Funkcijų optimizavimas

Temoje aiškinama:

- **Optimizavimo uždavinio pagrindinės sąvokos;**
- **Funkcijos gradientas 2D ir jo geometrinė prasmė;**
- **Gradiento kryptimi paremtas funkcijos minimizavimas;**
- **Daugelio kintamųjų funkcijos minimizavimas;**
- **Gradiento projekcijos metodas;**
- **Gradiento skaitinis įvertis. Kvazi-gradiento apskaičiavimas Broideno metodu;**
- **Funkcijos minimizavimu paremti netiesinių lygčių sistemų sprendimo metodai**
- **Kvazi-gradiento metodų taikymas NLS funkcijos minimizavimui**

Optimizavimo uždavinio pagrindinės sąvokos

- *Optimizavimas* **reiškia geriausio, mus pilnai tenkinančio sprendinio radimą iš galimų sprendinių aibės;**

- **Dažnai optimizavimo uždavinį galima suformuluoti kaip tam tikros funkcijos mažiausios reikšmės ir tą reikšmę atitinkančių funkcijos argumentų reikšmių radimą. Tokio uždavinio sprendimo eiga vadinama funkcijos** ***minimizavimu***

Funkcijos minimizavimo uždavinys

Skaliarinė <u>*tikslo funkcija*</u>

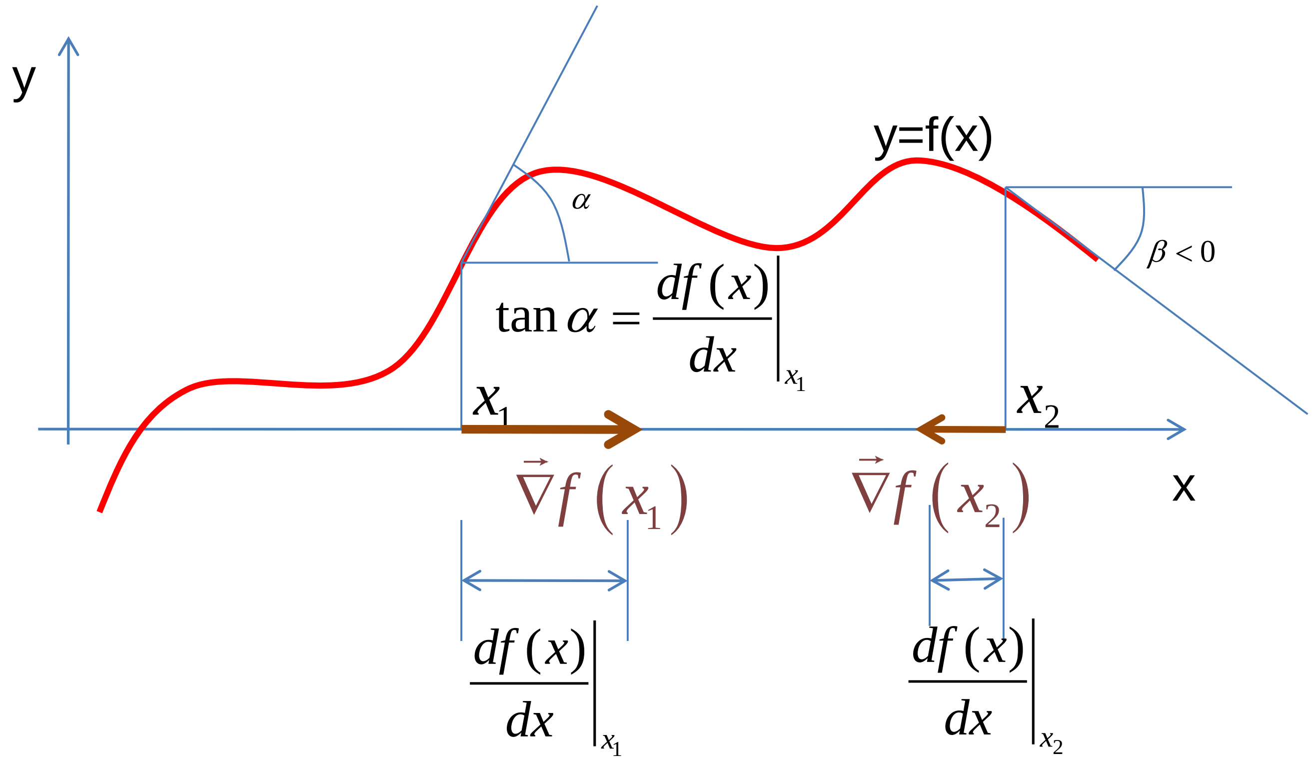
$$\min_{\{\mathbf{x}\}} \Psi(\mathbf{x})$$

