An Efficient Non-iterative Probabilistic Load Flow Analysis Method
Comparison and Worked Example

School of Engineering and Information Technology

Murdoch University – 2016

Richard Davis

BSc. Eng. (Civil)

Graduate Diploma in Energy Studies
1 Declaration and Acknowledgments

I hereby declare that this work is my own with references duly provided when used.

I would like to acknowledge the guidance and supervision provided by Dr Ali Arefi, in particular with regards to the MCGD calculations.
# Table of Contents

1 Declaration and Acknowledgments ........................................................................................................ ii

2 Abstract ...................................................................................................................................................... v

3 Introduction ............................................................................................................................................... 1

  4.1 Research Question and Design ............................................................................................................... 1

  4.2 Aims and Objectives ............................................................................................................................... 1

  4.3 Target Audience ..................................................................................................................................... 2

4 Background ............................................................................................................................................. 3

  5.1 Power Flow Analysis ............................................................................................................................... 4

    5.1.1 Power Flow Base Equations ............................................................................................................. 4

    5.1.2 Deterministic Load Flow Analysis ................................................................................................... 5

    5.1.3 Probabilistic Load Flow Analysis .................................................................................................... 5

  5.2 Monte Carlo Method ............................................................................................................................... 6

    5.2.1 Background and sample ................................................................................................................... 6

  5.3 Gaussian Distribution (GD) .................................................................................................................... 8

  5.4 Proposed Method (MCGD) ..................................................................................................................... 9

5 Power Flow Analysis Process Overview ............................................................................................... 10

  6.1 Direct Approach .................................................................................................................................. 10

6 Sample Network .................................................................................................................................... 13

  7.1 Loads and Layout ................................................................................................................................ 13

  7.2 Impedance .......................................................................................................................................... 14

  7.3 Calculation of Branch Injection Branch Current (BCBV) and Branch Injection Branch Current (BIBC) Matrices ................................................................................................................................. 15

    7.3.1 BIBC ............................................................................................................................................... 15

    7.3.2 BCBV ............................................................................................................................................. 16
8 Power Flow Analysis – Simulations ................................................................. 18

8.1 Monte Carlo Simulation using Direct-Method ........................................... 18
  8.1.1 Steps ........................................................................................................... 18
  8.1.2 Observations of Technique ..................................................................... 19

8.2 Copula Correlation ...................................................................................... 20
  8.2.1 Steps ........................................................................................................... 20

8.3 Multivariate Complex Gaussian Distribution (MCGD) using the Direct Approach ................................................................................................................. 23
  8.3.1 Steps ........................................................................................................... 24

9 Results .............................................................................................................. 30
  9.1 Interpretation of Results ............................................................................. 31

10 Future Work .................................................................................................... 33

11 Outcomes ......................................................................................................... 33

12 Conclusions ..................................................................................................... 34

13 Appendix .......................................................................................................... 35
  13.1 References .................................................................................................. 35
  13.2 Scripts .......................................................................................................... 38
    13.2.1 MC Pi Simulation .................................................................................... 38
    13.2.2 MC Copula Simulation .......................................................................... 39
    13.2.3 MCGD Calculations .............................................................................. 41
3 Abstract

This paper presents an analytical assessment method for probabilistic power flow analysis using a Multivariate Complex Gaussian Distribution (MCGD) and a Direct Approach for the load flow equations.

To provide a reference for comparison and assessment, a standard Monte Carlo (MC) simulation and a Copula coefficient MC simulation were undertaken. Both use the Direct Approach for load flow equations.

Details and steps of all methods are provided along with background information deemed relevant to provide a self-contained body of knowledge. This improves accessibility to a less mathematically advanced audience.

It will be shown that the proposed technique is significantly quicker to compute and that the magnitude of mean voltages closely match those obtained using the MCSs.

Further, it will be shown that standard deviations observed varied up to 45% between methods, at specific load points, and it is concluded that further analysis is required to be able to determine that the proposed method is sufficiently accurate.
4 Introduction

4.1 Research Question and Design

The research question was to assess a proposed method of probabilistic power flow analysis in radial distribution grids.

The method proposed is a non-iterative multivariate complex Gaussian Distribution solution that uses the Direct Approach for load flow calculations.

The research method is an analytical one that seeks to explore the feasibility of the proposed solution through comparison to generally accepted techniques, such as MC simulation, while at the same time detailing the steps and underlying concepts in enough detail to engage with the target audience.

4.2 Aims and Objectives

The aim of this paper is to assist in optimising the speed and accuracy of microgrid control system decision making. This in turn allows for robust and efficient microgrid optimisation, facilitating a transition from unsustainable fossil fuel power to renewable energy.

The associated objective is to assess and compare the accuracy of the proposed method against existing methods.
4.3 Target Audience

Many research papers present complex solutions that are based on the successive metaphorical shoulders of other researchers, often resulting in only terse reference to underlying concepts and methods. As part of my research this required a significant amount of ‘drill-down’ into the supporting mathematics and statistics and cited literature. Often it was found that cited papers themselves provided little background.

Considering the expected growth of renewables penetration into power grids – and the uncertainties that accompany this – exposure to the broader principles of probabilistic control systems could be of value to an audience whose undergraduate mathematics is decades in the past.

Therefore, I have chosen to include brief explanations of underlying concepts and methods employed in this research and document a worked example, rather than just the results of the proposed method’s performance.
5 Background

Traditional centralised power generation must overcome the challenges associated with a large distributed network, such as transmission losses, costs and maintenance. Further, historically centralised power generation in Australia is largely reliant on coal-fired plants, leading to significant amounts of pollution and environmental damage.

The emergence of small-scale renewable energy sources has led to the number of distributed energy generation (DEG) systems increasing. Within these systems a distributed energy resource (DER) could be connected to loads, storage, various switching mechanisms and controllers on a local grid system – called a microgrid. Distributed Generation (DG) refers to the generation of power on-site at the point of consumption, as opposed to in a large centralised power plant.

According to CIGRÉ C6.22 Working Group: “Microgrids are electricity distribution systems containing loads and distributed energy resources, (such as distributed generators, storage devices, or controllable loads) that can be operated in a controlled, coordinated way either while connected to the main power network or while is landed.” (“Microgrid Definitions | Building Microgrid” 2016)

Power flow analysis is undertaken in order to assess various aspects of a grid, such as voltage, current and power.
5.1 Power Flow Analysis

“A power flow study is a steady state analysis whose target is to determine the voltages, currents, real and reactive power flows in a system under a given load conditions. The purpose of power flow studies is to plan ahead and account for various hypothetical situations.” (Dubey 2016)

5.1.1 Power Flow Base Equations

The power flow equations are based on the following:

\[ i_i = \sum_{j=1}^{n} Y_{ij} V_j \]

Where \( i_i \) and \( V_j \) are the injected current at bus \( i \) and voltage at bus \( j \), respectively.

Relating current at a node to the power (complex) and voltage.

\[ i_i = \frac{P_i - jQ_i}{\dot{V}_i} \]

Where the current at node \( i \) is given by the real (P) and imaginary (Q) power components divided by the conjugate of the voltage at \( i \).

Combining:

\[ \frac{P_i - jQ_i}{\dot{V}_i} = \sum_{j=1}^{n} Y_{ij} \dot{V}_j \]

These form the base for many load flow techniques, including the ‘Direct Approach’ as used in this paper.
5.1.2 Deterministic Load Flow Analysis

When loads, generation and network topology are known with suitable confidence – if necessary across several scenarios – deterministic load flow methods are suitable. However, in situations where uncertainty exists statistical methods are required.

