Finance 937
Solving the Firm's Problem: Methods and Ideas

Joao Gomes
The Wharton School

September 23, 2019

Problem of the Firm in Discrete Time

The problem of the firm is as follows

$$
\begin{aligned}
v(a_0, k_0) &= \max_{\{k_{t+1}, i_t\}_{t=0}^{\infty}} \mathrm{E}_0 \sum_{t=0}^{\infty} M_{0,t} d_t \\
s.t.\ \ i_t &= k_{t+1} - (1-\delta) k_t \\
d_t &= \pi(a_t, k_t) - i_t - \Phi(i_t, k_t)
\end{aligned}
$$

- As before we assume financing does not matter (for now).

If both $\pi(a_t, k_t)$ and $\Phi(i_t, k_t)$ are continuously differentiable, the optimal first order conditions are:

$$
\begin{aligned}
q_t &= 1 + \Phi_i(i_t, k_t) \\
q_t &= E_t M_{t,t+1} \left[\pi_k(a_{t+1}, k_{t+1}) - \Phi_k(i_{t+1}, k_{t+1}) + (1-\delta) q_{t+1} \right]
\end{aligned}
$$

Some General Computational Issues

Two main issues

- Closed form solutions are not generally available
- General adjustment costs may not always be differentiable

We need to develop robust numerical methods to solve general versions of this problem.

Dynamic Programming: The Bellman Equation

Rewrite the value of the firm starting in any period t as:

$$v(a_t, k_t) = \max_{\left\{k_{t+j+1}, i_{t+j}\right\}_{j=0}^{\infty}} \mathrm{E}_t \sum_{j=0}^{\infty} M_{t,t+j} d(a_{t+j}, k_{t,t+j}, i_{t+j})$$

Then we can construct the following one period optimization problem for period 0 (and any period t after that):

$$v(a_0, k_0) = \max_{k_1, i_0} \left[d(a_0, k_0, i_0) + \mathrm{E}_0 M_{0,1} V(a_1, k_1)\right]$$

As above, this problem is also subject to the capital accumulation constraint.

Dynamic Program

The previous problem applies to any period t

- Drop time subscripts and use k' and a' to denote next period values of k and a:

$$v(a,k) = \max_{k',i} \left[d(a,k,i) + \mathrm{E}_a M(a,a') v(a',k') \right]$$

Where we define:

$$\begin{aligned} d(a,k,i) &= \pi(a,k) - i - \Phi(i,k) \\ \mathrm{E}_a &= \mathrm{E}[\cdot|a] \\ M_{t,t+1} &= M(a,a') \end{aligned}$$

Note

- This allows the discount factor to be stochastic and correlated with the firm's cash flows
- The firm can face aggregate or **systematic risk**.

Notation and Terminology

The full constrained optimization problem:

$$v(a,k) = \max_{k',I,q} \left[d(a,k,i) + \mathrm{E}_a M(a,a') v(a',k') + q\left(i + (1-\delta)k - k'\right) \right]$$

States, co-states and controls:

- k is the **endogenous state** variable and a is the **exogenous state** variable. They condition the optimization problem.
- q is the **co-state** variable. The marginal value, or **shadow price** of the endogenous state
- i is the **control** variable. It affects the evolution of the state but it can be computed from a static optimization

Policy functions:

- The solutions to the problem are the **optimal policy functions** $k'(a,k)$ and $i(a,k)$
- Like the **value function**, $v(a,k)$, they depend only on state variables.

Solution Approaches

Two basic alternatives to solve the problem of the firm

- Value Function Iteration (VFI) - directly computes $v(a, k)$ and uses it to obtain the optimal policy functions
  - Focuses on solving the Bellman equation directly
- Policy Function Iteration (PFI) - computes the optimal policies directly
  - Often relies on the first order conditions alone

Next we need to choose whether to deploy a global or local (perturbation) algorithm

- A local method is often inappropriate for firm level problems
- The value function may not be smooth, or at least concave
- For example with discontinuous adjustment costs, or when firms have the option to exit (later)

Some Issues with Methods

PFI is generally much faster:

- But the additional assumptions of differentiability and concavity are not always satisfied so we often can not use it.
- It is also usually very sensitive, as it relies on non-linear equation solvers.

VFI is extremely robust and can solve virtually any (well defined) dynamic programming problem

- But it can be slow and subject to a curse of dimensionality
- It relies on non-linear optimization, usually using discrete grids

The best approach is to first characterize the problem first and then choose the more suitable method.

