Lecture 12: NFXP

Dynamic Programming

Thomas Jørgensen

Outline for today

1. General formulation of discrete-choice problem
2. Focus on Rust (1987) model

Outline

Overview of Rust (1987)

- **The economic question:** For how long one should continue to operate and maintain a bus before it is optimal to replace or rebuild the engine?
- **The model:** The optimal replacement decision is the solution to a dynamic optimization problem that formalizes the trade-off between two conflicting objectives:
  - *minimizing maintenance and replacement costs versus minimizing unexpected engine failures*
  - (the productivity/efficiency of a machine declines over time, but replacing a machine is costly).
- **Empirical question:** Did the decision maker (the superintendent of maintenance, Harold Zurcher) behave according to the optimal replacement rule implied by the theory model?
- **Structural estimation:** Using data on *monthly mileage and engine replacements* for a sample of GMC busses, Rust estimate the structural parameters in the engine replacement model using NFXP

Overview of Rust (1987)

This is a path-breaking paper that introduces a methodology to estimate a single-agent dynamic discrete choice models.

Main contributions

1. Development and implementation of ***Nested Fixed Point Algorithm***
   1. Formulation of assumptions, that makes dynamic discrete choice models tractable.
   2. Bottom-up approach
   3. An illustrative application in a simple model of engine replacement.
   4. The first researcher to obtain ML estimates of discrete choice dynamic programming models

Policy experiments:

- How does changes in replacement cost affect
- the distribution of mileage
- the demand for engines

Outline

General Behavioral Framework

The decision maker chooses a sequence of discrete actions to maximize expected discounted utility over a finite horizon

$$V_\theta(s_t) = \sup_{\Pi} \mathbb{E}\left[\sum_{j=0}^{\infty} \beta^j U\left(s_{t+j}, d_{t+j}; \theta_1\right) | s_t, d_t\right]$$

where

- $\Pi = (d_t, d_{t+1}, ..,), d_t \in C(x_t) = \{1, 2, .., J\}$
- State-transitions, $s_t = (x_t, \varepsilon_t)$: $p(x_{t+1}, \varepsilon_{t+1} | x_t, \varepsilon_t, d_t, \theta_2, \theta_3)$
- $\beta \in (0, 1)$ is the discount factor
- $U(s_t, d_t; \theta_1)$ is a choice and state specific utility function
- $\mathbb{E}$ summarizes expectations of future states given s_t and d_t
- $\theta = (\beta, \theta_1, \theta_2, \theta_3)$
- $T = \infty$

Rust's Assumptions (now standard!)

Assumption (CI)

State variables, $s_t = (x_t, \varepsilon_t)$ obeys a (conditional independent) controlled Markov process with probability density

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

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 (i.e. Gumbel)

Object of interest: $\theta = (\beta, \theta_1, \theta_2, \theta_3)$
The vector of structural parameters to be estimated.

The Dynamic Programming Problem

- Under AS, the optimal decision solves the following DP problem

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

The Dynamic Programming Problem

- Under AS, the optimal decision solves the following DP problem

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

- Under (CI) and (XV) we can integrate out the unobserved state variables in **closed form** [lecture 5].
- The unknown function, EV_θ, no longer depends on ε_t:

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

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

- Reduced state-space + closed-form for part of the integral!
- Γ_θ is a *contraction mapping* with unique fixed point EV_θ, i.e. $\|\Gamma\left(EV\right) - \Gamma\left(W\right)\| \leq \beta \left\|EV - W\right\|$ [Theorem from lecture 4]

Outline

Zurcher's Bus Engine Replacement Problem

- **Choice set:** Binary choice set, $C(x_t) = \{0, 1\}$.
  Each bus comes in for repair once a month and Zurcher chooses between ordinary maintenance $(d_t = 0)$ and overhaul/engine replacement $(d_t = 1)$.
- **State variables:** (Harold Zurcher observes)
  - x_t: mileage at time t since last engine overhaul
    If engine is replaced, state of bus regenerates to $x_t = 0$.
  - $\varepsilon_t = (\varepsilon_t(d_t = 0), \varepsilon_t(d_t = 1))$: other variable (unobs. to us)

