# PORTFOLIO OPTIMIZATION THROUGH QUADRATIC PROGRAMMING WHEN THERE IS PERTURBATION IN THE RETURN MATRIX

N.T.Dharmathilaka

(148903X)

Degree of Master of Science

Department of Mathematics

University of Moratuwa
Sri Lanka

March 2017

# PORTFOLIO OPTIMIZATION THROUGH QUADRATIC PROGRAMMING WHEN THERE IS PERTURBATION IN THE RETURN MATRIX

N.T.Dharmathilaka

(148903X)

Thesis/Dissertation submitted in partial fulfillment of the requirements for the degree

Master of Science

Department of Mathematics

University of Moratuwa
Sri Lanka

March 2017

## DECLARATION OF THE CANDIDATE

I declare that this is my own work and this thesis/dissertation does not incorporate without acknowledgement any material previously submitted for a Degree or Diploma in any University or other institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

............................ .....................

**N.T.Dharmathilaka** **Date**

## DECLARATION OF THE SUPERVISOR

We have supervised and accepted this thesis for the submission of the degree.

.................................... ..........................

**Mr. A. R. Dissanayake** **Date**

Senior Lecturer,

Department of Mathematics,

University of Moratuwa.

..................................... ...........................

**Mr. L. P. Ranasinghe** **Date**

PhD,

Department of Mathematics,

University of Colombo.

## ACKNOWLEDGEMENT

I would like to convey my sincere gratitude to my parents, people who guided me to choose this Financial Mathematics MSc degree and the lecturers who guided through -out the whole two years, specially Mr.T.M.J.A. Cooray, our course coordinator and Mr.A.R. Dissanayake who was my research supervisor. Without them this attempt would not be possible and would not be worthy. Last but not least I would like to thank Mr.L.P. Ranasinghe who helped me and guided me in development of research structure and provided me with research raw data.

N.T. Dharmathilaka

# ABSTRACT

According to Finance the investor who invests in risky assets such as stocks, after forming a diversified portfolio or collection of securities; is interested in earning maximum return out of minimum risk, and it is more technically known as Portfolio Optimization(PO). The present problem is a Quadratic Programming problem is consisted of simultaneous variations of initial return vector of each company.
In this study the main objective was to study the behavior of covariance-variance matrix, correlation matrix and the optimum weights vector when there is small perturbation in the mean return vector. Then fitting a model between perturbation values versus optimum weights was also performed. The results show that there is a significant variation of optimum weights when there is small perturbation in the return matrix. And there is no change in the covariance or correlation matrices. This is done under the assumption that there is no short selling. Apart from that results show that when there is perturbation in the return matrix the expected return of the portfolio is also changing.
When the value of perturbation is increased individually for one company only, to drive away at least one company from the optimum weights (to zero the optimum weight of one company) it was observe that the perturbation value should be increased extensively for the SGX data sample. That means the weights are not very much sensitive to perturbations in the market.
If negatively mean companies are removed and perturbation is done for positively mean companies the effort to remove at least one company from the optimum weights is less.
**Keywords**: Portfolio Optimization, Quadratic Programming, Optimum weights, Expected return , Matlab

# CONTENTS

# LIST OF TABLES

# LIST OF FIGURES

# LIST OF ABBREVIATIONS

| | |
|---|---|
| CSV | Comma delimited |
| Matlab | Matrix Laboratory |
| MPT | Modern Portfolio Theory |
| MSE | Mean Squared Error |
| MV | Mean Variance |
| PO | Portfolio Optimization |
| QP | Quadratic Programming |
| SGX | Singapore Stock Exchange |
| 3D | Three Dimensional |

# Chapter 1

# INTRODUCTION

## 1.1 Background

In Finance the different types of risks are faced by an investor can be categorized in to two main groups, namely systematic risk and unsystematic risk. Systematic risk is macro in nature and uncontrollable from the organization's point of view and influenced by external factors such as government policies, interest rates, inflation etc. From the other hand unsystematic risk which is micro in nature and can be controllable within the organization itself. There are theories mainly focused on reducing unsystematic risk. According to Finance the investor who invests in risky securities; such as stocks, always carries the burden of losing his money due to poor performance of his security. One can really look at the company's performance which is measured by the return of the company. The risk is measured by the variance or simply the standard deviation of return. The common studies show that it is recommended to invest in group of securities rather than one security to reduce risk and maximize profit. This is also known as "Diversification". Financial professionals form a "Portfolio" which is the grouping of several securities such as stocks.[1]

Performing the task of minimizing the risk and maximizing the return at the same time under some constraints is called the PO (Portfolio Optimization). The method developed by Harry Markowitz or Mean-variance optimization are other names for it. The basic equation for the optimizing the portfolio is a Quadratic Programming Problem. Quadratic programming refers to the problem of minimizing a quadratic function subject to linear equality and inequality constraints. The present study is based on this idea. The known factor is that the optimization function is quadratic so the 'quadprog' inbuilt function in Matlab(Matrix Laboratory) can be used to solve the optimization problem. The current research flows through the idea of analysing what happens when the per-

turbation occur in the initial individual return matrix. When small variation is done to the return matrix how the optimal weights vectors, covariance matrices and correlation matrices behave is going to be analysed critically. A model is fitted to study how optimum portfolio varies with the perturbation value.
In 1991 Best and Grauer investigated the sensitivity of mean variance efficient portfolios to changes in the means of the individual assets. Another related research is in 2004 Hadigheh, Romanko and Terlaky had studied the behaviour of Convex Quadratic optimization problem when variation occurs simultaneously in the right hand side vector of the constraints and in the coefficient vector of the linear term in the objective function. The purpose of this research is to investigate about the behaviour of optimum portfolio when there is very small perturbation in the return matrix.

## 1.2 Problem definition

As the problem definition it can be expressed that how the covariance and correlation coefficients behave when there is very small random perturbation in the initial return matrix. For example if one company's mean return increased by 0.01percent and another company's mean return decreased by 0.1 percent randomly how it affects to the optimum portfolio and covariances are studied. A model is fitted to study how optimum portfolio varies with the perturbation value. The minimization problem is a Quadratic Optimization problem, so the inbuilt function 'quadprog' is used to solve the problem using the Matlab m file.

## 1.3 Objectives

The main objective of this research is to investigate how optimum weights of the portfolio behave when there is small variations in the mean vector of the ten selected companies.This was done using a Matlab code. Except that how correlation and covariance matrices are behaving, how optimum weights vs perturbation values are inter related is investigated by developing a neural network model. Also graphical representation of Efficient frontier and colormap of perturbation values vs optimum weights was done.

## 1.4 Significance of study

The most significant aspect of this study is using refined optimum portfolio weights versus difference of the return value a model can be fitted, and using this fitted model it can be performed a forecasting of optimum portfolio. It is easier than performing a traditional way of calculating a optimum portfolio.
The significance of study is it can be studied the behavior of optimum weights and correlations when small perturbation is applied to the mean of the each company. This is similar to the real random situation so results obtained can be applied to the real investing scenario. The significance of study is that the by taking the investment or budget as a unitary value, it can be calculated the optimum allocation of each company and to obtain the each money allocation only involves multiplying the result by budgeted money value. Another significance of study is number of investment channels are independent of the final results. Any number of securities can be added or removed. Solely model depends upon the historical data of daily, weekly or monthly prices. Historical data can be easily obtained in the web sites for low cost. Model is simple and easy to calculate. Simple mathematics involved.

## 1.5 Data collection

The daily stock prices of ten companies of SGX(Singapore Stock Exchange) from 4/4/2013 to 4/4/2014 is going to be used as raw data. These are quantitative and secondary data.

## 1.6 Content of the research

This research consists of five main chapters. They are Introduction, Literature review, Methodology, Data analysis and results and Conclusion and recommendations. Basically the Introduction is divided in to sub sections such as Background, Problem definition, Objectives, Significance of study, Data collection and Content of research. Literature review is the chapter where the extracted knowledge from various reading materials are documented. Used theories and important facts regarding the current research are thoroughly described here. The used methodology and models are explained in the third chapter; Methodology. All the results obtained by the Matlab m-file 'PortfolioOpt' are stated and the analysis of data is done in the fourth chapter; Data analysis and results. Last but not least; the most important, the stem of the research is the Conclusion and recommendations, which is the last chapter of the research.

# Chapter 2

# LITERATURE REVIEW

## 2.1 What is a Portfolio?

Portfolio is a financial term denoting a collection of investments held by an investment company, hedge fund, financial institution or individual. Portfolios are held directly by investors and/or managed by financial professionals. Prudence suggests that investors should construct an diversified investment portfolio in accordance with risk tolerance and investing objectives. Forming portfolios can eliminate non systematic risk.

## 2.2 Portfolio Optimization

Portfolio optimization is the process of choosing the portions of various assets to be held in a portfolio, in such a way as to make the portfolio better than any other according to some criterion. The criterion will combine, directly or indirectly, knowledge of the expected value of the portfolio's rate of return as well as of the return's dispersion and possibly other measures of financial risk.

## 2.3 Mathematical tools used in Portfolio Optimization[2]

From the complexed and larger scaled of optimizing portfolios to the simplest portfolio requires the work to be done by computer. Central to this optimization is the construction of the covariance matrix for the rates of return on the assets in the portfolio.
Techniques include:

- Quadratic programming
- Nonlinear programming
- Mixed integer programming

- Meta-Heuristic Methods
- Stochastic programming for multistage portfolio optimization

### 2.3.1 Quadratic Programming

Quadratic Programming (QP) is a special type of mathematical optimization problem specifically, the problem of optimizing (minimizing or maximizing) a quadratic function of several variables subject to linear constraints on these variables. It is a particular type of nonlinear programming.

## 2.4 Diversification

Modern portfolio theory relies on diversification to minimize individual security risk in a portfolio. The idea is that by holding a large number of different securities, no individual security can seriously affect the performance of the portfolio and the investor is left with only systemic risk, which is the risk that the entire sector or market, will decline. It is possible to hedge against systematic risk, but it cannot be fully mitigated without giving up a significant portion of the potential returns.Diversification strives to smooth out unsystematic risk events in a portfolio so the positive performance of some investments neutralizes the negative performance of others. Therefore, the benefits of diversification hold only if the securities in the portfolio are not perfectly correlated.
Studies and mathematical models have shown that maintaining a well-diversified portfolio of 25 to 30 stocks yields the most cost-effective level of risk reduction. Investing in more securities yields further diversification benefits, albeit at a drastically smaller rate [3]

## 2.5 Modern Portfolio Theory-MPT

Modern portfolio theory (MPT) is a theory on how an investors can construct portfolios to optimize or maximize expected return accordance with a given level of risk, emphasizing that risk is an inherent part of higher return. According to the theory, it's possible to construct an "efficient frontier" of optimal portfolios offering the maximum possible expected return for a given level of risk. This theory was pioneered by Harry Markowitz in his paper "Portfolio Selection," published in 1952 by the Journal of Finance.

An investor can construct a portfolio of multiple securities that will maximize returns for a given level of risk. Like that, given a desired level of expected return, an investor can construct a portfolio with the lowest possible risk. Prudence show that on statistical measures such as variance and correlation, an individual investment's return is less important than how the investment behaves in the context of the entire portfolio.

### 2.5.1 Efficient Frontier

Every possible combination of assets that exists can be plotted on a graph, with the portfolio's risk on the X-axis and the expected return on the Y-axis. This plot reveals the most desirable portfolios.
Every possible combination of the risky assets, without including any holdings of the risk-free asset, can be plotted in risk-expected return space, and the collection of all such possible portfolios defines a region in this space. The left boundary of this region is a hyperbola, and the upper edge of this region is the efficient frontier in the absence of a risk-free asset (sometimes called "the Markowitz bullet"). Combinations along this upper edge represent portfolios (including no holdings of the risk-free asset) for which there is lowest risk for a given level of expected return. Equivalently, a portfolio lying on the efficient frontier represents the combination offering the best possible expected return for given risk level. Investing in any portfolio not on this curve is not desirable[4]

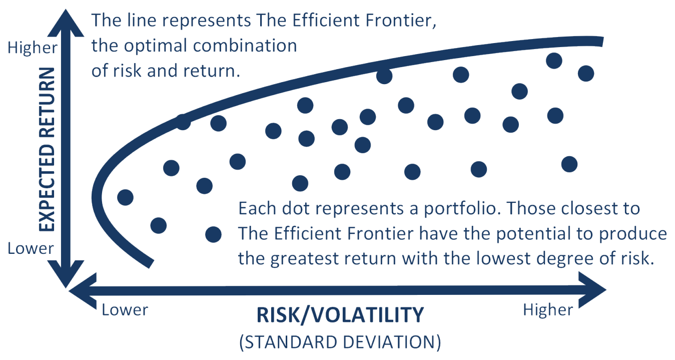

The line represents The Efficient Frontier, the optimal combination of risk and return.
Higher
EXPECTED RETURN
Lower
Each dot represents a portfolio. Those closest to The Efficient Frontier have the potential to produce the greatest return with the lowest degree of risk.
Lower
RISK/VOLATILITY
(STANDARD DEVIATION)
Higher

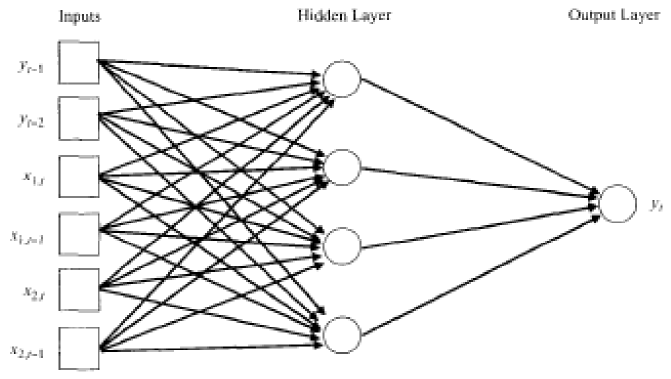
Figure 2.1: Efficient frontier

## 2.6 Neural Networks

Neural networks or artificial neural networks is motivated by the human brain processes. The neuron is the fundamental structural element which is the information processing module of the brain. The human brain has number of neurons and arranged in a highly complexed non linear parallel structure.

Artificial neural networks structured in a way that normal human brain would solve the problems. This is a black box type of model that is often used to model high dimensional non linear data.Hill(1994) is a basic reference on artificial neural networks and forecasting.

The model has several layers, the most common structure involves three layers, inputs which are the original predictors, the hidden layer comprised of a set of constructed variables and the output layer made up of responses. Each variable in a layer is called a node. In a node data is transformed. The transformation function or activation functions, are either sigmoidal (s shaped) or linear.



Let each of the k hidden layer nodes $a_{11}$be a linear combination of the input variables:

$$a_{ll} = \Sigma W_{1ju} X_j + \theta_u$$

where the $w_{1ju}$ are unknown parameters that must be estimated (called weights) and$\theta_u$ is a parameter that plays the role of an intercept in linear regression (this parameter is sometimes called the bias node).

Each node is transformed by the activation function $g()$. Much of the neural networks literature refers to these activation functions notationally as $\sigma_u$ because of their S shape. Let the output of node $a_u$ be denoted by $Zu = g(u)$. Now it is formed a linear combination of these outputs say b. Finally, the output response or the predicted value for y is a transformation of the b, say,$y = g(b)$, where $g(b)$ is the activation function for the response.

The neural network model is a very flexible form containing many parameters, and it is this feature that gives a neural network a nearly universal approximation property.

That is, it will fit many historical data sets very well. However the parameters in the underlying model must be estimated (parameter estimation is called "training" in the neural network literature), and there are a lot of them. The usual approach is to estimate the parameters by minimizing the overall residual sum of squares taken over all responses and all observations. This is a nonlinear least squares problem, and a variety of algorithms can be used to solve it. Often a procedure called backpropagation (which is a variation of steepest descent) is used, although derivative-based gradient methods have also been employed. As in any nonlinear estimation procedure starting values for the parameters must be specified in order to use these algorithms. It is customary to standardize all the input variables, so small essentially random values are chosen for the starting values[5]

### 2.6.1 Neural Networks (Matlab Neural Network toolbox)

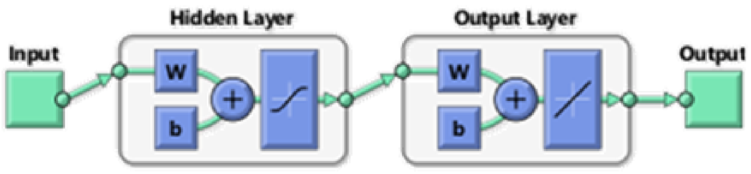


Figure 2.2: A Neural Network

A two layer feed forward network with sigmoid hidden neurons and linear output neurons can fit various multi-dimensional mapping problems with ease when consistent data and enough number of neurons to the hidden layer are given. The network will be trained with Levenberg-Marquardt back propagation algorithm unless there is not enough memory in which case Scaled Conjugate Gradient back propagation will be used. In validating and testing data 70 percent for training, 15 percent for validating and 15 percent for testing is used.
Training- These are presented to the network during training and network is adjusted according to its error. Followings are the training methods which can be used.
***Levenberg- Marquardt back propagation***
This algorithm typically requires more memory but less time. Training automatically stops when generalization stops improving, as indicated by an increase in the mean

square error of the validation samples.

***Bayesian Regularization***

This algorithm typically requires more time, but can result in good generalization for difficult, small or noisy datasets. Training stops according to adaptive weight minimization (regularization).

***Scaled Conjugate Gradient***

This algorithm requires less memory. Training automatically stops when generalization stops improving, as indicated by an increase in the mean square error of the validation samples.

Validation- These are used to measure network generalization and to halt training when generalization stops improving.

Testing- These have no effect on training and so provide an independent measure of network performance during and after training.

Training multiple times will generate different results due to different initial conditions and sampling. Mean Squared Error (mse) is the average squared difference between outputs and targets. Lower values are better. Zero means no error. Regression R values measure the correlation between outputs and targets. A r value 1 means a close relationship. 0 means random relationship.

## 2.7 Related other surveys

### 2.7.1 Sensitivity Analysis in Convex Quadratic Optimization: Simultaneous Perturbation of the Objective and Right-Hand-Side Vectors

*Alireza Ghaffari Hadigheh Oleksandr Romanko and Tam´ as Terlaky*

In this paper they have studied the behavior of Convex Quadratic Optimization problems when variation occurs simultaneously in the right-hand side vector of the constraints and in the coefficient vector of the linear term in the objective function. It is proven that the optimal value function is piecewise-quadratic. The concepts of transition point and invariancy interval are generalized to the case of simultaneous perturbation. Criteria for convexity, concavity or linearity of the optimal value function on invariancy intervals are derived. Furthermore, differentiability of the optimal value function is studied, and linear optimization problems are given to calculate the left and right derivatives. An algorithm, that is capable to compute the transition points and optimal partitions on all invariancy intervals, is outlined. They specialized the method to Linear Optimization problems and provide a practical example of simultaneous perturbation parametric quadratic optimization problem from electrical engineering[6]

### 2.7.2 An Interior Point Approach to Quadratic and Parametric Quadratic Optimization

*Oleksandr Romanko*

In this thesis sensitivity analysis for quadratic optimization problems is studied. In sensitivity analysis, which is often referred to as parametric optimization or parametric programming, a perturbation parameter is introduced into the optimization problem, which means that the coefficients in the objective function of the problem and in the right-hand-side of the constraints are perturbed. First, they describe quadratic programming problems and their parametric versions. Second, the theory for finding solutions of the parametric problems is developed. they also present an algorithm for solving such problems. In the implementation part, the implementation of the quadratic optimiza- tion solver is made. For that purpose, they extend the linear interior point package McIPM to solve quadratic problems. The quadratic solver is tested on the problems from the Maros and M´esz´aros test set. Finally, they implement the algorithm for parametric quadratic optimization. It utilizes the quadratic solver to solve auxiliary problems. they present numerical results produced by our parametric optimization package[7]

### 2.7.3 On the Sensitivity of Mean-Variance Efficient Portfolios to Changes in Asset Means: Some Analytical and Computational Results

*Michael J. Best University of Waterloo*

*Robert R. Grauer Simon Fraser University*

This paper investigates the sensitivity of mean- variance(MV-Mean Variance)-efficientportfolios to changes in the means of individual assets. When only a budget constraint is imposed on the investment problem, the analytical results indicate that an MV-efficient portfolio's weights, mean, and variance can be extremely sensitive to changes in asset means. When nonnegativity constraints are also imposed on the problem, the computational results confirm that a positively weighted MV-efficient portfolio's weights are extremely sensitive to changes in asset means, but the portfolio's returns are not. A surprisingly small increase in the mean of Just one asset drives half the securities from the portfolio.

Yet the portfolio's expected return and standard deviation are virtually unchanged[8]

### 2.7.4 Portfolio Selection

*Harry Markowitz*

The process of selecting a portfolio may be divided into two stages. The first stage starts with observation and experience and ends with beliefs about the future performances of available securities. The second stage starts with the relevant beliefs about future performances and ends with the choice of portfolio. This paper is concerned with the second stage. He first considers the rule that the investor does (or should) maximize discounted expected, or anticipated, returns. This rule is rejected both as a hypothesis to explain, and as a maximum to guide investment behavior. He next considers the rule that the investor does (or should) consider expected return a desirable thing and variance of return an undesirable thing. This rule has many sound points, both as a maxim for, and hypothesis about, investment behavior. He illustrates geometrically relations between beliefs and choice of portfolio according to the "expected returns-variance of returns" rule.

### 2.7.5 Portfolio optimization using the quadratic optimization system and publicly available information on the WWW

*Jivendra K. Kale*

Purpose - The purpose of this paper is to describe some optimization exercises which have proved to be very useful for introducing students to Markowitz-style mean-varience optimization.

Design/methodology/approach - This paper describes two exercises that walk students through the process of gathering security price and dividend data, estimating the parameters of the joint distribution of asset returns, and then using a portfolio optimizer to construct mean-variance efficient portfolios. It describes the basic methodology, and the more complex formulations of the portfolio optimization problem that are used in practice.

Practical implications - Portfolio selection is typically taught in finance courses as an abstract solution to a system of equations, and does little to connect the portfolio construction process to Exchange Traded Funds, stocks, bonds and other assets that are traded in markets. This study offers a practical approach to teaching portfolio optimiza-

tion, that starts with gathering market data and shows how a quadratic optimization system is used to construct mean-variance optimal portfolios.
Originality/value - The exercises in this case study prepare students to construct mean-variance efficient portfolios for asset allocation with Exchange Traded Funds, and for building stock and bond portfolios, using market data and a portfolio optimizer.

### 2.7.6 Sensitivity Analysis for Mean-Variance Portfolio Problems

*Michael J. Best*

*Robert R. Grauer*

This paper shows how to perform sensitivity analysis for Mean-Variance (MV) portfolio problems using a general form of parametric quadratic programming. The analysis allows an investor to examine how parametric changes in either the means or the right-hand side of the constraints affect the composition, mean, and variance of the optimal portfolio. The optimal portfolio and associated multipliers are piecewise linear functions of the changes in either the means or the right-hand side of the constraints. The parametric parts of the solution show the rates of substitution of securities in the optimal portfolio, while the parametric parts of the multipliers show the rates at which constraints are either tightening or loosening. Furthermore, the parametric parts of the solution and multipliers change in different intervals when constraints become active or inactive. The optimal MV paths for sensitivity analyses are piecewise parabolic, as in traditional MV analysis. However, the optimal paths may contain negatively sloping segments and are characterized by types of kinks, i.e., points of nondifferentiability, not found in MV analysis.

# Chapter 3

# METHODOLOGY

The methodology of this research is comprised of several main sections, which are data collection criterion, creating perturbed return matrix and finding optimum portfolio for each perturbed factor,creating color map, plotting efficient frontier and model fitting for output. During this project used data was secondary data since it had been collected from the internet . Also data was quantitative. The quantitative method is based on the transformation on information into numbers in order to make an assumption and to get a conclusion (Holme and Solvang, 1997). Apart from that the methodology has followed a deductive method. The research approach , will use is deductive, which means that it will be used existing theory and common principles as a starting point, and then base this thesis from the foundations from the theory. This way will give a logical conclusion if it is logically connected (Eriksson and Wiedersheim-Paul, 1999).

## 3.1 Data collection criterion

The purpose of this project is to investigate about the behavior of optimum portfolio when there is very small perturbation in the return matrix. To accomplish this quantitative data was used from the Singapore Stock Exchange's daily historical data from 4/4/2013 to 4/4/2014. The used numerical data is the daily trading prices of the randomly selected ten companies. The used data is called secondary data because it has been used readily available data from the internet. The selected ten companies are as given below.

1. United Fiber System Ltd- ( P30.SI)
2. Gallant Venture Ltd- (5IG.SI)
3. Baidu Inc- (BIDU)

4. China Mobile Ltd- (CHL)
5. China Southern Airlines Co. Ltd- (ZNH)
6. Dairy Farm International Holdings Ltd- (D01.SI)
7. DBS Group Holdings Ltd-( D05.SI)
8. Keppel Co. Ltd -( BN4.SI )
9. Koh Brothers Ltd - (K75.SI )
10. Delfi Ltd - (P34.SI)

### 3.2 Research method

As the next progression an Excel spreadsheet named SGXdata is prepared to calculate the daily return of the each security. The used formulae is,

$$\text{Return}=\frac{S_{i+1}-S_i}{S_i}$$

Where $S_{i+1}$ = One step ahead price of the security , $S_i$ = Present price of the security. Next step is to save SGXdata as a CSV(comma delimited) file and import it to the Matlab workspace.

The created PortfolioOpt mfile consisted of several sub sections. First section is creating the perturbed return matrix and calculating the optimum weights (x) of the portfolio related to the each perturbed return value groups (delta_$r$). Second section is the graphical representation of Efficient frontier of each perturbed return value groups. Third section is the plotting contour maps or color maps of the correlation values related to each perturbed return value group. Those are the targets to complete m file and from the obtained data of optimum weights(x) and perturbed return matrix ( delta_$r$) a model is fitted using Matlab Neural networks tool box.

Pursuing the first goal the first step is to call out the SGX data or return matrix of each company which is in the form of csv file to the m-script. Then forming the perturbed return values using random values between -1 and 1. The limits are -1 and 1 because here it has to be considered a very small change of return value of each company. Another fact is that when there is a small change in the return that value must change the whole lot of returns from time to time by the same magnitude of that considered company. This whole criteria is used because it has to be imagined a real situation where one company's return value increased by some amount and another company's value may go down by some amount etc. Before summing up with the real return matrix (SGXdata) the matrix of zeros are introduced to increase the efficiency. Next step is the summing up the SGXdata and delta_*r*. Then calculation of mean values, covariance values and inverse covariance values (Hessian matrix) is done.Also correlations. Finally optimum weights are calculated using matlab inbuilt function quadprog. Here the risk averse parameter is taken as 0.1.

Second section consists of graphical representation of Efficient frontier. First part of the code gives the calculation of minimum variance portfolio which determines the minimum variance and the related mean. Second part is the calculation of minimum variance line. For hundred points between minimum mean and maximum mean risk and return are calculated using a for loop. And finally efficient frontier is plotted with efficient part in green color and non- efficient part with red in color with dashed line. The minimum variance is marked with blue marker.

Then the optimum weights (x) and perturbed return values(delta_r) are written on two separate excel sheets using xlwrite inbuilt function. And third section is plotting contour3 map of the findings. The correlations are calculated and using contour3, a map between delta_r , x values and correlations is drawn.

Neural Networks is a tool box that comes with the Matlab software and by getting use of it a Neural network is fitted to the obtained data of perturbed values of return ;delta_r versus optimum weights (x) without short selling. Basically several networks are fitted and among them the best model is chosen as considering factors such as mean squared error (mse) of the model, performance graph and regression graphs etc.

First the rough idea about the number of hidden neurons is taken by running the code No_of_hidden_neurons code.To fit a model initial steps are like below. First opening the neural network start GUI with the command nnstart. Second step is clicking Fitting Tool to open the Neural Network Fitting Tool. Then using the Inputs and Targets options in the Select Data window data is loaded from the Matlab workspace. (delta_r and x).By clicking Next it displays the Validation and Test Data window. The validation and test data sets are each set to 15percent of the original data. Then it displays the Feed forward network. The standard network that is used for function fitting is a two-layer feedforward network, with a sigmoid transfer function in the hidden layer and a linear transfer function in the output layer. The default number of hidden neurons is set to 10. It might have to increase this number later, if the network training performance is poor. As the training method Scaled Conjugate Gradient (trainscg) is used as it uses gradient calculations which are more memory efficient than the Jacobian calculations the other two algorithms use. Then observing the plots it might have to train it again or increase the number of neurons. If satisfied by proceeding in to the next step , the matlab codes(MyNeuralNetworkfunction and Advanced Matlab script file) can be obtained.Finally by running Neural Network function for the PortfolioOptinsample values forecasted values are obtained and mse measurement is done with the obtained output from running Portfoliooptinsample in the PortfolioOpt matlab m file(PortfolioOptoutsample).

As the further developments when the value of perturbation is increased individually for one company only, to drive away at least one company from the optimum weights (to zero the optimum weight of one company) it was observed the behaviour of the perturbation value.If negatively mean companies are removed and perturbation is done for positively mean companies the effort to remove at least one company from the optimum weights is also observed.

## 3.3 The models

### 3.3.1 Model with perturbation

$$(QP) min c^T x + \frac{1}{2} x^T Qx : Ax = b, x \geq 0 \text{ ,}$$

Above model is related to Portfolio Optimization when vector c is equal to expected returns of the each company,Q is the covariance matrix, $Ax = b$ is the budget constraint and x is the unknown weights vector($x \geq 0$ because no short selling is allowed)

The model with perturbation is

$$(QP_{\lambda_c}) min(c + \lambda_c \bigtriangleup c)^T x + \frac{1}{2} x^T Qx : Ax = b, x \geq 0 \text{ ,}$$

$\lambda_c$ is the Risk aversion parameter.This is taken as -0.1 in the Matlab m file.

### 3.3.2 Quadratic Programming[9]

A general quadratic function of n variables can be written more compactly as,

$$f(x) = c'x + \frac{1}{2}x'Qx$$

It is assumed that Q is symmetric (for it is not it may be replaced by $\frac{1}{2}[Q+Q']$which is symmetric and leaves the value of f unchanged.) The gradient of f at x is the n-vector of its partial derivatives of f evaluated at x and is denoted by $\bigtriangledown f(x)$. will always be a covariance matrix and as such is symmetric and positive semi definite. Consider the model problem

$$min(c'x + \frac{1}{2}x'Qx|Ax = b)$$

where c and x are (n,1) , Q is (n,n), symmetric and positive semidefinite, A(m,n) and b is (m,1). Thus our model problem has n variables and m equality constraints. The constraints can equally well be written as

$$a_i'x = b_i, i = 1, 2, .....m,$$

where each $a_i$ is a n-vector, $A' = [a_1, a_2, \cdots, a_m]$ and $b = (b_1, b_2, \cdots, b_m)'$ In this format, $a_i$ is the gradient of the i-th constraint function.

$$\bigtriangledown f(x) = u_1a_1 + u_2a_2 + .......... + u_ma_m$$

which says that $\bigtriangledown f(x)$ must be a linear combination of the gradients of the constraints. replacing $u_i$ with $-u_i$and it can be written more compactly,

$$-\bigtriangledown f(x) = A'u,$$

where $u = (u_1, u_2, ....., u_m)'$

The optimality conditions for above, $min(c'x + \frac{1}{2}x'Qx|Ax = b)$ are (1) $Ax_0 = b$ (2) there exists a vector u with $-\bigtriangledown f(x) = A'u$,

### 3.3.3 Markowitz Portfolio Optimization model

The method developed by Harry Markowitz in 1956 to solve mean-variance problem is called as Markowitz Portfolio optimization method.

#### 3.3.3.1 Assumptions of the Markowitz Portfolio Model

- Investors consider each investment alternative as being represented by a probability distribution of expected returns over some holding period.
- Investors maximize one-period expected utility and their utility curves demonstrate diminishing marginal utility of wealth.
- Investors estimate risk on basis of variability of expected returns.
- Investors base decisions solely on expected return and risk.
- Investors prefer higher returns to lower risk and lower risk for the same level of return.

#### 3.3.3.2 Risk and Expected Return on a portfolio[10]

A portfolio constructed from n different securities can be described in terms of their weights

$$x_i = \frac{k_i S_i(0)}{V(0)}, i = 1, \ldots.., n$$

where $k_i$ is the number of shares of type i in the portfolio, $S_i(0)$ is the initial price of security i, and V(0) is the amount initially invested in the portfolio. It will prove convenient to arrange the weights into one-row matrix.

$$x = [x_1 x_2 \ldots\ldots.. x_n]$$

The weights can be added up to one, which can be written in the matrix form as,

$$1 = ux^T$$

$$u = [11.....1]$$

is one row matrix with all n entries equal to 1,$x^T$ is a one column matrix, the transpose of x, and the usual matrix multiplication rules apply. The attainable set consists of all portfolios with weights x satisfying called the attainable portfolios.
Suppose that the returns on the securities are K1,.....,Kn. The expeced returns $\mu_i = E(Ki)$ for $i = 1, \ldots., n$ will also be arranged into one row matrix.

$$m = [\mu_1 \mu_2 .....\mu_n]$$

The covariances between returns will be denoted by $q_{ij} = Cov(K_i, K_j)$. They are the entries of $n \times n$ covariance matrix.

$$Q = \begin{bmatrix} q_{11} & q_{12} & \cdots & q_{1n} \\ q_{21} & q_{22} & \cdots & q_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ q_{n1} & q_{n2} & \cdots & q_{nn} \end{bmatrix}$$

It is known that the covariance matrix is symmetric and positive definite. The diagonal elements are simply the variances of returns, $q_{ij} = Var(K_i)$. In what follows it can be assumed in addition that Q has an inverse $Q^{-1}$

*Proposition 3.1* The expected return $\mu_v = E(K_v)$ and variance $\sigma_v^2 = Var(K_v)$ of a portfolio with weights x are given by

$$\mu_v = mx^T$$

$$\sigma_v^2 = xQx^T$$

- To find a portfolio with the smallest variance in the attainable set. It will be called the minimum variance portfolio.
- To find a portfolio with the smallest variance among all portfolios in the attainable set whose expected return is equal to a given number $\mu_v$. The family of such portfolios parameterized by $\mu_v$, is called the minimum variance line.

#### 3.3.3.3 Finding minimum variance portfolio

The portfolio with the smallest variance in the attainable set has weights, $\frac{uQ^{-1}}{uQ^{-1}u^T}$ provided that the denominator is non-zero.

Where $x = [x_1 x_2 \cdots\cdots x_n]$ $x_i$=weights of securities

$u = [111 \cdots\cdot 1]$

$$Q = \begin{bmatrix} q_{11} & q_{12} & \cdots & q_{1n} \\ q_{21} & q_{22} & \cdots & q_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ q_{n1} & q_{n2} & \cdots & q_{nn} \end{bmatrix}$$

Suppose that the returns on the securities are $K_1 \cdots \cdot K_n$

#### 3.3.3.4 Finding minimum variance line with plotting efficient frontier

The portfolio with the smallest variance among attainable portfolios with expected return $\mu_v$ has weights,

$$\mathbf{w} = \frac{\begin{vmatrix} 1 & uQ^{-1}m^T \\ \mu v & mQ^{-1}m^T \end{vmatrix} uQ^{-1} + \begin{vmatrix} uQ^{-1}u^T & 1 \\ mQ^{-1}u^T & \mu v \end{vmatrix} mQ^{-1}}{\begin{vmatrix} uQ^{-1}u^T & uQ^{-1}m^T \\ mQ^{-1}u^T & mQ^{-1}m^T \end{vmatrix}}$$

provided that the determinant in the denominator is non-zero. The weights depend linearly on $\mu_v$.

