrix form Simulated conditional choice probabilities, $\hat{P}(d|x)$, are given by

$$\hat{P}(d|x) = \frac{1}{R}\sum_{r=1}^{R} I\left\{d = \arg\max_{j\in D(s)}\left[u(x,\varepsilon_r(j),j) + \beta\widehat{V}(x,j)\right]\right\}$$

$$\hat{\Gamma}\left(\widehat{V}\right)(x,d) = \frac{1}{R}\sum_{r=1}^{R}\max_{j\in D(x,\varepsilon)}\left[u(x'_r,\varepsilon'_r(j),j) + \beta B\left(x'_r\right)P_B\widehat{V}(x,j)\right]$$

where ε'_r and x'^r_j are draws from the distribution for the state variable processes $g(\varepsilon'|x')$ and $f(x'|x,d)$, respectively.

B is a matrix of basis functions evaluated at the nodes and $p_B = (B'B)^{-1}B'$

In the augmented model, Emax operator is smooth (the log sum) and choice probabilities are multinomial logistic

Approximation of Likelihood Function

- Conditional choice probability:

$$P_{\theta,\lambda}(d|x,\varepsilon) = \frac{\exp[\{u_{\theta_u}(x,\varepsilon,d) + \beta V_{\theta,\lambda}(x,\varepsilon,d)\}/\lambda]}{\sum_{j\in\mathcal{D}} \exp[\{u_{\theta_u}(x,\varepsilon,j) + \beta V_{\theta,\lambda}(x,\varepsilon,j)\}/\lambda]}.$$

- Thus, the likelihood of observables is given as

$$p_\lambda(\underline{x},\underline{d};\theta) = \int_{\mathcal{E}^T} p_\lambda(\underline{x},\underline{d},\underline{\varepsilon};\theta) d\underline{\varepsilon}_j,$$

where

$$\begin{aligned} & p_\lambda(\underline{x},\underline{d},\underline{\varepsilon};\theta) \\ = & \prod_{t=1}^{T} P_{\theta,\lambda}(d_t|x_t,\varepsilon_t) \times p_{\theta_p}(x_t,\varepsilon_t|x_{t-1},\varepsilon_{t-1},d_{t-1}). \end{aligned}$$

Approximation of Likelihood Function

- Given the sieve approximator $\hat{V}_{\theta,\lambda}$, draw $\underline{\varepsilon}^{(s)} \sim g(\underline{\varepsilon})$ from some density $g(\underline{\varepsilon})$ with support $\mathcal{E}^T$ and compute

$$\hat{p}_\lambda(\underline{x}, \underline{d}; \theta) = \frac{1}{S} \sum_{s=1}^{S} \frac{\hat{p}_\lambda(\underline{x}, \underline{d}, \underline{\varepsilon}^{(s)}; \theta)}{g(\underline{\varepsilon}^{(s)})},$$

  where $\hat{p}_\lambda(\underline{x}, \underline{d}, \underline{\varepsilon}^{(s)}; \theta)$ is evaluated using $\hat{V}_{\theta,\lambda}$.
- Computation can be sped up using more advanced simulators such as MCMC (Norets, 2009) or particle filtering (Brownlees, Kristensen and Shin, 2011).

Theory: $\hat{\Gamma}$ is a contraction mapping

Fixed-point algorithm converges towards the solution to $\min_{V \in \mathcal{V}_K} \left\| \hat{\Gamma}_{\theta,\lambda}(V) - V \right\|_{R_2,2}^2$ for any fixed K and R_2 when R_1 has been chosen large enough

- Formally, if the eigenvalues of $\mathbf{B}_K' \mathbf{B}_K \in \mathbb{R}^{K \times K}$ are bounded away from zero where $\mathbf{B}_K = \left[B_K\left(Z^{(1)}\right), ..., B_K\left(Z^{(R_2)}\right) \right]' \in \mathbb{R}^{K \times R_2}$.
- To see this, introduce projection operator $\hat{\pi}_K$ associated with the sieve space $\mathcal{V}_K$ and the grid $\left\{ Z^{(r)} : r = 1, ..., R_2 \right\}$,

$$\hat{\pi}_K(V)(z,d) = B_K(z)' \left[\mathbf{B}_K' \mathbf{B}_K \right]^{-1} \mathbf{B}_K' \mathbf{V}(d) \tag{5}$$

where $\mathbf{V}(d) = \left[V\left(Z^{(1)}, d\right), ..., V\left(Z^{(R_2)}, d\right) \right]' \in \mathbb{R}^{R_2}$.