5.1.3 Probabilistic Load Flow Analysis

A statistical procedure will produce potential outcomes as a probability distribution as opposed to specific values obtained from deterministic methods. Being aware of the statistical probability of a state, or value, is important in facilitating accurate decision-making by control systems or system modellers.

Renewable energy generation, such as solar and wind, are inherently variable. The ability to analyse power flows inclusive of this variability requires probabilistic methods. Specific to microgrids, power flow analysis is integral to the decision-making algorithms of system control units. Decisions need to be made in an accurate and near real-time manner, these include controlling the amount of power being injected by each DG source, directing power to dump loads, adjusting to islanding and grid connected environments and enabling self-protection steps.

Monte Carlo (MC) methods are commonly used to produce probability distributions for systems for which the inputs can be represented statistically (Allan and Silva 1981)(Nikmehr and Najafi Ravadanegh 2016)(Cao et al.
In this paper we will be using MC simulations and the Direct Approach of load flow analysis to calculate the mean and standard deviation (SD) of voltages at each load point in our sample grid. As it is a well-established technique, the output obtained will be used as a comparison against our proposed method.

Further, although there is variability in renewable energy sources such as wind and sunlight, assuming these and the load demands are independent from each other can lead to significant overestimation of the grid’s performance (Wang, Zhang, and Liu 2016), therefore a method for catering for dependencies using Copula correlations will be shown for the MC method. The proposed MCDG process incorporates correlation into its network modelling as part of the base method.

5.2 Monte Carlo Method

To provide a comparison to the proposed MCGD method, a MC simulation is undertaken. This is extended to include a MC solution using Copula correlations.

5.2.1 Background and sample

A MC simulation can be defined as:

“An analytical technique for solving a problem by performing a large number of trial runs, called simulations, and inferring a solution from the collective results of the trial runs. Method for calculating the probability distribution of possible
Monte Carlo methods seek to learn about a system by simulating it through inputting many random values.

A commonly used explanatory example of the MC method is the determination of the value of pi. It can be described as follows ("Estimating Pi Using the Monte Carlo Method" 2016):

As shown in Figure 1, plotted by a MatLab script (Mg 2012), a circle is placed within a square. The circle has a radius of 0.5 units and the square is 1x1 units.

The ratio of the circle to the square area would be given by $\pi r^2 / 1 \times 1$, with $r=0.5$ as defined, this gives $A_{\text{circle}}/A_{\text{square}} = \pi/4$.

If points were randomly chosen over the area and the ratio of the number of points within the circle area (blue) over the total number of points equated to the area of the circle divided by the area of the square we get:

$$\frac{\text{Points}_{\text{Circle}}}{\text{Points}_{\text{Total}}} = \frac{\pi}{4}$$

Rearranging,

$$\pi = 4 \times \left( \frac{\text{Points}_{\text{Circle}}}{\text{Points}_{\text{Total}}} \right)$$
Scripting in MatLab (Mg 2012) it can be shown that as the random number of points increases, so does the accuracy of $\pi$.

It can be clearly seen in Figure 2 that the estimation is approaching the value of $\pi$.

### 5.3 Gaussian Distribution (GD)

Gaussian distributions (GDs), or normal distributions, are continuous probability distributions that can be used to represent random variables and often represent distributions commonly seen in nature (“History of Normal Distribution” 2016).

GDs will be used to statistically describe the load magnitude probability at each load point in the system.

GDs that are applied to a single variable (therefore $R^2$ space) are termed univariate, while multiple variable distributions are deemed multivariate. Univariate distributions are used in the MC simulation and multivariate distributions as part of the MCGD method.

GD are effective as they exhibit the Central Limit Theory (CLT). This allows the representation of random variables whose distribution is not. The CLT states that the distribution of the mean values of many independent identically distributed variables will approximate a normal distribution. This occurs regardless of the underlying distribution of the random variables (“The Central Limit Theorem” 2016).
5.4 Proposed Method (MCGD)

The speed and accuracy of controller calculations is crucial to optimise the operation and reliability of microgrids and protect components from potential damage due to sudden changes or excessive variance of power.

To address these challenges the use of a non-iterative power flow analysis method is proposed, specifically a conditional multivariate complex Gaussian distribution using the Direct Approach. This is based on papers published for dynamic state estimation (Arefi, Ledwich, and Behi 2015) and the Direct Approach of power flow analysis (Jen-Hao Teng 2003).

"Multivariate Data Analysis refers to any statistical technique used to analyze data that arises from more than one variable. This essentially models reality where each situation, product, or decision involves more than a single variable." ("Multivariate Data Analysis (MVA): Powerful Statistics & Data Mining" 2016)

In the proposed method load variation, correlation and measurement errors are utilised in a single step to obtain the mean and standard deviation of the state variables. The process initially represents bus voltages, branch currents, and injection currents as multivariate complex Gaussian distributions then, using direct load flow and a linear transformation, calculates the mean and standard deviation of bus voltages using conditional multivariate complex Gaussian distribution and the estimation of variance method (Arefi, Ledwich, and Behi 2015).
6 Power Flow Analysis Process Overview

The power flow problem involves solving two of the four power flow variables (voltage, phase angle, real power and imaginary power) at each bus in the network.

A set of non-linear power equations are established and solved for the unknown variables. As the equations are non-linear, iterative procedures are often used to converge on a result (such as the Gauss-Seidel and Newton-Raphson methods).

Gauss-Seidel and Newton-Raphson methods are commonly used for transmission systems but can fail to provide suitable accuracy and robustness when applied to distribution systems. In particular when the distribution has characteristic traits such as being radial, or weakly meshed, containing unbalanced distributed loads and having a range of reactance and resistance (Jen-Hao Teng 2003).

The Direct Approach will be used for all methods going forward.

6.1 Direct Approach

The Direct Approach (Jen-Hao Teng 2003) optimises the use of the distribution system topology to solve the load flow problem.
For the iterative and deterministic case the solution steps are:

Two matrices are developed, Branch Injection Branch Current (BIBC) and Branch Current Branch Voltage (BCBV), which are then used with a current injection matrix (I). (I) is calculated using Equation 3, using voltage estimates for the first iteration.

The steps to develop the matrices are as follows:

1. Determine the BCBV and BIBC matrices as per section 7.3 and then multiply to obtain the Distributed Load Function (DLF) (Equation 2).
2. Calculating voltages using the DLF matrix (Equation 1).
3. Using the new voltages obtained, calculate the current at each load point (I) (Equation 3).
4. Use the new current values calculated in Step 3 to again calculate voltages in Step 2.
5. Repeat Steps 2 to 4 until suitable convergence of voltage values (~ 3 iterations)

\[ V_{\text{new}} = DLF \times I_{\text{Initial}} + V_{\text{Est}} \]

\[ DLF = BCBV \times BIBC \]

\[ I_{\text{next}} = (P + Q)/V_{\text{new}}/\sqrt{3} \]

For the network considered in this paper, the system was found to converge in three (3) iterations to four decimal places (Volts).
As an aside, solving for multiple nodes can be achieved using matrices:

Equation 4

\[ I_{\text{initial/next}} = \text{conj}(P_{\text{Mean}}' + 1i * Q_{\text{Mean}}') / V_{\text{base}} / \sqrt{3} \]

Equation 5

\[ V_{\text{new}} = -DLF * I_{\text{initial/next}} + V_{\text{Initial}} \]

Where P and Q are matrices of real and imaginary power.
7 Sample Network

7.1 Loads and Layout

The loads and layout for our sample network are defined as follows:

Some simplifying assumptions will be made with regards to the loads for a sample network.