Vektorinis argumentas

- **Funkcijos Ψ minimumas yra *<u>mažiausia jos reikšmė iš visų galimų</u>*;**
- **Funkcijos minimizavimo uždavinys yra rasti argumentų vektorių *x*, kuriam esant tikslo funkcijos reikšmė yra minimali;**
- **Jeigu visos x reikšmės yra leistinos (t.y. prasmingos sprendžiamo uždavinio atveju), turime *besąlyginio optimizavimo uždavinį.* Jeigu tam tikra reikšmių aibė yra neleistina, turime *sąlyginio optimizavimo uždavinį;***
- **Jį spręsime, taikydami *gradiento metodus.* Nustatome, kaip reikia pakeisti funkcijos argumentų reikšmes, kad funkcijos reikšmė sumažėtų. Minimumo ieškome nuosekliais artiniais (iteracijomis);**

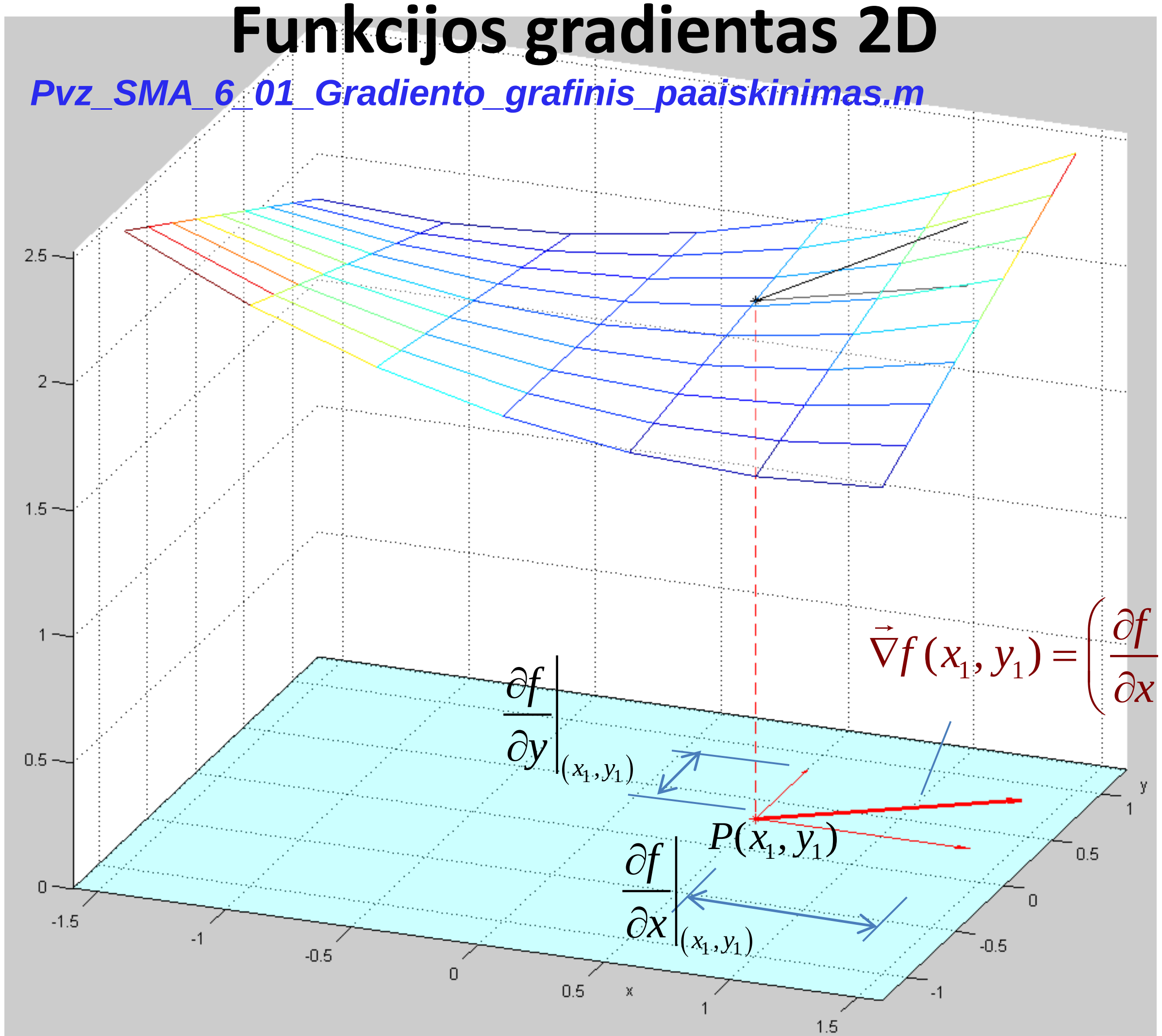
Funkcijos gradientas 2D ir jo geometrinė prasmė

