



UNIVERSITÀ DEGLI STUDI DI PADOVA

FACOLTA' DI INGEGNERIA

Corso di laurea in Ingegneria Elettrotecnica

**Smart Meters and Voltage estimation in radial
distribution networks**

Relatore: Prof. Roberto turri

Correlatore: Dott. Ekanayake Janaka

Laureando: Diego Beggiato

ANNO ACCADEMICO 2009/2010

Contents

Abstract

Introduction

1 Smart Meters

- 1.1 INTRODUCTION
- 1.2 SMART METERING IN THE UK-POLICY HISTORY AND CONTEXT
- 1.3 METERING MARKET IN UK & SMART METERS TECHNOLOGY
 - 1.3.1 METERING MARKET IN UK
 - 1.3.2 SMART METERS TECHNOLOGY
- 1.4 SMART METERING EXPERIENCE AND STUDIES
 - 1.4.1 INTERNATIONAL EXPERIENCE
 - 1.4.2 STUDIES OF THE IMPACT OF SMART METERING ON ENERGY SAVING
- 1.5 COSTS AND BENEFITS OF SMART METERS
 - 1.5.1 COSTS OF SMART METERS
 - 1.5.2 POTENTIAL BENEFITS OF SMART METERS
- 1.6 NEW TARIFFS AND NEW SERVICE

2 Load Flow for lower voltage system

- 2.1 INTRODUCTION
- 2.2 MATHEMATICAL FORMULATION
- 2.3 SOLUTION METHODOLOGY
- 2.4 APPLICATION
 - 2.4.1 NETWORK 1 (20000 V)
 - 2.4.2 NETWORK 2 (400 V)

3 State Estimation for lower voltage system

- 3.1 NODES VOLTAGE ESTIMATION (Special case)
 - 3.1.1 SOLUTION METHODOLOGY
 - 3.1.2 APPLICATION
- 3.2 NODES VOLTAGE ESTIMATION (Generalized case)
 - 3.2.1 FIRST PROGRAM (USES THE VOLTAGE MEASURES FROM S.M. ONLY)
 - 3.2.1.1 SOLUTION METHODOLOGY
 - 3.2.1.2 APPLICATION
 - 3.2.2 SECOND PROGRAM (USES ALL MEASURES FROM S.M.)
 - 3.2.1.1 SOLUTION METHODOLOGY
 - 3.2.1.2 APPLICATION

Conclusions

References

Appendix

Abstract

The management of distribution networks is changing. In UK like in other world country the idea to install smart meters in every building is now reality. The use of smart meters will introduce many benefits but also problems about their technology. One of problems is that some measures could be reported late or some smart meters could work incorrectly.

The paper presents an algorithm to estimate the voltage in radial distribution networks, where there are some measurements not available from smart meters.

The method, based on the assumption of some missing data refers to backward and forward sweep.

This is implemented in Matlab allows the estimation voltage with a very small error.

The report also highlights some success factors associated with smart meters.

Introduction

The recent legislative actions of the European Union (EU) and the UK's initiatives to achieve EU's 20/20/20 targets in CO₂ emission reductions, a massive deployment of smart meters is expected by 2020. The same targets expect significant increase of distributed generation and micro-generation. With the introduction of DGs in low voltage systems, voltage control will become difficult as the power flow will not be unidirectional any longer. The changes of power flow directions may violate the statutory voltage limits and also higher nodal voltages limit the DG's output power.

Traditionally the voltage control is being done through On Load Tap Changers (OLTC). The voltage controller of OLTCs usually uses local measurements and few remote measurements obtained from remote terminal units (RTU) to set the voltage at the primary substation such that the entire distribution network will be within the required voltage limits. With the addition of many DGs, a better control of distribution voltage is required which demands more data from distribution network.

Though deploying RTUs to collect more data is expensive, the smart meters will avail a better opportunity to collect required data from the distribution network. However processing this data to estimate the state of the distribution network is challenging due to:

- (a) the smart meter measurements are not heterogeneous;
- (b) communication channels used for smart meters may introduce delays as long as several minutes;
- (c) there may be erroneous measurements.

The broad scope of this research is to use smart meters' measurements for controlling the voltages of distribution network. As a part of this research, my target was to find at the ways of using the delayed or erroneous smart meter measurements and nodes where measurements are not at all available to estimate the voltages of each distribution network node and MV node using radial load flow analysis.

This report is the result of a project carried out for Cardiff University, particularly for the Institute of Energy. The Institute of Energy is an innovative and pioneering research centre with a mission to advance energy technology and play a key role in addressing the increasing demand for sustainable and low carbon technologies.

The Institute encompasses high quality research, education and training to address predicted skills shortages in the field of sustainable energy as the UK works towards the Government's targets for the reduction of CO₂ emissions.

The report also highlights some success factors associated with smart meters.

The government has an aspiration that all consumers will have a smart energy meter within ten years.

The potential consumer benefits from smart meters are considerable. These include:

- improved retail competition;
- more accurate bills;
- lower bills due to better customer feedback;
- reduced costs and increased convenience for pre-pay;
- less environmental pollution due to reduced carbon emissions;
- new tariffs; and
- new services for consumers, including vulnerable consumers.

This report is divided into two parts, the first identifies and examines the potential benefits and shortcomings of smart meter, and the second describes how the measures of smart meters can be used to estimate the voltages in a distribution network.

CHAPTER 1

Smart Meters

1.1 Introduction

The UK energy meter stock remains largely electro-mechanical and has not changed significantly in over a century, in marked contrast to many other technologies.

Advanced metering technology has a thirty year history in the UK but earlier initiatives had all but stalled until being given a new impetus by the EU Energy End-Use Efficiency and Energy Services Directive.

Smart meters are not an end in themselves but are an important gateway:

- *For energy suppliers:* to improve market operation through better ways of tackling energy management, and, finding new retail opportunities.
- *For SMEs and households:*
 - to achieve energy savings through improved feedback on energy consumption and expenditure;
 - to develop demand-response at an individual level;
 - to develop new scope for micro-generation.

The Energy Review in particular offers the opportunity to develop firm recommendations and a clear policy road map for the development of smart meters in the UK.

The government has an aspiration that all consumers will have a smart meter within ten years and is considering how this might be achieved [1].

Smart meters can record how much electricity or gas is used and at what time. The meter can store this information and then transmit it electronically to the energy supplier.

Consumption information can be displayed to consumers via a display in the home, or via the web or a mobile.

The potential consumer benefits from smart meters are considerable. These include [1]:

- improved retail competition;
- more accurate bills;
- lower bills due to better customer feedback;
- reduced costs and increased convenience for pre-pay.
- Less environmental pollution due to reduce carbon emission;
- New tariffs and new services.

This chapter identifies and examines the history and the potential benefits of smart meters under the following headings:

- *Smart metering in the UK-Policy history and context*
- *Metering market in UK & Smart Meters technology*
- *Smart metering experience and studies*
- *Costs and benefits of Smart Meters*
- *New tariffs and new service*

1.2 Smart metering in the UK-Policy history and context

Advanced metering in the UK has a thirty-year history, particularly for electricity. From the mid 1980's onwards, the former Electricity Council and a number of the Area Boards, actively explored 'new metering technologies', and initiated major field trials. Subsequently, energy-industry re-structuring and the introduction of retail-competition with associated metering developments, meant that few new metering initiatives took place at the residential and SME-level in the period 1990-98. From 2000 onwards, three significant policy reports were produced on smart metering in the UK, but few notable new policy or regulatory responses emerged [2].

- 1) April 2000: Review of the energy efficiency and other benefits of advanced utility metering.
Produced for BREMA by EA Technology.
Wide-ranging and authoritative desk-review of research on metering and energy saving. Inter alia, concluded that improved customer displays and tariff advice were central.
- 2) September 2001: DTI Smart Metering Working Group.
Made a recommendation for pilot studies 'to help determine whether the introduction of smart meters could result in benefits to consumers and companies and also help contribute to the achievement of Government. It seems that, in the absence of Treasury and Cabinet Office support, these pilots did not proceed.
- 3) May 2002: The new metering, Technology working Group.
An industry-wide group, convened by Elexon, reported at Ofgem's request.
Ofgem is the Office of the Gas and Electricity Markets.
They concluded that 'policies and decisions must be made now to enable the UK energy industry and their customers to benefit from advanced metering technology'.

In May 2002, Ofgem produced a final Metering Strategy, in expectation that 'Metering competition can deliver significant savings for consumers and encourage the application of innovative metering technologies'. A Metering Innovation Working Group, met intermittently, until 2005 [2].

The PIU Study (Feb 2002) and the Energy White Paper (Feb 2003) gave a new policy focus to demand-side and energy efficiency issues, with a succession of follow-on consultations, reviews and policy initiatives, but neither addressed smart metering in any detail. The 2002 policy-push on smart meters therefore seemed to have all-but stalled, but prospects of higher energy retail-prices, coupled with the risk of failing to meet UK targets for CO₂ reduction in 2010 and new EU-policy [2].

Other initiatives were produced on smart metering:

- 1) September 2005: Energywatch-
Initiated a widely welcomed debate on smart metering.
- 2) From July 2007 : In Europe.EU energy liberalisation legislation requiring full opening of electricity and gas retail markets from July 2007 has served as a catalyst in some member states for residential smart meters, where they are seen as a pre-condition for successful domestic retail competition.
- 3) November 2005: The energy End-use Efficiency and Energy Services Directive.
Finally reached political agreement in November 2005, with an implementation date of May 2008.
Along with a number of energy-saving measures, member states shall ensure that, in so far as it is technically possible, financially reasonable and proportionate in relation to the potential energy savings, final customers for electricity, natural gas, are provided with competitively priced individual meters that accurately reflect the final customer's actual energy consumption and that provide information on actual time of use.
When a new connection is made in a new building or a building undergoes major renovations as set out in Directive 2002/91/EC, such competitively priced individual meters shall always be provided.

Directive 2002/91/EC of the European Parliament and the council of 16 December 2002 concerns the energy performance of buildings.

Article 13.1:

Billing is based on actual energy consumption, and is presented in clear and understandable terms. Billing on the basis of actual consumption shall be performed frequently enough to enable customers to regulate their own energy consumption [6].

Article 13.2:

There are also additional billing-related requirements concerning tariff information and comparative historic consumption data in graphic format [6].

4) January 2006: Government Energy Review 2006.

This questions what more Government could do on the Demand-side, to ensure that carbon emission goals are met, and that homes are adequately and affordably heated.

5) February 2006: Domestic Metering Innovation.

Ofgem has been conducting an exercise to review progress on meter innovation in the context of the competitive regulatory framework for meter provision and meter services.

This review of the potential for smart metering therefore takes place at a crucial time.

In particular, the Ofgem consultation, together with the forthcoming consultation on the Energy End-Use Efficiency Directive, offer a major opportunity to define a clear policy pathway for smart meters, and to enable firm recommendations to feed into the Energy Review [2].

Thereafter, strong and continuing leadership by DTI, with appropriate support from Ofgem and DEFRA (Department for Environment, Food and Rural Affairs), will be a fundamental precondition to achieving a coherent policy framework, able to deliver smart-meters at a residential and SME level in the UK [2].

1.3 Metering market in UK & Smart Meters technology

1.3.1 Metering market in UK

Energy suppliers are responsible for [2]:

- making metering arrangements on behalf of their customers and can arrange for any type of meter to be installed, subject to it conforming to basic accuracy and safety requirements;
- Suppliers contract with others who provide meter assets, operate meter services (installation, maintenance, repair, replacement etc) and read meters. Data collection and aggregation for billing may also be undertaken as a separate service. Data processes and data flows are subject to common governance arrangements.

The UK meter sector therefore comprises [2]:

- the six large energy suppliers;
- a number of smaller ones;
- the fourteen geographic distributors (Geographic footprint of former Public Electricity Suppliers);
- four gas transporters (National Grid Gas (formerly Transco and Transco Metering Services), and three Independent Gas Transporters);
- and a number of stand-alone meter.

Some electricity and gas licensees are developing non-licensed meter businesses, and others are selling their meter assets [2].

An important input about SM will arrive from Renewable Energy, in fact a growing meter requirement for small-scale renewables and micro-generation, needing import / export measurement, represents a potentially sizeable new market segment.

1.3.2 Smart Meters Technology

There is no single definition of smart metering.

A smart-meter system comprises:

- An electronic box;
- a communications link.

At its most basic, a smart meter:

“measures electronically how much energy is used, and can communicate this information to another device”

For both electricity and gas, there are two main smart-meter types [2]:

- 1) **AMR** – *Automated Meter Reading*
One-Way Communication from the Meter to the Data Collector, as a minimum enabling *Automated Meter Reading*.
- 2) **AMM** - *Automated Meter Management*
Two-Way Communication between the Meter and the Supplier, enabling a wider range of functions known as *Automated Meter Management*. A further refinement of the AMM meter is an *Interval Meter*, a two-way meter with a capability to store and communicate consumption data by time-of-use (e.g. half-hourly intervals).

The key distinction between smart-meter types is therefore determined by:

- *their communication*, i.e. whether one-way or two-way;
- *data-storage capability*. These basic meter *capabilities* then determine the *functionality* that the meter might offer.

The following list captures the core capabilities that are consistently identified as particularly important for a smart meter system [2]:

- Measures energy consumed - quantity and when (a time-interval basis);
- Records ‘billing-level’ readings;
- Two-way communication;
- Stores interval-data and transfers it remotely to a data collector / utility;
- Capable of storing and displaying consumption and tariff information.

Communications combinations are still evolving with fixed, mobile, wireless, narrow-band and broadband options available. Reliability and data-accuracy are a recurring theme, whatever the chosen communications technology. Broadly speaking, communications from the meter to an *initial* data-collection point can be grouped between :

- **Fixed** e.g.
 - telephone landline - PSTN, ISTN, Cable/ADSL;
 - or the electricity distribution wires - Power Line Carrier (PLC).
- **Wireless** e.g.
 - GSM (system used by mobile phones);

- various forms of radio communications such as GPRS and Low Power Radio(LPR).

Different communications technologies seem best suited to different needs as follows.

- **GSM** - would allow Suppliers considerable flexibility in targeting particular customer groups, including SMEs, pre-payment or remote residential customers.
- **PLC – LPR** – are well suited to geographic-based smart meter rollout and are being extensively and successfully deployed in Italy and Scandinavia, for example.

In a number of countries where smart meters are being installed a range of communications technologies are being deployed to meet different needs, and many valuable lessons are being learned.

The importance of visual display

A lot of company as DEFRA, Energy Saving Trust, Energywatch, the Carbon Trust and the Association for the Conservation of Energy stressed that improved consumption feedback to create better awareness of energy use and carbon emissions, is potentially one of the most important ‘gains’ from a move to smart metering [2]. It is therefore important to note [2]:

- Meters may well be positioned out-of-sight, in fact most household meters are in under-stair or external cupboards.
- Smart meters will not necessarily have a user-friendly consumer display showing usage, expenditure and tariffs, unless part of a meter specification, as a separate consumer-display elsewhere in the home, wireless-linked to a smart meter is possible, but would add to costs.
- Even with smart meters, billing may well remain the main channel for providing consumers with feedback.
- Only two-way interval meters have the capability to display real-time time-of-use prices, to which consumers can actively respond.
- Consumers may choose a tariff package from their supplier, linking time-of-day with different tariffs, obviating the need for a display.
- Consumers could obtain data via other means, for example via the internet, interactive TV or mobile phone, provided they have a smart-meter which links to an external communications medium.
- Consumers could find consumption-breakdown between lighting and other electrical circuits of interest. However, separate measurement devices on each internal circuit would be needed to achieve this.

1.4 Smart metering experience and studies

1.4.1 International Experience

Much of the international experience (especially in the US) is with AMR only, was developed many years ago and driven by a tradition of monthly billing and hence a strong business case to cut meter reading costs [2]. The AMR systems used in the US have mainly involved fitting a small communications device to the meter to enable the utility to read meters from a van driving by houses, obviating the need to get into the property.

This will be of limited interest in the UK context given that [2]:

- the relative costs of AMM and AMR are closer;
- quarterly billing reduces the business case of AMR on its own;
- the main potential benefits (for the energy industry, and especially for customers and public benefits) are from AMM.

The focus therefore is one those examples that are AMM (with or without interval metering).

Generally, in most of the international examples, the benefits for the energy suppliers and distributors do not add up on their own to a business case for smart metering.

The exceptions to this are:

- *ITALY*, where particular circumstances made it sensible for ENEL to do smart metering, although even there the regulator has allowed some cost recovery;
- *SWEDEN*, where monthly meter reading created a business case for AMR.
- *NORTHERN IRELAND*, where the form of prepayment metering used was high cost, thus justifying a smart metering rollout, but only of prepayment meters.

The main societal or economy wide benefits identified by the use of smart metering are:

- 1) In **California, Ontario, Victoria, Italy**
Reductions in peak demand leading to better security of supply, less risk of blackouts.
- 2) In **California, Ontario, Victoria**
Reductions in peak demand leading to lower costs due to less need for peaking plant and power imports and avoided investment in distribution and transmission network upgrades.
- 3) In **Victoria**
Improved balance between supply and demand leading to greater efficiency of operation of the electricity wholesale market.
- 4) In **Netherland, Victoria, Sweden**
Customers enabled to switch supplier more easily, plus new service offers made possible by smart meters.
- 5) In **Sweden**
Environmental benefits from reduced power demand.

In each of these countries, a positive cost/benefit case has been established without taking into account any benefits that might accrue in terms of reduced carbon emissions [2].

1.4.2 Studies of the impact of smart metering on energy saving

It is generally assumed that smart metering, by providing more information to consumers, will encourage more efficient use and lead to energy savings in a number of countries.

A Wright et al.[7] identify eight techniques used to improve information and lead to energy savings in a number of countries. These include:

- 1) improved billing information;
- 2) customer displays;
- 3) load control via the meter;
- 4) use of the meter as a communications gateway to the home for load control and other services.

The main evidence is from Norway where the saving was on average 4% three years after consumption information was introduced compared to a 4% increase in households (so equates to an 8% fall) [2].

In UK, Wright et al. conclude by saying that “it is not possible to say how much energy would be saved [2]. Better billing feedback produced savings of up to 10% in electrically heated homes in cold climates, mainly using simple manual methods. In the absence of electric space heating smaller savings are likely, but some of the automatic measures here could produce new types of saving, for example in refrigeration which would not be possible manually. Load shifting is easier than load reduction, so cost savings are easier to achieve than energy savings, but both would probably lie in the 0-5% range for a home without electric heating.”

Studies made and therefore the evidences do suggest that the energy saving and environmental effects of smart metering should be great if [2]:

- clear, well positioned and easy to understand customer displays are provided,
- time of day or time of use tariffs are introduced alongside smart meters;
- information from smart metering is reinforced by other action, e.g:
 - offers of energy saving measures/packages
 - advice on behavioural changes
 - offers of micro-generation packages

Smart metering thus could have a role to play in an overall energy saving strategy that also encompasses these other measurements.

Work by the Carbon Trust suggests that there may be considerable potential for energy savings in the business sector through greater use of smart metering and associated services.

Early results from their trial of advanced metering plus professional advice on energy saving options in 575 SME premises have shown savings of 5% on average [8].

Evidence on peak demand reductions

There are three main potential benefits from peak demand reductions [2]:

- 1) Cost saving,
if it reduces need to upgrade networks or construct new power stations.
- 2) Improvement in security of supply,
as blackouts can be avoided and reliance on fuel imports reduced.
- 3) Possible emissions reductions,
but these depend upon what is the marginal generating plant and so could vary quite significantly,
subject to coal or gas-burn at peak.

Wright et al. [7] said that the carbon benefits of load shifting in the UK would be very small around 0.3%.

The main driver for peak demand reductions has been to improve security of supply but also to some extent to secure cost savings.

Potential emission reductions have not been an important driver.

The UK to achieve a substantial load-shift can:

- In the future can focus on smart-metering because they could achieve some load-shifting;
- In the present, it has already achieved a substantial load-shift to off-peak periods through off-peak tariffs which have been available for electricity, and particularly for some forms of electric space-heating in the UK, since the 1970s, and have been used in conjunction with automatic teleswitching of appliances or the use of simple timers.

However, the ability to use smart-meters to offer:

- a new range of time-of-use tariffs;
- and some additional automatic switching-on and off of appliances;
- or providing messages to consumers to switch off in expensive periods,

has opened up new possibilities.

In California, which is a key source of evidence of smart metering impact on peak demand, the serious power cuts experienced in 2001 acted as a spur to the use of smart metering [2].

A statewide pilot involving 2500 residential and small commercial customers, to study demand response to critical peak pricing with smart meters [2].

There were two different modes of action of smart meters:

- Some in the trial had automated response (the meter was linked to appliances and could change thermostat settings or switch off);
- others were given information about when prices were high for them to respond.

The effect ranged from [2]:

- 27% reductions (with automated response at the highest critical peak prices)
- to more typical 5-10% reductions without automated response.

However, the UK differs from many other countries where peak demand has been a key driver for smart metering because fewer than 10% of households use electricity for heating and there is virtually no air conditioning in households.

The household potential will be mainly:

“to switch washing machines, dishwashers and tumble driers to off-peak”

because many appliances need to be used when the need arises and so cannot be shifted to off-peak.

The main potential may be in the commercial sector where load shifting could be very significant due to use of air conditioning and refrigeration.

There are two areas where further work will be beneficial in the UK context:

- 1) on the potential effect of more clearly differentiated tariffs on residential appliance-use;
- 2) reduction of carbon emission as a result of cut peak and other changes to the shape of daily load curve.

1.5 Costs and benefits of Smart Meters

1.5.1 Costs of Smart Meters

Costs can vary considerably depending upon volume and technology assumptions, and there are a number of ways of assessing these [2].

For commercial reasons, much cost information provided to this study was broadly generic, and accordingly the costs below reflect a synthesis of two main sources of information:

- 1) known costs where smart metering has been introduced overseas;
- 2) based on certain volume and technology assumptions in the UK.

Knowing that a cost of basic meter is around £7-8 plus installation costs [2], see the costs of advanced meters.

There is no significant difference between the unit cost of AMR and AMM meters . The costs which follow are therefore only for AMM, given the greater functionality that this will provide.

The choice of communications technology has a significant impact on costs:

- PLC (Power line carrier);
- LPR low power radio and Fixed network radio;
- GSM, this is more expensive than other two.

The use of fixed telephone lines tends to be considered too expensive compared to other options.

The figures below are based on a number of different sources in the UK and international experience:

COST OF	£ (pounds)	
Base meter	15-17	
Specific functionality (each)	5-7	
Micro-generation	1-7	
Choice of communications	17-20	PLC
	12-15	LPR
	35-40	GSM

Table 1.1: Figures about Smart Meter [2]

Where:

Base meter ---> includes more functions than a basic meter and has appropriate software.

Specific functionality ---> for example they can be:

- an interval / data storage (enabling time-of-use) facility;
- a facility to operate in prepay mode (using keypad or smart card for prepayment);
- a contactor to connect and disconnect supply;
- a customer display on the meter itself.

Micro-generation ---> the ability to meter micro-generation at the customer's premises
i.e: to charge one rate for electricity supplied by the utility and credit the customer with a different rate for the electricity exported.

Choice of communications---> PLC (Power line carrier or fixed line radio)
LPR (Low power radio)
GSM (Global System for Mobile Communications)

Potential cost ranges for advanced meters will be:

- AMM + PLC ---> £37-62
- AMM + LPR ---> £32-57
- AMM + GSM ---> £54-80

Other cost must be added:

COST OF	£ (pounds)
Installation	20-30
Data Infrastructure / Systems	15-25
Separate Customer Display	28-45

Table 1.2 Add costs [2]

Where:

Installation ---> should not vary according to the type of meter (although GSM/LPR antenna positioning

might cause some additional site costs), but will vary with volume installed and whether meters are being rolled out on a geographic basis or more selectively.

Data Infrastructure / Systems ---> that need to be set up within the utility.

These will vary significantly in terms of cost per meter, depending upon volume, as there will be a high level of set up costs associated even with a small volume of meters installed.

Separate Customer Display ---> connected by PLC or wireless to the smart meter but conveniently located for customer use (e.g. kitchen).

Potential capital cost ranges for advanced meters will be:

- AMM + PLC or LPR ---> £70-115 (£98-160 including separate display)
- AMM + GSM ---> £95-135 (£98-160 including separate display)

Assuming a meter price of £120 installed, and an asset life of 15 years this gives a simple annualised cost per meter of £8, plus operating and maintenance costs of £5-10, making a total of £13-18.

For comparison in California cost per customer per year is US \$24; in Ontario CAN \$36-48, but Meter unit costs would fall considerably with larger volumes [2].

Supplier costs that may be impacted by smart meters

To assess whether the costs of smart meters are exceeded by the benefits of introducing them, it is useful first of all to examine the costs incurred by suppliers in metering and billing customers.

COST OF	DESCRIPTION	£ (pounds)
Meter reading	£6 per annum average to read a meter per 45 million customers	270 million
Call center activity	Average £10 per customer per 45 million customers	450 million
Billing	£4 per customer average per 45 million customers	180 million
Supplier switching	missing billing and problems associated with supplier switching	100-200 million
Lost revenue due to fraud/theft/bad debt	some estimates suggest that they are around 2% of turnover	100 million

Table 1.3: Quantified costs [2].

So the total cost of the above is in the range of £1.1 billion - £1.4 billion [2].

1.5.2 Potential benefits of Smart Meters

The main beneficiaries will be: suppliers; network operators and customers.

In addition there are potential public benefits (environment, security of supply, contribution to fuel poverty targets), but caution is needed to avoid double counting of benefits.

Supplier benefits

For suppliers a major attraction of smart meters could be the potential to reduce some of the above costs and to increase revenue.

Scope for cost reductions for suppliers:

- *Meter reading:* Smart meters would eliminate the need for a manual read, although if suppliers are still required to inspect meters once every two years then the cost savings will be lower.
- *Call center activity:* Smart metering should significantly reduce call centre activity due to the elimination of estimated bills and the need for customers to read meters.
- *Billing and payment:* Smart metering would eliminate estimated bills, but prepayment meter customers might still buy credit frequently so Costs particularly high for use of Paypoint.
- *Fewer call-outs:* Cost savings would result from smarter prepayment meters with fewer problems leading to call outs.
- *Switch-over to remove application:* There would be no need to visit to change tariffs or read meters for the token electricity prepayment meters.
- *Bad debt reduction/better detection of fraud/theft:* Due to elimination of estimated bills, scope for more frequent billing and automated detection of theft. Hence potential reductions in revenue lost.
- *Load management/electricity purchasing:* better data potentially enables ---> better management of power purchase agreements.