- A load can be classified as either residential or commercial. For use in the multivariate complex Gaussian distribution Direct Approach, it will be assumed that loads of the same type have a correlation coefficient of 0.6 and if different, 0.4.

- The standard deviation (SD) of current injection can be estimated by:

\[
SD = (u*err/300)
\]

Where \( u \) = mean current and \( err=50 \).
Using the above assumptions, a base voltage of 11kV and the equation \( I = \frac{P}{V/\sqrt{3}} \), the network values can be calculated as shown in Table 2.

Hence complex mean current is (Table 3):

These values will be used for both the Monte Carlo and MCGD calculations.

### 7.2 Impedance

Impedances are accounted for in the distribution branches and assumed to be uniform across the system and linear functions of the branch length (Table 2).

Values are assumed to be:

- Resistance (\( R \)): 0.886 Ω per meter
- Reactance (\( XL \)): 0.745 Ω per meter

The impedance is then calculated using:

\[
Z = R + iXL
\]

Therefore, for our sample network:

<table>
<thead>
<tr>
<th>Branch</th>
<th>Length (M)</th>
<th>Resistance (Ohms)</th>
<th>Reactance (Ohms)</th>
<th>Impedance</th>
</tr>
</thead>
<tbody>
<tr>
<td>B12</td>
<td>1500</td>
<td>0.886</td>
<td>1.329</td>
<td>1.329 + 1.1175i</td>
</tr>
<tr>
<td>B23</td>
<td>1800</td>
<td>0.886</td>
<td>1.595</td>
<td>1.5948 + 1.341i</td>
</tr>
<tr>
<td>B34</td>
<td>1900</td>
<td>0.886</td>
<td>1.683</td>
<td>1.6834 + 1.4155i</td>
</tr>
<tr>
<td>B45</td>
<td>1700</td>
<td>0.886</td>
<td>1.506</td>
<td>1.5062 + 1.2665i</td>
</tr>
<tr>
<td>B36</td>
<td>1800</td>
<td>0.886</td>
<td>1.595</td>
<td>1.5948 + 1.341i</td>
</tr>
</tbody>
</table>

Table 2
### 7.3 Calculation of Branch Injection Branch Current (BCBV) and Branch Injection Branch Current (BIBC) Matrices

The Direct Approach requires the BCBV and BIBC matrices to be calculated. As these matrices are only dependent on the network they only need to be recalculated if changes were made to the topology.

#### 7.3.1 BIBC

Starting with the power injection at each load point the current can be calculated as previously shown. Kirchhoff’s Current Law (KCL) can then be employed to obtain the relationship between the injected and branch current for each point.

For our sample network:

- \( B_1 = I_2 + I_3 + I_4 + I_5 + I_6 \)
- \( B_2 = I_3 + I_4 + I_5 + I_6 \)
- \( B_3 = I_4 + I_5 \)
- \( B_4 = I_5 \)
- \( B_5 = I_6 \)

Formulating this into matrix form gives:

\[
\begin{bmatrix}
B_1 \\
B_2 \\
B_3 \\
B_4 \\
B_5 \\
\end{bmatrix} =
\begin{bmatrix}
1 & 1 & 1 & 1 & 0 \\
0 & 1 & 1 & 1 & 0 \\
0 & 0 & 1 & 1 & 0 \\
0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 1 \\
\end{bmatrix}
\begin{bmatrix}
I_2 \\
I_3 \\
I_4 \\
I_5 \\
I_6 \\
\end{bmatrix}
\]
The BIBC matrix is the 5x5 zeros and ones matrix as shown above. To facilitate automation of the BIBC matrix construction a building algorithm was proposed (Jen-Hao Teng 2003).

1. For a distribution system with m-branch section and n-bus, the dimension of the BIBC matrix is mx(n-1)
2. If a line section Bk is located between bus i and bus j, copy the column of the i-th bus of the BIBC matrix to the column of the j-th bus and fill a +1 to the position of the k-th row and the j-th bus column.
3. Repeat procedure (2) until all line sections are included in the BIBC matrix.

7.3.2 BCBV

Equating load point voltages to losses in branches:

\[
\begin{align*}
V_1 - B_1Z_{12} &= V_2 \\
V_2 - B_2Z_{23} &= V_3 \\
V_3 - B_3Z_{34} &= V_4 \\
V_4 - B_4Z_{34} &= V_5 \\
V_5 - B_5Z_{56} &= V_6
\end{align*}
\]

Rearranging and substituting, we can write in the form \( V_1 - V_x \) for each load point (x):

\[
\begin{align*}
V_1 - V_2 &= Z_{12}B_1 \\
V_1 - V_3 &= Z_{12}B_1 + Z_{23}B_2 \\
V_1 - V_4 &= Z_{12}B_1 + Z_{23}B_2 + Z_{34}B_3 \\
V_1 - V_5 &= Z_{12}B_1 + Z_{23}B_2 + Z_{34}B_3 + Z_{45}B_4 \\
V_1 - V_6 &= Z_{12}B_1 + Z_{23}B_2 + Z_{36}B_5
\end{align*}
\]
Where \( Z_{xy} \) is shown in Table 2 for each corresponding branch \( B_{xy} \).

*These equations can also be derived by adding the impedances along the branches of the respective voltage drop in question.

Rearranging and formulating the matrix:

\[
\begin{bmatrix}
V_1 \\
V_2 \\
V_3 \\
V_4 \\
V_5 \\
V_6
\end{bmatrix}
= \begin{bmatrix}
Z_{12} & 0 & 0 & 0 & 0 \\
Z_{12} & Z_{23} & 0 & 0 & 0 \\
Z_{12} & Z_{23} & Z_{34} & 0 & 0 \\
Z_{12} & Z_{23} & Z_{34} & Z_{45} & 0 \\
Z_{12} & Z_{23} & 0 & 0 & Z_{36}
\end{bmatrix}
\begin{bmatrix}
B_1 \\
B_2 \\
B_3 \\
B_4 \\
B_5
\end{bmatrix}
\]

The BCBV matrix is the 5x5 impedance matrix as shown above. To facilitate automation of the BCBV matrix construction a building algorithm was proposed (Jen-Hao Teng 2003):

1. For a distribution system with \( m \)-branch section and \( n \)-bus, the dimension of the BCBV matrix is \((n-1)xm\).

2. If a line section \( B_k \) is located between bus \( i \) and bus \( j \), copy the row of the \( i \)-th bus of the BCBV matrix to the row of the \( j \)-th bus and fill the line impedance \( Z_{ij} \) to the position of the \( j \)-th bus row and the \( k \)-th column.

