

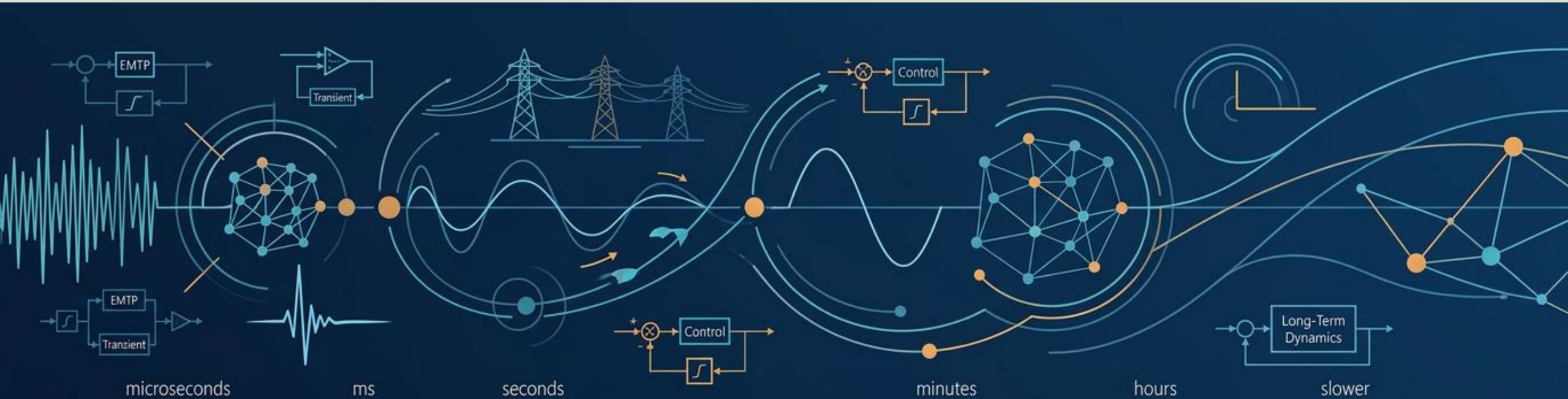


# Projektovanje pomoću računara u elektroenergetici

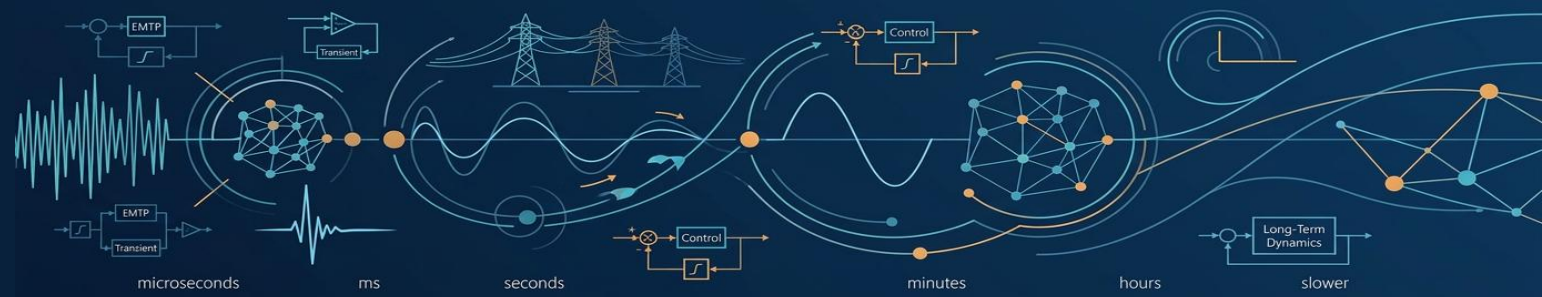
Prof. dr Goran Dobrić  
Prof. dr Mileta Žarković

# Steady-state analiza

## Nelinearne algebarske jednačine



# Ciljevi predavanja



## 1. Steady-State analiza distributivnih mreža

Razumeti **algebarske jednačine** tokova snaga (steady-state problem)

Uočiti **specifičnosti distributivnih mreža**: radijalna topologija i visok R/X odnos

## 2. Teorija i izvođenje jednačina

Izvesti **backward sweep** formule (proračun snaga od kraja ka izvoru)

Izvesti **forward sweep** formule (ažuriranje napona od izvora ka kraju)

Razumeti **iterativni proces** konvergencije

## 3. Pisanje algoritma i pseudokoda

Strukturirati problem kroz **pseudokod** sa jasnim koracima

Razumeti **algoritamsku logiku**: petlje, uslovi, konvergencija

## 4. Flowchart i vizuelizacija

Naučiti pravila za crtanje flowcharta (simboli, tok, grananja)

