



Escola Politècnica Superior
d'Enginyeria de Vilanova i la Geltrú

UNIVERSITAT POLITÈCNICA DE CATALUNYA

PROJECTE FI DE CARRERA

TÍTOL: Programa de càlcul del flux de càrregues (DaDa System) per sistemes elèctrics de potència. Part 1.

AUTOR: Damià Arnau Pont Pons

TITULACIÓ: Enginyeria Tècnica Industrial. Especialitat en electricitat.

DIRECTOR: Enric Ferrer Bardem

DEPARTAMENT: Enginyeria Elèctrica

DATA: 18 de Gener del 2010

TÍTOL: Programa de càlcul del flux de càrregues (DaDa System) per sistemes elèctrics de potència. Part 1.

COGNOMS: Pont Pons

NOM: Damià Arnau

TITULACIÓ: Enginyeria Tècnica Industrial

ESPECIALITAT: Electricitat

PLA: 95

DIRECTOR: Enric Ferrer Bardem

DEPARTAMENT: Enginyeria Elèctrica

QUALIFICACIÓ DEL PFC

TRIBUNAL

PRESIDENT	SECRETARI	VOCAL
JOSEP FONT MATEU	EUSEBIO MARTINEZ PIERA	INMACULADA MASSANA HUGAS

DATA DE LECTURA: 08/02/2010

Aquest Projecte té en compte aspectes mediambientals: Sí No

PROJECTE FI DE CARRERA

RESUM (màxim 50 línies)

L'objectiu principal d'aquest projecte consta de dues parts. En primer lloc consta de la introducció teòrica al flux de càrregues i com aquest es pot resoldre per diferents mètodes iteratius i en segon lloc el desenvolupament d'un software de calcul del mateix flux de càrregues.

En la primera part constara de una introducció inicial així com una valoració de la utilitat general del flux de carregues. Posteriorment es dara un estudi dels diferents elements tant de barres (Slack, PQ, PV i PQV) com d'elements de connexió (línies i transformadors). En darrer lloc s'explicaran varis sistemes de resolució del flux de càrregues(Gauss-Seidel, Newton-Raphson i Newton-Raphson modificat), tot i que es farà major incís en la resolució del sistema per mitja de Newton-Raphson. Aquest sistema, tot i la seva complicació inicial, ja que treballa amb matrius i derivades parcials, és el sistema que obté uns resultats més òptims amb les mínimes iteracions.

La segona part constara de la resolució teòrica dels càlculs necessaris per el desenvolupament del sistema de Newton-Raphson en un entorn informàtic. En aquesta part també es desenvolupara un software per mitja de Turbo Delphi, programa orientat a objectes que combina diferents llenguatges de programació.

Paraules clau (màxim 10):

Flux de càrregues	Barres	Línies	Transformador
Turbo Delphi	Newton-Raphson	Iteració	

1. INTRODUCCIÓ.....	1
1.1. INTRODUCCIÓ AL FLUX DE CÀRREGUES.....	1
1.1.1 UTILITATS DEL FLUX DE CÀRREGUES.....	1
1.1.2 RESOLUCIÓ I CONSIDERACIONS.....	2
1.2. TIPUS D'ELEMENTS EN UN SISTEMA ELÈCTRIC DE POTÈNCIA.....	4
1.2.1. TIPUS DE CONNEXIONS DE XARXA.....	4
1.2.1.1. MODELS DE LÍNIES	4
1.2.1.2. MODELS DE TRANSFORMADORS.....	5
1.2.2. TIPUS DE BUSOS.....	7
1.2.2.1. SLACK BUS.....	7
1.2.2.2. BUS DE CÀRREGA.....	7
1.2.2.3. BUS DE TENSIÓ CONTROLADA.....	7
1.3. RESOLUCIÓ DEL PROBLEMA.....	8
1.3.1. EQUACIONS.....	8
1.3.2. MÈTODES DE RESOLUCIÓ.....	9
1.3.2.1. MÈTODE DE GAUSS-SEIDEL.....	9
1.3.2.2. MÈTODE DE NEWTON-RAPHSON.....	10
1.3.2.3. MÈTODE DE NEWTON-RAPHSON MODIFICAT.....	14
2. PROGRAMA DE CÀLCUL.....	15
2.1. INTRODUCCIÓ AL PROGRAMADOR	15
2.1.1. PROGRAMACIÓ ORIENTADA A OBJECTES.....	15
2.1.2. TURBO DELPHI I RAVE REPORTS.....	16
2.2. FUNCIONS DEL PROGRAMA.....	19
2.2.1. ENTORN GRÀFIC	19
2.2.2. FUNCIONALITAT DEL PROGRAMA DaDaSystem.....	20
2.2.2.1. CÀLCUL DEL FLUX DE CÀRREGUES.....	20
2.2.2.1.1. INTRODUCCIÓ DE DADES.....	21
2.2.2.1.2. RECONeixEMENT D'INCOGNITES.....	23
2.2.2.1.3. CÀLCUL DE YBUS.....	24
2.2.2.1.4. CÀLCUL $f_{p,i}$ i $f_{q,i}$	26
2.2.2.1.5. CÀLCUL DE $Df(X)$	26
2.2.2.1.6. CÀLCUL DE $Df(X)^{-1}$	29
2.2.2.1.7. CÀLCUL $X_{(k+1)}$ DE L'ERROR I ITERACIÓ.....	30
2.2.2.1.8. CÀLCUL DEL FLUX DE CÀRREGUES.....	31
2.2.2.1.9. ESQUEMA ITERATIU.....	36
2.2.2.2. ESQUEMA UNIFILAR	37
2.2.2.3. CÀLCUL DEL FLUX DE CÀRREGUES.....	39
2.3. DIFICULTATS DE PROGRAMACIÓ.....	40
2.3.1. SELECCIÓ DEL PROGRAMARI.....	40
2.3.2. RESOLUCIÓ DELS CÀLCULS.....	40
2.3.3. ESQUEMA UNIFILAR.....	41
3. CONCLUSIONS.....	42
4. ANNEXES.....	43
4.1. ANNEX 1: EQUACIONS DELS ELEMENTS DE TRANSMISSIÓ.....	44
4.1.1. MODELS DE LÍNIES	44
4.1.2. MODELS DE TRANSFORMADORS.....	47
4.2. ANNEX 2: CODI DE PROGRAMACIÓ.....	51
4.3. ANNEX 3: TUTORIAL.....	116
4.4. ANNEX 4: EXEMPLES RESOLTS.....	134
5. BIBLIOGRAFIA.....	148

1. INTRODUCCIÓ

1.1. INTRODUCCIÓ AL FLUX DE CÀRREGUES

L'objectiu principal de un sistema elèctric de potència es la de satisfer la demanda. Com a conseqüència d'aquest principi sorgeix el problema de saber on s'ha de connectar l'alimentació i inclús preveure caigudes de tensió, regulació de transformadors, injecció de potència reactiva...

Els estudis del flux de potència, més comunament anomenats estudis del flux de càrrega, són summament importants per avaluar el funcionament dels sistemes de potència, el seu control i la planificació de futures expansions.

El calcul del flux de càrregues s'utilitza per expressar la solució del sistema de potència en règim permanent. Els principis de l'estudi són fàcils, però un estudi relatiu a un sistema de potència real s'ha de dur a terme per mitja de sistemes digitals, donada la seva complexitat de càlcul. Per realitzar aquests càlculs es necessari que s'executin per mitjà de un sistema iteratiu. Els sistemes iteratius mes utilitzats són el mètode Gauss-Seidel i el mètode Newton-Raphson. Per la resolució del sistema de potència és necessari tenir en compte tots els elements de la xarxa. Tot i això s'utilitzaran sempre els esquemes més senzills de representació compatible amb la màxima exactitud de la informació de la que disposam. S'utilitza la representació més simple ja que es inútil utilitzar models complicats tant de línies i maquines quan la informació d'altres elements del sistema es coneixen de forma limitada.

El calcul del flux de carregues d'un sistema ens permet obtenir el flux de potència, tant activa com reactiva, entre totes les línies com també obtenir la tensió de totes les barres de un sistema de potència concret.

1.1.1 UTILITATS DEL FLUX DE CÀRREGUES

Les principals funcions de l'estudi del flux de càrregues son el coneixement de les següents dades:

- 1) Flux de potència activa (P) i reactiva (Q) de cada una de les interconnexions de la xarxa.
- 2) La tensió (V) a les barres de distribució.
- 3) Influencia sobre la càrrega del sistema quan es readapten els components del circuit o quan s'incorporen circuits nous.
- 4) Coneixement de les pèrdues de generació i transport del sistema i com influeixen aquestes en la càrrega.
- 5) Coneixement de les condicions optimes de funcionament del sistema com també la correcta distribució de les càrregues.
- 6) Optimització de pèrdues en un sistema.
- 7) Valors nominals òptims i marge de regulació dels transformadors.

- 8) Millores obtingudes amb la variació de secció dels conductors com també en la variació de tensió.

Aquest estudi del flux de càrregues es sol realitzar, normalment, per dues condicions de funcionament: càrrega màxima i càrrega mínima.

El funcionament de càrrega màxima s'utilitza per poder saber, en condicions màximes, quins resultats obtendríem i com hauríem de planificar els plans alternatius a fi de poder atendre l'augment de demanda del sistema.

Per altre banda el funcionament de càrrega mínim es interessant saber-ho per poder preveure el flux continu de potencia que ha de mantenir la xarxa. Aquestes dades són interessants principalment per el govern o per les empreses subministradores.

1.1.2 RESOLUCIÓ I CONSIDERACIONS

El desenvolupament del problema del flux de càrregues es desenvolupa amb els següents apartats:

- 1) Formulació d'un model matemàtic adequat per la xarxa. Aquest model ha d'incloure, de forma adequada, les relacions entre voltatge i potències en el sistema.
- 2) Especificació de les restriccions de potencia i voltatge que s'han d'aplicar a tots es busos.
- 3) Calcul numèric de les equacions del flux de càrregues. D'aquestes equacions obtenim tots els voltatges, en angle i modul, de la xarxa.
- 4) Resolució del flux de potencia de tots els elements de transmissió i, alhora, les pèrdues de potència.

Per planificar la resolució del problema del flux de càrregues es necessari tenir en compte una serie de consideracions a l'hora de solucionar el problema.

Aquestes consideracions són les següents:

1. Els generadors són els únics encarregats de produir potencia activa (P). La localització i capacitat d'aquests generadors es fixa. La generació ha de ser, en tot moment, igual a la suma de la demanda i les pèrdues.
2. Els elements de transmissió poden transportar una certa quantitat de potencia, ens em d'assegurar que no es sobrepassa aquesta cargabilitat.
3. Els nivells de voltatge d'operació de certs busos s'ha de mantenir dins un marge de tolerància. Aquest fet es regula mitjançant la generació apropiada de potencia reactiva.
4. El xarxa elèctrica d'estudi forma part de un sistema més gran, també conegut com "power pool". Per tant s'hauran de mantenir un certs nivells de potencia apropiada en els bussos d'enllaç.

5. Les sortides del sistema, provocades per diversos disturbis del sistema, es poden minimitzar mitjançant un sistema d'estudi adequat realitzant múltiples estudis del flux de carrega.
6. Per dur a terme d'una manera apropiada i eficient la tasca de planificació és imprescindible l'estudi extens i acurat del flux de càrregues

De la formulació matemàtica del problema del flux de càrregues s'obté un sistema d'equacions algebraiques i no lineals, sistema d'equacions de segon grau. Al ser un sistema d'equacions d'aquest tipus no es poden resoldre amb una resolució analítica del sistema, sinó que es necessari utilitzar mètodes de resolució més complexos, normalment Gauss-Seidel o Newton-Raphson.

1.2. TIPUS D'ELEMENTS EN UN SISTEMA ELÈCTRIC DE POTÈNCIA

Els principals elements, en l'estudi del flux de càrregues en els sistemes elèctrics de potència són: línies, transformadors i càrregues.

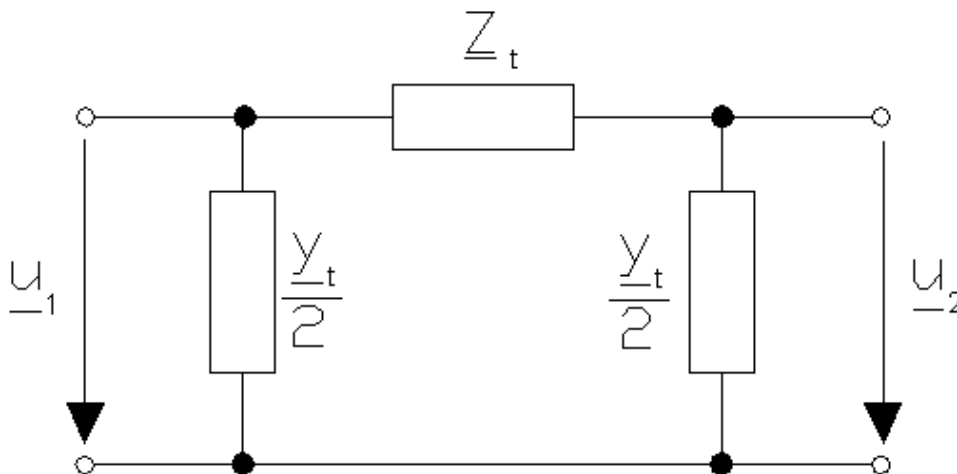
Les càrregues sempre vendran definides com a una potència absorbida, d'aquesta seran considerades fonts de potència constant.

Per tant els únics elements a tenir en compte seran línies i transformadors, dels quals només tindrèm en compte el de doble bobinat. Per tant aquests elements es poden simplificar en un esquema en " π ". Tot i que amb aquesta simplificació no es totalment precisa es valida ja que les dades que coneixem no tenen una precisió massa minuciosa.

1.2.1. TIPUS DE CONNEXIONS DE XARXA

1.2.1.1. MODELS DE LÍNIES

L'esquema equivalent en " π " de les línies vendria definit per el següent esquema:



Del qual la matriu d'admitàncies vendrà definit per

$$\underline{Y}_{BUS} = \begin{bmatrix} \frac{y_t}{2} + \frac{1}{z_t} & -\frac{1}{z_t} \\ -\frac{1}{z_t} & \frac{y_t}{2} + \frac{1}{z_t} \end{bmatrix}$$

D'aquesta forma, el flux de potència a través de la línia vendrà definit per:

$$S_{12}^* = u_1^* \cdot \left[\left[\frac{y_t}{2} + \frac{1}{z_t} \right] \cdot u_1 - \frac{1}{z_t} \cdot u_2 \right] ; S_{12}^* = u_1^* \cdot \left[\frac{y_t}{2} \cdot u_1 - \frac{u_1 - u_2}{z_t} \right]$$

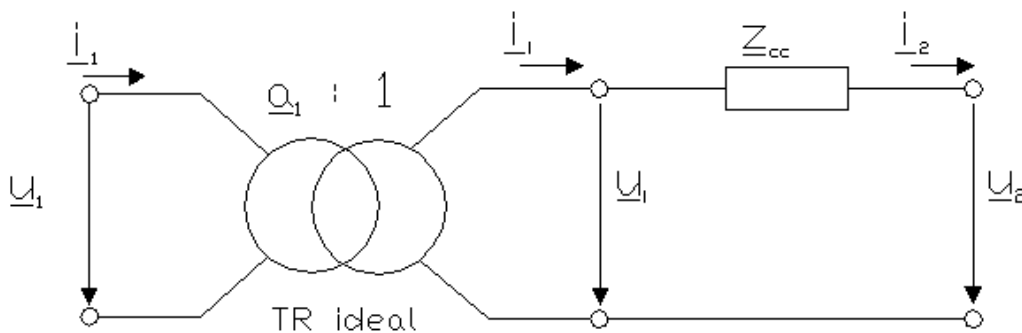
$$S_{21}^* = u_2^* \cdot \left[\frac{-1}{z_t} \cdot u_1 + \left[\frac{y_t}{2} + \frac{1}{y_t} \right] \cdot u_2 \right] ; S_{21}^* = u_2^* \cdot \left[\frac{y_t}{2} \cdot u_2 - \frac{u_2 - u_1}{z_t} \right]$$

Essent les pèrdues per transmissió en la línia:

$$\Delta S = s_{12} + s_{12} ; \Delta S = \left[\frac{y_t}{2} + \frac{1}{y_t} \right] \cdot (u_1^2 - u_2^2) - \frac{2 \cdot \Re \{ u_1^* \cdot u_2 \}}{z_t^*}$$

1.2.1.2. MODELS DE TRANSFORMADORS

D'aquesta forma l'esquema del transformador, menyspreant els elements transversals, es pot representar de la següent forma:



Del qual es pot definir la matriu d'admitàncies per el següent manera :

$$Y_{BUS} = \begin{bmatrix} \frac{z_{cc}^{-1}}{a^2} & -\frac{z_{cc}^{-1}}{a^*} \\ -\frac{z_{cc}^{-1}}{a} & \frac{z_{cc}^{-1}}{1} \end{bmatrix}$$

D'aquesta forma, el flux de potències a través del transformador vendrà serà:

$$\underline{S}_{12}^* = \underline{u}_1^* \cdot \left[\frac{\underline{u}_1}{a^2 \cdot \underline{z}_{cc}} - \frac{\underline{u}_2}{a \cdot \underline{z}_{cc}} \right] ; \quad \underline{S}_{12}^* = \frac{\underline{u}_1^*}{a \cdot \underline{z}_{cc}} \cdot \left[\frac{\underline{u}_1}{a} - \underline{u}_2 \right]$$

$$\underline{S}_{21}^* = \underline{u}_2^* \cdot \left[-\frac{\underline{u}_1}{a \cdot \underline{z}_{cc}} + \frac{\underline{u}_2}{\underline{z}_{cc}} \right] ; \quad \underline{S}_{12}^* = \frac{\underline{u}_2^*}{\underline{z}_{cc}} \cdot \left[\underline{u}_2 - \frac{\underline{u}_1}{a} \right]$$

Essent les pèrdues de transmissió:

$$\Delta \underline{S} = \underline{s}_{12} + \underline{s}_{21} ; \quad \underline{\Delta}_S = \frac{1}{\underline{z}_{cc}^*} \cdot \left[\frac{\underline{u}_1}{a} - \underline{u}_2 \right]^2$$

Considerem ara les següents particularitats:

- 1) Suposam que en el transformador no hi ha un decalatge de fase. En aquestes condicions : $\underline{a} = a$.

La nova matriu d'admitàncies vendrà definida per:

$$\underline{Y}_{BUS} = \begin{bmatrix} \underline{y}_1 + \underline{z}^{-1} & -\underline{z}^{-1} \\ -\underline{z}^{-1} & \underline{y}_2 + \underline{z}^{-1} \end{bmatrix} ; \quad \underline{Y}_{BUS} = \begin{bmatrix} \frac{\underline{z}_{cc}^{-1}}{a^2} & \frac{-\underline{z}_{cc}^{-1}}{a} \\ \frac{-\underline{z}_{cc}^{-1}}{a} & \frac{\underline{z}_{cc}^{-1}}{1} \end{bmatrix}$$

- 2) Si el transformador no disposa de relació de transformació variable, i tampoc dona lloc a decalatge de fase, $a=1$. En aquest cas el transformador correspon a un quadripol reciproc i simètric. Essent la seva impedància equivalent la impedància de curtcircuit del transformador.

La nova matriu d'admitàncies vendrà definida per:

$$\underline{Y}_{BUS} = \begin{bmatrix} \underline{z}_{cc}^{-1} & -\underline{z}_{cc}^{-1} \\ -\underline{z}_{cc}^{-1} & \underline{z}_{cc}^{-1} \end{bmatrix}$$

1.2.2. TIPUS DE BUSOS

A tota barra de un sistema elèctric de potència es consideren quatre magnituds bàsiques, que són: angle de fase i modul de la tensió i la potència activa i reactiva.

Normalment coneixem dues de les quatre magnituds anteriorment nombrades, segons aquestes dades podem dividir els busos amb les següents categories: Slack Bus, Bus PQ i Bus PV.

La relació d'aquests elements amb les dades conegudes i les seves incògnites, és la següent:

Tipus de bus	Variables conegudes				Incògnites					
Slack Bus	P_D	Q_D			$ V $	δ	P_G	Q_G		
Bus PQ	P_D	Q_D	P_G	Q_G					$ V $	δ
Bus PV	P_D	Q_D	P_G		$ V $			Q_G		δ

On el subíndex D fa referència a la demanda i el subíndex G fa referència a la generació.

1.2.2.1. SLACK BUS

El bus de referència o compensador, més conegut amb la seva nomenclatura anglesa com "Slack" o "Swing". Slack Bus és una barra en la que la tensió, tant en modul i en argument, es mantén constant d'aquesta manera estam definint una referència fasorial. Desconeixent, normalment, la potència activa i reactiva injectades a la xarxa, normalment es el nus on la potència s'aproxima mes a unes barres de potència infinita. Des de el punt de vista del circuit es considera un generador pur de tensió.

1.2.2.2. BUS DE CÀRREGA

En un bus de càrrega, també conegut com bus PQ, coneixem la potència tant activa com reactiva subministrades pel bus a la xarxa i que, per tant, tenim com a incògnita la tensió, tant en mòdul com en argument, de la mateixa. Tot i que en general són barres de consum, també poden ser de generació.

1.2.2.3. BUS DE TENSÍO CONTROLADA

En el bus de tensió controlada, també conegut com bus PV, coneixem la tensió en modul, però no el seu argument, també es coneix la potència activa. Per tant les incògnites d'aquestes barres són l'argument de la tensió i la potència reactiva injectada a la xarxa..

En les barres on hi ha transformadors amb relació de transformació variable, hi poden haver-hi tres magnituds especificades que es pretenen mantenir constants ajustant la toma de regulació del citat transformador, per tant les incògnites d'aquest nou cas seria l'angle de fase de la tensió i la toma de regulació de transformació del propi transformador.

1.3. RESOLUCIÓ DEL PROBLEMA

1.3.1 EQUACIONS

Considerem una xarxa elèctrica amb N busos. Per tant podem descriure les equacions del sistema a partir de les següents equacions del sistema a través dels nusos:

$$\underline{Y}_{BUS} \cdot \underline{V}_{BUS} = \underline{J}_{BUS}$$

$$\sum_{k=1}^n y_{ik} \cdot v_k = j_i \quad ; \quad i=1,2,3,\dots,n$$

Ara bé, en tots els busos de la xarxa es verifica que:

$$s_i^* = s_{gi}^* - s_{ci}^* \quad s_i^* = p_i - j \cdot q_i \quad s_i^* = v_i^* - j_i \quad ; \quad i=1,2,3,\dots,n$$

essent: s_i = potència aparent complexa injectada en el bus i.

j_i = intensitat del bus i ; es considera positiva si entra en el sistema.

D'aquesta forma s'obté:

$$j_i = \frac{s_i^*}{v_i^*} \quad ; \quad j_i = \frac{p_i - j \cdot q_i}{v_i^*} \quad ; \quad j_i = \frac{p_{Gi} - p_{Ci} - j \cdot (q_{Gi} - q_{Ci})}{v_i^*}$$

-

I substituint-la en la corresponent equació de nusos:

$$\sum_{k=1}^n y_{ik} \cdot v_k = \frac{s_i^*}{v_i^*} \quad ; \quad i=1,2,3,\dots,n$$

$$v_i^* \cdot \sum_{k=1}^n y_{ik} \cdot v_k = s_i^* \quad ; \quad i=1,2,3,\dots,n$$

Si la xarxa s'ha analitzat de tal forma que el bus slack es el primer de tots obtenim:

$$v_i^* \cdot \sum_{k=1}^n y_{ik} \cdot v_k = s_i^* \quad ; \quad i=1,2,3,\dots,n$$

de la qual podem determinar la tensió de la resta de busos

$$S_1^* = V_1^* \cdot \sum_{k=1}^n Y_{1k} \cdot V_k$$

La primera de les N equacions s'utilitzaran per determinar la potència injectada per el bus slack al sistema.

Una vegada obtingues les tensions es podrà procedir al calcul del flux de potencies a totes les ramificacions de la xarxa.

Com que ens trobam en un sistema d'equacions no lineals, esta clar que no es pot resoldre per sistemes de resolució simples i que es necessari la utilització de mètodes iteratius. Els més comuns son Gauss-Seidel i Newton-Raphson, que ambdós es resolen de la següent manera:

1) Es suposa una aproximació a la solució.

2) Aquesta aproximació, juntament amb el sistema d'equacions, s'utilitza per trobar una aproximació més exacte i això es va repetint fins que s'ha obtingut un error inferior a un error predeterminat.

Els distints mètodes utilitzen diferents algoritmes de calcul per obtenir les noves aproximacions. Com més complex sigui aquest algoritme, més exacte serà la solució.

1.3.2. MÉTODES DE RESOLUCIÓ

1.3.2.1. MÉTODE DE GAUSS-SEIDEL

Per el mètode de resolució de Gauss-Seidel es considera una xarxa de N busos numerats de la següent forma: el primer és el bus slack, els següents m-1 son els busos de tensió controlada i els restants son els busos de càrrega.

Les equacions del flux de càrregues a resoldre són les següents:

$$S_i^* = V_i^* \cdot \sum_{j=1}^n (Y_{ij} \cdot V_j) \quad ; \quad i=2,3,\dots,n$$

De les equacions anteriors, per determinar V_i aplicant el sistema de Gauss-Seidel obtenim:

$$Y_{ii} \cdot V_i \cdot V_i^* = S_i^* - V_i^* \cdot \sum_{\substack{j=1 \\ j \neq i}}^n (Y_{ij} \cdot V_j)$$

$$V_i = \frac{1}{V_i^*} \cdot \frac{S_i^*}{Y_{ii}} - \sum_{\substack{j=1 \\ j \neq i}}^n \left(\frac{Y_{ij}}{Y_{ii}} \cdot V_j \right)$$

En aquestes condicions el calcul iteratiu es realitzarà de la següent forma:

1) Per el busos de tensió controlada:

$$q_i^{(k)} = -\Im \left\{ (V_i^{(k)})^* \cdot \left[\sum_{j=1}^{i-1} Y_{ij} \cdot V_j^{k+1} + \sum_{j=i}^n Y_{ij} \cdot V_j^k \right] \right\}$$

$$\delta_{i,cal}^{(k+1)} = -\arg \left\{ \sum_{j=1}^{i-1} \frac{Y_{ij}}{Y_{ii}} \cdot V_j^{k+1} + \frac{1}{(V_i^{(k)})^*} \cdot \frac{p_i - jq_i^{(k)}}{Y_{ii}} - \sum_{j=i+1}^n \frac{Y_{ij}}{Y_{ii}} \cdot V_j^k \right\}$$

$$V_i^{k+1} = |V_i|_{esp} \cdot e^{j\delta_{i,cal}^{(k+1)}} ; i=2,3,\dots,m$$

2) Pels busos de càrrega

$$V_i^{k+1} = -\sum_{j=1}^{i-1} \frac{Y_{ij}}{Y_{ii}} \cdot V_j^{(k+1)} + \frac{1}{(V_i^{(k)})^*} \cdot \frac{S_i^*}{Y_{ii}} - \sum_{j=i+1}^n \frac{Y_{ij}}{Y_{ii}} \cdot V_j^{(k+1)} ; i=m+1, m+2, \dots, n$$

Un cop acabat el cicle iteratiu de calcul i obtingudes les solucions, s'haurà de comprovar la potencia reactiva dels busos de tensió controlada per assegurar-se que aquest estan dins els límits establerts. En el cas de que la potencia reactiva sobrepassi els límits indica que la tensió en modul no es pot mantenir i per tant s'haurà de realitzar el calcul iteratiu de nou posant com a potencia reactiva d'aquest bus la màxima, o mínima, segons el limit sobrepassat.

Finalment quan es disposin de totes les tensions de tots els busos de la xarxa ja es podran determinar el fluxos de potència activa i reactiva a través de totes les ramificacions del sistema.

1.3.2.2. MÉTODE DE NEWTON-RAPHSON

Consideram una xarxa de N busos, dels quals el primer es un Slack-bus, i els M-1 busos següents són busos de tensió controlada i la resta de busos son de càrrega:

Les equacions del flux de càrregues que s'han de resoldre numèricament són:

$$s_i^* = p_i - j \cdot q_i ; s_i^* = v_i^* \cdot \sum_{k=1}^n y_{ij} \cdot v_j ; i=1,2,3,\dots,n$$

Essent:

$$y_{ij} = g_{ij} - j \cdot b_{ij} ; v_i = |v_i| \cdot e^{j \cdot \delta_i}$$

s'obté,

$$p_i = |V_i| \cdot \left\{ \sum_{j=1}^n |V_j| \cdot [g_{ij} \cdot \cos(\delta_j - \delta_i) - b_{ij} \cdot \sin(\delta_j - \delta_i)] \right\}$$

$$q_i = -|V_i| \cdot \left\{ \sum_{j=1}^n |V_j| \cdot [g_{ij} \cdot \sin(\delta_j + \delta_i) - b_{ij} \cdot \cos(\delta_j - \delta_i)] \right\}$$

amb el qual podem definir:

$$f_{h,p} = P_h - \left\{ \sum_{j=1}^n |V_h| \cdot |V_j| \cdot [g_{hj} \cdot \cos(\delta_j - \delta_h) + b_{hj} \cdot \sin(\delta_j - \delta_h)] \right\}$$

$$2 \leq h \leq n$$

$$f_{l,q} = q_l + |V_l| \cdot \left\{ \sum_{j=1}^n |V_j| \cdot [g_{lj} \cdot \sin(\delta_j - \delta_l) + b_{lj} \cdot \cos(\delta_j - \delta_l)] \right\}$$

$$m+1 \leq l \leq n$$

En el sistema d'equacions anteriors, les incògnites a determinar són els arguments de les tensions de tots els busos (exceptuant el bus slack) i els mòdul de les tensions de tots els busos de càrrega. D'aquesta forma el vector incògnita és:

$$X = \begin{bmatrix} \delta \\ |V| \end{bmatrix} \text{ essent : } \delta = \begin{bmatrix} \delta_2 \\ \delta_3 \\ \vdots \\ \delta_n \end{bmatrix} ; \quad |V| = \begin{bmatrix} |V_{m+1}| \\ |V_{m+2}| \\ \vdots \\ |V_n| \end{bmatrix}$$

i anomenat ,

$$f = \begin{bmatrix} f_p \\ f_q \end{bmatrix} \text{ essent : } f_p = \begin{bmatrix} f_{2,p} \\ f_{3,p} \\ \vdots \\ f_{n,p} \end{bmatrix} ; \quad f_q = \begin{bmatrix} f_{m+1,q} \\ f_{m+2,q} \\ \vdots \\ f_{n,q} \end{bmatrix}$$

el sistema d'equacions a resoldre és: $f(x)=0$, amb el qual el mètode de Newton-Raphson proporciona les iteracions següents:

$$X^{(k+1)} = X^{(k)} - \{Df(x^{(k)})\}^{-1} \cdot f(x^{(k)})$$

a on el jacobí ve definit per:

$$Df_{(x)} = \begin{bmatrix} \frac{\partial f_p}{\partial \delta} & \frac{\partial f_p}{\partial |\underline{v}|} \\ \frac{\partial f_p}{\partial \delta} & \frac{\partial f_p}{\partial |\underline{v}|} \end{bmatrix} \quad \text{amb,}$$

$$\begin{bmatrix} \frac{\partial f_p}{\partial |\underline{v}|} \end{bmatrix} \quad \text{de dimensions } (n-1) \times (n-1)$$

$$\begin{bmatrix} \frac{\partial f_q}{\partial |\underline{v}|} \end{bmatrix} \quad \text{de dimensions } (n-1) \times (n-m)$$

$$\begin{bmatrix} \frac{\partial f_p}{\partial \delta} \end{bmatrix} \quad \text{de dimensions } (n-m) \times (n-1)$$

$$\begin{bmatrix} \frac{\partial f_q}{\partial \delta} \end{bmatrix} \quad \text{de dimensions } (n-m) \times (n-m)$$

Els distints elements del jacobí venen donats per:

$$\frac{\partial f_{h,p}}{\partial \delta_j} = |\underline{v}_h| \cdot |\underline{v}_j| \cdot [g_{hj} \cdot \sin(\delta_j - \delta_h) + b_{hj} \cdot \cos(\delta_j - \delta_h)] ; \quad h \neq j$$

$$\frac{\partial f_{h,p}}{\partial \delta_h} = - \sum_{\substack{j=1 \\ j \neq h}}^n |\underline{v}_h| \cdot |\underline{v}_j| \cdot [g_{hj} \cdot \sin(\delta_j - \delta_h) + b_{hj} \cdot \cos(\delta_j - \delta_h)]$$

$$\frac{\partial f_{h,p}}{\partial |\underline{v}_j|} = -|\underline{v}_h| \cdot [g_{hj} \cdot \cos(\delta_j - \delta_h) - b_{hj} \cdot \sin(\delta_j - \delta_h)] ; \quad h \neq j$$

$$\frac{\partial f_{h,p}}{\partial |\underline{v}_h|} = -2 \cdot g_{hh} \cdot |\underline{v}_h| - \sum_{\substack{j=1 \\ j \neq h}}^n |\underline{v}_j| \cdot [g_{hj} \cdot \cos(\delta_j - \delta_h) - b_{hj} \cdot \sin(\delta_j - \delta_h)]$$

$$\frac{\partial f_{l,q}}{\partial \delta_j} = |\underline{v}_l| \cdot |\underline{v}_j| \cdot [g_{lj} \cdot \cos(\delta_j - \delta_l) - b_{lj} \cdot \sin(\delta_j - \delta_l)] ; \quad l \neq j$$

$$\frac{\partial f_{l,q}}{\partial \delta_l} = - \sum_{\substack{j=1 \\ j \neq l}}^n |V_l| \cdot |V_j| \cdot [-g_{lj} \cdot \sin(\delta_j - \delta_l) + b_{lj} \cdot \cos(\delta_j - \delta_l)]$$

$$\frac{\partial f_{l,q}}{\partial |v_j|} = -|V_l| \cdot [g_{lj} \cdot \sin(\delta_j - \delta_l) + b_{lj} \cdot \cos(\delta_j - \delta_l)] ; \quad l \neq j$$

$$\frac{\partial f_{l,q}}{\partial |v_l|} = -2 \cdot g_{ll} \cdot |v_l| - \sum_{\substack{j=1 \\ j \neq l}}^n |V_j| \cdot [g_{lj} \cdot \sin(\delta_j - \delta_l) + b_{lj} \cdot \cos(\delta_j - \delta_l)]$$

Un possible criteri per determinar el càlcul iteratiu és el que ve donat pe l'equació:

$$\Delta x^{(k)} = x^{(k+1)} - x^{(k)}$$

Si en el sistema existeixen transformador amb regulació, en nombre t, els busos en els que es coneguin les dades de potencia activa, reactiva i el modul de la tensió, seran considerats busos de càrrega.

