

**UNIVERSITE D'ANTANANARIVO
ECOLE SUPERIEURE POLYTECHNIQUE**



**DEPARTEMENTS : GENIE MECANIQUE ET PRODUCTIQUE
GENIE ELECTRIQUE**

**Mémoire de fin d'études en vue de l'obtention
du diplôme de Licence ès Sciences techniques
Filière Génie Industriel**

N° d'ordre :/2010

**ELABORATION D'UN PROGRAMME POUR DETERMINER
LA NATURE ET LA SECTION DES CONDUCTEURS DANS
UN RESEAU ELECTRIQUE ARBORESCENT POUR
DIMINUER LA CHUTE DE TENSION MAXIMALE**

Présenté par : Monsieur RASAMIARINDRAINY Mika Ambinintsoa

Encadreur : Monsieur RAVELOMANANA Nathanaël

Année Universitaire 2009-2010

Date de soutenance : 12 Janvier 2011

UNIVERSITE D'ANTANANARIVO
ECOLE SUPERIEURE POLYTECHNIQUE



DEPARTEMENTS : GENIE MECANIQUE ET PRODUCTIQUE
GENIE ELECTRIQUE

Mémoire de fin d'études en vue de l'obtention
du diplôme de Licence ès Sciences techniques
Filière Génie Industriel

N° d'ordre :/2010

**ELABORATION D'UN PROGRAMME POUR DETERMINER
LA NATURE ET LA SECTION DES CONDUCTEURS DANS
UN RESEAU ELECTRIQUE ARBORESCENT POUR
DIMINUER LA CHUTE DE TENSION MAXIMALE**

Présenté par : Monsieur RASAMIARINDRAINY Mika Ambinintsoa

Président : Monsieur RAKOTONIAINA Solofo Hery, Enseignant Chercheur à l'ESPA

Rapporteur : Monsieur RAVELOMANANA Nathanaël, Enseignant Chercheur à l'ESPA

Examineurs : Monsieur RAVALOMANANA Olivier, Enseignant Chercheur à l'ESPA
Monsieur RAVOMANANA Jocelyn, Chef de Département Planification et
Développement des réseaux de distribution à la JIRAMA
Monsieur RANAIVOMAHASETRA, Enseignant à l'IST

Année Universitaire 2009-2010

Date de soutenance : 12 Janvier 2011



REMERCIEMENTS

Avant tout, nous aimerons de tout cœur rendre grâce à Dieu, Notre Seigneur, pour toutes ses bénédictions envers nous et ses réponses à nos prières pour l'accomplissement de notre étude.

Le présent mémoire couronne de laborieuses années d'études. Il n'aura pu être accompli sans l'enseignement et la contribution de diverses personnes auxquelles nous voudrions adresser nos remerciements les plus sincères, en particulier :

- Monsieur ANDRIANARY Philippe, Directeur de l'Ecole Supérieure Polytechnique d'Antananarivo, de nous avoir accueilli au sein de son établissement et de permettre la soutenance de ce travail,
- Monsieur RAKOTONIAINA Solofo Hery, Chef du Département Génie Electrique,
- Monsieur RAKOTOMANANA Rodin, Chef du Département Génie Mécanique,
- Monsieur RAVELOMANANA Nathanaël, Encadreur pédagogique du mémoire de fin d'études,
- Tous les membres du jury qui n'ont pas ménagé leur temps pour venir assister ce jour à la soutenance de cet ouvrage,
- A l'ensemble du corps enseignant ainsi que les membres professionnels de l'E.S.P.A. qui nous ont dispensé leur savoir et transmis leur connaissance avec compétence et générosité,
- Mes parents et ma famille qui nous ont soutenus moralement et financièrement et qui nous ont témoigné constamment leur confiance,
- Nos collègues et nos amis.
- Et enfin, tous ceux qui, de près ou de loin, ont prodigué le meilleur d'eux-mêmes pour la réalisation de ce mémoire.

Qu'ils veuillent bien trouver ici l'expression de notre haute considération et le témoignage de notre profonde gratitude.



LISTE DES TABLEAUX

Tableau 1: Classification des réseaux selon la tension nominale	6
Tableau 2: Les différentes fonctions des appareils MT	15
Tableau 3: Liste des contrôles usuels.....	24
Tableau 4: Liste des propriétés	25
Tableau 5: Liste des structures.....	29
Tableau 6: Listes des boucles.....	30
Tableau 7: Liste des procédures dans le code de travail Form1	46
Tableau 8: Listes des procédures générales dans le module1	47
Tableau 9 : Informations sur les nœuds	65
Tableau 10 : Informations sur les lignes	65



LISTE DES FIGURES

Figure 1 : Schéma d'un réseau ramifié simple.....	33
Figure 2 : Schéma équivalent à deux lignes.....	36
Figure 3 : Ligne équivalente à un groupe	36
Figure 4 : Exemple de transformation directe d'un réseau	37
Figure 5 : La barre d'outils	41
Figure 6 : Le menu	41
Figure 7 : Partie de l'interface pour les nœuds	43
Figure 8 : Partie de l'interface pour les lignes	44
Figure 9: Partie de l'interface pour les paramètres par défaut	45
Figure 10 : Structure du projet	46
Figure 11: Organigramme de « CmdValider_Click ».....	50
Figure 12: La procédure « DéfinirNdAmont »	52
Figure 13 : La procédure« RenumeroterNd »	55
Figure 14 : La procédure « DéfinirGroupeEtNiveau »	57
Figure 15 : La procédure « TransformationDirecte »	59
Figure 16: La procédure « TransfoGroupeEn1Ligne.....	60
Figure 17 : la procédure « « TransformationInverse ».....	63
Figure 18 : Réseau ramifié simple	65
Figure 19 : L'interface graphique pour l'exemple de calcul.....	68



LISTE DES ACRONYMES

AGP : Advanced Graphic Port

ASP : Active Server Pages

BT : Basse Tension

BTA : Basse Tension A

BTB : Basse Tension B

CPU : Central Processor Unit

DVD : Digital Versatil Disc

DVI : Digital Visual Interface

HT : Haute Tension

HTA : Haute Tension A

HTB : Haute Tension B

HVDC: High Voltage Direct Current

IDE: Integrated development Environment

MSDN : Microsoft Developer Network

MT: Moyenne Tension

NT : New Technology

OLE: Object Linking and Embedding

PCI: Peripheral Component Interconnect

PS/2 : Peersonal System/2

RAM: Random Access Memory

SATA: Serial Advanced Technology Attachment

TBT: Très Basse Tension

TV : Télévision

USB: Universal serial Bus

VB: Visual Basic

VBA: Visual Basic Application



VGA: Video graphics Array

XML: Extensible Markup Language

XP: Expérience



SOMMAIRE

INTRODUCTION	1
CHAPITRE I: GENERALITES SUR LES RESEAUX ELECTRIQUES	1
I.1. Historique	2
I.2. La classification des réseaux	3
I.3. Notions importantes sur les réseaux électriques.....	6
CHAPITRE II : LES PRINCIPAUX COMPOSANTS DES RESEAUX ELECTRIQUES	8
II.1. Les unités de production.....	8
II.2. Les ouvrages ou postes.....	12
II.3. Les matériels.....	13
CHAPITRE III : PROTECTIONS DES RESEAUX ELECTRIQUES	16
III.1. Les technologies de protection	16
III.2. Les matériels de conduite et de surveillance utilisés.....	17
CHAPITRE IV : CONSTITUTION D'UN ORDINATEUR	18
IV.1. La partie physique de l'ordinateur.....	18
IV.2. Partie logicielle de l'ordinateur	20
CHAPITRE V : LE LANGAGE DE PROGRAMMATION VISUAL BASIC 6.0	23
V.1. Les contrôles.....	23
V.2. Les variables.....	26
V.3. Les procédures.....	27
V.4. Les instructions.....	28
CHAPITRE VI : CALCUL DE LA SECTION D'UN CONDUCTEUR.....	32
VI.1. Calcul des sections des conducteurs d'une ligne.....	32
VI.2. Calcul des sections des conducteurs pour un réseau ramifié simple.....	33
VI.3. Calcul des sections des conducteurs dans un réseau arborescent complexe	35
CHAPITRE VII : LES DIFFERENTES ETAPES D'UNE PROGRAMMATION	39
VII.1. Spécifications des exigences et besoins des futurs utilisateurs	39
VII.2. Conception générale du programme.....	39
VII.3. Conception détaillée du programme.....	39
VII.4. Les diverses vérifications	40
CHAPITRE VIII : PRESENTATION GENERALE DU PROGRAMME.....	41
VIII.1. La description de l'interface utilisateur.....	41
VIII.2. Le fonctionnement général du programme	45
CHAPITRE IX : LES PROCEDURES GENERALES DU PROGRAMME.....	51
IX.1. Les procédures générales liées à l'organisation du réseau	51
IX.2. Les procédures utilisées pour la transformation directe	58
IX.3. La procédure « TransformationInverse »	62
CHAPITRE X : ANALYSES DES RESULTATS	65
X.1. Test de vérification du programme	65
X.2. Les impacts environnementaux	68
CONCLUSION.....	71



INTRODUCTION

L'électrification est un indicateur de développement d'un pays. En effet, pour les pays industrialisés, le taux d'électrification est très élevé tandis que pour les pays en voie de développement ce taux est très bas.

Ainsi, à Madagascar, le taux de communes électrifiées n'est que de 19.66%. L'électrification nécessite l'installation d'un réseau électrique. Ils transportent l'énergie électrique utilisée par les entreprises pour produire et les ménages dans leur vie quotidienne. Un des problèmes dans la mise en place de l'électrification est la nécessité d'un grand investissement dans l'installation.

A Madagascar, non seulement l'installation est chère mais il y a le problème de la chute de tension au bout des lignes entraînant les mécontentements des utilisateurs surtout dans les heures de pointe.

Des questions se posent : comment pourrait-on réduire ces coûts d'installation et par la même occasion comment donner satisfaction aux utilisateurs en leur fournissant une tension plus stable. Les conducteurs électriques constituent une importante partie de l'installation alors il s'avère nécessaire de bien choisir cet élément notamment ses sections.

C'est pourquoi, l'intitulé de ce mémoire est l'élaboration d'un programme pour déterminer la nature et la section des conducteurs dans un réseau arborescent pour diminuer la chute de tension maximale. Il pourra servir les entreprises dans leur calcul des sections des conducteurs pour la conception des réseaux électriques mais aussi, dans un but pédagogique, les étudiants de l'école supérieure polytechnique d'Antananarivo pour leurs connaissances en installation et distribution électrique.

Ce mémoire a pour but de faciliter le calcul et le choix des sections des conducteurs. Pour atteindre cet objectif global, il faut faire d'une part une réorganisation et une transformation du réseau en réseau équivalent simple et d'autre part, à partir de cette transformation, déterminer les sections des conducteurs.

La première partie sera consacrée au contexte général de l'étude. La deuxième parlera des matériels et des méthodes et dans la troisième partie, les applications et les résultats.

PARTIE 1 : CONTEXTE GENERAL DE L'ETUDE



CHAPITRE I: GENERALITES SUR LES RESEAUX ELECTRIQUES

Le réseau électrique est l'ensemble des moyens qui servent à transmettre l'énergie électrique d'une région où on la produit à une autre où on la consomme.

Pour comprendre les réseaux électriques, il faut connaître l'histoire, le mode de classification et les notions importantes à retenir.

I.1. Historique

Un réseau électrique étant composé de machines de production et de consommation, ainsi que de structures (lignes, transformateurs) pour les relier, les réseaux électriques ne sont apparus que vers la fin XIX^e siècle, lorsque chaque élément avait atteint une maturité technologique suffisante.

Lors de la première moitié du XIX^e siècle, les inventeurs mettent au point de nombreux types de moteurs électriques à courant continu, mais leur utilisation de manière industrielle ne sera permise qu'après l'invention de la dynamo (génératrice de courant continu) par Zénobe Gramme en 1869. À l'automne 1882, les premiers réseaux électriques apparaissent simultanément à New York et Bellegarde, en France. Vers la fin de ce siècle, Thomas Edison a fondé l'Edison Electric Light Co (qui deviendra en 1892 General Electric) et a créé le réseau électrique de New York. Ce réseau, qui souffre de nombreuses pannes, est constitué de petites centrales électriques (30 kW) et d'un réseau de distribution à 110 V. Il est cependant très limité car l'acheminement de l'électricité n'est possible que sur quelques kilomètres.

Un ingénieur et entrepreneur américain, George Westinghouse, avec l'aide de Tesla et avec le développement de la technologie du transformateur a développé le réseau électrique avec le courant alternatif. Une opposition sévère fait rage à cette époque aux États-Unis entre Edison (défenseur du courant continu) et George Westinghouse avec Tesla (défenseur du courant alternatif). La bataille décisive entre « courant continu » et « courant alternatif » se déroule autour d'un projet d'alimentation électrique de l'industrie de Buffalo par une centrale hydraulique de 75 MW située à Niagara Falls, à 32 km de distance. Edison proposait un projet en courant continu tandis que Tesla et Westinghouse proposaient un système en courant alternatif. Le contrat fut donné à Westinghouse. En 1896, la mise en service de la première ligne industrielle en triphasé fut un succès total et conduit pour un siècle au moins à imposer universellement le courant alternatif triphasé comme moyen de transport de l'énergie électrique, mieux adapté à cette époque au transport sur de longues distances.



Au début du XX^e siècle, les usages de l'électricité se multiplient, aussi bien au niveau domestique qu'industriel (notamment l'électrification des tramways, métros et chemins de fer). Les compagnies d'électricité construisent des centrales électriques et de petits réseaux locaux, chacun utilisant des fréquences et des niveaux de tensions différents. Les opérateurs se rendent compte tardivement de l'intérêt d'utiliser une fréquence unique (indispensable à l'interconnexion des réseaux), et l'on voit apparaître finalement deux standards de fréquence : le 60 Hz sur la majorité du continent américain et le 50 Hz quasiment partout dans le reste du monde.

Dans la première moitié du XX^e siècle les réseaux urbains des pays industrialisés se sont agrandis afin d'électrifier les campagnes. En parallèle, ces réseaux se sont interconnectés entre eux au niveau régional afin d'engranger des économies d'échelle sur la taille des centrales de production, et de mieux valoriser des ressources énergétiques géographiquement localisées, comme la production hydraulique située dans les zones montagneuses, éloignée des grands centres de consommation. Au fur et à mesure de l'augmentation des puissances appelées et des distances des lignes d'interconnexion, la tension d'exploitation des lignes a également augmentée (1^{re} ligne à 220 kV construite en 1923 aux États-Unis, celle à 380 kV en 1930 en Allemagne).

La deuxième moitié du XX^e siècle a connu en outre un renforcement des interconnexions intranationales et un développement significatif des interconnexions transnationales, dans le but principal de créer des capacités de secours mutuel entre opérateurs et d'améliorer globalement la stabilité des systèmes électriques, ainsi que, de façon plus ponctuelle, de créer des capacités d'échange d'énergie sur le long terme.

Les réseaux sont devenus, à l'heure actuelle, très complexes. Il est donc nécessaire de les classer.

I.2. La classification des réseaux

Il existe plusieurs facteurs qui permettent de classer les réseaux électriques. Ainsi, les réseaux peuvent être classifiés selon :

- Selon la fonction
- Selon le type de courant
- Selon l'architecture du réseau
- Selon la tension nominale



I.2.1. Classification selon la fonction

On peut subdiviser les réseaux en trois catégories :

- Les réseaux de transport
- Les réseaux de répartition
- Les réseaux de distribution

I.2.1.1. Les réseaux de transport

La finalité de ce réseau est triple :

- ✓ une fonction de “transport” dont le but est d’acheminer l’électricité des centrales de production aux grandes zones de consommation ;
- ✓ une fonction “d’interconnexion nationale” qui gère la répartition de l’offre en orientant la production en fonction de la répartition géographique et temporelle de la demande ;
- ✓ une fonction “d’interconnexion internationale” pour gérer des flux d’énergie entre les pays en fonction d’échanges programmés ou à titre de secours.

Les tensions sont généralement comprises entre 225 et 400 kV.

I.2.1.2. Les réseaux de répartition

La finalité de ce réseau est avant tout d’acheminer l’électricité du réseau de transport vers les grands centres de consommation. Ces centres de consommation sont :

- ✓ soit du domaine public avec l’accès au réseau de distribution MT,
- ✓ soit du domaine privé avec l’accès aux abonnés à grande consommation.

Les tensions sur ces réseaux sont comprises entre 25 et 275 kV.

I.2.1.3. Les réseaux de distribution

La finalité de ce réseau est d’acheminer l’électricité du réseau de répartition aux points de moyenne consommation. Ces points de consommation sont :

- ✓ soit du domaine public, avec accès aux postes de distribution publique MT/BT,
- ✓ soit du domaine privé, avec accès aux postes de livraison aux abonnés à moyenne consommation.

Les tensions sur ces réseaux sont comprises entre quelques kilovolts et 40 kV.



I.2.2. Classification selon le type de courant

Le transfert de l'énergie électrique sur les différents réseaux électriques se fait via le courant électrique. On distingue deux types de courant : le courant continu et le courant alternatif.

- Pour le courant continu : Les liaisons par courant continu ou HVDC (high voltage direct current) sont utilisées pour les échanges entre pays uniquement au niveau des réseaux de transport. Le choix de cette technique permet d'optimiser l'utilisation des câbles d'énergie, en particulier en supprimant les effets de «peau» ;
- Pour le courant alternatif : la plupart des réseaux électriques, en particulier celui de la distribution MT, utilisent le courant alternatif. En effet, sur ces réseaux, l'utilisation du courant continu ne serait pas rentable à cause des pertes réduites sur des réseaux courts et de la cherté des installations en convertisseurs courant continu/ alternatif. De plus, le courant alternatif est très adapté au changement de tension lors du cheminement de l'énergie électrique.

La fréquence de courant utilisé dans le monde, en dehors du continent américain qui est de 60 Hz, est 50Hz.

I.2.3. Classification selon l'architecture du réseau

Ils sont divisés en deux.

- Les réseaux ouverts : on y trouve les réseaux radiaux dont la sécurité d'alimentation, est assez élevée. Les réseaux arborescents dont la sécurité d'alimentation est faible puisqu'un défaut sur la ligne coupe l'ensemble des clients en aval se trouve aussi dans ce groupe.
- Les réseaux fermés : ce sont les réseaux maillés dont les postes électriques sont reliés entre eux par de nombreuses lignes électriques, apportant une grande sécurité d'alimentation. Il y a aussi les réseaux bouclés dont la sécurité d'alimentation est élevée.



I.2.4. Classification selon la tension nominale

Pour une même puissance transmise, le courant décroît avec l'augmentation de la tension nominale ce qui a pour conséquence l'augmentation des sections des conducteurs. Le tableau 1 ci-dessous récapitule la classification selon la tension nominale :

Domaines	Tension nominale	Utilisation
TBT	$U \leq 50 V$	Utilisation spécifique
BTA	$50 V \leq U \leq 500 V$	Distribution d'énergie électrique
BTB	$500 V \leq U \leq 1kV$	Distribution d'énergie électrique
HTA	$1kV \leq U \leq 50 kV$	Transport d'énergie électrique pour une distance moyenne
HTB	$> 50 kV$	Transport d'énergie électrique distant

Tableau 1: Classification des réseaux selon la tension nominale

I.3. Notions importantes sur les réseaux électriques

La connaissance de certaines notions est indispensable pour la compréhension des réseaux électriques à savoir les pertes en lignes par effet Joule et la chute de tension.

I.3.1. Les pertes en lignes par effet Joule

A cause de ces pertes, il est nécessaire de transporter l'énergie électrique en utilisant des tensions élevées. L'utilisation de ces tensions élevées est liée à un objectif économique. En effet, le transport de fortes puissances sur de longues distances permet la minimisation de l'effet Joule. Pour du courant triphasé on obtient :

$$\Delta P = RI^2 = R \frac{P^2}{3U^2} \quad [1]$$

ΔP : Pertes par effet Joule

R : Résistance de la ligne



I : Intensité du courant

P : Puissance transmise par la ligne

U : Tension

Pour une même puissance électrique transmise par la ligne, les pertes par effet Joule diminuent donc comme le carré de la tension. Ainsi un mètre de câble à 400 V provoque autant de pertes que 1000 km du même câble à 400 kV.

I.3.2. La chute de tension

La chute de tension est donnée par la formule suivante :

$$\Delta U = \frac{\rho}{sU} \left(1 + \frac{X'}{R'} \operatorname{tg} \varphi \right) \sum l_i P_i [2]$$

[2]

R' : résistance linéique

X' : réactance linéique

ρ : résistivité du conducteur ; on a $\rho = (0.0225 \Omega \text{ [mm]}^2)/m$ pour le cuivre

et $\rho = (0.036 \Omega \text{ [mm]}^2)/m$ pour l'aluminium

φ : angle de phase

l : longueur de la ligne [m]

P : puissance transmise par les lignes [kW]

s : section des conducteurs [mm²]

U : tension au départ [V]



CHAPITRE II : LES PRINCIPAUX COMPOSANTS DES RESEAUX ELECTRIQUES

Un réseau électrique est un ensemble d'infrastructures permettant d'acheminer l'énergie électrique des centres de production vers les consommateurs d'électricité. Il est constitué de lignes électriques exploitées à différents niveaux de tension, connectées entre elles dans des postes électriques. Les postes électriques permettent de répartir l'électricité et de la faire passer d'une tension à l'autre grâce aux transformateurs. Un réseau électrique doit aussi assurer la gestion dynamique de l'ensemble production - transport - consommation, mettant en œuvre des réglages ayant pour but d'assurer la stabilité de l'ensemble.

