

1. SVOJ PROJEKT:

- razmisliti o izboru sustava za svoj projekt koji ćete računati sami, nešto jednostavno, protein bez kofaktora i prostetičkih skupina (može i iznimka, ali je to onda puno teže), potražiti pdb od tog proteina, **PITANJE NA KOJE HOĆETE NAĆI ODGOVOR** – poslati do nedjelje u ponoć
- pokušajte to započeti slijedeći tjedan dok smo Natalia i ja tu

2. „JOURNAL CLUB” U SRIJEDU 25.2. POPODNE (publikaciju ćete svi dobiti u petak) – SRIJEDU U 10 SU VJEŽBE ILI IZRADA VAŠIH PROJEKATA.

3. ISPIT sredina 3. mjeseca? razmisliti – fiksirati na Journal clubu

**RAČUNALNE (SIMULACIJSKE) METODE U
ISTRAŽIVANJU BIOPOLIMERA (PROTEINA I
NUKLEINSKIH KISELINA)**

Kolegij:

Računalna biofizika



Table 1: Examples of levels of modeling in computational biochemistry and molecular biology.

Methods	Degrees of freedom	Properties, processes	Time scale
quantum dynamics	atoms, nuclei, electrons	excited states, relaxation, reaction dynamics	picoseconds
quantum mechanics (ab initio, density functional, semiempirical, valence bond methods)	atoms, nuclei, electrons	ground and excited states, reaction mechanisms	no time scale
classical statistical mechanics (MD, MC, force fields)	atoms, solvent	ensembles, averages, system properties, folding	nanoseconds
statistical methods (database analysis)	groups of atoms, amino acid residues, bases	structural homology and similarity	no time scale
continuum methods (hydrodynamics and electrostatics)	electrical continuum, velocity continuum etc.	rheological properties	supramolecular
kinetic equations	populations of species	population dynamics, signal transduction	macroscopic

Podijela računalnih metoda

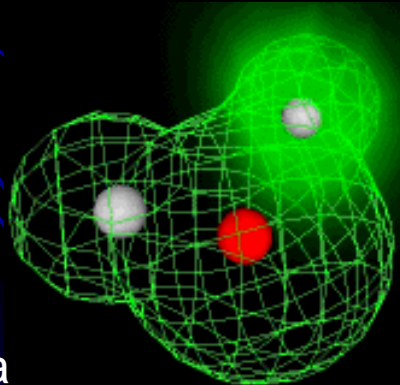
kvantno-mehaničke
metode

empirijske metode

$$H\Psi = E\Psi$$

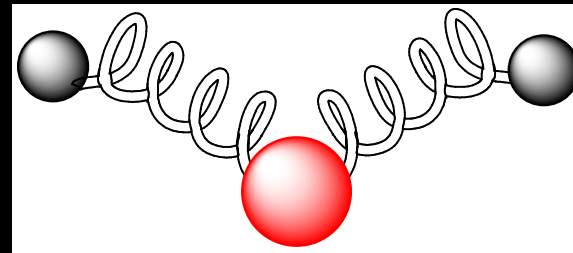
polyG - (Gly)₁₀₀ – 706 atoma

kvantna mehanika



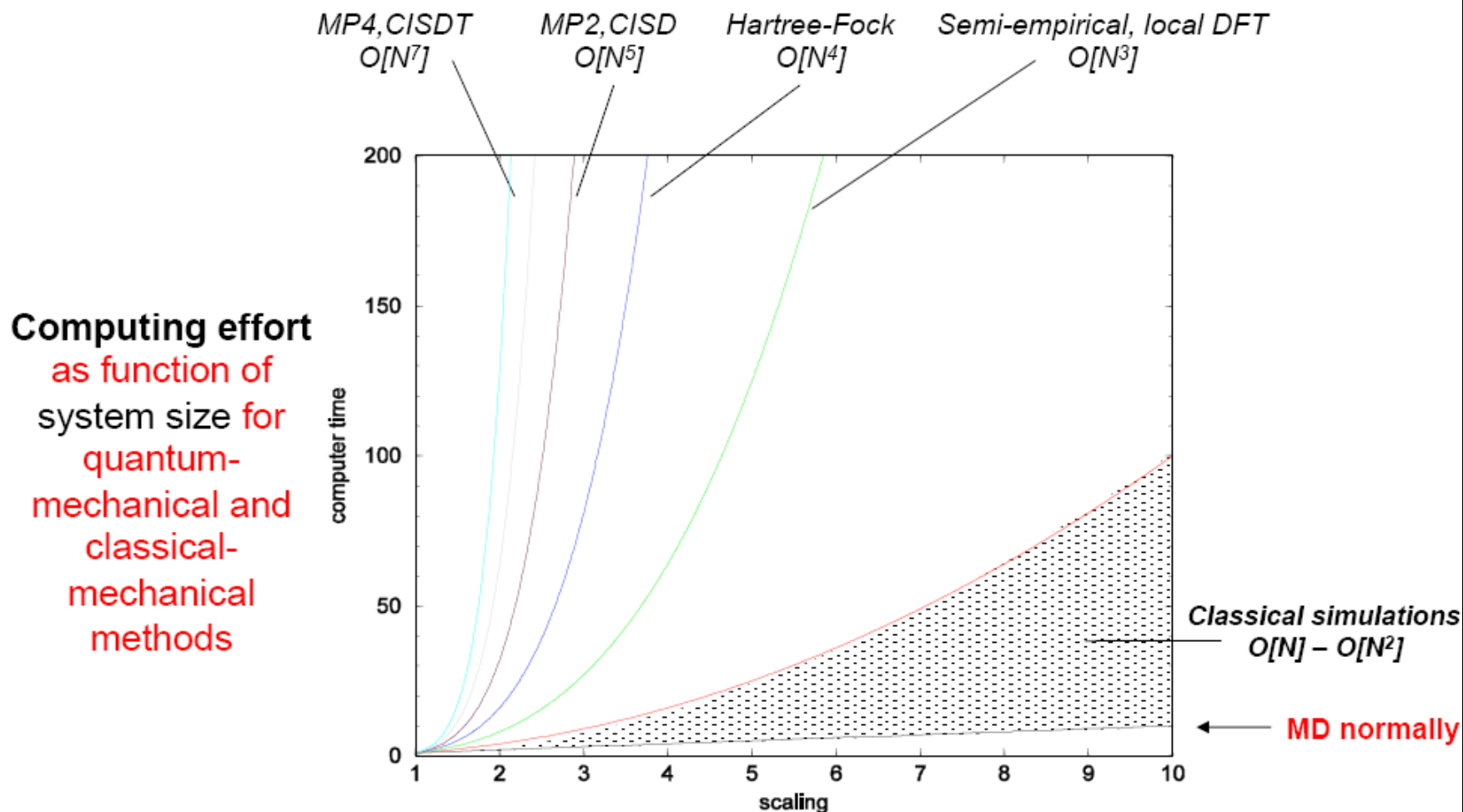
- cijepanje veza

molekulska mehanika



Podijela računalnih metoda

Quantum or classical treatment ?



1. A full quantum description only possible for small systems (100s atoms).
2. For **thermodynamic** properties of large systems quantum effects are generally negligible.