Theory: $\hat{\Gamma}$ is a contraction mapping

- We may then reformulate the nonlinear optimization as $\min_{V \in \mathcal{V}} \left\| \hat{\pi}_K \circ \hat{\Gamma}_{\theta,\lambda}(V) - \hat{\pi}_K(V) \right\|_{R_2,2}^2$
- The iterative procedure can be rewritten as

$$\hat{V}_{\theta,\lambda}^{(i)} = \hat{\pi}_K \circ \hat{\Gamma}_{\theta,\lambda}(\hat{V}_{\theta,\lambda}^{(i-1)}), \quad i = 1, 2, ...$$

- With eigenvalues of $\mathbf{B}_K'\mathbf{B}_K$ being bounded away from zero, the projection operator is non-expansive w.r.t. the L_2-norm defined by the grid, $\|\cdot\|_{R_2,2}$:

$$\begin{aligned}
& ||\hat{\pi}(V)(d) - \hat{\pi}_K(W)(d)||_{R_2,2} \\
= \; & [\mathbf{V} - \mathbf{W}](d)' \mathbf{B}_K \left[\mathbf{B}_K'\mathbf{B}_K\right]^{-1} \mathbf{B}_K'\mathbf{B}_K \left[\mathbf{B}_K'\mathbf{B}_K\right]^{-1} \mathbf{B}_K' [\mathbf{V} - \mathbf{W}](d) / R \\
= \; & [\mathbf{V} - \mathbf{W}](d)' \left[\mathbf{B}_K \left[\mathbf{B}_K'\mathbf{B}_K\right]^{-1} \mathbf{B}_K'\right] [\mathbf{V} - \mathbf{W}](d) / R_2 \\
\leq \; & [\mathbf{V} - \mathbf{W}](d)' [\mathbf{V} - \mathbf{W}](d) / R_2 \\
= \; & ||V(d) - W(d)||_{R_2,2}
\end{aligned}$$

- Since $\hat{\Gamma}$ is a contraction mapping with probability 1 for R_1 chosen large enough, $\hat{\pi}_K \circ \hat{\Gamma}$ is also a contraction mapping w.r.t. $\|\cdot\|_{R_2,2}$

Theory: Value Function Approximation

- Suppose $\exists \gamma_{\theta,\lambda} : ||B'_K \gamma_{\theta,\lambda} - V_{\theta,\lambda}||_\infty = O(K^{-\alpha})$ for some $\alpha > 0$.
- For example: If $z \mapsto V_{\theta,\lambda}(z,d)$ is s times differential and $B_K(z)$ is chosen as polynomials, then $\alpha = s/\dim(z)$.
- Also define $\zeta(K) := ||B_K||_\infty$. For example, with polynomials, $\zeta(K) = O(K^{1+2\dim(z)})$.

Theorem (1)

Under regularity conditions,

$$\begin{aligned} ||\hat{V}_{\theta,\lambda} - V_{\theta,\lambda}||_\infty &= O_P(\zeta(K)K^{-\alpha}) + O_P(\zeta(K)K^{1/2}/\sqrt{R_1 R_2}) \\ &= \textit{approximation bias} + \textit{simulation noise}, \end{aligned}$$

where $R_1 = \#$simulations and $R_2 = \#$random grid points.

Theory: Value Function Approximation

$$||\hat{V}_{\theta,\lambda} - V_{\theta,\lambda}||_\infty = O_P(\zeta(K)K^{-\alpha}) + O_P(\zeta(K)K^{1/2}/\sqrt{R_1 R_2}),$$

where $R_1 = \#$simulations and $R_2 = \#$random grid points.

