



Univerzitet u Beogradu  
Elektrotehnički fakultet

# PROJEKTOVANJE POMOĆU RAČUNARA U ELEKTROENERGETICI

## Osnovne akademske studije

Dr Zlatan Stojković, redovni profesor  
[zstojkovic@etf.rs](mailto:zstojkovic@etf.rs)  
<http://ees.etf.rs>

# **FORMIRANJE PROGRAMA**

---

# RAZVOJ ALGORITMA

zahtevi za  
resursima



Da bi se napisao dobar program u startu je potrebno imati dobar matematički model.

Pre početka pisanja programa potrebno je formirati kompletan algoritam.

Algoritam ne bi trebalo da sadrži detalje proračuna, već informacije šta je i kojim redosledom neophodno uraditi. Neophodno je definisati mesta odluke.

*Primer*

S obzirom da je programerski tim sastavljen od više članova, potrebo je usvojiti određene standarde i procedure:

1. Da li model bazirati na frekvencijski nezavisnim ili frekvencijski zavisnim parametrima?
2. Kakvu bi strukturu uzemljivača model trebalo da reprezentuje?
3. Da li obuhvatiti nelinearne efekte? Koje metode rešavanja nelinearnih sistema primeniti?



# ZAHTEVI ZA RESURSIMA



S obzirom na veličinu programa, potrebno je proceniti relaciju između veličine i brzine hardvera i veličine softvera.

Razvoj računara sve više eliminiše ovaj problem, međutim i dalje postoje problemi čija analiza i rešavanje zahtevaju kompleksan matematički aparat i znatno trajanje izvršenja programa (metoda konačnih elemenata ili konačnih razlika).

Na brzinu izvršenja programa, pored hardvera i softvera, utiču i stepen složenosti modela i režim rada korisnika.



# KARAKTERISTIKE PROGRAMA



Karakteristike programa se ogledaju u sledećem:

1. Struktura programa
2. Prenosivost programa
3. Ulaz/izlaz
4. Programska jezik
5. Dokumentacija
6. Razvoj programa
7. Testiranje programa

karakteristike  
programa



# STRUKTURA PROGRAMA

prenosivost  
programa



Svaki program je potrebno strukturirati tako da sadrži tri dela:

preprocesor

glavni program

postprocesor

1/2





# PRENOSIVOST PROGRAMA



Prenosivost je mera sposobnosti da se program formiran na jednom računaru može izvršavati na drugom računaru.

Programi pisani na mašinskom jeziku su u principu neprenosivi.

Programi pisani na višim programskim jezicima su prenosivi.

# PROGRAMSKI JEZIK

Pri izboru odgovarajućeg programskog jezika potrebno je voditi računa o sledećem:

Poželjno za programera	Poželjno za korisnika
Izvršiti potrebne operacije na pogoda način	Odgovarajući ulaz/izlaz (uključujući grafike)
Modularnost	Brz proračun
Sopstvena dokumentacija	
Prenosivost	
Lako održavanje i nadogradnja	

Tri često primenjivana programska jezika u elektroenergetici su:

1. FORTRAN (programski alat EMTP/ATP)
2. MATLAB
3. C/C++



Korišćenje programskog alata EMTP/ATP usmereno je ka rešavanju dve grupe problema:

1. Problemi koji se odnose na projektovanje elektroenergetskih sistema koji uključuju izbor opreme, koordinaciju izolacije, specifikaciju uređaja za zaštitu, projektovanje sistema automatskog upravljanja, itd.
2. Problemi vezani za planiranje i eksploraciju elektroenergetskih sistema





## Sklopni prenaponi

- deterministički; probabilistički; sklopna operacija jednim polom; isključenje sa APU; sklopne operacije sa kondenzatorskim baterijama; prelazni povratni napon

## Atmosferski prenaponi

- povratni preskok; indukovani prenaponi; prenaponi u razvodnim postrojenjima

## Koordinacija izolacije

- nadzemni vodovi; razvodna postrojenja na otvorenom; gasom izolovana razvodna postrojenja; odvodnici prenapona

