An Introduction to Computational Macroeconomics

Dynamic Programming: Chapter 1

John Stachurski

June – July 2022

Introduction

Summary of this lecture:

- Symbols and terminology
- 2 minute introduction to Julia
- Finite horizon job search
- Linear equations
- Fixed point theory
- Infinite horizon job search

Common Symbols

$\mathbb{1}\{P\}$	equals 1 if statement P true, 0 otherwise
$\alpha := 1$	α is defined as equal to 1
$\mathbb{N}$, $\mathbb{Z}$ and $\mathbb{R}$	natural numbers, integers and real numbers
$\mathbb{C}$	complex numbers
$\mathbb{Z}_+$, $\mathbb{R}_+$, etc.	the nonnegative elements of $\mathbb{Z}$, $\mathbb{R}$, etc.
$\|x\|$ for $x \in \mathbb{R}$	absolute value of x
$\|\lambda\|$ for $\lambda \in \mathbb{C}$	modulus of λ
$a \vee b$	$\max\{a, b\}$
$a \wedge b$	$\min\{a, b\}$
$\|B\|$	the cardinality of set B
$\mathbb{R}^n$	all n-tuples of real numbers
$x \leqslant y \;\; (x, y \in \mathbb{R}^n)$	$x_i \leqslant y_i$ for $i = 1, \ldots n$ (pointwise partial order)
$x \ll y \;\; (x, y \in \mathbb{R}^n)$	$x_i < y_i$ for $i = 1, \ldots n$
$\mathcal{D}(F)$	the set of distributions (or pmfs) on F
$\mathbb{R}^M$	all functions from M to $\mathbb{R}$

Let M be any set

If $f\colon M \to \mathbb{R}$, then we call f a **real-valued function** on M

Let $\mathbb{R}^M$ be the **set of all real-valued functions on M**

If $f, g \in \mathbb{R}^M$ and $\alpha, \beta \in \mathbb{R}$, then

- $\alpha f + \beta g \in \mathbb{R}^M$ with $(\alpha f + \beta g)(x) := \alpha f(x) + \beta g(x)$
- $fg \in \mathbb{R}^M$ with $(fg)(x) := f(x)g(x)$
- $f \vee g \in \mathbb{R}^M$ with $(f \vee g)(x) := f(x) \vee g(x)$
- $f \wedge g \in \mathbb{R}^M$ with $(f \wedge g)(x) := f(x) \wedge g(x)$
- etc.

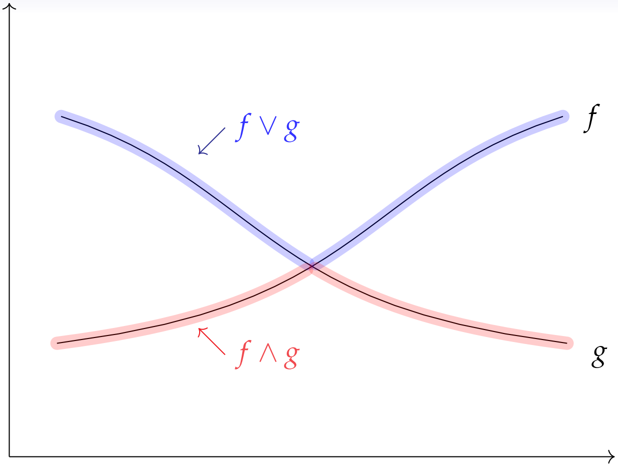


Figure: Functions $f \vee g$ and $f \wedge g$ when defined on a subset of $\mathbb{R}$

Operations on real numbers such as $|\cdot|$ and $\vee$ are applied to vectors element-by-element

Example.

$$a = \begin{pmatrix} a_1 \\ \vdots \\ a_n \end{pmatrix} \quad \Longrightarrow \quad |a| = \begin{pmatrix} |a_1| \\ \vdots \\ |a_n| \end{pmatrix}$$

$$a \vee b = \begin{pmatrix} a_1 \vee b_1 \\ \vdots \\ a_n \vee b_n \end{pmatrix} \quad \text{and} \quad a \wedge b = \begin{pmatrix} a_1 \wedge b_1 \\ \vdots \\ a_n \wedge b_n \end{pmatrix}$$

etc.

Topology in $\mathbb{R}^n$

Some quick reminders

A set $C \subset \mathbb{R}^n$ is called **closed** in $\mathbb{R}^n$ if

$$(u_m) \subset C \text{ and } u_m \to u \implies u \in C$$

A set G is called **open** if G^c is closed

$T \colon U \to V$ is called **continuous at $u \in U$** if

$$(u_m) \subset U \text{ and } u_m \to u \implies Tu_m \to Tu$$

We call T **continuous on U** if T is continuous at every $u \in U$

If $M = \{x_1, \ldots, x_n\}$ = some finite set, then

$$\mathbb{R}^M \text{ and } \mathbb{R}^n \text{ are the "same" set !}$$

Indeed, $f \in \mathbb{R}^M$ is defined by its values $f(x_1), \ldots, f(x_n)$

Hence $\exists$ a one-to-one correspondence between $\mathbb{R}^M$ and $\mathbb{R}^n$:

$$\mathbb{R}^M \ni f \quad \longleftrightarrow \quad (f(x_1), \ldots, f(x_n)) \in \mathbb{R}^n$$

Example.

- We call $C \subset \mathbb{R}^M$ closed if C is closed in $\mathbb{R}^n$, etc.
- If $f \in \mathbb{R}^M$, then $\|f\| :=$ the norm of $(f(x_1), \ldots, f(x_n))$

Julia Syntax: Two Minute Introduction

- Install from https://julialang.org/ (if you wish)
- To import `Library`, write `using Library`
- `f(x) = 2x` defines the function $f(x) = 2x$
- `cos.(x)` applies cos to each elements of vector x
- `x.^2` squares each element of vector x
- looping very similar to Python
- See also https://julia.quantecon.org/intro.html

```julia
# Defining functions, using conditions and loops

function f(x, y)                    # define a function
    if x < y                        # branch
        return sin(x + y)
    else
        return cos(x + y)
    end
end


function print_plurals(list_of_words)   # define a function
    for word in list_of_words           # loop
        println(word * "s")
    end
end
```

```julia
using LinearAlgebra          # import LinearAlgebra library

f(x) = 2x                    # simple function definition
f(x) = norm(x)               # norm defined in LinearAlgebra
g(x) = sum(x + x.^2)         # dot for pointwise operations
α, β = 2.0, -2.0             # unicode symbols

q(x) = sin(cos(x))           # another function

x = rand(5)
println(q(5))                  # OK
println(q.(x))                 # OK
println(q(x))                  # Error!
```

Let $f\colon A \to B$ and $g\colon B \to C$.