- For fixed R_1, this rate is identical to the one for nonparametric series regression estimators (Newey, 1997, Theorem 1).
- Bias: $\alpha = s/\dim(z)$. Thus, the smoother $V_{\theta,\lambda}(z,d)$ is the better is the rate. On the other hand, the larger $\dim(z)$ is the larger K has to be chosen.
- Parametric rate is attainable: Choosing $K = R_2^{1/\{2(1+2\dim(z)-\alpha)\}}$ and $R_1 = K\zeta^2(K)$,

$$||\hat{V}_{\theta,\lambda} - V_{\theta,\lambda}||_\infty = O_P(1/\sqrt{R_2}).$$

Theory: Simulated MLE

Theorem (3)

Under regularity conditions, the simulated MLE, $\hat{\theta}_{\text{approx}}$, satisfies:

$$\begin{aligned} & \|\hat{\theta}_{\text{approx}} - \hat{\theta}\| \\ = \; & O_P(\zeta(K)K^{-\alpha}) + \{O_P(\zeta(K)K^{1/2}/\sqrt{R_1 R_2}) + O_P(S^{-1/2})\} \\ = \; & \textit{bias} + \textit{variance}, \end{aligned}$$

where $\hat{\theta}$ is the exact MLE, $R_1 = \#$simulations used for Bellman operator, $R_2 = \#$random grid points, and $S = \#$simulations used to compute likelihood.

- The approximate MLE inherits the error of the value function.
- Conjecture: This rate is not sharp.
- Work in progress: Sharpening rate and obtain expression of leading bias and variance terms. This will allow for adjustment of estimator and/or standard errors to take into account approximation errors.

More of Harold Zurcher

Decisions: $d \in \{1, 2\}$ with $d = 2$ if engine replaced, and $d = 1$ otherwise, and x being elapsed millage.

Utiliy

$$u(x, d) = -\mathbb{I}\{d = 1\}C(x) - \mathbb{I}\{d = 2\}RC.$$

- RC: engine replacement cost
- $C(x) = c\sqrt{x}$ maintenance/operating costs,

States:

- x_t follows regenerating random walk
- ε is extreme value.

Sieve Approximation: Chebyshev polynomials with $K = m$ nodes are used to approximate value function.

- Where approximate "exact" solution and MLE with $R = 5,000$ simulations and $m = 50$ (49 degree Chebyshev polynomial).
- Compare this with approximate MLE with smaller R and m.

Numerical Performance - Choice Probabilities

APPROXIMATION ERROR IN CONDITIONAL CHOICE PROBABILITIES
ALTERNATIVE APPROXIMATIONS

Number of nodes, m	Based on ML estimates for alternative approximations				
	10	20	50	100	5000
2	6.31	6.31	6.31	6.31	6.31
4	1.26	1.16	1.20	1.20	1.17
6	0.16	0.10	0.09	0.09	0.10
8	0.17	0.13	0.11	0.11	0.13
10	0.07	0.03	0.01	0.01	0.02
50	0.05	0.00	0.03	0.02	0.00
	Based on ML estimates for R=5000 and m=50				
	10	20	50	100	5000
2	3.87	3.87	3.87	3.87	3.87
4	1.19	0.96	0.95	0.96	0.95
6	0.34	0.10	0.16	0.16	0.11
8	0.22	0.04	0.07	0.07	0.03
10	0.23	0.01	0.06	0.07	0.02
50	0.23	0.01	0.05	0.05	0.00

Note: All figures are measured in percentage points. Approximation error is measured as the maximum absolute deviation between the approximated choice probability and the "exact" choice probability. The "exact" solution was based on R=5000 and m=50. In the top panel, the choice