Essent enumerats abans que aquest, es a dir, m+1, m+2,, m+t.

En aquestes condicions, en el nou vector incògnita apareixeran les relacions de transformació dels citats transformadors, incloses en la matriu Y_{BUS} , ja que els seus elements seran:

$$Y_{ij} = g_{ij}(a_1, a_2, \dots, a_t) + j \cdot b_{ij}(a_1, a_2, \dots, a_t)$$

Així doncs el nou vector incògnita serà:

$$X = \begin{bmatrix} \delta \\ |v| \\ a \end{bmatrix} \quad \text{essent : } |v| = \begin{bmatrix} |v_{m+t+1}| \\ |v_{m+t+2}| \\ \dots \\ |v_n| \end{bmatrix} ; \quad a = \begin{bmatrix} a_2 \\ a_3 \\ \dots \\ a_n \end{bmatrix}$$

essent el jacobi:

$$Df_{(x)} = \begin{bmatrix} \frac{\partial f_p}{\partial \delta} & \frac{\partial f_p}{\partial |v|} & \frac{\partial f_p}{\partial a} \\ \frac{\partial f_q}{\partial \delta} & \frac{\partial f_q}{\partial |v|} & \frac{\partial f_q}{\partial a} \end{bmatrix} \quad \text{amb,}$$

$$\left[\frac{\partial f_p}{\partial \delta} \right] \text{ de dimensions } (n-1) \times (n-1)$$

$$\left[\frac{\partial f_p}{\partial |y|} \right] \text{ de dimensions } (n-1) \times (n-m-1)$$

$$\left[\frac{\partial f_p}{\partial a} \right] \text{ de dimensions } (n-1) \times t$$

$$\left[\frac{\partial f_q}{\partial \delta} \right] \text{ de dimensions } (n-m) \times (n-1)$$

$$\left[\frac{\partial f_q}{\partial |y|} \right] \text{ de dimensions } (n-m) \times (n-m-t)$$

$$\left[\frac{\partial f_q}{\partial a} \right] \text{ de dimensions } (n-m) \times t$$

1.3.2.3. MÈTODE DE NEWTON-RAPHSON MODIFICAT

Aquest mètode es basa en calcular tan sols un jacobi, el de la primera iteració. A partir d'aquest jacobi, i sense modificar-lo en totes les iteracions, obtindrem totes les solucions del procés iteratiu. Així doncs tenim que:

$$X^{(k+1)} = X^{(k)} - \{Df(x)^{(0)}\}^{-1} \cdot f(x)^{(k)}$$

Aquest sistema, tot i que redueix notablement el volum de càlculs a l'hora de realitzar les iteracions, no disposa de la convergència quadràtica que ens aporta el sistema de Newton-Raphson.

2. PROGRAMA DE CÀLCUL

2.1. INTRODUCCIÓ AL PROGRAMADOR

2.1.1. PROGRAMACIÓ ORIENTADA A OBJECTES

La Programació Orientada a Objectes(POO o OOP segons les seves sigles en anglès) es un sistema de programació que es basa en objectes i les interaccions d'aquestes per dissenyar aplicacions i programes d'ordinador. L'ús de la POO es va popularitzar als anys 90.

Els objectes de la POO són entitats que contenen tres elements bàsics:

- 1) Estat: l'estat esta compost per un o varis atributs que porten assignats uns determinats valors concrets(dades).
- 2)Comportament: està definit per procediments o mètodes de treball que potrealitzar,o no, aquest objecte a més de les operacions que si poden realitzar.
- 3)Identitat: és l'identificador que defineix l'element i el distingeix del resta. Aquest element ens defineix quin tipus d'element es.

Aquest tipus de programació orientada a objectes ens permet realitzar programes o mòduls molt més fàcils de escriure, mantenir i editar. Els elements de la POO que venen definits per defecte pel llenguatge de programació ens redueix, i facilita, notablement el treball de programació ja que a l'hora de implementar un element tan sols es necessari introduir l'objecte i assignar-li els valors.

Els conceptes bàsics de la POO, més enllà de la programació clàssica, que cal conèixer abans d'avançar més amb aquest tipus de programació. Els conceptes que són necessaris destacar principalment:

- 1)Classe: conté la definició de les propietats i el comportament d'un element concret. La instanciació és la lectura d'aquestes definicions i la conseqüent creació d'un objecte a partir d'elles.
- 2)Herència: és la interacció entre diferents elements que venen definits un a partir de l'altre. En la definició d'atributs i operacions de un element "B" que provenen de un altre element "A" com si directament fossin definides en "B".
- 3)Objecte: és una entitat que conté un conjunts de propietats o atributs i una serie e comportaments o funcionalitats. Són elements visuals o de funcionament intern del programa.
- 4)Mètode: Algoritme associat a un objecte que produeix un canvi d'estat segons la interacció que es fa a l'objecte. Un mètode pot produir un canvi en les propietats d'un objecte.
- 5)Esdeveniment : com el seu propi nom indica és un esdeveniment que es produeix quan l'usuari fa una acció i d'aquesta n'obté una reacció conseqüent amb aquesta acció.
- 6)Missatge: és una comunicació interna dirigida a un objecte que li diu com ha

d'actuar ja sigui per mitjà de un mètode o per un esdeveniment.

7) Propietat: conté la informació del tipus de dades que conté un objecte i quins poden ser modificats de forma externa i com.

8) Representació d'un objecte: un objecte es representa per mitjà d'una taula o una entitat que esta composta pels seus atributs i funcions corresponents.

2.1.2. TURBO DELPHI I RAVE REPORTS

Turbo Delphi, o Delphi, és un entorn de desenvolupament del software dissenyat especialment per la programació en entorn visual. Delphi utilitza un sistema de programació que és l'evolució d'un llenguatge de programació, pascal, que actualment es "Object Pascal". Aquest software és un software propietari, amb una versió lliure, distribuït per CodeGear/Embarcadero Technologies. Permet programar per diferents plataformes com són Windows, Linux i la plataforma .NET.

En la darrera versió RAD Studio 2009 inclou en el mateix entorn de programació els següents llenguatges:

- * Delphi per Win32
- * Delphi per .NET
- * C# per .NET
- * C++

També existeix una versió de Delphi per sistemes Unix i Linux denominada Kylix.

Com tot llenguatge de programació Delphi conté un tipus de variables acceptades i operables pel propi llenguatge:

Nom (Tipus)	Variable
Boolean	Valor lògic: Veritat (TRUE) o Fals (FALSE).
Char	Una sola lletra
Byte	Un nombre entre 0 a 255.
Smallint	Un nombre enter entre -32768 a 32767.
Word	Un nombre enter entre 0 a 65535.
Integer o Longint	Un nombre enter entre -2,147'483,648 a 2,147'483,647
Cardinal	Un nombre enter entre 0 a 2,147'483,647
Real	Nombre enter i decimal
Double	Un nombre enter i decimal més exacte que Real
String	Cadena de caràcters

Aquestes variables es poden emmagatzemar en valors simples, *array*(valors emmagatzemats en un vector) o *array of array*(matrius). Tot i això el tractament de dades sempre es fera de manera individual tot i que en una *array* es poden introduir diferent elements d'un mateix tipus o una mateixa funció arribant a ells mitjançant la introducció de la seva coordenada dins *l'array*.

A aquestes variables podem aplicar el següents tipus d'operacions aritmètiques bàsiques:

Operació	Descripció
+	Suma
-	Resta
*	Multiplicació
/	Divisió de nombres reals
Div	Divisió de nombres enters
Mod	Resta de nombres enters

A més d'aquestes operacions aritmètiques disposam també d'una seria d'operadors com són:

Operadors Relacionals	
Operació	Descripció
<	Menor
<=	Menor o igual
>	Major
>=	Major o igual
<>	Diferent
=	Igual

Operadors Lògics	
Operació	Descripció
And	i
Or	o
Xor	Bicondicional
Not	Negació
Ord()	Ordinal
Succ()	Successor
Pred()	Antecessor
Inc()	Increment
Dec()	Decrement

Per aplicar aquestes funcions als tipus de variables descrits anteriorment disposam de una serie de procediments per condicionar o crear un bucle per unes condicions determinades:

Procediment	Funció
For 0 to N Do	Repeteix l'algoritme N vegades
If <i>condició</i> then	Si la <i>condició</i> és veritat, procedeix a executar l'algoritme
While <i>condició</i> do	Mentre la <i>condició</i> és veritat repeteix l'algoritme
Repeat//Until(<i>Condició</i>)	Repeteix l'algoritme fins que la <i>condició</i> es veritat
Case(<i>Opció</i>)	Segons la <i>opció</i> d'entrada procedeix a executar un algoritme o un altre

Un cop conegudes totes aquestes operacions i procediments bàsics i tot i que es disposa de una amplia funcionalitat i possibilitats d'aplicació des de BDD(bases de dades) a interconnexions via Web nosaltres ens centrarem amb els operadors matemàtics que són els que utilitzarem principalment.

Des de el punt de vista numèric, per tant, podem disposar de les següents operacions matemàtiques:

Funció f(x)	Descripció
Sin(x)	Retorna el valor en radians del sinus de l'angle X
Cos(x)	Retorna el valor en radians del coseno de l'angle X
Arctan(x)	Retorna el valor en radians del arc tangent de l'angle X
Exp(x)	Retorna l'exponencial "e" de X
Ln(x)	Retorna el logaritme natural de l'argument X
Sqr(x)	Retorna el quadrat de l'argument X
Sqrt(x)	Retorna l'arrel quadrada de l'argument X
Abs(x)	Retorna el valor absolut de X
Round(x)	Retorna l'arredoniment de X
Trunc(x)	Retorna el valor de X sense fracció
Frac(x)	Retorna la part decimal de un nombre real X
Int(x)	Retorna la part entera de un nombre real X
Ord(x)	Retorna el valor associat a un caracter ASCII

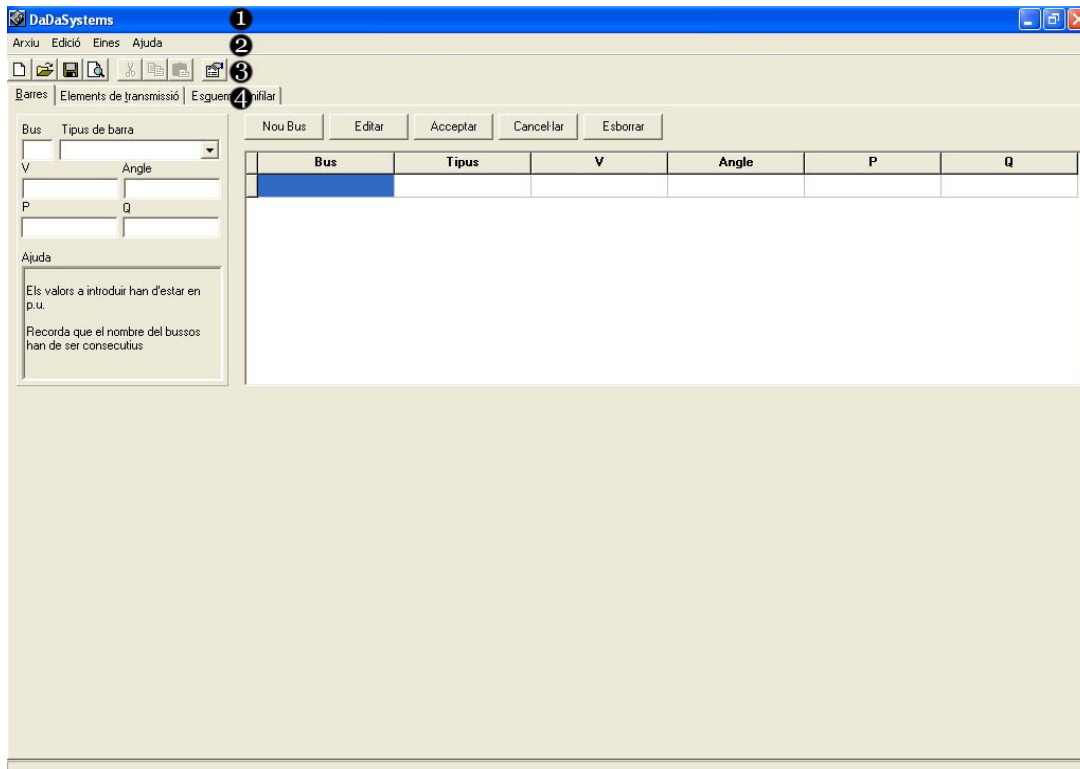
A més de tot això s'utilitzaran els objectes ja comentats de Object Pascal a fi de poder implementar tant els botons d'execució com la introducció de dades com de la mateixa forma introduir tota la interfície gràfica.

Per altre part Rave Reports, és una eina adherida a Delphi, que ens permet el tractament i exportació de documents tant en format Rave Reports com en altres formats com RTF, TXT o, el format més estandaritzat, PDF. El tractament de informació des de Rave Reports és molt versàtil i te un sistema de connexió molt àgil i que consumeix molts pocs recursos del sistema i que per tant es una bona eina complementaria per l'exportació de dades.

2.2. FUNCIONS DEL PROGRAMA

2.2.1 ENTORN GRÀFIC

L'entorn gràfic del programa DaDaSystem, es presenta com un entorn gràfic clàssic de Windows amb tot el que es refereix a barra de títol, barra de menú i barra d'eines.



Dibuix 1: Interfície gràfica

A la Barra de títol (veure 1 de *Dibuix 1*) tenim el nom de l'aplicació. També hi trobam els controls bàsic d'una aplicació: minimitzar, maximitzar i tancar. En la Barra de Menú (veure 2 de *Dibuix 1*) en aquesta barra podem trobar els següents menús: Arxíu, edició, eines i ajuda. Des de la edició podem modificar el valor de l'error predeterminat de iteració. I en darrer lloc en la barra d'eines i podem (veure 3 de *Dibuix 1*) trobar accessos ràpids a les principals funcions com nou/obrir/guardar, funcions de copiar i enganxar i començament del càlcul iteratiu.

El programa disposa de tres pestanyes d'acció definides (veure 4 de *Dibuix 1*) com Barres, Elements de transmissió i Esquema unifilar.

En la pestanya de barres esta pensada per introduir les dades de les barres així mateix com enumerar-les i definir-les. A l'hora de definir els busos tenim les següents opcions:

En la pestanya de elements de transmissió esta també pensada per introduir les dades dels elements de connexió entre busos com també les bateries de transformadors.

S'ha de tenir en compte que tant en la pestanya de Barres com en la pestanya d'Elements de transmissió l'usuari pot observar el procés iteratiu a la part inferior de la pantalla.

En la pestanya d'esquema unifilar es un entorn gràfic es pretén dotar de un sistema fàcil i pràctic per la representació de circuit de sistemes de potència. Per això en aquest sistema ens aporta uns icones, referents als tipus d'elements que podem trobar a una xarxa.

Disposam de línies, transformadors, generadors, càrregues, condensadors, barres i opció de les llegendes dels elements introduïts.

A part de tots els elements es pot observar una miniatura d'aquests, com també rotar-los o corregir els elements mal situats.

2.2.2. FUNCIONALITAT DEL PROGRAMA DaDaSystem

La part funcional del programa es pot dividir en dues parts diferenciades que són la part de càlcul del flux de càrregues i la part de la representació de l'esquema unifilar.

2.2.2.1. CÀLCUL DEL FLUX DE CÀRREGUES

Pel càlcul del flux de càrregues s'utilitzarà l'algoritme de Newton-Raphson a fi de trobar el valor real de totes les incògnites.

Sabent que la resolució iterativa per Newton-Raphson es basa en:

$$X^{(k+1)} = X^{(k)} - \{Df(x)^{(k)}\}^{-1} \cdot f(x)^{(k)}$$

On, en el cas que ens ocupa:

$$X = \begin{bmatrix} X_1 \\ X_2 \\ \dots \\ X_n \end{bmatrix} ; \text{on en el vector } X \text{ s'inclouen totes les incògnites del sistema, des de } X_1 \text{ fins a } X_n . \text{ Les variables podran ser } \delta, |V| \text{ i } a \text{ segons l'element que disposem.}$$

$$f(x) = \begin{bmatrix} f_1(X_1, \dots, X_n) \\ f_2(X_1, \dots, X_n) \\ \dots \\ f_m(X_1, \dots, X_n) \end{bmatrix} ; \text{on } f(x) \text{ són } f_{p,i} \text{ i } f_{q,i} \text{ en funció de tots els valors del sistema i de les incògnites amb les consideracions inicials. Hi ha d'haver tantes } f_{(m)} \text{ com } X_n \text{ incògnites}$$

$$Df(x) = \begin{bmatrix} \frac{df_1}{dX_1} & \frac{df_1}{dX_2} & \dots & \frac{df_1}{dX_n} \\ \frac{df_2}{dX_1} & \frac{df_2}{dX_2} & \dots & \frac{df_2}{dX_n} \\ \dots & \dots & \dots & \dots \\ \frac{df_m}{dX_1} & \frac{df_m}{dX_2} & \dots & \frac{df_m}{dX_n} \end{bmatrix}^{-1} ; \text{On } Df(x) \text{ és una matriu de } n \times m \text{ quadrada on es deriva parcialment cada } f(x) , \text{ de } f_{(1)} \text{ a } f_{(m)} , \text{ respecte a } X \text{ de } X_1 \text{ a } X_n .$$

2.2.2.1.1.INTRODUCCIÓ DE DADES

La part del càlcul del flux de càrregues en el programa DaDaSystem es pot configurar a partir de les pestanyes Barres i Elements de transmissió, comentades anteriorment.

En primer lloc cal introduir les dades, tant de busos com dels elements de transmissió en el programa, així doncs en la pestanya de barres esta pensada per introduir les dades de les barres (veure *Dibuix 2*) així mateix com enumerar-les i definir-les. A l'hora de definir els busos tenim les següents opcions:

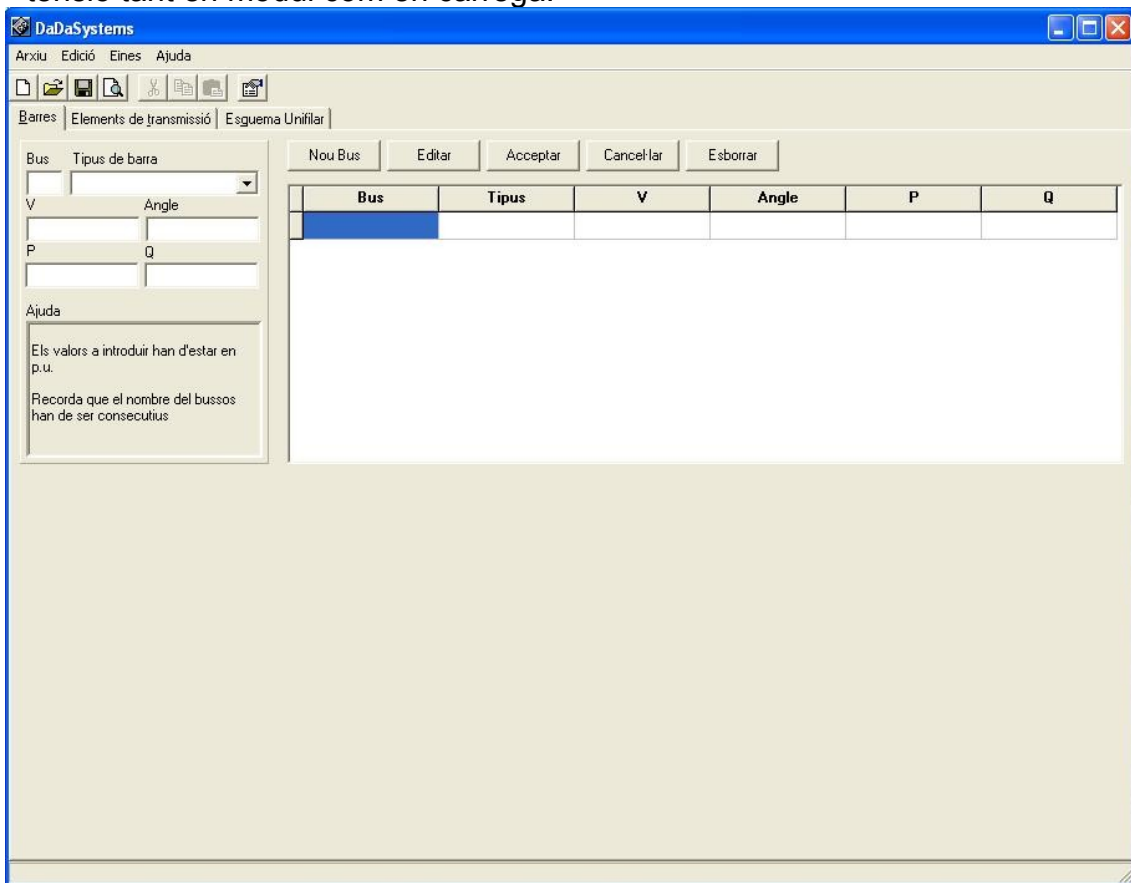
1) Slack Bus: Es coneixen, i mantenen constants, la tensió en modul i argument. Cada sistema només pot disposar de un slack bus.

2) Bus de tensió controlada: Hi ha dos tipus de busos de tensió controlada: PQV i PV.

PQV: Coneixem la tensió en modul però no l'argument en canvi coneixem tant la potència activa com la reactiva. Per transformadors on la toma de regulació servira per mantenir constant aquesta tensió en modul.

PV: Coneixem la tensió en modul però no l'argument i també coneixem la potència activa.

3) Bus de carrega: Tan sols coneixem la potència activa i reactiva. Desconeixem la tensió tant en modul com en càrrega.



Dibuix 2: Interfície d'introducció de les dades de les barres

A més s'han de tenir en compte aquestes consideracions:

1) La numeració de les barres ha de ser consecutiva i han de començar en el nombre 1. La barra 1 ha de ser sempre el bus Slack.

2) L'ordre d'introducció de barres no es important, aquestes s'ordenaran automàticament de 1 a N.

3) Totes les barres han de tenir definit el tipus de barra que és.

4) Cada tipus de barra ha de tenir totes les dades referents al propi tipus de barra. Si aquesta dada no s'incorpora, serà considerada 0.

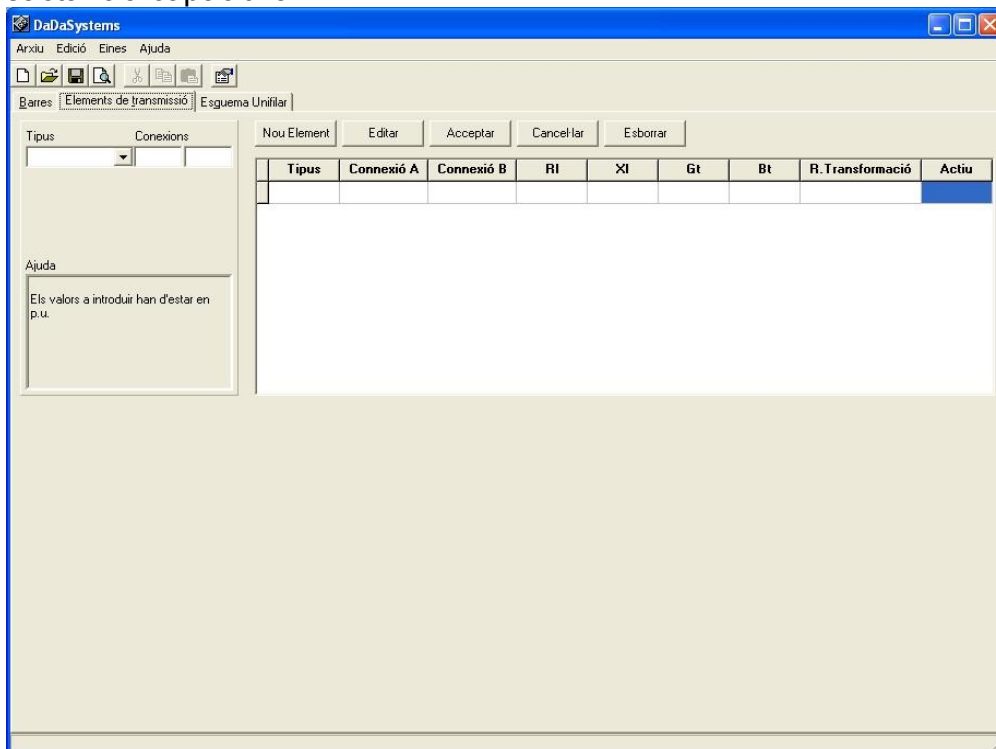
5) Les dades que no corresponguin al tipus de barra concret no es tendran en compte pels càlculs.

En la pestanya de elements de transmissió (veure *Dibuix 3*) esta també pensada per introduir les dades dels elements de connexió entre busos com també les bateries de transformadors. Els tipus d'elements de transmissió disponibles són:

1) Línies: coneixem com a mínim la resistència longitudinal però també podem conèixer la resistència transversal tot i que sol ser pròxima a 0.

2) Transformadors: del transformador coneixem la seva resistència equivalent Z_{cc} a més de la seva toma de regulació. Si es un transformador sense toma de regulació aquesta serà calculada pel programa.

3) Bateria de condensadors: connectada a un sol bus i de la que coneixem la seva reactància capacitiva.



Dibuix 3: Interfície d'introducció de dades dels elements de transmissió

Tenint en compte les següents consideracions:

- 1) Les connexions dels elements ha de fer referència a l'enumeració de les barres introduïdes en *Barres*.
- 2) L'ordre d'introducció dels elements de transmissió no es important.
- 3) Els elements de connexió han de connectar totes les barres, no hi poden haver barres sense connectar.
- 4) Tots els elements de transmissió han d'estar definits amb el tipus d'element que és.
- 5) Les bateries de condensadors ha de tenir el primer punt de connexió en la barra en que estan connectades.
- 6) Cada tipus d'element de transmissió ha de tenir totes les dades referents al propi tipus d'element. Si aquesta dada no s'incorpora, serà considerada 0.
- 7) Les dades que no corresponguin al tipus d'element de transmissió concret no es tendran en compte pels càlculs.
- 8) Revisar si, a l'hora de fer els càlculs, les bateries de condensadors i transformadors amb regulació estan actius o no.

Un cop introduïdes les dades i quan es procedeix al calcul del flux de càrregues el programa ha de procedir a efectuar un seguit d'operacions a fi d'arribar al a conèixer tots els valors de tots els elements de la xarxa.

2.2.2.1.2. RECOINEIXEMENT D'INCOGNITES

En primer lloc el programa analitzara el nombre de càrregues conegudes i el nombre d'elements incògnita dels que disposam. Un poc fet això el programa creara una *array* de $N \times k+1$ elements que anirà modificant segons les iteracions del programa, i al final tendrem una matriu on tendrem totes les solucions.

$$\begin{bmatrix} K_0 1 & K_0 2 & \dots & K_0(n-1) & K_0(n) \\ K_1 1 & K_1 2 & \dots & K_1(n-1) & K_1(n) \\ K_2 1 & K_2 2 & \dots & K_2(n-1) & K_2(n) \\ \dots & \dots & \dots & \dots & \dots \\ K_e 1 & K_e 2 & \dots & K_e(n-1) & K_e(n) \end{bmatrix}$$

On N es el nombre de incògnites que tenim i K és el nombre de iteració realitzada, la matriu anirà creixent mentre l'error en K sigui inferior de l'error preestablert.

Així mateix sabrem que els valor inicials de K_0 seran, segons el tipus d'element, els següents:

Per una barra PQ

$$|Z|=1 \quad \alpha=0$$

Per una barra PV i PQV

$$\alpha=0$$

Per un Transformador

$$a=1$$

2.2.2.1.3.CALCUL DE YBUS

Crearem una *Array of array* quadrada de dimensió N, essent N el nombre de barres del sistema. Com que el llenguatge de programació no accepta nombres complexos, crearem una *Array of array* pels valors reals i una altre pels valors complexos.

Per tant la introducció de dades de una línia vendrà donada per:

$$\begin{aligned} Y_{ii} &= \frac{1}{z_l} + \frac{y_l}{2} & Y_{ij} &= -\frac{1}{z_l} \\ Y_{ji} &= -\frac{1}{z_l} & Y_{jj} &= \frac{1}{z_l} + \frac{y_l}{2} \end{aligned}$$

De on sabem que $Z_l = R + j \cdot X$ i $Y_l = G + j \cdot B$.

Per tant per obtenir la part real i imaginària de $\frac{1}{z_l}$ es farà multiplicant pel seu conjugat:

$\frac{1}{z_l} \cdot \frac{z_l^*}{z_l^*}$ del que obtenim: $\frac{R - j \cdot X}{R^2 + X^2}$ i per tant obtenim els valors reals i imaginàris de Ybus d'una línia:

$$\Re\{z_l\} = \frac{R - X}{R^2 + X^2} ; \quad \Im\{z_l\} = \frac{-X}{R^2 + X^2} ;$$

Per obtenir la part real i imaginària de $\frac{y_l}{2}$ es fera simplement com a:

$$\frac{G}{2} ; \quad \frac{B}{2} ;$$

Per tant el nou Y_{BUS} de línia serà:

Part Real

$$\begin{aligned} \Re\{Y_{ii}\} &= \frac{R}{R^2 + X^2} + \frac{G}{2} & \Re\{Y_{ij}\} &= \frac{-R}{R^2 + X^2} \\ \Re\{Y_{ji}\} &= \frac{-R}{R^2 + X^2} & \Re\{Y_{jj}\} &= \frac{R}{R^2 + X^2} + \frac{G}{2} \end{aligned}$$

I part imaginaria

$$\begin{aligned}\Im\{Y_{ii}\} &= \frac{-I}{R^2 + X^2} + \frac{B}{2} & \Im\{Y_{ij}\} &= \frac{I}{R^2 + X^2} \\ \Im\{Y_{ji}\} &= \frac{i}{R^2 + X^2} & \Im\{Y_{jj}\} &= \frac{-I}{R^2 + X^2} + \frac{B}{2}\end{aligned}$$

Per la introducció de dades d'un transformador donada per:

$$\begin{aligned}Y_{ii} &= \frac{1}{a^2 \cdot Z_{cc}} & Y_{ij} &= \frac{-1}{a \cdot Z_{cc}} \\ Y_{ji} &= \frac{-1}{a \cdot Z_{cc}} & Y_{jj} &= \frac{1}{Z_{cc}}\end{aligned}$$

De on sabem que $Z_{cc} = R + Xi$ i a és la toma de regulació.

Per tant per obtenir la part real i imaginaria de $\frac{1}{Z_{cc}}$ es farà multiplicant pel seu conjugat:

$\frac{1}{Z_{cc}} \cdot \frac{Z_{cc}^*}{Z_{cc}^*}$ del que obtenim: $\frac{R-X}{R^2+X^2}$ i per tant obtenim els valors reals i imaginaris de Y_{BUS} d'una línia:

$$\Re\left\{\frac{1}{Z_{cc}}\right\} = \frac{R}{R^2 + X^2} ; \quad \Im\left\{\frac{1}{Z_{cc}}\right\} = \frac{-X}{R^2 + X^2} ;$$

Per tant el nou Y_{BUS} de línia serà:

Part Real

$$\begin{aligned}\Re\{Y_{ii}\} &= \frac{R}{a^2 \cdot (R^2 + X^2)} & \Re\{Y_{ij}\} &= \frac{-R}{a \cdot (R^2 + X^2)} \\ \Re\{Y_{ji}\} &= \frac{-R}{a \cdot (R^2 + X^2)} & \Re\{Y_{jj}\} &= \frac{R}{R^2 + X^2}\end{aligned}$$

I part imaginaria

$$\begin{aligned}\Im\{Y_{ii}\} &= \frac{-X}{a^2 \cdot (R^2 + X^2)} & \Im\{Y_{ij}\} &= \frac{X}{a \cdot (R^2 + X^2)} \\ \Im\{Y_{ji}\} &= \frac{X}{a \cdot (R^2 + X^2)} & \Im\{Y_{jj}\} &= \frac{-X}{R^2 + X^2}\end{aligned}$$

Per la introducció de una bateria de condensadors seria directe, ja que:

$$\underline{Y}_{BUS} = \underline{Y}_{bc} \quad \text{i per tant;}$$

$$\text{Part real} \quad \Re\{\underline{Y}_{ii}\} = 0$$

$$\text{Part imaginaria} \quad \Im\{\underline{Y}_{ii}\} = Y_{bc}$$

2.2.2.1.4. CALCUL $f_{p,i}$ i $f_{q,i}$

Un cop conegut els t \underline{Y}_{BUS} ant la part real com la part imaginari podem procedir al calcul tan de $f_{p,i}$ i $f_{q,i}$ d'una forma directe.

On:

$$f_{p,i} = P_i - |\underline{V}_i| \cdot \left\{ \sum_{j=1}^n |\underline{V}_j| \cdot [g_{ij} \cdot \cos(\delta_j - \delta_i) - b_{ij} \cdot \sin(\delta_j - \delta_i)] \right\}$$

$$f_{q,i} = q_i + |\underline{V}_i| \cdot \left\{ \sum_{j=1}^n |\underline{V}_j| \cdot [g_{ij} \cdot \sin(\delta_j + \delta_i) - b_{ij} \cdot \cos(\delta_j - \delta_i)] \right\}$$

2.2.2.1.5. CALCUL DE $Df(X)$

A partir de $f_{p,i}$ i $f_{q,i}$, i coneixent les incògnites que necessitam resoldre, podem crear el jacobí. El jacobí esta format per les derivades parcials de $f_{p,i}$ i $f_{q,i}$ respecte cada una de les solucions. Com que el llenguatge de programació no te com a operació cap derivada haurem de fer unes derivades parcials falses.