Recall that the **composition** of f and g is the map

$$g \circ f\colon A \to C, \quad A \ni a \mapsto g(f(a)) \in C$$

Example. $f(x) = x \wedge 0$ and $g(x) = x \vee 0$ implies $g \circ f \equiv 0$

In Julia we can compose as follows

```
f(x) = min(x, 0)
g(x) = max(x, 0)
h = f ∘ g        # type \circ and then hit tab
```

Dynamic program

an initial state X_0 is given
$t \leftarrow 0$
while $t < T$ **do**
  observe current state X_t
  choose action A_t
  receive reward R_t based on (X_t, A_t)
  state updates to X_{t+1}
  $t \leftarrow t + 1$
end

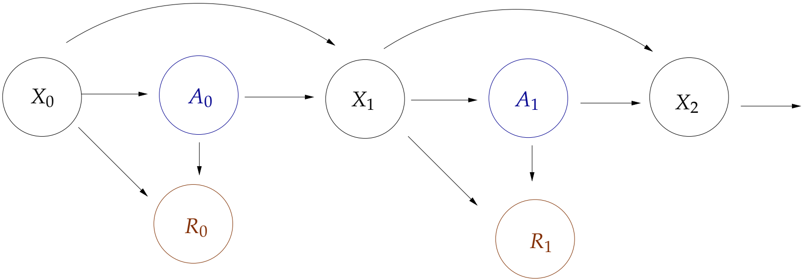


Figure: A dynamic program

Comments:

- Objective: maximize **lifetime rewards**
    - Some aggregation of $R_0, R_1, \ldots$
    - Example. $\mathbb{E}[R_0 + \beta R_1 + \beta^2 R_2 + \cdots]$ for some $\beta \in (0, 1)$
- If $T < \infty$ then the problem is called a **finite horizon** problem
- Otherwise it is called an **infinite horizon** problem
- The update rule can also depend on random elements:

$$X_{t+1} = F(X_t, A_t, \xi_{t+1})$$

Example. A retailer sets prices and manages inventories to maximize profits

- X_t measures
  - current business environment
  - the size of the inventories
  - prices set by competitors, etc.
- A_t specifies current prices and orders of new stock
- R_t is current profit π_t
- Lifetime reward is

$$\mathbb{E}\left[\pi_0 + \frac{1}{1+r}\pi_1 + \left(\frac{1}{1+r}\right)^2 \pi_2 + \cdots\right] = \text{EPV}$$

Flow of this lecture:

1. Begin with simple finite-horizon dynamic program
2. Introduce the recursive structure of dynamic programming
3. Shift to an infinite-horizon version
4. Show how the problem produces a system of nonlinear equations
5. Discuss how we can solve nonlinear equations
   - Fixed point theory

Finite-Horizon Job Search

A model of job search created by **John J. McCall**

We model the decision problem of an unemployed worker

Job search depends on

- current and likely future wage offers
- impatience, and
- the availability of unemployment compensation

We begin with a very simple version of the McCall model

(Later we consider extensions)

Set Up

An agent begins working life at time $t = 1$ without employment

Receives a new job offer paying wage w_t at each date t

She has two choices:

1. **accept** the offer and work permanently at w_t or
2. **reject** the offer, receive unemployment compensation c, and reconsider next period

Assume $\{w_t\}$ is $\overset{\text{IID}}{\sim}$ φ, where

- $\mathsf{W} \subset \mathbb{R}_+$ is a finite set of wage outcomes and
- $\varphi \in \mathcal{D}(\mathsf{W})$

The agent cares about the future but is **impatient**

Impatience is parameterized by a **time discount factor** $\beta \in (0, 1)$

- Present value of a next-period payoff of y dollars is βy

Trade off:

- $\beta < 1$ indicating some impatience
- hence the agent will be tempted to accept reasonable offers, rather than always waiting for a better one
- The key question is how long to wait

The Two Period Problem

Suppose that the working life is just two periods ($t = 1, 2$)

Let's start at $t = 2$, when w_2 is observed

backward induction – start at the end, work back

- If already employed, continue working at w_1
- If unemployed, take the max of c and w_2

Set $v_2(w_2) = \max\{c, w_2\}$ = the **time 2 value function**

- max value available for unemployed worker at $t = 2$ given w_2

Now we shift to $t = 1$

At $t = 1$, given w_1, the unemployed worker's options are

1. **accept** w_1 and receive it at $t = 1, 2$
2. **reject** it, receive compensation c, and then, at $t = 2$, get $v_2(w_2) = \max\{c, w_2\}$

Expected present value (**EPV**) of option 1 is $w_1 + \beta w_1$

- sometimes called the **stopping value**

EPV of option 2 is

$$h_1 := c + \beta \sum_{w' \in \mathsf{W}} v_2(w') \varphi(w'), \tag{1}$$

- sometimes called the **continuation value**

Decision at $t = 1$

1. Look at EPV of two choices (accept, reject)
2. Choose the one with highest EPV

Let's label the actions

$$0 := \text{reject}$$

$$1 := \text{accept}$$

Then optimal choice is

$$\mathbb{1}\{w_1 + \beta w_1 \geqslant h_1\}$$

$$=: \mathbb{1}\{\text{stopping value} \geqslant \text{continuation value}\}$$

Let

$$w_1^* := \frac{h_1}{1+\beta} := \textbf{reservation wage}$$

We have

$$\begin{aligned} w_1 \geqslant w^* \quad & \iff \quad w_1 \geqslant \frac{h_1}{1+\beta} \\ & \iff \quad w_1 + \beta w_1 \geqslant h_1 \\ & \iff \quad \text{stopping value} \geqslant \text{continuation value} \end{aligned}$$