Ainsi, un réseau est composé de divers éléments qui sont répartis comme suit :

- ❖ les unités de production
- ❖ les ouvrages
- ❖ les matériels

II.1. Les unités de production

Une centrale électrique est un site industriel destiné à la production d'électricité. Les centrales électriques transforment différentes sources d'énergie naturelle en énergie électrique afin d'alimenter en électricité les consommateurs : particuliers ou industriels.

Le type de turbine définit le type de centrale :

- Une turbine à vapeur dans une centrale à combustible fossile ou nucléaire et une turbine à combustion dans une centrale thermique ;
- Une turbine hydraulique dans une centrale hydroélectrique ou une centrale marémotrice.

L'énergie électrique est obtenue par conversion de l'énergie mécanique produite par une turbine hydraulique ou turbine à vapeur. Cette conversion est obtenue en couplant une dynamo ou un alternateur à la turbine.

Le rendement de conversion mécanique/électrique est d'environ 98%. L'essentiel des pertes se fait donc sur la conversion thermique-mécanique.



II.1.1. Les différents types de centrales

Il existe principalement deux types de centrales :

- Les centrales thermiques
- Les centrales utilisant les énergies renouvelables

II.1.1.1. Les centrales thermiques

Elles sont subdivisées en :

- ✓ Les centrales conventionnelles à chaudière ;
- ✓ Les centrales à turbine à gaz et à cycle forcé ;

a. Les centrales conventionnelles à chaudière

Les centrales les plus répandues sont constituées d'une chaudière et d'une turbine à vapeur. Leur carburant est le plus souvent du charbon mais on trouve aussi des chaudières utilisant de la biomasse, du gaz naturel, du pétrole, du fioul ou des déchets municipaux.

La plupart des centrales à charbon sont de type à « charbon pulvérisé », où le charbon est réduit en poudre très fine dans des broyeurs et injecté dans le foyer de la chaudière. Les centrales les plus récentes possèdent un cycle vapeur supercritique, qui permet d'avoir un rendement qui dépasse 45%.

b. Les centrales à turbine à gaz et à cycle forcé

Les turbines à gaz en cycle simple sont peu coûteuses à construire, de plus elles ont l'avantage de démarrer très rapidement contrairement aux centrales conventionnelles à vapeur qui ont une certaine inertie. Néanmoins, leur rendement faible, qui est de 35% au mieux, empêche de les utiliser directement pour la production d'électricité sans valoriser leur chaleur résiduelle, sauf en appoint lors des pics de demande ou à toute petite échelle.

Le cycle combiné consiste à récupérer l'énergie thermique des gaz très chauds de l'ordre de 600 °C à l'échappement de la turbine à combustion, pour produire dans une chaudière de récupération de la vapeur utilisée pour alimenter un groupe turbo-alternateur à vapeur. Le résultat permet donc une augmentation notable du rendement énergétique de la centrale. Ce type de centrale comprend deux alternateurs, l'un entraîné par la turbine à combustion, l'autre par la turbine à vapeur. Toutefois, certaines centrales n'ont qu'un seul alternateur, les deux turbines étant montées sur le même arbre.



L'utilisation de ces centrales a des impacts négatifs sur l'environnement. De plus, les énergies fossiles s'épuisent au fur à mesure de leur utilisation contrairement aux énergies renouvelables.

II.1.1.2. Les centrales utilisant les énergies renouvelables

Les centrales utilisant les énergies renouvelables sont constituées par :

- ✓ Les centrales hydroélectriques
- ✓ Les centrales solaires
- ✓ Les centrales éoliennes et géothermiques

a. Les centrales hydroélectriques

L'énergie hydraulique est depuis longtemps une solution mise en œuvre dans la production d'électricité, appelée aussi hydroélectricité, car elle utilise une énergie renouvelable.

À un étranglement des rives d'un cours d'eau, un barrage est érigé pour créer une retenue d'eau. Au pied de ce barrage, on installe des turbines reliées à des alternateurs. On alimente en eau sous pression les turbines par un système de canalisations et de régulateurs de débit.

Il y a différents types de centrales hydroélectriques, notamment les microcentrales, installées sur des rivières en tête de bassin, certaines avec un fort impact écologique.

Il existe également des centrales hydroélectriques de pompage-turbinage qui permettent d'accumuler l'énergie venant d'autres sites de productions peu flexibles telles les centrales nucléaires lorsque la consommation est basse, pour la restituer lors des pics de consommation.

b. Les centrales solaires

Elles sont subdivisées en :

- les centrales solaires photovoltaïques ;
- les centrales solaires thermiques.

i. Les centrales solaires photovoltaïques

Ce mode de production d'électricité avec l'énergie solaire utilise les rayonnements lumineux du soleil, qui sont directement transformés en un courant électrique par des cellules à base de silicium ou autre matériau ayant des propriétés de conversion lumière/électricité. Chaque cellule délivrant une faible tension, les cellules sont assemblées en panneaux.

Ce moyen peut être utilisé comme moyen de production à l'échelon individuel.

Dans une maison individuelle, 20m² de panneaux produisent, sur la totalité d'une année, ce que consomme un foyer de quatre personnes. Ils sont préconisés dans la réalisation des maisons dites « autonomes » ou « passives ».



ii. Les centrales solaires thermiques

Une centrale solaire thermique capte un maximum d'énergie thermique solaire en utilisant plusieurs rangées de miroirs disposés en arc de cercle face à la course du soleil, qui renvoient les rayons solaires en un seul point, le foyer. Pour que le foyer ne change pas de position en permanence, les miroirs sont orientables et pilotés par un système centralisé. À ce foyer, une chaudière contenant un liquide sert de capteur d'énergie.

Un autre système utilise des miroirs incurvés face au sud dans l'hémisphère nord munis d'un tube rempli d'un fluide qui s'échauffe aux rayons du soleil concentrés par le miroir. Le liquide est en général de l'eau qui surchauffée par l'énergie thermique solaire est conduite jusqu'à une turbine à vapeur.

Un autre système appelé tour solaire utilise l'énergie solaire pour chauffer l'air contenu dans une immense serre. L'air chauffé est alors plus léger et monte dans une cheminée où sont actionnées des turbines.

c. Les centrales éoliennes et géothermiques

Dans une centrale éolienne, l'énergie électrique est produite directement par des génératrices éoliennes. Ces machines formées d'un mat, surmonté d'un générateur électrique entraîné par une hélice, elles sont positionnées idéalement sur les plans d'eau ou les collines ventées. L'alternateur permet de transformer cette énergie mécanique en énergie électrique.

La terre est composée d'une croûte, posée sur un manteau de roche en fusion. Le principe de l'énergie géothermique consiste à creuser un trou dans cette croûte, à envoyer un fluide caloporteur au fond à l'aide d'un tuyau et à récupérer ce fluide chauffé remontant par un autre tuyau. Cette chaleur fait tourner des turbines qui entraînent des alternateurs. Cette énergie est d'un usage courant en Islande où elle est facile à mettre en œuvre.



II.2. Les ouvrages ou postes

Un poste ou ouvrage est une entité physique définie par sa localisation et ses fonctionnalités dans les réseaux électriques. La vocation d'un poste est avant tout d'assurer la transition entre deux niveaux de tension et/ou d'alimenter l'utilisateur final.

Il existe trois types de poste :

- Le poste HT/MT en distribution publique ;
- Le poste MT/MT en distribution publique ;
- Le poste MT/BT en distribution publique.

II.2.1. Le poste HT/MT en distribution publique

Cet ouvrage est présent dans toute structure électrique d'un pays ; il est situé entre le réseau de répartition et le réseau de distribution MT. Sa fonction est d'assurer le passage de la HT ($\gg 100$ kV) à la MT ($\gg 10$ kV).

II.2.2. Le poste MT/MT en distribution publique

Cet ouvrage peut réaliser deux fonctions :

- assurer la démultiplication des départs MT en aval des postes HT/MT. Dans ce cas, le poste ne comporte aucun transformateur. Il est constitué de deux arrivées MT et de 8 à 12 départs MT ;
- ✓ assurer le passage entre deux niveaux MT. De tels postes MT/MT intègrent des transformateurs. Ils sont nécessaires dans certains pays qui utilisent deux niveaux successifs de tension sur leur réseau MT.

II.2.3. Le poste MT/BT en distribution publique

Localisé entre le réseau de distribution MT et le réseau de distribution BT, cet ouvrage assure le passage de la MT ($\gg 10$ kV) à la BT ($\gg 100$ V).



II.3. Les matériels

Les lignes à haute tension et les appareillages constituent les matériels.

II.3.1. Les lignes à haute tension

La ligne à haute tension est le composant principal des grands réseaux de transport d'électricité. Elle transporte l'électricité de la centrale électrique au consommateur.

Les lignes à haute tension aériennes sont composées de câbles conducteurs, généralement en alliage d'aluminium, suspendus à des supports, pylônes ou poteaux. Ces supports peuvent être faits de bois, d'acier, de béton, d'aluminium ou parfois en matière plastique renforcée.

Le but d'utiliser ces lignes est de réduire les chutes de tension en ligne, les pertes en ligne, et également d'améliorer la stabilité des réseaux.

Les principaux composants sont les pylônes, les conducteurs, les isolateurs et câbles de garde.

II.3.1.1. Pylônes

Pour les lignes aériennes, des pylônes, généralement réalisés en treillis d'acier supportent et maintiennent les conducteurs à une distance suffisante du sol et des obstacles : ceci permet de garantir la sécurité et l'isolement par rapport à la terre, les câbles étant non isolés pour en limiter le poids et le coût. L'inconvénient est leur exposition aux intempéries telles que les, tempêtes, le poids de la glace qui peut les endommager.

II.3.1.2. Les conducteurs

Le courant électrique est transporté dans des conducteurs, généralement sous forme triphasée, avec au moins trois conducteurs par ligne. Pour une phase, on peut aussi trouver un faisceau de conducteurs de deux à quatre à la place d'un simple conducteur afin de limiter les pertes et d'augmenter la puissance pouvant transiter.

Les conducteurs en cuivre sont de moins en moins utilisés. On utilise en général des conducteurs en alliage d'aluminium, ou en combinaison aluminium-acier pour les câbles plus anciens ; ce sont des conducteurs composés d'une âme centrale en acier sur laquelle sont tressés des brins d'aluminium. Les conducteurs ne sont pas revêtus d'un isolant.

Les conducteurs hautes tensions sont principalement aériens ou quelque fois souterrains Les conducteurs aériens sont soumis à l'action des facteurs atmosphériques : température, vent, pluie, verglas etc. Ces facteurs interviennent de façon importante dans le choix des paramètres d'une ligne



haute-tension : type de conducteur électrique (matériaux et géométrie), hauteur et distance des pylônes, tension mécanique maximum sur le conducteur afin de maintenir une garde au sol suffisante, etc. Le choix de ces paramètres a une grande influence sur les coûts de construction et d'entretien d'une ligne de transmission, ainsi que sur sa fiabilité et sur sa longévité. Toutes choses égales par ailleurs la position des conducteurs influe sur l'intensité et la disposition du champ électromagnétique.

II.3.1.3. Isolateurs et câbles de garde

L'isolation entre les conducteurs et les pylônes est assurée par des isolateurs. Ceux-ci sont réalisés en verre, en céramique, ou en matériaux synthétiques. Les isolateurs en verre ou céramique ont en général la forme d'une assiette. On les associe entre eux pour former des chaînes d'isolateurs. Plus la tension de la ligne est élevée, plus le nombre d'isolateurs dans la chaîne est important.

Les câbles de garde ne transportent pas le courant. Ils sont situés au-dessus des conducteurs. Ils jouent un rôle de paratonnerre au-dessus de la ligne, en attirant les coups de foudre, et en évitant le foudroiement des conducteurs. Ils sont en général réalisés en almelec-acier.

II.3.2. Les appareillages

Les appareillages sont principalement utilisés dans le réseau MT. Il est nécessaire de comprendre leur fonction et les principaux appareils utilisés.

II.3.2.1. La fonction des appareillages

L'appareillage MT permet de réaliser les trois fonctions de base suivantes :

- ✓ le sectionnement qui consiste à isoler une partie d'un réseau pour y travailler en toute sécurité,
- ✓ la commande qui vise à ouvrir ou fermer un circuit dans ses conditions normales d'exploitation,
- ✓ la protection qui est dédié à isoler une partie d'un réseau en situation anormale.



Le tableau 2 ci-dessous montre les fonctions des appareils MT

Appareil MT	Sectionneur	Interrupteur	Disjoncteur	Interrupteur-sectionneur	Disjoncteur débrochable	fusible
Sectionnement	X			X	X	
Commande		X	X	X	X	
Protection			X		X	X

Tableau 2: Les différentes fonctions des appareils MT

II.3.2.2. Les principaux appareillages utilisés

Il se présente essentiellement sous trois formes :

- ✓ d'appareils en séparé ou fixés directement sur un mur et protégés d'accès par une porte grillagée,
- ✓ d'enveloppes métalliques ou cellules MT contenant ces appareils,
- ✓ de tableaux MT qui sont des associations de plusieurs cellules.

Deux appareils sont principalement utilisés et ce sont :

- ✓ le disjoncteur MT : Cet appareil, dont la fonction principale est la protection, assure également la fonction commande, et suivant son type d'installation le sectionnement
- ✓ l'interrupteur MT : Cet appareil, dont la fonction principale est la commande, assure aussi souvent la fonction sectionnement. De plus, il est complété de fusibles MT pour assurer la protection des transformateurs MT/BT



CHAPITRE III : PROTECTIONS DES RESEAUX ELECTRIQUES

La protection des réseaux électriques désigne l'ensemble des appareils de surveillance et de protection assurant la stabilité d'un réseau électrique. Cette protection est nécessaire pour éviter la destruction accidentelle d'équipements coûteux et pour assurer une alimentation électrique ininterrompue.

Pour comprendre les protections, il faut avoir des notions sur les technologies existants et les matériels de conduite et de surveillance utilisés.

III.1. Les technologies de protection

Une unité de protection a pour mission de surveiller en permanence les divers paramètres d'une partie d'un réseau tels que les lignes, câbles ou transformateurs, d'agir en situation anormale, et de plus en plus de transmettre des informations pour l'exploitation du réseau.

Longtemps limitées à la technologie électromécanique, les unités de protection connaissent aujourd'hui une évolution fondamentale avec l'utilisation des microprocesseurs.

Ainsi les matériels disponibles à ce jour reposent sur les trois technologies : électromécanique, analogique et numérique.

La plus ancienne est la technologie électromécanique, les relais sont simples et spécialisés, contrôle du courant, de la tension, de la fréquence,..., mais d'une faible précision, leurs réglages sont susceptibles de dérive dans le temps.

La technologie électronique analogique telle que le transistor plus récente a apporté la précision et la fidélité.

La technologie numérique a permis grâce à la puissance de traitement des microprocesseurs, de réaliser des unités de traitement de l'information qui peuvent : assurer globalement les diverses protections, remplacer l'automatisme de la cellule, fournir à l'exploitant la mesure des paramètres électriques.



III.2. Les matériels de conduite et de surveillance utilisés

Ce sont principalement :

- La télésurveillance
- La télécommande
- La téléconduite

III.2.1. La télésurveillance

Elle regroupe les signalisations de position des différents appareils MT, leur déclenchement éventuel sur défaut, la mesure des consommations instantanées ou pondérées dans les différentes parties du réseau électrique, et toute autre information permettant de connaître l'état, à jour, du réseau.

III.2.2. La télécommande

La commande à distance de l'ouverture et de la fermeture des appareils de puissance est l'exemple élémentaire de la télécommande. L'application pratique en est les interrupteurs et disjoncteurs MT télécommandés. D'autres actions peuvent être télécommandées telles que les réglages, l'automatisme.

III.2.3. La téléconduite :

Elle est une source d'économies au niveau de l'exploitation du réseau. En effet, sans avoir à se déplacer, l'exploitant peut en permanence contrôler et intervenir sur le fonctionnement de son réseau. Un exemple : suite à un défaut il est possible de changer rapidement le schéma d'exploitation du réseau afin de rendre minimale la partie de réseau non alimentée, et cela en consultant à distance les indicateurs de localisation de défaut installés en différents endroits sur le réseau, puis en agissant sur les interrupteurs télécommandés. Il en résulte une forte réduction de l'énergie non distribuée, mais aussi une optimisation de ce réseau avec les possibilités de gérer au mieux la répartition des charges.

PARTIE 2 : MATERIELS ET METHODES



CHAPITRE IV : CONSTITUTION D'UN ORDINATEUR

Pour les étudiants des diverses universités dans le monde, il est devenu impossible de se passer de l'ordinateur tant qu'il est indispensable dans leurs études supérieures. L'ordinateur est un outil de travail nécessaire pour faire des recherches en utilisant l'internet, pour les tâches bureautiques et pour utiliser les logiciels liés à leurs études. Pourtant, tout cela n'est que la partie logicielle d'un ordinateur, il y a aussi la partie physique.

Un ordinateur est composé de deux parties indissociables : la partie physique et la partie logicielle.

IV.1. La partie physique de l'ordinateur

La partie physique d'un ordinateur s'articule essentiellement autour du microprocesseur et des périphériques.

IV.1.1. Le microprocesseur.

Le microprocesseur (CPU) est comme le cœur de l'ordinateur, la mémoire vive (RAM) est l'endroit où le microprocesseur place les données sur lesquelles il travaille et la carte mère est le support du microprocesseur et de la RAM.

Tout élément qui transmet des données au microprocesseur ou reçoit des données du microprocesseur est un périphérique.

IV.1.2. Les périphériques.

Un périphérique est un élément matériel de l'ordinateur. Pour comprendre les périphériques, il est nécessaire d'avoir des notions autour des périphériques d'entrée et de sortie, des périphériques externe et interne et le mode de communication du périphérique avec l'ordinateur.

IV.1.2.1. Les périphériques d'entrées et de sortie

Un périphérique est appelé périphérique d'entrée quand il transmet des données vers l'ordinateur par exemple : souris, clavier, lecteur de cd-rom, disque dur.

Il est appelé périphérique de sortie quand il reçoit des données provenant de l'ordinateur par exemple : écran, imprimante, disque dur.

Un disque dur peut être considéré, à la fois, comme périphérique d'entrée, quand des données sont lues sur le disque dur, et périphérique de sortie, quand des données sont écrites sur le disque dur.



IV.1.2.2. Les périphériques externe et interne

Le périphérique interne est un périphérique situé à l'intérieur de l'ordinateur par exemple : carte graphique, disque dur, lecteur de cd-rom interne, tandis que, le périphérique externe est un périphérique situé à l'extérieur de l'ordinateur par exemple : clavier, souris, écran.

IV.1.2.3. Le mode de communication des périphériques

Un périphérique communique avec l'ordinateur par l'intermédiaire d'un port. Il existe deux types de port :

- ✓ Les ports internes ;
- ✓ Les ports externes.

a. Les ports internes

Les ports internes sont constitués par :

- IDE : les disques durs, lecteurs de cd-rom ;
- SATA : la nouvelle génération de disques durs ;
- PCI : les cartes graphiques, les cartes sons, les cartes réseaux ;
- AGP : les cartes graphiques ;
- PCI Express : la nouvelle génération de carte graphique.

b. Les ports externes

Ce sont :

- PS/2 : souris et claviers ;
- USB : tous sorts de périphériques externes ;
- VGA : écran classique et certains écrans plats ;
- DVI : la dernière génération d'écrans plats.

Pour fonctionner, les périphériques ont besoin de pilotes. Ces pilotes sont installées dans un système d'exploitation comme Windows par exemple. La partie logicielle de l'ordinateur est complémentaire à la partie physique.



IV.2. Partie logicielle de l'ordinateur

La partie logicielle de l'ordinateur est indissociable de la notion du système d'exploitation.

Un système d'exploitation est l'ensemble de logiciels d'un appareil informatique qui sert d'interface entre les périphériques et les logiciels applicatifs.

Il faut avoir des notions sur le type de système d'exploitation et les logiciels installés dans ce système.

IV.2.1. Le type de système d'exploitation : Windows XP

Windows XP est une ligne de systèmes d'exploitation multitâche propriétaire développé par Microsoft pour tout usage de l'ordinateur. Comprenant les ordinateurs fixes, portables et Media Center. Les lettres "XP" proviennent d'*eXPérience*. Il a pour nom de code "Whistler" et a été renommé enfin en Windows XP.

Windows XP est officiellement le successeur de Windows Me et de Windows 2000 et est le premier système d'exploitation familial développé par Microsoft basé sur le noyau de NT et l'architecture de Windows NT. Il correspond d'ailleurs à la version 5.1 de Windows NT.

L'édition la plus courante de Windows XP est la version Familiale qui est conçue pour les utilisateurs domestiques tandis que la Professionnelle qui comprend le partage avancé et le Symmetric multiprocessing en plus est destiné aux utilisateurs avec pouvoir sur le réseau et aux ordinateurs de société. Windows XP Media Center a des fonctionnalités multimédia complémentaires pour permettre par exemple de voir et d'enregistrer la TV, regarder plus confortablement des DVD's

Le développement de Windows XP a commencé en 1999, lorsqu'a commencé le développement de Windows Neptune, un système d'exploitation qui a essayé d'être la version "familiale" de Windows 2000. Ce projet a été abandonné et un nouveau a commencé : Whistler qui devint Windows XP. Un grand nombre d'idées des projets Neptune et Odyssey ont été intégrées à Windows XP.



