

# 1 The EGM<sup>n</sup> in Higher Dimensions

## 1.1 A more complex problem

The problem above demonstrates the simplicity of solving problems sequentially. However, as constructed, the problem has only 1 state variable and 1 post-decision state variable per stage. Can EGM<sup>n</sup> be used to solve higher dimensional problems? In short, yes, but it requires additional thought on interpolation.

For a demonstration, we now turn to the problem of a worker saving up for retirement. This worker must consume, save, and deposit resources into a tax-advantaged account that can not be liquidated until retirement. In the recursive problem, the worker begins a new period with a liquid account of market resources  $m_t$  and an illiquid account of retirement savings  $n_t$ . The worker maximizes their utility by choosing consumption  $c_t$  and pension deposit  $d_t$ . The pension deposit is set aside on a retirement account that is exposed to a risky return, while their post-consumption liquid assets accrue risk-free interest every period. The worker additionally receives an income which faces a permanent ( $\Gamma_{t+1}$ ) and a transitory ( $\theta_{t+1}$ ) shock every period. At the age of 65, the worker is retired and their assets are liquidated, at which point the state reduces to one liquid account of market resources. The worker's recursive problem is:

$$\begin{aligned}
 v_t(m_t, n_t) &= \max_{c_t, d_t} u(c_t) + \beta \mathbb{E}_t [\Gamma_{t+1}^{1-\rho} v_{t+1}(m_{t+1}, n_{t+1})] \\
 \text{s.t. } & c_t \geq 0, \quad d_t \geq 0 \\
 & a_t = m_t - c_t - d_t \\
 & b_t = n_t + d_t + g(d_t) \\
 & m_{t+1} = a_t R / \Gamma_{t+1} + \theta_{t+1} \\
 & n_{t+1} = b_t \mathbf{R}_{t+1} / \Gamma_{t+1}
 \end{aligned} \tag{1}$$

This problem can subsequently be broken down into 3 stages: a pension deposit stage, a consumption stage, and an income shock stage.

In the deposit stage, the worker begins with market resources and a retirement savings account. The worker must maximize their value of liquid wealth  $l_t$  and retirement balance  $b_t$  by choosing a pension deposit  $d_t$ , which must be positive. The retirement balance  $b$  is the cash-value of their retirement account plus their pension deposit and an additional amount  $g(d_t)$  that provides an incentive to save for retirement. As we'll see, this additional term will allow us to use the endogenous grid method to solve this sub-problem.

$$\begin{aligned}
 v_t(m_t, n_t) &= \max_{d_t} \tilde{\mathbf{v}}_t(l_t, b_t) \\
 \text{s.t. } & d_t \geq 0 \\
 & l_t = m_t - d_t \\
 & b_t = n_t + d_t + g(d_t)
 \end{aligned} \tag{2}$$

After making their pension decision, the worker begins their consumption stage with liquid wealth  $l_t$  and retirement balance  $b_t$ . From their liquid wealth, the worker must choose a level of consumption to maximize utility and continuation value  $w_t$ . After consumption, the worker is left with post-decision states that represent liquid assets  $a_t$  and retirement balance  $b_t$ , which passes through this problem unaffected because it can't be liquidated until retirement.

