University of California, Santa Barbara

MATH 104A

FINAL PROJECT

An Application of Numerical Analysis to Portfolio Theory

Authors:
Emily Lu
Li-Ting Chu
Mana Sanavi
Vista Darabian

Supervisor:
Professor Paul Atzberger

December 6th, 2019

Abstract

The objective of this project is to demonstrate the application of Numerical Analysis in Portfolio Theory by building a desirable portfolio made up of three stocks: Boeing, McDonald's, and Starbucks through simultaneous risk minimization and profit maximizations, based on the initial amount of risk we, as investors, are willing to incur and our desired payoff. To do so, we used the Markowitz Theory to construct a mathematical model of our portfolio, the Steepest Descent method to approximate the optimal weights for the aforementioned model, and the method of Lagrange Multipliers to validate our approximations. We found out that our numerical analysis approach through the Steepest Descent method produced nearly similar results as the results found through the Lagrangian Method.

Contents

1 Introduction **1**

2 Characterizing the Rate of Returns and Riskiness of Assets/Portfolios **1**

3 Portfolio Optimization **3**

4 Method of Steepest Descent for Approximation of Optimization Problem **4**

5 Method of Lagrange Multipliers **5**

6 Comparison of Steepest Descent Method and Lagrange Multipliers Method **6**

7 Concluding Remarks **7**

Appendix: Additional R Code **8**

Reference **12**

1 Introduction

In the world of investing, the most desirable portfolio is one that maximizes profit with minimal risk. However, this perfect portfolio often ceases to exists as risk and reward often do not go in hand with each other. Thus, this brings us to an old set of principles known as the portfolio theory. Starting with the assumption of investors being risk averse, portfolio theory works on the idea that investors could maximize returns and minimize risk by identifying which asset combinations to invest in. To exercize this theory and as an objective of this report, we will perform a numerical analysis towards the construction of a desirable portfolio given the initial wealth to be invested in the following three assets: Boeing (BA), McDonald's (MCD), and Starbucks (SBUX). In particular, we will use the Markowitz Theory to construct a mathematical model of our portfolio, the Steepest Descent method to approximate the optimal weights for the aforementioned model, and the method of Lagrange Multipliers to validate our approximations. The data used to model our analyses and results will be the monthly adjusted closing prices from November 30^{th}, 1999 to November 30^{th}, 2019 of Boeing, McDonald's, and Starbucks.

2 Characterizing the Rate of Returns and Riskiness of Assets/Portfolios

To start off, we shall define the two primary characteristics of assets/portfolios. The first common characterization is through the average rate of return of an asset/portfolio,

$$\mu = \mathbb{E}(\rho),$$

over a period of time. Using the average rate of return, we could capture the "expected" rate of return; however, it does not indicate how large the "expected" return rate may deviate from the actual return rate. Therefore to capture the variances/riskiness of our expected rate of returns, we use the second primary characteristic of assets/portfolios denoted as

$$\sigma^2 = \mathrm{Var}(\rho) = E(|\rho - \mu|^2).$$

Although we use σ_i^2 to capture the variance of returns of the i^{th} individual asset, it is also important to capture how the returns may be coupled among the assets. To do so, we calculate the covariance of the returns which is given by

$$\sigma_{i,j} = \mathbb{E}[(\rho_i - \mu_i)(\rho_j - \mu_j)].$$

Note that for $i = j$, $\sigma_{i,i} = \sigma_i^2$ and $\sigma_{i,j} = \sigma_{j,i}$.

Given the general description above, we shall now define the return and risk characteristics specific to our portfolio of three assets. First, we will denote $\sum_{i=1}^{3} W_i = W$ as the total initial wealth invested in our portfolio, $\omega_i = \frac{W_i}{W}$ as fractional share of the total wealth, and $S_i(t)$ to be the index value at time t for the $i^{th} \in \{1, 2, 3\}$ asset. We could then express the value of the portfolio at time t to be $S_p(t) = \sum_{i=1}^{3} \frac{W_i}{S_i(0)} S_i(t)$ and $S_p(0) = W$ by definition. Using the rate of change formula, we could derive the rate of return for our portfolio at time t to be

$$
\begin{aligned}
R_p(t) &= \frac{S_p(t) - S_p(0)}{S_p(0)} && (1)\\
&= \frac{\sum_{i=1}^{3} \frac{W_i}{S_i(0)} S_i(t) - W}{W} && (2)\\
&= \sum_{i=1}^{3} \frac{W_i}{W}\frac{S_i(t)}{S_i(0)} - \sum_{i=1}^{3} \frac{W_i}{W} && (3)\\
&= \sum_{i=1}^{3} \omega_i \frac{S_i(t) - S_i(0)}{S_i(0)} && (4)\\
&= \sum_{i=1}^{3} \omega_i R_i(t), && (5)
\end{aligned}
$$

which is just the weighted average of rates of return of the assets in our portfolio.