Funkcijos gradientas 1D



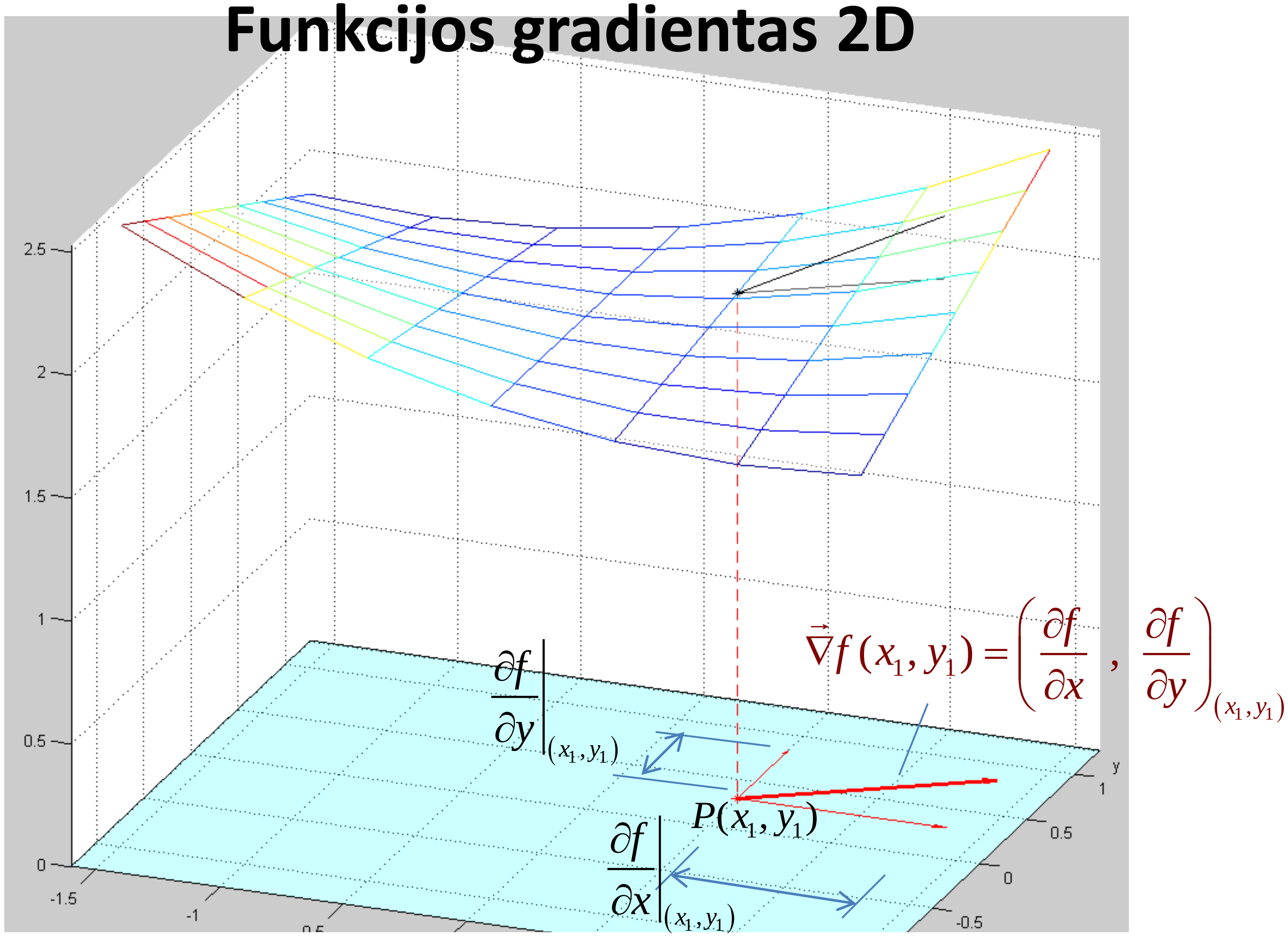
- Funkcijos gradientas yra vektorius, pavaizduojantis tam tikrame taške apskaičiuotą funkcijos išvestinę;
- Šiame pavyzdyje gradiento vektorius yra vienmatis, t.y. jis aprašomas viena projekcija;
- Taške apskaičiuoto gradiento vektoriaus kryptis parodo, kuria kryptimi reikia pakeisti argumento reikšmę, kad funkcijos reikšmė padidėtų. Vektoriaus ilgis parodo funkcijos kitimo spartą tame taške

