

Topics in Macroeconomics

Value function iteration (VFI)

Richard Foltyn

University of Glasgow

Solving household problems with VFI

We discuss the following models:

- 1 No uncertainty, no labour income: analytical solution
- 2 Certain labour income
- 3 Risky labour income

In all cases we assume CRRA preferences!

Solution methods

- 1 Grid search: no interpolation
- 2 “Unrestricted” maximisation:
 - 1 Linear interpolation
 - 2 Spline interpolation

VFI with analytical solution

Analytical solution

- Under some conditions value function has closed-form solution!
- Assumptions:
 - 1 CRRA preferences
 - 2 No labour income
 - 3 No uncertainty
- Solution methods:
 - 1 Iteration on value function (“manual” VFI)
 - 2 Guess and verify (not covered)

Illustration: manually iterate on closed form, compare with numerical solution.

Household problem

Analytical VFI

Consider infinite-horizon consumption-savings problem with log preferences:

$$\begin{aligned} V(a) &= \max_{c, a'} \left\{ \log(c) + \beta V(a') \right\} \\ \text{s.t. } \quad c + a' &= (1 + r)a \\ c \geq 0, \quad a' &\geq 0 \end{aligned}$$

where

a Beginning-of-period assets

a' Next-period assets (savings)

r Constant interest rate

β Discount factor $\beta \in (0, 1)$

Solving the household problem

Analytical VFI

How can we find V using VFI?

- Initial guess:
 - Consume everything: $c = (1 + r)a$
 - Continuation value is zero
- Value function in iteration 1:

$$V_1(a) = \log(c) = \log(1 + r) + \log(a)$$

- HH problem in iteration 2:

$$\begin{aligned} V_2(a) &= \max_{c, a'} \left\{ \log(c) + \beta V_1(a') \right\} \\ &= \max_{c, a'} \left\{ \log(c) + \beta \left[\log(1 + r) + \log(a') \right] \right\} \\ \text{s.t. } &c + a' = (1 + r)a \\ &c \geq 0, a' \geq 0 \end{aligned}$$

Solving the household problem

Analytical VFI

Iteration 2

- First-order conditions:

$$c^{-1} = \lambda$$

$$\beta (a')^{-1} = \lambda$$

where λ is Lagrange multiplier on budget constraint.

- Eliminate Lagrange multiplier: $a' = \beta c$
- Substitute for a' in budget constraint to find policy functions:

$$\begin{aligned} c + \beta c &= (1+r)a \implies c = (1+\beta)^{-1}(1+r)a \\ &\implies a' = \beta(1+\beta)^{-1}(1+r)a \end{aligned}$$

- Plug policy functions into value function:

$$\begin{aligned} V_2(a) &= \log \left((1+\beta)^{-1}(1+r)a \right) + \beta \left[\log(1+r) + \log \left(\beta(1+\beta)^{-1}(1+r)a \right) \right] \\ &= \beta \log \beta - (1+\beta) \log(1+\beta) + (1+2\beta) \log(1+r) + (1+\beta) \log(a) \end{aligned}$$

Solving the household problem

Analytical VFI

After 2 iterations we have:

$$V_1(a) = \underbrace{\log(1+r)}_{\chi_1} + \underbrace{1}_{\varphi_1} \times \log(a)$$

$$V_2(a) = \underbrace{\beta \log \beta - (1+\beta) \log(1+\beta) + (1+2\beta) \log(1+r)}_{\chi_2} + \underbrace{(1+\beta) \log(a)}_{\varphi_2}$$

- Continue iterating? – Expressions become too complicated!
- Instead conjecture that value function takes the form

$$V_n(a) = \chi_n + \varphi_n \log(a) \tag{1}$$

We have shown this to be true for $n = 1, 2$

Solving via induction

Analytical VFI

- Assume V has functional form given in (1) for some n
- Then V_{n+1} will be given by

$$V_{n+1}(a) = \chi_{n+1} + \varphi_{n+1} \log(a)$$

- Task: find $\chi_{n+1}, \varphi_{n+1}$ given χ_n, φ_n
- We do this by solving

$$\begin{aligned} V_{n+1}(a) &= \max_{c, a'} \left\{ \log(c) + \beta V_n(a') \right\} \\ &= \max_{c, a'} \left\{ \log(c) + \beta \left[\chi_n + \varphi_n \log(a') \right] \right\} \\ \text{s.t. } & c + a' = (1+r)a \\ & c \geq 0, a' \geq 0 \end{aligned}$$