Zurcher's Bus Engine Replacement Problem

- **Choice set:** Binary choice set, $C(x_t) = \{0, 1\}$.
  Each bus comes in for repair once a month and Zurcher chooses between ordinary maintenance $(d_t = 0)$ and overhaul/engine replacement $(d_t = 1)$.
- **State variables:** (Harold Zurcher observes)
  - x_t: mileage at time t since last engine overhaul
    If engine is replaced, state of bus regenerates to $x_t = 0$.
  - $\varepsilon_t = (\varepsilon_t(d_t = 0), \varepsilon_t(d_t = 1))$: other variable (unobs. to us)
- **Utility function:**

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

- **State variables process:** x_t (mileage since last replacement)

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

Likelihood Function

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

$$\ell^f\left(x_1, \ldots x_T, d_1, \ldots d_r | x_0, d_0, \theta\right) = \prod_{t=1}^{T} P\left(d_t | x_t, \theta\right) p\left(x_t | x_{t-1}, d_{t-1}, \theta_2\right)$$

where $P\left(d_t | x_t, \theta\right)$ is the ***choice probability*** given the observable state variable, x_t.

- **How to estimate the transition probability,** $p\left(x_t | x_{t-1}, d_{t-1}, \theta_2\right)$?
  - Can be estimated non-parametrically or by NLS
- **How to compute the choice probability,** $P\left(d_t | x_t, \theta\right)$?
  - Need to solve dynamic programming problem [next]

Conditional Choice Probabilities

- Under the extreme value assumption **choice probabilities** are multinomial logistic

$$P\left(d|x,\theta\right)=\frac{\exp\left\{u\left(x,d,\theta_1\right)+\beta EV_\theta\left(x,d\right)\right\}}{\sum_{j\in C(y)}\exp\left\{u\left(x,j,\theta_1\right)+\beta EV_\theta\left(x,j\right)\right\}} \tag{3}$$

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

$$\begin{aligned} EV_\theta\left(x,d\right) &= \Gamma_\theta(EV_\theta)\left(x,d\right) \qquad (4)\\ &= \int_y \ln\left[\sum_{d'\in D(y)}\exp\left[u(y,d';\theta_1)+\beta EV_\theta\left(y,d'\right)\right]\right] \\ &\quad \times p\left(dy|x,d,\theta_2\right) \end{aligned}$$

- **Structural Estimation**: Rust's *Nested Fixed Point Algorithm* (NFXP)

Structural Estimation: The Nested Fixed Point Algorithm

NFXP solves the optimization problem, where $L(\theta, EV_\theta) = \ell^f$,

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

2. **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

1. **Inner loop (fixed point algorithm):**
   The implicit function EV_θ defined by $EV_\theta = \Gamma(EV_\theta)$ is solved by:
   - Successive Approximations (SA): VFI
   - Newton-Kantorovich (NK) Iterations: Policy iteration

MATLAB implementation of the likelihood

```
function [f,g,h]=ll(data, mp, pnames, theta, ap)
  global ev0;
  % update model parameters
  mp=vec2struct(theta, pnames, mp);

  % Update u, du and P evaluated in grid points
  dc=0.001*mp.grid;
  cost=mp.c*0.001*mp.grid;
  P = zurcher.statetransition(mp.p, mp.n);

  % Solve model
  bellman= @(ev) zurcher.bellman(ev, P, cost, mp);
  [ev0, pk, dev]=solve.poly(bellman, ev0, ap, mp.beta);

  % Evaluate log-likelihood regarding replacement choice
  lp=pk(data.x);
  logl=log(lp+(1-2*lp).*(data.d));

  % add on log-likelihood for mileage process
  p=[mp.p; 1-sum(mp.p)];
  logl=logl + log(p(1+ data.dx1));

  % Objective function (negative mean log likleihood)
  f=mean(-logl);
```