Funkcijos gradientas 2D



- Dviejų kintamųjų funkcijos gradientas yra vektorius, plokštumoje xOy, pavaizduojantis tam tikrame taške apskaičiuotas funkcijos dalines išvestines;
- Taške apskaičiuoto gradiento vektoriaus kryptis parodo, kuria kryptimi reikia pakeisti argumento reikšmę, kad funkcijos reikšmė didėtų sparčiausiai. Vektoriaus ilgis parodo funkcijos kitimo spartą tame taške

Funkcijos gradientas 2D



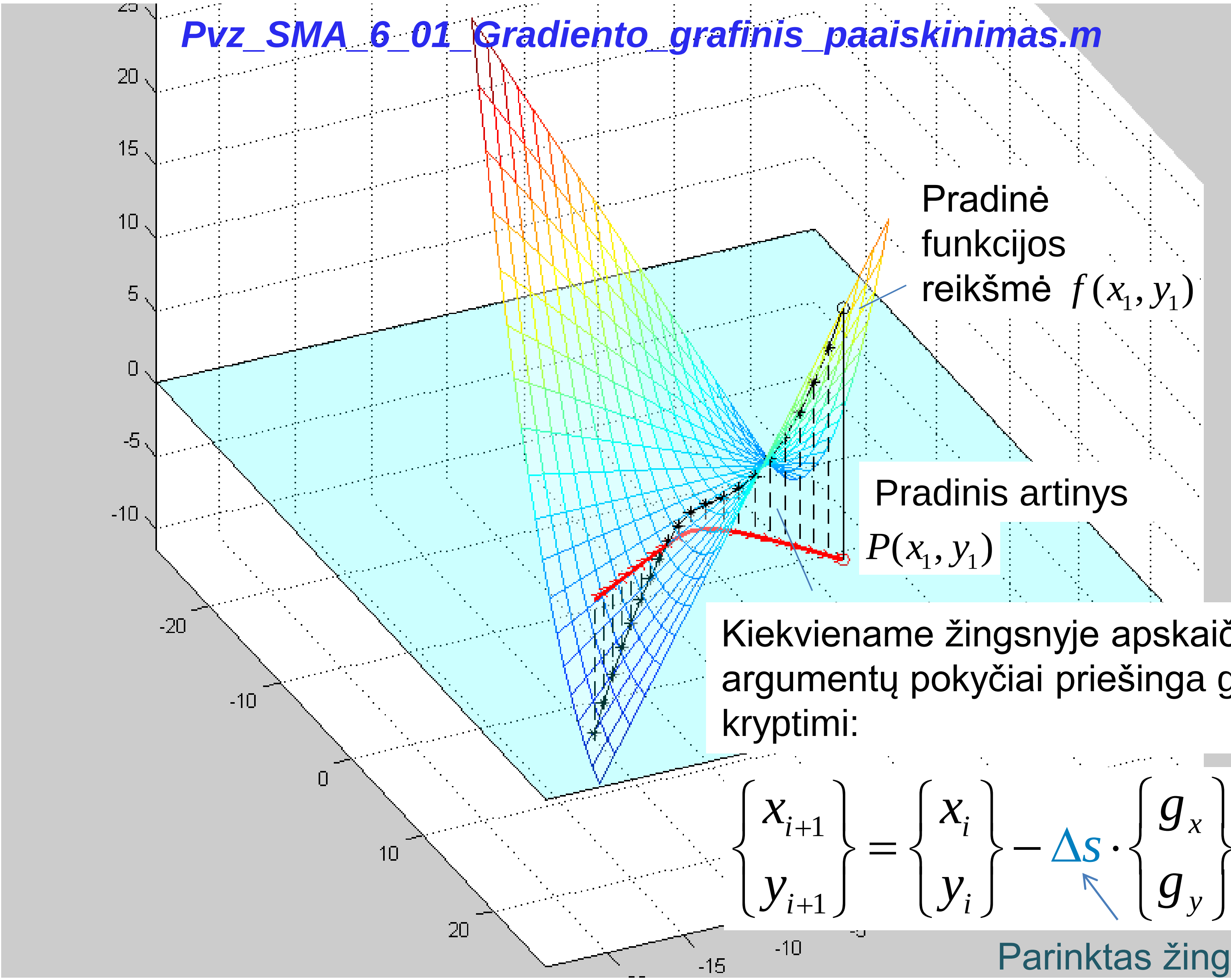
•**Gradiento vektoriaus ilgis *matuojamas skirtingais vienetais, nei funkcijos ir jos argumentų reikšmės. Todėl funkcijos ir jos gradiento vaizdavimo masteliai tame pačiame brėžinyje yra skirtingi;***