Sabent que les úniques variables de les que disposam son: $\delta, |\underline{V}|$ i a

Calcularem

La derivada de $\frac{\partial f_{p,i}}{\partial \delta_x}$ és el $\sum_{j=1}^{j=i} \frac{\partial f_{p,i}}{\partial \delta_x}$, però condicionat per:

$$\text{Si } \delta_x = \delta_j$$

$$-|\underline{V}_i| \cdot \left\{ |\underline{V}_j| \cdot [g_{ij} \cdot (-\sin(\delta_j - \delta_i)) - b_{ij} \cdot \cos(\delta_j - \delta_i)] \right\}$$

$$\text{Si } \delta_x = \delta_i$$

$$-|\underline{V}_i| \cdot \left\{ |\underline{V}_j| \cdot [g_{ij} \cdot (\sin(\delta_j - \delta_i)) + b_{ij} \cdot \cos(\delta_j - \delta_i)] \right\}$$

$$\text{Si } \delta_x = \delta_i \quad \text{i} \quad \delta_x = \delta_j$$

$$-|V_i|^2 \cdot b_{ij} \cdot \cos(\delta_j - \delta_i)$$

La derivada de $\frac{\partial f_{p,i}}{\partial |v_x|}$ és el $\sum_{j=1}^{j=n} \frac{\partial f_{p,i}}{\partial |v_x|}$, però condicionat per:

Si $V_x = V_j$

$$-|V_i| \cdot [g_{ij} \cdot \cos(\delta_j - \delta_i) - b_{ij} \cdot \sin(\delta_j - \delta_i)]$$

Si $V_x = V_i$

$$-|V_j| \cdot [g_{ij} \cdot \cos(\delta_j - \delta_i) - b_{ij} \cdot \sin(\delta_j - \delta_i)]$$

Si $V_x = V_j$ i $V_x = V_i$

$$-2 \cdot |V_i| \cdot [g_{ij} \cdot \cos(\delta_j - \delta_i) - b_{ij} \cdot \sin(\delta_j - \delta_i)]$$

La derivada de $\frac{\partial f_{p,i}}{\partial a_x}$ és el $\sum_{j=1}^{j=n} \frac{\partial f_{p,i}}{\partial a_x}$, però condicionat per:

$$-|V_i| \cdot |V_j| \cdot [g_{ij} \cdot \cos(\delta_j - \delta_i) - b_{ij} \cdot \sin(\delta_j - \delta_i)]$$

Tenguen en compte que no s'ha de derivar respecte la funció base sinó que s'ha de derivar Ybus, tant real com imaginari, respecte del factor de regulació: a . Per això tots els valors de Ybus que no contenguin a_x seran 0 i els que contenguin a_x vendran determinats per:

Part Real

$$\Re\{Y_{xixi}\} = \frac{2 \cdot R}{a^3 \cdot (R^2 + X^2)} \quad \Re\{Y_{xixj}\} = \frac{-R}{a^2 \cdot (R^2 + X^2)}$$

$$\Re\{Y_{xjxi}\} = \frac{-R}{a^2 \cdot (R^2 + X^2)} \quad \Re\{Y_{xjxj}\} = 0$$

I part imaginaria

$$\Im\{Y_{xixi}\} = \frac{-2 \cdot X}{|a|^3 \cdot (R^2 + X^2)} \quad \Im\{Y_{xixj}\} = \frac{X}{a^2 \cdot (R^2 + X^2)}$$

$$\Im\{Y_{xjxi}\} = \frac{X}{a^2 \cdot (R^2 + X^2)} \quad \Im\{Y_{xjxj}\} = 0$$

per tant quan $xj = j$ i $xi = i$ s'aplicaran els valors anterior i per la resta de casos el valor serà 0.

La derivada de $\frac{\partial f_{q,i}}{\partial \delta_x}$ és el $\sum_{j=1}^{j=n} \frac{\partial f_{q,i}}{\partial \delta_x}$, però condicionat per:

Si $\delta_x = \delta_j$

$$|L_i| \cdot \{|L_j| \cdot [g_{ij} \cdot (\cos(\delta_j - \delta_i)) - b_{ij} \cdot \sin(\delta_j - \delta_i)]\}$$

Si $\delta_x = \delta_i$

$$|L_i| \cdot \{|L_j| \cdot [-g_{ij} \cdot (\sin(\delta_j - \delta_i)) - b_{ij} \cdot \cos(\delta_j - \delta_i)]\}$$

Si $\delta_x = \delta_i$ i $\delta_x = \delta_j$

$$|L_i|^2 \cdot g_{ij} \cdot \cos(\delta_j - \delta_i)$$

La derivada de $\frac{\partial f_{q,i}}{\partial |v_x|}$ és el $\sum_{j=1}^{j=n} \frac{\partial f_{q,i}}{\partial |v_x|}$, però condicionat per:

Si $V_x = V_j$

$$|L_i| \cdot [g_{ij} \cdot \sin(\delta_j - \delta_i) + b_{ij} \cdot \cos(\delta_j - \delta_i)]$$

Si $V_x = V_i$

$$|L_j| \cdot [g_{ij} \cdot \sin(\delta_j - \delta_i) + b_{ij} \cdot \cos(\delta_j - \delta_i)]$$

Si $V_x = V_j$ i $V_x = V_i$

$$2 \cdot |L_i| \cdot [g_{ij} \cdot \sin(\delta_j - \delta_i) + b_{ij} \cdot \cos(\delta_j - \delta_i)]$$

La derivada de $\frac{\partial f_{q,i}}{\partial a_x}$ és el $\sum_{j=1}^{j=n} \frac{\partial f_{q,i}}{\partial a_x}$, però condicionat per:

$$|L_i| \cdot |L_j| \cdot [g_{ij} \cdot \sin(\delta_j - \delta_i) + b_{ij} \cdot \cos(\delta_j - \delta_i)]$$

Tenguen en compte que no s'ha de derivar respecte la funció base sinó que s'ha de derivar Ybus, tant real com imaginari, respecte del factor de regulació: a . Per això tots els valors de Ybus que no contenguin a_x seran 0 i els que contenguin a_x vendran determinats per:

Part Real

$$\Re(Y_{xixi}) = \frac{2 \cdot R}{a^3 \cdot (R^2 + X^2)} \quad \Re(Y_{xixj}) = \frac{-R}{a^2 \cdot (R^2 + X^2)}$$

$$\Re(Y_{xixi}) = \frac{-R}{a^2 \cdot (R^2 + X^2)} \quad \Re(Y_{xixj}) = 0$$

I part imaginaria

$$\Im(Y_{xixi}) = \frac{-2 \cdot X}{|a|^3 \cdot (R^2 + X^2)} \quad \Im(Y_{xixj}) = \frac{X}{a^2 \cdot (R^2 + X^2)}$$

$$\Im(Y_{xixi}) = \frac{X}{a^2 \cdot (R^2 + X^2)} \quad \Im(Y_{xixj}) = 0$$

per tant quan $xj = j$ i $xi = i$ s'aplicaran els valors anterior i per la resta de casos el valor serà 0.

2.2.2.1.6. CALCUL DE $Df(X)^{-1}$

4) Matriu inversa del jacobi

Per fer la matriu inversa del jacobi A , utilitzarem el mètode Gauss-Jordan. Aquest mètode, per mitja de la matriu identitat I podem obtenir, i la utilització d'operacions elementals per files ens proporciona la matriu inversa: A^{-1} .

Es partira de $\langle A|I \rangle$ per arribar a $\langle I|A^{-1} \rangle$ i per això es seguirà aquest procediment t senzill que te un algoritme molt simple:

Sabent que és una matriu quadrada de $n \times m$, on n i m és el nombre d'incògnites del sistema. Tenim que:

$$\langle A|I \rangle = \left(\begin{array}{cccc|cccc} A_{11} & A_{12} & \dots & A_{1n} & 1 & 0 & \dots & 0 \\ A_{21} & A_{22} & \dots & A_{2n} & 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ A_{m1} & A_{m2} & \dots & A_{mn} & 0 & 0 & \dots & 1 \end{array} \right)$$

Es comença dividint l'element que te els dos subíndex iguals per tota la fila, pel que obtenim:

$$\left(\begin{array}{cccc|cccc} A_{11}/A_{11} & A_{12}/A_{11} & \dots & A_{1n}/A_{11} & 1/A_{11} & 0/A_{11} & \dots & 0/A_{11} \\ A_{21} & A_{22} & \dots & A_{2n} & 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ A_{m1} & A_{m2} & \dots & A_{mn} & 0 & 0 & \dots & 1 \end{array} \right)$$

per tant

$$\left(\begin{array}{cccc|cccc} 1 & A_{12}/A_{11} & \dots & A_{1n}/A_{11} & 1/A_{11} & 0 & \dots & 0 \\ A_{21} - A_{21} & A_{22} - A_{12}/A_{11} \cdot A_{22} & \dots & A_{2n} - A_{1n}/A_{11} \cdot A_{2n} & 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ A_{m1} & A_{m2} & \dots & A_{mn} & 0 & 0 & \dots & 1 \end{array} \right)$$

Ara seguim el següent algoritme dese la fila 1 fins a la m , sense contar la fila en que

estem treballant, i multiplicam la fila en la treballam per cada una de les files en valor negatiu fins a arribar a A_m

$$\left(\begin{array}{cccc|cccc} 1 & A_{12}/A_{11} & \dots & A_{1n}/A_{11} & 1/A_{11} & 0 & \dots & 0 \\ A_{21}-A_{21} & A_{22}-A_{12}/A_{11}\cdot A_{22} & \dots & A_{2n}-A_{1n}/A_{11}\cdot A_{2n} & -1/A_{11} & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ A_{m1} & A_{m2} & \dots & A_{mn} & 0 & 0 & \dots & 1 \end{array} \right)$$

Si seguim obtenim:

$$\left(\begin{array}{cccc|cccc} 1 & A_{12}/A_{11} & \dots & A_{1n}/A_{11} & 1/A_{11} & 0 & \dots & 0 \\ 0 & A_{22}-A_{12}/A_{11}\cdot A_{22} & \dots & A_{2n}-A_{1n}/A_{11}\cdot A_{2n} & -1/A_{11} & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & A_{m2}-A_{12}/A_{11}\cdot A_{m2} & \dots & A_{mn}-A_{1n}/A_{11}\cdot A_{mn} & -1/A_{11} & 0 & \dots & 1 \end{array} \right)$$

Si procedim amb aquest algoritme des de la columna 1 fins a la columna n obtenim la matriu inversa de $Df(X)$:

$$\left(\begin{array}{cccc|cccc} 1 & 0 & \dots & 0 & A_{11}^{-1} & A_{12}^{-1} & \dots & A_{1n}^{-1} \\ 0 & 1 & \dots & 0 & A_{21}^{-1} & A_{22}^{-1} & \dots & A_{2n}^{-1} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 & A_{m1}^{-1} & A_{m2}^{-1} & \dots & A_{mn}^{-1} \end{array} \right)$$

2.2.2.1.7. CALCUL $X_{(k+1)}$ DE L'ERROR I ITERACIÓ

Coneixent $X_{(k)}$ es fàcil obtenir el valor de $X_{(k+1)}$ a partir del producte de $Df(X)^{-1}$ i $f(X)$. Arribant a la formula anunciada al anteriorment:

$$x^{(k+1)} = x^{(k)} - \{ Df(x^{(k)})^{-1} \cdot f(x^{(k)}) \}$$

El producte vectorial de una matriu $Df(X)^{-1}$ i un vector $f(X)$ es fa seguint el següent procediment:

$$\begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \cdot \begin{bmatrix} X_1 \\ X_2 \end{bmatrix} \quad \text{sabem que:}$$

$$Y_1 = A \cdot X_1 + B \cdot X_2 \quad ; \quad Y_2 = C \cdot X_1 + D \cdot X_2 \quad .$$

Així per tant ja coneixem $X_{(k+1)}$.

Un pic elaborat el cicle iteratiu s'ha de crear la condició d'iteració. Aquesta condició no serà altre que l'error obtingut sigui inferior a l'error preestablert inicial ment per l'usuari.

L'error que obtenim, ha de ser el major dels error, es a dir, tots els elements han de tenir un error inferior al preestablert.

L'error en $X_{(k+1)}$ és simplement l'increment de $X_{(k+1)}$ respecte $X_{(k)}$, per tant:

$$\Delta X_{max} < \text{Error màxim}$$

Error màxim predeterminat en el programa es de $1 \cdot 10^{-6}$ tot i que es pot modificar des de *Eines/Opcions*.

Mentre que l'error de X_{max} sigui superior a l'error màxim s'ha de seguir iterant en el programa.

2.2.2.1.7. CALCUL DEL FLUX DE CÀRREGUES

Per definir el flux de càrregues i coneixent la tensió en forma polar definirem la part real i la part imaginària dels busos de connexió dels elements:

$$\Re\{V_1\} = \underline{V}_1 \cdot \cos(\delta_1) \quad ; \quad \Im\{V_1\} = \underline{V}_1 \cdot \sin(\delta_1)$$

$$\Re\{V_2\} = \underline{V}_2 \cdot \cos(\delta_2) \quad ; \quad \Im\{V_2\} = \underline{V}_2 \cdot \sin(\delta_2)$$

Per el càlcul del flux de càrregues en una línia ve definit per

$$\underline{S}_{12}^* = \underline{u}_1^* \cdot \left[\frac{y_l}{2} \cdot \underline{u}_1 - \frac{\underline{u}_1 - \underline{u}_2}{z_l} \right] \quad \text{i} \quad \underline{S}_{21}^* = \underline{u}_2^* \cdot \left[\frac{y_l}{2} \cdot \underline{u}_2 - \frac{\underline{u}_2 - \underline{u}_1}{z_l} \right] \quad \text{del que podem definir que}$$

per a

Per a

$$\underline{S}_{12}^* = \underline{u}_1^* \cdot \underline{u}_1 \cdot \left[\frac{y_l}{2} + \frac{1}{z_l} \right] - \underline{u}_1^* \cdot \left[\frac{1}{z_l} \cdot \underline{u}_2 \right] \quad ; \quad \underline{S}_{12}^* = \underline{u}_1^* \cdot \left[\frac{y_l}{2} \cdot \underline{u}_1 - \frac{\underline{u}_1 - \underline{u}_2}{z_l} \right]$$

Podem dividir el producte i obtenim que

De $-\frac{\underline{u}_1 - \underline{u}_2}{z_l}$ podem obtenir les seves parts real i imaginària

Real

$$\Re \left\{ -\frac{\underline{u}_1 - \underline{u}_2}{z_l} \right\} = \frac{(\Re\{\underline{u}_1\} - \Re\{\underline{u}_2\}) \cdot \Re\{z_l\}}{\Re\{z_l\}^2 + \Im\{z_l\}^2} + \frac{(\Im\{\underline{u}_1\} - \Im\{\underline{u}_2\}) \cdot \Im\{z_l\}}{\Re\{z_l\}^2 + \Im\{z_l\}^2}$$

Imaginària

$$\Im \left\{ -\frac{\underline{u}_1 - \underline{u}_2}{z_l} \right\} = \frac{(\Im\{\underline{u}_1\} - \Im\{\underline{u}_2\}) \cdot \Re\{z_l\}}{\Re\{z_l\}^2 + \Im\{z_l\}^2} + \frac{(\Re\{\underline{u}_1\} - \Re\{\underline{u}_2\}) \cdot \Im\{z_l\}}{\Re\{z_l\}^2 + \Im\{z_l\}^2}$$

De $\frac{y_l}{2} \cdot \underline{u}_1$ podem obtenir les seves parts real i imaginària

Real

$$\Re \left\{ \frac{y_t}{2} \cdot u_1 \right\} = \frac{\Re \{y_t\}}{2} \cdot \Re \{u_1\} - \frac{\Im \{y_t\}}{2} \cdot \Im \{u_1\}$$

Imaginari

$$\Im \left\{ \frac{y_t}{2} \cdot u_1 \right\} = \frac{\Im \{y_t\}}{2} \cdot \Re \{u_1\} + \frac{\Re \{y_t\}}{2} \cdot \Im \{u_1\}$$

Del que obtenim:

Real

$$\Re \left\{ \frac{y_t}{2} \cdot u_1 - \frac{u_1 - u_2}{z_t} \right\} = \frac{(\Re \{u_1\} - \Re \{u_2\}) \cdot \Re \{z_t\}}{\Re \{z_t\}^2 + \Im \{z_t\}^2} + \frac{(\Im \{u_1\} - \Im \{u_2\}) \cdot \Im \{z_t\}}{\Re \{z_t\}^2 + \Im \{z_t\}^2} + \frac{\Re \{y_t\}}{2} \cdot \Re \{u_1\} - \frac{\Im \{y_t\}}{2} \cdot \Im \{u_1\}$$

Imaginari

$$\Im \left\{ \frac{y_t}{2} \cdot u_1 - \frac{u_1 - u_2}{z_t} \right\} = \frac{(\Im \{u_1\} - \Im \{u_2\}) \cdot \Re \{z_t\}}{\Re \{z_t\}^2 + \Im \{z_t\}^2} + \frac{(\Re \{u_1\} - \Re \{u_2\}) \cdot \Im \{z_t\}}{\Re \{z_t\}^2 + \Im \{z_t\}^2} + \frac{\Im \{y_t\}}{2} \cdot \Re \{u_1\} + \frac{\Re \{y_t\}}{2} \cdot \Im \{u_1\}$$

Per tant podem saber que

$$P_{12} = \Re \{S_{12}^*\} = \Re \{u_1^*\} \cdot \Re \left\{ \frac{y_t}{2} \cdot u_1 - \frac{u_1 - u_2}{z_t} \right\} + \Im \{u_1^*\} \cdot \Im \left\{ \frac{y_t}{2} \cdot u_1 - \frac{u_1 - u_2}{z_t} \right\}$$

$$Q_{12} = \Im \{S_{12}^*\} = -\Re \{u_1^*\} \cdot \Im \left\{ \frac{y_t}{2} \cdot u_1 - \frac{u_1 - u_2}{z_t} \right\} + \Im \{u_1^*\} \cdot \Re \left\{ \frac{y_t}{2} \cdot u_1 - \frac{u_1 - u_2}{z_t} \right\}$$

Per a

$$S_{21}^* = u_2^* \cdot \left[\frac{-1}{z_t} \cdot u_1 + \left[\frac{y_t}{2} + \frac{1}{y_t} \right] \cdot u_2 \right] ; \quad S_{21}^* = u_2^* \cdot \left[\frac{y_t}{2} \cdot u_2 - \frac{u_2 - u_1}{z_t} \right]$$

Podem dividir el producte i obtenim que

De $-\frac{u_2 - u_1}{z_t}$ podem obtenir les seves parts real i imaginaria

Real

$$\Re \left\{ -\frac{u_2 - u_1}{z_t} \right\} = \frac{(\Re \{u_2\} - \Re \{u_1\}) \cdot \Re \{z_t\}}{\Re \{z_t\}^2 + \Im \{z_t\}^2} + \frac{(\Im \{u_2\} - \Im \{u_1\}) \cdot \Im \{z_t\}}{\Re \{z_t\}^2 + \Im \{z_t\}^2}$$

Imaginaria

$$\Im \left\{ -\frac{u_2 - u_1}{z_t} \right\} = \frac{(\Im \{u_2\} - \Im \{u_1\}) \cdot \Re \{z_t\}}{\Re \{z_t\}^2 + \Im \{z_t\}^2} + \frac{(\Re \{u_2\} - \Re \{u_1\}) \cdot \Im \{z_t\}}{\Re \{z_t\}^2 + \Im \{z_t\}^2}$$

De $\frac{y_t}{2} \cdot u_2$ podem obtenir les seves parts real i imaginaria

Real

$$\Re \left\{ \frac{y_t}{2} \cdot u_2 \right\} = \frac{\Re \{y_t\}}{2} \cdot \Re \{u_2\} - \frac{\Im \{y_t\}}{2} \cdot \Im \{u_2\}$$

Imaginari

$$\Im \left\{ \frac{y_t}{2} \cdot u_2 \right\} = \frac{\Im \{y_t\}}{2} \cdot \Re \{u_2\} + \frac{\Re \{y_t\}}{2} \cdot \Im \{u_2\}$$

Per tant podem saber que

$$P_{21} = \Re \{S_{21}^*\} = \Re \{u_2^*\} \cdot \Re \left\{ \frac{y_t}{2} \cdot u_2 - \frac{u_2 - u_1}{z_t} \right\} + \Im \{u_2^*\} \cdot \Im \left\{ \frac{y_t}{2} \cdot u_2 - \frac{u_2 - u_1}{z_t} \right\}$$

$$Q_{21} = \Im \{S_{21}^*\} = -\Re \{u_2^*\} \cdot \Im \left\{ \frac{y_t}{2} \cdot u_2 - \frac{u_2 - u_1}{z_t} \right\} + \Im \{u_2^*\} \cdot \Re \left\{ \frac{y_t}{2} \cdot u_2 - \frac{u_2 - u_1}{z_t} \right\}$$

D'aquesta forma, el flux de potències a través del transformador vendrà serà vendrà definit per

$$S_{12}^* = u_1^* \cdot \left[\frac{u_1}{a^2 \cdot z_{cc}} - \frac{u_2}{a \cdot z_{cc}} \right] \quad \text{i} \quad S_{21}^* = u_2^* \cdot \left[-\frac{u_1}{a \cdot z_{cc}} + \frac{u_2}{z_{cc}} \right] \quad \text{del que podem definir que}$$

per a

$$S_{12}^* = u_1^* \cdot \left[\frac{u_1}{a^2 \cdot z_{cc}} - \frac{u_2}{a \cdot z_{cc}} \right]$$

Podem dividir el producte i obtenim que

De $\frac{u_1}{a^2 \cdot z_{cc}}$ podem obtenir les seves parts real i imaginaria

Real

$$\Re \left\{ \frac{u_1}{a^2 \cdot z_{cc}} \right\} = \frac{\Re \{u_1\} \cdot \Re \{z_{cc}\}}{a^2 \cdot (\Re \{z_{cc}\}^2 + \Im \{z_{cc}\}^2)} - \frac{\Im \{u_1\} \cdot \Im \{z_{cc}\}}{a^2 \cdot (\Re \{z_{cc}\}^2 + \Im \{z_{cc}\}^2)}$$

Imaginaria

$$\Im \left\{ \frac{u_1}{a^2 \cdot z_{cc}} \right\} = \frac{\Im \{u_1\} \cdot \Re \{z_{cc}\}}{a^2 \cdot (\Re \{z_{cc}\}^2 + \Im \{z_{cc}\}^2)} - \frac{\Re \{u_1\} \cdot \Im \{z_{cc}\}}{a^2 \cdot (\Re \{z_{cc}\}^2 + \Im \{z_{cc}\}^2)}$$

De $\frac{\underline{u}_2}{a \cdot \underline{z}_{cc}}$ podem obtenir les seves parts real i imaginaria

$$\begin{aligned} \text{Real} \\ \Re \left\{ \frac{\underline{u}_2}{a \cdot \underline{z}_{cc}} \right\} &= \frac{\Re\{\underline{u}_2\} \cdot \Re\{\underline{z}_{cc}\}}{a \cdot (\Re\{\underline{z}_{cc}\}^2 + \Im\{\underline{z}_{cc}\}^2)} - \frac{\Im\{\underline{u}_2\} \cdot \Im\{\underline{z}_{cc}\}}{a \cdot (\Re\{\underline{z}_{cc}\}^2 + \Im\{\underline{z}_{cc}\}^2)} \\ \text{Imaginaria} \\ \Im \left\{ \frac{\underline{u}_2}{a \cdot \underline{z}_{cc}} \right\} &= \frac{\Im\{\underline{u}_2\} \cdot \Re\{\underline{z}_{cc}\}}{a \cdot (\Re\{\underline{z}_{cc}\}^2 + \Im\{\underline{z}_{cc}\}^2)} - \frac{\Re\{\underline{u}_2\} \cdot \Im\{\underline{z}_{cc}\}}{a \cdot (\Re\{\underline{z}_{cc}\}^2 + \Im\{\underline{z}_{cc}\}^2)} \end{aligned}$$

Per tant podem saber que

$$\begin{aligned} P_{21} = \Re\{\underline{S}_{21}^*\} &= \Re\{\underline{u}_1^*\} \cdot \Re \left\{ \frac{\underline{u}_1}{a^2 \cdot \underline{z}_{cc}} - \frac{\underline{u}_2}{a \cdot \underline{z}_{cc}} \right\} + \Im\{\underline{u}_1^*\} \cdot \Im \left\{ \frac{\underline{u}_1}{a^2 \cdot \underline{z}_{cc}} - \frac{\underline{u}_2}{a \cdot \underline{z}_{cc}} \right\} \\ Q_{21} = \Im\{\underline{S}_{21}^*\} &= \Im\{\underline{u}_1^*\} \cdot \Re \left\{ \frac{\underline{u}_1}{a^2 \cdot \underline{z}_{cc}} - \frac{\underline{u}_2}{a \cdot \underline{z}_{cc}} \right\} + \Re\{\underline{u}_1^*\} \cdot \Im \left\{ \frac{\underline{u}_1}{a^2 \cdot \underline{z}_{cc}} - \frac{\underline{u}_2}{a \cdot \underline{z}_{cc}} \right\} \end{aligned}$$

per a

$$\underline{S}_{21}^* = \underline{u}_2^* \cdot \left[-\frac{\underline{u}_1}{a \cdot \underline{z}_{cc}} + \frac{\underline{u}_2}{\underline{z}_{cc}} \right]$$

Podem dividir el producte i obtenim que

De $\frac{\underline{u}_1}{a \cdot \underline{z}_{cc}}$ podem obtenir les seves parts real i imaginaria

$$\begin{aligned} \text{Real} \\ \Re \left\{ \frac{\underline{u}_1}{a \cdot \underline{z}_{cc}} \right\} &= \frac{\Re\{\underline{u}_1\} \cdot \Re\{\underline{z}_{cc}\}}{a \cdot (\Re\{\underline{z}_{cc}\}^2 + \Im\{\underline{z}_{cc}\}^2)} - \frac{\Im\{\underline{u}_1\} \cdot \Im\{\underline{z}_{cc}\}}{a \cdot (\Re\{\underline{z}_{cc}\}^2 + \Im\{\underline{z}_{cc}\}^2)} \\ \text{Imaginaria} \\ \Im \left\{ \frac{\underline{u}_1}{a \cdot \underline{z}_{cc}} \right\} &= \frac{\Im\{\underline{u}_1\} \cdot \Re\{\underline{z}_{cc}\}}{a \cdot (\Re\{\underline{z}_{cc}\}^2 + \Im\{\underline{z}_{cc}\}^2)} - \frac{\Re\{\underline{u}_1\} \cdot \Im\{\underline{z}_{cc}\}}{a \cdot (\Re\{\underline{z}_{cc}\}^2 + \Im\{\underline{z}_{cc}\}^2)} \end{aligned}$$

De $\frac{u_2}{z_{cc}}$ podem obtenir les seves parts real i imaginaria

Real

$$\Re \left\{ \frac{u_2}{z_{cc}} \right\} = \frac{\Re\{u_2\} \cdot \Re\{z_{cc}\}}{\Re\{z_{cc}\}^2 + \Im\{z_{cc}\}^2} - \frac{\Im\{u_2\} \cdot \Im\{z_{cc}\}}{\Re\{z_{cc}\}^2 + \Im\{z_{cc}\}^2}$$

Imaginaria

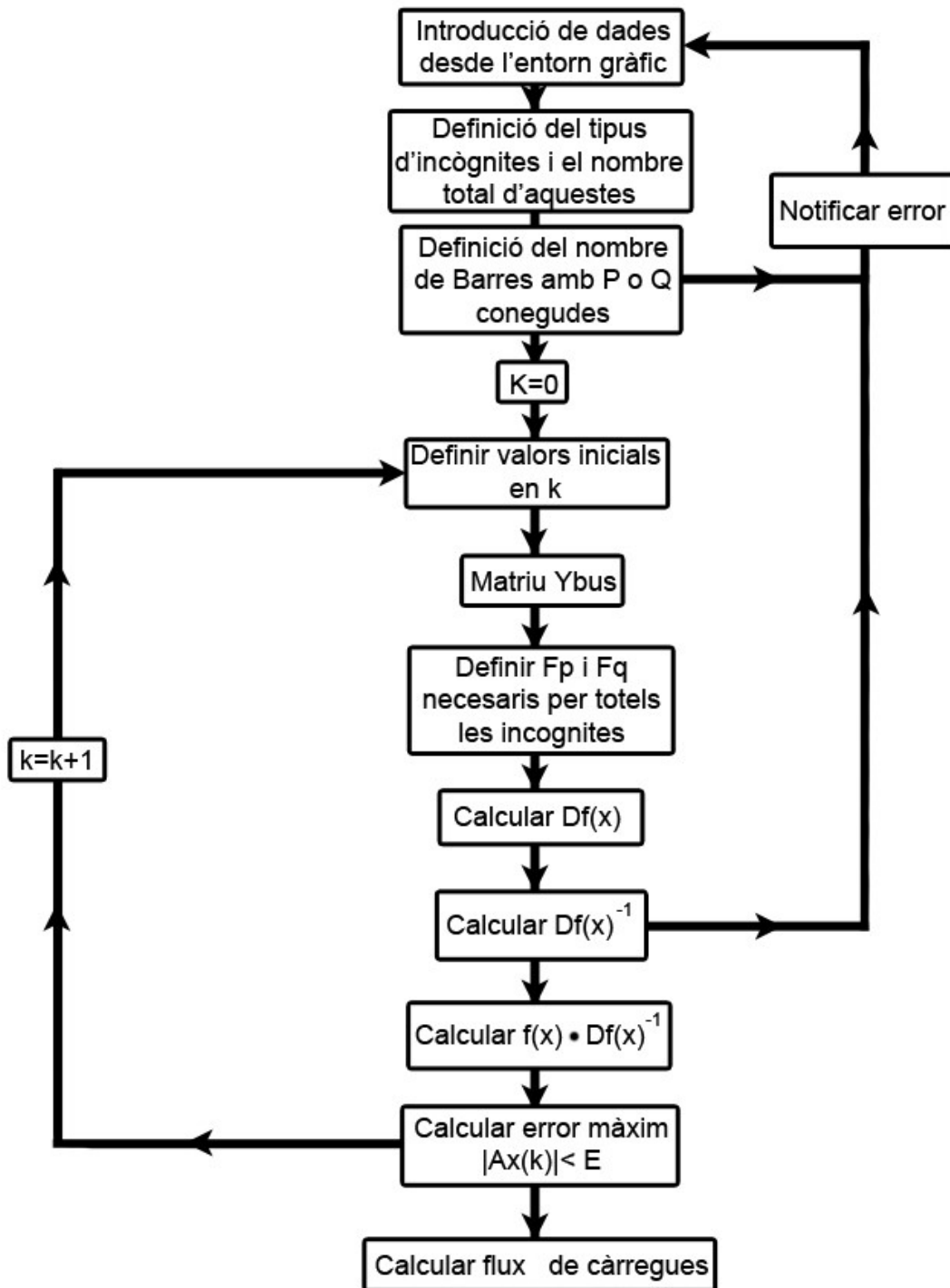
$$\Im \left\{ \frac{u_2}{z_{cc}} \right\} = \frac{\Im\{u_2\} \cdot \Re\{z_{cc}\}}{\Re\{z_{cc}\}^2 + \Im\{z_{cc}\}^2} - \frac{\Re\{u_2\} \cdot \Im\{z_{cc}\}}{\Re\{z_{cc}\}^2 + \Im\{z_{cc}\}^2}$$

Per tant podem saber que

$$P_{12} = \Re\{S_{12}^*\} = \Re\{u_2^*\} \cdot \Re \left\{ \frac{u_1}{a \cdot z_{cc}} - \frac{u_2}{z_{cc}} \right\} + \Im\{u_2^*\} \cdot \Im \left\{ \frac{u_1}{a \cdot z_{cc}} - \frac{u_2}{z_{cc}} \right\}$$

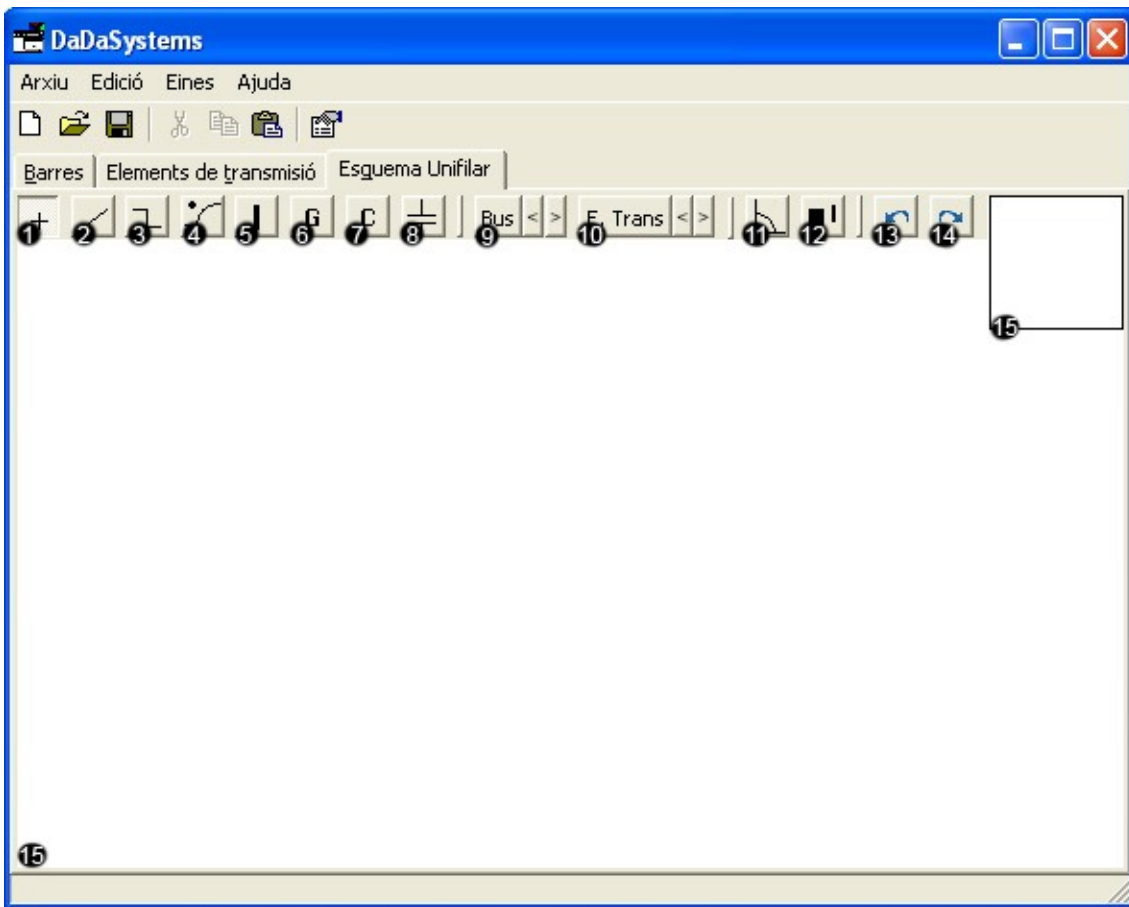
$$Q_{12} = \Im\{S_{12}^*\} = \Im\{u_2^*\} \cdot \Re \left\{ \frac{u_1}{a \cdot z_{cc}} - \frac{u_2}{z_{cc}} \right\} + \Re\{u_2^*\} \cdot \Im \left\{ \frac{u_1}{a \cdot z_{cc}} - \frac{u_2}{z_{cc}} \right\}$$

2.2.2.1.8. ESQUEMA ITERATIU



2.2.2.2.ESQUEMA UNIFILAR

Per la realització de l'esquema unifilar es disposa de un entorn gràfic interactiu en els que es pot accedir des de la pestanya Esquema Unifilar.



