# ANALYSIS OF FAIR FEE IN GUARANTEED LIFELONG WITHDRAWAL AND MARKOVIAN HEALTH BENEFITS

**Abhineet Singh**

**Parth Bhardwaj**

**Samyak Jain**

# QUICK RECAP

- Policyholder's health status is a continuous-time Markov chain with health states E = {1, 2, . . . , D}.
- The guarantee ($g$ % of $h_0$) is deducted by the policyholder over a discrete-time set

$$Ta = \{t \geq 0 : t = ak \, with \, a \in R^{+} \, and \, k \in N\}.$$

- The premium is a one-time lump sum investment of $h_0$ made by the insured at the start of the contract, which makes $h_0/s_0$ as the number of initial stocks.
- The insurance company's fee is deducted from the funds value, when the fund units are canceled, at a rate equal to γ .
- Death benefits are paid when the insured dies

$$EDB + ELB = h_0$$

## Calculating ELB

$$LB(0) = \sum_{t \in T_a} G(X(t)) h_0 e^{-rt} \mathbb{1}_{\{X(t) \neq D\}}.$$

$$ELB_i = \mathbb{E}[LB(0)|X(0) = i] = \mathbb{E}_i[LB(0)].$$

$$ELB_i = \sum_{t \in T_a} \mathbb{E}_i[G(X(t)) \mathbb{1}_{\{X(t) \neq D\}}] h_0 e^{-rt},$$

$$\boldsymbol{ELB} = \left(I - e^{-ra} \cdot \boldsymbol{P}(a)\right)^{-1} * \left(e^{-ra} \cdot \boldsymbol{P}(a) * \boldsymbol{G}\right),$$

## Calculating EDB

$$S_t = s_0 e^{\left(\mu - \frac{1}{2}\sigma^2\right)t + \sigma B_t}.$$

$$\begin{cases} dH_t = (\mu - \gamma) S_t dt + \sigma S_t dB_t, & \forall\, t : ak < t < a(k+1) \\ H_t = H_{t^-} - G(X(t)) h_0, & \forall\, t \in T_a. \end{cases}$$

$$\tau = \inf\{t \in \mathbb{R} : X(t) = D\}. \qquad DB_\beta(\gamma) = \max\{H_\beta, 0\}.$$

$$DB_\beta(\gamma) = e^{(\mu - \gamma - \sigma^2/2)\beta + \sigma B_\beta} \max[0, h_0 - \sum_{v=1}^{\lfloor \beta \rfloor} G(X(va)) h_0 e^{-(\mu - \gamma - \sigma^2/2)va - \sigma B_{va}}],$$

$$EDB(\gamma) = \int_0^\infty e^{-r\beta} \mathbb{E}[DB_\beta(\gamma)] \lambda(\beta) d\beta,$$

# MOTIVATION AND REASONING

- **Intuition**: By allowing the stock to grow without cancelling it at a constant rate gamma, we expect that the stock price would grow exponentially more in our case.
- This would mean that we can keep some part of EDB at the time of death and still ensure a fair fee for the insurer.
- This means that our policy is fair to the insurer as was the previous model, but in expectation generates more profit for the insurance company.
- This model also offers more transparency to the insurers in terms of their EDB.

# MODEL DESCRIPTION

- Health status X(t) is assumed to be a continuous time Markov chain with states {1,2,…,D}
- The benefits are calculated based on the policyholder's health evolution, and the fair fee is computed according to a balance condition between premium and expected benefits.
- Premium is a one-time investment $h_0$ invested in a single index with initial price S.
- The guarantee ($g$ % of $h_0$) is deducted by the policyholder over a discrete-time set

$$Ta = \{t \geq 0 : t = ak \, with \, a \in R^+ \, and \, k \in N\}$$

- Insurance company does not deduct any fee throughout the lifetime.
- At death, the company returns only partial death benefit, which is a function of the age of the person at the time of death.