•***Funkcijos sparčiausio kitimo krypčiai nusakyti naudojamas vienetinis gradiento vektorius***

$$\vec{\mathbf{g}} = \vec{\nabla} f(x_1, y_1) \Big/ \left\| \vec{\nabla} f(x_1, y_1) \right\|$$

Gradiento kryptimi paremtas funkcijos minimizavimas

Dviejų argumentų funkcijos minimizavimas priešinga gradientui kryptimi



Kiekviename žingsnyje apskaičiuojami argumentų pokyčiai priešinga gradientui kryptimi:

$$\begin{Bmatrix} x_{i+1} \\ y_{i+1} \end{Bmatrix} = \begin{Bmatrix} x_i \\ y_i \end{Bmatrix} - \Delta s \cdot \begin{Bmatrix} g_x \\ g_y \end{Bmatrix}$$

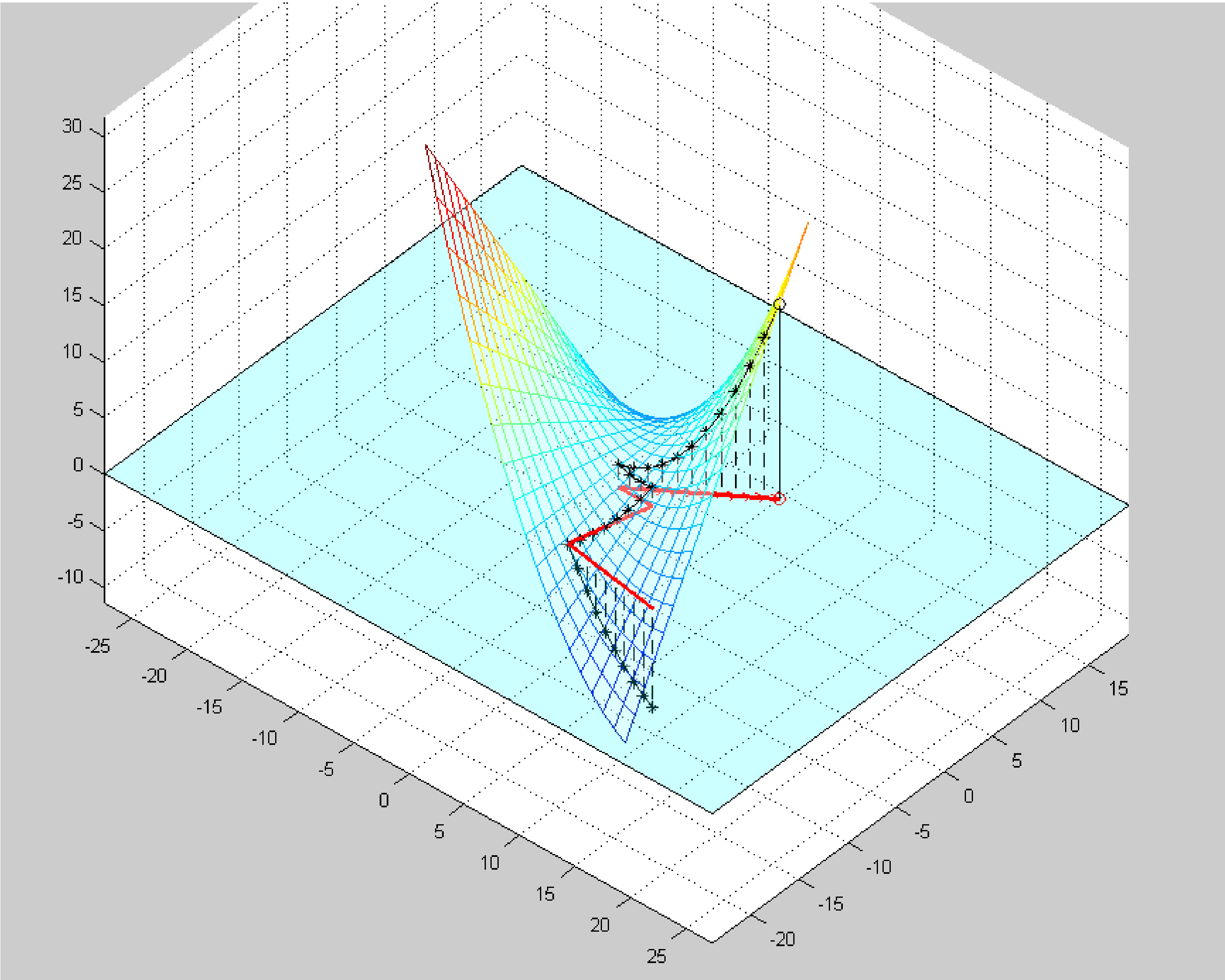
Parinktas žingsnio ilgis

