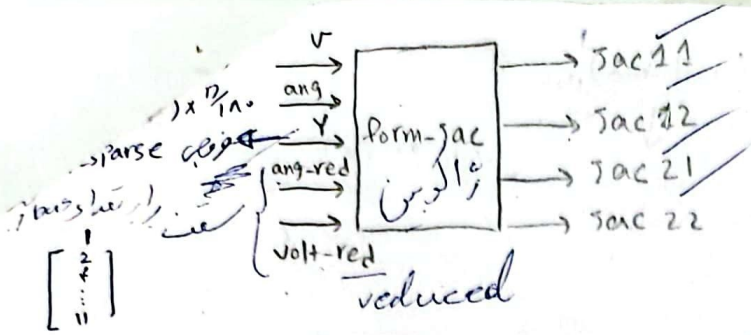


$\times \frac{\pi}{180}$

sparse

$\begin{bmatrix} 1 \\ 2 \\ 4 \\ \vdots \\ 11 \end{bmatrix}$

bus_type = round(bus(:,

نوع باس 11x1

می‌تواند نباشد

PQV_no = find(bus_type >= 2)

10x1

شماره باس ها

اندازه = 10 → ii = $\begin{bmatrix} 1 \\ \vdots \\ 10 \end{bmatrix}$

ترتیبی از ۱ تا ۱۰

یک عدد مرجع که ۱۱ کم شود

⇒ sparse ang_red = 10x11 تعداد اندازه

(1,1) = 1
(2,2) = 1
(3,4) = 1
(4,5) = 1
⋮
(10,11) = 1

1 = sparse_ang_red(i,j)

اگر نبود صفر

ii[K] pav[

PQ_no = find(bus_type == 3)

7x1

$\begin{bmatrix} 5 \\ 6 \\ 7 \\ \vdots \\ 11 \end{bmatrix}$ → اندازه = 7 ii = $\begin{bmatrix} 1 \\ \vdots \\ 7 \end{bmatrix}$

⇒ volt_red = spars 7x11

(1,5) = 1
(2,6) = 1
(3,7) = 1
⋮
(7,11) = 1

1 = volt_red_sparse(i,j)

اگر نبود صفر

ii[K] pq[K]