3. Repeat procedure (2) until all line sections are included in the BCBV matrix.

Substituting the impedance values from Table 2 for the BCBV matrix:

\[
\begin{bmatrix}
1.329 + 1.1175i & 0 & 0 & 0 & 0 \\
1.329 + 1.1175i & 1.5948 + 1.341i & 0 & 0 & 0 \\
1.329 + 1.1175i & 1.5948 + 1.341i & 1.6834 + 1.4155i & 0 & 0 \\
1.329 + 1.1175i & 1.5948 + 1.341i & 1.6834 + 1.4155i & 1.5062 + 1.2665i & 0 \\
-1.329 + 1.1175i & 1.5948 + 1.341i & 0 & 0 & 1.5948 + 1.341i
\end{bmatrix}
\]
8 Power Flow Analysis – Simulations

MatLab scripts were used to implement the power flow solutions shown below. The scripts used are documented in the Appendix.

8.1 Monte Carlo Simulation using Direct-Method

In our simulation, the MC method uses a Gaussian distribution of current at each load point (Figure 4).

The standard deviation and mean previously calculated are used to define the Gaussian distributions.

8.1.1 Steps

1. The MatLab 'normrnd' function is used to generate random variables from the Gaussian distribution of current for each load point.
2. A matrix implementation of the 'Direct Approach' (Equation 1) is then used to calculate the voltage at each load point and these values are stored.
3. Steps 1 and 2 and repeated many times (>10,000).
4. The mean and SD of stored voltage values are calculated.

The results were recorded for 200,000 iterations (taking ~53 seconds) (Table 5).
8.1.2 Observations of Technique

Observing the convergence of the mean and SD values can be of interest and offers insight into how the MC method works.

Figure 5 depicts the progression of the mean value of real voltage for load point 2. Each unit value on the x-axis equates to 100 iterations, so the x-axis spans 200,000 iterations.

It can be seen that the mean voltage variation varies significantly less after the first 50,000 iterations (500 on the x-axis). A similar trend can be seen in the variation of the SD (Figure 6).

Further, it can be observed that plotting the resulting voltage values on a histogram, again for the real values of Load Point 2, results in a Gaussian shaped distribution (Figure 7).
Probability distributions such as this allow control systems and operators to make more informed decisions around system design and control than would be possible with the single value produced through deterministic methods.

### 8.2 Copula Correlation

“Copulas are functions that describe dependencies among variables, and provide a way to create distributions that model correlated multivariate data. Using a copula, you can construct a multivariate distribution by specifying marginal univariate distributions, and then choose a Copula to provide a correlation structure between variables.” (“Copulas: Generate Correlated Samples - MATLAB & Simulink - MathWorks Australia” 2016).

Using a Copula generated distribution incorporates any inherent correlation in the underlying data (current in this case) and allows for a single sample to be taken when generating random variables (as opposed to one per load point).

The following steps were undertaken in a MatLab script (attached in Appendix):

#### 8.2.1 Steps

1. As before, the MatLab ‘normrnd’ function is used to generate random variables from the Gaussian distribution of current for each load point.

As the ksdensity functions proved time consuming to process the number of samples taken was reduced to 50,000. Observations from MC method previously showed convergence of mean and SD after 50,000 iterations.
2. The MatLab 'ksdensity' function is used to create a probability density estimate and transform the data from Step 1 to the copula scale (unit square).

Plots for the MC output values of real current (Load Point 2) and the transformed Copula scale data are shown in Figure 8.

![Figure 8](image1.png)

3. Correlation factors (rhohat) and the degrees of freedom (nuhat) are estimated from the current data using the MatLab function 'copulafit' Table 6.

A ‘t’ Copula fitting is used for estimation, employing the Approximate Maximum Likelihood method. The alternative Maximum Likelihood (ML) method is slower and better suited to small sample sizes. It was assumed that sufficient samples were taken in our case to use the approximated method.

![Table 4](image2.png)
4. Sample random variables using ‘copularnd’ from the Copula based distribution using the parameters estimated on Step 3. 50,000 random variables were sampled.

5. Using the ‘ksdensity’ function with an inverse probability function, transform the random sample back to the original scale of the data.

Figure 9 shows a combined scatter plot and histogram of the random samples (LP2 vs LP3) taken from the current probability distribution and the distribution of values obtained by sampling the multivariate distribution created using the Copulas.

This shows visually that the distributions are similar, and therefore that the Copula derived distribution provides a reasonable estimate of the combined univariate distributions of current at each load point.
6. Perform a MC simulation using the Copula sampled data from Step 5 and the load flow ‘Direct Approach’ to obtain the voltage mean and SD.

For comparison, the outputs from the multivariate Copula probability distribution are shown in Table 7 alongside the initial MC simulation results.

<table>
<thead>
<tr>
<th>Monte Carlo Simulation (200,000)</th>
<th>Monte Carlo Simulation- Copula Correlation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Real</td>
</tr>
<tr>
<td>Mean</td>
<td>SD</td>
</tr>
<tr>
<td>10885.592</td>
<td>31.089</td>
</tr>
<tr>
<td>10858.693</td>
<td>40.352</td>
</tr>
<tr>
<td>10848.264</td>
<td>46.120</td>
</tr>
<tr>
<td>10872.065</td>
<td>36.915</td>
</tr>
</tbody>
</table>

Table 5

Relevant voltages differ by less than $1/100^{th}$ of a percent and SD are within 4%, supporting the conclusion that the derived Copula function is a suitable estimator of the underlying univariate distributions.

8.3 Multivariate Complex Gaussian Distribution (MCGD) using the Direct Approach

The Multivariate Complex Gaussian Distribution (MCGD) leverages topological aspects of radial distribution systems and seeks to solve the power flow problem without iteration. The term ‘complex’ refers to the use of imaginary numbers in the power values.

As a high-level description, the method seeks to model the network by creating a multivariate Gaussian distribution from the univariate GDs at each load point. The behaviour of the interaction between load points is incorporated in the model using assumed correlation factors and the covariance matrices derived therefrom.
The MCGD assumes the form:

\[ Ms + b \sim \text{MCGD}(M\mu_s + b, M^{ss}M^h, MC_{ss}M^T) \]

Where, \( Ms + b \) can be taken to equate to the Direct Approach (Equation 1), specifically:

\[
\begin{align*}
M &= -\text{DLF} \\
S &= I_{\text{Mean}} \\
B &= V_{\text{initial}}
\end{align*}
\]

\( I_{ss} \) and \( C_{ss} \) are combination covariance matrices for real and imaginary components.

As for the Monte Carlo distribution, standard deviations of current at each load point are estimated by \((u*err/300)\).

The implementation of the method in this paper is based on the work done in the paper ‘An Efficient DSE Using Conditional Multivariate Complex Gaussian Distribution’ (Arefi, Ledwich, and Behi 2015).

8.3.1 Steps

1. Using the BCBV and BIBC matrices from before, the distribution load flow matrix (DLF) is calculated as per Equation 2.
2. Calculate the covariance matrices (CIVlxy) for the current at each load point.
This entails:

Creating a correlation matrix ('r' values) (Table 8) based on our assumptions on load types previously.

Then, knowing a correlation coefficient (r) can be defined in terms of standard deviations:

\[ r = \frac{SD_{xy}}{SD_x \times SD_y} \]

Rearranging,

\[ SD_{xy} = r \times SD_x \times SD_y \]

The format of the covariance matrix is given by (Table 9):

We can calculate the following covariance matrices using the correlation coefficient mapping (Table 8) and the standard deviations from Table:

Covariance matrices will use the naming convention:

\[ COV_{Iab} \]

Where, 'I' denotes a covariance matrix for current and 'a' and 'b' will be either 'r' for real values or 'i' for imaginary values (to allow for complex power depiction).