Greičiausio nusileidimo metodas

- **Minimizuojant *priešinga gradientui kryptimi*, gradiento vektorių tenka apskaičiuoti kiekviename žingsnyje. Tai užima nemažai skaičiavimo laiko;**
- **Racionaliau taikyti *greičiausio nusileidimo metodą.* Apskaičiavus gradiento vektorių, jam priešinga kryptimi einama tol, kol funkcija tolydžio mažėja;**
- **Funkcijai pradėjus vėl didėti, naujai apskaičuojame gradiento vektorių ir toliau minimizuojame priešinga jam kryptimi;**

Dviejų argumentų funkcijos minimizavimas greičiausio nusileidimo metodu

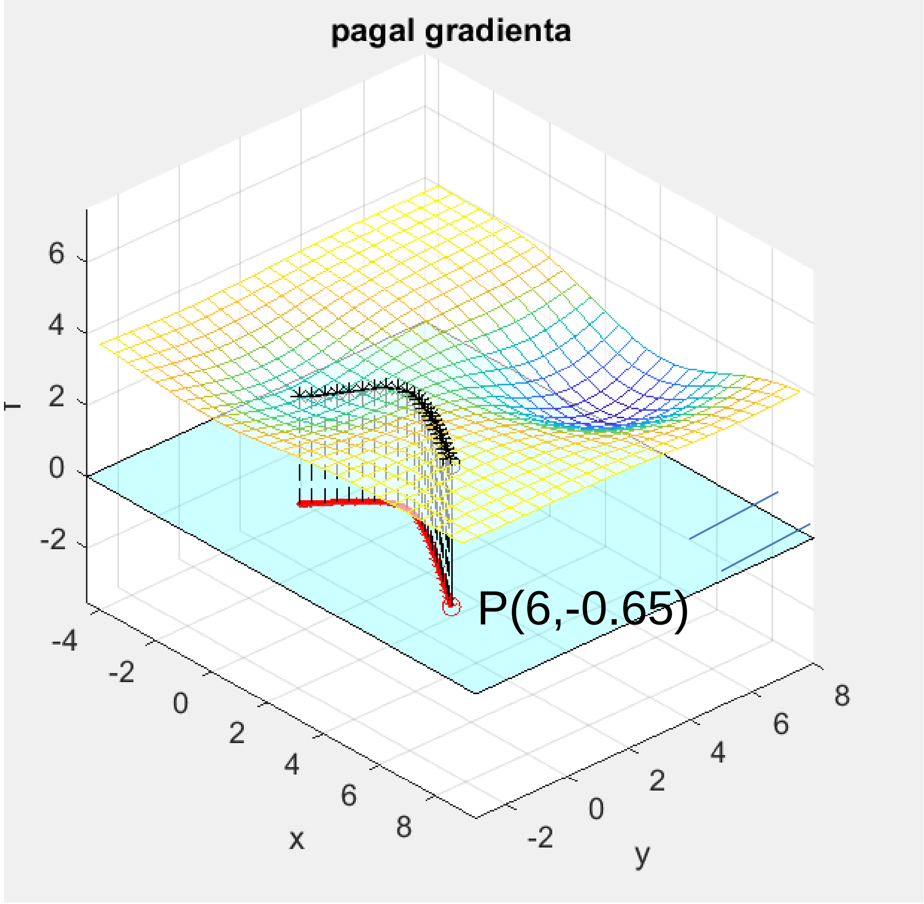
Pvz_SMA_6_01_Gradiento_grafinis_paaiskinimas.m