## Jednosmerni sistem visokog napona

- upravljanje; prelazni procesi; harmonici

## Statička kompenzacija

- upravljanje; prenaponi; harmonici





Programski alat EMTP/ATP uključuje sledeće komponente:

1. Nespregnute i spregnute linearne elemente sa koncentrisanim parametrima
2. Nadzemne vodove i kablove sa raspodeljenim i frekvencijski zavisnim parametrima
3. Nelinearne elemente: transformatore, uključujući pojavu zasićenja i histerezisa, odvodnike prenapona, električni luk
4. Prekidače, vremenski i naponski zavisne prekidače
5. Diode i tiristore
6. Trofazne sinhronne mašine
7. MODELE i TACS (elementi sistema automatskog upravljanja za koje se sprovodi analiza prelaznog režima)





Programski alat:

- Numerička analiza
- Matrični račun
- Obrada podataka
- Grafički prikaz

Programski jezik:

- Jednostavno rešavanje matematičkih problema
- Jednostavan zapis
- Potisnuo FORTRAN, BASIC, C





Simulinik – specijalizovan modul za simulaciju dinamike sistema u grafičkom okruženju:

- Analiza linearnih, nelinearnih, vremenski kontinualnih ili diskretnih multivarijabilnih sistema sa koncentrisanim parametrima

Blok-dijagram – matematički model koji ilustruje dinamičke karakteristike sistema, glavne promenljive sistema i veze između tih promenljivih.

Svakom Simulink modelu odgovara m-datoteka koja predstavlja niz MATLAB i Simulink komandi i funkcija kojima se formiraju Simulink modeli.





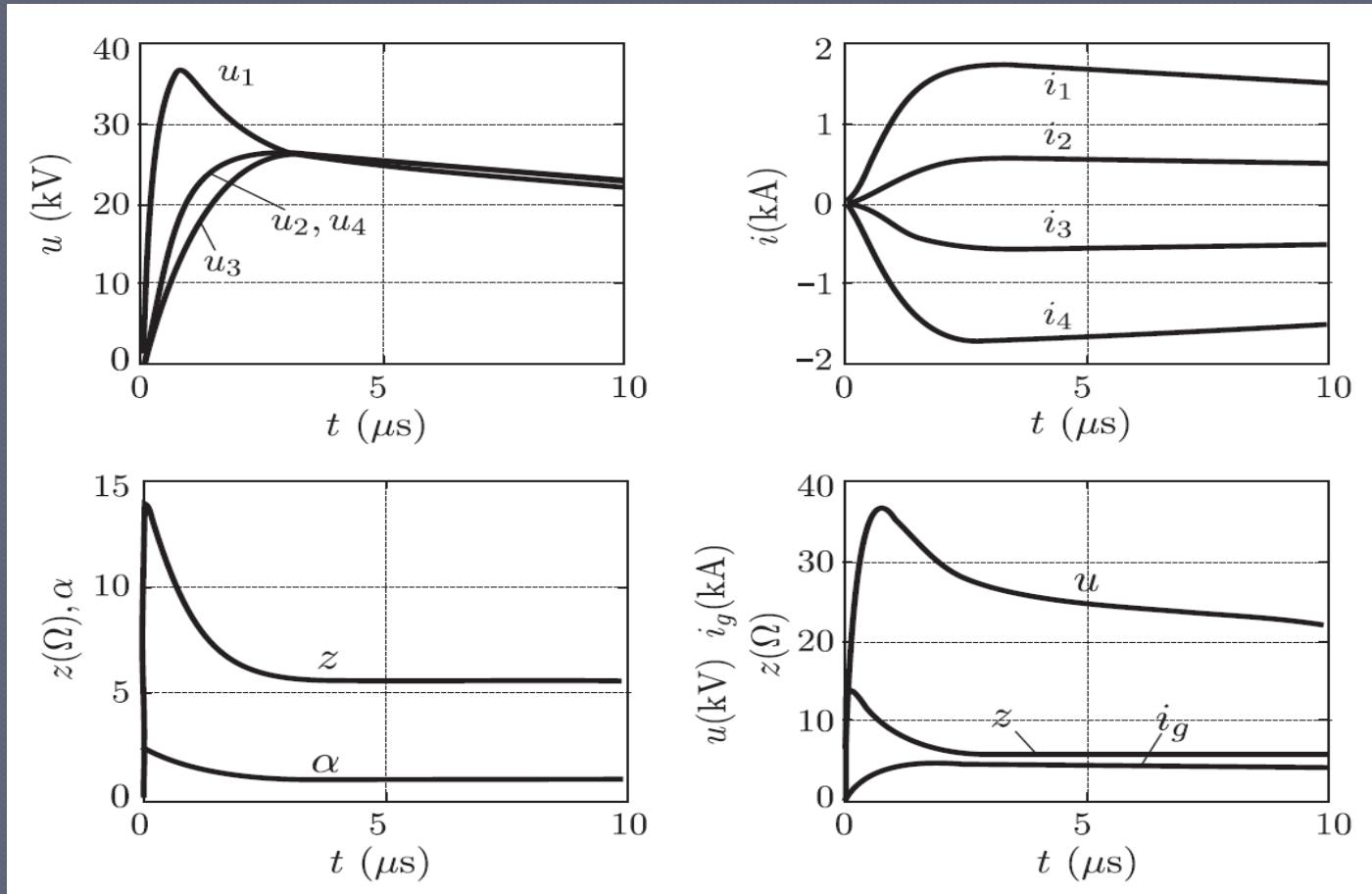
Primer programiranja u MATLAB-u:

- Za uzemljivač razmatran u prethodnim primerima proračunati:
  1. Udarnu impedansu uzemljivača u funkciji vremena
  2. Udarni koeficijent uzemljivača u funkciji vremena
  3. Konvencionalnu impedansu uzemljivača
  4. Konvencionalni koeficijent uzemljivača
- Ekran podeliti na četiri dela i nacrtati vremenske promene:
  1. Napona čvorova
  2. Struja induktivnih grana
  3. Udarne impedanse i udarnog koeficijenta uzemljivača
  4. Napona na mestu injektiranja, struje injektiranja i udarne impedanse uzemljivača





Rezultati proračuna udarnih karakteristika uzemljivača:





Programski jezik C je programski jezik opšte namene, srednjeg nivoa:

- Strukturirani tipovi datoteka i upravljačke strukture
- Manipulacija bitovima, procesorski registri, pristup podacima pomoću adrese, operatori orijentisani ka hardveru

Osnona verzija u Bell laboratoriji – operativni sistem UNIX

C → C++:

1. Unošenje izvornog teksta programa u datoteku na disku
2. Prevođenje izvornog teksta programa
3. Povezivanje prevedenog oblika programa sa potebnim korisničkim i sistemskim potprogramima u izvodljiv oblik
4. Izvršavanje programa