IV.2.2. Les logiciels installés

Ce sont principalement :

- Microsoft Office 2007
- Microsoft Visual Studio 6.0

IV.2.2.1. Microsoft Office 2007

Microsoft Office est une suite bureautique propriétaire. Elle est propriété de Microsoft et est développée pour les plateformes Windows et Macintosh. Microsoft Office inclut la suite bureautique, les serveurs associés et les services basés sur le Web. Les versions récentes de Microsoft Office s'appellent maintenant « Office System » plutôt que « Office Suite » pour mieux refléter le fait qu'elles incluent aussi les serveurs.

La première version de la suite bureautique contenait Word, Excel et PowerPoint. Il a également existé une offre commerciale « *Pro* » qui incluait Microsoft Access et Microsoft Schedule Plus. Au cours des années, les applications bureautiques se sont développées, partageant certains composants comme un correcteur orthographique, la possibilité d'intégrer un élément OLE et les scripts en VBA.

À noter qu'Office 2007, sorti fin janvier 2007, a une interface utilisateur différente et utilise un nouveau format basé sur le XML.

IV.2.2.2. Microsoft Visual Studio 6.0

Microsoft Visual Studio est une suite de logiciels de développement pour Windows conçu par Microsoft.

Visual Studio est un ensemble complet d'outils de développement permettant de générer des applications Web ASP.NET, des Services Web XML, des applications bureautiques et des applications mobiles. Visual Basic, Visual C++, Visual C# et Visual J# utilisent tous le même environnement de développement intégré (IDE, Integrated Development Environment), qui leur permet de partager des outils et facilite la création de solutions faisant appel à plusieurs langages.

Par ailleurs, ces langages permettent de mieux tirer parti des fonctionnalités du Framework .NET, qui fournit un accès à des technologies clés simplifiant le développement d'applications Web ASP et de Services Web XML grâce à Visual Web Developer.

La version 6.0 date de 1998 a été la base du développement pour Microsoft pour les quatre années suivantes, jusqu'au passage au développement .NET.



Visual Studio 6.0 a été la dernière version incluant Visual Basic tel que le connaissait la majeure partie des programmeurs VB. C'est également la dernière version à inclure Visual J++.



CHAPITRE V : LE LANGAGE DE PROGRAMMATION VISUAL BASIC 6.0

Le logiciel Visual Basic 6.0 fait partie des divers logiciels dans le Microsoft Visual Studio 6.0. Cette version est différente de celle qui est disponible dans le système Windows connu sous le nom de Visual Basic Application ou VBA. Le logiciel Visual Basic 6.0 nécessite l'installation de MSDN ou Microsoft Developer Network pour la documentation.

Le type de projet utilisé pour ce mémoire est l'EXE standard qui permet de créer un fichier exécutable standard.

Dans le langage de programmation Visual Basic, la notion d'objet est importante. Un objet est une combinaison de code et de données pouvant être traitée comme une entité, par exemple un contrôle, une feuille ou un composant d'application.

Pour comprendre le fonctionnement de Visual Basic 6.0, il est nécessaire d'avoir des notions sur : les contrôles, les variables, les procédures et les instructions.

V.1. Les contrôles.

C'est un objet graphique, tel qu'une zone de texte, un rectangle ou un bouton de commande, qu'on peut placer dans un formulaire pour faciliter sa lecture, afficher des données ou effectuer une opération.

Les principaux contrôles utilisés sont résumés dans le tableau N°3 ci-dessous :



Les feuilles « form »	C'est le contrôle de base incontournable en VB. On ne peut créer et utiliser d'autres contrôles que si ceux-ci font partie d'une Form. C'est pourquoi au démarrage de Visual Basic une feuille est proposée par défaut.
CommandButton	Ce contrôle permet à l'utilisateur de commencer ou d'interrompre une opération, ou encore d'y mettre fin.
Label	Ce contrôle est un contrôle graphique qui permet d'afficher du texte que l'utilisateur ne peut pas modifier directement.
TextBox	Ce contrôle, parfois appelé champ d'édition ou contrôle d'édition, affiche des informations entrées au moment de la création, tapées par l'utilisateur ou affectées au contrôle au moment de l'exécution.
VScrollBar et HScrollBar	Les barres de défilement VScrollBar et HScrollBar sont des contrôles permettant de parcourir rapidement une longue liste d'éléments ou un grand nombre d'informations.
OptionButton	Ce contrôle affiche une option qui peut être activée ou désactivée. Ils font généralement partie d'un groupe proposant à l'utilisateur plusieurs options dont il ne peut sélectionner qu'une seule.
PictureBox	Ce contrôle peut afficher un élément graphique issu d'une image bitmap, d'une icône ou d'un métafichier, ainsi que d'un métafichier étendu ou de fichiers JPEG ou GIF. L'élément graphique est découpé si le contrôle n'est pas assez grand pour afficher l'image complète. Il peut également regrouper des contrôles OptionButton.
ToolBar	Ce contrôle contient une collection d'objets Button utilisée pour créer une barre d'outils associée à une application. En général, il contient des boutons qui correspondent aux éléments d'un menu d'une application, fournissant ainsi une interface graphique par laquelle les utilisateurs peuvent accéder aux fonctions et aux commandes les plus fréquemment utilisées.
ImageList	Ce contrôle contient une collection d'objets ListImage qui peuvent chacun être référencés par leur index ou leur clé. Il est aussi destiné à être utilisé comme référentiel central à partir duquel les autres contrôles peuvent facilement obtenir des images.
CommonDialog	Ce contrôle fournit un ensemble de boîtes de dialogue standard destinées à des opérations telles que l'ouverture et l'enregistrement de fichiers, la définition des paramètres d'impression et la sélection de couleurs et de polices. Il permet également d'afficher une aide en exécutant le moteur d'aide de Windows.
Groupes de contrôles	C'est un ensemble de contrôles de même type et portant le même nom. Les contrôles d'un groupe partagent également les mêmes procédures d'événement mais les propriétés de ses éléments ont leur propre valeur

Tableau 3: Liste des contrôles usuels

Tous les contrôles possèdent un jeu de propriétés, de méthodes et d'événements.



V.1.1. Les propriétés.

Les propriétés sont les attributs nommés d'un objet. Elles définissent les caractéristiques du contrôle. Les propriétés utilisées par les divers contrôles cités précédemment sont explicitées dans le tableau N°4 ci-dessous:

Propriétés	Fonctions	Applications
Name	renvoie le nom utilisé dans le code pour identifier le contrôle.	Label, TextBox...
Alignement	renvoie ou définit une valeur qui détermine l'alignement d'un contrôle OptionButton ou du texte d'un contrôle	Label, OptionButton
BackColor	renvoie ou définit la couleur de l'arrière-plan.	CommandButton, Label,...
BorderStyle	renvoie ou définit le style de la bordure.	PictureBox, Label,...
Caption	détermine le texte affiché dans un contrôle ou en regard de celui-ci.	CommandButton, Label,...
Enabled	renvoie ou définit une valeur qui détermine si un contrôle peuvent répondre aux événements générés par l'utilisateur.	VScrollBar, OptionButton
Filename	renvoie ou définit le chemin d'accès et le nom du fichier sélectionné.	CommonDialog,...
Height et Width	renvoient ou définissent les dimensions d'un contrôle. Les dimensions sont mesurées à partir du centre de la bordure du contrôle de façon à ce que les contrôles ayant différentes largeurs de bordure s'alignent correctement.	Label, TextBox,...
Image	renvoie ou définit une valeur qui détermine l'objet ListImage d'un contrôle ImageList.	ToolBar,...
Index	renvoie ou définit le nombre qui identifie de façon unique un contrôle dans un groupe de contrôles. Cette propriété n'est disponible que si le contrôle fait partie d'un groupe de contrôles.	CommandButton, Label,...
InitDir	renvoie ou définit le dossier de fichier initial	CommonDialog,...
LargeChange	définit le nombre de graduations sur lesquelles le curseur se déplace lorsqu'il clique avec la souris à gauche ou à droite du curseur.	VScrollBar, HScrollBar, ...
SmallChange	définit le nombre de graduations sur lesquelles le curseur se déplace lorsque l'utilisateur appuie sur les touches flèches gauche ou droite.	VScrollBar, HScrollBar, ...
ToolTipText	renvoie ou définit une info-bulle	CommandButton, Label,...
Visible	renvoie ou définit une valeur indiquant si un objet est visible ou caché.	OptionButton, TextBox,...

Tableau 4: Liste des propriétés



V.1.2. Les méthodes

Une méthode est une action susceptible d'être exécutée par un contrôle. Tous les contrôles possèdent un jeu de méthodes. Ci-dessous se trouvent les méthodes :

- ShowSave : affiche la boîte de dialogue Enregistrer
- ShowOpen : affiche la boîte de dialogue Ouverture
- Show : Affiche un objet Form

La plupart des méthodes des autres contrôles n'ont pas été utilisées.

V.1.3. Les évènements

Un évènement est une action reconnue par un contrôle pour laquelle on peut écrire du code afin d'obtenir une réponse. Les principaux évènements utilisés sont :

- L'évènement Click : se produit lorsque l'utilisateur clique un bouton de la souris puis le relâche sur un objet. Dans le cas des contrôles CommandButton et OptionButton, l'évènement Click se produit uniquement lorsque l'utilisateur clique le bouton gauche de la souris
- L'évènement Change : indique que le contenu d'un contrôle a changé. Dans le cas des contrôles HScrollBar et VScrollBar (barres de défilement horizontales et verticales), l'évènement Change déplace le curseur de défilement sur la barre de défilement. Il se produit lorsque l'utilisateur fait défiler l'affichage

Il est à remarquer que le contrôle CommonDialog n'a pas un jeu d'évènements.

V.2. Les variables.

Les variables sont des espaces réservés destinés à stocker des valeurs. Elles possèdent un nom et un type de données. Le type de données par défaut d'une variable est Variant.

Pour comprendre les variables, il faut avoir des notions sur les types de données, les classes et les déclarations des variables.

V.2.1. Les différents types de données

Quand une variable est déclarée, l'utilisateur peut spécifier son type de donnée. Il existe plusieurs types de données dont les plus utilisés sont :

- Double : ils sont stockés sous la forme de nombres à virgules flottantes en double précision
- Integer : entiers relatifs variant entre -32 768 et 32 767
- Single : ils sont stockés sous la forme de nombre à virgules flottantes en simple précision



- Boolean : pour la valeur TRUE et FALSE
- String : pour les chaînes de caractères

L'utilisateur peut créer aussi un type de données avec l'instruction TYPE. Ce type de données contient un ou plusieurs éléments et il n'est utilisé qu'au niveau module.

V.2.2. Les déclarations des variables.

Les variables sont déclarées au début des feuilles de codes. L'utilisateur peut utiliser les instructions suivantes :

- **Dim** : déclare les variables. Les variables déclarées à l'aide de l'instruction Dim au niveau module sont disponibles pour toutes les procédures du module. Au niveau procédure, les variables ne sont disponibles qu'au sein de celle-ci.
- **Public** : déclare les variables publiques. Les variables déclarées avec l'instruction Public sont accessibles à toutes les procédures, dans l'ensemble des modules de toutes les applications.

V.2.3. Les classes.

Une classe est un modèle à partir duquel un objet est créé ; un module de classe est un module contenant la définition d'une classe, notamment les définitions de ses propriétés et de ses méthodes.

Le code d'un module de classe décrit les attributs et le comportement des objets créés à partir de la classe.

L'utilisateur peut déclarer des variables comme des classes.

V.3. Les procédures.

On appelle procédure un sous-programme qui permet d'effectuer un ensemble d'instructions par simple appel dans le corps du programme appelant.

Il est nécessaire d'avoir des notions sur les procédures de type SUB et celle FUNCTION.

V.3.1. Les procédures de type SUB.

La procédure SUB est un ensemble d'instructions ne retournant pas de valeur.

Parmi les procédures « Sub », on distingue encore deux catégories. Il y a :

- Les procédures événementielles : elles sont associées à un objet et sont exécutées quand l'évènement se réalise. La syntaxe est la suivante :

[Private][Static]Sub Objet_Eventement (**arguments**)

statement

**End sub**

- Les procédures publiques : elles sont utilisées lorsqu'un bloc d'instructions est utilisé dans plusieurs procédures. Il est nécessaire de les appeler contrairement aux procédures événementielles. La syntaxe est :

[Public][Static]Sub NomSub (**arguments**)

statement

End sub

L'instruction CALL transfère un contrôle à une procédure SUB. La syntaxe est :

Call name [*argumentlist*]

L'instruction CALL est facultative mais s'il est indiqué, il faut placer argumentlist entre parenthèses. Toutefois, cette instruction n'est pas nécessaire pour appeler la procédure, le nom name de la procédure suffit.

V.3.2. Les procédures de type FUNCTION.

Une procédure FUNCTION est un ensemble d'instructions retournant une valeur en sortie.

La syntaxe pour la procédure « Function » est :

[Private|Public][Static]Function name (**arguments**)[**as type**]

Statement

name = expression

End Function**V.3.3. Les instructions pour appeler les procédures**

Ce sont les instructions:

- **Call**: transfère un contrôle à une procédure Sub ou à une procédure Function.
- Pour les procédures de type "Function", ils peuvent être appelés en utilisant la syntaxe suivante : variable= NomFunction(Argument)

V.4. Les instructions.

Ils peuvent être regroupés en deux : les structures et les autres instructions.

V.4.1. Les structures.

Le programmeur est très souvent amené à tester des valeurs et à orienter le programme selon ces valeurs. L'utilisateur est parfois invité à faire des choix que le programme doit prendre en compte.



Plusieurs structures décisionnelles permettent ces traitements. Ces différentes structures sont développées dans le tableau N°5 ci-dessous :

Structure	Syntaxe
If...then	Exécution d'un traitement suite à l'évaluation booléenne d'une condition. Cette instruction peut s'écrire sur une ligne ou en bloc If condition Then [statements] If condition Then End If
If...Then ... Else	Evaluations imbriquées de conditions booléennes. Cette instruction peut s'écrire sur une ligne ou en bloc If condition Then [statement] Else If condition Then [statement] Else End If
Select Case	Exécution d'un traitement suite à l'évaluation d'une condition à valeur multiple Select Case Expression Case ListValeur1 [statement] Case ListValeur2 [statement] End Select

Tableau 5: Liste des structures



For...Next	<p>Si la valeur de l'incrément est négative, la boucle est décroissante.</p> <p>For Compteur = Début To Fin [Step Incrément]</p> <p>[Instructions]</p> <p>[... Exit For]</p> <p>[Instructions]</p> <p>Next [Compteur]</p>
Do...Loop	<p>Do{ While Until } condition]</p> <p>[statements]</p> <p>[Exit Do]</p> <p>[statements]</p> <p>Loop</p> <p>Répète un bloc d'instruction aussi longtemps qu'une condition est vraie ou jusqu'à ce qu'une condition devienne vraie</p>
For Each...Next	<p>For Each element In groupe</p> <p>[statements]</p> <p>[Exit For]</p> <p>[statements]</p> <p>Next</p> <p>Répète un groupe d'instruction pour chaque élément d'un tableau ou d'une collection</p>
While...Wend	<p>While condition</p> <p>[statements]</p> <p>Wend</p> <p>Exécute une série d'instruction dans une boucle aussi longtemps que la valeur d'une condition est True.</p>

Tableau 6: Listes des boucles



V.4.2. Les autres instructions.

Ce sont les instructions:

- **With...End With** : exécute une série d'instructions appliquées à un seul objet ou à un type défini par l'utilisateur.
- **On Error Resume Next** : Lorsqu'une erreur d'exécution survient, le contrôle est transmis à l'instruction qui suit immédiatement celle où l'erreur s'est produite, et l'exécution continue.
- **On Error GoTo** : invalide dans la procédure en cours tout gestionnaire d'erreurs validé.
- **Input#** : lit des données dans un fichier séquentiel ouvert et les attribue à des variables.
- **Write#** : écrit des données dans un fichier séquentiel.
- **Open** : permet d'exécuter une opération d'Entrée/Sortie sur un fichier.
- **GoTo** : effectue un branchement incondtionnel vers une ligne déterminée d'une procédure.

Le langage Visual Basic reste dans sa catégorie celui qui permet une meilleure prise en main pour son utilisateur. De plus, pour ceux qui ne sont pas habitués au langage de programmation, il facilite grandement leur apprentissage.



CHAPITRE VI : CALCUL DE LA SECTION D'UN CONDUCTEUR

Le calcul de la section des conducteurs dans un réseau arborescent complexe nécessite d'avoir des notions sur le calcul des sections des conducteurs d'une ligne et celui pour un réseau ramifié simple.

VI.1. Calcul des sections des conducteurs d'une ligne.

Pour déterminer les sections des conducteurs, les paramètres linéiques doivent tenir compte de certains facteurs. Ce sont :

- La résistance de chaque conducteur
- L'inductance propre des conducteurs
- L'inductance mutuelle entre les conducteurs
- Le phénomène d'induction électromagnétique dû au sol
- Les capacités sol-conducteurs
- Les capacités mutuelles des conducteurs

Ce sont les principaux facteurs qu'il faut tenir en compte.

Il est à noter que, pour toutes les installations, il y a une chute de tension à respecter.

Pour ce mémoire, la détermination des sections admissibles des conducteurs suit certaines hypothèses qui sont :

- Les récepteurs ont le même facteur de puissance
- Les déphasages entre les tensions des récepteurs sont faibles
- Les tensions efficaces sont sensiblement égales
- Les lignes sont constituées d'un même conducteur de même nature
- La puissance réactive est nulle.

La formule pour la détermination des sections d'une ligne est :

$$s_a = \frac{\rho}{U \Delta U_a} \left(1 + \frac{X'}{R} \operatorname{tg} \varphi P l \right) [3]$$

Où :

- s_a est la section admissible [mm^2]
- ρ , la résistivité des conducteurs [mm^2/m]
- U , la tension au départ [V]
- ΔU_a la chute de tension admissible [V]
- X' , la réactance linéique de la ligne



- R' la résistance linéique de la ligne
- $\operatorname{tg} \varphi$ la tangente de l'angle correspondant au facteur de puissance
- P , la puissance transportée par la ligne [kW]
- l , la longueur de la ligne [m]

Avec les hypothèses énoncées précédemment, on a :

$$s_a = \frac{\rho}{U \Delta U_a} P l [4]$$

VI.2. Calcul des sections des conducteurs pour un réseau ramifié simple.

Le schéma ci-dessous représente un réseau ramifié simple :

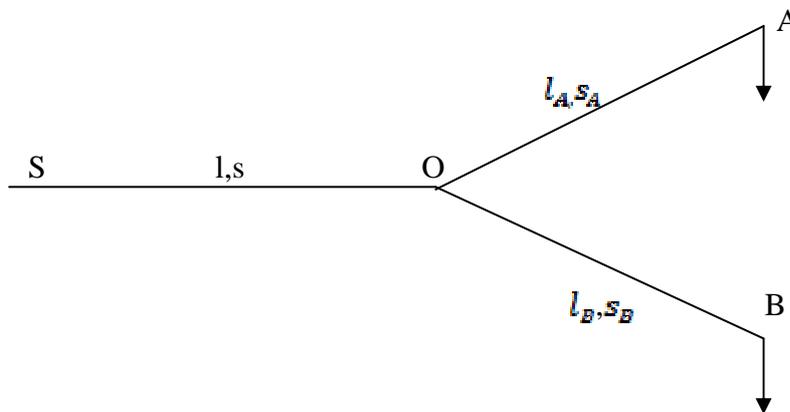


Figure 1 : Schéma d'un réseau ramifié simple

Les hypothèses de la section V.1 sont ici aussi appliquées pour le calcul. L'expression de la chute de tension sur une ligne est :

$$\Delta U = \frac{\rho l P}{U s} [5] \text{ Avec } \rho - \text{ la résistivité des conducteurs}$$

l - la longueur de la ligne [m]

P - la puissance transmise par la ligne [kW]

U - la tension au départ de la ligne [V]

s - la section des conducteurs [mm²]

D'où les expressions des chutes de tensions sur les portions OA, OB et SO du réseau de la figure :

$$\Delta U_A = \frac{\rho l_A P_A}{U s_A} [6]$$

$$\Delta U_B = \frac{\rho l_B P_B}{U s_B} [7]$$

$$\Delta U_{SO} = \frac{\rho l (P_A + P_B)}{U s} [8]$$



Les indices A, B et SO se rapportent aux portions de réseau OA, OB et SO

On peut en déduire la chute de tension entre O et A, puis les sections des conducteurs des portions SA et SB

$$\Delta U_{SA} = \frac{\rho}{U} \left[\frac{l_A P_A}{s_A} + \frac{l(P_A + P_B)}{s} \right] [9]$$

$$s_A = \frac{l_A P_A}{\frac{U \Delta U_{adm}}{\rho} - \frac{l(P_A + P_B)}{s}} [10]$$

$$s_B = \frac{l_B P_B}{\frac{U \Delta U_{adm}}{\rho} - \frac{l(P_A + P_B)}{s}} [11]$$

Le volume total des conducteurs dans tout le réseau sera alors :

$$V = sl + s_A l_A + s_B l_B [12]$$

Si on tient compte des expressions de s_A et de s_B , on peut affirmer que V est fonction de la variable s. Il passe par un minimum si sa dérivée par rapport à s est nulle

Nous ferons $\frac{\delta V}{\delta s} = 0$. Ceci est une équation à une inconnue s. Sa résolution mène à l'expression de la section admissible du tronçon OA correspondant à un volume total de conducteurs minimal:

$$s_{adm} = \frac{\rho P_{eq} (l + l_{eq})}{U \Delta U_{adm}} [13]$$

Avec :

$$P_{eq} = P_A + P_B [14]$$

$$l_{eq} = \sqrt{\frac{l_A^2 P_A + l_B^2 P_B}{P_A + P_B}} [15]$$

La section choisie pour la portion SO sera la section normalisée immédiatement supérieure à s_{adm} .

