Primena računara u fizičkoj hemiji

Nastavnik: ***prof. Miloš Mojović***

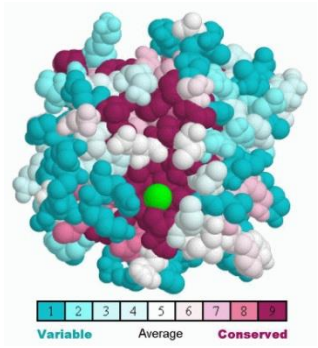
Asistent: ***dr Aleksandra Pavićević***

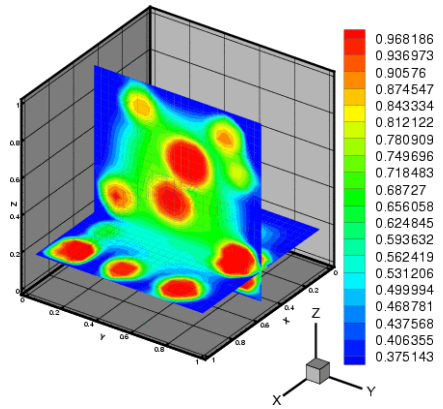
http://www.ffh.bg.ac.rs/примена-рачунара-у-физичкој-хемији

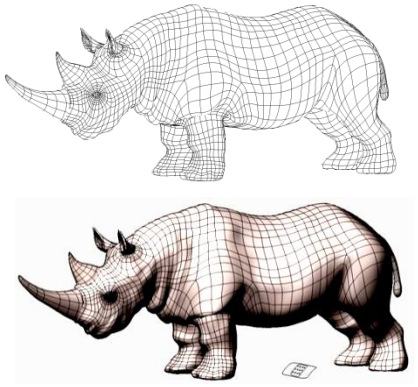
milos@ffh.bg.ac.rs

aleks.pavicevic@ffh.bg.ac.rs

Uvod u računarske simulacije:







Teorijske osnove računarskih simulacija:

- Napredak računarskih sistema doprineo je da računare možemo koristiti za simuliranje različitih fizičkohemijskih procesa.
- Karakterizacija nekog fizičkohemijskog sistema se, u principu, zasniva na numeričkim proračunima nekakavih fizičkih veličina prema zakonitostima koje određuje teorija.
- Rezultati dobijeni izračunavanjem nekih formula moraju imati i svoju eksperimentalnu potvrdu.
- Ukoliko se teorijski proračuni i praktično dobijeni rezultati poklope, teorija se može smatrati ispravnom (i obrnuto, eksperiment se može smatrati ispravno izvedenim).
- Međutim, između teorije i eksperimenta može se postaviti **SIMULACIJA** kao npr. sintetički izveden eksperiment koji se bazira na teorijskim konceptima.
- Proračuni koji se izvode na jednostavnim, idealnim, sistemima su relativno prosti i često ne predstavljaju problem. Međutim, ukoliko sistem ima puno parametara (tj. elemenata sistema), situacija se znatno usložnjava.

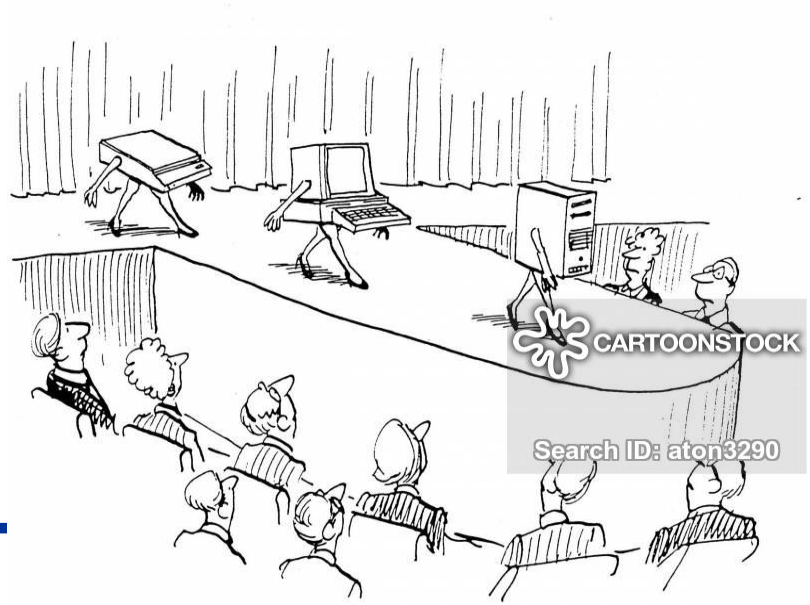
Tipovi računarskih simulacija:

- Realni sistemi, sa kojima se susrećemo u svakodnevnom životu, odnose se najvećim delom na veliki broj čestica sa još većim brojem interakcija između njih.
- Ponašanje ovih sistema najčešće sa ispituje putem **SIMULACIJA** preko konstrukcije **MODELA** koji predstavlja uprošćenu reprezentaciju realnog sistema.
- Ukoliko je model sličniji realnom sistemu simulacija će biti realnija (ali to često zahteva veoma jake računarske sisteme).
- Modeli mogu biti: eksperimentalni (npr. neka maketa u vakuumskoj komori) ili kompjuterski (danas se sve više koriste).

- Kompjuterske simulacije se generalno mogu podeliti na:

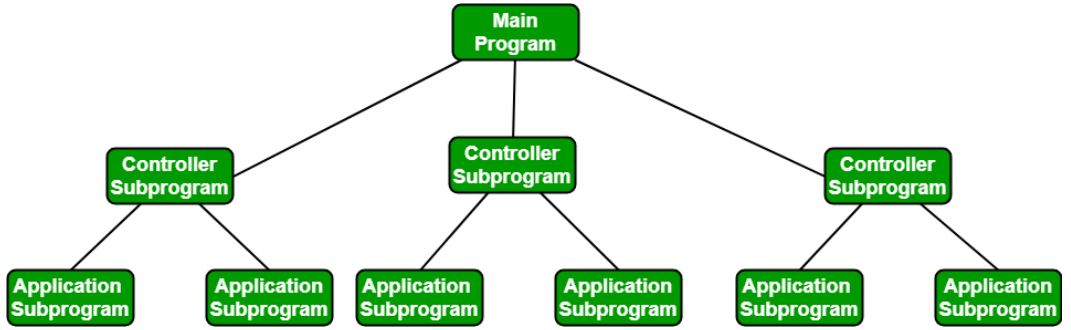
  - one koje koriste čestične sisteme (mikroskopski pristup) - koordinate, brzina čestica ...