Programski alat GIC – MATLAB + Visual C++

# DOKUMENTACIJA



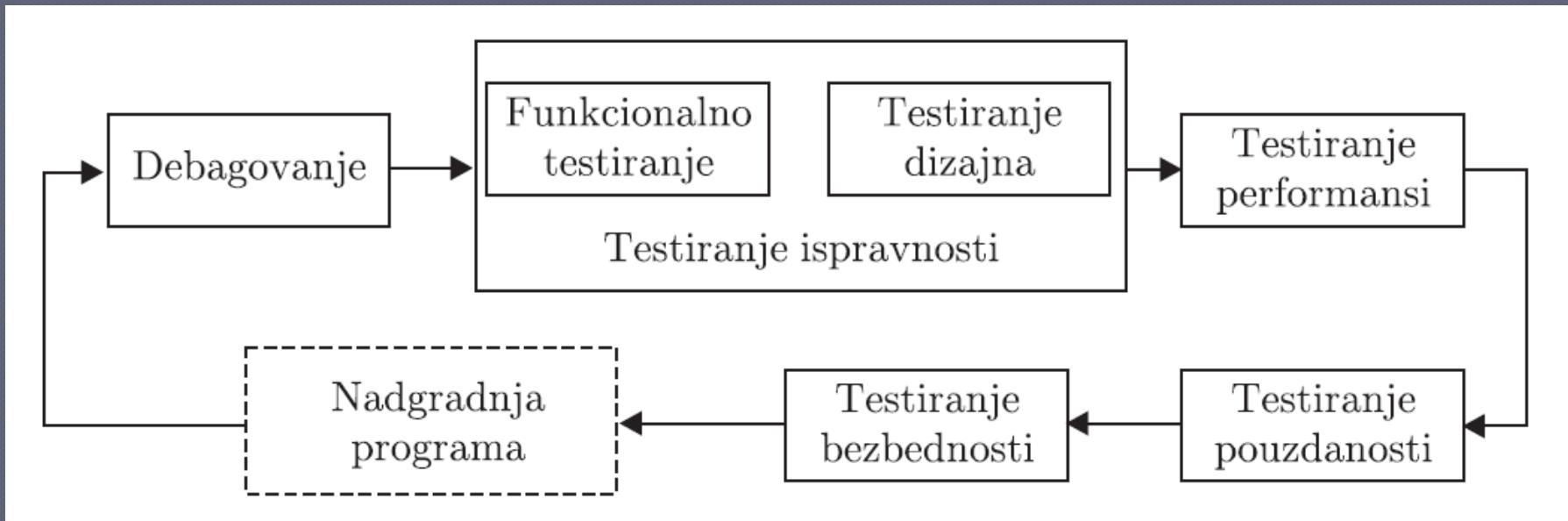
Redni broj	Korisnik	Programer	Dokumentacija
1	x	x	Opis šta program radi, metodi
2	x	x	Primer ulaza/izlaza
3	x	x	Opis ulaza/izlaza
4	x	x	Moguće greške u ulaznim podacima i poruke
5	x	x	Organizacija podataka, datoteke i izgled
6	x	x	Reference
7	x	x	Matematički model na kojem se bazira program
8	x	x	Zapis razvoja programa, ponovni pregled
9		x	Imena svih obuhvaćenih potprograma
10		x	Listing programa
11		x	Tumačenje pojmove promenljivih
12		x	Algoritam za proračun
13		x	Pisani komentari u vezi programa
14	x	x	Test primer sa kompletnim ulazom/izlazom

1. Formiranje novih modula u okviru postojećih softverskih alata
  - EMTP/ATP
  
2. Razvoj sopstvenog softverskog alata
  - Nedostupnost adekvatnog postojećeg softverskog alata
  - Usko specijalizovani softverski alati
  - Efikasniji i korisnički orijentisani



# TESTIRANJE PROGRAMA

Metodologija testiranja programa:





# TESTIRANJE PROGRAMA

Izvori mogućih razlika rezultata dobijenih proračunom i iz realnog pogona

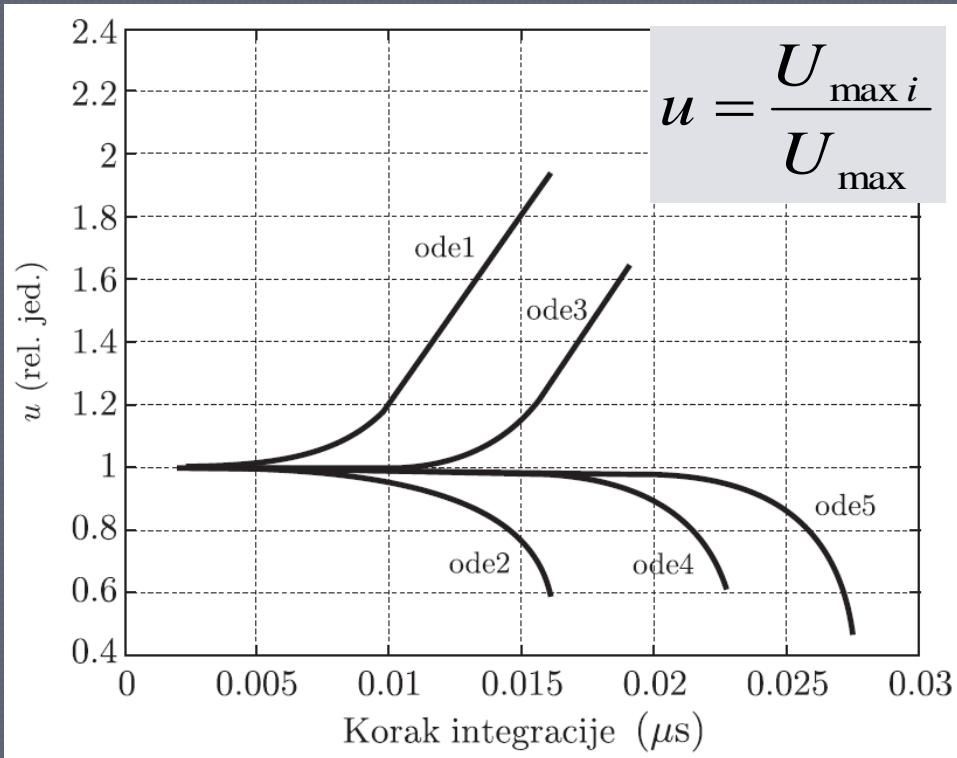
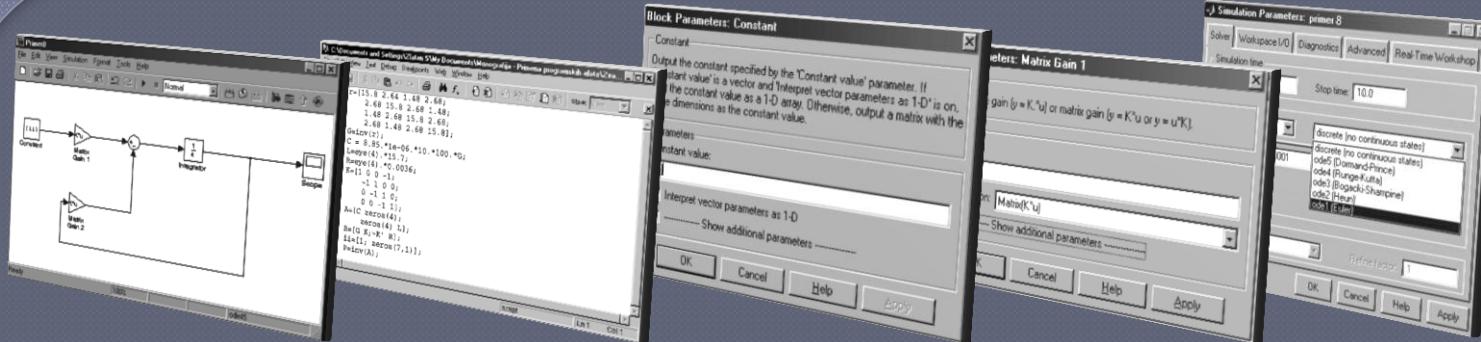
Pesimistički  
pristup

Kvalitet  
modela

Matematički  
model

Ulazni  
podaci





$u$  – relativna vrednost maksimuma napona na mestu injektiranja

$U_{\max i}$  – maksimalna vrednost napona na mestu injektiranja za  $i$ -tu vrednost koraka integracije

$U_{\max}$  – referentna vrednost maksimuma napona na mestu injektiranja za najmanju vrednost koraka integracije

Lokalna nestabilnost