Dibuix 4: Interfície de l'esquema unifilar

La pantalla de l'esquema unifilar ens permet la introducció de tots aquells elements en que ens podem trobar, per el cas concret de l'estudi. A més el programa permet la introducció de llegendes tant dels busos com dels elements de transmissió implementats a la xarxa. També inclou un sistema de rotació de les imatges que ens permetrà col·locar els elements en quatre posicions distintes (0, 90, 180 i 270 graus, veure 12 en *Dibuix 4*) com també activar o desactivar la miniatura del propi element. A part també disposam dels elements de pas endavant (veure 14 en *Dibuix 4*) i pas en darrera (veure 13 en *Dibuix 4*).

Els principals elements dels que disposem a l'hora de treballar son els següents:

- 1)Cursor(veure 1 en *Dibuix 4*): aquesta funció et permet desplaçar-te per la interfície sense fer cap modificació en ella.
- 2)Barra(veure 5 en *Dibuix 4*): es la representació d'una gràfica de una barra a la qual es connectaran tots els altres elements. La barra es la representació de un nus de tensió comuna de tots els elements connectats.

3) Generador (veure 6 en *Dibuix 4*): es la representació d'un generador, s'utilitzara principalment per el bus slack o una barra PV.

4) Càrrega (veure 7 en *Dibuix 4*): es la representació d'una càrrega. Esta pensat per barres PQ, PV o PQV, però no es requerida la seva implementació a l'esquema.

5) Bateria de condensadors (veure 8 en *Dibuix 4*): es la representació d'una bateria de condensadors.

6) Línia discontinua (veure 2 en *Dibuix 4*) o continua (veure 3 en *Dibuix 4*): Aquest element esta pensat especialment per la creació de línies de transmissió entre dues barres i en el primer punt que es marqui, serà el punt de inici, el següent serà el punt final. Si es tracta de la línia continua es seguira la línia fins que es seleccioni un nou element. Inclou també la opció de crear línies paral·leles al pla de treball

7) Transformador (veure 4 en *Dibuix 4*): es la representació del transformador, on amb un punt es marca el costat d'alta tensió.

8) Llegenda: Tant la de Barres (veure 9 en *Dibuix 4*) com la d'elements de transmissió (veure 10 en *Dibuix 4*) consten dels principals valors inicials de cada un dels tipus d'elements, així doncs cal primer introduir les dades per al calcul iteratiu abans de poder implementar aquestes llegendes ja que vendran donades per els elements dels que disposem en l'arxiu.

9) Quadricula: Quadricula, com el mateix nom indica, tot l'espai de dibuix. D'aquesta forma es fa mes fàcil posicionar els objectes i estructura l'esquema unifilar.

10) Forçat: Si aquest botó esta clickat, obliga que els punt de referència siguin els de la quadricula, per tant es mes fàcil mantenir diferents línies o objectes definits en un mateix punt concret.

11) Imatge en Miniatura: La representació de la imatge en miniatura (que es pot veure en veure 15 de *Dibuix 4*) és la representació en miniatura de l'element que estam treballant. Es pot activar o desactivar des de el seu boto (veure 12 en *Dibuix 4*).

Tots aquest elements són els que disposam per poder inserir en la nostra interfície de treball (veure 16 en *Dibuix 4*) per poder realitzar tots els esquemes necessaris per el càlcul del flux de càrregues.

Tota aquesta informació serà traslladada a la fulla d'exportació però no es un element indispensable per poder imprimir els càlculs resolts.

2.2.2.3.CALCUL DEL FLUX DE CÀRREGUES

L'exportació de dades es farà posteriorment al desenvolupament del calcul de tot el sistema. Per aquest càlcul, i posterior exportació de dades, es necessari que ja s'haguin calculat tots els valors de tensió (V , tant en modul com en fase) així com les tomes de regulació dels transformadors.

Per tant coneixerem tots els valors de les barres com també la matriu Y_{bus} de tot el sistema. Amb aquesta informació es calculara el flux de càrregues entres les diferents barres del sistema i les perdues en aquestes.

Un com conegut tot això, el resultat del flux de càrregues, vendrà determinada per:

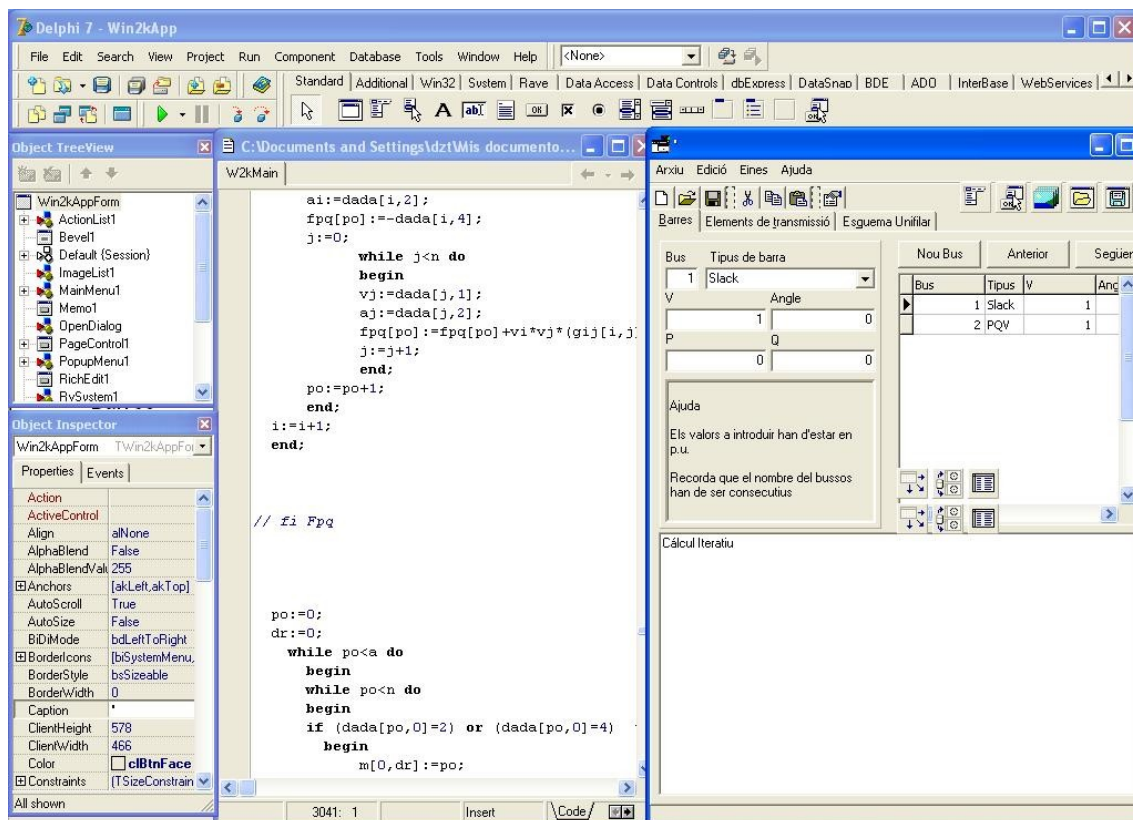
1. L'esquema unifilar, si cal, ja que no es requerit.
2. Dades inicials del sistema tant de barres com dels elements de transmissió
3. Solució iterativa on es mostren tots els elements amb els seus valors reals.
4. Càlcul del flux de càrregues entre tots els elements del sistema.

2.3. DIFICULTATS DE PROGRAMACIÓ

2.3.1. SELECCIÓ DEL PROGRAMARI

A l'hora de començar la programació es tenia com a primeres alternatives *Visual Basic* i *Turbo Delphi* entre altres alternatives com *Flash* o *Html*, per un desenvolupament web. Al final es va decantar per *Turbo Delphi* perquè, tot i ser una mica més complex que altres alternatives com *Visual Basic*, aporta un resultat molt més productiu i utilitza més profitosament els recursos de que disposa tant el propi programa com el sistema operatiu en el qual s'aplica.

El primer pas important a l'hora de la programació en *Turbo Delphi* és el pas de la programació clàssica a la *PPO*. Tot i que a la llarga simplifica el volum de treball, en un principi es complica d'entendre i fer funcionar, però un cop adaptat al sistema ens trobam en un entorn com el del *Dibuix 5*.



Dibuix 5: Entorn de treball

On disposam de un entorn de programació purament en codi i una altre part on disposam de la acció interactiva de disseny on podem afegir els diferents elements i modificar-los segons les nostres necessitats.

2.3.2. RESOLUCIÓ DELS CÀLCULS

La introducció i funcionament del sistema s'havia pensat completament per un sistema basat amb bases de dades però al final es va desestimar la idea ja que la seva funcionalitat esta molt reduïda tot i que la presentació i l'entorn gràfic respecte bases de dades era molt dinàmic i elegant, però d'això tan sols se n'ha mantingut la interfície d'introducció de dades.

Des de un punt de vista de calcul el problema ha estat el haver de treballar directament amb nombres reals i no disposar de funcions matemàtiques complexes, tan sols de les

elementals com s'ha comentat abans, per poder resoldre problemes com la matriu inversa o les derivades parcials. Per tant hi ha hagut una certa complicació a l'hora d'elaborar aquestes funcions i com aplicar-les en cada cas concret del sistema. Igualment hi ha hagut uns certs problemes a l'hora de calcular $f_{p,i}$ i $f_{q,i}$ ja que aquest s'havia de calcular progressivament i posteriorment s'havien de derivar parcialment i tot això tenint tots els valors introduïts ja que en funcionament no es pot tractar com a funcions. Per tant s'havia de crear un ordre clar de successió a l'hora de crear $f_{p,i}$ i $f_{q,i}$ i les derivades d'aquests a fi de que el jacobí sortís correctament i trobar els valors correctes en cada iteració.

2.3.3. ESQUEMA UNIFILAR

Per la creació de l'esquema unifilar s'havia pensat en crear un sistema automàtic de creació de línies, transformadors, busos... però en totes les opcions plantejades no es va obtenir cap resultat satisfactori ja que en petits sistemes si que funcionava correctament però amb sistemes més complexos sempre hi havia entrecreuament de línies o males connexions. En vista en diverses alternatives proposades o dutes a termes per altres softwares ens vam decantar per la introducció manual dels elements, pot ser més tediosa però al cap i a la fi aporten els criteris de creació que tan sols un usuari, en la creació en viu, pot elaborar.

3. CONCLUSIONS

El projecte ha constatat de dues parts principals: l'elaboració de un estudi matemàtic per la resolució del flux de carregues i la creació de un programa informàtic per facilitar-ne els càlculs.

El flux de carregues, pel sistema de Newton-Raphson, te una serie de problemes a resoldre per mitjà de llenguatge informàtic ja que no consta de tractament de calcul de matrius, derivades, nombres complexos... Per tant s'ha resolt aquesta part de programació simplificant alguns càlculs, ja que es tracten una serie de conceptes determinats i, per tant, es poden simplificar. Per altre banda s'han utilitzat teoremes com el de Gauss-Jordan per resoldre la matriu inversa.

El programa de resolució consta de dues parts principals que son la introducció de dades (barres i elements de transmissió) i la creació de l'esquema unifilar.

Per la introducció de dades es va decidir en crear un sistema de taula ja que, amb uns coneixements bàsics, és molt mes visual i pràctic la modificació i introducció de dades.

El programa tan sols resol sistemes en un regim permanent i que almenys disposen de dos elements incògnites alhora de resoldre el sistema iteratiu.

L'esquema unifilar potser es la part amb mes mancances ja que es simplement la creació de un dibuix que representa la xarxa. Tot i això l'esquema unifilar no serveix més que com a referència gràfica del que, en la taula, ja s'explica i no és necessari.

El desenvolupament de un sistema informàtic pot no tenir fi, sempre hi ha conceptes nous a introduir o millores a fer. Per aquest motiu el programa es va aturar en aquest nivell.

Aquest programa, degut al seu nivell de desenvolupament, està destinat a gent que s'estigui introduint al calcul de flux de càrregues o que estigui interessat tan sols en els sistemes en regim estacionari.

4.ANNEXES

4.1. ANNEX 1: EQUACIONS DELS ELEMENTS DE TRANSMISSIÓ: 4.1.1. MODELS DE LÍNIES.

Des de les equacions hiperbòliques, equacions de transmissió, d'una línia obtenim:

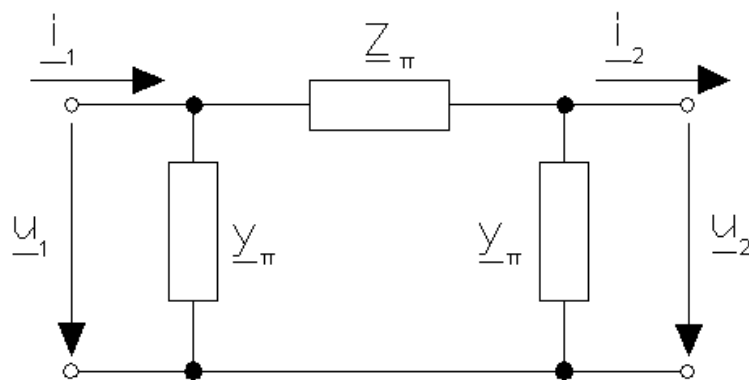
$$\underline{U}_1 = (ch\theta) \cdot \underline{U}_2 + (\underline{Z}_c \cdot sh\theta) \cdot i_2$$

$$i_1 = \left(\frac{sh\theta}{\underline{Z}_c}\right) \cdot u_2 + (ch\theta) \cdot i_2$$

essent:

$$\underline{Z}_c = \sqrt{\frac{\underline{Z}_l}{\underline{Y}_l}} \quad \varrho = \sqrt{\underline{Z}_l \cdot \underline{Y}_l}$$

podem obtenir de forma fàcil un esquema equivalent en “ π ” :



En l'esquema anterior es verifica que:

$$i_2 = \frac{u_1 - u_2}{\underline{Z}_\pi} - \underline{y}_\pi \cdot u_2 \quad u_1 = (1 + \underline{z}_\pi \cdot \underline{y}_\pi) \cdot u_2 + \underline{z}_\pi \cdot i_2$$

Identificant amb la primera equació de transmissió de la línia s'obté:

$$z_{\Pi} = z_c \cdot sh \varrho \quad ; \quad y_{\Pi} = \frac{ch \varrho - 1}{z_c \cdot sh \varrho} \quad y_{\Pi} = \frac{1}{z_c} \cdot tgh \left(\frac{\varrho}{2} \right)$$

O també tenint en compte que $z_t = z_c \cdot \theta$, $y_t = \frac{\theta}{z_c}$, és:

$$z_{\Pi} = z_t \cdot \frac{sh \varrho}{\theta} \quad ; \quad y_{\Pi} = y_t \cdot \frac{tgh \left(\frac{\varrho}{2} \right)}{\frac{\theta}{2}} \quad ; \quad y_{\Pi} = \frac{y_t}{2} \cdot \frac{tgh \left(\frac{\varrho}{2} \right)}{\frac{\theta}{2}}$$

Com que en la majoria dels casos ens trobam amb $|\varrho| \ll 1$ això fa que la serie de funcions hiperbòliques siguin:

$$sh \varrho = \varrho + \frac{\varrho^3}{3!} + \frac{\varrho^5}{5!} + \dots$$

$$ch \varrho = 1 + \frac{\varrho^2}{2!} + \frac{\varrho^4}{4!} + \dots$$

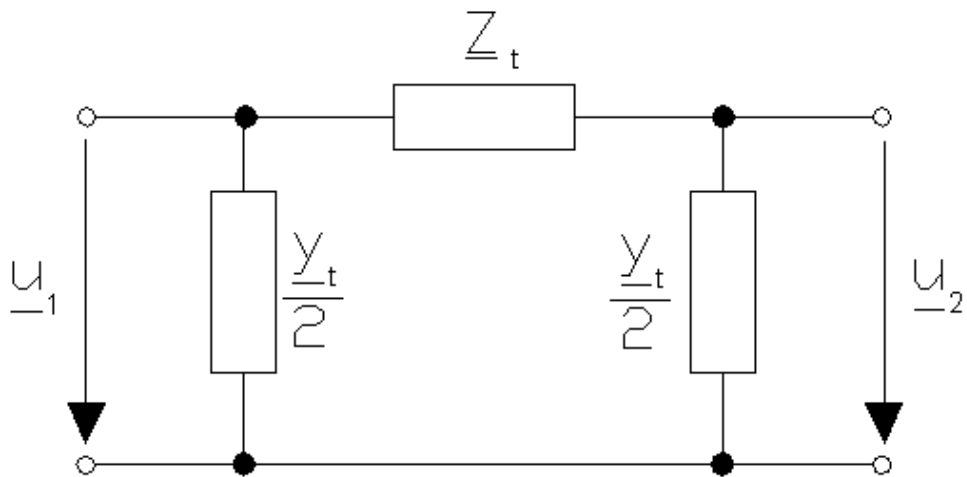
$$tgh \varrho = \varrho + \frac{\varrho^3}{3} + \frac{2}{15} \varrho^5 + \frac{17}{315} \varrho^7 \dots$$

I com a resultat s'obté:

$$z_{\Pi} = z_t \cdot \left[1 + \frac{\varrho^2}{3!} + \frac{\varrho^4}{5!} + \dots \right] \approx z_t$$

$$y_{\Pi} = \frac{y_t}{2} \cdot \left[1 - \frac{\left(\frac{\varrho}{2} \right)^2}{3} + \frac{2}{15} \cdot \left(\frac{\varrho}{2} \right)^4 - \dots \right] \approx \frac{y_t}{2}$$

Així l'esquema en “ π ” és suficient per tenint en compte les consideracions comentades anteriorment. L'esquema serà el següent:



La matriu d'admitàncies de l'esquema anterior vendrà definit per

$$\underline{Y}_{BUS} = \begin{bmatrix} \frac{y_t}{2} + \frac{1}{z_t} & -\frac{1}{z_t} \\ -\frac{1}{z_t} & \frac{y_t}{2} + \frac{1}{z_t} \end{bmatrix}$$

D'aquesta forma, el flux de potència a través de la línia vendrà definit per:

$$\underline{S}_{12}^* = \underline{u}_1^* \cdot \left[\left[\frac{y_t}{2} + \frac{1}{z_t} \right] \underline{u}_1 - \frac{1}{z_t} \underline{u}_2 \right] ; \quad \underline{S}_{12}^* = \underline{u}_1^* \cdot \left[\frac{y_t}{2} \underline{u}_1 - \frac{\underline{u}_1 - \underline{u}_2}{z_t} \right]$$

$$\underline{S}_{21}^* = \underline{u}_2^* \cdot \left[-\frac{1}{z_t} \underline{u}_1 + \left[\frac{y_t}{2} + \frac{1}{z_t} \right] \underline{u}_2 \right] ; \quad \underline{S}_{21}^* = \underline{u}_2^* \cdot \left[\frac{y_t}{2} \underline{u}_2 - \frac{\underline{u}_2 - \underline{u}_1}{z_t} \right]$$

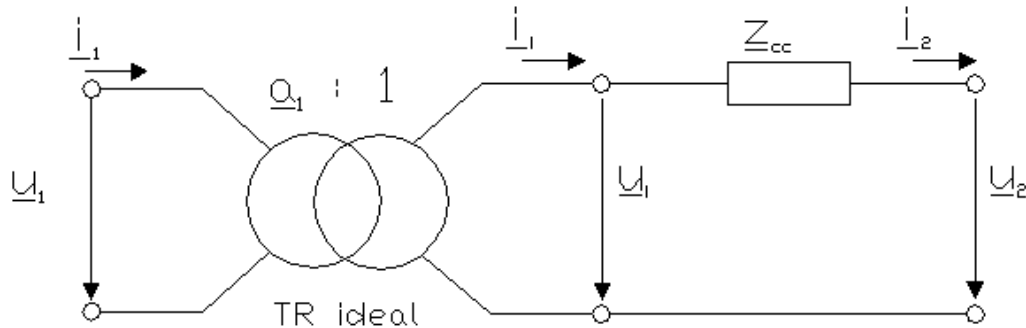
Essent les pèrdues per transmissió en la línia:

$$\Delta \underline{s} = \underline{s}_{12} + \underline{s}_{12} ; \quad \Delta \underline{s} = \left[\frac{y_t}{2} + \frac{1}{z_t} \right] \cdot (\underline{u}_1^2 - \underline{u}_2^2) - \frac{2 \cdot \Re(\underline{u}_1^* \cdot \underline{u}_2)}{z_t}$$

4.1.2. MODELS DE TRANSFORMADORS.

En aquest estudi tendrem en compte tan sols els transformadors de dos enrotllaments i, a més, menysprearem els paràmetres transversals.

D'aquesta forma l'esquema del transformador es pot representar de la següent forma:



En aquest esquema la z_{cc} és la impedància de curtcircuit del transformador referida als seus valor nominals. Si es volgués representar la impedància en el primari del transformador ideal, el seu valor seria $a^2 \cdot z_{cc}$.

En l'esquema anterior es verifica que:

$$\underline{u}_1 = a \cdot \underline{u}_i \quad ; \quad \underline{i}_1 = \frac{\underline{i}_i}{a^*} \quad ; \quad \underline{u}_1 \cdot \underline{i}_1^* = \underline{u}_i \cdot \underline{i}_i^*$$

Aleshores, les equacions de transmissió del transformador són:

$$\underline{u}_1 = a \cdot (\underline{u}_i + z_{cc} \cdot \underline{i}_i) \quad ; \quad \underline{u}_1 = a \cdot \underline{u}_i + a \cdot z_{cc} \cdot \underline{i}_i$$

$$\underline{i}_1 = \frac{\underline{i}_i}{a^*} \quad ; \quad \underline{i}_1 = \underline{i}_i \cdot (a^*)^{-1}$$

Ja que $\det(A) = \frac{a}{a^*} \neq 1$, al transformador li correspon un quadripol no reciproc, per tant, no es pot representar amb un esquema en “ π ” que tan sols contengui elements passius.

Les equacions d'admitàncies del transformador són:

$$\underline{i}_1 = (a^2 \cdot z_{cc})^{-1} \cdot \underline{u}_1 - (a^* \cdot z_{cc})^{-1} \cdot \underline{u}_2$$

$$-\underline{i}_2 = -(a \cdot z_{cc})^{-1} \cdot \underline{u}_1 + (z_{cc})^{-1} \cdot \underline{u}_2$$

Per tant:

$$\underline{Y}_{BUS} = \begin{bmatrix} \frac{\underline{z}_{cc}^{-1}}{a^2} & -\frac{\underline{z}_{cc}^{-1}}{a^*} \\ -\frac{\underline{z}_{cc}^{-1}}{a} & \frac{\underline{z}_{cc}^{-1}}{1} \end{bmatrix}$$

D'aquesta forma, el flux de potències a través del transformador vendrà serà:

$$\underline{S}_{12}^* = \underline{u}_1^* \cdot \left[\frac{\underline{u}_1}{a^2 \cdot \underline{z}_{cc}} - \frac{\underline{u}_2}{a^* \cdot \underline{z}_{cc}} \right] ; \quad \underline{S}_{12}^* = \frac{\underline{u}_1^*}{a^* \cdot \underline{z}_{cc}} \cdot \left[\frac{\underline{u}_1}{a} - \underline{u}_2 \right]$$

$$\underline{S}_{21}^* = \underline{u}_2^* \cdot \left[-\frac{\underline{u}_1}{a \cdot \underline{z}_{cc}} + \frac{\underline{u}_2}{\underline{z}_{cc}} \right] ; \quad \underline{S}_{12}^* = \frac{\underline{u}_2^*}{\underline{z}_{cc}} \cdot \left[\underline{u}_2 - \frac{\underline{u}_1}{a} \right]$$

Essent les pèrdues de transmissió:

$$\Delta \underline{S} = \underline{s}_{12} + \underline{s}_{21} ; \quad \Delta \underline{S} = \frac{1}{\underline{z}_{cc}^*} \cdot \left[\frac{\underline{u}_1}{a} - \underline{u}_2 \right]^2$$

Considerem ara les següents particularitats:

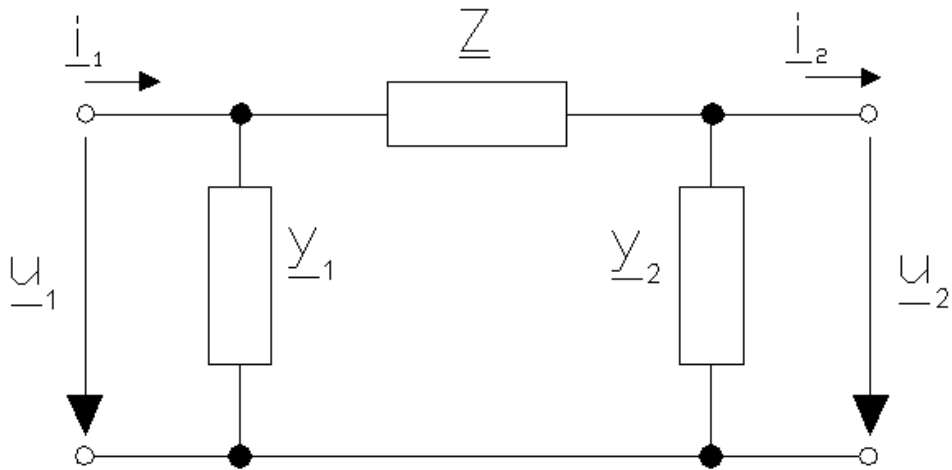
- 1) Suposam que en el transformador no hi ha un decalatge de fase. En aquestes condicions : $\underline{a} = a$.

Així doncs, les equacions de transmissió del transformador són:

$$\underline{u}_1 = a \cdot \underline{u}_2 + a \cdot \underline{z}_{cc} \cdot \underline{i}_2$$

$$\underline{i}_1 = \frac{1}{a} \cdot \underline{i}_2$$

En aquest cas el $\det(A) = 1$, per lo tant el transformador correspon a un quadripol recíproc, encara que no simètric, ja que $\underline{a}_{11} \neq \underline{a}_{22}$. Per tant podem obtenir aquest esquema equivalent en “ π ” .



En aquest esquema es verifica que:

$$i_1 = \frac{u_1 - u_2}{Z} + y_1 \cdot u_1$$

$$i_2 = \frac{u_1 - u_2}{Z} + y_2 \cdot u_2$$

Per tant les equacions d'aquest esquema són:

$$u_1 = (1 + Z \cdot y_2) \cdot u_2 + Z \cdot i_2$$

$$i_1 = (y_1 + y_2 + Z \cdot y_1 \cdot y_2) \cdot u_2 + (1 + Z \cdot y_1) \cdot i_2$$

I d'aquí s'obtenen els paràmetres següents:

$$Z = a \cdot Z_{cc} \quad ; \quad y_1 = \frac{1-a}{a^2} \cdot Z_{cc}^{-1} \quad ; \quad y_2 = \frac{a-1}{a} \cdot Z_{cc}^{-1}$$

Per tant la nova matriu d'admitàncies serà:

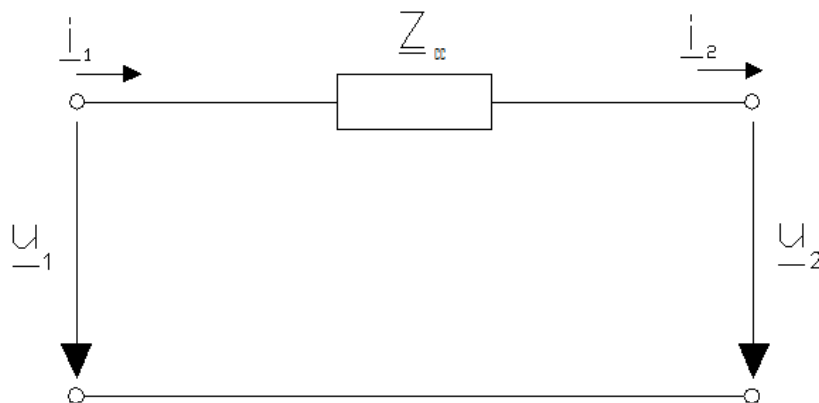
$$\underline{Y}_{BUS} = \begin{bmatrix} \underline{y}_1 + \underline{z}^{-1} & -\underline{z}^{-1} \\ -\underline{z}^{-1} & \underline{y}_2 + \underline{z}^{-1} \end{bmatrix}; \quad \underline{Y}_{BUS} = \begin{bmatrix} \frac{\underline{z}_{cc}^{-1}}{a^2} & -\frac{\underline{z}_{cc}^{-1}}{a} \\ -\frac{\underline{z}_{cc}^{-1}}{a} & \frac{\underline{z}_{cc}^{-1}}{1} \end{bmatrix}$$

- 2) Si el transformador no disposa de relació de transformació variable, i tampoc dona lloc a decalatge de fase, $a=1$ i per tant les equacions de transmissió són:

$$\underline{u}_1 = \underline{u}_2 + \underline{z}_{cc} \cdot \underline{i}_2$$

$$\underline{i}_1 = \underline{i}_2$$

En aquest cas el transformador correspon a un quadripol reciproc i simètric. Essent la seva impedància equivalent la impedància de curtcircuit del transformador



En aquest cas, la matriu d'admitàncies del transformador serà:

$$\underline{Y}_{BUS} = \begin{bmatrix} \underline{z}_{cc}^{-1} & -\underline{z}_{cc}^{-1} \\ -\underline{z}_{cc}^{-1} & \underline{z}_{cc}^{-1} \end{bmatrix}$$

4.2. ANNEX 2: CODI DE PROGRAMACIÓ.

unit W2kMain;

interface

uses Windows, Classes, Graphics, Forms, Controls, Menus,
Dialogs, StdCtrls, Buttons, ExtCtrls, ComCtrls, ImgList, StdActns,
ActnList, ToolWin, DB, DBTables, Provider, DBCtrls, Grids, DBGrids, Mask,shellapi,
RpDefine, RpBase, RpSystem, RpRenderRTF, RpRenderText, RpRender,
RpRenderPDF, RpRave, RpRenderHTML;

type

TWin2kAppForm = class(TForm)

 OpenDialog: TOpenDialog;

 SaveDialog: TSaveDialog;

 ActionList1: TActionList;

 FileNew1: TAction;

 FileOpen1: TAction;

 FileSave1: TAction;

 FileSaveAs1: TAction;

 FileExit1: TAction;

 EditCut1: TEditCut;

 EditCopy1: TEditCopy;

 EditPaste1: TEditPaste;

 HelpAbout1: TAction;

 StatusBar: TStatusBar;

 ImageList1: TImageList;

 RichEdit1: TRichEdit;

 MainMenu1: TMainMenu;

 File1: TMenuItem;

 FileNewItem: TMenuItem;

 FileOpenItem: TMenuItem;

 FileSaveItem: TMenuItem;

 FileSaveAsItem: TMenuItem;

 N2: TMenuItem;

 FileExitItem: TMenuItem;

 Edit1: TMenuItem;

 CutItem: TMenuItem;

 CopyItem: TMenuItem;

 PasteItem: TMenuItem;

 Help1: TMenuItem;

 HelpAboutItem: TMenuItem;

 SaveDialog1: TSaveDialog;

 ToolBar1: TToolBar;

 ToolButton1: TToolButton;

 ToolButton2: TToolButton;

 ToolButton3: TToolButton;

 ToolButton4: TToolButton;

 ToolButton6: TToolButton;

 ToolButton7: TToolButton;

ToolButton8: TToolButton;
ToolButton9: TToolButton;
PopupMenu1: TPopupMenu;
Cut1: TMenuItem;
Copy1: TMenuItem;
Paste1: TMenuItem;
ToolButton5: TToolButton;
PageControl1: TPageControl;
TabSheet1: TTabSheet;
Button1: TButton;
Panel1: TPanel;
Label5: TLabel;
Label6: TLabel;
Label7: TLabel;
Label8: TLabel;
Label9: TLabel;
Label10: TLabel;
Label11: TLabel;
Label12: TLabel;
Panel3: TPanel;
Button2: TButton;
Button3: TButton;
Button4: TButton;
Button7: TButton;
TabSheet2: TTabSheet;
Button8: TButton;
Panel2: TPanel;
Label1: TLabel;
Label2: TLabel;
Panel5: TPanel;
Panel7: TPanel;
Label16: TLabel;
Panel4: TPanel;
Button11: TButton;
Button12: TButton;
Button13: TButton;
Button14: TButton;
TabSheet3: TTabSheet;
Opciones1: TMenuItem;
Panel8: TPanel;
SpeedButton3: TSpeedButton;
Lin: TSpeedButton;
Carg: TSpeedButton;
LinC: TSpeedButton;
Trafo: TSpeedButton;
Buss: TSpeedButton;
Conde: TSpeedButton;
Gene: TSpeedButton;
SpeedButton2: TSpeedButton;
BitBtn3: TBitBtn;