  - one koji koriste kontinualne sisteme (makroskopski pristup) - pritisak, temperatura, gustina ...



Prototip i model:

- **Prototip** predstavlja similaciju deo (podsistem) realnog sistema čiji model želimo da simuliramo. On obično predstavlja kompleksni sistem koji se sastoji od više odvojenih podsistema koji međusobno interaguju.
- Naš model bi trebao da predstavlja sve podsisteme, ali ponekad je korisno da se neki od njih smatraju neaktivnim (ili konstantama pod datim uslovima).
- Predstavljanje modela pomoću računara bi takođe trebalo da prati ovakvu strukturu, što znači da bi program trebao da se sastoji iz više delova (potprograma ili podrutina) koje se po potrebi pozivaju u glavni program.
- Nakon izvršenih proračunavanja kao rezultat imamo neke podatke (ili najčešće velike grupe podataka) koje treba nekako jasno predstaviti (a to je uvek najbolje uraditi putem grafičkog prikaza). Iz tog razloga, rad sa računarskim simulacijama usko je povezan sa kompjuterskom grafikom.

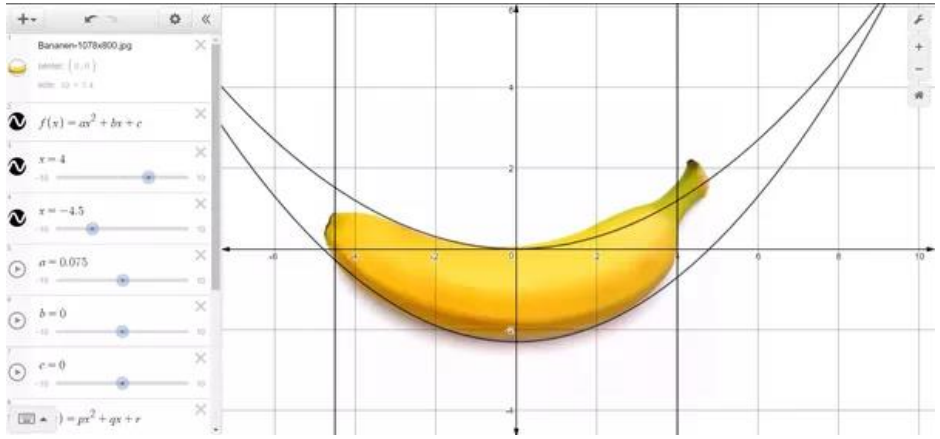


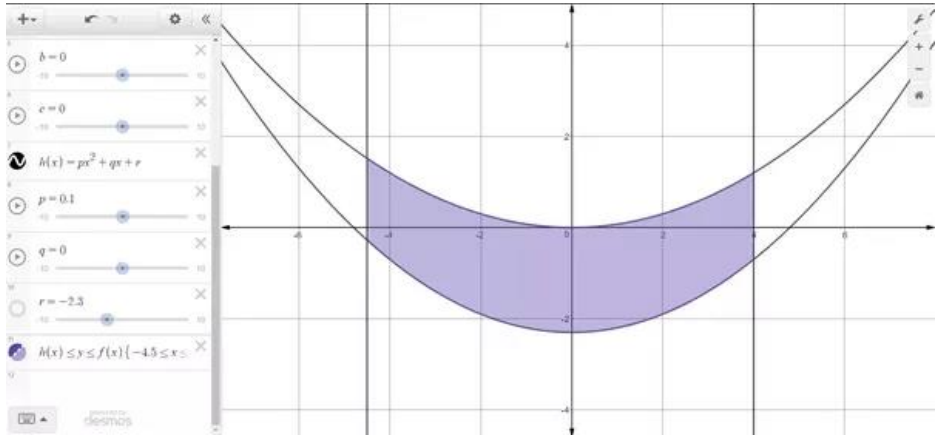
Validacija modela:

- Da bi bili sigurni da model ispravno reprezentuje eksperiment koji želimo da izvršimo, mora se izvršiti **validacija modela**. Ovaj postupak se sastoji iz nekoliko koraka:

  - provera grešaka na nivou programa (da se vidi da li program radi onaj posao za koji je napisan). Ovo se radi tako što se programu zadaju ulazni parametri za koje je rezultatat već poznat (mukotrpan posao koji je toliko uspešniji što je broj provera veći i raznovrsniji).

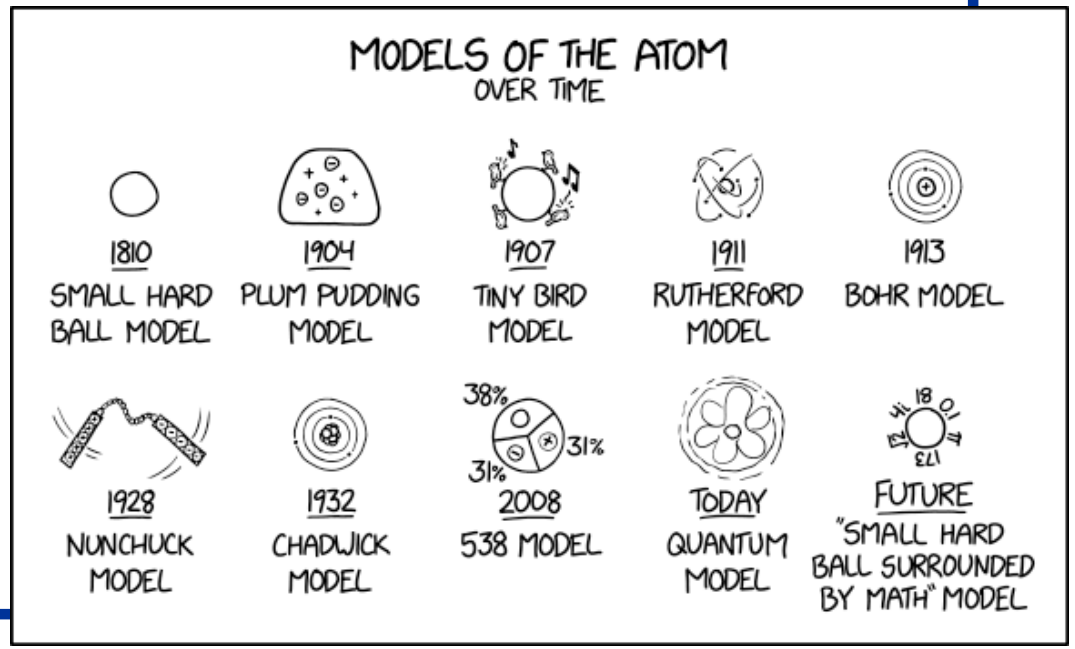
  - provera ispravnosti funkcionisanja modela (da li zaista predstavlja ono što smo želeli da simuliramo).