### 3.3.4 Neural network model

#### 3.3.4.1 Feed-forward neural networks

A neural network processes information from one layer to the next by an 'activation function'. Consider a feed-forward network with one hidden layer. The jth node in the hidden layer is defined as,

$$h_j = f_j\left(\alpha_{0j} + \sum_{i \to j} w_{ij} x_i\right),$$

where xi is the value of the ith input node, fj (.) is an activation function typically taken to be the logistic function,

$$f_j(z) = \frac{\exp(z)}{1 + \exp(z)},$$

$\alpha_{0j}$ is called the bias, the summation $i \longrightarrow j$ means summing over all input nodes feeding to j, and $w_{ij}$ are the weights

For the output layer, the node is defined as,

$$o = f_o\left(\alpha_{0o} + \sum_{j \to o} w_{jo} h_j\right)$$

where the activation function fo(.) is either linear or a Heaviside function. If fo(.) is linear, then

$$o = \alpha_{0o} + \sum_{j=1}^{k} w_{jo} h_j,$$

where k is the number of nodes in the hidden layer. By a Heaviside function, we mean $fo(z) = 1$ if $z > 0$ and $fo(z) = 0$ otherwise. A neuron with a Heaviside function is called a threshold neuron, with 1 denoting that the neuron fires its message.Combining the layers, the output of a feed-forward neural network can be written as

$$o = f_o \left[ \alpha_{0o} + \sum_{j \to o} w_{jo} f_j \left( \alpha_{0j} + \sum_{i \to j} w_{ij} x_i \right) \right]$$

If one also allows for direct connections from the input layer to the output layer, then the network becomes

$$o = f_o \left[ \alpha_{0o} + \sum_{i \to o} \alpha_{io} x_i + \sum_{j \to o} w_{jo} f_j \left( \alpha_{0j} + \sum_{i \to j} w_{ij} x_i \right) \right]$$

where the first summation is summing over the input nodes. When the activation function of the output layer is linear, the direct connections from the input nodes to the output node represent a linear function between the inputs and output. Consequently, in this particular case model is a generalization of linear models.

#### 3.3.4.2 Fitting data

Neural networks are good fitting tools. In fact there is a proof that a fairly simple neural network can model data of any practical function. The standard network that is used for function fitting is a two layer feed forward network with a sigmoid transfer in the hidden layer and a linear transfer function in the output layer. The default number of hidden neurons is set to 10. It might want to increase the this number later if the network training performance is poor.

Levenberg- Marquardt (trainlm) is recommended for most problems, but for some noisy and small problems Bayesian Regularization (trainbr) can take longer but obtain a better solution. For large problems however Scaled Conjugate Gradient (trainsg) is recommended as it uses gradient calculation which are more memory efficient than the Jacobian calculations the other two algorithm use.

#### 3.3.4.3 Number of layers and nodes in the feed forward neural network

Producing the neural network architecture means choosing the number of hidden layers and the number of neurons in the hidden layer and other layers.
The input layer- With respect to the number of neurons comprising this layer this parameter is completely and uniquely determined once it is known the shape of the training data. Specifically the number of neurons comprising the layer is equal to the number of features in the data. Some configurations add one additional node for bias term.
The output layer- Every neural network has exactly one output layer. Determining its size or the number of neurons is simple and it is solely depends on the configuration.
The hidden layers- There's a consensus that the performance difference from adding additional hidden layers, the situations in which performance improves with a second (or third etc) hidden layer are very small. One hidden layer is sufficient for the large majority of problems. For the size of the hidden layer there are some empirically derived rules of thumb of these the most commonly relied on is the optimal size of the hidden layer is usually between the size of the input and size of the output layers. For most of the problems one could probably get decent performance (even without a second optimization step) by setting the hidden layer configuration using just two rules. 1) number of hidden layers equals one and 2) the number of neurons in that layer is the mean of the neurons in the input and output layers.

#### 3.3.4.4 Improve Neural Network generalization and avoid overfitting

One of the problems that occur during neural network training is called overfitting. The error on the training set is driven to a very small value but when new data is presented to the network the error is large. The network has memorized the training examples but it has not learned to new circumstances or to generalize the situation.
One of the methods to improve generalization is to use a network that is just large enough to provide adequate fit. The larger the network used the more complex the functions the network can create.If a small enough network is used it will not have enough power to overfit the data.

## 3.4 Efficient Frontier

Rational investor always prefers lower risk. If a rational investor is given with two choices to select in between two securities he will select higher expected return and lower standard deviation.

A portfolio is called efficient if there is no other portfolio except it self that dominates it. The set of efficient portfolios among all attainable portfolios is called the efficient frontier.

Every rational investor will choose an efficient portfolio, always preferring a dominating portfolio to the most dominated one. How ever different investors may have different ideas but they also select portfolios on the efficient frontier depending on their individual preferences. Given two efficient portfolios with$\mu_1 \leq \mu_2$ and $\sigma_1 \leq \sigma_2$ , a cautious person or rational person may prefer that with lower risk $\sigma_1$ and lower expected return $\mu_1$ while others may choose a portfolio with higher risk $\sigma_2$, regarding the higher expected return $\mu_2$ as compensation for increased risk.

The value of the efficient frontier is that it takes the stem out of information having n assets into a two dimensional representation. In particular an efficient portfolio has the highest expected return among all attainable portfolios with the same standard deviation (the same risk),and has the lowest standard deviation (the lowest risk) among all attainable portfolios with the same expected return. As a result the efficient frontier must be a subset of the minimum variance line.

# Chapter 4

# DATA ANALYSIS AND RESULTS

This chapter is where descriptive analysis of the obtained results are presented. For the ease of presentation chapter is categorized in to several sections.The Matlab m-file output is critically analyzed as the first progression. As the final section analysis of the fitted model using Neural Network tool-box is presented.

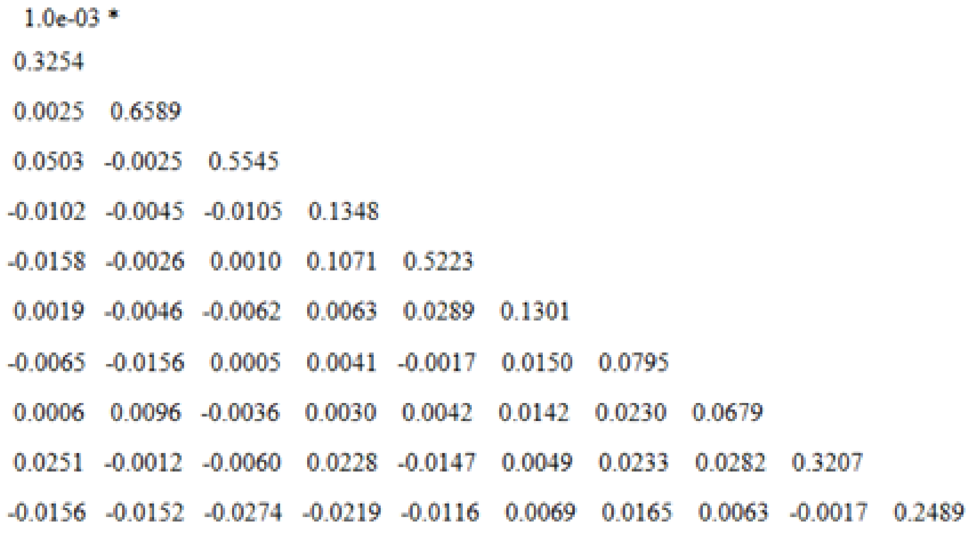
## 4.1 Analysis of Matlab output

### 4.1.1 Covariance matrix

According to the model,

$$(QP_{\lambda_c}) min(c + \lambda_c \bigtriangleup c)^T x + \tfrac{1}{2} x^T Qx : Ax = b, x \geq 0 \ ,$$

the covariance matrix output (matlab output) is given below. When it is observed the

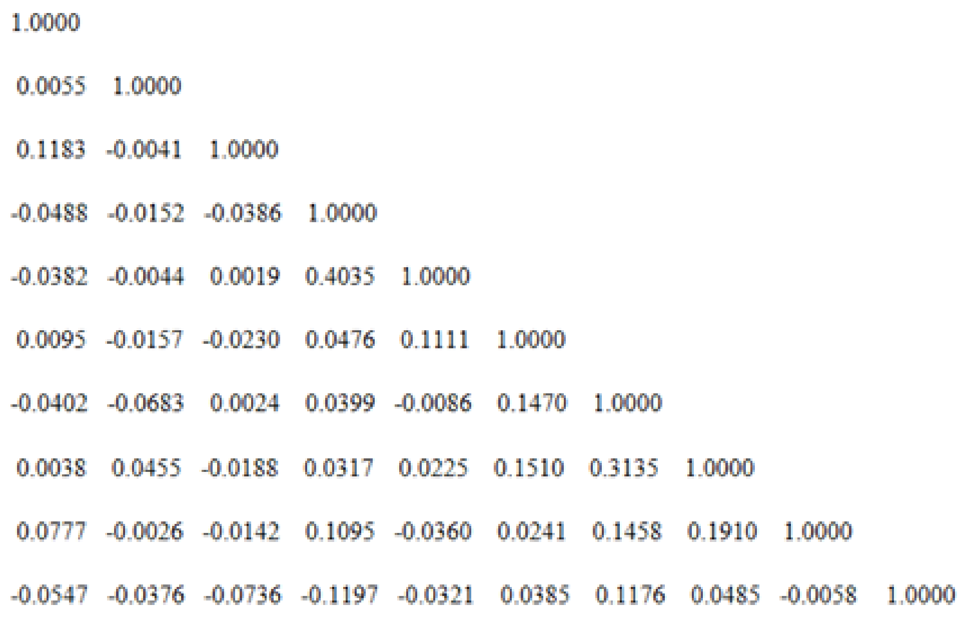


```
 1.0e-03 *
 0.3254
 0.0025   0.6589
 0.0503  -0.0025   0.5545
-0.0102  -0.0045  -0.0105   0.1348
-0.0158  -0.0026   0.0010   0.1071   0.5223
 0.0019  -0.0046  -0.0062   0.0063   0.0289   0.1301
-0.0065  -0.0156   0.0005   0.0041  -0.0017   0.0150   0.0795
 0.0006   0.0096  -0.0036   0.0030   0.0042   0.0142   0.0230   0.0679
 0.0251  -0.0012  -0.0060   0.0228  -0.0147   0.0049   0.0233   0.0282   0.3207
-0.0156  -0.0152  -0.0274  -0.0219  -0.0116   0.0069   0.0165   0.0063  -0.0017   0.2489
```

Figure 4.1: Covariance matrix output

Covariance-variance matrices related to each set of perturbed values of returns, they are actually same for about 14 decimal places. That means the covariance matrix does not change with the perturbation in the mean vector. And the covariance matrix is symmetric around the diagonal.
Here there are some positive covariances and some negative covariances. If two investments tend to be up or down during the same time periods, then they have positive covariance. The companies having positive covariances are companies (1,2), (1,3), (5,3), (5,4), (6,1), (6,4), (6,5), (7,3), (7,4), (7,6), (8,1), (8,2), (8,4), (8,5), (8,6), (8,7), (9,1) ,(9,4), (9,6), (9,7), (9,8), (10,6), (10,7), (10,8). All the other covariances are negative covariances except the diagonal which are the variances. That means one investment tends to be up while the other is down, so they have negative covariance.And covariance is a absolute measure of association between returns of two companies. Highest covariance or highest association is 0.1071 which is between company five and company four.Lowest covariance is between companies five and three.

### 4.1.2 Correlation matrix



```
 1.0000

 0.0055  1.0000

 0.1183 -0.0041  1.0000

-0.0488 -0.0152 -0.0386  1.0000

-0.0382 -0.0044  0.0019  0.4035  1.0000

 0.0095 -0.0157 -0.0230  0.0476  0.1111  1.0000

-0.0402 -0.0683  0.0024  0.0399 -0.0086  0.1470  1.0000

 0.0038  0.0455 -0.0188  0.0317  0.0225  0.1510  0.3135  1.0000

 0.0777 -0.0026 -0.0142  0.1095 -0.0360  0.0241  0.1458  0.1910  1.0000

-0.0547 -0.0376 -0.0736 -0.1197 -0.0321  0.0385  0.1176  0.0485 -0.0058  1.0000
```

Figure 4.2: Correlation matrix output

Correlation matrix is also symmetric around the diagonal. The Correlation matrix does not change with the perturbation in the mean vector. A Correlation coefficients measures the how much amount of the association between two variables. Each such coefficient must lie between -1 and +1, inclusive. Here also there are positive and negative correlations. A positive coefficient indicates a positive association: a greater-than-expected outcome for one variable is likely to be associated with a greater- than-expected outcome for the other while a smaller-than-expected outcome for one is likely to be associated with a smaller-than-expected outcome for the other. A negative coefficient indicates a negative association: a greater-than-expected outcome for one variable is likely to be associated with a smaller-than-expected outcome for the other while a smaller-than- expected outcome for one is likely to be associated with a greater-than-expected outcome for the other.

Because of the risk of losing money from the both investments simultaneously the positively correlated securities are more riskier. The most diversified portfolio consists of securities with the greatest negative correlation. A diversified portfolio can also be achieved by investing in uncorrelated assets, but there will be times when the investments will be both up or down, and thus, a portfolio of uncorrelated assets will have a greater degree of risk, but it is still significantly less than positively correlated investments. Here according to this figure and above numeric results there are 24 positive correlations and 21 negative correlations. So this portfolio is a fifty-fifty situation which means this portfolio is not perfectly diversified. However, even positively correlated investments will be less risky than single assets or investments that are perfectly positively correlated. However, there is no reduction in risk by combining assets that are perfectly correlated. Correlation is a relative measure of association between two companies.The highest association is between companies five and four. Lowest association is between companies five and three.

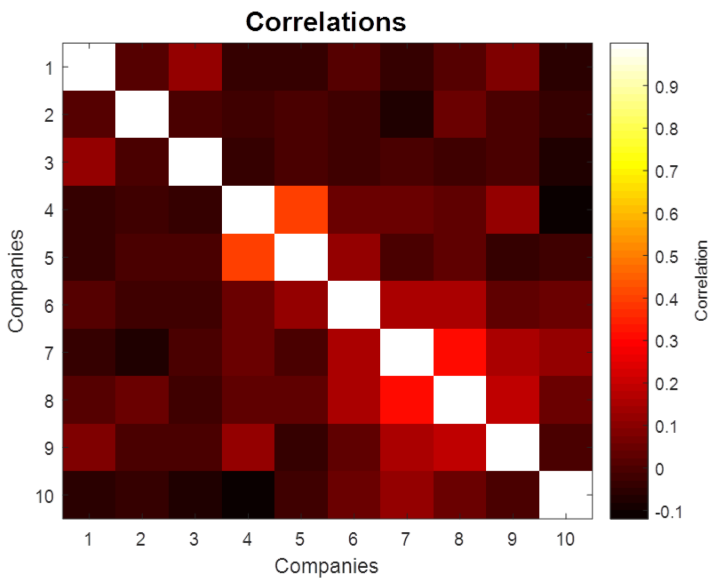


Figure 4.3: Correlation color map

#### 4.1.2.1 Why Correlation matrix is almost same for every perturbation value?

It is a known fact that if covariance-variance matrix does not change correlation matrix also does not change according to following formula.

$$\text{Correlation}= \frac{Covarince(x,y)}{Variance(x)*Variance(y)}$$

Then the question arises why the covariance doesn't change. The following model that had been used as our main model gives the answer. This shows that though there is perturbation there is no change in the covariance- variance matrix.Q is the covariance variance matrix and $\triangle\ c$ is the perturbation in mean.

$$(QP_{\lambda_c})min(c+\lambda_c \triangle c)^T x+0.5x^T Qx : Ax = b, x \geq 0 \ ,$$

### 4.1.3 Efficient frontier

When Efficient frontier is plotted the efficient part of the each frontier is plotted in green in color and inefficient part is plotted in red in color. For all the frontiers the standard deviation of the minimum variance portfolio is exactly 0.00484 and, but the mean of the minimum variance portfolio is spanned between 0.6287 and -0.6496. The mean or the expected return means the return obtained by each portfolio. So the from the span of the mean of the minimum variance portfolio it can be omitted the negative part because those portfolios don't give any return. Due to that reason from these portfolios the return that can be obtained is only up-to 0.6287 per one Singapore Dollar. But the minimum risk or the standard deviation doesn't change and remains at 0.00484. By definition, the variance of a portfolio's return is the expected value of the squared deviation of the actual return from the portfolio's expected return. The minimum standard deviation doesn?t change because the the formula of calculating minimum standard deviation is independent from $\mu$ or the mean of return vector. The mean of the optimum portfolio is a function of mean themselves and the change of the means.

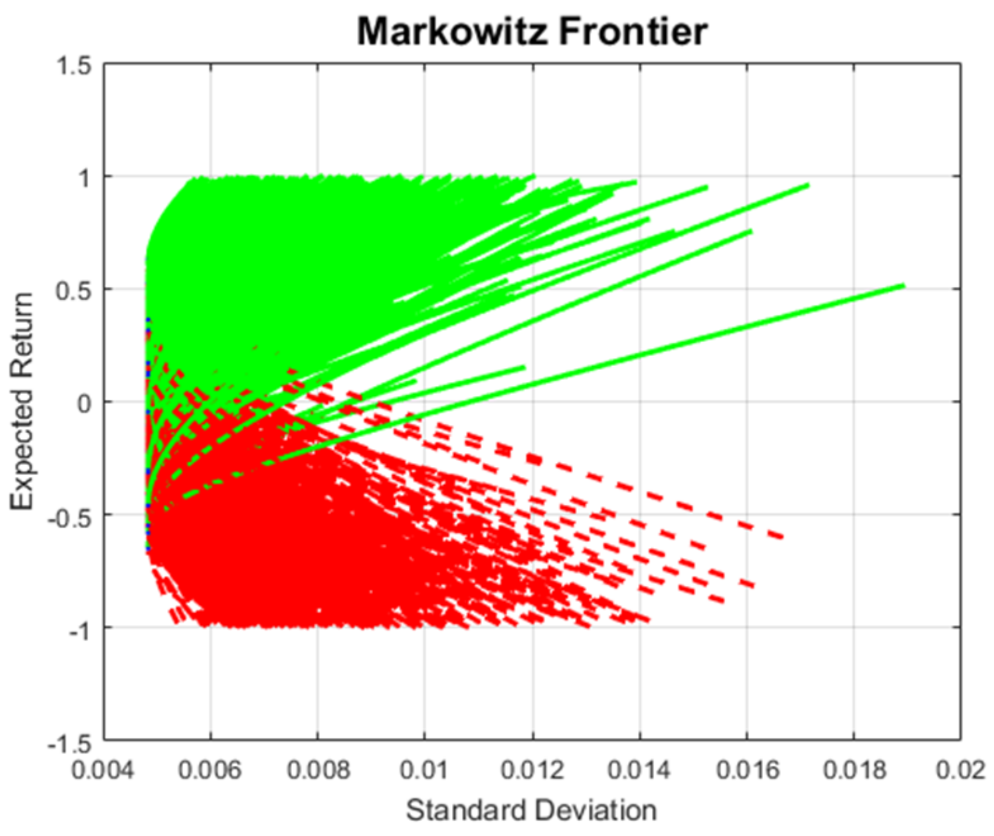


Figure 4.4: Plot of Efficient frontiers

### 4.1.4 Optimum weights

| P30.SI | 5IG.SI | BIDU | CHL | ZNH | D01.SI | D05.SI | BN4.SI | K75.SI | P34.SI |
|---|---|---|---|---|---|---|---|---|---|
| 0.103523 | 0.180792 | 0.159123 | 0.06677 | 0.178588 | 0.057143 | 0.039949 | 0.044388 | 0.116098 | 0.053626 |
| 0.103527 | 0.180741 | 0.159135 | 0.06677 | 0.178541 | 0.057141 | 0.039958 | 0.044399 | 0.116133 | 0.053654 |
| 0.103523 | 0.180695 | 0.159186 | 0.066772 | 0.178539 | 0.057141 | 0.039964 | 0.044403 | 0.116157 | 0.053621 |
| 0.103555 | 0.180739 | 0.159157 | 0.066768 | 0.178519 | 0.057137 | 0.039958 | 0.044397 | 0.116144 | 0.053626 |
| 0.103508 | 0.180804 | 0.159103 | 0.066783 | 0.178535 | 0.057152 | 0.039957 | 0.044397 | 0.116123 | 0.053637 |
| 0.103548 | 0.180728 | 0.159106 | 0.066789 | 0.178579 | 0.057148 | 0.039952 | 0.044394 | 0.116126 | 0.053631 |
| 0.103501 | 0.180765 | 0.159128 | 0.066785 | 0.178578 | 0.057129 | 0.039957 | 0.044397 | 0.116121 | 0.05364 |
| 0.103536 | 0.180748 | 0.159172 | 0.066763 | 0.178524 | 0.057127 | 0.039953 | 0.044394 | 0.116151 | 0.05363 |
| 0.103522 | 0.180728 | 0.159178 | 0.066765 | 0.178552 | 0.057132 | 0.039953 | 0.044391 | 0.116122 | 0.053658 |
| 0.103503 | 0.180761 | 0.159196 | 0.066779 | 0.178526 | 0.057148 | 0.039949 | 0.044391 | 0.116138 | 0.053611 |
| 0.103551 | 0.180722 | 0.159181 | 0.066772 | 0.178549 | 0.057136 | 0.039948 | 0.044386 | 0.116099 | 0.053656 |
| 0.103518 | 0.180761 | 0.159159 | 0.066779 | 0.178577 | 0.057133 | 0.039943 | 0.044388 | 0.116129 | 0.053614 |
| 0.103537 | 0.180789 | 0.159137 | 0.066758 | 0.17851 | 0.057143 | 0.039959 | 0.044396 | 0.116116 | 0.053655 |
| 0.103537 | 0.180754 | 0.159161 | 0.066774 | 0.178531 | 0.057134 | 0.039959 | 0.044397 | 0.116138 | 0.053616 |
| 0.103537 | 0.180702 | 0.159162 | 0.066766 | 0.178573 | 0.057129 | 0.039958 | 0.04439 | 0.116129 | 0.053653 |

Table 4.1: Part of the Optimum weights table

#### 4.1.4.1 Risk Aversion parameter

In practice, the risk aversion parameters signify trade-offs between the portfolio's risk or component risk and the expected return. They represent the relative importance of the risk terms they are associated with.In this research $RA = 0.1$ is used that means the risk and return terms will have equal weights. Setting large values for the risk aversion parameters will reduce the role of the asset excess return in portfolio selection and construction. On the other hand, adopting small values may lead to a portfolio where the risk is insufficiently accounted.

#### 4.1.4.2 About the Optimum weights

At a glance it can be observed that optimum wights are changing according to the variation of the return vector. It can be analyzed that the the optimum weight vector is changing significantly when there is perturbation in the return matrix. When the minimum variance line's mean return vector 'm' is replaced by (m + $\triangle$ m) ,the minimum variance line for perturbation model can be obtained. It can be observed that when there is small perturbation that delta m value (according to above notations) should be multiplied by inverse of covariance matrix which is a very large value because covariances are very small decimal numbers. That means when there is a small change in the returns there would be very huge change in the optimum weights.
When the value of perturbation is increased individually for one company only, to drive away at least one company from the optimum weights (to zero the optimum weight of one company) it was observe that the perturbation value should be increased extensively for the SGX data sample. That means the weights are not very much sensitive to perturbations in the market.
But the sensitivity for perturbation is decreasing according to below order for the companies(5IG.SI=BIDU=ZNH,CHL,P30.SI=P34.SI,K75.SI,D01.SI,D05.SI,BN4.SI). If negatively mean companies are removed and perturbation is done for positively mean companies the effort to remove at least one company from the optimum weights is less.

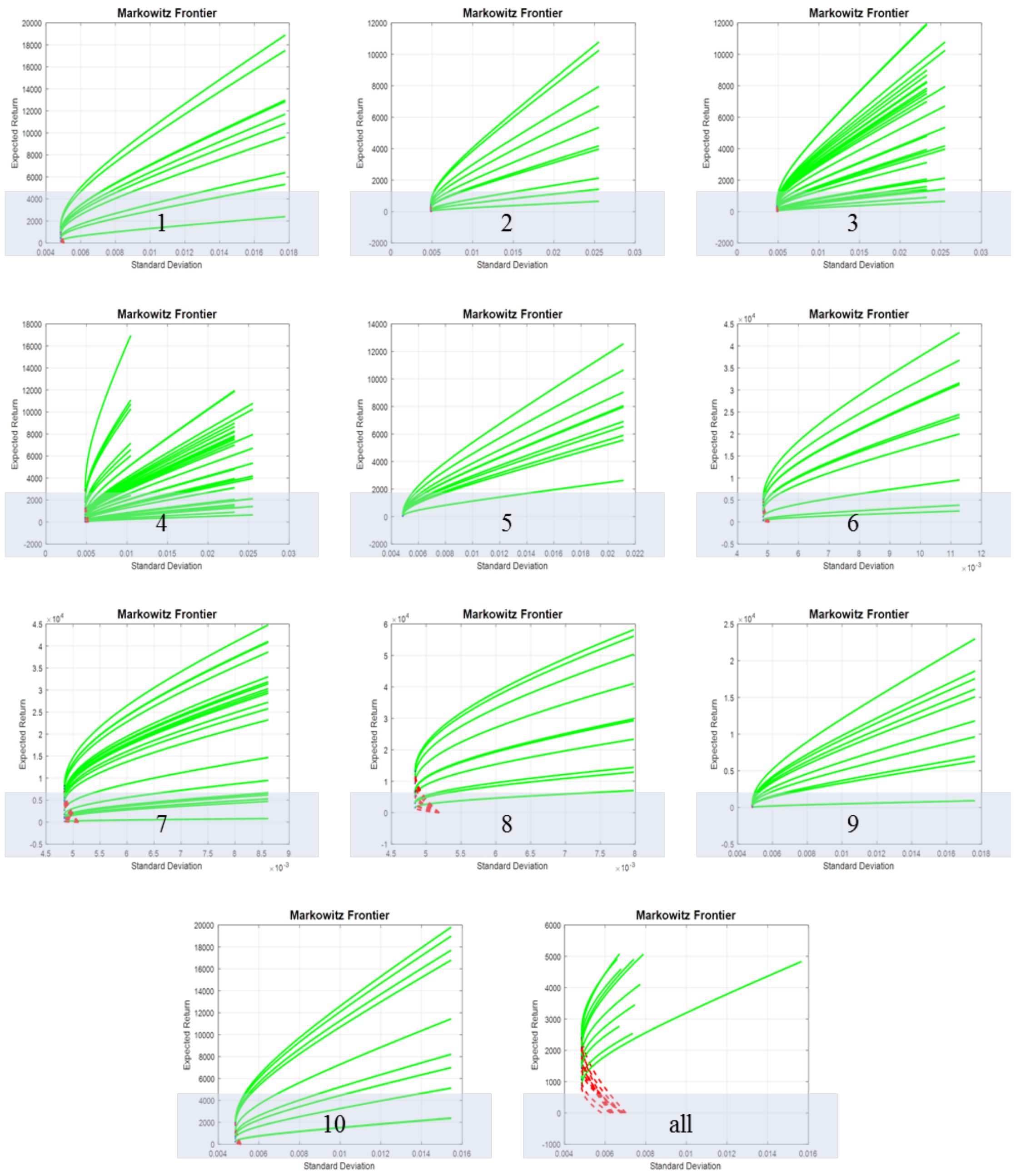


Figure 4.5: Efficient frontiers 1 to10- to chase away one company from the portfolio; Efficient frontier all- When negatively mean companies are removed to chase away one company from the portfolio

| Company of perturbation | Change of mean roughly | Range of expected return |
|---|---|---|
| 1 | 20100 | below 2000 |
| 2 | 12000 | below 1000 |
| 3 | 12100 | below 1000 |
| 4 | 17000 | below 3000 |
| 5 | 13000 | almost 0 compared to others |
| 6 | 45000 | below 5000 |
| 7 | 50000 | below 9000 |
| 8 | 60000 | below 15000 |
| 9 | 25000 | almost 0 compared to others |
| 10 | 20000 | below 2000 |
| All except negative means | 5100 | below 2500 |

Table 4.2: Expected return w.r.t. individual and positively mean company perturbation

By analysing above table the change of mean to chase away one company from the portfolio is less when negatively mean companies are removed from the portfolio as mentioned above. That means it is appropriate to choose positively mean companies when selecting companies for the portfolio formation. In the above table the expected return obtained at the moment of driving away one company from the portfolio is tabulated. Though the the mean is increased extensively companies 5 and 9 show almost less expected return equal to zero when compared with others. That means those companies's expected returns are not very much sensitive to perturbation of the mean of the company itself. Largest expected returns are shown by companies 8,7 and 6 respectively.But the mean should be increased extensively for those companies. With low change of mean compared to companies 1 and 10 company 4 gives higher expected return with respect to change in the mean. But with lower level of change in the mean of the companies the portfolio without negatively mean companies give higher expected return compared to change in the mean of the company individually.

### 4.1.5 Correlation colormap between perturbed values of returns vs optimum weights

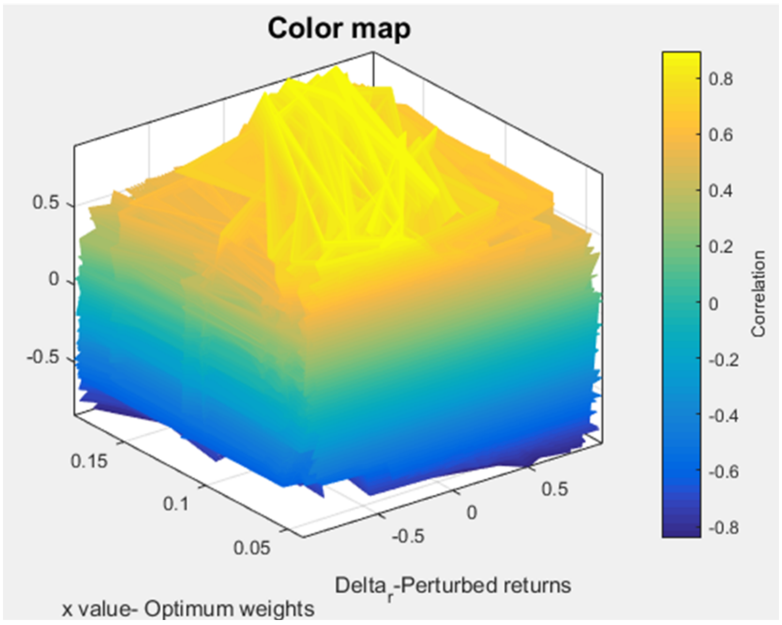


Figure 4.6: Correlation map-3D(Three Dimensional) view

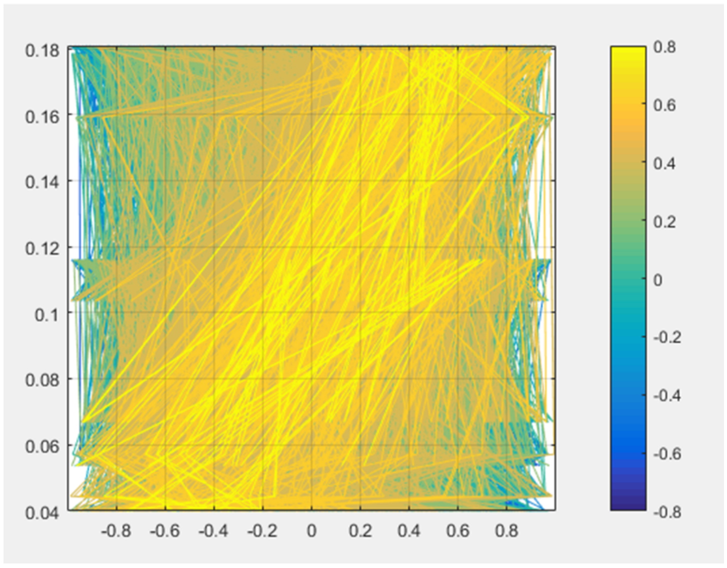


Figure 4.7: Plan view

According to the graph approximately there is equal blue colored part and the greenyellow colored part. That means there are equal negatively correlated values and positively correlated values. Correlations span between -1 and +1.
Observing the plan view graph it can be decide that correlations between perturbed returns and optimum weights are consisted of complex nexuses. By plan view it can be seen that relationships are linear. Some linear relationships show positive slope of optimum weights relative to perturbed return values. Some linear relationships show negative slope of optimum weights relative to perturbed return values. When correlation 3D map was rotated darkest blue lines(related to most negative correlations) and lightest yellow lines (most positive correlations) show narrow 2D pervasion on the graph.(When perturbation of returns is the x axis and optimum weights is the y axis)

## 4.2 Analysis of Neural Network model

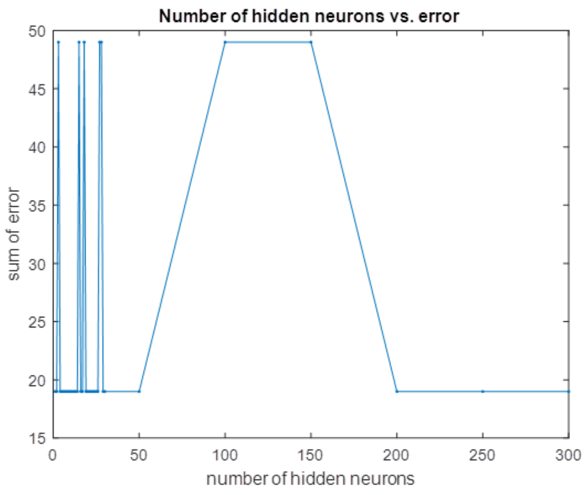


Figure 4.8: Number of hidden neurons vs sum of error(1-300)

Above results were obtained by the code No_of_hidden_neurons (which I found in link[11] but done some small adjustments for error measurement) and the graphs obtained are helpful to imagine the number of hidden neurons roughly. Because of that if it exceeds the number of hidden neurons model will be overfitting and if it is below the desired level it will be underfitting. After running the code several times the graphs showed roughly similar characteristics.So by observing above graph it can roughly get an idea that the hidden number of neurons are roughly with in the range of 8-15. Though error is mimimum, above 200, between 30 and 50, between 18-22 and 24-27 always

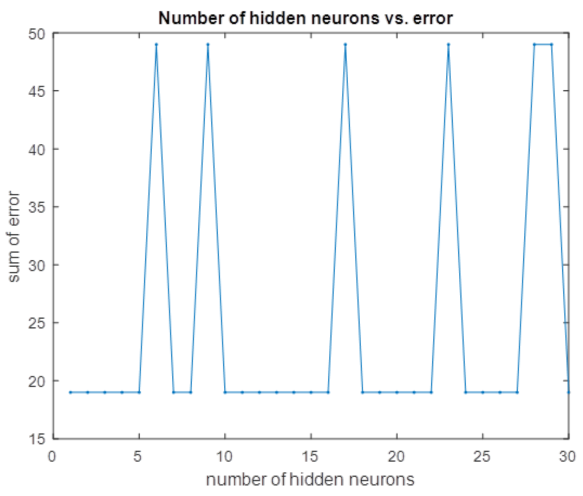


Figure 4.9: Number of hidden neurons vs sum of error(1-30)

minimum number is the best. Due to that checking in between 8-15 is suggested.

### 4.2.1 The Performance graph

Performance is reasonably good when,

- The final mean square error is small
- The test error and the validation set error have similar characteristics
- No significant overfitting has occurred

| No of hidden neurons | MSE of performance graph |
|---|---|
| 8 | 5.8595 e-07 |
| 9 | 4.1894 e-07 |
| 10 | 3.366 e-07 |
| 11 | 3.2053 e-07 |
| 12 | 4.0626 e-07 |
| 13 | 3.4623 e-07 |
| 14 | 4.0415 e-07 |
| 15 | 4.2313 e-07 |

Table 4.3: Minimum MSE of the performance graph

Here in this case there are thousand samples and there is almost no chance to occur overfitting. And almost all the graphs show that the test error and validation set error have the similar characteristics. And only factor to choose from is mean squared error. Among these MSEs of performance graph neurons numbers 10,11 and 13 show lower mse values.