The expected monthly rate of return of our portfolio at time t is

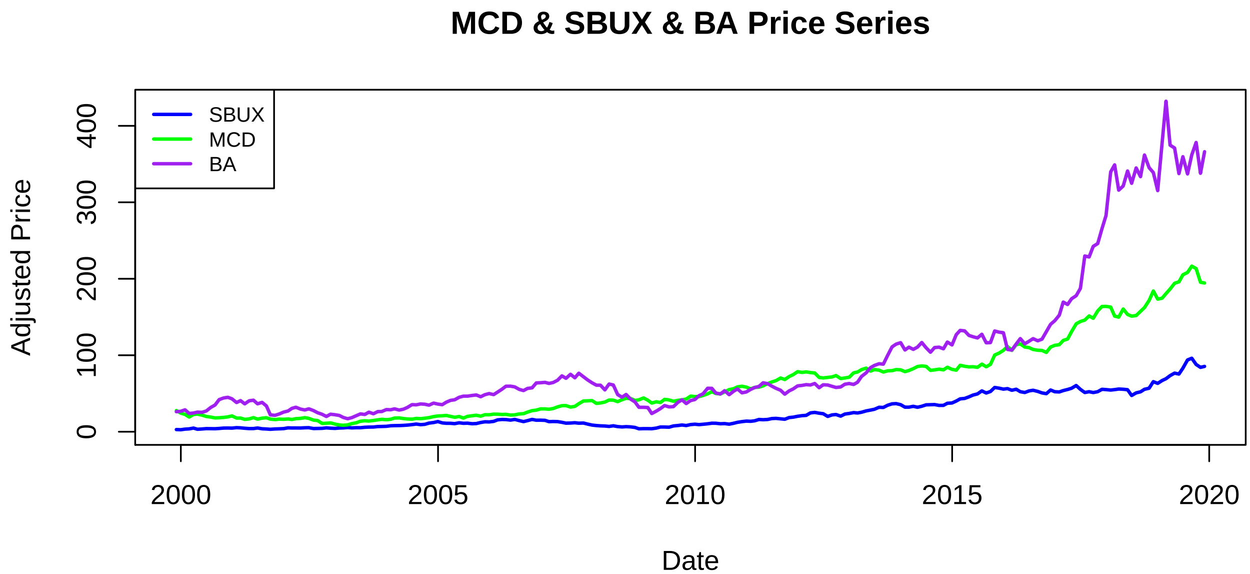
$$
\begin{aligned}
\mu_p(t) &= \mathbb{E}[R_p(t)] && (6)\\
&= \mathbb{E}\Big[\sum_{i=1}^{3} \omega_i R_i(t)\Big] && (7)\\
&= \sum_{i=1}^{3} \omega_i \mu_i. && (8)
\end{aligned}
$$

The variance of our portfolio return at time t is

$$
\begin{aligned}
\sigma_p^2(t) &= \mathbb{E}\Big[\Big|R_p - \mu_p\Big|^2\Big] && (9)\\
&= \mathbb{E}\Big[\Big|\sum_{i=1}^{3} \omega_i R_i(t) - \sum_{i=1}^{3} \omega_i \mu_i\Big|^2\Big] && (10)\\
&= \mathbb{E}\Big[\Big(\sum_{i=1}^{3} \omega_i (R_i(t) - \mu_i)\Big)\Big(\sum_{j=1}^{3} \omega_j (R_j(t) - \mu_j)\Big)\Big] && (11)\\
&= \sum_{i,j=1}^{3} \omega_i \omega_j \mathbb{E}[(R_i(t) - \mu_i)(R_j(t) - \mu_j)] && (12)\\
&= \sum_{i,j=1}^{3} \omega_i \omega_j \sigma_{i,j} && (13)\\
&= (\omega_1\ \omega_2\ \omega_3) \begin{pmatrix} \sigma_1^2 & \sigma_{1,2} & \sigma_{1,3} \\ \sigma_{1,2} & \sigma_2^2 & \sigma_{2,3} \\ \sigma_{1,3} & \sigma_{2,3} & \sigma_3^2 \end{pmatrix} \begin{pmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{pmatrix} && (14)\\
&= \omega^T V \omega && (15)
\end{aligned}
$$

