Appendix A

The load flow Algorithm in NEPLAN program:

The load flow problem is non-linear, because at every network node i, an apparent power S_i is given:

$$S_i = P_i + jQ_i = V_i * \overset{*}{I}_i \quad (* = conjugate \quad complex)$$

The power Application software (PAS) is driven by off line and on line data, for study purposes, all functions of PAS can also be operated off line.

Given:

a) off line data:

- Elements with impedances (lines/cables, transformer, generators) with limits of operation, these 2 terminal element are represented by their branch four-poles and the individual branch admittance matrices.
- Elements without impedances (bus-bar, circuit breakers isolators, CTs, PTs, etc) with limit of operation.
- Static network: topology (Node-Branch representation, static connection of elements without and with impedances, regardless of the positions of the circuit breaker and isolators).
- Generator and load data: attached to bus-bar node type: PQ, PV, Slack.

b) On Line: data:

- Actual network: topology (position of the circuit breakers and isolators, closed, open running, under-fined etc).
- Actual measuring values: at bus-bars for load flow additionally branch values for state estimation to be calculated:

a) All node voltages, voltage. Vector, vector of state variables define the actual steady state of the system.

b) Results derived from the state vector:

- Branch currents and powers, overload indication.
- Node voltages with indication of limit violation.
- Balances of active and reactive power, active and reactive losses.

Application of the load flow Algorithm:

a) Off line for study purposes:

Function in NEPLAN 2000

b) On-line in PAS (power application software):

- SSE state estimation.
- Optimal load flow.
- CAN contingency analysis etc.

Solution methods:

Overview: Gauss Seidel, current iteration, Newton-Raphson

Causs- Seided:

Historical, no inversion of Y matrix needed:

a) Set state vector V to + j0 pu (flat start).

b) For each node i DO:

V i is Calculated, the other Vj remain constant joint solution of:

$$I_i = Y_{ii} * V_i + \sum Y_{ij} * Vi_j$$
$$and$$
$$Ii = (S_i / V_i)$$

Special treatment of PV nodes. Slak node:

Vi is not changed

c) The calculation is stopped, if the hange of state vector is below a specified accuracy limit:

$$|\Delta V_i| \leq \varepsilon$$

Much iteration is required, convergence of the iteration process is questionable, but storage requirement are small.

Current Iteration method:

Basic: Factorized, Y – matrix

a) Set state vector $\underline{V}$ to $\underline{1}$ + J0 pu (flat start).

b) Repeat until convergence is reached.

b1) for each node i Do:

$$I_i = (\underline{S}_i / \underline{V}_i)$$

Special treatment of PV node. Slack in node: Vi is not changed.

b2) Network solution:

Solution of V = Z*I forward substitution

Result: New state vector Vnew

b3) Convergence check: the check: The calculation is stopped, if the change of the state vector is below a specified a currency limit.

$$\Delta |V_i| = |V_{new}| - |V_i| \le \varepsilon$$

b4) if convergence not yet reached it

$$V_i = V_{new}$$

b5) END REPEAT.

Just a few iterations are required typically 4…7, even for large networks, convergence is in general reached, if there exists also a physical solution the problem and the static stability limit are not violated.

Newton Raphson:

Prioiple: minimization of a mismatch function:

Search for the point, defined by the state vector V, where the mismatch function zero.

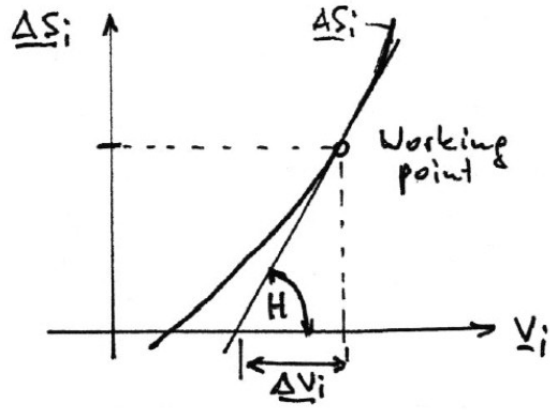
The mismatch function in our case is the difference between the actual and the given nodal apparent powers:

$$\Delta S_i = S_i \ actual - S_i \ given$$
$$S_i \ actual = V_i^{*} * I_i^{*}$$

Because the Ii depend also from Vi, the first term in the above equation is nonlinear in Vi. All deviations ΔS_i are caused by deviations in the state variables, ΔV_i and we can make a formal linearization a round the working point:

$$\Delta S_i = \partial(V_i * I_i) / \partial(V_c) * \Delta V_i \quad [*(A)]$$

This can be illustrated by fig below:



Because the (partial) derivative is not defined for complex numbers all values S = P + jQ, V, I , Z , Y must be broken down intro real numbers either is Cartesian or in polar coordinates.