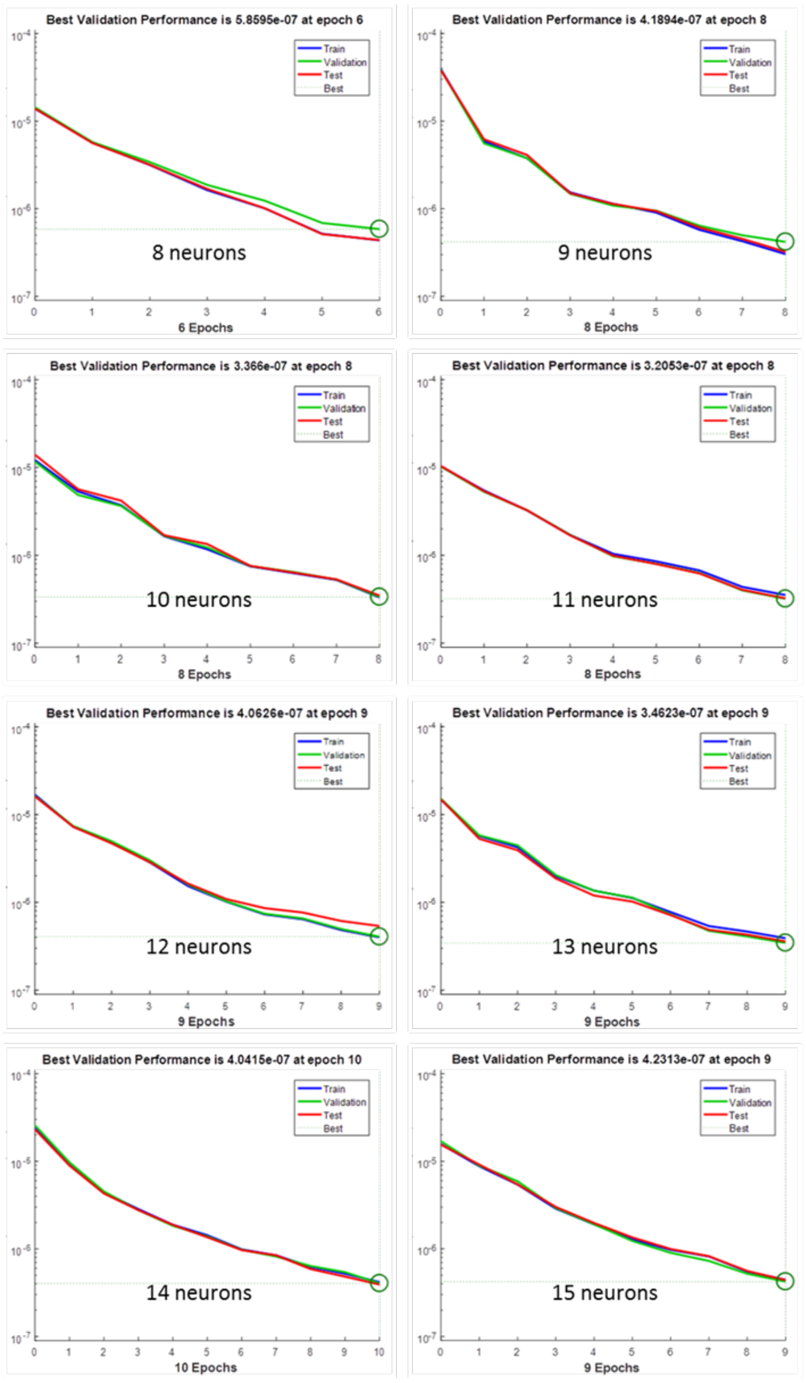


Figure 4.10

Figure 4.11: Performance graphs of each number of hidden neurons

### 4.2.2 Regression graph

For a perfect fit the data should fall along a 45 degree line where the network outputs are equal to the targets. For this problem all the graphs show almost very good fit. Regression R values measure the correlation between out puts and targets. An R value of 1 means a close relationship.Among R values 9,10 and 11,13 show higher R values close to 1.

| Number of neurons | R value |
|---|---|
| 8 | 0.99992 |
| 9 | 0.99995 |
| 10 | 0.99995 |
| 11 | 0.99994 |
| 12 | 0.99993 |
| 13 | 0.99994 |
| 14 | 0.99993 |
| 15 | 0.99993 |

Table 4.4: R values for different no of neurons

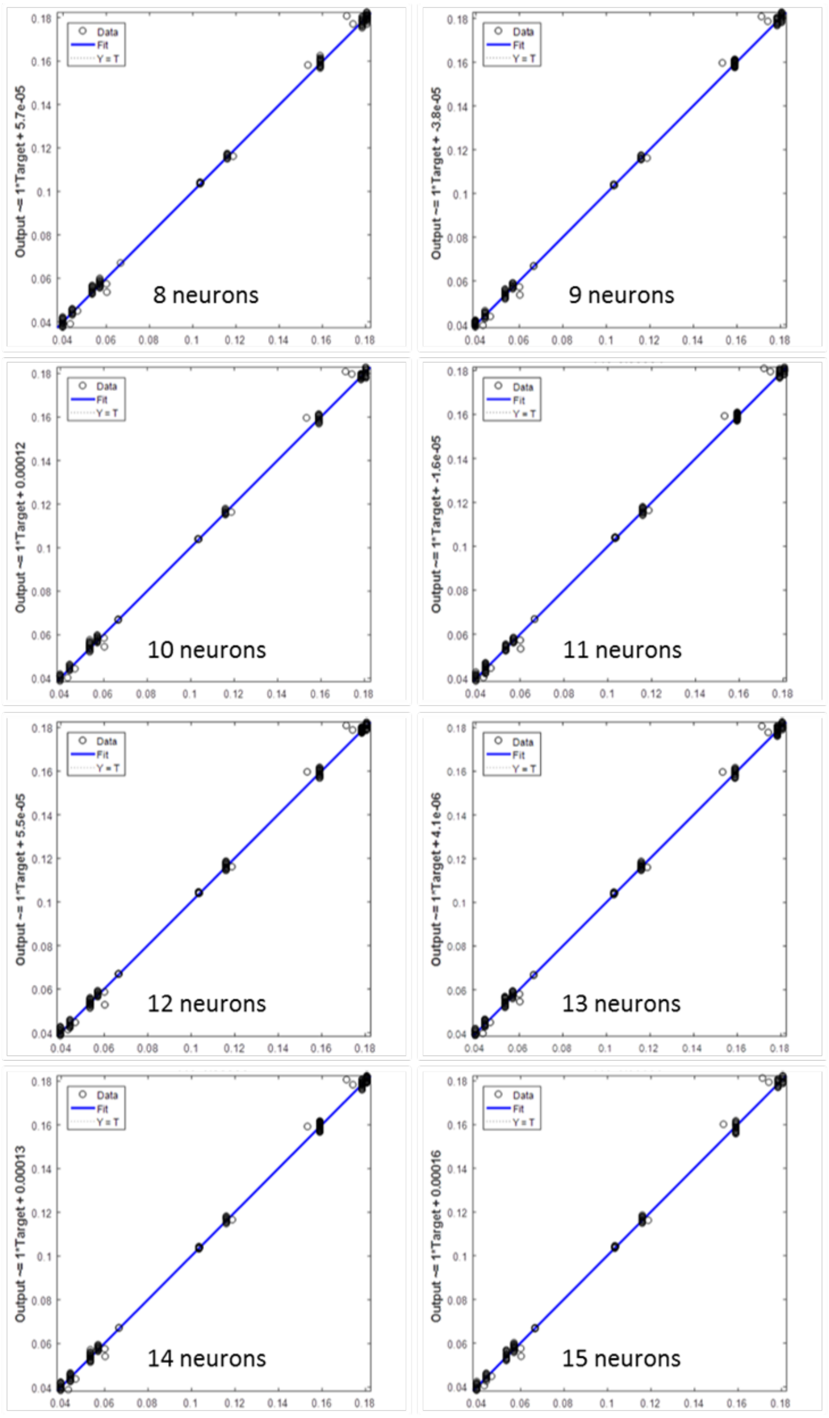


Figure 4.12: Regression graphs of each number of neurons

| Number of neurons | MSE of forecasted sample |
|---|---|
| 8 | 3.773403739866344e-07 |
| 9 | 2.559320260355951e-07 |
| 10 | 3.192714742027959e-07 |
| 11 | 3.773761479409590e-07 |
| 12 | 3.891015587113294e-07 |
| 13 | 2.731106172000422e-07 |
| 14 | 3.230091050640401e-07 |
| 15 | 3.700696130421751e-07 |

Table 4.5: MSE of forecasted samples

According to above table mean squares of error of forecasted samples are minimum in the 9 and 13 neurons. Considering all these factors number of neurons in hidden layer should be 13. The final neural network should be ,

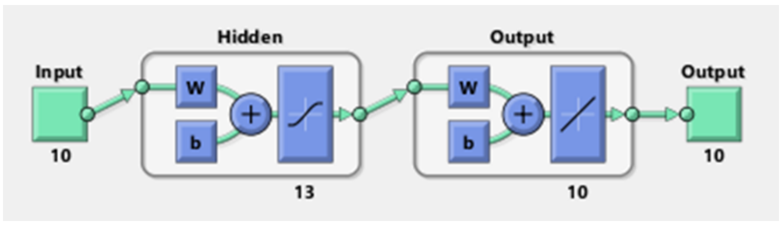


Figure 4.13: Final Neural Network

The relevant Neural Network function and the Advanced Matlab script file is attached in the appendix.The MSE is within $10^{-7}$ range and the calculated optimum values are changing in the $10^{-5}$ range. So the model accurate to about 7 decimal places.

# Chapter 5

# CONCLUSION AND RECOMMENDATIONS

## 5.1 Conclusions

The main supposition out of this entire research is there is significant amount of variation in the optimum weights after portfolio optimization, when there is perturbation in the initial mean returns of the companies. Here in the research methodology mean returns are changed only between -1 to +1 and there was variation in the optimum weights from about fifth decimal place (0.1035..-change after fifth decimal place). The conclusion out of these comments is that the investors must be thoroughly sensitive to the small changes of the returns of the their portfolio companies because small variations of the returns may change the weights or percentages one must invest in order to get higher returns with minimum risk. Another aspect is the variation in the mean returns of the companies is also changing the expected return of the portfolio (Portfolio return). It is wanted to be stressed that the main assumption in the research is there is no short selling.

When the value of perturbation is increased individually for one company only, to drive away at least one company from the optimum weights (to zero the optimum weight of one company) it was observe that the perturbation value should be increased extensively for the SGX data sample. That means the weights are not very much sensitive to perturbations in the market.

If negatively mean companies are removed and perturbation is done for positively mean companies the effort to remove at least one company from the optimum weights is less. Another deduction through this study is when the mean of return is fluctuating there is no significant change in the covariance-variance matrix and the correlation matrix. Covariance-variance matrix and correlation matrix are independent of the perturbation of the mean return.

When the Efficient frontier is considered it can be seen that expected return is changing since mean is perturbing, but the minimum standard deviation doesn't change because the the formula of calculating minimum standard deviation is independent from $\mu$ or the mean of return vector.
According to the color map graph approximately there is equal blue colored part and the greenyellow colored part. That means there are equal negatively correlated values and positively correlated values.

## 5.2 Recommendations

As further proceedings one can observe the behavior of optimum weights when there is short selling incurred. Then the variation of expected return and standard deviation of the portfolio can be closely inspected. Other than that when perturbation value increased over +1 and below -1 the optimum range for getting maximum return can be studied. What happens if optimization is done with another optimization method, will it give same results? That's another path one can look forward. Further the model fitting can be done through a different software and can improve the accuracy.

## References

1. Hill,Robert. A.(2010).*Portfolio theory and Financial Analyses*, pp.10-20

2. https://en.wikipedia.org/wiki/Portfolio_optimization

3. http://www.slideshare.net/bhargavibhanu10/portfolio-analysis-33849624

4. https://en.wikipedia.org/wiki/Modern_portfolio_theory

5. Montgomery,Douglas C.Jennings, Cheryl L.Kulahci, Murat .(2008). *Introduction to Time Series Analysis and Forecasting*. pp 372-374

6. Hadigheh, Alireza. Ghaffari. Romanko,Oleksandr. Terlaky, Tamas. (2004). *Sensitivity Analysis in Convex Quadratic Optimization: Simultaneous perturbation of the objective and right-hand-side vectors*

7. Romanko, Oleksandr (2004). *An Interior Point Approach to Quadratic and Parametric Quadratic Optimization*

8. Best, Michael J. Grauer, Robert. R.(1991) . O*n the Sensitivity of Mean- Variance-Efficient Portfolios to Changes in Asset Means: Some Analytical and Computational Results*

9. Best ,Michael. J.(2010) . *Portfolio Optimization* ,pp.1-36,81-136.

10. Capinski, Marek. and Zastawniak, Tomasz. (2003) . *Mathematics for Finance: An Introduction to Financial Engineering* ,pp.91-117.

11. http://blogs.mathworks.com/loren/2015/08/04/artificial−neural−networks−for-beginners/s_tid=answers_rc2−1_p4

# Appendices

## Appendix 1-Portfolio_Opt

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                          %
%                                                                          %
%             Created by : N.T.Dharmathilaka  (148903X)                    %
%                                                                          %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Finding weights vector using quadprog
R=SGXdata;                     %Calling initial returns of the companies
delta_r=unifrnd(-1,1,1000,10); %Random value matrix as perturbed returns
Ro=zeros(261,10);              %Improving efficiency(Prelocating)

for t=1:length(delta_r(:,1))   %Assigning perturbed values for whole time
                               %points
Ro=delta_r([t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,...
    t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,...
    t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,...
    t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,...
    t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,...
    t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,...
    t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,...
    t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t],:);

i=1:length(R);
j=1:length(R(1,:));
k=1:length(Ro);
m=1:length(Ro(1,:));

Rn= R(i,j) + Ro(k,m);           %Summing up of return matrix and perturbed
                                %return matrix
M=mean(Rn);                     %Mean
CorRn=corr(Rn);                 %Plotting Correlation colormap of Rn
figure(1);
colormap('hot');
imagesc(CorRn);
cb=colorbar;
title('Correlations','fontsize',15)
xlabel('Companies');
ylabel('Companies');
cb=ylabel(cb,'Correlation');

C=cov(Rn);                      %Covariance
H=inv(C);                       %Hessian matrix

A=ones(1,10);
b=1;
Aeq=ones(1,10);
beq=1;
lb=zeros(10,1);
[x,fval,exitflag,output,lambda]=quadprog(H,-0.1*M,A,b,Aeq,beq,lb);%x=weights

delta_rtr=delta_r';             %Transpose of delta_r
Cor=corr(delta_rtr(:,t),x);     %Correlation
Correla(:,t)=Cor(:,1);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%Plotting Efficient Frontier
u=ones(1,10);                    %unit vector
ut=u';                           %transpose of unit vector
A=u*H;
B=A*ut;                          %Calculating weights
w=A/B;
rett=M*w';                       %expected return of minimum variance
portfolio
riskk=sqrt(w*C*w');              %minimum variance


D=M*H;
Mt=M';                           %transpose of M
E=D*Mt;
F=A*Mt;
v=linspace(min(M),max(M),100);
ret=zeros(1,100);
risk=zeros(1,100);
for i=1:100

   W=(E*A-F*D+(B*v(i)*D)-(F*v(i)*A))/(B*E-F*F); % Minimum variance line
    ret(i)=M*W';
    risk(i)=sqrt(W*C*W');

end

m1=(v>rett); % Values more than minimum expected return of portfolio
m2=(v<rett); % Values less than minimum expected return of portfolio
%Markowitz bullet or Efficient frontier
figure(2);
Z=plot(risk(m1),ret(m1),'g-',risk(m2),ret(m2),'r--',riskk,rett,'b.');hold on

title('Markowitz Frontier','fontsize',15);
grid on; xlabel('Standard Deviation'); ylabel('Expected Return');
set(Z(1:2),'linewidth',2);
set(Z(3),'markersize',5);

X(:,t)=x(:,1);
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Writing delta_r and x values in an excel sheet

Input=delta_r;
Output=X';

In=xlswrite('Portfolio_Optin.xlsx',Input);
Out=xlswrite('Portfolio_Optout.xlsx',Output);
T=Correla';
CoR=xlswrite('Correlation.xlsx', T);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Plotting Colormap and correlations maps and matrix plots
```

```
figure(3);

X0=delta_r;
Y0=X';
Z0=[T,T,T,T,T,T,T,T,T,T,T];
contour3(X0,Y0,Z0,1000)
cb=colorbar;
title('Color map','fontsize',15);
xlabel('Delta_r-Perturbed returns');
ylabel('x value- Optimum weights');
cb=ylabel(cb,'Correlation');

figure(4);

plotmatrix(X0,Y0);
title('Matrix plot','fontsize',15);
xlabel('Delta_r-Perturbed returns');
ylabel('x value- Optimum weights');

figure(5);

plotmatrix(Y0,Z0);
title('Matrix plot','fontsize',15);
xlabel('x value- Optimum weights');
ylabel('Correlations');


clear;
```

## Appendix 2-Analysis of Portfolio_Opt

Following is the step by step analysis of the code Portfolio_Opt. There are four parts in it. First part is finding weights vector using quadprog inbuilt function. Second part is plotting the efficient frontiers and third is the writing the output in an excel sheet. Fourth and final part is plotting the colormap of perturbed return values versus optimum weights of portfolio.

### Finding the weights vector using quadprog

**R=SGXdata**

Initially a csv file called SGXdata was prepared using the raw data; that is time to time returns of each of the ten companies. This raw data is called to the m-file using the above command. Then a matrix called R is prepared in the within the m-file itself.

**delta_*r*=unifrnd(-1,1,1000,10)**

Above command is for creating a matrix called delta_*r* for used as perturbed return values. Each row of the matrix consists of ten values between -1 and 1 which are used to run the for loop below one row at a time. There are thousand rows which give thousand different outputs.

**Ro=zeros(261,10)**

This command is for prelocate Ro matrix to improve efficiency because it is large in size.

**t=1:length(delta_*r*(:,1))**

The values taken by t are from one to the length of the matrix delta_*r*.

**Ro=delta_*r*([t,t,t $\cdots$],:)**

There are 261 't's are in this code. For each and every row of delta_*r*, that row value is assigned 261 times to build the matrix Ro.

**i=1:length(R)**

The values taken by i are from one to the length of the matrix R(along a column)

**j=1:length(R(1,:))**

The values taken by j are from one to the length of the matrix R(along a row)

**k=1:length(Ro)**

The values taken by k are from one to the length of the matrix Ro(along a column)

**m=1:length(Ro(1,:))**

The values taken by m are from one to the length of the matrix R(along a row)

**Rn= R(i,j) + Ro(k,m)**

Each and every element in the R matrix is summed up with the relevant element in the marix Ro.

**M=mean(Rn)**

Calculating the mean of the new matrix Rn.

**CorRn=corr(Rn)**

**figure(1)**

**colormap('hot')**

**imagesc(CorRn)**

**cb=colorbar**

**title('Correlations','fontsize',15)**

**xlabel('Companies')**

**ylabel('Companies')**

**cb=ylabel(cb,'Correlation')**

Above code is for plotting the covariance matrix of Rn matrix in a color map.

**C=cov(Rn)**

Calculating the covariance of the matrix Rn.

**H=inv(C)**

Calculating the Hessian matrix or the inverse of the covariance matrix.

**A=ones(1,10)**

**b=1**

**Aeq=ones(1,10)**

**beq=1**

**lb=zeros(10,1)**

**[x,fval,exitflag,output,lambda]=quadprog(H,-0.1*M,A,b,Aeq,beq,lb)**

X = QUADPROG(H,f,A,b) attempts to solve the quadratic programming problem:
$min 0.5 * x' * H * x + f' * x$ subject to: $A * x <= b$

The parameters of the function quadprog are supplied as A,b,Aeq,beq and lb. A and b are inequality conditions and Aeq and beq are equality conditions. lb is lower bound.Equality condition expresses the budget constraint and inequality expresses that there is no short selling.

$[X, FVAL, EXITFLAG, OUTPUT, LAMBDA] = QUADPROG(H, f, A, b)$ returns the set of Lagrangian multipliers LAMBDA, at the solution: LAMBDA.ineqlin for the linear inequalities A, LAMBDA.eqlin for the linear equalities Aeq, LAMBDA.lower for LB

**Plotting the efficient frontiers**

**Calculating Minimum Variance portfolio**

**u=ones(1,10)**

**ut=u'**

A unit vector with one row and ten columns is assigned with u=ones(1,10) and unit vector's transpose is calculated using $u = u'$.

**A=u*H**

**B=A*ut**

**w=A/B**

A vector is calculated by multiplying unit vector with inverse of variance- covariance matrix(Hessian matrix). B vector is calculated by multiplying A vector with transpose of the unit vector. And the weights of the minimum variance portfolio are obtained by dividing A with B.

**rett=M*w'**

**riskk=sqrt(w*c*w')**

he expected return of the minimum variance portfolio is obtained by multiplying mean, M with transpose of the weight vector, w'. And the standard deviation of the minimum variance portfolio is by taking square root of multiplication of weight vector with co-variance matrix and transpose of the weight vector.

**Calculating Minimum Variance Line and plotting Efficient Frontier**

**D=M*H**

**Mt=M'**

**E=D*Mt**

**F=A*Mt**

Solution for the second objective is obtained by assigning D matrix by multiplying mean vector, M with inverse covariance matrix. Then E matrix by multiplying D with transpose of mean ,Mt and F matrix by multiplying A vector with transpose of mean, Mt.

```
linspace(min(M),max(M),100)
```

Targeted expected return ,v assigned by linspace(min(M),max(M),100). Hundred values between minimum of mean, M and maximum of mean M are generated.

```
ret=zeros(1,100)
risk=zeros(1,100)
```

To improve the efficiency for vectors called ret and risk zero matrices ware assigned.

```
for i=1:100;
W=(E*A-F*D+(B*v(i)*D)-(F*v(i)*A))/(B*E-F*F);
ret(i)=m*W’;
risk(i)=sqrt(W*c*W’);
end
```

Then for loop is written from i=1:100 to calculate weights, standard deviations for the targeted expected return values.

```
m1=(v>rett)
m2=(v<rett)
```

Finally m1 is used to denoted values more than the expected return value of the minimum variance portfolio and m2 is used to denoted values less than expected return value of the minimum variance portfolio.

```
Z=plot(risk(m1),ret(m1),’g-’,risk(m2),ret(m2),’r–’,riskk,rett,’b.’); hold on
```

Then it is plotted the Markowitz bullet by variance Standard Deviation vs Expected Return. By the hold on command all the efficient frontiers inside the for loop are plotted.

```
title(’Markowitz Frontier’,’fontsize’,15)
grid on; xlabel(’Standard Deviation’); ylabel(’Expected Return’)
set(Z(1:2),’linewidth’,2)
set(Z(3),’markersize’,5)
```

The following code is used to change the properties of the graph. title is used to give a heading to the graph. Grid on is to mark the grid on the surface. xlabel and ylabel are for naming the x and y axes. Last two codes are used to increase the line width and increase the marker size.

**Writing the output in an excel sheet**

```
delta_rtr=delta_r'
Cor=corr(delta_rtr(:,t),x)
Correla(:,t)=Cor(:,1)
X(:,t)=x(:,1)
```

Transpose of delta_r is denoted as delta_rtr. Correlation between delta_rtr and x vector is calculated using inbuilt function corr. And finally correlation matrix is prepared using the third line. Forth line is used to prepare optimum weights matrix (X).

```
In=xlswrite('Portfolio_Optin.xlsx',Input)
Out=xlswrite('Portfolio_Optout.xlsx',Output)
T=Correla'
CoR=xlswrite('Correlation.xlsx', T)
```

The outputs of delta_r, optimum weights and correlation vector are written on excel sheets using inbuilt function xlswrite.

**Plotting contour3 map**

```
figure(3)
X0=delta_r
Y0=X'
Z0=[T,T,T,T,T,T,T,T,T,T]
contour3(X0,Y0,Z0,1000)
cb=colorbar
title('Color map','fontsize',15)
xlabel('Delta_r-Perturbed returns')
ylabel('x value- Optimum weights')
cb=ylabel(cb,'Correlation')
```

A figure called figure 2 is plotted using delta_r, Optimum weights and Correlation data. It is a 3D plot.