It is difficult to quantify what these cost savings might add up to, but discussions with suppliers suggest it is reasonable to assume that at least 15-25% of supplier costs outlined above might be saved from the introduction of smart metering [2].

Revenue improvements for suppliers:

- *Improved use of working capital:* the gap between reading meter, billing and collecting money all shortens, hence a cash flow benefit as money comes in faster in relation to costs incurred.
- *New retail packages:* Ability to offer customers new packages, e.g: time of use tariffs and energy services micro-generation.
- *Wider scope for Pay-as-You-Go:* More ability to use pay as you go because it becomes more attractive– e.g. for students, houses in multiple occupation, properties where tenants change frequently, some SMEs.

The last point could be categorised as cost reduction (reducing debt problems), but the main benefit may be revenue enhancement from better cash flow and possibly the willingness of customers to pay more for a better pay as you go service.

The above analysis suggests that supplier annual cost savings might be about £5 on average per smart meter installed, compared to a cost of £13-18. Therefore there is currently no business case for suppliers to introduce smart meters on a widespread basis [2].

Finally it is worth noting that suppliers may lose revenue if smart metering reduces total energy demand and this may act as a disincentive.

Nevertheless there is a business case for suppliers to introduce smart meters to some segments of the market:

- Token pre-payment meters;
- Credit customers who are costly to serve, e.g:

- where it takes several attempts to get a meter reading;
 - customers in remote areas;
 - debt prone/late paying customers;
 - properties with frequent tenancy changes;
- Customers willing to pay for a smart meter, because they value the services it may offer may be a large group in the business sector but probably more limited amongst households.

Customer benefits

Customer benefits could come in three main ways :

- improvements in billing accuracy and payment arrangements;
- enabling energy saving (and hence money saving);

Improvements in billing accuracy and payment arrangements:

- more accurate billing ---> avoid risk of debt or credit;
- no meter reading ---> no need to let people into the house;
- more payment options ---> payment on actual use might suit some and could also encourage energy saving

Energy saving:

- More information provided via a conveniently sited display ---> (e.g. in kitchen) or via the internet or mobile phone messages could enable behavioural change and encourage investment in energy saving measures, that could lead to lower bills. This could involve absolute reductions in use (energy saving) or shifting use to off-peak times (via time of use pricing);
- Smart meters with two-way communications could be used to send messages ---> energy saving tips;

Benefits for distribution/transmission network owners/operators

- Potentially some benefits through better demand response (i.e. time of use tariffs) that leads to avoided peak capacity investment and thus more economic use of networks.
- Metering could facilitate separate import and export prices for customers with on-site generation.

Overall, both suppliers and the distribution/transmission businesses see the benefits here as relatively small, perhaps worth around £1-2 per meter per year on average.

The largest benefit that has been assessed in most international examples comes from reductions in peak demand.

In conclusion, if smart metering is left to the market, it is likely to happen only on a relatively small-scale basis over the short to medium term, because the costs exceed the benefits that would accrue directly to suppliers, or to network operators.

This means therefore that the benefits to customers and the environment become key to whether the cost benefit case is positive.

Summary of costs and benefits

Capital and operating costs of smart meters	£ (pounds)
Capital cost of each AMM-Interval Meter installed	£120 per meter (range £70 - £180)
Cost per meter per customer pa	£8 per customer pa
Annual maintenance / operational costs	£5-10 pa
Total nominal cost per customer	£13 – 18 pa
Benefits of smart electricity meters	
Potential supplier benefits at end of 100% roll-out	£125 million pa (£5 per customer)
Potential benefits to customers	
1 - 3% pa energy saving/load reduction	£3.50 -£10.50 per annum
Possible 1% load shifting due to time of use tariffs	£3.50 per annum
Accurate bills / scope for better budgeting	Estimate £1 per customer pa
Improved service	Estimate £1 per customer pa
Total potential customer benefits	£9-16.00 a year
Network Benefits (cost savings on upgrading)	£1-2 per annum per meter
Total benefits (excluding any value for carbon)	£15-23 per year

Table 1.4: Summary of costs and benefits [2].

1.6 New tariffs and new service

Currently, domestic tariffs for gas and electricity are flat, averaged across the day and year and do not directly reflect the costs of producing, transporting and supplying power and gas [9].

Meters which record energy-use by time-period, open the way for energy suppliers to introduce time- or volume-related tariffs, just as for phone charges.

If customers respond to these price signals by:

- changing the pattern of their energy-use;
- or reducing their energy demand

it may eventually allow some parts of the energy supply system to be operated or to be planned differently and create potential cost-savings. Some of these savings may fall to energy suppliers, who may then choose to share some of these savings with their customers.

By creating a price-related demand response, variable tariffs will impact throughout the energy market [1]:

- on energy suppliers;
- on networks;
- on generators;
- on the system operator;
- as well as on consumers.

Major considerations for consumers include:

Economic efficiency: There will be winners and losers in moving from flat tariffs to more cost reflective ones. For example, consumers with high peak-use are presently subsidised by consumers whose usage is mainly at lower-cost times.

Fairness: tariffs which increase with volume or at peak time, will benefit those consumers able to respond by reducing or shifting their load. While consumers with inflexible consumption may be disadvantaged.

Environment: If new tariffs reduce energy use, carbon emission benefits should follow [10]. Tariffs designed to reduce system peaks (to avoid or delay new investment in plant or networks) could have some energy demand-reduction effect too.

Security of supply: Peak-related tariffs can improve system security by flattening demand peaks.

Impact on bills: Impact on bills will depend on how the tariff is structured. Peak-related tariffs could reduce the average bill if a consumer used the same amount of electricity or gas but at cheaper off-peak times (or even used a little more) [11].

The objective of the new tariffs could be:

- peak-load-shifting;
- energy or carbon reduction;
- lower average bills;

or some combination.

In most overseas trials (Ontario, California, Australia and Norway) the main goal of new tariffs has been to reduce electrical peak-load, caused by summer air-conditioning or winter space and water-heating [].

Types of new tariffs:

NAME	DESCRIPTION
Block/Tiered tariffs	Payment for fixed-blocks of energy at pre-agreed prices. Price-blocks can increase, or decrease. First block can be higher-priced to cover fixed costs, or lower-priced to ensure a basic supply. Could be an effective tariff for reducing demand, but would not impact on daily or seasonal peak-use.
Time of Use tariffs	Differing unit-price for different pre-fixed time periods. Time-of-use periods relate to timing of peak system demands over the day, week or year and could change seasonally
Critical Peak Pricing tariffs	High per-unit price for usage during periods nominated by the supplier as 'critical peak periods'. A typical critical peak price ratio is 5:1 (or more) against the standard tariff. There are many variations of critical peak pricing, including rebates for customers who keep demand below a pre-agreed target.
Real-Time Pricing tariffs	Unlikely to be offered to small users Varies continuously through time rather than a pre-fixed tariff

Table 1.5 Types of new tariffs [1].

Consumers can take the benefit of new tariffs by:

- Using less energy overall ---> which will keep down fuel bills and reduce carbon;
- Using energy at different times ---> which could reduce bills but may have an uncertain carbon impact;
- Shifting to different fuels – could reduce bills – but have an unknown carbon effect.
For example: with a variable electricity tariff and a flat gas tariff, a consumer may switch some usage to gas for example for cooking or water heating.

Price and lower bills will play a major part in consumer acceptance of smart meters and new tariffs. To reduce energy use overall (for example not just to achieve peak-price response), a shared goal of carbon reduction would be needed, and is most likely to be achieved by smart meter tariffs in combination with new automated controls for gas-boilers and thermostats.

New services

Together with new tariffs, smart meters will enable a range of new services. Some likely developments are as follows:

NAME	DESCRIPTION
Electricity and gas sold in combination with other services	Such as broadband or cable packages. Perhaps for communications, entertainment and fuel, or with banking, insurance or home-security services. New combined packages, many of which may be e-account based, could be a growth marketing area
Smarter appliances and smarter homes	Together with smart meters, a variety of in-home wireless controls and sensors may be installed to create smarter homes. This includes individual appliance switching, load-control and thermostat-control. Consumers want better understanding of, and control over the consumption of individual appliances. In-home networks with micro-processor controls could enable this.
Automated demand-side control	For the future, time-of-use tariffs facilitated by smart meters could also be important in helping to manage any major increase in electrical load which may result from a radical increase in electric vehicle use.
Additional safety features	Every year there are a small number of electrical fatalities. Smart meters could facilitate additional safety features. For example, a switch with micro-processor controls could provide a very sensitive residual current device facility helping to detect earth leakage.
Metering for micro-generation	Import and export capability is likely to become a standard functionality of all electricity smart meters without material incremental cost [12], and could enable householders to benefit from the electricity they export. Micro-generation metering will also provide better information to the

Table 1.6 New services [1].

In time, there are likely to be many new services offered to consumers in tandem with new tariffs that could be of great interest.

Not all marketing developments will centre around energy efficiency or energy saving, but a post-2011 supplier obligation for either energy or carbon reduction would keep energy-saving firmly to the fore as a main supplier marketing goal.

Load flow for lower voltage system

2.1 Introduction

Generally, distribution feeder have a high R/X ratio and their configuration is radial. These reasons make that distribution systems are ill-conditioned and thus conventional method as Newton-Raphson, fast decoupled load flow and their modifications to are unsuitable for solving load flow for most cases and fail to converge [3].

Literature survey shows that several non-Newton efficient algorithms based on backward and forward sweeps are reported (Haque,1996) to (Ranjan and Das,2003).

Haque (1996) has developed a method for radical and mesh networks. In the mesh networks the loops are opened and in the loop break point a dummy bus is added [3].

The power flow through the branch that makes the loop is simulated by injection of the same power in the dummy bus. The method uses the backward and forward sweeps with initial voltage of all the nodes assumed to be equal to that of the source bus which is took as reference.

Ghosh and Das (1999) also uses backward and forward sweep with an initial voltage of all the nodes beyond the line branches. The method involve the evaluation of algebraic expression. It only permit the calculation of the nodes voltage rms values.

Ranjan et al. (2003) presents a method based on the load flow algorithm developed by Das which is modified to incorporate a composite load models. This methods also apply the backward and forward sweeps with an initial values of the nodes voltage assumed to be of 1 p.u [3].

Ranjan and Das (2003) method uses the basic principal of circuit theory but first, the authors have developed an algorithm for determining the nodes after each branches of the network.

However the method gives only the voltage magnitude of each node on the basis of algebraic equations. As convergence criterion of the algorithm, author have proposed the difference of the reactive and the active power at the sub-station end of two successive iterations. If this difference is less than 0.1kw and 0.1 kvar, the solution is reached.

In the present study, my main aim is the development of an efficient method for solving radial distribution feeder. A fast and easy to understand algorithm for determining nodes after branches based on the study presented by Anguillaro et al. (2001) is given. Other benefits of the method lies in the evaluation of the voltage phase-angles which becomes necessary if the load flow solution is used in the capacitors sizing problem. The test carried out on several feeders shows that my model takes few time to reach the solution. The results obtained by my method are compared to other method.

The results obtained from load flow calculation will come used in voltage estimation algorithm as measurements from smart meters.

2.1.2 Node and branch numbering

The numbering scheme is not required for the proposed load flow solution of radial distribution networks. However and for convenience, to perform the numbering scheme we consider the example line of Fig. 2.1

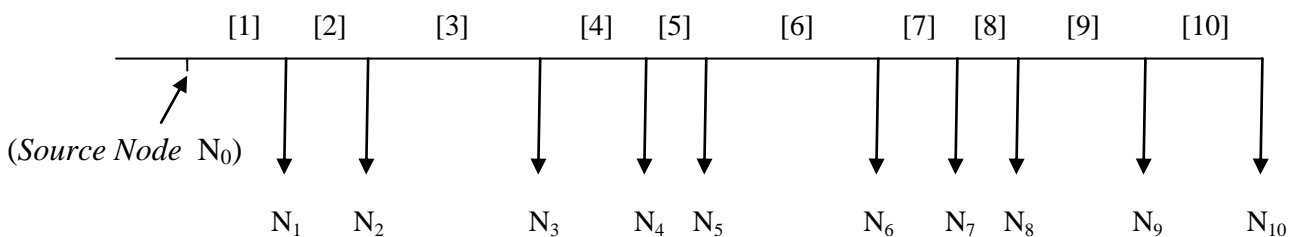


Fig. 2.1 : Nodes and branches numbering scheme

We number the nodes of the feeder. The source node is numbered as bus number 0. The node just ahead the source node is labeled node 1 and so on until the end-node. The feeder connectivity of Fig. 2.1 is presented in Table 2.1.

Table 2.1: Feeder connectivity

Branch	Sending-end [SE(i)]	Receiving-end [RE(i)]
1	0	1
2	1	2
3	2	3
4	3	4
5	4	5
6	5	6
7	6	7
8	7	8
9	8	9
10	9	10

2.2 Mathematical Formulation

Assumptions: It is assumed that the three-phase radial distribution network is balanced and thus can be represented by its one-line diagram. Distribution lines are of medium level voltage then the shunt capacitance are small and thus ignored. The single-line equivalent diagram of a such line is shown on Fig.2.2

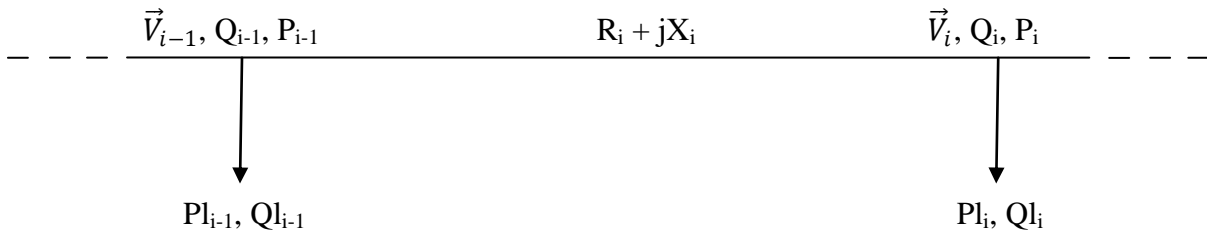


Fig 2.2: Branch One-line diagram

Mathematical models: The load flow of radial distribution network can be solved iteratively from two sets of recursive equations. The first set concern the determination of the branches current by going up the line (backward sweep). The second one allow us to determine the nodes voltage by going down the line (forward sweep) [3].

Branches power and current:

From the branch electric equivalent shown in Fig. 2.2 we can write the set of the above equations.

$$P_i = \sum_{k=i}^{if} P_{l_k} + \sum_{k=i+1}^{if} P_{loss_k} \quad (2.1)$$

$$Q_i = \sum_{k=i}^{if} Ql_k + \sum_{k=i+1}^{if} Qloss_k \quad (2.2)$$

where:

- P_i = Is the active power fed through bus i. It is equal to the sum of the active power of all the loads beyond node i (node i included) plus the sum of the active power loss in the branches beyond node i (branch i not included). (Active Power in the ith branch)
- Q_i = Is the reactive power fed through bus i. It is equal to the sum of the reactive power of all the loads beyond node i (node i included) plus the sum of the reactive power loss in the branches beyond node i (branch i not included). (Reactive Power in the ith branch)
- k is $\rightarrow i \leq k \leq if$ i = Number of node and branches
 if = Node and branch final of the network
- P_{l_k} = Is the active power of the load at node k.
- Q_{l_k} = Is the reactive power of the load at node k.
- P_{loss_k} = Is the active power loss in the kth branch.
- Q_{loss_k} = Is the reactive power loss in the kth branch.

The active and reactive power loss are given by:

$$P_{loss_k} = \frac{Pl_k^2 + Ql_k^2}{V_k^2} \cdot R_k \quad (2.3)$$

$$Q_{loss_k} = \frac{Pl_k^2 + Ql_k^2}{V_k^2} \cdot X_k \quad (2.4)$$

where:

- R_k = Is the resistance of the kth branch.
- X_k = Is the reactance of the kth branch.

The current that flowing through the ith branch is given by:

$$\vec{I}_i = \frac{P_i - jQ_i}{conj\vec{V}_i} \quad (2.5)$$

where:

- \vec{I}_i = Current in the ith branch
- $conj\vec{V}_i$ = Conjugate Voltage at the node i

If the complex voltage at the node i is:

$\vec{V}_i = V_i (\cos\varphi_i + j\sin\varphi_i)$, (3) can be expressed as:

$$\vec{I}_i = \frac{P_i \cos\varphi_i + Q_i \sin\varphi_i}{V_i} - j \frac{Q_i \cos\varphi_i - P_i \sin\varphi_i}{V_i} \quad (2.6)$$

The d and q components of the current (4) are:

$$I_{d_i} = \frac{P_i \cos\varphi_i + Q_i \sin\varphi_i}{V_i} \quad (2.7)$$

$$I_{q_i} = \frac{Q_i \cos\varphi_i - P_i \sin\varphi_i}{V_i} \quad (2.8)$$

Nodes Voltage:

For the nodes voltage and regarding our numbering scheme, we can write the following complex expression:

$$\vec{V}_{(RE(i))} = \vec{V}_{(SE(i))} - [R_{(RE(i))} + jX_{(RE(i))}] \cdot [I_{d_{(RE(i))}} + jI_{q_{(RE(i))}}] \quad (2.9)$$

the d and q components of which are:

$$V_{d_{(RE(i))}} = V_{d_{(SE(i))}} - [R_{(RE(i))} \cdot I_{d_{(RE(i))}}] - [X_{(RE(i))} \cdot I_{q_{(RE(i))}}] \quad (2.10)$$

$$V_{q_{(RE(i))}} = V_{q_{(SE(i))}} + [R_{(RE(i))} \cdot I_{q_{(RE(i))}}] - [X_{(RE(i))} \cdot I_{d_{(RE(i))}}] \quad (2.11)$$

where:

- $RE_{(i)}$ = Is the receiving-end of the branch i.
- $SE_{(i)}$ = Is the sending-end of the branch i.

For the first branch the d and q components of sending-end of the branch one are respectively equal to 1.0 in p.u. and zero. This correspond to the source node (node 0) which is also the reference node.

The voltage rms value and phase-angle of the receiving-end of the branch i are given by:

$$V_{(RE(i))} = \sqrt{V_{d_{(RE(i))}}^2 + V_{q_{(RE(i))}}^2} \quad (2.12)$$

$$\varphi_{(RE(i))} = \text{atan} \frac{V_{q_{(RE(i))}}}{V_{d_{(RE(i))}}} \quad (2.13)$$

2.3 Solution methodology

To determine the voltage at each node of radial distribution networks, the proposed method can be summarized in the following algorithm.

- STEP 1: Read the line data.
- STEP 2: Initialize the voltage of all the nodes to 1 p.u. and phase-angle to zero.
- STEP 3: Perform the backward sweep to obtain the current in each branch by using Equation (1) to (5).
- STEP 4: Perform the forward sweep to calculate the voltage rms value and phase-angle at each node by using Equation (7) and (8).
- STEP 5: If the voltage at each node for two successive iterations is within a certain tolerance (0.00001 p.u.) the solution is reached go to step 6 else, repeat step 4 to 6 until the convergence criterion is reached.
- STEP 6: Read the results.

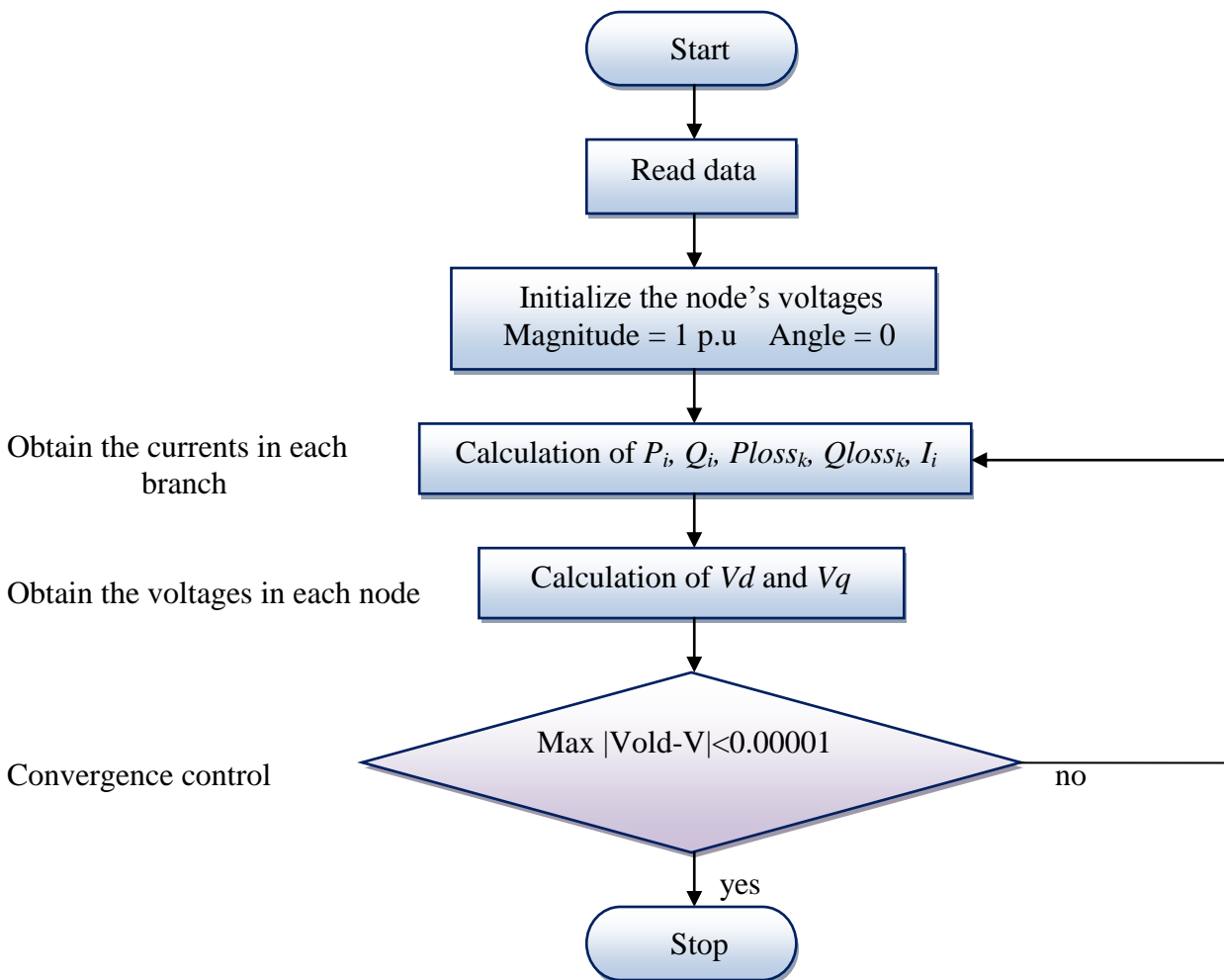


Fig. 2.3: Flowchart

To get the values of node's voltages and the currents flowing through the branches, it was carried a fully implemented algorithm in Matlab.

This program is structured so that it can be used for any radial network, regardless of the number of nodes. Structure is given in Appendix 1.

2.4 Applications

2.4.1 Network1

The first application that we see is a feeder network with only 6 nodes, where the voltage at Source nodes (balance node) is 20 kV.

Feeder network is shown in Fig.2.4

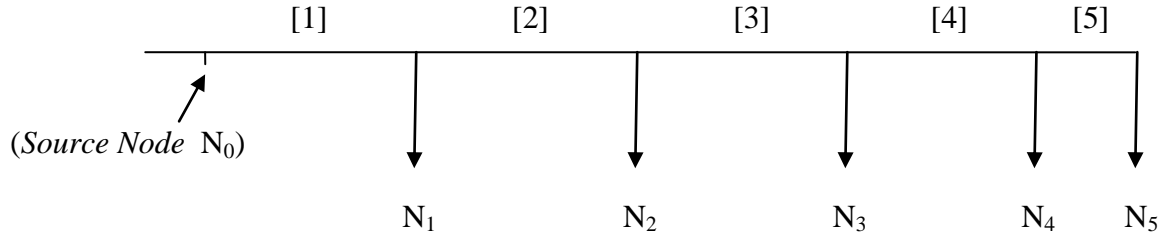


Fig. 2.4: Feeder Network 1

Data of feeder network of Fig. 2.4 are presented in Table 2.2 and in Table 2.3.

SE node	RE node	Voltage [V]	d [km]	r [Ω /km]	xl [Ω /km]	xc [Ω /km]
0	1	20e3 + 0j	9	0,118	0,35	0
1	2	20e3 + 0j	13	0,118	0,35	0
2	3	20e3 + 0j	8	0,118	0,35	0
3	4	20e3 + 0j	10	0,118	0,35	0
4	5	20e3 + 0j	5	0,118	0,35	0

Table 2.2: Branches data of the feeder network of Fig 2.4.

Node	Active Power [W]	Reactive Power [var]	Voltage [V]
0	0	0	20e3 + 0j
1	3,23e5	2,00e5	20e3 + 0j
2	3,20e5	2,40e5	20e3 + 0j
3	3,15e5	1,52e5	20e3 + 0j
4	4,00e5	3,00e5	20e3 + 0j
5	3,38e5	2,98e5	20e3 + 0j

Table 2.3: Loads data of the feeder network of Fig. 2.4.

From load flow calculation the following results in absolute values are obtained:

NODE'S VOLTAGES (absolute values)		
Node	Magnitude Voltage [V]	Angle Voltage [$^{\circ}$]
0	20000.000	0.00
1	19717.643	-0.59
2	19375.242	-1.30
3	19213.894	-1.65

4	19058.154	-1.94
5	19020.239	-2.01

BRANCH'S CURRENTS (absolute values)

Branch	Magnitude Current [A]	Angle Current [°]
1	60.642	-35.63
2	50.410	-37.07
3	38.841	-37.10
4	28.772	-40.95
5	13.678	-43.41

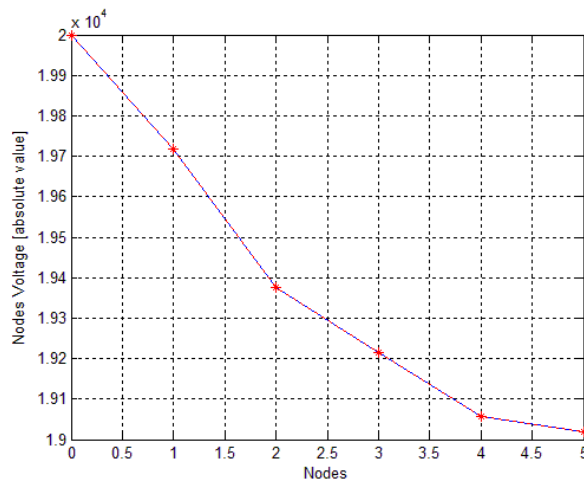
LOAD'S CURRENTS (absolute values)

Node	Magnitude Current [A]	Angle Current [°]
1	10.325	-28.59
2	11.569	-36.97
3	10.317	-26.30
4	15.118	-38.73
5	13.678	-43.41

LOAD'S POWERS (absolute values)

Node	Active Power [W]	Reactive Power [var]
1	311375.277	165519.891
2	315407.850	226379.212
3	312046.407	143239.343
4	399623.605	298883.571
5	337998.689	297996.117

We can see the node's voltages in Graph 2.1



Graph 2.1: Node's Voltages in absolute values (network 1).