Value Function Iteration: Basic Ideas

We focus on the more common and robust method

- The first step is to characterize the problem to ensure that is well behaved - details in Fnce 924 and Stokey and Lucas (1989)
- The Bellman equation needs to satisfy monotonicity and discounting
- The existence of a solution to the maximum problem, usually requires continuity and a compact of the choice set
- The transition function between a_t and a_{t+1} needs to be continuous

Value Function Iteration: Initial Guess

The first step is to provide a guess for next period's or terminal value function $v^0(a, k)$

- We know that the value function generally inherits the properties of $d(a, k, i)$

This usually implies that the value function is:

- Continuous and increasing in k,
- Increasing in a,
- Concave and differentiable in k **if** $d(a, k, i)$ also satisfies these properties and the choice set is convex.

Value Function Iteration: Initial Guess

Combined with our earlier results this means that a good first guesses for $v^0(a, k)$ is:

- $v^0(a, k) = k$ or
- $v^0(a, k) = \gamma\pi(a, k) + k$.

where γ captures the effects of both discounting and decreasing returns to scale.

Remember

- A **good first guess** is one of the most important steps in numerical work.

Value Function Iteration: Updating

Next, construct a revised guess for the value function, $v^1(a,k)$, by solving the problem:

$$\begin{aligned} v^1(a,k) &= \max_{k',i,q}\left[d(a,k,i) + \mathrm{E}_a M(a,a')v^0(a',k') \right. \\ &\qquad \left. + q\left(I + (1-\delta)k - k'\right)\right] \end{aligned}$$

- This is guaranteed to exist because the right hand side is continuous and the choice set is compact.
- The maximization may not be solved using FOC, but (global) value function iteration always works.
- Her we will only discuss the easy and reliable (but slow) grid method.
- For advanced, projection based methods, refer to Jesus's excellent semester long class in the Econ department.

Value Function Iteration: Convergence

After we obtain the updated value function $v^1(a,k)$:

- We repeat this iterative procedure so that at every step n

$$v^{n+1}(a,k) = \max_{k',I} \Big[d(a,k,i) + \mathrm{E}_a M(a,a') v^n(a',k') + q\left(i + (1-\delta)k - k'\right)\Big] = T(v^n(a,k))$$

- Under the assumptions above we expect that this method will converge so that $v = T(v)$.
- We stop when the distance, $d^n = |v^{n+1}(k) - v^n(k)| < \varepsilon$,

Value Function Iteration: Grid Method

At each point in time the problem is solved at discrete nodes for the state variable $k \in \{k_1, k_2, ...k_{nk}\}$ and $a \in \{a_1, a_2, ...a_{na}\}$.

- The number of points on the grid is be a key element in determining the speed of the computation.
- It is **very important** to solve the problem over the region where the firm is likely to spend its time.
- This requires looking at the grid range and where it is centered

Grid for the Endogenous State

The grid should be centered around "average" value of the state variable.

- The optimal "long run" value for k obeys the Jorgenson, user cost, relation:

$$1 = \mathrm{E}M' \left[\pi_k(a', k') + (1 - \delta) \right]$$

  - In the long run there are no adjustment costs
  - This is what we expect long run capital to converge to and where the firm should spend more time

- With Cobb-Douglas, and average discount rate $\bar{r}$, the center of the grid should be set at:

$$\bar{k} = \left[\frac{\alpha \mathrm{E}[a]}{\bar{r} + \delta} \right]^{1/(1-\alpha)}$$

- Numerical accuracy is often improved by picking $\mathrm{E}[a]$ so that $\bar{k} = 1$

Grid for the Endogenous State

The grid maximum value will never exceed looking at the maximum sustainable capital level in steady-state:

- Set $k'_{nk} = k_{nk}$
- Set $d = 0$ - the firm will not operate if dividends are sustainable negative and use the firm's resource constraint to get:

$$\pi(a_{na}, k_{nk}) - \delta k_{nk} = 0$$

  - With Cobb-Douglas

$$k_{nk} = \left[\frac{a_{na}}{\delta}\right]^{1/(1-\alpha)}$$

The minimum level , k_1, could be set to 0 or the Jorgensonian level, at the lowest productivity and discount factor (highest interest rate):

$$1 = \mathrm{E} M_1 \left[\pi_k(a_1, k_1) + (1-\delta)\right]$$

However both grid bounds **should** be much tighter in many problems.

Grid for the Exogenous State

Generally we assume that uncertainty about the exogenous state a is driven by a first order Markov process.

- This is not a very restrictive assumption since any n-order Markov process can be be reduced to a first-order process.
- For example:

  $$a_t = \rho_1 a_{t-1} + \rho_2 a_{t-2} + \epsilon_t$$

  can be transformed into:

  $$\begin{bmatrix} a_t \\ a_{t-1} \end{bmatrix} = \begin{bmatrix} \rho_1 & \rho_2 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} a_{t-1} \\ a_{t-2} \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} \epsilon_t$$

  which is a VAR(1) process.