```
figure(4)
plotmatrix(X0,Y0)
title('Matrix plot','fontsize',15)
xlabel('Delta_r-Perturbed returns')
```

**ylabel('x value- Optimum weights')**

This is related to the plotting of matrix plot of perturbed return values vs optimum weights.

**figure(5)**

**plotmatrix(Y0,Z0)**

**title('Matrix plot','fontsize',15)**

**xlabel('x value- Optimum weights')**

**ylabel('Correlations')**

This also related to plotting of matrix plot of optimum weights vs correlations.

## Appendix 3-Optimum weights

| P30.SI | 5IG.SI | BIDU | CHL | ZNH | D01.SI | D05.SI | BN4.SI | K75.SI | P34.SI |
|---|---|---|---|---|---|---|---|---|---|
| 0.103523 | 0.180792 | 0.159123 | 0.06677 | 0.178588 | 0.057143 | 0.039949 | 0.044388 | 0.116098 | 0.053626 |
| 0.103527 | 0.180741 | 0.159135 | 0.06677 | 0.178541 | 0.057141 | 0.039958 | 0.044399 | 0.116133 | 0.053654 |
| 0.103523 | 0.180695 | 0.159186 | 0.066772 | 0.178539 | 0.057141 | 0.039964 | 0.044403 | 0.116157 | 0.053621 |
| 0.103555 | 0.180739 | 0.159157 | 0.066768 | 0.178519 | 0.057137 | 0.039958 | 0.044397 | 0.116144 | 0.053626 |
| 0.103508 | 0.180804 | 0.159103 | 0.066783 | 0.178535 | 0.057152 | 0.039957 | 0.044397 | 0.116123 | 0.053637 |
| 0.103548 | 0.180728 | 0.159106 | 0.066789 | 0.178579 | 0.057148 | 0.039952 | 0.044394 | 0.116126 | 0.053631 |
| 0.103501 | 0.180765 | 0.159128 | 0.066785 | 0.178578 | 0.057129 | 0.039957 | 0.044397 | 0.116121 | 0.05364 |
| 0.103536 | 0.180748 | 0.159172 | 0.066763 | 0.178524 | 0.057127 | 0.039953 | 0.044394 | 0.116151 | 0.05363 |
| 0.103522 | 0.180728 | 0.159178 | 0.066765 | 0.178552 | 0.057132 | 0.039953 | 0.044391 | 0.116122 | 0.053658 |
| 0.103503 | 0.180761 | 0.159196 | 0.066779 | 0.178526 | 0.057148 | 0.039949 | 0.044391 | 0.116138 | 0.053611 |
| 0.103551 | 0.180722 | 0.159181 | 0.066772 | 0.178549 | 0.057136 | 0.039948 | 0.044386 | 0.116099 | 0.053656 |
| 0.103518 | 0.180761 | 0.159159 | 0.066779 | 0.178577 | 0.057133 | 0.039943 | 0.044388 | 0.116129 | 0.053614 |
| 0.103537 | 0.180789 | 0.159137 | 0.066758 | 0.17851 | 0.057143 | 0.039959 | 0.044396 | 0.116116 | 0.053655 |
| 0.103537 | 0.180754 | 0.159161 | 0.066774 | 0.178531 | 0.057134 | 0.039959 | 0.044397 | 0.116138 | 0.053616 |
| 0.103537 | 0.180702 | 0.159162 | 0.066766 | 0.178573 | 0.057129 | 0.039958 | 0.04439 | 0.116129 | 0.053653 |
| 0.103559 | 0.180712 | 0.159161 | 0.066764 | 0.178515 | 0.057148 | 0.039964 | 0.0444 | 0.116125 | 0.053653 |
| 0.103543 | 0.180725 | 0.159174 | 0.066768 | 0.178551 | 0.05714 | 0.039957 | 0.044394 | 0.11613 | 0.053618 |
| 0.103536 | 0.180755 | 0.159106 | 0.066771 | 0.178569 | 0.057131 | 0.039946 | 0.044394 | 0.116145 | 0.053647 |
| 0.103538 | 0.180759 | 0.159175 | 0.066778 | 0.178573 | 0.057124 | 0.039938 | 0.044384 | 0.116099 | 0.053632 |
| 0.103536 | 0.180778 | 0.159109 | 0.066787 | 0.178572 | 0.057148 | 0.039953 | 0.044398 | 0.116108 | 0.053612 |
| 0.103498 | 0.180721 | 0.159134 | 0.066777 | 0.17859 | 0.057149 | 0.039959 | 0.044394 | 0.116131 | 0.053647 |
| 0.103519 | 0.180711 | 0.159205 | 0.066773 | 0.17857 | 0.057141 | 0.039946 | 0.044387 | 0.116106 | 0.053641 |
| 0.103496 | 0.180725 | 0.159148 | 0.066792 | 0.178599 | 0.057143 | 0.039949 | 0.04439 | 0.116135 | 0.053622 |
| 0.103534 | 0.180774 | 0.159157 | 0.066775 | 0.178539 | 0.057135 | 0.039944 | 0.044397 | 0.116119 | 0.053627 |
| 0.103561 | 0.180714 | 0.159112 | 0.06676 | 0.178522 | 0.057147 | 0.039957 | 0.044405 | 0.116155 | 0.053667 |
| 0.103535 | 0.180746 | 0.159174 | 0.066766 | 0.178567 | 0.057145 | 0.039947 | 0.044395 | 0.116095 | 0.053629 |
| 0.103532 | 0.180799 | 0.159165 | 0.066762 | 0.178556 | 0.057131 | 0.03995 | 0.04439 | 0.116106 | 0.05361 |
| 0.103498 | 0.180801 | 0.159109 | 0.066778 | 0.17855 | 0.057146 | 0.03995 | 0.044392 | 0.116132 | 0.053644 |
| 0.103512 | 0.180759 | 0.159129 | 0.06678 | 0.178543 | 0.057128 | 0.039947 | 0.044392 | 0.116155 | 0.053654 |
| 0.103532 | 0.180795 | 0.159129 | 0.066763 | 0.178514 | 0.057141 | 0.039956 | 0.0444 | 0.116133 | 0.053637 |
| 0.103519 | 0.180791 | 0.159106 | 0.066781 | 0.178555 | 0.057141 | 0.039958 | 0.044402 | 0.116121 | 0.053626 |
| 0.103542 | 0.180782 | 0.15912 | 0.06676 | 0.178541 | 0.057151 | 0.039947 | 0.044398 | 0.116102 | 0.053657 |
| 0.10354 | 0.18077 | 0.15914 | 0.066762 | 0.178524 | 0.057136 | 0.039946 | 0.044398 | 0.11613 | 0.053654 |
| 0.103509 | 0.180721 | 0.159149 | 0.066792 | 0.178612 | 0.057154 | 0.039949 | 0.044388 | 0.116098 | 0.053629 |
| 0.103514 | 0.180741 | 0.159165 | 0.066776 | 0.178557 | 0.057132 | 0.039948 | 0.044387 | 0.116141 | 0.053638 |
| 0.103536 | 0.180771 | 0.159145 | 0.066781 | 0.178556 | 0.057133 | 0.039948 | 0.04439 | 0.116127 | 0.053614 |
| 0.10352 | 0.180775 | 0.159122 | 0.066784 | 0.178561 | 0.057145 | 0.039948 | 0.04439 | 0.116141 | 0.053613 |
| 0.103508 | 0.18069 | 0.159197 | 0.066776 | 0.178566 | 0.057141 | 0.039956 | 0.044398 | 0.116154 | 0.053616 |
| 0.103494 | 0.18072 | 0.159132 | 0.066796 | 0.178618 | 0.057143 | 0.039954 | 0.044392 | 0.116139 | 0.053614 |
| 0.103492 | 0.180813 | 0.1591 | 0.066775 | 0.178579 | 0.057142 | 0.039951 | 0.044396 | 0.116114 | 0.053637 |
| 0.103499 | 0.180777 | 0.159152 | 0.066787 | 0.178573 | 0.057129 | 0.039947 | 0.044388 | 0.116135 | 0.053614 |
| 0.103512 | 0.180749 | 0.159178 | 0.066785 | 0.178583 | 0.05713 | 0.039951 | 0.044391 | 0.116121 | 0.053602 |
| 0.103528 | 0.180777 | 0.159146 | 0.066773 | 0.178512 | 0.057131 | 0.03995 | 0.044394 | 0.116142 | 0.053646 |
| 0.103542 | 0.18077 | 0.159132 | 0.066774 | 0.178547 | 0.057138 | 0.039949 | 0.044396 | 0.116125 | 0.053628 |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 0.103525 | 0.180709 | 0.159082 | 0.06679 | 0.178602 | 0.05715 | 0.039954 | 0.04439 | 0.116147 | 0.053651 |
| 0.103547 | 0.180763 | 0.159107 | 0.06677 | 0.178547 | 0.057146 | 0.039947 | 0.044392 | 0.116147 | 0.053634 |
| 0.103509 | 0.180708 | 0.159191 | 0.066762 | 0.178526 | 0.057148 | 0.03996 | 0.044394 | 0.116134 | 0.053668 |
| 0.103531 | 0.180756 | 0.159159 | 0.066764 | 0.178559 | 0.057151 | 0.039946 | 0.044387 | 0.116102 | 0.053646 |
| 0.103545 | 0.180738 | 0.159166 | 0.066768 | 0.178521 | 0.057148 | 0.039953 | 0.044397 | 0.116114 | 0.05365 |
| 0.10351 | 0.1808 | 0.159181 | 0.066757 | 0.178543 | 0.057142 | 0.039943 | 0.044389 | 0.116091 | 0.053645 |
| 0.103559 | 0.180719 | 0.159183 | 0.066769 | 0.178528 | 0.05715 | 0.039959 | 0.044396 | 0.116108 | 0.053628 |
| 0.10355 | 0.180811 | 0.159107 | 0.066778 | 0.178524 | 0.057133 | 0.039944 | 0.04439 | 0.11615 | 0.053612 |
| 0.10349 | 0.180791 | 0.159156 | 0.066783 | 0.178579 | 0.057142 | 0.039943 | 0.04439 | 0.116095 | 0.053632 |
| 0.103542 | 0.18074 | 0.159117 | 0.066772 | 0.178537 | 0.057132 | 0.039954 | 0.044396 | 0.116154 | 0.053655 |
| 0.103508 | 0.180772 | 0.159156 | 0.066767 | 0.178551 | 0.057136 | 0.03995 | 0.044396 | 0.116144 | 0.05362 |
| 0.103502 | 0.180763 | 0.159161 | 0.066779 | 0.178571 | 0.057139 | 0.039954 | 0.044389 | 0.116111 | 0.053629 |
| 0.103526 | 0.180778 | 0.159121 | 0.066769 | 0.17853 | 0.057137 | 0.039954 | 0.044394 | 0.116144 | 0.053648 |
| 0.103508 | 0.180714 | 0.159176 | 0.066771 | 0.178576 | 0.057148 | 0.039946 | 0.044391 | 0.116134 | 0.053637 |
| 0.103513 | 0.180722 | 0.159152 | 0.066785 | 0.178557 | 0.057139 | 0.039959 | 0.044399 | 0.116137 | 0.053637 |
| 0.103515 | 0.180707 | 0.159175 | 0.06678 | 0.178611 | 0.057142 | 0.039949 | 0.044395 | 0.116111 | 0.053616 |
| 0.103504 | 0.180724 | 0.159123 | 0.066781 | 0.17856 | 0.057147 | 0.039956 | 0.044397 | 0.116152 | 0.053656 |
| 0.103536 | 0.180712 | 0.159167 | 0.066775 | 0.178586 | 0.057137 | 0.039943 | 0.044387 | 0.116122 | 0.053634 |
| 0.10355 | 0.180742 | 0.159141 | 0.066767 | 0.178521 | 0.05714 | 0.039955 | 0.044397 | 0.116149 | 0.053637 |
| 0.103505 | 0.180788 | 0.159142 | 0.066762 | 0.17852 | 0.057144 | 0.039953 | 0.044392 | 0.116149 | 0.053645 |
| 0.103523 | 0.180719 | 0.159212 | 0.06676 | 0.178523 | 0.05713 | 0.039954 | 0.044397 | 0.116137 | 0.053645 |
| 0.103533 | 0.180799 | 0.15915 | 0.066777 | 0.178522 | 0.057135 | 0.039948 | 0.044393 | 0.116128 | 0.053615 |
| 0.103545 | 0.180812 | 0.159114 | 0.06676 | 0.178523 | 0.057152 | 0.039957 | 0.044398 | 0.116121 | 0.053617 |
| 0.103524 | 0.180758 | 0.159156 | 0.06678 | 0.178567 | 0.057135 | 0.039946 | 0.044394 | 0.116128 | 0.053614 |
| 0.103522 | 0.180741 | 0.159121 | 0.066778 | 0.178574 | 0.05715 | 0.039946 | 0.044397 | 0.116151 | 0.05362 |
| 0.103482 | 0.180759 | 0.15911 | 0.06678 | 0.178593 | 0.057153 | 0.039957 | 0.044401 | 0.116121 | 0.053644 |
| 0.103536 | 0.180756 | 0.15914 | 0.066763 | 0.178522 | 0.057149 | 0.039955 | 0.044399 | 0.116117 | 0.053664 |
| 0.103557 | 0.18075 | 0.159199 | 0.066755 | 0.178508 | 0.057138 | 0.039945 | 0.044392 | 0.116133 | 0.053623 |
| 0.103492 | 0.180689 | 0.159146 | 0.066791 | 0.178617 | 0.057153 | 0.039957 | 0.044394 | 0.116111 | 0.05365 |
| 0.103512 | 0.180814 | 0.159098 | 0.066773 | 0.178559 | 0.05713 | 0.03995 | 0.044395 | 0.116123 | 0.053647 |
| 0.103509 | 0.180703 | 0.159129 | 0.06679 | 0.178607 | 0.057153 | 0.039957 | 0.044393 | 0.116116 | 0.053643 |
| 0.103517 | 0.18074 | 0.159163 | 0.066775 | 0.178533 | 0.057128 | 0.039955 | 0.044393 | 0.116143 | 0.053651 |
| 0.10352 | 0.180753 | 0.159153 | 0.066777 | 0.178536 | 0.057132 | 0.039954 | 0.044392 | 0.116136 | 0.053649 |
| 0.103553 | 0.18076 | 0.159127 | 0.066773 | 0.178554 | 0.057132 | 0.039944 | 0.044392 | 0.116128 | 0.053636 |
| 0.103496 | 0.180814 | 0.159108 | 0.066784 | 0.178553 | 0.057139 | 0.039954 | 0.044395 | 0.116141 | 0.053614 |
| 0.103505 | 0.180812 | 0.159124 | 0.066776 | 0.178598 | 0.05713 | 0.03995 | 0.044388 | 0.116103 | 0.053615 |
| 0.103555 | 0.180691 | 0.159154 | 0.066757 | 0.178515 | 0.05715 | 0.039957 | 0.044396 | 0.11616 | 0.053663 |
| 0.103533 | 0.180796 | 0.159102 | 0.066769 | 0.17853 | 0.057139 | 0.039946 | 0.044391 | 0.116145 | 0.053649 |
| 0.103521 | 0.180757 | 0.159139 | 0.066761 | 0.178519 | 0.057134 | 0.039963 | 0.0444 | 0.116143 | 0.053662 |
| 0.103504 | 0.180725 | 0.15912 | 0.066789 | 0.178601 | 0.057135 | 0.039957 | 0.044391 | 0.116126 | 0.053652 |
| 0.103526 | 0.18079 | 0.159121 | 0.066763 | 0.178565 | 0.05713 | 0.039954 | 0.044394 | 0.116108 | 0.05365 |
| 0.10354 | 0.180801 | 0.159124 | 0.066758 | 0.178514 | 0.057131 | 0.03995 | 0.044394 | 0.116151 | 0.053638 |
| 0.103526 | 0.180792 | 0.159155 | 0.066767 | 0.178547 | 0.057134 | 0.039952 | 0.044394 | 0.116116 | 0.053618 |
| 0.103559 | 0.180728 | 0.159126 | 0.066769 | 0.178537 | 0.057145 | 0.039962 | 0.044402 | 0.116138 | 0.053634 |
| 0.103489 | 0.180785 | 0.159164 | 0.066782 | 0.178548 | 0.057131 | 0.039951 | 0.044388 | 0.116135 | 0.053626 |
| 0.103518 | 0.180728 | 0.159129 | 0.066785 | 0.178602 | 0.057141 | 0.039952 | 0.044394 | 0.116139 | 0.053612 |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 0.103497 | 0.180748 | 0.159132 | 0.066782 | 0.178572 | 0.057137 | 0.039948 | 0.044392 | 0.116158 | 0.053634 |
| 0.103537 | 0.180788 | 0.159162 | 0.066758 | 0.17851 | 0.057137 | 0.039948 | 0.044388 | 0.116129 | 0.053643 |
| 0.10354 | 0.180743 | 0.159106 | 0.066782 | 0.178583 | 0.057156 | 0.03995 | 0.044395 | 0.116112 | 0.053634 |
| 0.103531 | 0.180744 | 0.159182 | 0.066768 | 0.178531 | 0.057146 | 0.039957 | 0.044391 | 0.116108 | 0.053642 |
| 0.103502 | 0.180717 | 0.159178 | 0.066786 | 0.178546 | 0.057142 | 0.039961 | 0.0444 | 0.116122 | 0.053646 |
| 0.103493 | 0.180772 | 0.159189 | 0.06677 | 0.178535 | 0.057141 | 0.039955 | 0.044388 | 0.116099 | 0.053659 |
| 0.103551 | 0.180784 | 0.159105 | 0.066785 | 0.178566 | 0.057136 | 0.039943 | 0.044388 | 0.116121 | 0.053621 |
| 0.103508 | 0.180726 | 0.159167 | 0.066771 | 0.178567 | 0.057139 | 0.039961 | 0.044401 | 0.116143 | 0.053616 |
| 0.103542 | 0.180724 | 0.159168 | 0.066767 | 0.178544 | 0.057137 | 0.039952 | 0.044396 | 0.116119 | 0.05365 |
| 0.103517 | 0.180783 | 0.159186 | 0.066753 | 0.178522 | 0.057144 | 0.039952 | 0.04439 | 0.116096 | 0.053656 |
| 0.10352 | 0.180816 | 0.159154 | 0.06677 | 0.178532 | 0.057135 | 0.039953 | 0.044393 | 0.116099 | 0.053628 |
| 0.10349 | 0.180757 | 0.159165 | 0.066785 | 0.178586 | 0.05714 | 0.039951 | 0.044392 | 0.116124 | 0.053609 |
| 0.10355 | 0.180716 | 0.159116 | 0.066785 | 0.178558 | 0.057153 | 0.039949 | 0.044394 | 0.116155 | 0.053624 |
| 0.103531 | 0.180764 | 0.159191 | 0.066762 | 0.178536 | 0.057133 | 0.039953 | 0.044399 | 0.116109 | 0.053624 |
| 0.103501 | 0.180754 | 0.159154 | 0.066782 | 0.178589 | 0.057131 | 0.039955 | 0.04439 | 0.116109 | 0.053635 |
| 0.103518 | 0.180753 | 0.159183 | 0.066759 | 0.178514 | 0.057149 | 0.039949 | 0.044399 | 0.116126 | 0.053651 |
| 0.103528 | 0.180719 | 0.159141 | 0.066781 | 0.178588 | 0.057157 | 0.039949 | 0.044397 | 0.116122 | 0.053618 |
| 0.103524 | 0.18073 | 0.159154 | 0.06678 | 0.178549 | 0.057135 | 0.03995 | 0.044396 | 0.116143 | 0.05364 |
| 0.103535 | 0.180808 | 0.159139 | 0.066759 | 0.17852 | 0.057131 | 0.03995 | 0.044395 | 0.116143 | 0.05362 |
| 0.103527 | 0.180782 | 0.159105 | 0.066774 | 0.178577 | 0.057147 | 0.039944 | 0.044396 | 0.116131 | 0.053618 |
| 0.103485 | 0.180787 | 0.159161 | 0.066768 | 0.178543 | 0.057141 | 0.039949 | 0.04439 | 0.116138 | 0.053639 |
| 0.103501 | 0.18075 | 0.159108 | 0.066791 | 0.178593 | 0.057142 | 0.039952 | 0.044393 | 0.116143 | 0.053626 |
| 0.103523 | 0.180716 | 0.159154 | 0.06678 | 0.178592 | 0.057139 | 0.039955 | 0.044393 | 0.116132 | 0.053617 |
| 0.103537 | 0.180768 | 0.159114 | 0.066774 | 0.178555 | 0.057132 | 0.039947 | 0.044389 | 0.116153 | 0.053629 |
| 0.103552 | 0.180742 | 0.159205 | 0.066766 | 0.178543 | 0.057134 | 0.039947 | 0.044389 | 0.116103 | 0.05362 |
| 0.103531 | 0.180744 | 0.159153 | 0.06678 | 0.178541 | 0.057154 | 0.039957 | 0.044401 | 0.116109 | 0.053629 |
| 0.103509 | 0.1808 | 0.159096 | 0.066783 | 0.178564 | 0.057128 | 0.039952 | 0.044397 | 0.116154 | 0.053617 |
| 0.103544 | 0.180782 | 0.159175 | 0.066752 | 0.178512 | 0.057138 | 0.039945 | 0.044391 | 0.116113 | 0.053647 |
| 0.103504 | 0.18074 | 0.159097 | 0.066793 | 0.178613 | 0.057144 | 0.039951 | 0.044395 | 0.116139 | 0.053623 |
| 0.103507 | 0.180807 | 0.159125 | 0.066764 | 0.178534 | 0.057135 | 0.039951 | 0.044397 | 0.116132 | 0.053649 |
| 0.103534 | 0.18072 | 0.159115 | 0.066791 | 0.178587 | 0.05714 | 0.039949 | 0.044393 | 0.116157 | 0.053615 |
| 0.103533 | 0.180703 | 0.159164 | 0.066765 | 0.178571 | 0.057144 | 0.039957 | 0.044398 | 0.116138 | 0.053628 |
| 0.103547 | 0.180726 | 0.159136 | 0.066769 | 0.178555 | 0.057153 | 0.039953 | 0.044393 | 0.116115 | 0.053653 |
| 0.103541 | 0.1807 | 0.159107 | 0.066787 | 0.178584 | 0.057148 | 0.039959 | 0.0444 | 0.116134 | 0.05364 |
| 0.103545 | 0.180786 | 0.159171 | 0.06675 | 0.17852 | 0.057144 | 0.039947 | 0.044385 | 0.116106 | 0.053647 |
| 0.103512 | 0.18075 | 0.159114 | 0.066776 | 0.178584 | 0.05715 | 0.039955 | 0.04439 | 0.116131 | 0.053636 |
| 0.103555 | 0.180719 | 0.159188 | 0.066766 | 0.178568 | 0.057139 | 0.039946 | 0.044385 | 0.116097 | 0.053636 |
| 0.103546 | 0.180729 | 0.159175 | 0.066774 | 0.178562 | 0.05713 | 0.039944 | 0.044385 | 0.116132 | 0.053624 |
| 0.103527 | 0.180793 | 0.159173 | 0.066763 | 0.178544 | 0.057129 | 0.039951 | 0.044389 | 0.116098 | 0.053632 |
| 0.103547 | 0.180739 | 0.15916 | 0.066763 | 0.178525 | 0.05715 | 0.039959 | 0.044395 | 0.116105 | 0.053655 |
| 0.103549 | 0.180708 | 0.159189 | 0.06678 | 0.178557 | 0.057135 | 0.039943 | 0.044385 | 0.11612 | 0.053634 |
| 0.103542 | 0.180767 | 0.159098 | 0.066767 | 0.178569 | 0.057146 | 0.039955 | 0.044396 | 0.116142 | 0.053618 |
| 0.103507 | 0.180771 | 0.159085 | 0.066779 | 0.178564 | 0.057141 | 0.039957 | 0.044398 | 0.116141 | 0.053657 |
| 0.103536 | 0.180773 | 0.159094 | 0.066769 | 0.178541 | 0.05714 | 0.039955 | 0.044401 | 0.116156 | 0.053636 |
| 0.103546 | 0.180765 | 0.159128 | 0.06678 | 0.178558 | 0.057134 | 0.039949 | 0.044395 | 0.116112 | 0.053632 |
| 0.103541 | 0.180784 | 0.15911 | 0.066761 | 0.17851 | 0.057131 | 0.039956 | 0.044403 | 0.116155 | 0.053649 |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 0.103547 | 0.18074 | 0.159196 | 0.066767 | 0.17856 | 0.057125 | 0.039943 | 0.044384 | 0.116102 | 0.053635 |
| 0.103522 | 0.180737 | 0.159145 | 0.066783 | 0.17855 | 0.057152 | 0.039959 | 0.044392 | 0.116109 | 0.053651 |
| 0.103544 | 0.180678 | 0.159171 | 0.066786 | 0.178571 | 0.057135 | 0.039948 | 0.044397 | 0.116155 | 0.053614 |
| 0.103553 | 0.180759 | 0.15917 | 0.066773 | 0.178524 | 0.057143 | 0.03995 | 0.044397 | 0.116105 | 0.053625 |
| 0.103517 | 0.180712 | 0.159181 | 0.066789 | 0.17858 | 0.057133 | 0.039956 | 0.044392 | 0.116114 | 0.053627 |
| 0.103505 | 0.180713 | 0.159193 | 0.066769 | 0.178584 | 0.057144 | 0.039954 | 0.044388 | 0.116095 | 0.053654 |
| 0.103548 | 0.180721 | 0.159142 | 0.066773 | 0.178575 | 0.057134 | 0.039946 | 0.044388 | 0.11611 | 0.053662 |
| 0.103529 | 0.180756 | 0.159121 | 0.066787 | 0.178581 | 0.057134 | 0.039943 | 0.044387 | 0.116139 | 0.053623 |
| 0.103555 | 0.180703 | 0.159117 | 0.066779 | 0.178566 | 0.057145 | 0.039957 | 0.0444 | 0.116113 | 0.053666 |
| 0.103521 | 0.180741 | 0.159148 | 0.066778 | 0.178534 | 0.057144 | 0.039951 | 0.044398 | 0.116149 | 0.053636 |
| 0.103509 | 0.180698 | 0.159155 | 0.066781 | 0.17861 | 0.057154 | 0.039954 | 0.044399 | 0.116115 | 0.053623 |
| 0.103527 | 0.18073 | 0.159154 | 0.066775 | 0.17859 | 0.057146 | 0.039948 | 0.044389 | 0.116112 | 0.053628 |
| 0.103554 | 0.180711 | 0.159166 | 0.066774 | 0.178574 | 0.057134 | 0.039944 | 0.044391 | 0.116136 | 0.053615 |
| 0.103526 | 0.180751 | 0.159133 | 0.066764 | 0.178557 | 0.057135 | 0.03995 | 0.044395 | 0.116146 | 0.053642 |
| 0.103503 | 0.180787 | 0.1591 | 0.06678 | 0.178574 | 0.057152 | 0.03995 | 0.044396 | 0.116111 | 0.053647 |
| 0.103534 | 0.180706 | 0.159183 | 0.066765 | 0.178515 | 0.057136 | 0.039956 | 0.044402 | 0.116164 | 0.053638 |
| 0.10352 | 0.180709 | 0.159195 | 0.066761 | 0.178542 | 0.057131 | 0.03996 | 0.044396 | 0.116141 | 0.053644 |
| 0.103496 | 0.18073 | 0.159113 | 0.066781 | 0.178605 | 0.057154 | 0.039957 | 0.044402 | 0.116108 | 0.053654 |
| 0.103541 | 0.180747 | 0.159198 | 0.066761 | 0.178531 | 0.057136 | 0.039949 | 0.044398 | 0.116115 | 0.053624 |
| 0.103546 | 0.180734 | 0.159189 | 0.066766 | 0.178574 | 0.057135 | 0.039946 | 0.044395 | 0.116103 | 0.053613 |
| 0.103538 | 0.180731 | 0.159129 | 0.066781 | 0.178557 | 0.057137 | 0.039961 | 0.044396 | 0.116124 | 0.053647 |
| 0.103509 | 0.180734 | 0.159137 | 0.06676 | 0.17854 | 0.057143 | 0.039959 | 0.044403 | 0.116152 | 0.053663 |
| 0.103542 | 0.180766 | 0.159168 | 0.066772 | 0.178536 | 0.057135 | 0.03995 | 0.044386 | 0.116096 | 0.053648 |
| 0.103546 | 0.180753 | 0.159174 | 0.06676 | 0.178523 | 0.057129 | 0.039953 | 0.044399 | 0.116152 | 0.05361 |
| 0.103533 | 0.180718 | 0.159172 | 0.066774 | 0.17859 | 0.057143 | 0.039944 | 0.044384 | 0.116092 | 0.05365 |
| 0.103531 | 0.180774 | 0.159152 | 0.066754 | 0.178503 | 0.057142 | 0.039961 | 0.044401 | 0.116124 | 0.053659 |
| 0.103526 | 0.180777 | 0.159154 | 0.066776 | 0.178578 | 0.057133 | 0.039941 | 0.044391 | 0.116107 | 0.053617 |
| 0.103512 | 0.180711 | 0.159133 | 0.066785 | 0.178561 | 0.057142 | 0.039956 | 0.044398 | 0.116139 | 0.053663 |
| 0.103526 | 0.180742 | 0.159153 | 0.066781 | 0.178566 | 0.057134 | 0.039952 | 0.044393 | 0.116132 | 0.053621 |
| 0.103544 | 0.180799 | 0.15909 | 0.066779 | 0.17854 | 0.057141 | 0.039956 | 0.044395 | 0.116118 | 0.053639 |
| 0.103516 | 0.180746 | 0.15916 | 0.066774 | 0.178575 | 0.057126 | 0.039948 | 0.044392 | 0.116134 | 0.053629 |
| 0.103528 | 0.180749 | 0.159175 | 0.066772 | 0.178581 | 0.05714 | 0.039947 | 0.044391 | 0.116088 | 0.053628 |
| 0.103519 | 0.180777 | 0.159178 | 0.066779 | 0.178558 | 0.057126 | 0.039949 | 0.044391 | 0.116108 | 0.053616 |
| 0.103545 | 0.18073 | 0.159109 | 0.066786 | 0.178531 | 0.057152 | 0.039961 | 0.044397 | 0.116169 | 0.05362 |
| 0.103504 | 0.180765 | 0.15918 | 0.066778 | 0.178552 | 0.057131 | 0.039947 | 0.04439 | 0.11614 | 0.053612 |
| 0.10351 | 0.180744 | 0.159117 | 0.066778 | 0.178602 | 0.057139 | 0.039954 | 0.044397 | 0.116106 | 0.053653 |
| 0.103494 | 0.180745 | 0.159169 | 0.066768 | 0.178547 | 0.057139 | 0.039955 | 0.044395 | 0.116146 | 0.053641 |
| 0.103536 | 0.180768 | 0.15915 | 0.066761 | 0.178523 | 0.057142 | 0.039955 | 0.044391 | 0.116122 | 0.053654 |
| 0.103549 | 0.180749 | 0.159107 | 0.06677 | 0.178544 | 0.057152 | 0.039958 | 0.044402 | 0.11614 | 0.053631 |
| 0.103548 | 0.180731 | 0.159121 | 0.06678 | 0.178529 | 0.05715 | 0.039962 | 0.044404 | 0.116158 | 0.053618 |
| 0.103535 | 0.180798 | 0.159149 | 0.066762 | 0.178529 | 0.057142 | 0.039941 | 0.044389 | 0.116122 | 0.053632 |
| 0.103503 | 0.180746 | 0.159168 | 0.066786 | 0.178544 | 0.057141 | 0.039947 | 0.044391 | 0.116156 | 0.053617 |
| 0.103522 | 0.18076 | 0.159096 | 0.066781 | 0.178545 | 0.057152 | 0.039951 | 0.0444 | 0.116136 | 0.053658 |
| 0.103544 | 0.180768 | 0.159116 | 0.066778 | 0.178554 | 0.057142 | 0.039946 | 0.044392 | 0.116138 | 0.053623 |
| 0.103545 | 0.180777 | 0.159125 | 0.066779 | 0.178528 | 0.057146 | 0.039956 | 0.044394 | 0.116126 | 0.053623 |
| 0.10352 | 0.180742 | 0.159167 | 0.066762 | 0.178506 | 0.057144 | 0.039958 | 0.044402 | 0.11616 | 0.053637 |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 0.103507 | 0.1807 | 0.159167 | 0.066785 | 0.178586 | 0.057144 | 0.03995 | 0.044397 | 0.116146 | 0.053618 |
| 0.103506 | 0.180787 | 0.159166 | 0.066775 | 0.178532 | 0.057152 | 0.039959 | 0.044394 | 0.116111 | 0.053617 |
| 0.103526 | 0.180781 | 0.159165 | 0.066754 | 0.178508 | 0.057145 | 0.039951 | 0.044397 | 0.116123 | 0.05365 |
| 0.103519 | 0.18072 | 0.159131 | 0.066779 | 0.178575 | 0.057135 | 0.039956 | 0.04439 | 0.116144 | 0.053651 |
| 0.103493 | 0.180764 | 0.159189 | 0.066759 | 0.178537 | 0.057128 | 0.039951 | 0.044396 | 0.116126 | 0.053656 |
| 0.1035 | 0.180778 | 0.159095 | 0.066783 | 0.178583 | 0.057145 | 0.039947 | 0.044396 | 0.11613 | 0.053643 |
| 0.103514 | 0.180735 | 0.159156 | 0.066783 | 0.178559 | 0.057143 | 0.039954 | 0.044393 | 0.116138 | 0.053624 |
| 0.103545 | 0.180773 | 0.159166 | 0.066777 | 0.178556 | 0.057131 | 0.039947 | 0.044387 | 0.116106 | 0.053612 |
| 0.103547 | 0.180753 | 0.159131 | 0.066762 | 0.178541 | 0.05714 | 0.039949 | 0.04439 | 0.116138 | 0.05365 |
| 0.103536 | 0.180711 | 0.159131 | 0.066783 | 0.178533 | 0.057153 | 0.039951 | 0.044394 | 0.116162 | 0.053647 |
| 0.103509 | 0.180716 | 0.159178 | 0.066772 | 0.178584 | 0.057136 | 0.03995 | 0.044396 | 0.116121 | 0.053637 |
| 0.103532 | 0.180698 | 0.159183 | 0.066774 | 0.178574 | 0.057145 | 0.039957 | 0.044394 | 0.116126 | 0.053618 |
| 0.103538 | 0.180755 | 0.159111 | 0.06678 | 0.178588 | 0.057137 | 0.039952 | 0.044394 | 0.116116 | 0.053629 |
| 0.103507 | 0.180791 | 0.159114 | 0.066771 | 0.178557 | 0.05714 | 0.03995 | 0.044398 | 0.116134 | 0.053638 |
| 0.103512 | 0.180776 | 0.159184 | 0.066762 | 0.178533 | 0.057141 | 0.039946 | 0.04439 | 0.11612 | 0.053636 |
| 0.103506 | 0.18079 | 0.159179 | 0.066771 | 0.178498 | 0.057131 | 0.039952 | 0.044389 | 0.116147 | 0.053638 |
| 0.10356 | 0.180692 | 0.159204 | 0.066759 | 0.17853 | 0.057142 | 0.039948 | 0.044391 | 0.116133 | 0.05364 |
| 0.10352 | 0.180727 | 0.159142 | 0.066792 | 0.178606 | 0.057141 | 0.039951 | 0.04439 | 0.116118 | 0.053613 |
| 0.10353 | 0.18079 | 0.159123 | 0.066772 | 0.178567 | 0.057144 | 0.039942 | 0.044386 | 0.116096 | 0.05365 |
| 0.103543 | 0.180714 | 0.159156 | 0.066775 | 0.178561 | 0.057152 | 0.039956 | 0.044394 | 0.116117 | 0.053631 |
| 0.103555 | 0.180723 | 0.159162 | 0.066763 | 0.178529 | 0.057141 | 0.039952 | 0.044392 | 0.116143 | 0.05364 |
| 0.10355 | 0.180776 | 0.159173 | 0.066773 | 0.178523 | 0.057137 | 0.039943 | 0.044386 | 0.116109 | 0.053629 |
| 0.103532 | 0.180786 | 0.159167 | 0.066769 | 0.178511 | 0.057138 | 0.039944 | 0.04439 | 0.116128 | 0.053635 |
| 0.103529 | 0.180757 | 0.159193 | 0.066756 | 0.178537 | 0.057147 | 0.039946 | 0.044396 | 0.116092 | 0.053646 |
| 0.103528 | 0.180746 | 0.159108 | 0.066767 | 0.178526 | 0.057153 | 0.039963 | 0.044403 | 0.116155 | 0.053651 |
| 0.103497 | 0.180816 | 0.159135 | 0.066771 | 0.178539 | 0.057147 | 0.039956 | 0.044399 | 0.116112 | 0.053629 |
| 0.103552 | 0.180777 | 0.159167 | 0.066771 | 0.178515 | 0.057134 | 0.039942 | 0.044392 | 0.116139 | 0.053611 |
| 0.10354 | 0.180777 | 0.159111 | 0.066781 | 0.17856 | 0.05713 | 0.039953 | 0.04439 | 0.116107 | 0.05365 |
| 0.103527 | 0.180774 | 0.15912 | 0.066787 | 0.178578 | 0.057128 | 0.039944 | 0.044387 | 0.11612 | 0.053635 |
| 0.103504 | 0.180778 | 0.159162 | 0.066772 | 0.178533 | 0.057143 | 0.039954 | 0.044388 | 0.116131 | 0.053633 |
| 0.103509 | 0.180808 | 0.15911 | 0.066771 | 0.178523 | 0.05715 | 0.039959 | 0.044402 | 0.116133 | 0.053636 |
| 0.103494 | 0.180813 | 0.15912 | 0.066777 | 0.178527 | 0.057143 | 0.039952 | 0.044398 | 0.116137 | 0.05364 |
| 0.103513 | 0.180784 | 0.159122 | 0.066774 | 0.178572 | 0.057134 | 0.039946 | 0.044395 | 0.116141 | 0.05362 |
| 0.103531 | 0.180729 | 0.159133 | 0.066783 | 0.17856 | 0.057153 | 0.03995 | 0.044392 | 0.116116 | 0.053654 |
| 0.103546 | 0.180715 | 0.15912 | 0.066784 | 0.178545 | 0.057136 | 0.039957 | 0.044402 | 0.116154 | 0.053642 |
| 0.103482 | 0.180779 | 0.159129 | 0.06679 | 0.17859 | 0.057135 | 0.039947 | 0.044392 | 0.116131 | 0.053626 |
| 0.103506 | 0.180717 | 0.159129 | 0.066786 | 0.178594 | 0.057133 | 0.039959 | 0.044393 | 0.116153 | 0.053631 |
| 0.103524 | 0.180694 | 0.159157 | 0.06677 | 0.178576 | 0.057146 | 0.039954 | 0.044399 | 0.116139 | 0.053641 |
| 0.103507 | 0.180778 | 0.159085 | 0.066786 | 0.178557 | 0.057142 | 0.039958 | 0.044394 | 0.116155 | 0.053638 |
| 0.103557 | 0.180766 | 0.159174 | 0.066776 | 0.178555 | 0.057127 | 0.039943 | 0.044387 | 0.116101 | 0.053615 |
| 0.103543 | 0.180723 | 0.159137 | 0.066784 | 0.178595 | 0.057143 | 0.039947 | 0.044388 | 0.116098 | 0.053642 |
| 0.103554 | 0.18076 | 0.159165 | 0.066769 | 0.17852 | 0.057134 | 0.039949 | 0.044397 | 0.116134 | 0.053617 |
| 0.103496 | 0.180792 | 0.159119 | 0.066795 | 0.178605 | 0.057128 | 0.039948 | 0.044388 | 0.116111 | 0.053619 |
| 0.103495 | 0.180762 | 0.159113 | 0.066771 | 0.17858 | 0.057153 | 0.039953 | 0.044398 | 0.116128 | 0.053646 |
| 0.103529 | 0.180751 | 0.159147 | 0.066786 | 0.178587 | 0.057133 | 0.039944 | 0.044395 | 0.116108 | 0.053619 |
| 0.10353 | 0.180756 | 0.159153 | 0.066765 | 0.178545 | 0.057141 | 0.039951 | 0.044397 | 0.116132 | 0.053631 |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 0.103534 | 0.180709 | 0.159107 | 0.066776 | 0.178589 | 0.057153 | 0.039956 | 0.044401 | 0.116151 | 0.053624 |
| 0.103511 | 0.18073 | 0.159116 | 0.066794 | 0.178625 | 0.057151 | 0.039947 | 0.044396 | 0.116109 | 0.053621 |
| 0.103528 | 0.180739 | 0.159137 | 0.066774 | 0.178533 | 0.057135 | 0.039955 | 0.044401 | 0.116137 | 0.053661 |
| 0.103512 | 0.180731 | 0.159114 | 0.066794 | 0.17861 | 0.057145 | 0.039957 | 0.044394 | 0.11611 | 0.053632 |
| 0.103496 | 0.180815 | 0.159103 | 0.066768 | 0.178571 | 0.057142 | 0.039954 | 0.044401 | 0.116109 | 0.05364 |
| 0.103502 | 0.180781 | 0.15912 | 0.06677 | 0.178556 | 0.057151 | 0.039959 | 0.0444 | 0.11614 | 0.053623 |
| 0.10352 | 0.180776 | 0.15915 | 0.066774 | 0.178548 | 0.057146 | 0.039943 | 0.044396 | 0.116142 | 0.053605 |
| 0.103509 | 0.180733 | 0.159145 | 0.066788 | 0.178587 | 0.057134 | 0.039946 | 0.044397 | 0.116139 | 0.053621 |
| 0.103556 | 0.180713 | 0.15916 | 0.066769 | 0.178571 | 0.057153 | 0.039951 | 0.044396 | 0.116101 | 0.053628 |
| 0.103532 | 0.180765 | 0.159185 | 0.066773 | 0.178567 | 0.057123 | 0.03994 | 0.044387 | 0.116112 | 0.053616 |
| 0.103527 | 0.180795 | 0.159157 | 0.066764 | 0.178506 | 0.057132 | 0.039952 | 0.044395 | 0.116131 | 0.05364 |
| 0.103537 | 0.180727 | 0.15917 | 0.066773 | 0.178545 | 0.057146 | 0.039957 | 0.044395 | 0.11612 | 0.05363 |
| 0.103557 | 0.180718 | 0.159152 | 0.066761 | 0.178549 | 0.057144 | 0.039958 | 0.044396 | 0.116107 | 0.053658 |
| 0.103507 | 0.1808 | 0.159105 | 0.066767 | 0.178555 | 0.057139 | 0.039955 | 0.044391 | 0.116129 | 0.053651 |
| 0.10354 | 0.180785 | 0.159177 | 0.066757 | 0.178535 | 0.057125 | 0.039945 | 0.044389 | 0.116122 | 0.053625 |
| 0.103546 | 0.180699 | 0.15918 | 0.06678 | 0.178588 | 0.057131 | 0.03995 | 0.04439 | 0.116116 | 0.053621 |
| 0.103531 | 0.180781 | 0.159155 | 0.066772 | 0.178547 | 0.057134 | 0.039949 | 0.044394 | 0.116112 | 0.053625 |
| 0.103492 | 0.180746 | 0.159099 | 0.066795 | 0.178588 | 0.057155 | 0.039961 | 0.0444 | 0.116136 | 0.053628 |