---

exponential

exp_ang = $e^{(jay * ang)}$

11x1

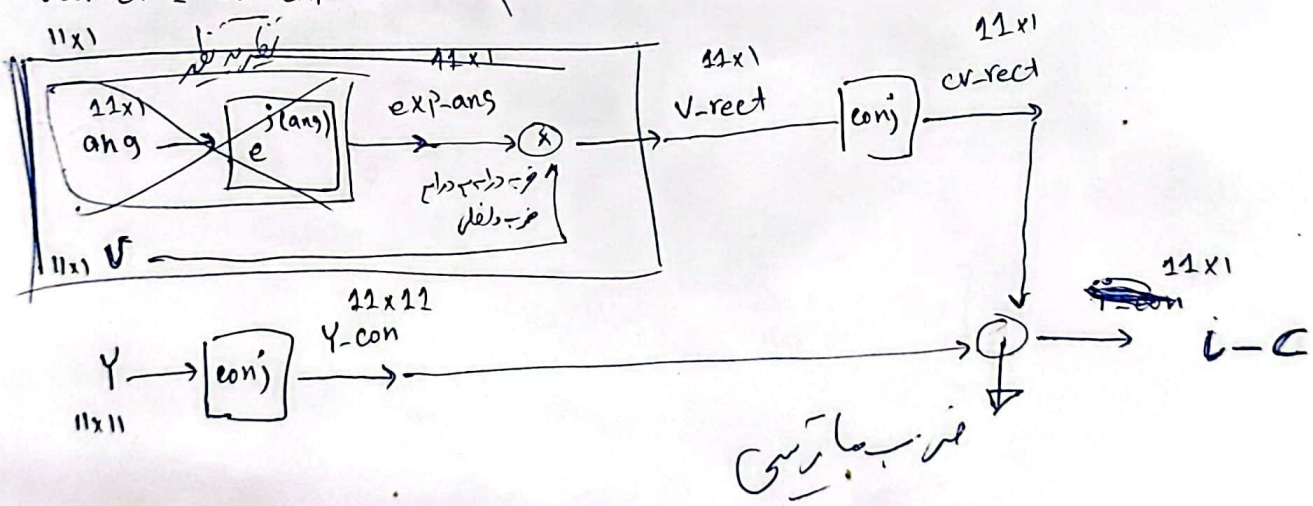
$\sqrt{-1}\,\theta$

نباید θ باشد

V_rect = V. exp_ang

11x1

نظیر به نظیر



11x1 V_rect

ضرب داخل نظیر به نظیر

11x1 i_c

قطر اصلی

11x1 S اسپارس

11x11 تبدیل به ماتریس قطری

قطر اصلی

اسپارس Vdia → قطری 11x11 از V_rect → conj → 11x11 cvdia

Vmag → قطری 11x11 از V

form-sac ضمیمه‌ی تابع ← تقسیم

subject: date:

(اصلی)

قطری $S \rightarrow 11 \times 11$

اسپارس $S1 \rightarrow$ 〃 $\Rightarrow (S+S1) \rightarrow 11 \times 11$ اسپارس

$V_{mag} \rightarrow 11 \times 11$ قطری (اصلی)

$(S+S1)/V_{mag} \xrightarrow{?} 11 \times 11$

تمام درایه‌های ستون j ام از ماتریس (S+S1) به درایه‌ی (j,j) ام ماتریس V_{mag} تقسیم خواهد شد. (j از 1 تا 11)

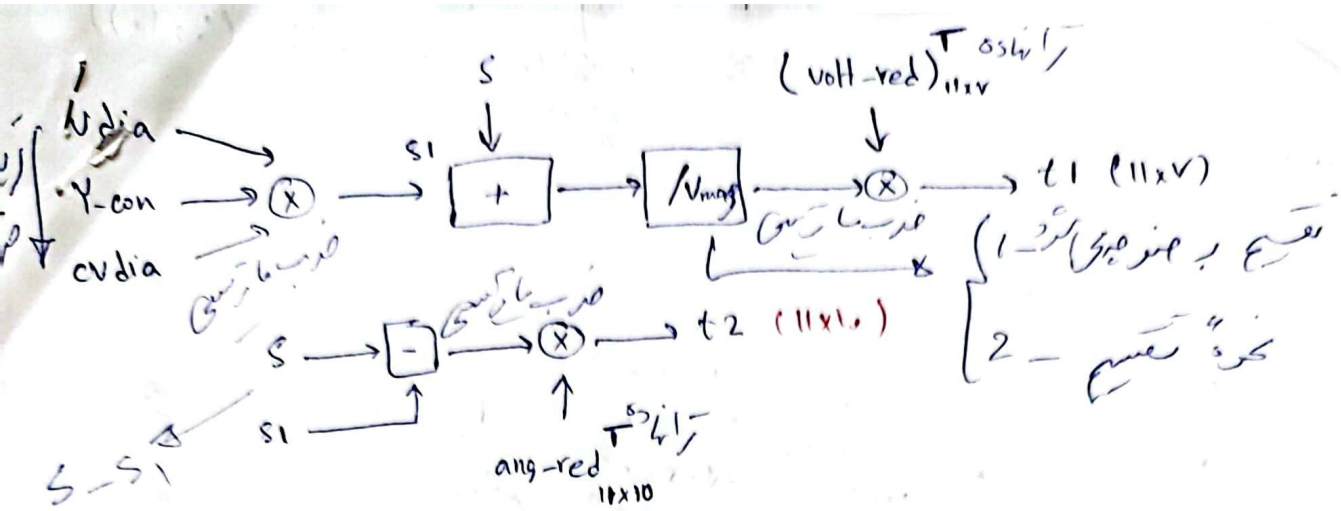
ماتریس حاصل $(i,j) = \dfrac{[S+S1](i,j)}{V_{mag}(j,j)}$ $i,j = 1,\dots,11$

نمی‌تواند صفر باشد

کار بر حق شماره صفر وارد کنید

اگر خودمان حساب کنیم قطعاً صفر نیست

Arman



Locals =>

10x10 S_{ac}^{11} : ang-red * imag(t2)

