ELS Pricing Final Project

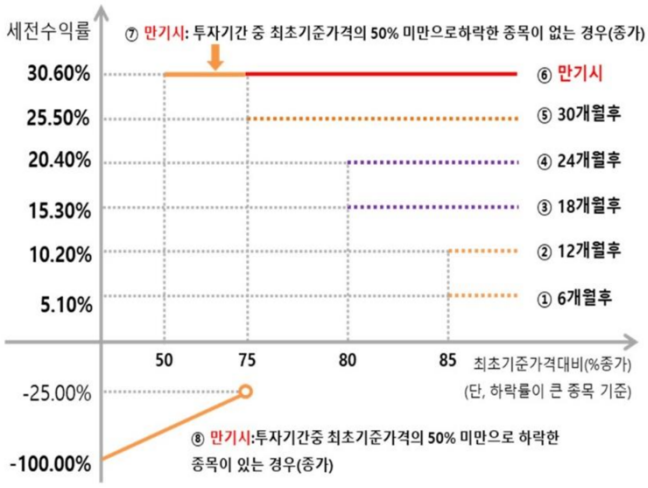
20223888 이선우

1. Name of ELS

TRUE ELS 15365회 (SK하이닉스, NAVER)

기초자산 : SK하이닉스 보통주, NAVER 보통주
발행인이 제시한 공정가액 : 8863.69원
만기 평가일 : 2025년 8월 29일 (3년)

2. Payoff Structure



본 상품의 주요 Payoff는 좌측과 같다.

조건 충족 시 수익률 : 연 10.20%
조건 미충족 시 최대손실 : 100%

조기 상환 Step Down 형식의 상품으로,
3년간, 6개월 간격, 조기상환여부를 결정한다.

적용되는 행사가격과 실현가능 수익률은 각각 아래와 같다.

- 06개월 : 2종목 모두 종가기준 85% 이상이면 5.10% 수익률
- 12개월 : 2종목 모두 종가기준 85% 이상이면 10.20% 수익률
- 18개월 : 2종목 모두 종가기준 80% 이상이면 15.30% 수익률
- 24개월 : 2종목 모두 종가기준 80% 이상이면 20.40% 수익률
- 30개월 : 2종목 모두 종가기준 75% 이상이면 25.50% 수익률
- 36개월 : 2종목 모두 종가기준 75% 이상이면 30.60% 수익률

만일, 위 6개의 조기상환, 만기상환이 발생하지 않으면, 아래의 수익을 제공한다. (Barrier 50%)

- 투자기간동안 두 종목 모두 최초가격의 50% 미만 하락한적이 없다면 > 30.60% 수익률
- 투자기간동안 두 종목 중 하나라도 최초가격의 50% 미만으로 하락한 적이 있는 경우
  >>> 원금 × 기준종목의 (만기평가가격/최초기준가격) 중 더 낮은 수익률 (Min)

3. Specify Parameters Precisely

본 상품을 평가하기 위해 필요한 Parameter들은 다음과 같다.

변동성(자산1)	0.3888	무위험이자율	0.0282
변동성(자산2)	0.3962	두 자산의 상관계수	0.5287
배당률(자산1)	0.0162	배당률(자산2)	0.0022

투자설명서에 제시된 값을 이용했다.

- SK하이닉스, NAVER의 변동성: VIX 방법론을 이용해 산출
- 상관계수: 180영업일의 Daily Return에 대한 Historical Correlation을 통해 산출

시장모델을 이용

- 무위험이자율: 발행일인 8월 31일의 "90일 CD금리"
- 각 자산의 배당률: 8월 31일 이전 180영업일의 "평균배당수익률"
- 각 자산의 변동성: VIX 방법론을 이용해 산출된 변동성
- 자산 간 상관계수: 최근 180일 영업일간 Daily Return의 Historical Correlation

OSFDM 계산을 위해 필요한 Parameter는 다음과 같다.

만기(T)	3	낙인 배리어(B)	50%
최초가격(자산1)	100	최대주가(자산1)	3 * 100
최초가격(자산2)	100	최대주가(자산2)	3 * 100
주가간격(M, parsing)	91	시간간격(N)	360 * 3 = 1080
조기상환수익률	수익률 구간	조기상환기준가격	상환 행사가격

투자설명서를 기반으로 Parameter를 설정

- (자산1, 자산2) 최초가격과 최대주가, 주가간격: 계산의 편의를 위해 동일하게 설정.
- 시간간격: 하루 단위를 사용해 계산을 진행
- 조기상환수익률: [5.1%, 10.2%, 15.3%, 20.4%, 25.5%, 30.6%] (투자설명서 기반)
- 조기상환기준가격: [85%, 85%, 80%, 80%, 75%, 75%] (투자설명서 기반)

4. Price the ELS using OS FDM

$X =$ SK하이닉스, $Y =$ NAVER라 하자.