Il faut tenir compte aussi des chutes de tension sur les portions OA et OB. On en déduit les sections des portions OA et OB.

$$U_o = U_s + \Delta U_{so} \quad \text{avec} \quad \Delta U_{so} = \frac{\rho l (P_A + P_B)}{s_{adm} U} [16]$$

$$\Delta U_{adm\ OA} = \Delta U_{adm\ OB} = \Delta U_{adm} - \Delta U_{so} [17]$$



$$S_{adm\ OA} = \frac{\rho I_A P_A}{U_0 \Delta U_{adm\ OA}} \quad \text{et} \quad S_{adm\ OB} = \frac{\rho I_B P_B}{U_0 \Delta U_{adm\ OB}} [18]$$

Les sections choisies pour les portions OA et OB seront les sections normalisées immédiatement supérieures à $S_{adm\ OA}$ et $S_{adm\ OB}$.

Dans le cas où les valeurs obtenues sont inférieures mais proches d'une section normalisée, moins de 10%, la valeur de la section normalisée prise n'est pas celle qui est immédiatement supérieure à cette section admissible mais la section normalisée suivante.

Ce raisonnement est utilisé pour la détermination des sections dans un réseau arborescent complexe.

VI.3. Calcul des sections des conducteurs dans un réseau arborescent complexe.

Trois notions importantes doivent être assimilées : la règle des moments des puissances, la transformation directe et la transformation inverse.

VI.3.1. Règle des moments des puissances

Considérons les réseaux de la figure N° 2 composé de deux lignes de longueurs l_x et l_y .

Du point de vue détermination de la section ces deux lignes sont équivalentes à une ligne unique de longueur $l = l_x + l_y$ si on a les relations :

$$P \cdot l_x = P_y \cdot l \quad \text{et} \quad P \cdot l_y = P_x \cdot l [19]$$

P : La puissance consommée à un nœud quelconque sur la ligne de longueur l

P_x : La puissance consommée au nœud X

P_y : La puissance consommée au nœud Y

l : La longueur de la ligne unique

l_x : La longueur entre le nœud quelconque et le point X

l_y : La longueur entre le nœud quelconque et le point Y

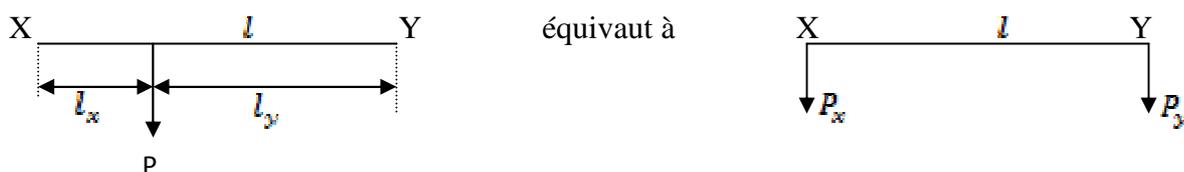


Figure 2 : Schéma équivalent à deux lignes

VI.3.2. Transformation directe.

Pour déterminer les sections des conducteurs dans un réseau complexe, il faut appliquer les résultats du paragraphe précédent d'une part et/ou la règle des moments de puissance d'autre part.

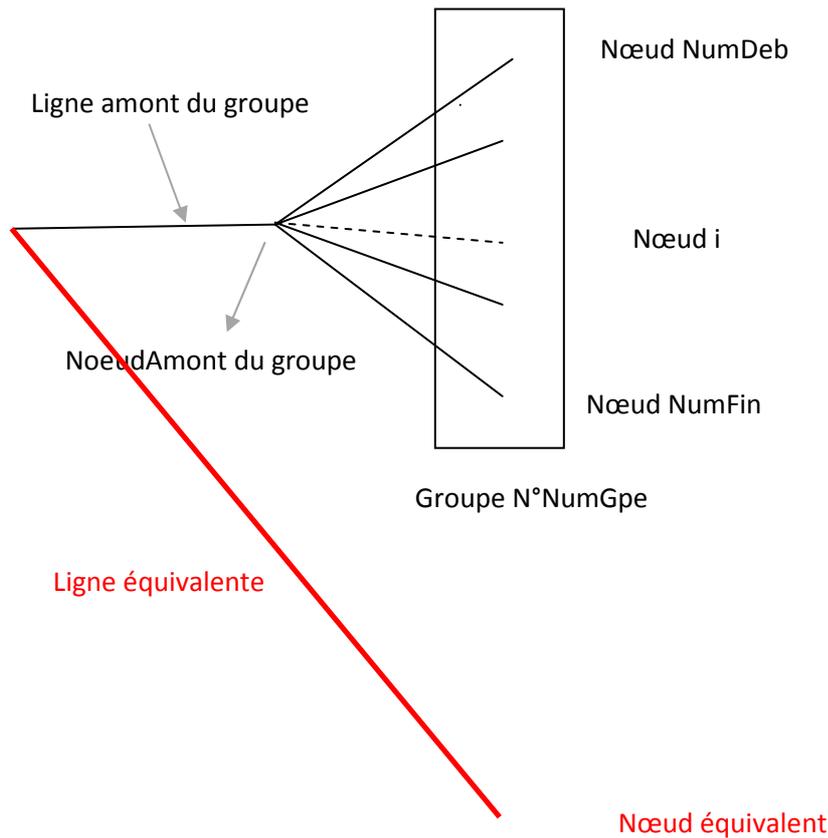


Figure 3 : Ligne équivalente à un groupe

NoeudAmont du groupe : le nœud commun aux différents nœuds du groupe

Ligne amont du groupe : la ligne amont du nœud commun du groupe

Groupe N°NumGpe : un groupe de nœud dont le numéro est la variable NumGpe

Les nœuds ont le numéro i qui varie de NumDeb, le premier nœud du groupe de nœuds, à NumFin, le dernier nœud du groupe.

Dans une première phase, on transforme le schéma jusqu'à l'obtention d'un schéma à une seule ligne : Chaque groupe de nœuds ayant le même nœud amont et la ligne qui se trouve en amont de ce groupe sont remplacés par une seule ligne équivalente et un nœud équivalent, cf figure N°3. On obtient ainsi un schéma équivalent plus simple. On refait la même transformation jusqu'à l'obtention d'un schéma à une seule ligne.

Un exemple de transformation de réseau est montré sur la figure N° 4. Le réseau initial est composé des lignes AB, BC, BD, DE et EF. Une première transformation de I à II consiste à remplacer les lignes BD, DE et DF par une seule ligne BG. Une deuxième transformation de II à III remplace les lignes AB, BC et BG par la ligne AH. Ainsi, tout le réseau sera remplacé par la ligne unique AH.

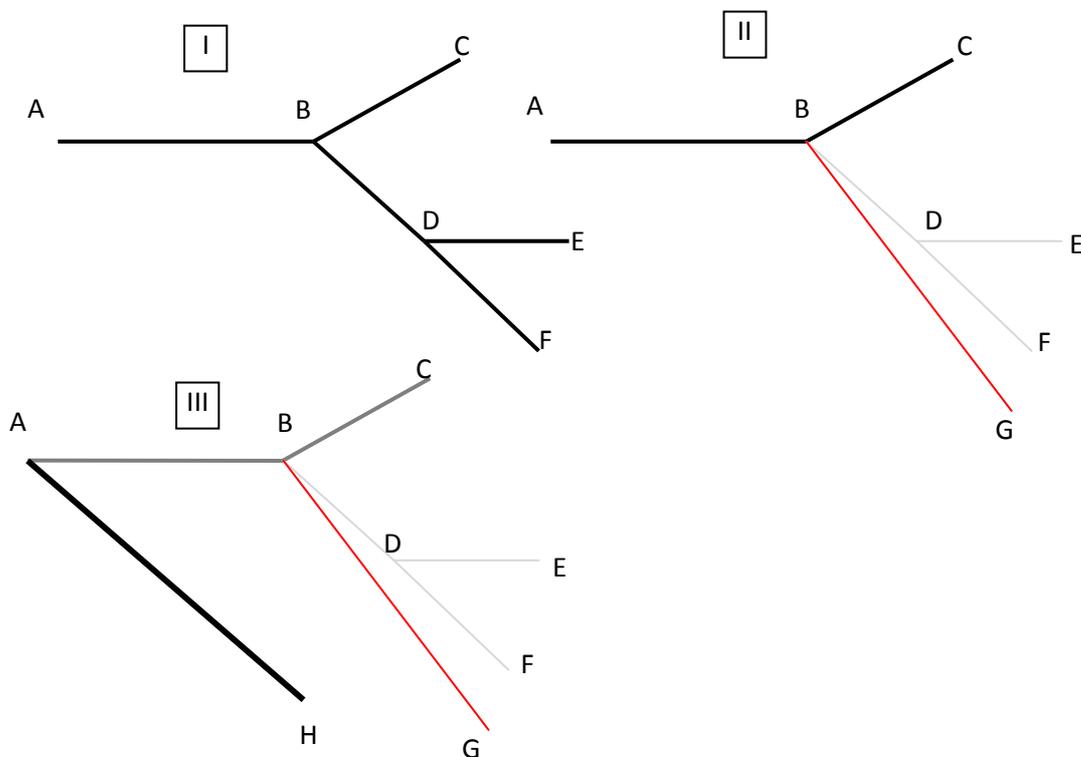


Figure 4 : Exemple de transformation directe d'un réseau



VI.3.3. Transformation inverse.

La transformation inverse consiste à calculer les sections des lignes successives à partir du nœud source, jusqu'aux nœuds bout de lignes. On calcule la section admissible de la ligne équivalente et on en déduit la section de la ligne en amont du groupe. Sachant la section de la ligne en amont du groupe, on peut alors calculer la chute de tension sur cette ligne, puis la tension au nœud amont du groupe et la chute de tension autorisée pour les lignes du groupe. De cette manière, on peut déterminer les sections de toutes les lignes du groupe.



CHAPITRE VII : LES DIFFERENTES ETAPES D'UNE PROGRAMMATION

Le terme « Programmer » veut dire écrire dans un langage de programmation informatique une suite d'instructions, organisée en algorithme dans un but précis, exécutable par un ordinateur. L'élaboration d'un programme nécessite donc une méthode bien définie qui est essentiellement la conception et le codage. En effet, la conception d'un programme répond surtout aux exigences de futurs utilisateurs. Ainsi, en sachant ces exigences, le programmeur définit la conception généralisée du programme. Après, il entre dans la conception des différents sous-programmes. Une minutieuse vérification du programme est nécessaire avant toute validation de celui-ci.

VII.1. Spécifications des exigences et besoins des futurs utilisateurs.

Cette étape est souvent négligée par les programmeurs. En effet, ceux-ci s'intéressent surtout à la partie de création du code. Pourtant, les utilisateurs jugent l'utilité d'un programme s'il répond à un quelconque besoin de leur part. Ainsi, la conception d'un programme est surtout le fruit d'une attente précise des utilisateurs.

VII.2. Conception générale du programme.

La phase de conception générale définit le but du programme. Si on fait une rapide analyse fonctionnelle d'un programme, on détermine essentiellement les données qu'il va traiter (données d'entrée), la méthode employée (appelée l'algorithme), et le résultat (données de sortie). Les données d'entrée et de sortie peuvent être de nature très diverses. On peut décrire la méthode employée pour accomplir le but d'un programme à l'aide d'un algorithme (par exemple un organigramme).

VII.3. Conception détaillée du programme.

Une fois la conception générale finie, on entre dans la phase de création de code. Le codage dépend de l'algorithme défini précédemment ainsi que du langage de programmation utilisé. Cette phase comprend la création des différents sous-programmes en plus de la création de l'interface utilisée par les utilisateurs.



Lorsqu'il s'agit de logiciels qui nécessitent la création de nombreux programmes, cette phase de conception détaillée peut être sous-traitée. Le plus important pour un programmeur est la conception générale du programme

VII.4. Les diverses vérifications.

Avant qu'un programme puisse être définitivement déclaré comme opérationnel, il doit subir une série de tests. Ces tests servent à déterminer si le programme répond effectivement aux exigences des futurs utilisateurs.

Il y a le test unitaire et le test d'intégration.

VII.4.1. Le test unitaire

En programmation, le test unitaire est un procédé permettant de s'assurer du fonctionnement correct d'une partie déterminée d'un logiciel ou d'une portion d'un programme. Il s'agit pour le programmeur de tester un module ou une procédure, indépendamment du reste du programme, ceci afin de s'assurer qu'il répond aux spécifications fonctionnelles.

VII.4.2. Le test d'intégration

Un test d'intégration est un test qui se déroule dans une phase d'un projet informatique suivant les tests unitaires. Il consiste, une fois que les développeurs ont validés leurs développements ou leurs correctifs, à regrouper ensemble leurs modifications dans le cadre d'une livraison. Il s'agit d'établir une nouvelle version basée soit sur une version de maintenance, soit sur une version de développement.

Une fois toutes les vérifications terminées, l'élaboration du programme est terminée.

PARTIE 3 : APPLICATIONS ET RESULTATS

CHAPITRE VIII : PRESENTATION GENERALE DU PROGRAMME

Les notions de Visual Basic ont permis l'élaboration du programme de calcul des sections de conducteurs. Comme tout programme, celui-ci est composé d'une interface pour les utilisateurs et de sous-programmes destinés à le faire fonctionner. Pour comprendre le programme, il faut connaître d'une part la description de l'interface utilisateur et d'autre part le fonctionnement général du programme.

VIII.1. La description de l'interface utilisateur.

L'interface revêt une importance particulière car elle est destinée aux utilisateurs. En effet, la fonctionnalité du programme dépend de sa simplicité d'utilisation. La création de l'interface a respecté ce paramètre. Il est nécessaire pour cela de connaître le menu et la barre d'outils, les diverses entrées de données, la sortie de résultats de calcul et la barre de défilement.

VIII.1.1. Le menu et la barre d'outils

Ces deux fonctionnalités du programme permettent d'accéder aux contrôles suivants :

- Le bouton « Nouveau »
- Le bouton « Ouvrir »
- Le bouton « Enregistrer »
- Le bouton « calculer les sections des conducteurs »
- Le bouton « Quitter »

Les figures N°5 et 6 suivantes montrent cette partie de l'interface :



Figure 5 : La barre d'outils

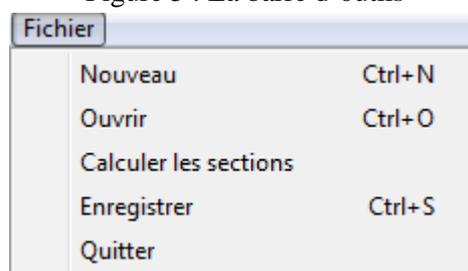


Figure 6 : Le menu



VIII.1.1.1. Le bouton « Nouveau »

Il permet d'initialiser les divers paramètres concernant les nœuds et les lignes. Il permet également de supprimer les sections admissibles que le programme a pu dans sa mémoire. Si ce bouton a été sélectionné par l'utilisateur, il peut entrer de nouvelles données et peut changer les paramètres par défaut du programme concernant la chute de tension admissible, la tension nominale, le type de conducteur et le numéro du nœud source.

VIII.1.1.2. Le bouton « Ouvrir »

Il permet d'ouvrir un fichier contenant les données concernant les nœuds, les lignes, la chute de tension admissible, la tension nominale, le type de conducteur et le numéro du nœud source. Le programme lit ces données et les affiche.

VIII.1.1.3. Le bouton « Enregistrer »

Ce contrôle a pour fonction d'enregistrer les données sur les nœuds, les lignes, la chute de tension admissible, la tension nominale, le numéro du nœud source et le type de conducteur dans un fichier défini par l'utilisateur. Ainsi, l'utilisateur pourra ultérieurement ouvrir le fichier de sauvegarde et reproduire le travail enregistré.

VIII.1.1.4. Le bouton « calculer les sections des conducteurs »

En utilisant ce contrôle, l'utilisateur demande au programme de calculer les sections admissibles correspondantes aux différentes données. Ce bouton fait appel à la même procédure que le bouton VALIDER.

VIII.1.1.5. Le bouton « Quitter »

Ce bouton permet comme son nom l'indique de quitter le programme.

VIII.1.2. Les diverses entrées de données et la sortie des résultats de calcul

L'interface utilisateur est composée essentiellement de zones de textes réservées à l'entrée de données et d'étiquettes réservées aux sorties de résultats de calcul.

Les entrées de données sont subdivisées en trois parties : celle concernant les nœuds, celle concernant les lignes et celle concernant les paramètres par défaut.

VIII.1.2.1. Données concernant les nœuds

La figure N°7 suivante montre la partie de l'interface sur les données concernant les nœuds :

nombre de nœuds <input type="text"/>			
Noeud n°	X	Y	Pi[kW]
<input type="text" value="1"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text" value="2"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text" value="3"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text" value="4"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text" value="5"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text" value="6"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text" value="7"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text" value="8"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text" value="9"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text" value="10"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text" value="11"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text" value="12"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text" value="13"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text" value="14"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

Figure 7 : Partie de l'interface pour les nœuds

En haut de la figure se trouve un TextBox étiqueté « nombre de nœuds ». Il permet d'entrer le nombre de nœuds existants dans le réseau.

La colonne portant le nom de « nœud n° » contient des « TextBox » indiquant le numéro des nœuds correspondants. Leur propriété « enabled » est « false ». L'utilisateur ne peut modifier ce numéro.

Les colonnes « X » et « Y » permettent d'entrer les abscisses et les ordonnées des nœuds.

La colonne « Pi » est réservée pour la puissance dissipée au nœud exprimée en kW

VIII.1.2.2. Données concernant les lignes

La figure N°8 suivante montre cette partie de l'interface :

nombre de lignes					
Ligne n°	X1	Y1	X2	Y2	L[m]
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					

Figure 8 : Partie de l'interface pour les lignes

Tout comme pour les nœuds, en haut, on trouve la zone de saisie du nombre des lignes.

L'entrée des données sur les lignes est divisée en six colonnes :

- ✓ Les numéros des lignes : la propriété « enabled » du TextBox est « false », l'utilisateur ne peut modifier le chiffre ;
- ✓ La colonne « X1 » : elle correspond à l'abscisse du départ de la ligne ;
- ✓ La colonne « Y1 » : elle désigne l'abscisse de l'arrivée de la ligne ;
- ✓ La colonne « X2 » : elle est consacrée à l'ordonnée du départ de la ligne ;
- ✓ La colonne « Y2 » : elle est dédié à l'ordonnée de l'arrivée de la ligne ;
- ✓ La colonne « L » : elle va contenir à la valeur de la longueur de la ligne.

VIII.1.2.3. Données concernant les paramètres par défaut

Il existe des données par défaut pour le programme :

- ✓ La chute de tension maximale est prise par défaut à 7% ;
- ✓ La tension nominale est de 220 [V] ;
- ✓ Le numéro du nœud source est 1 ;
- ✓ Le type de conducteur choisi par défaut est le cuivre.

La figure N°9 suivante montre les paramètres par défaut :

dUmax %	<input type="text" value="7"/>
Unom [V]	<input type="text" value="220"/>
N°du noeud source	<input type="text" value="1"/>

Conducteur

Cuivre

Aluminium

Figure 9: Partie de l'interface pour les paramètres par défaut

VIII.1.2.4. Données concernant la sortie des résultats

Cette partie de l'interface montre les valeurs des sections des conducteurs dans les lignes. Il faut cliquer sur la touche « VALIDER » pour calculer et faire apparaître les résultats.

VIII.1.3. Barre de défilement

Une barre de défilement vertical est prévue à droite de l'écran pour prévoir un réseau de dimension relativement grande.

VIII.2. Le fonctionnement général du programme.

Pour comprendre le fonctionnement du programme, il faut connaître la structure du projet et l'organigramme de la procédure « CmdValider_Click » qui valide les données sur le réseau.

VIII.2.1. La structure du projet

La structure du projet définit les feuilles de travail, les modules et les modules de classe à créer.

Ci-dessous se trouve la structure du projet correspondant à la figure 10:

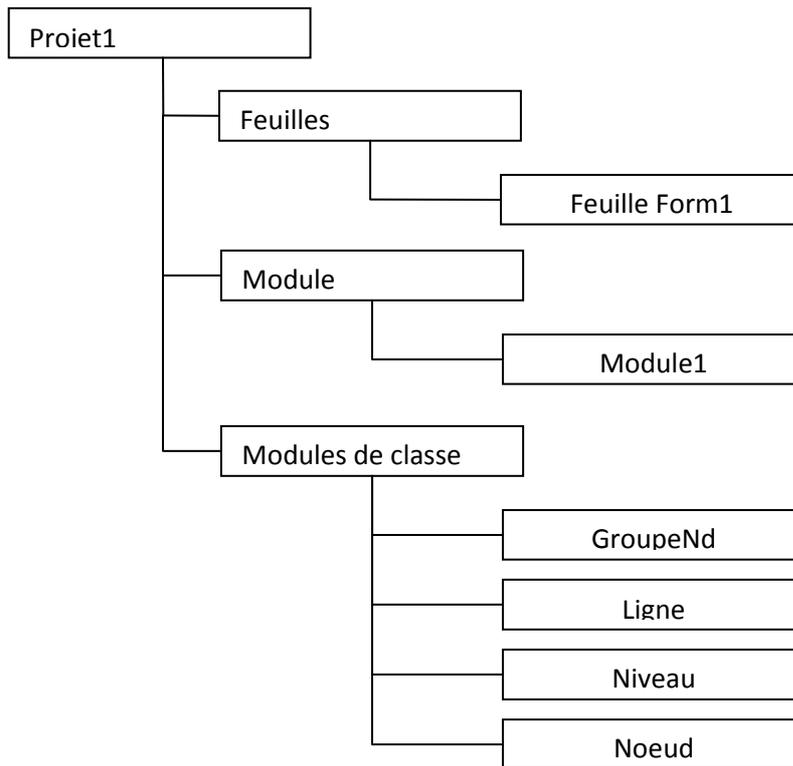


Figure 10 : Structure du projet

VIII.2.1.1. La feuille de travail « form1 »

La feuille « Form1 » contient l'interface graphique et son code, les procédures d'événement des contrôles sur l'interface graphique. (cf Annexe 1)

Le tableau N° 6 montre les procédures d'événement dans le code de la feuille.