As we can see from the results and the graph, the node's voltages decrements, this is correct because our network is passive and it is not active. If hypothetically we had had active loads as generators for example, the nodes voltage would increase in the nodes where there are generators.

We can see the same results in p.u. values rather than absolute values:

$$\begin{aligned} \text{where: } S_{base} &= S \text{ of the first node} = P + jQ \\ &= 323000 + 200000j \text{ [VA]} \\ &= \sqrt{P^2 + Q^2} = 380000 \text{ [VA]} \end{aligned}$$

$$V_{base} = 20000 + 0j \text{ [V]}$$

$$I_{base} = \frac{S_{base}}{V_{base} \cdot \sqrt{3}}$$

From load flow calculation the following results in p.u. values are obtained:

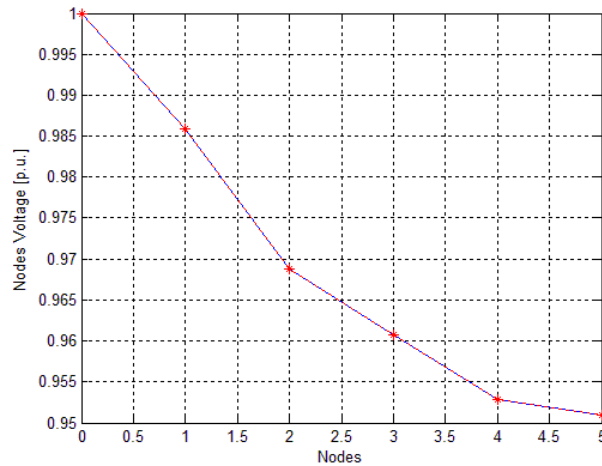
NODE'S VOLTAGES (pu)		
Node	Magnitude Voltage [pu]	Angle Voltage [°]
0	1.000	0.00
1	0.986	-0.59
2	0.969	-1.30
3	0.961	-1.65
4	0.953	-1.94
5	0.951	-2.01

BRANCH'S CURRENTS (pu)		
Branch	Magnitude Current [pu]	Angle Current [°]
1	5.957	-35.63
2	4.952	-37.07
3	3.816	-37.10
4	2.826	-40.95
5	1.344	-43.41

LOAD'S CURRENTS (pu)		
Node	Magnitude Current [pu]	Angle Current [°]
1	1.014	-28.59
2	1.136	-36.97
3	1.014	-26.30
4	1.485	-38.73
5	1.344	-43.41

LOAD'S POWERS (pu)		
Node	Active Power [pu]	Reactive Power [pu]
1	0.883	0.471
2	0.894	0.644
3	0.885	0.413
4	1.133	0.856
5	0.958	0.852

We can see the nodes voltage in a graph (Graph 2.2)



Graph. 2.2: Node's Voltages in p.u.

Both results, absolute values and p.u. have been calculated with a tolerance of 0.00001. The convergence criterion is reached after 5 iterations.

The convergence criterion is the following:

When the node's voltages are calculated, they come compared with old voltages, and if the difference between new and old voltages is smaller than tolerance the algorithm finish to work, while if the difference is greater than tolerance the algorithm does another iteration.

Example:

Node	Vold [V]	Vnew [V]	Vold-Vnew	abs(Vold-Vnew)
0	20000+0j	20000	0	0
1	20000+0j	19716,59-203,93j	283,41+203,93j	349,15
2	20000+0j	19377,80-440,36j	622,20+440,36j	762,27
3	20000+0j	19221,27-552,38j	778,73+552,38j	954,75
4	20000+0j	19071,54-646,24j	928,46+646,24j	1131,22
5	20000+0j	19035,41-667,03j	964,59+667,03j	1172,76

After the first iteration max abs(Vold-Vnew) = 1172,76

Node	Vold [V]	Vnew [V]	Vold-Vnew	abs(Vold-Vnew)
0	20000	20000	0	0
1	19717,59-203,93j	19717,59-203,93j	0	0
2	19377,80-440,36j	19370,25-440,18j	7,55-0,18j	7,55
3	19221,27-552,38j	19206,03-552,06j	15,24-0,32j	15,24
4	19071,54-646,24j	19047,43-645,17j	24,11-1,07j	24,13
5	19035,41-667,03j	19008,83-665,66j	26,58-1,37j	26,62

After the second iteration max abs(Vold-Vnew) = 26,62

Node	Vold [V]	Vnew [V]	Vold-Vnew	abs(Vold-Vnew)
0	20000	20000	0	0
1	19717,59-203,93j	19717,59-203,93j	0	0
2	19370,25-440,18j	19370,25-440,18j	0	0
3	19206,03-552,06j	19205,97-552,10j	0,06+0,04j	0,07
4	19047,43-645,17j	19047,24-645,29j	0,19+0,12j	0,12
5	19008,83-665,66j	19008,59-665,80j	0,24+0,14j	0,28

After the third iteration max abs(Vold-Vnew) = 0,28

Node	Vold [V]	Vnew [V]	Vold-Vnew	abs(Vold-Vnew)
0	20000	20000	20000	0
1	19717,59-203,93j	19717,59-203,93j	0	0
2	19370,25-440,18j	19370,25-440,18j	0	0
3	19205,97-552,10j	19205,97-552,10j	0	0
4	19047,24-645,29j	19047,24-645,29j	0	0
5	19008,59-665,80j	19008,59-665,80j	13-0,01j	13e-4

After the fourth iteration max abs(Vold-Vnew) = 13e-4

Node	Vold [V]	Vnew [V]	Vold-Vnew	abs(Vold-Vnew)
0	20000	20000	0	0
1	19717,59-203,93j	19717,59-203,93j	0	0
2	19370,25-440,18j	19370,25-440,18j	0	0
3	19205,97-552,10j	19205,97-552,10j	0	0
4	19047,24-645,29j	19047,24-645,29j	0	0
5	19008,59-665,80j	19008,59-665,80j	(15,18+9,03j)e-6	17,66e-6

After the fourth iteration max abs(Vold-Vnew) = 17,66e-6

As we can see after 5 iterations the convergence criterion is reached because the error is smaller than 10^{-4} .

2.4.2 Network 2

That seen for the network 1 we can see for the network 2.

The second application that we see is a feeder network with 11 nodes, where the voltage at Source nodes (balance node) is 400 V.

Feeder network is shown in Fig.2.5

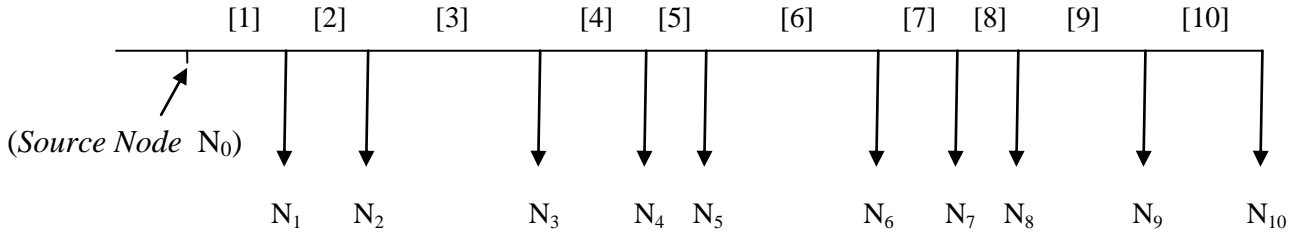


Fig. 2.5: Feeder Network 2

Data of feeder network of Fig. 2.5 are presented in Table 2.4 and in Table 2.5.

SE node	RE node	Voltage [V]	d [km]	r [Ω /km]	xl [Ω /km]	xc [Ω /km]
0	1	400+0j	0,2	0,328	0,0965	0
1	2	400+0j	0,13	0,328	0,0965	0
2	3	400+0j	0,4	0,328	0,0965	0
3	4	400+0j	0,2	0,328	0,0965	0
4	5	400+0j	0,1	0,328	0,0965	0
5	6	400+0j	0,4	0,328	0,0965	0
6	7	400+0j	0,15	0,328	0,0965	0
7	8	400+0j	0,12	0,328	0,0965	0
8	9	400+0j	0,3	0,328	0,0965	0
9	10	400+0j	0,2	0,328	0,0965	0

Table 2.4: Branches data of the feeder network of Fig 2.5.

Node	Active Power [W]	Reactive Power [var]	Voltage [V]
0	0	0	400+0j
1	3,00e3	2,00e3	400+0j
2	2,00e3	1,00e3	400+0j
3	3,00e3	1,00e3	400+0j
4	8,00e3	6,00e3	400+0j
5	3,00e3	2,00e3	400+0j
6	2,00e3	1,00e3	400+0j
7	1,00e3	1,00e3	400+0j
8	2,00e3	2,00e3	400+0j
9	1,00e3	1,00e3	400+0j
10	1,00e3	1,00e3	400+0j

Table 2.4: Loads data of the feeder network of Fig 2.5.

From load flow calculation the following results in absolute values are obtained:

NODE'S VOLTAGES (absolute values)

Node	Magnitude Voltage [V]	Angle Voltage [°]
0	400.000	0.00
1	395.836	0.19
2	393.503	0.29
3	386.979	0.60
4	384.256	0.76
5	383.198	0.82
6	380.171	1.03
7	379.332	1.09
8	378.794	1.14
9	378.120	1.19
10	377.895	1.21

BRANCH'S CURRENTS (absolute values)

Branch	Magnitude Current [A]	Angle Current [°]
1	36.805	-33.51
2	31.738	-33.40
3	28.953	-33.92
4	24.641	-36.81
5	19.294	-37.84
6	14.000	-39.58
7	10.762	-43.91
8	8.622	-43.86
9	4.319	-43.81
10	2.161	-43.79

LOAD'S CURRENTS (absolute values)

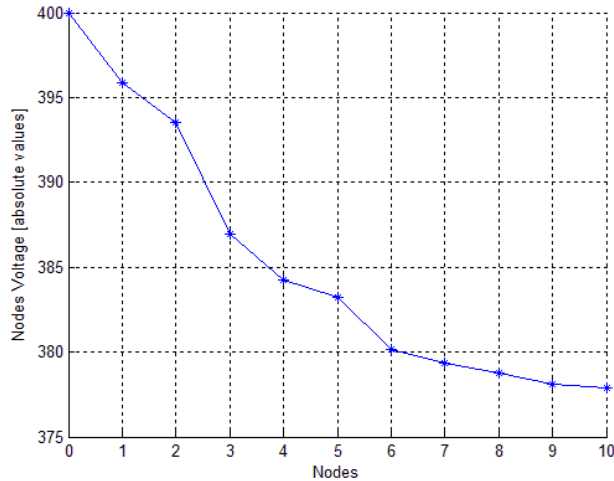
Node	Magnitude Current [A]	Angle Current [°]
1	5.067	-34.18
2	2.799	-28.05
3	4.518	-17.92
4	5.360	-33.12
5	5.318	-33.25
6	3.367	-25.63
7	2.140	-44.09
8	4.303	-43.92
9	2.158	-43.82
10	2.161	-43.79

LOAD'S POWERS (absolute values)

Node	Active Power [W]	Reactive Power [var]
1	2867.230	1960.941
2	1679.069	905.583
3	2871.656	962.242

4	2961.916	1988.801
5	2923.854	1977.604
6	1981.784	994.643
7	991.172	997.405
8	1994.489	1998.389
9	999.063	999.724
10	999.999	1000.000

We can see the node's voltages in graph 2.3:



Graph 2.3: Node's Voltages in absolute values (Network 2)

All that said for the first network is validated for the second network. We can see the same results in p.u. rather than absolute values:

where: $S_{base} = S$ of the first node $= P + jQ$
 $= 3000 + 2000j$ [VA]
 $= \sqrt{P^2 + Q^2} = 3605$ [VA]

$$V_{base} = 400 + 0j$$
 [V]

$$I_{base} = \frac{S_{base}}{V_{base} \cdot \sqrt{3}}$$

From load flow calculation the following results in p.u. values are obtained:

Node	NODE'S VOLTAGES (pu)	
	Magnitude Voltage [pu]	Angle Voltage [°]
0	1.000	0.00
1	0.990	0.19
2	0.984	0.29
3	0.967	0.60
4	0.961	0.76

5	0.958	0.82
6	0.950	1.03
7	0.948	1.09
8	0.947	1.14
9	0.945	1.19
10	0.945	1.21

BRANCH'S CURRENTS (pu)

Branch	Magnitude Current [pu]	Angle Current [°]
1	6.904	-33.51
2	5.953	-33.40
3	5.431	-33.92
4	4.622	-36.81
5	3.619	-37.84
6	2.626	-39.58
7	2.019	-43.91
8	1.617	-43.86
9	0.810	-43.81
10	0.405	-43.79

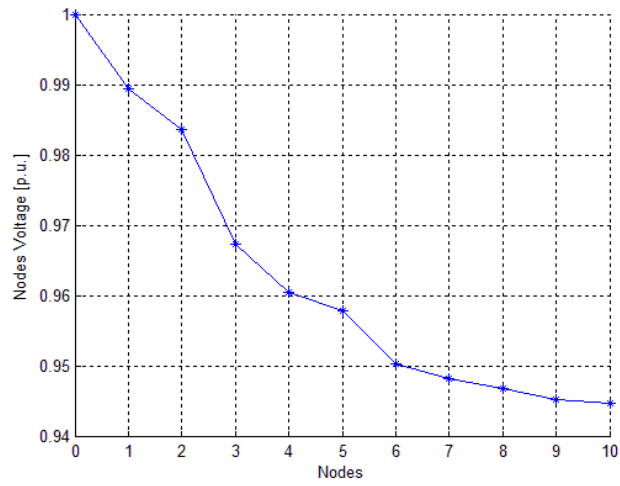
LOAD'S CURRENTS (pu)

Node	Magnitude Current [pu]	Angle Current [°]
1	5.067	-34.18
1	0.950	-34.18
2	0.525	-28.05
3	0.848	-17.92
4	1.005	-33.12
5	0.998	-33.25
6	0.632	-25.63
7	0.401	-44.09
8	0.807	-43.92
9	0.405	-43.82
10	0.405	-43.79

LOAD'S POWERS (pu)

Node	Active Power [pu]	Reactive Power [pu]
1	0.776	0.531
2	0.455	0.253
3	0.777	0.269
4	0.802	0.544
5	0.792	0.543
6	0.537	0.275
7	0.268	0.271
8	0.540	0.540
9	0.270	0.270
10	0.271	0.272

We can see the nodes voltage in a graph (Graph 2.4)



Graph. 2.4: Nodes Voltage in p.u.(network 2)

Both results, absolute values and p.u. have been calculated with a tolerance of 0.0001. The convergence criterion is reached after 4 iterations.

State Estimation for lower voltage system

Traditionally the voltage control is being done through On Load Tap Changers (OLTC). The voltage controller of OLTCs usually uses local measurements and few remote measurements obtained from remote terminal units (RTU) to set the voltage at the primary substation such that the entire distribution network will be within the required voltage limits. With the addition of many DGs, a better control of distribution voltage is required which demands more data from distribution network.

Though deploying RTUs to collect more data is expensive, the smart meters will avail a better opportunity to collect required data from the distribution network. However processing this data to estimate the state of the distribution network is challenging due to:

- (a) the smart meter measurements are not heterogeneous;
- (b) communication channels used for smart meters may introduce delays as long as several minutes;
- (c) there may be erroneous measurements.

The broad scope of this research is to use smart meters' measurements for controlling the voltages of distribution network. As a part of this research, my target was to find at the ways of using the delayed or erroneous smart meter measurements and nodes where measurements are not at all available to estimate the voltages of each distribution network node.

3.1 Nodes voltage estimation (Special case)

After using the algorithm for load flow calculation, I wrote a new algorithm to calculate the node's voltages at the nodes where power and voltage measurements are not available.

This way it has been possible to calculate the node voltages only if there are not adjacent nodes where measurements are not available.

3.1.1 Solution methodology

STEP1 : The algorithm calculates current flowing through the branch after node with unknown voltage as shown in fig 3.1:

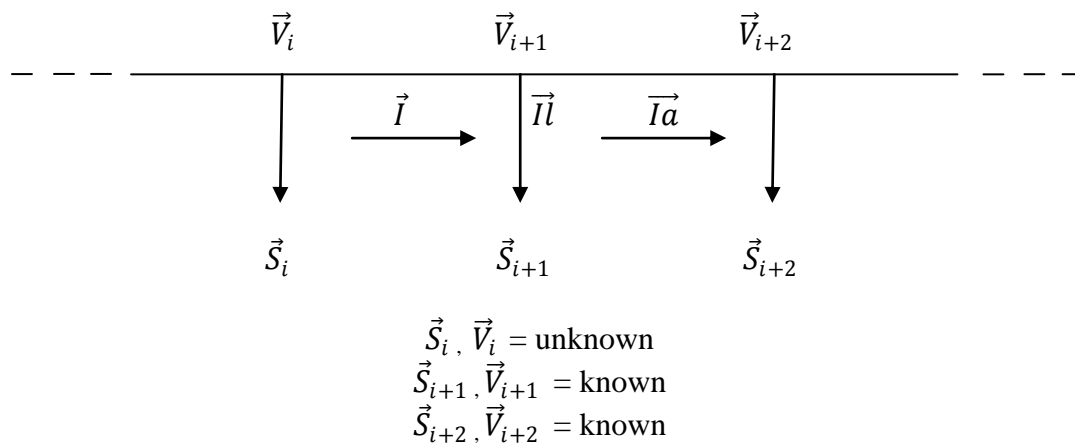


Fig. 3.1: Part of the network to show the step 1

Ohm's law can be applied because I know R and X of the branch after the node where I do not know the voltage and I know the load after.

If node voltage and load power are available, it is possible to calculate the load current:

$$\vec{I}_l = \frac{\text{conj}\vec{S}_{i+1}}{\sqrt{3} \cdot \text{conj}\vec{V}_{i+1}} \quad (3.1)$$

Then algorithm can calculate the current after the node where voltage and power are not available:

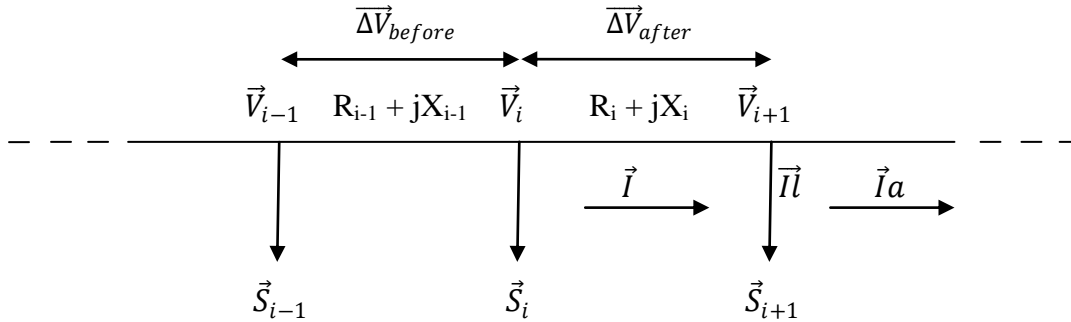
$$\vec{I}_a = \frac{\vec{V}_{i+1} - \vec{V}_{i+2}}{(R_{i+1} + jX_{i+1}) \cdot \sqrt{3}} \quad (3.2)$$

where R and X are the resistance and the reactance of the branch.

Finally the algorithm calculates the current flows in a branch i (I), after node with unknown voltage.

$$\vec{I} = \vec{I}_a + \vec{I}_l \quad (3.3)$$

STEP2 : the algorithm calculates voltage drop after node with unknown voltage as shown in fig. 3.2:



$$\begin{aligned} \vec{S}_{i-1}, \vec{V}_{i-1} &= \text{known} \\ \vec{S}_i, \vec{V}_i &= \text{unknown} \\ \vec{S}_{i+1}, \vec{V}_{i+1} &= \text{known} \end{aligned}$$

Fig 3.2: Part of the network to show the step 2

Where:

$\vec{\Delta V}_{before}$ = is the voltage drop calculated before the node where power measurement is not available.

$\vec{\Delta V}_{after}$ = is the voltage drop calculated after the node where power measurement is not available.

Ohm's law can be apply because I know R and X of the branch and voltage drop.

$$\overline{\Delta V}_{after} = \vec{I} \cdot (R_i + jX_i) \cdot \sqrt{3} \quad (3.4)$$

STEP3 : The algorithm calculates unknown voltage and unknown power as shown in fig 3.3:

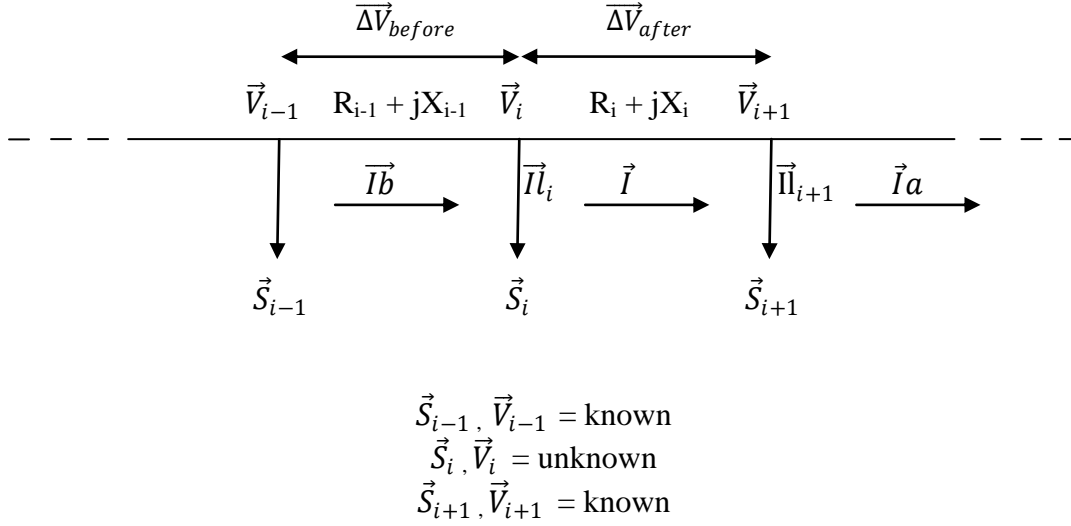


Fig: 3.3: Part of the network to show the step 3.

$$\vec{V}_i = \vec{V}_{i+1} + \overline{\Delta V}_{after} \quad (3.5)$$

After the algorithm calculated the unknown voltage, it can calculate the load current:

$$\vec{I}_i = \vec{I}_b + \vec{I} \quad (3.6)$$

where I_i is a load current to the node i and \vec{I}_b is the current flows before node i (node where voltage are not available)

$$\vec{I}_b = \frac{\overline{\Delta V}_{before}}{(R_{i-1} + jX_{i-1}) \cdot \sqrt{3}} \quad (3.7)$$

$$\vec{S}_i = \vec{V}_i \cdot \text{conj} \vec{I}_i \cdot \sqrt{3} \quad (3.8)$$

S_i = is the load power to the node i (unknown node).

In this mode the algorithm can calculate voltages and powers in all nodes.

