

Cheatsheet Octave para Circuitos (Mallas/Nodos, Fasores) y Laplace (simbólico)

Compacto, directo y listo para copiar/pegar. Incluye plantillas “todo en uno” para tu circuito.

0) Setup rápido

```
format short g          % salida compacta
j = 1i;                 % unidad imaginaria
deg2rad = @(d) d*pi/180;
rad2deg = @(r) r*180/pi;
```

1) Complejos, fasores e impedancias

```
% Fasores básicos
Vs = 54*exp(j*deg2rad(30));          % 54∠30°
abs(Vs), rad2deg(angle(Vs));          % módulo y fase (°)
conj(Vs); real(Vs); imag(Vs);

% Impedancias a una frecuencia (ω fijo)
R = @(r) r;                          % Resistor
L = @(w,Lh) j*w*Lh;                  % ZL = jωL
C = @(w,Ch) 1./(j*w*Ch);             % ZC = 1/(jωC)
```

Impresión rápida en polar para cualquier vector:

```
to_polar = @(x) arrayfun(@(z) ...
    printf("%.6g ∠ %.2f°\n", abs(z), rad2deg(angle(z))), x);
```

2) Método de mallas (KVL) – Plantilla general

Armas la matriz de impedancias **Z** y el vector **V** (términos independientes), resolvés $\mathbf{I} = \mathbf{Z} \backslash \mathbf{V}$.

```
% Z*I = V
Z = [...];          % matriz de impedancias por malla
V = [...];          % fuentes equivalentes por malla (signo según KVL)
I = Z \ V;          % corrientes de malla
to_polar(I);        % imprimir en módulo ∠ fase°
```

Regla práctica para construir Z (copedancias):

- **Diagonal (Z_{ii}):** suma de **todas** las impedancias que toca la malla i.
 - **Fuera de diagonal (Z_{ij}):** **menos** la suma de impedancias **compartidas** entre i y j.
Memotecnia: “**diagonal suma, fuera resta común**”. Si no comparten, es 0.
-

3) Método de nodos (KCL) – Plantilla general

```
% Y*Vn = I_iny      (Y: conductancias nodales)
% Armá Y sumando 1/Z por ramas entre nodos. Vn: tensiones de nodos (referencia a tierra).

Y = [...];
I_iny = [...];       % corrientes inyectadas (convención elegida)
Vn = Y \ I_iny;
```

Truco útil si ya tenés impedancias en red: $Y = \text{inv}(Z_{\text{red}})$ (cuando aplica), pero lo robusto es **armar Y por KCL**.

4) Ramas, potencias y chequeos

```
% Corriente en una rama de impedancia Zb con caída Vb
I_b = Vb / Zb;

% Potencia compleja en una rama
S = Vb * conj(I_b); % S = P + jQ
P = real(S); Q = imag(S);

% Residuos KVL/Nodos (deben ser ~ 0)
res1 = ...; res2 = ...; res3 = ...;
printf("r1|=%.2e |r2|=%.2e |r3|=%.2e\n", abs(res1), abs(res2), abs(res3));
```

5) Gráficos rápidos (barrido de ω)

```
w = logspace(0,4,500); % 1 a 1e4 rad/s
Zc = 1./(j*w*10e-6); % C = 10 μF
plot(w, abs(Zc)); grid on;
xlabel('\omega [rad/s]'); ylabel('|Z_C| [\Omega]');
set(gca,'XScale','log','YScale','log');
```

6) Simbólico en Octave (Laplace, fracciones parciales)

Necesita el paquete **symbolic** (usa SymPy).

Instalar (una vez):

```
pkg install -forge symbolic
```

Cargar (cada sesión):

```
pkg load symbolic
```

6.1) Laplace / Inversa / Elementos en s

```
syms t s a w L C R real
f = exp(-a*t)*sin(w*t)*heaviside(t);
F = laplace(f, t, s); % F(s)
f2 = ilaplace(F, s, t); % inversa

% Elementos en s
ZL = s*L; % inductor
ZC = 1/(s*C); % capacitor
ZR = R; % resistor
```

6.2) Sistemas simbólicos (mallas/nodos en s)

```
syms Vs s R1 R2 L1 C1
Z11 = R1 + s*L1 + 1/(s*C1);
Z12 = -R1;
Z21 = -R1;
Z22 = R1 + R2 + 1/(s*C1);

Z = [Z11, Z12; Z21, Z22];
V = [Vs; 0];
```

```
I_sym = simplify(Z \ V);          % corrientes simbólicas
pretty(I_sym);
```