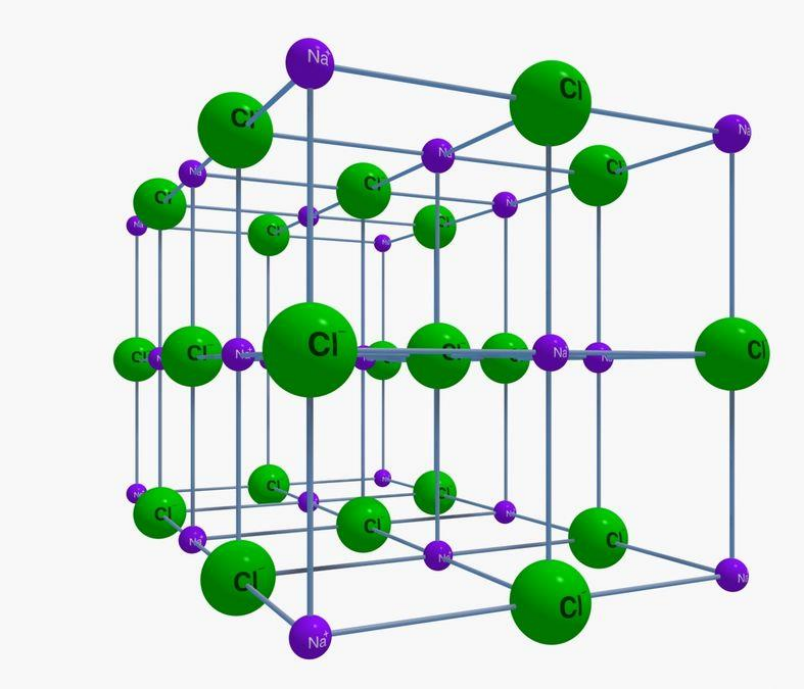
Model i simulacija:

- Suština **modela** je to da on predstavlja uprošćenu reprezentaciju nekog realnog objekta ili fizičkog procesa u cilju reševanja nekog (makar ograničenog) problema.
- Modeli mogu opisivati realnu situaciju sa manjom ili većom preciznošću i treba imati u vidu koliko precizan model je potrebno simulirati da bi se dobili zadovoljavajući rezultati.
- Jedan od modela svima poznat je napravio Nils Bor 1913. (Borov model atoma) gde su elektroni predstavljani kao male naelektrisane čestice koje se vrte oko pozitivno naelektrisanog jezgra (planetarni model). Iako je on daleko od preciznog opisa realne situacije, on je do izvesnog stepena sasvim dobro mogao da nam pojasni neke atomske interakcije.
- Problem kod ovog modela je to što je ubrzano kretanje elektrona trebalo da uslovljava emitovanje EM zračenja, ali je Bor zato uveo postulate o postojanju samo diskretnih E nivoa (i prelazima između njih).
- Naime, Borov model sasvim lepo funkcioniše za H-atom, ali je za potpuno opisivanje složenijih atoma potrebno uvesti izvesne korekcije u vidu kvantno-mehaničkih modela Šredingera i Hajzenberga (1926. godina).
- Ovde se vidi prednost matematičkog u odnosu na fizički model. Fizički model može biti realno konstruisan (od kuglica i žica) ali neće biti svrsishodan kao matematički model koji može pokazati prednosti i mane nekog modela.



Primeri modelovanja:

- Još jedan od primera kada je korisno praviti modele je kada proučavamo kristalnu strukturu materijala. Na slici je primer kubne rešetke sa atomima koji su postavljeni u rogljevima elementarne ćelije.

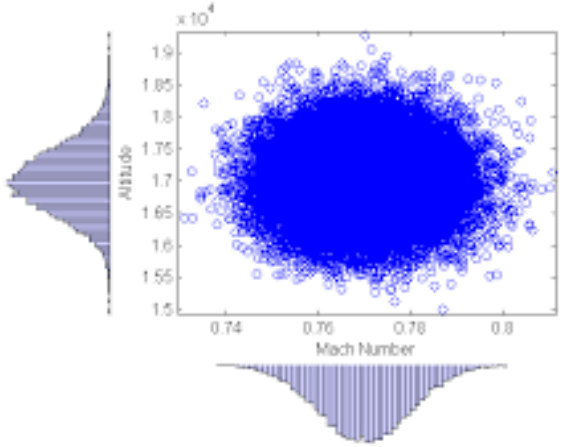


- U ovom klasičnom modelu smatra se da su svi atomi u stanju mirovanja iako u realnosti oni vibriraju oko ravnotežnih položaja.
- Ukoliko posmatramo sistem koji pravi male oscilacije možemo primeniti harmoničnu aproksimaciju, ali ukoliko atomi počnu da, pod određenim uslovima, više odstupaju od ravnotežnog položaja, neophodno je uključiti i jednačine za anharmonične vibracije.
- Koji model ćemo primeniti u našoj simulaciji, dakle, zavisi od vrste problema koji treba rešiti i od eksperimentalnih uslova pod kojima želimo da se ispitivani sistem nalazi. Kako to drugi rade?

Primeri programa koji simuliraju različite sisteme

Tipovi simulacija i njihove primene:

- Postoji nekoliko osnovnih načina na koje možemo izvršiti simulacije različitih tipova problema.