Hence

$$\text{accept} \quad \iff \quad w_1 \geqslant w_1^*$$

The **time 1 value function** v_1 is

$$v_1(w_1) = \max \left\{ w_1 + \beta w_1, \, c + \beta \sum_{w' \in \mathsf{W}} v_2(w') \varphi(w') \right\}$$

$$= \max \{ w_1 + \beta w_1, \, h_1 \}$$

$$= \max \{ \text{stopping value}, \; \text{continuation value} \}$$

The maximum lifetime value available at $t = 1$ given

- currently unemployed
- current offer w_1

```julia
using Distributions

"Creates an instance of the job search model, stored as a NamedTuple."
function create_job_search_model(;
        n=50,        # wage grid size
        w_min=10.0,  # lowest wage
        w_max=60.0,  # highest wage
        a=200,       # wage distribution parameter
        b=100,       # wage distribution parameter
        β=0.96,      # discount factor
        c=10.0       # unemployment compensation
    )
    w_vals = collect(LinRange(w_min, w_max, n+1))
    ϕ = pdf(BetaBinomial(n, a, b))
    return (; n, w_vals, ϕ, β, c)
end

" Computes lifetime value at t=1 given current wage w_1 = w. "
function v_1(w, model)
    (; n, w_vals, ϕ, β, c) = model
    h_1 = c + β * max.(c, w_vals)'ϕ
    return max(w + β * w, h_1)
end

" Computes reservation wage at t=1. "
function res_wage(model)
    (; n, w_vals, ϕ, β, c) = model
    h_1 = c + β * max.(c, w_vals)'ϕ
    return h_1 / (1 + β)
end
```

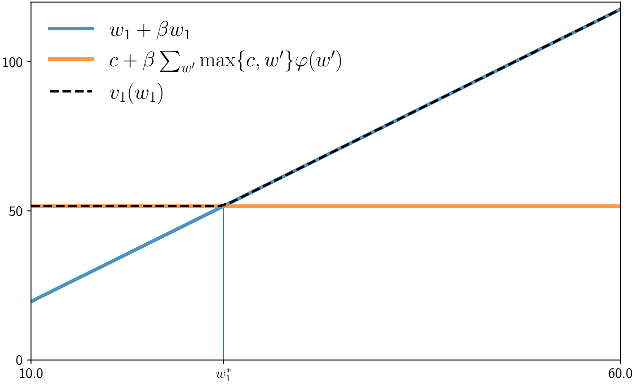


Figure: The value function v_1 and the reservation wage

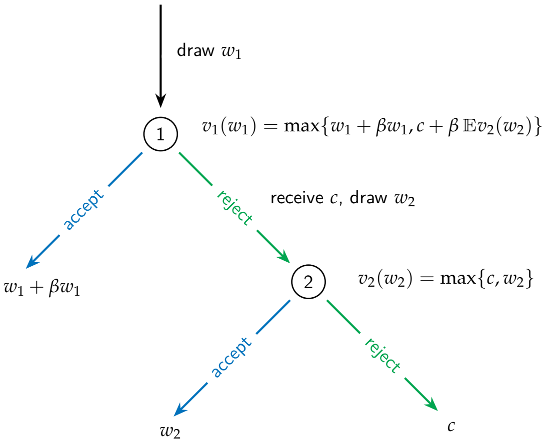


Figure: Decision tree for the two period problem

Three Period Problem

Now suppose we extend to three periods, with $t = 0, 1, 2$

At $t = 0$, the EPV of accepting w_0 is $w_0 + \beta w_0 + \beta^2 w_0$

Maximal EPV of rejecting is

1. unemployment compensation plus
2. max value we can expect from $t = 1$ when unemployed

That is,

$$\text{continuation value} = h_0 := c + \beta \sum_{w'} v_1(w') \varphi(w')$$

Putting it together,

$$v_2(w_2) = \max\{c, w_2\}$$

$$v_1(w_1) = \max\left\{w_1 + \beta w_1,\, c + \beta \sum_{w' \in \mathsf{W}} v_2(w')\varphi(w')\right\}$$

$$v_0(w_0) = \max\left\{w_0 + \beta\, w_0 + \beta^2\, w_0,\, c + \beta \sum_{w' \in \mathsf{W}} v_1(w')\varphi(w')\right\}$$

- **solve for all v_t** by **backward induction** (start from top)

From Values to Choices

Now we know the optimal values we can make optimal choices

At time $t = 0$

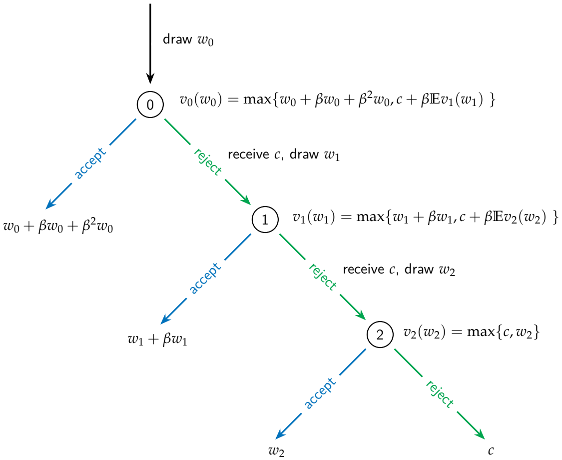
$$\text{action} \;\; = \mathbb{1}\left\{ w_0 + \beta\, w_0 + \beta^2\, w_0 \geqslant c + \beta \sum_{w' \in \mathsf{W}} v_1(w')\varphi(w') \right\}$$

At time $t = 1$, if still unemployed

$$\text{action} \;\; = \mathbb{1}\left\{ w_1 + \beta w_1 \geqslant c + \beta \sum_{w' \in \mathsf{W}} v_2(w')\varphi(w') \right\}$$

At time $t = 2$, if still unemployed

$$\text{action} \;\; = \mathbb{1}\{w_2 \geqslant c\}$$


draw w_0
0
$v_0(w_0) = \max\{w_0 + \beta w_0 + \beta^2 w_0, c + \beta \mathbb{E} v_1(w_1)\}$
accept
reject
receive c, draw w_1
$w_0 + \beta w_0 + \beta^2 w_0$
1
$v_1(w_1) = \max\{w_1 + \beta w_1, c + \beta \mathbb{E} v_2(w_2)\}$
accept
reject
receive c, draw w_2
$w_1 + \beta w_1$
2
$v_2(w_2) = \max\{c, w_2\}$
accept
reject
w_2
c

Summary

We reduced the multi-stage problem to two period problems

- the **key idea** of dynamic programming!

The equation

$$v_0(w_0) = \max \left\{ w_0 + \beta\, w_0 + \beta^2\, w_0,\ c + \beta \sum_{w' \in \mathsf{W}} v_1(w')\varphi(w') \right\}$$

is an example of a **Bellman equation**

Similiar ideas easily extend to time T:

$$v_T(w_T) = \max\{c, w_T\}$$

and

$$v_t(w_t) =$$

$$\max \left\{ w_t + \beta\, w_t + \cdots + \beta^{T-t}\, w_t,\; c + \beta \sum_{w' \in \mathsf{W}} v_{t+1}(w') \varphi(w') \right\}$$

for $t = 0, \ldots, T - 1$

Infinite Horizons

Now let us consider a worker who aims to maximize

$$\mathbb{E} \sum_{t=0}^{\infty} \beta^t Y_t, \quad Y_t \in \{c, W_t\} \text{ is earnings at time } t \tag{2}$$

- $\{W_t\} \stackrel{\text{IID}}{\sim} \varphi$ for $\varphi \in \mathcal{D}(\mathsf{W})$
- $\mathsf{W} \subset \mathbb{R}_+$ with $|\mathsf{W}| < \infty$
- c and β are positive and $\beta < 1$
- jobs are <u>permanent</u>

What is max EPV of each option when lifetime is infinite?