BitBtn2: TBitBtn;
SpeedButton4: TSpeedButton;
BitBtn4: TBitBtn;
BitBtn1: TBitBtn;
BBS: TSpeedButton;
ETR: TSpeedButton;
SpeedButton6: TSpeedButton;
BitBtn6: TBitBtn;
BitBtn7: TBitBtn;
BitBtn8: TBitBtn;
BitBtn9: TBitBtn;
utorial1: TMenuItem;
Opcions2: TMenuItem;
Memo2: TMemo;
Memo3: TMemo;
ToolButton10: TToolButton;
Label3: TLabel;
Label4: TLabel;
Label17: TLabel;
Label18: TLabel;
Label19: TLabel;
Clculiteratiu1: TMenuItem;
SpeedButton1: TSpeedButton;
BitBtn5: TBitBtn;
BitBtn10: TBitBtn;
Tbus: TStringGrid;
Image1: TImage;
Tele: TStringGrid;
EV: TEdit;
EAngle: TEdit;
EP: TEdit;
EQ: TEdit;
Ebus: TEdit;
EtipusB: TComboBox;
Etipuse: TComboBox;
ECA: TEdit;
ECB: TEdit;
Eyt: TEdit;
Bcheck: TCheckBox;
ERL: TEdit;
EXL: TEdit;
EBT: TEdit;
EGT: TEdit;
Soluc: TStringGrid;
Image5: TImage;
Button15: TButton;
RvSystem1: TRvSystem;
Image3: TImage;
ScrollBox1: TScrollBox;
Image4: TImage;

```

Edit4: TEdit;
Edit5: TEdit;
BitBtn11: TBitBtn;
Label20: TLabel;
Label21: TLabel;
SpeedButton5: TSpeedButton;
Panel9: TPanel;
Image2: TImage;
RvRenderHTML1: TRvRenderHTML;
Ereg: TEdit;
Label13: TLabel;
RegCheck: TCheckBox;
RvRenderPDF1: TRvRenderPDF;
RvRenderRTF1: TRvRenderRTF;
RvRenderText1: TRvRenderText;
N1: TMenuItem;
Exportar1: TMenuItem;
procedure FileOpen1Execute(Sender: TObject);
procedure FileSave1Execute(Sender: TObject);
procedure FileSaveAs1Execute(Sender: TObject);
procedure FileExit1Execute(Sender: TObject);
procedure HelpAbout1Execute(Sender: TObject);
procedure FormCreate(Sender: TObject);
procedure ToolButton5Click(Sender: TObject);
procedure PageControl1Change(Sender: TObject);
procedure Button12Click(Sender: TObject);
procedure DataSource2DataChange(Sender: TObject; Field: TField);
procedure BBSClick(Sender: TObject);
procedure utorial1Click(Sender: TObject);
procedure Opciones2Click(Sender: TObject);
procedure Clculiteratiu1Click(Sender: TObject);
procedure ToolButton1Click(Sender: TObject);
procedure FileNewItemClick(Sender: TObject);
procedure FileNew1Execute(Sender: TObject);
procedure Image1Click(Sender: TObject);
procedure Image3Click(Sender: TObject);
procedure BitBtn1Click(Sender: TObject);
procedure BitBtn4Click(Sender: TObject);
procedure SpeedButton3Click(Sender: TObject);
procedure LinClick(Sender: TObject);
procedure LinCClick(Sender: TObject);
procedure TrafoClick(Sender: TObject);
procedure BussClick(Sender: TObject);
procedure GeneClick(Sender: TObject);
procedure CargClick(Sender: TObject);
procedure CondeClick(Sender: TObject);
procedure Image4MouseDown(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
procedure BitBtn2Click(Sender: TObject);
procedure BitBtn3Click(Sender: TObject);

```

```

procedure BitBtn5Click(Sender: TObject);
procedure BitBtn10Click(Sender: TObject);
procedure RvSystem1Print(Sender: TObject);
procedure TeleDrawCell(Sender: TObject; ACol, ARow: Integer;
  Rect: TRect; State: TGridDrawState);
procedure EtipusBChange(Sender: TObject);
procedure Button3Click(Sender: TObject);
procedure TbusDrawCell(Sender: TObject; ACol, ARow: Integer;
  Rect: TRect; State: TGridDrawState);
procedure Button1Click(Sender: TObject);
procedure Button4Click(Sender: TObject);
procedure TbusSelectCell(Sender: TObject; ACol, ARow: Integer;
  var CanSelect: Boolean);
procedure Button2Click(Sender: TObject);
procedure TbusClick(Sender: TObject);
procedure Button7Click(Sender: TObject);
procedure Button8Click(Sender: TObject);
procedure Button11Click(Sender: TObject);
procedure Button13Click(Sender: TObject);
procedure Button14Click(Sender: TObject);
procedure TeleSelectCell(Sender: TObject; ACol, ARow: Integer;
  var CanSelect: Boolean);
procedure EtipuseChange(Sender: TObject);
procedure FormResize(Sender: TObject);
procedure Button15Click(Sender: TObject);
procedure ToolButton10Click(Sender: TObject);
procedure Button5Click(Sender: TObject);
procedure BitBtn11Click(Sender: TObject);
procedure ETRClick(Sender: TObject);
procedure BitBtn6Click(Sender: TObject);
procedure BitBtn9Click(Sender: TObject);
procedure BitBtn8Click(Sender: TObject);
procedure BitBtn7Click(Sender: TObject);
procedure TeleClick(Sender: TObject);
procedure EtipuseSelect(Sender: TObject);
procedure TeleDb1Click(Sender: TObject);
procedure TbusDb1Click(Sender: TObject);
procedure RegCheckClick(Sender: TObject);
private
  FileName:string;

public
  { Public declarations }
end;

var
  Win2kAppForm: TWin2kAppForm;
  Bus,Tran: Array of Array of real;//dades inicials
  Msol,Bij,Gij,dBij,dGij: Array of Array of real;// Part real i imaginaria de Ybuss
  M,Mi: Array of Array of real;//matriu, matriu inversa,Vector de F

```

```

Ref: Array of Array of Integer;
Fpq:Array of real;
ti,Nu,Nua,Nub,tib,pt,pb,exp,pd,Rowe,cole,Row,col,HV,rpo,k,i,j,nb,nt:integer;//tamany de
buss, Marcador,Marcadors H,V, numero de incognites;nombre d'iteraciÃ³
Re,Im, RM,at,erra,erref:real; //component real i imaginari, valor-marca, At=a regulaciÃ³
editar,forza,cuadri,posi,PA,propi:boolean;
pr,tnom:string;

```

```

Const
e=4;

```

```

implementation

```

```

uses

```

```

SysUtils, Mapi, about, unit1, SHFolder;

```

```

{$R *.dfm}

```

```

resourcestring

```

```

SUntitled = 'Sense Títol';

```

```

SOverwrite = 'Vols reemplaçar-ho %s';

```

```

function Valor(A, B: real) : Real;

```

```

Begin

```

```

Result:= A/(sqr(B)+Sqr(A));

```

```

End;

```

```

Procedure dtrafo(ElCanvas: Tcanvas; e,x,y: integer);

```

```

begin

```

```

with elcanvas do

```

```

begin

```

```

moveto(x div e,y div e);

```

```

if HV=3 then

```

```

begin

```

```

lineto(x div e,(y+10 * 5) div e);

```

```

ellipse((x-10 * 5) div e,(y+10 * 5) div e,(x+10 * 5) div e,(y+30 * 5) div e);

```

```

ellipse((x-10 * 5) div e,(y+25 * 5) div e,(x+10 * 5) div e,(y+45 * 5) div e);

```

```

moveTo((x div e),(y+45 * 5) div e);

```

```

lineto((x div e),(y+60 * 5) div e);

```

```

end;

```

```

if HV=0 then

```

```

begin

```

```

lineto((x+10* 5) div e,y div e);

```

```

ellipse((x+10* 5) div e,(y-10* 5) div e,(x+30* 5) div e,(y+10* 5) div e);

```

```

ellipse((x+25* 5) div e,(y-10* 5) div e,(x+45* 5) div e,(y+10* 5) div e);

```

```

moveto((x+45* 5) div e,y div e);

```

```

lineto((x+60* 5) div e,y div e);

```

```

end;

```

```

if HV=1 then

```

```

begin

```

```

lineto(x div e,(y-10 * 5) div e);

```

```

ellipse((x-10 * 5) div e,(y-10 * 5) div e,(x+10 * 5) div e,(y-30 * 5) div e);
ellipse((x-10 * 5) div e,(y-25* 5) div e,(x+10 * 5) div e,(y-45* 5) div e);
moveTo(x div e,(y-45* 5) div e);
lineto(x div e,(y-60 * 5) div e);
end;
if HV=2 then
begin
lineto((x-10 * 5) div e,y div e);
ellipse((x-10 * 5) div e,(y-10 * 5) div e,(x-30 * 5) div e,(y+10 * 5) div e);
ellipse((x-25* 5) div e,(y-10 * 5) div e,(x-45* 5) div e,(y+10 * 5) div e);
moveto((x-45* 5) div e,y div e);
lineto((x-60* 5) div e,y div e);
end;
end;
end;

```

```

procedure titol(Sender: TObject);
begin
with Sender as TBaseReport do
begin
SetFont('Arial', 14);
gotoXY(1,1);
Print('DDS 1.0
gotoXY(1,1.2);
Print(pr);
SetFont('Arial', 12);
gotoXY(1,1.8);
end;
end;

```

Càlcul del flux de càrregues');

```

Procedure dconde(EICanvas: Tcanvas; e,x,y: integer);
begin
with elcanvas do
begin
moveto(x div e,y div e);
Pen.Color:= clBlack;
if HV=3 then
begin
lineto((x) div e,(y+10 * 5) div e);
moveto((x-10 * 5) div e,(y+10 * 5) div e);
lineto((x+10 * 5) div e,(y+10 * 5) div e);
moveto((x-10 * 5) div e,(y+15* 5) div e);
lineto((x+10 * 5) div e,(y+15* 5) div e);
moveto((x) div e,(y+15* 5) div e);
lineto((x) div e,(y+30* 5) div e);
end;
if HV=0 then
begin
lineto((x+10 * 5) div e,(y) div e);

```

```

moveto((x+10 * 5) div e,(y-10 * 5) div e);
lineto((x+10 * 5) div e,(y+10 * 5) div e);
moveto((x+15* 5) div e,(y-10 * 5) div e);
lineto((x+15* 5) div e,(y+10 * 5) div e);
moveto((x+15* 5) div e,(y) div e);
lineto((x+30* 5) div e,(y) div e);
end;
if HV=1 then
begin
lineto((x) div e,(y-10 * 5) div e);
moveto((x-10 * 5) div e,(y-10 * 5) div e);
lineto((x+10 * 5) div e,(y-10 * 5) div e);
moveto((x-10 * 5) div e,(y-15* 5) div e);
lineto((x+10 * 5) div e,(y-15* 5) div e);
moveto((x) div e,(y-15* 5) div e);
lineto((x) div e,(y-30 * 5) div e);
end;
if HV=2 then
begin
lineto((x-10 * 5) div e,(y) div e);
moveto((x-10 * 5) div e,(y-10 * 5) div e);
lineto((x-10 * 5) div e,(y+10 * 5) div e);
moveto((x-15* 5) div e,(y-10 * 5) div e);
lineto((x-15* 5) div e,(y+10 * 5) div e);
moveto((x-15* 5) div e,(y) div e);
lineto((x-30 * 5) div e,(y) div e);
end;
end;
end;

```

```

Procedure Letran(EICanvas: Tcanvas; Ti,Nua,Nub,e,x,y: integer);
begin
x:=x div e;
y:=y div e;
with elcanvas do
begin
if Ti=2 then
begin
Textout(x+3,y+3,'Línia'+ ' ' + inttostr(Nua)+'-' + inttostr(Nub));
end;
if Ti=3 then
begin
Textout(x+3,y+3,'Transformador'+ ' ' + inttostr(Nua)+'-' + inttostr(Nub));
end;
if Ti=4 then
begin
Textout(x+3,y+3,'B.Condensadors'+ ' ' + inttostr(Nua)+'-' + inttostr(Nub));
end;
end;
end;
end;

```

```

Procedure Lebus(EICanvas: Tcanvas; Ti,Nu,e,x,y: integer);
begin
with elcanvas do
begin
x:=x div e;
y:=y div e;
  if Ti=5 then
  begin
Textout(x+3,y+3,'Slack'+ ' '+ inttostr(Nu));
end;
  if Ti=6 then
  begin
Textout(x+3,y+3,'PV'+ ' '+ inttostr(Nu));
end;
  if Ti=7 then
  begin
Textout(x+3,y+3,'PQ'+ ' '+ inttostr(Nu));
end;
  if Ti=8 then
  begin
Textout(x+3,y+3,'PQV'+ ' '+ inttostr(Nu));
end;
end;
end;
end;
Procedure dbus(EICanvas: Tcanvas; e,x,y: integer);
begin
with elcanvas do
begin
moveto(x,y);
Pen.Color:= clBlack;
Brush.Color:= clBlack;
if HV=0 then
begin
rectangle(x div e,y div e,(x+30 * 5) div e,(y+5* 5) div e);
end;
if HV=3 then
begin
rectangle(x div e,y div e,(x+5* 5) div e,(y+30* 5) div e);
end;
if HV=2 then
begin
rectangle(x div e,y div e,(x-30* 5) div e,(y-5* 5) div e);
end;
if HV=1 then
begin
rectangle(x div e,y div e,(x-5* 5) div e,(y-30* 5) div e);
end;
Brush.Color:= clwhite;
end;
end;
end;

```

end;

```
Procedure dgen(EICanvas: Tcanvas; e,x,y: integer);
begin
with elcanvas do
begin
moveto((x) div e,(y) div e);
if (HV=0) then
begin
lineto((x+20 * 5) div e,(y) div e);
moveto((x+20 * 5) div e,(y) div e);
ellipse((x+20 * 5) div e,(y-10 * 5) div e,(x+40 * 5) div e,(y+10 * 5) div e);
end;
if (HV=1) then
begin
lineto((x) div e,(y-20 * 5) div e);
moveto((x) div e,(y-20 * 5) div e);
ellipse((x-10 * 5) div e,(y-20 * 5) div e,(x+10 * 5) div e,(y-40 * 5) div e);
end;
if (HV=2) then
begin
lineto((x-20 * 5) div e,(y) div e);
moveto((x-20 * 5) div e,(y) div e);
ellipse((x-20 * 5) div e,(y+10 * 5) div e,(x-40 * 5) div e,(y-10 * 5) div e);
end;
if (HV=3) then
begin
lineto((x) div e,(y+20 * 5) div e);
moveto((x) div e,(y+20 * 5) div e);
ellipse((x-10 * 5) div e,(y+20 * 5) div e,(x+10 * 5) div e,(y+40 * 5) div e);
end;
end;
end;
end;
```

```
Procedure dcarg(EICanvas: Tcanvas; e,x,y: integer);
begin
with elcanvas do
begin
moveto((x) div e,(y) div e);
if HV=0 then
begin
lineto((x+30 * 5) div e,(y) div e);
lineto((x+25* 5) div e,(y+2* 5) div e);
lineto((x+25* 5) div e,(y-2* 5) div e);
lineto((x+30* 5) div e,(y) div e);
brush.color:=clblack;
floodfill((x+26* 5) div e,(y) div e+1,clblack,fsborder);
floodfill((x+26* 5) div e,(y) div e-1,clblack,fsborder);
brush.color:=clwhite;
end;
end;
```



```

if HV=1 then
begin
lineto((x) div e,(y-30* 5) div e);
lineto((x+2* 5) div e,(y-25* 5) div e);
lineto((x-2* 5) div e,(y-25* 5) div e);
lineto((x) div e,(y-30* 5) div e);
brush.color:=clblack;
floodfill((x) div e+1,(y-26* 5) div e,clblack,fsborder);
floodfill((x) div e-1,(y-26* 5) div e,clblack,fsborder);
brush.color:=clwhite;
end;
if HV=2 then
begin
lineto((x-30* 5) div e,(y) div e);
lineto((x-25* 5) div e,(y+2* 5) div e);
lineto((x-25* 5) div e,(y-2* 5) div e);
lineto((x-30* 5) div e,(y) div e);
brush.color:=clblack;
floodfill((x-26* 5) div e,(y div e)+1,clblack,fsborder);
floodfill((x-26* 5) div e,(y div e)-1,clblack,fsborder);
brush.color:=clwhite;
end;
if HV=3 then
begin
lineto((x) div e,(y+30* 5) div e);
lineto((x+2* 5) div e,(y+25* 5) div e);
lineto((x-2* 5) div e,(y+25* 5) div e);
lineto((x) div e,(y+30* 5) div e);
brush.color:=clblack;
floodfill((x div e)+1,(y+26* 5) div e,clblack,fsborder);
floodfill((x div e)-1,(y+26* 5) div e,clblack,fsborder);
brush.color:=clwhite;
end;
end;
end;

function DefaultSaveLocation: string;
var
P: PChar;
begin
{
returns the location of 'My Documents' if it exists, otherwise it returns
the current directory.
}
P := nil;
try
P := AllocMem(MAX_PATH);
if SHGetFolderPath(0, CSIDL_PERSONAL, 0, 0, P) = S_OK then
Result := P
else

```

```

    Result := GetCurrentDir;
finally
    FreeMem(P);
end;
end;

```

```

procedure TWin2kAppForm.FileNew1Execute(Sender: TObject);
begin
    SaveDialog.InitialDir := DefaultSaveLocation;
    FFileName := SUntitled;
    RichEdit1.Lines.Clear;
    RichEdit1.Modified := False;
end;

```

```

procedure TWin2kAppForm.FormCreate(Sender: TObject);
var
temp:integer;
begin
    HV:=3;
    image4.Canvas.Create;
    image2.canvas.create;
    row:=1;
    rowe:=1;
    col:=1;
    cole:=1;
    Rpo:=1;
    cuadri:=true;
    SetLength(Ref, rpo+1,3 );
    FileNew1.Execute; { sets the default file name and clears the RichEdit Control }
    tbus.Font.Style;
    Tbus.cells[1,0]:= 'Bus';
    Tbus.cells[2,0]:= 'Tipus';
    Tbus.cells[3,0]:= 'V';
    Tbus.cells[4,0]:= 'Angle';
    Tbus.cells[5,0]:= 'P';
    Tbus.cells[6,0]:= 'Q';
    temp:=tbus.width;
    temp:=(temp-20) div 6;
    tbus.DefaultColWidth:=temp;

```

```

    Tele.cells[1,0]:= 'Tipus';
    tele.ColWidths[2]:=80;
    Tele.cells[2,0]:= 'Connexió A';
    tele.ColWidths[3]:=80;
    Tele.cells[3,0]:= 'Connexió B';
    Tele.cells[4,0]:= 'RI';
    Tele.cells[5,0]:= 'XI';
    Tele.cells[6,0]:= 'Gt';
    Tele.cells[7,0]:= 'Bt';
    tele.ColWidths[8]:=110;

```

```

Tele.cells[8,0]:= 'R.Transformació';
Tele.cells[9,0]:= 'Actiu';
end;

```

```

procedure TWin2kAppForm.GeneClick(Sender: TObject);
begin
bitbtn1.Click;
end;

```

```

procedure TWin2kAppForm.FileOpen1Execute(Sender: TObject);
Var
pox,pof,PoC,PoE,PoD,W:integer;Refa:string;cnt:integer;
begin
if OpenFileDialog.Execute then
begin
RichEdit1.Lines.LoadFromFile(OpenDialog.FileName);
FFilename := OpenFileDialog.FileName;
RichEdit1.SetFocus;
RichEdit1.Modified := False;
RichEdit1.ReadOnly := ofReadOnly in OpenFileDialog.Options;
end;

```

```

Poc:=richedit1.Lines.IndexOf('Components');
Pox:=richedit1.Lines.IndexOf('Conexions');
Poe:=richedit1.Lines.IndexOf('Esquema');
PoF:=richedit1.Lines.IndexOf('Fi');
tbus.RowCount:=2;
poc:=poc+2;
while poc<pox do
Begin
Refa:=richedit1.Lines.Strings[poc];
W:=Pos('|',Refa);
if w>1 then
delete(Refa,1,w);
W:=Pos('|',Refa);
tbus.RowCount:=tbus.RowCount+1;
for cnt:=0 to 6 do
begin
if w>1 then
tbus.Cells[cnt,poc-1]:=Copy(Refa, 0, W-1));
delete(Refa,1,w);
W:=Pos('|',Refa);
end;
poc:=poc+1;
End;
tbus.RowCount:=tbus.RowCount-1;
pox:=pox+2;
tele.RowCount:=2;

```

```

while pox<poe do
Begin
tele.RowCount:=tele.RowCount+1;
Refa:=richedit1.Lines.Strings[pox];
W:=Pos('|',Refa);
if w>1 then
delete(Refa,1,w);
W:=Pos('|',Refa);
for cnt:=0 to 10 do
begin
if w>1 then
tele.Cells[cnt,pox-poc-1]:=(Copy(Refa, 0, W-1));
delete(Refa,1,w);
W:=Pos('|',Refa);
end;
pox:=pox+1;
End;
tele.RowCount:=tele.RowCount-1;
image4.canvas.pen.Color:=clwhite;
image4.Canvas.Rectangle(0,0,1500,1500);
image4.canvas.pen.Color:=clblack;
Rpo:=0;
poe:=poe+1;
while poe<pof do
Begin
Rpo:=Rpo+1;
Refa:=richedit1.Lines.Strings[poe];
SetLength(Ref, Rpo, 3);
W:=Pos('|',Refa);
Ref[rpo-1,0]:=strtoint(Copy(Refa, 0, W-1));
delete(Refa,1,W);
W:=Pos('|',Refa);
Ref[rpo-1,1]:=strtoint(Copy(Refa, 0, W-1));
delete(Refa,1,W);
W:=Pos('|',Refa);
Ref[rpo-1,2]:=strtoint(Copy(Refa, 0, W-1));
poe:=poe+1;
if poe=pof then
begin
rpo:=rpo-1;
bitbtn3.Click;
end;
end;

end;

procedure TWin2kAppForm.FileSave1Execute(Sender: TObject);
var
cnt:integer;

```

```

begin
  RichEdit1.Lines.Clear;
  richedit1.Lines.Add('Components');
  for cnt:=0 to tbus.RowCount-1 do
  begin
    richedit1.Lines.add(tbus.Cells[0,cnt]+'|'+tbus.Cells[1,cnt]+'|'+tbus.Cells[2,cnt]
+'|'+tbus.Cells[3,cnt]+'|'+tbus.Cells[4,cnt]+'|'+tbus.Cells[5,cnt]+'|'+tbus.Cells[6,cnt]+'|'); end;
    richedit1.Lines.Add('Conexions');
    for cnt:=0 to tele.RowCount-1 do
    begin
      richedit1.Lines.Add(tele.Cells[0,cnt]+'|'+tele.Cells[1,cnt]+'|'+tele.Cells[2,cnt]
+'|'+tele.Cells[3,cnt]+'|'+tele.Cells[4,cnt]+'|'+tele.Cells[5,cnt]+'|'+tele.Cells[6,cnt]
+'|'+tele.Cells[7,cnt]+'|'+tele.Cells[8,cnt]+'|'+tele.Cells[9,cnt]+'|');
    end;
    richedit1.Lines.Add('Esquema');
    for pd := 0 to rpo do
    richedit1.Lines.add(inttostr(Ref[pd,0])+'|'+inttostr(Ref[pd,1])+'|'+inttostr(Ref[pd,2])+'|');
    richedit1.Lines.Add('fi');

    if (FFileName = SUntitled) or (FFileName = "") then
      FileSaveAs1Execute(Sender)
    else
      begin
        RichEdit1.Lines.SaveToFile(FFileName);
        RichEdit1.Modified := False;
      end;
    end;
end;

procedure TWin2kAppForm.FileSaveAs1Execute(Sender: TObject);
var
cnt:integer;
begin
  RichEdit1.Lines.Clear;
  richedit1.Lines.Add('Components');
  RichEdit1.Lines.Clear;
  richedit1.Lines.Add('Components');
  for cnt:=0 to tbus.RowCount-1 do
  begin
    richedit1.Lines.add(tbus.Cells[0,cnt]+'|'+tbus.Cells[1,cnt]+'|'+tbus.Cells[2,cnt]
+'|'+tbus.Cells[3,cnt]+'|'+tbus.Cells[4,cnt]+'|'+tbus.Cells[5,cnt]+'|'+tbus.Cells[6,cnt]+'|');
    end;
    richedit1.Lines.Add('Conexions');
    for cnt:=0 to tele.RowCount-1 do
    begin
      richedit1.Lines.Add(tele.Cells[0,cnt]+'|'+tele.Cells[1,cnt]+'|'+tele.Cells[2,cnt]
+'|'+tele.Cells[3,cnt]+'|'+tele.Cells[4,cnt]+'|'+tele.Cells[5,cnt]+'|'+tele.Cells[6,cnt]
+'|'+tele.Cells[7,cnt]+'|'+tele.Cells[8,cnt]+'|'+tele.Cells[9,cnt]+'|');
    end;
    richedit1.Lines.Add('Esquema');
    for pd := 0 to rpo do

```

```
richedit1.Lines.add(inttostr(Ref[pd,0])+'|'+inttostr(Ref[pd,1])+'|'+inttostr(Ref[pd,2])+'|');
richedit1.Lines.Add('fi');
```

```
with SaveDialog do
begin
  FileName := FFileName;
  if Execute then
  begin
    if FileExists(FileName) then
      if MessageDlg(Format(SOverwrite, [FileName]),
        mtConfirmation, mbYesNoCancel, 0) <> idYes then Exit;
    RichEdit1.Lines.SaveToFile(FileName);
    FFileName := FileName;
    RichEdit1.Modified := False;
  end;
end;
end;
```

```
procedure TWin2kAppForm.BBSClick(Sender: TObject);
var
x,y:integer;temp:string;
begin
image2.Canvas.rectangle(0,0,70,70);
x:=0;
y:=0;
Nu:=strtoint(tbus.Cells[1,pb+1]);

  if (tbus.Cells[2,pb+1]='Slack') or (tbus.Cells[2,row]='sl') or (tbus.Cells[2,row]='SL') then
  tib:=5;
  if (tbus.Cells[2,pb+1]='PV') or (tbus.Cells[2,row]='pv') then
  tib:=6;
  if (tbus.Cells[2,pb+1]='PQ') or (tbus.Cells[2,row]='pq') then
  tib:=7;
  if (tbus.Cells[2,pb+1]='PQV') or (tbus.Cells[2,row]='pqv') then
  tib:=8;

lebus(image2.canvas,tib,Nu,e,x,y);

end;
```

```
procedure TWin2kAppForm.BitBtn10Click(Sender: TObject);
var
pd,x,y,refh,refv,rt:integer;
begin
if cuadri then
begin
image4.Canvas.Pen.color:=clwhite;
image4.Canvas.Rectangle(0,0,1500,1500);
```

```

image4.Canvas.Pen.color:=clblack;
refH:=image4.Width;
refV:=image4.Height;
with image4.Canvas do
begin
pen.Color:=clMenuBar;
rt:=0;
while rt<refh do
begin
  moveto(rt,0);
  lineto(rt,refv);
  rt:=rt+10;
end;
rt:=0;
while rt<refv do
begin
  moveto(0,rt);
  lineto(refh,rt);
  rt:=rt+10;
end;
pen.Color:=clblack;
cuadri:=false;
end;
end
else
begin
image4.Canvas.Pen.color:=clwhite;
image4.Canvas.Rectangle(0,0,1500,1500);
image4.Canvas.Pen.color:=clblack;
cuadri:=true;
end;
for pd := 0 to rpo do
begin
x:=Ref[pd,0];
y:=Ref[pd,1];
HV:=(ref[pd,2]mod 10) div 10;
with image4.canvas do
begin
if (ref[pd,2]mod 10)=0 then
moveto(x,y);
if (ref[pd,2]mod 10)=1 then
lineto(x,y);
end;
if (ref[pd,2]mod 10)=2 then
dtrafo(image4.Canvas,e,x,y);
if (ref[pd,2]mod 10)=3 then
dbus(image4.Canvas,e,x,y);
if (ref[pd,2]mod 10)=4 then
dgen(image4.Canvas,e,x,y);
if (ref[pd,2]mod 10)=5 then

```

```

dcarg(image4.Canvas,e,x,y);
if (ref[pd,2]mod 10)=6 then
dconde(image4.Canvas,e,x,y);
end;
end;

```

```

procedure TWin2kAppForm.BitBtn1Click(Sender: TObject);

```

```

var
x,y:integer;
begin
if HV=3 then
HV:=0
else
HV:=HV+1;
image2.Canvas.rectangle(0,0,70,70);
if linc.down then
begin
with image2.canvas do
begin
moveto(10,30);
lineto(30,50);
lineto(50,50);
lineto(50,20);
end;
end;

```

```

if lin.down then
begin
with image2.canvas do
begin
moveto(10,30);
lineto(30,60);
moveto(50,50);
lineto(50,20);
end;
end;

```

```

if carg.down then
begin
X:=35*e;
y:=35*e;
dcarg(image2.Canvas,e,x,y);
end;

```

```

if trafo.down then
begin
if HV=0 then
begin
X:=10*e;
y:=35*e;

```



```

end;
if HV=1 then
begin
  X:=35*e;
  y:=60*e;
end;
if HV=2 then
begin
  X:=60*e;
  y:=35*e;
end;
if HV=3 then
begin
  X:=35*e;
  y:=10*e;
end;
dtrafo(image2.Canvas,e,x,y);
end;

if conde.down then
begin
x:=35*e;
y:=35*e;
dconde(image2.Canvas,e,x,y);
end;

if buss.down then
begin
x:=35*e;
y:=35*e;

dbus(image2.Canvas,e,x,y);
end;

if Gene.down then
begin
if HV=0 then
begin
  X:=10*e;
  y:=35*e;
end;
if HV=1 then
begin
  X:=35*e;
  y:=60*e;
end;
if HV=2 then
begin
  X:=60*e;
  y:=35*e;

```

```

end;
if HV=3 then
begin
  X:=35*e;
  y:=10*e;
end;
dgen(image2.Canvas,e,x,y);
end;

with image2.canvas do
begin
  pen.Color:=clred;
  brush.Color:=clred;
  ellipse(x div e-3,y div e-3,x div e+3,y div e+3);
  pen.color:=clblack;
  brush.Color:=clwhite;
end;
end;

procedure TWin2kAppForm.BitBtn2Click(Sender: TObject);
var
  refH,refv,rt,pd,x,y:integer;
begin
  if not cuadri then
  begin
    image4.Canvas.Pen.color:=clwhite;
    image4.Canvas.Rectangle(0,0,image4.Width,image4.Height);
    image4.Canvas.Pen.color:=clblack;
    refH:=image4.Width;
    refV:=image4.Height;
    with image4.Canvas do
    begin
      pen.Color:=clMenuBar;
      rt:=0;
      while rt<refh do
      begin
        moveto(rt,0);
        lineto(rt,refv);
        rt:=rt+10;
      end;
      rt:=0;
      while rt<refv do
      begin
        moveto(0,rt);
        lineto(refh,rt);
        rt:=rt+10;
      end;
    end;
    pen.Color:=clblack;
  end;
end
end

```

```

else
begin
image4.Canvas.Pen.color:=clwhite;
image4.Canvas.Rectangle(0,0,image4.Width,image4.Height);
image4.Canvas.Pen.color:=clblack;
end;

```

```

rpo:=rpo-1;
if rpo>0 then
for pd := 0 to rpo do
begin
x:=Ref[pd,0];
y:=Ref[pd,1];
HV:=(ref[pd,2]) div 10;
with image4.canvas do
begin
if (ref[pd,2]mod 10)=0 then
moveto(x div e,y div e);
if (ref[pd,2]mod 10)=1 then
lineto(x div e,y div e);
end;
if (ref[pd,2]mod 10)=2 then
dtrafo(image4.Canvas,e,x,y);
if (ref[pd,2]mod 10)=3 then
dbus(image4.Canvas,e,x,y);
if (ref[pd,2]mod 10)=4 then
dgen(image4.Canvas,e,x,y);
if (ref[pd,2]mod 10)=5 then
dcarg(image4.Canvas,e,x,y);
if (ref[pd,2]mod 10)=6 then
dconde(image4.Canvas,e,x,y);
if (ref[pd,2]mod 10)=7 then
begin
tib:=ref[pd,2]div 10000;
Nu:=(ref[pd,2]mod 10000)div 100;
lebus(image4.canvas,tib,Nu,e,x,y);
end;
if (ref[pd,2]mod 10)=8 then
begin
ti:=ref[pd,2]div 1000000;
Nua:=(ref[pd,2]mod 1000000)div 10000;
Nub:=(ref[pd,2]mod 10000)div 100;
letran(image4.canvas,ti,Nua,Nub,e,x,y);
end;
end;
end;
end;

```

```

procedure TWin2kAppForm.BitBtn3Click(Sender: TObject);

```

```

var
refv,refh,rt,pd,x,y:integer;
begin
if not cuadri then
begin
image4.Canvas.Pen.color:=clwhite;
image4.Canvas.Rectangle(0,0,image4.Width,image4.Height);
image4.Canvas.Pen.color:=clblack;
refH:=image4.Width;
refV:=image4.Height;
with image4.Canvas do
begin
pen.Color:=clMenuBar;
rt:=0;
while rt<refh do
begin
moveto(rt,0);
lineto(rt,refv);
rt:=rt+10;
end;
rt:=0;
while rt<refv do
begin
moveto(0,rt);
lineto(refh,rt);
rt:=rt+10;
end;
pen.Color:=clblack;
end;
end
else
begin
image4.Canvas.Pen.color:=clwhite;
image4.Canvas.Rectangle(0,0,image4.Width,image4.Height);
image4.Canvas.Pen.color:=clblack;
end;

if rpo<length(Ref)-1 then
rpo:=rpo+1;
for pd := 0 to rpo do
begin
x:=Ref[pd,0];
y:=Ref[pd,1];
HV:=(ref[pd,2]) div 10;
with image4.canvas do
begin
if (ref[pd,2]mod 10)=0 then
moveto(x div e,y div e);
if (ref[pd,2]mod 10)=1 then
lineto(x div e,y div e);

```

```

end;
if (ref[pd,2] mod 10)=2 then
dtrafo(image4.Canvas,e,x,y);
if (ref[pd,2] mod 10)=6 then
dconde(image4.Canvas,e,x,y);
if (ref[pd,2] mod 10)=3 then
dbus(image4.Canvas,e,x,y);
if (ref[pd,2] mod 10)=4 then
dgen(image4.Canvas,e,x,y);
if (ref[pd,2] mod 10)=5 then
dcarg(image4.Canvas,e,x,y);
if (ref[pd,2] mod 10)=7 then
begin
tib:=ref[pd,2] div 10000;
Nu:=(ref[pd,2] mod 10000) div 100;
lebus(image4.canvas,tib,Nu,e,x,y);
end;
if (ref[pd,2] mod 10)=8 then
begin
ti:=ref[pd,2] div 1000000;
Nua:=(ref[pd,2] mod 1000000) div 10000;
Nub:=(ref[pd,2] mod 10000) div 100;
letran(image4.canvas,ti,Nua,Nub,e,x,y);
end;

end;
end;

procedure TWin2kAppForm.BitBtn4Click(Sender: TObject);
begin
if panel9.Visible then
panel9.Visible:=false
else
panel9.Visible:=true;
end;

procedure TWin2kAppForm.BitBtn5Click(Sender: TObject);
begin
if forza then
forza:=false
else
forza:=true;
end;

procedure TWin2kAppForm.BussClick(Sender: TObject);
begin
bitbtn1.Click;
end;

procedure TWin2kAppForm.Button12Click(Sender: TObject);

```

```

begin
// if editar then
// begin
tele.Options:= tele.Options - [goRowSelect];
Tele.cells[1,rowe]:=Etipuse.Text;
Tele.cells[2,rowe]:=ECA.Text;
Tele.cells[3,rowe]:=ECB.Text;
If Etipuse.text='B.Condensadors' then
begin
Tele.cells[7,rowe]:=Eyt.Text;
end
else
begin
Tele.cells[4,rowe]:=ERL.Text;
Tele.cells[5,rowe]:=EXL.Text;
Tele.cells[6,rowe]:=EGT.Text;
Tele.cells[7,rowe]:=EBT.Text;
Tele.cells[8,rowe]:=EREG.Text;
end;
If (RegCheck.Checked or bcheck.checked) then
Tele.cells[9,rowe]:='N'
else
Tele.cells[9,rowe]:='S';

editar:=false;
//end;
end;

procedure TWin2kAppForm.CargClick(Sender: TObject);
begin
bitbtn1.Click;
end;

procedure TWin2kAppForm.Clculiteratiu1Click(Sender: TObject);
begin
toolbutton5.Click;
end;

procedure TWin2kAppForm.CondeClick(Sender: TObject);
begin
bitbtn1.Click;
end;

procedure TWin2kAppForm.DataSource2DataChange(Sender: TObject; Field: TField);
begin
memo2.readonly:=false;
if (Etipuse.text='Linia') then
begin
Panel5.show;