계산의 편의를 위해 다음과 같이 구간을 정의한다.

$$\delta\tau = \frac{T-t}{N} = k, \qquad (\delta x, \delta y) = \left(\frac{x_{max}}{Mx}, \frac{y_{max}}{My}\right) = h$$

생성되는 Mesh Point의 좌표는 다음과 같다.

$$Mesh\ Point(\tau_n, x_i, y_j) = u(\tau_n, x_i, y_j) \equiv (n\delta\tau, i\delta x, j\delta y) = \boldsymbol{u_{ij}^n}$$

X, Y 두 종목에 대한 블랙-숄즈 방정식과 그 미분은 아래와 같다.

$$u_\tau = \mu_X x u_X + \mu_Y y u_Y + \frac{1}{2}\sigma_X^2 x^2 u_{xx} + \frac{1}{2}\sigma_Y^2 y^2 u_{yy} + \rho\sigma_X\sigma_Y xy u_{xy} - ru$$

$$\frac{\partial u}{\partial \tau} = \frac{(\sigma_X x)^2}{2}\frac{\partial^2 u}{\partial x^2} + \frac{(\sigma_Y y)^2}{2}\frac{\partial^2 u}{\partial y^2} + rx\frac{\partial u}{\partial x} + ry\frac{\partial u}{\partial y} + \rho\sigma_x\sigma_y xy\frac{\partial u^2}{\partial x \partial y} - ru$$

위 식에서 상관계수항과 상수항은 평균해서 정리해준다. ($\rho\sigma_X\sigma_Y xy u_{XY},\ ru$)

$$\begin{aligned}
\frac{u_{i,j}^{n^*} - u_{i,j}^{n}}{k} &= L_{OS}^{x} u_{i,j}^{n^*} \\
&= \mu_x x_i \frac{u_{i+1,j}^{n^*} - u_{i,j}^{n^*}}{h} + (\sigma_X x_X)^2 \frac{u_{i+1,j}^{n^*} - 2u_{i,j}^{n^*} + u_{i-1,j}^{n^*}}{2h^2} \\
&\quad + \frac{\rho\sigma_X\sigma_Y x_i y_j}{2} \frac{u_{i+1,j+1}^{n} + u_{i-1,j-1}^{n} - u_{i-1,j+1}^{n} - u_{i+1,j-1}^{n}}{4h^2} - \frac{r}{2} u_{i,j}^{n^*} \\
\frac{u_{i,j}^{n+1} - u_{i,j}^{n^*}}{k} &= L_{OS}^{y} u_{i,j}^{n+1} \\
&= \mu_Y y_j \frac{u_{i,j+1}^{n+1} - u_{i,j}^{n+1}}{h} + (\sigma_Y y_j)^2 \frac{u_{i,j+1}^{n+1} - 2u_{i,j}^{n+1} + u_{i,j-1}^{n+1}}{2h^2} \\
&\quad + \frac{\rho\sigma_X\sigma_Y x_i y_j}{2} \frac{u_{i+1,j+1}^{n^*} + u_{i-1,j-1}^{n^*} - u_{i-1,j+1}^{n^*} - u_{i+1,j-1}^{n^*}}{4h^2} - \frac{r}{2} u_{i,j}^{n+1}
\end{aligned}$$

두 식을 더하면 다음과 같다

$$\frac{u_{i,j}^{n^*} - u_{i,j}^{n}}{k} + \frac{u_{i,j}^{n+1} - u_{i,j}^{n^*}}{k} = \frac{u_{i,j}^{n+1} - u_{i,j}^{n}}{k} = L_{OS}^{x} u_{i,j}^{n^*} + L_{OS}^{y} u_{i,j}^{n+1}$$

위 식에서, x, y 각각 implicit FDM을 통해 해를 구하고, 식을 $a_i,\ b_i,\ c_i,\ d_i$로 간소화한다.

$L_{OS}^{x} u_{i,j}^{n^*}$ 의 계수들, $\qquad a_i u_{i-1,j}^{n^*} + b_i u_{i,j}^{n^*} + c_i u_{i+1,j}^{n^*} = d^n$

$$a_i = -\frac{\sigma_X^2 x_i^2}{2h^2} + \frac{(rf - q_X)x_i}{2h}, \qquad b_i = \frac{1}{k} + \frac{\sigma_X^2 x_i^2}{2h^2} + \frac{(rf - q_X)}{2}, \qquad c_i = -\frac{\sigma_X^2 x_i^2}{2h^2} - \frac{(rf - q_X)x_i}{2h}$$

$$d^n = \frac{u_{i,j}^{n}}{k} + \frac{\rho\sigma_X\sigma_Y x_i y_j}{2} \frac{u_{i+1,j+1}^{n} + u_{i-1,j-1}^{n} - u_{i-1,j+1}^{n} - u_{i+1,j-1}^{n}}{8h^2}$$