Numerical Performance - Bias in MLE

BIAS IN ML ESTIMATES FOR ALTERNATIVE APPROXIMATIONS
RUST'S ENGINE REPLACEMENT MODEL
COST FUNCTION: $C(x) = c\sqrt{x}$

Number of nodes, m	R	Engine replacement costs, RC 10	20	5000	Cost function parameter, c 10	20	5000
2		-3.59 (0.58)	-3.59 (0.58)	-3.59 (0.58)	3.83 (8.06)	3.86 (8.07)	3.86 (8.07)
4		0.70 (1.73)	0.64 (1.72)	0.64 (1.72)	0.38 (3.84)	0.75 (3.95)	0.73 (3.94)
6		0.05 (1.60)	-0.02 (1.57)	-0.02 (1.57)	-0.19 (3.86)	-0.04 (3.90)	-0.05 (3.90)
8		0.09 (1.60)	0.03 (1.58)	0.03 (1.58)	-0.11 (3.84)	0.05 (3.90)	0.05 (3.89)
10		0.06 (1.59)	0.00 (1.57)	0.00 (1.57)	-0.16 (3.83)	0.00 (3.88)	-0.01 (3.87)
50		0.06 (1.59)	0.00 (1.56)	0.00 (1.56)	-0.16 (3.81)	0.01 (3.85)	0.00 (3.85)
Paramter estimate ("exact" solution)				11.14			16.49

Numerical Performance - Likelihood Function

TABLE 3
LIKELIHOOD RATIO
ALTERNATIVE APPROXIMATIONS AGAINST "EXACT" SOLUTION

Number of nodes, *m*	Based on ML estimates for alternative approximations				
	10	20	50	100	5000
2	-7.10	-7.10	-7.10	-7.10	-7.10
4	1.04	0.99	1.01	1.01	0.99
6	-0.14	-0.18	-0.17	-0.17	-0.18
8	0.09	0.05	0.06	0.06	0.05
10	0.03	-0.01	0.00	0.00	-0.01
50	0.04	0.00	0.01	0.01	0.00
Likelihood ("exact" solution)					-298.57

Note: The table presents the log-likelihood value under alternative levels of approximation, differenced against the "exact" solution. The "exact" solution were based on, *R=5000*, and n=50.