1) Monte Karlo metoda

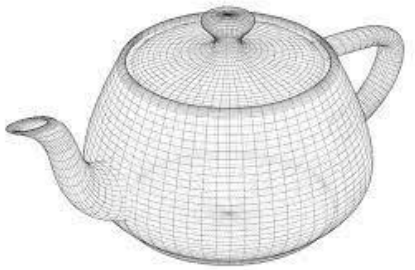
2) Metoda čestica

3) Metoda konačnih promena

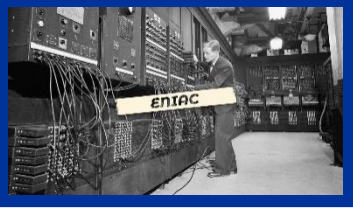
4) Metoda konačnih elemenata

5) Simulacije kompleksnih sistema

- Ovo su metode koje se najčešće koriste i biće ukratko objašnjeno na kom principu svaka od njih funkcioniše, kao i na kojim tipovima problema se svaka od njih može primeniti.
- Pored ovih, postoji još puno drugih metoda koje su specifične za simulacije određenih fizičkohemijskih problema.



Monte Karlo metoda:



1) Monte Karlo metoda

Tokom II svetskog rata američki naučnici su intenzivno radili na dizajniranju atomske bombe. U svom radu često su se susretali sa problemom brzog rešavanja zadataka sa kojima do sada nikada nisu imali nikakvih iskustava (npr. problem kako će neutroni prolaziti kroz nekakvu barijeru). Kao što je poznato, neutroni sa nekom sredinom ili: ne interaguju, elestično ili neelastično bivaju odbijeni, ili mogu biti absorbovani. Verovatnoća svakog od ovih procesa zavisi od energije neutrona. Iako je zavisnost verovatnoće svakog od ovih procesa dobro poznata, ne postoji nikakav način da se ovaj problem reši konvencionalnom matematičkom analizom.

Ulam i **Fon Nojman** su ovaj problem uspeli da reše na sledeći način: Posmatrali su prolazak jednog neutrona kroz barijeru. Da bi odlučili šta će se sa njim dalje dešavati, okretali su točak na ruletu. Prateći kretanje neutrona i praveći Monte Karlo odluku o njihovoj daljoj interakciji sa okolnom sredinom oni su uspeli da, ponavljajući ovaj postupak veoma veliki broj puta, veoma uspešno predvide ponašanje realnog fizičkog sistema (tj. da ustanove koliki broj neutrona je prošlo ili nije prošlo barijeru).

Ulam

Fon Nojman

STAN ULAM, JOHN VON NEUMANN, *and the* MONTE CARLO METHOD

by Roger Eckhardt

The Monte Carlo method is a statistical sampling technique that over the years has been applied successfully to a vast number of scientific problems. Although the computer codes that implement Monte Carlo have grown ever more sophisticated, the essence of the method is captured in some unpublished remarks Stan made in 1983 about solitaire.

"The first thoughts and attempts I made to practice [the Monte Carlo method] were suggested by a question which occurred to me in 1946 as I was convalescing from an illness and playing solitaires. The question was what are the chances that a Canfield solitaire laid out with 52 cards will come out successfully? After spending a lot of time trying to estimate them by pure combinatorial calculations, I wondered whether a more practical method than "abstract thinking" might not be to lay it out say one hundred times and simply observe and count the number of successful plays. This was already possible to envisage with the beginning of the new era of fast computers, and I immediately thought of problems of neutron diffusion and other questions of mathematical physics, and more generally how to change processes described by certain differential equations into an equivalent form interpretable as a succession of random operations. Later... [in 1946, I] described the idea to John von Neumann and we began to plan actual calculations."

Von Neumann was intrigued. Statistical sampling was already well known in mathematics, but he was taken by the idea of doing such sampling using the newly developed electronic computing techniques. The approach seemed especially suitable for exploring the behavior of neutron chain reactions in fission devices. In particular, neutron multiplication rates could be estimated and used to predict the explosive behavior of the various fission weapons then being designed.

In March of 1947, he wrote to Robert Richtmyer, at that time the Theoretical Division Leader at Los Alamos (Fig. 1). He had concluded that "the statistical approach is very well suited to a digital treatment," and he outlined in some detail how this method could be used to solve neutron diffusion and multiplication problems in fission devices for the case "of 'inert' criticality" (that is, approximated as momentarily static config-

Los Alamos Science Special Issue 1987

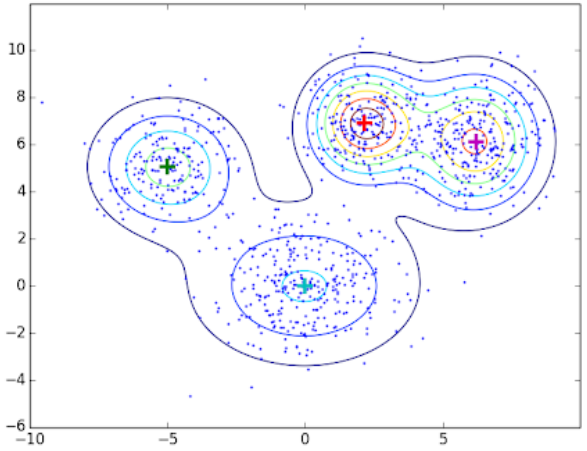
131

Metoda čestica:

2) Metoda čestica

Često je potrebno simulirati sisteme sa veoma velikim brojem čestica (npr. galaksija koja sadrži 10^{11} zvezda ili interkcije u tečnosti koja sadrži 10^{24} čestica). Današnje simulacije iz molekulske dinamike mogu obuhvatati 10^{7}-10^{8} čestica i to ako su uključene samo interakcije kratkog dometa (izm. najbližih suseda). Postoji jedan bazični pristup koji znatno smanjuje broj čestica koji treba posmatrati, a to je da sistem posmatramo preko tzv. **superčestica** pri čemu svaka predstavlja veliki broj realnih čestica.