10xV S_{ac}^{12} : ang_red * real(t1)

Vx10 S_{ac}^{21} : volt_red x real(t2)

VxV S_{ac}^{22} : volt_red x imag(t1)

صفحه 10

صفحه 1

ورودی [bus] ⟵ شماره باس‌ها — ستون اول ماتریس — n×15

bus ستون دوم = V n×1

bus ستون سوم × $\frac{\pi}{180}$ = ang n×1

4 ″ = Pg n×1

5 ″ = Qg n×1

6 ″ = Pl n×1

7 ″ = Ql n×1

تابع CalC:

CalC ← bustype n×1 ، Ybus n×n ، tol = 10^{-9} (مثلاً عدد مثبت کوچک برای خطا)

CalC → ΔP ✓ ، ΔQ ✓ ، P ✓ ، Q ✓ ، ConV_flag ✓ (0 or 1)

بخش کدهای لازم برای initialization

محل قرارگیری عدد حقیقی (برخی از ستون اول) bus

bustype = [2; 2; 1; 2; 3; 3] → n=#1

1 swing — مرجع

2 PV — ولتاژ ثابت

3 PQ — بار

bustype ← ماتریس Ybus n×n ، شماره bus (سطر اول) ، ردیف (سطر دیگر دارد)

محاسبه

sw_bno = [1 1 0 ... 1] 1×n

sw_bno: محل قرارگیری باس‌هایی که نمی‌توانند مرجع باشند

g_bno = [0 0 1 0 0 1 1 ...] 1×n

g_bno: محل قرارگیری باس‌هایی که نمی‌توانند PV باشند

باس‌ها:

① جاهایی که bustype مساوی 1 باشد صفر.

② جاهایی که bustype مساوی 2 باشد صفر.

باس type از نوع بار ماتریس سطر و ستون مثل سوئینگ و PV ندارد.

V_rect (n×1) = [V (n×1) نقطه‌به‌نقطه ضرب در e^(j[θ]) (n×1)] θ = ang

Cur_inj (n×1) = Y (n×n) . V_rect (n×1)

ضرب خارجی

S (n×1) = [V_rect (n×1)] ⊗ [Cur_inj (n×1)]* (Conjugate)

ضرب نقطه‌به‌نقطه

نتیجه‌ی کل ⟵ عدد مختلط

P (n×1) = real{S} (قسمت حقیقی) ، Q (n×1) = imag{S} (قسمت موهومی)

ΔP (n×1) = (Pg − Pl − P) (n×1) × (sw_bno)

ΔQ (n×1) = (Qg − Ql − Q) (n×1) × (sw_bno) (n×1) × (g_bno) (n×1)

Pmis (1×1) = max(|ΔP|) ، Qmis (1×1) = max(|ΔQ|) ⟵ بزرگترین عدد مطلق یا بیشترین افت

mism = pmis + ...

if mism > tol ⟹ ConV_flag = 1 (همگرا نشده)

تابع chq-lim :

$q_g\text{-max}$ = ستون ۱۱ ام bus nx1 → $\boxed{\text{chq-lim}}$ → f

$q_g\text{-min}$ = ۱۲ ام " bus nx1 →

خروجی = f = { 1 اگر حد Q یکی از ژنراتورها به limit پایین یا بالای خود رسیده باشد → محاسبات لازم در ادامه دیده می‌شوند

0 نرسیده باشد " " " " " → عملیاتی انجام نمی‌شود

اگر به حد خود رسیده باشد ← { bus-type آن نیروگاه از 2 به 3 تغییر داده می‌شود ← PQ ← PV

$Q_g \neq 0$ تولیدی آن برابر صفر شده $Q_l = -Q_{min}^{max}$ قرار داده می‌شود

نیروگاه

صفحه ۲

محاسبات: فقط روی باس‌های ژنراتور PV ← هرجا bus-type = 2 است فقط چک شود!

نقطه شروع: lim-flag = 0

منظور کل تابع chq-lim است

اگر $Q_g[\text{bus type} == 2] > q_g^{max}[\text{bus-type} == 2] \Rightarrow q_g\text{-max-indx} = [1 \ldots 1 \ldots]$ هرجا به حد رسید

عملاً همان شرط توضیح داده شده در اول کار است.