Podijela računalnih metoda

The many-particle problem

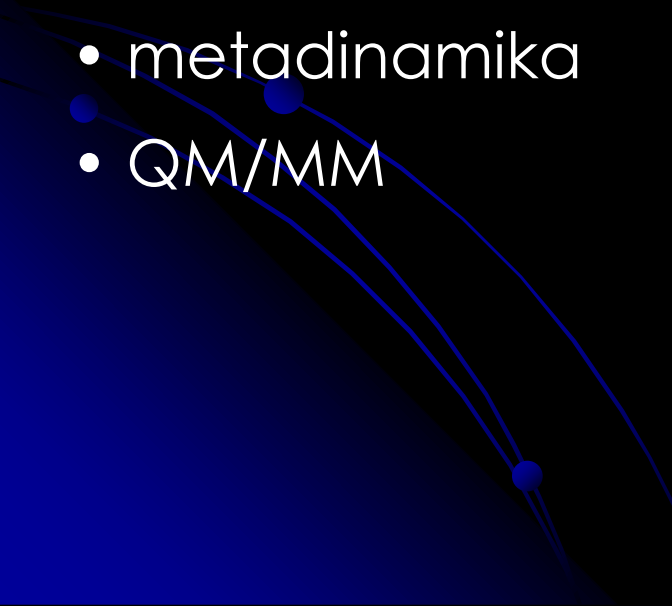
	Crystalline State	Liquid State Macromolecules	Gas Phase
Quantum Mechanics	possible	very difficult	possible
Classical Mechanics	easy	essential many particle problem: simulation	trivial

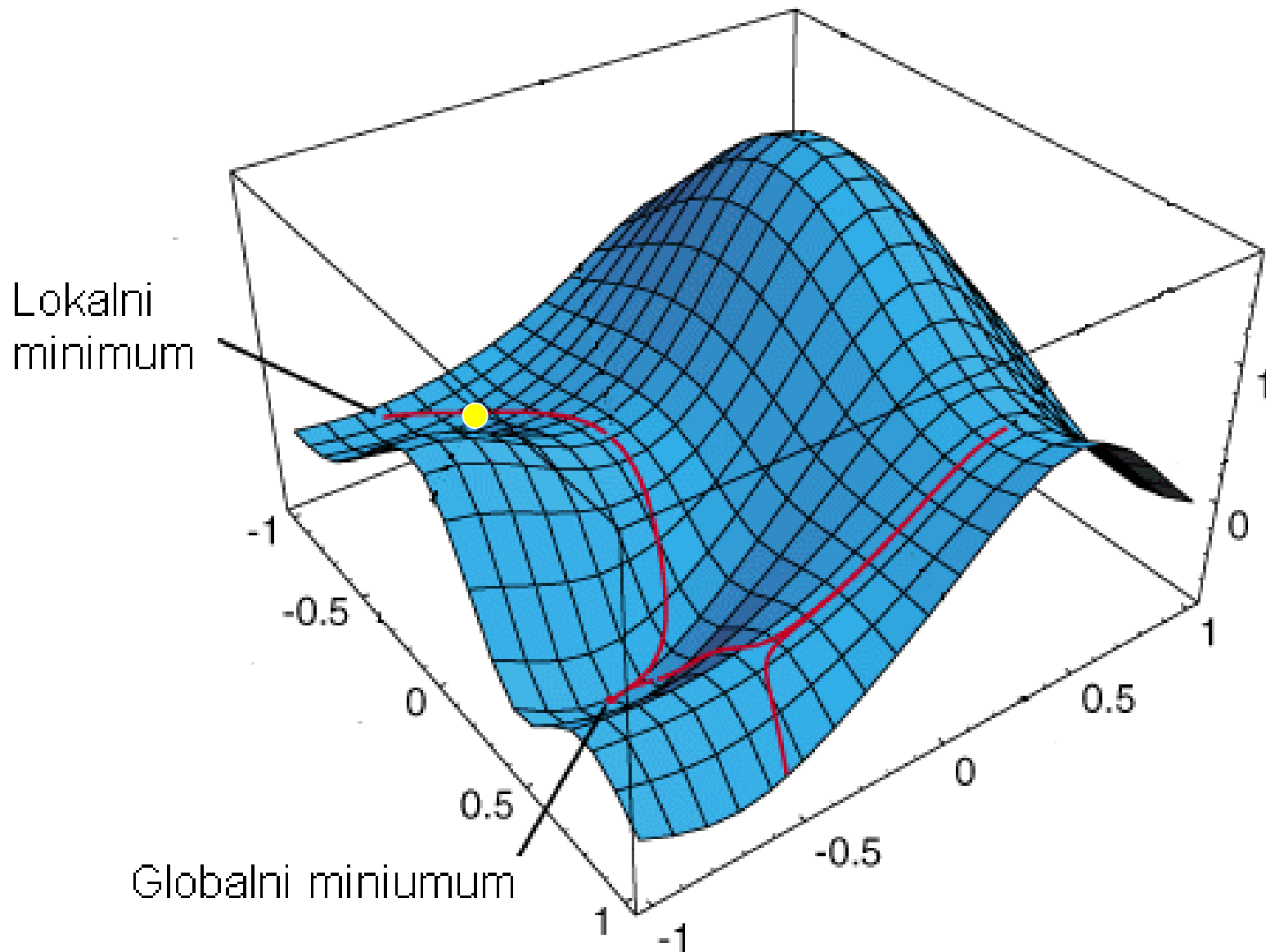
↑
Reduced to just a
few degrees of
freedom by
symmetry

↑
Reduced to just a
few degrees of
freedom by
dilution

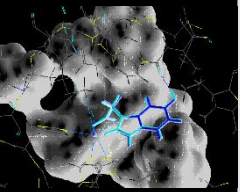
Empirijske metode

- računalne metode temeljene na polju sila:

- molekularna mehanika (MM)
 - molekularna dinamika (MD)
 - Monte Carlo konformacijska pretraga (MC)
 - molekularna dinamika s nasumičnim ubrzanjem (RAMD)
 - metadinamika
 - QM/MM
- 