| 0.103519 | 0.18072 | 0.159197 | 0.06677 | 0.178594 | 0.057145 | 0.039945 | 0.044387 | 0.116095 | 0.053629 |
| 0.103508 | 0.180702 | 0.159162 | 0.066787 | 0.178594 | 0.057141 | 0.039956 | 0.044389 | 0.11611 | 0.053652 |
| 0.103526 | 0.180707 | 0.159185 | 0.066781 | 0.178576 | 0.057129 | 0.039946 | 0.044391 | 0.116115 | 0.053644 |
| 0.103515 | 0.180784 | 0.159173 | 0.066775 | 0.178542 | 0.057134 | 0.039948 | 0.044391 | 0.116117 | 0.053621 |
| 0.103519 | 0.180702 | 0.159127 | 0.06677 | 0.178532 | 0.057156 | 0.039965 | 0.0444 | 0.116166 | 0.053665 |
| 0.103552 | 0.180772 | 0.159113 | 0.066762 | 0.178516 | 0.057153 | 0.039956 | 0.044402 | 0.116158 | 0.053617 |
| 0.103514 | 0.180701 | 0.159147 | 0.066781 | 0.178595 | 0.057142 | 0.039949 | 0.044397 | 0.116156 | 0.053618 |
| 0.10349 | 0.180743 | 0.159124 | 0.06678 | 0.178551 | 0.057148 | 0.039963 | 0.044397 | 0.116152 | 0.053651 |
| 0.103552 | 0.180717 | 0.159182 | 0.066771 | 0.178543 | 0.05714 | 0.039956 | 0.044396 | 0.11612 | 0.053624 |
| 0.103529 | 0.180796 | 0.159192 | 0.066765 | 0.178506 | 0.05714 | 0.039953 | 0.044391 | 0.116115 | 0.053614 |
| 0.103518 | 0.180754 | 0.159121 | 0.066773 | 0.17854 | 0.057159 | 0.039959 | 0.044395 | 0.116116 | 0.053666 |
| 0.103508 | 0.180728 | 0.159117 | 0.066788 | 0.178595 | 0.057129 | 0.039953 | 0.044396 | 0.116138 | 0.053648 |
| 0.10354 | 0.180767 | 0.159197 | 0.06676 | 0.178512 | 0.057131 | 0.039944 | 0.044392 | 0.116114 | 0.053642 |
| 0.10352 | 0.180731 | 0.159136 | 0.066793 | 0.178608 | 0.057141 | 0.039952 | 0.044387 | 0.116106 | 0.053625 |
| 0.103526 | 0.180716 | 0.159144 | 0.066762 | 0.178531 | 0.057137 | 0.039963 | 0.044402 | 0.11615 | 0.053668 |
| 0.103542 | 0.180731 | 0.159132 | 0.066782 | 0.178548 | 0.057144 | 0.039958 | 0.044394 | 0.116131 | 0.053638 |
| 0.1035 | 0.180726 | 0.159153 | 0.066786 | 0.178587 | 0.057153 | 0.039953 | 0.044398 | 0.116105 | 0.053638 |
| 0.103546 | 0.180799 | 0.159114 | 0.066765 | 0.178508 | 0.057129 | 0.039953 | 0.044402 | 0.116162 | 0.053623 |
| 0.103546 | 0.180769 | 0.159137 | 0.066767 | 0.178508 | 0.057129 | 0.039953 | 0.044397 | 0.116155 | 0.053639 |
| 0.10352 | 0.1807 | 0.159183 | 0.066786 | 0.17857 | 0.057131 | 0.039952 | 0.044396 | 0.116145 | 0.053618 |
| 0.103541 | 0.180814 | 0.15917 | 0.06676 | 0.178505 | 0.057138 | 0.039943 | 0.044386 | 0.116102 | 0.053641 |
| 0.103527 | 0.180765 | 0.159139 | 0.066785 | 0.178549 | 0.057139 | 0.039956 | 0.044394 | 0.116126 | 0.053621 |
| 0.103536 | 0.180781 | 0.159165 | 0.066769 | 0.178522 | 0.05714 | 0.03995 | 0.044388 | 0.116142 | 0.053608 |
| 0.103503 | 0.180761 | 0.159128 | 0.066787 | 0.17855 | 0.057137 | 0.039959 | 0.0444 | 0.116153 | 0.053622 |
| 0.10353 | 0.180703 | 0.159147 | 0.066784 | 0.178595 | 0.057152 | 0.039952 | 0.044392 | 0.11612 | 0.053625 |
| 0.103543 | 0.180701 | 0.159165 | 0.066771 | 0.178569 | 0.05714 | 0.03996 | 0.044399 | 0.116125 | 0.053627 |
| 0.103498 | 0.180792 | 0.159112 | 0.066789 | 0.178573 | 0.057145 | 0.039949 | 0.044395 | 0.116122 | 0.053624 |
| 0.103503 | 0.180763 | 0.159137 | 0.066782 | 0.178581 | 0.057135 | 0.03995 | 0.044393 | 0.116129 | 0.053626 |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 0.103527 | 0.180752 | 0.159148 | 0.066776 | 0.178538 | 0.057157 | 0.039955 | 0.044403 | 0.11612 | 0.053623 |
| 0.103516 | 0.180789 | 0.159129 | 0.066777 | 0.178535 | 0.057136 | 0.039955 | 0.0444 | 0.116154 | 0.05361 |
| 0.103541 | 0.180771 | 0.159147 | 0.066765 | 0.178548 | 0.057135 | 0.039947 | 0.044394 | 0.116143 | 0.05361 |
| 0.103513 | 0.180735 | 0.159203 | 0.066766 | 0.178523 | 0.057132 | 0.039953 | 0.044391 | 0.116141 | 0.053644 |
| 0.103546 | 0.180752 | 0.159173 | 0.066773 | 0.17852 | 0.057133 | 0.03995 | 0.044388 | 0.116149 | 0.053614 |
| 0.103515 | 0.180744 | 0.159142 | 0.066771 | 0.17858 | 0.057145 | 0.039949 | 0.044397 | 0.116113 | 0.053643 |
| 0.103533 | 0.180732 | 0.159216 | 0.06676 | 0.178516 | 0.057131 | 0.039951 | 0.044397 | 0.116139 | 0.053625 |
| 0.103524 | 0.180738 | 0.159172 | 0.066759 | 0.178509 | 0.05715 | 0.039957 | 0.044402 | 0.116135 | 0.053654 |
| 0.103555 | 0.180753 | 0.159189 | 0.066765 | 0.178539 | 0.057147 | 0.039946 | 0.044396 | 0.116104 | 0.053607 |
| 0.1035 | 0.180799 | 0.159139 | 0.066774 | 0.178524 | 0.057144 | 0.039956 | 0.044395 | 0.116134 | 0.053634 |
| 0.103545 | 0.180787 | 0.159104 | 0.066762 | 0.178532 | 0.05714 | 0.039954 | 0.044392 | 0.116125 | 0.053659 |
| 0.103529 | 0.180698 | 0.159193 | 0.066765 | 0.178527 | 0.057147 | 0.03996 | 0.044394 | 0.116139 | 0.053647 |
| 0.103516 | 0.18069 | 0.159165 | 0.066778 | 0.178548 | 0.057152 | 0.039955 | 0.044401 | 0.116145 | 0.053651 |
| 0.103548 | 0.180712 | 0.159111 | 0.066789 | 0.178583 | 0.05715 | 0.039954 | 0.044397 | 0.116125 | 0.053632 |
| 0.103514 | 0.180776 | 0.159173 | 0.066764 | 0.17853 | 0.057122 | 0.039951 | 0.044392 | 0.116132 | 0.053646 |
| 0.103513 | 0.180765 | 0.159114 | 0.06679 | 0.178584 | 0.05713 | 0.039945 | 0.044397 | 0.11614 | 0.053621 |
| 0.103541 | 0.180685 | 0.159186 | 0.066778 | 0.178537 | 0.057134 | 0.039952 | 0.04439 | 0.116152 | 0.053645 |
| 0.103502 | 0.180797 | 0.159095 | 0.066779 | 0.178573 | 0.057145 | 0.039949 | 0.044392 | 0.116125 | 0.053642 |
| 0.103551 | 0.180707 | 0.159102 | 0.066792 | 0.178588 | 0.057147 | 0.039956 | 0.044393 | 0.116128 | 0.053634 |
| 0.10352 | 0.180737 | 0.159178 | 0.06678 | 0.17856 | 0.057138 | 0.039957 | 0.044397 | 0.116103 | 0.05363 |
| 0.103537 | 0.180718 | 0.159175 | 0.066773 | 0.17857 | 0.057125 | 0.039945 | 0.044394 | 0.116147 | 0.053618 |
| 0.103535 | 0.180751 | 0.159131 | 0.066789 | 0.178558 | 0.057132 | 0.039947 | 0.044399 | 0.116145 | 0.053613 |
| 0.103504 | 0.180699 | 0.159104 | 0.066792 | 0.178584 | 0.057156 | 0.03996 | 0.044395 | 0.11615 | 0.053655 |
| 0.103542 | 0.180726 | 0.159161 | 0.066773 | 0.178579 | 0.057143 | 0.039947 | 0.044394 | 0.116105 | 0.053631 |
| 0.103548 | 0.180726 | 0.159146 | 0.066785 | 0.178573 | 0.057143 | 0.039952 | 0.044392 | 0.116117 | 0.053618 |
| 0.103491 | 0.180705 | 0.15918 | 0.066778 | 0.178591 | 0.057133 | 0.039955 | 0.044394 | 0.116142 | 0.05363 |
| 0.103527 | 0.180792 | 0.159174 | 0.066749 | 0.178493 | 0.057144 | 0.039949 | 0.044391 | 0.116131 | 0.053649 |
| 0.103515 | 0.180696 | 0.159163 | 0.066782 | 0.178548 | 0.05714 | 0.039961 | 0.0444 | 0.116159 | 0.053635 |
| 0.103531 | 0.180786 | 0.159177 | 0.066766 | 0.178529 | 0.057133 | 0.039951 | 0.044389 | 0.116104 | 0.053634 |
| 0.103521 | 0.180731 | 0.159114 | 0.066779 | 0.178555 | 0.05715 | 0.039961 | 0.0444 | 0.116152 | 0.053635 |
| 0.103543 | 0.180772 | 0.159123 | 0.066765 | 0.178521 | 0.057147 | 0.039957 | 0.044402 | 0.116152 | 0.053618 |
| 0.103526 | 0.180816 | 0.159097 | 0.066769 | 0.178536 | 0.057137 | 0.039947 | 0.044391 | 0.116151 | 0.05363 |
| 0.103537 | 0.180726 | 0.159124 | 0.066787 | 0.178563 | 0.057151 | 0.039956 | 0.044403 | 0.11612 | 0.053635 |
| 0.103516 | 0.180791 | 0.15917 | 0.066774 | 0.178528 | 0.057151 | 0.039947 | 0.044391 | 0.116118 | 0.053614 |
| 0.103527 | 0.180757 | 0.15914 | 0.066769 | 0.17852 | 0.057147 | 0.039958 | 0.044396 | 0.11616 | 0.053627 |
| 0.103507 | 0.180689 | 0.159166 | 0.066773 | 0.178571 | 0.057135 | 0.039959 | 0.044397 | 0.11615 | 0.053652 |
| 0.103553 | 0.180761 | 0.159175 | 0.066757 | 0.178512 | 0.057145 | 0.039947 | 0.044391 | 0.11611 | 0.053649 |
| 0.103547 | 0.180743 | 0.159114 | 0.066783 | 0.178595 | 0.057144 | 0.039949 | 0.044398 | 0.116116 | 0.053612 |
| 0.103531 | 0.180805 | 0.159129 | 0.066772 | 0.178535 | 0.057152 | 0.039953 | 0.044397 | 0.116106 | 0.05362 |
| 0.10354 | 0.180765 | 0.159103 | 0.066761 | 0.178549 | 0.057136 | 0.039957 | 0.0444 | 0.11613 | 0.053658 |
| 0.103554 | 0.18072 | 0.159158 | 0.066763 | 0.178549 | 0.05714 | 0.039952 | 0.044389 | 0.116122 | 0.053654 |
| 0.103532 | 0.180781 | 0.159118 | 0.066781 | 0.178586 | 0.057136 | 0.039942 | 0.044387 | 0.11611 | 0.053627 |
| 0.103541 | 0.180765 | 0.159184 | 0.066767 | 0.178534 | 0.057127 | 0.039949 | 0.044395 | 0.116108 | 0.053628 |
| 0.103508 | 0.180803 | 0.159093 | 0.066778 | 0.178583 | 0.057133 | 0.039954 | 0.044397 | 0.116103 | 0.053648 |
| 0.103522 | 0.180798 | 0.159186 | 0.066761 | 0.178546 | 0.057127 | 0.039952 | 0.044386 | 0.116092 | 0.05363 |
| 0.103497 | 0.180739 | 0.159124 | 0.066776 | 0.178594 | 0.057158 | 0.03996 | 0.044393 | 0.116111 | 0.053646 |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 0.103544 | 0.180736 | 0.15917 | 0.066765 | 0.178534 | 0.057138 | 0.039945 | 0.044387 | 0.116131 | 0.053652 |
| 0.103501 | 0.180778 | 0.159086 | 0.066787 | 0.178596 | 0.057148 | 0.039954 | 0.044393 | 0.116126 | 0.053633 |
| 0.103522 | 0.18076 | 0.159131 | 0.066774 | 0.17856 | 0.057141 | 0.03995 | 0.044395 | 0.116133 | 0.053633 |
| 0.103542 | 0.180781 | 0.1591 | 0.066773 | 0.178548 | 0.057145 | 0.039952 | 0.044389 | 0.116129 | 0.053641 |
| 0.103522 | 0.180742 | 0.159194 | 0.066774 | 0.178555 | 0.057147 | 0.039946 | 0.044392 | 0.116117 | 0.053612 |
| 0.103515 | 0.180767 | 0.159148 | 0.06678 | 0.178536 | 0.057129 | 0.039952 | 0.044399 | 0.116126 | 0.053649 |
| 0.103494 | 0.180811 | 0.159152 | 0.066779 | 0.178573 | 0.057132 | 0.039951 | 0.044398 | 0.116098 | 0.053611 |
| 0.103544 | 0.180705 | 0.159124 | 0.066772 | 0.178576 | 0.057151 | 0.039962 | 0.044401 | 0.116115 | 0.05365 |
| 0.103533 | 0.180729 | 0.159135 | 0.066773 | 0.178587 | 0.05715 | 0.03995 | 0.044388 | 0.11612 | 0.053635 |
| 0.103527 | 0.180733 | 0.159162 | 0.066772 | 0.178562 | 0.05714 | 0.039954 | 0.044394 | 0.116124 | 0.053633 |
| 0.103541 | 0.180777 | 0.15914 | 0.066769 | 0.178564 | 0.05713 | 0.039952 | 0.044389 | 0.116097 | 0.053641 |
| 0.103544 | 0.18071 | 0.15916 | 0.066773 | 0.178542 | 0.057147 | 0.03996 | 0.044396 | 0.116115 | 0.053652 |
| 0.10354 | 0.180745 | 0.159131 | 0.066783 | 0.178537 | 0.057136 | 0.039964 | 0.044403 | 0.116119 | 0.053642 |
| 0.103531 | 0.180772 | 0.159126 | 0.066766 | 0.178523 | 0.05713 | 0.03995 | 0.044394 | 0.116153 | 0.053656 |
| 0.103537 | 0.18075 | 0.15915 | 0.066767 | 0.178574 | 0.057147 | 0.039942 | 0.04439 | 0.116102 | 0.053642 |
| 0.103491 | 0.180732 | 0.159146 | 0.066782 | 0.178584 | 0.057138 | 0.039949 | 0.044397 | 0.116134 | 0.053647 |
| 0.103529 | 0.180737 | 0.159127 | 0.066768 | 0.178538 | 0.057153 | 0.039962 | 0.044402 | 0.116138 | 0.053646 |
| 0.103521 | 0.180751 | 0.159197 | 0.066763 | 0.178526 | 0.057136 | 0.039958 | 0.0444 | 0.116123 | 0.053623 |
| 0.103508 | 0.180744 | 0.159157 | 0.066768 | 0.178576 | 0.057146 | 0.039955 | 0.04439 | 0.116113 | 0.053645 |
| 0.103504 | 0.180699 | 0.159195 | 0.066772 | 0.178553 | 0.057151 | 0.039955 | 0.044398 | 0.116115 | 0.053656 |
| 0.10355 | 0.180712 | 0.159139 | 0.066773 | 0.178603 | 0.057148 | 0.039953 | 0.044388 | 0.116104 | 0.05363 |
| 0.103495 | 0.180755 | 0.159183 | 0.066774 | 0.178548 | 0.057146 | 0.039946 | 0.044395 | 0.116136 | 0.053622 |
| 0.103546 | 0.180758 | 0.159153 | 0.066769 | 0.178538 | 0.05715 | 0.039955 | 0.044396 | 0.116113 | 0.053622 |
| 0.103541 | 0.180802 | 0.159185 | 0.066753 | 0.178519 | 0.057128 | 0.039942 | 0.044386 | 0.116098 | 0.053645 |
| 0.103496 | 0.18077 | 0.15911 | 0.066776 | 0.178556 | 0.057136 | 0.039965 | 0.044404 | 0.116131 | 0.053657 |
| 0.103496 | 0.180728 | 0.159155 | 0.066786 | 0.178563 | 0.057151 | 0.039953 | 0.044392 | 0.11615 | 0.053625 |
| 0.103552 | 0.180715 | 0.159199 | 0.066766 | 0.178525 | 0.057132 | 0.039949 | 0.044396 | 0.116113 | 0.053652 |
| 0.103526 | 0.180704 | 0.159161 | 0.066787 | 0.178615 | 0.057134 | 0.039948 | 0.044389 | 0.116118 | 0.053617 |
| 0.103546 | 0.180715 | 0.159156 | 0.066776 | 0.178552 | 0.057147 | 0.039959 | 0.044398 | 0.116135 | 0.053615 |
| 0.1035 | 0.180722 | 0.159167 | 0.066773 | 0.178575 | 0.057132 | 0.039957 | 0.044395 | 0.116123 | 0.053656 |
| 0.103539 | 0.180748 | 0.159157 | 0.066765 | 0.178565 | 0.057128 | 0.039943 | 0.044386 | 0.116134 | 0.053635 |
| 0.103511 | 0.180773 | 0.15915 | 0.066778 | 0.178549 | 0.057141 | 0.039948 | 0.044391 | 0.116106 | 0.053651 |
| 0.103543 | 0.180761 | 0.159136 | 0.066778 | 0.178552 | 0.057145 | 0.039952 | 0.044389 | 0.116137 | 0.053607 |
| 0.103546 | 0.180745 | 0.159161 | 0.066767 | 0.178514 | 0.057142 | 0.039956 | 0.0444 | 0.11615 | 0.053619 |
| 0.103536 | 0.180712 | 0.159161 | 0.066778 | 0.178533 | 0.057148 | 0.039961 | 0.044399 | 0.116156 | 0.053616 |
| 0.103548 | 0.180779 | 0.159115 | 0.066775 | 0.178554 | 0.057132 | 0.039948 | 0.044398 | 0.116112 | 0.053639 |
| 0.103543 | 0.180722 | 0.159145 | 0.066775 | 0.17854 | 0.057141 | 0.039953 | 0.044393 | 0.116121 | 0.053665 |
| 0.103546 | 0.180733 | 0.159197 | 0.066758 | 0.178513 | 0.05715 | 0.039956 | 0.044396 | 0.116118 | 0.053633 |
| 0.103527 | 0.180724 | 0.159157 | 0.066783 | 0.178569 | 0.05715 | 0.039948 | 0.044396 | 0.116121 | 0.053625 |
| 0.103503 | 0.180779 | 0.159145 | 0.066775 | 0.178531 | 0.057148 | 0.039953 | 0.044395 | 0.116136 | 0.053635 |
| 0.103556 | 0.180693 | 0.159199 | 0.066773 | 0.178511 | 0.057141 | 0.039958 | 0.0444 | 0.116156 | 0.053613 |
| 0.103516 | 0.180775 | 0.159111 | 0.066772 | 0.178588 | 0.05713 | 0.039943 | 0.044388 | 0.116133 | 0.053644 |
| 0.103556 | 0.180709 | 0.159193 | 0.066763 | 0.17856 | 0.057129 | 0.039947 | 0.044387 | 0.116132 | 0.053624 |
| 0.10351 | 0.180708 | 0.159147 | 0.066791 | 0.178609 | 0.057148 | 0.039956 | 0.044392 | 0.116122 | 0.053615 |
| 0.103501 | 0.18068 | 0.159167 | 0.066792 | 0.17861 | 0.057154 | 0.039959 | 0.044392 | 0.116117 | 0.053628 |
| 0.103543 | 0.180707 | 0.159174 | 0.066778 | 0.17856 | 0.057144 | 0.039948 | 0.044391 | 0.116132 | 0.053623 |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 0.10351 | 0.180718 | 0.159111 | 0.066778 | 0.178604 | 0.057148 | 0.03996 | 0.044403 | 0.116137 | 0.053632 |
| 0.103503 | 0.180718 | 0.159202 | 0.066779 | 0.178598 | 0.057145 | 0.039952 | 0.044393 | 0.116099 | 0.053612 |
| 0.103543 | 0.180793 | 0.15913 | 0.066779 | 0.178529 | 0.057128 | 0.039944 | 0.044393 | 0.116149 | 0.053613 |
| 0.103526 | 0.180715 | 0.159096 | 0.066793 | 0.178608 | 0.057152 | 0.039956 | 0.044396 | 0.116127 | 0.053632 |
| 0.103562 | 0.180726 | 0.159132 | 0.066781 | 0.178549 | 0.057153 | 0.039953 | 0.044396 | 0.116121 | 0.053627 |
| 0.103522 | 0.180786 | 0.159198 | 0.066756 | 0.178509 | 0.057139 | 0.039952 | 0.044394 | 0.116116 | 0.053628 |
| 0.103546 | 0.180775 | 0.159157 | 0.066768 | 0.178507 | 0.057136 | 0.039949 | 0.044397 | 0.116134 | 0.05363 |
| 0.103544 | 0.180766 | 0.159148 | 0.066759 | 0.178526 | 0.05714 | 0.039952 | 0.044394 | 0.116134 | 0.053637 |
| 0.103518 | 0.180731 | 0.159163 | 0.066775 | 0.178545 | 0.057137 | 0.039957 | 0.044391 | 0.116145 | 0.053638 |
| 0.103548 | 0.180774 | 0.159116 | 0.066766 | 0.178515 | 0.057152 | 0.039956 | 0.044402 | 0.116143 | 0.053628 |
| 0.103513 | 0.180719 | 0.15917 | 0.066786 | 0.178585 | 0.057137 | 0.039947 | 0.044387 | 0.116135 | 0.053619 |
| 0.103499 | 0.180726 | 0.159117 | 0.066787 | 0.178608 | 0.057156 | 0.039957 | 0.044398 | 0.116132 | 0.053621 |
| 0.10353 | 0.180748 | 0.159175 | 0.066762 | 0.178541 | 0.057131 | 0.039952 | 0.044397 | 0.116121 | 0.053644 |
| 0.103509 | 0.180707 | 0.15919 | 0.066779 | 0.178582 | 0.057145 | 0.039946 | 0.044387 | 0.116135 | 0.053622 |
| 0.103486 | 0.180764 | 0.159158 | 0.066774 | 0.178566 | 0.057138 | 0.039955 | 0.044397 | 0.116122 | 0.05364 |
| 0.10353 | 0.18077 | 0.159189 | 0.066747 | 0.178495 | 0.05714 | 0.039959 | 0.0444 | 0.116122 | 0.053648 |
| 0.103532 | 0.180685 | 0.159177 | 0.066779 | 0.178582 | 0.057139 | 0.039958 | 0.044394 | 0.116106 | 0.05365 |
| 0.103527 | 0.180718 | 0.159138 | 0.066773 | 0.17858 | 0.057149 | 0.039947 | 0.044392 | 0.116138 | 0.053639 |
| 0.103544 | 0.180736 | 0.15916 | 0.066779 | 0.17853 | 0.057138 | 0.039959 | 0.0444 | 0.116124 | 0.05363 |
| 0.10354 | 0.180792 | 0.159152 | 0.066772 | 0.178507 | 0.057131 | 0.03995 | 0.04439 | 0.116139 | 0.053628 |
| 0.103497 | 0.1808 | 0.159133 | 0.066769 | 0.178553 | 0.057139 | 0.039955 | 0.044402 | 0.116107 | 0.053645 |
| 0.103529 | 0.180727 | 0.159135 | 0.066776 | 0.178583 | 0.057129 | 0.039951 | 0.044399 | 0.11614 | 0.053631 |
| 0.103522 | 0.180765 | 0.159158 | 0.06677 | 0.17859 | 0.05714 | 0.039945 | 0.044391 | 0.116107 | 0.053612 |
| 0.103532 | 0.180764 | 0.159173 | 0.066769 | 0.178517 | 0.057145 | 0.039957 | 0.044399 | 0.116124 | 0.053622 |
| 0.103549 | 0.18077 | 0.159135 | 0.066772 | 0.178534 | 0.057142 | 0.039948 | 0.04439 | 0.116106 | 0.053654 |
| 0.103513 | 0.180817 | 0.159166 | 0.06677 | 0.178523 | 0.057134 | 0.039947 | 0.044393 | 0.116116 | 0.053621 |
| 0.103493 | 0.180787 | 0.15914 | 0.066776 | 0.178562 | 0.057131 | 0.039953 | 0.0444 | 0.11614 | 0.053619 |
| 0.103511 | 0.18076 | 0.15911 | 0.06678 | 0.178551 | 0.057133 | 0.039957 | 0.0444 | 0.116159 | 0.053639 |
| 0.103547 | 0.180706 | 0.159136 | 0.066775 | 0.178542 | 0.057155 | 0.039956 | 0.044397 | 0.116152 | 0.053634 |
| 0.103518 | 0.180803 | 0.159103 | 0.066771 | 0.178532 | 0.057143 | 0.039959 | 0.044401 | 0.116113 | 0.053656 |
| 0.103523 | 0.180721 | 0.15919 | 0.066777 | 0.178534 | 0.057131 | 0.03996 | 0.044394 | 0.116124 | 0.053645 |
| 0.103496 | 0.180732 | 0.159152 | 0.066775 | 0.178579 | 0.057133 | 0.039957 | 0.044394 | 0.116142 | 0.053641 |
| 0.103511 | 0.180733 | 0.159186 | 0.06678 | 0.178577 | 0.057135 | 0.039954 | 0.044392 | 0.116106 | 0.053626 |
| 0.103536 | 0.180774 | 0.159189 | 0.066761 | 0.178544 | 0.057129 | 0.03994 | 0.044388 | 0.116136 | 0.053603 |
| 0.103537 | 0.180777 | 0.159151 | 0.066764 | 0.178553 | 0.057146 | 0.039943 | 0.044395 | 0.116116 | 0.053619 |
| 0.103528 | 0.180772 | 0.159132 | 0.06677 | 0.17851 | 0.057137 | 0.039958 | 0.0444 | 0.116151 | 0.053642 |
| 0.103526 | 0.180795 | 0.15915 | 0.066769 | 0.178557 | 0.057128 | 0.039945 | 0.044389 | 0.116123 | 0.053617 |
| 0.103548 | 0.180678 | 0.159189 | 0.066761 | 0.178566 | 0.05713 | 0.039958 | 0.044397 | 0.116121 | 0.053652 |
| 0.103526 | 0.180775 | 0.159164 | 0.066776 | 0.178526 | 0.057127 | 0.039952 | 0.044393 | 0.116151 | 0.053609 |
| 0.103515 | 0.180741 | 0.159172 | 0.066774 | 0.178567 | 0.057146 | 0.039945 | 0.044395 | 0.1161 | 0.053643 |
| 0.103508 | 0.180731 | 0.159146 | 0.066789 | 0.178607 | 0.057153 | 0.039947 | 0.044389 | 0.11611 | 0.053619 |
| 0.103534 | 0.180776 | 0.159105 | 0.066784 | 0.178553 | 0.057154 | 0.03995 | 0.044401 | 0.116118 | 0.053626 |
| 0.103519 | 0.180738 | 0.159122 | 0.066772 | 0.178563 | 0.057141 | 0.039955 | 0.044401 | 0.116149 | 0.053637 |
| 0.10351 | 0.18071 | 0.159149 | 0.06678 | 0.178561 | 0.057153 | 0.039962 | 0.044397 | 0.116137 | 0.053641 |
| 0.10353 | 0.180727 | 0.15911 | 0.066782 | 0.178596 | 0.057143 | 0.039947 | 0.044389 | 0.116136 | 0.053642 |
| 0.103544 | 0.1807 | 0.159147 | 0.066771 | 0.178515 | 0.057142 | 0.039965 | 0.044402 | 0.116152 | 0.053662 |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 0.103553 | 0.180715 | 0.159127 | 0.066777 | 0.178586 | 0.057138 | 0.039958 | 0.044394 | 0.116112 | 0.05364 |
| 0.103538 | 0.180724 | 0.159125 | 0.066779 | 0.178557 | 0.05713 | 0.039951 | 0.044398 | 0.116158 | 0.053638 |
| 0.10394 | 0.171433 | 0.153469 | 0.066945 | 0.174353 | 0.060254 | 0.043491 | 0.046822 | 0.118874 | 0.060419 |
| 0.103515 | 0.180699 | 0.159132 | 0.066791 | 0.178587 | 0.057137 | 0.039954 | 0.044396 | 0.116151 | 0.053638 |
| 0.103513 | 0.180713 | 0.159187 | 0.066776 | 0.178566 | 0.057145 | 0.039957 | 0.044394 | 0.116112 | 0.053637 |
| 0.103509 | 0.180689 | 0.159176 | 0.066779 | 0.178548 | 0.057136 | 0.039961 | 0.0444 | 0.116156 | 0.053646 |
| 0.10353 | 0.18077 | 0.159181 | 0.066777 | 0.178558 | 0.057121 | 0.039942 | 0.044386 | 0.116122 | 0.053613 |
| 0.103549 | 0.18076 | 0.159165 | 0.066759 | 0.178522 | 0.057142 | 0.039944 | 0.044397 | 0.116141 | 0.053621 |
| 0.103535 | 0.180707 | 0.159134 | 0.066776 | 0.178533 | 0.057141 | 0.039965 | 0.044403 | 0.116153 | 0.053653 |
| 0.103545 | 0.180763 | 0.159177 | 0.066767 | 0.178495 | 0.057122 | 0.039955 | 0.044397 | 0.116153 | 0.053627 |
| 0.103514 | 0.180724 | 0.159162 | 0.066781 | 0.178556 | 0.05715 | 0.039955 | 0.044394 | 0.116119 | 0.053647 |
| 0.103535 | 0.180711 | 0.159133 | 0.066777 | 0.178601 | 0.05714 | 0.039958 | 0.044399 | 0.116108 | 0.053636 |
| 0.103509 | 0.18078 | 0.159115 | 0.066777 | 0.178607 | 0.057148 | 0.039949 | 0.044392 | 0.1161 | 0.053624 |
| 0.103497 | 0.180718 | 0.159103 | 0.066792 | 0.178575 | 0.057141 | 0.03996 | 0.044403 | 0.116167 | 0.053643 |
| 0.103548 | 0.180702 | 0.15912 | 0.066786 | 0.178536 | 0.057138 | 0.03996 | 0.044402 | 0.116171 | 0.053637 |
| 0.103523 | 0.180701 | 0.15919 | 0.066783 | 0.178557 | 0.05713 | 0.039951 | 0.044399 | 0.116154 | 0.053612 |
| 0.103498 | 0.180698 | 0.159111 | 0.066803 | 0.178628 | 0.057143 | 0.03995 | 0.04439 | 0.116141 | 0.053638 |
| 0.103547 | 0.180701 | 0.159182 | 0.066778 | 0.17858 | 0.057145 | 0.039951 | 0.044387 | 0.116101 | 0.053629 |
| 0.103516 | 0.180701 | 0.159114 | 0.06679 | 0.178613 | 0.057155 | 0.039957 | 0.044396 | 0.11614 | 0.053617 |
| 0.103532 | 0.180691 | 0.159188 | 0.066772 | 0.178593 | 0.057147 | 0.039949 | 0.044396 | 0.116118 | 0.053614 |
| 0.103514 | 0.180734 | 0.159164 | 0.066777 | 0.17854 | 0.057136 | 0.03996 | 0.044396 | 0.116147 | 0.05363 |
| 0.103537 | 0.180767 | 0.159125 | 0.066776 | 0.178534 | 0.057141 | 0.039948 | 0.04439 | 0.116119 | 0.053663 |
| 0.103508 | 0.180717 | 0.15916 | 0.066777 | 0.178538 | 0.057148 | 0.039961 | 0.044401 | 0.116153 | 0.053636 |
| 0.103523 | 0.180814 | 0.159177 | 0.066755 | 0.178512 | 0.057139 | 0.039947 | 0.044388 | 0.116099 | 0.053646 |
| 0.103515 | 0.180746 | 0.159103 | 0.06679 | 0.178589 | 0.057137 | 0.03995 | 0.044396 | 0.116149 | 0.053624 |
| 0.103516 | 0.180716 | 0.159118 | 0.066783 | 0.178557 | 0.057149 | 0.039963 | 0.044403 | 0.116151 | 0.053644 |
| 0.103508 | 0.180786 | 0.159155 | 0.066787 | 0.178562 | 0.057142 | 0.039945 | 0.044389 | 0.116118 | 0.053607 |
| 0.103536 | 0.180775 | 0.15912 | 0.066763 | 0.178512 | 0.057136 | 0.039955 | 0.044402 | 0.116152 | 0.053649 |
| 0.103517 | 0.180721 | 0.159142 | 0.066777 | 0.178595 | 0.05714 | 0.039952 | 0.044395 | 0.116129 | 0.053632 |
| 0.103558 | 0.18075 | 0.159185 | 0.066761 | 0.178521 | 0.057132 | 0.039944 | 0.044388 | 0.116124 | 0.05364 |
| 0.103543 | 0.180706 | 0.15915 | 0.06677 | 0.178564 | 0.057143 | 0.039955 | 0.04439 | 0.116142 | 0.053636 |
| 0.103516 | 0.180744 | 0.159169 | 0.066774 | 0.178535 | 0.057143 | 0.03995 | 0.044398 | 0.116154 | 0.053619 |
| 0.103507 | 0.180746 | 0.15917 | 0.066764 | 0.178563 | 0.057136 | 0.039955 | 0.044398 | 0.11611 | 0.053651 |
| 0.103509 | 0.180792 | 0.159186 | 0.066758 | 0.178521 | 0.057124 | 0.039952 | 0.044396 | 0.116139 | 0.053623 |
| 0.103509 | 0.180785 | 0.159109 | 0.066777 | 0.17858 | 0.057142 | 0.039944 | 0.044392 | 0.116137 | 0.053625 |
| 0.103517 | 0.180704 | 0.159169 | 0.06678 | 0.178538 | 0.057136 | 0.039955 | 0.044402 | 0.116148 | 0.053651 |
| 0.103532 | 0.180712 | 0.159166 | 0.066787 | 0.178555 | 0.057144 | 0.039949 | 0.04439 | 0.116134 | 0.053632 |
| 0.103548 | 0.180724 | 0.159161 | 0.066773 | 0.178538 | 0.057152 | 0.039958 | 0.044394 | 0.116114 | 0.053639 |
| 0.103532 | 0.180774 | 0.159128 | 0.06677 | 0.17855 | 0.057146 | 0.039956 | 0.044398 | 0.116108 | 0.053638 |
| 0.103504 | 0.180806 | 0.15917 | 0.066763 | 0.178541 | 0.057131 | 0.039946 | 0.044386 | 0.116105 | 0.053648 |
| 0.103551 | 0.180746 | 0.15914 | 0.066762 | 0.178537 | 0.057153 | 0.039956 | 0.044392 | 0.116117 | 0.053646 |
| 0.103547 | 0.180742 | 0.159115 | 0.066784 | 0.178577 | 0.057132 | 0.039945 | 0.044392 | 0.116113 | 0.053652 |
| 0.103501 | 0.180691 | 0.159181 | 0.066779 | 0.178592 | 0.057135 | 0.039958 | 0.044397 | 0.11615 | 0.053616 |
| 0.103536 | 0.180771 | 0.159089 | 0.066783 | 0.178552 | 0.057143 | 0.03995 | 0.044391 | 0.116133 | 0.053651 |
| 0.103507 | 0.180789 | 0.159101 | 0.066776 | 0.178587 | 0.057136 | 0.039946 | 0.044396 | 0.116134 | 0.053628 |
| 0.103507 | 0.180777 | 0.159125 | 0.066775 | 0.17858 | 0.057146 | 0.039944 | 0.044391 | 0.116102 | 0.053652 |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 0.103505 | 0.180804 | 0.159093 | 0.066783 | 0.178598 | 0.057135 | 0.039942 | 0.044393 | 0.116137 | 0.05361 |
| 0.103527 | 0.180701 | 0.159208 | 0.066777 | 0.178545 | 0.057142 | 0.039957 | 0.044396 | 0.116106 | 0.05364 |
| 0.103514 | 0.180794 | 0.159163 | 0.066783 | 0.178554 | 0.057145 | 0.039945 | 0.04439 | 0.116104 | 0.053608 |
| 0.103498 | 0.180744 | 0.159158 | 0.066778 | 0.178581 | 0.057146 | 0.039951 | 0.044395 | 0.116132 | 0.053616 |
| 0.103524 | 0.180706 | 0.159115 | 0.066787 | 0.178578 | 0.057144 | 0.039958 | 0.044393 | 0.116148 | 0.053648 |
| 0.1035 | 0.180705 | 0.159154 | 0.066779 | 0.178611 | 0.05714 | 0.039952 | 0.04439 | 0.116121 | 0.053648 |
| 0.103501 | 0.18079 | 0.15909 | 0.066766 | 0.178556 | 0.057148 | 0.039952 | 0.044399 | 0.116143 | 0.053656 |
| 0.103512 | 0.180787 | 0.159099 | 0.066784 | 0.17856 | 0.057133 | 0.039953 | 0.044389 | 0.116144 | 0.053639 |
| 0.103511 | 0.180789 | 0.159187 | 0.066782 | 0.178554 | 0.057132 | 0.039943 | 0.044389 | 0.1161 | 0.053612 |
| 0.103518 | 0.180804 | 0.159106 | 0.066776 | 0.17855 | 0.057133 | 0.039953 | 0.044391 | 0.116123 | 0.053647 |
| 0.103533 | 0.180736 | 0.159132 | 0.066769 | 0.178558 | 0.05713 | 0.039954 | 0.044394 | 0.116158 | 0.053637 |
| 0.103493 | 0.180704 | 0.159172 | 0.066784 | 0.178567 | 0.057151 | 0.039956 | 0.044391 | 0.116128 | 0.053654 |
| 0.103539 | 0.180716 | 0.159143 | 0.066777 | 0.178544 | 0.057147 | 0.039951 | 0.044396 | 0.116142 | 0.053646 |
| 0.103542 | 0.180703 | 0.15918 | 0.066773 | 0.178555 | 0.057133 | 0.039946 | 0.044393 | 0.11614 | 0.053636 |
| 0.10351 | 0.180702 | 0.159196 | 0.06677 | 0.178524 | 0.057147 | 0.039955 | 0.044399 | 0.116149 | 0.053647 |
| 0.103513 | 0.18074 | 0.159132 | 0.066768 | 0.178533 | 0.057149 | 0.039964 | 0.0444 | 0.116166 | 0.053635 |
| 0.10349 | 0.180708 | 0.159186 | 0.066778 | 0.17859 | 0.057131 | 0.039952 | 0.044393 | 0.11613 | 0.053641 |
| 0.103531 | 0.180723 | 0.159152 | 0.066784 | 0.178581 | 0.057136 | 0.039949 | 0.044391 | 0.116106 | 0.053648 |
| 0.103538 | 0.180728 | 0.159188 | 0.066762 | 0.178507 | 0.057149 | 0.03996 | 0.0444 | 0.116119 | 0.053648 |
| 0.10354 | 0.180758 | 0.159148 | 0.06677 | 0.17854 | 0.05715 | 0.039955 | 0.044394 | 0.116113 | 0.053631 |
| 0.103523 | 0.180733 | 0.159183 | 0.066786 | 0.178548 | 0.057142 | 0.03995 | 0.044391 | 0.116123 | 0.053622 |
| 0.103553 | 0.180724 | 0.159107 | 0.06677 | 0.178586 | 0.05714 | 0.03995 | 0.044398 | 0.116122 | 0.053651 |
| 0.103522 | 0.180746 | 0.159172 | 0.066771 | 0.178573 | 0.057142 | 0.039951 | 0.044387 | 0.116107 | 0.053629 |
| 0.103499 | 0.180749 | 0.159144 | 0.066769 | 0.178568 | 0.057148 | 0.039957 | 0.044393 | 0.116113 | 0.053659 |
| 0.103508 | 0.180796 | 0.159117 | 0.066771 | 0.178568 | 0.057132 | 0.039953 | 0.044393 | 0.116133 | 0.05363 |
| 0.10352 | 0.180736 | 0.159165 | 0.066765 | 0.178536 | 0.05715 | 0.039952 | 0.044393 | 0.116121 | 0.053663 |
| 0.103537 | 0.180783 | 0.15909 | 0.06678 | 0.17855 | 0.057141 | 0.039955 | 0.044392 | 0.116117 | 0.053654 |
| 0.103493 | 0.180798 | 0.159186 | 0.066763 | 0.17855 | 0.057148 | 0.039951 | 0.044395 | 0.116094 | 0.053624 |
| 0.103499 | 0.180757 | 0.159139 | 0.066785 | 0.178573 | 0.057126 | 0.039955 | 0.044394 | 0.116137 | 0.053636 |
| 0.103529 | 0.180731 | 0.159124 | 0.066768 | 0.178571 | 0.057156 | 0.039958 | 0.044401 | 0.116109 | 0.053653 |
| 0.103538 | 0.180762 | 0.159161 | 0.066766 | 0.178537 | 0.057144 | 0.039945 | 0.044389 | 0.116126 | 0.053631 |
| 0.103505 | 0.180713 | 0.159182 | 0.066786 | 0.178614 | 0.057136 | 0.039947 | 0.044391 | 0.116103 | 0.053623 |
| 0.103542 | 0.180766 | 0.159125 | 0.066781 | 0.178554 | 0.057136 | 0.039944 | 0.044393 | 0.116151 | 0.053609 |
| 0.103499 | 0.180709 | 0.159153 | 0.066784 | 0.178552 | 0.057141 | 0.039963 | 0.044403 | 0.116138 | 0.053658 |
| 0.103512 | 0.180701 | 0.159103 | 0.066781 | 0.178587 | 0.057154 | 0.039955 | 0.044399 | 0.116159 | 0.053647 |
| 0.103513 | 0.180703 | 0.159176 | 0.066783 | 0.178616 | 0.057155 | 0.039948 | 0.044391 | 0.116098 | 0.053617 |
| 0.103559 | 0.180693 | 0.159202 | 0.066762 | 0.178538 | 0.057144 | 0.039947 | 0.044392 | 0.11614 | 0.053622 |
| 0.103523 | 0.180795 | 0.159129 | 0.066755 | 0.178508 | 0.057141 | 0.039957 | 0.044401 | 0.11614 | 0.05365 |
| 0.103525 | 0.180757 | 0.159136 | 0.066781 | 0.17854 | 0.057145 | 0.039959 | 0.044398 | 0.116144 | 0.053616 |
| 0.103537 | 0.180775 | 0.159184 | 0.066769 | 0.178503 | 0.057129 | 0.039946 | 0.044389 | 0.116126 | 0.053642 |