" $< q_g^{min}$ " $\Rightarrow q_g\text{-min-indx} = [-1 \ -1]$

در calc توضیح داده شد

آپدیت کردن g-bno

اگر به حد max رسید ($q_g\text{-max-indx}$ عنصر 1 دارد) ← $Q_g(\text{indx همان}) = 0$

$Q_l(\ ''\ ) = \boxed{Q_l - q_g^{max}(\text{همان indx})}$

آپدیت ← bus-type(همان) = 3 ← volt-ved, ang-ved

lim-flag = 1 ← scalar

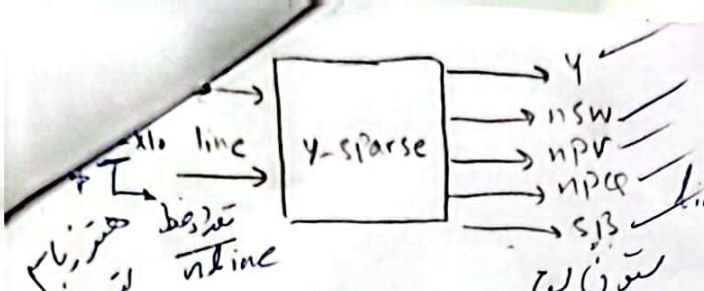
" " min رسید ($[1 \text{ obj } 1] = q_g\text{-min-ind}$) ← $Q_g(\ ) = 0$

$Q_l(\ ) = \boxed{Q_l(\ ) - q_g^{min}(\ )}$

آپدیت g-bno ← bus-type( ) = 3

lim-flag = 1

خروجی f = lim-flag



هنوز باسم ستون آخر bus کاری نداریم

تعداد خط nline

nline , nbus , ibus : [1 ⋮ 17]

12 , 11

اندازه سطر باس — اندازه سطر line

مشابه iline ساخته می‌شود

ستون سوم

متغیرها :

r = line(:,3) , 12x1 مقاومت

rx = line(:,4) , 12x1 راکتانس

chrg = قطری 12x12 اسپارس — jay x 1/2 x line(:,5) — روی قطر اصلی می‌چینیم

$\sqrt{-1}$ مختلط

z = r + jay * rx 12x1

y = قطری 12x12 اسپارس , $y_{11} = \frac{1}{z_{11}}$ ($y_{ii} = \frac{1}{z_{ii}}$) روی قطر اصلی می‌چینیم

حالت خاص (مثال) — حالت کلی

صفحه A

* از خطوط ۴۷ تا ۵۰ ⇐ برای شماره‌گذاری باس‌ها به ترتیبی که در ماتریس bus وارد شده است و bus-int را می‌سازد

شماره باس ۵ را ترتیبی از ۱ تا n می‌چینیم

متغیرهای دیگر :

12x1

from-bus = round(line(:,1)) — می‌توان قید گذاشت که صحیح وارد شوند

from-int = bus-int(from-bus) 12x1

to-bus = round(line(:,2)) 12x1

to-int = bus-int(to-bus)

مبدأ خط

مقصد خط

۶۵ تا ۶۷ ⇒ tap = [1 همه] 12x1 — به جنبه درایه‌ای که متناظر با آن در ماتریس line، ستون ۶ ام = 1/عدد

1 < 12x1 — شرط

1 , 1/ak

درایه‌های tap دو مقدار دارند

ak = line col 6

Phase-shift = line(:,7) 12x1

tap = tap x $e^{-j \cdot \text{phase-shift} \times \pi/180}$ (ضرب درایه به درایه) — به روز رسانی tap

12x1

متغیر iline = [1 : 1 , nline]^T ترانهاده — از ۱ تا nline به گام ۱

12x1 — شروع — گام — انتها

C_from = (from-int , iline) = tap — C_from(i,j) = tap(K)

