# Counterparty Credit Risk

To compute expected exposure profiles, we will need: (1) A model to simulate paths of interest rates, in a given probability measure. (2) A pricer to turn the paths of interest rates into paths of swap prices, properly aging the swap as we go along.

At time $t \in [T_0, T_1)$ (a stub treatment) the value of a Δ-period receiver swap (based on SOFR) with future payment dates $T_1, \ldots, T_N$ is:

$$V_{rec}(t) = c\,\Delta \sum_{1}^{N} P(t, T_i) - \beta(t)/\beta(T_0) + P(t, T_N)$$

We construct a Markov Model that approximates the price funciton using P(t,T) = F(t, x(t),T) where x(t) is a Markov Process. To establish the Expected Exposure profile for a Swap Value V on a discretized time grid $t_j$ as:

1. Simulate a path of x(t), for $t \in t_j$.
2. Use the reconstitution formula to establish r(t) (spot rates), for $t \in t_j$.
3. Use numerical integration (e.g. trapezoid) to establish β(t), for $t \in t_j$.
4. Use the above formula to value the swap on the time grid, establishing a path of $V_{rec}(t_j)$.
5. Compute a path of exposures, using $E(t) = V_{rec}(t)^+$ for $t \in tj$.
6. Average over many independent path

We state a dynamic model that allows us to simulate the vector x(t) and to have a concrete expression for the reconstitution function F. In this model the stochasticity is generated by Brownian motions. Specifically, we take specialization of HJM models that allow for a Markov representation of the general form. The markov vector $x(t) = (x_1(t), x_2(t))^T$ is chosen that allows for “twists” of the forward curve, where the short end moves opposite that of the long end.

$$F(t, T, x(t)) = \frac{P(0, T)}{P(0, t)} e^{A(t,T) + C(t,T)^T x(t))}$$

$$C(t, T) = -\int_t^T M(t, u) du$$

$$M(t, u)^T = \begin{bmatrix} 1 & e^{-k_2(T-t)} \end{bmatrix}$$

$$G(t, T) = -C(t, T) = \int_t^T M(t, u) du = \int_t^T \begin{bmatrix} 1 \\ e^{-k_2(u-t)} \end{bmatrix} du$$

$$G(t, T) = \begin{bmatrix} T - t \\ \frac{-1}{k_2} e^{-k_2(T-t)} + \frac{1}{k_2} \end{bmatrix}$$

$$y(t) = B(t) \left( \int_0^t \begin{bmatrix} \sigma_1 & p_x \sigma_1 \sigma_2 e^{k_2 u} \\ p_x \sigma_1 \sigma_2 e^{k_2 u} & \sigma_2 e^{k_2 u} \end{bmatrix} \begin{bmatrix} \sigma_1 & p_x \sigma_1 \sigma_2 e^{k_2 u} \\ p_x \sigma_1 \sigma_2 e^{k_2 u} & \sigma_2 e^{k_2 u} \end{bmatrix} du \right) B(t)$$

$$y(t) = B(t) \left( \int_0^t \begin{bmatrix} \sigma_1^2 + p_x^2 \sigma_1^2 \sigma_2^2 e^{2k_2 u} & p_x \sigma_1^2 \sigma_2 e^{k_2 u} + p_x \sigma_1 \sigma_2^2 e^{2k_2 u} \\ p_x \sigma_1^2 \sigma_2 e^{k_2 u} + p_x \sigma_1 \sigma_2^2 e^{2k_2 u} & p_x^2 \sigma_1^2 \sigma_2^2 e^{2k_2 u} + \sigma_2^2 e^{2k_2 u} \end{bmatrix} du \right) B(t)$$

$$y(t) = B(t) \left( \begin{bmatrix} \sigma_1^2 t + \frac{1}{2k_2} p_x^2 \sigma_1^2 \sigma_2^2 (e^{2k_2 t} - 1) & \frac{1}{k_2} p_x \sigma_1^2 \sigma_2 (e^{k_2 t} - 1) + \frac{1}{2k_2} p_x \sigma_1 \sigma_2^2 (e^{2k_2 t} - 1) \\ \frac{1}{k_2} p_x \sigma_1^2 \sigma_2 (e^{k_2 t} - 1) + \frac{1}{2k_2} p_x \sigma_1 \sigma_2^2 (e^{2k_2 t} - 1) & \frac{1}{2k_2} p_x^2 \sigma_1^2 \sigma_2^2 (e^{2k_2 t} - 1) + \frac{1}{2k_2} \sigma_2^2 (e^{2k_2 t} - 1) \end{bmatrix} \right) B(t)$$

$$B(t) = \begin{bmatrix} 1 & 0 \\ 0 & e^{-k_2 t} \end{bmatrix}$$

$$y(t) = \begin{bmatrix} 1 & 0 \\ 0 & e^{-k_2 t} \end{bmatrix} \left( \begin{bmatrix} \sigma_1^2 t + \frac{1}{2k_2} p_x^2 \sigma_1^2 \sigma_2^2 (e^{2k_2 t} - 1) & \frac{1}{k_2} p_x \sigma_1^2 \sigma_2 (e^{k_2 t} - 1) + \frac{1}{2k_2} p_x \sigma_1 \sigma_2^2 (e^{2k_2 t} - 1) \\ \frac{1}{k_2} p_x \sigma_1^2 \sigma_2 (e^{k_2 t} - 1) + \frac{1}{2k_2} p_x \sigma_1 \sigma_2^2 (e^{2k_2 t} - 1) & \frac{1}{2k_2} p_x^2 \sigma_1^2 \sigma_2^2 (e^{2k_2 t} - 1) + \frac{1}{2k_2} \sigma_2^2 (e^{2k_2 t} - 1) \end{bmatrix} \right) \begin{bmatrix} 1 & 0 \\ 0 & e^{-k_2 t} \end{bmatrix}$$

$$y(t) = \left( \begin{bmatrix} \sigma_1^2 t + \frac{1}{2k_2} p_x^2 \sigma_1^2 \sigma_2^2 (e^{2k_2 t} - 1) & \frac{1}{k_2} p_x \sigma_1^2 \sigma_2 (e^{k_2 t} - 1) + \frac{1}{2k_2} p_x \sigma_1 \sigma_2^2 (e^{2k_2 t} - 1) \\ \frac{1}{k_2} p_x \sigma_1^2 \sigma_2 (1 - e^{-k_2 t}) + \frac{1}{2k_2} p_x \sigma_1 \sigma_2^2 (e^{k_2 t} - e^{-k_2 t}) & \frac{1}{2k_2} p_x^2 \sigma_1^2 \sigma_2^2 (e^{k_2 t} - e^{-k_2 t}) + \frac{1}{2k_2} \sigma_2^2 (e^{k_2 t} - e^{-k_2 t}) \end{bmatrix} \right) \begin{bmatrix} 1 & 0 \\ 0 & e^{-k_2 t} \end{bmatrix}$$