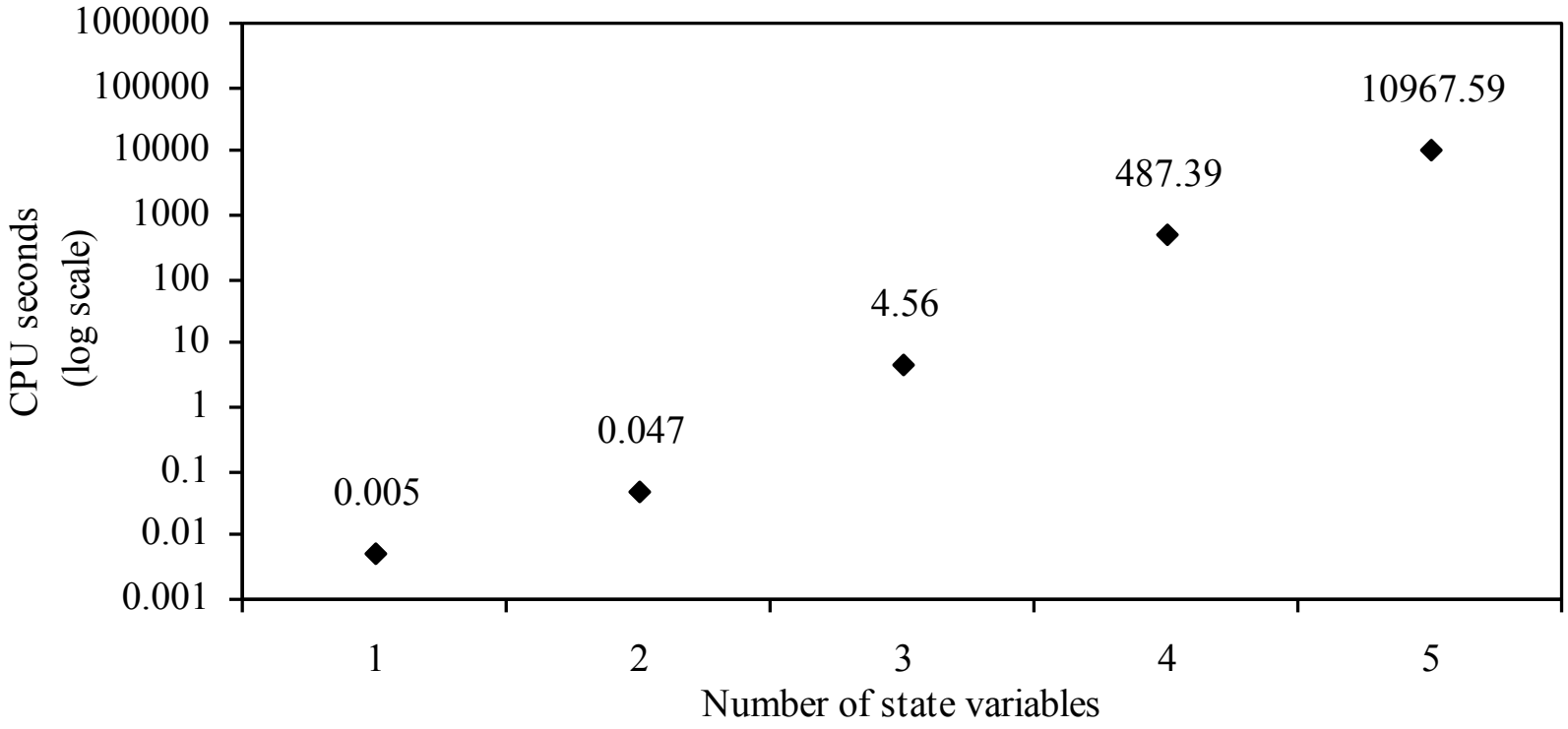
Consequences of discretizing the data

TABLE 4
BIAS AND APPROXIMATION ERROR DUE TO DISCRETIZATION
RUST'S ENGINE REPLACEMENT MODEL
COST FUNCTION: $C(x) = c\sqrt{x}$

Number of grid points	Bias (Standard error)			Mean approx. error (Standard deviation of approx. error)		
	c	RC	LR	x_t (m)	x_t (m)-x_{t-1} (m)	dx_t (m)
2	-10.55	-4.21	-29.85	25.55	-1.85	327.43
	(1.02)	(0.36)		(62.26)	(18.18)	(37.83)
4	-5.26	-1.99	-3.34	5.84	-1.01	162.05
	(2.57)	(1.14)		(32.30)	(15.94)	(18.66)
6	-4.68	-1.67	-4.38	2.89	-0.66	106.93
	(2.33)	(1.03)		(21.34)	(13.81)	(12.28)
8	-3.55	-1.08	-2.27	1.55	-0.53	79.37
	(2.93)	(1.33)		(16.04)	(12.16)	(9.11)
10	-2.46	-0.68	-0.32	0.94	-0.34	62.83
	(2.98)	(1.32)		(13.01)	(11.11)	(7.21)
25	-0.90	-0.11	-0.23	0.20	-0.12	23.14
	(3.57)	(1.54)		(5.26)	(6.75)	(2.80)
50	-0.18	0.06	0.17	0.07	-0.04	9.92
	(3.77)	(1.59)		(2.62)	(4.10)	(1.59)
75	0.10	0.10	0.35	0.02	-0.02	5.60
	(3.87)	(1.61)		(1.74)	(2.64)	(1.35)
90	0.06	0.07	0.29	0.04	-0.02	4.59
	(3.85)	(1.59)		(1.45)	(2.05)	(1.37)
100	0.00	0.00	0.03	0.03	-0.01	4.27
	(3.85)	(1.59)		(0.26)	(0.36)	(0.27)
	Parameter Estimates (Standard error)		Likelihood	Mean of variable (Standard deviation of variable)		
Continuous data	16.39	11.14	-298.56	115.91	3.32	3.32
	(3.83)	(1.57)		(84.77)	(1.42)	(1.42)

CPU time

FIGURE 1
CPU TIME USED TO SOLVE MODEL



Note: When the models were solved, I used 100 Halton Draws to calculate integrals and 6 Chebyshev coefficients in each dimension of the state space for the models with up to 4 state variables. For the model with 5 state variables, I used only 5 Cebyshev coefficient in each dimension of the state space. The models were solved using a IMB ThinkPad T41 with a 1.6 GHz Pentium M processor and 2 GB RAM.

