Q3_main

2024-10-30

```
simulate_S_paths <- function(S0,r,q,sigma,n,m,delta_t) {
  S = rep(S0,n)
  S_df = data.frame(matrix(ncol = 0, nrow = n))
  for (i in 1:m) {

  delta_S =  S * (r-q) * delta_t + sigma * S * sqrt(delta_t) * rnorm(n)
  S = S + delta_S
  S_df[[paste0("S",i)]] = S
  }
  return(S_df)
}

laguerre_poly <- function(X, i) {
  # Ensure X is a vector
  if (!is.vector(X)) stop("X must be a vector")

  # Calculate Laguerre polynomial L_i(X)
  result <- 0
  for (n in 0:i) {
    result <- result + ((-1)^n * choose(i, n) * X^n / factorial(n))
  }

  return(result * exp(-X / 2))  # Apply exponential decay for the generalized form
}

LSM_put <- function(S0,K,r,q,sigma,t,n,m, k_regressors) {

  delta_t = t/m


  S_df = simulate_S_paths(S0,r,q,sigma,n,m,delta_t)


  exercise_times = rep(m,n)
  payoff_df = data.frame(apply(K - S_df, c(1, 2), function(x) max(x, 0)))

  payoff_scaled_df = payoff_df
  S_scaled_df = S_df/K


  for (i in (m-1):1) {
    itm_idx = payoff_df[,i] > 0
    future_cashflows = mapply(function(row, col) payoff_df[row, col], row = 1:nrow(payoff_scaled_df), c
    discount_times = delta_t * (exercise_times - i)
```

```
    Y = future_cashflows * exp(-r*discount_times[itm_idx])
    # X = S_df[,i][itm_idx]
    S_itm = S_scaled_df[,i][itm_idx]

    # Laguerre polynomial
    X = data.frame(matrix(ncol = 0, nrow = length(S_itm)))
    for (o_i in 0:(k_regressors-1)) {

      X[[paste0("L",o_i)]] = laguerre_poly(S_itm, o_i)
      # X[[paste0("X",o_i + 1)]] = S_itm**(o_i + 1)
    }

    model = lm(Y ~ ., data=X)
    #summary(model)
    cond_exp_Y = predict(model, newdata = data.frame(X))
    names(cond_exp_Y) = NULL
    current_itm_payoff = payoff_scaled_df[,i][itm_idx]
    exercise_times[itm_idx] = ifelse(current_itm_payoff > cond_exp_Y, i, exercise_times[itm_idx])
  }

  payoff_decisions = mapply(function(row, col) payoff_df[row, col], row = 1:nrow(payoff_df), col = exer
  discount_times = delta_t * (exercise_times - i)
  option_path_values = payoff_decisions * exp(-r*discount_times)
  option_value = mean(option_path_values)
  se = sd(option_path_values) / sqrt(n)

  early_exercise_portion = mean(exercise_times < m)
  return(list(value = option_value, se=se, early_portion=early_exercise_portion, ee_times=exercise_times

}
```

```
european_put_BS <- function(S,K,r,q,sigma,t) {

  d1 = (log(S / K) + (r - q + 0.5 * sigma^2) * t) / (sigma * sqrt(t))
  d2 = d1 - sigma * sqrt(t)

  put_price = K * exp(-r * t) * pnorm(-d2) - S0 * exp(-q * t) * pnorm(-d1)
  return(put_price)
}
```

```
S0 = K = 100
t = 1/12
r = 0.04
q = 0.02
sigma = 0.2

european_put_value = european_put_BS(S0,K,r,q,sigma,t)

m_list = c(10,20,30,40,50)
# m_list = c(10,20)
n_list = c(1000,1000*4, 1000*4**2, 1000*4**3, 1000*4**4)
# n_list = c(1000,1000*4)
k_regressors_list = c(1,2,3)
```

```
put_LSM_results = data.frame()
put_LSM_results2 = data.frame()

for (k_regressors in k_regressors_list) {
  temp_results = data.frame(matrix(ncol = 0, nrow = length(n_list)))
  temp_results2 = data.frame(matrix(ncol = 0, nrow = length(n_list)))

  for (m in m_list) {
    value_list = c()
    se_list = c()

    early_portion_list = c()
    early_value_list = c()
    for (n in n_list) {
      option_LSM_res = LSM_put(S0,K,r,q,sigma,t,n,m, k_regressors)
      value_list = c(value_list, option_LSM_res$value)
      se_list = c(se_list, option_LSM_res$se)

      early_value_list = c(early_value_list, option_LSM_res$value - european_put_value)
      early_portion_list = c(early_portion_list, option_LSM_res$early_portion)

    }
    temp_results[[paste0("value_m",m)]] = value_list
    temp_results[[paste0("se_m",m)]] = se_list

    temp_results2[[paste0("EE_value_m",m)]] = early_value_list
    temp_results2[[paste0("Pct_EE_m",m)]] = early_portion_list
  }

  n_names = c()
  for (n in n_list) {n_names = c(n_names, paste0("k",k_regressors,",","n",n))}
  rownames(temp_results) = n_names
  rownames(temp_results2) = n_names

  put_LSM_results = rbind(put_LSM_results, temp_results)
  put_LSM_results2 = rbind(put_LSM_results2, temp_results2)
}
```