| 0.103487 | 0.180753 | 0.159082 | 0.066787 | 0.178605 | 0.05715 | 0.039958 | 0.044397 | 0.116126 | 0.053656 |
| 0.103538 | 0.180762 | 0.159179 | 0.066771 | 0.178564 | 0.057126 | 0.039943 | 0.044393 | 0.116109 | 0.053616 |
| 0.103521 | 0.180718 | 0.159158 | 0.066777 | 0.17856 | 0.057146 | 0.039952 | 0.044396 | 0.116126 | 0.053646 |
| 0.103496 | 0.180801 | 0.159182 | 0.066759 | 0.17854 | 0.057128 | 0.039947 | 0.044387 | 0.116108 | 0.053651 |
| 0.103502 | 0.180748 | 0.159182 | 0.066787 | 0.178584 | 0.05715 | 0.039944 | 0.044389 | 0.116101 | 0.053614 |
| 0.103503 | 0.1807 | 0.159099 | 0.066792 | 0.178612 | 0.057147 | 0.039951 | 0.044399 | 0.11615 | 0.053646 |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 0.10349 | 0.180785 | 0.15916 | 0.066767 | 0.178571 | 0.057148 | 0.039949 | 0.044389 | 0.116098 | 0.053642 |
| 0.103536 | 0.180775 | 0.159155 | 0.066764 | 0.178514 | 0.057131 | 0.03995 | 0.044392 | 0.116125 | 0.053659 |
| 0.103498 | 0.180745 | 0.159154 | 0.06678 | 0.178567 | 0.057155 | 0.039953 | 0.044393 | 0.116122 | 0.053634 |
| 0.103515 | 0.180768 | 0.159116 | 0.066779 | 0.178566 | 0.057132 | 0.039951 | 0.044388 | 0.116143 | 0.053642 |
| 0.103526 | 0.18075 | 0.159119 | 0.066791 | 0.178557 | 0.057136 | 0.039956 | 0.044394 | 0.116155 | 0.053617 |
| 0.103513 | 0.18071 | 0.159142 | 0.066793 | 0.178621 | 0.057145 | 0.039956 | 0.044394 | 0.116105 | 0.053621 |
| 0.103521 | 0.180749 | 0.159152 | 0.066775 | 0.178561 | 0.057144 | 0.039949 | 0.044394 | 0.116125 | 0.05363 |
| 0.103502 | 0.180714 | 0.15914 | 0.066794 | 0.178606 | 0.05715 | 0.039956 | 0.044393 | 0.116117 | 0.053626 |
| 0.103555 | 0.180748 | 0.159173 | 0.066762 | 0.178548 | 0.057139 | 0.039952 | 0.044389 | 0.116102 | 0.053632 |
| 0.103491 | 0.180759 | 0.159166 | 0.066772 | 0.178538 | 0.057138 | 0.039956 | 0.044398 | 0.116123 | 0.053658 |
| 0.103523 | 0.180727 | 0.159112 | 0.066773 | 0.178592 | 0.057151 | 0.039958 | 0.044391 | 0.116121 | 0.05365 |
| 0.103526 | 0.180719 | 0.15916 | 0.066776 | 0.178523 | 0.057137 | 0.039964 | 0.044396 | 0.116164 | 0.053635 |
| 0.103511 | 0.180805 | 0.159166 | 0.066772 | 0.17852 | 0.057146 | 0.039955 | 0.044398 | 0.116109 | 0.053618 |
| 0.103492 | 0.180701 | 0.159121 | 0.066776 | 0.178598 | 0.057149 | 0.039956 | 0.0444 | 0.116154 | 0.053653 |
| 0.103526 | 0.180709 | 0.159168 | 0.066772 | 0.178564 | 0.057149 | 0.039954 | 0.044392 | 0.116118 | 0.053647 |
| 0.103535 | 0.180698 | 0.159175 | 0.06677 | 0.178538 | 0.057135 | 0.039962 | 0.044399 | 0.116155 | 0.053632 |
| 0.10353 | 0.180796 | 0.159122 | 0.066782 | 0.178582 | 0.057132 | 0.039949 | 0.044395 | 0.116105 | 0.053608 |
| 0.103536 | 0.180728 | 0.159151 | 0.06678 | 0.178585 | 0.057136 | 0.039955 | 0.044387 | 0.116106 | 0.053635 |
| 0.103496 | 0.180788 | 0.159158 | 0.066763 | 0.178564 | 0.05714 | 0.039947 | 0.044391 | 0.116131 | 0.053621 |
| 0.103503 | 0.18076 | 0.159118 | 0.066792 | 0.178583 | 0.057138 | 0.039949 | 0.044393 | 0.116143 | 0.053621 |
| 0.103508 | 0.180759 | 0.15909 | 0.066793 | 0.178581 | 0.057144 | 0.039955 | 0.044392 | 0.116144 | 0.053634 |
| 0.103545 | 0.180721 | 0.159172 | 0.066762 | 0.178521 | 0.057141 | 0.039964 | 0.0444 | 0.116133 | 0.053642 |
| 0.103499 | 0.180714 | 0.159147 | 0.066774 | 0.178593 | 0.057144 | 0.039951 | 0.044398 | 0.116137 | 0.053642 |
| 0.103516 | 0.180752 | 0.159118 | 0.066793 | 0.178583 | 0.05714 | 0.039952 | 0.044397 | 0.116126 | 0.053624 |
| 0.103502 | 0.180719 | 0.159193 | 0.066772 | 0.178536 | 0.057143 | 0.039951 | 0.044391 | 0.116156 | 0.053638 |
| 0.103486 | 0.180788 | 0.159173 | 0.066783 | 0.178587 | 0.057122 | 0.039941 | 0.044386 | 0.116112 | 0.053623 |
| 0.10354 | 0.180729 | 0.159152 | 0.066779 | 0.178527 | 0.057149 | 0.039955 | 0.0444 | 0.116133 | 0.053636 |
| 0.103541 | 0.180807 | 0.159167 | 0.066769 | 0.17851 | 0.057132 | 0.039946 | 0.044393 | 0.116102 | 0.053633 |
| 0.103543 | 0.180777 | 0.159138 | 0.066775 | 0.178509 | 0.05714 | 0.039956 | 0.044392 | 0.116147 | 0.053623 |
| 0.103546 | 0.180727 | 0.159123 | 0.066768 | 0.178552 | 0.057151 | 0.039952 | 0.04439 | 0.116135 | 0.053656 |
| 0.103529 | 0.180697 | 0.159102 | 0.066792 | 0.17861 | 0.057147 | 0.039955 | 0.044393 | 0.116126 | 0.053648 |
| 0.103502 | 0.180737 | 0.159126 | 0.06679 | 0.178564 | 0.057148 | 0.039957 | 0.044401 | 0.116147 | 0.053628 |
| 0.103493 | 0.180733 | 0.159156 | 0.066784 | 0.178572 | 0.05714 | 0.039958 | 0.044394 | 0.116113 | 0.053657 |
| 0.103541 | 0.180707 | 0.159121 | 0.066791 | 0.178587 | 0.057155 | 0.03996 | 0.044402 | 0.116118 | 0.053616 |
| 0.103546 | 0.180827 | 0.159132 | 0.066757 | 0.178516 | 0.057143 | 0.039944 | 0.044389 | 0.11612 | 0.053626 |
| 0.103532 | 0.180709 | 0.15919 | 0.06678 | 0.178556 | 0.05714 | 0.039953 | 0.044386 | 0.116104 | 0.053649 |
| 0.103537 | 0.1807 | 0.159154 | 0.066774 | 0.178536 | 0.057137 | 0.03996 | 0.044398 | 0.116159 | 0.053647 |
| 0.10351 | 0.180756 | 0.159164 | 0.066775 | 0.178537 | 0.057143 | 0.03995 | 0.044392 | 0.116147 | 0.053626 |
| 0.103544 | 0.180764 | 0.15911 | 0.066775 | 0.178529 | 0.057133 | 0.039952 | 0.044399 | 0.116137 | 0.053657 |
| 0.103561 | 0.180711 | 0.159136 | 0.066771 | 0.178531 | 0.057134 | 0.039957 | 0.044399 | 0.116137 | 0.053664 |
| 0.103518 | 0.180726 | 0.159169 | 0.066777 | 0.178536 | 0.057147 | 0.039952 | 0.044393 | 0.116137 | 0.053644 |
| 0.103515 | 0.180775 | 0.159172 | 0.066764 | 0.178541 | 0.057145 | 0.039952 | 0.044392 | 0.116107 | 0.053638 |
| 0.103509 | 0.180808 | 0.159092 | 0.066784 | 0.1786 | 0.057143 | 0.039946 | 0.044392 | 0.116092 | 0.053634 |
| 0.103536 | 0.180747 | 0.159167 | 0.066768 | 0.17853 | 0.057138 | 0.039951 | 0.044396 | 0.116113 | 0.053653 |
| 0.103538 | 0.180768 | 0.159141 | 0.066768 | 0.178547 | 0.057138 | 0.039945 | 0.044394 | 0.116108 | 0.053653 |
| 0.103561 | 0.18069 | 0.159203 | 0.066761 | 0.178512 | 0.05713 | 0.039951 | 0.044392 | 0.11616 | 0.053638 |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 0.103499 | 0.180721 | 0.159104 | 0.066802 | 0.178627 | 0.057138 | 0.039953 | 0.044394 | 0.116118 | 0.053645 |
| 0.103517 | 0.180775 | 0.159146 | 0.06678 | 0.178584 | 0.05714 | 0.039943 | 0.044393 | 0.116084 | 0.05364 |
| 0.10351 | 0.180724 | 0.159157 | 0.06678 | 0.178553 | 0.057139 | 0.039954 | 0.044401 | 0.116121 | 0.053661 |
| 0.103502 | 0.180697 | 0.159171 | 0.066775 | 0.178597 | 0.057146 | 0.039955 | 0.044397 | 0.11612 | 0.05364 |
| 0.103508 | 0.180779 | 0.159113 | 0.066774 | 0.178523 | 0.057132 | 0.039962 | 0.044398 | 0.116162 | 0.053648 |
| 0.103523 | 0.180729 | 0.159177 | 0.066776 | 0.178581 | 0.057139 | 0.039952 | 0.044386 | 0.116096 | 0.053641 |
| 0.103529 | 0.180738 | 0.15912 | 0.066779 | 0.178609 | 0.05714 | 0.039947 | 0.044389 | 0.116112 | 0.053636 |
| 0.10352 | 0.180804 | 0.159163 | 0.066768 | 0.178544 | 0.057141 | 0.039944 | 0.044389 | 0.116104 | 0.053624 |
| 0.10354 | 0.180765 | 0.159143 | 0.066777 | 0.178526 | 0.057136 | 0.039954 | 0.044391 | 0.116154 | 0.053613 |
| 0.103526 | 0.18073 | 0.159146 | 0.066775 | 0.178519 | 0.057142 | 0.039959 | 0.044396 | 0.116161 | 0.053646 |
| 0.103529 | 0.180722 | 0.159162 | 0.066768 | 0.178566 | 0.057131 | 0.039955 | 0.044396 | 0.116147 | 0.053624 |
| 0.103522 | 0.180713 | 0.159125 | 0.066779 | 0.178571 | 0.057134 | 0.03996 | 0.044393 | 0.116158 | 0.053644 |
| 0.103516 | 0.180734 | 0.159131 | 0.066795 | 0.178607 | 0.057134 | 0.039949 | 0.044398 | 0.11612 | 0.053614 |
| 0.103509 | 0.180799 | 0.159173 | 0.066764 | 0.178538 | 0.057148 | 0.03995 | 0.044393 | 0.116108 | 0.053616 |
| 0.103546 | 0.180785 | 0.159181 | 0.066758 | 0.178504 | 0.057136 | 0.039944 | 0.044389 | 0.116109 | 0.053647 |
| 0.103516 | 0.180801 | 0.159118 | 0.066758 | 0.178508 | 0.057142 | 0.039954 | 0.044394 | 0.116156 | 0.053654 |
| 0.103547 | 0.180722 | 0.159141 | 0.066772 | 0.178519 | 0.057135 | 0.039952 | 0.044392 | 0.116161 | 0.053658 |
| 0.103524 | 0.180771 | 0.15911 | 0.066783 | 0.178552 | 0.05713 | 0.03995 | 0.04439 | 0.116141 | 0.053648 |
| 0.103519 | 0.180718 | 0.159199 | 0.066777 | 0.17855 | 0.057149 | 0.039949 | 0.04439 | 0.116115 | 0.053635 |
| 0.103536 | 0.18071 | 0.159188 | 0.06677 | 0.178563 | 0.057144 | 0.039947 | 0.044394 | 0.116105 | 0.053643 |
| 0.103495 | 0.180783 | 0.159173 | 0.066773 | 0.178551 | 0.057129 | 0.039951 | 0.044398 | 0.116111 | 0.053636 |
| 0.103487 | 0.180784 | 0.159158 | 0.066772 | 0.178581 | 0.057145 | 0.039943 | 0.044385 | 0.116122 | 0.053622 |
| 0.103556 | 0.180789 | 0.159164 | 0.066762 | 0.178515 | 0.057124 | 0.039943 | 0.044387 | 0.116139 | 0.053621 |
| 0.103532 | 0.180768 | 0.159111 | 0.066777 | 0.178567 | 0.05714 | 0.03995 | 0.044393 | 0.116118 | 0.053644 |
| 0.103544 | 0.180717 | 0.159112 | 0.066772 | 0.178528 | 0.05715 | 0.039961 | 0.0444 | 0.116157 | 0.053659 |
| 0.103542 | 0.180718 | 0.159179 | 0.066775 | 0.178572 | 0.057137 | 0.03995 | 0.044387 | 0.116115 | 0.053624 |
| 0.10353 | 0.180729 | 0.159182 | 0.066783 | 0.178562 | 0.057133 | 0.03995 | 0.044395 | 0.116113 | 0.053623 |
| 0.103531 | 0.180786 | 0.159158 | 0.066772 | 0.178537 | 0.057136 | 0.039944 | 0.044389 | 0.116121 | 0.053626 |
| 0.103538 | 0.180706 | 0.159138 | 0.066776 | 0.17855 | 0.057141 | 0.039956 | 0.044397 | 0.116144 | 0.053653 |
| 0.103511 | 0.180771 | 0.159159 | 0.066786 | 0.178568 | 0.057144 | 0.039949 | 0.044391 | 0.116105 | 0.053616 |
| 0.103523 | 0.180773 | 0.159171 | 0.066773 | 0.178552 | 0.057133 | 0.039952 | 0.044388 | 0.116101 | 0.053634 |
| 0.103545 | 0.180734 | 0.159136 | 0.06676 | 0.178536 | 0.057152 | 0.039963 | 0.044403 | 0.116112 | 0.053658 |
| 0.103491 | 0.180727 | 0.159115 | 0.066789 | 0.178622 | 0.057142 | 0.039954 | 0.0444 | 0.116107 | 0.053653 |
| 0.103523 | 0.180743 | 0.159128 | 0.066786 | 0.178584 | 0.057134 | 0.039953 | 0.044399 | 0.116106 | 0.053644 |
| 0.103518 | 0.180729 | 0.159125 | 0.066773 | 0.178554 | 0.05715 | 0.039954 | 0.044393 | 0.116154 | 0.053649 |
| 0.10351 | 0.180742 | 0.159134 | 0.066774 | 0.178559 | 0.057157 | 0.039957 | 0.044398 | 0.11612 | 0.053649 |
| 0.103521 | 0.180809 | 0.159119 | 0.066771 | 0.178524 | 0.05715 | 0.039952 | 0.044399 | 0.11613 | 0.053624 |
| 0.103535 | 0.180722 | 0.159139 | 0.066772 | 0.178571 | 0.057127 | 0.039954 | 0.044387 | 0.116144 | 0.05365 |
| 0.103525 | 0.180771 | 0.159096 | 0.066779 | 0.178547 | 0.057147 | 0.03996 | 0.044399 | 0.116133 | 0.053643 |
| 0.103542 | 0.180725 | 0.15915 | 0.066782 | 0.178567 | 0.057134 | 0.039946 | 0.04439 | 0.116116 | 0.053647 |
| 0.10355 | 0.180735 | 0.159163 | 0.066757 | 0.178496 | 0.057144 | 0.039955 | 0.044395 | 0.116156 | 0.053648 |
| 0.103506 | 0.180673 | 0.159171 | 0.066791 | 0.178588 | 0.057144 | 0.039957 | 0.04439 | 0.116147 | 0.053633 |
| 0.103541 | 0.180748 | 0.159159 | 0.066766 | 0.178529 | 0.057145 | 0.039954 | 0.044391 | 0.116138 | 0.053629 |
| 0.103543 | 0.180721 | 0.159147 | 0.066772 | 0.178594 | 0.057148 | 0.039945 | 0.044392 | 0.116098 | 0.05364 |
| 0.103527 | 0.180783 | 0.15919 | 0.066754 | 0.178509 | 0.057128 | 0.039955 | 0.044393 | 0.116114 | 0.053647 |
| 0.103527 | 0.180795 | 0.159138 | 0.066775 | 0.178576 | 0.057144 | 0.039943 | 0.04439 | 0.1161 | 0.053613 |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 0.103526 | 0.180728 | 0.159125 | 0.066785 | 0.178567 | 0.05715 | 0.039954 | 0.044394 | 0.116127 | 0.053644 |
| 0.103517 | 0.180755 | 0.159105 | 0.066786 | 0.178573 | 0.057142 | 0.039957 | 0.044396 | 0.116156 | 0.053613 |
| 0.103507 | 0.180785 | 0.159163 | 0.066765 | 0.178536 | 0.057139 | 0.039949 | 0.044399 | 0.116132 | 0.053623 |
| 0.103516 | 0.180791 | 0.159113 | 0.066769 | 0.178529 | 0.057139 | 0.039952 | 0.044397 | 0.116137 | 0.053657 |
| 0.103521 | 0.18077 | 0.159122 | 0.06677 | 0.178541 | 0.057153 | 0.039949 | 0.044399 | 0.116138 | 0.053636 |
| 0.103493 | 0.180693 | 0.159197 | 0.066796 | 0.178614 | 0.057144 | 0.039956 | 0.04439 | 0.116107 | 0.053608 |
| 0.103535 | 0.180719 | 0.159184 | 0.066776 | 0.178575 | 0.057146 | 0.039941 | 0.044386 | 0.116125 | 0.053611 |
| 0.10351 | 0.180708 | 0.159214 | 0.066779 | 0.178558 | 0.057132 | 0.039953 | 0.04439 | 0.116123 | 0.053634 |
| 0.103515 | 0.180742 | 0.1591 | 0.066789 | 0.178589 | 0.057136 | 0.039949 | 0.044396 | 0.116138 | 0.053647 |
| 0.10352 | 0.180773 | 0.159112 | 0.066772 | 0.178563 | 0.057142 | 0.039953 | 0.044396 | 0.116119 | 0.05365 |
| 0.103539 | 0.180778 | 0.159154 | 0.066762 | 0.178517 | 0.057146 | 0.039955 | 0.044401 | 0.116118 | 0.053629 |
| 0.103514 | 0.180809 | 0.159147 | 0.06678 | 0.178551 | 0.057135 | 0.039954 | 0.044398 | 0.116103 | 0.053609 |
| 0.103504 | 0.180754 | 0.159146 | 0.066758 | 0.178511 | 0.057147 | 0.039961 | 0.044403 | 0.11615 | 0.053666 |
| 0.103505 | 0.180777 | 0.159148 | 0.066766 | 0.178564 | 0.057133 | 0.039945 | 0.044393 | 0.116136 | 0.053632 |
| 0.103489 | 0.180752 | 0.159137 | 0.06678 | 0.1786 | 0.057151 | 0.039958 | 0.0444 | 0.116114 | 0.053619 |
| 0.103537 | 0.180687 | 0.159182 | 0.066787 | 0.178585 | 0.057143 | 0.039954 | 0.044392 | 0.116106 | 0.053626 |
| 0.103512 | 0.180751 | 0.159168 | 0.066778 | 0.178587 | 0.057145 | 0.03995 | 0.044391 | 0.116097 | 0.053622 |
| 0.103531 | 0.18073 | 0.15915 | 0.06677 | 0.178553 | 0.057133 | 0.039956 | 0.044392 | 0.116148 | 0.053637 |
| 0.103531 | 0.180739 | 0.159129 | 0.066776 | 0.178555 | 0.057137 | 0.039954 | 0.0444 | 0.116127 | 0.053651 |
| 0.103515 | 0.180713 | 0.159159 | 0.066787 | 0.178582 | 0.057131 | 0.039958 | 0.044396 | 0.116145 | 0.053615 |
| 0.103513 | 0.180735 | 0.159188 | 0.06678 | 0.178575 | 0.057147 | 0.039949 | 0.044386 | 0.116119 | 0.053607 |
| 0.103492 | 0.180786 | 0.159177 | 0.066776 | 0.17854 | 0.057139 | 0.039944 | 0.044392 | 0.116132 | 0.053623 |
| 0.103522 | 0.180734 | 0.15918 | 0.066765 | 0.178549 | 0.05715 | 0.039955 | 0.044401 | 0.116121 | 0.053623 |
| 0.103524 | 0.180743 | 0.159127 | 0.066767 | 0.178535 | 0.057144 | 0.039956 | 0.044398 | 0.11614 | 0.053666 |
| 0.103522 | 0.180725 | 0.15918 | 0.066778 | 0.178554 | 0.057128 | 0.039952 | 0.044394 | 0.116143 | 0.053625 |
| 0.103544 | 0.180757 | 0.159115 | 0.066771 | 0.178527 | 0.057134 | 0.039962 | 0.044401 | 0.116154 | 0.053635 |
| 0.103531 | 0.180776 | 0.159163 | 0.06677 | 0.17856 | 0.057128 | 0.039943 | 0.044385 | 0.116136 | 0.053608 |
| 0.103548 | 0.18073 | 0.159111 | 0.06677 | 0.178578 | 0.057133 | 0.039946 | 0.04439 | 0.116135 | 0.053659 |
| 0.10352 | 0.180678 | 0.159154 | 0.066772 | 0.178567 | 0.057151 | 0.039961 | 0.044401 | 0.116144 | 0.053654 |
| 0.103545 | 0.180729 | 0.159105 | 0.066782 | 0.178578 | 0.057134 | 0.039951 | 0.044399 | 0.11616 | 0.053616 |
| 0.103517 | 0.18071 | 0.15918 | 0.066778 | 0.178562 | 0.057141 | 0.039955 | 0.044398 | 0.116137 | 0.053621 |
| 0.103502 | 0.180794 | 0.159109 | 0.066775 | 0.178595 | 0.057143 | 0.039949 | 0.044391 | 0.116096 | 0.053647 |
| 0.103535 | 0.180721 | 0.159119 | 0.066791 | 0.17858 | 0.05714 | 0.039961 | 0.044402 | 0.116125 | 0.053626 |
| 0.10352 | 0.180781 | 0.159115 | 0.06678 | 0.178538 | 0.057145 | 0.039956 | 0.0444 | 0.116142 | 0.053624 |
| 0.103511 | 0.180777 | 0.159147 | 0.066786 | 0.178595 | 0.057139 | 0.039945 | 0.044392 | 0.116091 | 0.053617 |
| 0.103523 | 0.180769 | 0.159123 | 0.066775 | 0.178527 | 0.057133 | 0.039955 | 0.044398 | 0.116166 | 0.053632 |
| 0.103525 | 0.180689 | 0.159199 | 0.066763 | 0.178524 | 0.057139 | 0.039959 | 0.044396 | 0.116147 | 0.053657 |
| 0.103505 | 0.180764 | 0.159101 | 0.066784 | 0.178538 | 0.057148 | 0.039959 | 0.044406 | 0.11617 | 0.053625 |
| 0.103505 | 0.180775 | 0.159142 | 0.06678 | 0.178585 | 0.057135 | 0.039952 | 0.044394 | 0.116113 | 0.053618 |
| 0.103494 | 0.180775 | 0.159156 | 0.066769 | 0.178516 | 0.057142 | 0.039957 | 0.044391 | 0.116141 | 0.053659 |
| 0.103541 | 0.180784 | 0.159123 | 0.066773 | 0.178541 | 0.057145 | 0.039943 | 0.044389 | 0.116137 | 0.053624 |
| 0.103528 | 0.180782 | 0.159102 | 0.066767 | 0.178563 | 0.057143 | 0.03995 | 0.044391 | 0.116122 | 0.053651 |
| 0.103542 | 0.180794 | 0.159153 | 0.066772 | 0.178533 | 0.05714 | 0.039947 | 0.044394 | 0.116119 | 0.053606 |
| 0.103526 | 0.180704 | 0.159131 | 0.066782 | 0.178592 | 0.057157 | 0.039951 | 0.044391 | 0.116138 | 0.053628 |
| 0.103505 | 0.180769 | 0.159139 | 0.066768 | 0.178567 | 0.057146 | 0.039953 | 0.044392 | 0.116128 | 0.053632 |
| 0.103528 | 0.180694 | 0.15914 | 0.066782 | 0.178599 | 0.057147 | 0.03995 | 0.044396 | 0.116136 | 0.053628 |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 0.103517 | 0.180731 | 0.15916 | 0.066768 | 0.178582 | 0.057149 | 0.03995 | 0.044391 | 0.116101 | 0.05365 |
| 0.103518 | 0.180753 | 0.159174 | 0.066769 | 0.178572 | 0.057126 | 0.039943 | 0.044385 | 0.116129 | 0.05363 |
| 0.103537 | 0.180786 | 0.159183 | 0.066756 | 0.178498 | 0.057136 | 0.039952 | 0.04439 | 0.116144 | 0.053616 |
| 0.103513 | 0.180742 | 0.159184 | 0.066777 | 0.178536 | 0.057146 | 0.039955 | 0.044393 | 0.116123 | 0.053631 |
| 0.103535 | 0.180757 | 0.159135 | 0.066775 | 0.178584 | 0.057146 | 0.039945 | 0.04439 | 0.11612 | 0.053613 |
| 0.103527 | 0.180794 | 0.159156 | 0.06676 | 0.178537 | 0.057136 | 0.03995 | 0.044397 | 0.116101 | 0.053643 |
| 0.103516 | 0.180786 | 0.159111 | 0.066778 | 0.178566 | 0.057133 | 0.039948 | 0.044396 | 0.116113 | 0.053653 |
| 0.103512 | 0.180789 | 0.159133 | 0.066759 | 0.178531 | 0.05714 | 0.039953 | 0.044392 | 0.116135 | 0.053655 |
| 0.103503 | 0.180749 | 0.159139 | 0.066793 | 0.17861 | 0.057135 | 0.039955 | 0.044396 | 0.116102 | 0.053618 |
| 0.103505 | 0.18075 | 0.159146 | 0.066771 | 0.178583 | 0.057142 | 0.039957 | 0.044395 | 0.116113 | 0.053639 |
| 0.103499 | 0.18078 | 0.1591 | 0.066772 | 0.178578 | 0.057143 | 0.039951 | 0.044396 | 0.116136 | 0.053645 |
| 0.103543 | 0.180779 | 0.159123 | 0.066771 | 0.178553 | 0.057131 | 0.039951 | 0.044393 | 0.116135 | 0.053621 |
| 0.103505 | 0.180784 | 0.159162 | 0.066767 | 0.178504 | 0.057144 | 0.039959 | 0.044397 | 0.116137 | 0.053641 |
| 0.103497 | 0.180683 | 0.159181 | 0.066787 | 0.178587 | 0.057131 | 0.039957 | 0.0444 | 0.116154 | 0.053623 |
| 0.103523 | 0.180691 | 0.159171 | 0.066776 | 0.178536 | 0.057149 | 0.039963 | 0.044399 | 0.116149 | 0.053644 |
| 0.103494 | 0.180756 | 0.159091 | 0.066781 | 0.178593 | 0.057153 | 0.039956 | 0.044402 | 0.116143 | 0.05363 |
| 0.103499 | 0.180768 | 0.159125 | 0.066785 | 0.178575 | 0.057128 | 0.039953 | 0.04439 | 0.116131 | 0.053645 |
| 0.103505 | 0.180735 | 0.159167 | 0.066777 | 0.178575 | 0.057147 | 0.039954 | 0.044395 | 0.116118 | 0.053628 |
| 0.103531 | 0.180781 | 0.159113 | 0.06677 | 0.178548 | 0.057151 | 0.039947 | 0.044396 | 0.116119 | 0.053643 |
| 0.103514 | 0.180776 | 0.15914 | 0.066774 | 0.178589 | 0.057131 | 0.039947 | 0.044398 | 0.116113 | 0.053619 |
| 0.103525 | 0.180775 | 0.159177 | 0.066778 | 0.178542 | 0.057128 | 0.03995 | 0.044391 | 0.116108 | 0.053626 |
| 0.103534 | 0.180809 | 0.159132 | 0.066762 | 0.178528 | 0.057133 | 0.039949 | 0.044392 | 0.116117 | 0.053644 |
| 0.103539 | 0.180792 | 0.159153 | 0.066757 | 0.178505 | 0.057141 | 0.039954 | 0.044398 | 0.116143 | 0.053618 |
| 0.103544 | 0.18072 | 0.159186 | 0.066765 | 0.178524 | 0.057145 | 0.039952 | 0.044397 | 0.116129 | 0.053637 |
| 0.103512 | 0.180694 | 0.159145 | 0.066795 | 0.178583 | 0.05715 | 0.039962 | 0.044396 | 0.116138 | 0.053626 |
| 0.1035 | 0.18071 | 0.159186 | 0.066777 | 0.178601 | 0.057143 | 0.039955 | 0.04439 | 0.116106 | 0.053632 |
| 0.103523 | 0.18076 | 0.159151 | 0.066768 | 0.178562 | 0.05714 | 0.039951 | 0.044391 | 0.116122 | 0.053633 |
| 0.10351 | 0.180714 | 0.159108 | 0.066784 | 0.178587 | 0.057134 | 0.039955 | 0.044392 | 0.11615 | 0.053666 |
| 0.103554 | 0.180718 | 0.15921 | 0.066776 | 0.178554 | 0.057132 | 0.039951 | 0.04439 | 0.116104 | 0.05361 |
| 0.103504 | 0.18073 | 0.159139 | 0.06678 | 0.178589 | 0.057145 | 0.039951 | 0.044396 | 0.116147 | 0.053619 |
| 0.10353 | 0.18074 | 0.159113 | 0.066785 | 0.178581 | 0.057137 | 0.039958 | 0.044396 | 0.116136 | 0.053624 |
| 0.103524 | 0.180719 | 0.159119 | 0.066774 | 0.178556 | 0.057152 | 0.039954 | 0.044402 | 0.116169 | 0.053631 |
| 0.103515 | 0.180756 | 0.159101 | 0.066769 | 0.178559 | 0.057151 | 0.039955 | 0.044396 | 0.116151 | 0.053648 |
| 0.103541 | 0.180729 | 0.159125 | 0.066767 | 0.17855 | 0.05714 | 0.039962 | 0.044402 | 0.116121 | 0.053662 |
| 0.10355 | 0.180712 | 0.159137 | 0.066773 | 0.178538 | 0.057147 | 0.039963 | 0.044403 | 0.116136 | 0.053641 |
| 0.103501 | 0.180773 | 0.159143 | 0.066788 | 0.178583 | 0.057145 | 0.039951 | 0.044395 | 0.116112 | 0.053609 |
| 0.103556 | 0.180732 | 0.159194 | 0.066772 | 0.178542 | 0.05714 | 0.039951 | 0.044388 | 0.116109 | 0.053616 |
| 0.103536 | 0.180719 | 0.159155 | 0.066772 | 0.178541 | 0.057133 | 0.039956 | 0.044397 | 0.11615 | 0.053641 |
| 0.103487 | 0.180786 | 0.159147 | 0.066782 | 0.178584 | 0.057148 | 0.039944 | 0.04439 | 0.116109 | 0.053623 |
| 0.103538 | 0.180805 | 0.159108 | 0.066764 | 0.178532 | 0.057148 | 0.039946 | 0.044393 | 0.116117 | 0.053649 |
| 0.103527 | 0.180797 | 0.159106 | 0.066762 | 0.178529 | 0.057137 | 0.039951 | 0.044394 | 0.116145 | 0.053653 |
| 0.10353 | 0.180762 | 0.159162 | 0.066766 | 0.178512 | 0.057144 | 0.039951 | 0.0444 | 0.116125 | 0.053647 |
| 0.103516 | 0.180725 | 0.159136 | 0.066789 | 0.178595 | 0.057151 | 0.039953 | 0.044391 | 0.116129 | 0.053615 |
| 0.103509 | 0.180716 | 0.159141 | 0.066783 | 0.178566 | 0.057134 | 0.039959 | 0.044397 | 0.116151 | 0.053645 |
| 0.103506 | 0.180794 | 0.15912 | 0.06677 | 0.17853 | 0.057135 | 0.039958 | 0.044399 | 0.116151 | 0.053637 |
| 0.103525 | 0.180767 | 0.159145 | 0.066764 | 0.178529 | 0.057142 | 0.039947 | 0.044391 | 0.116138 | 0.053652 |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 0.103545 | 0.18072 | 0.159211 | 0.066761 | 0.17853 | 0.057127 | 0.039948 | 0.044393 | 0.116146 | 0.053619 |
| 0.103531 | 0.180814 | 0.159138 | 0.06676 | 0.178501 | 0.057126 | 0.039949 | 0.044393 | 0.116148 | 0.053642 |
| 0.103526 | 0.180757 | 0.159121 | 0.066776 | 0.178548 | 0.057138 | 0.039963 | 0.044403 | 0.116136 | 0.05363 |
| 0.103557 | 0.180722 | 0.159143 | 0.066773 | 0.178535 | 0.057152 | 0.039957 | 0.044391 | 0.116111 | 0.053658 |
| 0.103526 | 0.180743 | 0.159163 | 0.066775 | 0.178519 | 0.057138 | 0.039953 | 0.044399 | 0.116163 | 0.053623 |
| 0.103554 | 0.180704 | 0.159182 | 0.066784 | 0.178579 | 0.057135 | 0.039948 | 0.044392 | 0.116106 | 0.053616 |
| 0.103489 | 0.180782 | 0.159119 | 0.066793 | 0.178591 | 0.057135 | 0.039946 | 0.044389 | 0.116143 | 0.053614 |
| 0.103514 | 0.180778 | 0.159134 | 0.066781 | 0.17854 | 0.057129 | 0.03995 | 0.044398 | 0.116152 | 0.053625 |
| 0.103532 | 0.180749 | 0.15915 | 0.066754 | 0.17851 | 0.057145 | 0.03996 | 0.044401 | 0.116147 | 0.053653 |
| 0.103542 | 0.180699 | 0.15918 | 0.066772 | 0.178536 | 0.057137 | 0.039957 | 0.044398 | 0.116144 | 0.053636 |
| 0.103526 | 0.180712 | 0.159169 | 0.066767 | 0.178569 | 0.057137 | 0.039952 | 0.044399 | 0.11613 | 0.053639 |
| 0.10353 | 0.180792 | 0.159112 | 0.06678 | 0.178537 | 0.057147 | 0.039955 | 0.044399 | 0.116132 | 0.053617 |
| 0.103537 | 0.180723 | 0.159185 | 0.066767 | 0.178553 | 0.057141 | 0.039954 | 0.044398 | 0.116114 | 0.053628 |
| 0.103542 | 0.180799 | 0.159137 | 0.06676 | 0.178528 | 0.057146 | 0.039945 | 0.044395 | 0.116103 | 0.053645 |
| 0.103512 | 0.180706 | 0.159152 | 0.066777 | 0.178601 | 0.05714 | 0.039951 | 0.044388 | 0.116114 | 0.053661 |
| 0.103525 | 0.18076 | 0.15919 | 0.066761 | 0.178538 | 0.057127 | 0.039953 | 0.044387 | 0.116137 | 0.053622 |
| 0.103543 | 0.180702 | 0.159197 | 0.06677 | 0.178583 | 0.057143 | 0.039956 | 0.044395 | 0.116098 | 0.053613 |
| 0.103523 | 0.180728 | 0.159175 | 0.066789 | 0.178607 | 0.057138 | 0.039943 | 0.044386 | 0.116096 | 0.053615 |
| 0.103516 | 0.180724 | 0.159132 | 0.066786 | 0.178603 | 0.057139 | 0.039948 | 0.044388 | 0.116109 | 0.053656 |
| 0.103534 | 0.180776 | 0.159159 | 0.066768 | 0.178528 | 0.057127 | 0.03995 | 0.044386 | 0.116132 | 0.053639 |
| 0.103534 | 0.180792 | 0.159139 | 0.066758 | 0.178511 | 0.057149 | 0.03995 | 0.044391 | 0.116129 | 0.053647 |
| 0.103507 | 0.180785 | 0.159146 | 0.066761 | 0.178521 | 0.057138 | 0.039961 | 0.044397 | 0.116127 | 0.053659 |
| 0.103518 | 0.180781 | 0.159154 | 0.066753 | 0.178517 | 0.057137 | 0.039951 | 0.044397 | 0.116136 | 0.053655 |
| 0.103495 | 0.180723 | 0.159119 | 0.066775 | 0.178567 | 0.057142 | 0.03996 | 0.044402 | 0.116156 | 0.053662 |
| 0.103516 | 0.180697 | 0.159112 | 0.06679 | 0.178606 | 0.057141 | 0.039955 | 0.044401 | 0.116136 | 0.053645 |
| 0.103535 | 0.180758 | 0.159106 | 0.066779 | 0.178584 | 0.057127 | 0.039947 | 0.044393 | 0.116123 | 0.053648 |
| 0.103488 | 0.18075 | 0.159112 | 0.066783 | 0.178589 | 0.057146 | 0.039953 | 0.044394 | 0.116135 | 0.05365 |
| 0.103553 | 0.180763 | 0.159109 | 0.066763 | 0.178528 | 0.057153 | 0.039958 | 0.044398 | 0.116127 | 0.053648 |
| 0.103504 | 0.180718 | 0.159147 | 0.066772 | 0.178543 | 0.05714 | 0.039961 | 0.044401 | 0.116157 | 0.053656 |
| 0.103507 | 0.18078 | 0.15918 | 0.06677 | 0.178581 | 0.057145 | 0.039944 | 0.044394 | 0.11609 | 0.05361 |
| 0.103498 | 0.180775 | 0.15916 | 0.06678 | 0.178553 | 0.057145 | 0.039948 | 0.044388 | 0.116133 | 0.053618 |
| 0.103506 | 0.180713 | 0.159135 | 0.066791 | 0.178585 | 0.057144 | 0.039956 | 0.044394 | 0.11613 | 0.053645 |
| 0.103525 | 0.180759 | 0.159178 | 0.066782 | 0.178553 | 0.057133 | 0.039944 | 0.044389 | 0.116108 | 0.053629 |
| 0.103505 | 0.18081 | 0.159094 | 0.066778 | 0.178568 | 0.057134 | 0.039957 | 0.044395 | 0.116132 | 0.053627 |
| 0.103499 | 0.180744 | 0.159169 | 0.066774 | 0.178528 | 0.057135 | 0.039958 | 0.0444 | 0.116139 | 0.053654 |
| 0.103514 | 0.180784 | 0.159117 | 0.066769 | 0.17858 | 0.057147 | 0.039949 | 0.044395 | 0.116124 | 0.053622 |
| 0.103497 | 0.180743 | 0.159114 | 0.066789 | 0.178579 | 0.057141 | 0.039959 | 0.04439 | 0.116138 | 0.053649 |
| 0.103509 | 0.180734 | 0.15917 | 0.066784 | 0.178592 | 0.057148 | 0.039953 | 0.044392 | 0.116101 | 0.053616 |
| 0.103532 | 0.180698 | 0.159143 | 0.066772 | 0.178558 | 0.057135 | 0.039957 | 0.044397 | 0.116151 | 0.053658 |
| 0.103518 | 0.180742 | 0.159108 | 0.06679 | 0.178601 | 0.057143 | 0.039948 | 0.044392 | 0.116138 | 0.053621 |
| 0.103509 | 0.180781 | 0.159136 | 0.066777 | 0.178572 | 0.057138 | 0.039948 | 0.044397 | 0.116138 | 0.053604 |
| 0.103516 | 0.180811 | 0.159117 | 0.066779 | 0.178528 | 0.057134 | 0.039949 | 0.044401 | 0.116137 | 0.053629 |
| 0.103515 | 0.180731 | 0.159116 | 0.066783 | 0.178603 | 0.057133 | 0.039954 | 0.044396 | 0.116136 | 0.053633 |
| 0.103508 | 0.180708 | 0.159139 | 0.066782 | 0.178585 | 0.057141 | 0.039963 | 0.044397 | 0.116119 | 0.053658 |
| 0.103514 | 0.180713 | 0.159114 | 0.066775 | 0.178562 | 0.057156 | 0.039957 | 0.044397 | 0.116142 | 0.05367 |
| 0.103497 | 0.180721 | 0.15913 | 0.066776 | 0.178583 | 0.057152 | 0.03996 | 0.044401 | 0.116154 | 0.053625 |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 0.103488 | 0.180761 | 0.159094 | 0.066781 | 0.178611 | 0.057159 | 0.039959 | 0.044394 | 0.116106 | 0.053647 |
| 0.103533 | 0.180794 | 0.159101 | 0.066775 | 0.17857 | 0.05713 | 0.039955 | 0.044396 | 0.116119 | 0.053628 |
| 0.103519 | 0.180776 | 0.159115 | 0.066781 | 0.178573 | 0.057149 | 0.039954 | 0.044396 | 0.116101 | 0.053636 |
| 0.103538 | 0.180807 | 0.159117 | 0.066755 | 0.178507 | 0.057128 | 0.039953 | 0.044396 | 0.116149 | 0.05365 |
| 0.103521 | 0.180751 | 0.15913 | 0.066778 | 0.178579 | 0.057139 | 0.039952 | 0.044394 | 0.116125 | 0.053631 |
| 0.103512 | 0.180717 | 0.159123 | 0.066783 | 0.178595 | 0.057152 | 0.039953 | 0.044398 | 0.116149 | 0.053619 |
| 0.103488 | 0.180789 | 0.159127 | 0.066783 | 0.178587 | 0.057144 | 0.039942 | 0.044392 | 0.116105 | 0.053644 |
| 0.103494 | 0.180727 | 0.159143 | 0.06678 | 0.178574 | 0.057143 | 0.039952 | 0.044398 | 0.116144 | 0.053645 |
| 0.10352 | 0.180768 | 0.159131 | 0.066772 | 0.178537 | 0.057136 | 0.039956 | 0.044402 | 0.116145 | 0.053635 |
| 0.10353 | 0.180757 | 0.159182 | 0.066763 | 0.178549 | 0.05714 | 0.039944 | 0.044393 | 0.116136 | 0.053607 |
| 0.103551 | 0.180797 | 0.159111 | 0.066763 | 0.178522 | 0.057131 | 0.039956 | 0.044397 | 0.11614 | 0.053632 |
| 0.103531 | 0.18073 | 0.159187 | 0.066775 | 0.178565 | 0.057133 | 0.039945 | 0.044392 | 0.116135 | 0.053606 |
| 0.1035 | 0.180759 | 0.159185 | 0.066785 | 0.178558 | 0.05713 | 0.039949 | 0.044393 | 0.116103 | 0.053638 |
| 0.103529 | 0.180705 | 0.159147 | 0.066776 | 0.178555 | 0.057139 | 0.039956 | 0.044401 | 0.116152 | 0.053638 |
| 0.103515 | 0.180769 | 0.159114 | 0.066779 | 0.178594 | 0.057136 | 0.039955 | 0.044396 | 0.116118 | 0.053624 |