Nacrtati kompletan flowchart Shimohammadi metoda

Koristiti Draw.io alat za crtanje

## 5. Praktična realizacija u MATLAB-u

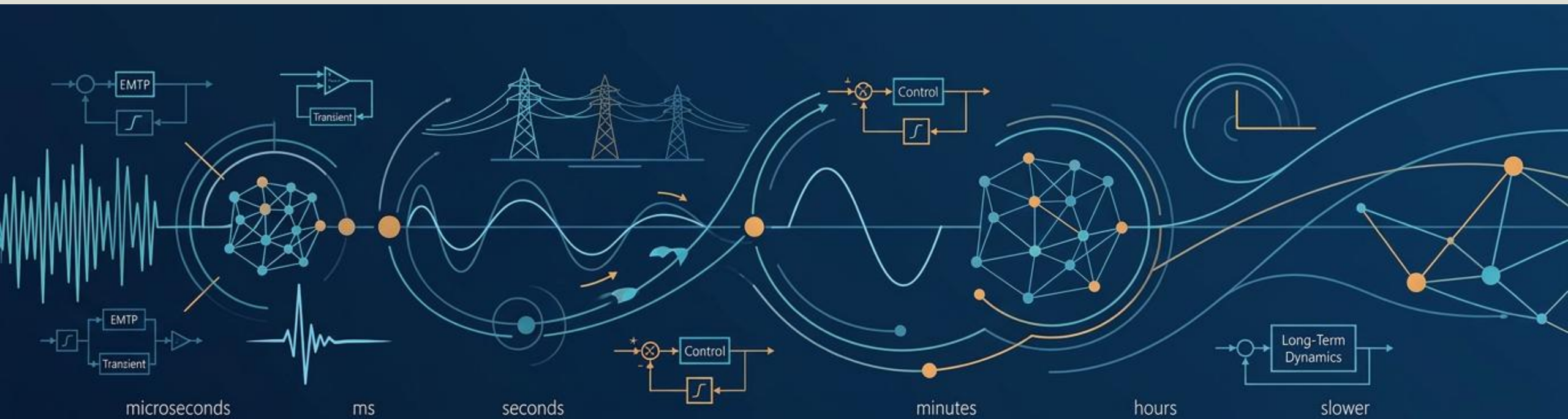
Implementirati metod korak-po-korak sa objašnjenjem

Testirati na radijalnom test sistemu

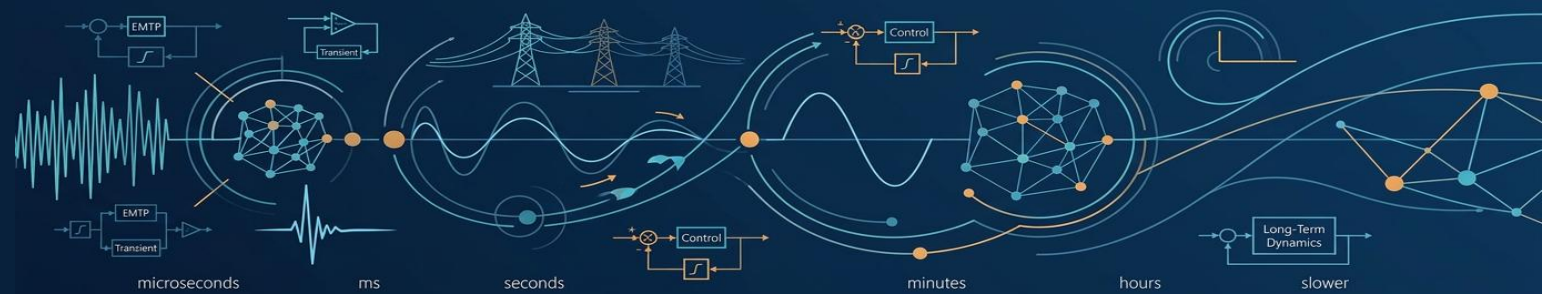
Analizirati konvergenciju i rezultate (naponi, gubici)

Vizualizovati naponski profil mreže

# 1. Steady-State analiza distributivnih mreža



# Uvod i motivacija



## Šta smo radili na prvom času?

Pregled vremenskih konstanti u elektroenergetici ( $\mu\text{s}$  do h)

Razlike između steady-state, RMS i EMT simulacija

Tipovi matematičkih jednačina: Algebarske, ODE, DAE

## Zašto su tokovi snaga važni?

Tokovi snaga (load flow/power flow) = osnovna analiza ustaljenog stanja mreže

Primer primene: planiranje, optimizacija, analiza napona i gubitaka

Razlika između prenosnih i distributivnih mreža

## Problem klasičnih metoda na distributivnim mrežama

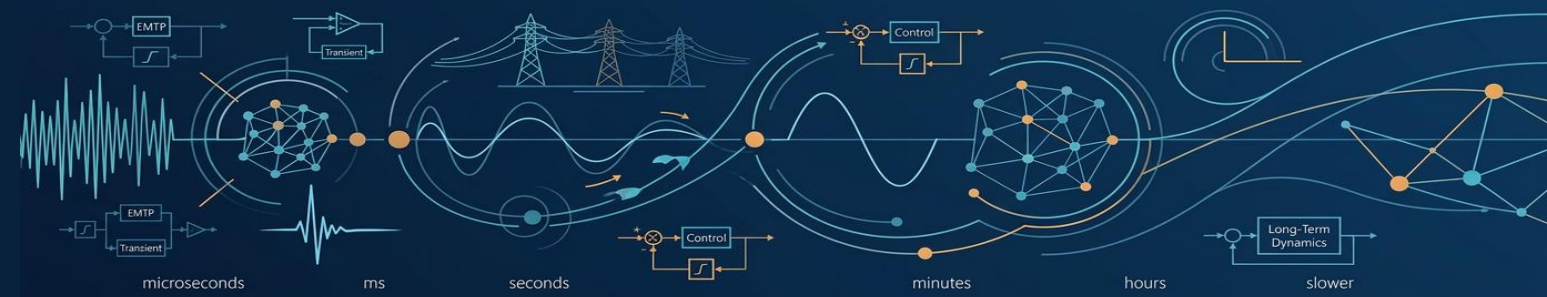
Newton-Raphson i Gauss-Seidel razvijeni za prenosne mreže

Distributivne mreže: **radijalna struktura** + **visok R/X odnos**

Konvergencija klasičnih metoda se **pogoršava** kod radijalnih sistema

Motivacija: potreban nam je specijalizovan metod!