where $V = \begin{pmatrix} \sigma_1^2 & \sigma_{1,2} & \sigma_{1,3} \\ \sigma_{1,2} & \sigma_2^2 & \sigma_{2,3} \\ \sigma_{1,3} & \sigma_{2,3} & \sigma_3^2 \end{pmatrix}$ is the covariance matrix.

To demonstrate these derivations, we graphed below the monthly rate of returns of each asset in our portfolio from November 30^{th}, 1999 to November 30^{th}, 2019.



Afterwards, we then calculated and found the expected rates of monthly return, covariances, and variances of each asset in our portfolio to be:

Stock	Mean	Variance	Pair i,j	Covariance i,j
SBUX	1.81%	0.75%	(SBUX,MCD)	0.06%
MCD	0.98%	0.32%	(SBUX,BA)	0.15%
BA	1.44%	0.66%	(MCD,BA)	0.14%

3 Portfolio Optimization

By Markowitz Theory, a desirable portfolio is one where if for an expected of return, u_p, the portfolio has the least variance, σ_p^2. Finding such a portfolio is known as the Markowitz problem and it could be mathematically modelled as a contrained optimization problem:

$$\text{minimize } f(\omega) = \frac{1}{2}\sum_{i,j=1}^{n} \omega_i \omega_j \sigma_{i,j}$$

$$\text{subject to } g_1(\omega) = \sum_{i=1}^{n} \omega_i \mu_i = 0 \;\;\&\;\; g_2(\omega) = \sum_{i=1}^{n} \omega_i - 1 = 0.$$

There are several numerical methods to solve for this contrained optimization problem, but for our case, we will use the Method of Steepest Descent to find the numerical approximations and the Lagrangian Method to find the true numerical values. Note that the optimal portfolio we will be finding for both methods refers to the portfolio that has the minimum variance out of all portfolios.

We will also be finding efficient frontier portfolios which are convex combination of any two minimum variance portfolios with different target returns and have greater expected returns than the expected returns of the minimum variance portfolios. Unlike the global minimum variance portfolio, frontier portfolio minimizes risk based on a target return. To solve for frontier portfolios in our case, let x and y be any two minimum variance portfolios with different targeted expected returns: $x'\mu = \mu_{p,0} \neq y'\mu = \mu_{p,1}$. Portfolio x solves

$$\min \sigma_{p,x}^2 = x'\Sigma x \text{ s.t. } x'\mu = u_{p,0} \; and \; x'1 = 1$$

and portfolio y solves

$$\min \sigma_{p,y}^2 = y'\Sigma y \text{ s.t. } y'\mu = u_{p,1} \; and \; y'1 = 1.$$

Then let α be any constant and define z to the linear combination of portfolios x and y, which is the frontier portfolio.

4 Method of Steepest Descent for Approximation of Optimization Problem

The method of Steepest Descent is used to solve systems of nonlinear equations given an initial approximation of solutions. It determines the local minimum of a multivariable function g of the form: $\mathbb{R}^n \to \mathbb{R}$. Unlike other numerical methods, the Steepest Descent method has the advantage of usually converging even for poor initial approximations. This works perfectly in our favor since we do know what our initial approximations of the weights for our optimized portfolio should be. To start off, we will list the methods of Steepest Descent for finding a local minimum of an arbitrary function g from $\mathbb{R}^n \to \mathbb{R}$:

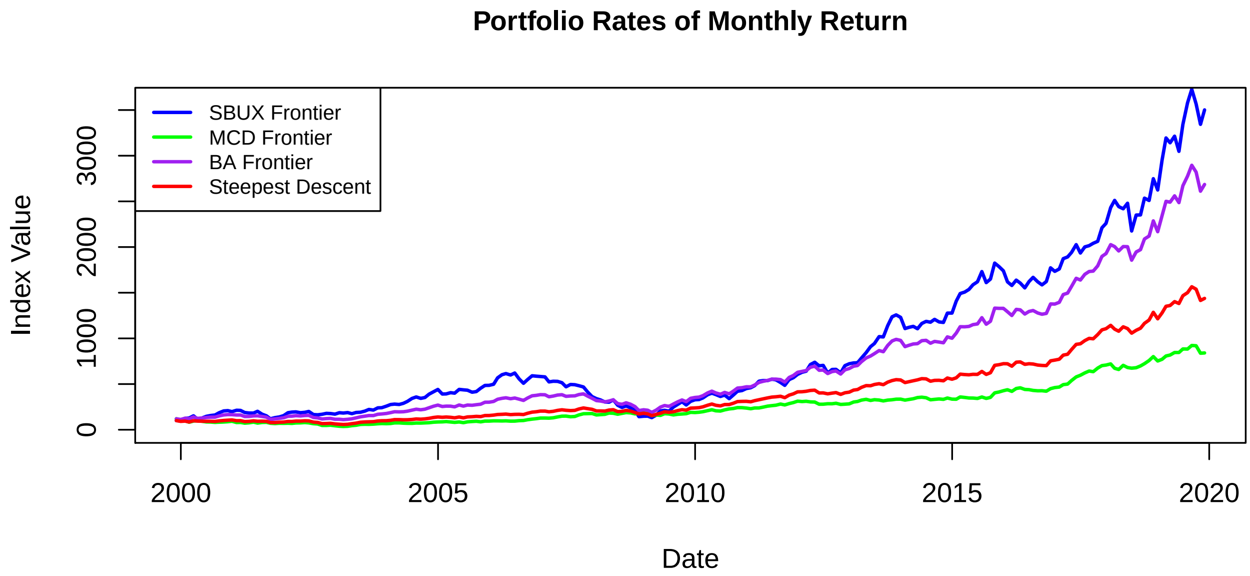
1. Evaluate g with an initial approximation $x^{(0)} = (x_1^{(0)}, ..., x_n^{(0)})^t$.
2. Determine a direction from $x^{(0)}$ that will decrease the value of g, which is the negative gradient of g at $x^{(0)}$ i.e. $-\nabla g(x^{(0)})$.
3. Move an appropriate amount towards this direction and call the new value $x^{(1)}$. An appropriate amount is determined by solving for the value α that minimize the function: $h(\alpha) = g(x^{(0)} - \alpha \nabla g(x^{(0)}))$.
4. Repeat steps 1 through 3 with $x^{(0)}$ replaced by $x^{(1)}$.

Through the method of Steepest Descent, we calculated and found the following statistics:

	SBUX	MCD	BA	Variance	Avg. Monthly R.o.R.
Steepest Descent	20.01%	60.72%	19.27%	0.23%	1.23%

Using the weights derived from the Steepest Descent Method, we graphed the rates of return from November 30^{th}, 1999 to November 30^{th}, 2019 below using one hundred dollars as the initial wealth invested in our portfolio and compared them against frontier portfolios with the same expected return as each asset in our portfolio.

Note: "SBUX Frontier" refers to a frontier portfolio with same expected return as Starbucks and so on.



5 Method of Lagrange Multipliers

In the method of the Lagrange Multipliers, the constraints are taken into account by solving the unconstrained optimization problem:

$$L(w, \lambda_1, \lambda_2) = \frac{1}{2}\omega^T V\omega + \lambda_1(\mu_p - \omega^T \mu) + \lambda_2(1 - \omega^T 1).$$

To solve for the minima and maxima, we need to first assume that expected monthly return rates on the three assets are linearly independent and not equal to each other. Afterwards, we take the partial derivatives with respect to each variable in L to obtain a set of equations known as the first order conditions and set them equal to 0. The set of equations we obtained are:

$$\nabla_w L = V\omega_p - \lambda_1\mu - \lambda_2 1 = 0$$

and

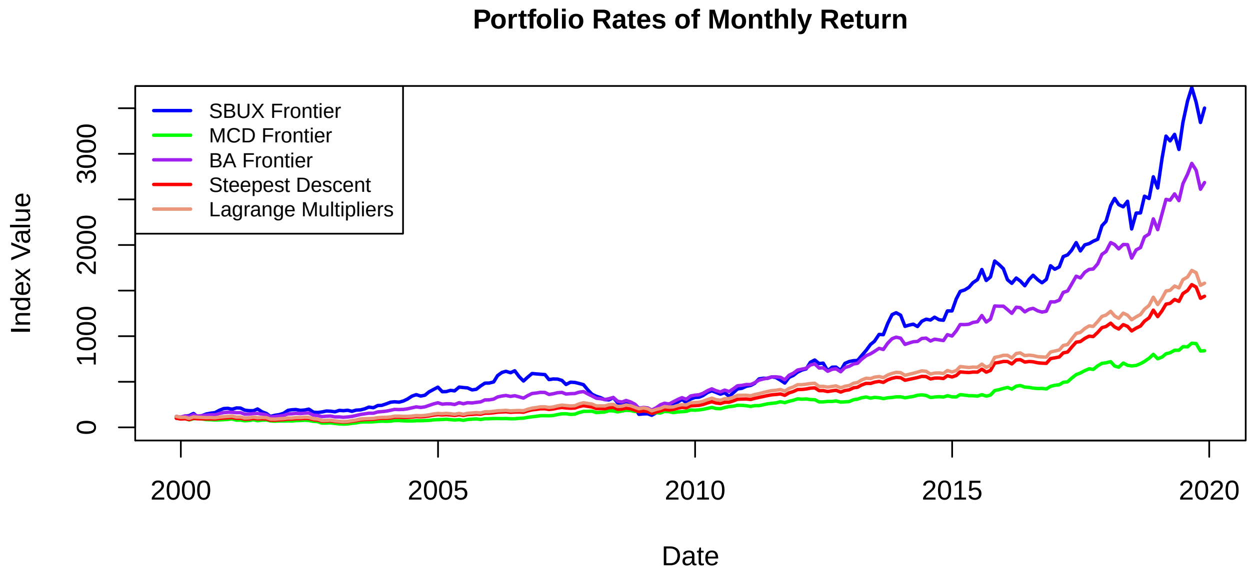
$$\frac{\partial L}{\partial \lambda_1} = \mu_p - w_p^T \mu = 0$$
$$\frac{\partial L}{\partial \lambda_2} = 1 - w_p^T 1 = 0.$$

Solving for the variables will give us the critical points of our optimization problem f.

Through the method of Lagrange multipliers, we calculated and found the following statistics:

	SBUX	MCD	BA	Variance	Avg. Monthly R.o.R.
Lagrange Multipliers	15.75%	64.90%	19.34%	0.23%	1.20%

Using the weights derived from the Lagrangian Multipliers Method, we graphed the rates of monthly return from November 30^{th}, 1999 to November 30^{th}, 2019 below with one hundred dollars as the initial wealth invested along with previous results.



6 Comparison of Steepest Descent Method and Lagrange Multipliers Method

	SBUX	MCD	BA	Variance	Avg. Monthly R.o.R.
Steepest Descent	20.01%	60.72%	19.27%	0.23%	1.23%
Lagrange Multipliers	15.75%	64.90%	19.34%	0.23%	1.20%
SBUX Frontier	76.61%	-18.10%	41.49%	0.37%	1.81%
MCD Frontier	-6.37%	95.07%	11.29%	0.18%	0.98%
BA Frontier	40.41%	31.27%	28.32%	0.28%	1.44%

Taking the values acquired from the Steepest Descent method as our approximated values and the values acquired from the Lagrange Multipler method as our true values, we performed an error analysis between the values. For the expected rate of monthly return, we found the absolute error to be 3.50x10^{-4} and the relative error to be 2.92x10^{-2}. As for the variance of monthly returns, we found the absolute error to be 7.34x10^{-5} and the relative error to be 3.24x10^{-2}. For our purpose, we focused only on the relative errors since it takes into account consideration the size of the value. The relative error for the expected rate of monthly return is slighter smaller than the relative error for variance of monthly returns, which is could be perhaps contributed to the varied distributions of weights found by the Steepest Descent method and Lagrange Multiplier method. This tells us that a slight increase in share(s) of one or more of the assets would lead to different variances and expected rates of return. Additionally, we concluded that the expected return and variance of expected return share a direct correlation. As we could see in the table above, the portfolio with largest variance also has the largest average rate of return and vice versa.