| 0.103518 | 0.180739 | 0.159116 | 0.06679 | 0.178589 | 0.057152 | 0.039957 | 0.044397 | 0.116114 | 0.053629 |
| 0.103528 | 0.18071 | 0.159211 | 0.066772 | 0.178554 | 0.057134 | 0.039954 | 0.04439 | 0.116111 | 0.053637 |
| 0.103495 | 0.18076 | 0.159156 | 0.066775 | 0.178526 | 0.057148 | 0.03996 | 0.044399 | 0.116129 | 0.053651 |
| 0.103521 | 0.180815 | 0.159125 | 0.066762 | 0.178512 | 0.05713 | 0.039953 | 0.044393 | 0.116135 | 0.053653 |
| 0.103543 | 0.180741 | 0.159165 | 0.066773 | 0.178519 | 0.057131 | 0.039952 | 0.044389 | 0.116142 | 0.053645 |
| 0.103545 | 0.180706 | 0.159108 | 0.066771 | 0.178586 | 0.057132 | 0.039957 | 0.044394 | 0.116145 | 0.053655 |
| 0.103539 | 0.18073 | 0.159162 | 0.066771 | 0.17857 | 0.057148 | 0.039951 | 0.044391 | 0.116114 | 0.053624 |
| 0.103553 | 0.180684 | 0.159175 | 0.066783 | 0.178566 | 0.057142 | 0.039945 | 0.044387 | 0.116147 | 0.053619 |
| 0.103508 | 0.18071 | 0.159149 | 0.066777 | 0.178533 | 0.057157 | 0.039968 | 0.044398 | 0.116132 | 0.053669 |
| 0.103539 | 0.180709 | 0.15912 | 0.066789 | 0.17856 | 0.057159 | 0.039963 | 0.044405 | 0.116128 | 0.053627 |
| 0.103523 | 0.180769 | 0.159166 | 0.066773 | 0.178529 | 0.057149 | 0.039955 | 0.044397 | 0.116119 | 0.053619 |
| 0.103528 | 0.18076 | 0.159162 | 0.066766 | 0.178527 | 0.057152 | 0.039951 | 0.044393 | 0.116106 | 0.053655 |
| 0.103515 | 0.180748 | 0.159123 | 0.066785 | 0.178545 | 0.057153 | 0.039959 | 0.044397 | 0.116148 | 0.053626 |
| 0.103511 | 0.180769 | 0.159171 | 0.066771 | 0.178511 | 0.057132 | 0.039959 | 0.0444 | 0.116134 | 0.053641 |
| 0.103514 | 0.180779 | 0.159148 | 0.066777 | 0.178526 | 0.05713 | 0.039959 | 0.044398 | 0.116121 | 0.053648 |
| 0.10351 | 0.180804 | 0.159116 | 0.066781 | 0.178537 | 0.057145 | 0.039957 | 0.044399 | 0.116135 | 0.053616 |
| 0.10352 | 0.180766 | 0.159136 | 0.066774 | 0.17853 | 0.057141 | 0.03995 | 0.044392 | 0.116127 | 0.053664 |
| 0.103497 | 0.180747 | 0.159158 | 0.066778 | 0.178576 | 0.057138 | 0.039956 | 0.044394 | 0.116136 | 0.053619 |
| 0.103516 | 0.180778 | 0.159137 | 0.066768 | 0.178584 | 0.057148 | 0.039948 | 0.044395 | 0.116107 | 0.053619 |
| 0.10355 | 0.180771 | 0.159154 | 0.066764 | 0.178529 | 0.057129 | 0.039947 | 0.044389 | 0.11613 | 0.053636 |
| 0.103516 | 0.180797 | 0.159135 | 0.066766 | 0.178566 | 0.057139 | 0.039945 | 0.044396 | 0.116111 | 0.053628 |
| 0.103517 | 0.180797 | 0.159144 | 0.066765 | 0.178536 | 0.057129 | 0.039953 | 0.044397 | 0.116119 | 0.053642 |
| 0.103508 | 0.180737 | 0.159112 | 0.06679 | 0.178555 | 0.057136 | 0.039963 | 0.044396 | 0.116153 | 0.05365 |
| 0.103534 | 0.180777 | 0.159163 | 0.066764 | 0.178533 | 0.057144 | 0.039945 | 0.044387 | 0.116105 | 0.053648 |
| 0.103531 | 0.180751 | 0.1592 | 0.066779 | 0.17854 | 0.057148 | 0.039946 | 0.044386 | 0.116107 | 0.053612 |
| 0.10351 | 0.180763 | 0.159144 | 0.066773 | 0.178593 | 0.057139 | 0.039956 | 0.044393 | 0.116102 | 0.053627 |
| 0.103549 | 0.180723 | 0.159141 | 0.066769 | 0.17856 | 0.057138 | 0.03995 | 0.044398 | 0.116111 | 0.053661 |
| 0.103501 | 0.180747 | 0.159155 | 0.066769 | 0.178564 | 0.057146 | 0.039957 | 0.044391 | 0.116123 | 0.053647 |
| 0.103533 | 0.180714 | 0.15913 | 0.066771 | 0.178545 | 0.057152 | 0.039958 | 0.044399 | 0.116147 | 0.053653 |
| 0.103511 | 0.180727 | 0.159109 | 0.066785 | 0.17855 | 0.057155 | 0.039964 | 0.044397 | 0.116158 | 0.053644 |
| 0.103526 | 0.180784 | 0.159122 | 0.066765 | 0.178569 | 0.057147 | 0.039945 | 0.044392 | 0.116107 | 0.053643 |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 0.103499 | 0.18077 | 0.159125 | 0.06678 | 0.178553 | 0.057149 | 0.039955 | 0.044395 | 0.116126 | 0.053646 |
| 0.10352 | 0.180745 | 0.159165 | 0.066776 | 0.178596 | 0.05715 | 0.039942 | 0.044392 | 0.116103 | 0.053611 |
| 0.103539 | 0.180712 | 0.159185 | 0.066765 | 0.178532 | 0.057139 | 0.03995 | 0.044391 | 0.116141 | 0.053646 |
| 0.103533 | 0.180685 | 0.15913 | 0.066782 | 0.178583 | 0.057143 | 0.039958 | 0.044398 | 0.116144 | 0.053642 |
| 0.103549 | 0.180692 | 0.159119 | 0.066787 | 0.178561 | 0.057151 | 0.039956 | 0.044396 | 0.11614 | 0.053649 |
| 0.103492 | 0.180809 | 0.159103 | 0.066789 | 0.178558 | 0.057142 | 0.039951 | 0.044392 | 0.116155 | 0.053609 |
| 0.103549 | 0.180791 | 0.159139 | 0.066774 | 0.178544 | 0.057143 | 0.039951 | 0.044391 | 0.116099 | 0.053617 |
| 0.103533 | 0.180745 | 0.159145 | 0.066777 | 0.178529 | 0.057142 | 0.039962 | 0.044401 | 0.116133 | 0.053632 |
| 0.103485 | 0.180791 | 0.159172 | 0.066782 | 0.178575 | 0.057134 | 0.039941 | 0.044393 | 0.11611 | 0.053618 |
| 0.103503 | 0.180777 | 0.159178 | 0.066773 | 0.178522 | 0.05714 | 0.039945 | 0.044386 | 0.116131 | 0.053646 |
| 0.10351 | 0.180748 | 0.15912 | 0.066773 | 0.17858 | 0.05713 | 0.039949 | 0.044399 | 0.116136 | 0.053656 |
| 0.103517 | 0.180689 | 0.159174 | 0.066774 | 0.17854 | 0.057149 | 0.039962 | 0.044402 | 0.116156 | 0.053637 |
| 0.103543 | 0.18072 | 0.159161 | 0.066766 | 0.178551 | 0.057148 | 0.03995 | 0.044395 | 0.116117 | 0.053649 |
| 0.103539 | 0.180775 | 0.159137 | 0.066765 | 0.178539 | 0.057138 | 0.039949 | 0.044391 | 0.11612 | 0.053647 |
| 0.103532 | 0.180726 | 0.159186 | 0.066779 | 0.178558 | 0.057146 | 0.039954 | 0.044392 | 0.116102 | 0.053626 |
| 0.103554 | 0.180779 | 0.159182 | 0.066767 | 0.178505 | 0.057131 | 0.039955 | 0.044392 | 0.116116 | 0.053617 |
| 0.103542 | 0.18071 | 0.159159 | 0.066772 | 0.17857 | 0.057135 | 0.039953 | 0.04439 | 0.116131 | 0.053638 |
| 0.10352 | 0.180784 | 0.159091 | 0.066776 | 0.178539 | 0.057146 | 0.039953 | 0.044401 | 0.116132 | 0.053658 |
| 0.103545 | 0.180696 | 0.15914 | 0.06679 | 0.178582 | 0.057154 | 0.039955 | 0.044395 | 0.116122 | 0.053621 |
| 0.103534 | 0.180768 | 0.15913 | 0.066773 | 0.17854 | 0.057146 | 0.039953 | 0.044397 | 0.116104 | 0.053655 |
| 0.103501 | 0.180706 | 0.159163 | 0.066791 | 0.178579 | 0.057141 | 0.039962 | 0.044399 | 0.116125 | 0.053633 |
| 0.103533 | 0.180807 | 0.159102 | 0.066769 | 0.178565 | 0.057131 | 0.039941 | 0.044392 | 0.11614 | 0.053619 |
| 0.103519 | 0.180743 | 0.159137 | 0.066779 | 0.178553 | 0.057144 | 0.039959 | 0.044396 | 0.11613 | 0.05364 |
| 0.10352 | 0.18073 | 0.159145 | 0.066782 | 0.178555 | 0.05715 | 0.039961 | 0.044395 | 0.116121 | 0.053642 |
| 0.103539 | 0.180742 | 0.1591 | 0.066783 | 0.178561 | 0.057152 | 0.03996 | 0.044399 | 0.116133 | 0.053631 |
| 0.10348 | 0.180789 | 0.159085 | 0.066781 | 0.178585 | 0.05715 | 0.039955 | 0.044401 | 0.116147 | 0.053626 |
| 0.103494 | 0.180749 | 0.159129 | 0.066782 | 0.178591 | 0.057137 | 0.039957 | 0.044401 | 0.116139 | 0.053622 |
| 0.10355 | 0.180696 | 0.159148 | 0.066769 | 0.178591 | 0.057149 | 0.039948 | 0.044397 | 0.116099 | 0.053654 |
| 0.103513 | 0.180769 | 0.159126 | 0.066784 | 0.178603 | 0.057142 | 0.039942 | 0.044387 | 0.116123 | 0.053611 |
| 0.103522 | 0.180733 | 0.159178 | 0.066767 | 0.178562 | 0.057128 | 0.039956 | 0.044398 | 0.116131 | 0.053625 |
| 0.103562 | 0.180723 | 0.159182 | 0.066766 | 0.178518 | 0.057136 | 0.03996 | 0.044397 | 0.116135 | 0.053623 |
| 0.103505 | 0.180784 | 0.159109 | 0.066778 | 0.178603 | 0.057136 | 0.039942 | 0.044386 | 0.116104 | 0.053653 |
| 0.103534 | 0.180761 | 0.159116 | 0.06679 | 0.178566 | 0.057137 | 0.039954 | 0.044392 | 0.116138 | 0.053613 |
| 0.103545 | 0.180701 | 0.159184 | 0.06677 | 0.178555 | 0.057138 | 0.03995 | 0.044391 | 0.116144 | 0.053622 |
| 0.103513 | 0.180817 | 0.159173 | 0.066772 | 0.178529 | 0.057128 | 0.039945 | 0.044388 | 0.116106 | 0.053628 |
| 0.103514 | 0.180781 | 0.159115 | 0.06677 | 0.178567 | 0.057141 | 0.039955 | 0.044396 | 0.116113 | 0.053649 |
| 0.103524 | 0.180708 | 0.15911 | 0.066778 | 0.178589 | 0.057152 | 0.039957 | 0.044402 | 0.116137 | 0.053643 |
| 0.10353 | 0.180767 | 0.159138 | 0.066764 | 0.178545 | 0.05713 | 0.039954 | 0.0444 | 0.116149 | 0.053621 |
| 0.103517 | 0.18079 | 0.15913 | 0.066783 | 0.178574 | 0.057132 | 0.039941 | 0.044392 | 0.116124 | 0.053617 |
| 0.103517 | 0.180717 | 0.159138 | 0.066769 | 0.178563 | 0.057153 | 0.039959 | 0.044399 | 0.116131 | 0.053654 |
| 0.103507 | 0.180765 | 0.159179 | 0.066782 | 0.178573 | 0.057148 | 0.039955 | 0.044388 | 0.116096 | 0.053607 |
| 0.103555 | 0.180747 | 0.159209 | 0.066767 | 0.178513 | 0.057125 | 0.039948 | 0.044392 | 0.116129 | 0.053615 |
| 0.103551 | 0.180748 | 0.159142 | 0.066765 | 0.178546 | 0.05715 | 0.039952 | 0.044389 | 0.116107 | 0.05365 |
| 0.103545 | 0.1807 | 0.159125 | 0.066788 | 0.178564 | 0.057139 | 0.039961 | 0.044402 | 0.116152 | 0.053624 |
| 0.10354 | 0.180766 | 0.159176 | 0.066776 | 0.17855 | 0.057128 | 0.039939 | 0.044385 | 0.116124 | 0.053617 |
| 0.103521 | 0.180802 | 0.159162 | 0.066771 | 0.178541 | 0.057129 | 0.039954 | 0.044396 | 0.116103 | 0.053621 |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 0.103495 | 0.180736 | 0.159178 | 0.066779 | 0.178557 | 0.05713 | 0.03995 | 0.044394 | 0.116133 | 0.053647 |
| 0.103524 | 0.180741 | 0.159147 | 0.066772 | 0.178554 | 0.057144 | 0.039953 | 0.044391 | 0.116129 | 0.053645 |
| 0.103498 | 0.180811 | 0.159155 | 0.066775 | 0.17855 | 0.057127 | 0.039947 | 0.044387 | 0.116093 | 0.053655 |
| 0.103542 | 0.180729 | 0.159131 | 0.066768 | 0.178571 | 0.057137 | 0.039953 | 0.044393 | 0.116148 | 0.053627 |
| 0.103548 | 0.180717 | 0.159101 | 0.066787 | 0.178572 | 0.057146 | 0.039957 | 0.044396 | 0.116137 | 0.053637 |
| 0.103494 | 0.180796 | 0.15915 | 0.066767 | 0.178578 | 0.057132 | 0.039945 | 0.044391 | 0.1161 | 0.053646 |
| 0.103513 | 0.180775 | 0.15917 | 0.066762 | 0.178531 | 0.057143 | 0.039952 | 0.044394 | 0.116129 | 0.053631 |
| 0.103533 | 0.180692 | 0.15915 | 0.066781 | 0.178582 | 0.057141 | 0.039948 | 0.044394 | 0.116131 | 0.053648 |
| 0.103534 | 0.180719 | 0.159129 | 0.066774 | 0.178555 | 0.057149 | 0.039964 | 0.044395 | 0.116133 | 0.053648 |
| 0.103536 | 0.180736 | 0.159135 | 0.066785 | 0.178548 | 0.05714 | 0.039958 | 0.044397 | 0.116143 | 0.053623 |
| 0.103504 | 0.180723 | 0.159182 | 0.066784 | 0.178534 | 0.05714 | 0.039961 | 0.044395 | 0.116154 | 0.053622 |
| 0.103526 | 0.180794 | 0.15914 | 0.066759 | 0.178506 | 0.057143 | 0.039958 | 0.044396 | 0.116151 | 0.053628 |
| 0.103522 | 0.180774 | 0.159133 | 0.066786 | 0.178586 | 0.057128 | 0.039951 | 0.044391 | 0.116118 | 0.05361 |
| 0.103532 | 0.180699 | 0.159123 | 0.066785 | 0.178606 | 0.057146 | 0.039952 | 0.044388 | 0.116118 | 0.053651 |
| 0.103501 | 0.180754 | 0.159119 | 0.066775 | 0.178538 | 0.057135 | 0.039964 | 0.044402 | 0.116149 | 0.053663 |
| 0.103531 | 0.180744 | 0.159174 | 0.066784 | 0.178565 | 0.05713 | 0.039943 | 0.044391 | 0.11614 | 0.053597 |
| 0.103513 | 0.180739 | 0.159143 | 0.066766 | 0.178515 | 0.057156 | 0.039967 | 0.044406 | 0.116165 | 0.05363 |
| 0.103532 | 0.180779 | 0.159123 | 0.066767 | 0.178528 | 0.057153 | 0.039962 | 0.044399 | 0.116132 | 0.053625 |
| 0.103553 | 0.180716 | 0.159159 | 0.066773 | 0.178528 | 0.057131 | 0.039961 | 0.044398 | 0.116141 | 0.05364 |
| 0.103547 | 0.180706 | 0.159094 | 0.06679 | 0.1786 | 0.057148 | 0.039947 | 0.044395 | 0.116145 | 0.053629 |
| 0.103557 | 0.180713 | 0.159159 | 0.066774 | 0.178555 | 0.05713 | 0.039952 | 0.044393 | 0.116154 | 0.053613 |
| 0.103511 | 0.180776 | 0.159131 | 0.066778 | 0.178514 | 0.057138 | 0.039953 | 0.044395 | 0.116157 | 0.053648 |
| 0.103536 | 0.18071 | 0.159169 | 0.066765 | 0.178578 | 0.057148 | 0.039945 | 0.044393 | 0.116113 | 0.053643 |
| 0.103508 | 0.180737 | 0.159126 | 0.066787 | 0.178617 | 0.057136 | 0.039958 | 0.044391 | 0.116116 | 0.053626 |
| 0.10354 | 0.180741 | 0.159139 | 0.066774 | 0.178553 | 0.057149 | 0.039956 | 0.044399 | 0.11611 | 0.053639 |
| 0.103539 | 0.180727 | 0.159169 | 0.066762 | 0.178554 | 0.05715 | 0.039949 | 0.044394 | 0.116101 | 0.053655 |
| 0.103501 | 0.18074 | 0.159106 | 0.066773 | 0.178568 | 0.057137 | 0.03996 | 0.044395 | 0.116161 | 0.053659 |
| 0.103529 | 0.180765 | 0.159142 | 0.066763 | 0.178519 | 0.057148 | 0.039952 | 0.044401 | 0.116153 | 0.053628 |
| 0.103535 | 0.180746 | 0.159149 | 0.066775 | 0.178549 | 0.057138 | 0.039954 | 0.044391 | 0.116128 | 0.053636 |
| 0.103533 | 0.180736 | 0.159096 | 0.066794 | 0.178602 | 0.057149 | 0.039946 | 0.044398 | 0.11614 | 0.053606 |
| 0.103544 | 0.180767 | 0.159173 | 0.066755 | 0.178512 | 0.057144 | 0.039954 | 0.044401 | 0.116133 | 0.053616 |
| 0.103534 | 0.180716 | 0.159176 | 0.066766 | 0.178553 | 0.057137 | 0.039951 | 0.04439 | 0.116133 | 0.053646 |
| 0.103508 | 0.180731 | 0.159151 | 0.066782 | 0.178585 | 0.057152 | 0.039953 | 0.044394 | 0.116103 | 0.05364 |
| 0.103497 | 0.180725 | 0.159105 | 0.066798 | 0.178605 | 0.057139 | 0.039961 | 0.044392 | 0.116113 | 0.053664 |
| 0.103511 | 0.180785 | 0.159121 | 0.066794 | 0.178603 | 0.057125 | 0.039941 | 0.044386 | 0.116124 | 0.05361 |
| 0.103503 | 0.180726 | 0.159198 | 0.066774 | 0.178553 | 0.057137 | 0.039954 | 0.044391 | 0.116112 | 0.053652 |
| 0.103501 | 0.180815 | 0.159114 | 0.066787 | 0.178555 | 0.057143 | 0.039959 | 0.0444 | 0.116107 | 0.053618 |
| 0.103523 | 0.180703 | 0.15918 | 0.066772 | 0.178551 | 0.057134 | 0.039957 | 0.044393 | 0.116145 | 0.053641 |
| 0.103544 | 0.180784 | 0.159149 | 0.066766 | 0.178526 | 0.057142 | 0.039946 | 0.044395 | 0.116131 | 0.053618 |
| 0.103555 | 0.180723 | 0.159128 | 0.066777 | 0.178542 | 0.057145 | 0.039952 | 0.044395 | 0.116161 | 0.053622 |
| 0.103505 | 0.180749 | 0.159148 | 0.066782 | 0.17859 | 0.057151 | 0.039957 | 0.044397 | 0.1161 | 0.053622 |
| 0.103529 | 0.180767 | 0.159146 | 0.066768 | 0.178545 | 0.057152 | 0.039953 | 0.044401 | 0.116112 | 0.053628 |
| 0.103543 | 0.18067 | 0.159178 | 0.066783 | 0.178566 | 0.057135 | 0.039955 | 0.044392 | 0.116148 | 0.05363 |
| 0.103498 | 0.180737 | 0.159124 | 0.066788 | 0.178575 | 0.057139 | 0.039957 | 0.044391 | 0.116148 | 0.053644 |
| 0.103537 | 0.180719 | 0.15911 | 0.066785 | 0.1786 | 0.057142 | 0.039948 | 0.044388 | 0.116113 | 0.053657 |
| 0.103491 | 0.180801 | 0.159111 | 0.06678 | 0.17856 | 0.057136 | 0.039957 | 0.044394 | 0.116127 | 0.053643 |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 0.103549 | 0.180787 | 0.159122 | 0.066766 | 0.178549 | 0.05714 | 0.039945 | 0.044391 | 0.116106 | 0.053645 |
| 0.103525 | 0.180693 | 0.159177 | 0.066766 | 0.178557 | 0.057153 | 0.039961 | 0.044401 | 0.116134 | 0.053632 |
| 0.103553 | 0.180728 | 0.159163 | 0.066766 | 0.178553 | 0.057138 | 0.039949 | 0.044393 | 0.116132 | 0.053625 |
| 0.103536 | 0.180716 | 0.159159 | 0.066777 | 0.178555 | 0.057139 | 0.039956 | 0.044395 | 0.116137 | 0.053629 |
| 0.103536 | 0.180718 | 0.159161 | 0.066771 | 0.178566 | 0.057126 | 0.039948 | 0.044387 | 0.116141 | 0.053647 |
| 0.103494 | 0.180761 | 0.159151 | 0.066779 | 0.178549 | 0.057133 | 0.039959 | 0.044394 | 0.116146 | 0.053632 |
| 0.103501 | 0.180792 | 0.159121 | 0.066774 | 0.178532 | 0.057152 | 0.039961 | 0.044401 | 0.116107 | 0.053659 |
| 0.103511 | 0.180766 | 0.159115 | 0.066783 | 0.178567 | 0.057143 | 0.03995 | 0.044396 | 0.116112 | 0.053657 |
| 0.103485 | 0.180768 | 0.159159 | 0.066779 | 0.178573 | 0.057127 | 0.039953 | 0.044396 | 0.116118 | 0.053642 |
| 0.103534 | 0.18074 | 0.159128 | 0.066768 | 0.178587 | 0.057138 | 0.039946 | 0.044392 | 0.116113 | 0.053653 |
| 0.103546 | 0.180786 | 0.159165 | 0.066763 | 0.178535 | 0.05713 | 0.039947 | 0.044391 | 0.116112 | 0.053626 |
| 0.103528 | 0.180759 | 0.159168 | 0.066773 | 0.178531 | 0.05714 | 0.039957 | 0.044403 | 0.116118 | 0.053624 |
| 0.103502 | 0.180813 | 0.159164 | 0.066776 | 0.178522 | 0.057142 | 0.039952 | 0.044395 | 0.11611 | 0.053623 |
| 0.103521 | 0.180762 | 0.159103 | 0.06678 | 0.178574 | 0.05715 | 0.039955 | 0.0444 | 0.116147 | 0.053609 |
| 0.10354 | 0.180687 | 0.159126 | 0.066772 | 0.178595 | 0.057156 | 0.039959 | 0.044399 | 0.116119 | 0.053648 |
| 0.103521 | 0.180761 | 0.15915 | 0.066777 | 0.178566 | 0.057125 | 0.039952 | 0.044386 | 0.116115 | 0.053647 |
| 0.103485 | 0.180775 | 0.159169 | 0.06678 | 0.178587 | 0.057144 | 0.039945 | 0.044386 | 0.116098 | 0.05363 |
| 0.10352 | 0.180743 | 0.159133 | 0.06678 | 0.178566 | 0.057135 | 0.039952 | 0.044393 | 0.116143 | 0.053635 |
| 0.103539 | 0.180693 | 0.15919 | 0.066764 | 0.178555 | 0.057154 | 0.03996 | 0.044395 | 0.116126 | 0.053625 |
| 0.103515 | 0.180774 | 0.159168 | 0.066782 | 0.178556 | 0.057127 | 0.03994 | 0.044389 | 0.116136 | 0.053612 |
| 0.103548 | 0.180687 | 0.15918 | 0.066771 | 0.178588 | 0.057134 | 0.039951 | 0.044394 | 0.116107 | 0.05364 |
| 0.103497 | 0.180735 | 0.159151 | 0.066787 | 0.178589 | 0.057148 | 0.039948 | 0.044395 | 0.116125 | 0.053625 |
| 0.103541 | 0.180798 | 0.159116 | 0.06676 | 0.1785 | 0.057146 | 0.039959 | 0.044398 | 0.116132 | 0.053651 |
| 0.103517 | 0.180752 | 0.159195 | 0.066762 | 0.178558 | 0.05715 | 0.039952 | 0.044392 | 0.116095 | 0.053627 |
| 0.103507 | 0.180699 | 0.159188 | 0.066775 | 0.178582 | 0.057147 | 0.039954 | 0.044393 | 0.116137 | 0.053618 |
| 0.1035 | 0.180744 | 0.159182 | 0.066764 | 0.178525 | 0.057151 | 0.039962 | 0.044399 | 0.116118 | 0.053656 |
| 0.10354 | 0.180774 | 0.159179 | 0.066763 | 0.178511 | 0.057147 | 0.039955 | 0.044395 | 0.116106 | 0.05363 |
| 0.10352 | 0.180729 | 0.159152 | 0.066777 | 0.178566 | 0.057141 | 0.039957 | 0.044399 | 0.116132 | 0.053628 |
| 0.103522 | 0.180804 | 0.159184 | 0.06675 | 0.178495 | 0.057127 | 0.039946 | 0.044397 | 0.116137 | 0.053639 |
| 0.103511 | 0.180736 | 0.159183 | 0.066777 | 0.178554 | 0.057143 | 0.039944 | 0.044386 | 0.116135 | 0.053632 |
| 0.103517 | 0.180734 | 0.159105 | 0.066772 | 0.178541 | 0.05715 | 0.039955 | 0.044401 | 0.116159 | 0.053667 |
| 0.103516 | 0.180722 | 0.159182 | 0.066784 | 0.17857 | 0.05714 | 0.039952 | 0.044391 | 0.116128 | 0.053615 |
| 0.103513 | 0.180785 | 0.15913 | 0.066764 | 0.178556 | 0.057149 | 0.039947 | 0.044397 | 0.116118 | 0.053642 |
| 0.103549 | 0.18077 | 0.159147 | 0.066761 | 0.1785 | 0.057146 | 0.039947 | 0.044398 | 0.116133 | 0.053648 |
| 0.103511 | 0.180756 | 0.159185 | 0.066774 | 0.178572 | 0.057124 | 0.039945 | 0.044393 | 0.116129 | 0.05361 |
| 0.10356 | 0.180707 | 0.159195 | 0.066769 | 0.178518 | 0.057133 | 0.039949 | 0.044387 | 0.116128 | 0.053654 |
| 0.103499 | 0.180785 | 0.159187 | 0.066762 | 0.178553 | 0.057125 | 0.039946 | 0.04439 | 0.116107 | 0.053645 |
| 0.103544 | 0.180738 | 0.159128 | 0.066782 | 0.178561 | 0.057148 | 0.039952 | 0.044392 | 0.116116 | 0.05364 |
| 0.103536 | 0.180767 | 0.159124 | 0.066766 | 0.178554 | 0.057138 | 0.039956 | 0.04439 | 0.116124 | 0.053644 |
| 0.103536 | 0.180789 | 0.159117 | 0.066766 | 0.17857 | 0.057138 | 0.039942 | 0.044388 | 0.116124 | 0.05363 |
| 0.103541 | 0.18072 | 0.159149 | 0.066784 | 0.178548 | 0.05714 | 0.039954 | 0.044392 | 0.11615 | 0.053623 |
| 0.103497 | 0.180763 | 0.159148 | 0.06678 | 0.17854 | 0.057153 | 0.039961 | 0.044394 | 0.116121 | 0.053643 |
| 0.103533 | 0.180711 | 0.159143 | 0.066771 | 0.178536 | 0.057149 | 0.039961 | 0.044405 | 0.11615 | 0.053641 |
| 0.103526 | 0.180676 | 0.159177 | 0.066779 | 0.178574 | 0.057143 | 0.03996 | 0.044398 | 0.116142 | 0.053626 |
| 0.103491 | 0.180798 | 0.159157 | 0.066765 | 0.178571 | 0.057147 | 0.039948 | 0.044387 | 0.116096 | 0.053639 |
| 0.103484 | 0.180743 | 0.159162 | 0.066771 | 0.178583 | 0.057139 | 0.03996 | 0.044399 | 0.116107 | 0.053652 |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 0.103516 | 0.180792 | 0.159129 | 0.066784 | 0.178549 | 0.057135 | 0.039955 | 0.044401 | 0.116115 | 0.053623 |
| 0.103548 | 0.180716 | 0.159109 | 0.066778 | 0.178553 | 0.05715 | 0.039949 | 0.04439 | 0.116148 | 0.053659 |
| 0.103519 | 0.180763 | 0.159091 | 0.066777 | 0.178566 | 0.057145 | 0.039951 | 0.044394 | 0.116143 | 0.053652 |
| 0.103547 | 0.180692 | 0.159163 | 0.066772 | 0.178562 | 0.057151 | 0.039958 | 0.044398 | 0.116125 | 0.053633 |
| 0.103517 | 0.180775 | 0.159158 | 0.066763 | 0.178537 | 0.057143 | 0.03995 | 0.044389 | 0.116115 | 0.053653 |
| 0.103545 | 0.1807 | 0.159132 | 0.066774 | 0.178572 | 0.057146 | 0.039959 | 0.044394 | 0.116155 | 0.053625 |
| 0.103525 | 0.180773 | 0.159098 | 0.066783 | 0.178558 | 0.057151 | 0.039947 | 0.044393 | 0.116134 | 0.053637 |
| 0.103532 | 0.180798 | 0.159098 | 0.066773 | 0.17859 | 0.057141 | 0.039945 | 0.044394 | 0.116088 | 0.053642 |
| 0.1035 | 0.180774 | 0.159144 | 0.06678 | 0.178533 | 0.057135 | 0.039955 | 0.044397 | 0.116152 | 0.053629 |
| 0.103503 | 0.18072 | 0.15917 | 0.066781 | 0.178583 | 0.057152 | 0.039955 | 0.04439 | 0.116126 | 0.053621 |
| 0.103533 | 0.180733 | 0.159169 | 0.066768 | 0.178569 | 0.05713 | 0.039951 | 0.044388 | 0.11614 | 0.053618 |
| 0.103524 | 0.180695 | 0.1592 | 0.066764 | 0.178548 | 0.057136 | 0.039954 | 0.044393 | 0.116145 | 0.053641 |
| 0.103488 | 0.180746 | 0.15917 | 0.066788 | 0.178585 | 0.057149 | 0.039956 | 0.044389 | 0.116098 | 0.05363 |
| 0.103504 | 0.180798 | 0.159125 | 0.066777 | 0.178574 | 0.057144 | 0.039944 | 0.044396 | 0.116111 | 0.053626 |
| 0.103506 | 0.180771 | 0.159119 | 0.06678 | 0.17853 | 0.057131 | 0.039958 | 0.044393 | 0.116156 | 0.053656 |
| 0.103513 | 0.18074 | 0.159107 | 0.066781 | 0.178588 | 0.057149 | 0.03995 | 0.044392 | 0.116128 | 0.053652 |
| 0.103497 | 0.180755 | 0.159145 | 0.066795 | 0.178606 | 0.057142 | 0.039946 | 0.044386 | 0.116113 | 0.053615 |
| 0.103493 | 0.18075 | 0.159098 | 0.066783 | 0.178602 | 0.057153 | 0.039947 | 0.044398 | 0.116122 | 0.053654 |
| 0.103525 | 0.180748 | 0.159103 | 0.066778 | 0.178585 | 0.057142 | 0.039953 | 0.044399 | 0.116155 | 0.053613 |
| 0.10354 | 0.180717 | 0.159191 | 0.06677 | 0.178561 | 0.057141 | 0.03995 | 0.04439 | 0.116128 | 0.053614 |
| 0.103517 | 0.180744 | 0.159157 | 0.066766 | 0.178518 | 0.057146 | 0.03996 | 0.044399 | 0.116162 | 0.053631 |
| 0.103552 | 0.180749 | 0.159175 | 0.066769 | 0.178513 | 0.057151 | 0.039958 | 0.044396 | 0.116112 | 0.053624 |
| 0.103513 | 0.180743 | 0.159137 | 0.066771 | 0.178547 | 0.057147 | 0.039962 | 0.044402 | 0.116128 | 0.053648 |
| 0.103506 | 0.180746 | 0.159144 | 0.066773 | 0.178587 | 0.057137 | 0.039948 | 0.044388 | 0.116116 | 0.053655 |
| 0.103519 | 0.180758 | 0.15915 | 0.066769 | 0.178559 | 0.057143 | 0.039956 | 0.04439 | 0.116099 | 0.053657 |
| 0.103536 | 0.180777 | 0.159177 | 0.066769 | 0.178561 | 0.057139 | 0.039938 | 0.044384 | 0.116116 | 0.053603 |
| 0.103507 | 0.180748 | 0.159188 | 0.066769 | 0.178525 | 0.057128 | 0.039951 | 0.044398 | 0.116156 | 0.053629 |
| 0.103553 | 0.18078 | 0.159139 | 0.066752 | 0.178502 | 0.057138 | 0.039957 | 0.044397 | 0.116121 | 0.053662 |
| 0.103531 | 0.18081 | 0.159155 | 0.066765 | 0.178533 | 0.057132 | 0.039945 | 0.04439 | 0.116111 | 0.053628 |
| 0.103502 | 0.180734 | 0.159122 | 0.066793 | 0.1786 | 0.057145 | 0.03996 | 0.0444 | 0.116121 | 0.053624 |
| 0.103544 | 0.18078 | 0.159179 | 0.066776 | 0.178525 | 0.057137 | 0.039946 | 0.044388 | 0.116114 | 0.053611 |
| 0.103506 | 0.180788 | 0.159115 | 0.066782 | 0.1786 | 0.057132 | 0.039946 | 0.044387 | 0.116124 | 0.05362 |
| 0.103534 | 0.180725 | 0.159176 | 0.066772 | 0.178586 | 0.057126 | 0.039945 | 0.044389 | 0.116113 | 0.053635 |
| 0.103555 | 0.18075 | 0.159146 | 0.066784 | 0.178558 | 0.057139 | 0.039945 | 0.04439 | 0.116118 | 0.053615 |
| 0.103526 | 0.180736 | 0.15913 | 0.066783 | 0.178587 | 0.057133 | 0.039947 | 0.04439 | 0.11613 | 0.053638 |
| 0.103529 | 0.180734 | 0.159131 | 0.066781 | 0.178584 | 0.057139 | 0.039956 | 0.044393 | 0.116115 | 0.053638 |
| 0.103497 | 0.180803 | 0.159104 | 0.066776 | 0.178594 | 0.057143 | 0.039944 | 0.044394 | 0.116112 | 0.053633 |
| 0.103512 | 0.18075 | 0.159176 | 0.06676 | 0.178542 | 0.057127 | 0.039954 | 0.044391 | 0.116129 | 0.053659 |
| 0.103538 | 0.180714 | 0.159195 | 0.066783 | 0.178538 | 0.057149 | 0.039949 | 0.044388 | 0.116127 | 0.053616 |
| 0.103499 | 0.180748 | 0.159151 | 0.066779 | 0.178565 | 0.057137 | 0.039955 | 0.044401 | 0.116136 | 0.053629 |
| 0.10351 | 0.180788 | 0.15918 | 0.066771 | 0.178536 | 0.057135 | 0.039949 | 0.044392 | 0.116113 | 0.053627 |
| 0.103525 | 0.180696 | 0.159161 | 0.06678 | 0.1786 | 0.057149 | 0.039956 | 0.04439 | 0.116097 | 0.053646 |
| 0.103492 | 0.180764 | 0.1591 | 0.066791 | 0.178602 | 0.057143 | 0.039949 | 0.044398 | 0.116117 | 0.053645 |
| 0.103507 | 0.180798 | 0.159108 | 0.066778 | 0.178581 | 0.057138 | 0.039949 | 0.044392 | 0.116132 | 0.053619 |
| 0.103534 | 0.180786 | 0.159114 | 0.066759 | 0.178531 | 0.05715 | 0.03996 | 0.044398 | 0.116112 | 0.053656 |
| 0.103541 | 0.180721 | 0.159147 | 0.066767 | 0.178543 | 0.057155 | 0.039952 | 0.044394 | 0.116145 | 0.053635 |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 0.103523 | 0.180797 | 0.159159 | 0.066758 | 0.178532 | 0.057124 | 0.039942 | 0.044396 | 0.116144 | 0.053623 |
| 0.103554 | 0.180742 | 0.159184 | 0.066775 | 0.178547 | 0.057128 | 0.039953 | 0.044397 | 0.116108 | 0.053612 |
| 0.103529 | 0.180764 | 0.159113 | 0.066781 | 0.178587 | 0.05714 | 0.039948 | 0.044393 | 0.116117 | 0.053629 |
| 0.103519 | 0.180706 | 0.159186 | 0.066789 | 0.178562 | 0.057132 | 0.03996 | 0.044392 | 0.116137 | 0.053618 |
| 0.103532 | 0.180769 | 0.159127 | 0.066775 | 0.178546 | 0.057135 | 0.03995 | 0.044398 | 0.116136 | 0.053632 |
| 0.103544 | 0.180695 | 0.159163 | 0.066777 | 0.178534 | 0.057129 | 0.039956 | 0.044397 | 0.116155 | 0.053652 |
| 0.103544 | 0.180715 | 0.159096 | 0.066786 | 0.178593 | 0.057135 | 0.039953 | 0.044394 | 0.116139 | 0.053645 |
| 0.103543 | 0.180701 | 0.159135 | 0.066783 | 0.178549 | 0.057135 | 0.039956 | 0.044399 | 0.116148 | 0.05365 |
| 0.103517 | 0.180729 | 0.15918 | 0.066775 | 0.178554 | 0.057143 | 0.039953 | 0.044398 | 0.116136 | 0.053615 |
| 0.103545 | 0.180784 | 0.159163 | 0.066769 | 0.178538 | 0.057137 | 0.039945 | 0.044393 | 0.116104 | 0.053621 |
| 0.103544 | 0.180735 | 0.159181 | 0.066767 | 0.178538 | 0.05714 | 0.039946 | 0.044393 | 0.116126 | 0.053629 |
| 0.103535 | 0.180793 | 0.159132 | 0.066772 | 0.178533 | 0.057151 | 0.039954 | 0.044396 | 0.116121 | 0.053613 |
| 0.103523 | 0.180764 | 0.159159 | 0.066771 | 0.178541 | 0.057148 | 0.03996 | 0.044399 | 0.116108 | 0.053627 |
| 0.103513 | 0.180713 | 0.159128 | 0.066793 | 0.178588 | 0.057139 | 0.039959 | 0.044401 | 0.116116 | 0.053651 |
| 0.103526 | 0.180712 | 0.15913 | 0.066786 | 0.178561 | 0.057151 | 0.039958 | 0.044402 | 0.116155 | 0.053619 |
| 0.103509 | 0.18079 | 0.159191 | 0.066762 | 0.17854 | 0.057132 | 0.039949 | 0.044386 | 0.116102 | 0.053639 |
| 0.103536 | 0.180776 | 0.159113 | 0.066782 | 0.17858 | 0.057144 | 0.039952 | 0.044394 | 0.116117 | 0.053606 |
| 0.103506 | 0.180788 | 0.159166 | 0.066778 | 0.178564 | 0.05714 | 0.039944 | 0.044391 | 0.116104 | 0.053619 |
| 0.103551 | 0.180724 | 0.159172 | 0.066775 | 0.178589 | 0.057135 | 0.039946 | 0.044386 | 0.116108 | 0.053612 |
| 0.103527 | 0.18081 | 0.159094 | 0.066769 | 0.178578 | 0.05715 | 0.039943 | 0.044391 | 0.116105 | 0.053634 |
| 0.10354 | 0.180768 | 0.159146 | 0.066764 | 0.178543 | 0.057146 | 0.039945 | 0.044391 | 0.116136 | 0.053622 |
| 0.103553 | 0.18077 | 0.159171 | 0.066773 | 0.178534 | 0.057141 | 0.039949 | 0.044391 | 0.116102 | 0.053615 |
| 0.103553 | 0.180764 | 0.15914 | 0.066767 | 0.178508 | 0.057134 | 0.039955 | 0.044398 | 0.116151 | 0.053631 |
| 0.103542 | 0.180791 | 0.159114 | 0.066768 | 0.178535 | 0.057143 | 0.039955 | 0.044393 | 0.116119 | 0.05364 |
| 0.103497 | 0.180755 | 0.15914 | 0.066781 | 0.17855 | 0.057151 | 0.039956 | 0.044401 | 0.116146 | 0.053623 |
| 0.103546 | 0.18073 | 0.159195 | 0.066761 | 0.178538 | 0.057133 | 0.039944 | 0.04439 | 0.116121 | 0.053642 |
| 0.103518 | 0.180701 | 0.159193 | 0.066776 | 0.178596 | 0.057135 | 0.039955 | 0.044396 | 0.116106 | 0.053624 |
| 0.103516 | 0.180698 | 0.159206 | 0.066791 | 0.17861 | 0.057129 | 0.039945 | 0.044389 | 0.116108 | 0.053609 |
| 0.103528 | 0.180707 | 0.159187 | 0.066776 | 0.178571 | 0.05715 | 0.039947 | 0.04439 | 0.116114 | 0.053629 |
| 0.103543 | 0.180726 | 0.159157 | 0.066772 | 0.178551 | 0.05714 | 0.039945 | 0.044395 | 0.116152 | 0.05362 |
| 0.10354 | 0.180707 | 0.159205 | 0.066765 | 0.178532 | 0.05715 | 0.039951 | 0.044397 | 0.116119 | 0.053634 |
| 0.103505 | 0.180797 | 0.15917 | 0.066771 | 0.178532 | 0.05714 | 0.039946 | 0.044397 | 0.116139 | 0.053604 |
| 0.103505 | 0.180708 | 0.159158 | 0.066787 | 0.178593 | 0.05714 | 0.039949 | 0.044386 | 0.116135 | 0.053639 |
| 0.103511 | 0.180828 | 0.159122 | 0.066764 | 0.178532 | 0.057134 | 0.03995 | 0.0444 | 0.116137 | 0.053621 |
| 0.10353 | 0.180721 | 0.159128 | 0.066773 | 0.178548 | 0.057156 | 0.039955 | 0.044396 | 0.116154 | 0.053639 |
| 0.103507 | 0.180745 | 0.159152 | 0.066775 | 0.178551 | 0.057149 | 0.039958 | 0.044401 | 0.116126 | 0.053636 |

## Appendix 4-No_of_hidden_neurons