# Karakteristike distributivnih mreža



## Radijalna topologija

**Definicija:** Jedan izvor (napajanje), granasta struktura bez petlji

**Vizualizacija:** stablo sa krenom (slack bus) i granama

**Prednost:** jednostavnija zaštita i kontrola

**Nedostatak za proračun:** klasični matrcini metodi nisu efikasni

## Visok R/X odnos

Prenosna mreža:  $R/X \approx 0.1-0.3$  (induktivna)

Distributivna mreža:  $R/X \approx 1-10$  (rezistivna)

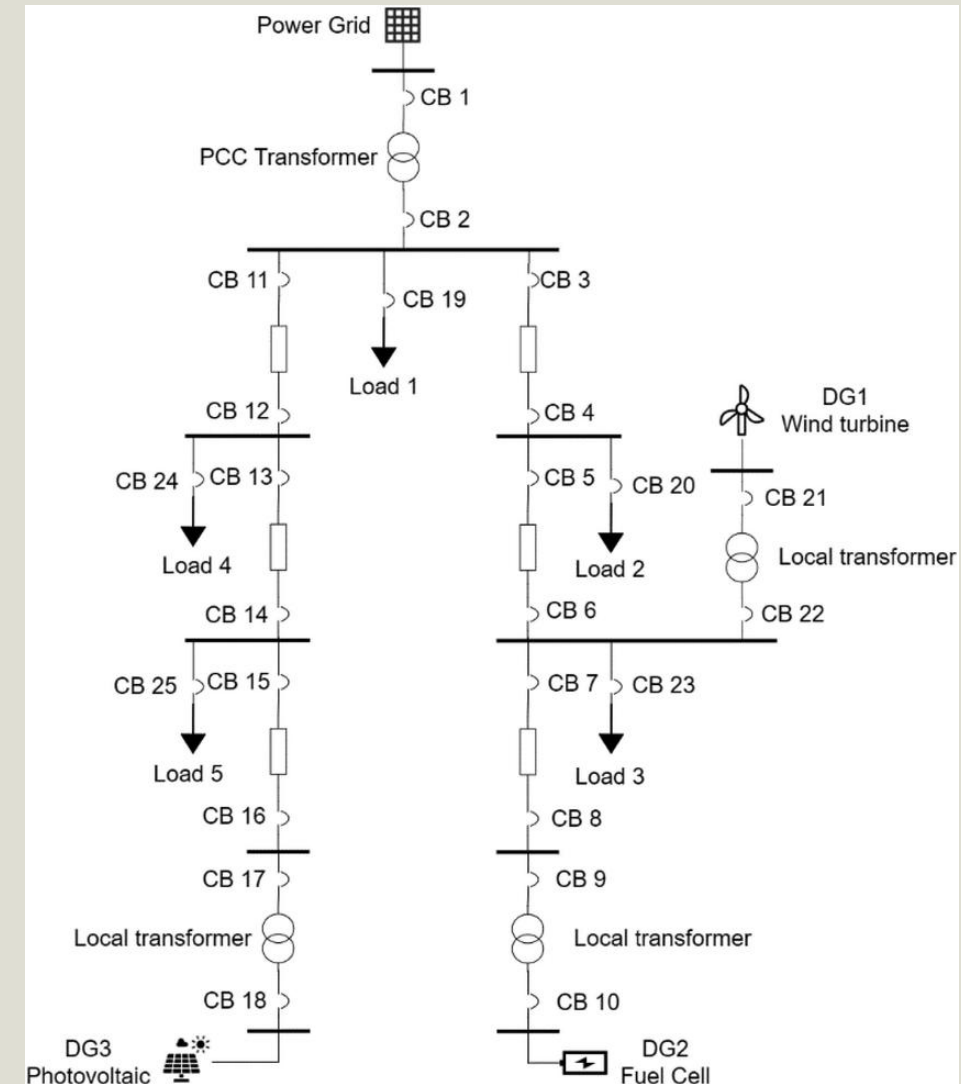
**Posledica:** jaka sprežnost između P-V i Q- $\delta$

## Karakteristike opterećenja

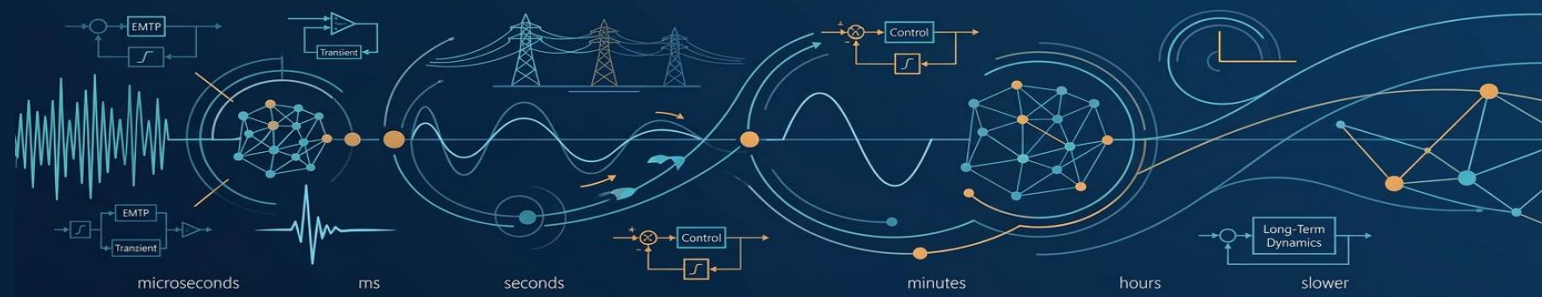
Raspodeljeno opterećenje duž vodova

Voltage-dependent load models (konstantna impedansa, struja, snaga)

Nebalansirane trofazne mreže (u realnosti)



# Karakteristike distributivnih mreža



## Radijalna topologija

**Definicija:** Jedan izvor (napajanje), granasta struktura bez petlji

Vizualizacija: stablo sa krenom (slack bus) i granama

Prednost: jednostavnija zaštita i kontrola

Nedostatak za proračun: klasični matrcini metodi nisu efikasni

## Visok R/X odnos

Prenosna mreža:  $R/X \approx 0.1-0.3$  (induktivna)