Data

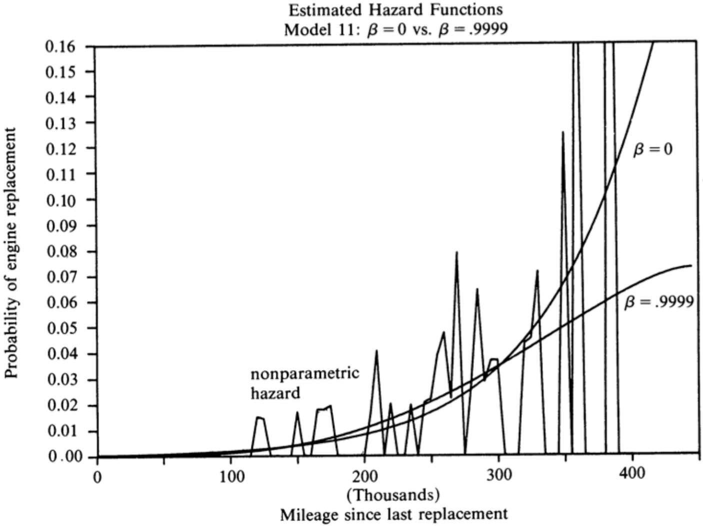
- Harold Zurcher's Maintenance records of 162 busses
- Monthly observations of mileage on each bus (odometer reading)
- Data on maintenance operations

1. Routine, periodic maintenance (e.g. brake adjustments)
2. Replacement or repair at time of failure
3. Major engine overhaul and/or replacement

Rust focus on 3)

Overview of Rust (1987)
General Framework
An Empirical Model of Harold Zurcher

Estimated Hazard Functions



Specification Search

TABLE VIII

SUMMARY OF SPECIFICATION SEARCH[a]

	Bus Group		
Cost Function	1, 2, 3	4	1, 2, 3, 4
Cubic $c(x, \theta_1) = \theta_{11}x + \theta_{12}x^2 + \theta_{13}x^3$	Model 1 −131.063 −131.177	Model 9 −162.885 −162.988	Model 17 −296.515 −296.411
quadratic $c(x, \theta_1) = \theta_{11}x + \theta_{12}x^2$	Model 2 −131.326 −131.534	Model 10 −163.402 −163.771	Model 18 −297.939 −299.328
linear $c(x, \theta_1) = \theta_{11}x$	Model 3 −132.389 −134.747	Model 11 −163.584 −165.458	Model 19 −300.250 −306.641
square root $c(x, \theta_1) = \theta_{11}\sqrt{x}$	Model 4 −132.104 −133.472	Model 12 −163.395 −164.143	Model 20 −299.314 −302.703
power $c(x, \theta_1) = \theta_{11}x^{\theta_{12}}$	Model 5[b] N.C. N.C.	Model 13[b] N.C. N.C.	Model 21[b] N.C. N.C.
hyperbolic $c(x, \theta_1) = \theta_{11}/(91 - x)$	Model 6 −133.408 −138.894	Model 14 −165.423 −174.023	Model 22 −305.605 −325.700
mixed $c(x, \theta_1) = \theta_{11}/(91 - x) + \theta_{12}\sqrt{x}$	Model 7 −131.418 −131.612	Model 15 −163.375 −164.048	Model 23 −298.866 −301.064
nonparametric $c(x, \theta_1)$ any function	Model 8 −110.832 −110.832	Model 16 −138.556 −138.556	Model 24 −261.641 −261.641

[a] First entry in each box is (partial) log likelihood value ℓ^2 in equation (5.2)) at $\beta = .9999$. Second entry is partial log likelihood value at $\beta = 0$.