Dalja aproksimacija može se izvršiti ukoliko se posmatra da se grupacije čestica nalaze u paralelopipedima (često, ali ne uvek u kocki). Uticaj svih čestica u okviru te kocke na neku udaljenu česticu posmatra se kao da se sve čestice nalaze u centru kocke.

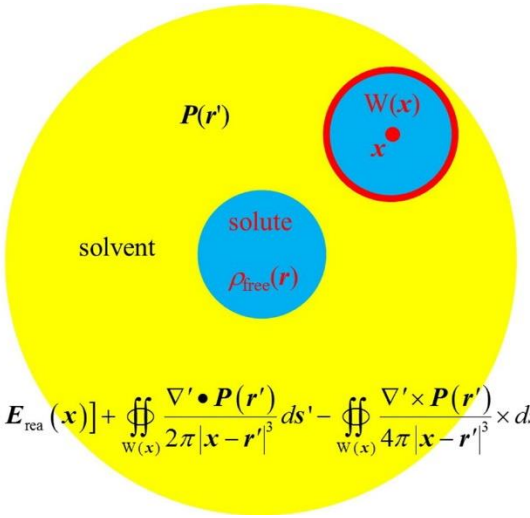


Metoda konačnih promena:

3) Metoda konačnih promena

Koristi se ukoliko je potrebno simulirati sistem čije prostorne i vremenske koordinate parametara kao što su temperatura, gustina, koncentracija, itd. kontinualno variraju sa nekom osobinom koja može biti opisana parcijalnim diferencijalnim jednačinama.

Ova metoda se bazira na tome da osobinu koju posmatramo (recimo da je u pitanju koncentracija) definišemo samo u tačkama virtuelne mreže koja pokriva oblast od interesa. U ovoj aproksimaciji, prave vrednosti parametra koje opisuju diferencijalne jednačine bivaju aproksimirane linearnom kombinacijom vrednosti parametara u tačkama mreže u sadašnjem momentu (poznate vrednosti) i vrednostima parametara nakon određenog vremenskog intervala (one koje će biti određene).



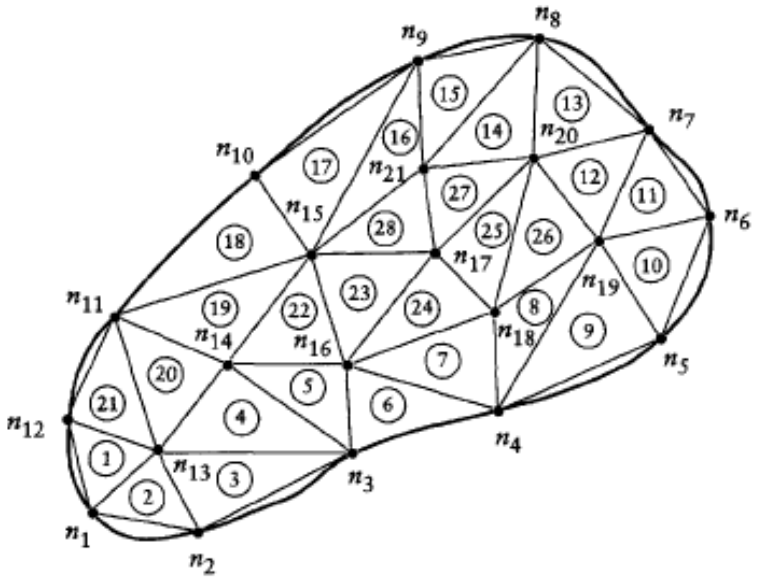
$$\varepsilon_0 \nabla_x^2 [E(x) - E_{rea}(x)] + \oiint_{W(x)} \frac{\nabla' \bullet P(r')}{2\pi |x - r'|^3} ds' - \oiint_{W(x)} \frac{\nabla' \times P(r')}{4\pi |x - r'|^3} \times ds' = \nabla \rho_{free}(r)$$

Ovom metodom uspešno možemo simulirati difuzione procese koji uključuju protok toplote, kretanje talasa i sistema koji mogu biti opisani Poasonovom jednačinom

Metoda konačnih elemenata:

4) Metoda konačnih elemenata

Koristi se najviše za 2D i 3D simulacije ravnotežnih sistema sa malom simetrijom. Specifični region koji treba simulirati se definiše setom tačaka koje se zovu **NODOVI.** Njihovim povezivanjem se dobijaju **ELEMENTI** i i **GRANICE** simuliranog sistema.



Sistem kompleksnog oblika koji je predstavljen od 21 noda i 28 trouglasta elementa.

Ovako prezentovani sistem je mnogo lakše posmatrati i uključiti u matematičke operacije (integracija površina i funkcionalna zavisnost nodalnih vrednosti.

Simulacije kompleksnih sistema

5) Simulacije kompleksnih sistema

Ukoliko je sistem koji želimo da simuliramo isuviše kompleksan u smislu velikog broja interakcija između elemenata sistema, najbolji pristup je "razbiti" kompleksan sistem na set međusobno interagujuh podsistema. U mnogim slučajevima, veliki broj procesa u okviru složenog sistema se odigrava simultano.

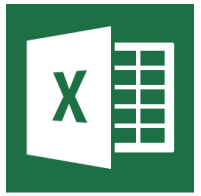
U ovim slučajevima, pribegava se metodi sekvencijalnog vremenskog progresa (engl. time-splitting), pri čemu se svaki proces posmatra u okviru veoma kratkog vremenskog intervala čineći ga nezavisnim od drugih paralelnih procesa.

Jedno od mogućih uprošćenja modela može biti i uključivanje samo onih interakcija koje daju značajan doprinos sistemu, a zanemarivanje onih koje to ne čine. Posebno su važni slučajevi u kojima slabe interakcije postaju značajne u zavsnosti od veličine sistema (npr. površinski napon i gravitaciona sila su zanemarljivi ukoliko je u pitanju mala, dok postaju značajan element simulacije u koliko je u pitanju velika količina materije).