$$y(t) = \begin{bmatrix} \sigma_1^2 t + \frac{1}{2k_2} p_x^2 \sigma_1^2 \sigma_2^2 (e^{2k_2 t} - 1) & \frac{1}{k_2} p_x \sigma_1^2 \sigma_2 (1 - e^{-k_2 t}) + \frac{1}{2k_2} p_x \sigma_1 \sigma_2^2 (e^{k_2 t} - e^{-k_2 t}) \\ \frac{1}{k_2} p_x \sigma_1^2 \sigma_2 (1 - e^{-k_2 t}) + \frac{1}{2k_2} p_x \sigma_1 \sigma_2^2 (e^{k_2 t} - e^{-k_2 t}) & \frac{1}{2k_2} p_x^2 \sigma_1^2 \sigma_2^2 (1 - e^{-2k_2 t}) + \frac{1}{2k_2} \sigma_2^2 (1 - e^{-2k_2 t}) \end{bmatrix}$$

$$y(t) = \begin{bmatrix} \sigma_1^2 t + \frac{1}{2k_2} p_x^2 \sigma_1^2 \sigma_2^2 (e^{2k_2 t} - 1) & \frac{1}{k_2} p_x \sigma_1 \sigma_2 [\sigma_1 + \frac{\sigma_2}{2} e^{k_2 t} - (\sigma_1 + \frac{\sigma_2}{2}) e^{-k_2 t}] \\ \frac{1}{k_2} p_x \sigma_1 \sigma_2 [\sigma_1 + \frac{\sigma_2}{2} e^{k_2 t} - (\sigma_1 + \frac{\sigma_2}{2}) e^{-k_2 t}] & \frac{\sigma_2^2}{2k_2} (p_x^2 \sigma_1^2 + 1)(1 - e^{-2k_2 t}) \end{bmatrix}$$

Since $y_{10} = y_{01}$, so we can simplify the expression for $y(t)$ as follows

We now want to simulate the process for $x(t)$ on a discrete time grid $t_j$ that spans the horizon $[0, 10yrs]$. We also fix all our model constants to $a = 2.5\%$, $b = 0.2\%$, $\sigma_r = 2\%$, $c = 40\%$, $\kappa_2 = 5\%$, $\rho_\infty = 40\%$.

$$y(t) = \begin{bmatrix} y_{00} & y_{01} \\ y_{10} & y_{11} \end{bmatrix} = \begin{bmatrix} y_{00} & y_{10} \\ y_{10} & y_{11} \end{bmatrix}$$

$$A(t,T) = -\int_t^T M(t,u)^T y(t) G(t,u) du$$

$$A(t,T) = -\int_t^T \begin{bmatrix} 1 & e^{-k_2(u-t)} \end{bmatrix} \begin{bmatrix} y_{00} & y_{10} \\ y_{10} & y_{11} \end{bmatrix} \begin{bmatrix} u-t \\ \frac{-1}{k_2} e^{-k_2(u-t)} + \frac{1}{k_2} \end{bmatrix} du$$

$$A(t,T) = -\int_t^T \begin{bmatrix} y_{00} + y_{10} e^{-k_2(u-t)} & y_{10} + y_{11} e^{-k_2(u-t)} \end{bmatrix} \begin{bmatrix} u-t \\ \frac{-1}{k_2} e^{-k_2(u-t)} + \frac{1}{k_2} \end{bmatrix} du$$

$$A(t,T) = -\int_t^T (u-t)(y_{00} + y_{10} e^{-k_2(u-t)}) - (\frac{1}{k_2} e^{-k_2(u-t)} - \frac{1}{k_2})(y_{10} + y_{11} e^{-k_2(u-t)}) du$$

$$A(t,T) = -\frac{(u-t)^2}{2} y_{00} + y_{10}[\frac{-1}{k_2}[(u-t) + \frac{1}{k_2}]e^{-k_2(u-t)}] - \frac{1}{k_2}[\frac{y_{10}}{k_2} e^{-k_2(u-t)} - \frac{y_{11}}{2k_2^2} e^{-2k_2(u-t)} - y_{10} u - \frac{y_{11}}{k_2^2} e^{-k_2(u-t)}]\Big|^{u=}_{u=}$$

$$A(t,T) = -\frac{(u-t)^2}{2} y_{00} + y_{10}[\frac{-1}{k_2}[(u-t) + \frac{1}{k_2}]e^{-k_2(u-t)} - \frac{1}{k_2^2} e^{-k_2(u-t)} - \frac{u}{k_2}] - \frac{y_{11}}{k_2^2}[\frac{1}{2} e^{-2k_2(u-t)} - e^{-k_2(u-t)}]\Big|^{u=T}_{u=t}$$

$$A(t,T) = -\frac{(T-t)^2}{2} y_{00} + y_{10}[\frac{-(T-t)}{k_2} e^{-k_2(T-t)} - \frac{1}{k_2^2} e^{-k_2(T-t)} + \frac{1}{k_2^2} + \frac{T-t}{k_2} - \frac{1}{k_2^2} + \frac{e^{-k_2(T-t)}}{k_2^2}] - \frac{y_{11}}{2k_2^2}[(e^{-2k_2(T-t)} - 1) \cdot$$

$$A(t,T) = -\frac{(T-t)^2}{2} y_{00} + y_{10}[\frac{-(T-t)}{k_2} e^{-k_2(T-t)} + \frac{T-t}{k_2}] - \frac{y_{11}}{2k_2^2}[(e^{-2k_2(T-t)} - 1) - (2e^{-k_2(T-t)} - 2)]$$

$$A(t,T) = -\frac{(T-t)^2}{2} y_{00} - y_{10} \frac{1}{k_2}(T-t)(1 - e^{-k_2(T-t)}) - \frac{y_{11}}{2k_2^2}(1 - e^{-k_2(T-t)})^2$$

$$C(t,T)^T x(t) = \begin{bmatrix} (T-t) & \frac{-1}{k_2} e^{-k_2(T-t)} + \frac{1}{k_2} \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix}$$