Grid for the Exogenous State

We the usually want to approximate the continuous stochastic process:

$$a' = (1-\rho)\bar{a} + \rho a + \varepsilon, \text{ with } \varepsilon \sim N(0,\sigma)$$

with a Markov chain on a discrete grid for $a = \{a_1, a_2, ...a_{na}\}$.

- Specifically, we want to the transition matrix P satisfying:

$$a' = \begin{bmatrix} a_1 \\ a_2 \\ \\ a_{na} \end{bmatrix} = \begin{bmatrix} p_{1,1} & p_{1,2} & & p_{1,na} \\ p_{2,1} & p_{2,2} & & p_{2,na} \\ \\ p_{na,1} & p_{na,na} & & p_{na,na} \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \\ a_{na} \end{bmatrix} = Pa$$

- where $p_{i,j} = P[a' = a_i | a = a_j]$ is the probability of moving from state j today to state i tomorrow.

A 5-state Markov Chain

$$P = \begin{bmatrix} 0.7376 & 0.1947 & 0.0113 & 0.0001 & 0.0000 \\ 0.2473 & 0.5555 & 0.2221 & 0.0169 & 0.0002 \\ 0.0150 & 0.2328 & 0.5333 & 0.2328 & 0.0150 \\ 0.0002 & 0.0169 & 0.2221 & 0.5555 & 0.2473 \\ 0.0000 & 0.0001 & 0.0113 & 0.1947 & 0.7376 \end{bmatrix}$$

In this case $p_{3,1}$ is probability of moving from state 1 today to state 3 tomorrow and

$$\sum_j p_{i,j} = 1$$

Grid for the Exogenous State: Tauchen (1986)

Let $\Delta a = a_j - a_{j-1}$ be the **constant** interval between grid points.

- The upper and lower bound are set to m unconditional standard deviations on either side of the unconditional mean $\bar{a}$:

$$\begin{aligned} a_1 &= \bar{a} - m\frac{\sigma}{\sqrt{1-\rho^2}} \\ a_{na} &= \bar{a} + m\frac{\sigma}{\sqrt{1-\rho^2}} \end{aligned}$$

- For this to be sensible we should have an **odd** number of grid points and $m = (na-1)/2$
- Thus $a_{m+1} = \bar{a}$

Grid for the Exogenous State: Tauchen (1986)

Compute the transition probabilities as:

$$p_{i,j} = P[a' < a_i + \Delta a/2 | a = a_j] - P[a' < a_i - \Delta a/2 | a = a_j]$$

Given normality this is equal to:

$$p_{i,j} = \int_{a_i - 0.5\Delta a}^{a_i + 0.5\Delta a} N\left(\frac{a' - (1-\rho)\bar{a} - \rho a_j}{\sigma}\right) \Delta a'$$

Adjust at the boundaries to ensure probabilities sum to 1

$$p_{1,j} = N\left(\frac{a_1 + 0.5\Delta a - (1-\rho)\bar{a} - \rho a_j}{\sigma}\right)$$

$$p_{na,j} = 1 - N\left(\frac{a_{na} - 0.5\Delta a - (1-\rho)\bar{a} - \rho a_j}{\sigma}\right)$$

Unless the stochastic process is **very** persistent, an approximation with $na = 9$ points works very well.

Grid for the Exogenous State: Tauchen and Hussey (1991)

By choosing the grid points a_i better we can improve the approximation - and reduce the grid size

- Gauss-Hermite quadrature methods - approximate the numerical integral

$$\int_{a_i-0.5\Delta a}^{a_i+0.5\Delta a} f\left(a\right)\Delta a = \sum_{i=1}^{na} \omega_i f(a_i)$$

- Generally we get an exact approximation if $f(a)$ is a polynomial of order $2na - 1$
- With 5 nodes we get exact answer if $f(a)$ is a 9th-order polynomial!

Implementing Quadrature Methods

Scale productivity so that $[a_i - 0.5\Delta a, a_i + 0.5\Delta a] = [-1, 1]$. How to get the nodes and weights to approximate:

$$\int_{-1}^{1} f(a)\Delta a \approx \sum_{i=1}^{na} \omega_i f(a_i)$$

- Right answer for $f(a) = 1$

$$\int_{-1}^{1} 1\Delta a = \sum_{i=1}^{na} \omega_i 1$$

- Right answer for $f(a) = a$

$$\int_{-1}^{1} a\Delta a = \sum_{i=1}^{na} \omega_i a_i$$

- Right answer for $f(a) = a^q$

$$\int_{-1}^{1} a^q\Delta a = \sum_{i=1}^{na} \omega_i a_i^q$$

Implementing Quadrature Methods: Example

We get a system of $2na$ equations in $2na$ unknowns. If $na = 2$:

$$\int_{-1}^{1} 1\Delta a = 2 = \omega_1 + \omega_2$$

$$\int_{-1}^{1} a\Delta a = 0 = \omega_1 a_1 + \omega_2 a_2$$

$$\int_{-1}^{1} a^2\Delta a = 2/3 = \omega_1 a_1^2 + \omega_2 a_2^2$$

$$\int_{-1}^{1} a^3\Delta a = 0 = \omega_1 a_1^3 + \omega_2 a_2^3$$

Solving for (ω_1, ω_2) and a_1, a_2 implies the solution (check):

$$\int_{-1}^{1} f(a)\Delta a \approx \sum_{i=1}^{na} \omega_i f(a_i) = 1.f(-\sqrt{1/3}) + 1.f(\sqrt{1/3})$$

This computes exactly the integral of any cubic function (check):

$$\int_{-1}^{1} [q_0 + q_1 a + q_2 a^2 + q_3 a^3]\Delta a$$

Gauss Hermite Quadrature: Some Additional Issues

To deal with Normal distributions we instead construct the nodes and weights using:

$$\int_{-1}^{1} a^q e^{-a^2} da \approx \sum_{i=1}^{n} \omega_i a_i^q$$

- The weighting function e^{-x^2} is captured in the coefficients, ω_i.
- To exactly implement this we need to perform a change of variable before integration so that x is a standard normal

Very Persistent Processes - Rouwenhorst (1995)

The Tauchen-Hussey method is less accurate with very persistent processes: $\rho \approx 1$

- The Rouwenhorst (1995) procedure often works better.

Again Δa is constant. The transition matrix for $na = 2$ is simply:

$$P_2 = \begin{bmatrix} p & 1-p \\ 1-q & q \end{bmatrix}$$

For $na > 2$, P_{na} can be constructed recursively as follows:

$$\begin{aligned} P_{na} &= p \begin{bmatrix} P_{na-1} & \mathbf{0} \\ \mathbf{0}' & 0 \end{bmatrix} + (1-p) \begin{bmatrix} \mathbf{0} & P_{na-1} \\ 0 & \mathbf{0}' \end{bmatrix} \\ &+ (1-q) \begin{bmatrix} \mathbf{0}' & 0 \\ P_{na-1} & \mathbf{0} \end{bmatrix} + q \begin{bmatrix} 0 & \mathbf{0}' \\ \mathbf{0} & P_{na-1} \end{bmatrix} \end{aligned}$$

then divide all but the top and bottom rows by two so that the conditional probabilities sum to one.
Note: $\mathbf{0}$ is a $na-1$ column vector and $\mathbf{0}'$ is a $na-1$ row vector.

Very Persistent Processes - Rouwenhorst (1995)

Comments:

- It can be shown that regardless of the choice of *na* the first-order serial correlation of this process will always be $p + q - 1$
- Hence we can ensure the discrete process has the same first-order persistence as the continuous process.
- Setting $p \neq q$ introduces conditional heteroscedasticity in the shocks
- The standard deviation of the approximated process equals $\frac{\sqrt{na-1}}{2}\Delta a$
- Hence we can choose Δa so that:

$$\frac{\sqrt{na-1}}{2}\Delta a = \frac{\sigma}{\sqrt{1-\rho^2}}$$

Firm Problem in Continuous Time: Deterministic Case

The problem of the firm can be written as follows:

$$
\begin{aligned}
v(k(0)) \quad &= \quad \max_{\{i(t)\}_{t\geq 0}} \int_0^{\infty} M_{0,t} d(k_t, i(t)) dt \\
s.t. \quad & \quad dk/dt = \dot{k}(t) = f(i(t), k(t)) \\
& \quad lim_{T\to\infty} k(T) M_{0,T} = 0
\end{aligned}
$$

Comments

- Without aggregate risk, the instantaneous discount factor becomes

$$M_{0,t} = e^{-rt}$$

- The capital accumulation is:

$$\dot{k}(t) = i(t) - \delta k(t)$$

Next: practical approaches to solve these problems

- Theory: Kamien and Schwarz (1991) and Oksendal (2010)

Solving the Firm's Problem: The Maximum Principle

Construct the Lagrangian (finite horizon)

$$\begin{aligned} L &= \max_{\{i(t)\}_{t\geq 0}} \int_0^T e^{-rt} d(k(t), i(t)) + \mu(t) \left[f(i(t), k(t)) - \dot{k}(t) \right] dt \\ &+ \nu k(T) e^{-rT} \end{aligned}$$