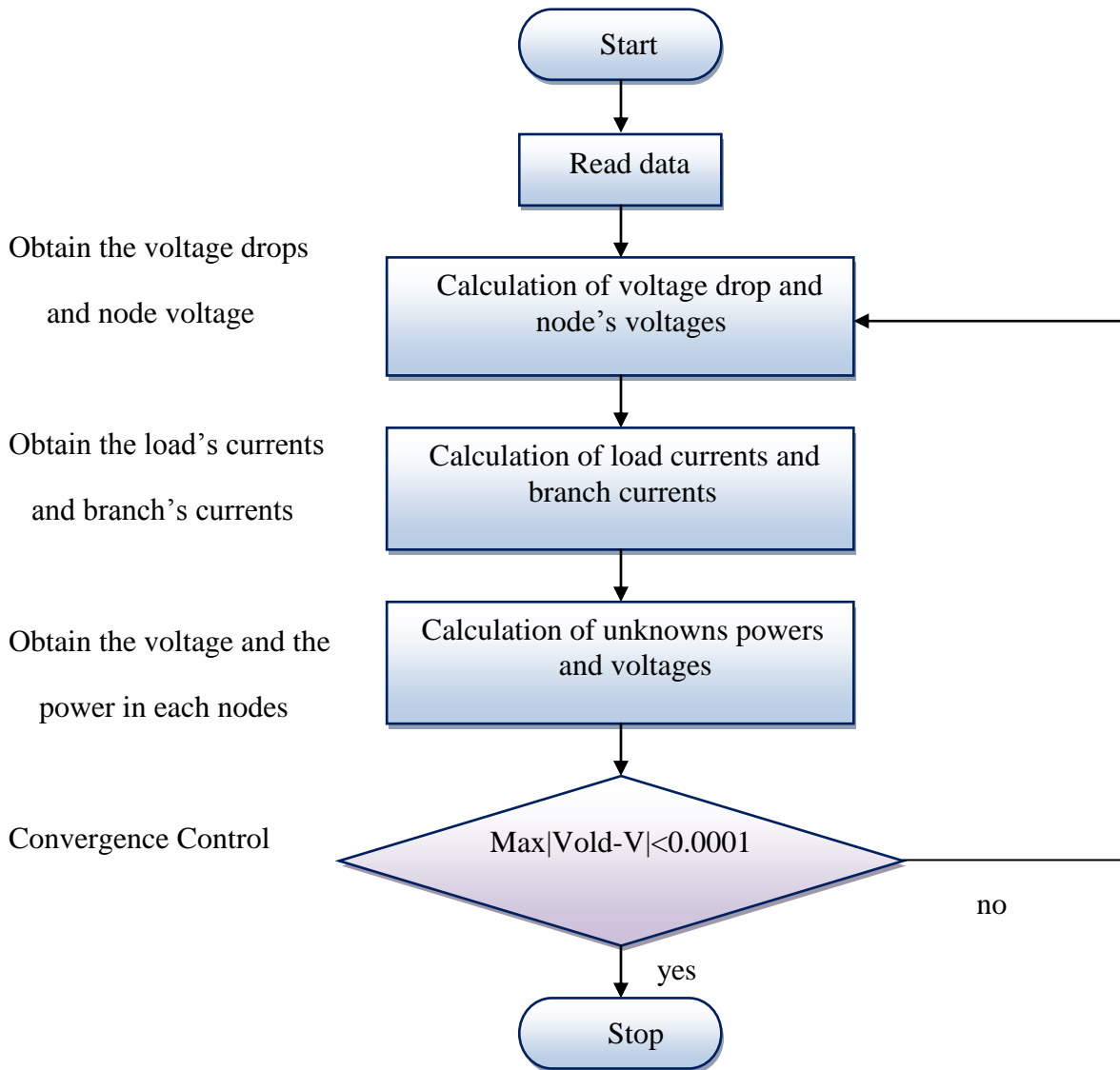


Fig 3.4: Flowchart

3.1.2 Application

Test Network that we see is a feeder network with 11 nodes, where the voltage at Source node (balance node) is 400 V. Feeder network is shown in Fig.3.5:

Voltage $N_0 = 400$ V

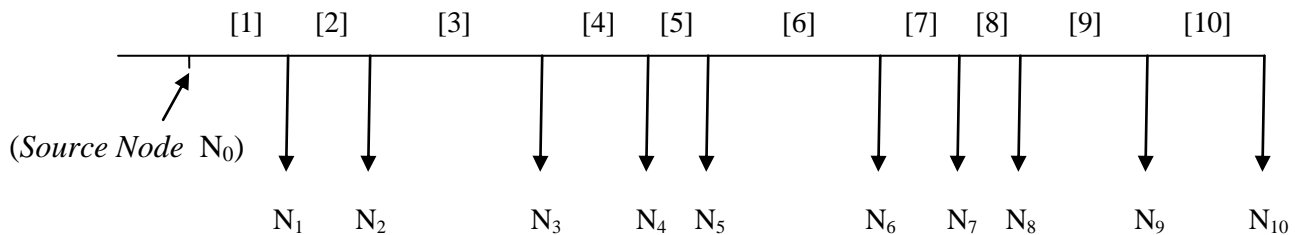


Fig 3.5: Test Network

In this network voltage and power measurements at nodes 2, 5 and 9 are unknowns. Branch data and load data are shown in Table 2.4 and 2.5 in Chapter 2.

From load flow calculation the following results are obtained:

NODE'S VOLTAGES (absolute values)

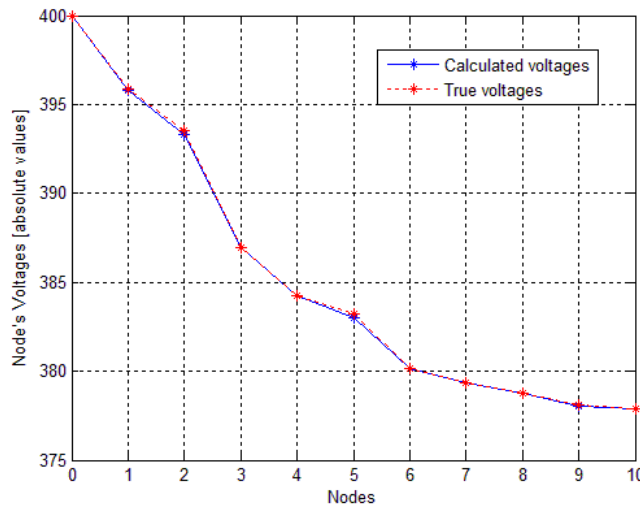
Node	Magnitude Voltage [V]	Angle Voltage [°]
0	400.000	0.00
1	395.832	0.19
2	393.515	0.30
3	386.981	0.60
4	384.254	0.76
5	383.202	0.86
6	380.171	1.02
7	379.329	1.10
8	378.795	1.14
9	378.119	1.23
10	377.895	1.21

If we compare these results (calculated voltages) with those calculated using load flow algorithm (true voltages) knowing all load powers, we can see that the difference between results is negligible. This is because the voltage values are calculated by equations and they are not estimated. Results calculated with load flow algorithm are shown below:

NODE'S VOLTAGES (absolute values)

Node	Magnitude Voltage [V]	Angle Voltage [°]
0	400.000	0.00
1	395.836	0.19
2	393.503	0.29
3	386.979	0.60
4	384.256	0.76
5	383.198	0.82
6	380.171	1.03
7	379.332	1.09
8	378.794	1.14
9	378.120	1.19
10	377.895	1.21

If we put in a graph both nodes voltage values, we can see that the difference between results is negligible:



Graph 3.1: Compare between true voltages and calculated voltages.

However if measurements in two adjacent nodes are not known, the algorithm does not work because there are too many unknowns.

This is a limitation because in a real network it is very likely to have missing adjacent measurements.

3.2 Nodes voltage estimation (Generalized case)

3.2.1 First program (Uses only the voltage measurements from S.M.)

After using the algorithm for the calculation of the node voltages assuming that the network has not adjacent nodes where power and voltage measurements are not available. I tried to use a new algorithm to calculate the unknowns node voltages.

This way it has been possible to calculate the node voltages even if we do not have adjacent voltage and power measurements.

Due to the additional unknown measurements, this case is more prone to errors than the previous case.

It is impossible to accurately estimate node voltages, if there are adjacent nodes where power and voltage measurements are not available.

3.3.1.1 Solution methodology

STEP 1 : The algorithm calculates the voltage at the first node where the voltage measurement is not available (from the end of network). This step is same to the previous algorithm.

STEP 1.1 : The algorithm calculates current flowing through the branch after node with unknown voltage as shown in fig 3.1.

STEP 1.2 : The algorithm calculates voltage drop after the node where voltage is not available as shown in fig. 3.7:

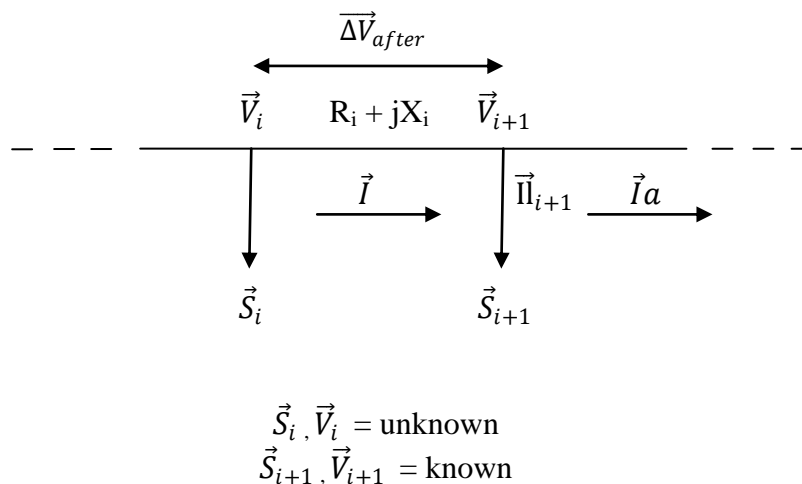


Fig 3.7: Part of the network to show the step 1.2

Where:

$\overline{\Delta V}_{after}$ = is the voltage drop calculated after the node where power measurement is not available. Ohm's law can be applied because I know R and X of the branch and current.

$$\overline{\Delta V}_{after} = \vec{I} \cdot (R_i + jX_i) \cdot \sqrt{3} \quad (3.9)$$

Finally the algorithm can calculate the voltage:

$$\vec{V}_i = \vec{V}_{i+1} + \overline{\Delta V}_{after} \quad (3.10)$$

This way it is possible to calculate the voltage and the new voltage array is obtained.

STEP 2 : The algorithm checks where are nodes with voltage measurements not available and checks how many adjacent nodes there are with measurements not available.

STEP 3 : When the algorithm knows the number of adjacent nodes in different sections of network where the measurements are not available, calculates unknown node voltages.

Explanation of the method used to estimate the voltage

Example with two adjacent nodes as shown in Fig 3.8:

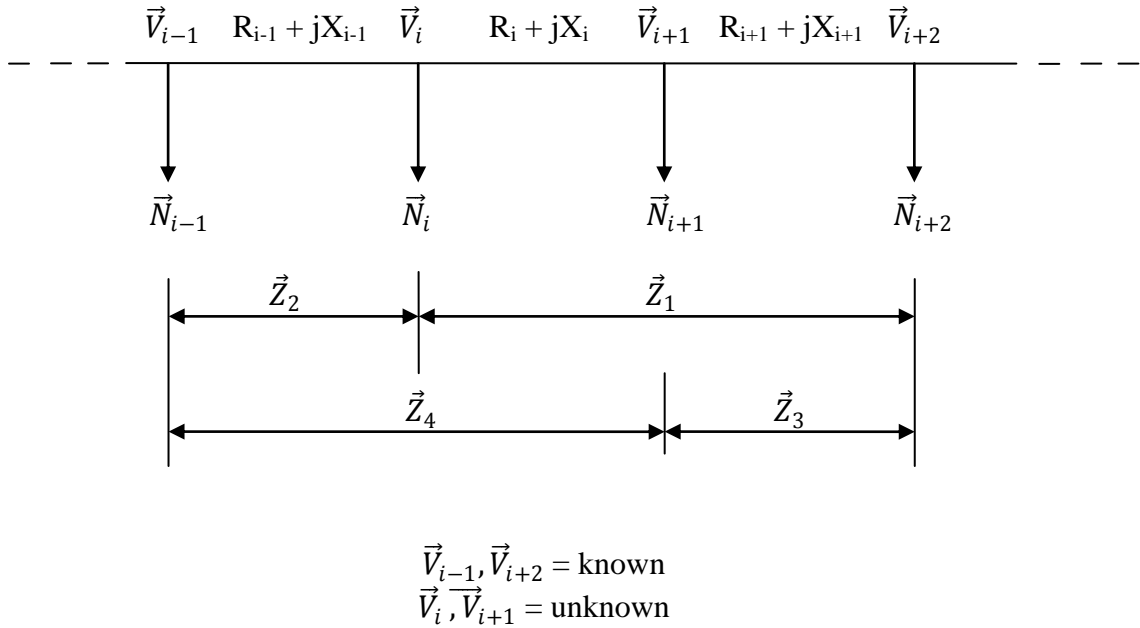


Fig 3.8: Part of the network to show the step 3.

Where N_i represents the number of node;

R_i and X_i represent resistance and reactance of branches;

Z_1, Z_2, Z_3, Z_4 are the impedances of branches.

If we know the node voltage before and after the unknown adjacent node voltages, we can estimate their voltages if we assume that the profile voltage is linear and decreasing.

The voltage at node “i” (where voltage is not available) is possible to calculate it, using closest nodes where voltages are available, as shown in fig 3.8.

The case of two adjacent nodes where voltage is not available, the voltage at “node i-1” and the voltage at node “i+2” are available. The two voltages will have a different weight to estimate the unknown voltage, the voltage that will have the greatest weight will be the closest.

To make this distinction we can calculate the impedance of the branches because we know R and X of the network.

The equation to calculate the voltage at node “i” is the following:

$$\vec{V}_i = \frac{\vec{V}_{i-1} \cdot \vec{Z}_1 + \vec{V}_{i+2} \cdot \vec{Z}_2}{\vec{Z}_t} \quad (3.11)$$

Where Z_1 is the impedance between node “i” and node “i+2”;

Z_2 is the impedance between node “i-1” and node “i”;

Z_t is the total impedance, sum of Z_1 and Z_2 .

While the equation to calculate the voltage at node “i+1” is the following:

$$\vec{V}_{i+1} = \frac{\vec{V}_{i-1} \cdot \vec{Z}_3 + \vec{V}_{i+2} \cdot \vec{Z}_4}{\vec{Z}_t} \quad (3.12)$$

Where Z_3 is the impedance between node “i+1” and node “i+2”;

Z_4 is the impedance between node “i-1” and node “i+1”;

Z_t is the total impedance, sum of Z_3 and Z_4 .

The same method we use to estimate nodes voltage when the network has three adjacent nodes where voltage is not available, as shown in Fig 3.9:

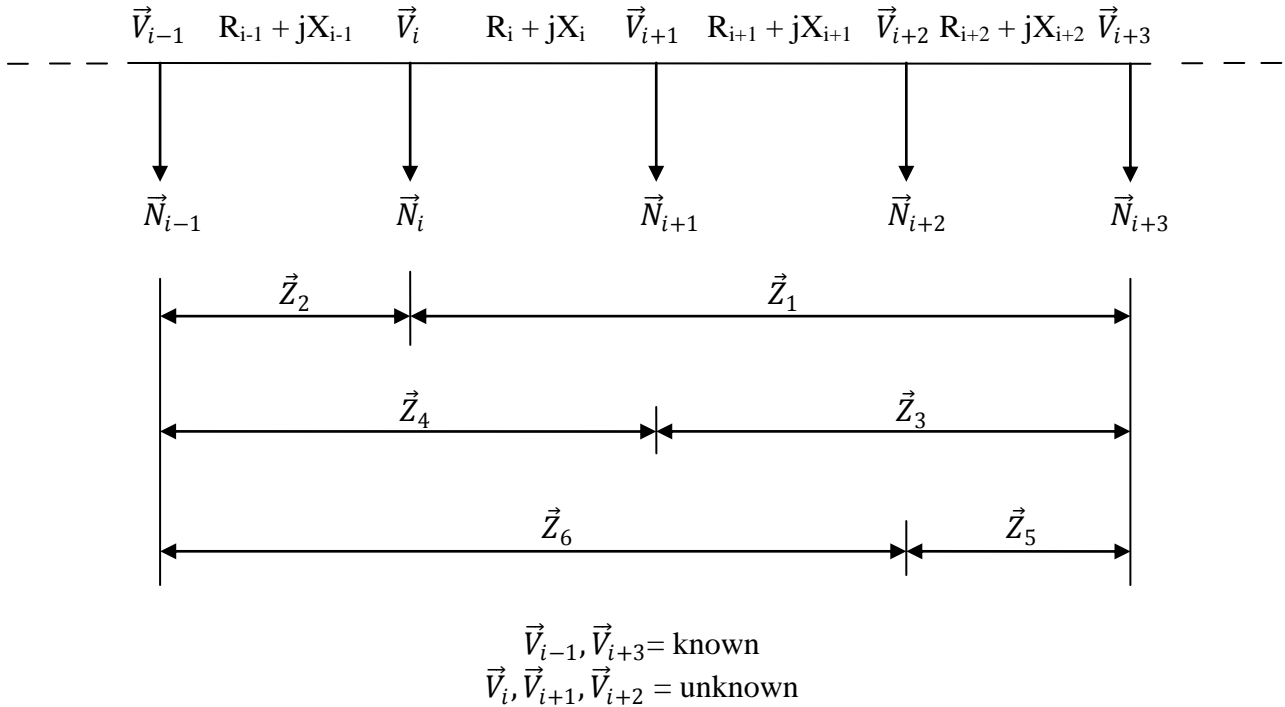


Fig 3.9: Part of the network to show the step 3

Equations to calculate the voltage at node where the voltage is not know as follows:

$$\vec{V}_i = \frac{\vec{V}_{i-1} \cdot \vec{Z}_1 + \vec{V}_{i+3} \cdot \vec{Z}_2}{\vec{Z}_t} \quad (3.13)$$

Where Z_1 is the impedance between node “i” and node “i+3”;
 Z_2 is the impedance between node “i-1” and node “i”;
 Z_t is the total impedance, sum of Z_1 and Z_2 .

$$\vec{V}_{i+1} = \frac{\vec{V}_{i-1} \cdot \vec{Z}_3 + \vec{V}_{i+2} \cdot \vec{Z}_4}{\vec{Z}_t} \quad (3.14)$$

Where Z_3 is the impedance between node “i+1” and node “i+3”;
 Z_4 is the impedance between node “i-1” and node “i+1”;
 Z_t is the total impedance, sum of Z_3 and Z_4 .

$$\vec{V}_{i+2} = \frac{\vec{V}_{i-1} \cdot \vec{Z}_5 + \vec{V}_{i+2} \cdot \vec{Z}_6}{\vec{Z}_t} \quad (3.15)$$

Where Z_5 is the impedance between node “i+2” and node “i+3”;
 Z_6 is the impedance between node “i-1” and node “i+2”;
 Z_t is the total impedance, sum of Z_5 and Z_6 .

The method is always the same regardless of the number of adjacent nodes where measurements are not available. In this way we can estimate the voltage in all nodes.

We can summarize all in a flow chart as shown in Fig 3.10:

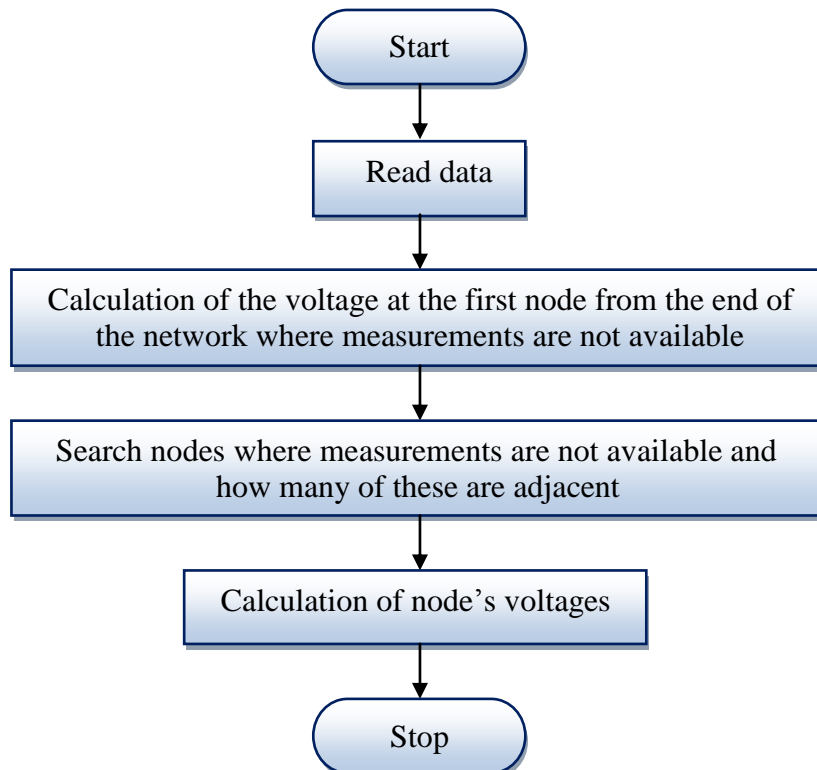


Fig 3.10: Flowchart

3.2.1.2 Application

Network test that we see is a feeder network with 11 nodes, where the voltage at Source node (balance node) is 400 V.

Feeder network is shown in Fig.3.11:

Voltage $N_0 = 400$ V

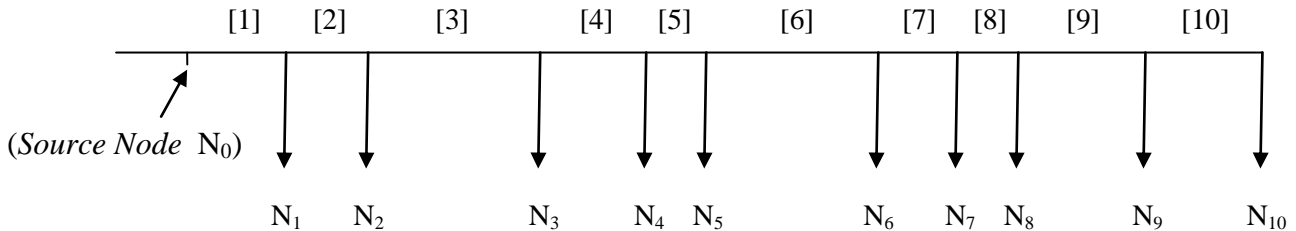


Fig. 3.11: Test Network

Simulation 1:

In this network load power and node voltage are not available in 3 nodes, two of which are adjacent. Voltage and power measurements at nodes 2, 5 and 9 are unknowns.

Branch data and load data are shown in Table 2.4 and 2.5 in Chapter 2.

For our simulations, we used load flow calculation data as if node voltages and load powers were available from smart meters.

From algorithm the following results are obtained:

NODE'S VOLTAGES (absolute values)

Node	Magnitude Voltage [V]	Angle Voltage [°]
0	400.000	0.00
1	395.832	0.19
2	393.495	0.29
3	386.981	0.60
4	385.033	0.72
5	384.060	0.78
6	380.171	1.02
7	379.329	1.10
8	378.795	1.14
9	378.027	1.25
10	377.895	1.21

If we compare these results (estimated voltages) with those calculated using load flow algorithm (true voltages) knowing all load powers, we can see that the difference between results is negligible.

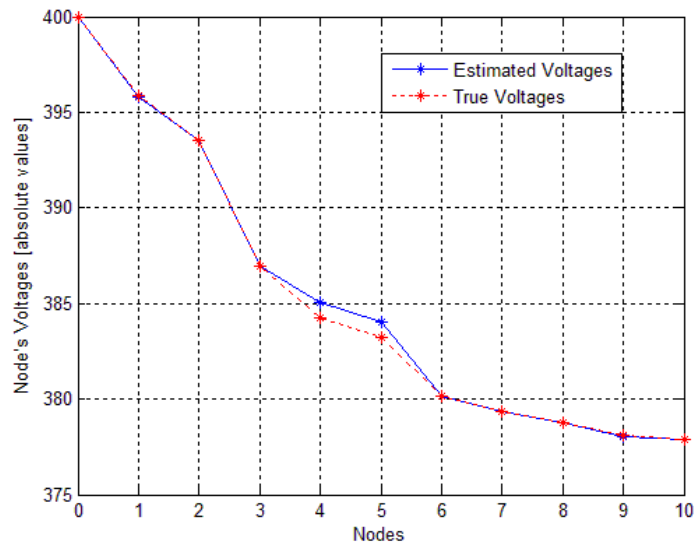
Results calculated with load flow algorithm are shown below:

NODE'S VOLTAGES (absolute values)

Node	Magnitude Voltage [V]	Angle Voltage [°]
0	400.000	0.00
1	395.836	0.19
2	393.503	0.29

3	386.979	0.60
4	384.256	0.76
5	383.198	0.82
6	380.171	1.03
7	379.332	1.09
8	378.794	1.14
9	378.120	1.19
10	377.895	1.21

If we put in a graph both nodes voltage values, we can see the difference between results:



Graph 3.2: Compare between true voltages and estimated voltages.

Nodes where measurements are not available, the error between true voltage and estimated voltage is shown in Table 3.1:

Node	Estimated Voltage	True Voltage	Error (absolute value)	Percentage error
4	385.033	384.256	0.777	0.194
5	384.060	383.198	0.862	0.215
9	378.027	378.120	0.093	0.023

Table 3.1: Error between true voltage and estimated voltage.

When load powers are 100%, the max error between true voltage and estimated voltage is about 0.24% of real voltage when there are two adjacent measurements not available.

Simulation 2:

In this network load power and node voltage are not available in 4 nodes, three of which are adjacent. Voltage and power measurements at nodes 3,4,5 and 9 are unknowns.

For our simulations, we used load flow calculation data as if node voltages and load powers were available from smart meters.

From algorithm the following results are obtained:

NODE'S VOLTAGES (absolute values)

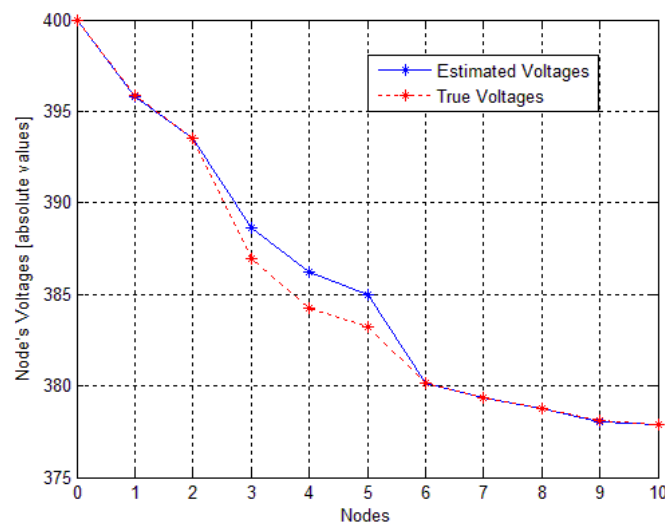
Node	Magnitude Voltage [V]	Angle Voltage [°]
0	400.000	0.00
1	395.832	0.19
2	393.495	0.29
3	388.643	0.55
4	386.219	0.68
5	385.009	0.75
6	380.171	1.02
7	379.329	1.10
8	378.795	1.14
9	378.027	1.25
10	377.895	1.21

If we compare these results (estimated voltages) with those calculated using load flow algorithm (true voltages) knowing all load powers, we can see that the difference between results is negligible. Results calculated from load flow calculation are shown below:

NODE'S VOLTAGES (absolute values)

Node	Magnitude Voltage [V]	Angle Voltage [°]
0	400.000	0.00
1	395.836	0.19
2	393.503	0.29
3	386.979	0.60
4	384.256	0.76
5	383.198	0.82
6	380.171	1.03
7	379.332	1.09
8	378.794	1.14
9	378.120	1.19
10	377.895	1.21

If we put in a graph both nodes voltage values, we can see the difference between results:



Graph 3.3: Compare between true voltages and estimated voltages.

Nodes where measurements are not available, the error between true voltage and estimated voltage is shown in Table 3.2:

Node	Estimated Voltage	True Voltage	Error (absolute value)	Percentage error
3	388.643	386.979	1.664	0.416
4	386.219	384.256	1.963	0.491
5	385.009	383.198	1.811	0.453
9	378.027	378.120	0.093	0.023

Table 3.2: Error between true voltage and estimated voltage.

When load powers are 100%, the max error between true voltage and estimated voltage is about 0.5% of true voltage when there are three adjacent measurements not available.

3.2.2 Second program (Uses all measurements from S.M.)

With the first program we can estimate only the node voltages. Now, I have tried to make a new program to calculate the voltage and power at the nodes where measurements are not available.