# MODEL DESCRIPTION

- ELB calculation does not change. Hence ELB for the final model is

$$\boldsymbol{ELB} = \left(I - e^{-ra} \cdot \boldsymbol{P}(a)\right)^{-1} * \left(e^{-ra} \cdot \boldsymbol{P}(a) * \boldsymbol{G}\right),$$

- EDB calculation changes

$$DB_\beta = f(\beta) * \exp\left(\left(\mu - \frac{\sigma^2}{2}\right)\beta + \sigma B_\beta\right) \max\left[0,\ h_0 - \sum_{\nu=1}^{\lfloor\beta\rfloor} G\left(X(\nu a)\right) h_0 \exp\left(-\left(\mu - \frac{\sigma^2}{2}\right)\nu a - \sigma B_{\nu a}\right)\right]$$

$$\text{EDB} = \int_0^\infty e^{-r\beta}\, E\left[DB(\beta)\right] \lambda(\beta)\, d\beta.$$

# HOW TO SOLVE THIS?

- We used numerical analysis and Monte Carlo simulation to solve the equation
- We simulate a single trajectory of health status Markov chain and a Brownian motion using Monte Carlo Simulation to give death benefit for a single trajectory.

```python
def single_trajectory_simulation(args):
    beta, h0, G, mu, sigma, P, a = args
    trajectory = generate_trajectory(beta, P, a)
    val1 = 0
    total1 = 0
    random_normal_number = 0
    steps = beta
    for time_step in range(steps):
        random_normal_number = np.random.normal(loc=0, scale=math.sqrt(2)) + val1
        value = G[trajectory[time_step]] * h0 *
        math.exp(- (mu - (sigma ** 2) / 2) * a * time_step - sigma * random_normal_number)
        total1 += value
        val1 = random_normal_number
    total1 = max(0, h0 - total1) * math.exp(sigma * random_normal_number )
    return total1
```

```python
def generate_trajectory(beta, P, a):
    P = give_P_matix(P)
    states = np.arange(len(P))
    current_state = 0
    trajectory = []

    for _ in range(beta - 1):
        current_state = np.random.choice(states, p=P[current_state])
        trajectory.append(current_state)
    trajectory.append(len(P))
    return trajectory
```

# HOW TO SOLVE THIS?

- We calculate EDB (numerical integration) by using Runge Kutta method of order 4.
- We now make a system of equations in variables $f(\beta_i)$'s and solve them putting constraints on each $0 \leq f(\beta_i) \leq 1$ to obtain f($\beta_i$)'s.
- Here are the plots:

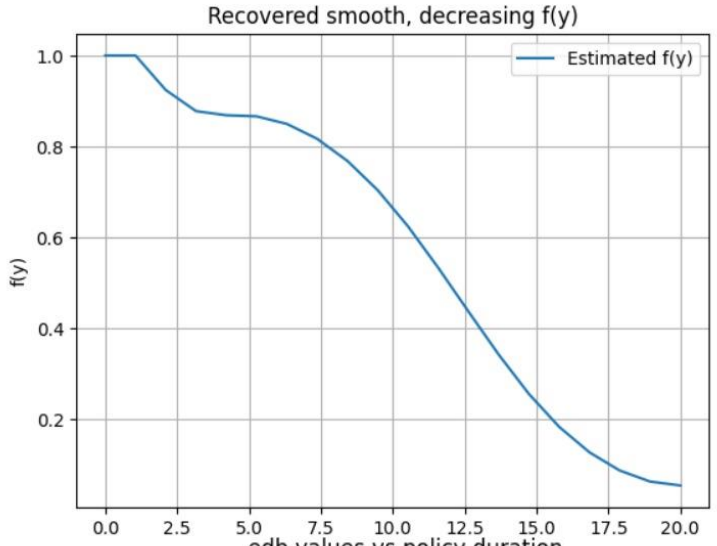
```
def compute_row(h0, start_amount, r, P, G, alpha_1, Q, mu, sigma, num_of_eqns, num_trajectory,a):
    print("computing row")
    lhs_value = h0 * (1 - calculate_ELB(r, P, G, 1)[0])
    row_values = []
    for eqn_num in range(num_of_eqns):
        edb_val = give_EDB_beta_seq(eqn_num, h0, G, mu, sigma, num_trajectory, P,a)
        h = a / 10
        F = 0.0
        def f(b):
            lam_val = lambda_beta(b, alpha_1, Q)[0]
            return math.exp(-r * a *eqn_num+((mu - (sigma ** 2) / 2) * eqn_num)) * lam_val

        b = a * eqn_num
        for _ in range(10):
            k1 = f(b)
            k2 = f(b + h/2)
            k3 = f(b + h/2)
            k4 = f(b + h)
            F += (h/6) * (k1 + 2*k2 + 2*k3 + k4)
            b += h

        row_values.append(F * edb_val)
    return (h0 - start_amount, lhs_value, row_values)
```