$$C(t,T)^T x(t) = (T-t) x_1(t) + \frac{1}{k_2}(1 - e^{k_2(T-t)}) x_2(t)$$

$$\frac{P(0,T)}{P(0,t)} = e^{-a(T-t) - \frac{b}{2}(T^2 - t^2)}$$

$$F(t,T,x(t)) = \frac{P(0,T)}{P(0,t)} e^{A(t,T) + C(t,T)^T x(t))}$$

$$F(t,T,x(t)) = e^{-a(T-t)-\frac{b}{2}(T^2-t^2)}e^{-\frac{(T-t)^2}{2}y_{00}-y_{10}\frac{1}{k_2}(T-t)(1-e^{-k_2(T-t)})-\frac{y_{11}}{2k_2^2}(1-e^{-k_2(T-t)})^2+(T-t)x_1(t)+\frac{1}{k_2}(1-e^{k_2(T-t)})x_2(t)}$$

$$dx(t) = (y_t\mathbf{1} - kx(t))dt + \sigma_x^{*T}dW_t^*$$

$$\mathbf{1} = (1,1)^T$$

$$k = \begin{bmatrix} 0 & 0 \\ 0 & k_2 \end{bmatrix}$$

$$\sigma_x^* = \begin{bmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{bmatrix}$$

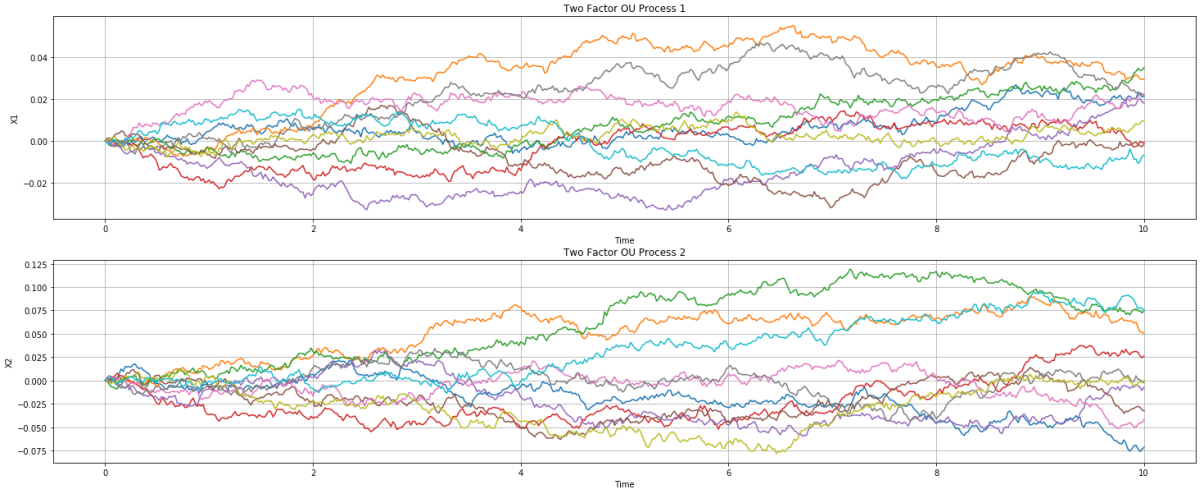
$$dx(t) = \begin{bmatrix} \sigma_1^2 t + \frac{1}{2k_2}p_x^2\sigma_1^2\sigma_2^2(e^{2k_2t}-1) & \frac{1}{k_2}p_x\sigma_1\sigma_2[\sigma_1 + \frac{\sigma_2}{2}e^{k_2} - (\sigma_1+\frac{\sigma_2}{2})e^{-k_2t}] \\ \frac{1}{k_2}p_x\sigma_1\sigma_2[\sigma_1 + \frac{\sigma_2}{2}e^{k_2t} - (\sigma_1+\frac{\sigma_2}{2})e^{-k_2t}] & \frac{\sigma_2^2}{2k_2}(p_x^2\sigma_1^2+1)(1-e^{-2k_2t}) \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} - \begin{bmatrix} 0 & 0 \\ 0 & k_2 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} dt + \begin{bmatrix} \sigma \\ ( \end{bmatrix}$$

$$dx_1(t) = [\sigma_1^2 t + \frac{1}{2k_2}p_x^2\sigma_1^2\sigma_2^2(e^{2k_2t}-1) + \frac{1}{k_2}p_x\sigma_1\sigma_2[\sigma_1 + \frac{\sigma_2}{2}e^{k_2} - (\sigma_1+\frac{\sigma_2}{2})e^{-k_2t}]]dt + \sigma_1 dW_t^*$$

$$dx_2(t) = [\frac{1}{k_2}p_x\sigma_1\sigma_2[\sigma_1 + \frac{\sigma_2}{2}e^{k_2t} - (\sigma_1+\frac{\sigma_2}{2})e^{-k_2t}] + \frac{\sigma_2^2}{2k_2}(p_x^2\sigma_1^2+1)(1-e^{-2k_2t}) - k_2x_2(t)]dt + \sigma_2 dW_t^*$$

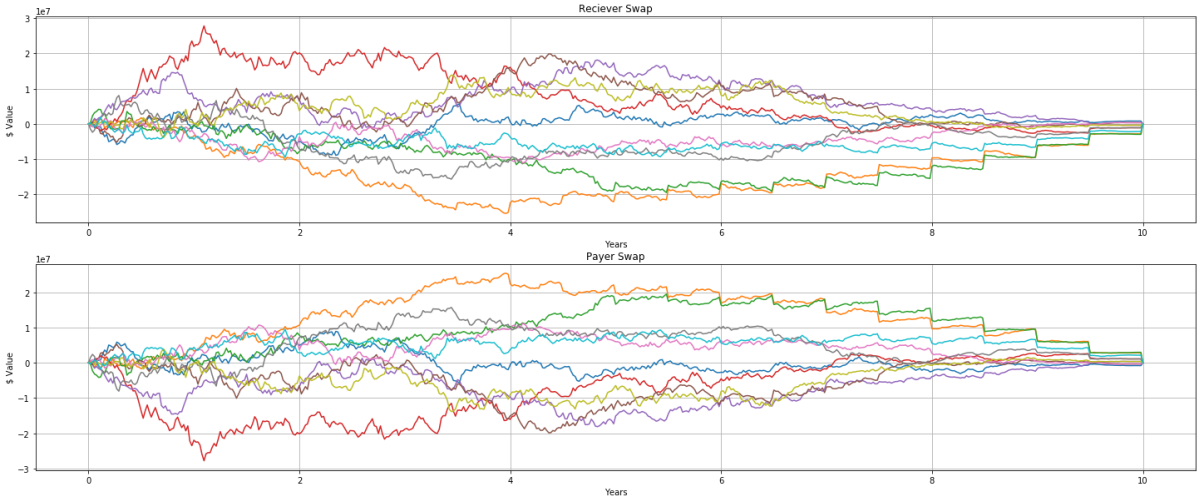
# Simulation of 2-factor Markov Model and OU process