Nom de la procédure	Contrôle concerné	Evénement	Fonction
CmdValider_Click	CmdValider	Click	Calcul et affichage des résultats
Option1_Click	Option1	Click	Type de conducteur choisi : cuivre
Option2_Click	Option2	Click	Type de conducteur choisi : aluminium
SubmenuCalculer_Click	SubmenuCalculer	Click	Sortie de résultats
SubmenuNew_Click	SubmenuNew	Click	Initialisation
SubmenuOpen_Click	SubmenuOpen	Click	Ouverture d'un fichier dans un répertoire
SubmenuQuit_Click	SubmenuQuit	Click	Fermeture du programme
SubmenuSave_Click	SubmenuSave	Click	Enregistrement des entrées et des résultats de calcul dans un répertoire
Toolbar1_ButtonClick	Toolbar1	Click	Choix des opérations choisies par l'utilisateur
VScroll1_Change	VScroll1	Change	Déplacement vertical de Picture1

Tableau 7: Liste des procédures dans le code de travail Form1

Les autres procédures se trouvent dans le module1.

**VIII.2.1.2. Le module « Module1 »**

Le module « Module1 » contient les procédures de portée générales du programme et les déclarations des variables publiques.

Le tableau N° 7 montre les procédures générales du programme.

Nom de la procédure	Fonction
Initialisation	Initialise de toutes les variables dans le projet
EntreDonne	Lit et collecte les données sur les nœuds, sur les lignes et les paramètres par défaut
SectionCond	Détermine la section normalisée
IntervertirExtremiteLigne	Intervertisse les extrémités d'une ligne pour la réorganisation du réseau
DefinirBoutdeLigne	Détermine les nœuds bout de ligne du réseau
DefinirNdAmont	Détermine pour chaque nœud du réseau son nœud amont
AutreExtremite	Cherche pour chaque ligne du réseau, les nœuds qui se trouvent à ses extrémités
RenumeroterNd	Renumérote tous les nœuds du réseau selon leur éloignement avec le nœud source, celui-ci a le numéro un
RajoutNdImaginaire	Rajoute un nœud imaginaire de numéro zéro et une ligne de numéro zéro qui seront le nœud en amont de la ligne unique du réseau lorsque celui-ci est transformé en une seule ligne unique et deux nœuds
NouveauGroupe	Crée un nouveau groupe de nœuds lors de la réorganisation du réseau
DefinirGroupeEtNiveau	Répartisse les nœuds du réseau dans des groupes qui sont eux-mêmes répartis dans des niveaux
NouveauNiveau	Crée un nouveau niveau lors de la réorganisation du réseau
TransfoGroupeEn1Ligne	Transforme un groupe de nœuds et sa ligne amont en une seule ligne unique avec un seul nœud en aval
TransformationDirecte	Transforme le réseau de façon à ce que celui-ci ne soit plus qu'une seule ligne unique avec le nœud de numéro zéro en amont et un seul nœud en aval
PuissanceTransmiseLigne	Détermine la puissance transmise par les lignes
TransformationInverse	Détermine les sections des lignes du réseau
CalculSectionCond	Calcule la section admissible de la ligne
SortieDonne	Affiche les résultats du calcul
Ouvrir	Ouvre un fichier dans un répertoire
Enregistrer	Enregistre les données sur les nœuds, les lignes et les paramètres par défaut dans un fichier

Tableau 8: Listes des procédures générales dans le module1

Ces procédures seront développées dans le chapitre suivant.



VIII.2.1.3. Les modules de classe

Le projet contient quatre modules de classe appelés « GroupeNd », « Ligne », « Nœud » et « Niveau ». (cf Annexe 2)

Une classe « GroupeNd » représente un groupe de nœuds du réseau qui ont le même nœud amont.

Ses propriétés sont :

- ✓ NdDeb : le numéro du premier nœud du groupe
- ✓ NdFin : le numéro du dernier nœud du groupe
- ✓ NdAmont : le numéro du nœud commun au groupe
- ✓ GroupeAmont : le numéro du groupe amont du groupe
- ✓ Niveau : le numéro du niveau du groupe

Une classe « Ligne » représente une ligne du réseau. Ses propriétés sont :

- ✓ XDeb, XFin, YDeb, YFin : les coordonnées de la ligne
- ✓ Ro : la valeur de la résistivité du conducteur
- ✓ L : la valeur de la longueur de la ligne
- ✓ Sadm : la valeur de la section admissible des conducteurs de la ligne
- ✓ dU : la valeur de la chute de tension sur la ligne
- ✓ pTransmise : la valeur de la puissance transmise par la ligne

Une classe « Niveau » représente un niveau dans le réseau. Ses propriétés sont :

- ✓ GpDeb : le numéro du premier groupe du niveau
- ✓ GpFin : le numéro du dernier groupe du niveau
- ✓ GpAmont : le numéro du groupe en amont.

Une classe « Nœud » représente un nœud du réseau. Ses propriétés sont :

- ✓ X et Y : les coordonnées du nœud
- ✓ P : la puissance consommée au nœud
- ✓ U : la tension au nœud
- ✓ dUrest : la chute de tension autorisée entre le nœud et les « bout de ligne »
- ✓ NdBoutDeLigne : pour les nœuds bouts de ligne la valeur de cette propriété est « True »
- ✓ NdAmont : le numéro du nœud amont
- ✓ LgnAmont : le numéro de la ligne amont
- ✓ Groupe : le numéro du groupe contenant le nœud
- ✓ Niveau : le numéro du niveau contenant le nœud



- ✓ Remplace : cette propriété prend la valeur « True » si le nœud a été remplacé lors des transformations

VIII.2.2. L'organigramme de la procédure « CmdValider_Click »

Le but de cette procédure « CmdValider_Click » est de calculer les sections normalisées des lignes et de les afficher sur l'interface graphique. Ci-dessous les différentes étapes de la procédure « CmdValider_Click » :

- le programme collecte les informations entrées par l'utilisateur sur les nœuds, les lignes, la chute de tension admissible, la tension nominale, le numéro du nœud source et le type de conducteur à l'aide de la procédure « EntreDonne ».
- il détermine les nœuds bouts de ligne du réseau pour améliorer la connaissance de la topologie du réseau. Les nœuds bout de ligne n'ont pas de nœuds en aval, la connaissance de cette spécificité est essentielle pour la détermination des nœuds amont
- il détermine le nœud amont de chaque nœud du réseau. C'est encore dans le but de connaître la topologie du réseau
- Ces deux propriétés des nœuds connues, les nœuds sont renumérotés selon leur éloignement par rapport au nœud source qui a le numéro un
- il crée un nœud imaginaire de numéro zéro et une ligne imaginaire de numéro zéro. Ce nœud imaginaire est nécessaire lors de la transformation du réseau en réseau équivalent simple principalement dans le cas où le réseau se ramifie dès le nœud source
- Dans le but de réorganiser le réseau, les nœuds renumérotés sont répartis dans des groupes de nœuds qui sont eux-mêmes répartis dans des niveaux
- Le programme détermine la puissance transmise par les lignes avant de transformer le réseau
- En utilisant la topologie du réseau réorganisé, le réseau est transformé de façon qu'il n'y ait qu'une ligne unique et un nœud unique en aval du nœud source.
- le schéma équivalent connu, le programme détermine, en faisant les étapes inverses de la transformation directe, les sections des conducteurs
- il affiche les résultats

Ci-dessous la figure N°11 traduit l'algorithme de la procédure :

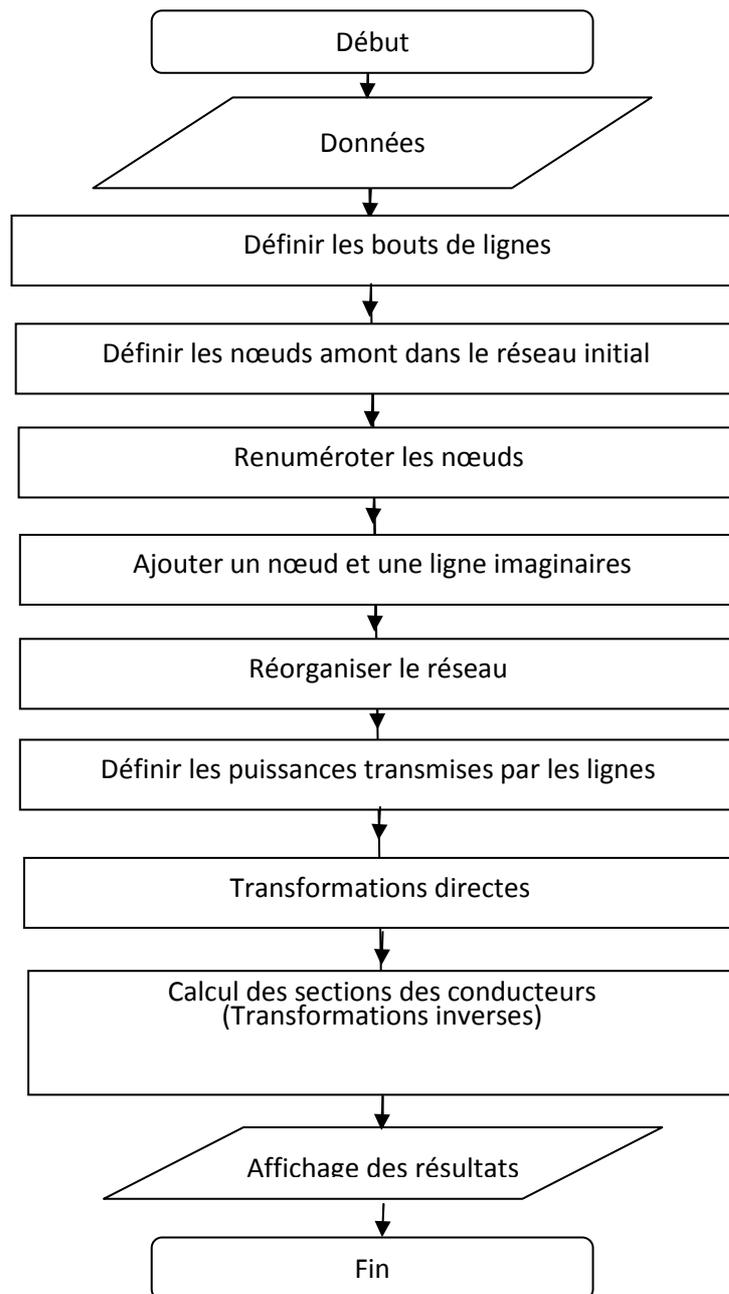


Figure 11: Organigramme de « CmdValider_Click »



CHAPITRE IX : LES PROCEDURES GENERALES DU PROGRAMME

L'organigramme de la procédure « CmdValider_Click » montre les différentes procédures générales appelées pour faire le calcul de la section des conducteurs.

Dans un premier temps, ce chapitre développera les procédures liées à l'organisation du réseau avant les transformations. Ensuite, il parlera de la transformation directe. Et enfin, il traitera la transformation inverse.

IX.1. Les procédures générales liées à l'organisation du réseau.

Ces différentes procédures sont :

- celle pour la détermination des nœuds amont
- celle pour la renumérotation des nœuds
- celle pour la détermination des groupes et des niveaux

IX.1.1. Les procédures pour la détermination des nœuds amont : la procédure « DéfinirNdAmont »

Pour un réseau arborescent, chaque utilisateur ne peut être alimenté que par un seul chemin. Chaque nœud n'a alors qu'un seul nœud qui le précède (nœud amont) du point de vue flux de l'énergie. La connaissance des nœuds amont des nœuds font partie de la connaissance de la topologie du réseau. La connaissance du nœud amont est essentielle pour la renumérotation des nœuds et les futures transformations à faire.

Il est donc nécessaire de connaître les détails de la procédure « DéfinirNdAmont » qui sert pour la détermination du nœud amont d'un nœud. (*cf Annexe 3*)

La figure N°12 suivante traduit l'algorithme de cette procédure :

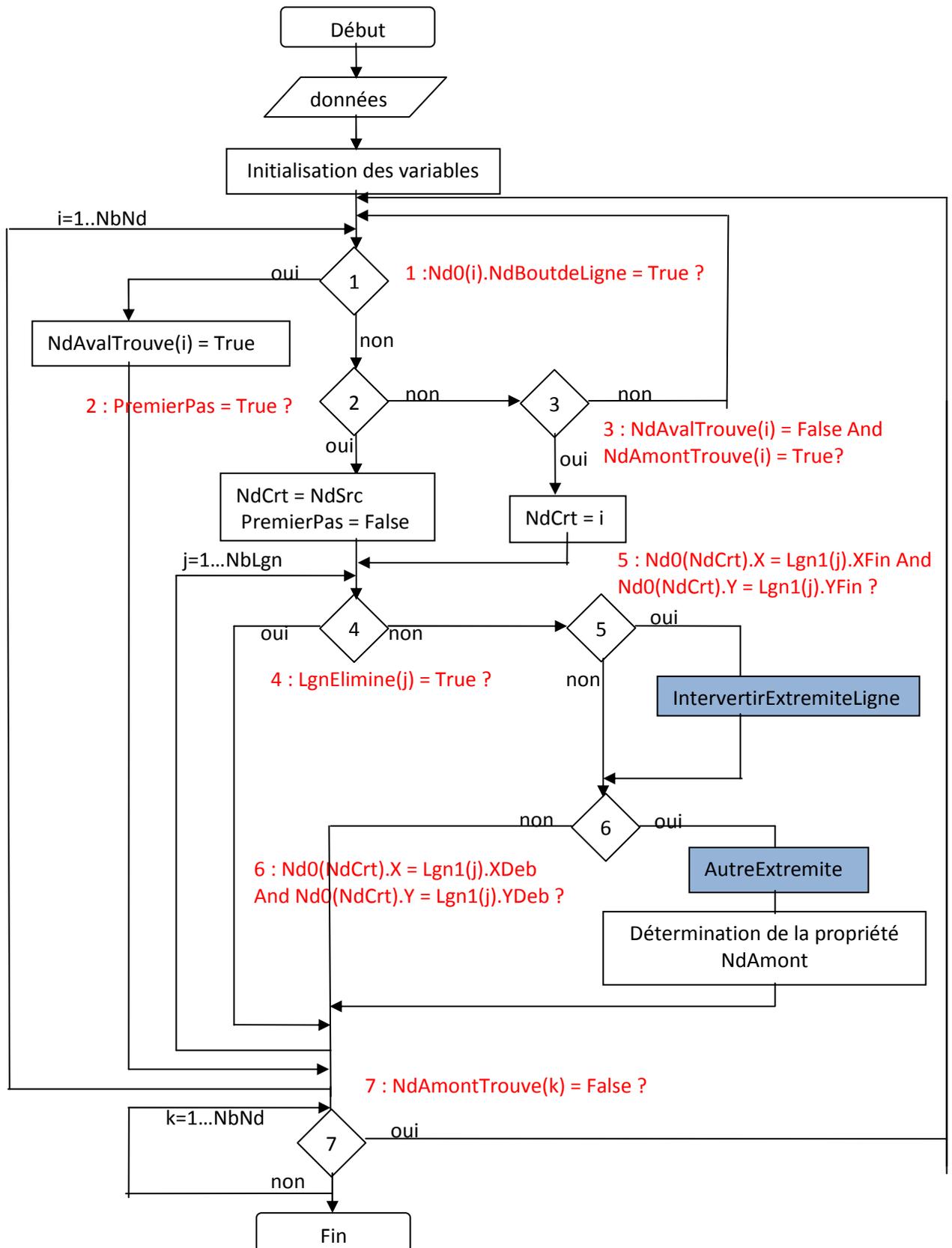


Figure 12: La procédure « DefinirNdAmont »



Les données utilisées sont :

- tous les nœuds du réseau avec leurs propriétés, ils sont déclarés comme une classe Nd0()
- toutes les lignes du réseau avec leurs propriétés, elles sont déclarées comme une classe Lgn1()
- le nombre des nœuds : NbNd
- le nombre des lignes : NbLgn
- le numéro du nœud source : NdSrc.

Les variables locales sont :

- NdAmontTrouve(30) : son rôle est de déterminer si le nœud amont est trouvé
- NdAvalTrouve(30) : pour déterminer si le nœud aval est trouvé
- LgnElimine(30) : cette variable sert à éliminer une ligne dont le nœud amont du nœud aval est connu
- PremierPas : c'est pour le cas spécial du nœud source
- NdCrt : le numéro de nœud

Les variables à initialiser sont les suivantes :

- Pour toutes les lignes du réseau : une classe Lgn1() est créée, celle-ci est une copie de la classe Lgn0() ; la création est nécessaire car cette procédure modifie les propriétés des lignes
- Pour toutes les lignes du réseau : la valeur de LgnElimine() est FALSE car aucune ligne n'est encore éliminée
- Pour tous les nœuds du réseau : la valeur de NdAmontTrouve() est FALSE et celle de Nd0().NdAmont est -1 car aucun nœud amont n'est encore trouvé
- Les valeurs de NdAmontTrouve pour le nœud source et PremierPas sont TRUE car le cas du nœud source n'est pas encore traité

On parcourt tous les nœuds du réseau dans une boucle de paramètre i.

Si Nd0(i).NdBoutdeLigne prend la valeur TRUE, la valeur TRUE est affectée à la variable NdAvalTrouve(i) pour ce nœud i. Dans ce cas, on passe au nœud i suivant.

Pour Nd0(i).NdBoutdeLigne FALSE :

Si la valeur de PremierPas est TRUE, le numéro NdCrt est le numéro NdSrc du nœud source et la valeur de PremierPas est maintenant toujours FALSE. Pour toutes les lignes du réseau, on cherche le nœud qui a les coordonnées de fin de la ligne dont le début est le nœud source. C'est le cas spécial du nœud source.



Une fois le cas du nœud source traité : si la valeur de $NdAvalTrouve(i)$ est FALSE et celle de $NdAmontTrouve(i)$ est TRUE, le numéro $NdCrt$ est i .

On parcourt toutes les lignes du réseau dans une boucle de paramètres j de un à la dernière.

Si la valeur $LgnElimine(j)$ est TRUE, on passe à la ligne j suivante. Sinon, la valeur est « false » et on détermine si les coordonnées du nœud ayant comme numéro $NdCrt$ correspondent à des coordonnées de fin d'une ligne j . Si c'est le cas, la procédure $IntervertirExtremiteLigne$ intervertit les extrémités des lignes dans le but d'organiser le réseau.

Si les coordonnées du nœud ayant comme numéro $NdCrt$ ne sont pas des coordonnées de fin d'une ligne j , alors on détermine si les coordonnées du nœud ayant comme numéro $NdCrt$ sont les coordonnées de début d'une ligne j . Si oui, la procédure $AutreExtremite$ est appelée pour déterminer le nœud à l'autre extrémité de la ligne j . Le nœud amont de ce dernier est le nœud dont le numéro est $NdCrt$.

Les variables suivantes prennent les valeurs respectives :

- $NdAmontTrouve(a) = True$: pour le nœud a , qui est le nœud à l'autre extrémité de la ligne j , le nœud amont a a été trouvé
- $NdAvalTrouve(NdCrt) = True$: le nœud aval de nœud de numéro $NdCrt$ a été trouvé
- $Nd0(a).NdAmont = NdCrt$: le nœud de a est le nœud $NdCrt$
- $LgnElimine(j) = True$: la ligne j est éliminée car le nœud amont pour le nœud ayant les coordonnées de fin est déterminé

Puis, on passe à la ligne j suivante.

Avant de terminer la procédure, le programme effectue un test de vérification qui a pour but de déterminer si pour chaque nœud du réseau, son nœud amont est connu

Si oui, la procédure est terminée.

IX.1.2. La procédure « RenumeroterNd »

Le réseau comporte des nœuds. Ces nœuds sont introduits dans le programme dans un ordre quelconque. Pour une bonne organisation du réseau, les nœuds doivent être renumérotés. Dans la nouvelle numérotation le nœud numéro un est le nœud source.

En partant du nœud source, on renumérote les autres nœuds. Ils seront numérotés selon l'éloignement du nœud source.