Monte Carlo - Unobserved heterogeneity

Rust's model with random coefficients

Experimental design:

- Replacemen costs, RC_i
  - *bus specific* and *randomly distributed* in the population of busses
  - normally distributed with mean $\overline{RC}$ and variance σ^2_{RC}.
- Linear cost function $C(x) = cx$
- Parameters: c and $\overline{RC}$, are set roughly equal to the ML estimates from one of the linear specifications of Rust (1987). [1]
- Sample sizes of $N = 100$, $T = 250$
- I draw 5000 Monte Carlo samples, and for each of them, I obtain partial ML estimates for models estimated with and without unobserved heterogeneity.

[1] Rust (1987) report ML estimates for bus groups 1,2 and 3 as $RC = 11.7270$ and $c = 0.001 * 4.8259$. However, the unit of measurement for c is in units of the discretized state variable, $x_t^d = 1, 2, .., 90$. With the chosen dimension of the grid for mileage, the coefficient to the discretized variable x_t^d must be divided by 5, since the interval length corresponds to 5000 miles in continuous measurement. I therefore set $\bar{c} = 1$ and $\overline{RC} = 1$

Random vs Fixed Coefficients

TABLE 5
MONTE CARLO EXPERIMENT
FIXED AND RANDOM COEFFICIENTS

Monte Carlo Distribution of ML and MSL estimates

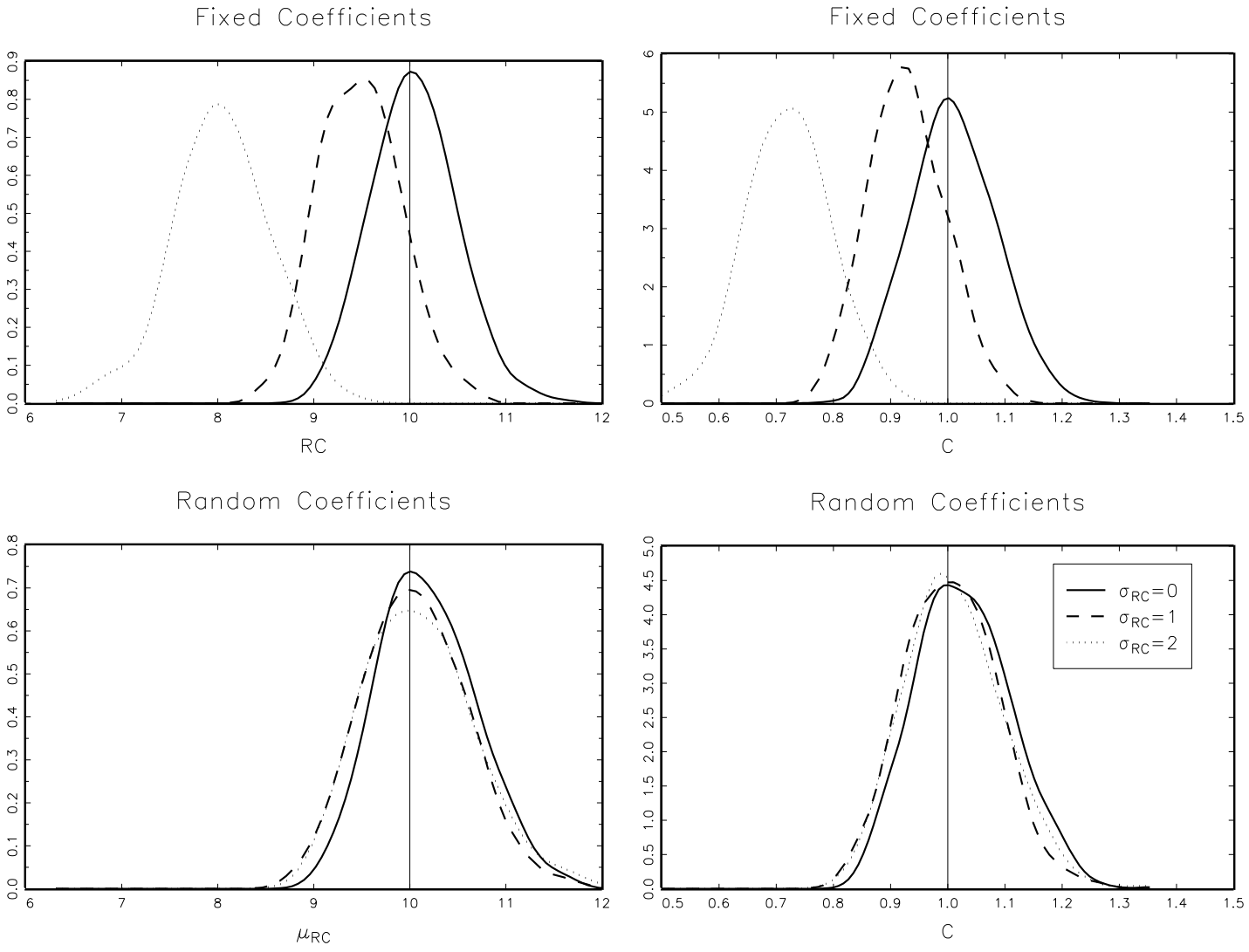
		Fixed Coefficients		Random Coefficients		
Statistic	σ_{RC}^{dgp}	RC	c	μ_{RC}	σ_{RC}	c
Mean Bias	0	0.05	0.01	0.19	0.41	0.03
	1	-0.52	-0.07	0.05	-0.02	0.01
	2	-1.96	-0.28	0.10	0.08	0.01
Mean Absolute E	0	0.35	0.06	0.43	0.41	0.07
	1	0.57	0.08	0.43	0.29	0.06
	2	1.96	0.28	0.46	0.30	0.07
Monte Carlo	0	0.44	0.074	0.51	0.38	0.082
std. dev	1	0.42	0.066	0.54	0.37	0.080
	2	0.51	0.075	0.57	0.38	0.084
Mean std. Error	0	0.45	0.073	0.46	0.13	0.071
	1	0.41	0.068	0.52	0.08	0.079
	2	0.28	0.053	0.56	0.06	0.080