In [6]:
```
Plot(ts,path1,path2,"Two Factor OU Process 1","Two Factor OU Process
2","Time","Time","X1","X2")
```



## Evolution of spot rates and corresponding valuation of SOFR swap both Payer and Reciever ends

In [14]:
```
Plot(ts,v_swap_10*SwapNt,-v_swap_10*SwapNt,"Reciever Swap","Payer Swa
p","Years","Years","$ Value","$ Value")
```




## In a Cox process setting, we can use previous results (for Recovery Amount pricing) to state that a contract paying some possibly stochastic amount Y(τ) at the time of default, provided default takes place before time T, will have time 0 value of

$$\mathrm{E}(1_{\boldsymbol{\tau}<T} Y(\boldsymbol{\tau}) e^{-\int_0^t r(u)\,du}) = \mathrm{E}(\int_0^{T_{max}} Y(s)\lambda(s) e^{-\int_0^s \lambda(u)\,du} e^{-\int_0^s r(u)\,du} ds)$$

## For a portfolio of trades in a netting set, we can associate it with an exposure variable E(t), which is the amount that is at risk if there is a default and the portfolio is settled at time t.

## Expected Exposure evolution for swap

If the portfolio value (to us) is V(t), and there is no collateral, then E(t) = V(t)+ Mathematically speaking the Expected Exposure (EE) and Present Value of Expected Exposure (PVEE) is given as

$$EE(t) = \mathrm{E}(E(t)) and PVEE(t) = \mathrm{E}(E(t) * \mathrm{e}^{-\int_0^t r(u)\,du})$$

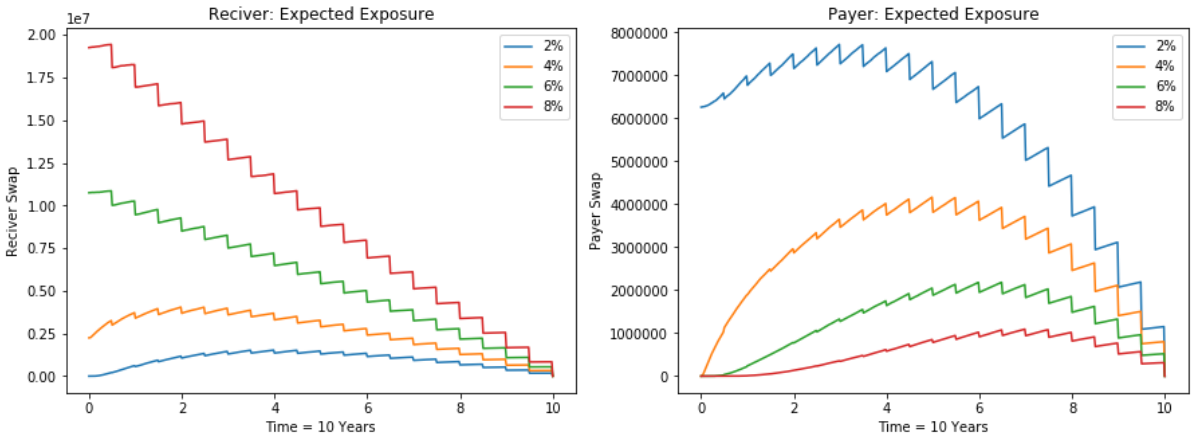
EE and PVEE profiles with various coupons for Swap portfolio with no collateral for various Swap Coupon rates

In [28]:
```
figure, axis = plt.subplots(1,2)
figure.set_figwidth(15)
figure.set_figheight(5)

axis[0].set_title("Reciver: Expected Exposure")
axis[0].set_xlabel("Time = 10 Years")
axis[0].set_ylabel("Reciver Swap")
axis[0].plot(ts,res_nocol_EE_rcv[0,:], label = "2%")
axis[0].plot(ts,res_nocol_EE_rcv[1,:], label = "4%")
axis[0].plot(ts,res_nocol_EE_rcv[2,:], label = "6%")
axis[0].plot(ts,res_nocol_EE_rcv[3,:], label = "8%")
axis[0].legend()

axis[1].set_title("Payer: Expected Exposure")
axis[1].set_xlabel("Time = 10 Years")
axis[1].set_ylabel("Payer Swap")
axis[1].plot(ts,res_nocol_EE_pay[0,:], label = "2%")
axis[1].plot(ts,res_nocol_EE_pay[1,:], label = "4%")
axis[1].plot(ts,res_nocol_EE_pay[2,:], label = "6%")
axis[1].plot(ts,res_nocol_EE_pay[3,:], label = "8%")
axis[1].legend()

plt.show()
```



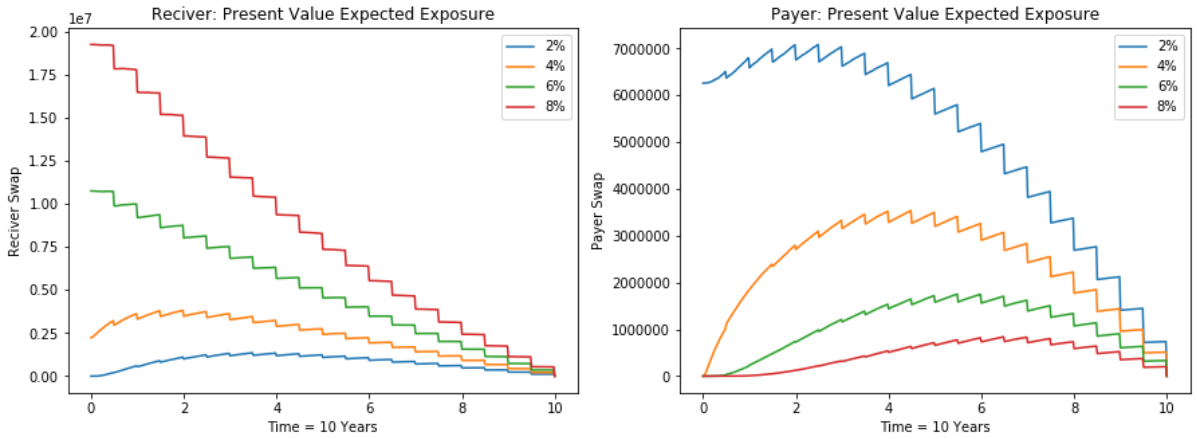
In [29]:
```
figure, axis = plt.subplots(1,2)
figure.set_figwidth(15)
figure.set_figheight(5)

axis[0].set_title("Reciver: Present Value Expected Exposure")
axis[0].set_xlabel("Time = 10 Years")
axis[0].set_ylabel("Reciver Swap")
axis[0].plot(ts,res_nocol_PVEE_rcv[0,:], label = "2%")
axis[0].plot(ts,res_nocol_PVEE_rcv[1,:], label = "4%")
axis[0].plot(ts,res_nocol_PVEE_rcv[2,:], label = "6%")
axis[0].plot(ts,res_nocol_PVEE_rcv[3,:], label = "8%")

axis[0].legend()

axis[1].set_title("Payer: Present Value Expected Exposure")
axis[1].set_xlabel("Time = 10 Years")
axis[1].set_ylabel("Payer Swap")
axis[1].plot(ts,res_nocol_PVEE_pay[0,:], label = "2%")
axis[1].plot(ts,res_nocol_PVEE_pay[1,:], label = "4%")
axis[1].plot(ts,res_nocol_PVEE_pay[2,:], label = "6%")
axis[1].plot(ts,res_nocol_PVEE_pay[3,:], label = "8%")
axis[1].legend()

plt.show()
```