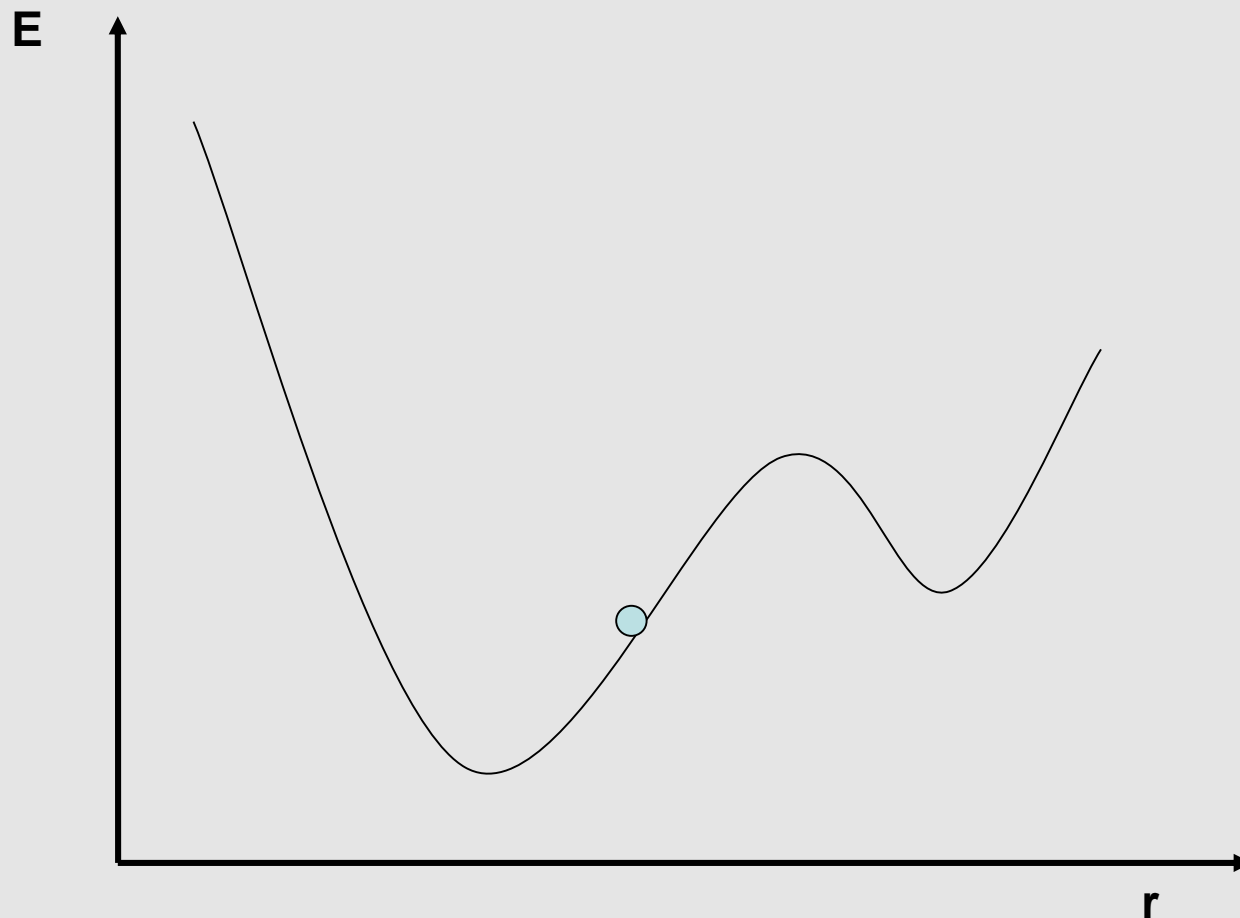


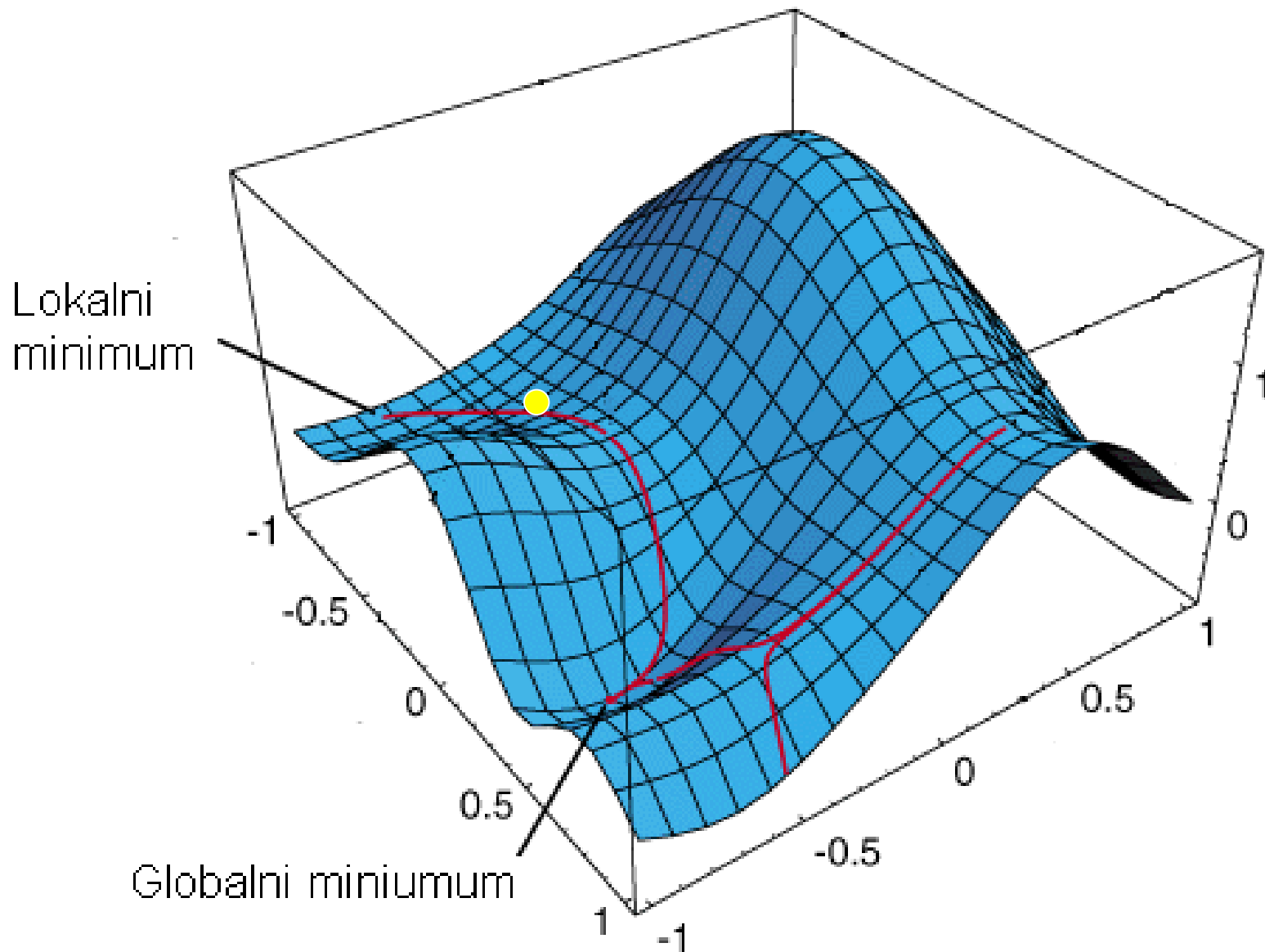
Ploha potencijalne energije molekule u 3D presjeku (prikazana je ovisnost energije o dvije interne koordinate)



MOLEKULSKA MEHANIKA

$$E = E_{stretch} + E_{bend} + E_{tors} + E_{oop} + E_{el} + E_{vdw} + \sum E_{cross}$$





Ploha potencijalne energije molekule u 3D presjeku
(prikazana je ovisnost energije o dvije interne koordinate)

MOLEKULSKA MEHANIKA

$$E = E_{stretch} + E_{bend} + E_{tors} + E_{oop} + E_{el} + E_{vdw} + \sum E_{cross}$$

Tehnike (algoritmi) u optimizaciji geometrije:

- Linijsko pretraživanje (*line search*)
- Metoda najstrmijeg spusta (*steepest descent*)
- Metoda konjugiranih gradijenata (*conjugated gradients*)
- Newton Rapson-ova (*Full matrix NR, Block diagonal NR, BFGS, truncated NR*)
- Metoda traženja sedla

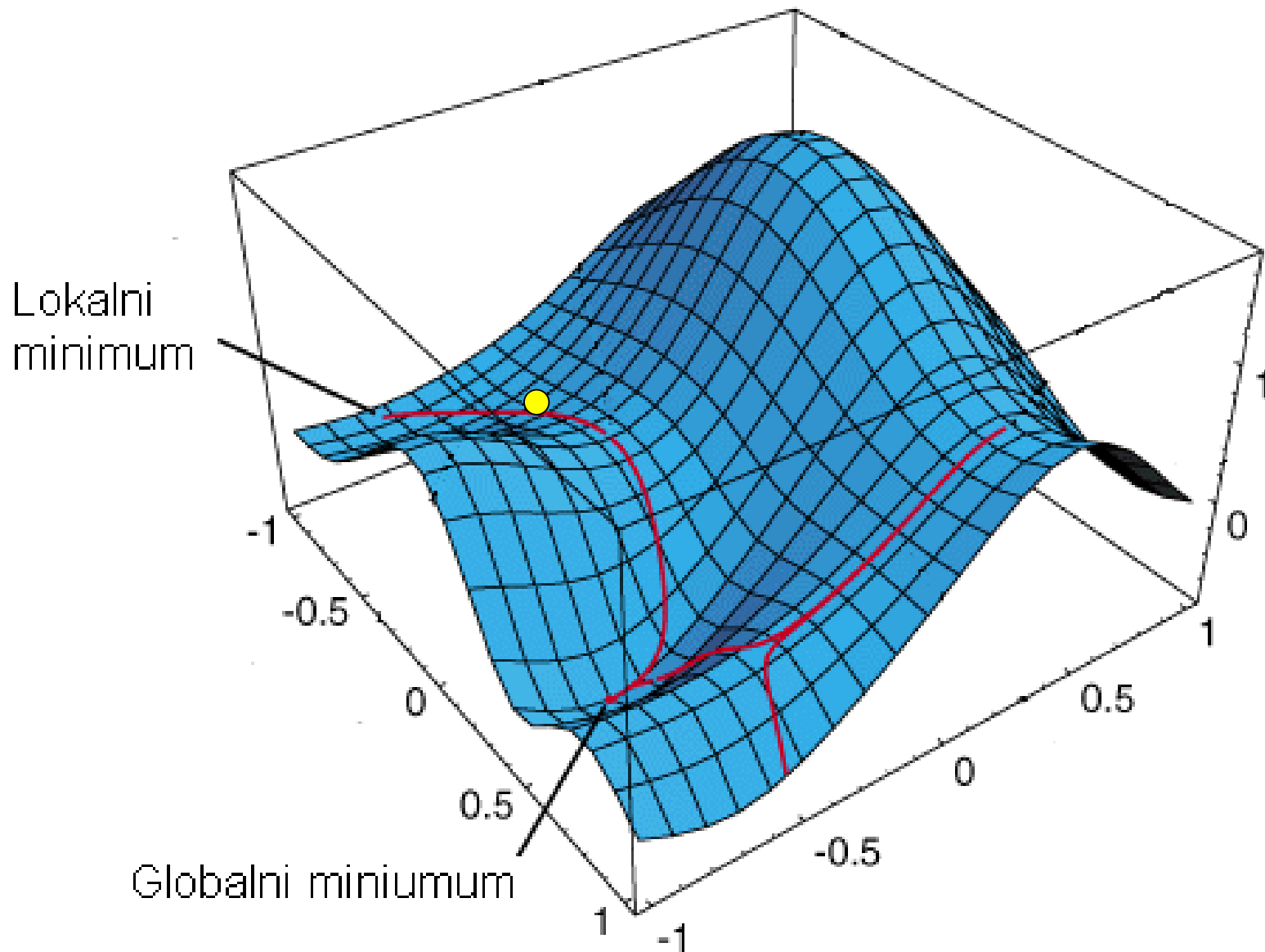
SISTEMATSKO PRETRAŽIVANJE KONFORMACIJE

- za pretraživanje konformacijskog prostora biomakromolekula najčešće se koriste **molekulska dinamika** i/ili **Monte Carlo** metoda