The four required covariance matrices are calculated as per the above and shown in Table 10. The covariance matrices indicate the degree of correlation.
between the various load points the real and the real and imaginative power and voltages.

<table>
<thead>
<tr>
<th>Load Point</th>
<th>COVrr</th>
<th>COVri</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>27.5478</td>
<td>11.4324</td>
</tr>
<tr>
<td>3</td>
<td>11.4324</td>
<td>13.1791</td>
</tr>
<tr>
<td>4</td>
<td>8.2664</td>
<td>5.7163</td>
</tr>
<tr>
<td>5</td>
<td>14.0493</td>
<td>9.7175</td>
</tr>
<tr>
<td>6</td>
<td>9.9173</td>
<td>6.8595</td>
</tr>
</tbody>
</table>

Table 8

*COVrr and COVri have a factor 0.9 applied*

3. Calculate the covariance matrix $\Gamma_{ss}I$ and relation matrix $C_{ss}I$ for complex current from current covariance matrices. The postscript 'I' denoting current.

Where,

$$\Gamma_{ss}I = \text{COVrr} + \text{COVlii} + i(\text{COVlr} - \text{COVrii})$$

$$C_{ss}I = \text{COVrr} - \text{COVlii} + i(\text{COVlr} + \text{COVrii})$$

This results in:

<table>
<thead>
<tr>
<th>Load Point</th>
<th>$\Gamma_{ss}I$</th>
<th>$C_{ss}I$</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>30.923 + 0i</td>
<td>15.193 - 6.0682i</td>
</tr>
<tr>
<td>3</td>
<td>15.193 + 6.0682i</td>
<td>24.818 + 0i</td>
</tr>
<tr>
<td>4</td>
<td>10.386 + 2.8512i</td>
<td>9.656 + 1.062i</td>
</tr>
<tr>
<td>5</td>
<td>18.388 + 6.7314i</td>
<td>17.775 - 0.5021i</td>
</tr>
<tr>
<td>6</td>
<td>12.81 + 4.314i</td>
<td>12.231 - 0.657i</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Load Point</th>
<th>$\Gamma_{ss}I$</th>
<th>$C_{ss}I$</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>24.174 + 17.3554i</td>
<td>6.143 + 8.0579i</td>
</tr>
<tr>
<td>3</td>
<td>7.672 + 13.2707i</td>
<td>1.54 + 22.2934i</td>
</tr>
<tr>
<td>4</td>
<td>6.143 + 8.0579i</td>
<td>1.777 + 8.6074i</td>
</tr>
<tr>
<td>5</td>
<td>9.711 + 15.5826i</td>
<td>1.66 + 15.936i</td>
</tr>
<tr>
<td>6</td>
<td>7.025 + 10.562i</td>
<td>1.488 + 10.9463i</td>
</tr>
</tbody>
</table>
4. Calculate mean voltage using the direct method (Equation 1):

\[ V_{\text{mean}} = DLF \cdot I_{\text{mean}} + V_{\text{initial}} \]

Where \( I_{\text{mean}} \) is as per input data (Table ) and \( V_{\text{initial}} \) is a 5x1 column vector of 11,000V.

This gives mean voltages for system (Table 12).

<table>
<thead>
<tr>
<th>Load Point</th>
<th>Mean Voltage</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>10931.904 -275.1642i</td>
</tr>
<tr>
<td>3</td>
<td>10885.63 -545.5526i</td>
</tr>
<tr>
<td>4</td>
<td>10858.759 -685.7927i</td>
</tr>
<tr>
<td>5</td>
<td>10848.354 -755.2692i</td>
</tr>
<tr>
<td>6</td>
<td>10872.103 -621.2275i</td>
</tr>
</tbody>
</table>

Table 10

5. Calculate the Correlation matrices for voltage.

Having calculated the MCGD components of the current for the system, we can now get the MCGD components (Mean, \( I_{ssV} \) and \( C_{ssV} \)) for voltage, using:

\[ I_{ssV} = M I_{ssi} M^H \quad \text{and} \quad C_{ssV} = M C_{ssi} M^T \]

Where:

\[ M = -DLF \]

Giving:

<table>
<thead>
<tr>
<th>Load Point</th>
<th>( I_{ssV} )</th>
<th>( C_{ssV} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 11
6. Calculate the corresponding covariance matrices

Using:

\[ \text{COV}_{rr} = 0.5 \times \text{real} \left( r_{ssV} + c_{ssV} \right) \]

\[ \text{COV}_{vi} = 0.5 \times \text{imag} \left( -r_{ssV} + c_{ssV} \right) \]

\[ \text{COV}_{vi} = 0.5 \times \text{imag} \left( r_{ssV} + c_{ssV} \right) \]

\[ \text{COV}_{ii} = 0.5 \times \text{real} \left( r_{ssV} - c_{ssV} \right) \]

<table>
<thead>
<tr>
<th>Load Point</th>
<th>( \text{COV}_{rr} )</th>
<th>( \text{COV}_{vr} )</th>
<th>( \text{COV}_{vi} )</th>
<th>( \text{COV}_{vv} )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>140.6440</td>
<td>249.3123</td>
<td>308.9694</td>
<td>336.6436</td>
</tr>
<tr>
<td>3</td>
<td>249.3123</td>
<td>455.2748</td>
<td>567.7189</td>
<td>618.9589</td>
</tr>
<tr>
<td>4</td>
<td>308.9694</td>
<td>567.7189</td>
<td>713.5065</td>
<td>779.7780</td>
</tr>
<tr>
<td>5</td>
<td>336.6436</td>
<td>618.9589</td>
<td>855.7569</td>
<td>979.7780</td>
</tr>
<tr>
<td>6</td>
<td>270.5738</td>
<td>514.0130</td>
<td>638.8525</td>
<td>694.5776</td>
</tr>
</tbody>
</table>

Table 12

7. Calculate the standard deviation (for the voltages at each load point).

The SD for the voltages can be given by:

\[
\text{SD}_{v} = \sqrt{\sum_{i} \text{Var}(v_i)}
\]

Where

\[
\text{Var}(v_i) = \begin{bmatrix} v_i^R \\ v_i^I \end{bmatrix} \times \begin{bmatrix} \text{COV}_{rr} & \text{COV}_{vr} \\ \text{COV}_{vr} & \text{COV}_{vv} \end{bmatrix} \times \begin{bmatrix} v_i^R \\ v_i^I \end{bmatrix} / (v_i^R)^2 + (v_i^I)^2
\]

\[ v_i^R \text{ and } v_i^I \text{ are the real and imaginary voltages for load point I and COV values are obtained from the respective COV}_{xy} \text{ matrices.} \]

This was implemented in MatLab as:

\[
v_{v_{\text{vari}}} = [\text{Vrimean}(i,1), \text{Vrimean}(i,2)]*[\text{COV}_{rr}(i,i), \text{COV}_{vr}(i,i); \text{COV}_{vi}(i,i), \text{COV}_{vv}(i,i)]*[\text{Vrimean}(i,1); \text{Vrimean}(i,2)] / (\text{Vrimean}(i,1)^2 + \text{Vrimean}(i,2)^2)
\]
Where \( V_{rimean} \) is a 5x2 matrix constructed by concatenating the real and imaginary voltages respectively, for each load point (i).