This has made it possible to calculate the nodes voltage, even if we do not know the voltage measurements of the adjacent nodes.

The error will be reduced compared to the previous case because we will use the measurements from smart meters to estimate the node voltages.

Mistakes are inevitable if we want to estimate the nodes voltage, when we have adjacent nodes where measurements are not available.

In all cases the error voltage will be minimized, whereas the error power will be big because I do not know how the loads of power are divided between the nodes where measurements are not available. When I have two or three adjacent nodes where I do not have measurements, it is very difficult to estimate the power that the loads absorb.

The method implemented to estimate node voltages and powers is explained below.

3.2.2.1 Solution methodology

STEP 1 : The algorithm checks where are nodes with voltage measurements not available and how many adjacent nodes there are.

STEP 2 : The algorithm calculates the voltage at the first node where the voltage measurement is not available (from the end of network). This step is same to the previous program.

STEP 3 : The algorithm calculates the node currents where voltage and power measurements are available. This has made it possible to assume the current value where it is not available, equal the current calculated at the node where the measurement is available.

Example with two adjacent nodes as shown in Fig 3.12:

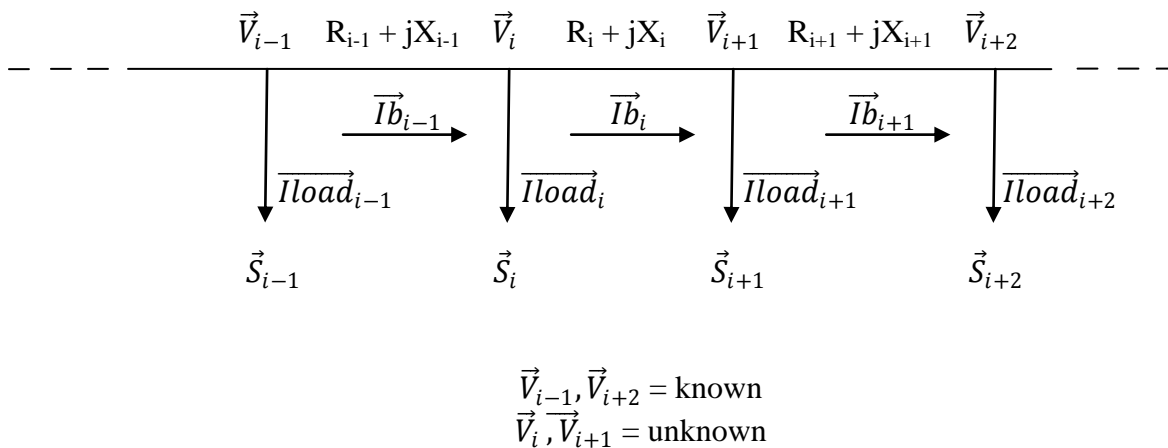


Fig 3.12: Part of the network to show the step 3.

Where we do not have a current value because measurements are not available, the initial current value assumed can be the following:

- 1) We can assume the current at node (i-1) because measurements are available;
- 2) Alternatively we can assumed half current value at node (i-1);

- 3) Another method is to assume half current at node (i-1) and half current at node (i+2), then summing the two currents and dividing by two.

Each method is correct to start the estimation of the nodes voltage. This current value is only a initial value.

To calculate the current at node (i-1) and at node (i+2) we can do as follows:

$$\overrightarrow{Iload}_{i-1} = \frac{conj\vec{S}_{i-1}}{conj\vec{V}_{i-1} \cdot \sqrt{3}} \quad (3.16)$$

$$\overrightarrow{Iload}_{i+2} = \frac{conj\vec{S}_{i+2}}{conj\vec{V}_{i+2} \cdot \sqrt{3}} \quad (3.17)$$

It is possible to calculate just these currents because other nodes do not have measurements.

Next step is to assume the node current values.

Before, we have listed three different method to assume the value current where measurements are not available.

- 1) $Iload_i = Iload_{(i+1)} = Iload_{(i-1)}$
- 2) $Iload_i = Iload_{(i+1)} = Iload_{(i-1)}/2$
- 3) $Iload_i = Iload_{(i+1)} = (Iload_{(i-1)}/2 + Iload_{(i-1)}/2)/2$

These are three possible methods to assume the initial current value where measurements are not available.

STEP 4 : The algorithm calculates branch currents because all load currents are available. The algorithm uses “backward sweep” method, in this way it calculates branch currents starting from last node.

With reference to fig. 3.13, the algorithm calculates branch currents with this equation:

$$\overrightarrow{Ib}_i = \overrightarrow{Iload}_{i+1} + \overrightarrow{Ib}_{i+1} \quad (3.18)$$

This equation is valid for all nodes because all node currents are available, some assumed and others known from load flow calculation (from smart meters).

STEP 5 : The algorithm calculates nodes voltages because all currents are available (branches currents and load currents). The algorithm can use backward sweeps or forward sweeps because it is considered only a part of the network.

If the network reference to fig 3.13 is considered, to calculate node voltages with backward sweep or forward sweep is the same because extreme node voltages are available.

For example in our case with reference to fig. 3.13, voltage V(i-1) and voltage V(i+2) are available.

The algorithm to calculate the node voltage (i-1) and node voltage (i+2) uses the following equation:

$$\vec{V}_i = \vec{V}_{i+1} + (R_i + jX_i) \cdot \overrightarrow{Ib}_i \quad (3.19)$$

The algorithm uses this equation if we want to use backward sweep, alternatively it uses the following equation:

$$\vec{V}_i = \vec{V}_{i-1} - (R_{i-1} + jX_{i-1}) \cdot \overrightarrow{Ib}_{i-1} \quad (3.20)$$

With this equation it calculates node voltages by forward sweep method. To use the first or the second equation is the same.

STEP 6: If the voltage calculated at node where measurements are available, for example node (i-1) or node (i+2), (depending on whether the algorithm uses backward sweep or forward sweep), is different from less than 0.06 Volt (about 0.00015 in p.u.) from true voltage calculated from load flow calculation, estimated voltage is accepted and the solution is reached, otherwise the algorithm have to repeat step 3 to 6 until the convergence criterion is reached. The convergence criterion has just been described.

When the algorithm repeats step 3, the load current assumed where measurement are not available, it is equal to that calculated at the previous iteration increased or decreased by 2%. Determine whether it should be increased or decreased is simple. Looking at the figure 3.13 is easy to understand it.

After the first iteration there are two voltage's values at node (i-1), one is estimated from the algorithm and one is true calculated from load flow calculation. If we compare the results there may be two cases:

- 1) $V_{true(i-1)} - V_{estimated(i-1)} < 0.06 \text{ V}$;
- 2) $V_{true(i-1)} - V_{estimated(i-1)} > 0.06 \text{ V}$;

If we have the first case the algorithm finishes to work, while if we have the second case, there are other two possibilities:

- 2.1) $V_{true(i-1)} < V_{estimated(i-1)}$;
- 2.2) $V_{true(i-1)} > V_{estimated(i-1)}$;

If we have 2.1) ie, the estimated voltage is bigger than real voltage (calculated from load flow calculation), the load current assumed at previous iteration must be decreased, because in this way both the voltage drop and the estimated voltage decreased.

After this iteration the difference between V_{true} and $V_{estimated}$ will be diminished. The steps in terms of equations are following:

$$\overline{Iload}_{i+1}^{(II)} = \overline{Iload}_{i+1}^{(I)} \cdot 0.98 \quad (3.21)$$

$$\overline{Ib}_i^{(II)} = \overline{Iload}_{i+1}^{(II)} + \overline{Ib}_{i+1}^{(I)} \quad (3.22)$$

$$\overline{Iload}_i^{(II)} = \overline{Iload}_i^{(I)} \cdot 0.98 \quad (3.23)$$

$$\overline{Ib}_{i-1}^{(II)} = \overline{Iload}_i^{(II)} + \overline{Ib}_i^{(I)} \quad (3.24)$$

Where subscript indicates the iteration's number.

This way the algorithm calculates new current values. Then the algorithm can calculate the new voltage's values as described in step 5.

Finally we compare the new result with true voltage value calculated from load flow calculation and we verify if the convergence criterion is reached.

Exactly the opposite will happen in the other case (2.2).

If we have 2.2) ie, the estimated voltage is smaller than true voltage (calculated from load flow calculation), the load current value assumed at previous iteration must be increased, because in this way both the voltage drop and the estimated voltage increased.

After this iteration the difference between V_{true} and $V_{estimated}$ will be diminished. The steps in terms of equations are following:

$$\overrightarrow{Iload}_{i+1}^{(II)} = \overrightarrow{Iload}_{i+1}^{(I)} \cdot 1.02 \quad (3.25)$$

$$\overrightarrow{Ib}_i^{(II)} = \overrightarrow{Iload}_{i+1}^{(II)} + \overrightarrow{Ib}_{i+1}^{(I)} \quad (3.26)$$

$$\overrightarrow{Iload}_i^{(II)} = \overrightarrow{Iload}_i^{(I)} \cdot 1.02 \quad (3.27)$$

$$\overrightarrow{Ib}_{i-1}^{(II)} = \overrightarrow{Iload}_i^{(II)} + \overrightarrow{Ib}_i^{(II)} \quad (3.28)$$

Where subscript indicates the iteration's number.

This way the algorithm calculates new current values. Then the algorithm can calculate the new voltage's values as described in step 5. Finally we compare the new result with true voltage value calculated from load flow calculation and we verify if the convergence criterion is reached.

This algorithm can estimate the voltage and the power in all nodes.

We can all summarize in a flowchart as shown in Fig 3.13:

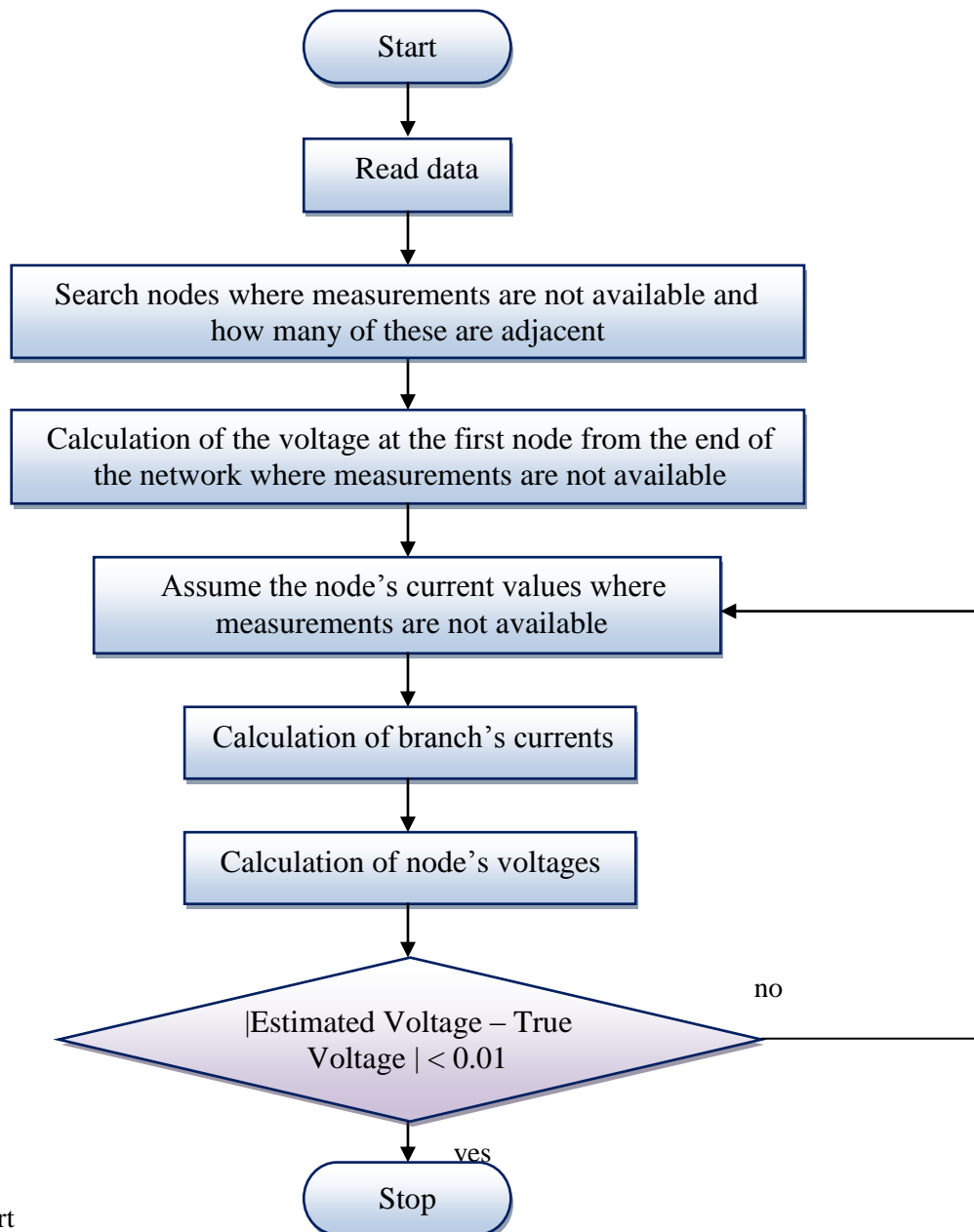


Fig. 3.13: Flowchart

3.2.2.2 Application

Test Network that we see is a feeder network with 11 nodes, where the voltage at Source node (balance node) is 400 V. Feeder network is shown in Fig.3.14:

Voltage $N_0 = 400$ V

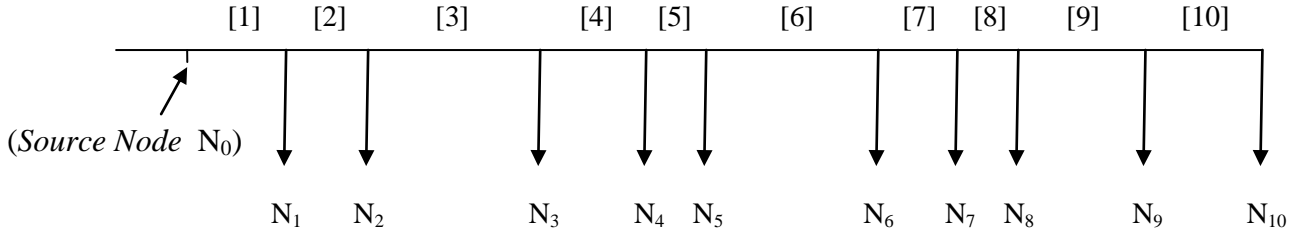


Fig. 3.14: Test Network

Simulation 1:

In this network load power and node voltage are not available in 3 nodes. Voltage and power measurements at nodes 2, 5 and 9 are unknowns.

Branch data and load data are shown in Table 2.4 and 2.5 in Chapter 2.

For our simulations, we used load flow calculation data as if node's voltages and load's powers were available from smart meters.

From algorithm the following results are obtained:

NODE'S VOLTAGES (absolute values)

Node	Magnitude Voltage [V]	Angle Voltage [°]
0	400.000	0.00
1	395.832	0.19
2	393.515	0.29
3	386.981	0.60
4	384.254	0.76
5	383.203	0.82
6	380.171	1.02
7	379.329	1.10
8	378.795	1.14
9	378.119	1.19
10	377.895	1.21

If we compare these results with those calculated using load flow algorithm, knowing all load powers, we can see that the difference between results is negligible. This is because the algorithm converges when the difference between estimated voltage and true voltage is very little.

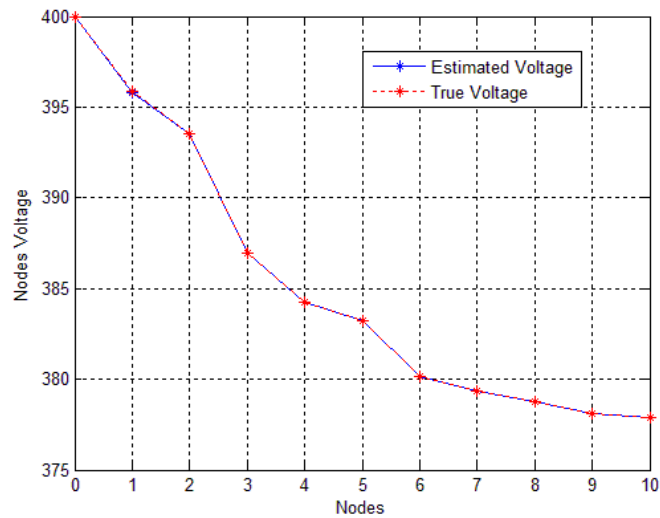
Results calculated with load flow algorithm are shown below:

NODE'S VOLTAGES (absolute values)

Node	Magnitude Voltage [V]	Angle Voltage [°]
0	400.000	0.00
1	395.836	0.19

2	393.503	0.29
3	386.979	0.60
4	384.256	0.76
5	383.198	0.82
6	380.171	1.03
7	379.332	1.09
8	378.794	1.14
9	378.120	1.19
10	377.895	1.21

If we put in a graph both node voltage values, we can see the difference between results:



Graph 3.4: Compare between Estimated voltages and True voltages.

Nodes where measurements are not available, the difference between estimated voltage and true voltage is negligible, this is possible because the algorithm makes a lot of iterations to reached the convergence. The difference is shown in table 3.3:

Node	Estimated Voltage	True Voltage	Error (absolute value)	Percentage error
2	393.515	393.503	0.012	0.003
5	383.203	383.198	0.005	0.001
9	378.119	378.120	0.001	0.0003

Table 3.3: Difference between estimated voltage and true voltage.

The same we can see for the power. From the algorithm, the following results in absolute values are obtained:

Node	NODE'S POWERS (absolute values)	
	Active Power [W]	Reactive Power [var]
1	2867.200	1960.90
2	1501.377	758.76
3	2871.700	962.20
4	2961.900	1988.80
5	2926.462	1761.82
6	1981.800	994.60

7	991.200	997.40
8	1994.500	1998.40
9	986.288	988.63
10	1000.000	1000.00

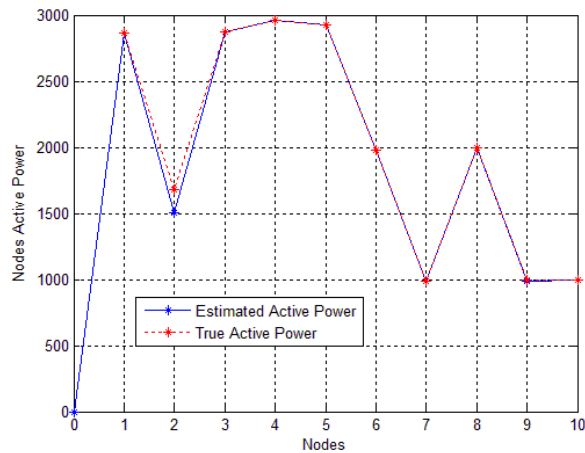
If we compare these results with those calculated using load flow algorithm, knowing all load powers, we can see that the difference between results is almost negligible.

This is because: - the algorithm converges when the difference between estimated voltage and true voltage is very little;
- there are no adjacent nodes where measurement are not available.

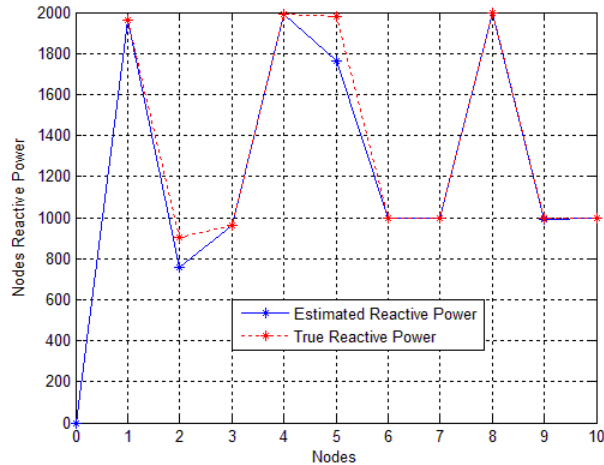
Results calculated with load flow algorithm are shown below:

Node	NODE'S POWERS (absolute values)	
	Active Power [W]	Reactive Power [var]
1	2867.230	1960.94
2	1679.069	905.58
3	2871.656	962.24
4	2961.916	1988.80
5	2923.854	1977.60
6	1981.784	994.64
7	991.172	997.40
8	1994.489	1998.38
9	999.063	999.72
10	999.999	1000.00

If we put in a graph both nodes voltage values, we can see the difference between results:



Graph 3.5: Compare between Estimated Active Power and True Active Power.



Graph 3.6: Compare between Estimated Reactive Power and True Reactive Power.

Nodes where measurements are not available, the difference between estimated power and true power is shown in table 3.4 and table 3.5:

Node	Estimated Active Power [W]	True Active Power [W]	Error (absolute value) [W]
2	1501.377	1679.069	177.692
5	2926.462	2923.854	2.608
9	986.288	999.063	12.775

Table 3.4: Difference between Estimated Active Power and True Active Power.

Node	Estimated Reactive Power [var]	True Reactive Power [var]	Error (absolute value) [var]
2	758.763	905.579	146.816
5	1761.829	1977.602	215.773
9	988.632	999.726	11.094

Table 3.5: Difference between Estimated Reactive Power and True Reactive Power.

The errors between true power and estimated power are bigger than the previous case which the comparison was between true voltage and estimated voltage.

This is because it is difficult or impossible to estimate accurate power values if there are measurements not available.

Simulation 2:

In this network load power and node voltage are not available in 4 nodes. Voltage and power measurements at nodes 2,5,8 and 9 are not available.

Branch data and load data are shown in Table 2.4 and 2.5 in Chapter 2.

For our simulations, we used load flow calculation data as if node voltages and load powers were available from smart meters.

From algorithm the following results are obtained:

NODE'S VOLTAGES (absolute values)

Node	Magnitude Voltage [V]	Angle Voltage [°]
0	400.000	0.00
1	395.832	0.19
2	393.515	0.29
3	386.981	0.60
4	384.254	0.76
5	383.203	0.82
6	380.171	1.02
7	379.329	1.10
8	378.865	1.13
9	378.119	1.19
10	377.895	1.21

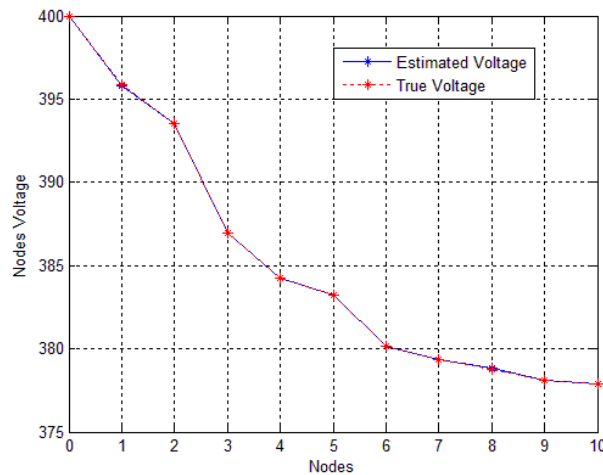
If we compare these results with those calculated using load flow algorithm, knowing all load powers, we can see that the difference between results is negligible. This is possible because the algorithm makes a lot of iterations to reached the convergence.

Results calculated with load flow algorithm are shown below:

NODE'S VOLTAGES (absolute values)

Node	Magnitude Voltage [V]	Angle Voltage [°]
0	400.000	0.00
1	395.836	0.19
2	393.503	0.29
3	386.979	0.60
4	384.256	0.76
5	383.198	0.82
6	380.171	1.03
7	379.332	1.09
8	378.794	1.14
9	378.120	1.19
10	377.895	1.21

If we put in a graph both nodes voltage values, we can see the difference between results:



Graph 3.7: Compare between Estimated voltage and True voltage.

Nodes where measurements are not available, the difference between estimated voltage and true voltage is negligible.

The difference is shown in table 3.6:

Node	Estimated Voltage	True Voltage	Error (absolute value)	Percentage error
2	393.515	393.503	0.012	0.003
5	383.203	383.198	0.005	0.001
8	378.865	378.794	0.071	0.018
9	378.119	378.120	0.001	0.0003

Table 3.6: Difference between estimated voltage and true voltage.

The same we can see for the power. From the algorithm, the following results in absolute values are obtained:

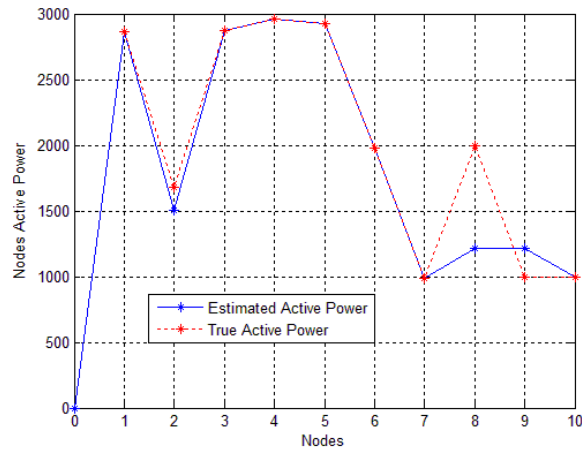
Node	NODE'S POWERS (absolute values)	
	Active Power [W]	Reactive Power [var]
1	2867.200	1960.90
2	1501.377	758.76
3	2871.700	962.20
4	2961.900	1988.80
5	2926.462	1761.82
6	1981.800	994.60
7	991.200	997.40
8	1214.844	1217.70
9	1211.146	1216.61
10	1000.000	1000.00

If we compare these results with those calculated using load flow algorithm, knowing all loads powers, we can see that the difference between results is bigger than the previous simulation, this is because there are adjacent nodes where measurements are not available.

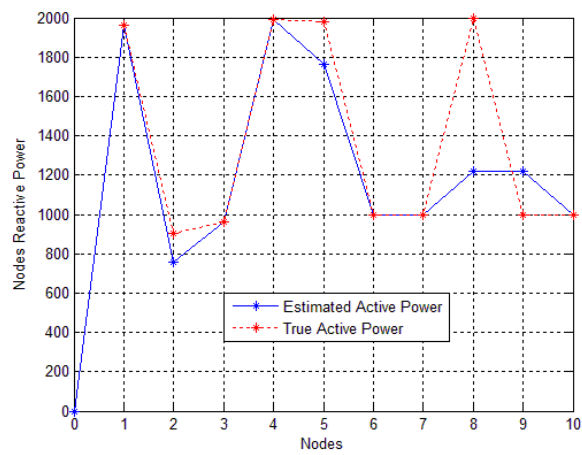
Results calculated with load flow algorithm are shown below:

Node	NODES POWER (absolute values)	
	Active Power [W]	Reactive Power [var]
1	2867.230	1960.94
2	1679.069	905.58
3	2871.656	962.24
4	2961.916	1988.80
5	2923.854	1977.60
6	1981.784	994.64
7	991.172	997.40
8	1994.489	1998.38
9	999.063	999.72
10	999.999	1000.00

If we put in a graph both Active Power values in a graph 3.8 and Reactive Power values in a graph 3.9, we can see the difference between results:



Graph 3.8: Compare between Estimated Active Power and True Active Power.