$L_{OS}^{y} u_{i,j}^{n+1}$ 의 계수들, $\qquad a_j u_{i,j-1}^{n+1} + b_j u_{i,j}^{n+1} + c_j u_{i,j+1}^{n+1} = d^{n^*}$

$$a_i = -\frac{\sigma_Y^2 y_j^2}{2h^2} + \frac{(rf - q_Y)y_j}{2h}, \qquad b_i = \frac{1}{k} + \frac{\sigma_Y^2 y_j^2}{2h^2} + \frac{(rf - q_Y)}{2}, \qquad c_i = -\frac{\sigma_Y^2 y_j^2}{2h^2} - \frac{(rf - q_Y)y_j}{2h}$$

$$d^{n^*} = \frac{u_{i,j}^{n^*}}{k} + \frac{\rho\sigma_X\sigma_Y x_i y_j}{2} \frac{u_{i+1,j+1}^{n^*} + u_{i-1,j-1}^{n^*} - u_{i-1,j+1}^{n^*} - u_{i+1,j-1}^{n^*}}{8h^2}$$

```
a_i[:] = -0.5 * (x_vol * x[1:parsing-1]/hx) ** 2 + (rf - x_q)*x[1:parsing-1] / (2*hx)
b_i[:] = 1/k  + (x_vol * x[1:parsing-1]/hx) ** 2 + (rf - x_q)*0.5
c_i[:] = -0.5 * (x_vol * x[1:parsing-1]/hx) ** 2 - (rf - x_q)*x[1:parsing-1] / (2*hx)

a_j[:] = -0.5 * (y_vol * y[1:parsing-1]/hy)**2 + (rf - x_q)*y[1:parsing-1] / (2*hy)
b_j[:] = 1/k  + (y_vol * y[1:parsing-1]/hy)**2 + (rf - y_q) / 2
c_j[:] = -0.5 * (y_vol * y[1:parsing-1]/hy)**2 - (rf - x_q)*y[1:parsing-1] / (2*hy)

coef_list = [a_i, b_i, c_i, a_j, b_j, c_j]
```

먼저 x축을 통해 d^n과 계수들을 Thomas Algorithm에 적용해 $u_{i,j}^{n^*}$ 좌표의 값을 먼저 구한 뒤, 이후 y축을 통해 d^{n^*}과 계수들을 Thomas Algorithm에 적용해 $u_{i,j}^{n+1}$ 좌표의 값을 구한다.

```
def OSFDM (coef_list, u, parsing):

    coef = coef_list
    a_i = coef[0]
    b_i = coef[1]
    c_i = coef[2]
    a_j = coef[3]
    b_j = coef[4]
    c_j = coef[5]

    d_i = np.zeros(parsing-2)
    d_j = np.zeros(parsing-2)

    tmp = u
    # Solve with x-direction
    for j in range(1, parsing-1):
        d_i[0:parsing-1] = tmp[1:parsing-1, j] / k + (rho * x_vol * y_vol *  x[1:parsing-1] * y[j]) / 2 * \
                           (tmp[2:parsing, j+1] - tmp[2:parsing, j-1] - tmp[0:parsing-2, j+1] + tmp[0:parsing-2, j-1] ) / \
                           (4*hx ** 2)
        u[1:parsing-1, j] = Thomas_Algorithm(a_i, b_i, c_i, d_i)


    tmp = u
    # Solve with y-direction
    for i in range(1, parsing-1):
        d_j[0:parsing-1] = tmp[i, 1:parsing-1] / k + (rho * x_vol * y_vol * x[i] * y[1:parsing-1]) / 2 * \
                           (tmp[i+1, 2:parsing] - tmp[i+1,0:parsing-2] - tmp[i-1,2:parsing] + tmp[i-1,0:parsing-2])  / \
                           (4*hy**2)

        u[i, 1:parsing-1] = Thomas_Algorithm(a_j, b_j, c_j, d_j)

    return u
```