# Collateralized Positions and Valuation Adjustments

## Variation Margin

The Credit Support Annex document describes how collateral is computed; how often; and how quickly it must be posted. Variation Margin is posted to track moves in the value of the underlying portfolio, making sure that the exposure originating from directional moves are not allowed to grow too big. Variation Margin accounts for a Margin Period of Risk (number of days between default actually happening versus the cashflows are recieved / contract is settled) given by $\delta$. The exposure is now defined as

$$E(t) = (V(t) - V(t - \delta)) +$$

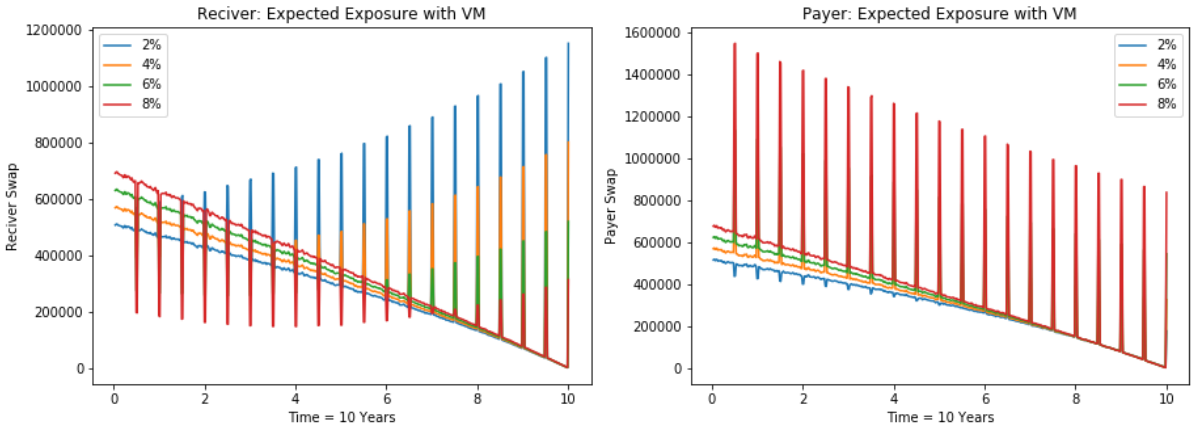
In [30]:
```
figure, axis = plt.subplots(1,2)
figure.set_figwidth(15)
figure.set_figheight(5)

axis[0].set_title("Reciver: Expected Exposure with VM")
axis[0].set_xlabel("Time = 10 Years")
axis[0].set_ylabel("Reciver Swap")
axis[0].plot(ts[2:],res_vm_EE_rcv[0,2:], label = "2%")
axis[0].plot(ts[2:],res_vm_EE_rcv[1,2:], label = "4%")
axis[0].plot(ts[2:],res_vm_EE_rcv[2,2:], label = "6%")
axis[0].plot(ts[2:],res_vm_EE_rcv[3,2:], label = "8%")
axis[0].legend()

axis[1].set_title("Payer: Expected Exposure with VM")
axis[1].set_xlabel("Time = 10 Years")
axis[1].set_ylabel("Payer Swap")

axis[1].plot(ts[2:],res_vm_EE_pay[0,2:], label = "2%")
axis[1].plot(ts[2:],res_vm_EE_pay[1,2:], label = "4%")
axis[1].plot(ts[2:],res_vm_EE_pay[2,2:], label = "6%")
axis[1].plot(ts[2:],res_vm_EE_pay[3,2:], label = "8%")
axis[1].legend()

plt.show()
```