Solving via induction

Analytical VFI

Iteration $n + 1$

- First-order conditions for the $(n + 1)$ -th iteration:

$$c^{-1} = \lambda \qquad \beta \varphi_n (a')^{-1} = \lambda$$

- Substitute for a' in budget constraint to find policy functions:

$$c + \beta \varphi_n c = (1 + r)a \implies c = (1 + \beta \varphi_n)^{-1} (1 + r)a \quad (2)$$

$$\implies a' = \frac{\beta \varphi_n}{1 + \beta \varphi_n} (1 + r)a \quad (3)$$

- Plug policy functions into value function:

$$V_{n+1}(a) = \log\left((1 + \beta \varphi_n)^{-1} (1 + r)a\right) + \beta \left[\chi_n + \varphi_n \log\left(\frac{\beta \varphi_n}{1 + \beta \varphi_n} (1 + r)a\right) \right]$$

Solution for V

Analytical VFI

- Collect terms:

$$V_{n+1}(a) = \underbrace{\beta\chi_n + \beta\varphi_n \log(\beta\varphi_n) - (1 + \beta\varphi_n) \left[\log(1 + \beta\varphi_n) + \log(1 + r) \right]}_{\chi_{n+1}} + \underbrace{(1 + \beta\varphi_n) \log(a)}_{\varphi_{n+1}}$$

- Pin down sequence of $(\varphi_n)_{n=1}^{\infty}$:
 - Follows first-order linear difference equation

$$\varphi_{n+1} = 1 + \beta\varphi_n$$

- General solution (φ_0 pinned down by $\varphi_1 = 1$):

$$\varphi_n = \beta^n \left(\varphi_0 - \frac{1}{1 - \beta} \right) + \frac{1}{1 - \beta}$$

- Convergence to limit:

$$\varphi = \lim_{n \rightarrow \infty} \varphi_n = \frac{1}{1 - \beta}$$

Solution for V

Analytical VFI

- Difference equation for χ_n is much more complicated (depends on φ_n !)
- Compute only limiting value (move $\beta\chi_n$ to l.h.s.):

$$\lim_{n \rightarrow \infty} \chi_{n+1} - \beta\chi_n =$$
$$\lim_{n \rightarrow \infty} \beta\varphi_n \log(\beta\varphi_n) - (1 + \beta\varphi_n) \left[\log(1 + \beta\varphi_n) + \log(1 + r) \right]$$

- Limiting value given by:

$$\chi = \lim_{n \rightarrow \infty} \chi_n = \frac{\beta}{1 - \beta} \log \beta + \log(1 - \beta) + \frac{1}{1 - \beta} \log(1 + r)$$

- Converged value function V :

$$V(a) = \chi + \frac{1}{1 - \beta} \log(a)$$

Value function convergence

Analytical VFI

Convergence of coefficients χ_n and φ_n in

$$V_n(a) = \chi_n + \varphi_n \log(a)$$

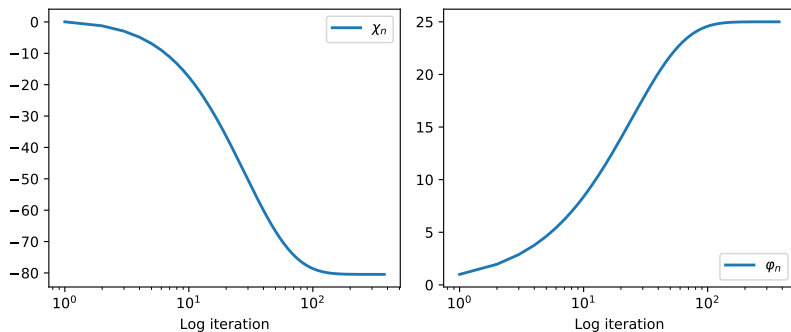


Figure 1: Convergence of analytical value function coefficients.