What if we **accept** $w \in \mathsf{W}$ now?

$$\text{EPV} \;= \text{stopping value} \;= w + \beta w + \beta^2 w + \cdots = \frac{w}{1-\beta}$$

What if we **reject**?

$$\begin{aligned}\text{EPV} \;&= \text{continuation value}\\ &= \text{EPV of } \underline{\text{optimal}} \text{ choice in each subsequent period}\end{aligned}$$

But what are optimal choices?!

Calculating optimal choice requires knowing optimal choice!

The Value Function

Let $v^*(w) :=$ max lifetime EPV given wage offer w

We call v^* the **value function**

Suppose that we know v^*

Then the (maximum) **continuation value** is

$$h^* := c + \beta \sum_{w' \in \mathsf{W}} v^*(w') \varphi(w')$$

$$= \text{ max EPV conditional on decision to continue}$$

The optimal choice is then

$$\mathbb{1} \{\text{stopping value} \geqslant \text{continuation value}\} = \mathbb{1} \left\{ \frac{w}{1-\beta}, \; h^* \right\}$$

But how can we calculate v^*?

Key idea: We can use the Bellman equation to solve for v^*

Theorem. The value function v^* satisfies the **Bellman equation**

$$v^*(w) = \max \left\{ \frac{w}{1-\beta}, c + \beta \sum_{w' \in \mathsf{W}} v^*(w') \varphi(w') \right\} \qquad (w \in \mathsf{W})$$

Intuition:

- If accept, get $w/(1-\beta)$
- If reject and then choose optimally, get max continuation value
- Max value today is max of these alternatives

Full proof coming later!

So how can we use the Bellman equation

$$v^*(w) = \max \left\{ \frac{w}{1-\beta}, \, c + \beta \sum_{w' \in \mathsf{W}} v^*(w') \varphi(w') \right\} \qquad (w \in \mathsf{W})$$

to solve for v^*?

For this we need **fixed point theory**

Fixed point theory is used to solve equations

We start begin with the linear case

Linear Equations

Given one-dimensional equation $x = ax + b$, we have

$$|a| < 1 \quad \implies \quad x^* = \frac{b}{1-a} = \sum_{k \geqslant 0} a^k b$$

How can we extend this beyond one dimension?

We define the **spectral radius** of square matrix A as

$$r(A) := \max\{|\lambda| : \lambda \text{ is an eigenvalue of } A\}$$

Key idea:

- $r(A) < 1$ is a generalization of $|a| < 1$

Neumann Series Lemma

Suppose b is a column vector in $\mathbb{R}^n$ and A is $n \times n$

Let I be the $n \times n$ identity matrix

Theorem. If $r(A) < 1$, then

1. $I - A$ is nonsingular,
2. the sum $\sum_{k \geqslant 0} A^k$ converges,
3. $(I - A)^{-1} = \sum_{k \geqslant 0} A^k$, and
4. the vector equation $x = Ax + b$ has the unique solution

$$x^* := (I - A)^{-1} b = \sum_{k \geqslant 0} A^k b$$

Intuitive idea: with $S := \sum_{k \geqslant 0} A^k$, we have

$$I + AS = I + A(I + A + \cdots) = I + A + A^2 + \cdots = S$$

Rearranging $I + AS = S$ gives $S = (I - A)^{-1}$

The equation $x = Ax + b$ is equivalent to $(I - A)x = b$

Unique solution is $x^* = (I - A)^{-1} b = Sb$, as claimed

However, still need to show that

- $\sum_{k \geqslant 0} A^k$ converges
- the matrix $I - A$ is invertible

To complete the proof, we introduce the **matrix norm**

$$\|B\|_\infty := \max_{i,j} |b_{ij}|$$

Lemma. If B is any square matrix, then

- $r(B)^k \leqslant \|B^k\|_\infty$ for all $k \in \mathbb{N}$ and
- **Gelfand's formula** holds: $\|B^k\|_\infty^{1/k} \to r(B)$ as $k \to \infty$

Ex. Prove: $r(A) < 1 \implies \sum_{k \geqslant 0} A^k$ converges

- Hint 1: Suffices to show $\lim_{N \to \infty} \| \sum_{k \geqslant 0}^{N} A^k \|_\infty < \infty$
- Hint 2: Use triangle inequality and Cauchy's root test

To complete the proof, we introduce the **matrix norm**

$$\|B\|_\infty := \max_{i,j} |b_{ij}|$$

Lemma. If B is any square matrix, then

- $r(B)^k \leqslant \|B^k\|_\infty$ for all $k \in \mathbb{N}$ and
- **Gelfand's formula** holds: $\|B^k\|_\infty^{1/k} \to r(B)$ as $k \to \infty$

Ex. Prove: $r(A) < 1 \implies \sum_{k \geqslant 0} A^k$ converges

- Hint 1: Suffices to show $\lim_{N\to\infty} \|\sum_{k\geqslant 0}^N A^k\|_\infty < \infty$
- Hint 2: Use triangle inequality and Cauchy's root test

To complete the proof, we introduce the **matrix norm**

$$\|B\|_\infty := \max_{i,j} |b_{ij}|$$

Lemma. If B is any square matrix, then

- $r(B)^k \leqslant \|B^k\|_\infty$ for all $k \in \mathbb{N}$ and
- **Gelfand's formula** holds: $\|B^k\|_\infty^{1/k} \to r(B)$ as $k \to \infty$

Ex. Prove: $r(A) < 1 \implies \sum_{k \geqslant 0} A^k$ converges

- Hint 1: Suffices to show $\lim_{N \to \infty} \| \sum_{k \geqslant 0}^N A^k \|_\infty < \infty$
- Hint 2: Use triangle inequality and Cauchy's root test

Final step: Show that $(I - A)^{-1}$ exists:

- Suffices to show existence of a right inverse
    - See, e.g., §6.1.4.5 of networks.quantecon.org
- That is, we need an S such that $(I - A)S = I$
- Let $S = \sum_{k \geqslant 0} A^k$

We have

$$(I - A)S = I \sum_{k \geqslant 0} A^k - A \sum_{k \geqslant 0} A^k = \sum_{k \geqslant 0} A^k - \sum_{k \geqslant 1} A^k = I$$