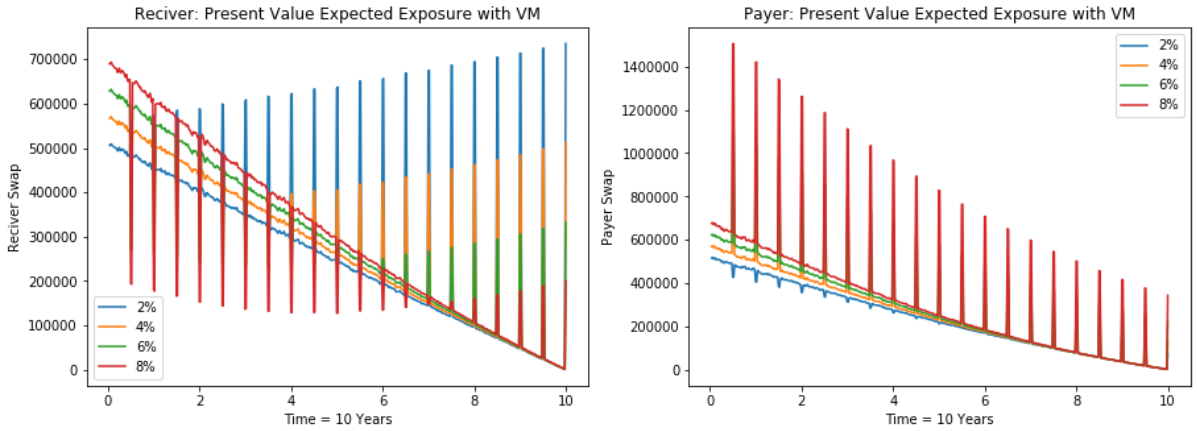


In [31]:
```
figure, axis = plt.subplots(1,2)
figure.set_figwidth(15)
figure.set_figheight(5)

axis[0].set_title("Reciver: Present Value Expected Exposure with VM")
axis[0].set_xlabel("Time = 10 Years")
axis[0].set_ylabel("Reciver Swap")
axis[0].plot(ts[2:],res_vm_PVEE_rcv[0,2:], label = "2%")
axis[0].plot(ts[2:],res_vm_PVEE_rcv[1,2:], label = "4%")
axis[0].plot(ts[2:],res_vm_PVEE_rcv[2,2:], label = "6%")
axis[0].plot(ts[2:],res_vm_PVEE_rcv[3,2:], label = "8%")
axis[0].legend()

axis[1].set_title("Payer: Present Value Expected Exposure with VM")
axis[1].set_xlabel("Time = 10 Years")
axis[1].set_ylabel("Payer Swap")
axis[1].plot(ts[2:],res_vm_PVEE_pay[0,2:], label = "2%")
axis[1].plot(ts[2:],res_vm_PVEE_pay[1,2:], label = "4%")
axis[1].plot(ts[2:],res_vm_PVEE_pay[2,2:], label = "6%")
axis[1].plot(ts[2:],res_vm_PVEE_pay[3,2:], label = "8%")
axis[1].legend()

plt.show()
```



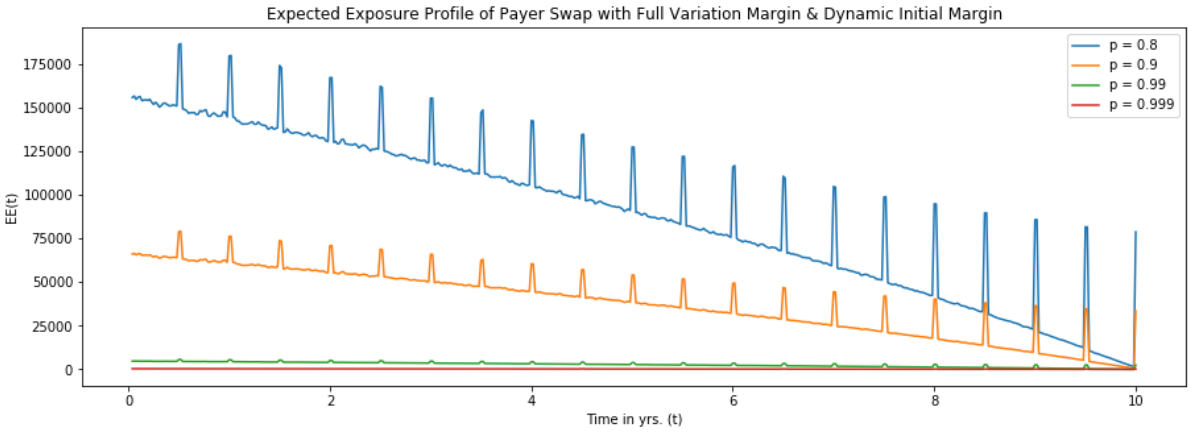

# Intial Margin

IM is an additional buffer that covers against adverse moves of the portfolio on the MPoR. If the IM is X, then (as both counterparties must post) 2X in total will be squirreled away for a rainy day. (Assuming IM is symmetric, which is often nearly the case). The regulatory rule is – roughly – that IM be set as the 99% VaR of V(t)–V(t –δ), with δ = 10 business days

In [32]:
```
plt.figure(figsize=(15,5))
plt.title("Expected Exposure Profile of Payer Swap with Full Variation Margin & Dynamic Initial Margin")
plt.xlabel("Time in yrs. (t)")
plt.ylabel("EE(t)")

for i in range(len(probs)):
    plt.plot(ts[2:],VM_IM_EE_pay[i,2:], label=f'p = {probs[i]}')

plt.legend()
plt.show()
```




# CVA

Credit valuation adjustments (CVAs) are accounting adjustments made to reserve a portion of profits on uncollateralized financial derivatives. They are charged by a bank to a risky (capable of default) counterparty to compensate the bank for taking on the credit risk of the counterparty during the life of the transaction. For small Margin Period of Risk and assuming that the interest rates and default intensity are independent, where CVA can be approximated as for a Counterparty A as:

$$CVA \approx (1 - R_A) * \int_0^{T_{max}} PVEE(s) * Q(\tau \in ds)$$

where Q(τA ∈ ds) can be found from the time 0 survival probability curve for A, denoted XA(0, t), t > 0. That is, XA(0,t) = Q(τA > t)

# Unilateral CVA

In [35]:
```
df_UNI_B_2 = pd.DataFrame({'Coupon':harr,
                    'Unilateral RECIVER: No Collateral': CVA(f1,f2,RR,l_
A,l_B,ts,"Uni")[0],
                    'Unilateral RECIVER: Only VM': CVA(fv1,fv2,RR,l_A,l_
B,ts,"Uni")[0],
                    'Unilateral RECIVER: VM + IM': CVA(fvi1,fvi2,RR,l_A,l
_B,ts,"Uni")[0],
                    'Unilateral PAYER: No Collateral': CVA(f1,f2,RR,l_A,l
_B,ts,"Uni")[1],
                    'Unilateral PAYER: Only VM': CVA(fv1,fv2,RR,l_A,l_B,t
s,"Uni")[1],
                    'Unilateral PAYER: VM + IM': CVA(fvi1,fvi2,RR,l_A,l_
B,ts,"Uni")[1]
                   })
pd.options.display.float_format = '{:,.2f}'.format

print("Unilateral CVA calculations using PVEE")

df_UNI_B_2
```

```
Unilateral CVA calculations using PVEE
```

Out[35]:

| | Coupon | Unilateral RECIVER: No Collateral | Unilateral RECIVER: Only VM | Unilateral RECIVER: VM + IM | Unilateral PAYER: No Collateral | Unilateral PAYER: Only VM | Unilateral PAYER: VM + IM |
|---|---|---|---|---|---|---|---|
| **0** | 0.02 | 240,277.35 | 64,410.69 | 547.11 | 1,004,092.10 | 64,989.33 | 552.03 |
| **1** | 0.04 | 634,955.30 | 67,283.60 | 571.51 | 425,652.84 | 69,172.63 | 587.56 |
| **2** | 0.06 | 1,344,511.60 | 74,378.07 | 631.78 | 162,091.92 | 77,577.48 | 658.95 |
| **3** | 0.08 | 2,213,208.00 | 82,266.49 | 698.78 | 57,671.12 | 86,776.29 | 737.09 |

As seen in the exposure profile, the payer has the most exposure at a 2% coupon and the least at a 8% coupon. It is the opposite case with the reciever i.e. the most exposure at a 8% coupon and the least at 2%. The CVA value is proportional to the expected positive exposure of the trade.