§ Optimizacija

➤ uvod

- **globalni minimum** je najmanja vrijednost funkcije na njenom području definiranosti (*npr. energetski najstabilnija konformacija molekule*)
- **globalni maksimum** je najveća vrijednost funkcije na njenom području definiranosti
- nije moguće konstruirati algoritam koji bi pronalazio globalni ekstrem proizvoljne funkcije
- **lokalni minimum** je minimum funkcije u nekom intervalu koji ne mora, ali može biti globalni minimum
- **lokalni maksimum** je maksimum funkcije u nekom intervalu koji ne mora, ali može biti globalni maksimum



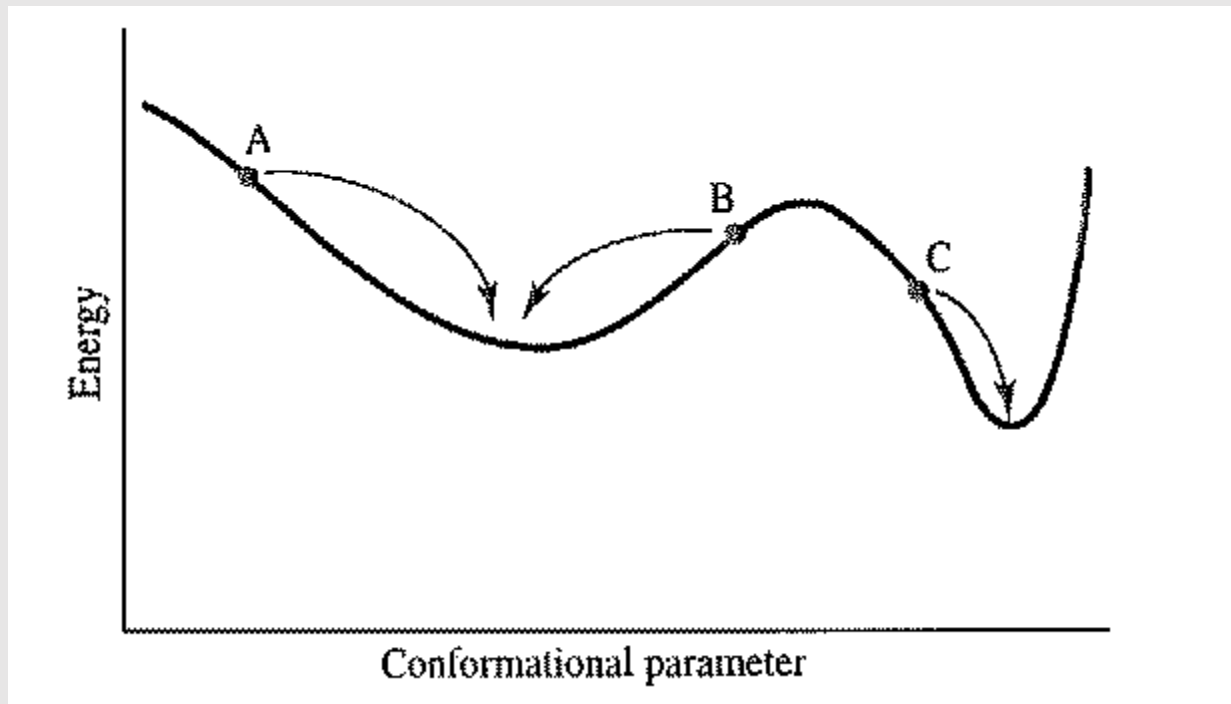
Ploha potencijalne energije molekule u 3D presjeku (prikazana je ovisnost energije o dvije interne koordinate)

§ Optimizacija –METODE OPTIMIZACIJE

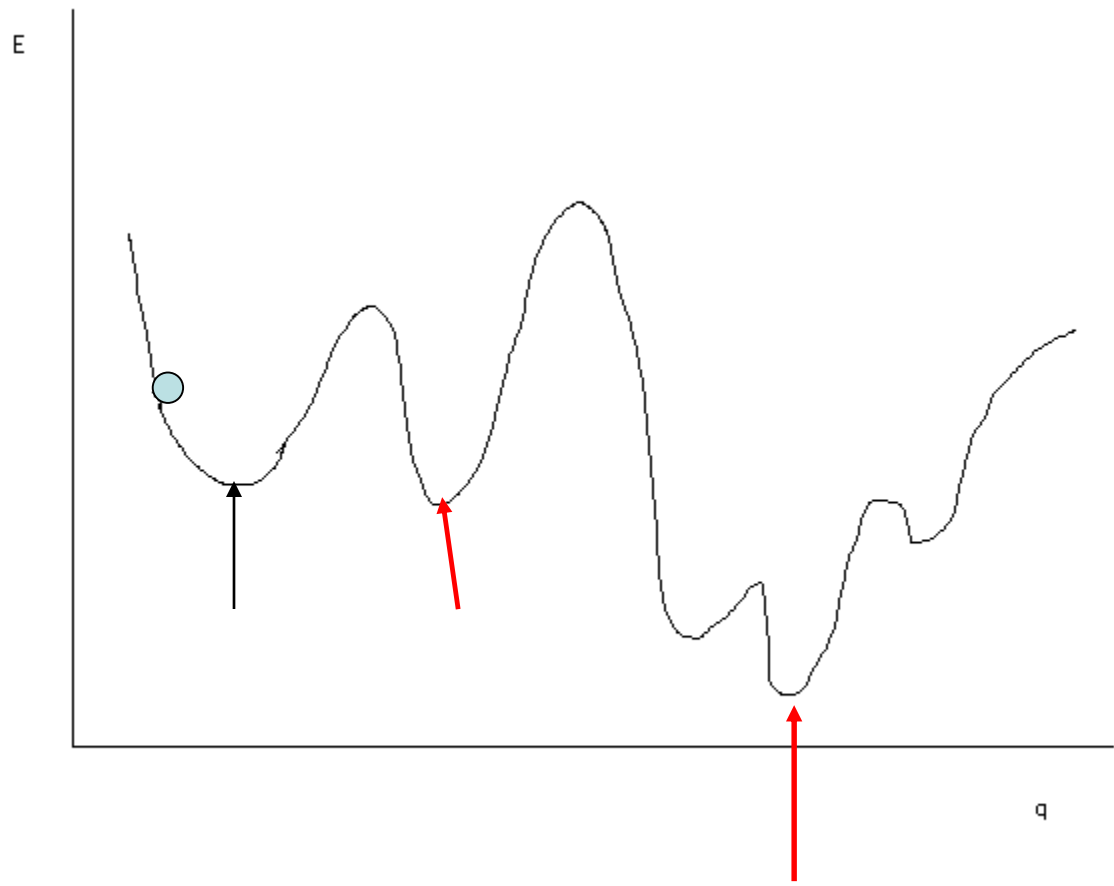
- kod jednostavne analitičke funkcije je jednostavno naći ekstrem
- kod **PPE molekule** koja je funkcija $3N$ koordinata u kartezijevom sustavu, odnosno $3N-6$ internih koordinata, to je jako teško pa koristimo numeričke metode
- treba izabrati najbolju metodu i s obzirom na **brzinu i efikasnost**, ali i s obzirom na **preciznost i točnost**
- **različite metode** imaju prednosti i mane, **niti jedna nije savršena**
- ovisi i o složenosti problema koji rješavamo (*npr. minimizacija proteina i male molekule nije ista*)
- metode koje koriste derivacije su bolje jer daju uvid i u izgled plohe potencijalne energije, ali ponekad nam to i ne treba ili ionako koristimo numeričku derivaciju pa nam taj podatak ne bi bio toliko koristan i radije izaberemo metodu koja ne koristi derivacije
- metoda koja je izvrsna za QM, ne mora biti dobra i za MM, i obrnuto!