Distributivna mreža:  $R/X \approx 1-10$  (rezistivna)

Posledica: jaka sprežnost između P-V i Q- $\delta$

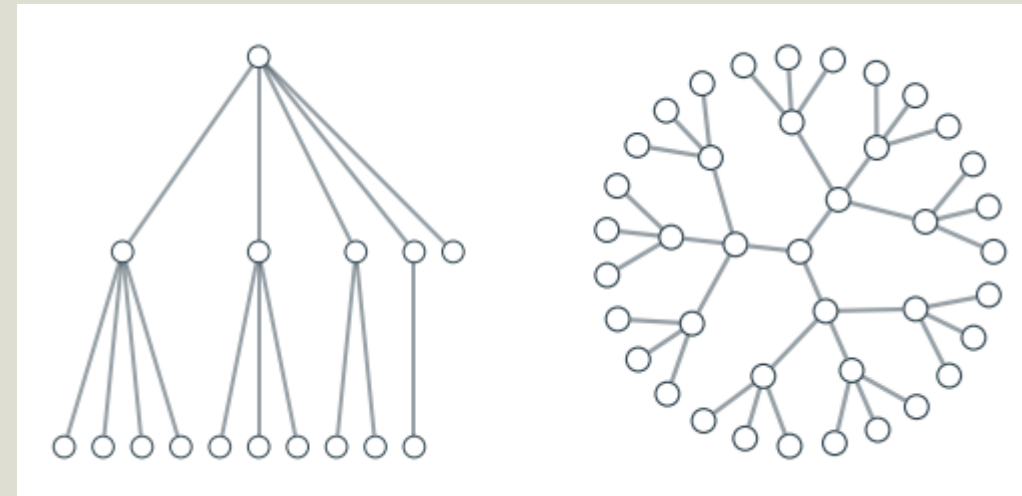
Zašto Newton-Raphson ima problema: Jacobian matrica loše uslovljena

## Karakteristike opterećenja

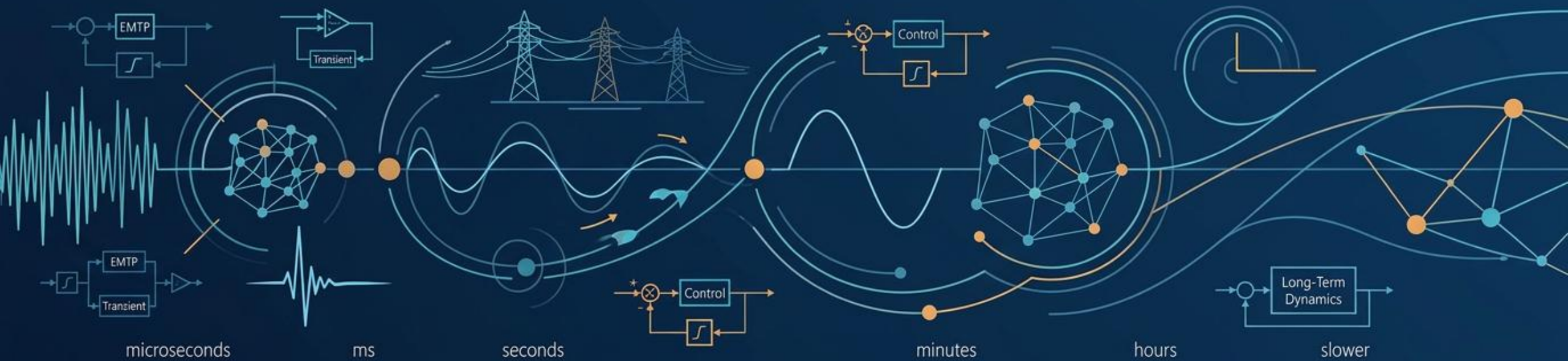
Raspodeljeno opterećenje duž vodova

Voltage-dependent load models (konstantna impedansa, struja, snaga)

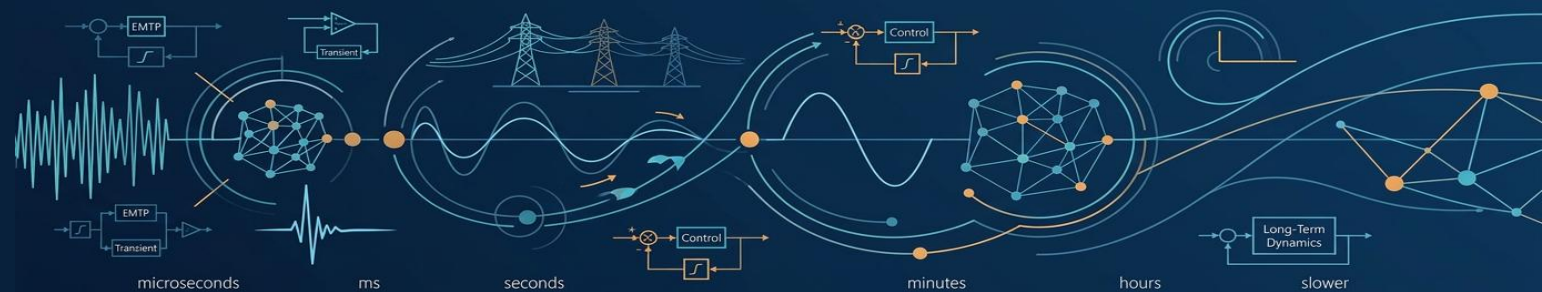
Nebalansirane trofazne mreže (u realnosti)