Simulacije i aplikativni softver

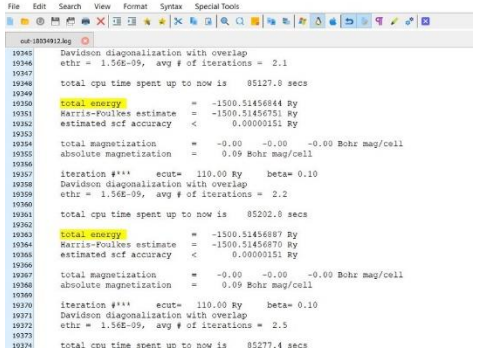
- Kompjuterski programi danas mogu u trenutku rešiti komplikovane matematičke i fizičkohemijske probleme za čije je rešavanje ranije trebalo mnogo više vremena.
- Danas postoji veliki broj kompjuterskih programa koji su napisani za ovu svrhu i stoga prosečnom korisniku nije potrebno detaljno poznavanje programiranja da bi rešio svoj specifični problem. Međutim, korisnik mora da poznaje princip rada ovih programa i način na koji može da izvrši njihovu pripremu za rešavanje svog specifičnog problema.
- Treba se suočiti sa time da kompjuterski programi nisu savršeni niti da mogu rešiti bez greške svaki postavljeni problem.
- Programi o kojima će u ovom kursu u daljem izlaganju biti reči su:

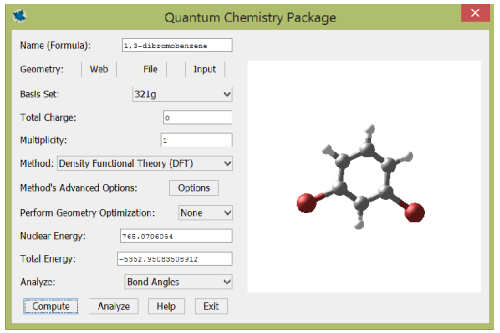
  Excel (dopunske opcije), **Matlab** i **Mathematica**.
- Naravno, treba imati u vidu da je ovo je samo mali segment programa koji se mogu iskoristiti za ove svrhe.



Poznavanje problema:

- Računar može relativno brzo (i uspešno) rešiti zadati problem. Veći problem je zadati (postaviti) zadatak na onaj način kako ga računar može prepoznati.
- Da bi se to uspešno moglo izvesti neophodno je poznavati fizički fenomen koji treba reštiti i postaviti ga u matematički kontekst koji računar može rešiti.
- Problem najčešće može biti rešen na nekoliko načina, upotrebom različitih matematičkih programa od kojih svaki može imati svoje prednosti i nedostatke.
- Na korisniku je da izabere na koji način će problem rešiti i pri tome treba da ima u vidu i način na koji softver koji primenjuje može komunicirati sa drugim programima.
- Počećemo prvo od rešavanja najjednostavnijih jednačina sa jednom nepoznatom pomoću računara.

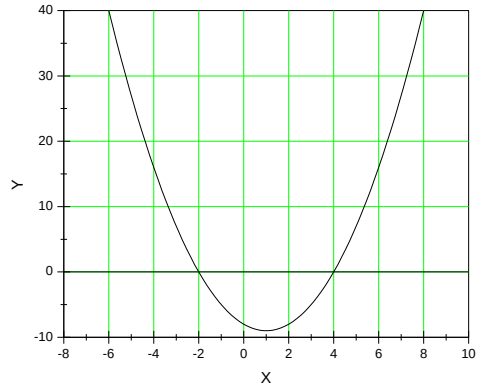




Rešavanje jednačina u programu Excel:

- Primer kako možemo rešiti prostu jednačinu pomoću računara u programu Excel: Jednačina sa jednom nepoznatom

$$f(x) = x^2 - 2x - 8$$



X1=-2
X2=4

- U Excelu se jednačine mogu rešiti pomoću opcije "**Goal Seek**". Otvori se nova radna sveska i u B1 upiše zadata jednačina:

```
=A1*A1-2*A1-8
```

- Cilj je pronaći "nulu" jednačine. Ići na Data-Forecast-What if-Goal Seek i u prikazanom meniju otkucati:

Setcell	B1
To value	0.
By changing cell	A1

Origin: File-New-Function

Rešavanje jednačina u programu Excel:

-2.000007 4.1137E-06

- Kao što se vidi, računar je pronašao rešenje, ali sa malom greškom. Takođe, ni rešenje koje je računar pronašao nije baš tačno nula funkcije, već broj koji je veoma blizu nje (ova tačnost je najčešće dovoljna za većinu zadataka).
- Ukoliko želimo da budemo manje "tolerantni" prema greški koju računar pravi, to možemo regulisati u File-Option-Formulas-Maximum, pa dodati još jednu 0 u maksimalni broj iteracija i tri nule u Maximum change.
- Sada će novo rešenje istog problema biti:

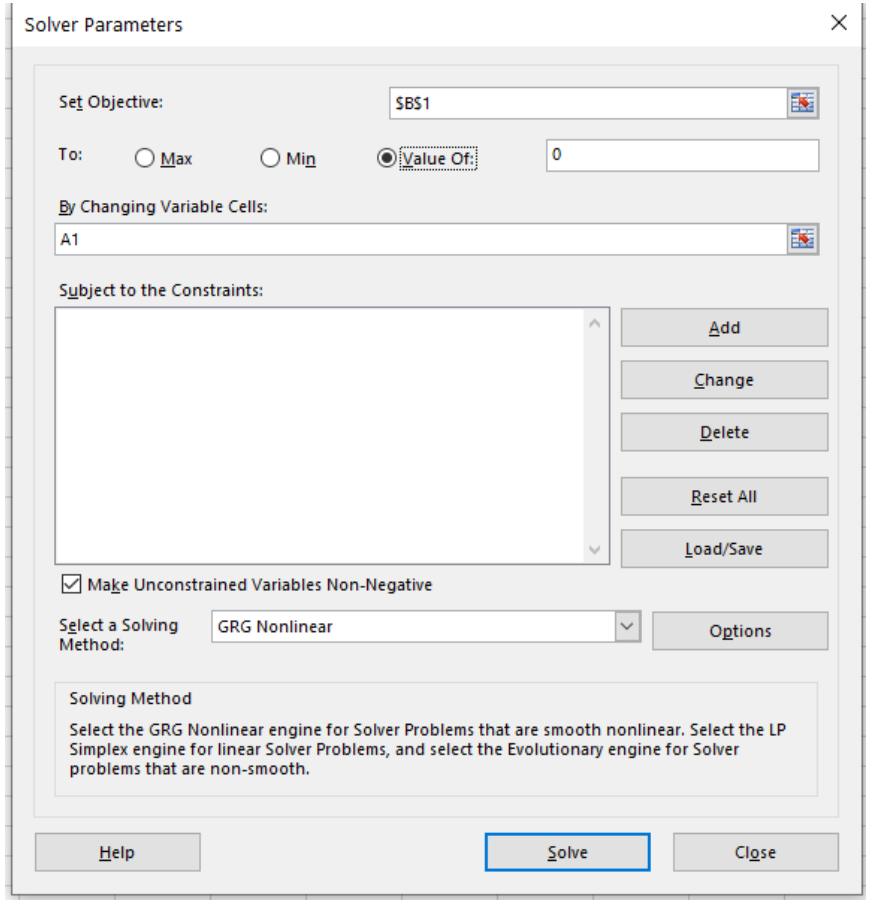
-2 -1.376E-8

- Da bi našli drugi koren jednačine, u A1 ukucati npr. 3 pa zatim ponovo uraditi Goal Seek. Šta se događa ako u A1 ukucamo tačno 4? Šta smo iz ovoga zaključili, kako računar rešava jednačine?

Drugi način rešavanja jednačine u Excelu:

- Isti problem se u Excelu može rešti na još jedan način, upotrebom programa **Solver**.
- File-Options-Add ins-Solver Add-in-Go
- Ići na: Tools-Solver. Ukoliko tu ne postoji Sover, to znači da u toku instalacije ova opcija Excela nije instalisana, tako da je treba potražiti na Tools-Add Ins-Solver (iz Analysis Tool paketa).

- Isti princip se primenjije i ovde, samo što se sve opcije koje se tiču preciznosti dobijenog rezultata nalaze u Options meniju.
- Drugo rešenje se dobija na identičan način kao i u prethodnom primeru.



Primena u fizičkoj hemiji:

- Isti princip rešavanja problema možemo primeniti za rešavanje nekih fizičkohemijskih problema.
- Npr. zadatak je naći specifičnu zapreminu n-butana na 500K i 18 atmosfera korišćenjem Redlih Kwong-ove jednačine realnog gasnog stanja.

$$\hat{v}^3(p) - \hat{v}^2(RT) + \hat{v}\,(a - pb^2 - RTb) - ab = 0$$

$$a = 0.42748\left(\frac{R^2T_c^2}{p_c}\right)\alpha, \quad b = 0.08664\left(\frac{RT_c}{p_c}\right), \qquad T_r = \frac{T}{T_c}, \quad \alpha = \frac{1}{T_r^{0.5}}$$

- Prvi korak je pronaći kritičnu temperaturu i pritisak u literaturi: T_c=245.2K i p_c=37.5 atm (R=0.08206 l atm/g mol K).
- Drugi korak je izračunati vrednosti za koeficijenate a i b.

Rešavanje problema u Excelu:

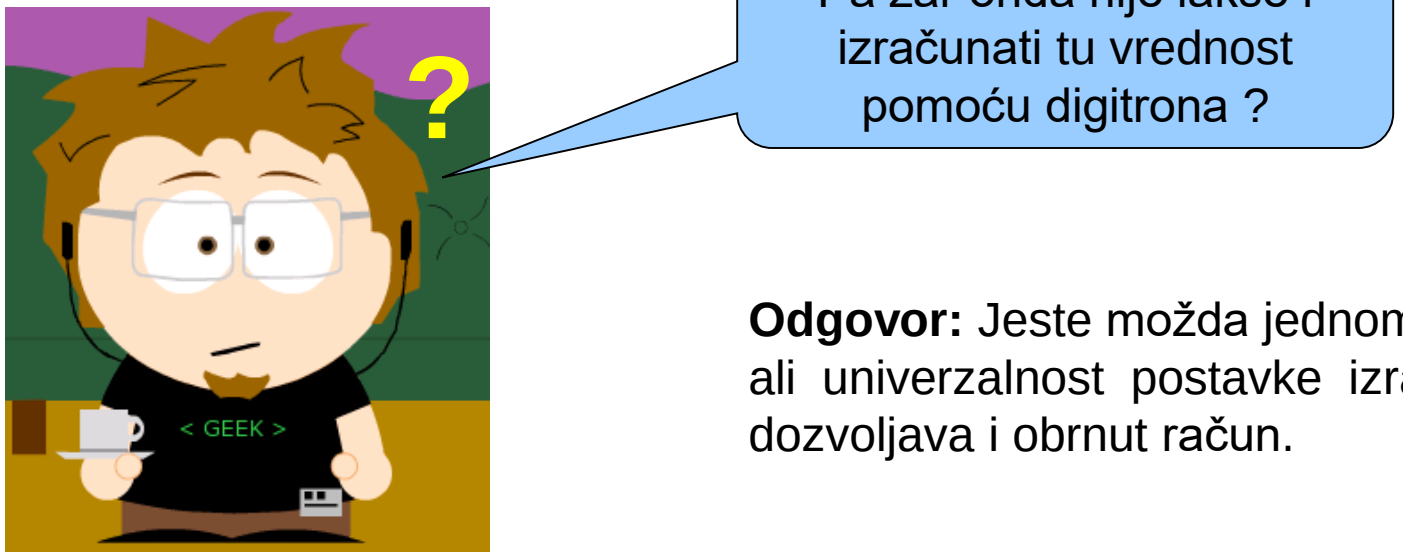
- Napraviti nov dokument u Excel-u i upisati sve poznate parametre i jednačine u odgovarajuća mesta:

	A	B	C	D	E	F	G
18							
19			*n*-Butane				
20							
21							
22		Parameters				Data	
23	Tc	425.2	K		T	500	K
24	pc	37.5	atm		p	18	atm
25	R	0.08206	liter-atm/g mol K				
26							
27		Formulas				Results	
28	Tr	F23/B23			Tr	1.1759	
29	a	0.42748*(B25^2*B23^2/B24)*(B23/F23)^0.5			a	12.7981	
30	b	0.08664*(B25*B23/B24)			b	0.0806	
31	f(v)	F24*F31^3-B25*F23*F31^2			v	2.0377	liter/g mol
32		+(F29-B25*F23*F30-F24*F30^2)*F31-F29*F30			f(v)	1.957E-07	
33	ideal gas	volume	B25*F23/F24		ideal gas	2.2794	

- Pravilno obeležiti poznate i nepoznate veličine i odvojiti ih radi preglednosti. Iskoristiti **Goal Seek** komandu da bi izraz f(v) izjednačili sa nulom i našli rešenje variranjem parametra "v".
- Dobija se rešenje 2.0333 l/gmol (koje možemo porediti sa jedn. ideal gasnog stanja RT/p) .... ali..... kako možemo proveriti da li je ovo tačno?