Structural Estimates, n=90

TABLE IX

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

FIXED POINT DIMENSION = 90

(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 ($df = 4$)	Marginal Significance Level
$\beta = .9999$	RC	11.7270 (2.602)	10.0750 (1.582)	9.7558 (1.227)	85.46	1.2E − 17
	θ_{11}	4.8259 (1.792)	2.2930 (0.639)	2.6275 (0.618)		
	θ_{30}	.3010 (.0074)	.3919 (.0075)	.3489 (.0052)		
	θ_{31}	.6884 (.0075)	.5953 (.0075)	.6394 (.0053)		
	LL	−2708.366	−3304.155	−6055.250		
$\beta = 0$	RC	8.2985 (1.0417)	7.6358 (0.7197)	7.3055 (0.5067)	89.73	1.5E − 18
	θ_{11}	109.9031 (26.163)	71.5133 (13.778)	70.2769 (10.750)		
	θ_{30}	.3010 (.0074)	.3919 (.0075)	.3488 (.0052)		
	θ_{31}	.6884 (.0075)	.5953 (.0075)	.6394 (.0053)		
	LL	−2710.746	−3306.028	−6061.641		
Myopia test:	LR Statistic ($df = 1$)	4.760	3.746	12.782		
$\beta = 0$ vs. $\beta = .9999$	Marginal Significance Level	0.0292	0.0529	0.0035		

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		

Estimating parameters, bustypes 1,2,3,4 (model 19)

```
Output from run_busdata.m

Structural Estimation using busdata from Rust(1987)
Beta            =    0.99990
n               =  175.00000
Sample size     = 8156.00000


   Param.                   Estimates          s.e.        t-stat
---------------------------------------------------------------------
   RC                          9.7498        1.2249        7.9596
   c                           1.3385        0.3143        4.2589
   p                 (1)       0.1070        0.0034       31.2107
   p                 (2)       0.5152        0.0055       93.0556
   p                 (3)       0.3622        0.0053       68.0405
   p                 (4)       0.0143        0.0013       10.8946
   p                 (5)       0.0009        0.0003        2.6469
---------------------------------------------------------------------
log-likelihood     = -8607.89002
runtime (seconds) =     0.36119
```

Infinite Horizon “Speed Up tricks”

- Solving the fixed-point problem by VFI or “successive approximations” can be quite slow if β is close to 1.
  - The convergence rate of VFI is linear with a scale equal to β
- There are some ideas and tricks to speed up the solution:

1. Newton-Kantorovich Iterations
2. MPEC
3. Nested Pseudo Likelihood (NPL): Next time

1: Newton-Kantorovich Iterations (1/3)

- **Everything is in Function space (Banach), but very similar to what we already know**
- Recall the Bellman operator

$$EV(x,d) = \Gamma(EV)(x,d)$$

- We can write this as

$$\underbrace{(I - \Gamma)EV}_{"f(x)"} = 0$$

1: Newton-Kantorovich Iterations (1/3)

- **Everything is in Function space (Banach), but very similar to what we already know**
- Recall the Bellman operator

$$EV(x,d) = \Gamma(EV)(x,d)$$

- We can write this as

$$\underbrace{(I-\Gamma)EV}_{''f(x)''} = 0$$

- We can then use Newtons method to solve for a zero:

$$\underbrace{EV^{k+1}}_{''x^{k+1}''} = \underbrace{EV^{k}}_{''x^{k}''} - \underbrace{(I-\Gamma')^{-1}}_{''f'(x^k)^{-1}''}\underbrace{(I-\Gamma)EV^{k}}_{''f(x^k)''}$$

- Where Γ' is the **Fréchet derivative** of the operator Γ.