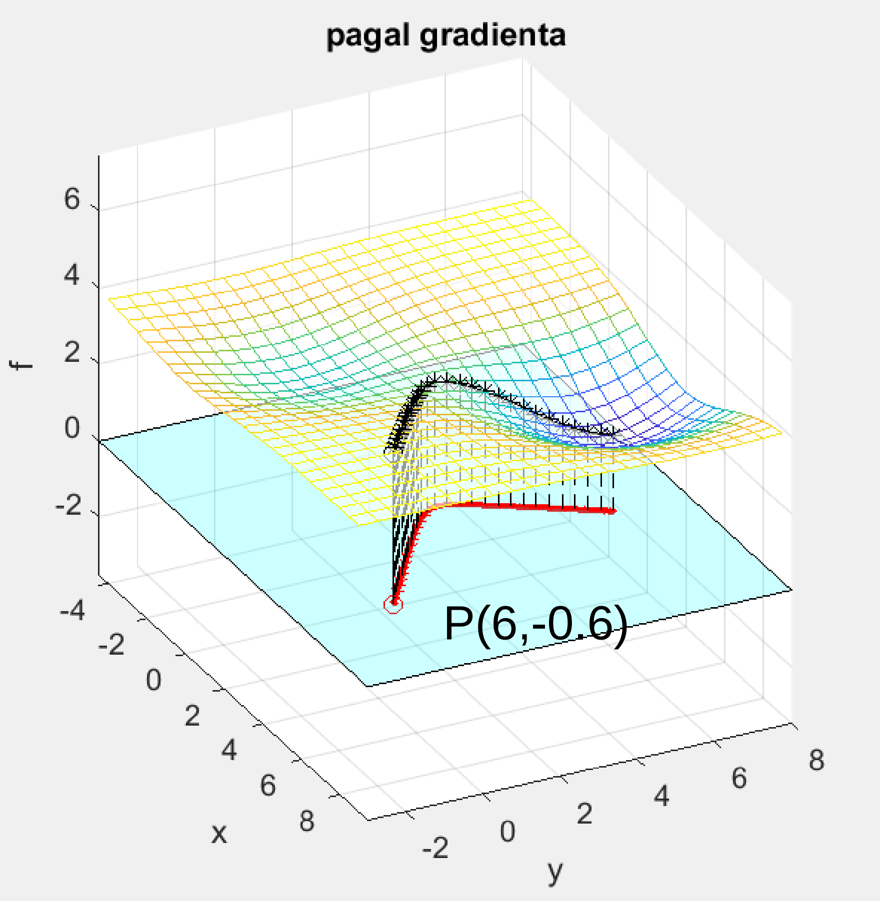


Lokalieji minimumai

Pvz_SMA_6_02_Gradientas_lokalus.m

- Artiniams patekus į ***lokalaus minimumo*** zoną, gradientiniai metodai nebegali rasti kitų minimumų, nors juose funkcijos reikšmė gali būti ir dar mažesnė. Juos galima rasti, tik parinkus kitokius pradinius artinius;
- Egzistuoja *globaliojo optimizavimo metodai*, įgalinantys sistemiškai ieškoti globalaus funkcijos minimumo



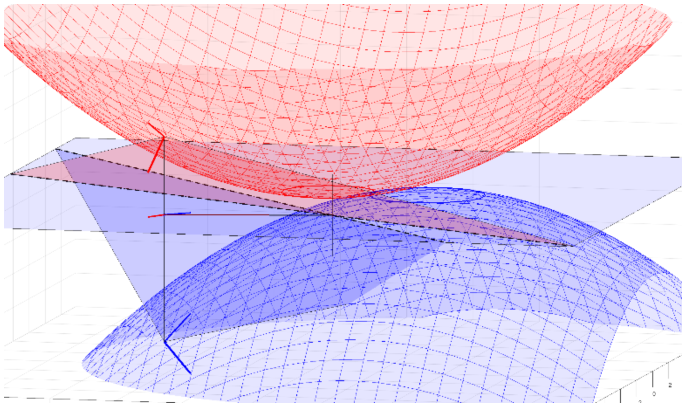


Žingsnio valdymas, minimizuojant gradientiniais metodais

- Minimizavimo žingsnis Δs yra parenkamas empiriškai, todėl jo pradinis dydis gali būti ir ne visai racionalus arba netgi netinkamas;
- Žingsnio dydį reikia koreguoti, vykstant minimizavimui:
  - Jeigu skaičiuojant vis naujus artinius funkcijos reikšmė mažėja, galima bandyti žingsnį padidinti;
  - Jeigu apskaičiavus naują artinį funkcijos reikšmė padidėja, reiškia funkcijos minimumo taškas jau pereitas. Reikia grįžti į ankstesnįjį artinį, sumažinti žingsnį ir tęsti minimumo paiešką;
  - Šiuos veiksmus galima kartoti;
  - Minimumo taškas identifikuojamas, kai žingsnis sumažėja iki pasirinkto mažo dydžio

Kaip siejasi lygčių sistemų sprendimas Niutono metodu su minimizavimu pagal gradientą