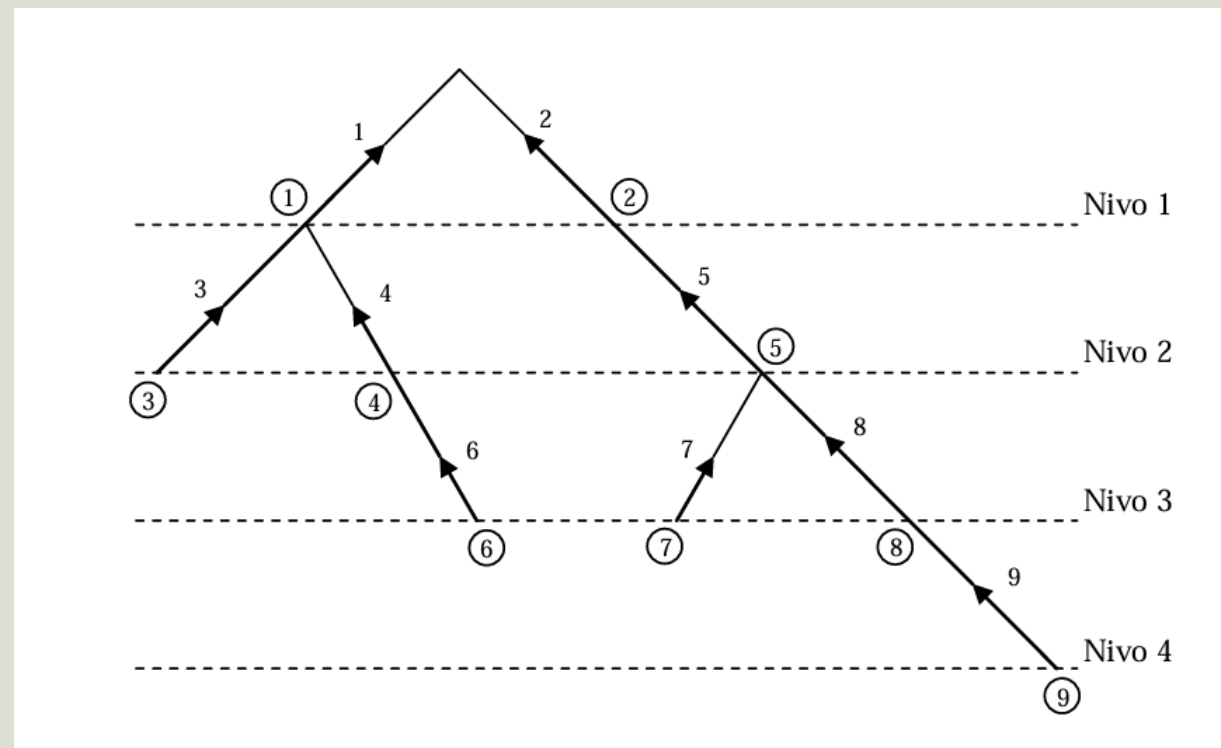
## 2. Teorija i izvođenje jednačina



# Osnovna ideja: Forward-Backward Sweep



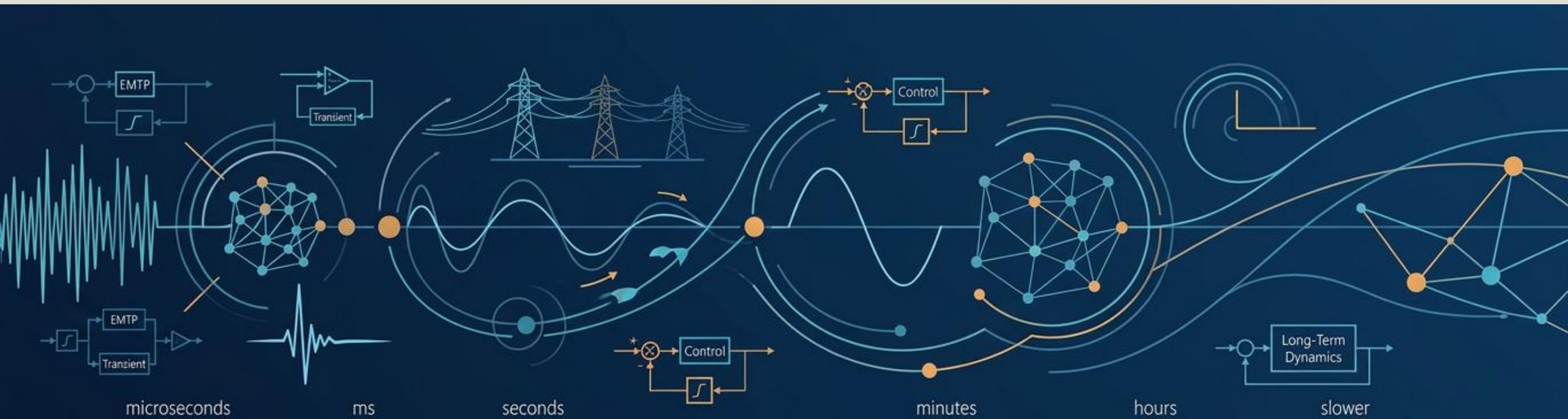
- 1. Backward sweep:** Proračun tokova snage od kraja ka izvoru
- 2. Forward sweep:** Ažuriranje napona od izvora ka kraju
3. Iterativni proces do konvergencije