	value_m10	se_m10	value_m20	se_m20	value_m30	se_m30	value_m40	se_m40	value_m50	se_m50
k1,n1000	2.244003	0.0926097	2.157604	0.0855481	2.227771	0.0878390	2.175247	0.0890546	2.202994	0.0938404
k1,n4000	2.220882	0.0452730	2.281288	0.0468543	2.215374	0.0452281	2.174692	0.0438203	2.217167	0.0451154
k1,n16000	2.223186	0.0230657	2.272631	0.0234660	2.195185	0.0217681	2.210109	0.0226266	2.204409	0.0217934
k1,n64000	2.207165	0.0115226	2.207865	0.0111969	2.218574	0.0112222	2.210261	0.0111262	2.219233	0.0111191
k1,n256000	2.216780	0.0057656	2.218247	0.0056639	2.199092	0.0055350	2.212745	0.0055493	2.207404	0.0054995
k2,n1000	2.251306	0.0950062	2.394823	0.0909027	2.025101	0.0782879	2.319820	0.0968155	2.396936	0.0922530
k2,n4000	2.276385	0.0474774	2.210542	0.0451888	2.235069	0.0448596	2.282570	0.0453790	2.319925	0.0449778
k2,n16000	2.224510	0.0224584	2.216977	0.0217304	2.222601	0.0214077	2.228284	0.0215469	2.207382	0.0216281
k2,n64000	2.226092	0.0112861	2.227294	0.0111539	2.199531	0.0108588	2.218476	0.0109201	2.233093	0.0109340
k2,n256000	2.229335	0.0056330	2.221866	0.0055147	2.206458	0.0054137	2.217589	0.0053988	2.226828	0.0053975
k3,n1000	2.067823	0.0875236	2.143367	0.0927040	2.491521	0.0960144	2.208239	0.0958243	2.365252	0.0929499
k3,n4000	2.203729	0.0455855	2.233370	0.0455633	2.236384	0.0452773	2.239018	0.0436957	2.251022	0.0457607
k3,n16000	2.197055	0.0220978	2.189398	0.0216855	2.216994	0.0216771	2.214891	0.0233936	2.241734	0.0220874
k3,n64000	2.226053	0.0113112	2.219996	0.0109498	2.224232	0.0111237	2.229365	0.0110330	2.230979	0.0109004
k3,n256000	2.222390	0.0057045	2.219505	0.0056063	2.223838	0.0054811	2.224665	0.0055591	2.223447	0.0054886

	EE_value_m10	Pct_EE_m10	EE_value_m20	Pct_EE_m20	EE_value_m30	Pct_EE_m30	EE_value_m40	Pct_EE_m40	EE_value_m50	Pct_EE_m50
k1,n1000	0.0289464	0.1780000	-0.0574530	0.2860000	0.0127142	0.1930000	-0.0398098	0.4420000	-0.0120621	0.1980000
k1,n4000	0.0058253	0.1810000	0.0662314	0.2627500	0.0003179	0.2692500	-0.0403641	0.3097500	0.0021109	0.3567500
k1,n16000	0.0081289	0.1810000	0.0575747	0.1993750	-0.0198713	0.2365625	-0.0049472	0.1705000	-0.0106472	0.2651875
k1,n64000	-0.0078916	0.1747656	-0.0071914	0.2206094	0.0035170	0.2445625	-0.0047956	0.2693125	0.0041760	0.2723906
k1,n256000	0.0017237	0.1777461	0.0031905	0.2112695	-0.0159647	0.2522031	-0.0023115	0.2557578	-0.0076521	0.2841367
k2,n1000	0.0362490	0.1820000	0.1797663	0.3870000	-0.1899560	0.3420000	0.1047637	0.3670000	0.1818797	0.2870000
k2,n4000	0.0613287	0.3082500	-0.0045151	0.2440000	0.0200125	0.2527500	0.0675139	0.3295000	0.1048683	0.3057500
k2,n16000	0.0094537	0.2803125	0.0019207	0.3011875	0.0075445	0.2922500	0.0132277	0.2896250	-0.0076743	0.3165000
k2,n64000	0.0110356	0.2388438	0.0122377	0.2769375	-0.0155255	0.2899844	0.0034190	0.2899844	0.0180367	0.2896406
k2,n256000	0.0142786	0.2425312	0.0068098	0.2725703	-0.0085987	0.2861758	0.0025326	0.3003945	0.0117717	0.2983828
k3,n1000	-0.1472336	0.2390000	-0.0716894	0.3720000	0.2764646	0.4750000	-0.0068174	0.3670000	0.1501958	0.3070000
k3,n4000	-0.0113278	0.3267500	0.0183130	0.2995000	0.0213275	0.3582500	0.0239615	0.3652500	0.0359651	0.3607500
k3,n16000	-0.0180016	0.3015625	-0.0256587	0.3216250	0.0019378	0.3398750	-0.0001653	0.3035000	0.0266775	0.3435000
k3,n64000	0.0109963	0.2908281	0.0049393	0.3179531	0.0091749	0.3382031	0.0143084	0.3277188	0.0159223	0.3468594
k3,n256000	0.0073338	0.2745156	0.0044484	0.3168555	0.0087814	0.3308203	0.0096085	0.3388047	0.0083899	0.3471094