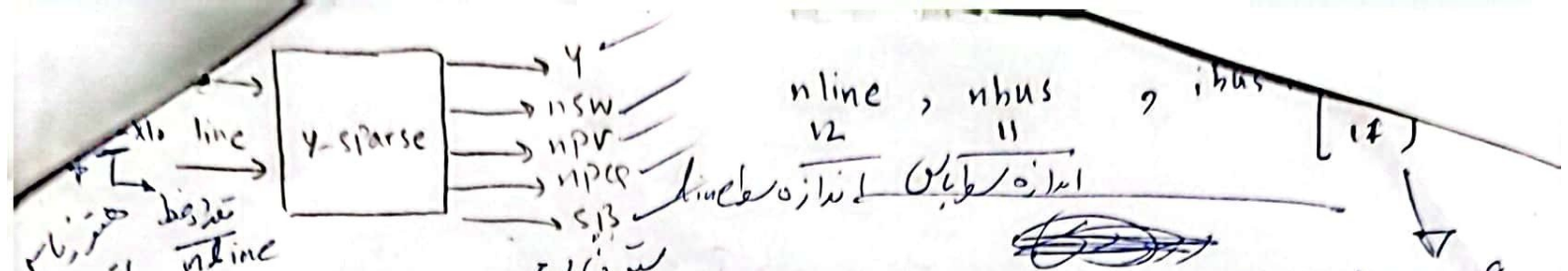
nbus x nline (11x12) اسپارس

from-int[K] iline[K] — ۱۲ تا ...

C_to = (to-int , iline) = 1 — مشابه بالا فقط ۱ را درون C_to قرار می‌دهد (به جای tap)

nbus x nline 11x12

11x12 c_line = C_from - C_to



ها به nline اضافه می‌شود

متغیرها:

$r = line(:,3)$, 12×1 مقاومت

$rx = line(:,4)$, 12×1 راکتانس

$chrg =$ ماتریس قطری 12×12 $\quad jay \times \frac{1}{2} \times line(:,5)$ ← روی قطر اصلی می‌چینه

$z = r + jay * rx$ 12×1

$y =$ ماتریس قطری 12×12 , $y_{11} = \frac{1}{z_{11}}$ $\left(y_{ii} = \frac{1}{z_{ii}}\right)$ روی قطر اصلی می‌چینه

حالت خاص (مثال) — حالت کلی

صفحه A

* از خطوط 45 تا 50 ← برای شماره‌گذاری باس‌ها ترتیبی که در ماتریس bus وارد شده است bus_int را می‌سازد.

متغیرهای دیگر:

from_bus = round(line(:,1)) 12×1 ← شماره باس‌ها را ترتیبی از آن می‌چیند

from_int = bus_int(from_bus) 12×1 ← مبدا فعل

to_bus = round(line(:,2)) 12×1 ← معدوم فعل

to_int = bus_int(to_bus)

می‌توان نوشت که صحیح را برگرداند

$[\text{شماره باس} \times 3]$

$[\text{شماره باس} \times 5]$

$tap = [1_{\text{فعل}}]_{12\times1}$ ← برای خطوطی که ضریب تاپ‌شان در ماتریس line ستون ۹ام = ۱ است

tap‌های دو سطر دارند

$ak = line\ col\ 6$

$\begin{cases} 1 & \text{شرط} \\ \frac{1}{ak} \end{cases}$

phase_shift = line(:,7) 12×1

$tap = tap \times e^{-j \cdot phase\_shift \times \pi/180}$ ← به روزرسانی tap (فرمول دارد)

$iline = [1:1:nline]^T$ 12×1 ← از ۱ به تعداد خطوط nline

$C\_from = \left(\begin{matrix}from\\-int\end{matrix}, iline\right) = tap$ ← $C\_from(i,j) = tap(K)$