Provera dobijenog rezultata:

- Kao i kada nešto računate na digitronu, niste nikada sigurni da li ste dobro otkucali sve brojeve i računske operacije. I ovde treba proveriti dobijeni rezultatat :)
- Prvo proveriti da li su sve formule dobro unesene, zatim proveriti da li su svuda usklađene jedinice.
- Znači, provera dobijenog rezultata počiva na prethodnom korišćenju papira i olovke, proveri jedinica i konačnog izraza formule koju kasnije samo treba pažljivo uneti u računar:



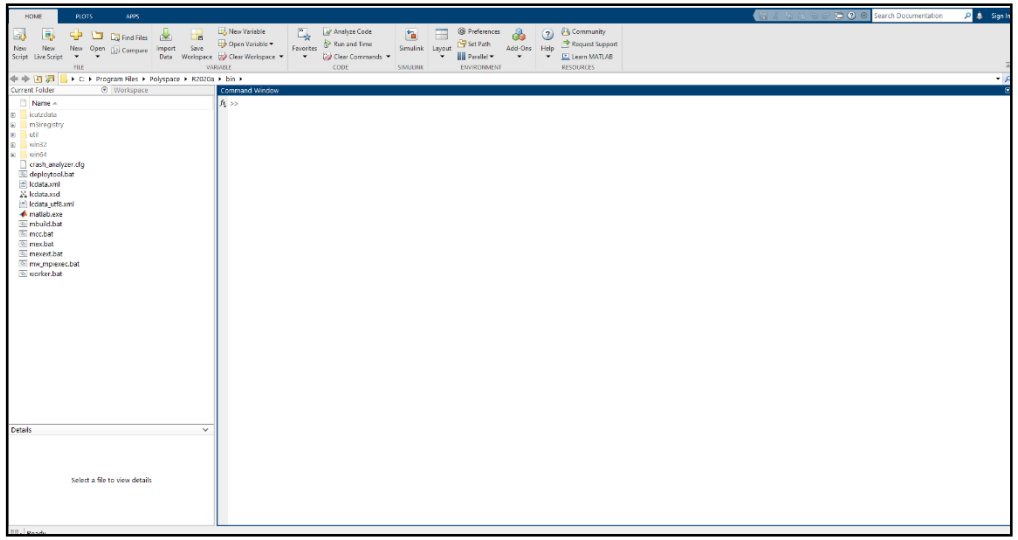
Odgovor: Jeste možda jednom, ali univerzalnost postavke izraza dozvoljava i obrnut račun.

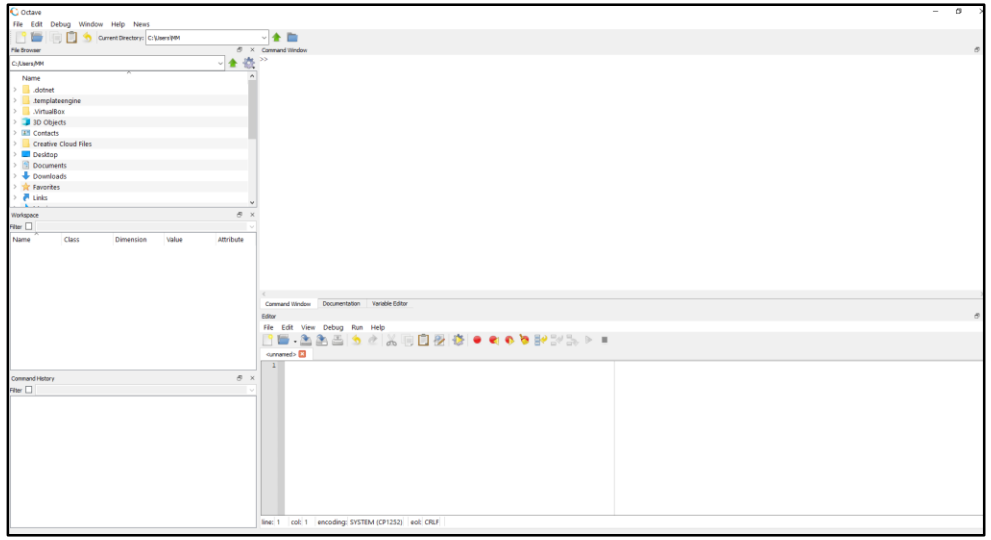
Rešavanje jednačina korišćenjem programa MATLAB:

- Nelinearne jednačine mogu biti rešene i korišćenjem programa **MATLAB**. Prvo šta treba uraditi je definisati problem koji treba rešiti pravljenjem tzv. "m-fajla".
- Sledeći korak je proveriti postavku zadatka a zatim treba pronaći opciju za njegovo rešavanje unutar programa (ovo su koraci analogni onima koji se izvode u Excel-u).
- Ali da bi napravili m-fajl potrebno je znati raditi u komandnom prozoru MATLAB-a, i zato prvo ... **upoznavanje sa MATLAB programom**.
- Počinjemo: Otvoriti **MATLAB,** ići na File-New-M-file. Radna površina se sastoji iz različitih prozora. Direktorijum u kome se radi prikazan je u gornjem meniju.
- Ici na Desktop-Desktop Layout-Default da bi dobili standardni meni u radu sa programom.
- Na ekranu se mogu prepoznati sledeći osnovni prozori:

  Current Directory (koji se može promeniti u Wokspace prozor), Command History, Command Window i Editor.

Upoznavanje sa radnim okruženjem:





Matlab

GNU Octave