Table 15 shows the calculated variances. Table 16 summarises the results of the MCGD calculations.

<table>
<thead>
<tr>
<th>Load Point</th>
<th>Var (I)</th>
<th>SD (I)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>126.542218</td>
<td>11.2490985</td>
</tr>
<tr>
<td>3</td>
<td>369.746783</td>
<td>19.2288009</td>
</tr>
<tr>
<td>4</td>
<td>550.561931</td>
<td>23.4640562</td>
</tr>
<tr>
<td>5</td>
<td>646.374108</td>
<td>25.4238885</td>
</tr>
<tr>
<td>6</td>
<td>462.927983</td>
<td>21.5157613</td>
</tr>
</tbody>
</table>

Table 15

<table>
<thead>
<tr>
<th>Load Point</th>
<th>MCGD Direct Method</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Real</td>
<td>SD</td>
</tr>
<tr>
<td></td>
<td>Mean</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>10931.904</td>
<td>11.249</td>
</tr>
<tr>
<td>3</td>
<td>10885.630</td>
<td>19.229</td>
</tr>
<tr>
<td>4</td>
<td>10858.759</td>
<td>23.464</td>
</tr>
<tr>
<td>5</td>
<td>10848.354</td>
<td>25.424</td>
</tr>
<tr>
<td>6</td>
<td>10872.103</td>
<td>21.516</td>
</tr>
</tbody>
</table>

Table 16
9 Results

Table 17 shows a summary of the resultant statistical descriptors of the voltage at each load point in the network and Figure provides a graphical comparison of voltages at each load point.

Table 18 displays a comparison of real voltage magnitude and standard deviation for both the MCGD and MC Copula method, using the base MCS as a reference.

![Table 17](image1)

![Table 18](image2)

![Figure 10](image3)

![Figure 11](image4)
The following can be observed:

- The magnitude of the real mean voltages are similar, with less than one hundredth of a percentage difference between all methods.
- The standard deviations are within 4% between the Copula MC and base MC methods.
- Variations in SD of between 27% and 44% are observed between the MCGD and base MCS.
- The MCGD method offers significant computation time advantages* over the other methods (Figure 10).

*Calculations were performed in MatLab R2016b on a computer with specifications - CPU : 1.3 GHz Intel Core i5 and RAM : 4 GB 1600 MHz DDR3

9.1 Interpretation of Results

Significant points to consider from interpretation of the results are the execution time and accuracy of the MCGD method.

The MC based calculations took up to 1650 times longer than the MCGD method. The speed of calculation using the MCGD method is due to its non-iterative nature

The mean voltage magnitudes obtained by all methods were so similar they could be considered equal. This supports the accuracy of the MCGD technique. However, variations in SD of 27% to 45% were observed between the MCGD and base MCS methods. This would be worthy of further investigation and perhaps the inclusion of additional power flow analysis techniques to which comparisons can be made. The assumed correlation factors may have contributed to the
variations observed, however, this could not be investigated within the timeframes of the project.

Further time limitations included:

- The MCGD technique was not further extended to maximise the use of the developed multivariate Gaussian distributions.
- A more complex bus was not modelled.
- The computer algorithms developed to calculate statistical parameters from hourly load data (for a period of a year) were not used.
- Sensitivity analysis was not undertaken. Of particular interest would be assessing the effect of altering the assumed load type correlation factors on the standard deviations calculated by the MCGD method.

Worth noting:

- The Copula and MC base methods were undertaken using 50,000 random samples. Based on the convergence observed in the mean and SDs, this was considered a sufficient minimum.
- The Copula method was very computationally intensive. It was noted that the ‘ksdensity’ functions accounted for over 90% of the time taken.
  - Optimising the number of points considered and the Copula scaling mechanism will likely greatly reduce the calculation times, although, due to it still relying on a MC sampling mechanism, at best it be similar to the time taken for the base MC method.
10 Future Work

The generalised and analytical work in this paper could be extended to both control systems and more powerful modelling tools that abstract the user from the complexity of developing the network topology matrices and support a ‘pluggable’ system where power system elements can be added or removed. This should further allow for connectivity between microgrids and the main grid.

In addition, the estimation of correlation and standard deviations, used for convenience in this paper, should optimally be replaced with real values calculated from measured load data (preferably multiple years) or partial data extended and varied in a product such as Homer. It is not unfeasible to consider that a control system itself could regularly assimilate network data and statistical load data over the period of its operation.

Only one load level was considered in the statistical analysis of the network. Introducing a structure of various load levels will allow greater granularity of statistical data and, therefore, more accurate network modelling.

11 Outcomes

It was observed that the proposed method achieves significant reductions in calculation times when compared to iterative methods.

Implementing non-iterative probabilistic methods can improve microgrid performance by optimising control system algorithms.
12 Conclusions

In can be concluded that the use of non-iterative techniques, even in today's environment of high-speed computers, has significant calculation time benefits. This is of great importance for control systems in environments where the speed of probabilistic assessment is important.

Although excellent matching was obtained in voltage calculations, the relatively large differences in the standard deviations of the voltage (up to 45%) do not allow us to conclude that the method employed was sufficiently accurate in assessing the power flows in the network. It should be emphasised that this does not lead to the conclusion that the results obtained are inaccurate, it merely indicates that the scope of the comparative techniques chosen and time available for the research did not allow for sufficient comparisons and investigation to make conclusions on accuracy with sufficient confidence.

The commonality of the underlying load flow analysis technique, Direct Approach, likely resulted in the voltage magnitudes between methods being comparable, however, as standard deviations showed notable variation further insight into the accuracy of these results could be provided using additional power flow techniques, or perhaps a commercial software package.

As an aside, during research and implementation it was noted that the Copula MCS could be useful in other applications such as quantifying the correlation of wind on wind farms using the underlying Weibull distributions.
13 Appendix

13.1 References


doi:10.1109/TSG.2014.2385871.


doi:10.1109/TPWRD.2003.813818.


13.2 Scripts

13.2.1 MC Pi Simulation

```matlab
clc;
n=input('Number of points: ');
x=rand(n,1);
y=rand(n,1);
figure('color','white');
hold all
axis square;
x1=x-0.5; % circle has centre at (0.5,0.5)
y1=y-0.5; %
r=x1.^2+y1.^2;
m=0; % Number of points inside circle
for i=1:n
    if r(i)<=0.25
        m=m+1;
        plot(x(i),y(i),'b.');
    else
        plot(x(i),y(i),'r.');
    end
end
m/(0.25*n)
```
13.2.2 MC Copula Simulation