6.3) Fracciones parciales / residuos

```
% Simbólico
syms s
F = (s + 2)/((s + 1)*(s + 3));
F_part = partfrac(F, s);
pretty(F_part);

% Numérico (coeficientes)
b = [1 2];          % N(s) = s + 2
a = conv([1 1],[1 3]); % D(s) = (s+1)(s+3)
[r,p,k] = residue(b, a); % r: residuos, p: polos, k: parte polinómica
```

6.4) Evaluar simbólicos en valores numéricos

```
% Ejemplo: reemplazar parámetros y evaluar en jω
I_num = subs(I_sym, {R1,R2,L1,C1,Vs}, {8,16,9e-3,6e-3, 54*exp(j*pi/6)}); % ejemplo
I_num = subs(I_num, s, j*2*pi*60);      % ω = 2π·60 rad/s
double(I_num);
```

6.5) Más dígitos (“vpa” estilo)

```
digits(20);
vpa(F, 15);
```

7) Control (funciones de transferencia, Bode)

```
pkg load control
s = tf('s');
G = (s + 2)/((s + 1)*(s + 3));
bode(G); grid on;
step(G); impulse(G);
```

8) Plantilla completa AC por mallas (tu circuito)

Tres mallas horarias; la fuente Vs está en la rama izquierda; la rama central (-j6) es común.

```
format short g
j = 1i; deg2rad=@(d) d*pi/180; rad2deg=@(r) r*180/pi;

Vs = 54*exp(j*deg2rad(30)); % 54∠30°
to_polar=@(x) arrayfun(@(z) printf("%.5g ∠ %.2f°\n", abs(z), rad2deg(angle(z))), x);

% Matriz de impedancias por mallas (diagonal suma, fuera resta común)
Z = [ 8 - j,    -8 + 6*j,    6*j;
     -8 + 6*j,  24 - 6*j,    6*j;
     6*j,      6*j,        3*j ];

% Vector de fuentes (misma fuente aparece con signo opuesto en mallas 1 y 2)
V = [ Vs; -Vs; 0 ];

I = Z \ V;          % [I1; I2; I3]
disp('I (mallas) en polar:');
to_polar(I);

% Corrientes de ramas (convenciones típicas)
I_8   = I(1) - I(2);
I_j5  = I(1);
I_16  = I(2);
```

```

I_C   = I(1) - I(2) - I(3);    % por el capacitor (-j6), positiva hacia abajo
I_j9  = I(3);

% Tensiones de rama
V_8   = 8*I_8;
V_j5  = j*5*I_j5;
V_16  = 16*I_16;
V_C   = -j*6*I_C;
V_j9  = j*9*I_j9;

% Chequeo KVL (residuos ~ 0)
res1 = 8*(I(1)-I(2)) + j*5*I(1) + (-j*6)*(I(1)-I(2)-I(3)) - Vs;
res2 = 8*(I(2)-I(1)) + 16*I(2) + (-j*6)*(I(2)-I(1)-I(3)) + Vs;
res3 = (-j*6)*(I(3)-I(1)-I(2)) + j*9*I(3);
printf("Residuos: %.2e %.2e %.2e\n", abs(res1), abs(res2), abs(res3));

```

9) Plantilla s-domain simbólica (misma topología)

Reemplazá $j5 \rightarrow sL1$, $(-j6) \rightarrow 1/(sC)$, $j9 \rightarrow sL2$, etc.

```

pkg load symbolic
syms s L1 C1 L2 R8 R16 Vs

Z11 = R8 + s*L1 + 1/(s*C1);
Z22 = R8 + R16 + 1/(s*C1);
Z33 = 1/(s*C1) + s*L2;

Z12 = -(R8 + 1/(s*C1));
Z13 = -(1/(s*C1));
Z23 = -(1/(s*C1));

Z = [ Z11, Z12, Z13;
      Z12, Z22, Z23;
      Z13, Z23, Z33 ];

V = [ Vs; -Vs; 0 ];
I_sym = simplify(Z \ V);
pretty(I_sym);          % I1(s), I2(s), I3(s)

% Ejemplo: corriente por C
I_C_s = simplify(I_sym(1) - I_sym(2) - I_sym(3));
pretty(I_C_s);

```

Memotecnias express

- **Copedancias:** diagonal suma; fuera, **resta** lo compartido.
- **Rama común con fuente:** recorridos opuestos \Rightarrow **signos opuestos**.
- **Nodos:** suma de corrientes que salen (o entran) = 0, sé consistente.
- **Laplace:** $R \rightarrow R$, $L \rightarrow sL$, $C \rightarrow 1/(sC)$; para AC a ω : $s = j\omega$.
- **Potencias:** $S = V \cdot I^*$, primero sentidos, luego números.