위와 같은 과정을, 각 Time Node 별로 반복적으로 시행해 시점별 가격을 반영해준다.
이때, u는 배리어를 터치하지 않았을 경우, w는 배리어를 터치하지 않았을 경우로 설정해준다.
반복하며 조기상환일에 해당하는 경우, 조건을 충족했다면 쿠폰을 적용해주며 u, w를 갱신한다.

```
checker = 0
for n in range(numT):

    if (n==step[checker]):
        x_Barrier_i = np.min(np.where(x >= x0_price * strike[checker+1]))
        y_Barrier_j = np.min(np.where(y >= y0_price * strike[checker+1]))

        u[x_Barrier_i:parsing-1, y_Barrier_j:parsing-1] = facevalue * (1 + coupon[checker+1])
        w[x_Barrier_i:parsing-1, y_Barrier_j:parsing-1] = facevalue * (1 + coupon[checker+1])

        checker += 1

    x_Barrier_i = np.min(np.where(x >= x0_price * barrier))
    y_Barrier_j = np.min(np.where(y >= y0_price * barrier))

    u[:, 0:y_Barrier_j+1] = w[:,0:y_Barrier_j+1]
    u[0:x_Barrier_i+1, :] = w[0:x_Barrier_i+1, :]

    u = OSFDM(coef_list, u, parsing)
    w = OSFDM(coef_list, w, parsing)


x_idx = np.where((x0_price-1 <= x) & (x < x0_price+1))[0][0]
y_idx = np.where((y0_price-1 <= y) & (y < y0_price+1))[0][0]

return x_idx, y_idx, u
```

OSFDM 이용해 구한 Two-Star ELS의 가격은 8897.68원으로, 이론가(8863.69원)과 차이가 있다.

```
    x_idx, y_idx, u = Two_Star_ELS_OSFDM(facevalue, x0_price, y0_price, parsing, rf, T, x_vol, y_vol, rho, x_q, y_q, barrier, coupon, strike)
    print(f'Two Star ELS PRICE by OSFDM = {np.round(u[x_idx, y_idx].item(),6)}')
✓ 0.4s

Two Star ELS PRICE by OSFDM = 8897.683918
```

5. Compare above result to simulation result

10000번의 몬테카를로 시뮬레이션을 이용해 구한 ELS의 공정가격과 표준편차는 아래와 같다

```
ELS price = 8869.668994447533, standard error = 30.773652273182503
```

공정가격: 8869.67원, 표준편차: 30.77원이다.

위 가격은 OSFDM을 통해 구한 8897.68원과 28.01원의 차이가 있으며, 아래의 이유에서 파생되는 것으로 볼 수 있다.

1) 몬테카를로 시뮬레이션의 시행횟수에 따른 불안정한 가격 변동폭 (많은 시행횟수의 필요성)
2) OSFDM 계산시 부족한 Parsing으로 인해 발생한 오차 (세밀한 Mesh의 필요성)

6. Greeks 및 Hedging Anlaysis

$$\Delta = \frac{V_{i+1}^n - V_{i-1}^n}{h}, \qquad \Gamma = \frac{V_{i+1}^n - 2V_i^n + V_{i-1}^n}{h^2}, \qquad Vega = \frac{V_i^n \big|_{\sigma} - V_i^n \big|_{\sigma'}}{\sigma - \sigma'}$$

위 수식에 따라 OSFDM으로 계산된 Greek은 아래와 같다

```
Two Star ELS "DELTA" by OSFDM = 1.2527498422259764
Two Star ELS "GAMMA" by OSFDM = -1.985277714180559
Two Star ELS "Vega" by OSFDM = 1.066787456716498
```

1) 델타는 기초자산 가격 변화량에 따른 상품 가격 변화량으로, 델타가 양의 값이므로, 기초자산가격인 주식가격이 증가한다면, Knock In 확률이 감소하거나, 조기상환 확률이 높아지므로 ELS의 상품가격이 상승할 것으로 볼 수 있다. 즉, 기초자산가격이 $\frac{xmax}{parsing} = \frac{300}{91}$ 이므로, 기초자산가격이 3.3% 상승하면, ELS가격이 약 1.2 증가할 것으로 예측할 수 있다.

ELS 발행기관은 주가가 떨어지면 주식을 사고, 주가가 오르면 주식을 팔며, 기초자산을 유지하되 운용수익을 챙길 수 있다.

2) 감마의 경우, 기초자산 가격 변화에 따른 델타의 변화량으로, 주식가격이 증가한다면, 델타가 감소할 수 있음을 의미한다. 즉, 기초자산가격이 3.3% 상승한다면, 델타가 -1.9 정도 감소할 수 있음을 보인다.

ELS 발행기관은 롱감마 포지션을 가지게 된다. 롱감마의 경우, 동적 델타 헷징을 통해 수익을 낼 수 있다.

3) 크로스 감마의 경우, 이 경우, 하나의 자산의 가격변화에 대한 다른 하나의 자산의 델타의 변화량을 의미한다. 만일 두 자산이 함께 움직이는 경우, 크로스 감마는 높아지게 된다.

4) 베가는 기초자산의 가격 변화에 따른 상품의 가격변화로, 기초자산가격이 3.3% 상승한다면, ELS가격 역시 1.06 정도 증가할 것으로 볼 수 있다.