Hence $(I - A)^{-1}$ exists and equals $\sum_{k \geqslant 0} A^k$

Fixed Points

To solve more complex equations we use **fixed point theory**

Recall that, if S is any set then

- T is a **self-map** on S if T maps S into itself
- $x^* \in S$ is called a **fixed point** of T in S if $Tx^* = x^*$

Example. Every x in set S is fixed under the **identity map**

$$I \colon x \mapsto x$$

Example. If $S = \mathbb{N}$ and $Tx = x + 1$, then T has no fixed point

Example. If $S = \mathbb{R}$ and $Tx = x^2$, then T has fixed points at $0, 1$

Example. If $S = \mathbb{R}^n$ and $Tx = Ax + b$, then

$r(A) < 1 \implies x^* := (I - A)^{-1} b$ is the unique f.p. of T in S

Example. If $S \subset \mathbb{R}$, $Tx = x \iff T$ meets the 45 degree line

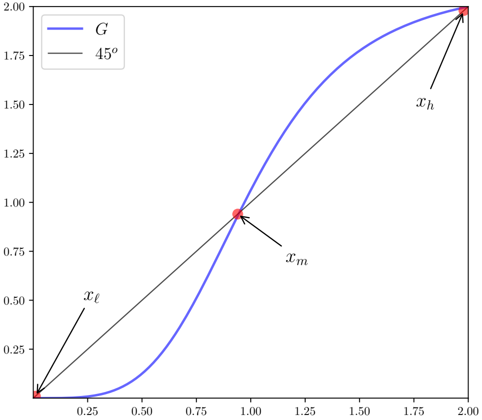


Figure: Graph and fixed points of $G\colon x \mapsto 2.125/(1 + x^{-4})$

Given self-map T on S, common to

- write Tx instead of $T(x)$ and
- call T an **operator** rather than a function

Key idea:

$$\text{solving equation } x = Tx \iff \text{finding fixed points of } T$$

Example. If $S = \mathbb{R}^n$ and $Tx = Ax + b$, then

$$x^* \text{ solves equation } x = Ax + b \iff x^* \text{ is a fixed point of } T$$

(But fixed point theory is mainly for nonlinear equations)

Point on notation:

- $T^2 = T \circ T$
- $T^3 = T \circ T \circ T$
- etc.

Example. $Tx = Ax + b$ implies $T^2 x = A(Ax + b) + b$

Lemma Let S be any set and let T be a self-map on S. If

$$\exists\, \bar{x} \in S,\ m \in \mathbb{N} \ \text{ s.t. } \ T^k x = \bar{x} \text{ for all } x \in S \text{ and } k \geqslant m$$

then $\bar{x}$ is the unique fixed point of T in S.

Proof of uniqueness:

Let x and y be any two two fixed points of T in S

Since $T^m x = \bar{x}$ and $T^m y = \bar{x}$, we have $T^m x = T^m y$

But x and y are fixed points, so

$$x = T^m x \text{ and } y = T^m y$$

We conclude that $x = y$, so uniqueness holds

Proof of existence:

We claim that $\bar{x}$ is a fixed point

To see this, recall that

$$T^k x = \bar{x} \text{ for } k \geqslant m \text{ and all } x \in S$$

Hence $T^m \bar{x} = \bar{x}$ and $T^{m+1} \bar{x} = \bar{x}$

But then

$$T\bar{x} = T(T^m \bar{x}) = T^{m+1} \bar{x} = \bar{x}$$

That is, $\bar{x}$ is a fixed point of T

Let T be a self-map on $S \subset \mathbb{R}^d$

Ex. Prove the following: If

1. $T^m u \to u^*$ as $m \to \infty$ for some pair $u, u^* \in S$ and
2. T is continuous at u^*

then u^* is a fixed point of T

Answer: Assume hypotheses and let $u_m := T^m u$ for all $m \in \mathbb{N}$

By continuity and $u_m \to u^*$ we have $Tu_m \to Tu^*$

But $(Tu_m)_{m \geqslant 1}$ is just $(u_2, u_3, \ldots)$

Since $u_m \to u^*$, we just have $Tu_m \to u^*$

Limits are unique, so $u^* = Tu^*$

Self-map T is called **globally stable** on S if

1. T has a unique fixed point x^* in S and
2. $T^k x \to x^*$ as $k \to \infty$ for all $x \in S$

Example. If $S = \mathbb{R}^n$ and $Tx = Ax + b$, then

$$T^k x = A^k x + A^{k-1} b + A^{k-2} b + \cdots + Ab + b \qquad (x \in S, k \in \mathbb{N})$$

If $r(A) < 1$, then $A^k x \to 0$ and $\sum_{i=0}^{k} A^i \to (I - A)^{-1}$, so

$$\lim_{k \to \infty} T^k x = \lim_{k \to \infty} \left[A^k x + \sum_{i=0}^{k} A^{i-1} b \right] = (I - A^{-1}) b = x^*$$

Example. Consider Solow–Swan growth dynamics

$$k_{t+1} = g(k_t) := sAk_t^{\alpha} + (1-\delta)k_t, \qquad t = 0, 1, \ldots,$$

where

- k_t is capital stock per worker,
- A, α are production parameters,
- $s > 0$ is a savings rate, and
- $\delta \in (0, 1)$ is a rate of depreciation

Iterating with g from k_0 generates a time path for capital stock

The map g is globally stable on $(0, \infty)$

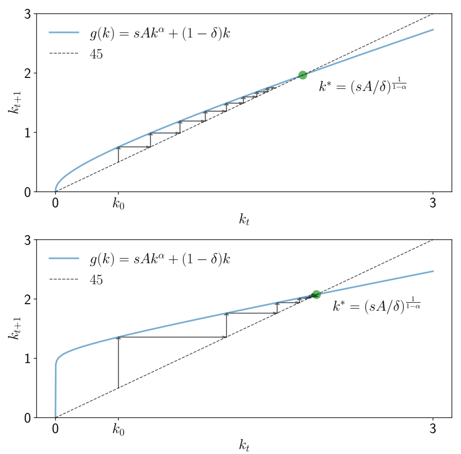


Figure: Global stability for the Solow–Swan model

Note from last slide

- If g is flat near k^*, then $g(k) \approx k^*$ for k near k^*
- A flat function near the fixed point $\implies$ fast convergence

Conversely

- If g is close to the 45 degree line near k^*, then $g(k) \approx k$
- Close to 45 degree line means high persistence, slow convergence

Let T be a self-map on $S \subset \mathbb{R}^n$.