Analytical vs. numerical iteration

Analytical VFI

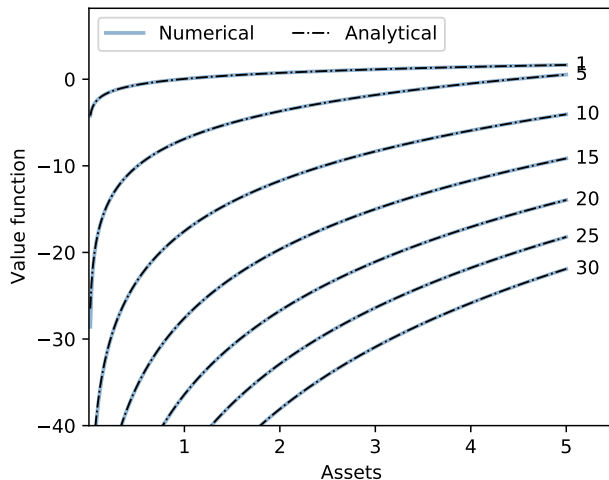


Figure 2: Value function V_n for the first few iterations.

Policy function convergence

Analytical VFI

Apply same reasoning to policy functions

- Define MPC as $\kappa_n \equiv (1 + \beta\varphi_n)^{-1}$
- Rewrite policy functions (2) and (3) at iteration $n + 1$ as:

$$c_{n+1} = \kappa_n(1 + r)a$$

$$a'_{n+1} = (1 - \kappa_n)(1 + r)a$$

- $\lim_{n \rightarrow \infty} \kappa_n = 1 - \beta$
- Policy functions usually converge faster than value functions!

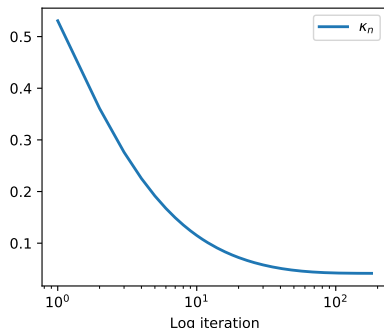


Figure 3: Convergence of policy function coefficient.

Analytical vs. numerical solution

Analytical VFI

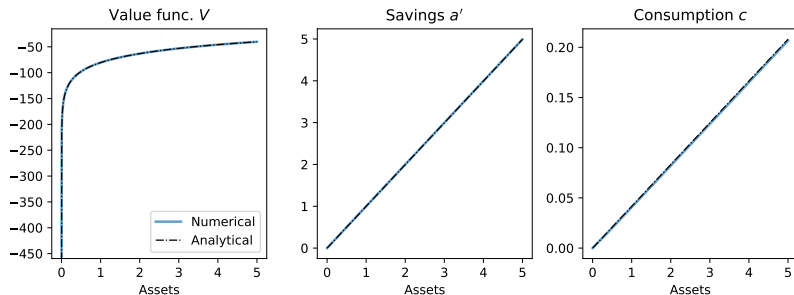


Figure 4: Converged value and policy functions.

VFI with grid search

VFI with grid search

- Restriction: solution method forces next-period assets to be **exactly** on discretized grid: $a' \in \mathcal{G}_a$
- Advantages:
 - 1 Easy to implement
 - 2 Derivative-free method
 - 3 Fast (unless grid is very dense)
- Disadvantages:
 - 1 Imprecise
 - 2 Policy functions are not smooth (unless grid is very dense)
 - 3 Does not scale well to multiple dimensions