Bias in estimated coefficients ratios

	σ_{RC}^{dgp}	RC/c	μ_{RC}/c
Mean Bias	0	-0.3%	-0.7%
	1	2.4%	0.2%
	2	12.1%	0.2%

Random vs Fixed Coefficients

FIXED AND RANDOM COEFFICIENTS



Note: See the note in Table 5 for a description of the experimental design.

Conclusion

- A computationally efficient, general method for the implementation and estimation of discrete Markov decision models has been proposed.
- Its theoretical properties is analyzed.
- Its numerical performance investigated through Monte Carlo studies.
- The method works very well and can handle a wide variety of specifications without problems.

Future Research

- Obtain sharper rates for the approximation errors in the MLE.
- Derive expressions of the leading bias and variance terms.

- PART III

A model of Vehicle Ownership, Type Choice and Usage

Model

Household's problem:

$$\max_{\{d_t, vkt_t\}_{t=1}^{T}} E \sum_{t=1}^{T} \beta^t u(d_t, vkt_t; \tau_t, a_t, x_t, \eta, p_t, m_t)$$

where $d_t \in \{(1, \ldots, \overline{\tau}) \times (0, 1, \ldots, \overline{a}), 0, -1\}$, $vkt_t \in \mathbb{R}_{++}$

The household can either

1. sell the existing car and not buy a new one, $d = -1$
2. keep the existing car another year, $d = 0$
3. trade the existing car for another vehicle, $d = (\tau', a')$

where $\tau \in (1, \ldots, \overline{\tau})$ denote the "type" or "class" of vehicle and $a \in \{0, 1, \ldots, \overline{a}\}$ denotes the *age* of the car a