We call $C \subset S$ **invariant** for T if

$$u \in C \implies Tu \in C$$

Lemma. If T is globally stable on $S \subset \mathbb{R}^n$ with fixed point u^* and C is nonempty, closed and invariant for T, then $u^* \in C$

Proof: Let the stated hypotheses hold and fix $u \in C$

By global stability we have $T^k u \to u^*$

Since T is invariant on C we have $(T^k u)_{k \in \mathbb{N}} \subset C$

Since C is closed, this implies that the limit is in C

Hence $u^* \in C$, as claimed

Given a self-map T on S, we typically ask

- Does T have at least one fixed point on S (existence)?
- Does T have at most one fixed point on S (uniqueness)?
- How can we compute fixed points of T?

For the last question, we seek an algorithm

Then we investigate its properties

Successive Approximation

A natural algorithm for approximating the fixed point in S:

---

fix x_0 and $k = 0$
while *some stopping condition fails* **do**
  $x_{k+1} \leftarrow T x_k$
  $k \leftarrow k + 1$
end
return x_k

If T is globally stable on S, then $(x_k) = (T^k x_0)$ converges to x^*

$$\text{hence output} \approx x^*$$

The algorithm just described is called **successive approximation**

```julia
function successive_approx(T,                    # Operator (callable)
                           x_0;                  # Initial condition
                           tolerance=1e-6,       # Error tolerance
                           max_iter=10_000,      # Max iteration bound
                           print_step=25)        # Print at multiples
    x = x_0
    error = Inf
    k = 1
    while (error > tolerance) & (k <= max_iter)
        x_new = T(x)
        error = maximum(abs.(x_new - x))
        if k % print_step == 0
            println("Completed iteration $k with error $error.")
        end
        x = x_new
        k += 1
    end
    if k < max_iter
        println("Terminated successfully in $k iterations.")
    else
        println("Warning: Iteration hit max_iter bound $max_iter.")
    end
    return x
end
```

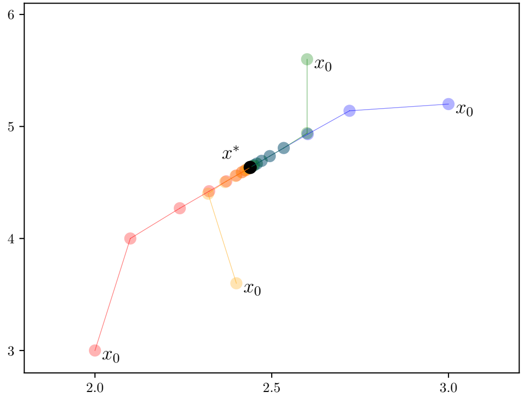


Figure: Successive approximation from different initial conditions

Newton's Method

Let h be a differentiable real-valued function on $(a, b) \subset \mathbb{R}$

We seek a **root** of h, which is an x^* such that $h(x^*) = 0$

We start with guess x_0 and then update it

To do this we use $h(x_1) \approx h(x_0) + h'(x_0)(x_1 - x_0)$

Setting the RHS $= 0$ and solving for x_1 gives

$$x_1 = x_0 - \frac{h(x_0)}{h'(x_0)}$$

Continuing in the same way, we set

$$x_{k+1} = q(x_k) \quad \text{where} \quad q(x) := x - \frac{h(x)}{h'(x)},$$

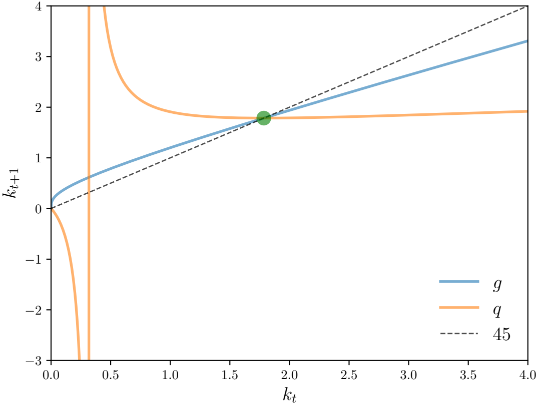


Figure: Successive approximation vs Newton's method

Comments:

- The map q is flat close to the fixed point k^*
- Hence Newton's method converges quickly near k^*
- But Newton's method is not globally convergent
- Successive approximation is slower but more robust

Key ideas

- There is almost always a trade-off between robustness and speed
- Speed requires assumptions, and assumptions can fail

Newton's method extends naturally to **multiple dimensions**

When h is a map from $S \subset \mathbb{R}^n$ to itself, we use

$$x_{k+1} = x_k - [J(x_k)]^{-1} h(x_k)$$

Here $J_h(x_k) :=$ the Jacobian of h evaluated at x_k

Comments

- Typically faster but less robust
- Matrix operations can be parallelized
- Automatic differentiation can be helpful

Norms in Vector Space

We want to use fixed point theory in $\mathbb{R}^n$

For this purpose it will be helpful to study alternative norms on $\mathbb{R}^n$

A function $\|\cdot\|\colon \mathbb{R}^n \to \mathbb{R}$ is called a **norm** on $\mathbb{R}^n$ if, for any $\alpha \in \mathbb{R}$ and $u, v \in \mathbb{R}^n$,

(a) $\|u\| \geqslant 0$

(b) $\|u\| = 0 \iff u = 0$

(c) $\|\alpha u\| = |\alpha| \|u\|$ and

(d) $\|u + v\| \leqslant \|u\| + \|v\|$

Example. The **Euclidean norm** $\|u\| := \sqrt{\langle u, u\rangle}$ obeys (a)–(d)

Example. The ℓ_1 **norm** of a vector $u \in \mathbb{R}^n$ is defined by

$$u = (u_1, \ldots, u_n) \mapsto \|u\|_1 := \sum_{i=1}^{n} |u_i|$$

Example. The **supremum norm**, defined by

$$\|u\|_\infty := \max_{i=1}^{n} |u_i|$$

is also a norm on $\mathbb{R}^n$

Ex. Verify that

1. the ℓ_1 norm on $\mathbb{R}^n$ satisfies (a)–(d) above
2. the supremum norm on $\mathbb{R}^n$ satisfies (a)–(d) above

Equivalence of Norms

Let u and $(u_m) := (u_m)_{m \in \mathbb{N}}$ be elements of $\mathbb{R}^n$

We say that (u_m) **converges** to u and write $u_m \to u$ if

$$\|u_m - u\| \to 0 \text{ as } m \to \infty \text{ for some norm } \|\cdot\| \text{ on } \mathbb{R}^n$$

Do we need to say "convergence <u>with respect to</u> $\|\cdot\|$"?