Example: HH problem with constant labour income

VFI with grid search

- Infinitely-lived HH solves consumption-savings problem

$$\begin{aligned} V(a) &= \max_{c, a' \in \mathcal{G}_a} \left\{ u(c) + \beta V(a') \right\} \\ \text{s.t. } \quad &c + a' = (1 + r)a + y \\ &c \geq 0, \quad a' \geq 0 \end{aligned}$$

where

\mathcal{G}_a Beginning-of-period asset grid

y Constant labour income

- Preferences are assumed to be CRRA with relative risk aversion γ :

$$u(c) = \frac{c^{1-\gamma} - 1}{1 - \gamma}$$

- Note that with $\gamma = 1$, $u(c) = \log(c)$ as before

Solution algorithm

VFI with grid search

- 1 Create asset grid $\mathcal{G}_a = (a_1, \dots, a_{N_a})$
- 2 Pick initial guess for value function, V_0
- 3 In iteration n , perform the following steps
 - 1 For each asset level a_i , find all feasible next-period asset levels $a_j \leq (1+r)a_i + y$
 - 2 For each j , compute consumption $c_j = (1+r)a_i + y - a_j$
 - 3 For each j , compute utility

$$U_j = u(c_j) + \beta V_n(a_j) \tag{4}$$

- 4 Find the index k that maximises (4):

$$k = \arg \max_j \left\{ u(c_j) + \beta V_n(a_j) \right\}$$

- 5 Set $V_{n+1}(a_i) = U_k$ and store k as the optimal choice at a_i

Parametrisation for problem with constant labour income

VFI with grid search

- The next slides show solutions for the following parametrisation:

	Description	Value
β	Discount factor	0.96
σ	Coef. of relative risk aversion	2
r	Interest rate	0.04
y	Labour income	1
N_a	Asset grid size	50, 100, 1000

Table 1: Parameters for HH problem with constant labour income

- Each graph compares three solution methods:
 - 1 VFI with grid search
 - 2 VFI with linear interpolation
 - 3 VFI with cubic spline interpolation

Solution for $N_a = 50$

VFI with grid search

Grid search is quite sensitive to grid size!

Compare results for $N_a = 50$, $N_a = 100$ and $N_a = 1000$.

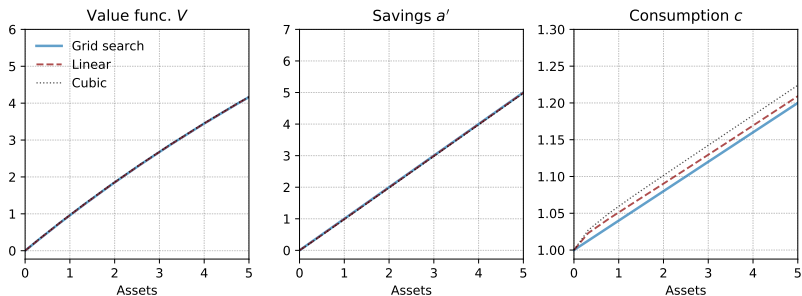


Figure 5: Solution with 50 asset grid points.

Solution for $N_a = 100$

VFI with grid search

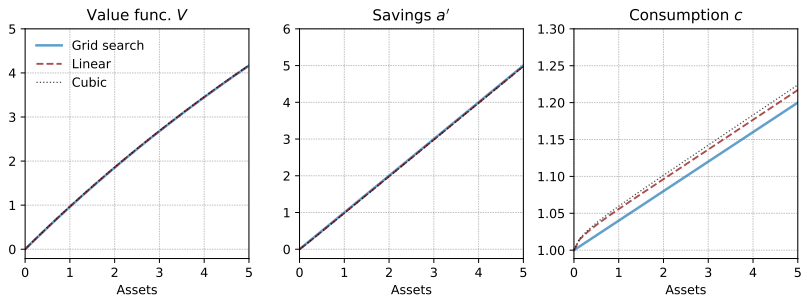


Figure 6: Solution with 100 assets grid points.

Solution for $N_a = 1000$

VFI with grid search

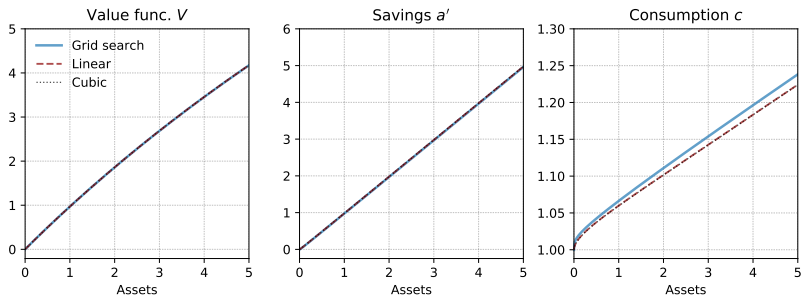


Figure 7: Solution with 1000 assets grid points.