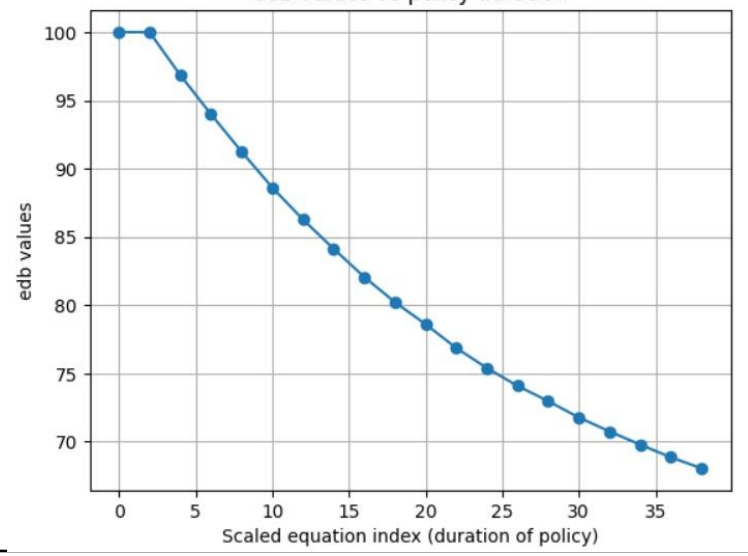
# PLOTS

- **f** represents the fraction of EDB (Expected Death Benefit) returned to the insurer.

- The plot is decreasing because individuals who die earlier receive a larger share of the investment returns, leaving less to be returned to the insurer.

- This graph is subjective to G and Q and hence can be altered according to the company's need. G taken here is [0.0312, 0.0312, 0.0412, 0.0412, 0.0512, 0.0512, 0.00]
and Q is the generator matrix for 65 years of age.



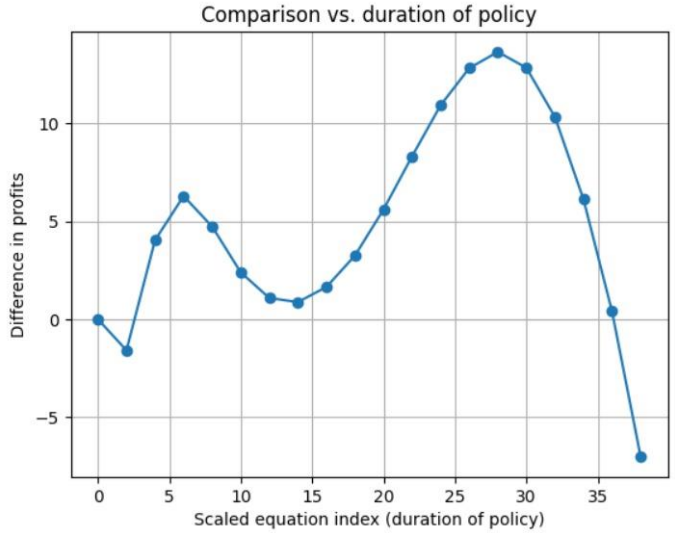



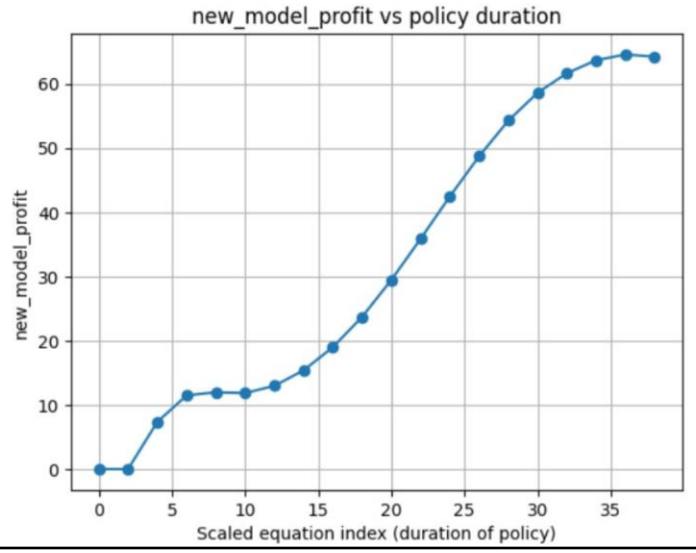
# PLOTS

- This plot shows the comparison between profit earned by the firm according to our model and the previous model.
- The difference is computed using the below formula

$$\text{comp} = \big(1 - f(\beta)\big)\,\mathrm{E}[\mathrm{DB}(\beta)] - \frac{h_0}{\mu}\,\gamma\big(e^{\mu\beta} - 1\big)$$

- The graph shows that the net profit ($our\ model - prev\ model$) is always positive except for very small and very large insurance duration



# THANK YOU