No because any two norms $\|\cdot\|_a$ and $\|\cdot\|_b$ on $\mathbb{R}^n$ are **equivalent**

That is, for any such pair, $\exists\, M, N$ such that

$$M\|u\|_a \leqslant \|u\|_b \leqslant N\|u\|_a \quad \text{for all } u \in \mathbb{R}^n$$

- See, e.g., Kreyszig (1978)

Hence convergence is independent of the norm

Ex. Let $\|\cdot\|_a$ and $\|\cdot\|_b$ be any two norms on $\mathbb{R}^n$

Given u in $\mathbb{R}^n$ and a sequence (u_m) in $\mathbb{R}^n$, confirm that

$$\|u_m - u\|_a \to 0 \text{ implies } \|u_m - u\|_b \to 0 \text{ as } m \to \infty$$

<u>Proof</u>: Let $\|\cdot\|_a$, $\|\cdot\|_b$, u and (u_m) be as stated

We can find an $M \in \mathbb{R}$ with

$$0 \leqslant \|u_m - u\|_b \leqslant M\|u_m - u\|_a \text{ for all } m \in \mathbb{N}$$

Since $\|u_m - u\|_a \to 0$, we also have $\|u_m - u\|_b \to 0$

Contractions

Let

- U be a nonempty subset of $\mathbb{R}^n$,
- $\| \cdot \|$ be a norm on $\mathbb{R}^n$, and
- T be a self-map on U

T is called a **contraction** on U with respect to $\| \cdot \|$ if

$$\exists\, \lambda < 1 \text{ such that } \|Tu - Tv\| \leqslant \lambda \|u - v\| \quad \text{for all} \quad u, v \in U$$

Example. $Tx = ax + b$ is a contraction on $\mathbb{R}$ with respect to $| \cdot |$ if and only if $|a| < 1$

Indeed,

$$|Tx - Ty| = |ax + b - ay - b| = |a||x - y|$$

Ex. Prove: If T is a contraction on U with respect to any norm, then

1. T is continuous on U and
2. T has at most one fixed point in U

Let's check part 2 under the stated hypotheses

If u, v are fixed points of T in U, then

$$\|u - v\| = \|Tu - Tv\| \leqslant \lambda \|u - v\| \quad \text{for some } \lambda < 1$$

$$\therefore \quad \|u - v\| = 0$$

$$\therefore \quad u = v$$

Banach's Contraction Mapping Theorem

Theorem If

1. U is closed in $\mathbb{R}^n$ and
2. T is a contraction of modulus λ on U with respect to some norm $\|\cdot\|$ on $\mathbb{R}^n$,

then T has a unique fixed point u^* in U and

$$\|T^n u - u^*\| \leqslant \lambda^n \|u - u^*\| \quad \text{for all } n \in \mathbb{N} \text{ and } u \in U$$

In particular, T is globally stable on U

Proof: See the course notes

Infinite-Horizon Job Search

Let's now return to the job search problem

Recall that that the value function v^* solves the Bellman equation

That is,

$$v^*(w) = \max \left\{ \frac{w}{1-\beta}, c + \beta \sum_{w' \in \mathsf{W}} v^*(w') \varphi(w') \right\} \qquad (w \in \mathsf{W})$$

The infinite-horizon **continuation value** is defined as

$$h^* := c + \beta \sum_{w'} v^*(w') \varphi(w')$$

Key question: how to solve for v^*?

We introduce the **Bellman operator**, defined at $v \in \mathbb{R}^{\mathsf{W}}$ by

$$(Tv)(w) = \max \left\{ \frac{w}{1-\beta}, \, c + \beta \sum_{w' \in \mathsf{W}} v(w') \varphi(w') \right\} \qquad (w \in \mathsf{W})$$

By construction, $Tv = v \iff v$ solves the Bellman equation

Let $\mathcal{V} := \mathbb{R}_+^{\mathsf{W}}$

Proposition. T is a contraction $\mathcal{V}$ with respect to $\| \cdot \|_\infty$

In the proof, we use the elementary bound

$$|\alpha \vee x - \alpha \vee y| \leqslant |x - y| \qquad (\alpha, x, y \in \mathbb{R})$$

Fixing f, g in $\mathcal{V}$ fix any $w \in \mathsf{W}$, we have

$$|(Tf)(w) - (Tg)(w)| \leqslant \left| \beta \sum_{w'} f(w')\varphi(w') - \beta \sum_{w'} g(w')\varphi(w') \right|$$

$$= \beta \left| \sum_{w'} [f(w') - g(w')]\varphi(w') \right|$$

Applying the triangle inequality,

$$|(Tf)(w) - (Tg)(w)| \leqslant \beta \sum_{w'} |f(w') - g(w')|\varphi(w') \leqslant \beta \|f - g\|_\infty$$

$$\therefore \quad \|Tf - Tg\|_\infty \leqslant \beta \|f - g\|_\infty$$

Recall: The optimal decision at any given time, facing current wage draw $w \in \mathsf{W}$, is

$$\mathbb{1}\left\{\frac{w}{1-\beta} \geqslant h^*\right\}$$

Let's try to write this in the language of dynamic programming

Dynamic programming centers around the problem of finding optimal **policies**

Optimal Policies

In general, for a dynamic program, choices consist of a sequence $(A_t)_{t \geqslant 0}$

- specifies how the agent acts at each t

Since agents are not clairvoyant, so we assume that A_t cannot depend on future events

In other words, for some function σ_t,

$$A_t = \sigma_t(X_t, A_{t-1}, X_{t-1}, A_{t-2}, X_{t-2}, \ldots A_0, X_0)$$

In dynamic programming, σ_t is called a **policy function**

Key idea Design the state such that X_t is

- sufficient to determine the optimal current action
- but not so large as to be unmanagable

- Finding the state is an art!

Example. Recall retailer who chooses stock orders and prices in each period

What to include in the current state?

- level of current inventories
- interest rates and inflation?
- the rate at which inventories have changed?
- competitors prices?