VFI with interpolation

VFI with interpolation

- Grid search is rarely used today
- We prefer solution algorithms which find local maximum for each point on the grid (i.e. solution satisfies first-order conditions)
- Optimal points need not be on the grid, hence we have to **interpolate**
- Advantages:
 - 1 “Exact” solution (in a numerical sense)
 - 2 Less affected by curse of dimensionality in case of multiple choice variables
 - 3 Easier to spot mistakes since policy functions don’t have artificial kinks as in grid search
- Disadvantages:
 - 1 Likely slower than grid search
 - 2 More complex to implement:
 - Need maximisation or root-finding routine
 - Need to compute derivatives of objective function or first-order condition, unless we use derivative-free methods or numerical differentiation.

Example: HH problem with risky labour income

VFI with interpolation

- Illustrate VFI with interpolation using standard Bewley/Huggett/Aiyagari problem with risky labour income
- Infinite-lived HH solves consumption-savings problem

$$\begin{aligned} V(a, y) = \max_{c, a'} & \left\{ u(c) + \beta \mathbb{E} \left[V(a', y') \mid y \right] \right\} \\ \text{s.t.} \quad & c + a' = (1 + r)a + y \\ & c \geq 0, \quad a' \geq 0 \end{aligned}$$

where

- y Labour income process on state space \mathcal{G}_y with transition probability $\Pr(y' = y_j \mid y = y_i) = \pi_{ij}$
- As before, $u(\bullet)$ is CRRA
- Note that now we have a **two-dimensional state space** on $\mathcal{G}_a \times \mathcal{G}_y$.

Solution algorithm

VFI with interpolation

- 1 Create asset grid $\mathcal{G}_a = (a_1, \dots, a_{N_a})$
- 2 Create discrete labour income process with states $\mathcal{G}_y = (y_1, \dots, y_{N_y})$ and transition matrix Π_y
- 3 Pick initial guess for value function, V_0
- 4 In iteration n , perform the following steps
 - 1 For each point (a_i, y_j) in the state space, find

$$a^\star = \arg \max_{a' \in [0, x_{ij}]} \left\{ u(x_{ij} - a') + \beta \sum_{k=1}^{N_y} \pi_{jk} V_n(a', y_k) \right\}$$

where $x_{ij} = (1+r)a_i + y_j$ is the cash at hand.

- 2 Compute value at optimum,

$$V^\star = u(x_{ij} - a^\star) + \beta \mathbf{E} \left[V_n(a^\star, y') \mid y_j \right]$$

- 3 Set $V_{n+1}(a_i, y_j) = V^\star$ and store $A_{n+1}(a_i, y_j) = a^\star$ as the savings policy function.

Solution algorithm

VFI with interpolation

How do we find a^* ?

- 1 We use a maximiser that finds the maximum $a^* \in [0, x_{ij}]$ of the function

$$f(a' | a_i, y_j) = u(x_{ij} - a') + \beta \sum_{k=1}^{N_y} \pi_{jk} V_n(a', y_k)$$

for given (a_i, y_j) .

- Need to interpolate $V_n(\bullet, y_k)$ onto arbitrary a'
 - Need to either use derivative-free maximizer, or differentiate df/da' numerically
- 2 In principle, we could perform *root-finding* on the FOC

$$-u'(x_{ij} - a') + \beta \sum_{k=1}^{N_y} \pi_{jk} dV_n(a', y_k) / da' = 0$$

This is rarely done since we don't know $dV_n(a', y_k) / da'$ and fast root-finders *additionally* need the derivative of the FOC!

Parametrisation for problem with risky labour

VFI with interpolation

- Assume labour process follows AR(1),

$$y_{t+1} = \rho y_t + \varepsilon_{t+1} \quad \varepsilon_{t+1} \stackrel{\text{iid}}{\sim} \mathcal{N}(0, \sigma^2)$$

which we discretise as a Markov chain using the Rouwenhorst (1995) or Tauchen (1986) methods.