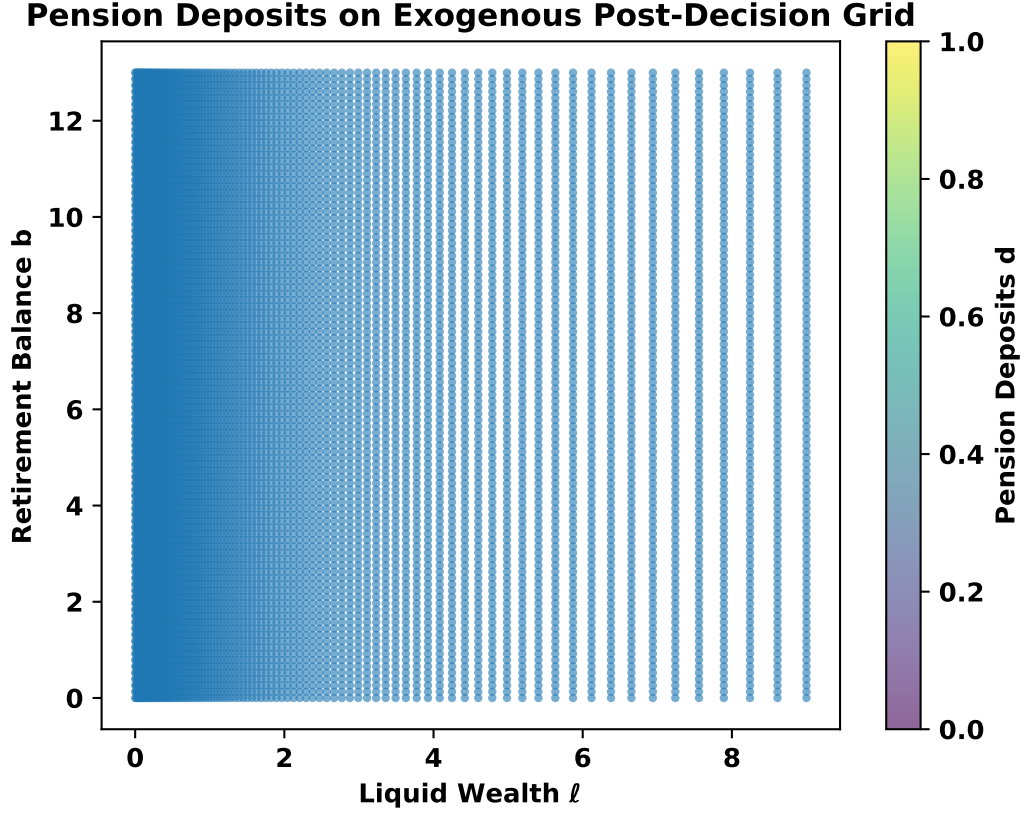
$$\begin{aligned}
\tilde{v}_t(l_t, b_t) &= \max_{c_t} u(c_t) + \beta w_t(a_t, b_t) \\
\text{s.t. } c_t &\geq 0 \\
a_t &= m_t - c_t - d_t \\
b_t &= n_t + d_t + g(d_t)
\end{aligned} \tag{3}$$

Finally, the post-decision value function  $w_t$  represents the value of both liquid and illiquid account balances before the realization of uncertainty regarding the risky return and income shocks. Since we are dealing with a normalized problem, this stage handles the normalization of state variables and value functions into the next period. The advantage of conceptualizing this sub-problem as a separate stage is that we can construct a function  $w_t$  and use it in the prior optimization problems without having to worry about stochastic optimization.

$$\begin{aligned}
w_t(a_t, b_t) &= \mathbb{E}_t [\Gamma_{t+1}^{1-\rho} v_{t+1}(m_{t+1}, m_{t+1})] \\
\text{s.t. } a_t &\geq 0, \quad b_t \geq 0 \\
m_{t+1} &= a_t \mathbf{R} / \Gamma_{t+1} + \theta_{t+1} \\
n_{t+1} &= b_t \mathbf{R}_{t+1} / \Gamma_{t+1}
\end{aligned} \tag{4}$$

## 1.2 The pass-through case

As seen in the consumption stage above, the retirement balance  $b_t$  passes through the problem unaffected because it can't be liquidated until retirement. In this sense, it is already a post-decision state variable. To solve this problem, we can use a fixed grid of  $b_t$  and for each obtain endogenous consumption and ex-ante market resources using the simple endogenous grid method for the consumption problem.



In the deposit stage, both the state variables and post-decision variables are different since both are affected by the pension deposit decision.

First, we can rewrite the pension deposit problem more compactly as:

$$v_t(m_t, n_t) = \max_{d_t} \tilde{v}_t(m_t - d_t, n_t + d_t + g(d_t)) \quad (5)$$

The first order condition is

$$\tilde{v}_t^l(l_t, b_t)(-1) + \tilde{v}_t^b(l_t, b_t)(1 + g'(d_t)) = 0 \quad (6)$$

Rearranging this equation gives

$$g'(d_t) = \frac{\tilde{v}_t^l(l_t, b_t)}{\tilde{v}_t^b(l_t, b_t)} - 1 \quad (7)$$

Assuming that  $g'(d)$  exists and is invertible, we can find

$$d_t(l_t, b_t) = g'^{-1} \left( \frac{\tilde{v}_t^l(l_t, b_t)}{\tilde{v}_t^b(l_t, b_t)} - 1 \right) \quad (8)$$