So suppose state X_t determines the current action A_t

Then we can write $A_t = \sigma(X_t)$ for some function σ

Note that we dropped the time subscript on σ

No loss of generality: can include time in the current state

- i.e., expand X_t to $\hat{X}_t = (t, X_t)$

Depends on the problem at hand

- For the job search model with finite horizon, the date matters
- For the infinite horizon version of the problem, however, the agent always looks forward toward an infinite horizon

For job search model,

- state = current wage offer and
- possible actions are accept (1) or reject (0)

A policy is a map σ from W to $\{0, 1\}$

Let Σ be the set of all such maps

For each $v \in \mathcal{V}$, let us define a **v-greedy policy** to be a $\sigma \in \Sigma$ satisfying

$$\sigma(w) = \mathbb{1}\left\{\frac{w}{1-\beta} \geqslant c + \beta \sum_{w' \in \mathsf{W}} v(w')\varphi(w')\right\} \quad \text{for all } w \in \mathsf{W}$$

Accepts iff $w/(1-\beta) \geqslant$ continuation value computed using v

Optimal choice:

- agent should adopt a v^*-greedy policy
- Sometimes called **Bellman's principle of optimality**

We can also express a v^*-greedy policy via

$$\sigma^*(w) = \mathbb{1}\{w \geqslant w^*\} \quad \text{where} \;\; w^* := (1-\beta)h^* \tag{3}$$

The term w^* in (3) is called the **reservation wage**

- Same ideas as before, different language
- We prove optimality more carefully later

Computation

Since T is globally stable on $\mathcal{V}$, we can compute an approximate optimal policy by

1. applying successive approximation on T to compute v^*
2. calculate a v^*-greedy policy

In dynamic programming, this approach is called **value function iteration**

input $v_0 \in \mathcal{V}$, an initial guess of v^*
input τ, a tolerance level for error
$\varepsilon \leftarrow \tau + 1$
$k \leftarrow 0$
while $\varepsilon > \tau$ **do**
    for $w \in$ W **do**
        $v_{k+1}(w) \leftarrow (Tv_k)(w)$
    end
    $\varepsilon \leftarrow \|v_k - v_{k+1}\|_\infty$
    $k \leftarrow k + 1$
end
Compute a v_k-greedy policy σ
return σ

```julia
include("two_period_job_search.jl")
include("s_approx.jl")

" The Bellman operator. "
function T(v, model)
    (; n, w_vals, ϕ, β, c) = model
    return [max(w / (1 - β), c + β * v'ϕ) for w in w_vals]
end

" Get a v-greedy policy. "
function get_greedy(v, model)
    (; n, w_vals, ϕ, β, c) = model
    σ = w_vals ./ (1 - β) .>= c .+ β * v'ϕ  # Boolean policy vector
    return σ
end

" Solve the infinite-horizon IID job search model by VFI. "
function vfi(model=default_model)
    (; n, w_vals, ϕ, β, c) = model
    v_init = zero(model.w_vals)
    v_star = successive_approx(v -> T(v, model), v_init)
    σ_star = get_greedy(v_star, model)
    return v_star, σ_star
end
```

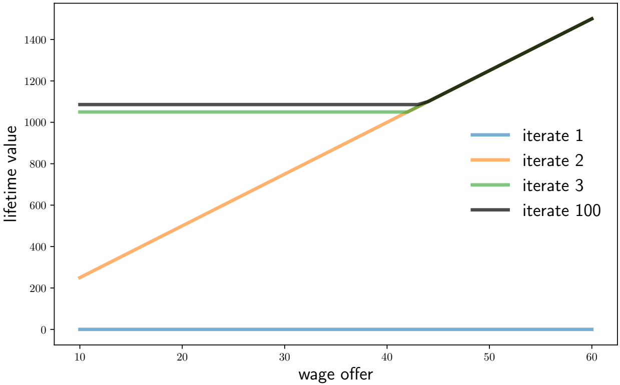


Figure: A sequence of iterates of the Bellman operator

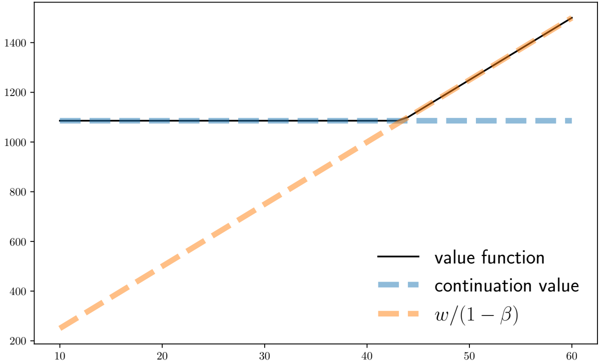


Figure: The approximate value function for job search

Computing the Continuation Value Directly

We used a standard dynamic programming approach to solve this problem

Sometimes we can find more efficient ways to solve particular problems

For the infinite horizon job search problem, a more efficient way exists

The idea is to compute the continuation value directly

This shifts the problem from n-dimensional to one-dimensional

Method: Recall that

$$v^*(w) = \max\left\{\frac{w}{1-\beta}, \, c + \beta \sum_{w'} v^*(w')\varphi(w')\right\} \qquad (w \in \mathsf{W})$$

Using the definition of h^*, we can write

$$v^*(w') = \max\left\{w'/(1-\beta), \, h^*\right\} \qquad (w' \in \mathsf{W})$$

Take expectations, multiply by β and add c to obtain

$$h^* = c + \beta \sum_{w'} \max\left\{\frac{w'}{1-\beta}, \, h^*\right\} \varphi(w')$$

How to find h^* from the equation

$$h^* = c + \beta \sum_{w'} \max \left\{ \frac{w'}{1-\beta}, h^* \right\} \varphi(w') \tag{4}$$

We introduce the map $g \colon \mathbb{R}_+ \to \mathbb{R}_+$ defined by

$$g(h) = c + \beta \sum_{w'} \max \left\{ \frac{w'}{1-\beta}, h \right\} \varphi(w')$$

By construction, h^* solves (4) if and only if h^* is a fixed point of g

Ex. Show that g is a contraction map on $\mathbb{R}_+$

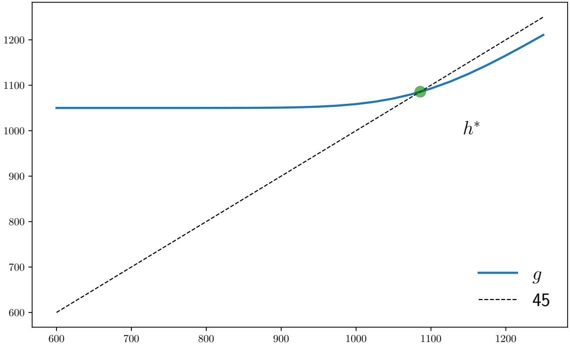


Figure: Computing the continuation value as the fixed point of g

New algorithm:

1. Compute h^* via successive approximation on g
    - Iteration in $\mathbb{R}$, not $\mathbb{R}^n$
2. Optimal policy is

$$\sigma^*(w) = \mathbb{1}\left\{\frac{w}{1-\beta} \geqslant h^*\right\}$$