Graph 3.9: Compare between Estimated Reactive Power and True Reactive Power.

Nodes where measurements are not available, the difference between estimated power and true power is shown in table 3.7 and table 3.8:

Node	Estimated Active Power [W]	True Active Power [W]	Error (absolute value) [W]
2	1501.377	1679.069	177.692
5	2926.462	2923.854	2.608
8	1214.844	1994.489	779.644
9	1211.146	999.063	212.083

Table 3.7: Difference between Estimated Active Power and True Active Power.

Node	Estimated Reactive Power [var]	True Reactive Power [var]	Error (absolute value) [var]
2	758.762	905.579	146.817
5	1761.829	1977.602	215.773
8	1217.706	1998.38	780.674
9	1216.613	999.726	216.887

Table 3.8: Difference between Estimated Reactive Power and True Reactive Power.

The errors between true power and estimated power are bigger than the previous case which the comparison was between true voltage and estimated voltage. This is because it is difficult or impossible to estimate accurate power values if there are adjacent nodes where measurements are not available.

Simulation 3:

In this network load power and node voltage are not available in 4 nodes. Voltage and power measurements at nodes 2,5,7,8 and 9 are not available.

Branch data and load data are shown in Table 2.4 and 2.5 in Chapter 2. For our simulations, we used load flow calculation data as if node voltages and load powers were available from smart meters.

From algorithm the following results are obtained:

NODE'S VOLTAGES (absolute values)

Node	Magnitude Voltage [V]	Angle Voltage [°]
0	400.000	0.00
1	395.832	0.19
2	393.515	0.29
3	386.981	0.60
4	384.254	0.76
5	383.149	0.75
6	380.171	1.02
7	379.351	1.12
8	378.878	1.15
9	378.119	1.19
10	377.895	1.21

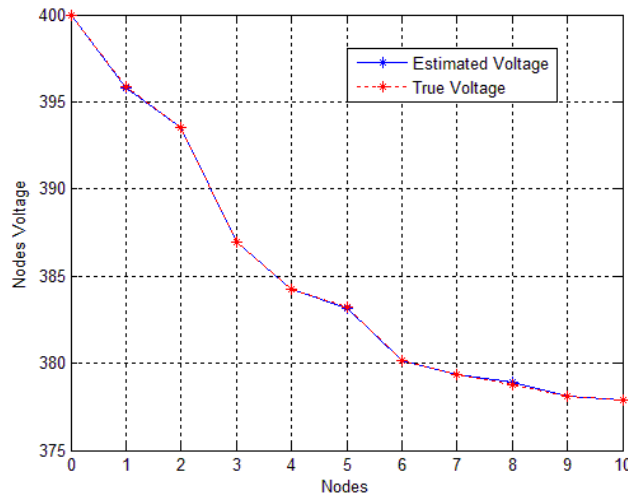
If we compare these results with those calculated using load flow algorithm, knowing all load powers, we can see that the difference between results is negligible. This is possible because the algorithm makes a lot of iterations to reached the convergence.

Results calculated with load flow algorithm are shown below:

NODE'S VOLTAGES (absolute values)

Node	Magnitude Voltage [V]	Angle Voltage [°]
0	400.000	0.00
1	395.836	0.19
2	393.503	0.29
3	386.979	0.60
4	384.256	0.76
5	383.198	0.82
6	380.171	1.03
7	379.332	1.09
8	378.794	1.14
9	378.120	1.19
10	377.895	1.21

If we put in a graph both nodes voltage values, we can see the difference between results:



Graph 3.10: Compare between Estimated voltage and True voltage.

Nodes where measurements are not available, the difference between estimated voltage and true voltage is negligible.

The difference is shown in table 3.9:

Node	Estimated Voltage	True Voltage	Error (absolute value)	Percentage error
2	393.515	393.503	0.012	0.003
5	383.149	383.198	0.049	0.012
7	379.351	379.332	0.019	0.005
8	378.878	378.794	0.084	0.021
9	378.119	378.120	0.001	0.0003

Table 3.9: Difference between estimated voltage and true voltage.

The same we can see for the power. From the algorithm, the following results in absolute values are obtained:

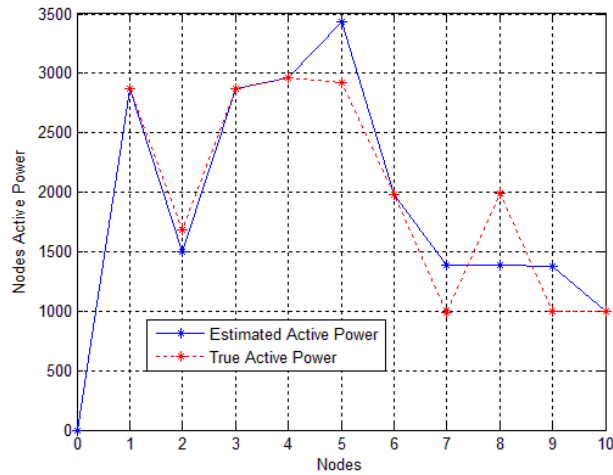
Node	NODE'S POWERS (absolute values)	
	Active Power [W]	Reactive Power [var]
1	2867.200	1960.90
2	1501.377	758.76
3	2871.700	962.20
4	2961.900	1988.80
5	3436.138	2063.42
6	1981.800	994.60
7	1385.653	928.58
8	1383.488	928.08
9	1379.943	927.37
10	1000.000	1000.00

If we compare these results with those calculated using load flow algorithm, knowing all loads powers, we can see that the difference between results is bigger than the previous simulation, this is because there are more adjacent nodes where measurements are not available.

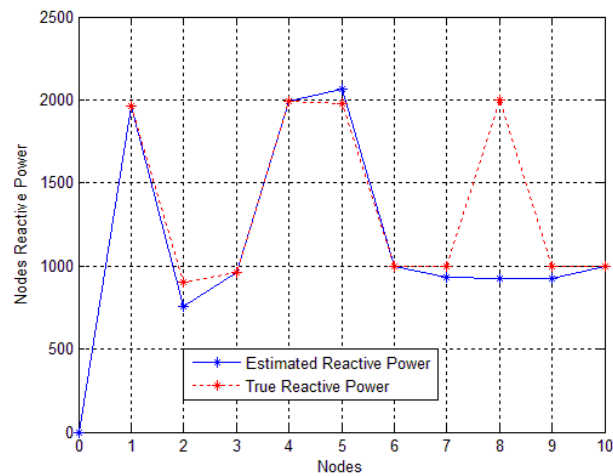
Results calculated with load flow algorithm are shown below:

Node	NODE'S POWERS (absolute values)	
	Active Power [W]	Reactive Power [var]
1	2867.230	1960.94
2	1679.069	905.58
3	2871.656	962.24
4	2961.916	1988.80
5	2923.854	1977.60
6	1981.784	994.64
7	991.172	997.40
8	1994.489	1998.38
9	999.063	999.72
10	999.999	1000.00

If we put in a graph both Active Power values in a graph 3.11 and Reactive Power values in a graph 3.12, we can see the difference between results:



Graph 3.11: Compare between Estimated Active Power and True Active Power.



Graph 3.12: Compare between Estimated Reactive Power and True Reactive Power.

Nodes where measurements are not available, the difference between estimated power and true power is shown in table 3.10 and table 3.11:

Node	Estimated Active Power [W]	True Active Power [W]	Error (absolute value) [W]
2	1501.377	1679.069	177.692
5	3436.138	2923.854	512.284
7	1385.653	991.172	394.4808
8	1383.488	1994.489	611.001
9	1379.943	999.063	380.880

Table 3.10: Difference between Estimated Active Power and True Active Power.

Node	Estimated Reactive Power [var]	True Reactive Power [var]	Error (absolute value) [var]
2	758.762	905.579	146.817
5	2063.422	1977.602	85.820
7	928.582	997.407	68.825
8	928.084	1998.380	1070.296
9	927.372	999.726	72.354

Table 3.11: Difference between Estimated Reactive Power and True Reactive Power.

The errors between true power and estimated power are bigger than the previous case which the comparison was between true voltage and estimated voltage.

This is because it is difficult or impossible to estimate accurate power values if there are adjacent nodes where measurements are not available.

Conclusions

In this report two problems have been addressed:

- 1) The importance of smart meters in energy saving;
- 2) Voltage estimation in radial distribution networks where Smart Meters do not work very well.

Regarding the first problem, we can say that Smart Meters offer opportunities both for suppliers and for consumers.

They are a gateway:

- for suppliers to improve market operation, both by identifying better ways to tackle their existing customer-service activities and by offering scope for new retail opportunities;
- for SMEs and households to achieve energy savings through improved feedback on consumption and expenditure, and, by beginning to enable demand-response at an individual level;
- for micro-generation.

Smart metering could help to boost security of supply:

- Through interval metering and time of use tariffs it could stimulate better demand response, demand reduction at peaks can become a resource.
This may apply particularly to business customers who are willing to shift load to save money.
- By metering import and export, smart meters could help facilitate the growth of micro-generation, which could also help with security of supply.

Carbon reduction is a very important target, and this could come about through:

- Energy saving if smart meters encourage consumers to use less energy in total;
- Load shifting if consumers respond to time of use or time of day tariffs by shifting their use to off-peak times. In this case the effect would depend upon the carbon intensity of the marginal plant.

If smart meters lead to energy saving this will benefit both customers and the environment. Based on the evidence available to date, a reasonable assumption would be a 1-3% reduction.

Customers may also be willing to load shift in response to differential tariffs.

When reduction is at the 3% level, supplier and network benefits is broadly equivalent to the installed costs of smart metering (£13-18 costs and £14-23 benefits).

Regarding the second problem we can say that to estimate the state of the distribution network is challenging due to:

1. the smart meter measurements are not heterogeneous
2. communication channels used for smart meters may introduce delays as long as several minutes
3. there may be erroneous measurements.

The scope of my research it was to seek a way to estimate the voltages of each distribution network node using radial load flow analysis.

To achieve this goal have been implemented several programs.

Initially I wrote a program in Matlab to calculate voltages and currents at each network node using radial flow method (backward and forward sweep) .

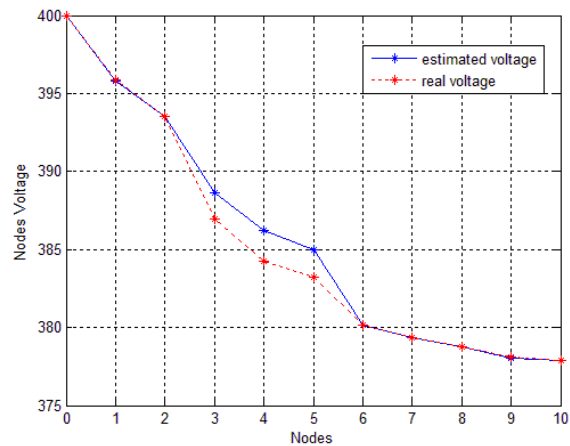
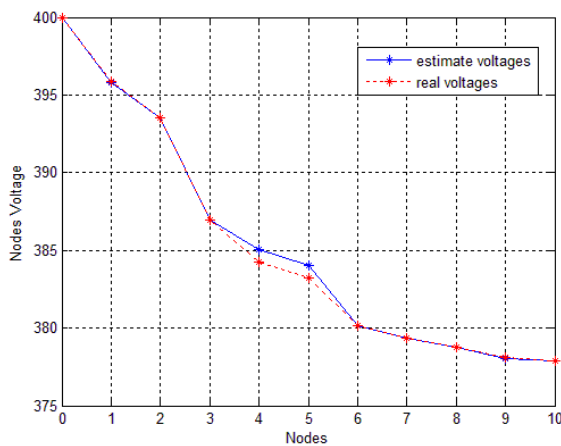
This was necessary to get all network information as if we had all measurements from Smart Meters.

Program converges very fast, after a few iterations the solution was found. As we can see in chapter 2, after 5 iterations the method converges because the error is less than 10^{-4} .

The next step was to write a program to estimate the voltages of each network node when more than a few smart meters not working properly.

The first attempt was to write a program to estimate the voltage where smart meters not working, assuming that the voltage profile was linear, consequently this drop voltage was linear.

The error obtained between the estimated voltage and true voltage is below:



In the first graph the error is smaller than in the second graph, because in the first graph we have two adjacent nodes where voltage and power measurements are not available (two adjacent smart meters that do not work properly), while in the second graph we have three adjacent nodes where voltage and power measurements are not available.

In both cases the maximum error that the program makes is respectively:

In the first graph:

Node	Estimated Voltage	True Voltage	Error (absolute value)	Percentage error
5	384.060	383.198	0.862	0.215

In the second graph:

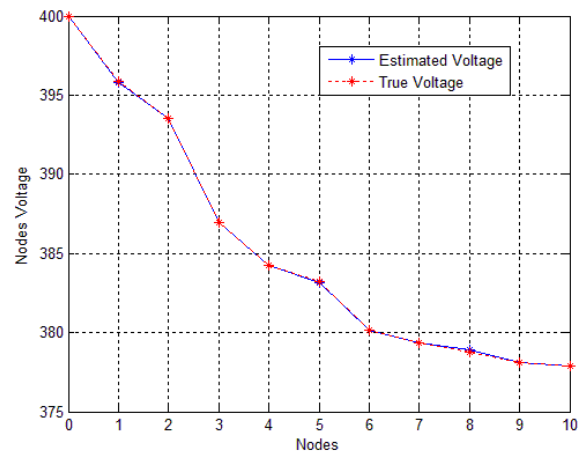
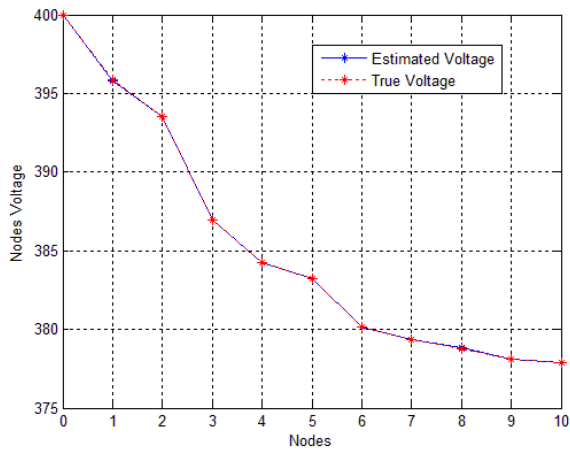
Node	Estimated Voltage	True Voltage	Error (absolute value)	Percentage error
4	386.219	384.256	1.963	0.491

We can see that the percentage error doubles in the second case where we have a major unknowns number.

Although this program may seem like a good way, but we must take note that if the voltage drop is linear, at node where measurements are not available the load's power is zero. This can be a great approximation.

To avoid this approximation as the second attempt, I wrote a program assuming a current value where voltage and power measurements are not available because smart meter not working. In this way each iteration, the current value changes of 2%, and the program calculates the new voltage and power value at each node. When the program achieves convergence the error is very little.

The error obtained between the estimated voltage and true voltage is below:



The first graph shows the error between estimated voltage and true voltage when there are two adjacent nodes where measurements are not available (node 8 and 9 addition to node 5). The second graph shows the estimated voltage profile and true voltage profile when there are three adjacent nodes where measurements are not available (node 7,8 and 9 addition to node 5).

In both cases the maximum error that the program makes is respectively:

In the first graph:

Node	Estimated Voltage	True Voltage	Error (absolute value)	Percentage error
8	378.865	378.794	0.071	0.018

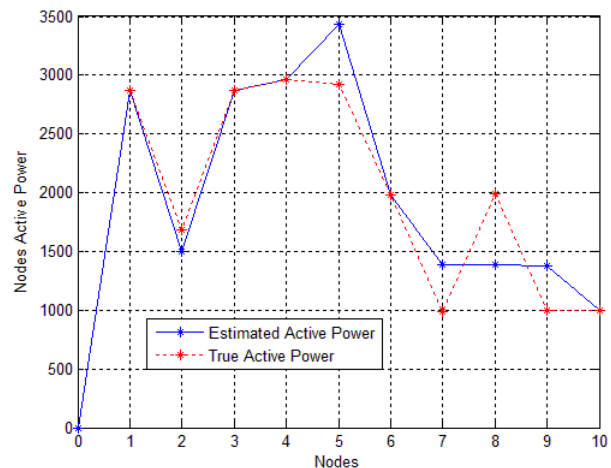
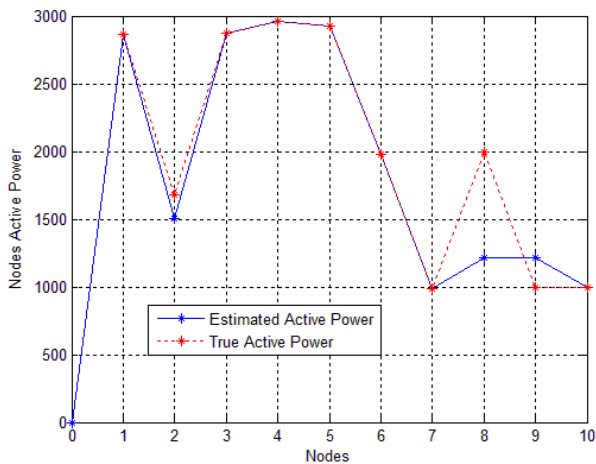
In the second graph:

Node	Estimated Voltage	True Voltage	Error (absolute value)	Percentage error
8	378.878	378.794	0.084	0.021

About the estimated voltage profile we can see that the error is very little, so the program works good.

Regarding load's powers the error between real power and estimated power is higher than error's voltage because is difficult to know how power distributed where there are more than one unknown adjacent measurement.

Active Power graphs are as follow:



As we can see the error is higher than previous case, in the first graph we have nodes 9,8 and 5 where measurements are not available, in the second graph we have nodes 9,8,7 and 5 where measurements are not available.

We can conclude by saying that the estimated voltage profile is very close to the real voltage profile while estimated loads power is not very close to the real loads power, for the reason explained above.

The result of my research about the estimated voltage it can be considered good because we can estimate the voltages of each distribution network node with a very small error regard true voltages.

REFERENCES

- [1] Gil Owen and Judith Ward. The consumer implications of smart meters. Sustainability first.
- [2] Gil Owen and Judith Ward. Smart meters: commercial, Policy and Regulatory Drivers. March 2006. Sustainability first.
- [3] Abdellatif Hamouda and Khaled Zehar. Efficient Load Flow Method for Radial Distribution Feeders. 2006 Asian Network for Scientific Information.
- [4] Nick Jenkins, Ron Allan, Peter crossley, Daniel kirschen and Goran Strbac. Embedded Generation. IET power and Energy series 31.
- [5] Power System
- [6] Directive 2002/91/EC
- [7] Wright et al. A review of the energy efficiency and other benefits of advanced utility metering. EA Technology, 2000.
- [8] The carbon Trust's advanced metering project. Metering International, Issue 3, 2005 and Carbon Trust presentation to Ofgem seminar, 02.03.06.
- [9] Except off-peak tariffs for electric heating and hot water, such as Economy 7.
- [10] Owen G and Ward J. Smart meters in Great Britain.
- [11] NIE keypad "Powershift" trial.
- [12] Energy Retail Association Specification.

Appendix 1

This appendix explains the algorithm used to calculate load flow in the second chapter of the paper. The algorithm is composed from:

- 1) main.m = is the main program that calls other programs
- 2) readdata.m = is the program that reads network's data and calculates load flow
- 3) results.m = is the program that writes results in a file "txt"
- 4) data.m = is a input file data

Inside the programs the current has been called with letter F and not with letter I.

1) main.m

```
% Efficient Load Flow Method for Radial Distribution Feeders
```

```
clear all
clc
```

```
FileDati=input('filename data (.m) = ','s') % File input data
FileOut=['load flow','.txt']; % File output data
eval(FileDati)
fprintf(1,'Calculation load flow')
```

```
readdata
```

```
fid=1;
results
fid=fopen(FileOut,'wt');
results;
fclose(fid);
```

2) readdata.m

```
% Data Acquisition Lines and Loads
```

```
% Branches
```

```
Nbranches=size(BRANCHES,1) % Numbers of Branches
Nnodes=size(LOADS,1) % Numbers of Loads
for kk=1:Nbranches % Read data
    N1=BRANCHES(1:kk,1) % Sending-end of branch
    N2=BRANCHES(1:kk,2) % Recieving-end of brunch
    Dist=BRANCHES(1:kk,4) % Branch lenght
    r=BRANCHES(1:kk,5) % Branch resistance
    xl=BRANCHES(1:kk,6) % Branch inductance
    xc=BRANCHES(1:kk,7) % Branch capacity
    Condsec=BRANCHES(1:kk,8) % Conductors sections
end
```

```
% Loads
```

```
for kk=1:Nnodes
    N=LOADS(1:kk,1); % Nodes Loads
    P=LOADS(1:kk,2); % Real Power
    Q=LOADS(1:kk,3); % Reactive Power
```

```

V=LOADS(1:kk,4);    % Nodes Voltage
end

Vbase=LOADS(1,4)    % Base Voltage
Nbase=sqrt(((LOADS(2,2))^2)+((LOADS(2,3))^2)) % Base Power
Fbase=Nbase/(Vbase*sqrt(3)) % Base current

% Calculating the transpose of vectors for row vectors

N1=transpose(N1)
N2=transpose(N2)
Dist=transpose(Dist)
r=transpose(r)
R=(r).*(Dist)
xl=transpose(xl)
X=(xl).*(Dist)
xc=transpose(xc)
Vbase=transpose(Vbase)
N=transpose(N)
P=transpose(P)
Q=transpose(Q)
V=transpose(V)

% Mathematical Formulation

cont=0;    % Variable count iterations
mass=100; % Variable tolerance
while mass > 0.0001 && cont < 10
cont=cont+1;

% Determination of the branches current by going up the line (backward
% sweep)
for k=1:Nnodes
    Pnodes(1,k)=sum(P(1,k:Nnodes));    % Nodes Active Power
end
Nnodes1=(Nnodes-1)
for k=1:Nnodes1
    Pbranches(1,k)=sum(P(1,k+1:Nnodes));    % Branches Active Power
end

for k=1:Nnodes
    Qnodes(1,k)=sum(Q(1,k:Nnodes));    % Nodes Reactive Power
end
for k=1:Nnodes1
    Qbranches(1,k)=sum(Q(1,k+1:Nnodes));    % Branches Reactive Power
end
Pnodes;
Qnodes;
Pbranches;
Qbranches;
for k=1:Nnodes1
    Vbranches(1,k)=V(1,k);    % Branches Voltage
end
Vb=abs(Vbranches);
Ploss=((Pbranches.*Pbranches+(Qbranches.*Qbranches))./(Vb.*Vb)).*R; % Active power loss

```

```

Qloss=((Pbranches.*Pbranches+(Qbranches.*Qbranches))./(Vb.*Vb)).*X; % Reactive power loss
Ploss;
Qloss;
Pi=(Pbranches)+(Ploss); % Branches Total Active Power
Qi=(Qbranches)+(Qloss); % Branches Total Reactive Power
Fi=(Pi-(Qi*j))./conj(Vbranches); % Branches Current
Fdi=real(Fi); % d components of the current
Fqi=imag(Fi); % q components of the current

% Determination the nodes voltage by going down the line (forward sweep)
Vold=V; % Old Voltage
for k=1:Nbranches
    V(1,k+1)=V(1,k)-((R(1,k)+(X(1,k)*j)).*Fi(1,k));
end
abs(V);
angle(V);
fi=angle(V)*180/pi;
diffV=(Vold)-(V);
abs(diffV);
mass=abs(max((Vold)-(V))); % Tolerance voltage nodes
end

```

3) results.m

```
Risp=input('Do you want to work in absolute values [1] or in p.u. [2] ?');
```

```

if Risp==1
    fprintf(fid,'\n          NODES VOLTAGE (val. assoluti)\n');
    fprintf(fid,' Node          Magnitude Voltage [V]          Angle Voltage [°]\n');
    for i=1:Nnodes
        Node=i;
        ModVolt=abs(V(1,i));
        ArgVolt=angle(V(1,i))*180/pi;
        fprintf(fid,'%5.0i      %18.3f      %18.2f\n',Node , ModVolt, ArgVolt);
    end

    fprintf(fid,'\n          BRANCHES CURRENT (val. assoluti)\n');
    fprintf(fid,' Branch          Magnitude Current [A]          Angle Current [°]\n');
    Pp=0;
    for i=1:Nbranches
        ModFbranch=abs(Fi(1,i));
        ArgFbranch=angle(Fi(1,i))*180/pi;
        fprintf(fid,'%5.0i      %18.3f      %18.2f\n',i , ModFbranch, ArgFbranch);
        Pp=3*BRANCHES(i,4)*BRANCHES(i,5)*ModFbranch^2+Pp;
    end
    fprintf(fid,'\n ACTIVE POWER LOSS BRANCH (W) %3.2f',Pp);

else
    fprintf(fid,'\n          NODES VOLTAGE (p.u.)\n');
    fprintf(fid,' Node          Magnitude Voltage [p.u.]          Angle Voltage [°]\n');
    for i=1:Nnodes
        Node=i;
        ModVolt=abs(V(1,i))/Vbase;
        ArgVolt=angle(V(1,i)/Vbase)*180/pi;

```

```

    fprintf(fid,'%5.0i      % 18.3f      % 18.2f\n',Node , ModVolt, ArgVolt);
end

fprintf(fid,'\n          BRANCHES CURRENT (p.u.)\n');
fprintf(fid,' Branch      Magnitude Current [p.u.]      Angle Current [°]\n');
Pp=0;
for i=1:Nbranches
    Pp=3*BRANCHES(i,4)*BRANCHES(i,5)*(abs(Fi(1,i))^2)+Pp;
    ModFbranch=abs(Fi(1,i))/Fbase;
    ArgFbranch=angle(Fi(1,i)/Fbase)*180/pi;
    fprintf(fid,'%5.0i      % 18.3f      % 18.2f\n',i , ModFbranch, ArgFbranch);
end
Pp=Pp/Nbase;
fprintf(fid,'\n ACTIVE POWER LOSS BRANCH (p.u.) %3.2f,Pp);
end

```

4) data.m

```

% BRANCHES: contains data of the branches
% [n1 n2 Vnom(V) d(km) r(ohm/km) xl(ohm/km) xc(ohm/km)
% Nom.Sect(mm2) Max.Curr(A)]
%
% LOADS: contains data of the loads
% [node ActiveP(W) ReactiveQ(var) Vbase(V)]

```

```

BRANCHES=[0 1 20e3+0j 9 0.118 0.35 0 150 477
           1 2 20e3+0j 13 0.118 0.35 0 150 477
           2 3 20e3+0j 8 0.118 0.35 0 150 477
           3 4 20e3+0j 10 0.118 0.35 0 150 477
           4 5 20e3+0j 5 0.118 0.35 0 150 477];

```

```

LOADS=[0 0 0 20e3+0j
        1 323e3 200e3 20e3+0j
        2 320e3 240e3 20e3+0j
        3 315e3 152e3 20e3+0j
        4 400e3 300e3 20e3+0j
        5 338e3 298e3 20e3+0j];

```

Appendix 2

This appendix explains the algorithm used to calculate the voltage in the special case of the third chapter.