La figure N°13 montre l'organigramme de la procédure (*cf Annexe 4 pour le code*) :

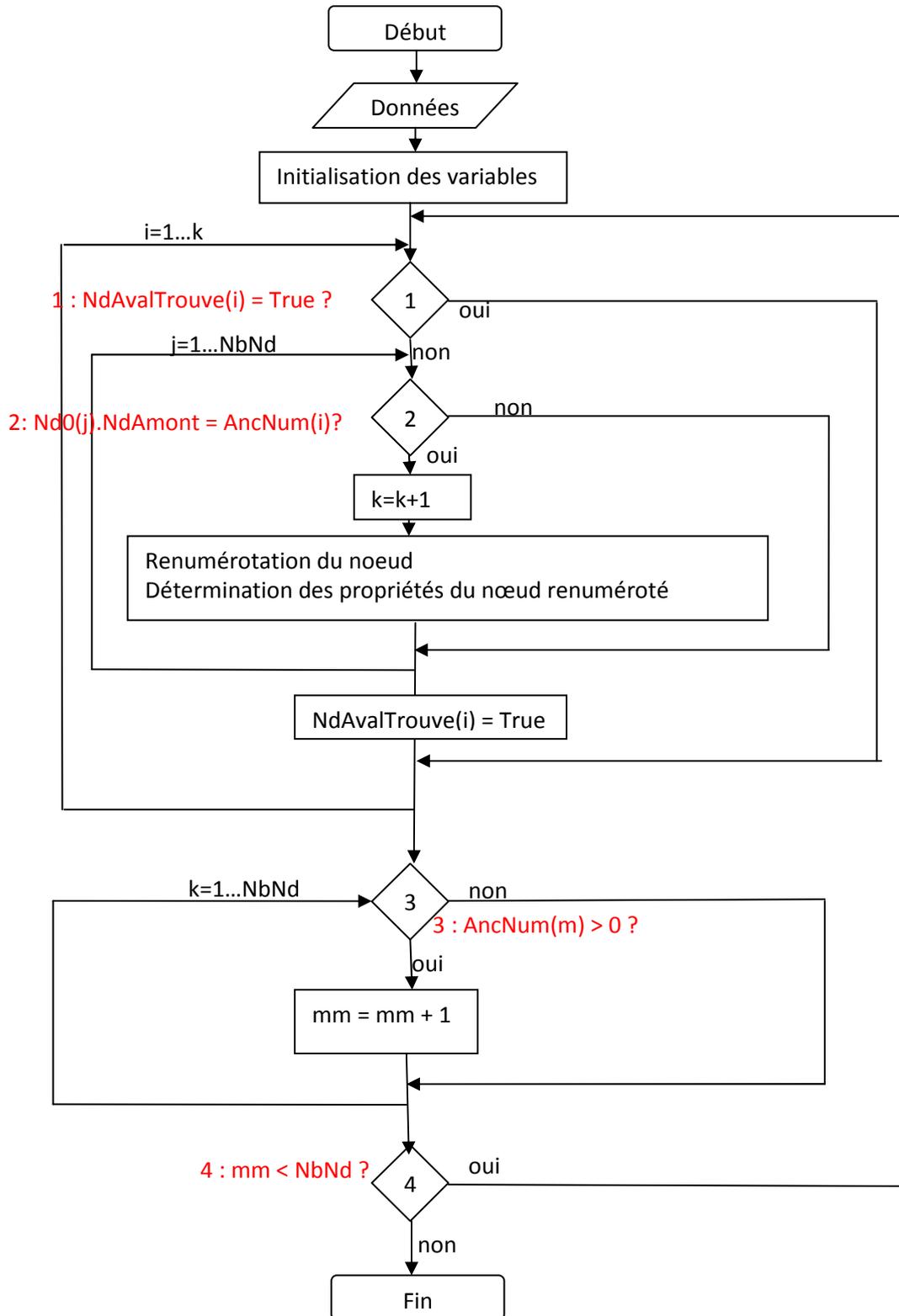


Figure 13 : La procédure« RenumeroterNd »



Les données utilisées sont :

- la classe Nd0() : regroupe tous les nœuds du réseau avec leur numéro initial
- une nouvelle classe Nd() : regroupe tous les nœuds avec leur nouveau numéro
- le nombre de nœuds NbNd,
- le numéro du nœud source NdSrc

Les variables locales sont :

- AncNum(30) : les anciens numéros des nœuds
- NdAvalTrouve(30) : pour déterminer si le nœud aval est trouvé
- NdAmontTrouve(30) : son rôle est de déterminer si le nœud amont est trouvé
- NouvNum(30) : les anciens numéros des nœuds

Les variables à initialiser sont :

- Le numéro du nœud source dans la nouvelle numérotation est 1
- La valeur de la variable k est 1, cette variable k est le nouveau numéro nouvellement créé
- Les propriétés du nœud source dans la classe Nd0 sont copiées vers la classe Nd

On parcourt les nouveaux nœuds dans une boucle de paramètre i de un à la valeur k.

Si la valeur de NdAvalTrouve de i est TRUE pour le nœud de numéro i, on passe au nœud i suivant.

Sinon, on parcourt tous les nœuds du réseau dans la boucle de paramètre j.

Si l'ancien numéro de i est le nœud amont du nœud de numéro j, alors k est augmenté d'une unité.

Ce qui correspond à un nouveau nœud dans la nouvelle numérotation. Le nouveau numéro du nœud j est la valeur de k et le nœud amont de k est le nouveau numéro du nœud amont de j dans l'ancienne numérotation. Les propriétés du nœud k tels que la propriété bout de ligne, la puissance et les coordonnées (x,y) sont les propriétés du nœud j.

Avant de passer à la valeur i suivante, la variable NdAvalTrouve du nœud i devient TRUE.

Avant de terminer la procédure, un test de vérification des nœuds renumérotés est effectué. Le nombre de nœuds renumérotés est déterminé. Si un nœud n'a pas encore été renuméroté alors la procédure ne touche pas à son terme.

Si Le nombre de nœuds renumérotés est le nombre de nœuds du réseau, alors la procédure est terminée.

IX.1.3. La procédure « DéfinirGroupeEtNiveau »

Pour bien présenter la topologie du réseau, les nœuds seront groupés en groupes de nœuds et les groupes de nœuds en niveaux. Un groupe de nœuds est l'ensemble de nœuds qui ont le même nœud amont. Un niveau est l'ensemble de groupes de nœuds ayant même groupe amont. Un niveau est séparé du nœud source par un même nombre de lignes (nœuds).

La figure N°14 suivante montre le fonctionnement de la procédure (cf Annexe 5 pour le code) :

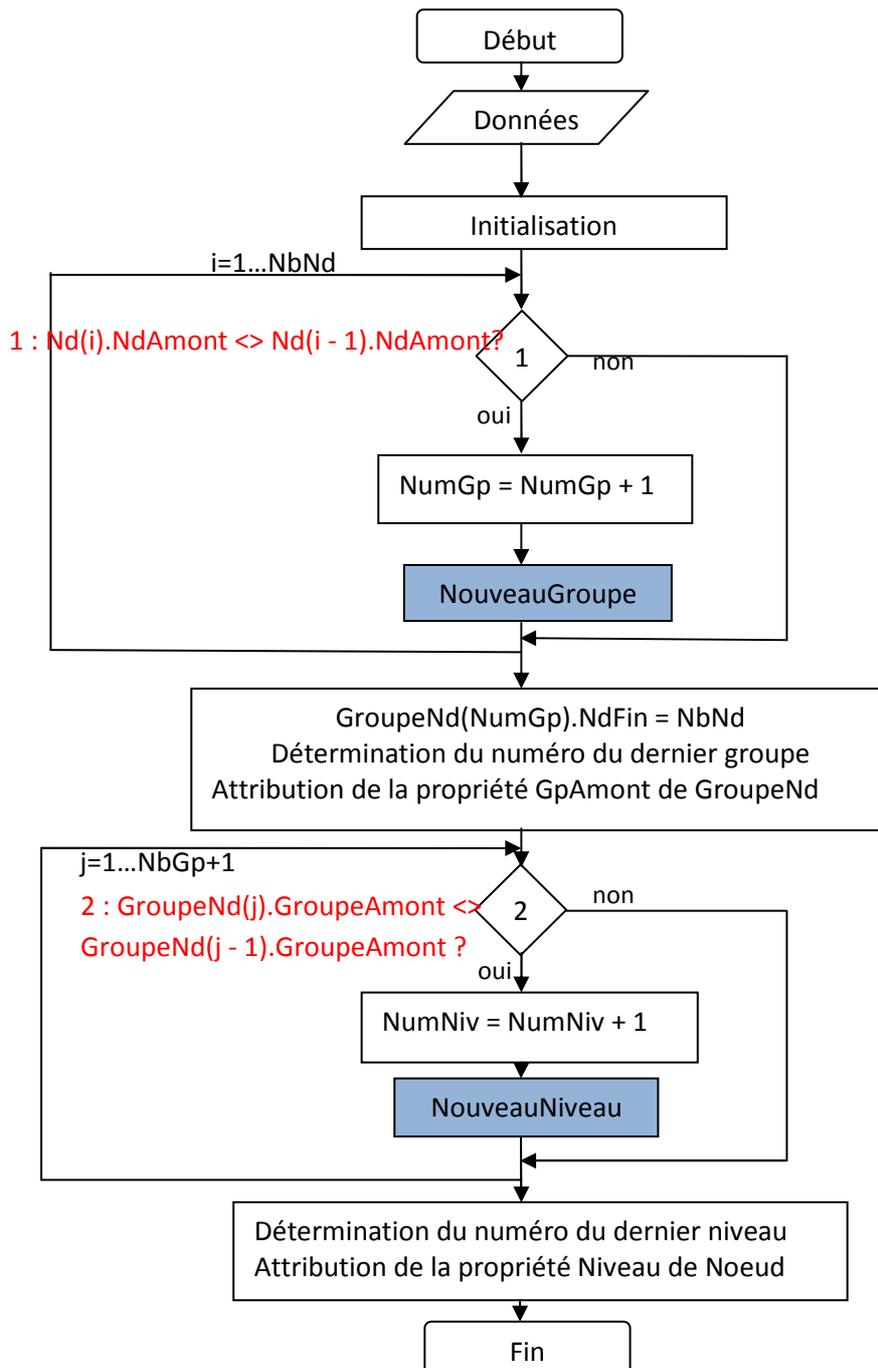


Figure 14 : La procédure « DéfinirGroupeEtNiveau »



Les données utilisées sont :

- la classe GroupeNd() : les groupes de nœuds du réseau
- NivCls() : les niveaux du réseau
- Nd() : les nœuds du réseau
- NbNd : le nombre de nœuds du réseau.

Les variables locales sont les variables :

- NumGp : le numéro du groupe de nœuds
- NumNiv : le numéro du niveau

Les variables à initialiser sont :

- NumGp : la valeur est zéro car aucun groupe n'a encore été créé
- GroupeNd(0) : Le groupe de nœuds du nœud imaginaire de numéro zéro
- NivCls(0) : le niveau du nœud imaginaire de numéro zéro

On parcourt tous les nœuds du réseau dans une boucle de paramètre i pour classer les nœuds dans différents groupes.

Si le nœud amont de deux nœuds successifs est différent, alors la valeur de la variable NumGp est augmentée d'une unité car un nouveau groupe de nœuds est créé. La procédure NouveauGroupe est appelée pour créer un nouveau groupe de nœuds. Sinon, on passe au nœud i suivant.

Une fois que tous les nœuds du réseau sont répartis dans des groupes, on détermine la propriété GroupeAmont pour les GroupeNd().

Pour classer les groupes de nœuds dans des niveaux, il faut parcourir tous les groupes du réseau dans une boucle de paramètre j de un au nombre de groupe dans le réseau NbGp plus un.

Si le groupe amont de deux groupes successifs est différent, alors la valeur de la variable NbNiv est augmentée d'une unité car un nouveau niveau est créé. La procédure NouveauNiveau est appelée pour créer un nouveau niveau. Sinon, on passe au groupe de nœud j suivant.

Une fois que tous les groupes de nœuds sont répartis dans des niveaux, tous les nœuds du réseau reçoivent leur propriété Niveau.

IX.2. Les procédures utilisées pour la transformation directe.

Pour calculer les sections, il est nécessaire de transformer le réseau en un réseau simple. C'est le but de la transformation directe. Celle-ci nécessite l'utilisation de deux procédures générales qui sont :

- une procédure générale pour la transformation directe proprement dite : la procédure « TransformationDirecte »
- une procédure générale pour la transformation d'un groupe de nœuds avec la ligne amont du groupe en un nœud unique avec une seule ligne amont

IX.2.1. La procédure « TransformationDirecte »

La figure N°15 montre l'organigramme de la procédure (cf Annexe 6 pour le code) :

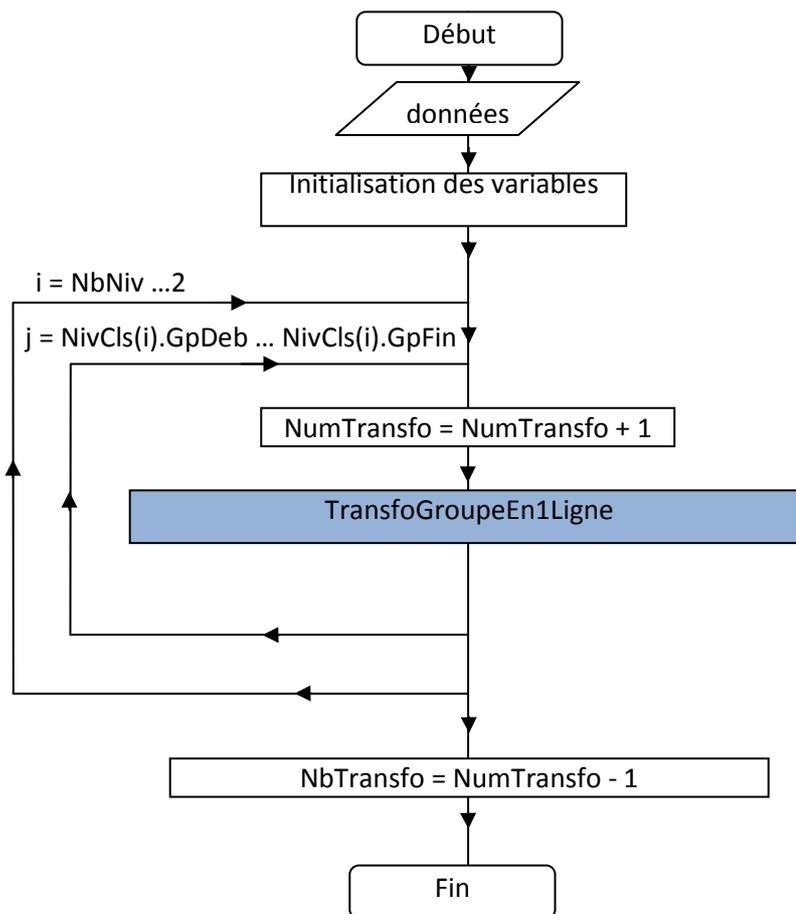


Figure 15 : La procédure « TransformationDirecte »

Les données sont :

- la classe NivCls() : les niveaux du réseau
- NbNd : le nombre de nœuds du réseau.

La variable locale est NumTransfo qui le numéro de transformation du groupe de nœuds et de sa ligne amont en un nœud unique avec une seule ligne amont

Les variables à initialiser sont :

- NumTransfo : la valeur est égale à zéro car il n'y a pas encore eu de transformations

- NbNiv : Le nombre de niveau dont la valeur est le niveau du dernier nœud du réseau
- NdRemplacePar() : la valeur est -1 car le programme considère qu'aucun nœud du réseau n'a encore été remplacé lors d'une transformation

On parcourt tous les niveaux du réseau. Pour chaque groupe d'un niveau, le groupe considéré muni de la ligne amont de celui-ci est transformé en un seul nœud avec une seule ligne amont.

La procédure « TransfoGroupeEn1Ligne » est la procédure qui s'occupe de cette transformation.

IX.2.2. La procédure « TransfoGroupeEn1Ligne »

La procédure « TransfoGroupeEn1Ligne » a pour but de transformer un groupe de nœuds avec la ligne amont du nœud commun au groupe en un nœud unique avec une seule ligne en amont.

L'algorithme de cette procédure est traduit par l'organigramme ci-dessous correspondant à la figure N°16 (cf Annexe 7 pour le code):

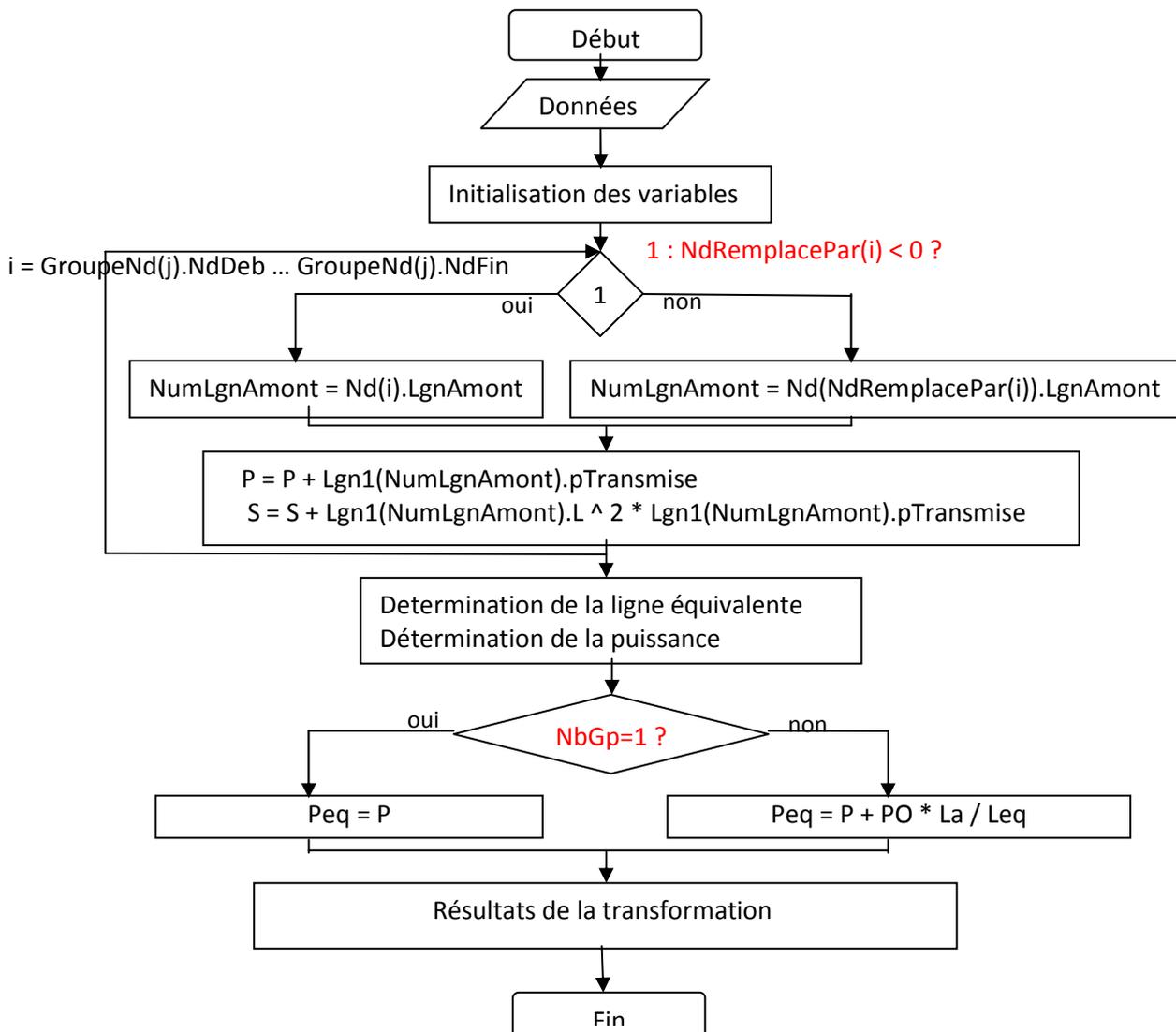


Figure 16: La procédure « TransfoGroupeEn1Ligne »



Les données sont :

- La classe Nd() : les nœuds du réseau
- La classe Lgn1() : les lignes du réseau
- La classe GroupeNd() : les groupes de nœuds du réseau
- NbNd : le nombre de nœuds du réseau,
- NbLgn : le nombre de lignes du réseau,
- Transfo() : la variable définie par l'utilisateur

Les variables locales sont :

- Lambda : le radical d'expression $\sqrt{\frac{\sum l^2_i P_{oi}}{\sum P_{oi}}}$
- S : une somme dont la formule est $\sum l^2_i P_{oi}$
- P : la somme des puissances consommées aux noeuds
- NdCommunGp : le nœud amont du groupe de noeuds
- La : la longueur de la ligne amont du nœud amont du groupe de nœud
- Leq : longueur de la ligne équivalente
- Peq : la puissance équivalente
- PO : la puissance consommée du nœud amont du groupe de nœud

Après l'entrée de données, les variables locales utilisées par la procédure doivent être initialisées. Leur valeur initiale sera zéro.

On parcourt une boucle de paramètre i du premier nœud d'un groupe de nœuds au dernier de celui-ci pour déterminer S et P de ce groupe.

Deux cas peuvent se présenter :

- si le nœud i n'a pas été encore remplacé, alors la valeur de NumLgnAmont est le numéro de la ligne amont du nœud i.
- sinon, c'est le nœud i est déjà un nœud remplacé, alors la valeur de NumLgnAmont est le numéro de la ligne amont du nœud remplacé.

Le numéro du groupe est déterminé par la valeur de l'argument j de la procédure.

Une fois que P et S sont déterminés pour ce groupe de nœuds, on passe à la valeur i suivante.

En utilisant les valeurs de P et S, on détermine les valeurs de la longueur de la ligne équivalente et de la puissance équivalente de ce groupe de nœuds.



Le groupe et la ligne amont du groupe est remplacé par une ligne unique et un nœud unique imaginaire.

Les propriétés du nœud imaginaire créées telles que la puissance et le numéro de la ligne amont sont attribuées. Pour la ligne aussi, les propriétés telles que la puissance transmise et la longueur sont attribuées.

Les caractéristiques de cette transformation sont mémorisées dans la variable `Transfo()`.