§ Optimizacija –METODE OPTIMIZACIJE

- uvijek krećemo od početne točke i postepeno se približavamo minimumu
- minimizacijski algoritmi se kreću samo “nizbrdo” te će nas odvesti u lokalni minimum koji je u susjedstvu početnoj točki
- ukoliko želimo naći više minimuma metodom optimizacije, potrebno nam je više početnih točaka



nije moguće konstruirati algoritam koji bi pronalazio globalni ekstrem proizvoljne funkcije



§ Optimizacija –METODE OPTIMIZACIJE

➤ metode koje ne koriste derivacije funkcije - **simpleks**

- *simpleks metoda* (metoda nultog reda)
- simpleks je geometrijska figura koja ima $N+1$ međusobno spojenih vrhova (N je dimenzionalnost funkcije)
- npr. za funkciju dvije varijable, simpleks će imati tri vrha
- simpleks algoritam nalazi minimum funkcije pomičući se slično gibanju amebe

§ Optimizacija

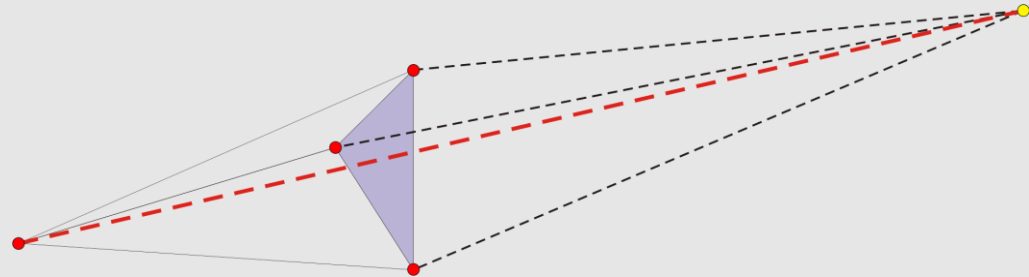
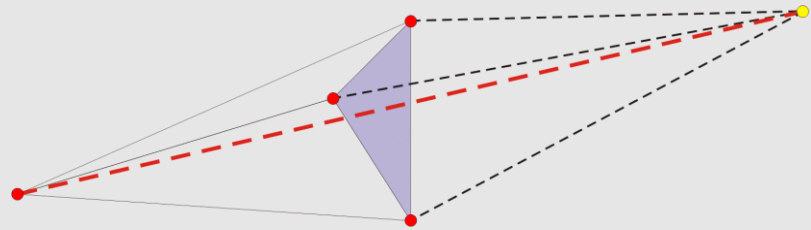
➤ metode koje ne koriste derivacije funkcije - **simpleks**

- postoje tri osnovna načina gibanja simpleksa u pokušaju generiranja nove točke koja ima manju funkcijsku vrijednost
- 1. **refleksija** točke s najvećom funkcijskom vrijednosti kroz nasuprotnu plohu simpleksa, ukoliko nova točka ima manju funkcijsku vrijednost od svih ostalih točaka uz refleksiju se može primijeniti i **ekspanzija**
- 2. **kontrakcija duž jedne dimenzije** iz točke s najvećom funkcijskom vrijednosti
- 3. **kontrakcija u svim smjerovima** oko točke s najmanjom funkcijskom vrijednosti

§ Optimizacija

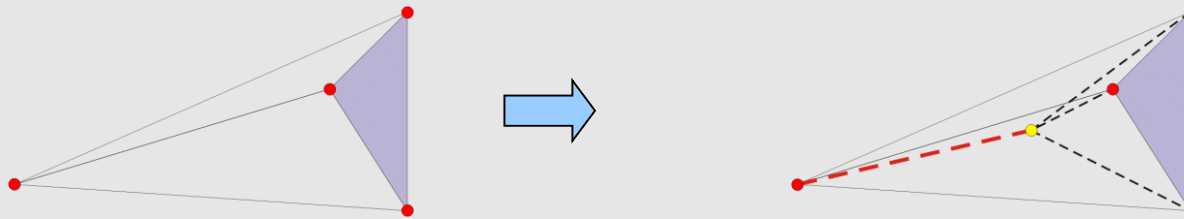
➤ metode koje ne koriste derivacije funkcije - **simpleks**

● *refleksija* i '*refleksija i ekspanzija*'



§ Optimizacija

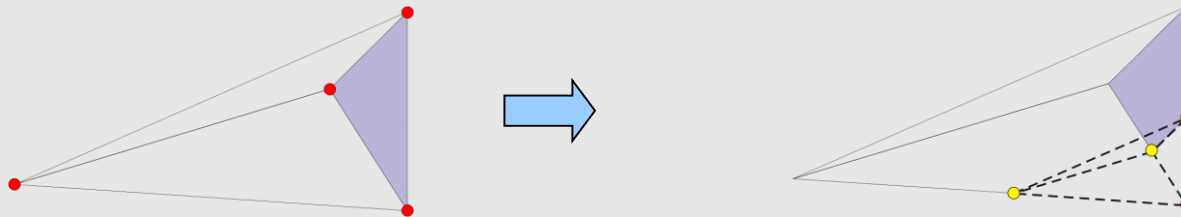
- metode koje ne koriste derivacije funkcije - **simpleks**
 - *kontrakcija duž jedne dimenzije*



§ Optimizacija

➤ metode koje ne koriste derivacije funkcije - **simpleks**

- *kontrakcija u svim smjerovima oko točke s najmanjom funkcijskom vrijednosti*



§ Optimizacija – PRIMJER SIMPLEKSA

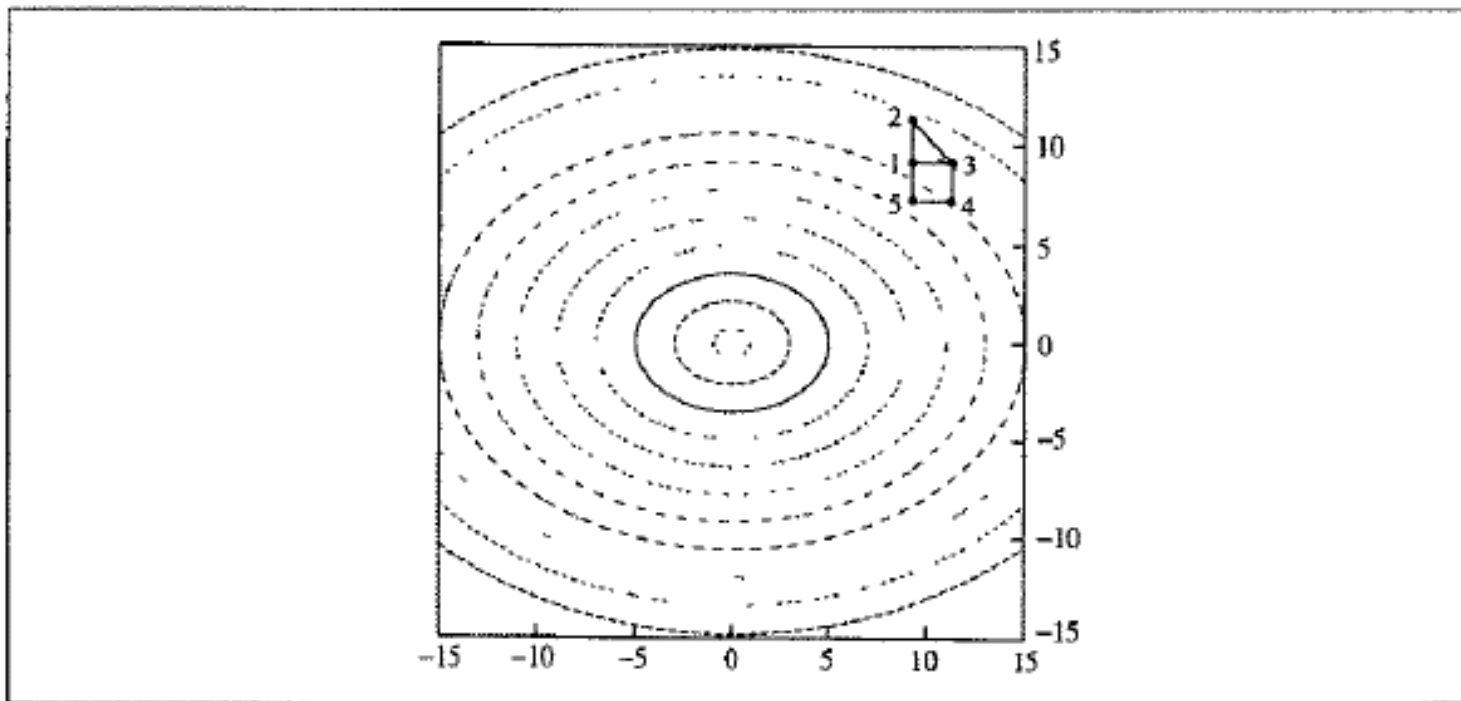


Fig 5.5. The first few steps of the simplex algorithm with the function $x^2 + 2y^2$. The initial simplex corresponds to the triangle 123. Point 2 has the largest value of the function and the next simplex is the triangle 134. The simplex for the third step is 145.