Current utility

Utility (direct):

$$u(d_t, vkt_t; \tau_t, a_t, x_t, \eta, p_t, m_t) = u(Inc_t - replacement\ cost_t - fuel\ cost_t - operating/maintenance\ cost_t) + driving\ utility_t,$$

$$\begin{aligned} Inc_t &= y(x_t, age_t, m_t) \\ replace.\ cost_t &= \mathbf{1}_{\{trade\}}[P_t^{buy} - P_t^{sell} + \gamma^{\text{transact}}] \\ fuel\ cost_t &= \frac{p_t}{fe_t(a_t, \tau_t)} vkt_t \\ o/m\ cost_t &= c(a_t, \tau_t) \\ driving\ utility_t &= u^{drive}(vkt_t, a_t, \tau_t, \eta) \end{aligned}$$

Optimal Driving

- We assume that households disregard the effect driving have on the value of the car.
- Therefore driving decision is purely static and we can derive driving, vkt_t^*, only conditional car type and age, (a_t, τ_t),
- Conditional on optimal driving, we can simply solve the model as a dynamic discrete choice problem (type/car age choice and replacement/ownership decisions)
- Given optimal vkt_t^* for each discrete vehicle choice, we can derive *indirect* utility for each decision

$$
\begin{aligned}
u_{Keep}(\tau, a, x, \eta, p, m) &= u(d_t = 0, vkt_t^*; \tau, a, x, \eta, p, m) \\
u_{Sell}(\tau, a, x, \eta, p, m) &= u(d_t = -0, vkt_t^*; \tau, a, x, \eta, p, m) \\
u_{Repl.}(\tau, a, \tau', a', x, \eta, p, m) &= u(d_t = (\tau', a'), vkt_t^*; \tau, a, x, \eta, p, m)
\end{aligned}
$$

The dynamic program

Keep car, $d_t = 0$

The recursive Bellman-like equation for v_s for a household that decides to keep its current vehicle, $d = 0$ is given by

$$v_s(\tau, a, x, \eta, p, m, d = 0) = u_{Keep}(\tau, a, x, \eta, p, m) +$$
$$\beta_s \int_{x'} \int_{p'} \int_{m'} \log \left(\sum_{d' \in D} \exp\{v_{s+1}(\tau, a+1, x', \eta, p', m', d')\} \right)$$
$$g_s(x'|x, p', m') dx' \, h(p', m'|p, m) dp' \, dm'$$

where

- β_s is a mortality-adjusted discount factor for the household,
- $g_s(x'|x, p', m')$ is a transition probability density for the household's state x' at age $s+1$ given their current state x at age s and the next period macro state (p', m'),
- $h(p', m'|p, m)$ is a transition density for gas prices and the macro state.

The dynamic program

Sell car, do not replace, $d_t = -1$

If the household chooses $d = -1$, i.e. to sell its vehicle but not replace it with a new one, the Bellman equation is given by

$$v_s(\tau, a, x, \eta, p, m, d = -1) = u_{Sell}(\tau, a, x, \eta, p, m) +$$
$$\beta_s \int_{x'} \int_{p'} \int_{m'} \log \left(\sum_{d' \in D} \exp\{v_{s+1}(\emptyset, \emptyset, x', \eta, p', m', d')\} \right)$$
$$g_s(x'|x, p', m') dx' h(p', m'|p, m) dp' dm'$$

- $u_{Sell}(\tau, a, x, \eta, p, m)$ does not include utility of driving, but depends on the value of selling existing car (τ, a).

The dynamic program

Trade current vehicle (τ, a) for a new one, $d_t = \{\tau', a'\}$

If the household chooses to trade their current vehicle (τ, a) for a new one (τ', a') the Bellman equation is given by

$$v_s(\tau, a, x, \eta, p, m, d = \{\tau', a'\}) = u_{Repl.}(\tau, a, \tau', a', x, \eta, p, m) +$$
$$\beta_s \int_{x'} \int_{p'} \int_{m'} \log \left(\sum_{d' \in D} \exp\{v_{s+1}(\tau', a' + 1, x', \eta, p', m', d')\} \right)$$
$$g_s(x'|x, p', m') dx' h(p', m'|p, m) dp' dm'$$

where

- $u_{Repl.}(\tau, a, \tau', a', x, \eta, p, m)$ depends on optimal driving and the *total expected transactions cost* involved in selling the existing car (τ, a) to buy another car (τ', a').