%30 Nov - Use this version
% R. Davis 2016
% MCGD Calculations as described in research project
% Set samples to take from underlying current distributions GD's
mcLoopCount=50000
mean_sd_real=[31.4918,5.249;21.782,3.630;23.619,3.937;26.768,4.461;28.343,4.724];
mean_sd_imag=[11.022,1.837;20.470,3.412;17.321,2.887;23.619,3.936;23.619,3.936];
current_real=[0;0;0;0;0];
current_imag=[0;0;0;0;0];
iCounter=1;
for mcloop=1:mcLoopCount
for l = 1:5
    current_real(mcloop,l)=normrnd(mean_sd_real(l,1),mean_sd_real(l,2));
    current_imag(mcloop,l)=normrnd(mean_sd_imag(l,1),mean_sd_imag(l,2));
end
PDFIRealLP2 = ksdensity(current_real(:,1),current_real(:,1),'function','cdf');
PDFIRealLP3 = ksdensity(current_real(:,2),current_real(:,2),'function','cdf');
PDFIRealLP4 = ksdensity(current_real(:,3),current_real(:,3),'function','cdf');
PDFIRealLP5 = ksdensity(current_real(:,4),current_real(:,4),'function','cdf');
PDFIRealLP6 = ksdensity(current_real(:,5),current_real(:,5),'function','cdf');
PDFIimagLP2 = ksdensity(current_imag(:,1),current_imag(:,1),'function','cdf');
PDFIimagLP3 = ksdensity(current_imag(:,2),current_imag(:,2),'function','cdf');
PDFIimagLP4 = ksdensity(current_imag(:,3),current_imag(:,3),'function','cdf');
PDFIimagLP5 = ksdensity(current_imag(:,4),current_imag(:,4),'function','cdf');
PDFIimagLP6 = ksdensity(current_imag(:,5),current_imag(:,5),'function','cdf');
% Set samples to take from Copula function
CorrSampleCount = 50000
% Preallocate for speed
v2_real= zeros(1,CorrSampleCount);
v3_real= zeros(1,CorrSampleCount);
v4_real= zeros(1,CorrSampleCount);
v5_real= zeros(1,CorrSampleCount);
v6_real= zeros(1,CorrSampleCount);
v2_imag= zeros(1,CorrSampleCount);
v3_imag= zeros(1,CorrSampleCount);
v4_imag= zeros(1,CorrSampleCount);
v5_imag= zeros(1,CorrSampleCount);
v6_imag= zeros(1,CorrSampleCount);
v2_complex= zeros(1,CorrSampleCount);
v3_complex= zeros(1,CorrSampleCount);
v4_complex= zeros(1,CorrSampleCount);
v5_complex= zeros(1,CorrSampleCount);
v6_complex= zeros(1,CorrSampleCount);
% Approximate ML
[Rho,nu] = copulafit('t',[PDFIRealLP2 PDFIRealLP3 PDFIRealLP4 PDFIRealLP5 PDFIRealLP6], 'Method','ApproximateML');
rReal = copularnd('t',Rho,nu,CorrSampleCount);
IrealCop_LP2 = rReal(:,1);
IrealCop_LP3 = rReal(:,2);
IrealCop_LP4 = rReal(:,3);
IrealCop_LP5 = rReal(:,4);
IrealCop_LP6 = rReal(:,5);
[Rho2,nu2] = copulafit('t',[PDFIimagLP2 PDFIimagLP3 PDFIimagLP4 PDFIimagLP5 PDFIimagLP6], 'Method','ApproximateML');
rImag = copularnd('t',Rho2,nu2,CorrSampleCount);
IimagCop_LP2 = rImag(:,1);
IimagCop_LP3 = rImag(:,2);
IimagCop_LP4 = rImag(:,3);
IimagCop_LP5 = rImag(:,4);
IimagCop_LP6 = rImag(:,5);
% Transform the random sample back to the original scale of the data.
% Use a subset or original data as per RowCount value below
RowCount = 2000
CorrPDFIRealLP2 = ksdensity(current_real(:,1:RowCount,1), IrealCop_LP2, 'function','icdf');
CorrPDFIRealLP3 = ksdensity(current_real(:,1:RowCount,2), IrealCop_LP3, 'function','icdf');
CorrPDFIRealLP4 = ksdensity(current_real(:,1:RowCount,3), IrealCop_LP4, 'function','icdf');
CorrPDFIRealLP5 = ksdensity(current_real(:,1:RowCount,4), IrealCop_LP5, 'function','icdf');
CorrPDFIRealLP6 = ksdensity(current_real(:,1:RowCount,5), IrealCop_LP6, 'function','icdf');
CorrPDFIimagLP2 = ksdensity(current_imag(:,1:RowCount,1), IimagCop_LP2, 'function','icdf');
CorrPDFIimagLP3 = ksdensity(current_imag(:,1:RowCount,2), IimagCop_LP3, 'function','icdf');
CorrPDFIimagLP4 = ksdensity(current_imag(:,1:RowCount,3), IimagCop_LP4, 'function','icdf');
CorrPDFIimagLP5 = ksdensity(current_imag(:,1:RowCount,4), IimagCop_LP5, 'function','icdf');
CorrPDFIimagLP6 = ksdensity(current_imag(:,1:RowCount,5), IimagCop_LP6, 'function','icdf');
% We now have $x$ amount of points of correlated current samples
% Do MCS using Copula function obtained values
for MCCurrentLoop=1:CorrSampleCount
    Vnew = -DLF*(transpose(CorrIreal(MCCurrentLoop,:)) + 1i*(transpose(CorrIimag(MCCurrentLoop,:)))) + Vinitial;
    v2_real(1,MCCurrentLoop) = real(Vnew(1));
    v3_real(1,MCCurrentLoop) = real(Vnew(2));
    v4_real(1,MCCurrentLoop) = real(Vnew(3));
    v5_real(1,MCCurrentLoop) = real(Vnew(4));
    v6_real(1,MCCurrentLoop) = real(Vnew(5));
    v2_imag(1,MCCurrentLoop) = imag(Vnew(1));
    v3_imag(1,MCCurrentLoop) = imag(Vnew(2));
    v4_imag(1,MCCurrentLoop) = imag(Vnew(3));
    v5_imag(1,MCCurrentLoop) = imag(Vnew(4));
    v6_imag(1,MCCurrentLoop) = imag(Vnew(5));
    v2_complex(1,MCCurrentLoop) = Vnew(1);
    v3_complex(1,MCCurrentLoop) = Vnew(2);
    v4_complex(1,MCCurrentLoop) = Vnew(3);
    v5_complex(1,MCCurrentLoop) = Vnew(4);
    v6_complex(1,MCCurrentLoop) = Vnew(5);
end
% Calculate mean and SD from data generated in MCS sampling of Copula
VrealmeanMC=[mean(v2_real);mean(v3_real);mean(v4_real);mean(v5_real);mean(v6_real)]
VrealstdMC=[std(v2_real);std(v3_real);std(v4_real);std(v5_real);std(v6_real)]
VimagmeanMC=[mean(v2_imag);mean(v3_imag);mean(v4_imag);mean(v5_imag);mean(v6_imag)]
VimagstdMC=[std(v2_imag);std(v3_imag);std(v4_imag);std(v5_imag);std(v6_imag)]
VcomplexmeanMC=[mean(v2_complex);mean(v3_complex);mean(v4_complex);mean(v5_complex);mean(v6_complex)]
VcomplexstdMC=[std(v2_complex);std(v3_complex);std(v4_complex);std(v5_complex);std(v6_complex)]
% Return to void plots
% return;
figure;
scatterhist(current_real(:,1),current_real(:,2))
title('Real Current - GD Sampled')
grid on
xlabel('LP2')
ylabel('LP3')
figure;
scatterhist(current_imag(:,1),current_imag(:,2))
title('Imaginary Current - GD Sampled')
grid on
xlabel('LP2')
ylabel('LP3')
figure;
scatterhist(CorrPDFIRealLP2,CorrPDFIRealLP3)
title('Real Current - MGD Copula Sampled')
grid on
xlabel('LP2')
ylabel('LP3')
figure;
scatterhist(CorrPDFIimagLP2,CorrPDFIimagLP3)
title('Imaginary Current - MGD Copula Sampled')
grid on
xlabel('LP2')
ylabel('LP3')
%MCGD Calculations for research project
%R. Davis - 2016