nbus × nline (11×12) ماتریس

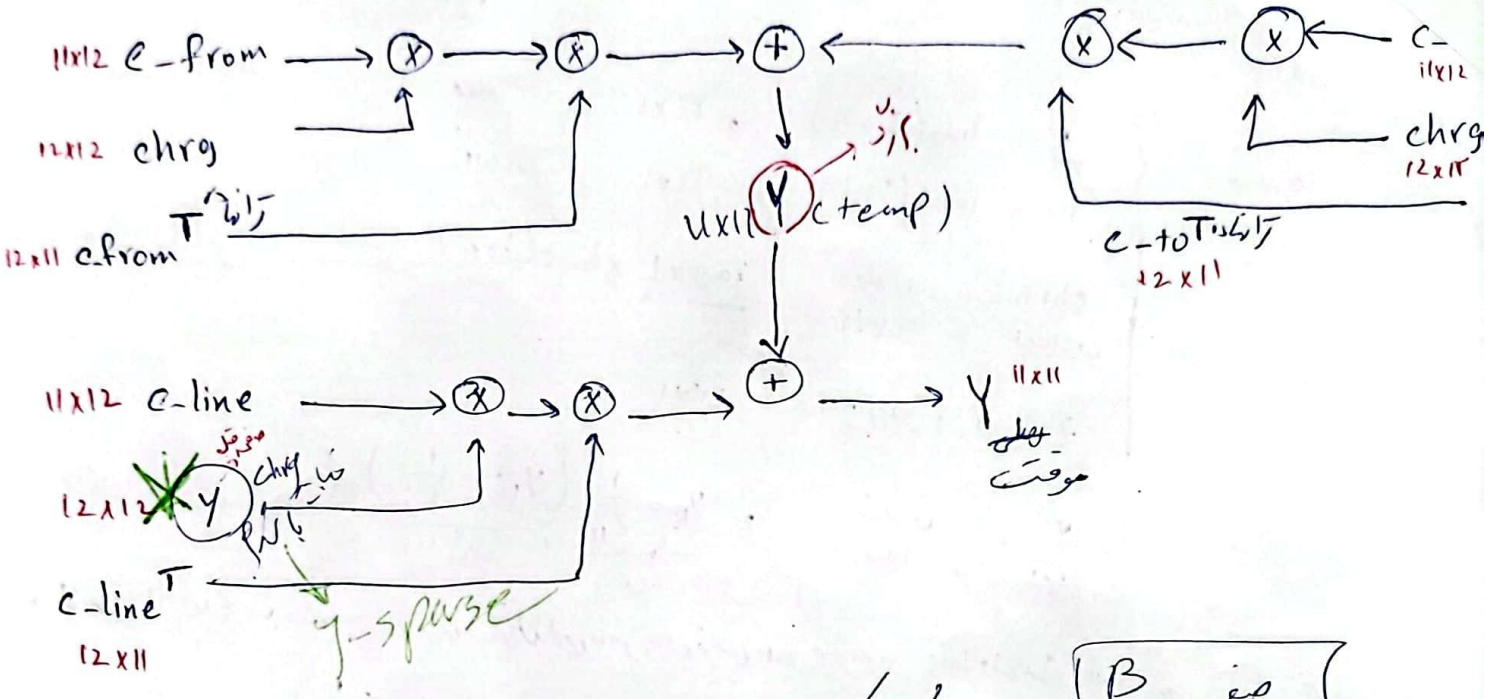
$from\_int[K]$ $iline[K]$

$C\_to = \left(to\_int, iline\right) = 1$ ← شبیه بالا فقط از درون C_to آرایه‌ی دوم تاپ‌ها را

nbus × nline (11×12) ماتریس

$e\_line = C\_from - C\_to$ 11×12

محاسبات : Y



صفحه B

در مرحله آخر :

11x1
Gb = bus(:, 8) کندالکتانس
Bb = bus(:, 9) سوسپتانس
11x1

$$Y = Y + \underbrace{\left((ibus, ibus) = Gb + jBb\right)}_{nbus \times nbus}$$

مختلط

K از 1 تا 11

مشابه صفحه قبل

11 x 11

می تواند نباشد

تعداد bus هایی که type برابر PQ است

فروجی های دیگر : bus_type = round(bus(:, 10))