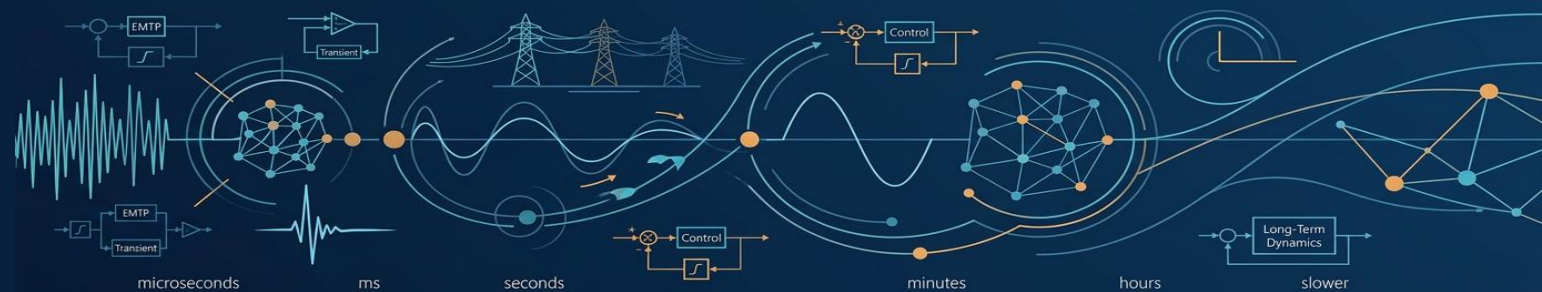
$$S = P + jQ = V \cdot I^* \quad I_{ij} = \frac{V_i - V_j}{Z_{ij}} = \frac{V_i - V_j}{R_{ij} + jX_{ij}}$$

$$I^* = \frac{S}{V} = \frac{P + jQ}{V} \quad I = \frac{S^*}{V^*} = \frac{P - jQ}{V^*}$$

### 3. Pseudokod i flowchart



# Pseudokod



## Šta je?

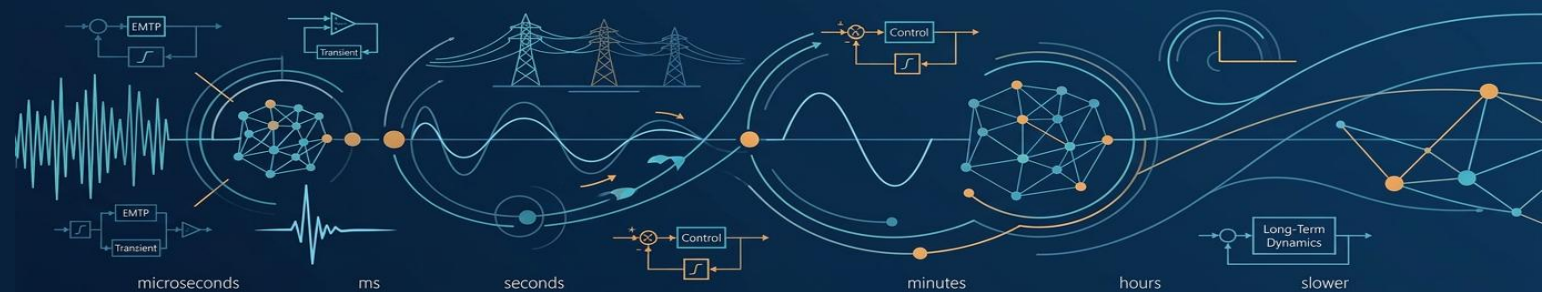
Pseudokod je tekstualni opis algoritma napisan u strukturiranoj formi koja liči na programski kod, ali koristi prirodan jezik (srpski/engleski) umesto sintakse konkretnog programskog jezika.

## Zašto ga koristimo?

Služi kao most između matematičkih jednačina i programskog koda - omogućava nam da osmislimo logiku algoritma (petlje, uslove, korake) bez brige o sintaksnim detaljima kao što su tačka-zarez, deklaracije tipova ili specifična sintaksa programskog jezika.

Aspekt	Matematika	Pseudokod	MATLAB kod
Jezik	Simboli, jednačine	<b>Reči + struktura</b>	Sintaksa jezika
Primer	$V_j = V_i - \Delta V$	<b>V[j] = V[i] - delta_V</b>	V(j)=V(i)-deltaV;
Cilj	Tačan izraz	<b>Algoritamska logika</b>	Izvršiv program

# Pseudokod



## PRIPREMA PODATAKA

### UNOS PODATAKA

- Bus data: P\_load[], Q\_load[], V\_initial[]
- Branch data: from[], to[], R[], X[]
- Slack bus: V\_slack

### FORMIRANJE TOPOLOGIJE STABLA

- Identifikuj strukturu stabla (parent-child relacije)
- Odredi redosled čvorova

## INICIJALIZACIJA

### INICIJALIZACIJA

- iteration = 0
- V[all buses] = 1.0 pu // Početni naponi
- V[slack] = V\_slack
- converged = FALSE

```
WHILE (NOT converged) AND (iteration < max_iter):
```

```
    iteration = iteration + 1
```

```
    V_old = V // Čuvanje za proveru konvergencije
```

```
// ===== BACKWARD SWEEP =====
```

```
FOR each branch from END to SOURCE:
```

```
    Struja opterećenja + struja downstream grana
```

```
    I_total [j] = I_load [j] + SUM(I_downstream)
```

```
END FOR
```

```
// ===== FORWARD SWEEP =====
```

```
FOR each branch from SOURCE to END:
```

```
    V [j] = V_upstream - Z[j]*I[j]
```

```
END FOR
```

```
// ===== CHECK CONVERGENCE =====
```