Ce sont les propriétés :

- `GroupeTransforme` : le numéro du groupe transformé
- `Leq` : la longueur de la ligne équivalente
- `NdCommun` : le nœud amont pour le groupe de nœud transformé
- `Peq` : la puissance équivalente
- `NumLigneRemplacee` : le numéro de la ligne amont du nœud amont du groupe de nœud transformé

IX.3. La procédure « TransformationInverse ».

Cette procédure permet de calculer les sections des conducteurs des lignes. En partant du réseau simplifié issu de la transformation directe, on détermine les sections du groupe de nœuds remplacé lors de la transformation directe.

Les données utilisées sont :

- la classe `Nd()` : les nœuds du réseau
- la classe `Lgn1()` : les lignes du réseau
- la classe `GroupeNd` : les groupes de noeuds
- `NdRemplacePar()` : cette variable est négative si le nœud considéré n'est pas encore remplacé

Les variables locales sont :

- `iTr` : le numéro de la transformation
- `LL` : le numéro de la ligne amont
- `LgnCommunRemplace` : le numéro de la ligne amont du groupe de noeuds
- `NdCommun` : le numéro du nœud commun au groupe de noeuds
- `Peq` : la puissance équivalente
- `Leq` : la longueur équivalente
- `deltaU` : la chute de tension
- `deltaURest` : la chute de tension restante pour le groupe de nœuds

- k : le numéro du groupe transformé

Ci-dessous l'organigramme de la procédure (cf Annexe 8 pour le code) :

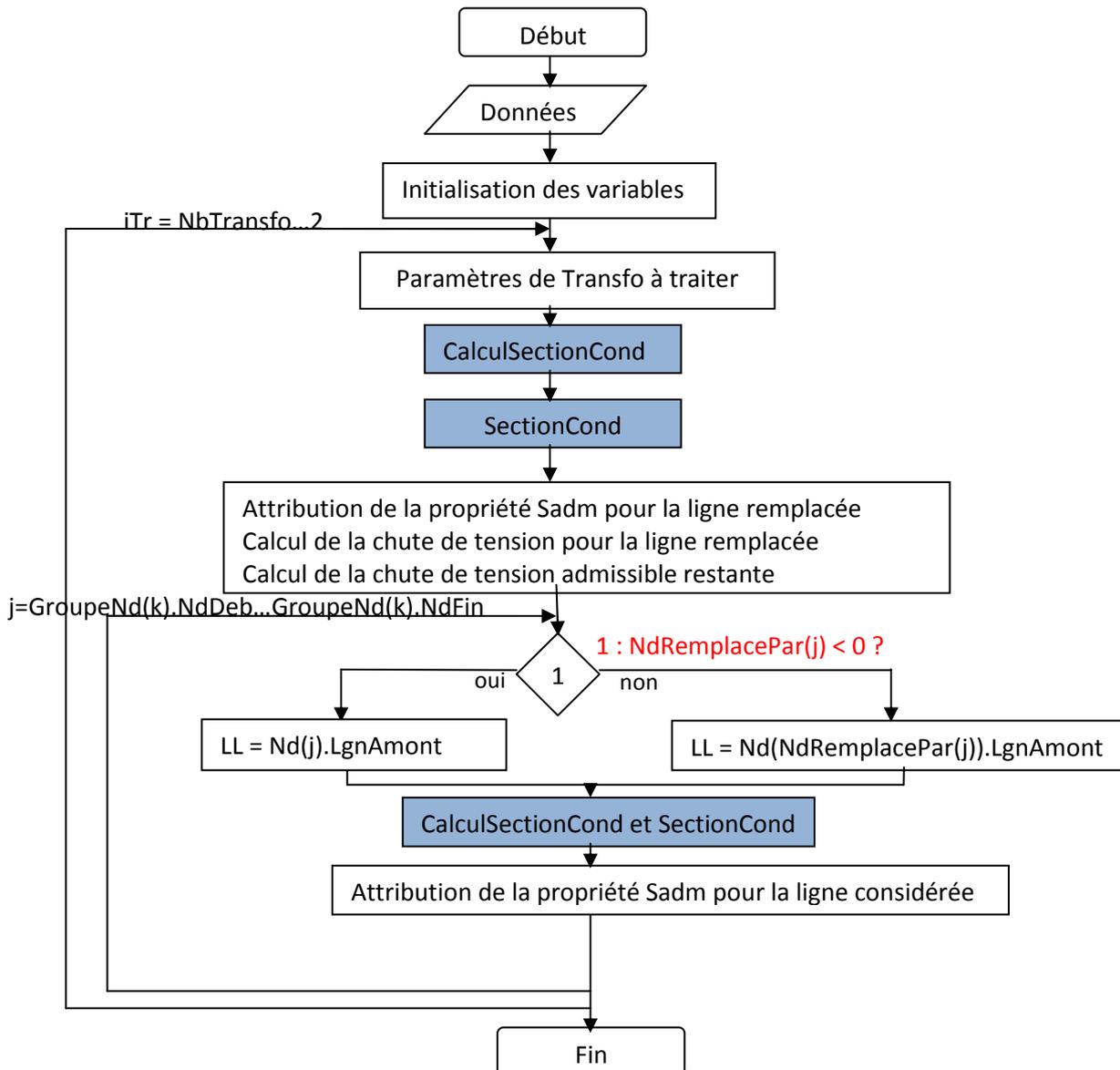


Figure 17 : la procédure « TransformationInverse »

Les variables à initialiser sont :

- dU_{rest} : la valeur de la chute de tension aux nœuds « zéro et un » est la chute de tension maximale admissible dU_{max} du réseau exprimée en Volts



- U : la tension au niveau des nœuds un et zéro est la tension U_{nom}

Pour déterminer la section des lignes des nœuds du groupe et de la ligne amont de celui-ci. On parcourt les transformations effectuées lors de la transformation directe dans une boucle de paramètre iTr de $Nbtransfo$ à un.

Les propriétés de *Transfo* mémorisées dans la procédure « *TransfoGroupeEn1Ligne* » sont utilisées. Ce sont le numéro du groupe transformé k , le nœud amont $NdCommun$ pour le groupe de nœud transformé, le numéro de la ligne amont du nœud amont du groupe de nœud transformé $LgnCommunRemplace$, la puissance équivalente Peq et la longueur de la ligne équivalente Leq .

La procédure *CalculSectionCond* est appelée pour calculer la section de la ligne amont du nœud amont du groupe de nœuds. La valeur issue de cette procédure devient l'argument de la procédure *SectionCond* qui détermine la section normalisée correspondante.

On détermine après la chute de tension sur cette ligne et la chute de tension restante admissible pour tout le groupe de nœuds

Pour déterminer les sections des lignes amont des nœuds du groupe, on parcourt tous les nœuds du groupe de nœuds dans une boucle de paramètre j du premier nœud du groupe jusqu'au dernier.

Si la valeur de $NdRemplacePar()$ pour un j est strictement négative, ce qui veut dire que ce nœud j n'est pas un nœud imaginaire alors la valeur de LL , le numéro de la ligne dont la section est à déterminer, est le numéro de la ligne amont de ce j

La procédure *CalculSectionCond* est appelée pour calculer la section puis la procédure *SectionCond* est appelée pour donner la valeur de la section normalisée correspondante. La propriété de la ligne pour la section est attribuée.

Sinon, la valeur de LL est le numéro de la ligne amont du nœud remplacé lors de la transformation directe.

Les procédures *CalculSectionCond* et *SectionCond* sont appelées pour déterminer la valeur de la section normalisée correspondante. La propriété $Sadm$ de cette ligne est attribuée. Une fois que c'est terminé, on passe à la valeur d' iTr suivante.

CHAPITRE X : ANALYSES DES RESULTATS

Le but de ce présent mémoire étant l'élaboration d'un programme pour le calcul des sections des conducteurs, il est nécessaire de vérifier si le programme est opérationnel à l'aide d'un exemple concret et de déterminer aussi les impacts environnementaux du programme.

X.1. Test de vérification du programme.

Tout programme subit une série de tests avant d'être déclaré opérationnel.

Ci-dessous la figure N°16 d'un schéma d'un réseau ramifié simple :

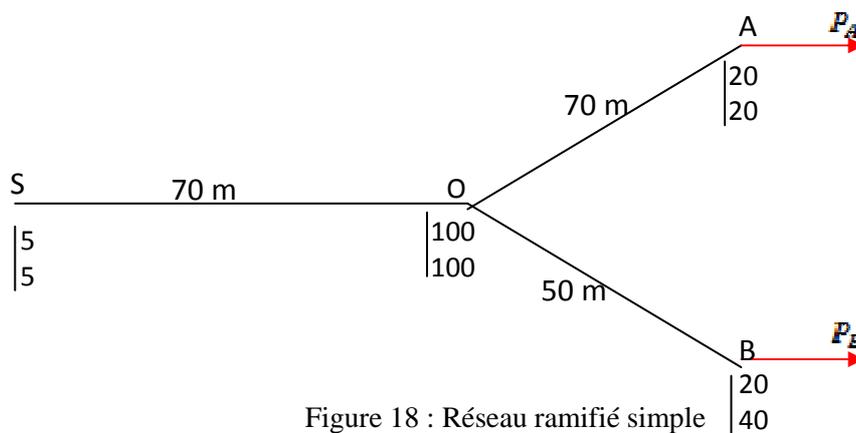


Figure 18 : Réseau ramifié simple

Les données sur les nœuds ainsi que celles sur les lignes sont explicitées dans les tableaux N°8 et N°9 suivants :

Nœud	X	Y	P
A	20	20	20
B	20	40	15
O	100	100	0
S	5	5	0

Tableau 9 : Informations sur les nœuds

Ligne	X1	Y1	X2	Y2	L
OA	100	100	20	20	70
OB	100	100	20	40	50
SO	5	5	100	100	70

Tableau 10 : Informations sur les lignes



X.1.1. Calculs manuels

La longueur équivalente des portions OA et OB :

$$l_{eq} = \sqrt{\frac{l^2_{OA}P_A + l^2_{OB}P_B}{P_A + P_B}} \quad [15]$$

$$l_{eq} = 62 \text{ m}$$

La section normalisée de la portion SO :

$$s_{adm\ SO} = \frac{\rho P_{eq} (l_{SO} + l_{eq})}{U \Delta U_{adm}} \quad [13]$$

Avec :

$$P_{eq} = P_A + P_B = 35 \text{ kW} \quad [14]$$

$$\rho = 36 \Omega \cdot \text{mm}^2 / \text{m}$$

$$\Delta U_{adm} = \Delta U * 0.07 = 15.4 \text{ V}$$

$$l_{SO} = 70 \text{ m}$$

La valeur numérique est :

$$s_{adm\ SO} = 49.09 \text{ mm}^2$$

La valeur de la section admissible étant inférieure mais proche d'une valeur d'une section normalisée, la section normalisée pour la portion SO est $s_{SO} = 70 \text{ mm}^2$

Les sections normalisées des portions OA et OB :

➤ Pour la portion OA

$$s_{adm\ OA} = \frac{\rho l_{OA} P_A}{U_0 \Delta U_{adm\ OA}} \quad [18]$$

Avec :

$$U_0 = U - \frac{\rho l_{SO} (P_A + P_B)}{s_{adm\ SO} U} = 211.82 \text{ V}$$

$$\Delta U_{adm\ OA} = \Delta U_{adm} - \frac{\rho l_{SO} (P_A + P_B)}{s_{adm\ SO} U} = 7.22 \text{ V}$$

La valeur numérique est



$$s_{adm OA} = 32.95 \text{ mm}^2$$

La valeur de la section normalisée correspondante est :

$$s_{OA} = 50 \text{ mm}^2$$

➤ Pour la portion OB

$$s_{adm OB} = \frac{\rho l_{OB} P_B}{U_o \Delta U_{adm OB}} [18]$$

Avec :

$$U_o = U - \frac{\rho l_{SO} (P_A + P_B)}{s_{adm SO} U} = 211.82 \text{ V}$$

$$\Delta U_{adm OB} = \Delta U_{adm} - \frac{\rho l_{SO} (P_A + P_B)}{s_{adm SO} U} = 7.22 \text{ V}$$

La valeur numérique est

$$s_{adm OB} = 17.67 \text{ mm}^2$$

La valeur de la section normalisée correspondante est :

$$s_{OB} = 25 \text{ mm}^2$$

X.1.2. Le test de vérification du programme

Les études théoriques permettent de confronter les valeurs.

Ci-dessous, la figure N°19 qui montre l'interface utilisateur du programme avec vues sur les entrées et la sortie de données

nombre de noeuds : 4				nombre de lignes : 3						
Noeud n°	X	Y	Pi[kW]	Ligne n°	X1	Y1	X2	Y2	L[m]	Section[mm²]
1	20	20	20	1	100	100	20	20	70	50,0
2	20	40	15	2	100	100	20	40	50	25,0
3	100	100	0	3	5	5	100	100	70	70,0
4	5	5	0	4						
5				5						
6				6						
7				7						
8				8						
9				9						

dUmax %	7
Unom [V]	220
N° du noeud source	4
Conducteur	<input type="radio"/> Cuivre <input checked="" type="radio"/> Aluminium

Figure 19 : L'interface graphique pour l'exemple de calcul

Les valeurs sont conformes aux valeurs des études théoriques.

X.2. Les impacts environnementaux.

La situation environnementale de la planète est, aujourd'hui, alarmante. Plusieurs constats montrent l'ampleur du problème. Le réchauffement climatique et plus directement les émissions de gaz à effet de serre sont les plus médiatisés et elles interpellent le plus le public. Pourtant, l'utilisation de l'outil informatique a des impacts non moindres sur l'environnement.

L'élaboration du programme de calcul des sections des conducteurs a nécessité l'utilisation d'outils informatiques.



Pour comprendre les conséquences réelles de l'usage de l'outil informatique sur l'environnement, il est nécessaire d'avoir des notions sur le processus de fabrication, l'utilisation et l'élimination de l'outil informatique.

X.2.1. Impacts pendant le processus de fabrication de l'outil informatique

Deux problèmes importants apparaissent à ce stade du cycle de vie de l'outil informatique.

D'une part, les appareils électroniques utilisent des quantités non négligeables de matières premières. De ce fait, les problèmes environnementaux à ce stade de production sont des problèmes de ressources : le problème de ressources en eau et le problème de ressources en matières premières.

Le problème de ressource en eau :

la fabrication des outils informatiques nécessite l'utilisation d'une très grande quantité d'eau. l'eau utilisable est environ de 0.5 % de l'hydrosphère, or l'utilisation de cet eau n'est pas rationnel non seulement l'eau est rare mais en plus de cela, l'homme fait une mauvaise de gestion de cette eau .

Le problème de ressources en matières premières :

Et d'autre part, les appareils électroniques utilisent de quantités non négligeables de matériaux. En effet, ils sont composés d'un mélange complexe de plusieurs centaines de matériaux dont un grand nombre contiennent des métaux lourds tels que le plomb, le mercure, le cadmium, le béryllium ainsi que des substances chimiques dangereuses, comme les retardateurs de flamme bromés - polybromodiphényles, éthers diphényliques polybromés et tétrabromobisphénol-A.

X.2.2. Impacts pendant l'utilisation de l'outil informatique

L'utilisation de l'outil informatique génère des problèmes de ressources en énergie.

Le problème de ressources en énergie :

La plupart des utilisateurs d'outils informatiques dans le monde ne sont conscient de la dépense énergétique liée à leur utilisation. D'une part, la consommation énergétique de l'Internet croit annuellement de 20%. Ce qui aura pour effet dans un quart de siècle de faire d'internet le premier consommateur d'énergie dans le monde. D'autre part, les infrastructures liées à internet ont une empreinte écologique non négligeable. Ainsi, le milliard de serveurs en service sur la planète consomment d'avantage tant que la demande ne baissera pas. De plus, l'augmentation des débits sur la toile a engendré immédiatement l'augmentation des besoins en climatisation des centraux hébergeant les nœuds de communication des fournisseurs d'accès.



X.2.3. Impacts sur l'élimination de l'outil informatique

C'est le problème majeur à l'heure actuelle. Ils représentent désormais 5% de la totalité des déchets municipaux solides mondiaux, soit presque autant que les emballages plastiques.

Le problème de recyclage :

Compte tenu de leur toxicité, les déchets électroniques sont considérés comme des déchets dangereux. Le recyclage des matériels est la solution utilisée dans le monde mais il est assez complexe et nécessite des intervenants spécialisés. Ce qui a pour effet d'envoyer un grand nombre des outils informatiques dans des centres de stockage. Or, les déchets ont un effet nocif sur la santé des personnes vivant à proximité des centres de stockage.

X.2.4. Impacts sur l'homme

Les appareils électroniques contiennent des substances chimiques dangereuses qui génèrent une forte pollution et des risques sanitaires pour les travailleurs qui les produisent ou les éliminent. L'exposition au plomb et au mercure des enfants et des femmes enceintes est particulièrement préoccupante. Même à faible niveau d'exposition, ces métaux sont extrêmement toxiques et peuvent porter atteinte aux enfants et aux fœtus.

En outre, les utilisateurs risquent d'avoir le syndrome des yeux secs et le trouble du corps humain.



CONCLUSION

Le présent mémoire n'aurait abouti sans les connaissances générales dans le domaine des réseaux électriques en particulier celui des sections des conducteurs. Il est le fruit de la mise en pratique des connaissances en langage de programmation Visual Basic et des méthodes de calcul des sections des conducteurs.

Le programme informatique réalisé a pour but de résoudre le problème de la chute de tension au bout des lignes et de réduire le coût de l'installation électrique.

Le programme facilite le calcul des sections des conducteurs pour les réseaux arborescents complexes. En effet, les calculs faits manuellement sont longs et complexes et les erreurs de calcul ne sont pas rares. Dans ce mémoire, les résultats d'exemple de calcul fait manuellement ont été comparés avec les résultats de l'utilisation du programme. Les résultats sont conformes. L'utilisation du programme donne un gain de temps non négligeable.

Dans le futur, les résultats du travail dans ce mémoire pourront être inclus dans un logiciel d'installations électriques.



BIBLIOGRAPHIE

- [1] : http://fr.wikipedia.org/wiki/réseau_électrique du 1/12/2010
- [2] : http://fr.wikipedia.org/wiki/Ligne_à_haute_tension du 1/12/2010
- [3] : http://fr.wikipedia.org/wiki/Centrale_électriques du 4/12/2010
- [4] : <http://www.schneider-electric.com/corporate/fr/produit-services/cahiers-techniques/réseaux> du 24/12/2010
- [5] : http://www.intersections.schneider-electric.com/La_protection_des_installations_électriques_contre_la_foudre du 29/12/2010
- [6] : http://www.global-download.shneider-electric.com/La_protection_des_circuits du 29/12/2010
- [7] : <http://www.aide-windows.net/périphériques.php> du 31/12/2010
- [8] : <http://ww.techno-science.net/?onglet=glossaire&definition> du 31/12/2010
- [9] : PURET. Christian. « *Les réseaux de distribution publique dans le monde* », Septembre 1991, 28 pages
- [10] : THOMASSET. Georges. « *La conception des réseaux industriels en HT* », Octobre 1993, 24 pages

ANNEXES



ANNEXES

Annexe 1 : Les procédures dans le feuille de travail « form1 »

Annexe 2 : Les modules de classe

Annexe 3 : La procédure « DéfinirNdAmont »

Annexe 4 : La procedure « RenumeroterNd »

Annexe 5 : La procedure « DéfinirGroupeEtNiveau »

Annexe 6 : La procedure « TransformationDirecte »

Annexe 7: La procedure « TransfoGroupeEn1Ligne »

Annexe 8 : La procédure « TransformationInverse »

**Annexe 1 : Les procédures dans le feuille de travail « form1 »**

```
Private Sub CmdValider_Click()
    Call Calcul
End Sub

Private Sub Form_Load()
    Show
    On Error Resume Next
    For i = 1 To 30
        form1.LgnSadmLbl(i).BackColor = &HC0C0C0
    Next
    On Error GoTo 0
    Call Initialisation
    For i = 1 To 30
        Set Nd0(i) = New Noeud
    Next
    For i = 1 To 30
        Set Lgn0(i) = New Ligne
    Next
    For i = 0 To 30
        Set Lgn1(i) = New Ligne
    Next
    For i = 0 To 50
        Set Nd(i) = New Noeud
    Next
    For i = 0 To 30
```



Set GroupeNd(i) = New GroupeNd

Next

For i = 0 To 10

Set NivCls(i) = New Niveau

Next

Snorm(1) = 1.5

Snorm(2) = 2.5

Snorm(3) = 4

Snorm(4) = 6

Snorm(5) = 10

Snorm(6) = 16

Snorm(7) = 25

Snorm(8) = 35

Snorm(9) = 50

Snorm(10) = 70

Snorm(11) = 95

Snorm(12) = 120

Snorm(13) = 150

Snorm(14) = 185

Snorm(15) = 240

Snorm(16) = 300

Snorm(17) = 400

Snorm(18) = 600

Snorm(19) = 630

Toolbar1.ImageList = ImageList1

For i = 1 To 5



```
Toolbar1.Buttons(i).Image = i
Next
End Sub
Private Sub Option1_Click()
    'pour le cuivre
    Ro = 22.5
End Sub
Private Sub Option2_Click()
    'pour l'aluminium
    Ro = 36
End Sub
Private Sub SubmenuCalculer_Click()
    Call Calcul
End Sub
Private Sub SubmenuNew_Click()
    Call Initialisation
End Sub
Private Sub SubmenuOpen_Click()
    Call Initialisation
    Call Ouvrir
End Sub
Private Sub SubmenuQuit_Click()
    End
End Sub
Private Sub SubmenuSave_Click()
    Call Enregistrer
```