Integrate the second term by parts to get:

$$\begin{aligned} L &= \max_{\{i(t)\}_{t\geq 0}} \int_0^T e^{-rt} d(k(t), i(t)) + \left[\mu(t) f(i(t), k(t)) + \dot{\mu}(t) k(t) \right] dt \\ &+ \mu(0) k(0) - \mu(T) k(T) + \nu k(T) e^{-rT} \\ &= \max_{\{i(t)\}_{t\geq 0}} \int_0^T \hat{H}(k(t), i(t)) + \dot{\mu}(t) k(t) dt \\ &+ \mu(0) k(0) - \mu(T) k(T) + \nu k(T) e^{-rT} \end{aligned}$$

where we defined the **Hamiltonian** function:

$$\hat{H}(k(t), i(t)) = e^{-rt} d(k(t), i(t)) + \mu(t) f(i(t), k(t))$$

Solving the Firm's Problem: The Maximum Principle

Using standard optimization methods, the FOC to the problem above are:

$$\begin{aligned} \frac{\partial \hat{H}}{\partial i} &= 0 \\ \frac{\partial \hat{H}}{\partial k} &= -\dot{\mu}(t) \\ \mu(T) &= \nu e^{-rT} \end{aligned}$$

The final boundary condition follows from optimality of $k(T)$, and implies the TVC:

$$\mu(T)k(T) = 0$$

For infinite horizon problems this becomes

$$\lim_{T \to \infty} \mu(T)k(T) = 0$$

Current and Discounted Value Hamiltonian

In economics and finance we usually want to work with the present/discounted value of a return function (utility or dividends) so

$$\hat{H}(k(t), i(t)) = e^{-rt} d(k(t), i(t)) + \mu(t) f(i(t), k(t))$$

It is then also useful to define the present/discounted value multipliers:

$$q(t) = e^{-rt} \mu(t)$$

and work with the current value Hamiltonian

$$H(k(t), i(t)) = d(k(t), i(t)) + q(t) f(i(t), k(t)) = e^{-rt} \hat{H}(k(t), i(t))$$

In this case the second FOC above can be rewritten as:

$$\frac{\partial H}{\partial k} = \qquad rq(t) - \dot{q}(t)$$

Hamilton-Jacobi-Bellman Equation

Dynamic programming, or recursive approach:

- How does firm value change over any given instant, dt

$$dv = -rdt + d(i,k)dt + \frac{\partial v}{\partial k}\frac{dk}{dt}dt + \frac{\partial v}{\partial t}dt$$

- In infinite horizon problems v is time-invariant ($\frac{\partial v}{\partial t} = 0$), so:

$$dv = -rdt + d(i,k)dt + v'(k)\dot{k}dt$$

At the optimum, $dv = 0$, and this yields Hamilton-Jacobi-Bellman (HJB) equation:

$$rv(k) = \max_i d(i,k) + v'(k)f(i,k) = H(k,i)$$

where $H(\cdot)$ is again the **current value Hamiltonian**.

- The left hand side is simply the instantaneous rate of return on an investment of size $v(k)$.
- The right hand side equals the sum of instantaneous dividends, $d(i,k)$ plus capital gains, $\dot{v}(t) = v'(k)\dot{k}(t)$

Adding (Easy) Uncertainty: Poisson Shocks

Suppose that $a \in \{a_1, a_2\}$ with jump intensities p_1, p_2.

- This can be implemented with a simple Markov transition matrix for the exogenous states - as in the discrete time examples.

The HJB equation now becomes:

$$rv_i(k) = H(k, v'(k)) = \max_i d(i,k) + v_i'(k) f(i,k) + p_i[v_j(k) - v_i(k)]$$

where $v_i(k)$ is the value of the firm when $a = a_i$ for each $i = 1, 2$

- The capital gain is now augmented with the possibility that the value of the firm will jump discretely in the next instant with intensity equal to p_i.

Numerical Solution of HJB Equations

Finite Difference Method

$$rv(k) = \max_{\{i(t)\}_{t\geq 0}} d(i,k) + v'(k)f(i,k)$$

Some key technical references:

- Barles and Souganidis (1991) and Tourin (2013)

Key ideas:

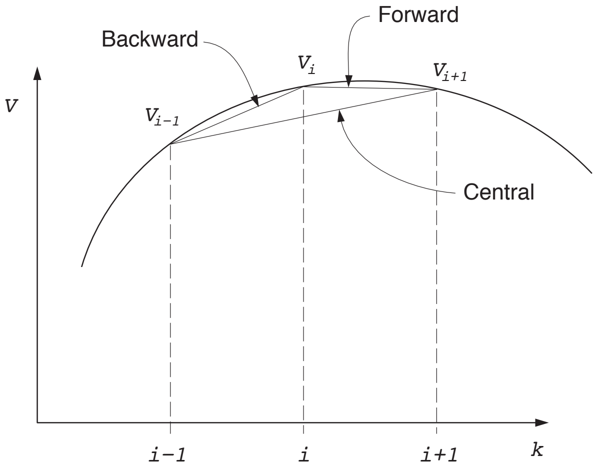
- Construct a discrete equally spaced grid for $k \in \{k_1, ...k_{nk}\}$.
- Let $\Delta k = k_i - k_{i-1}$ and denote $v_i = v(k_i)$
- Approximate $v'(k_i)$ with the following differences:

$$v'(k_i) \approx \frac{v_i - v_{i-1}}{\Delta k} = v'_{i,B} \qquad \text{Backward Difference}$$

$$v'(k_i) \approx \frac{v_{i+1} - v_i}{\Delta k} = v'_{i,F} \qquad \text{Forward Difference}$$

$$v'(k_i) \approx \frac{v_{i+1} - v_{i-1}}{2\Delta k} = v'_{i,C} \qquad \text{Central Difference}$$

Finite Difference Approximations



Note: By concavity $v'_{i,F} \leq v'_{i,C} \leq v'_{i,B}$

Finite Difference Approximation

To facilitate exposition, suppose in what follows that adjustment costs are CRS so that

$$\Phi_i(i,k) = \phi'(i/k)$$

Then at each grid point, the approximate HJB becomes:

$$rv_i = \max_i d(i_i, k_i) + v_i' f(i_i, k_i)$$

where

$$i_i = (\phi')^{-1}(v_i' - 1)k_i$$

How to construct the derivatives, v_i? Use:

- $v'_{i,F}$ if $\dot{k}_{i,F} = i_{i,F} - \delta k_i = [(\phi')^{-1}(v'_{i,F} - 1) - 1]k_i > 0$,
- $v'_{i,B}$ if $\dot{k}_{i,B} = i_{i,B} - \delta k_i = [(\phi')^{-1}(v'_{i,B} - 1) - 1]k_i < 0$,
- $v'_{i,C}$ if $[(\phi')^{-1}(v'_{i,F} - 1) - 1]k_i < 0 < [(\phi')^{-1}(v'_{i,B} - 1) - 1]k_i$

The last option will not be needed in the approximation.

Matrix Representation and Sparsity

Discretized HJB is:

$$rv_i = d_i + \frac{v_{i+1} - v_i}{\Delta k}\dot{k}_{i,F}\chi_{\dot{k}_{i,F}>0} + \frac{v_i - v_{i-1}}{\Delta k}\dot{k}_{i,B}\chi_{\dot{k}_{i,B}<0}$$

Matrix form

$$rv = d + \mathbf{B}v$$

where $\mathbf{B}$ is a very sparse matrix of size $nk \times nk$. Entries in row i:

$$\begin{bmatrix} \underbrace{-\frac{\dot{k}_{i,B}}{\Delta k}}_{>0} & \underbrace{\frac{\dot{k}_{i,B}}{\Delta k} - \frac{\dot{k}_{i,F}}{\Delta k}}_{<0} & \underbrace{\frac{\dot{k}_{i,F}}{\Delta k}}_{<0} \end{bmatrix} \begin{bmatrix} v_{i-1} \\ v_i \\ v_{i+1} \end{bmatrix}$$

However, d and $\mathbf{B}$ depend on v - nonlinear equation that requires iteration.

Value Function Iteration: Explicit Method

Algorithm:

1. Guess an initial v_i^0 for $i = 1, 2, ..nk$
2. At each stage, compute $(v^n)'$ and i^n
3. Update your guess using:

$$\frac{v^{n+1} - v^n}{\Delta} + rv^n = d^n + \mathbf{B}^n v^n$$

4. Stop when $||v^{n+1} - v^n|| < \epsilon$

Comments:

- Step size Δ cannot be too large
- Hence, it may take a lot of iterations, although they are should be very fast

Value Function Iteration: Implicit Method

To speed up computations we can instead iterate on:

$$\frac{v^{n+1} - v^n}{\Delta} + rv^{n+1} = d^n + \mathbf{B}^n v^{n+1}$$

Now, each step involves solving the **linear** system:

$$\left[(r + \frac{1}{\Delta})\mathbf{I} - \mathbf{B}^n\right] v^{n+1} = d^n + \frac{1}{\Delta} v^n$$

Because $\mathbf{B}^n$ is very sparse, this is extremely fast.