```

```

Panel7.Hide;
memo2.Clear;
memo2.lines.Add("");
memo2.lines.Add('Recorda que al ser un element simètric és indiferent com es
conecti la línia');
end;
if (Etipuse.text='B.Condensadors')then
begin
Panel5.Hide;
Panel7.show;
memo2.Clear;
memo2.lines.Add("");
memo2.lines.Add('La bateria de condensadors tan sols esta connectada en una barra,
per tant només hi haurà una connexió vèlida que és la primera.');
```

```

end;
if (Etipuse.text='Trafo')then
begin
Panel5.show;
Panel7.Hide;
memo2.Clear;
memo2.lines.Add("");
memo2.lines.Add('Recorda que la primera connexió es la part de alta tensió.
Connecta-ho correctament ja que no es un element simètric.');
```

```

end;
memo2.readonly:=true;
end;

procedure TWin2kAppForm.FileExit1Execute(Sender: TObject);
begin
Close;
end;

procedure TWin2kAppForm.FileNewItemClick(Sender: TObject);
begin
SaveDialog.InitialDir := DefaultSaveLocation;
FFilename := SUntitled;
RichEdit1.Lines.Clear;
RichEdit1.Modified := False;
end;

```

```

procedure TWin2kAppForm.HelpAbout1Execute(Sender: TObject);
begin
AboutBox.ShowModal;
end;

```

```

procedure TWin2kAppForm.Image1Click(Sender: TObject);
begin
image1.Canvas.Pen.Width:=10;

```

```

image1.Canvas.LineTo(100,300);
end;

procedure TWin2kAppForm.Image3Click(Sender: TObject);
begin
image1.Canvas.LineTo(200,500);
end;

procedure TWin2kAppForm.Image4MouseDown(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
begin
if forza then
begin
if (x mod 10)<>0 then
  begin
    if (x mod 10)<5 then
      x:=x-(x mod 10)
    else
      x:=x+(10-x mod 10);
    end;

if (y mod 10)<>0 then
  begin
    if (Y mod 10)<5 then
      y:=y-(y mod 10)
    else
      y:=y+(10-y mod 10);
    end;

end;
x:=x*e;
Y:=y*e;

if bbs.Down then
begin
lebus(image4.canvas,tib,Nu,e,x,y);
rpo:=rpo+1;
SetLength(Ref, rpo+1,3 );
Ref[rpo,2]:=7+(((tib*100)+Nu)*100;

end;

if ETR.Down then
begin
letran(image4.canvas,ti,Nua,Nub,e,x,y);
rpo:=rpo+1;
SetLength(Ref, rpo+1,3 );
Ref[rpo,2]:=8+((((tib*100)+Nua)*100)+Nub)*100;
end;

```



```

with image4.canvas do
begin
if posi then
begin
if lin.down then
begin
lineto(x div e,y div e);
rpo:=rpo+1;
SetLength(Ref, rpo+1,3 );
Ref[rpo,2]:=1;
posi:=false;
end;
if linc.Down then
begin
lineto(x div e,y div e);
rpo:=rpo+1;
SetLength(Ref, rpo+1,3 );
Ref[rpo,2]:=1;
end;
end
else
begin
if lin.down then
begin
moveto(x div e,y div e);
rpo:=rpo+1;
SetLength(Ref, rpo+1,3 );
Ref[rpo,2]:=0;
posi:=true;
end;
if linc.Down then
begin
rpo:=rpo+1;
SetLength(Ref, rpo+1,3 );
Ref[rpo,2]:=0;
moveto(x div e,y div e);
posi:=true;
end;
end
end;

if trafo.down then
begin
rpo:=rpo+1;
SetLength(Ref, rpo+1,3 );
Ref[rpo,2]:=HV*10+2;
dtrafo(image4.Canvas,e,x,y);
end;
if buss.down then

```

```

begin
  rpo:=rpo+1;
  SetLength(Ref, rpo+1,3 );
  Ref[rpo,2]:=HV*10+3;
  dbus(image4.Canvas,e,x,y);
end;
if Gene.down then
begin
  rpo:=rpo+1;
  SetLength(Ref, rpo+1,3 );
  Ref[rpo,2]:=HV*10+4;
  dgen(image4.Canvas,e,x,y);
end;
if Carg.down then
begin
  rpo:=rpo+1;
  SetLength(Ref, rpo+1,3 );
  Ref[rpo,2]:=HV*10+5;
  dcarg(image4.Canvas,e,x,y);
end;
if conde.down then
begin
  rpo:=rpo+1;
  SetLength(Ref, rpo+1,3 );
  Ref[rpo,2]:=HV*10+6;
  dconde(image4.Canvas,e,x,y);
end;
Ref[rpo,0]:=x;
Ref[rpo,1]:=Y;
end;

procedure TWin2kAppForm.LinCClick(Sender: TObject);
begin
  posi:=false;
  bitbtn1.Click;
end;

procedure TWin2kAppForm.LinClick(Sender: TObject);
begin
  posi:=false;
  bitbtn1.Click;
end;

procedure TWin2kAppForm.Opciones2Click(Sender: TObject);
begin
  form1.Show;
end;

procedure TWin2kAppForm.PageControl1Change(Sender: TObject);
var

```

```

ns,po,ret:integer;
begin
if pagecontrol1.ActivePageIndex=2 then
begin
panel9.Visible:=true;
soluc.Visible:=false;
end
else
begin
panel9.Visible:=false;
if soluc.colcount>2 then
soluc.Visible:=true;
end;

```

```

if pagecontrol1.ActivePageIndex=2 then
begin
if bus=nil then
begin
end;
end;
end;
procedure TWin2kAppForm.SpeedButton3Click(Sender: TObject);
begin
bitbtn1.Click;
end;
procedure TWin2kAppForm.ToolButton1Click(Sender: TObject);
begin
  SaveDialog.InitialDir := DefaultSaveLocation;
  FFileName := SUntitled;
  RichEdit1.Lines.Clear;
  RichEdit1.Modified := False;
end;
procedure TWin2kAppForm.ToolButton5Click(Sender: TObject);
var
RI,XI,Gt,Bt,vi,ai,vj,aj,drq,drp:real;j,i,ns,dr,dj,di,po:integer; conv:string;
begin
soluc.CleanupInstance;
soluc.Visible:=true;
soluc.RowCount:=2;
soluc.ColCount:=1;
bus:=nil;
msol:=nil;
tran:=nil;
Ns:=0;
K:=0;
//Pasam de Bdd a Array
soluc.ColWidths[0]:=30;
soluc.Cells[0,0]='k';

```

```

Nb:=tbus.RowCount-1;
SetLength(bus, Nb, 5);
row:=1;
while row< Nb+1 do
begin
if tbus.Cells[3,row]<>" then
bus[row-1,1]:=strtofloat(tbus.Cells[3,row]);
if tbus.Cells[4,row]<>" then
bus[row-1,2]:=strtofloat(tbus.Cells[4,row]);
if tbus.Cells[5,row]<>" then
bus[row-1,3]:=strtofloat(tbus.Cells[5,row]);
if tbus.Cells[6,row]<>" then
bus[row-1,4]:=strtofloat(tbus.Cells[6,row]);

if (tbus.Cells[2,row] ='Slack') or (tbus.Cells[2,row] ='sl') or (tbus.Cells[2,row] ='SL') then
bus[row-1,0]:=1;

if (tbus.Cells[2,row] ='PV') or (tbus.Cells[2,row] ='pv') then
begin
bus[row-1,0]:=2;
bus[row-1,2]:=0;
end;
if (tbus.Cells[2,row] ='PQ') or (tbus.Cells[2,row] ='pq') then
begin
bus[row-1,0]:=3;
bus[row-1,2]:=0;
bus[row-1,1]:=1;
end;
if (tbus.Cells[2,row] ='PQV') or (tbus.Cells[2,row] ='pqv') then
begin
bus[row-1,0]:=4;
bus[row-1,2]:=0;
end;
row:=row+1;
end;

for po := 0 to nb - 1 do
begin
if bus[po,0]<>1 then
Begin
soluc.ColCount:=soluc.ColCount+1;
soluc.Cells[soluc.colcount-1,0]:='a'+inttostr(po+1);
soluc.Cells[soluc.colcount-1,1]:='0';
ns:=ns+1;
SetLength(Msol, ns, k+1);
Msol[ns-1,0]:=0;
End;
end;
for po := 0 to nb - 1 do
begin

```

```

if bus[po,0]=3 then
  begin
  soluc.ColCount:=soluc.ColCount+1;
  soluc.Cells[soluc.colcount-1,0]='V'+inttostr(po+1);
  soluc.Cells[soluc.colcount-1,1]='1';
  ns:=ns+1;
  SetLength(Msol, ns, k+1);
  Msol[ns-1,0]:=1;
  end;
end;

```

```

nt:=0;
rowe:=1;
//ns:=tele.RowCount;
while rowe< tele.RowCount do

```

```

  begin
  nt:=nt+1;
  SetLength(tran, nt,8);
  if tele.Cells[2,rowe]<>" then
  tran[nt-1,1]:=strtofloat(tele.Cells[2,rowe]);
  if tele.Cells[3,rowe]<>" then
  tran[nt-1,2]:=strtofloat(tele.Cells[3,rowe]);
  if tele.Cells[4,rowe]<>" then
  tran[nt-1,3]:=strtofloat(tele.Cells[4,rowe]);
  if tele.Cells[5,rowe]<>" then
  tran[nt-1,4]:=strtofloat(tele.Cells[5,rowe]);
  if tele.Cells[6,rowe]<>" then
  tran[nt-1,5]:=strtofloat(tele.Cells[6,rowe]);
  if tele.Cells[7,rowe]<>" then
  tran[nt-1,6]:=strtofloat(tele.Cells[7,rowe]);
  if tele.Cells[8,rowe]<>" then
  tran[nt-1,7]:=strtofloat(tele.Cells[8,rowe]);

```

```

  if (tele.Cells[1,rowe]='Linia') or (tele.Cells[1,rowe]='Lin') or (tele.Cells[1,rowe]='LIN') or
  (tele.Cells[1,rowe]='lin') then
  tran[nt-1,0]:=1;

```

```

  if (tele.Cells[1,rowe]='Trafo') or (tele.Cells[1,rowe]='Tr') or (tele.Cells[1,rowe]='TR') or
  (tele.Cells[1,rowe]='tr') then

```

```

    if tele.Cells[9,rowe] = 'S'then
      begin
        tran[nt-1,0]:=3;
        tran[nt-1,7]:=1;
        ns:=ns+1;
        SetLength(Msol, ns, k+1);
        Msol[ns-1,0]:=1;
        soluc.ColCount:=soluc.ColCount+1;
        soluc.Cells[soluc.colcount-1,0]='R'+inttostr(nt);
        soluc.Cells[soluc.colcount-1,1]='1';

```

```

        end
    else
        tran[nt-1,0]:=2;

        if tele.Cells[1,rowe]='B.Cond' then
        if tele.Cells[9,rowe] = 'N' then
            tran[nt-1,0]:=0
        else
            tran[nt-1,0]:=4;

    rowe:=rowe+1;
end;
soluc.ColCount:=soluc.ColCount+1;
soluc.Cells[soluc.colcount-1,0]='Error';
soluc.Cells[soluc.colcount-1,1]='0';

erref:=unit1.erref;
erra:=1;

while erra>erref do
begin
//definico de tamanys de les arrays

Fpq:=nil;
M:=nil;
Mi:=nil;
Gij:=nil;
Bij:=nil;
    SetLength(Fpq, ns);//Funcions Fp i Fq
    SetLength(M, ns, ns);//Matriu(inicial)
    SetLength(mi, ns, ns);//Matriu inversa
    SetLength(Gij, Nb, Nb);//Ybus real
    SetLength(Bij, Nb, Nb);//Ybus imaginari

//fi definico de tamanys de les arrays

//definim Ybus

for po := 0 to nt- 1 do
begin
    conv:=floattostr(tran[po,1]);
    i:=strtoint(conv)-1;
    conv:=floattostr(tran[po,2]);
    j:=strtoint(conv)-1;
    Rl:=tran[po,3];
    Xl:=tran[po,4];
    Gt:=tran[po,5];

```

```

Bt:=tran[po,6];
at:=tran[po,7];

//linea
if tran[po,0]=1 then
  begin
    if Gt<>0 then
      begin
        Gij[i, i]:=Gij[i, i]+Valor(RI,XI)+1/2*Gt;
        Gij[i, j]:=Gij[i, j]-Valor(RI,XI);
        Gij[j, i]:=Gij[j, i]-Valor(RI,XI);
        Gij[j, j]:=Gij[j, j]+Valor(RI,XI)+1/2*Gt;
      end
    else
      begin
        Gij[i, i]:=Gij[i, i]+Valor(RI,XI);
        Gij[i, j]:=Gij[i, j]-Valor(RI,XI);
        Gij[j, i]:=Gij[j, i]-Valor(RI,XI);
        Gij[j, j]:=Gij[j, j]+Valor(RI,XI);
      end;

    if Bt<>0 then
      begin
        Bij[i, i]:=Bij[i, i]-Valor(XI,RI)+1/2*Bt;
        Bij[i, j]:=Bij[i, j]+Valor(XI,RI);
        Bij[j, i]:=Bij[j, i]+Valor(XI,RI);
        Bij[j, j]:=Bij[j, j]-Valor(XI,RI)+1/2*Bt;
      end
    else
      begin
        Bij[i, i]:=Bij[i, i]-Valor(XI,RI);
        Bij[i, j]:=Bij[i, j]+Valor(XI,RI);
        Bij[j, i]:=Bij[j, i]+Valor(XI,RI);
        Bij[j, j]:=Bij[j, j]-Valor(XI,RI);
      end;
    end;
  //fi linea

  //trafo variable
  if(tran[po,0]=2) or (tran[po,0]=3) then
    begin
      Gij[i, i]:=Gij[i, i]+(1/(at*at)*valor(RI,XI));
      Gij[i, j]:=Gij[i, j]-(1/(at)*valor(RI,XI));
      Gij[j, i]:=Gij[j, i]-(1/(at)*valor(RI,XI));
      Gij[j, j]:=Gij[j, j]+Valor(RI,XI);

      Bij[i, i]:=Bij[i, i]-(1/(at*at)*valor(XI,RI));
      Bij[i, j]:=Bij[i, j]+(1/(at)*valor(XI,RI));
      Bij[j, i]:=Bij[j, i]+(1/(at)*valor(XI,RI));
      Bij[j, j]:=Bij[j, j]-Valor(XI,RI);
    end;
  end;

```

```

end;
//fi trafo

//B.Condensadors
if (tran[po,0]=4) then
Bij[i, i]:=bij[i,i]+Bt;
//fi B.Condensadors

end;
//fi definicio Ybus

//funcions Fpq
po:=0;
for l := 0 to Nb - 1 do
begin
if (bus[i,0]<>1) and (po<ns) then
begin
vi:=bus[i,1];
ai:=bus[i,2];
fpq[po]:=bus[i,3];
for J := 0 to Nb - 1 do
begin
vj:=bus[j,1];
aj:=bus[j,2];
fpq[po]:=fpq[po]-vi*vj*(gij[i,j]*cos(aj-ai)-bij[i,j]*sin(aj-ai));
end;
po:=po+1;
end;
end;
for l := 0 to Nb - 1 do
begin
if ((bus[i,0]=3) or (bus[i,0]=4)) and (po<ns) then
begin
vi:=bus[i,1];
ai:=bus[i,2];
fpq[po]:=bus[i,4];
for J := 0 to Nb - 1 do
begin
vj:=bus[j,1];
aj:=bus[j,2];
fpq[po]:=fpq[po]+vi*vj*(gij[i,j]*sin(aj-ai)+bij[i,j]*cos(aj-ai));
end;
po:=po+1;
end;
end;
end;

// fi Fpq

//CÃ lcul de la matriu

```



```

dr:=0;
po:=0;
// on de mir[1,dr]= 1(angle)/ 2 (V,tensio)/3 (RegulaciÃ³)
for po := 0 to nb - 1 do
begin
if ((bus[po,0]=2) or (bus[po,0]=3) or (bus[po,0]=4)) and (dr<ns) then
begin
m[0,dr]:=po+1;
m[1,dr]:=1;
dr:=dr+1;
end;
end;
for po := 0 to nb - 1 do
begin
if (bus[po,0]=3) and (dr<ns) then
begin
m[0,dr]:=po+1;
m[1,dr]:=2;
dr:=dr+1;
end;
end;

for po := 0 to nt - 1 do
begin
if (tran[po,0]=3) and (dr<ns) then
begin
m[0,dr]:=po+1;
m[1,dr]:=4;
dr:=dr+1
end;
end;
end;

dr:=0;
while dr<ns do
begin
po:=0;
drp:=m[0,dr]-1;
M[0,dr]:=0;
drq:=m[1,dr];
M[1,dr]:=0;
if drq=4 then
begin
dGij:=nil;
dBij:=nil;
SetLength(dGij, Nb, Nb);//real
SetLength(dBij, Nb, Nb);//imaginari
RI:=tran[strtoint(floattostr(drp)),3];
XI:=tran[strtoint(floattostr(drp)),4];
at:=tran[strtoint(floattostr(drp)),7];
conv:=floattostr(tran[strtoint(floattostr(drp)),1]);

```

```

        di:=strtoint(conv)-1;
        conv:=floattostr(tran[strtoint(floattostr(drp)),2]);
        dj:=strtoint(conv)-1;

        dGij[di, di]:=dGij[di, di]+(2/(abs(at*at*at))*valor(RI,XI));
        dGij[di, dj]:=dGij[di, dj]-(1/(at*at)*valor(RI,XI));
        dGij[dj, di]:=dGij[dj, di]-(1/(at*at)*valor(RI,XI));

        dBij[di, di]:=dBij[di, di]-(2/(abs(at*at*at))*valor(XI,RI));
        dBij[di, dj]:=dBij[di, dj]+(1/(at*at)*valor(XI,RI));
        dBij[dj, di]:=dBij[dj, di]+(1/(at*at)*valor(XI,RI));
    end;

for I := 0 to Nb - 1 do
    begin
        if (bus[i,0]<>1) and (po<ns) then
            begin
                vi:=bus[i, 1];
                ai:=bus[i,2];
                for J := 0 to Nb - 1 do
                    begin
                        vj:=bus[j, 1];
                        aj:=bus[j,2];
                        if drq=1 then
                            begin
                                if (drp=i) and (drp<>j) then
                                    m[po,dr]:=m[po,dr]-vi*vj*(gij[i,j]*(sin(aj-ai))+bij[i,j]*cos(aj-ai));
                                if (drp<>i) and (drp=j) then
                                    m[po,dr]:=m[po,dr]-vi*vj*(gij[i,j]*(-sin(aj-ai))-bij[i,j]*cos(aj-ai));
                            end;

                        if drq=2 then
                            begin
                                if (drp=i) and (drp=j) then
                                    m[po,dr]:=m[po,dr]-2*vi*gij[i,j];
                                if (drp=i) and (drp<>j) then
                                    m[po,dr]:=m[po,dr]-vj*(gij[i,j]*cos((aj-ai))-bij[i,j]*sin((aj-ai)));
                                if (drp<>i) and (drp=j) then
                                    m[po,dr]:=m[po,dr]-vi*(gij[i,j]*cos((aj-ai))-bij[i,j]*sin((aj-ai)));
                            end;
                        if drq=4 then
                            begin
                                m[po,dr]:=m[po,dr]+vi*vj*(dgij[i,j]*cos(aj-ai)-dbij[i,j]*sin(aj-ai));
                            end;
                        end;
                        po:=po+1;
                    end;
                end;
            end;
        end;
    end;
end;

```

```

for l := 0 to Nb - 1 do
  begin
    if (bus[i,0]=3) or (bus[i,0]=4) then
      begin
        vi:=bus[i,1];
        ai:=bus[i,2];

        for J := 0 to Nb - 1 do
          begin
            vj:=bus[j,1];
            aj:=bus[j,2];
            if drq=1 then
              begin
                if (drp=i) and (drp<>j) then
                  M[po,dr]:=M[po,dr]+vi*vj*(gij[i,j]*(-cos((aj-ai)))+bij[i,j]*sin((aj-ai)));
                if (drp<>i) and (drp=j) then
                  M[po,dr]:=M[po,dr]+vi*vj*(gij[i,j]*cos((aj-ai))-bij[i,j]*sin((aj-ai)));
                end;
                if drq=2 then
                  begin
                    if (drp=i) and (drp=j) then
                      m[po,dr]:=m[po,dr]+2*vi*bij[i,j];
                    if (drp=i) and (drp<>j) then
                      m[po,dr]:=m[po,dr]+vj*(bij[i,j]*cos((aj-ai))+gij[i,j]*sin((aj-ai)));
                    if (drp<>i) and (drp=j) then
                      m[po,dr]:=m[po,dr]+vi*(bij[i,j]*cos((aj-ai))+gij[i,j]*sin((aj-ai)));
                    end;

                    if drq=4 then
                      begin
                        m[po,dr]:=m[po,dr]-vi*vj*(dgij[i,j]*sin(aj-ai)+dbij[i,j]*cos(aj-ai));
                      end;
                    end;
                po:=po+1;
              end;
            end;
          end;
        end;
      end;
    end;
  end;

```

```

end;
dr:=dr+1;
end;

```

```

//matriu inversa "original"
i:=0;
while i<ns do
  begin
    mi[i,i]:=1;
    i:=i+1;
  end;
//fi matriu inversa "original"

```

```

//transformaciÃ³ inversa
for Po := 0 to ns - 1 do
begin
propi:=true;
for i := 0 to ns - 1 do
begin
if propi then
begin
RM:=M[po,po];
for J := 0 to ns - 1 do
begin
m[po,j]:= m[po,j]/RM;
Mi[po,j]:= Mi[po,j]/RM;
end;
propi:=false;
end;
if po<>i then
begin
RM:=M[i,po];
for J := 0 to ns - 1 do
begin
Mi[i,j]:=Mi[i,j] - RM * Mi[po,j];
m[i,j]:=m[i,j]- RM * m[po,j];
end;
end;
end;
end;

// fi transformaciÃ³ inversa
K:=k+1;
soluc.RowCount:=k+1;
SetLength(Msol, ns, k+1);//Funcions
soluc.Cells[0,k]:=floattostr(k);
for l := 0 to ns - 1 do
begin
Msol[i,k]:=Msol[i,k-1];
for j := 0 to ns - 1 do
begin
Msol[i,k]:=Msol[i,k]-fpq[j]*mi[i,j];
end;
soluc.Cells[i+1,k]:=floattostr(msol[i,k]);
end;

dr:=0;
while dr<ns do
begin
for po := 0 to nb - 1 do
begin
if ((bus[po,0]<>1) and (dr<ns))then
begin

```

```

        bus[po,2]:=Msol[dr,k];
        dr:=dr+1;
        end;
    end;
for po := 0 to nb - 1 do
begin
    if ((bus[po,0]=3) and (dr<ns)) then
        begin
            bus[po,1]:=Msol[dr,k];
            dr:=dr+1;
            end;
        end;
for po := 0 to nt - 1 do
begin
    if ((tran[po,0]=3) and (dr<ns) )then
        begin
            tran[po,7]:=Msol[dr,k];
            dr:=dr+1;
            end;
        end;
end;
end;

```

```

erra:=abs(Msol[0,k]-Msol[0,k-1]);
for po := 1 to ns - 1 do
begin
if erra<abs((Msol[po,k]-Msol[po,k-1])) then
erra:=abs((Msol[po,k]-Msol[po,k-1]));
end;
soluc.Cells[ns+1,k]:=floattostr(erra);
end;
end;

```

```

procedure TWin2kAppForm.TrafoClick(Sender: TObject);
begin
bitbtn1.Click;
end;

```

```

procedure TWin2kAppForm.utorial1Click(Sender: TObject);
begin
shellexecute(Handle,'open',Pchar('Tutorial.pdf'),Nil,Nil,sw_shownormal);
end;

```

```

procedure TWin2kAppForm.RvSystem1Print(Sender: TObject);
var
xx,v1,v2,a1,a2,v1r,v1i,v2r,v2i,ztr,zti,ylr,yli,p,q,a,Clin,Ymo:real;po,pd,x,y,xm,xmin,ymin,ym:i

```

```

integer;blk:timage;temp:string;esu:tbitmap;
begin
with Sender as TBaseReport do
begin
if rpo>2 then
begin
HV:=Ref[2,2] div 10;
if hv=0 then
begin
xm:=Ref[2,0]+150*e;
ym:=Ref[2,1]+80*e;
xmin:=Ref[2,0]-80*e;
ymin:=Ref[2,1]-80*e;
end;
if hv=1 then
begin
xm:=Ref[2,0]+80*e;
ym:=Ref[2,1]+80*e;
xmin:=Ref[2,0]-80*e;
ymin:=Ref[2,1]-150*e;
end;
if hv=2 then
begin
xm:=Ref[2,0]+80*e;
ym:=Ref[2,1]+80*e;
xmin:=Ref[2,0]-150*e;
ymin:=Ref[2,1]-80*e;
end;
if hv=3 then
begin
xm:=Ref[2,0]+80*e;
ym:=Ref[2,1]+150*e;
xmin:=Ref[2,0]-80*e;
ymin:=Ref[2,1]-80*e;
end;

for pd := 2 to rpo do
begin
x:=Ref[pd,0];
y:=Ref[pd,1];
HV:=Ref[pd,2] div 10;
if (Ref[pd,2] mod 10 >6) and((y+150)>ym) then
begin
xm:=xm+50*e;
ym:=y+150*e;
end;

if (x-20<=xmin) or ((HV=2)and((x-80)<xmin)) then
begin
if HV=2 then

```

```

xmin:=x-80*e
else
xmin:=x-20*e;
end;
if (y-20<=ymin) or ((HV=1)and((y-80)<ymin)) then
begin
if HV=1 then
ymin:=y-80*e
else
ymin:=y-20*e;
end;
if (x+20>=xm) or ((HV=0)and((x+80)>(xm-50))) then
begin
if HV=0 then
xm:=x+80*e
else
xm:=x+20*e;
end;
if (y+20>=ym) or ((HV=3)and((y+80)>(ym-150))) then
begin
if HV=3 then
ym:=y+80*e
else
ym:=y+20*e;
end;
END;

```

```

if ((ym-ymin) div 1500)> ((xm-xmin) div 350) then
exp:=(ym-ymin) div 1500
else
exp:=((xm-xmin) div 350);

```

```

xmin:=xmin;
ymin:=ymin;
xm:=(xm-xmin)div exp;
ym:=(ym-ymin)div exp;
image3.Height:=ym;
image3.Width:=xm;
image3.Picture.Bitmap.Width:=image3.Width;
image3.picture.Bitmap.Height:=image3.Height;
image3.Canvas.Pen.color:=clwhite;
image3.Canvas.Rectangle(0,0,image3.width,image3.Height);
image3.Canvas.Pen.color:=clblack;
ESU:=tbitmap.Create;
for pd := 0 to rpo do
begin
x:=Ref[pd,0]-xmin;
y:=Ref[pd,1]-ymin;

```

```

HV:=(ref[pd,2]) div 10;
  with image3.canvas do
    begin
      if (ref[pd,2]mod 10)=0 then
        moveto((x) div exp,(y) div exp);
      if (ref[pd,2]mod 10)=1 then
        lineto((x) div exp,(y) div exp);
      end;
    if (ref[pd,2]mod 10)=2 then
      dtrafo(image3.Canvas,exp,x,y);
    if (ref[pd,2]mod 10)=3 then
      dbus(image3.Canvas,exp,x,y);
    if (ref[pd,2]mod 10)=4 then
      dgen(image3.Canvas,exp,x,y);
    if (ref[pd,2]mod 10)=5 then
      dcarg(image3.Canvas,exp,x,y);
    if (ref[pd,2]mod 10)=6 then
      dconde(image3.Canvas,exp,x,y);
    if (ref[pd,2]mod 10)=7 then
      begin
        tib:=ref[pd,2]div 10000;
        Nu:=(ref[pd,2]mod 10000)div 100;
        lebus(image3.canvas,tib,Nu,exp,x,y);
      end;
    if (ref[pd,2]mod 10)=8 then
      begin
        ti:=ref[pd,2]div 1000000;
        Nua:=(ref[pd,2]mod 1000000)div 10000;
        Nub:=(ref[pd,2]mod 10000)div 100;
        letran(image3.canvas,ti,Nua,Nub,exp,x,y);
      end;

    esu.Assign(image3.Picture.bitmap);
    xx:=1+6/500*(((550-Xm))/2);
    PrintBitmap(xx,2,1,1, ESU);

  end;
end;

titol(sender);
SetFont('Arial', 16);
if rpo>2 then
  begin
    NewLine;
    NewLine;
  //una pagina sencera son 11 pulgades, partint des punt 1, 48 linies )
  gotoXY(1,1.6);
  PrintCenter('Esquema unifilar',4);
  Ymo:=Ym*6/500+2;
  gotoXY(1,Ymo);

```



```
newline;  
end  
else  
gotoXY(1,1);
```

```
Newpage;  
Titol(sender);  
newline;  
SetFont('Arial', 16);
```

```
PrintCenter('Dades inicials',4);
```

```
SetFont('Arial', 14);
```

```
NewLine;
```

```
NewLine;
```

```
Print('    Barres');
```

```
NewLine;
```

```
NewLine;
```

```
ClearTabs;
```

```
SetTab(0.5, pjLeft, 0.6, 0, 0, 0);
```

```
SetTab(0.8, pjleft, 1, 0, 0, 0);
```

```
SetTab(1, pjRight, 1.8, 0, 0, 0);
```

```
SetTab(1.8, pjRight, 2.6, 0, 0, 0);
```

```
SetTab(2.6, pjRight, 3.4, 0, 0, 0);
```

```
SetTab(3.4, pjRight, 4.2, 0, 0, 0);
```

```
SetFont('Arial', 10);
```

```
Bold := True;
```

```
PrintTab('No');
```

```
PrintTab('Tipus');
```

```
PrintTab('Tensió (v)');
```

```
PrintTab('Fase (rad)');
```

```

PrintTab('Potencia activa (P)');

PrintTab('Potencia reactiva (Q)');

Bold:= False;

NewLine;
for po:=1 to tbus.RowCount do
begin
PrintTab(tbus.cells[1,po]);
PrintTab(tbus.cells[2,po]);
  if (tbus.Cells[2,po] ='Slack') or (tbus.Cells[2,po] ='sl') or (tbus.Cells[2,po] ='SL') then
  begin
    PrintTab(tbus.cells[3,po]);
    PrintTab(tbus.cells[4,po]);
    PrintTab("");
    PrintTab("");
  end;
  if (tbus.Cells[2,po] ='PV') or (tbus.Cells[2,po] ='pv') then
  begin
    PrintTab(tbus.cells[3,po]);
    PrintTab("");
    PrintTab(tbus.cells[5,po]);
    PrintTab("");
  end;
  if (tbus.Cells[2,po] ='PQ') or (tbus.Cells[2,po] ='pq') then
  begin
    PrintTab("");
    PrintTab("");
    PrintTab(tbus.cells[5,po]);
    PrintTab(tbus.cells[6,po]);
  end;
  if (tbus.Cells[2,po] ='PQV') or (tbus.Cells[2,po] ='pqv') then
  begin
    PrintTab(tbus.cells[3,po]);
    PrintTab("");
    PrintTab(tbus.cells[5,po]);
    PrintTab(tbus.cells[6,po]);
  end;
end;

```

```

newline;
end;

```

```

SetFont('Arial', 14);

```

```

NewLine;

```

```

NewLine;

Print('    Elements de transmissió');

NewLine;

NewLine;

ClearTabs;

SetTab(0.2, pjright, 0.8, 0, 0, 0);

SetTab(0.8, pjRight, 1, 0, 0, 0);

SetTab(1.2, pjRight, 1.4, 0, 0, 0);

SetTab(1.7, pjRight, 1.8, 0, 0, 0);

SetTab(2.2, pjRight, 2.4, 0, 0, 0);

SetTab(2.7, pjRight, 3, 0, 0, 0);

SetTab(3.2, pjRight, 4.2, 0, 0, 0);

SetFont('Arial', 10);

Bold := True;

PrintTab('Element');

PrintTab('Conexions');

PrintTab('R(long)');

PrintTab('X(long)');

PrintTab('G(trans)');

PrintTab('B(trans)');

PrintTab('R. Transmissió');

Bold:= False;

NewLine;

  for x:=1 to nt do
  begin
if tran[x-1,0]=1 then
Printtab('Línia');

```

```

if tran[x-1,0]=3 then
Printtab('TrafoV');
if tran[x-1,0]=2 then
Printtab('TrafoC');
if tran[x-1,0]=4 then
Printtab('B. Conensadors');

printtab(floattostr(tran[x-1,1])+'-'+floattostr(tran[x-1,2]));
printtab(floattostr(tran[x-1,3]));
printtab(floattostr(tran[x-1,4]));
printtab(floattostr(tran[x-1,5]));
printtab(floattostr(tran[x-1,6]));
if (tran[x-1,0]=2) then
printtab(floattostr(tran[x-1,7]))
else
printtab("");

newline;
end;
newpage;
Titol(sender);
SetFont('Arial', 16);
PrintCenter('Flux de càrregues',4);

//
NewLine;
ClearTabs;

SetTab(0.2, pjleft, 0.6, 0, 0, 0);

SetTab(0.9, pjleft, 1, 0, 0, 0);

SetTab(1.2, pjRight, 1.8, 0, 0, 0);

SetTab(2, pjRight, 2.6, 0, 0, 0);

SetTab(2.8, pjRight, 3.4, 0, 0, 0);

SetTab(3.6, pjRight, 4.2, 0, 0, 0);

SetTab(4.4, pjRight, 5, 0, 0, 0);

SetFont('Arial', 10);

Bold := True;

PrintTab('Element');

PrintTab('Conexions');