load_index = find(bus_type == 3) → length → npQ

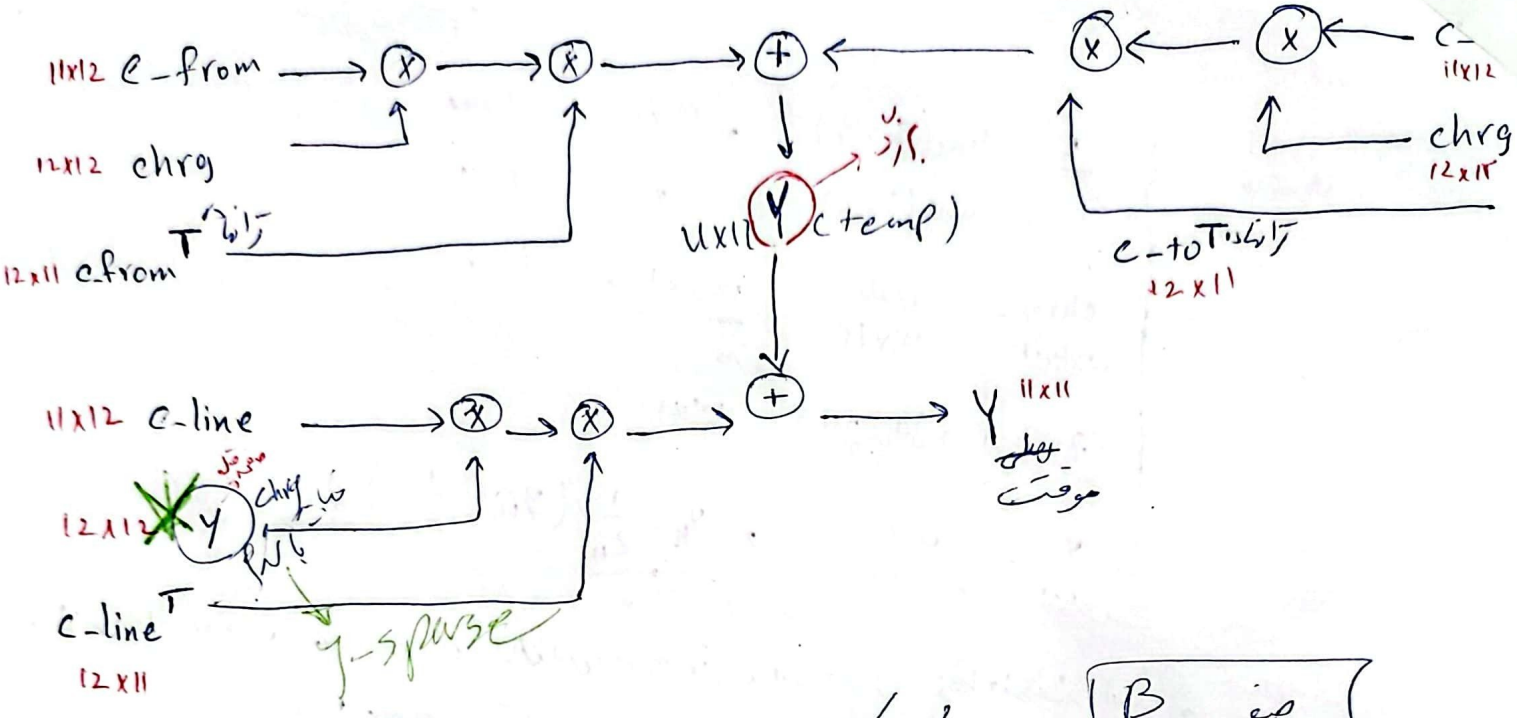
gen_index = " " " 2 → " → npV

SB = " " " 1 → " → nsw

~~Scalar~~ vector

Scalar
nsw =
npv =
npQ

محاسبات : Y



صفحه B

در مرحله آخر :

11x1
Gb = bus(:, 8) کندالکتانس
Bb = bus(:, 9) سوسپتانس
11x1

$$Y = Y + \underbrace{\left((ibus, ibus) = G_b + j B_b\right)}_{nbus \times nbus \quad 11 \times 11}$$

مختلط

K از آتا 11

مشابه صفحه قبل

تعداد bus هایی که type برابر PQ است

می تواند نباشد

فروجی های دیگر : bus_type = round(bus(:, 10))

~~Scalar~~ vector

load_index = find(bus_type == 3) → ~~length~~ → npQ
gen_index = " " " 2 → ~~[ ]~~ → npV
SB = " " " 1 → ~~[ ]~~ → nsw

Scalar
nsw =
npv =
npQ