The first algorithm is composed from:

- 1) main.m = is the main program that calls other programs;
- 2) readdata1.m = is the program that reads network's data and calculates voltages where measurements are not available;
- 3) readdata2.m = is the program that reads network's data and calculates load flow;
- 4) results1.m = is the program that writes calculated voltage results in a file "txt";
- 5) results2.m = is the program that writes load flow results in a file "txt";
- 6) difference.m = is the program that calculates the difference between calculated voltage and true voltage;
- 7) data.m = is a input file data.

Inside the programs the current has been called with letter F and not with letter I.

1) main.m

```
% main.m
% Efficient Load Flow Method for Radial Distribution Feeders

clear all
clc

FileDati=input('filename data (.m) = ','s') % File input data
FileOut1=['load flow without some measurements','.txt'];
FileOut2=['load flow','.txt'];           % File output data
FileOut3=['difference between true values and calculated voltages','.txt'];
eval(FileDati)
fprintf(1,'Calculation load flow')

readdata1

fid=1;
results1
fid=fopen(FileOut1,'wt');
results1;
fclose(fid);

readdata2

fid=1;
results2
fid=fopen(FileOut2,'wt');
results2;
fclose(fid);

fid=1;
difference
fid=fopen(FileOut3,'wt');
difference;
fclose(fid);

plot(N,abs(V),'b-*');
```

```

xlabel('Nodes')
ylabel('Nodes Voltage [p.u.]')
grid on
hold on
plot(N,abs(V2),'r:*');
hold on

```

2) readdata1.m

```
% Data Acquisition Lines and Loads
```

```
% Branches
```

```

Nbranches=size(BRANCHES,1); % Numbers of Branches
Nnodes=size(LOADS,1); % Numbers of Loads
for kk=1:Nbranches; % Read data
    N1=BRANCHES(1:kk,1); % Sending-end of branch
    N2=BRANCHES(1:kk,2); % Recieving-end of brunch
    Dist=BRANCHES(1:kk,4); % Branch lenght
    r=BRANCHES(1:kk,5); % Branch resistance
    xl=BRANCHES(1:kk,6); % Branch inductance
    xc=BRANCHES(1:kk,7); % Branch capacity
    Condsec=BRANCHES(1:kk,8); % Conductors sections
end

```

```
% Loads
```

```

for kk=1:Nnodes
    N=LOADS(1:kk,1); % Nodes Loads
    P=LOADS(1:kk,2); % Real Power
    Q=LOADS(1:kk,3); % Reactive Power
    V=LOADS(1:kk,4); % Nodes Voltage
end

```

```
% Calculating the transpose of vectors for row vectors
```

```

N1=transpose(N1);
N2=transpose(N2);
Dist=transpose(Dist);
r=transpose(r);
R=(r).*(Dist);
xl=transpose(xl);
X=(xl).*(Dist);
xc=transpose(xc);
N=transpose(N);
P=transpose(P);
Q=transpose(Q);
V=transpose(V);

```

```
% Mathematical Formulation
```

```

T=0;
while T==0

```

```

Vinv=V(1,Nnodes:-1:1);
[T,I] = min(Vinv); % nodo where I don't know the measures

```



```

k=Nnodes+1-I;
s=Nnodes;

cont1=0; % Variable count iterations
mass1=100; % Variable tolerance
while mass1 > 0.00001 && cont1 < 1
cont1=cont1+1;

% Found node calculating the nodes voltage
Vold=V; % old nodes voltage
Vc=conj(V)
if k<(s-1)
F=((P(1,k+1)+(Q(1,k+1)*j))/(sqrt(3)*(V(1,k+1))))+(V(1,k+1)-
V(1,k+2))/(sqrt(3)*((R(1,k+1)+(X(1,k+1)*j)))); % current flowing through the branch after node with
unknown voltage
deltaVafter=(R(1,k)+(X(1,k)*j))*F*sqrt(3); % drop voltage after node with unknown voltage
Vunknown=V(1,k+1)+deltaVafter % unknown voltage
V(1,k)=Vunknown; % unknown voltage array component
Vall=V; % nodes voltage array
moduleV=abs(Vall);
argV=angle(Vall);
deltaVbefore=V(1,k-1)-V(1,k);
Fbefore=deltaVbefore/(sqrt(3)*(R(1,k-1)+(X(1,k-1)*j))) % current flowing through the branch before node
with unknown voltage
Fafter=F % current flowing through the branch after node with unknown
voltage
Fi=Fbefore-Fafter % current of unknown load
moduleF=abs(Fi);
argF=angle(Fi);
Ni=sqrt(3)*V(1,k)*(Fi); % power of unknown load
else
if k==(s-1)
deltaVbetween=V(1,k-1)-V(1,k+1); % voltage drop between the two nodes straddling the
unknown voltage
F=((P(1,k+1)+(Q(1,k+1)*j))/(sqrt(3)*(V(1,k+1)))); % current flowing through the branch after node
with unknown voltage
deltaVafter=(R(1,k)+(X(1,k)*j))*F*sqrt(3); % drop voltage after node with unknown voltage
deltaVbefore=deltaVbetween-deltaVafter; % drop voltage before node with unknown voltage
Vunknown=V(1,k+1)+deltaVafter; % unknown voltage
V(1,k)=Vunknown; % unknown voltage array component
Vall=V; % nodes voltage array
argV=angle(Vall);
Fbefore=deltaVbefore/(sqrt(3)*(R(1,k-1)+(X(1,k-1)*j))); % current flowing through the branch before
node with unknown voltage
Fafter=deltaVafter/(sqrt(3)*(R(1,k)+(X(1,k)*j))); % current flowing through the branch after node
with unknown voltage
Fi=Fbefore-Fafter; % current of unknown load
moduleF=abs(Fi);
Ni=V(1,k)*(Fi)*sqrt(3); % power of unknown load

% alternativ mode Vunknown=V(1,k+1)+deltaVafter

else
deltaV=V(1,k-2)-V(1,k-1); % drop voltage before node with unknown voltage

```

```

    FBefore=deltaV/sqrt(3)*(R(1,k-2)+(X(1,k-2)*j));          % current flowing through the branch before
node with unknown voltage
    FLoad=((P(1,k-1)+(Q(1,k-1)*j))/(sqrt(3)*(V(1,k-1)))); % current load (load before the unknown load)
    F=FBefore-FLoad;                                       % current of unknown load
    Vunknown=V(1,k-1)-(sqrt(3)*F*(R(1,k-1)+(X(1,k-1)*j))); % unknown voltage
    V(1,k)=Vunknown;                                       % unknown voltage array component
    Vall=V;                                                % nodes voltage array
    Ni=V(1,k)*(F)*sqrt(3);                                  % power of unknown load
    end
end

```

```

V=Vall;
mass1=abs(max((Vold)-(V)));
end
P(1,k)=real(Ni);
Q(1,k)=imag(Ni);
P=P;          % Active Power load
Q=Q;          % Reactive Power load
[T,I] = min(V);
end

```

```

Fload=(P+(Q*j))./(sqrt(3)*(V)); % Current load

```

3) readdata2.m

```

% Data Acquisition Lines and Loads

```

```

% Branches
Nbranches=size(BRANCHES,1); % Numbers of Branches
Nnodes=size(LOADS2,1);     % Numbers of Loads
for kk=1:Nbranches         % Read data
    N1=BRANCHES(1:kk,1);   % Sending-end of branch
    N2=BRANCHES(1:kk,2);   % Recieving-end of brunch
    Dist=BRANCHES(1:kk,4); % Branch lenght
    r=BRANCHES(1:kk,5);    % Branch resistance
    xl=BRANCHES(1:kk,6);   % Branch inductance
    xc=BRANCHES(1:kk,7);   % Branch capacity
    Condsec=BRANCHES(1:kk,8); % Conductors sections
end

```

```

% Loads
for kk=1:Nnodes
    N2=LOADS2(1:kk,1); % Nodes Loads
    P2=LOADS2(1:kk,2); % Real Power
    Q2=LOADS2(1:kk,3); % Reactive Power
    V2=LOADS2(1:kk,4); % Nodes Voltage
end

```

```

Vbase=LOADS2(1,4); % Base Voltage
Nbase=sqrt(((LOADS2(2,2))^2)+((LOADS2(2,3))^2)); % Base Power
Fbase=Nbase/(Vbase*sqrt(3)); % Base current

```

```

% Calculating the transpose of vectors for row vectors

```

```

N1=transpose(N1);
N2=transpose(N2);
Dist=transpose(Dist);
r=transpose(r);
R=(r).*(Dist);
x1=transpose(x1);
X=(x1).*(Dist);
xc=transpose(xc);
Vbase=transpose(Vbase);
N2=transpose(N2);
P2=transpose(P2);
Q2=transpose(Q2);
V2=transpose(V2);

% Mathematical Formulation

cont=0; % Variable count iterations
mass=100; % Variable tolerance
while mass > 0.0001 && cont < 10
cont=cont+1;

% Determination of the branches current by going up the line (backward
% sweep)
for k=1:Nnodes
P2nodes(1,k)=sum(P2(1,k:Nnodes)); % Nodes Active Power
end
Nnodes1=(Nnodes-1);
for k=1:Nnodes1
P2branches(1,k)=sum(P2(1,k+1:Nnodes)); % Branches Active Power
end

for k=1:Nnodes
Q2nodes(1,k)=sum(Q2(1,k:Nnodes)); % Nodes Reactive Power
end
for k=1:Nnodes1
Q2branches(1,k)=sum(Q2(1,k+1:Nnodes)); % Branches Reactive Power
end
P2nodes;
Q2nodes;
P2branches;
Q2branches;
for k=1:Nnodes1
V2branches(1,k)=V2(1,k); % Branches Voltage
end
Vb=abs(V2branches);
P2loss=((P2branches.*P2branches+(Q2branches.*Q2branches))./(Vb.*Vb)).*R; % Active power loss
Q2loss=((P2branches.*P2branches+(Q2branches.*Q2branches))./(Vb.*Vb)).*X; % Reactive power loss
P2loss;
Q2loss;
Pi2=(P2branches)+(P2loss); % Branches Total Active Power
Qi2=(Q2branches)+(Q2loss); % Branches Total Reactive Power
Fi2=(Pi2-(Qi2*j))./(sqrt(3)*conj(V2branches)); % Branches Current
Fdi2=real(Fi2); % d components of the current
Fqi2=imag(Fi2); % q components of the current

```

```

% Determination the nodes voltage by going down the line (forward sweep)
Vold2=V2 % Old Voltage
for k=1:Nbranches
    V2(1,k+1)=V2(1,k)-(sqrt(3)*((R(1,k)+(X(1,k)*j)).*Fi2(1,k)));
end
V2=V2
abs(V2);
angle(V2);
fi=angle(V2)*180/pi;
diffV2=(Vold2)-(V2)
abs(diffV2);
mass=abs(max((Vold2)-(V2))) % Tolerance voltage nodes
end

for kk=1:Nbranches
    if kk<Nbranches
        Fload(1,kk)=Fi2(1,kk)-Fi2(1,kk+1);
    else
        Fload(1,kk)=Fi2(1,kk);
    end
end

for kk=1:Nbranches
    Nload(1,kk)=sqrt(3)*V2(1,kk+1)*conj(Fload(1,kk));
end
Nload;
Pload=real(Nload);
Qload=imag(Nload);

```

4) results1.m

```
Risp=input('Do you want to work in absolute values [1] or in p.u. [2] ?');
```

```

if Risp==1
    fprintf(fid,'\n          NODES VOLTAGE (absolute values)\n');
    fprintf(fid,' Node          Magnitude Voltage [V]          Angle Voltage [°]\n');
    for i=1:Nnodes
        Node=i;
        ModVolt=abs(V(1,i));
        ArgVolt=angle(V(1,i))*180/pi;
        fprintf(fid,'%5.0i      %18.3f      %18.2f\n',Node , ModVolt, ArgVolt);
    end
    fprintf(fid,'\n          LOADS CURRENT (absolute values)\n');
    fprintf(fid,' Node          Magnitude Current [A]          Angle Current [°]\n');
    for i=1:Nnodes
        Node=i;
        ModFload=abs(Fload(1,i));
        ArgFload=angle(Fload(1,i))*180/pi;
        fprintf(fid,'%5.0i      %18.3f      %18.2f\n',Node , ModFload, ArgFload);
    end
    fprintf(fid,'\n          NODES POWER (absolute values)\n');
    fprintf(fid,' Node          Active Power [W]          Reactive Power [var]\n');
    for i=1:Nnodes
        Node=i;

```

```

    ActivePower=P(1,i);
    ReactivePower=Q(1,i);
    fprintf(fid,'%5.0i    %18.3f    %18.2f\n',Node , ActivePower, ReactivePower);
end
else
    fprintf(fid,'\n          NODES VOLTAGE (p.u.)\n');
    fprintf(fid,' Node          Magnitude Voltage [p.u.]          Angle Voltage [°]\n');
    Vbase=LOADS(1,4);          % Base Voltage
    Nbase=sqrt((LOADS(2,2))^2+((LOADS(2,3))^2)); % Base Power
    Fbase=Nbase/(Vbase*sqrt(3)); % Base current
    V=V./Vbase;
    for i=1:Nnodes
        Node=i;
        ModVolt=abs(V(1,i));
        ArgVolt=angle(V(1,i))*180/pi;
        fprintf(fid,'%5.0i    %18.3f    %18.2f\n',Node , ModVolt, ArgVolt);
    end
    Fload=Fload./Fbase;
    fprintf(fid,'\n          LOADS CURRENT (p.u.)\n');
    fprintf(fid,' Node          Magnitude Current [p.u.]          Angle Current [°]\n');
    for i=1:Nnodes
        Node=i;
        ModFload=abs(Fload(1,i));
        ArgFload=angle(Fload(1,i))*180/pi;
        fprintf(fid,'%5.0i    %18.3f    %18.2f\n',Node , ModFload, ArgFload);
    end
    fprintf(fid,'\n          NODES POWER (p.u.)\n');
    fprintf(fid,' Node          Active Power [p.u.]          Reactive Power [p.u.]\n');
    P=P./Nbase;
    Q=Q./Nbase;
    for i=1:Nnodes
        Node=i;
        ActivePower=P(1,i);
        ReactivePower=Q(1,i);
        fprintf(fid,'%5.0i    %18.3f    %18.2f\n',Node , ActivePower, ReactivePower);
    end
end
end

```

5) results2.m

Risp=input('Do you want to work in absolute values [1] or in p.u. [2] ?');

```

if Risp==1
    fprintf(fid,'\n          NODES VOLTAGE (absolute values)\n');
    fprintf(fid,' Node          Mod. Voltage [V]          Arg. Voltage [°]\n');
    for i=1:Nnodes
        Node=i;
        ModVolt=abs(V2(1,i));
        ArgVolt=angle(V2(1,i))*180/pi;
        fprintf(fid,'%5.0i    %18.3f    %18.2f\n',Node , ModVolt, ArgVolt);
    end

    fprintf(fid,'\n          BRANCHES CURRENT (absolute values)\n');
    fprintf(fid,' Branch          Mod. Current [A]          Arg. Current [°]\n');

```

```

for i=1:Nbranches
    ModFbranch=abs(Fi2(1,i));
    ArgFbranch=angle(Fi2(1,i))*180/pi;
    fprintf(fid,'%5.0i    %18.3f    %18.2f\n',i , ModFbranch, ArgFbranch);
end

fprintf(fid,'\n          LOADS CURRENT (absolute values)\n');
fprintf(fid,' Node          Mod. Current [A]          Arg. Current [°]\n');
for i=1:Nbranches
    Node=i+1;
    ModFload=abs(Fload(1,i));
    ArgFload=angle(Fload(1,i))*180/pi;
    fprintf(fid,'%5.0i    %18.3f    %18.2f\n',Node , ModFload, ArgFload);
end

fprintf(fid,'\n          LOADS POWER (absolute values)\n');
fprintf(fid,' Node          Active Power [W]          Reactive Power [var]\n');
for i=1:Nbranches
    Node=i+1;
    ActivePower=Pload(1,i);
    ReactivePower=Qload(1,i);
    fprintf(fid,'%5.0i    %18.3f    %18.2f\n',Node , ActivePower, ReactivePower);
end
else
    Vbase=LOADS(1,4); % Base Voltage
    Nbase=sqrt(((LOADS(2,2))^2)+((LOADS(2,3))^2)); % Base Power
    Fbase=Nbase/(Vbase*sqrt(3)); % Base current
    fprintf(fid,'\n          NODES VOLTAGE (p.u.)\n');
    fprintf(fid,' Node          Mod. Voltage [p.u.]          Arg. Voltage [°]\n');
    V2=V2./Vbase;
    for i=1:Nnodes
        Node=i;
        ModVolt=abs(V2(1,i));
        ArgVolt=angle(V2(1,i))*180/pi;
        fprintf(fid,'%5.0i    %18.3f    %18.2f\n',Node , ModVolt, ArgVolt);
    end

    fprintf(fid,'\n          BRANCHES CURRENT (p.u.)\n');
    fprintf(fid,' Branch          Mod. Current [p.u.]          Arg. Current [°]\n');
    Fi2=Fi2./Fbase;
    for i=1:Nbranches
        ModFbranch=abs(Fi2(1,i));
        ArgFbranch=angle(Fi2(1,i))*180/pi;
        fprintf(fid,'%5.0i    %18.3f    %18.2f\n',i , ModFbranch, ArgFbranch);
    end

    fprintf(fid,'\n          LOADS CURRENT (p.u.)\n');
    fprintf(fid,' Node          Mod. Current [p.u.]          Arg. Current [°]\n');
    Fload=Fload./Fbase;
    for i=1:Nbranches
        Node=i+1;
        ModFload=abs(Fload(1,i));
        ArgFload=angle(Fload(1,i))*180/pi;
        fprintf(fid,'%5.0i    %18.3f    %18.2f\n',Node , ModFload, ArgFload);
    end
end

```

```

fprintf(fid,'\n                LOADS POWER (p.u.)\n');
fprintf(fid,' Node          Active Power [p.u.]          Reactive Power [p.u.]\n');
Pload=Pload./Nbase;
Qload=Qload./Nbase;
for i=1:Nbranches
    Node=i+1;
    ActivePower=Pload(1,i);
    ReactivePower=Qload(1,i);
    fprintf(fid,'%5.0i          %18.3f          %18.2f\n',Node , ActivePower, ReactivePower);
end
end

```

6) difference.m

```
Risp=input('Do you want to work in absolute values [1] or in p.u. [2] ?');
```

```

if Risp==1
fprintf(fid,'\n DIFFERENCE BETWEEN TRUE VOLTAGE & CALCULATED VOLTAGE (absolute
values)\n');
fprintf(fid,' Node          Magnitude Voltage [V]\n');
for i=1:Nnodes
    Node=i;
    ModVolt=abs(V(1,i)-V2(1,i));
    fprintf(fid,'%5.0i          %18.3f\n',Node , ModVolt);
end
else
Vbase=LOADS(1,4);          % Base Voltage
fprintf(fid,'\n DIFFERENCE BETWEEN TRUE VOLTAGE & CALCULATED VOLTAGE (p.u.)\n');
fprintf(fid,' Node          Magnitude Voltage [p.u.]\n');
V2=V2./Vbase;
V=V./Vbase;
for i=1:Nnodes
    Node=i;
    ModVolt=abs(V(1,i)-V2(1,i));
    fprintf(fid,'%5.0i          %18.3f\n',Node , ModVolt);
end
end

```

7) data.m

```

% realnetworksmart.m
% BRANCHES: contains data of the branches
% [n1 n2 Vnom(V) d(km) r(ohm/km) xl(ohm/km) xc(ohm/km)
% Nom.Sect(mm2) Max.Curr(A)]
%
% LOADS: contains data of the loads
% [node ActiveP(W) ReactiveQ(var) Vbase(V)]

```

```

BRANCHES=[0 1 400+0j 0.2 0.328 0.0965 0 75 192
           1 2 400+0j 0.13 0.328 0.0965 0 75 192
           2 3 400+0j 0.4 0.328 0.0965 0 75 192

```

```

3 4 400+0j 0.2 0.328 0.0965 0 75 192
4 5 400+0j 0.1 0.328 0.0965 0 75 192
5 6 400+0j 0.4 0.328 0.0965 0 75 192
6 7 400+0j 0.15 0.328 0.0965 0 75 192
7 8 400+0j 0.12 0.328 0.0965 0 75 192
8 9 400+0j 0.3 0.328 0.0965 0 75 192
9 10 400+0j 0.2 0.328 0.0965 0 75 192];

```

```

LOADS=[0 0 0 400
1 2867.2 1960.9 395.83 + 1.28i
2 0 0 0
3 2871.7 962.2 386.96 + 4.06i
4 2961.9 1988.8 384.22 + 5.08i
5 0 0 0
6 1981.8 994.6 380.11 + 6.80i
7 991.2 997.4 379.26 + 7.25i
8 1994.5 1998.4 378.72 + 7.53i
9 0 0 0
10 1000 1000 377.81 + 8.00i];

```

```

LOADS2=[0 0 0 400+0j
1 3e3 2e3 400+0j
2 2e3 1e3 400+0j
3 3e3 1e3 400+0j
4 3e3 2e3 400+0j
5 3e3 2e3 400+0j
6 2e3 1e3 400+0j
7 1e3 1e3 400+0j
8 2e3 2e3 400+0j
9 1e3 1e3 400+0j
10 1e3 1e3 400+0j];

```


Appendix 3

This appendix explains the algorithm used to estimate the voltage in the generalized case of the third chapter.

The first algorithm is composed from:

- 1) main.m = is the main program that calls other programs;
- 2) calculation.m = is the program that reads network's data and estimates voltages where measurements are not available;
- 3) readdata2.m = is the program that reads network's data and calculates load flow;
- 4) results1.m = is the program that writes estimated voltage results in a file "txt";
- 5) results2.m = is the program that writes load flow results in a file "txt";
- 6) difference.m = is the program that calculates the difference between calculated voltage and true voltage;
- 7) data.m = is a input file data.

Inside the programs the current has been called with letter F and not with letter I.