In this case the matrix describing the partial derivatives in *(A) is a real matrix with the dimension 2N* 2N and is called it or Jacobian (Jacobi's matrix).

The vectors of power and voltage differences have the dimension 2N each.

According to *(A) the change of the state vector ΔV_i can be calculated by multiplying the inverse of H, H^{-1} with the given power mismatches Si.

Discretion of iteration:

a) Set state vector V to 1 + j0 pu (flat start).

b) Repeat until convergence is reached.

b1) for each node i Do:

I_i by using

$S_{iactual} = Vi * I_i$ and S_i actual = S_{igiven}

ΔS_i by using

$\Delta S_i = S_{iactual} = S_{igiven}$

Special treatment of PV nodes slack node:

V_i is not changed

b2) Network solution:

Setup of H and factorization:

for each iteration step or:

set up of H and factorization once at the beginning (constant slope).

Solution of *(A) by for word and backward substitution result new state vecor

$V_{new} = V_{old} + \Delta V$

b3) convergence check: The calculation is stopped if the change of the state vector is below a specified accuracy limit.

b4) if convergence not yet reached:

Set $\underline{V}_i = \underline{V}_{new}$

b) END REPEAT:

Just a few iterations are required (typically 4…..7, even for large network).

Convergence is in general reached, if there exists also a physical solution to the problem and the static stability limits are not violated.

In order to minimize the calculation burden several treatment of H are in use:

- Decoupled load flow: only the dependence of P on the voltage angles and of Q on the voltage magnitudes is considered.
- H calculated factorized once, only after a certain number of iterations or at each iteration.
- etc. etc.

APPENDICES B

NEPLAN DATA

LOAD DATA:

Name	P/MW	Q/MVAR
Atbara (NEC) 220kv	68	42.140
Aroma	4.25	2.630
Banat 110	46.750	28.970
Eldebba 220	12.750	7.900
Dongle 220	32.2	21.070
Eid Babker 110	42.500	26.340
Elbager 110	21.250	13.750
Elfao 110	4.250	2.630
Elgriba 220	11.333	2.578
Elobeid 220	21.250	13.170
Faroug 110	42.500	20.340
Free zone	59.500	36.870
Elgederef 110	17.659	8.713
Gamuoia	51.500	30.000
Giad 110	21.590	20.00
Hag Abdallah 110	10.500	5.270
Hassa Heisa	42.500	28.340
Hawata 220	12.500	5.270
Izergab 110	85.000	52.680
Jebel Aulia 110	58.244	21.038
Khartoum East 110	127.500	68.570
Kassala 220	12.108	5.270
Khartoum North 110	127.501	79.02
Kilo3 66	6.293	5.669
Kilox 110	42.500	19.340
Kuku 110	59.500	36.870
Local market 110	101.000	53.750
Mahadia 220	58.500	38.460
Mahadia 110	43.000	26.340
Managil	21.250	17.662
Mashkur	12.750	7.900
Maringan 110	51.000	35.610
Merowe town	6.500	5.270
Mina sharif 110	19.496	10.540
New Halfa	14.450	6.960
Mugran 110	103.000	60.020
Omdurman 110	76.500	47.410

Name	P/MW	Q/MVAR
Port Sudan	10.000	11.070
Rabak 110	17.250	10.540
Rank 220	4.250	2.630
Rawashda	5.530	3.930
Sennar 110	25.504	10.578
Shagara 110	71.911	52.680
Shendi 220	18.000	13.700
Singa 220kv	9.874	6.862
Tandulti 220	12.750	7.850
Umrawaba 220	8.500	5.270