- Linearity implies that the step size does not really matter anymore
- Can handle many more grid points

However: seems to be sensitive to central derivative

- Maybe better to just impose $v'_{i,C} = 1$

The Firm's Problem in Continuous Time: Policy Functions

FOC with CRS

$$
\begin{aligned}
\frac{\partial H}{\partial i} &= d_i(k(t), i(t)) + q(t) f_i k(t), i(t)) = 0 \\
& 1 + \Phi_i(i(t)/k(t)) = q(t) \implies i(t) = \Phi_i^{-1}(q(t) - 1)k(t) \\
\frac{\partial H}{\partial k} &= d_k(k(t), i(t)) + q(t) f_k(k(t), i(t)) \\
& \pi_k(i(t), k(t)) - \Phi_k(i(t)/k(t)) - q(t)\delta = rq(t) - \dot{q}(t)
\end{aligned}
$$

Plus the law of motion for capital:

$$\dot{k}(t) = i(t) - \delta k(t)$$

Two ODE's (recognizing $i(t) = i(k(t), q(t))$:

$$
\begin{aligned}
&\dot{q}(t) = (r + \delta) q(t) - [\pi_k(i(t), k(t)) - \Phi_k(i(t)/k(t))] \\
&\dot{k}(t) = i(t) - \delta k(t)
\end{aligned}
$$

The Firm's Problem in Continuous Time: Policy Functions

Two boundary conditions:

$$k(0) = k_0$$
$$\lim_{T \to \infty} \mu(T) k(T) = 0$$

The terminal condition always makes it difficult to solve the system of ODE's directly.

- This is another way to see why we will often need numerical methods

But we can can easily compute the steady-state - just a system of (non-linear) equations:

$$\dot{k}(t) = 0 \implies i^\star = \delta k^\star \implies q^\star = 1$$
$$\dot{q}(t) = 0 \implies r + \delta = \pi_k(k^\star) \implies k^\star = \pi_k^{-1}(r + \delta)$$

Next, use FD method to compute the solution to ODEs.

Shooting Algorithm

Methodology:

- Denote distance between grid points by Δt.
- Approximate $k(t)$ and $q(t)$ at N discrete points in the time dimension, t^n, $n = 1, ..., N$. Let $k^n = k(t^n)$
- Approximate the derivatives:

$$\dot{k}(t_n) = \frac{k^{n+1} - k^n}{\Delta t}$$

- Approximate the ODEs as:

$$\frac{q^{n+1} - q^n}{\Delta t} = [1 - (r + \delta)]q^n - [\pi_k^n - \Phi_k^n]$$
$$\frac{k^{n+1} - k^n}{\Delta t} = i^n - \delta k^n$$

Shooting Algorithm

Methodology:

- Guess k^0 and q^0
- Generate a sequence k^n, q^n, $n = 1, ..., N$ by running the approximate ODE's forward in time.
- If the sequence converges to $k^\star, q^\star$, then we have obtained the correct solution.
- Also called the unique "saddle path".
- Otherwise try different value for k^0.

Continuous Time Stochastic Processes

Suppose now $\pi(t) = \pi(a(t), k(t))$, where $a(t)$ is a **diffusion**:

- A process with continuous sample paths, i.e. no jumps
- Simply a continuous-time Markov

Simplest diffusion is the Brownian/Wiener process:

$$dw(t) = \lim_{dt \to 0} [w(t + dt) - w(t)] = \lim_{dt \to 0} \epsilon \sqrt{dt},$$
$$\epsilon \sim N(0, 1) \qquad w(0) = 0;$$

This is the continuous time analogue to a random walk process

$$w(t + 1) - w(t) = \epsilon, \qquad \epsilon \sim N(0, 1)$$

It is easy to see that:

$$w(t) \sim N(0, t)$$

so the variance of the process blows up over time.

Continuous Time Stochastic Processes

A more general stochastic process is the Brownian motion with drift:

$$da(t) = \mu dt + \sigma dw(t)$$

Now

$$E(a(t)) = a(0) + \mu t, \qquad V(a(t)) = \sigma^2 t$$

- This is also an example of a **stochastic differential equation** which we will now need to solve.

A generalized diffusion process is then:

$$da(t) = \mu(a)dt + \sigma(a)dw(t)$$

where $\mu(a)$ and $\sigma(a)$ can be general functions.

Popular Continuous Time Processes

Geometric Brownian motion (GBM):

- Used for log-normal variables,
- Often most likely to yield closed-form solutions

$$da = \mu a dt + \sigma a dw \implies da/a = \mu dt + \sigma dw$$

- Now $\ln a(t) \sim N(\mu - \sigma^2/2, \sigma^2 t)$

Ornstein-Uhlenbeck Process:

- The analogous to an AR(1) process

$$da = \theta(\bar{a} - a)dt + \sigma dw$$

- Mean-reverting process that implies stationary distribution for the process $a \sim N(\bar{a}, \sigma^2/(2\theta))$