With the Variation Margin taken into account, the valuation is reduced and the exposure trends for both payer and reciever swaps are the same.

Further, with the Initial Margin included, the CVA is reduced to a very small amount as expected.

In both cases of Only VM and VM + IM, the CVA values are directly proportional to the coupon rate, increasing as the coupon rate increases.

# BiLateral CVA

In [37]:
```
df_BI_B_2 = pd.DataFrame({'Coupon':harr,
                          'BiLateral RECIVER: No Collateral': CVA(f1,f2,RR,l_A,
l_B,ts,"Bi")[0],
                          'BiLateral RECIVER: Only VM': CVA(fv1,fv2,RR,l_A,l_B,
ts,"Bi")[0],
                          'BiLateral RECIVER: VM + IM': CVA(fvi1,fvi2,RR,l_A,l_
B,ts,"Bi")[0],
                          'BiLateral PAYER: No Collateral': CVA(f1,f2,RR,l_A,l_
B,ts,"Bi")[1],
                          'BiLateral PAYER: Only VM': CVA(fv1,fv2,RR,l_A,l_B,t
s,"Bi")[1],
                          'BiLateral PAYER: VM + IM': CVA(fvi1,fvi2,RR,l_A,l_B,
ts,"Bi")[1]
                         })
pd.options.display.float_format = '{:,.2f}'.format

print("BiLateral CVA calculations using PVEE")
df_BI_B_2
```

```
BiLateral CVA calculations using PVEE
```

Out[37]:

| | Coupon | BiLateral RECIVER: No Collateral | BiLateral RECIVER: Only VM | BiLateral RECIVER: VM + IM | BiLateral PAYER: No Collateral | BiLateral PAYER: Only VM | BiLateral PAYER: VM + IM |
|---|---|---|---|---|---|---|---|
| **0** | 0.02 | 218,751.35 | 60,350.17 | 512.62 | 938,581.41 | 61,207.82 | 519.91 |
| **1** | 0.04 | 588,990.29 | 63,271.42 | 537.44 | 392,036.15 | 64,938.66 | 551.60 |
| **2** | 0.06 | 1,260,755.05 | 70,328.46 | 597.38 | 147,016.70 | 72,805.29 | 618.42 |
| **3** | 0.08 | 2,082,133.94 | 78,108.11 | 663.46 | 51,611.39 | 81,394.53 | 691.37 |

As seen in the exposure profile, the payer has the most exposure at a 2% coupon and the least at a 8% coupon. It is the opposite case with the reciever i.e. the most exposure at a 8% coupon and the least at 2%. The CVA value is proportional to the expected positive exposure of the trade.

With the Variation Margin taken into account, the valuation is reduced and the exposure trends for both payer and reciever swaps are the same.

Further, with the Initial Margin included, the CVA is reduced to a very small amount as expected.

In both cases of Only VM and VM + IM, the CVA values are directly proportional to the coupon rate, increasing as the coupon rate increases.

# FVA

Funding valuation adjustment reflects the funding cost of uncollateralised derivatives above the risk-free rate of return. It represents the costs and benefits of writing a hedge for a client who is not posting collateral, and then hedging that trade with a collateralised one in the interbank market. For a Bank B with survival intensity given by $\lambda_B$ and Recovery amount $R_B$ the spread is given as $S_B(t) = \lambda_B(t)(1 - R_B)$ FVA for uncollateralized position is given as:

$$FVA = \int_0^{T_{max}} \mathrm{E}(S_B(t) * V(t) * e^{-\int_0^t \lambda_B(u)\,du} * e^{-\int_0^t r(u)\,du})dt$$

and since $V(t) = V(t)^+ - V(t)^-$ the FVA can be broken into FCA (Funding Cost Adjustment) and FBA (Funding Benifit Adjustment) therefore, FVA = FCA - FBA

In [41]:
```
S_B = 0.012
df_FVA = pd.DataFrame({'Coupon':harr,
                       'Reciver FVA': FVA(f1,f2,l_B,ts,S_B),
                        'Payer FVA': -1*FVA(f1,f2,l_B,ts,S_B),
                      })
pd.options.display.float_format = '{:,.2f}'.format
df_FVA
```

Out[41]:

| | Coupon | Reciver FVA | Payer FVA |
|---|---|---|---|
| **0** | 0.02 | -364,993.24 | 364,993.24 |
| **1** | 0.04 | 100,169.39 | -100,169.39 |
| **2** | 0.06 | 565,332.03 | -565,332.03 |
| **3** | 0.08 | 1,030,494.66 | -1,030,494.66 |

FVA Values are negative when FBA is greater than FCA i.e. essentilly when B is recieving funding from its counterparty. This is visible when coupon rate is less than the par rate (~3.4%).

FVA is negative if FCA is smaller than FBA. This is the case if B is short derivative assets, and is therefore in the aggregate receiving funding from its counterparties.

For example B enters a reciver swap at 2% coupon (that is less than 3.4%) the counterparty must incentivize, B to enter the transaction thereby making a payment. Using this cash B can theoretically extinguish its outstanding debt and thus the funding benefit is greater than the funding cost, implying a negative FVA.

The inverse would be true if we are paying a coupon higher than hpar (payer swap) or we were the receiver in the swap desrcibed above. The payer FVA and receiver FVA change between positive and negative values above the hpar rate because of the cost of carry on the trade.

# MVA

FVA originating with IM postings is denoted Margin Valuation Adjustment (MVA). For IM(t) be the initial margin posted at time t then, in the spirit of FVA, we can define

$$MVA = \int_0^{T_{max}} \mathrm{E}(IM(t) * S_B(t) * e^{-\int_0^t \lambda_B(u)\,du} * e^{-\int_0^t r(u)\,du}) dt$$

In [43]:
```
S_B = 0.012
df_MVA = pd.DataFrame({'Coupon':harr,
                       'Reciver MVA': MVA(fv1,fv2,l_B,ts,S_B,0.99)[0].resha
pe(1,-1)[0],
                      'Payer MVA': MVA(fv1,fv2,l_B,ts,S_B,0.99)[1].reshape
(1,-1)[0],
                     })
pd.options.display.float_format = '{:,.2f}'.format
df_MVA
```

Out[43]:

| | Coupon | Reciver MVA | Payer MVA |
|---|---|---|---|
| **0** | 0.02 | 181,597.69 | 173,625.44 |
| **1** | 0.04 | 185,791.17 | 194,525.46 |
| **2** | 0.06 | 192,829.57 | 218,270.41 |
| **3** | 0.08 | 202,186.06 | 244,333.43 |

In order to reduce CVA from roughly from a magnidute of 10^4 to 10^2, an MVA charge of roughly $10^5 order has been created. The MVA ends up being over a lot greater than the CVA with variation margin and initial margin (at p=99\%); this is difficult to justify on economic terms, but is required by regulation.