```

```

PrintTab('P(ca-cb)');

PrintTab('P(cb-ca)');

PrintTab('Q(ca-cb)');

PrintTab('Q(cb-ca)');

Bold:= False;

NewLine;

for x:=0 to nt-1 do
begin
if tran[x,0]<>4 then
begin
temp:=floattostr(tran[x,1]-1);
V1:=bus[strtoint(temp),1];
temp:=floattostr(tran[x,2]-1);
V2:=bus[strtoint(temp),1];
temp:=floattostr(tran[x,1]-1);
a1:=bus[strtoint(temp),2];
temp:=floattostr(tran[x,2]-1);
a2:=bus[strtoint(temp),2];
V1r:=v1*cos(a1);
V1i:=v1*sin(a1);
V2r:=v2*cos(a2);
V2i:=v2*sin(a2);

Ztr:=tran[x,3];
Zti:=tran[x,4];
Ylr:=tran[x,5];
Yli:=tran[x,6];
a:=tran[x,7];

If tran[x,0]=1 then
begin
PrintTab('Línia');
PrintTab(floattostr(tran[x,1])+'-'+floattostr(tran[x,2]));
Re:=(V1r-V2r)*valor(Ztr,Zti)+(V1i-V2i)*valor(Zti,Ztr)+Ylr/2*V1r-Yli/2*V1i;
Im:=(V1i-V2i)*valor(Ztr,Zti)-(V1r-V2r)*valor(Zti,Ztr)+Yli/2*V1r+Ylr/2*V1i;
P:=(v1r*Re+v1i*Im);
Q:=(-v1r*Im+v1i*Re);

temp:=floattostr(tran[x,1]-1);
if bus[strtoint(temp),0]=1 then
begin
bus[strtoint(temp),3]:=bus[strtoint(temp),3]+p;
bus[strtoint(temp),4]:=bus[strtoint(temp),4]+q;
end;

```

```

if bus[strtoint(temp),0]=2 then
begin
bus[strtoint(temp),4]:=bus[strtoint(temp),4]+q;
end;

V2r:=v1*cos(a1);
V2i:=v1*sin(a1);
V1r:=v2*cos(a2);
V1i:=v2*sin(a2);
Re:=(V1r-V2r)*valor(Ztr,Zti)-(V1i-V2i)*valor(-Zti,Ztr)+Ylr/2*V1r-Yli/2*V1i;
Im:=(V1i-V2i)*valor(Ztr,Zti)+(V1r-V2r)*valor(-Zti,Ztr)+Yli/2*V1r+Ylr/2*V1i;
PrintTab(floattostr(P));
P:=(v1r*Re+v1i*Im);
PrintTab(floattostr(P));
PrintTab(floattostr(Q));
Q:=(-v1r*Im+v1i*Re);
PrintTab(floattostr(Q));

temp:=floattostr(tran[x,2]-1);
if bus[strtoint(temp),0]=1 then
begin
bus[strtoint(temp),3]:=bus[strtoint(temp),3]+p;
bus[strtoint(temp),4]:=bus[strtoint(temp),4]+q;
end;
if bus[strtoint(temp),0]=2 then
begin
bus[strtoint(temp),4]:=bus[strtoint(temp),4]+q;
end;

end;
If tran[x,0]=2 then//no variable
begin
PrintTab('Trfc');
PrintTab(floattostr(tran[x,1]+'-'+floattostr(tran[x,2])));
V1r:=v1*cos(a1);
V1i:=v1*sin(a1);
V2r:=v2*cos(a2);
V2i:=v2*sin(a2);
Re:=(V1r*valor(Ztr,Zti)+V1i*valor(Zti,Ztr))/(a*a)-(V2r*valor(Ztr,Zti)+V2i*valor(Zti,Ztr))/(a);
Im:=(V1i*valor(Ztr,Zti)+V1r*valor(-Zti,Ztr))/(a*a)-(V2i*valor(Ztr,Zti)+V2r*valor(-Zti,Ztr))/(a);
P:=(v1r*Re+v1i*Im);
Q:=(-v1r*Im+v1i*Re);

temp:=floattostr(tran[x,1]-1);
if bus[strtoint(temp),0]=1 then
begin
bus[strtoint(temp),3]:=bus[strtoint(temp),3]+p;
bus[strtoint(temp),4]:=bus[strtoint(temp),4]+q;
end;
if bus[strtoint(temp),0]=2 then

```

```

begin
bus[strtoint(temp),4]:=bus[strtoint(temp),4]+q;
end;

V2r:=v1*cos(a1);
V2i:=v1*sin(a1);
V1r:=v2*cos(a2);
V1i:=v2*sin(a2);
Re:=(V1r*valor(Ztr,Zti)+V1i*valor(Zti,Ztr))-(V2r*valor(Ztr,Zti)+V2i*valor(Zti,Ztr))/(a);
Im:=(V1i*valor(Ztr,Zti)+V1r*valor(-Zti,Ztr))-(V2i*valor(Ztr,Zti)+V2r*valor(-Zti,Ztr))/(a);
PrintTab(floattostr(P));
P:=(v1r*Re+v1i*Im);
PrintTab(floattostr(P));
PrintTab(floattostr(Q));
Q:=(-v1r*Im+v1i*Re);
PrintTab(floattostr(Q));

temp:=floattostr(tran[x,2]-1);
if bus[strtoint(temp),0]=1 then
begin
bus[strtoint(temp),3]:=bus[strtoint(temp),3]+p;
bus[strtoint(temp),4]:=bus[strtoint(temp),4]+q;
end;
if bus[strtoint(temp),0]=2 then
begin
bus[strtoint(temp),4]:=bus[strtoint(temp),4]+q;
end;
end;
If tran[x,0]=3 then
begin
PrintTab('TrfV');
PrintTab(floattostr(tran[x,1])+'-'+floattostr(tran[x,2]));
V1r:=v1*cos(a1);
V1i:=v1*sin(a1);
V2r:=v2*cos(a2);
V2i:=v2*sin(a2);
Re:=(V1r*valor(Ztr,Zti)+V1i*valor(Zti,Ztr))/(a*a)-(V2r*valor(Ztr,Zti)+V2i*valor(Zti,Ztr))/(a);
Im:=(V1i*valor(Ztr,Zti)+V1r*valor(-Zti,Ztr))/(a*a)-(V2i*valor(Ztr,Zti)+V2r*valor(-Zti,Ztr))/(a);
P:=(v1r*Re+v1i*Im);
Q:=(-v1r*Im+v1i*Re);

temp:=floattostr(tran[x,1]-1);
if bus[strtoint(temp),0]=1 then
begin
bus[strtoint(temp),3]:=bus[strtoint(temp),3]+p;
bus[strtoint(temp),4]:=bus[strtoint(temp),4]+q;
end;
if bus[strtoint(temp),0]=2 then
begin
bus[strtoint(temp),4]:=bus[strtoint(temp),4]+q;

```

```

end;

V2r:=v1*cos(a1);
V2i:=v1*sin(a1);
V1r:=v2*cos(a2);
V1i:=v2*sin(a2);
Re:=(V1r*valor(Ztr,Zti)+V1i*valor(Zti,Ztr))-(V2r*valor(Ztr,Zti)+V2i*valor(Zti,Ztr))/(a);
Im:=(V1i*valor(Ztr,Zti)+V1r*valor(-Zti,Ztr))-(V2i*valor(Ztr,Zti)+V2r*valor(-Zti,Ztr))/(a);
PrintTab(floattostr(P));
PrintTab(floattostr(Q));
P:=(v1r*Re+v1i*Im);
PrintTab(floattostr(P));
Q:=(-v1r*Im+v1i*Re);
PrintTab(floattostr(Q));

temp:=floattostr(tran[x,2]-1);
if bus[strtoint(temp),0]=1 then
begin
bus[strtoint(temp),3]:=bus[strtoint(temp),3]+p;
bus[strtoint(temp),4]:=bus[strtoint(temp),4]+q;
end;
if bus[strtoint(temp),0]=2 then
begin
bus[strtoint(temp),4]:=bus[strtoint(temp),4]+q;
end;
end;
end;

newline;
end;

newpage;
Titol(sender);
SetFont('Arial', 16);
PrintCenter('Solució Iterativa',4);

SetFont('Arial', 14);

newline;

Print('    Barres');

NewLine;

NewLine;

ClearTabs;

SetTab(0.5, pjLeft, 0.6, 0, 0, 0);

```



```

SetTab(0.8, pjleft, 1, 0, 0, 0);
SetTab(1, pjRight, 1.8, 0, 0, 0);
SetTab(1.8, pjRight, 2.6, 0, 0, 0);
SetTab(2.6, pjRight, 3.4, 0, 0, 0);
SetTab(3.4, pjRight, 4.2, 0, 0, 0);
SetFont('Arial', 10);
Bold := True;
PrintTab('No');
PrintTab('Tipus');
PrintTab('Tensió (v)');
PrintTab('Fase (rad)');
PrintTab('Potencia activa (P)');
PrintTab('Potencia reactiva (Q)');
Bold:= False;
NewLine;
for x:=1 to nb do
begin
PrintTab(inttostr(x));
if bus[x-1,0]=1 then
PrintTab('Slack');
if bus[x-1,0]=2 then
PrintTab('PV');
if bus[x-1,0]=3 then
PrintTab('PQ');
if bus[x-1,0]=4 then
PrintTab('PQV');

PrintTab(floattostr(bus[x-1,1]));
PrintTab(floattostr(bus[x-1,2]));
PrintTab(floattostr(bus[x-1,3]));
PrintTab(floattostr(bus[x-1,4]));
PrintTab(floattostr(bus[x-1,5]));
newline;
end;
newline;
Print('      Marge d'error '+floattostr(erref)+' . Error obtingut '+floattostr(erra)+' en la

```

```
'+inttostr(k)+' iteració');  
newline;  
  
newline;  
SetFont('Arial', 14);  
  
NewLine;  
NewLine;  
  
Print('      Elements de transmissió');  
  
NewLine;  
  
NewLine;  
  
ClearTabs;  
  
SetTab(0.2, pjright, 0.8, 0, 0, 0);  
SetTab(0.8, pjRight, 1, 0, 0, 0);  
SetTab(1.2, pjRight, 1.4, 0, 0, 0);  
SetTab(1.7, pjRight, 1.8, 0, 0, 0);  
SetTab(2.2, pjRight, 2.4, 0, 0, 0);  
SetTab(2.7, pjRight, 3, 0, 0, 0);  
SetTab(3.2, pjRight, 4.2, 0, 0, 0);  
  
SetFont('Arial', 10);  
  
Bold := True;  
  
PrintTab('Element');  
PrintTab('Conexions');  
PrintTab('R(long)');  
PrintTab('X(long)');  
PrintTab('G(trans)');  
PrintTab('B(trans)');  
PrintTab('R. Transmissió');
```

```
Bold:= False;
```

```
NewLine;
```

```
  for x:=1 to nt do
  begin
if tran[x-1,0]=1 then
Printtab('Línia');
if tran[x-1,0]=3 then
Printtab('TrafoV');
if tran[x-1,0]=2 then
Printtab('TrafoC');
if tran[x-1,0]=4 then
Printtab('B. Conensadors');

printtab(floattostr(tran[x-1,1]+1)+'-'+floattostr(tran[x-1,2]+1));
printtab(floattostr(tran[x-1,3]));
printtab(floattostr(tran[x-1,4]));
printtab(floattostr(tran[x-1,5]));
printtab(floattostr(tran[x-1,6]));
if (tran[x-1,0]=2) or (tran[x-1,0]=3) then
printtab(floattostr(tran[x-1,7]));

newline;
end;
```

```
//
end;
end;
```

```
procedure TWin2kAppForm.Button15Click(Sender: TObject);
var
Pi,Pf:integer;
begin
tnom:=filename;
pr:=tnom;
Pi:=Pos('\',tnom);

while pi>0 do
begin
pf:=Length(pr);
pr:=copy(pr,pi+1,pf-pi);
Pi:=Pos('\',pr);
end;
pf:=Length(pr);
Pi:=Pos('.',pr);
```

```
pr:=copy(pr,0,pi-1);
win2kappform.Caption:='DaDaSystem-'+pr;
```

```
end;
```

```
procedure TWin2kAppForm.TeleDrawCell(Sender: TObject; ACol, ARow: Integer;
  Rect: TRect; State: TGridDrawState);
```

```
var
```

```
  sTexto: String;           // Texto que va a imprimir en la celda actual
  Alineacion: TAlignment;  // Alineación que le vamos a dar al texto
  iAnchoTexto: Integer;    // Ancho del texto a imprimir en pixels
```

```
begin
```

```
  tele.ColWidths[0]:=10;
```

```
  with tele.Canvas do
```

```
    begin
```

```
      Font.Name := tele.Font.Name;
```

```
      Font.Size := tele.Font.Size;
```

```
      if arow=0 then
```

```
        Alineacion := taCenter
```

```
      else
```

```
        Alineacion := taleftjustify;
```

```
      if gdFixed in State then
```

```
        begin
```

```
          Brush.Color := clBtnFace;
```

```
          Font.Color := clblack;
```

```
          Font.Style := [fsBold];
```

```
        end;
```

```
        sTexto := tele.Cells[ACol,ARow];
```

```
        FillRect( Rect );
```

```
        iAnchoTexto := TextWidth( sTexto );
```

```
        case Alineacion of
```

```
          taLeftJustify: TextOut( Rect.Left + 5, Rect.Top + 2, sTexto );
```

```
          taCenter: TextOut( Rect.Left + ( ( Rect.Right - Rect.Left ) - iAnchoTexto ) div 2,
Rect.Top + 2, sTexto );
```

```
          taRightJustify: TextOut( Rect.Right - iAnchoTexto - 2, Rect.Top + 2, sTexto );
```

```
        end;
```

```
      end;
```

```
end;
```

```
procedure TWin2kAppForm.EtipusBChange(Sender: TObject);
```

```
begin
```

```
  memo3.readonly:=false;
```

```
  if (EtipusB.text='Slack o Bus de referencia') then
```

```
    begin
```

```
      memo3.Clear;
```

```
      memo3.lines.Add("");
```

```
      memo3.lines.Add('Recorda que tan sols hi ha un bus Slack i que aquest ha de ser el
primer bus.');
```

```

end;
if (EtipusB.text='PQ o Bus de Carga') then
begin
memo3.Clear;
memo3.lines.Add("");
memo3.lines.Add('Recorda que en unes barres PQ tan sols coneixem la potència activa
i reactiva');
end;
if (EtipusB.text='PQV o Bus de tensio controlada') then
begin
memo3.Clear;
memo3.lines.Add("");
memo3.lines.Add('Recorda que les barres PQV són unes barres PQ on d'alguna
manera podem controlar la tensiÃ³ en modul, com per exemple amb un transformador
amb toma de regulaciÃ³ variable');
end;
if (EtipusB.text='PV o Bus de tensio Controlada') then
begin
memo3.Clear;
memo3.lines.Add("");
memo3.lines.Add('Recorda que en unes barres coneixem la potencia activa i la tensiÃ³
en modul');
end;
memo3.readonly:=true;

```

```
end;
```

```

procedure TWin2kAppForm.Button3Click(Sender: TObject);
begin
tbus.Options:= tbus.Options + [goRowSelect];
editar:=true;
end;

```

```

procedure TWin2kAppForm.TbusDrawCell(Sender: TObject; ACol, ARow: Integer;
Rect: TRect; State: TGridDrawState);
var
sTexto: String; // Texto que va a imprimir en la celda actual
Alineacion: TAlignment; // Alineaci3n que le vamos a dar al texto
iAnchoTexto: Integer; // Ancho del texto a imprimir en pixels
begin
tbus.ColWidths[0]:=10;
with tbus.Canvas do
begin
Font.Name := tbus.Font.Name;
Font.Size := tbus.Font.Size;
if arow=0 then
Alineacion := taCenter
else
Alineacion := taLeftJustify;

```

```

if gdFixed in State then
begin
  Brush.Color := clBtnFace;
  Font.Color := clblack;
  Font.Style := [fsBold];
end;
sTexto := tbus.Cells[ACol,ARow];
FillRect( Rect );
iAnchoTexto := TextWidth( sTexto );

case Alineacion of
  taLeftJustify: TextOut( Rect.Left + 5, Rect.Top + 2, sTexto );
  taCenter: TextOut( Rect.Left + ( ( Rect.Right - Rect.Left ) - iAnchoTexto ) div 2,
Rect.Top + 2, sTexto );
  taRightJustify: TextOut( Rect.Right - iAnchoTexto - 2, Rect.Top + 2, sTexto );
end;
end;
end;

```

```

procedure TWin2kAppForm.Button1Click(Sender: TObject);
var
j,i:integer;
temp : TStringList;
begin
tbus.RowCount:=tbus.RowCount+1;
temp:=tstringlist.create;
with tbus do
begin
j:=rowcount-1;
  while j>row do
    begin
      rows[j].assign(rows[j-1]);
      j:=j-1;
    end;
temp.Clear;
rows[j].Clear;
end;
button3.Click;
end;

```

```

procedure TWin2kAppForm.Button4Click(Sender: TObject);
begin
  editar:=false;
  tbus.Options:= tbus.Options - [goRowSelect];
  Tbus.cells[1,row]:=Ebus.Text;
  Tbus.cells[2,row]:=copy(ETipusB.Text,1,5);
  Tbus.cells[3,row]:=EV.Text;
  Tbus.cells[4,row]:=EAngle.Text;

```

```
Tbus.cells[5,row]:=EP.Text;  
Tbus.cells[6,row]:=EQ.Text;
```

```
end;
```

```
procedure TWin2kAppForm.TbusSelectCell(Sender: TObject; ACol,  
  ARow: Integer; var CanSelect: Boolean);
```

```
begin  
Col:=ACol;  
Row:=ARow;  
end;
```

```
procedure TWin2kAppForm.Button2Click(Sender: TObject);  
begin
```

```
  editar:=false;  
  tbus.Options:= tbus.Options - [goRowSelect];  
  Ebus.Text:= "";  
  ETipusB.Text:= "";  
  EV.Text:= "";  
  EAngle.Text:= "";  
  EP.Text:= "";  
  EQ.Text:= "";
```

```
end;
```

```
procedure TWin2kAppForm.TbusClick(Sender: TObject);  
begin
```

```
  if editar then  
  begin  
    Ebus.Text:=Tbus.cells[1,row];  
    ETipusB.Text:=Tbus.cells[2,row];  
    EV.Text:=Tbus.cells[3,row];  
    EAngle.Text:=Tbus.cells[4,row];  
    EP.Text:=Tbus.cells[5,row];  
    EQ.Text:=Tbus.cells[6,row];  
  end;  
end;
```

```
procedure TWin2kAppForm.Button7Click(Sender: TObject);  
begin
```

```
  if row<>0 then  
  if tbus.rowcount >2 then  
  begin  
    with tbus do  
    begin  
      j:=row;  
      while j<rowcount-1 do  
        begin  
          rows[j].assign(rows[j+1]);
```

```

    j:=j+1;
    end;
rows[j].Clear;
rowcount:=rowcount-1;
end;
end
else
tbus.Rows[1].Clear;
end;

```

```

procedure TWin2kAppForm.Button8Click(Sender: TObject);
var
j,i:integer;
temp : TStringList;
begin
tele.RowCount:=tele.RowCount+1;
temp:=tstringlist.create;

with tele do
begin
j:=rowcount-1;
while j>rowe do
begin
rows[j].assign(rows[j-1]);
j:=j-1;
end;
temp.Clear;
rows[j].Clear;
end;
end;

```

```

procedure TWin2kAppForm.Button11Click(Sender: TObject);
begin
tele.Options:= tele.Options + [goRowSelect];
editar:=true;
Etipuse.Text:= Tele.cells[1,rowe];
ECA.Text:= Tele.cells[2,rowe];
ECB.Text:= Tele.cells[3,rowe];
ERL.Text:= Tele.cells[4,rowe];
EXL.Text:= Tele.cells[5,rowe];
EGT.Text:= Tele.cells[6,rowe];
EBT.Text:= Tele.cells[7,rowe];
EREG.Text:=Tele.cells[8,rowe];
etipuse.Enabled;
If tele.cells[9,rowe]='S' then
begin
RegCheck.Checked:=true;
Bcheck.Checked:=true;
end;
end;

```



```

procedure TWin2kAppForm.Button13Click(Sender: TObject);
begin
  editar:=false;
  tbus.Options:= tbus.Options - [goRowSelect];
end;

```

```

procedure TWin2kAppForm.Button14Click(Sender: TObject);
begin
if rowe<>0 then
if tele.rowcount >2 then
begin
  for i := rowe to tele.rowcount - 1 do
  begin
    tele.rows[i][0] := tele.rows[i + 1][0];
    tele.rows[i][1] := tele.rows[i + 1][1];
  end;
  tele.RowCount := tele.RowCount - 1;
end
else
tele.Rows[rowe].Clear;

end;

```

```

procedure TWin2kAppForm.TeleSelectCell(Sender: TObject; ACol,
  ARow: Integer; var CanSelect: Boolean);
begin
  Cole:=ACol;
  Rowe:=ARow;
end;

```

```

procedure TWin2kAppForm.EtipuseChange(Sender: TObject);
begin
  memo2.readonly:=true;
  memo2.Clear;
if (EtipusE.text='Línia') then
  begin
    panel5.Visible:=true;
    panel7.hide;
    memo2.lines.Add("");
    memo2.lines.Add('Recorda que al ser un element simètric és indiferent com es connecti
la línia');
    label13.Visible:=false;
    Ereg.Visible:=false;
    RegCheck.Visible:=false;
  end;
if (EtipusE.text='Trafo') then
  begin
    panel5.Visible:=true;
    panel7.hide;

```

```

memo2.lines.Add("");
memo2.lines.Add('Recorda que la primera connexió és la part de alta tensió. Connecta-
ho correctament ja que no es un element simètric.');
```

label13.Visible:=true;

```

Ereg.Visible:=true;
RegCheck.Visible:=true;
end;
if (EtipusE.text='B.Condensadors') then
begin
panel5.Hide;
panel7.Visible:=true;
memo2.lines.Add("");
memo2.lines.Add('La bateria de condensadors tan sols esta connectada en una barra,
per tant només hi haurà una connexió vàlida que és la primera.');
```

end;

```

memo2.readonly:=false;
end;
```

```

procedure TWin2kAppForm.FormResize(Sender: TObject);
var
pos,Ws:integer;
begin
if tbus.Width >400 then
begin
tbus.ColWidths[0]:=10;
ws:=(tbus.Width-10) div 6;
pos:=1;
while pos<(tbus.ColCount) do
begin
tbus.ColWidths[pos]:=ws;
pos:=pos+1;
end;
end;
```

```

if tele.Width >650 then
begin
tele.ColWidths[0]:=10;
ws:=(tele.Width-(20)) div 9;
pos:=1;
while pos<(tele.ColCount) do
begin
tele.ColWidths[pos]:=ws;
pos:=pos+1;
end;
end
else
begin
tele.ColWidths[0]:=10;
pos:=1;
while pos<(tele.ColCount) do
```

```

begin
tele.ColWidths[pos]:=80;
pos:=pos+1;
end;
end;
end;

```

```

procedure TWin2kAppForm.ToolButton10Click(Sender: TObject);
begin
rvsystem1.Execute;
end;

```

```

procedure TWin2kAppForm.Button5Click(Sender: TObject);
begin
image4.Stretch:=true;
image4.Width:=image4.Width-10;
image4.Height:=image4.Height-10;
end;

```

```

procedure TWin2kAppForm.BitBtn11Click(Sender: TObject);
begin
// TEnir en compte per es reedit de s'imatge final no fa falte fer un tamany enorme inicial!
image4.Width:=strtoint(edit4.Text);
image4.Height:=strtoint(edit5.Text);
image4.Picture.Graphic.Width:=strtoint(edit4.Text);
image4.Picture.Graphic.Height:=strtoint(edit5.Text);

end;

```

```

procedure TWin2kAppForm.ETRClick(Sender: TObject);
var
x,y:integer;temp:string;
begin
image2.Canvas.rectangle(0,0,70,70);
x:=0;
y:=0;
nua:=strtoint(tele.Cells[2,pt+1]);
if tele.Cells[3,pt+1]<>" then
nub:=strtoint(tele.Cells[3,pt+1]);
if (tele.Cells[1,pt+1]='Linia') or (tele.Cells[1,rowe]='Lin') or (tele.Cells[1,rowe]='LIN') or
(tele.Cells[1,rowe]='lin') then
ti:=2;
if (tele.Cells[1,pt+1]='Trafo') or (tele.Cells[1,rowe]='Tr') or (tele.Cells[1,rowe]='TR') or
(tele.Cells[1,rowe]='tr') then
ti:=3;
if tele.Cells[1,pt+1]='B.Cond' then
ti:=4;

```

```

letran(image2.canvas,ti,Nua,Nub,e,x,y);

end;

procedure TWin2kAppForm.BitBtn6Click(Sender: TObject);
begin
if pb<(tbus.RowCount-2) then
pb:=pb+1;

if bbs.Down then
bbs.Click;
end;

procedure TWin2kAppForm.BitBtn9Click(Sender: TObject);
begin
if pb>0 then
pb:=pb-1;

if bbs.Down then
bbs.Click;
end;

procedure TWin2kAppForm.BitBtn8Click(Sender: TObject);
begin
if pt<(tele.RowCount-2) then
pt:=pt+1;

if etr.Down then
etr.Click;
end;

procedure TWin2kAppForm.BitBtn7Click(Sender: TObject);
begin
if pt>0 then
pt:=pt-1;

if etr.Down then
etr.Click;

end;

procedure TWin2kAppForm.TeleClick(Sender: TObject);
begin
if editar then
button11.Click;
end;

procedure TWin2kAppForm.EtipuseSelect(Sender: TObject);

```

```

begin
  memo2.readonly:=false;
  memo2.Clear;
if (EtipusE.text='Línia') then
  begin
    panel5.Visible:=true;
    panel7.hide;
    memo2.lines.Add("");
    memo2.lines.Add('Recorda que al ser un element simètric és indiferent com es connecti
la línia');
    end;
if (EtipusE.text='Trafo') then
  begin
    panel5.Visible:=true;
    panel7.hide;
    memo2.lines.Add("");
    memo2.lines.Add('Recorda que la primera connexió és la part de alta tensió. Connecta-
ho correctament ja que no es un element simètric.');
```

end;

```

if (EtipusE.text='B.Condensadors') then
  begin
    panel5.Hide;
    panel7.Visible:=true;
    memo2.lines.Add("");
    memo2.lines.Add('La bateria de condensadors tan sols esta connectada en una barra,
per tant només hi haurà una connexió vàlida que és la primera.');
```

end;

```

memo2.readonly:=true;
end;

procedure TWin2kAppForm.TeleDbClick(Sender: TObject);
var
  i,j:integer;
  temp : TStringList;
begin
  temp:=tstringlist.create;
  with tele do
  begin
  for i:=fixedrows to rowcount-2 do
    begin
    for j:=i+1 to rowcount-1 do
      begin
        if AnsiCompareText(Cells[cole,i], Cells[Cole,j]) < 0 then
          begin
            temp.assign(rows[j]);
            rows[j].assign(rows[i]);
            rows[i].assign(temp);
          end
        else // descendente

```

```

if AnsiCompareText(Cells[cole,i], Cells[cole,j]) > 0 then
begin
temp.assign(rows[j]);
rows[j].assign(rows[i]);
rows[i].assign(temp);
end;
end;
end;
end;
end;
temp.free;
end;

```

```

procedure TWin2kAppForm.TbusDbfClick(Sender: TObject);
var
i,j:integer;
temp : TStringList;
begin
i:=0;
j:=0;

```

```

temp:=tstringlist.create;
with tbus do
begin
for i:=fixedrows to rowcount-2 do
begin
for j:=i+1 to rowcount-1 do
begin
if strtoint(Cells[1,i])> strtoint(Cells[1,j]) then
begin
temp.assign(rows[j]);
rows[j].assign(rows[i]);
rows[i].assign(temp);
end
end;
end;
temp.free;
end;

```

```

procedure TWin2kAppForm.RegCheckClick(Sender: TObject);
begin
if not Regcheck.checked then
begin
label13.Font.Color:=clMedGray;
Ereg.Enabled:=false;
Ereg.color:=clBtnFace;

```

```
end
else
begin
  label13.Font.Color:=clWindowText;
  ereg.Enabled:=true;
  ereg.color:=clwindow;
end;
end;

end.
```

4.3. ANNEX 3: TUTORIAL

DaDaSystem 1.2

TUTORIAL

INDEX

1 Introducció a DaDaSystem 1.0.

2 Instal·lació de DaDaSystem 1.0.

1.2.1 Prerequisits.

1.2.2 Instal·lació des de arxiu InstDDS.EXE.

3 Execució de DaDaSystem 1.0 .

4 Entorn gràfic de DaDaSystem 1.0.

1.4.1 Barra de títol.

1.4.2 Barra de Menú.

1.4.3 Barra d'eines.

1.4.4 Entorn de treball.

1.4.4.1 Barres.

1.4.4.2 Elements de transmissió.

1.4.4.3 Esquema unifilar.

5 Utilització del programa DaDaSystem 1.0.

5.1 Introducció de dades.

5.1.1 Introducció de barres.

5.1.1.1 Conceptes bàsics.

5.1.1.2 Tipus de barres.

5.1.2 Introducció dels elements de transmissió.

5.1.2.1 Conceptes bàsics.

5.1.2.2 Tipus de d'elements de transmissió.

5.2 Representació de l'esquema unifilar.

5.3 Resolució del problema i exportació de dades.

6 Obrir/Guardar arxius del programa DaDaSystem 1.0.

6.1 Nou arxiu.

6.2 Obrir arxiu.

6.3 Guardar arxiu.

7. Exportació dels resultats del programa DaDaSystem 1.0.

1.1 Introducció a DaDaSystem 1.0

DaDaSystem 1.0 és un programa de càlcul del flux de càrregues per sistemes de potencia basat amb el sistema de resolució de Newton-Raphson. Aquest programa funciona en sistemes operatius Windows 9.X o superiors.

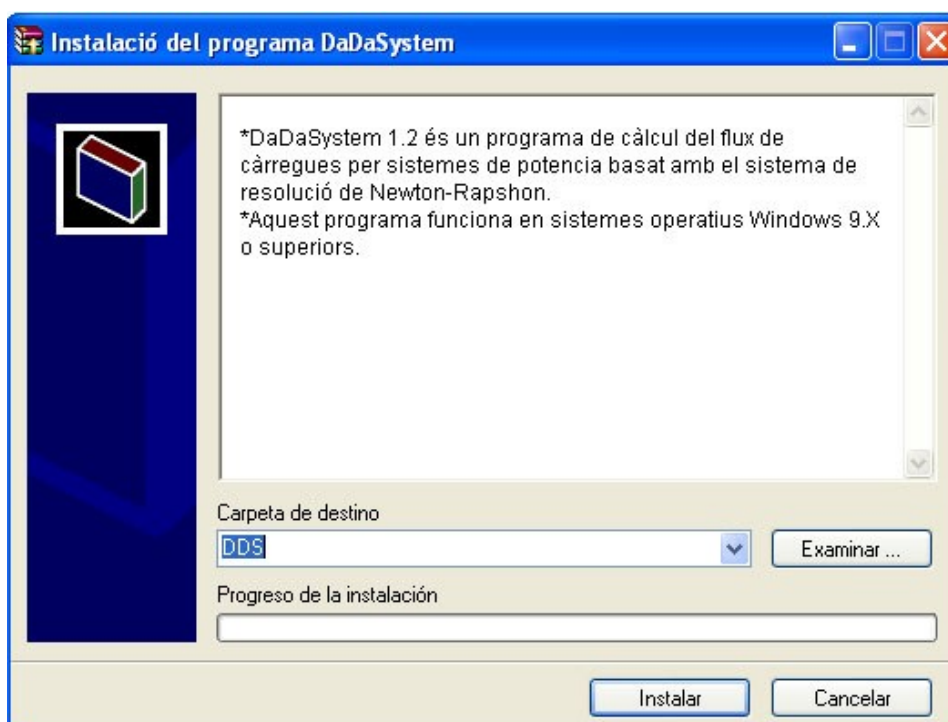
1.2.1 Instal·lació de DaDaSystem 1.0

1.2.1.1 Prerequisits

Per poder instal·lar DaDaSystem al teu ordinador es necessari que aquest tingui unes prestacions per suportar un entorn compatible amb Windows 9.x o superior. A més has de tenir instal·lat un lector d'arxius PDF.

1.2.1.2 Instal·lació des de arxiu InstDDS.EXE

Executi l'arxiu InstDDS.exe



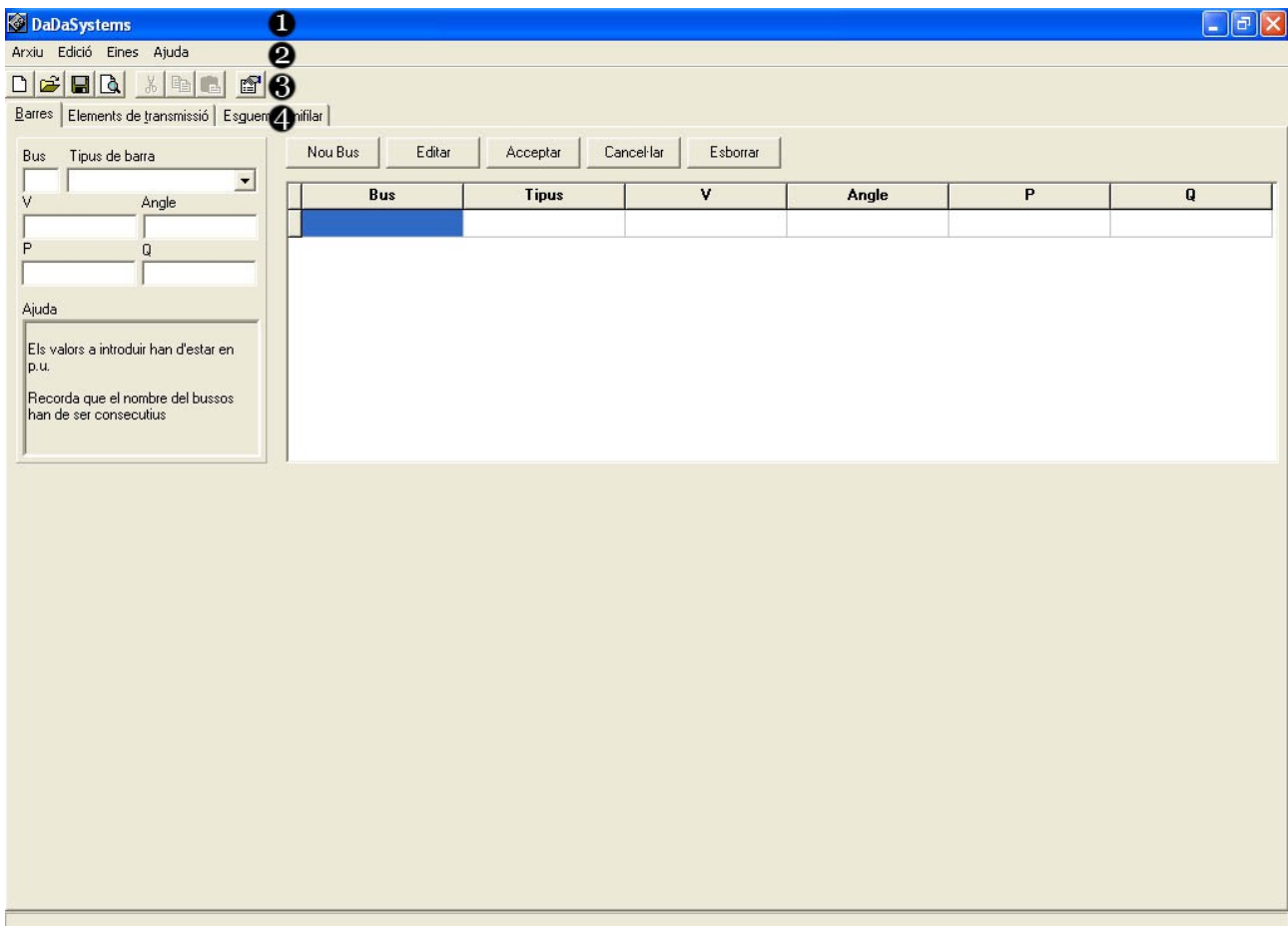
Selecciona la carpeta de destí, per defecte DDS.