- The next slides show solutions for the following parametrisation:

	Description	Value
β	Discount factor	0.96
σ	Coef. of relative risk aversion	2
r	Interest rate	0.04
ρ	Autocorrelation of AR(1) process	0.95
σ	Conditional std. dev. of AR(1) process	0.20
N_y	Number of states for Markov chain	3
N_a	Asset grid size	50, 100, 1000

Table 2: Parameters for HH problem with risky labour income

Solution for $N_a = 50$

VFI with interpolation

Solution for different income levels: low, middle, high

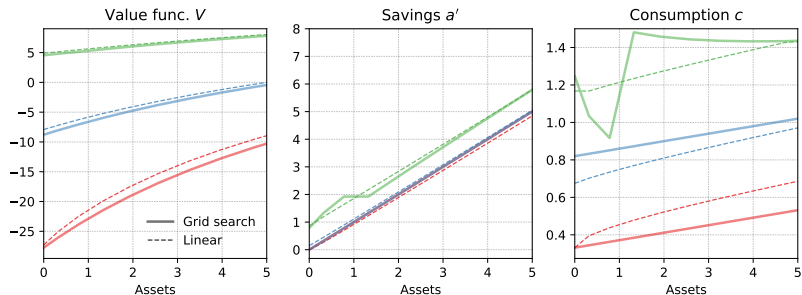


Figure 8: Solution with 50 asset grid points.

Solution for $N_a = 100$

VFI with interpolation

Solution for different income levels: low, middle, high

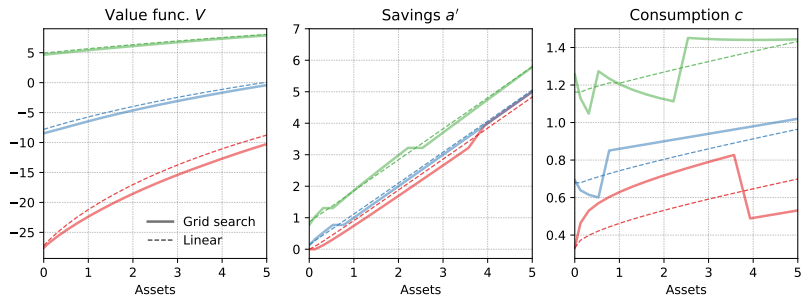


Figure 9: Solution with 100 assets grid points.

Solution for $N_a = 1000$

VFI with interpolation

Solution for different income levels: low, middle, high

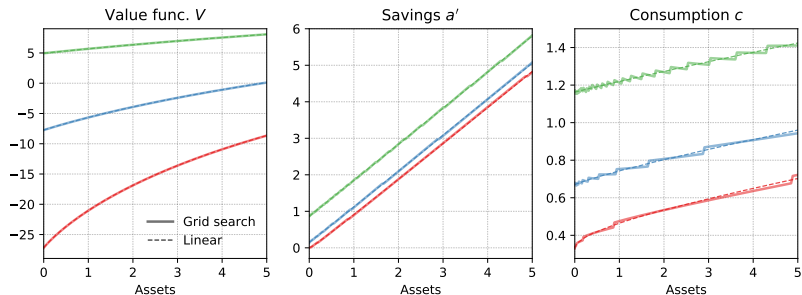


Figure 10: Solution with 1000 assets grid points.

Main take-aways

- Avoid grid search if you can!
- Test sensitivity of your solution to chosen grid size:
 - Check policy functions, value function almost always looks smooth!

Appendix

References I

- Rouwenhorst, Geert K. (1995). "Asset Pricing Implications of Equilibrium Business Cycle Models". In: **Frontiers of Business Cycle Research**. Ed. by Thomas F. Cooley. Vol. 10. Princeton University Press. Chap. 10, pp. 294–330.
- Tauchen, George (1986). "Finite state markov-chain approximations to univariate and vector autoregressions". In: **Economics Letters** 20.2, pp. 177–181.