# Regulatory Capital under Basel 2 and the IRB Approach

Replacing the old Current Exposure Method (CEM) and SA-CCR owing to their primitive basic rules for regulatory credit capital, the instutions under Basel 2 approach use an Internal Ratings Based Approach (IRB) to ensure that the regulatory capital is closer to the economic capital. The infinite diversification (similar to Vasicek portfolio) helps achieve portfolio Invariance: a capital treatment given to a loan position with a given counterparty should be identical from one bank to the next, and should not depend on exposure to other counterparties. The Risk Weighted Capital required required for Credit Risk is RWA = 12.5 *EAD* RW where

$$RW = (LossGivenDefault) * (\mathbf{\Phi}(\frac{\mathbf{\Phi}^{-1}(p) - \sqrt{(}\rho(p))\mathbf{\Phi}^{-1}(q)}{\sqrt{(}1-\rho(p)} - p)) * k(M,p)$$

p = 1-year probability of default (PD)\ q = 0.001 (“once in a thousand years”)\ M = effective maturity\
$\rho(p) = 0.24 - 0.12(1 - e^{-50p})$

The computation of EAD and M is governed by a regulatory framework called internal models methodology (IMM). The purpose of IMM is to find a way to take an arbitrary derivatives portfolio (including its collateral) and replace it with a “representative” loan notional. Expected Positive Exposure (EPE) is the equivalent for such a loan notional. Given by

$$E(E(\tau)\,|\,\tau \leq T) \approx \frac{1}{T}\int_0^T EE(t)dt \triangleq (EPE(t))dt$$

However, it ignores wrong way risk and is not suitable for portfolios with maturity << 1 year. Hence a new measure called Effective Expected Positive Exposure (EEPE) has been found as running maximum of EE profiles:

$$EE^*(t_i) = max(EE^*(t_{i-1}), EE(t_i))$$

and

$$EEPE = \frac{1}{T}\int_0^T EE^*(t)dt$$

In [49]:
```
pd.DataFrame({'Coupon':harr,
              'EPE(1) Reciever': EPE(res_nocol_EE_rcv),
              'EPE(1) Payer' : EPE(res_nocol_EE_pay),
              'EEPE(1) Reciever': EEPE(res_nocol_EE_rcv),
              'EEPE(1) Payer': EEPE(res_nocol_EE_pay),
              'EAD Reciever': EAD(EEPE(res_nocol_EE_rcv),1.4,CVA(res_noc
ol_PVEE_rcv,res_nocol_PVEE_pay,RR,l_A,l_B,ts,"Uni")[0]),
              'EAD Payer': EAD(EEPE(res_nocol_EE_pay),1.4,CVA(res_nocol_
PVEE_rcv,res_nocol_PVEE_pay,RR,l_A,l_B,ts,"Uni")[1]),
              'M Reciever' : np.array([calc_M(res_nocol_EE_rcv[i],dfs,t
s) for i in range(0,4)]),
              'M Payer' : np.array([calc_M(res_nocol_EE_pay[i],dfs,ts) f
or i in range(0,4)])
             })
```

Out[49]:

| | Coupon | EPE(1) Reciever | EPE(1) Payer | EEPE(1) Reciever | EEPE(1) Payer | EAD Reciever | EAD Payer | Rec |
|---|---|---|---|---|---|---|---|---|
| **0** | 0.02 | 943,345.46 | 5,824,151.00 | 1,305,873.62 | 7,538,140.10 | 1,750,797.47 | 9,977,842.72 | |
| **1** | 0.04 | 2,612,302.82 | 2,858,466.56 | 3,931,289.44 | 3,548,330.96 | 5,277,077.62 | 4,708,905.50 | |
| **2** | 0.06 | 5,690,089.95 | 1,301,611.88 | 10,855,763.53 | 1,586,112.31 | 14,850,653.50 | 2,116,405.84 | |
| **3** | 0.08 | 9,583,112.58 | 559,992.71 | 19,435,332.47 | 676,068.15 | 26,701,908.97 | 906,851.92 | |

In [52]:
```
AIRB_1  = pd.DataFrame({'Coupon':harr,
              'EAD Reciever': EAD(EEPE(res_nocol_EE_rcv),1.4,CVA(res_noc
ol_PVEE_rcv,res_nocol_PVEE_pay,RR,l_A,l_B,ts,"Uni")[0]),
              'EAD Payer': EAD(EEPE(res_nocol_EE_pay),1.4,CVA(res_nocol_
PVEE_rcv,res_nocol_PVEE_pay,RR,l_A,l_B,ts,"Uni")[1]),
              'M Reciever' : np.array([calc_M(res_nocol_EE_rcv[i],dfs,t
s) for i in range(0,4)]),
              'M Payer' : np.array([calc_M(res_nocol_EE_pay[i],dfs,ts) f
or i in range(0,4)])
             })
AIRB_1['RW reciever'] = Risk_Weight(0.005,0.6,AIRB_1['M Reciever'],0.00
1)
AIRB_1['RW payer']    = Risk_Weight(0.005,0.6,AIRB_1['M Payer'],0.001)
AIRB_1['RC reciever'] = AIRB_1['RW reciever'] * AIRB_1['EAD Reciever']
AIRB_1['RC payer']    = AIRB_1['RW payer'] * AIRB_1['EAD Payer']

AIRB_1
```

Out[52]:

| | Coupon | EAD Reciever | EAD Payer | M Reciever | M Payer | RW reciever | RW payer | RC reciever | RC paye |
|---|---|---|---|---|---|---|---|---|---|
| **0** | 0.02 | 1,750,797.47 | 9,977,842.72 | 5.00 | 5.00 | 0.11 | 0.11 | 184,304.61 | 1,050,357.0 |
| **1** | 0.04 | 5,277,077.62 | 4,708,905.50 | 5.00 | 5.00 | 0.11 | 0.11 | 555,512.43 | 495,701.5 |
| **2** | 0.06 | 14,850,653.50 | 2,116,405.84 | 4.85 | 5.00 | 0.10 | 0.11 | 1,535,023.70 | 222,791.8 |
| **3** | 0.08 | 26,701,908.97 | 906,851.92 | 4.58 | 5.00 | 0.10 | 0.11 | 2,670,935.84 | 95,463.3 |