Power at nodes 2,3,4,5,6 - Given for sample network

VpMean = [600; 415; 450; 510; 540];
VqMean = [210; 390; 330; 450; 450];
Initial estimates
Vinitial = [11; 11; 11; 11; 11];
Vbase = 1000*11;

%Calculate current
IpMean = VpMean/Vbase/sqrt(3);
IqMean = VqMean/Vbase/sqrt(3);
IcMean = IpMean + 1i*IqMean;

%Calculating SD from (u*err/300)
err = 50;
SDp = IpMean*err/300;
SDq = IqMean*err/300;

%Get impedance from given resistance and reactance
Z = [1.329 + 1.1175i; 1.5948 + 1.3411i; 1.4155i; 1.5062 + 1.2665i; 1.5948 + 1.3411i];

%Future work, add algorithm to construct this as per JNL paper
BIBC = [1,1,1,1,1; 0,1,1,1,1; 0,0,1,1,0; 0,0,0,1,0; 0,0,0,0,1];

%Corr matrix maps disimilar load types using 0.4, same load type as 0.6 and self mapping as 1
Corr = [1,0.6,0.4,0.6,0.4; 0.6,1,0.4,0.6,0.4; 0.4,0.4,1,0.4,0.4; 0.6,0.6,0.4,1,0.4; 0.4,0.4,0.4,0.4,1];

%Calculate Covariance matrices for current
for colNo = 1:5
    currSDp = SDp(colNo);
    currSDq = SDq(colNo);
    for rowNo = 1:5
        COVpp(rowNo,colNo) = Corr(colNo,rowNo) * currSDp;
        COVqq(rowNo,colNo) = Corr(colNo,rowNo) * currSDq;
        COVqp(rowNo,colNo) = 0.9 * Corr(colNo,rowNo) * currSDp;
        COVpq(rowNo,colNo) = 0.9 * Corr(colNo,rowNo) * currSDq;
    end
end

%Additional constructs
%Impedance matrix
%Specifically set this, but look to make generic in future
BCBV = zeros(5,5);
for colNo = 1:5
    switch (colNo)
        case 1
            BCBV(1, colNo) = Z(1);
            BCBV(2, colNo) = Z(1);
            BCBV(3, colNo) = Z(1);
            BCBV(4, colNo) = Z(1);
            BCBV(5, colNo) = Z(1);
        case 2
            BCBV(2, colNo) = Z(colNo);
            BCBV(3, colNo) = Z(colNo);
            BCBV(4, colNo) = Z(colNo);
            BCBV(5, colNo) = Z(colNo);
        case 3
            BCBV(3, colNo) = Z(colNo);
            BCBV(4, colNo) = Z(colNo);
        case 4
            BCBV(4, colNo) = Z(colNo);
        case 5
            BCBV(5, colNo) = Z(colNo);
    end
end

DLF = BCBV * BIBC

GammassI = COVpp + COVqq + j*(COVqp - COVpq)
CssI = COVpp - COVqq + j*(COVpq + COVqp)

VmeanOut = -DLF * IcMean + Vinitial
VmeanOut = real(VmeanOut)
VmeanOut = imag(VmeanOut)

%Now calculate variance per load point for voltage
GammassV = -DLF * GammassI * (-DLF)'
CssV = -DLF * CssI * (-DLF)'

%Calculating SD from (u*err/300)
err = 50;
SDp = IpMean*err/300;
SDq = IqMean*err/300;

%Get impedance from given resistance and reactance
Z = [1.329 + 1.1175i; 1.5948 + 1.3411i; 1.4155i; 1.5062 + 1.2665i; 1.5948 + 1.3411i];

%Future work, add algorithm to construct this as per JNL paper
BIBC = [1,1,1,1,1; 0,1,1,1,1; 0,0,1,1,0; 0,0,0,1,0; 0,0,0,0,1];

%Corr matrix maps disimilar load types using 0.4, same load type as 0.6 and self mapping as 1
Corr = [1,0.6,0.4,0.6,0.4; 0.6,1,0.4,0.6,0.4; 0.4,0.4,1,0.4,0.4; 0.6,0.6,0.4,1,0.4; 0.4,0.4,0.4,0.4,1];

%Calculate Covariance matrices for current
for colNo = 1:5
    currSDp = SDp(colNo);
    currSDq = SDq(colNo);
    for rowNo = 1:5
        COVpp(rowNo,colNo) = Corr(colNo,rowNo) * currSDp;
        COVqq(rowNo,colNo) = Corr(colNo,rowNo) * currSDq;
        COVqp(rowNo,colNo) = 0.9 * Corr(colNo,rowNo) * currSDp;
        COVpq(rowNo,colNo) = 0.9 * Corr(colNo,rowNo) * currSDq;
    end
end

%Additional constructs
%Impedance matrix
%Specifically set this, but look to make generic in future
BCBV = zeros(5,5);
for colNo = 1:5
    switch (colNo)
        case 1
            BCBV(1, colNo) = Z(1);
            BCBV(2, colNo) = Z(1);
            BCBV(3, colNo) = Z(1);
            BCBV(4, colNo) = Z(1);
            BCBV(5, colNo) = Z(1);
        case 2
            BCBV(2, colNo) = Z(colNo);
            BCBV(3, colNo) = Z(colNo);
            BCBV(4, colNo) = Z(colNo);
            BCBV(5, colNo) = Z(colNo);
        case 3
            BCBV(3, colNo) = Z(colNo);
            BCBV(4, colNo) = Z(colNo);
        case 4
            BCBV(4, colNo) = Z(colNo);
        case 5
            BCBV(5, colNo) = Z(colNo);
    end
end

DLF = BCBV * BIBC

GammassI = COVpp + COVqq + j*(COVqp - COVpq)
CssI = COVpp - COVqq + j*(COVpq + COVqp)

VmeanOut = -DLF * IcMean + Vinitial
VmeanOut = real(VmeanOut)
VmeanOut = imag(VmeanOut)

%Now calculate variance per load point for voltage
GammassV = -DLF * GammassI * (-DLF)'
CssV = -DLF * CssI * (-DLF)'
%Calculate voltage covariances

\[
\text{COV}_{vrr} = 0.5 \times \text{real}(\text{GammassV} + \text{CssV})
\]
\[
\text{COV}_{vri} = 0.5 \times \text{imag}(-1 \times \text{GammassV} + \text{CssV})
\]
\[
\text{COV}_{vir} = 0.5 \times \text{imag}(\text{GammassV} + \text{CssV})
\]
\[
\text{COV}_{vii} = 0.5 \times \text{real}(\text{GammassV} - \text{CssV})
\]

% l = load point number minus 1
Vvariance=zeros(5,1)
Vstd=zeros(5,1)
for l=1:5
    Vvariance(l,1) = (VpmeanOut(l), VqmeanOut(l)) * [COV_{vrr}(l,l), COV_{vri}(l,l); COV_{vir}(l,l), COV_{vii}(l,l)] *
    (VpmeanOut(l))^2 + (VqmeanOut(l))^2;
    Vstd(l,1)=sqrt(Vvariance(l,1));
end
%Display
Vvariance
Vstd