Using this, we can back out  $n_t$  as

$$\mathbf{n}_t(l_t, b_t) = b_t - \mathfrak{d}_t(l_t, b_t) - g(\mathfrak{d}_t(l_t, b_t)) \quad (9)$$

and  $m_t$  as

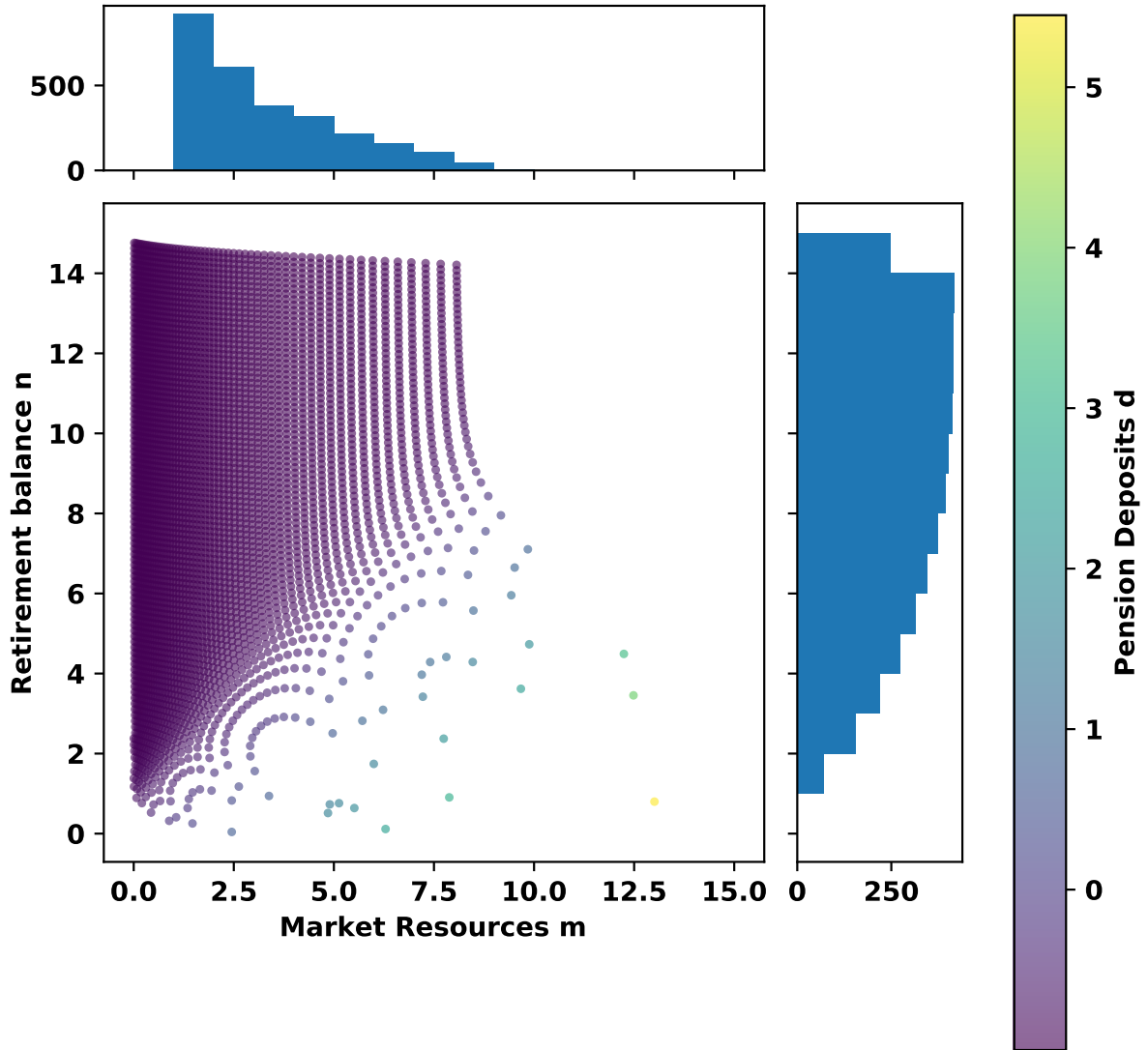
$$\mathbf{m}_t(l_t, b_t) = l_t + \mathfrak{d}_t(l_t, b_t) \quad (10)$$

In sum, given an exogenous grid  $(l_t, b_t)$  we obtain the triple  $(\mathbf{m}_t(l_t, b_t), \mathbf{n}_t(l_t, b_t), \mathfrak{d}_t(l_t, b_t))$ , which we can use to create an interpolator for the decision rule  $d_t$ .

To close the solution method, the envelope conditions are

$$\begin{aligned} v_t^m(m_t, n_t) &= \tilde{\mathfrak{v}}_t^l(l_t, b_t) \\ v_t^n(m_t, n_t) &= \tilde{\mathfrak{v}}_t^b(l_t, b_t) \end{aligned} \quad (11)$$

## Pension Deposit on Endogenous Grid



### 1.3 Unstructured Grid Interpolation

To interpolate a function defined on an unstructured grid, we use Gaussian Process Regression as in Scheidegger and Bilonis (2019).