1.3 Execució de DaDaSystem 1.0

Una vegada instal·lat el programa pot accedir a ell per l'accés directa de l'escriptori, *DDS* o per la carpeta arrel que per defecte es *C:\DaDaSystem\DDS.EXE*

1.4 Entorn gràfic de DaDaSystem 1.0

Una cop s'ha executat el programa *DDS* es troba davant aquesta pantalla:



On:

- 1 Barra de títol
- 2 Barra de Menú
- 3 Barra d'eines
- 4 Entorn de treball

1.4.1 Barra de títol

A la *Barra de títol* tenim el nom de l'aplicació així com el nom de l'arxiu amb el que estem treballant. També hi trobam els controls bàsic d'una aplicació: minimitzar,maximitzar/reduir i tancar.

1.4.2 Barra de Menú

A la *Barra de Menú* en aquesta barra podem trobar els següents menús:

Arxiu, amb tecla d'accés ràpid alt+A:

Nou: Crees un arxiu nou. Si estàs treballant en un arxiu et sollicitara que vols fer amb l'arxiu actual. Un cop guardat, si cal, es procedirà a presentar un entorn nou.

Obrir: Obris un arxiu de text anteriorment guardat per DDS. Si no es un arxiu creat per DDS aquest no tindrà cap efecte pràctic.

Guardar: Guardes un l'arxiu en que estàs treballant guardant les dades del les barres i elements de transmissió com també l'esquema unifilar. Aquesta opció de guardat et permet una posterior edició o modificació de l'arxiu.

Guardar com...: Fa la mateixa funció que *guardar* però et deixa guardar l'arxiu actual però canviant-li el nom.

Exportar: Et permet exportar els resultats, un cop realitzats tots els càlculs, s'utilitza un exportador de dades anomenat Rave Reports. Aquest programa permet l'exportació a diferents formats com PDF, RTF, TXT i NDR. Aquest arxius ja estan guardats en mode presentació però es recomana, per un resultat optm, la utilització del format PDF.

Sortir: Tanca l'aplicació.

Edició, amb tecla d'accés ràpid alt+E:

Retallar: Copia al porta papers i elimina la selecció actual.

Copiar: Copia al porta papers.

Enganxar: Enganxa la informació del porta papers.

Eines, amb tecla d'accés ràpid alt+N:

Opcions: Et permet modificar el nivell d'error màxim a l'hora de calcular les iteracions.

Càlcul Iteratiu: Comença el càlcul iteratiu, per què aquest càlcul tengui efecte han d'estar definit tot el sistema així com s'explica a l'apartat 1.5.1 *Introducció de dades* d'aquest tutorial

Ajuda, amb tecla d'accés ràpid alt+U:

Tutorial: Accés al tutorial del programa.

Sobre...: Informació del programa i del creador.

1.4.3 Barra d'eines

A la *Barra d'eines* tenim varis accessos ràpids de funcions comuns:



Accés ràpid a *Arxiu/Nou*.



Accés ràpid a *Arxiu/Obrir*.



Accés ràpid a *Arxiu/Guardar*.



Accés ràpid a *Arxiu/Exportar*.



Accés ràpid a *Edició/Retallar*.



Accés ràpid a *Edició/Copiar*.



Accés ràpid a *Edició/Enganxar*.



Accés ràpid a *Eines/Càlcul iteratiu*

1.4.4 Entorn de treball

En l'*Entorn de treball* es on introduïrem i modificarem les dades. Tenim tres pestanyes: *Barres*, *elements de transmissió* i *Esquema unifilar*.

En l'*Entorn de treball* es poden introduir les dades, tant a *Barres* com a *Elements de transmissió*, per dues vies. La inserció es pot fer pas a pas com s'explicara a continuació o modificant els valors directament en la taula tot i que no es recomana fer-ho a menys que siguis un usuari expert.

1.4.4.1 Barres

Bus	Tipus	V	Angle	P	Q
1	Slack	1,05		0,25	0,25
2	PV	1,03		0,25	0,25
3	PQ			0,6	0,3

a2	a3	V3	Error
-0,015204226628486	-0,0145879723410034	1,03484320557491	0,034843205574913
-0,015277034528548	-0,0143011661415709	1,03358813268386	0,00125507289105542
-0,0152774980755862	-0,014300959979686	1,03358654119653	1,59148732703507E-6
-0,0152774980766715	-0,014300959977905	1,03358654119396	2,56950016819246E-12

Barres: amb Tecla d'accés ràpid alt+B:

on: 1 Formulari d'introducció de dades.

2 Barra de funcions.

3 Informació/Ajuda.

4 Taula de dades.

5 Resultat del càlcul iteratiu.

Formulari d'introducció de dades: Aquí s'introdueixen les dades de cada barra definides de la següent manera:

Bus: és l'enumeració de la barra de la qual introduïm les dades.

Tipus de Barra: Com el seu nom indica aquí es defineix el tipus de barra que es: Bus de referència(Slack), Bus de tensió controlada(PV o PQV) o bus de carrega(PQ)

V: Tensió en la barra de la qual introduïm dades.

Angle: Angle de decalatge en la barra de la qual introduïm dades.

P: Potència activa injectada en la barra de la qual introduïm dades.

Q: Potència reactiva injectada en la barra de la qual introduïm dades.

Informació/Ajuda: Mostra informació o ajuda referent al concepte en que estàs treballant.

Barra de funcions: Et permet moure't i modificar les dades de les barres.

Nou Bus: Crea una nova barra en blanc que posteriorment es pot editar.

Acceptar: Accepta les característiques de la barra i les traspassa a la taula de dades.

Editar: Et permet modificar una barra ja existent.

Cancel·lar: Anul·la les modificacions fetes a les barres.

Eliminar: Elimina un bus ja existent.

Taula de dades

A la taula de dades es mostren totes les dades introduïdes anteriorment pel *Formulari d'introducció de dades*.

Aquest formulari et permet modificar les dades (recomanat per usuaris avançats) així com ordenar els elements segons la columna en la qual es faci doble click.

Resultat del calcul iteratiu

Es mostra el resultat del calcul iteratiu. Es mostra el K(nombre d'iteració), la solució de les incògnites dels elements i , per ultim, l'error màxim en K.

1.4.4.2 Elements de transmissió

Elements de transmissió: amb tecla d'accés ràpid alt+t:

The screenshot shows the 'DaDaSystem-ejemplo2' application window. The main area is titled 'Elements de transmissió' and contains several components:

- 1** Formulari d'introducció de dades: A form with a dropdown menu and input fields for 'Conexions'.
- 2** Barra de funcions: A toolbar with buttons for 'Els Elements', 'Editar', 'Acceptar', 'Cancel·lar', and 'Esborrar'.
- 3** Informació/Ajuda: A text box with the message 'Els valors a introduir han d'estar en p.u.'.
- 4** Taula de dades: A table with the following data:

Tipus	Connexió A	Connexió B	RI	XI	Gt	Bt	R. Transformació	Actiu
Linea	1	2	0,028	0,096				S
Linea	1	3	0,01	0,03				
Linea	2	3	0,02	0,06				
- 5** Resultat del calcul iteratiu: A table with the following data:

	a2	a3	V3	Error
1	-0,015204226628486	-0,0145879723410034	1,03484320557491	0,034843205574913
2	-0,015277034528548	-0,0143011661415709	1,03358813268386	0,00125507289105542
3	-0,0152774980755862	-0,0143009559979686	1,03358654119653	1,59148732703507E-6
4	-0,0152774980766715	-0,0143009559977905	1,03358654119396	2,56950016819246E-12

on: 1 Formulari d'introducció de dades.

2 Barra de funcions.

3 Informació/Ajuda.

4 Taula de dades.

5 Resultat del calcul iteratiu.

Formulari d'introducció de dades: Aquí s'introdueixen les dades de cada barra definides de la següent manera:

Tipus: Es selecciona el tipus d'element que es: Transformador, línia

de transmissió o bateria de condensadors.

Línia: S'introdueixen les dades de Z_t i Y_l .

Transformador: S'introdueixen les dades de Z_{cc} i el factor de regulació i si aquest està connectat o no

B. condensadors: S'introdueixen les dades Y_t i si aquesta bateria de condensadors està connectada o no.

Connexions: És la connexió entre quins busos està connectat l'element.

Informació/Ajuda: Mostra informació o ajuda referent al concepte en que estàs treballant.

Barra de funcions: Et permet moure't i modificar les dades de les barres.

Nou Bus: Crea una nova barra en blanc que posteriorment es pot editar.

Anterior: Retrocedeix una casella de la taula de barres.

Següent: Avança una casella de la taula de barres.

Acceptar: Accepta les característiques de la barra i les traspassa a la taula de dades.

Editar: Et permet modificar una barra ja existent.

Cancel·lar: Anul·la les modificacions fetes a les barres.

Eliminar: Elimina un bus ja existent.

Taula de dades

A la taula de dades es mostren totes les dades introduïdes anteriorment pel *Formulari d'introducció de dades*. Aquest formulari et permet modificar les dades (recomanat per usuaris avançats) així com ordenar els elements

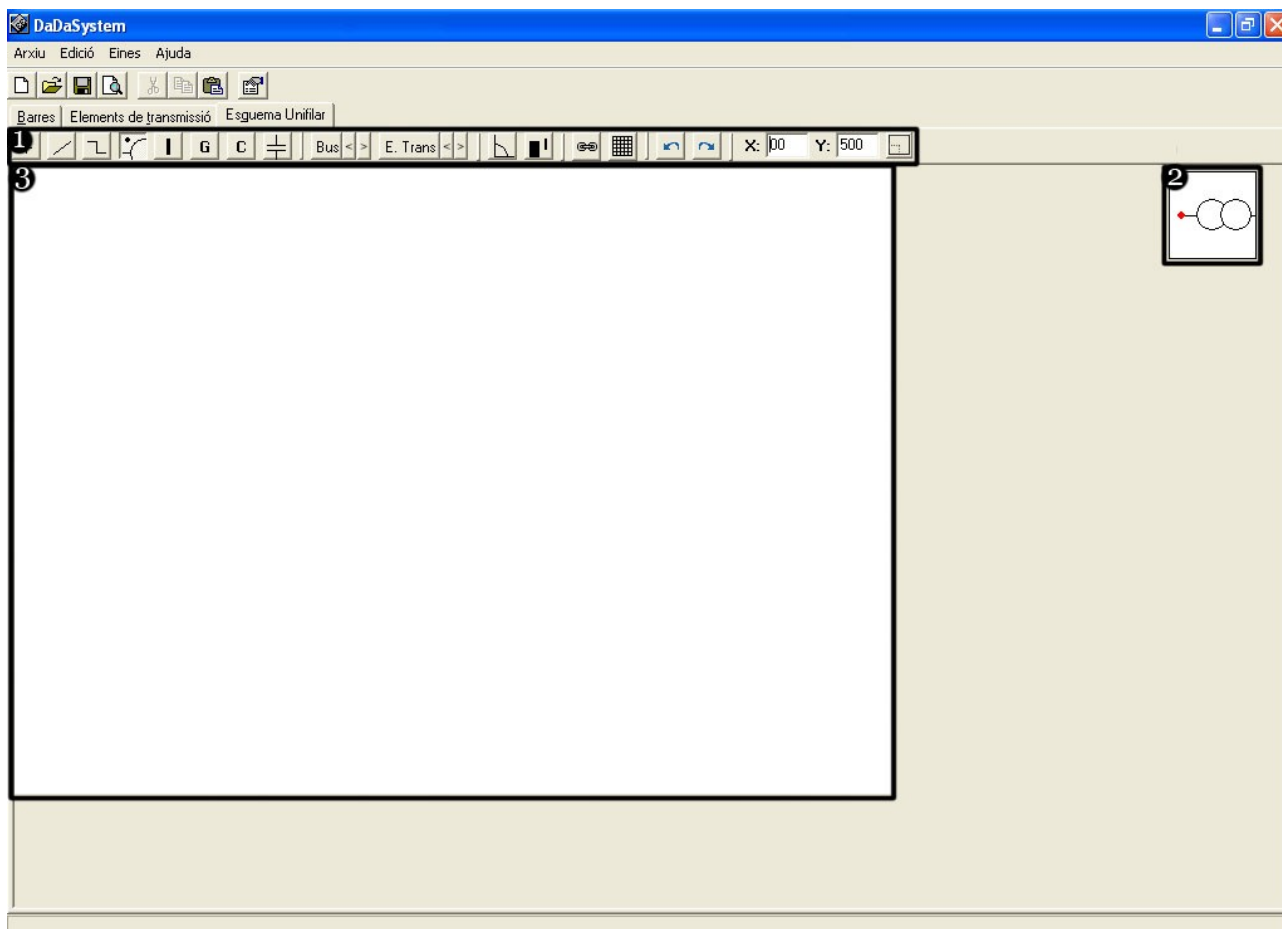
segons la columna en la qual es faci doble click.

Resultat del calcul iteratiu

Es mostra el resultat del calcul iteratiu. Es mostra el K(nombre d'iteració), la solució de les incògnites dels elements i , per ultim, l'error màxim en K.

1.4.4.3 Esquema unifilar

Esquema Unifilar: amb tecla d'accés ràpid alt+Q:



on: 1 Barra de funcions.

2 Imatge en miniatura.

3 Espai de representació.

Barra de funcions: Disposes del següents botons de funció. Aquest botons et permeten seleccionar el tipus d'element a afegir al gràfic. La funció d'aquests depèn de la seva activació:



Cursor: aquesta funció et permet desplaçar-te per l'interfície sense fer cap modificació en ella.



Línia discontinua: en el primer punt que es marqui, serà el punt de inici, el següent serà el punt final. Si es pressiona control a l'hora de seleccionar el punt final, et fera una línia de 0° o 90° dependent de quin sigui el punt més pròxim



Línia continua: el primer punt que es marqui serà el punt inicial però la línia serà continua fins que es seleccioni un altre element. Si es pressiona control a l'hora de seleccionar el punt final, et fera una línia de 0° o 90° dependent de quin sigui el punt més pròxim.



Transformador: es la representació del transformador, on el punt marca el costat de alta tensió. El punt vermell de la imatge en miniatura marca el punt de referència a l'hora de dibuixar-ho.



Barra: es la representació d'una barra. El punt vermell de la imatge en miniatura marca el punt de referència a l'hora de dibuixar-ho.



Generador: es la representació d'un generador. El punt vermell de la imatge en miniatura marca el punt de referència a l'hora de dibuixar-ho.



Carrega: es la representació d'una càrrega. El punt vermell de la imatge en miniatura marca el punt de referència a l'hora de dibuixar-ho.



Bateria de condensadors: es la representació d'una bateria de

condensadors. El punt vermell de la imatge en miniatura marca el punt de referència a l'hora de dibuixar-ho.



Llegenda de Barres: es la implementació de la llegenda de la barra en qüestió, les fletxes esquerra i dreta permeten desplaçar-se per tots els busos del sistema.



Llegenda d'elements de transmissió: es la implementació de la llegenda del bus en qüestió, les fletxes esquerra i dreta permeten desplaçar-se per tots els busos del sistema.



Rotació: Mou la representació 90° cada cop, te les opcions de 0, 90, 180 o 270 graus. Es pot veure l'inclinació de l'element a la dreta de la representació sempre que *Miniatura* estigui actiu.



Miniatura: es la representació en miniatura de l'element que estam treballant. Es pot activar o desactivar.



Pas enrere: es mou un pas enrere de la representació eliminant l'últim canvi realitzat.



Pas endavant: es mou un pas endavant, sempre i quan no ens trobem en l'últim pas de la representació.



Quadricula: Quadricula, com el mateix nom indica, tot l'espai de dibuix. D'aquesta forma es fa més fàcil posicionar els objectes i estructura l'esquema unifilar.



Forçat: Si aquest botó esta clickat, obliga que els punt de referència siguin els de la quadricula, per tant es mes fàcil mantenir diferents línies o objectes definits en un mateix punt concret.

Imatge en miniatura: Es veu la miniatura de el tipus d'element en que s'està treballant. El punt vermell indica el punt de referència a l'hora d'implantar-ho.

1.5 Utilització del programa DaDaSystem 1.0

Tot i poder-se utilitzar diferents vies per la resolució del problema del flux de carregues per mitjà de DaDaSystem 1.0 es recomana utilitzar l'algoritme de resolució següent:

- 1)Definir i numerar totes les barres començant des de *Bus 1*(slack) fins a *Bus N*.
- 2)Definir els elements de connexió i entre quines barres estan connectats, com també definir les bateries de condensadors i en quina barra estan connectades.
- 3)Tenir constància de l'estat de connexió del les bateries de condensadors o transformadors.
- 4)Introduir les dades del les barres en la pestanya *Barres*.
- 5)Introduir les dades dels elements de transmissió i bateria de condensadors en la pestanya *Elements de transmissió*.
- 6)Realitzar el calcul iteratiu des de l'acció *Barra d'eines/Començar calcul iteratiu*
- 7)Realitzar, si cal, la representació de l'esquema unificar en la pestanya *Esquema unifilar*.

1.5.1 Introducció de dades

1.5.1.1 Introducció de barres

1.5.1.1.1 Conceptes bàsics.

Per introduir les dades de les barres s'ha de tenir en compte els següents

conceptes:

- 1) La numeració de les barres ha de ser consecutiva i han de començar en el nombre 1. La barra 1 ha de ser sempre el bus Slack.
- 2) L'ordre d'introducció de barres no es important, aquestes s'ordenaran automàticament de 1 a N.
- 3) Totes les barres han de tenir definit el tipus de barra que és.
- 4) Cada tipus de barra ha de tenir totes les dades referents al propi tipus de barra. Si aquesta dada no s'incorpora, serà considerada 0.
- 5) Les dades que no corresponguin al tipus de barra concret no es tendran en compte pels càlculs.

1.5.1.1.2. Tipus de barres

Els tipus de barres que es poden introduir al programa poden ser: Slack Bus, Bus de tensió controlada i Bus de carrega

- 1) Slack Bus: Es coneixen, i mantenen constants, la tensió en modul i argument. Cada sistema només pot disposar de un slack bus.
- 2) Bus de tensió controlada: Hi ha dos tipus de busos de tensió controlada: PQV i PV.
 - PQV: Coneixem la tensió en modul però no l'argument però coneixem tant la potència activa com la reactiva.
 - PV: Coneixem la tensió en modul però no l'argument però coneixem tant la potència activa.
- 3) Bus de carrega: Tan sols coneixem la potència activa i reactiva. Desconeixem la tensió tant en modul com en càrrega.

1.5.1.2 Introducció dels elements de transmissió

1.5.1.2.1. Conceptes bàsics.

Per introduir les dades dels elements de transmissió s'ha de tenir en compte els

següents conceptes:

- 1) Les connexions dels elements ha de fer referència a l'enumeració de les barres introduïdes en *Barres*.
- 2) L'ordre d'introducció dels elements de transmissió no es important, aquestes s'ordenaran segons les connexions.
- 3) Els elements de connexió han de connectar totes les barres, no hi poden haver barres sense connectar.
- 4) Tots els elements de transmissió han d'estar definits amb el tipus d'element que és.
- 5) Les bateries de condensadors ha de tenir el primer punt de connexió en la barra en que estan connectades.
- 6) Cada tipus d'element de transmissió ha de tenir totes les dades referents al propi tipus d'element. Si aquesta dada no s'incorpora, serà considerada 0.
- 7) Les dades que no corresponguin al tipus d'element de transmissió concret no es tendran en compte pels càlculs.
- 8) Revisar si, a l'hora de fer els càlculs, les bateries de condensadors i transformadors amb regulació estan actius o no.

1.5.1.2.2. Tipus de d'elements de transmissió

Els tipus d'elements de transmissió que es poden introduir al programa poden ser: Línies, transformadors o bateria de condensadors.

- 1) Línies: coneixem com a mínim la resistència longitudinal però també

podem conèixer la resistència transversal tot i que sol ser pròxima a 0.

2)Transformadors: del transformador coneixem la seva resistència equivalent Z_{cc} a més de la seva toma de regulació. Si es un transformador sense toma de regulació aquesta serà calculada pel programa.

3)Bateria de condensadors: connectada a un sol bus i de la que coneixem la seva reactància capacitiva.

1.5.2 Representació de l'esquema unifilar.

Per la representació de l'esquema unifilar cal tenir clar el disseny que es te pensat i introduir un a un tots els elements a fi de realitzar l'esquema concret.

Es recomana començar d'esquerra a dreta i optimitzant l'espai al màxim.

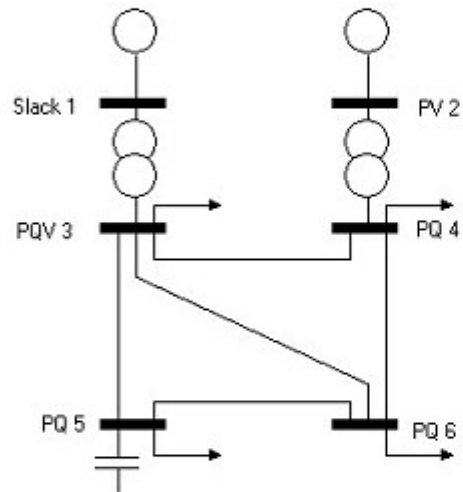
La representació de l'esquema unifilar es independent del calcul iteratiu però si les dades de les barres i els elements de transmissió no estan introduïdes no es podran aplicar les llegendes.

1.5.3 Resolució del problema i exportació de dades.

En primer lloc es pot consultar i el marge d'error que tenim es l'apropiat, de no ser així es pot modificar en *Eines/Propietats* .Un cop fer això i per començar amb la resolució i calcul del flux de carregues es necessari polsar el botó *Començar el calcul iteratiu* de la *Barra d'eines* un cop fet això es començara el calcul iteratiu i el resultat de les iteracions es podrà veure en la taula del *Resultat iteratiu* on a la fila inicial es pot veure la variable respecta la qual s'està donant la solució i la columna inicial ens diu el nombre de iteració des de 0 fins a que l'error sigui inferior a l'error màxim definit.

4.4. ANNEX 4: EXEMPLES RESULTS

Esquema unifilar



Dades inicials

Barres

No	Tipus	Tensió (v)	Fase (rad)	Potencia activa (P)	Potencia reactiva (Q)
1	Slack	1,03	0		
2	PV	1,02		1,2	
3	PQV	1,01		-0,7	-0,3
4	PQ			-0,5	-0,2
5	PQ			-0,6	-0,3
6	PQ			-1,5	-0,4

Elements de transmissió

Element	Conexions	R(long)	X(long)	G(trans)	B(trans)	R. Transmissió
TrafoV	1-3	0	0,101	0	0	
TrafoC	2-4	0	0,12	0	0	1
Linia	3-4	0,02	0,06	0	0,105	
Linia	3-5	0,01	0,035	0	0,08	
Linia	3-6	0,032	0,11	0	0,15	
Linia	4-6	0,015	0,04	0	0,09500001	
Linia	5-6	0,025	0,075	0	0,15	
B. Condensad	5-0	0	0	0	0,25	

Flux de càrregues

Element	Conexions	P(ca-cb)	P(cb-ca)	Q(ca-cb)	Q(cb-ca)
TrfV	1-3	2,12643402922363	0,844651048763346	-2,12643402922363	-0,382472173133478
Trfc	2-4	1,2	-1,2	0,24360274397415	-0,0706682229970183
Llnla	3-4	0,173119775785971	-0,172378154428484	0,0350784939716569	-0,139049972807278
Llnla	3-5	0,834551134720021	-0,827684667479914	0,0222255700513603	-0,0789932032834379
Llnla	3-6	0,418763118717638	-0,412937792444963	0,0251681091104602	-0,154650022923292
Llnla	4-6	0,872378154428483	-0,860943900928007	0,00971819580428802	-0,073087201325581
Llnla	5-6	0,227684667479914	-0,22611830662703	0,0289698818612729	-0,172262775751121

Solució Iterativa

Barres

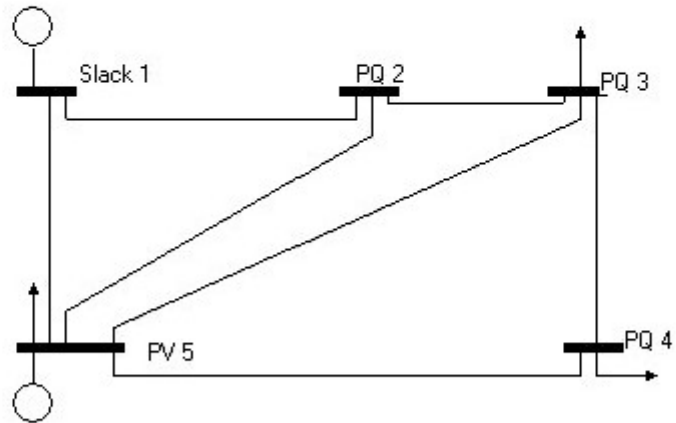
No	Tipus	Tensió (v)	Fase (rad)	Potència activa (P)	Potència reactiva (Q)
1	Slack	1,03	0	8,50573611689451	3,37860419505339
2	PV	1,02	-0,0671997842470111	1,2	0,974410975896601
3	PQV	1,01	-0,200140350917185	-0,7	-0,3
4	PQ	1,0013428403716	-0,208658235370599	-0,5	-0,2
5	PQ	0,999953356067831	-0,22844148232789	-0,6	-0,3
6	PQ	0,98656963177231	-0,243117053592686	-1,5	-0,4

Marge d'error 1E-6. Error obtingut 1,33526523171668E-12 en la 5 iteració

Elements de transmissió

Element	Conexions	R(long)	X(long)	G(trans)	B(trans)	R. Transmissió
TrafoV	2-4	0	0,101	0	0	0,96297879797462
TrafoC	3-5	0	0,12	0	0	1
Linia	4-5	0,02	0,06	0	0,105	
Linia	4-6	0,01	0,035	0	0,08	
Linia	4-7	0,032	0,11	0	0,15	
Linia	5-7	0,015	0,04	0	0,09500001	
Linia	6-7	0,025	0,075	0	0,15	
B. Condensad	6-1	0	0	0	0,25	

Esquema unifilar



Dades inicials

Barres

No	Tipus	Tensió (v)	Fase (rad)	Potencia activa (P)	Potencia reactiva (Q)
1	Slack	1,05	0		
2	PQ			-0,96	-0,62
3	PQ			-0,35	-0,14
4	PQ			-0,16	-0,08
5	PV	1,02		-0,24	

Elements de transmissió

Element	Conexions	R(long)	X(long)	G(trans)	B(trans)	R. Transmissió
Linia	4-5	0,1	0,5	0	0,075	
Linia	1-5	0,05	0,25	0	0,02	
Linia	1-2	0,02	0,1	0	0,03	
Linia	3-4	0,05	0,25	0	0,02	
Linia	2-3	0,04	0,2	0	0,025	
Linia	2-5	0,05	0,25	0	0,02	
Linia	3-5	0,08	0,4	0	0,01	

Flux de càrregues

Element	Conexions	P(ca-cb)	P(cb-ca)	Q(ca-cb)	Q(cb-ca)
Línia	4-5	-0,125120132711364	0,12720894628832	-0,0994030848031434	0,0353462887909986
Línia	1-5	0,517686829150482	-0,505407222790846	0,0415646107049581	-0,00159557890677682
Línia	1-2	1,24788548369529	-1,21338177918997	0,570657978898111	-0,428973267539736
Línia	3-4	0,0349881995397706	-0,0348798872886365	-0,0376738559031765	0,0194030848031432
Línia	2-3	0,254946655451387	-0,252218746501597	-0,0111423282200183	0,00117802784978823
Línia	2-5	-0,00156487626141401	0,00308744248793711	-0,179884404240244	0,167562361260706
Línia	3-5	-0,132769453038173	0,135112834014588	-0,103504171946613	0,105342975837094

Solució Iterativa

Barres

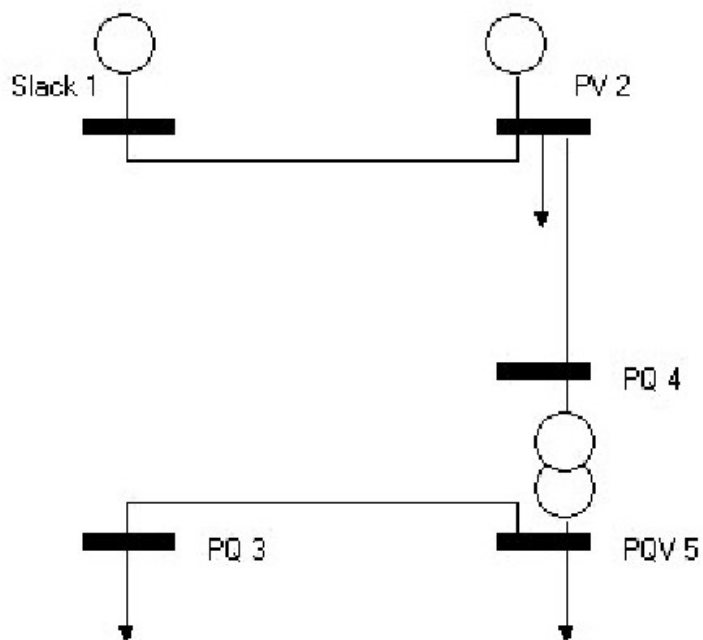
No	Tipus	Tensió (v)	Fase (rad)	Potència activa (P)	Potència reactiva (Q)
1	Slack	1,05	0	1,76557231284578	0,612222589603069
2	PQ	0,976261958295689	-0,110504137300019	-0,96	-0,62
3	PQ	0,96706783542776	-0,164505443147096	-0,35	-0,14
4	PQ	0,972637156010105	-0,175310529614792	-0,16	-0,08
5	PV	1,02	-0,118665085324758	-0,24	0,196656046982022

Marge d'error 1E-6. Error obtingut 2,50621659536598E-9 en la 4 iteració

Elements de transmissió

Element	Conexions	R(long)	X(long)	G(trans)	B(trans)	R. Transmissió
Línia	5-6	0,1	0,5	0	0,075	
Línia	2-6	0,05	0,25	0	0,02	
Línia	2-3	0,02	0,1	0	0,03	
Línia	4-5	0,05	0,25	0	0,02	
Línia	3-4	0,04	0,2	0	0,025	
Línia	3-6	0,05	0,25	0	0,02	
Línia	4-6	0,08	0,4	0	0,01	

Esquema unifilar



Dades inicials

Barres

No	Tipus	Tensió (v)	Fase (rad)	Potencia activa (P)	Potencia reactiva (Q)
1	Slack	1	0		
2	PV	1		1,5	
4	PQ			0	0
3	PQ			-0,5	-0,1
5	PQV	1		1	-0,5
3	PV	1		1	

Elements de transmissió

Element	Conexions	R(long)	X(long)	G(trans)	B(trans)	R. Transmissió
Línia	2-4	0,001	0,005	0	0,1	
Línia	1-2	0,001	0,005	0	0,1	
Línia	4-3	0,001	0,005	0	0,1	
TrafoV	5-4	0	0,1	0	0	

Flux de càrregues

Element	Conexions	P(ca-cb)	P(cb-ca)	Q(ca-cb)	Q(cb-ca)
Línia	2-4	-0,499393312019733	0,499997511110864	0,545655446580726	-0,6423873481918
Línia	1-2	-1,9952443463237	1,99939331201973	0,35983618007444	-0,439091351594267
Línia	4-3	2,48888913949873E-6	-3,37306250002324E-15	-0,0995182353435752	-9,65813533189167E-15
TrfV	5-4	0,999999999999911	-0,499999999998563	-0,999999999999911	0,64190558353462

Solució Iterativa

Barres

No	Tipus	Tensió (v)	Fase (rad)	Potencia activa (P)	Potencia reactiva (Q)
1	Slack	1	0	-3,99048869264739	0,719672360148881
2	PV	1	0,0103862446451129	1,5	0,213128189972919
3	PQ	0,997775352697544	0,0134365295174158	0	0
4	PQ	0,997525910106901	0,0134865420204999	-0,5	-0,1
5	PQV	1	0,120502815883686	1	-0,5

Marge d'error 1E-6. Error obtingut 3,11097338112631E-7 en la 4 iteració

Elements de transmissió

Element	Conexions	R(long)	X(long)	G(trans)	B(trans)	R. Transmissió
Línia	3-5	0,001	0,005	0	0,1	
Línia	2-3	0,001	0,005	0	0,1	
Línia	5-4	0,001	0,005	0	0,1	
TrafoV	6-5	0	0,1	0	0	1,06547860996358

5. BIBLIOGRAFÍA

CHARTE OJEDA, F. (2003) *Delphi 7* . España, ANAYA.

CORIA CISNEROS, L. (2006) *Modelado y operacion de lineas de transmision*. México .ITM.

GÓMEZ EXPÓSITO, A. (2002)*Análisis y operación de sistemas de energía eléctrica*. Madrid. McGrawHill

MILLA LOSTAUMAU, L. (2000) *Modelo de ecuaciones de flujos de potencia real y reactiva en sistemas de energia electrica*. Perú .UNMSM.

OLIVARES, D. (2000) *Sistemas Eléctricos de potencia*. Perú .UNMSM.

VV. AA. (2005) *Simulación de sistemas eléctricos*. Madrid . Pearson Prentice Hall.

<http://www.scribd.com/doc/3272851/Iniciacion-a-Delphi>

<http://www.turboexplorer.com/>