7 Concluding Remarks

In this project, we have shown two methods of calculating the optimal weights or shares to be invested in each asset of a global minimum variance portfolio. Although we used real data to calculate the optimal weights and validate through two different methods, we realized that this theory is actually difficult to perform in real practice. In real practices, investors ideally would like a portfolio that is optimized against future risks. However, it is nearly impossible to predict the exact future. Thus, our project was limited in that we only focused on the past results and didn't model any predicted results.

Appendix: Additional R Code

```r
library(dplyr)
library(BatchGetSymbols)
library(knitr)
library(formattable)
library(numDeriv)

getMonthly <- function(ticker, first.date, last.date){

  stock <- get.clean.data(ticker, src="yahoo", first.date, last.date)
  stock$mon_yr = format(stock$ref.date, "%Y-%m")
  final <- stock %>% group_by(mon_yr) %>% filter(ref.date == max(ref.date))
  final <- final[, (names(final) %in% c('ref.date', 'price.adjusted'))]
  final$Returns <- calc.ret(final$price.adjusted, type.return='arit')

  return(final)}

# Data Retrieval
first.date <- "1999-11-30"; last.date <- "2019-11-30"
SBUX <- getMonthly('SBUX', first.date, last.date)
MCD <- getMonthly('MCD', first.date, last.date)
BA <- getMonthly('BA', first.date, last.date)

# Mean, Variance, Covariance Calculations
asset.names <- c("SBUX", "MCD", "BA")
n <- nrow(SBUX)
mu.vec <- c(percent(mean(SBUX$Returns[2:n])),
            percent(mean(MCD$Returns[2:n])),
            percent(mean(BA$Returns[2:n])))
var_vec <- c(percent(var(SBUX$Returns[2:n])),
             percent(var(MCD$Returns[2:n])),
             percent(var(BA$Returns[2:n])))
cov_vec <- c(percent(cov(SBUX$Returns[2:n], MCD$Returns[2:n])),
             percent(cov(SBUX$Returns[2:n], BA$Returns[2:n])),
             percent(cov(MCD$Returns[2:n], BA$Returns[2:n])))

asset_info <- data.frame("Stock"=c("SBUX","MSFT","BA"),"Mean"=mu.vec,
                         "Variance"=var_vec, "Pair i,j" =
                           c("(SBUX,MCD)","(SBUX,BA)","(MCD,BA)") ,
                         "Covariance i,j"=cov_vec)

# Defining the initial guessed weights and sigma matrices
w <- rep(1/3, 3)
sigma.mat <- matrix(c(var_vec[1], cov_vec[1], cov_vec[2],
                      cov_vec[2], var_vec[2], cov_vec[3],
                      cov_vec[3], cov_vec[3], var_vec[3]),
```

```
                nrow = 3, ncol = 3)
dimnames(sigma.mat) <- list(asset.names, asset.names)

# Portfolio Returns
R <- matrix(SBUX$Returns, nrow = 1); R <- rbind(R, MCD$Returns)
R <- rbind(R, BA$Returns)
portf_ret <- t(w) %*% R

# Expected Return of the Portfolio given initial weights
portf_mu <- percent(t(w) %*% mu.vec)

# Expected Variance of the Portfolio given initial weights
portf_var <- percent(t(w) %*% sigma.mat %*% w)

# Optimization ftn to minimize with large lambda1 and lambda2 as default
E <- function(a, lambda1=100, lambda2=100){

  one_matrix <- c(1, 1, 1)
  Q1 <- lambda1*(portf_mu-t(a)%*%mu.vec)
  Q2 <- lambda2*(1-t(a)%*%one_matrix)