Popular Continuous Time Processes

Feller square root process:

$$da = \theta(\bar{a} - a)dt + \sigma\sqrt{a}dw$$

- The "Cox-Ingersoll-Ross" process (CIR)
- This process ensures $a(t) \geq 0 \qquad \forall t$
- Very useful to model interest rates in finance

Stationary distribution of a is Gamma:

$$f(a) \propto e^{-\beta a} a^{\gamma - 1}$$

with $\beta = 2\theta/\sigma^2$ and $\gamma = \beta\bar{a}$

Stochastic Hamilton-Jacobi-Bellman Equation

The Hamilton-Jacobi-Bellman (HJB) equation:

- The **expected** change in firm value over an instant dt obeys:

$$rv(a,k)dt = \mathrm{E} \max_i d(a,i,k)dt + \frac{\partial v}{\partial k}dk + \frac{\partial v}{\partial a}da + \frac{1}{2}\frac{\partial^2 v}{\partial a^2}(da)^2$$

Why the second order term?

- Because dw is of order $\sqrt{t}$, for a generalized Brownian motion:

$$\mathrm{E}(da)^2 = (\mu(a)dt + \sigma(a)dw(t))^2 = \sigma(a)^2 dt + o(dt^2)$$

The HJB for a general firm problem now becomes a **partial differential equation** (PDE):

$$rv(a,k) = \max_i d(a,i,k) + \frac{\partial v}{\partial k}[i - \delta k] + \frac{\partial v}{\partial a}\mu(a) + \frac{\partial^2 v}{\partial a^2}\frac{\sigma(a)^2}{2}$$

This "derivation" is an application of **Ito's Lemma**.

Solving the Stochastic HJB

The Finite Difference method can easily be generalized to solve the general stochastic HJB

- Need to discretize and create a grid for a with $ia = 1, 2, ...na$ points, equally spaced at intervals Δa
- Of course this only makes sense if the long run distribution of a is stationary

We can then compute the approximate derivatives:

$$\frac{\partial v}{\partial a}(k_{ik} a_{ia}) = \frac{v(k_{ik}, a_{ia+1}) - v(k_{ik}, a_{ia-1})}{2\Delta a}$$
$$\frac{\partial^2 v}{\partial a^2}(da)^2 = \frac{v(k_{ik}, a_{ia+1}) - 2v(k_{ik}, a_{ia}) + v(k_{ik}, a_{ia-1})}{(2\Delta a)^2}$$

- Note that these are just the central differences.

The method can then be applied as discussed earlier.

Example of Closed Form Solutions: Real Options

Consider the case of a firm where:

- $k = 0$ and there is a **single** option to invest a fixed amount $i > 0$
- a follows a geometric Brownian motion (GBM)

In this case the HJB is simply

- Before investment

$$rv^0(a) = 0 + \frac{\partial v^0}{\partial a}\mu + \frac{\partial^2 v^0}{\partial a^2}\frac{\sigma^2}{2}$$

- After investment

$$rv^1(a) = \pi(a)$$

When does investment take place?

- When productivity reaches a critical threshold $a = a^\star$

Example of Closed Form Solutions: Real Options

Solution (check):

$$v^0(a) = A_1 a^{\eta_1} + A_2 a^{\eta_2}$$
$$v^1(a) = \pi(a)/r$$

where $\eta_1 < 0$, $\eta_2 > 1$. What about the constants?

- **Boundary Condition**:

$$\lim_{a \to 0} v^0(a) = \lim_{a \to 0} A_1 a^{\eta_1} + A_2 a^{\eta_2} = 0 \implies A_1 = 0$$

- **Value Matching condition**:

$$v^0(a^\star) = v^1(a^\star) - i \implies A_2 (a^\star)^{\eta_2} = \pi(a^\star)/r - i$$

- **Smooth Pasting condition**:

$$\frac{\partial v^0}{\partial a}(a^\star) = \frac{\partial v^1}{\partial a}(a^\star) \implies \eta_2 A_2 (a^\star)^{\eta_2 - 1} = 1/r$$

Example of Closed Form Solutions: Real Options

The last two equations can be solved for $a^\star$ and A_2

- This yields an exact closed form solution for the value function
- This is a benefit of some continuous time versions of the model

$$v(a) = \begin{cases} A_2 a^{\eta_2}, \text{ if } a < a^\star \\ \pi(a)/r - i, \text{ if } a \geq a^\star \end{cases}$$

Some useful intuition:

- The value function is convex in productivity in the inaction region
- A typical option valuation result - firm only faces upside risk
- Optimal investment requires $\pi(a^\star)/r - i = A_2(a^\star)^{\eta_2} > 0$
- There is a value in waiting to invest until a is sufficiently high
- A simple take or leave it decision would imply the investment cutoff obeys $\pi(a^\star)/r = i$