End Sub

Private Sub Toolbar1_ButtonClick(ByVal Button As MSComctlLib.Button)

ButtonKey = Button.Key

 Select Case Button.Key

 Case "Nouveau"

 Call Initialisation

 Case "Ouvrir"

 Call Initialisation

 Call Ouvrir

 Case "Enregistrer"

 Call Enregistrer

 Case "Calculer les sections des conducteurs"

 Call Calcul

 Case "Quitter"

 End

 End Select

End Sub

Private Sub VScroll1_Change()

form1.Picture1.Top = -VScroll1 * 10

End Sub



Annexe 2 : Les modules de classe

GroupeNd

Public NdDeb As Integer

Public NdFin As Integer

Public NdAmont As Integer

Public GroupeAmont As Integer

Public Niveau As Integer

Ligne

Public XDeb As Integer, YDeb As Integer

Public XFin As Integer, YFin As Integer

Public Ro As Double

Public L As Double

Public Sadm As Double

Public dU As Double

Public pTransmise As Double

Niveau

Public GpDeb As Integer

Public GpFin As Integer

Public GpAmont As Integer

Nœud

Public X As Integer

Public Y As Integer

Public P As Double



Public U As Double

Public dUrest As Double

Public NdBoutdeLigne As Boolean

Public NdAmont As Integer

Public LgnAmont As Integer

Public Groupe As Integer

Public Niveau As Integer

Public Remplace As Boolean

**Annexe 3 : La procédure « DéfinirNdAmont »**

```
Public Sub DéfinirNdAmont()  
  
    Dim NdAmontTrouve(30) As Boolean  
  
    Dim NdAvalTrouve(30) As Boolean  
  
    Dim LgnElimine(30) As Boolean  
  
    Dim PremierPas As Boolean  
  
    Dim NdCrt  
  
    For i = 1 To NbLgn  
        Set Lgn1(i) = Lgn0(i)  
  
    Next  
  
    For i = 1 To NbLgn  
        LgnElimine(i) = False  
  
    Next  
  
    For i = 1 To NbNd  
        NdAmontTrouve(i) = False  
  
        Nd0(i).NdAmont = -1  
  
    Next  
  
    NdAmontTrouve(NdSrc) = True  
  
    PremierPas = True  
  
    TrouveNdAmont:  
  
    For i = 1 To NbNd  
        If Nd0(i).NdBoutdeLigne = True Then  
            NdAvalTrouve(i) = True GoTo nexti
```



End If

'cas spécial du noeud source

If PremierPas = True Then

NdCrt = NdSrc

PremierPas = False

Else

If NdAvalTrouve(i) = False And NdAmontTrouve(i) = True Then

NdCrt = i

GoTo 10

Else

GoTo nexti

End If

End If

10

For j = 1 To NbLgn

If LgnElimine(j) = True Then GoTo nextj

If Nd0(NdCrt).X = Lgn1(j).XFin And Nd0(NdCrt).Y = Lgn1(j).YFin Then

Call IntervertirExtremiteLigne(Lgn1(j))

End If

If Nd0(NdCrt).X = Lgn1(j).XDeb And Nd0(NdCrt).Y = Lgn1(j).YDeb Then

a = AutreExtremite(j, Nd0(NdCrt).X, Nd0(NdCrt).Y)

NdAmontTrouve(a) = True

NdAvalTrouve(NdCrt) = True

Nd0(a).NdAmont = NdCrt



LgnElimine(j) = True

End If

nextj: Next j

GoTo verif

nexti:

Next i

verif:

For k = 1 To NbNd

If NdAmontTrouve(k) = False Then

GoTo TrouveNdAmont:

End If

Next

End Sub

**Annexe 4 : La procedure « RenumeroterNd »**

Public Sub RenumeroterNd()

'variable locale

Dim AncNum(30) As Integer

Dim NdAvalTrouve(30) As Boolean

Dim NdAmontTrouve(30) As Boolean

Dim NouvNum(30) As Integer

'initialisation

NdSrcIni = NdSrc

NouvNum(NdSrc) = 1

AncNum(1) = NdSrc

Set Nd(1) = Nd0(NdSrc)

k = 1

'determination

.....

'cas du noeud source

ndsreprime = NdSrc

NdSrc = 1

Nd(1).NdBoutdeLigne = False

Nd(1).LgnAmont = 0

Nd(1).P = Nd0(ndsreprime).P

Nd(1).X = Nd0(ndsreprime).X

Nd(1).Y = Nd0(ndsreprime).Y



'mémorisation de l'ancien numéro du noeud source

NouvNum(ndsrcprime) = 1

AncNum(1) = ndsrcprime

.....

k = 1

boucle1:

For i = 1 To k

If NdAvalTrouve(i) = True Then GoTo nexti

For j = 1 To NbNd

If Nd0(j).NdAmont = AncNum(i) Then

k = k + 1

NouvNum(j) = k

AncNum(k) = j

Nd(k).LgnAmont = Nd0(j).LgnAmont

a = Nd0(j).NdAmont

Nd(k).NdAmont = NouvNum(a)

'attribution des propriétés des noeuds nd() Nd(k).NdBoutdeLigne =
Nd0(j).NdBoutdeLigne

Nd(k).X = Nd0(j).X

Nd(k).Y = Nd0(j).Y

Nd(k).P = Nd0(j).P

.....

End If

Next j

NdAvalTrouve(i) = True



nexti:

Next i

mm = 0

For m = 1 To NbNd

 If AncNum(m) > 0 Then mm = mm + 1

Next

If mm < NbNd Then GoTo boucle1 End Sub

**Annexe 5 : La procedure « DéfinirGroupeEtNiveau »**

```
Public Sub DéfinirGroupeEtNiveau()

Dim NumGp As Integer

Dim NumNiv As Integer

Dim i As Integer

Dim j As Integer

'determination des groupes de noeuds

NumGp = 0

GroupeNd(0).NdDeb = 0

GroupeNd(0).NdFin = 0

NivCls(0).GpDeb = 0

NivCls(0).GpFin = 0

For i = 1 To NbNd

    If Nd(i).NdAmont <> Nd(i - 1).NdAmont Then

        NumGp = NumGp + 1

        Call NouveauGroupe(i, NumGp)

    End If

Next

GroupeNd(NumGp).NdFin = NbNd

For m = GroupeNd(NumGp).NdDeb To GroupeNd(NumGp).NdFin

    Nd(m).Groupe = NumGp

Next

'determination de la propriété Gpamont
```



```
NbGp = NumGp
For i = 1 To NbGp
    GroupeNd(i).GroupeAmont = Nd(Nd(GroupeNd(i).NdDeb).NdAmont).Groupe
Next i
'determination du niveau des groups
GroupeNd(0).GroupeAmont = -1
NumNiv = 0
For j = 1 To NbGp + 1
    If GroupeNd(j).GroupeAmont <> GroupeNd(j - 1).GroupeAmont Then
        NumNiv = NumNiv + 1
        Call NouveauNiveau(j, NumNiv)
    End If
Next j
For n = 1 To NbGp
    For k = NivCls(n).GpDeb To NivCls(n).GpFin
        GroupeNd(k).Niveau = n
    Next k
Next n
'determination du niveau d'un nœud
For i = 1 To NbNd
    Nd(i).Niveau = GroupeNd(Nd(i).Groupe).Niveau
Next
End Sub
```

**Annexe 6 : La procedure « TransformationDirecte »**

```
Public Sub TransformationDirecte()  
  
Dim NumTransfo As Integer  
  
Dim j As Integer  
  
NumTransfo = 0  
  
NbNiv = Nd(NbNd).Niveau  
  
For k = 0 To NbNd  
  
    NdRemplacePar(k) = -1  
  
    Next  
  
'determination de la transformation directe  
  
For i = NbNiv To 2 Step -1  
  
    For j = NivCls(i).GpDeb To NivCls(i).GpFin  
  
        NumTransfo = NumTransfo + 1  
  
        Call TransfoGroupeEn1Ligne(j, NumTransfo)  
  
    Next  
  
Next  
  
NbTransfo = NumTransfo - 1  
  
End Sub
```

**Annexe 7: La procedure « TransfoGroupeEn1Ligne »**

Public Sub TransfoGroupeEn1Ligne(j As Integer, NumTransfo As Integer)

Dim Lambda As Double

Dim S As Double

Dim P As Double

Dim NdCommunGp As Integer

Dim La As Double

Dim Leq As Double

Dim Peq As Double

Dim PO As Double

'initialisation

lamda = 0

S = 0

P = 0

NdCommunGp = 0

La = 0

Leq = 0

Peq = 0

PO = 0

'determination de S et P

For i = GroupeNd(j).NdDeb To GroupeNd(j).NdFin

 If NdRemplacePar(i) < 0 Then

 NumLgnAmont = Nd(i).LgnAmont



Else

$$\text{NumLgnAmont} = \text{Nd}(\text{NdRemplacePar}(i)).\text{LgnAmont}$$

End If

$$P = P + \text{Lgn1}(\text{NumLgnAmont}).p\text{Transmise}$$

$$S = S + \text{Lgn1}(\text{NumLgnAmont}).L^2 * \text{Lgn1}(\text{NumLgnAmont}).p\text{Transmise}$$

Next

$$\text{Lambda} = \text{Sqr}(S / P)$$

$$L_a = \text{Lgn1}(\text{Nd}(\text{GroupeNd}(j)).\text{NdAmont}).\text{LgnAmont}).L$$

$$L_{eq} = L_a + \text{Lambda} \text{ 'Puissance équivalente}$$

$$P_O = \text{Nd}(\text{GroupeNd}(j)).\text{NdAmont}).P$$

If NbGp = 1 Then

$$P_{eq} = P$$

Else

$$P_{eq} = P + P_O * L_a / L_{eq}$$

End

'Résultat de la transformation

$$\text{NdCommunGp} = \text{GroupeNd}(j).\text{NdAmont}$$

$$\text{Nd}(\text{NdCommunGp}).\text{Remplace} = \text{True}$$

$$\text{NdRemplacePar}(\text{NdCommunGp}) = \text{NbNd} + \text{NumTransfo}$$

$$ii = \text{NbNd} + \text{NumTransfo}$$

$$jj = \text{NbLgn} + \text{NumTransfo}$$

$$\text{Nd}(ii).\text{LgnAmont} = \text{NbLgn} + \text{NumTransfo}$$

$$\text{Nd}(ii).P = P_{eq}$$

$$\text{Lgn1}(jj).p\text{Transmise} = P_{eq}$$



Lgn1(jj).L = Leq

Transfo(NumTransfo).GroupeTransforme = j

Transfo(NumTransfo).Leq = Leq

Transfo(NumTransfo).NdCommun = NdCommunGp

Transfo(NumTransfo).Peq = Peq

Transfo(NumTransfo).NumLigneRemplacee = Nd(NdCommunGp)

End Sub

**Annexe 8 : La procédure « TransformationInverse »**

Public Sub TransformationInverse()

Dim iTr As Integer

Dim LL As Double

Dim LgnCommunRemplace As Integer

Dim NdCommun As Integer

Dim Peq As Double

Dim Leq As Double

Dim deltaU As Double

Dim deltaURest As Double

Dim k As Integer

'initialisation

Nd(1).dUrest = (dUmax / 100) * Unom

Nd(0).dUrest = (dUmax / 100) * Unom

Nd(1).U = Unom

Nd(0).U = Unom

For iTr = NbTransfo To 1 Step -1

 k = Transfo(iTr).GroupeTransforme

 NdCommun = Transfo(iTr).NdCommun

 LgnCommunRemplace = Transfo(iTr).NumLigneRemplacee

 Leq = Transfo(iTr).Leq

 Peq = Transfo(iTr).Peq

 Call CalculSectionCond(Ro, Peq, Leq, Nd(Nd(NdCommun).NdAmont).U,
Nd(Nd(NdCommun).NdAmont).dUrest, Sadm)



a = SectionCond(Sadm)

Lgn1(LgnCommunRemplace).Sadm = a

deltaU = (Ro * Peq * Lgn1(LgnCommunRemplace).L) / (Nd(Nd(NdCommun).NdAmont).U
* Sadm)

Nd(NdCommun).U = Nd(Nd(NdCommun).NdAmont).U - deltaU

deltaURest = Nd(Nd(NdCommun).NdAmont).dUrest - deltaU

Nd(NdCommun).dUrest = deltaURest

For j = GroupeNd(k).NdDeb To GroupeNd(k).NdFin

 If NdRemplacePar(j) < 0 Then

 LL = Nd(j).LgnAmont

 Call CalculSectionCond(Ro, Nd(j).P, Lgn1(LL).L,
Nd(NdCommun).U, Nd(NdCommun).dUrest, Sadm)

 b = SectionCond(Sadm)

 Lgn1(LL).Sadm = b

 Else:

 LL = Nd(NdRemplacePar(j)).LgnAmont

 Call CalculSectionCond(Ro, Nd(NdRemplacePar(j)).P, Lgn1(LL).L,
Nd(NdCommun).U, Nd(NdCommun).dUrest, Sadm)

 c = SectionCond(Sadm)

 Lgn1(LL).Sadm = c

 End If

Next

Next

End Sub



TABLE DES MATIERES

REMERCIEMENTS	i
LISTE DES TABLEAUX.....	ii
LISTE DES FIGURES.....	iii
LISTE DES ACRONYMES	iv
SOMMAIRE	vi
INTRODUCTION	1
CHAPITRE I: GENERALITES SUR LES RESEAUX ELECTRIQUES	2
I.1. Historique	2
I.2. La classification des réseaux	3
I.2.1. Classification selon la fonction	4
I.2.1.1. Les réseaux de transport	4
I.2.1.2. Les réseaux de répartition.....	4
I.2.1.3. Les réseaux de distribution.....	4
I.2.2. Classification selon le type de courant.....	5
I.2.3. Classification selon l'architecture du réseau.....	5
I.2.4. Classification selon la tension nominale.....	6
I.3. Notions importantes sur les réseaux électriques.....	6
I.3.1. Les pertes en lignes par effet Joule	6
I.3.2. La chute de tension	7
CHAPITRE II : LES PRINCIPAUX COMPOSANTS DES RESEAUX ELECTRIQUES	8
II.1. Les unités de production.....	8
II.1.1. Les différents types de centrales	9
II.1.1.1. Les centrales thermiques.....	9
II.1.1.2. Les centrales utilisant les énergies renouvelables	10
II.2. Les ouvrages ou postes.....	12
II.2.1. Le poste HT/MT en distribution publique	12
II.2.2. Le poste MT/MT en distribution publique.....	12
II.2.3. Le poste MT/BT en distribution publique.....	12
II.3. Les matériels.....	13
II.3.1. Les lignes à haute tension	13
II.3.1.1. Pylônes.....	13
II.3.1.2. Les conducteurs	13
II.3.1.3. Isolateurs et câbles de garde	14
II.3.2. Les appareillages.....	14
II.3.2.1. La fonction des appareillages	14
II.3.2.2. Les principaux appareillages utilisés	15



CHAPITRE III : PROTECTIONS DES RESEAUX ELECTRIQUES	16
III.1. Les technologies de protection	16
III.2. Les matériels de conduite et de surveillance utilisés	17
III.2.1. La télésurveillance	17
III.2.2. La télécommande	17
III.2.3. La téléconduite	17
CHAPITRE IV : CONSTITUTION D'UN ORDINATEUR	18
IV.1. La partie physique de l'ordinateur	18
IV.1.1. Le microprocesseur	18
IV.1.2. Les périphériques	18
IV.1.2.1. Les périphériques d'entrées et de sortie	18
IV.1.2.2. Les périphériques externe et interne	19
IV.1.2.3. Le mode de communication des périphériques	19
IV.2. Partie logicielle de l'ordinateur	20
IV.2.1. Le type de système d'exploitation : Windows XP	20
IV.2.2. Les logiciels installés	21
IV.2.2.1. Microsoft Office 2007	21
IV.2.2.2. Microsoft Visual Studio 6.0	21
CHAPITRE V : LE LANGAGE DE PROGRAMMATION VISUAL BASIC 6.0	23
V.1. Les contrôles	23
V.1.1. Les propriétés	25
V.1.2. Les méthodes	26
V.1.3. Les évènements	26
V.2. Les variables	26
V.2.1. Les différents types de données	26
V.2.2. Les déclarations des variables	27
V.2.3. Les classes	27
V.3. Les procédures	27
V.3.1. Les procédures de type SUB	27
V.3.2. Les procédures de type FUNCTION	28
V.3.3. Les instructions pour appeler les procédures	28
V.4. Les instructions	28
V.4.1. Les structures	28
V.4.2. Les autres instructions	31
CHAPITRE VI : CALCUL DE LA SECTION D'UN CONDUCTEUR	32
VI.1. Calcul des sections des conducteurs d'une ligne	32
VI.2. Calcul des sections des conducteurs pour un réseau ramifié simple	33
VI.3. Calcul des sections des conducteurs dans un réseau arborescent complexe	35



VI.3.1. Règle des moments des puissances	35
VI.3.2. Transformation directe	36
VI.3.3. Transformation inverse	38
CHAPITRE VII : LES DIFFERENTES ETAPES D'UNE PROGRAMMATION	39
VII.1. Spécifications des exigences et besoins des futurs utilisateurs	39
VII.2. Conception générale du programme.....	39
VII.3. Conception détaillée du programme.....	39
VII.4. Les diverses vérifications	40
VII.4.1. Le test unitaire	40
VII.4.2. Le test d'intégration.....	40
CHAPITRE VIII : PRESENTATION GENERALE DU PROGRAMME.....	41
VIII.1. La description de l'interface utilisateur.....	41
VIII.1.1. Le menu et la barre d'outils.....	41
VIII.1.1.1. Le bouton « Nouveau ».....	42
VIII.1.1.2. Le bouton « Ouvrir »	42
VIII.1.1.3. Le bouton « Enregistrer »	42
VIII.1.1.4. Le bouton « calculer les sections des conducteurs »	42
VIII.1.1.5. Le bouton « Quitter ».....	42
VIII.1.2. Les diverses entrées de données et la sortie des résultats de calcul	42
VIII.1.2.1. Données concernant les nœuds	43
VIII.1.2.2. Données concernant les lignes	44
VIII.1.2.3. Données concernant les paramètres par défaut.....	44
VIII.1.2.4. Données concernant la sortie des résultats	45
VIII.1.3. Barre de défilement	45
VIII.2. Le fonctionnement général du programme	45
VIII.2.1. La structure du projet	45
VIII.2.1.1. La feuille de travail « form1 »	46
VIII.2.1.2. Le module « Module1 ».....	47
VIII.2.1.3. Les modules de classe	48
VIII.2.2. L'organigramme de la procédure « CmdValider_Click ».....	49
CHAPITRE IX : LES PROCEDURES GENERALES DU PROGRAMME.....	51
IX.1. Les procédures générales liées à l'organisation du réseau	51
IX.1.1. Les procédures pour la détermination des nœuds amont : la procédure «	51
IX.1.2. La procédure « RenumeroterNd ».....	54
IX.1.3. La procédure « DefinirGroupeEtNiveau ».....	57
IX.2. Les procédures utilisées pour la transformation directe	58
IX.2.1. La procédure « TransformationDirecte ».....	59



IX.2.2. La procédure « TransfoGroupeEn1Ligne »	60
IX.3. La procédure « TransformationInverse »	62
CHAPITRE X : ANALYSES DES RESULTATS	65
X.1. Test de vérification du programme	65
X.1.1. Calculs manuels	66
X.1.2. Le test de vérification du programme	68
X.2. Les impacts environnementaux	68
X.2.1. Impacts pendant le processus de fabrication de l'outil informatique.....	69
X.2.2. Impacts pendant l'utilisation de l'outil informatique.....	69
X.2.3. Impacts sur l'élimination de l'outil informatique	70
X.2.4. Impacts sur l'homme.....	70
CONCLUSION	71
BIBLIOGRAPHIE	72
ANNEXES	I
TABLE DES MATIERES	103

Auteur : RASAMIARINDRAINY Mika Ambinintsoa

Tél : 033 01 384 07, **e-mail :** miiik27a@yahoo.fr

Adresse : Lot VO 8 Bis J Miandrarivo Antananarivo 101

Titre du mémoire :

« ELABORATION D'UN PROGRAMME POUR DETERMINER LA NATURE ET LA
SECTIONS DES CONDUCTEURS DANS UN RESEAU ARBORESCENT POUR DIMINUER
LA CHUTE DE TENSION MAXIMALE »

Nombre de pages : 71 pages

Nombre de tableaux : 9

Nombre de figures : 19

Résumé

Pour résoudre le problème des chutes de tension aux bouts des lignes et pour diminuer le coût des installations électriques, un programme de calcul de la section des conducteurs a été réalisé.

Etant donné la place de l'informatique dans le monde actuel, la connaissance de plusieurs langages de programmation est indispensable.

Ce programme a nécessité la connaissance du langage de programmation Visual Basic ainsi que les théories de calcul des sections des conducteurs.

Ce présent mémoire servira de support pédagogique pour les étudiants et d'outils de travail pour les techniciens.

Abstract

To solve the problem of voltage drops at the ends of lines and decrease the cost of electrical installations, a program for calculating the conductor has been achieved.

Given the role of informatics in today's world, knowledge of multiple programming languages is essential.

This program required the knowledge of the programming language Visual Basic theories and calculating of conductor cross sections.

What this memo will serve as a teaching aid for students and working tools for technicians.

Mots clés : Réseau électrique, arborescent, chute de tension, section des conducteurs, programmation visual basic

Directeur de mémoire : Nathanaël RAVELOMANANA