  Ex <- .5*t(a)%*%sigma.mat%*%a

  return(Ex+Q1+Q2)}

# Method of Steepest Descent function
steepestDescent <- function(x0, TOL = 10^(-10), N = 100, gx = E){

eps <- 10^(-10)
k <- 1
while(k <= N){

  g1 <- gx(x0)

  ## grad() calculates the gradient using the central difference
  ## formula, introduing some error
  z <- grad(gx, x0); z1 <- sqrt(sum(z^2))
  if (z1 == 0) {
    warning(
    paste("Zero gradient at:", x0, g1, "-- not applicable.\n"))
    return(list(xmin = NA, fmin = NA, niter = k))}
  z <- z/z1; a1 <- 0; a3 <- 1; g3 <- gx(x0 - a3*z)

  while (g3 >= g1){
    a3 = a3/2; g3 = gx(x0 - a3*z)
    if (a3 < TOL/2){
      warning(
      paste("No likely improvement.\n"))
```

```r
      x0[x0 < eps] <- 0
      return(list(xmin = x0, fmin = gx(x0), niter = k))}}
  a2 <- a3/2; g2 <- gx(x0 - a2*z)
  h1 <- (g2 - g1)/a2; h2 <- (g3 - g2)/(a3 - a2); h3 <- (h2 - h1)/a3
  a0 <- 0.5*(a2 - h1/h3); g0 <- gx(x0 - a0*z)

  if (g0 < g3)
    a <- a0
  else a <- a3
    x0 <- x0 - a*z
  k <- k+1}
if (k > N)
warning("Maximum number of iterations reached -- not converged.\n")
return(list(xmin = NA, fmin = NA, niter = k))}

# Minimize the function and the optimal weights involved
minimized <- steepestDescent(x0=w)$xmin

sum_weights <- sum(as.vector(minimized))
opt_weights <- c(as.vector(minimized)[1]/sum_weights,
                 as.vector(minimized)[2]/sum_weights,
                 as.vector(minimized)[3]/sum_weights)

# Weights for minimum variance portfolio
top.mat <- cbind(2*sigma.mat, rep(1, 3)); bott.vec <- c(rep(1, 3), 0)
TB.mat <- rbind(top.mat, bott.vec); b.vec <- c(rep(0, 3), 1)
z.mat <- solve(TB.mat)%*% b.vec; w.vec <- z.mat[1:3,1]

# Weights for frontier portfolio: Starbucks
x.vec = rep(1,3)/3
names(x.vec) = asset.names
mu.p.x = crossprod(x.vec,mu.vec)
sig2.p.x = t(x.vec) %*%
      sigma.mat %*%
      x.vec
sig.p.x = sqrt(sig2.p.x)

top.mat = cbind(2*sigma.mat, mu.vec, rep(1, 3))
mid.vec = c(mu.vec, 0, 0)
bott.vec = c(rep(1, 3), 0, 0)
A.mat = rbind(top.mat, mid.vec, bott.vec)
bmsft.vec = c(rep(0, 3), mu.vec[1], 1)

z.mat = solve(A.mat) %*% bmsft.vec
x.vec = z.mat[1:3,]

# Weights for frontier portfolio: McDonalds
```

```r
x.vec = rep(1,3)/3
names(x.vec) = asset.names
mu.p.x = crossprod(x.vec,mu.vec)
sig2.p.x = t(x.vec) %*%
      sigma.mat %*%
      x.vec
sig.p.x = sqrt(sig2.p.x)

top.mat = cbind(2*sigma.mat, mu.vec, rep(1, 3))
mid.vec = c(mu.vec, 0, 0)
bott.vec = c(rep(1, 3), 0, 0)
A.mat = rbind(top.mat, mid.vec, bott.vec)
bmsft.vec = c(rep(0, 3), mu.vec[2], 1)

z.mat = solve(A.mat) %*% bmsft.vec
x.vec = z.mat[1:3,]

# Weights for frontier portfolio: Boeing
x.vec = rep(1,3)/3
names(x.vec) = asset.names
mu.p.x = crossprod(x.vec,mu.vec)
sig2.p.x = t(x.vec) %*%
      sigma.mat %*%
      x.vec
sig.p.x = sqrt(sig2.p.x)

top.mat = cbind(2*sigma.mat, mu.vec, rep(1, 3))
mid.vec = c(mu.vec, 0, 0)
bott.vec = c(rep(1, 3), 0, 0)
A.mat = rbind(top.mat, mid.vec, bott.vec)
bmsft.vec = c(rep(0, 3), mu.vec[3], 1)

z.mat = solve(A.mat) %*% bmsft.vec
x.vec = z.mat[1:3,]
```

Reference

Numerical Analysis 10th Ed. by R. Burden, J. D. Faires, A. M. Burden

An Introduction to Portfolio Theory by P. Atzberger