1) **main.m**

```
clear all
clc
```

```
FileDati=input('filename data (.m) = ','s') % File input data
FileOut1=['load flow without some measurements','.txt'];
FileOut2=['load flow','.txt'];           % File output data
FileOut3=['difference between true value and calculated voltage','.txt'];
eval(FileDati)
fprintf(1,'Calculation load flow')
```

```
calculation
```

```
fid=1;
results1
fid=fopen(FileOut1,'wt');
results1;
fclose(fid);
```

```
readdata2
```

```
fid=1;
results2
fid=fopen(FileOut2,'wt');
results2;
fclose(fid);
```

```
fid=1;
difference
fid=fopen(FileOut3,'wt');
difference;
fclose(fid);
```

```
plot(N,abs(V),'b-*');;
xlabel('Nodes')
ylabel('Nodes Voltage')
grid on
```

```

hold on
plot(N,abs(V2),'r:*');
hold on

```

2) calculation.m

```
% Data Acquisition Lines and Loads
```

```

% Branches
Nbranches=size(BRANCHES,1); % Numbers of Branches
Nnodes=size(LOADS,1); % Numbers of Loads
for kk=1:Nbranches; % Read data
    N1=BRANCHES(1:kk,1); % Sending-end of branch
    N2=BRANCHES(1:kk,2); % Recieving-end of brunch
    Dist=BRANCHES(1:kk,4); % Branch lenght
    r=BRANCHES(1:kk,5); % Branch resistance
    xl=BRANCHES(1:kk,6); % Branch inductance
    xc=BRANCHES(1:kk,7); % Branch capacity
    Condsec=BRANCHES(1:kk,8); % Conductors sections
end

```

```

% Loads
for kk=1:Nnodes
    N=LOADS(1:kk,1); % Nodes Loads
    P=LOADS(1:kk,2); % Real Power
    Q=LOADS(1:kk,3); % Reactive Power
    V=LOADS(1:kk,4); % Nodes Voltage
end

```

```
% Calculating the transpose of vectors for row vectors
```

```

N1=transpose(N1);
N2=transpose(N2);
Dist=transpose(Dist);
r=transpose(r);
R=(r).*(Dist);
xl=transpose(xl);
X=(xl).*(Dist);
xc=transpose(xc);
N=transpose(N);
P=transpose(P);
Q=transpose(Q);
V=transpose(V);

```

```
% Mathematical Formulation
```

```

T=0;
while T==0

```

```

Vinv=V(1,Nnodes:-1:1);
[T,I] = min(Vinv); % nodo where I don't know the measures

```

```

k=Nnodes+1-I;
s=Nnodes;

```

```

% Found node calculating the nodes voltage
Vold=V; % old nodes voltage
Vc=conj(V);
if k<(s-1)
    F=((P(1,k+1)+(Q(1,k+1)*j))/(sqrt(3)*(V(1,k+1))))+(V(1,k+1)-
V(1,k+2))/(sqrt(3)*((R(1,k+1)+(X(1,k+1)*j)))); % current flowing through the branch after node with
unknown voltage
    deltaVafter=(R(1,k)+(X(1,k)*j))*F*sqrt(3); % drop voltage after node with unknown voltage
    Vunknown=V(1,k+1)+deltaVafter % unknown voltage
    V(1,k)=Vunknown; % unknown voltage array component
    Vall=V; % nodes voltage array
    moduleV=abs(Vall);
    argV=angle(Vall);

else % k==(s-1)
    F=((P(1,k+1)+(Q(1,k+1)*j))/(sqrt(3)*(V(1,k+1)))); % current flowing through the branch after node
with unknown voltage
    deltaVafter=(R(1,k)+(X(1,k)*j))*F*sqrt(3); % drop voltage after node with unknown voltage
    Vunknown=V(1,k+1)+deltaVafter; % unknown voltage
    V(1,k)=Vunknown; % unknown voltage array component
    Vall=V; % nodes voltage array
    argV=angle(Vall);

end

V=Vall;
T=1;
end

% Fino a qui il programma calcola la tensione al primo nodo dove non
% conosco la tensione senza errore.

% Ora iniziamo con il secondo programma

T=0;
while T==0

while T==0
Vinv=V(1,Nnodes:-1:1);
[T,I] = min(Vinv); % nodo where I don't know the measures

k=Nnodes+1-I;
kfirst=k;
V(1,k)=1;

Vinv=V(1,Nnodes:-1:1);
[T,I] = min(Vinv); % nodo where I don't know the measures
k=Nnodes+1-I;
ksecond=k;

if ksecond==kfirst-1
    adj=2;
else adj=1;
end
end

```

```

if  adj==1;
    k=kfirst;
    Z1=sum(R(1,k:(k+(adj-1))))+(sum(X(1,k:(k+(adj-1))))*j);
    Z2=sum(R(1,k-1))+sum(X(1,k-1))*j;
    Zt=Z1+Z2;
    V(1,k)=(V(1,k-1)*Z1 + V(1,k+adj)*Z2)/Zt;

else
    k=ksecond;
    V(1,k)=1;
    Vinv=V(1,Nnodes:-1:1);
    [T,I] = min(Vinv);
    k=Nnodes+1-I;
    kthird=k;
end
if  adj==1
    break

elseif  kthird==ksecond-1
    adj=3;

else  adj=2
end
if  adj==1
    break
elseif  adj==2

    k=ksecond;
    Z1=sum(R(1,k:(k+(adj-1))))+(sum(X(1,k:(k+(adj-1))))*j);
    Z2=sum(R(1,k-1))+sum(X(1,k-1))*j;
    Zt=Z1+Z2;
    V(1,k)=(V(1,k-1)*Z1 + V(1,k+adj)*Z2)/Zt;

    Z4=sum(R(1,k-1:(k+(adj-2))))+(sum(X(1,k-1:(k+(adj-2))))*j);
    Z3=sum(R(1,k+adj-1))+sum(X(1,k+adj-1))*j;
    Zt=Z3+Z4;
    V(1,k+1)=(V(1,k-1)*Z3 + V(1,k+adj)*Z4)/Zt;

else

    k=kthird;
    V(1,k)=1;
    Vinv=V(1,Nnodes:-1:1);
    [T,I] = min(Vinv);
    k=Nnodes+1-I;
    kfourth=k;
end
if  adj==1 || adj==2
    break

```

```

elseif kfourth==kthird-1
    adj=4;

else adj=3
end
if adj==1 || adj==2
    break
elseif adj==3
    k=kthird;
    Z1=sum(R(1,k:(k+(adj-1))))+(sum(X(1,k:(k+(adj-1))))*j);
    Z2=sum(R(1,k-1))+(sum(X(1,k-1))*j);
    Zt=Z1+Z2;
    V(1,k)=(V(1,k-1)*Z1 + V(1,k+adj)*Z2)/Zt;

    Z4=sum(R(1,k-1:(k+(adj-2))))+(sum(X(1,k-1:(k+(adj-2))))*j);
    Z3=sum(R(1,k+adj-1))+(sum(X(1,k+adj-1))*j);
    Zt=Z3+Z4;
    V(1,k+2)=(V(1,k-1)*Z3 + V(1,k+adj)*Z4)/Zt;

    Z5=sum(R(1,k+1:(k+(adj-1))))+(sum(X(1,k+1:(k+(adj-1))))*j);
    Z6=sum(R(1,k-1:k))+(sum(X(1,k-1:k))*j);
    Zt=Z1+Z2;
    V(1,k+1)=(V(1,k-1)*Z5 + V(1,k+adj)*Z6)/Zt;

else

    fprintf('Error: Too adjacent S.M are not working');
end
end
V=V;
[T,I] = min(V);
end

V=V;

```

- 3) **readdata2.m** is identical to program in Appendix 2
- 4) **results1.m** is identical to program in Appendix 2
- 5) **results2.m** is identical to program in Appendix 2
- 6) **difference.m** is identical to program in Appendix 2
- 7) **data.m** is identical to program in Appendix 2

The second algorithm is composed from:

- 1) main.m = is the main program that calls other programs;
- 2) calculation.m = is the program that reads network's data and estimates voltages where measurements are not available;
- 3) readdata2.m = is the program that reads network's data and calculates load flow;
- 4) resultscomplet.m = is the program that writes estimated voltage results in a file "txt";
- 5) results2complet.m = is the program that writes load flow results in a file "txt";
- 6) difference.m = is the program that calculates the difference between calculated voltage and true voltage;
- 7) data.m = is a input file data.

Inside the programs the current has been called with letter F and not with letter I.

1) main.m

```
clear all
clc
```

```
FileDati=input('filename data (.m) = ','s') % File input data
FileOut1=['load flow without some measurements','.txt'];
FileOut2=['load flow','.txt']; % File output data
FileOut3=['difference between true voltage and calculated voltage','.txt'];
eval(FileDati)
fprintf(1,'Calculation load flow')
```

```
calculation
```

```
fid=1;
resultscomplet
fid=fopen(FileOut1,'wt');
resultscomplet;
fclose(fid);
```

```
readdata2
```

```
fid=1;
results2complet
fid=fopen(FileOut2,'wt');
results2complet;
fclose(fid);
```

```
fid=1;
difference
fid=fopen(FileOut3,'wt');
difference;
fclose(fid);
```

```
for g=1:(Nnodes-1)
    n(1,g)=N(1,g+1);
end
```

```
figure(1)
plot(N,abs(V),'b-*');
xlabel('Nodes')
ylabel('Nodes Voltage')
```

```

grid on
hold on
plot(N,abs(V2),'r:*');
hold off

```

```

figure(2)
plot(N,P,'b-*');
xlabel('Nodes')
ylabel('Nodes Active Power')
grid on
hold on
plot(n,Pload,'r:*');
hold off

```

```

figure(3)
plot(N,Q,'b-*');
xlabel('Nodes')
ylabel('Nodes Reactive Power')
grid on
hold on
plot(n,Qload,'r:*');
hold off

```

2) calculation.m

```
% Data Acquisition Lines and Loads
```

```

% Branches
Nbranches=size(BRANCHES,1); % Numbers of Branches
Nnodes=size(LOADS,1); % Numbers of Loads
for kk=1:Nbranches; % Read data
    N1=BRANCHES(1:kk,1); % Sending-end of branch
    N2=BRANCHES(1:kk,2); % Recieving-end of brunch
    Dist=BRANCHES(1:kk,4); % Branch lenght
    r=BRANCHES(1:kk,5); % Branch resistance
    xl=BRANCHES(1:kk,6); % Branch inductance
    xc=BRANCHES(1:kk,7); % Branch capacity
    Condsec=BRANCHES(1:kk,8); % Conductors sections
end

```

```

% Loads
for kk=1:Nnodes
    N=LOADS(1:kk,1); % Nodes Loads
    P=LOADS(1:kk,2); % Real Power
    Q=LOADS(1:kk,3); % Reactive Power
    V=LOADS(1:kk,4); % Nodes Voltage
end

```

```
% Calculating the transpose of vectors for row vectors
```

```

N1=transpose(N1);
N2=transpose(N2);
Dist=transpose(Dist);
r=transpose(r);

```

```

R=(r).*(Dist);
xl=transpose(xl);
X=(xl).*(Dist);
xc=transpose(xc);
N=transpose(N);
P=transpose(P);
Q=transpose(Q);
V=transpose(V);

% Mathematical Formulation

T=0;
while T==0

while T==0
Vinv=V(1,Nnodes:-1:1);
[T,I] = min(Vinv); % nodo where I don't know the measures

k=Nnodes+1-I;
kfirst=k;
V(1,k)=1;

Vinv=V(1,Nnodes:-1:1);
[T,I] = min(Vinv); % nodo where I don't know the measures
k=Nnodes+1-I;
ksecond=k;

if ksecond==kfirst-1
    adj=2;
else
    adj=1;
end

if adj==1;
    k=kfirst;
    s=Nnodes;
    t=kfirst+adj-1;

% Found node calculating the nodes voltage
    Vold=V; % old nodes voltage
    Vc=conj(V);
    if t<(s-1)
        Fb(1,k)=((P(1,k+1)-(Q(1,k+1)*j))/(sqrt(3)*(Vc(1,k+1))))+(V(1,k+1)-
V(1,k+2))/(sqrt(3)*((R(1,k+1)+(X(1,k+1)*j)))); % current flowing through the branch after node with
unknown voltage
        deltaVafter=(R(1,k)+(X(1,k)*j))*Fb(1,k)*sqrt(3); % drop voltage after node with unknown
voltage
        Vunknown=V(1,k+1)+deltaVafter; % unknown voltage
        V(1,k)=Vunknown; % unknown voltage array component
        Vall=V; % nodes voltage array
        moduleV=abs(Vall);
        argV=angle(Vall);
    else
        Fb(1,k)=((P(1,k+1)-(Q(1,k+1)*j))/(sqrt(3)*(Vc(1,k+1)))); % current flowing through the branch after
node with unknown voltage

```



```

    deltaVafter=(R(1,k)+(X(1,k)*j))*Fb(1,k)*sqrt(3);          % drop voltage after node with unknown
voltage
    Vunknown=V(1,k+adj)+deltaVafter;          % unknown voltage
    V(1,k)=Vunknown;          % unknown voltage array component
    Vall=V;          % nodes voltage array
    argV=angle(Vall);
end
Fload(1,k-1)=(P(1,k-1)-(Q(1,k-1)*j))/(sqrt(3)*conj(V(1,k-1)));
Fload(1,k+adj)=(P(1,k+adj)-(Q(1,k+adj)*j))/(sqrt(3)*conj(V(1,k+adj)))
t=kfirst
for k=t:(t+adj-1)
    Fload(1,k)=(Fload(1,k-1))/2+(Fload(1,k+adj))/2)/2;
end
H=0
cont=0
while H==0 && cont<100
    cont=cont+1
    Fb(1,k-1)=Fb(1,k)+Fload(1,k);
    V(1,k-1)=V(1,k)+(R(1,k-1)+(X(1,k-1)*j))*sqrt(3)*Fb(1,k-1)
    DIFF=abs(Vold(1,k-1)-V(1,k-1))
    Vold(1,k-1)
    V(1,k-1)
    Fold(1,k)=Fload(1,k)
    if abs(Vold(1,k-1)-V(1,k-1))<0.01
        break
    elseif V(1,k-1)>Vold(1,k-1)
        Fload(1,k)=Fold(1,k)*0.98
        H=0;
    else
        Fload(1,k)=Fold(1,k)*1.02
        H=0;
    end
end
end
V(1,k-1)=Vold(1,k-1);
S(1,k)=V(1,k)*sqrt(3)*conj(Fload(1,k));
P(1,k)=real(S(1,k));
Q(1,k)=imag(S(1,k));
else
    k=ksecond;
    V(1,k)=1;
    Vinv=V(1,Nnodes:-1:1);
    [T,I] = min(Vinv);
    k=Nnodes+1-I;
    kthird=k;
end
if adj==1
    break

elseif kthird==ksecond-1
    adj=3;

else adj=2
end
if adj==1
    break

```

```

elseif adj==2
    k=ksecond;
    s=Nnodes;
    t=ksecond+adj-1;
    Vold=V; % old nodes voltage
    Vc=conj(V);
    if t<(s-1)

        Fb(1,k+1)=((P(1,k+adj)-(Q(1,k+adj)*j))/(sqrt(3)*(Vc(1,k+adj))))+(V(1,k+adj)-
V(1,k+adj+1))/(sqrt(3)*((R(1,k+adj)+(X(1,k+adj)*j)))); % current flowing through the branch after node
with unknown voltage
        deltaVafter=(R(1,k+adj-1)+(X(1,k+adj-1)*j))*Fb(1,k+1)*sqrt(3); % drop voltage after node
with unknown voltage
        Vunknown=V(1,k+adj)+deltaVafter; % unknown voltage
        V(1,k+1)=Vunknown; % unknown voltage array component
        Vall=V; % nodes voltage array
        moduleV=abs(Vall);
        argV=angle(Vall);

    else

        Fb(1,k+1)=((P(1,k+adj)-(Q(1,k+adj)*j))/(sqrt(3)*(Vc(1,k+adj)))) % current flowing through the
branch after node with unknown voltage
        deltaVafter=(R(1,k+adj-1)+(X(1,k+adj-1)*j))*Fb(1,k+1)*sqrt(3) % drop voltage after node
with unknown voltage
        Vunknown=V(1,k+adj)+deltaVafter; % unknown voltage
        V(1,k+1)=Vunknown % unknown voltage array component
        Vall=V; % nodes voltage array
        argV=angle(Vall);

    end

    Fload(1,k-1)=(P(1,k-1)-(Q(1,k-1)*j))/(sqrt(3)*conj(V(1,k-1)));
    Fload(1,k+adj)=(P(1,k+adj)-(Q(1,k+adj)*j))/(sqrt(3)*conj(V(1,k+adj)));
    t=ksecond
    for k=t:(t+adj-1)
        Fload(1,k)=((Fload(1,k-1))/2+(Fload(1,t+adj))/2)/2;
    end
    H=0
    cont=0
    while H==0 && cont<100
        cont=cont+1
        k=ksecond;
        Fb(1,k)=Fb(1,k+1)+Fload(1,k+1);
        V(1,k)=V(1,k+1)+(R(1,k)+(X(1,k)*j))*sqrt(3)*Fb(1,k);
        Fb(1,k-1)=Fb(1,k)+Fload(1,k);
        V(1,k-1)=V(1,k)+(R(1,k-1)+(X(1,k-1)*j))*sqrt(3)*Fb(1,k-1);
        DIFF=abs(Vold(1,k-1)-V(1,k-1))
        Vold(1,k-1)
        V(1,k-1)
        Fold(1,k)=Fload(1,k)
        if abs(Vold(1,k-1)-V(1,k-1))<0.01
            break
        elseif V(1,k-1)>Vold(1,k-1)
            Fload(1,k)=Fold(1,k)*0.98;

```

```

        Fload(1,k+1)=Fload(1,k);
        H=0;
    else
        Fload(1,k)=Fold(1,k)*1.02;
        Fload(1,k+1)=Fload(1,k);
        H=0;
    end
end
V(1,k-1)=Vold(1,k-1);
for k=t:(t+adj-1)
    S(1,k)=V(1,k)*sqrt(3)*conj(Fload(1,k));
    P(1,k)=real(S(1,k));
    Q(1,k)=imag(S(1,k));
end

else

    k=kthird;
    V(1,k)=1;
    Vinv=V(1,Nnodes:-1:1);
    [T,I] = min(Vinv);
    k=Nnodes+1-I;
    kfourth=k;
end

if adj==1 || adj==2
    break

elseif kfourth==kthird-1
    adj=4;

else adj=3
end
if adj==1 || adj==2
    break
elseif adj==3
    k=kthird
    s=Nnodes
    t=kthird+adj-1
    Vold=V % old nodes voltage
    Vc=conj(V)
    if t<(s-1)

        Fb(1,k+2)=((P(1,k+adj)-(Q(1,k+adj)*j))/(sqrt(3)*(Vc(1,k+adj))))+(V(1,k+adj)-
V(1,k+adj+1))/(sqrt(3)*((R(1,k+adj)+(X(1,k+adj)*j)))); % current flowing through the branch after node
with unknown voltage
        deltaVafter=(R(1,k+adj-1)+(X(1,k+adj-1)*j))*Fb(1,k+2)*sqrt(3); % drop voltage after
node with unknown voltage
        Vunknown=V(1,k+adj)+deltaVafter; % unknown voltage
        V(1,k+2)=Vunknown; % unknown voltage array component
        Vall=V; % nodes voltage array
        moduleV=abs(Vall);
        argV=angle(Vall);

    else

```

```

        Fb(1,k+2)=(P(1,k+adj)-(Q(1,k+adj)*j))/(sqrt(3)*(Vc(1,k+adj))) % current flowing through
the branch after node with unknown voltage
        deltaVafter=(R(1,k+adj-1)+(X(1,k+adj-1)*j))*Fb(1,k+2)*sqrt(3) % drop voltage after
node with unknown voltage
        Vunknown=V(1,k+adj)+deltaVafter; % unknown voltage
        V(1,k+2)=Vunknown % unknown voltage array component
        Vall=V; % nodes voltage array
        argV=angle(Vall);

end

Fload(1,k-1)=(P(1,k-1)-(Q(1,k-1)*j))/(sqrt(3)*conj(V(1,k-1)));
Fload(1,k+adj)=(P(1,k+adj)-(Q(1,k+adj)*j))/(sqrt(3)*conj(V(1,k+adj)));
t=kthird
for k=t:(t+adj-1)
    Fload(1,k)=(Fload(1,k-1))/2+(Fload(1,t+adj))/2;
end
H=0
cont=0
while H==0 && cont<100
    cont=cont+1
    k=kthird;
    Fb(1,k+1)=Fb(1,k+2)+Fload(1,k+2);
    V(1,k+1)=V(1,k+2)+(R(1,k+1)+(X(1,k+1)*j))*sqrt(3)*Fb(1,k+1);
    Fb(1,k)=Fb(1,k+1)+Fload(1,k+1);
    V(1,k)=V(1,k+1)+(R(1,k)+(X(1,k)*j))*sqrt(3)*Fb(1,k);
    Fb(1,k-1)=Fb(1,k)+Fload(1,k);
    V(1,k-1)=V(1,k)+(R(1,k-1)+(X(1,k-1)*j))*sqrt(3)*Fb(1,k-1);
    DIFF=abs(Vold(1,k-1)-V(1,k-1))
    Vold(1,k-1)
    V(1,k-1)
    Fold(1,k)=Fload(1,k)
    if abs(Vold(1,k-1)-V(1,k-1))<0.01
        break
    elseif V(1,k-1)>Vold(1,k-1)
        Fload(1,k)=Fold(1,k)*0.98;
        Fload(1,k+1)=Fload(1,k);
        Fload(1,k+2)=Fload(1,k);
        H=0;
    else
        Fload(1,k)=Fold(1,k)*1.02;
        Fload(1,k+1)=Fload(1,k);
        Fload(1,k+2)=Fload(1,k);
        H=0;
    end
end
end
V(1,k-1)=Vold(1,k-1);
for k=t:(t+adj-1)
    S(1,k)=V(1,k)*sqrt(3)*conj(Fload(1,k));
    P(1,k)=real(S(1,k));
    Q(1,k)=imag(S(1,k));
end
end
else

```

```

        fprintf('Error: Too adjacent S.M are not working');
    end
end
V=V;
Vinv=V(1:Nnodes:-1:1);
[T,I] = min(Vinv);
end

V=V;
Vc=conj(V);
for y=1:Nnodes
Fload(1,k)=(P(1,y)-(Q(1,y)*j))/Vc(1,y)
end

```

3) readdata2.m is identical to program in Appendix 2

4) resultscmplet.m

```
Risp=input('Do you want to work in absolute values [1] or in p.u. [2] ?');
```

```

if Risp==1
    fprintf(fid,'\n          NODES VOLTAGE (absolute values)\n');
    fprintf(fid,' Node          Magnitude Voltage [V]          Angle Voltage [°]\n');
    for i=1:Nnodes
        Node=i;
        ModVolt=abs(V(1,i));
        ArgVolt=angle(V(1,i))*180/pi;
        fprintf(fid,'%5.0i      %18.3f      %18.2f\n',Node , ModVolt, ArgVolt);
    end
    fprintf(fid,'\n          LOADS CURRENT (absolute values)\n');
    fprintf(fid,' Node          Magnitude Current [A]          Angle Current [°]\n');
    for i=2:Nnodes
        Node=i;
        ModFload=abs(Fload(1,i));
        ArgFload=angle(Fload(1,i))*180/pi;
        fprintf(fid,'%5.0i      %18.3f      %18.2f\n',Node , ModFload, ArgFload);
    end
    fprintf(fid,'\n          NODES POWER (absolute values)\n');
    fprintf(fid,' Node          Active Power [W]          Reactive Power [var]\n');
    for i=2:Nnodes
        Node=i;
        ActivePower=P(1,i);
        ReactivePower=Q(1,i);
        fprintf(fid,'%5.0i      %18.3f      %18.2f\n',Node , ActivePower, ReactivePower);
    end
else
    fprintf(fid,'\n          NODES VOLTAGE (p.u.)\n');
    fprintf(fid,' Node          Magnitude Voltage [p.u.]          Angle Voltage [°]\n');
    Vbase=LOADS(1,4); % Base Voltage
    Nbase=sqrt(((LOADS(2,2))^2)+((LOADS(2,3))^2)); % Base Power
    Fbase=Nbase/(Vbase*sqrt(3)); % Base current
    V=V./Vbase;
    for i=1:Nnodes
        Node=i;

```

```

    ModVolt=abs(V(1,i));
    ArgVolt=angle(V(1,i))*180/pi;
    fprintf(fid,'%5.0i      % 18.3f      % 18.2f\n',Node , ModVolt, ArgVolt);
end
Fload=Fload./Fbase;
fprintf(fid,'\n          LOADS CURRENT (p.u.)\n');
fprintf(fid,' Node          Magnitude Current [p.u.]          Angle Current [°]\n');
for i=1:Nbranches
    Node=i;
    ModFload=abs(Fload(1,i));
    ArgFload=angle(Fload(1,i))*180/pi;
    fprintf(fid,'%5.0i      % 18.3f      % 18.2f\n',Node , ModFload, ArgFload);
end
fprintf(fid,'\n          NODES POWER (p.u.)\n');
fprintf(fid,' Node          Active Power [p.u.]          Reactive Power [p.u.]\n');
P=P./Nbase;
Q=Q./Nbase;
for i=1:Nbranches
    Node=i;
    ActivePower=P(1,i);
    ReactivePower=Q(1,i);
    fprintf(fid,'%5.0i      % 18.3f      % 18.2f\n',Node , ActivePower, ReactivePower);
end
end

```

5) results2complet.m

Risp=input('Do you want to work in absolute values [1] or in p.u. [2] ?');

```

if Risp==1
    fprintf(fid,'\n          NODES VOLTAGE (absolute values)\n');
    fprintf(fid,' Node          Magnitude Voltage [V]          Angle Voltage [°]\n');
    for i=1:Nnodes
        Node=i;
        ModVolt=abs(V2(1,i));
        ArgVolt=angle(V2(1,i))*180/pi;
        fprintf(fid,'%5.0i      % 18.3f      % 18.2f\n',Node , ModVolt, ArgVolt);
    end

    fprintf(fid,'\n          BRANCHES CURRENT (absolute values)\n');
    fprintf(fid,' Branch          Magnitude Current [A]          Angle Current [°]\n');
    for i=1:Nbranches
        ModFbranch=abs(Fi2(1,i));
        ArgFbranch=angle(Fi2(1,i))*180/pi;
        fprintf(fid,'%5.0i      % 18.3f      % 18.2f\n',i , ModFbranch, ArgFbranch);
    end

    fprintf(fid,'\n          LOADS CURRENT (absolute values)\n');
    fprintf(fid,' Node          Magnitude Current [A]          Angle Current [°]\n');
    for i=1:Nbranches
        Node=i+1;
        ModFload=abs(Fload(1,i));
        ArgFload=angle(Fload(1,i))*180/pi;
        fprintf(fid,'%5.0i      % 18.3f      % 18.2f\n',Node , ModFload, ArgFload);
    end

```

```

end

fprintf(fid,'\n                LOADS POWER (absolute values)\n');
fprintf(fid,' Node          Active Power [W]          Reactive Power [var]\n');
for i=1:Nbranches
    Node=i+1;
    ActivePower=Pload(1,i);
    ReactivePower=Qload(1,i);
    fprintf(fid,'%5.0i      %18.3f      %18.2f\n',Node , ActivePower, ReactivePower);
end
else
Vbase=LOADS(1,4);                % Base Voltage
Nbase=sqrt(((LOADS(2,2))^2)+((LOADS(2,3))^2)); % Base Power
Fbase=Nbase/(Vbase*sqrt(3));      % Base current
fprintf(fid,'\n                NODES VOLTAGE (p.u.)\n');
fprintf(fid,' Node          Magnitude Voltage [p.u.]      Angle Voltage [°]\n');
V2=V2./Vbase;
for i=1:Nnodes
    Node=i;
    ModVolt=abs(V2(1,i));
    ArgVolt=angle(V2(1,i))*180/pi;
    fprintf(fid,'%5.0i      %18.3f      %18.2f\n',Node , ModVolt, ArgVolt);
end

fprintf(fid,'\n                BRANCHES CURRENT (p.u.)\n');
fprintf(fid,' Branch        Magnitude Current [p.u.]      Angle Current [°]\n');
Fi2=Fi2./Fbase;
for i=1:Nbranches
    ModFbranch=abs(Fi2(1,i));
    ArgFbranch=angle(Fi2(1,i))*180/pi;
    fprintf(fid,'%5.0i      %18.3f      %18.2f\n',i , ModFbranch, ArgFbranch);
end

fprintf(fid,'\n                LOADS CURRENT (p.u.)\n');
fprintf(fid,' Node          Magnitude Current [p.u.]      Angle Current [°]\n');
Fload=Fload./Fbase;
for i=1:Nbranches
    Node=i+1;
    ModFload=abs(Fload(1,i));
    ArgFload=angle(Fload(1,i))*180/pi;
    fprintf(fid,'%5.0i      %18.3f      %18.2f\n',Node , ModFload, ArgFload);
end

fprintf(fid,'\n                LOADS POWER (p.u.)\n');
fprintf(fid,' Node          Active Power [p.u.]      Reactive Power [p.u.]\n');
Pload=Pload./Nbase;
Qload=Qload./Nbase;
for i=1:Nbranches
    Node=i+1;
    ActivePower=Pload(1,i);
    ReactivePower=Qload(1,i);
    fprintf(fid,'%5.0i      %18.3f      %18.2f\n',Node , ActivePower, ReactivePower);
end
end
end

```

- 6) **difference.m** is identical to program in Appendix 2
- 7) **data.m** is identical to program in Appendix 2