§ Optimizacija

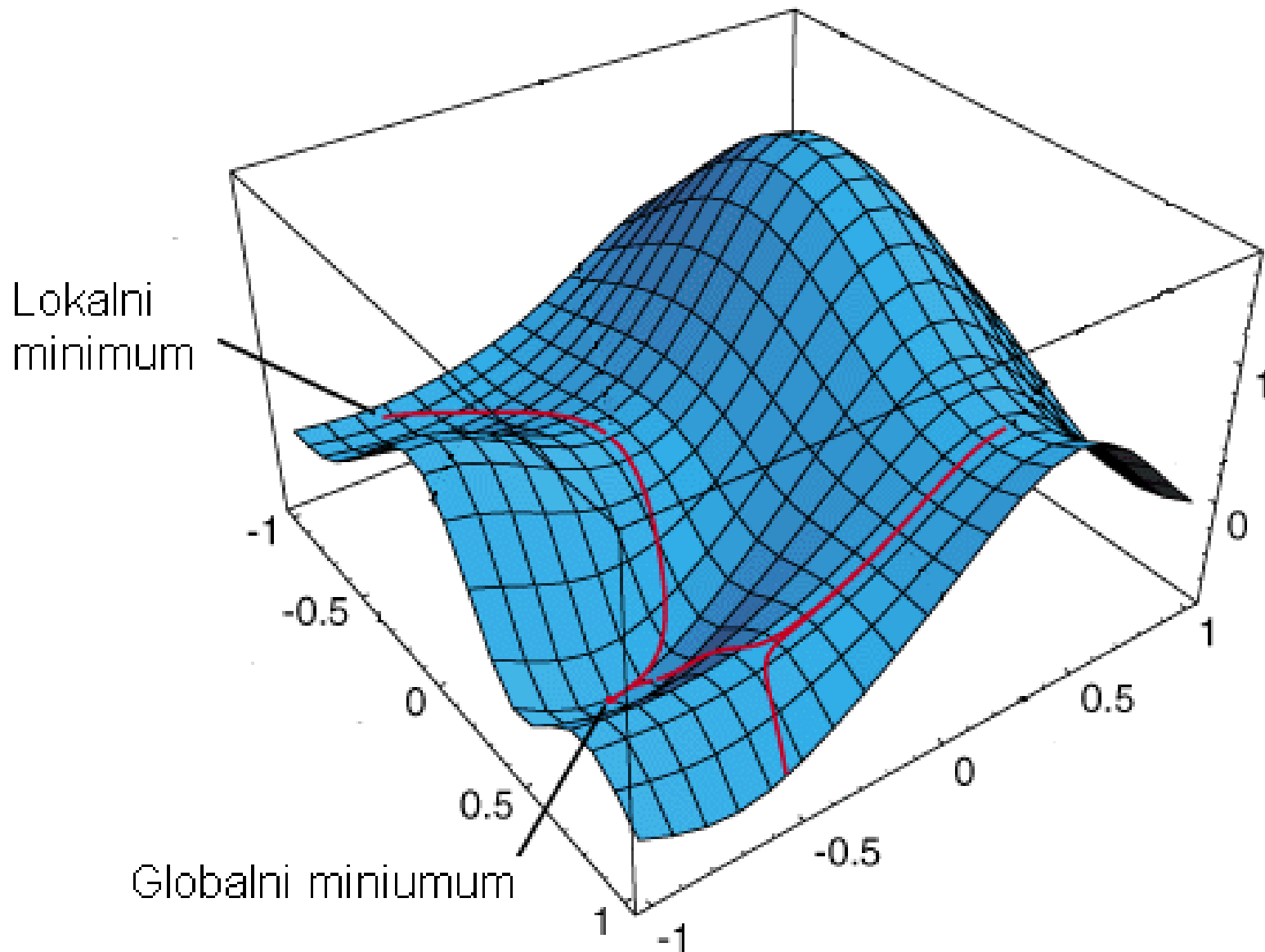
➤ metode koje ne koriste derivacije funkcije - **simpleks**

- za implementaciju simpleks metode prvo je potrebno generirati **točke početnog simpleksa**, početna konfiguracija sustava odgovara jednoj od tih točaka dok se ostale mogu generirati na razne načine, najčešće dodajući konstantni iznos po svakoj koordinati
- simpleks metoda je najkorisnija tamo gdje je **početna funkcijska vrijednost jako velika u odnosu na minimalnu vrijednost**
- metoda može biti prilično **zahtjevna računski** zbog velikog broja izračuna vrijednosti funkcije (samo početni simpleks zahtijeva $N+1$ vrijednosti)
- iz tih razloga simpleks metoda se obično koristi u kombinaciji s drugim minimizacijskim algoritmima - **nakon nekoliko koraka sa simpleks metodom koristi se naprednija metoda optimizacije**

§ Optimizacija

➤ metode koje koriste derivacije funkcije

- objasni gradijent, odnosno funkciju više varijabli i njene derivacije
- smjer prve derivacije funkcije pokazuje gdje se minimum nalazi dok iznos gradijenta pokazuje strmost lokalnog nagiba funkcije
- druga derivacija funkcije pokazuje zakrivljenost funkcije, odnosno informacije koje se mogu iskoristiti za predviđanje promjene smjera funkcije (dakle kada će proći kroz minimum ili maksimum, odnosno stacionarnu točku)
- metode koje koriste derivacije funkcija se klasificiraju prema najvišem redu derivacije koja se koristi (npr. metode prvog reda koriste prve derivacije, dok metode drugog reda koriste i prve i druge derivacije funkcije)



Ploha potencijalne energije molekule u 3D presjeku (prikazana je ovisnost energije o dvije interne koordinate)

§ Optimizacija

➤ metode koje koriste derivacije funkcije

- kod metoda koje koriste derivacije funkcije korisno je funkciju zapisati u obliku Taylorovog reda oko neke točke x_k

$$f(x) = f(x_k) + (x - x_k) f'(x_k) + \frac{(x - x_k)^2}{2!} f''(x_k) + \dots$$

- za funkcije više varijabli, varijabla x se zamjenjuje vektorom x dok se za derivacije koriste matrice!
- npr. ako je potencijalna energija $f(x)$ funkcija $3N$ Cartesiusovih koordinata, tada vektor x ima $3N$ komponenti, a x_k odgovara trenutnoj konfiguraciji sustava (geometriji molekule)
- $f'(x_k)$ je gradijent, matrica dimenzija $3N \times 1$ s elementima $\frac{\partial f}{\partial x_i}$ odnosno vektor koji se označava se s g_k , a elementi su mu derivacije funkcije po varijablama

§ Optimizacija

metode koje koriste derivacije funkcije

- $f'' \mathbf{x}_k$ je matrica dimenzije $3N \times 3N$, a njeni elementi su druge parcijalne derivacije funkcije energije s obzirom na koordinate x_i i x_j

$$\frac{\partial^2 f}{\partial x_i \partial x_j}$$

- ta matrica drugih derivacija naziva se **Hessian** ili matrica konstanti sila (engl. *force constant matrix*)
- Taylorov red za višedimenzijjski slučaj se može zapisati kao

$$f(\mathbf{x}) = f(\mathbf{x}_k) + (\mathbf{x} - \mathbf{x}_k)^T \mathbf{f}'(\mathbf{x}_k) + \frac{1}{2!} (\mathbf{x} - \mathbf{x}_k)^T \mathbf{f}''(\mathbf{x}_k) (\mathbf{x} - \mathbf{x}_k) + \dots$$