```
sweep = [1:1:30];                        % parameter values to test
squarederror = zeros(length(sweep), 1); % pre-allocation
models = cell(length(sweep), 1);        % pre-allocation
x = PortfolioOptin';                     % inputs
t = PortfolioOptout';                    % targets
trainFcn = 'trainscg';
perfomFcn='mse';

for i = 1:length(sweep)
    hiddenLayerSize = sweep(i);          % number of hidden layer neurons
    net = patternnet(hiddenLayerSize,trainFcn,perfomFcn);  % pattern
recognition network
    net.divideParam.trainRatio = 70/100;% 70% of data for training
    net.divideParam.valRatio = 15/100;  % 15% of data for validation
    net.divideParam.testRatio = 15/100; % 15% of data for testing
    net = train(net, x, t);             % train the network
    models{i} = net;                    % store the trained network
    p = net(Xtest');                    % predictions
    [~, p] = max(p);                    % predicted labels
    error(i) = sum(abs(Ytest-p'));      % Sum of error

end

figure
plot(sweep, error, '.-')
xlabel('number of hidden neurons')
ylabel('sum of error')
title('Number of hidden neurons vs. error')
```

## Appendix 5-Advanced script file

```
% Solve an Input-Output Fitting problem with a Neural Network
% Script generated by Neural Fitting app
% Created 23-Oct-2016 18:20:52
%
% This script assumes these variables are defined:
%
%   PortfolioOptin - input data.
%   PortfolioOptout - target data.

x = PortfolioOptin';
t = PortfolioOptout';

% Choose a Training Function
% For a list of all training functions type: help nntrain
% 'trainlm' is usually fastest.
% 'trainbr' takes longer but may be better for challenging problems.
% 'trainscg' uses less memory. Suitable in low memory situations.
trainFcn = 'trainscg';  % Scaled conjugate gradient backpropagation.

% Create a Fitting Network
hiddenLayerSize = 13;
net = fitnet(hiddenLayerSize,trainFcn);

% Choose Input and Output Pre/Post-Processing Functions
% For a list of all processing functions type: help nnprocess
net.input.processFcns = {'removeconstantrows','mapminmax'};
net.output.processFcns = {'removeconstantrows','mapminmax'};

% Setup Division of Data for Training, Validation, Testing
% For a list of all data division functions type: help nndivide
net.divideFcn = 'dividerand';  % Divide data randomly
net.divideMode = 'sample';  % Divide up every sample
net.divideParam.trainRatio = 70/100;
net.divideParam.valRatio = 15/100;
net.divideParam.testRatio = 15/100;

% Choose a Performance Function
% For a list of all performance functions type: help nnperformance
net.performFcn = 'mse';  % Mean Squared Error

% Choose Plot Functions
% For a list of all plot functions type: help nnplot
net.plotFcns = {'plotperform','plottrainstate','ploterrhist', ...
    'plotregression', 'plotfit'};

% Train the Network
[net,tr] = train(net,x,t);

% Test the Network
y = net(x);
e = gsubtract(t,y);
performance = perform(net,t,y)

% Recalculate Training, Validation and Test Performance
trainTargets = t .* tr.trainMask{1};
```

```
valTargets = t .* tr.valMask{1};
testTargets = t .* tr.testMask{1};
trainPerformance = perform(net,trainTargets,y)
valPerformance = perform(net,valTargets,y)
testPerformance = perform(net,testTargets,y)

% View the Network
view(net)

% Plots
% Uncomment these lines to enable various plots.
figure, plotperform(tr)
figure, plottrainstate(tr)
figure, ploterrhist(e)
figure, plotregression(t,y)
%figure, plotfit(net,x,t)

% Deployment
% Change the (false) values to (true) to enable the following code blocks.
% See the help for each generation function for more information.
if (false)
    % Generate MATLAB function for neural network for application
    % deployment in MATLAB scripts or with MATLAB Compiler and Builder
    % tools, or simply to examine the calculations your trained neural
    % network performs.
    genFunction(net,'myNeuralNetworkFunction');
    y = myNeuralNetworkFunction(x);
end
if (true)
    % Generate a matrix-only MATLAB function for neural network code
    % generation with MATLAB Coder tools.
    genFunction(net,'myNeuralNetworkFunction','MatrixOnly','yes');
    y = myNeuralNetworkFunction(x);
end
if (false)
    % Generate a Simulink diagram for simulation or deployment with.
    % Simulink Coder tools.
    gensim(net);
end
```

## Appendix 6-MyNeuralNetworkfunction

```
function [y1] = myNeuralNetworkFunction13n(x1)
%MYNEURALNETWORKFUNCTION neural network simulation function.
%
% Generated by Neural Network Toolbox function genFunction, 05-Nov-2016
11:30:36.
%
% [y1] = myNeuralNetworkFunction(x1) takes these arguments:
%   x = 10xQ matrix, input #1
% and returns:
%   y = 10xQ matrix, output #1
% where Q is the number of samples.

%#ok<*RPMT0>

% ===== NEURAL NETWORK CONSTANTS =====

% Input 1
x1_step1.xoffset = [-0.999852132845892;-0.999734912909109;-
0.998190802293883;-0.996606791881713;-0.999695500848871;-0.99920578818652;-
0.999903310884981;-0.993389700562756;-0.997459674664558;-0.99868850769668];
x1_step1.gain =
[1.00062535300866;1.00177967874599;1.00184093952136;1.0053396197367;1.0005867
9153159;1.00055813125213;1.00092443992081;1.00431069442484;1.00282513870305;1
.00125017918711];
x1_step1.ymin = -1;

% Layer 1
b1 = [-1.9301290161547824;-
1.5951933754694749;1.2612172468014131;1.2443501052252852;0.52644604316175314;
0.014389445380966263;0.31398801486371941;-
1.0280315884506439;0.49842616486732749;-
1.4007827201683862;1.281363410781869;2.0609985606072647;1.9164235641978995];
IW1_1 = [0.29348284639179167 -0.21291855286245409 -0.66185515745513712
0.12685006658852355 -0.48115830106050461 0.95104740983241265 -
0.059568346402238771 0.61895964181236141 -0.35492026951937666
0.60670475777087474;0.18740119938931935 -0.29740488347284655 -
0.59561299738772566 0.137351220751314 0.9788755306506203 8
0.62134212359827345 -0.64624961607580989 0.21145041419155053 -
0.70051617627533991 -0.18921995025721156;-0.34488575211055422
0.74062579753043278 0.35583744689100977 -0.63135506353216475
0.50269802755808968 -0.39441928998294495 -0.25561458233642154
0.81006584848309O6 0.49591070466284892 0.071964157890781186;-
0.12724618095347265 0.083808181086462832 -0.60023444847850249
0.12216482637419 16 0.27017175243833397 0.54801949588699439 -
0.90422391594681584 -0.046178646170865513 0.265811401094662
0.38116825788023778;-0.5072350702851407 0.16609444204989182
0.77126612159555197 0.11803391491560901 0.39289393090015806 -
0.72335321191531288 -0.88227935880825836 0.062941364606925967
0.217247418584253 2 0.39493120062784881;0.7366390994969047 6
0.27992498450479508 0.42365104046483182 -0.92248330030374392
0.49118045867883908 -0.12514196104926112 0.20105176632041441 -
0.81468604330649175 0.4239366936699443 -0.44209210651693753;-
0.38895733268280552 0.55349330636865635 -0.47692129202008215
0.13409569083890011 0.71127796886763439 0.20169346030147772 -
0.19422807652981788 0.89175409004151562 0.52539356245266922 -
0.59240265336138731;0.0025382191079272413 0.48546157540618701
```

```
0.62282258852520889 -0.51782701107537754 0.085313844846789277 -
0.238214363743293 0.56777973924175618 -0.063859884426483576
0.49531283460763131 -0.7312409749048831б;-0.13659364663763962 -
0.080911468002849124 1.0016138786429658 -0.23931446329128744 -
0.31024918501429732 -0.48064416480590771 0.2632202479429507 9 -
0.66205387639108459 0.32621552185232383 0.42917155236733884;-
0.16126984144442125 0.24313096391661695 0.72764600192203943 -
0.047151630690307554 -0.38400229776564249 -0.29811414217067539
0.77062656654552908 0.34010198510498452 0.29370374025489188
0.2653832580453814 9;0.58476895824203423 0.95589345775780155
0.37528450975612682 -0.37205588099185588 0.50211498542329114
0.025678598290809665 0.60496237400186637 -0.11234970040370328
0.54954437781659449 -0.1618728570809522;0.44334337118119682
0.43600320804533255 0.060472934215061343 0.043074553093463523 -
0.3829361355762797 0.69171615445940782 0.17622306746260799 -
0.47982310227268765 0.21641124792060634 -
0.15768126833995641;0.58103060118258976 -0.56509829709441728 -
0.60498864932065288 -0.12010991656532893 -0.26124604027832166 -
0.31253402333993019 -0.72749375693704654 -0.63057191355318021 -
0.41206275555725014 0.47151032215421473];

% Layer 2
b2 =
[0.5095566324597152 8;0.93621935003685419;0.36254735126913162;0.67693872382198
561;-0.56621039431908093;-
0.086492923744872788;0.80081147171534983;0.58766937946874576;-
0.61536041046276879;-0.65031118231940843];
LW2_1 = [0.35693708703504484 0.46528881452301235 -0.98892441279371024
0.77525899623060801 0.51427944297170891 0.45472305001540186 -
0.054276777415267478 -0.62724025587562626 0.70660072907883942 -
0.80921815617167503 0.455896000824353 -0.50929759622462889
0.65040030546702066;0.15945837607471275 0.055866584073840368
0.11884807758800664 -0.24950046998347342 -0.10192272152654357
0.21050405686327911 0.27431172824637873 -0.39112502243641811
0.27355359381178851 0.044917601540473515 -0.37583199277021562
0.4974381903131232 -0.21973210675377566;-0.75517224911264147
0.027398328760361097 0.033018453100240007 0.054871148889510099
0.013790815611000058 0.052924428963081326 -0.081221576681418411 -
0.44403907900554246 -0.12891940924804196 0.48385253516307664 -
0.041222835354731632 0.45033138698738551 -0.2372658944040931;-
0.14269886431437037 0.86891021654761647 -0.66302307309410546
0.15048700254654923 -0.29844516096596058 -0.63765825054839265 -
0.54240713376209948 0.90463584105134831 0.32307486730523105
0.98691791007820606 -0.65615250725898044 -0.096832736120229468 -
0.83161779109924749;-0.17070382083428512 -0.058412942610872952 -
0.16537126792528464 0.00087480735266249297 -0.1110609571837341 -
0.12471532317173839 0.49092911284781321 -0.65126999001293617
0.78050472763761458 -0.68551278815583061 0.50235286686592373
0.22758916572820051 -0.26916401816903057;0.18134635297094509
1.2836169952823515 0.034737689320501 35 0.37170902604115635 -
0.12959158238956656 -0.61452813607589651 -0.40844328553761489
0.94339625755661893 0.47388139163599829 0.19245915823764759
0.38119857484696457 0.35655171064025837
0.17679698217593051;0.10040757084679314 -0.57938927251472983 -
0.53058470883462783 -0.56766932248054991 0.42011277870099123
0.82775236631293991 0.19599631403763756 -0.86696572089700963 -
0.5919278179520101 0.694736053711832 -0.50352684716644025 -
```

```
0.53204290330928494 -0.53908204306605567;-0.23530000523582845
0.73801994909257107 -0.49787283406810828 -0.34406367241885871
0.23538441543536345 -0.68584318805707478 0.074288643247786765
0.8521523054789274 -0.48493767502072971 0.72386477355020684
0.68481710689709374 -0.48157674017532709
0.59873955725762906;0.62781350954025805 -0.49867598299908383
0.404373021976O564 -0.80253506690057352 0.1837091486979478 -0.15452231566398 2
```

```
  a = 2 ./ (1 + exp(-2*n)) - 1;
end

% Map Minimum and Maximum Output Reverse-Processing Function
function x = mapminmax_reverse(y,settings)
  x = bsxfun(@minus,y,settings.ymin);
  x = bsxfun(@rdivide,x,settings.gain);
  x = bsxfun(@plus,x,settings.xoffset);
end
```

## Appendix 7-Important Matlab inbuilt functions used

***MEAN Average or mean value*** For vectors, MEAN(X) is the mean value of the elements in X. For matrices, MEAN(X) is a row vector containing the mean value of each column. For N-D arrays, MEAN(X) is the mean value of the elements along the first non-singleton dimension of X.
***COV Covariance matrix*** COV(X), if X is a vector, returns the variance. For matrices, where each row is an observation, and each column a variable, COV(X) is the covariance matrix. DIAG(COV(X)) is a vector of variances for each column, and SQRT(DIAG(COV(X))) is a vector of standard deviations. COV(X,Y), where X and Y are matrices with the same number of elements, is equivalent to COV([X(:) Y(:)]).
COV(X) or COV(X,Y) normalizes by (N-1) if N>1, where N is the number of observations. This makes COV(X) the best unbiased estimate of the covariance matrix if the observations are from a normal distribution. For N=1, COV normalizes by N.
COV(X,1) or COV(X,Y,1) normalizes by N and produces the second moment matrix of the observations about their mean. COV(X,Y,0) is the same as COV(X,Y) and COV(X,0) is the same as COV(X).
***INV Matrix inverse*** INV(X) is the inverse of the square matrix X. A warning message is printed if X is badly scaled or nearly singular.
***QUADPROG Quadratic programming*** X=QUADPROG(H,f,A,b) attempts to solve the quadratic programming problem:

min 0.5*x'*H*x + f'*x subject to: A*x <= b x

X=QUADPROG(H,f,A,b,Aeq,beq) solves the problem above while additionally satisfying the equality constraints Aeq*x = beq.

X=QUADPROG(H,f,A,b,Aeq,beq,LB,UB) defines a set of lower and upper bounds on the design variables, X, so that the solution is in the range LB <= X <= UB. Use empty matrices for LB and UB if no bounds exist. Set LB(i) = -Inf if X(i) is unbounded below; set UB(i) = Inf if X(i) is unbounded above.

X=QUADPROG(H,f,A,b,Aeq,beq,LB,UB,X0) sets the starting point to X0.

X=QUADPROG(H,f,A,b,Aeq,beq,LB,UB,X0,OPTIONS) minimizes with the default optimization parameters replaced by values in the structure OPTIONS, an argument created with the OPTIMSET function. See OPTIMSET for details. Used options are Display, Diagnostics, TolX, TolFun, HessMult, LargeScale, MaxIter, PrecondBandWidth, TypicalX, TolPCG, and MaxPCGIter. Currently, only 'final' and 'off' are valid values for the parameter Display ('iter' is not available).

X=QUADPROG(Hinfo,f,A,b,Aeq,beq,LB,UB,X0,OPTIONS,P1,P2,...) passes the pro
blem dependent parameters P1,P2,... directly to the HMFUN function when OPTIMSET('HessMult',HMFUN) is set. HMFUN is provided by the user. Pass empty matrices for A, b, Aeq, beq, LB, UB, XO, OPTIONS, to use the default values.

X = QUADPROG(PROBLEM) finds the minimum for PROBLEM. PROBLEM is a structure with matrix 'H' in PROBLEM.H, the vector 'f' in PROBLEM.f, the linear inequality constraints in PROBLEM.Aineq and PROBLEM.bineq, the linear equality constraints in PROBLEM.Aeq and PROBLEM.beq, the lower bounds in PROBLEM.lb, the upper bounds in PROBLEM.ub, the start point in PROBLEM.x0, the options structure in PROBLEM.options, and solver name 'quadprog' in PROBLEM.solver. Use this syntax to solve at the command line a problem exported from OPTIMTOOL. The structure PROBLEM must have all the fields.

[X,FVAL]=QUADPROG(H,f,A,b) returns the value of the objective function at X: FVAL = 0.5*X'*H*X + f'*X.

[X,FVAL,EXITFLAG] = QUADPROG(H,f,A,b) returns an EXITFLAG that describes the exit condition of QUADPROG. Possible values of EXITFLAG and the corresponding exit conditions are
1 QUADPROG converged with a solution X.
3 Change in objective function value smaller than the specified tolerance.
4 Local minimizer found.
0 Maximum number of iterations exceeded.

-2 No feasible point found.

-3 Problem is unbounded.

-4 Current search direction is not a descent direction; no further progress can be made.

-7 Magnitude of search direction became too small; no further progress can be made. The problem is ill-posed or badly conditioned.

[X,FVAL,EXITFLAG,OUTPUT] = QUADPROG(H,f,A,b) returns a structure OUTPUT with the number of iterations taken in OUTPUT.iterations, the type of algorithm used in OUTPUT.algorithm, the number of conjugate gradient iterations (if used) in OUTPUT.cgiterations, a measure of first order optimality (large-scale method only) in OUTPUT.firstorderopt, and the exit message in OUTPUT.message.

[X,FVAL,EXITFLAG,OUTPUT,LAMBDA]=QUADPROG(H,f,A,b) returns the set of Lagrangian multipliers LAMBDA, at the solution: LAMBDA.ineqlin for the linear inequalities A, LAMBDA.eqlin for the linear equalities Aeq, LAMBDA.lower for LB, and LAMBDA.upper for UB.
***CORR Linear or rank correlation*** RHO = CORR(X) returns a P-by-P matrix containing the pairwise linear correlation coefficient between each pair of columns in the N-by-P matrix X.
RHO = CORR(X,Y,...) returns a P1-by-P2 matrix containing the pairwise correlation coefficient between each pair of columns in the N-by-P1 and N-by-P2 matrices X and Y. ***XLSWRITE Stores numeric array or cell array in Excel workbook***
(SUCCESS,MESSAGE)=XLSWRITE(FILE,ARRAY,SHEET,RANGE) writes ARRAY to the Excel workbook, FILE, into the area, RANGE in the worksheet specified in SHEET. FILE and ARRAY must be specified. If either FILE or ARRAY is empty, a error is thrown and XLSWRITE terminates. The first worksheet of the workbook is the default. If SHEET does not exist, a new sheet is added at the end of the worksheet collection. If SHEET is an index larger than the number of worksheets, new sheets are appended until the number of worksheets in the workbook equals SHEET. The size defined by the RANGE should fit the size of ARRAY or contain only the first cell, e.g. 'A2'. If RANGE is larger than the size of ARRAY, Excel will fill the remainder of the region with N/A . If RANGE is smaller than the size of ARRAY, only the sub-array that

fits into RANGE will be written to FILE. The success of the operation is returned in SUCCESS and any accompanying message, in MESSAGE. On error, MESSAGE shall be a struct, containing the error message and message ID.

***CONTOUR3 3-D contour plot*** contour3(...) is the same as CONTOUR(...) except the contour lines are drawn in multiple planes. Each line is drawn in a horizontal plane at a height equal to the corresponding contour level.

$[C,H]$ = contour3(...) returns contour matrix C and a handle, H, to a contour object.

***FEEDFORWARDNET Feedforward neural network*** Two (or more) layer feedforward networks can implement any finite input-output function arbitrarily well given enough hidden neurons.

feedforwardnet(hiddenSizes,trainFcn) takes a 1xN vector of N hidden layer sizes, and a backpropagation training function, and returns a feed-forward neural network with N+1 layers.

Input, output and output layers sizes are set to 0. These sizes will automatically be configured to match particular data by train. Or the user can manually configure inputs and outputs with configure.

Defaults are used if feedforwardnet is called with fewer arguments. The default arguments are (10,'trainlm').

***MSE Mean squared error performance function***

mse(net,targets,outputs,errorWeights,...parameters...) calculates a network performance given targets, outputs, error weights and parameters as the mean of squared errors.

Only the first three arguments are required. The default error weight is 1, which weights the importance of all targets equally.

Parameters are supplied as parameter name and value pairs:

'regularization' - a fraction between 0 (the default) and 1 indicating the proportion of performance attributed to weight/bias values. The larger this value the network will be penalized for large weights, and the more likely the network function will avoid overfitting.

'normalization' - this can be 'none' (the default), or 'standard', which results in outputs and targets being normalized to [-1, +1], and therefore errors in the range [-2, +2), or 'percent' which normalizes outputs and targets to [-0.5, 0.5] and errors to [-1, 1].

Here a network's performance with 0.1 regularization is calculated.

```
perf = mse(net,targets,outputs,1,'regularization',0.1)
```