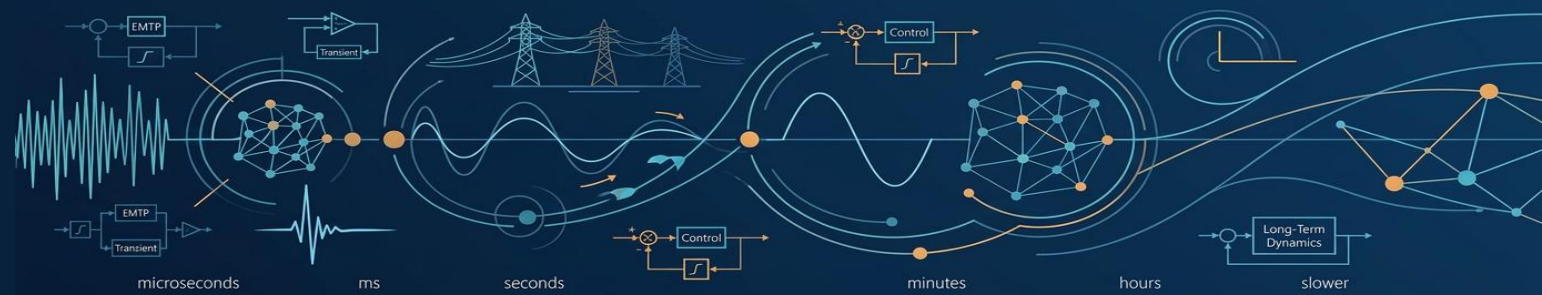
```
error = MAX(|V - V_old|)
```

```
IF error < tolerance:
```

```
    converged = TRUE
```