§ Optimizacija

➤ metode koje koriste derivacije funkcije

- funkcije koje se koriste u modeliranju rijetko su kvadratne funkcije te ekspanzija u Taylorov red i zadržavanje prva tri člana predstavlja **aproksimaciju**, ta aproksimacija je to bolja što smo bliže minimumu (*harmonijski i anharmonijski oscilator*)
- posljedica te aproksimacije su slijedeći mogući problemi:
- **PROBLEM 1.** minimizacija dane funkcije neće biti tako dobra kao što bi bila minimizacija kvadratne funkcije - **u slučaju kvadratne funkcije metode optimizacije drugog reda daju rješenje u jednom koraku, kod modeliranja potrebna je iteracija!**
- **PROBLEM 2.** ako smo daleko od minimuma, onda **harmonična aproksimacija** puno slabije vrijedi i metoda drugog reda bi mogla imati ozbiljnih problema u nalaženju minimuma te se preporuča korištenje **robusnije metode**

§ Optimizacija

➤ metode prvog reda - metoda najstrmijeg spusta

- engl. *steepest descent method* ili *gradient descent*
- u metodi najstrmijeg spusta nove koordinate se dobivaju pomakom u smjeru paralelnom gradijentu funkcije što je analogno hodanju (padanju) niz strminu
- za funkciju energije koja ovisi o $3N$ Cartesiusovih koordinata **smjer** se reprezentira $3N$ dimenzijskim jediničnim vektorom \mathbf{s}_k

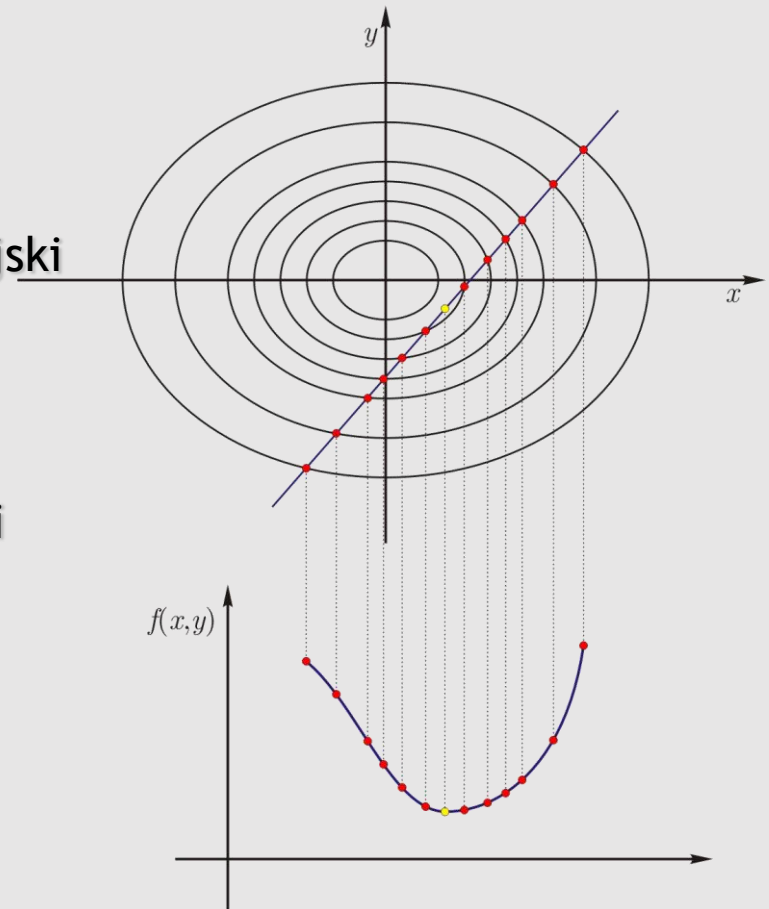
$$\mathbf{s}_k = -\frac{\mathbf{g}_k}{|\mathbf{g}_k|}$$

- nakon što je definiran smjer u kojem će doći do pomaka, potrebno je odrediti koliki će biti iznos pomaka
- iznos pomaka može biti *proizvoljan* ili je moguće provesti *linijsko pretraživanje*

§ Optimizacija

➤ metode prvog reda - **metoda najstrmijeg spusta** - linijsko pretraživanje

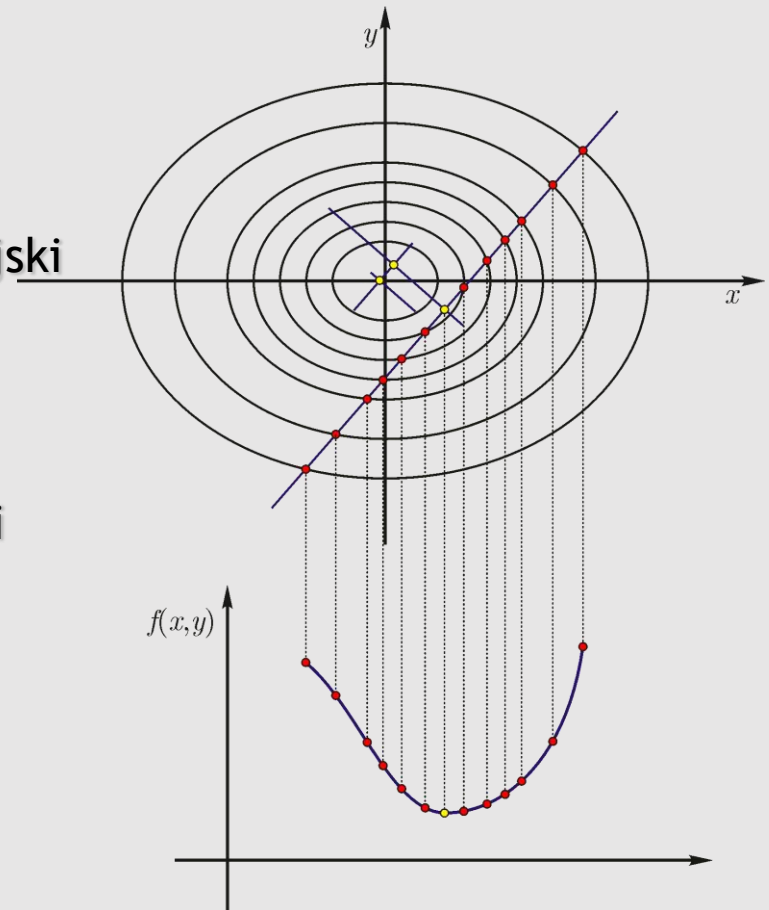
- smjer gradijenta od početne točke je duž plave linije
- donja slika predstavlja jednodimenzijски presjek prikazane površine
- potrebno je locirati minimum u tom jednodimenzijском presjeku i onda pogledati smjer gradijenta u toj točki minimuma



§ Optimizacija

☛ metode prvog reda - **metoda najstrmijeg spusta** - linijsko pretraživanje

- smjer gradijenta od početne točke je duž plave linije
- donja slika predstavlja jednodimenzijski presjek prikazane površine
- potrebno je locirati minimum u tom jednodimenzijskom presjeku i onda pogledati smjer gradijenta u toj točki minimuma



§ Optimizacija

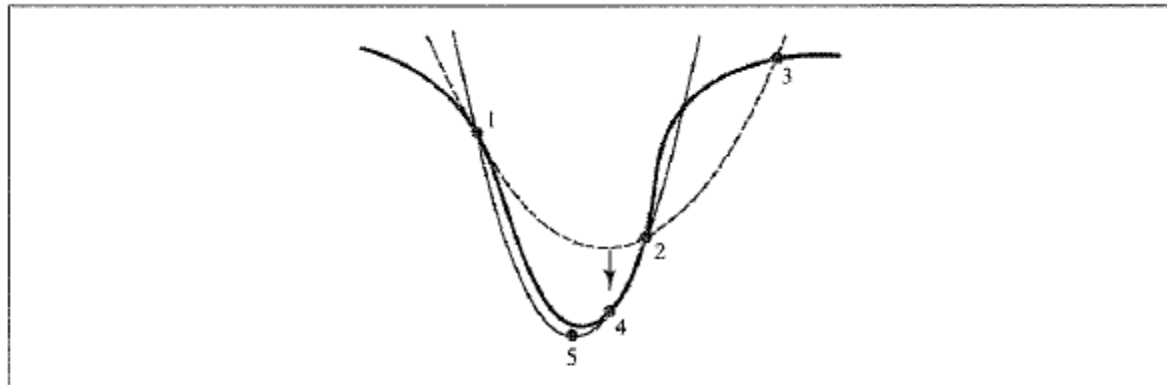
➤ metoda najstrmijeg spusta - linijsko pretraživanje