1: Newton-Kantorovich Iterations (2/3)

- **The Fréchet derivative**
  - In terms of its finite-dimensional approximation, Γ'_θ takes the form of an $N \times N$ matrix equal to the partial derivatives of the $N \times 1$ vector $\Gamma_\theta(EV_\theta)$ with respect to the $N \times 1$ vector EV_θ
  - Γ'_θ is simply β times the transition probability matrix for the controlled process $\{d_t, x_t\}$
  - Recall that we have

$$\begin{aligned} EV(x,d) &= \int_y \ln\left[\sum_{d' \in D(y)} \exp\left[u(y,d') + \beta EV(y,d')\right]\right] \times p(dy|x,d) \\ &\approx \sum_{x'} p(x'|x,d) \sum_{d'} P(d'|x')\left[u(x',d') + \beta EV(x',d')\right] \end{aligned}$$

such that the (k,j)-element of the derivative matrix:

$$\begin{aligned} \Gamma_{(k)}(x_{(j)},d)' &= \text{"}\frac{\partial \Gamma(EV)\left(x_{(j)},d\right)}{\partial EV_{(k)}}\text{"} \\ &= \beta \sum_{x'} \mathbf{1}_{\{x'=k\}} p\left(x'|x_{(j)},d\right) \sum_{d'} P(d'|x') \end{aligned}$$

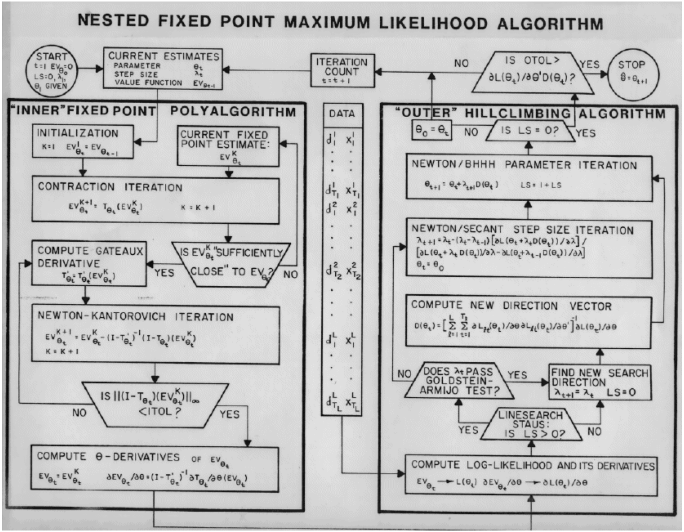
1: Newton-Kantorovich Iterations (3/3)

- **The convergence rate**:
  - We 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

1: Newton-Kantorovich Iterations (3/3)

- **The convergence rate**:
  - We 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
- **Not global convergence:**
  - The method only works well within a "pool of attraction", however
  - Use a poly-algorithm that starts with VFI then switch to NK
- **When to switch to Newton-Kantorovich**?
  - Recall from VFI: $tol_k = \|EV_{k+1} - EV_k\| < \beta\|EV_k - EV\|$
  - Relative tolerance tol_{k+1}/tol_k approach β
  - Switch to Newton whenever tol_{k+1}/tol_k is sufficiently close to β

John’s pocket guide



2: MPEC, Su and Judd (2012) (1/2)

- **NFXP** solves the *unconstrained* optimization problem

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

  - **Outer loop:** Likelihood function $L(\theta, EV_{\theta})$ is maximized w.r.t. θ
  - **Inner loop:** The implicit function EV_{θ} defined by $EV_{\theta} = \Gamma(EV_{\theta})$ is solved by VFI and NK

- **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

  - **Constraint** is

$$EV(x,d) = \sum_{x'} p\left(x'|x,d;\theta\right) \sum_{d'} P\left(d'|x';\theta\right) \left[u(x',d';\theta) + \beta EV\left(x',d'\right)\right]$$

2: MPEC: ECTA comment (2/2)

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

Until next time

- **Ensure that you understand:**
  1. Discrete choice problems
  2. infinite horizon speed tricks. Especially Newton-Kontorowich
- **Next time:** Nested Pseudo Likelihood (NPL)
  "Swapping the nested fixed point": another way to reduce estimation time!