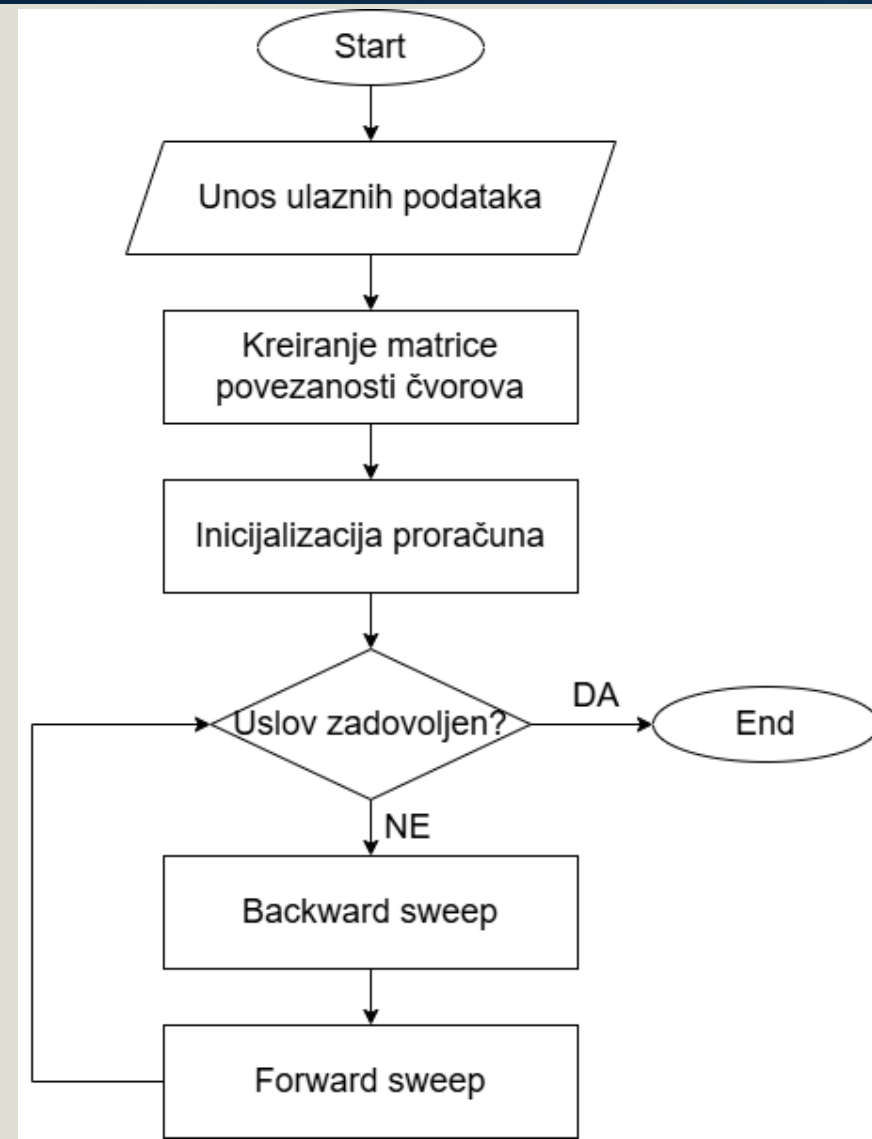
```
END IF, END WHILE
```

# Blok dijagram Flowchart

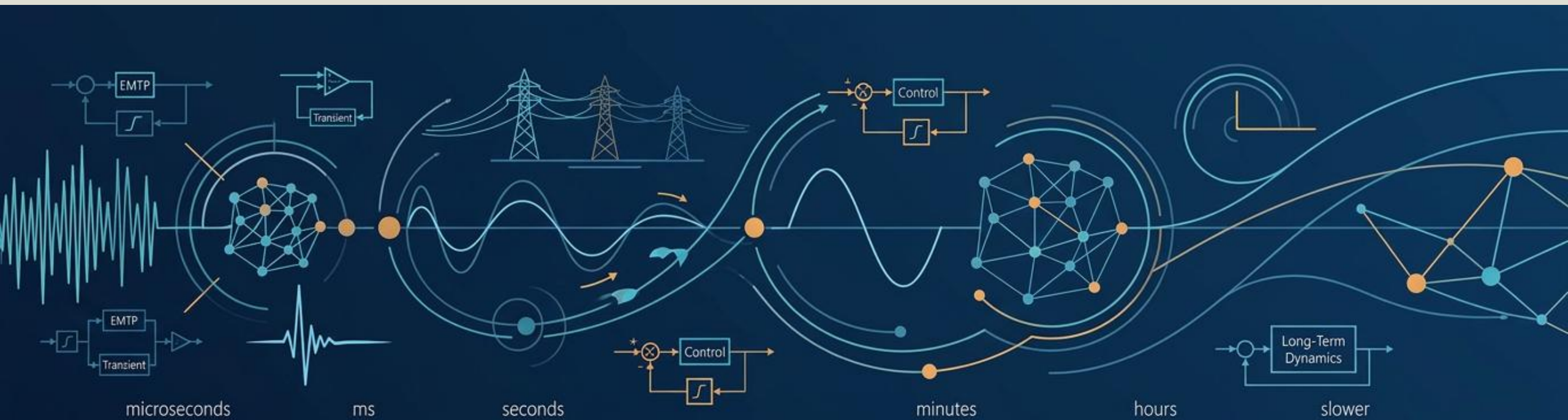


Simbol	Značenje	Primer
Elipsa	Početak/Kraj	START, END
Pravougaonik	Proces/računanje	$V[j] = V[i] - \Delta V$
Romb	Odluka (uslov)	$error < \epsilon ?$
Paralelogram	Ulaz/Izlaz	INPUT: Bus data
Strelica	Tok izvršavanja	→

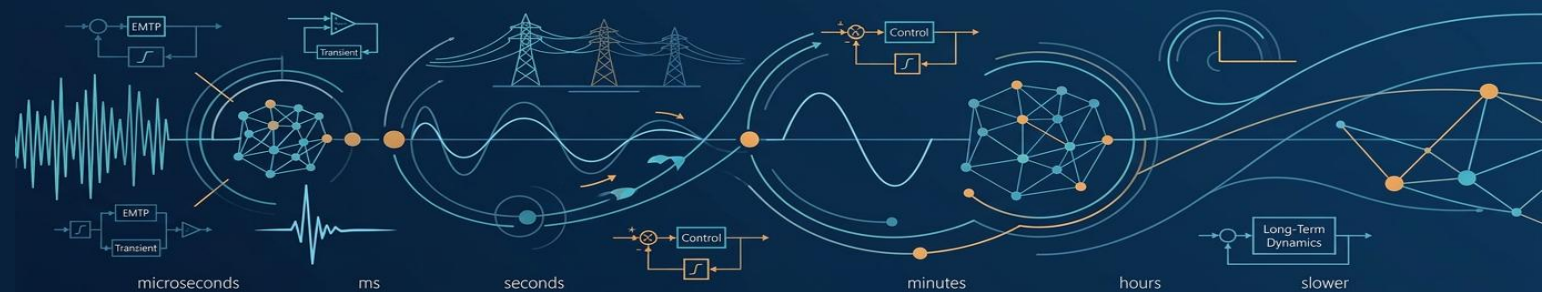
<https://app.diagrams.net/>



# 4. Praktična realizacija u MATLAB-u

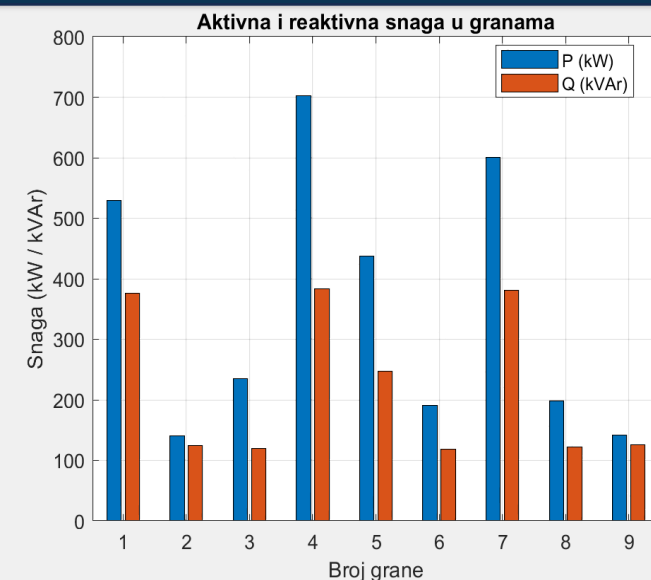
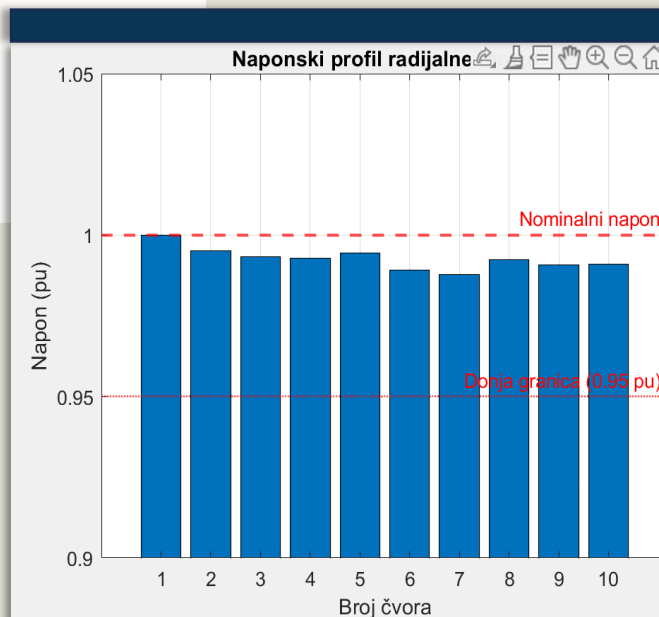


# MATLAB implementacija



shirmohammadi\_project/

- |– main\_shirmohammadi.m % Glavni fajl
- |– load\_test\_system.m % Script: učitavanje podataka
- |– initialize\_system.m % Script: inicijalizacija
- |– build\_topology.m % Script: topologija
- |– backward\_sweep.m % Script: backward sweep
- |– forward\_sweep.m % Script: forward sweep
- |– check\_convergence.m % Script: konvergencija
- |– calculate\_results.m % Script: rezultati
- |\_ plot\_results.m % Script: grafici



# Hvala na pažnji!