- svrha linijskog pretraživanja je pronalaženje minimuma duž određenog smjera tj. duž linije presjeka u višedimenzijskom prostoru
- prvo je potrebno **izolirati minimum** u nekom segmentu zadane funkcije za što je potrebno pronaći **tri točke** duž linije takve da je vrijednost funkcije u središnjoj točki manja od vrijednosti u vanjskim točkama
- **ukoliko postoje tri takve točke, tada između dviju vanjskih točaka mora postojati barem jedan minimum funkcije**
- za pronalazak minimuma mogu se iskoristiti iteracijski postupci

§ Optimizacija

➤ metoda najstrmijeg spusta - linijsko pretraživanje

- linijsko pretraživanje je konceptualno lagan postupak, ali zahtijeva značajan broj izračuna funkcijskih vrijednosti
- alternativa je interpolacija kvadratne funkcije kroz tri zadane točke, diferenciranje te kvadratne funkcije (koje je moguće provesti analitički) daje aproksimacija minimuma duž linijskog presjeka
- minimum daje novu točku pa je opet moguće interpolirati kvadratičnu funkciju kroz tri nove točke
- **gradijent u točki minimuma dobivenoj linijskim pretraživanjem će biti okomit na prethodni smjer gradijenta**



§ Optimizacija

➤ metoda najstrmijeg spusta - pristup proizvoljnog koraka

- obično se koristi ako je poznato da će linijsko pretraživanje računalo biti prezahtjevno
- nove koordinate se dobivaju pomicanjem duž jediničnog vektora gradijenta za **proizvoljan korak** λ_k

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \lambda_k \mathbf{s}_k$$

- u većini aplikacija koje koriste metodu najstrmijeg spusta, λ_k ima predodređenu vrijednost; ukoliko prva iteracija vodi **smanjenju energije**, λ_k se **povećava** množenjem s nekom konstantom vrijednošću, npr. 1,1
- ukoliko sljedeći korak **poveća energiju**, λ_k se **reducira** množenjem s npr. 0,9

§ Optimizacija

➤ metode prvog reda - **metoda najstrmijeg spusta**

-prednosti: **robustna metoda**, izvrsna da nas približi minimumu

- nedostaci: u neposrednoj blizini minimuma može “plesati”, kad je oštar minimum potrebno je puno malih koraka, ...

§ Optimizacija

➤ metode prvog reda - **metoda konjugiranog gradijenta**

- engl. *conjugate gradients method*
- producira skup smjerova koji ne pokazuju oscilacijsko ponašanje kao metoda najstrmijeg spusta kod površina s blagim nagibom
- **u metodi najstrmijeg spusta** su i gradijenti i smjerovi uzastopnih koraka ortogonalni, dok su **u metodi konjugiranog gradijenta** gradijenti uzastopnih koraka ortogonalni, a smjerovi uzastopnih koraka konjugirani
- skup konjugiranih smjerova uzastopnih koraka ima svojstvo da će za kvadratnu funkciju od N varijabli, minimum biti dosegnut u N koraka

§ Optimizacija

➤ metode prvog reda - metoda konjugiranog gradijenta

- u metodi konjugiranog gradijenta koordinate se pomiču u **smjeru** \mathbf{v}_k koji se računa iz **trenutnog gradijanta** i **prethodnog vektora smjera** \mathbf{v}_{k-1}

$$\mathbf{v}_k = -\mathbf{g}_k + \gamma_k \mathbf{v}_{k-1}$$

- γ_k je skalarna konstanta definirana sljedećim izrazom

$$\gamma_k = \frac{\mathbf{g}_k \cdot \mathbf{g}_k}{\mathbf{g}_{k-1} \cdot \mathbf{g}_{k-1}}$$

- ove jednadžbe mogu se koristiti od **drugog iteracijskog koraka**; prvi iteracijski korak jednak je kao i u metodi najstrmijeg spusta

➤ metode prvog reda - metoda konjugiranog gradijenta

- prednost u odnosu na SD je što **ne oscilira** kada se nađe u blizini oštrog minimuma
- postoji **puno modifikacija** CG metode
- izneseni algoritam je originalni **Fletcher-Reevesov algoritam**
- često se koristi **Polak-Ribierov algoritam** kod kojeg se skalarna konstanta računa na nešto malo drugačiji način što u nekim slučajevima rezultira efikasnijom optimizacijom

§ Optimizacija

➤ metode drugog reda - **Newton-Raphsonova metoda**

- metode drugog reda koriste ne samo prve derivacije funkcije (gradijente) već i druge derivacije funkcije
- Newton-Raphsonova metoda je najjednostavnija metoda drugog reda
- Taylorov red oko točke x_k je

$$f(x) = f(x_k) + (x - x_k) f'(x_k) + \frac{(x - x_k)^2}{2!} f''(x_k) + \dots$$

$$f'(x) = f'(x_k) + (x - x_k) f''(x_k)$$

- ukoliko je funkcija kvadratna, tada će druga derivacija funkcije biti jednaka u svakoj točki, odnosno vrijedi

$$f''(x) = f''(x_k)$$

§ Optimizacija

➤ metode drugog reda - **Newton-Raphsonova metoda**

- u točki minimuma $x = x_{\min}$ prva derivacija funkcije će biti jednaka nuli

$$f'(x_{\min}) = 0$$
$$x_{\min} = x_k - \frac{f'(x_k)}{f''(x_k)}$$

- za višedimenzijску funkciju taj izraz postaje

$$\mathbf{x}_{\min} = \mathbf{x}_k - \mathbf{f}''^{-1}(\mathbf{x}_k) \mathbf{f}'(\mathbf{x}_k)$$

- \mathbf{f}''^{-1} je inverzna matrica hessiana čije računanje može biti zahtjevno ukoliko sustav ima mnogo atoma
- Newton-Raphsonova metoda je prikladnija za funkcije s manje varijabli (npr. izračun energije za sustav s manje od 100 atoma)

☛ metode drugog reda - **Newton-Raphsonova metoda**

- za kvadratnu funkciju nalazi minimum u jednom koraku
- u praksi je potrebna iteracija jer funkcija nije kvadratna, odnosno to je aproksimacija
- kada smo **daleko od minimuma**, onda aproksimacija da je funkcija kvadratna je jako loša te **metoda postane nestabilna** (pogotovo ako Hessian nije pozitivan!)

➤ metode drugog reda - kvazi Newton-Raphsonove metode

- s obzirom da je ponekad računalno zahtjevno računati matricu drugih derivacija u svakom koraku optimizacije, napravljene su **kvazi** N.-R. metode kod kojih se taj korak nastoji reducirati
- jedna od mogućnosti jest da se ista matrica drugih derivacija koristi za nekoliko sukcesivnih koraka
- kod nekih se u jednom koraku pomiče samo jedan atom te se tako Hessian sukcesivno gradi tijekom optimizacije

§ Optimizacija

➤ iteracijski kriterij

- u većini slučajeva, optimizacija bi trajala “zauvijek” beskonačno se približavajući vrijednosti minimuma (ili maksimuma)
- stoga moramo nametnuti iteracijski kriterij koji određuje jesmo li “dovoljno” blizu minimuma (ili maksimuma) te možemo li zaustaviti iteraciju
- često korišteni iteracijski kriteriji: **energija, promjena u koordinatama, RMS gradijenta**

$$\text{RMS} = \sqrt{\frac{\mathbf{g}^T \mathbf{g}}{3N}}$$

§ Optimizacija

➤ odabir metode

- ne postoji savršena metoda
- odabir metode određuje veliki broj faktora, neki od njih su:
- koliko daleko od minimuma se nalazi početna konformacija
- koliko precizno želimo odrediti minimum
- koliko je velik sustav koji istražujemo
- koliko računalnog vremena smo spremni potrošiti
- ...
- općenito, našeće je najbolje kombinirati više metoda na način da se najprije krene s robustnijim metodama, a zatim se primjenjuju sofisticiranije metode