Pvz_SMA_6_03_Niutono_metodas_2_LS_gradientai.m



Jakobio matricos eilutėse įrašyti lygčių sistemos funkcijų *gradientų vektoriai*

$$\begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \cdots & \frac{\partial f_n}{\partial x_n} \end{bmatrix}_{\mathbf{x}} \begin{Bmatrix} \Delta x_1 \\ \Delta x_2 \\ \vdots \\ \Delta x_n \end{Bmatrix} = -\begin{Bmatrix} f_1(\mathbf{x}) \\ f_2(\mathbf{x}) \\ \vdots \\ f_n(\mathbf{x}) \end{Bmatrix}$$

- Niutono metodu atliekamas netiesinių lygčių sistemos sprendimas *nėra tapatus minimizavimui pagal gradientą*, nors sąsajų yra;
- Sprendžiant lygčių sistemą Niutono metodu, funkcijų gradientai, esantys Jakobio matricos eilutėse, įgalina apskaičiuoti kiekvienos LS funkcijos normalių vektorius artinio taške pagal formulę $\left(\frac{\partial f_i}{\partial x_1}, \quad \frac{\partial f_i}{\partial x_2}, \quad -1 \right)$.

  Pagal normalių vektorius gaunamos liestinės plokštumos, kurių susikirtimo su nulio plokštuma tiesių susikirtimo taškas yra sekantis artinys;
- Minimizuojant funkciją, jos argumentų prieaugio vektoriaus kryptis tiesiogiai nustatoma pagal gradientą;
- Šie skirtumai kyla iš to, kad sprendžiant LS tikslas yra *artėti link*

Daugelio kintamųjų funkcijos minimizavimas

Daugelio kintamųjų funkcijos minimizavimas (1)

$$\vec{\nabla} f(x_1, x_2, \ldots, x_n) = \left\{ \frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \ldots, \frac{\partial f}{\partial x_n} \right\}$$

$$\vec{\mathbf{g}} = \vec{\nabla} f \Big/ \left\| \vec{\nabla} f \right\|$$

$$\mathbf{x}^{i+1} = \mathbf{x}^i - \Delta s \cdot [\mathbf{g}]^T$$

Gradientas yra vektorius-eilutė. Toks užrašas dera su Jakobio matricos užrašu

Parinktas žingsnio “ilgis”(norma) argumentų erdvėje

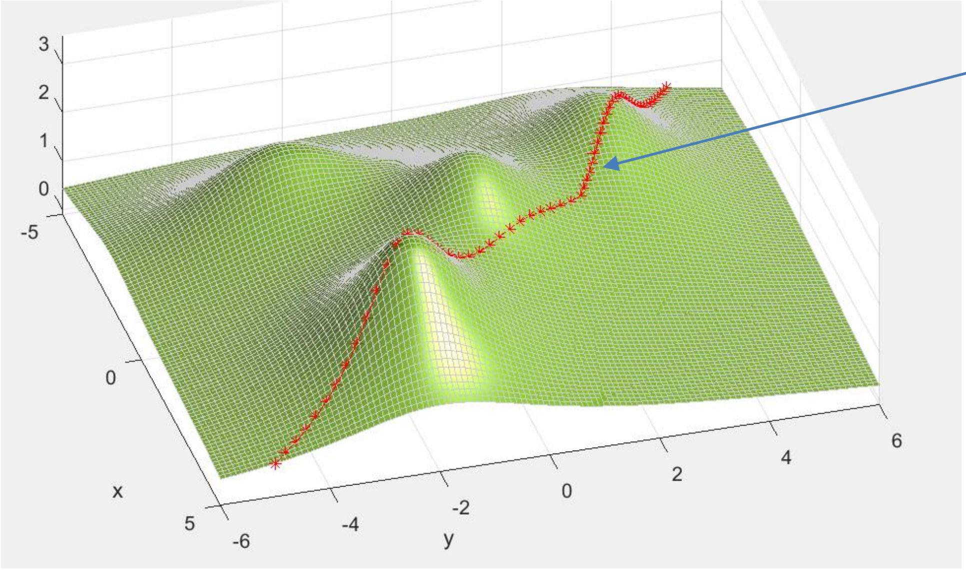
Daugelio kintamųjų funkcijos atveju taikomi tokie patys minimizavimo algoritmai, kaip ir dviejų kintamųjų atveju. Tačiau sprendimo procesą grafiškai pavaizduoti yra keblu.

Daugelio kintamųjų funkcijos minimizavimas (2)

Pvz_SMA_6_05_Gradiento_taikymas_optimizavimui.m

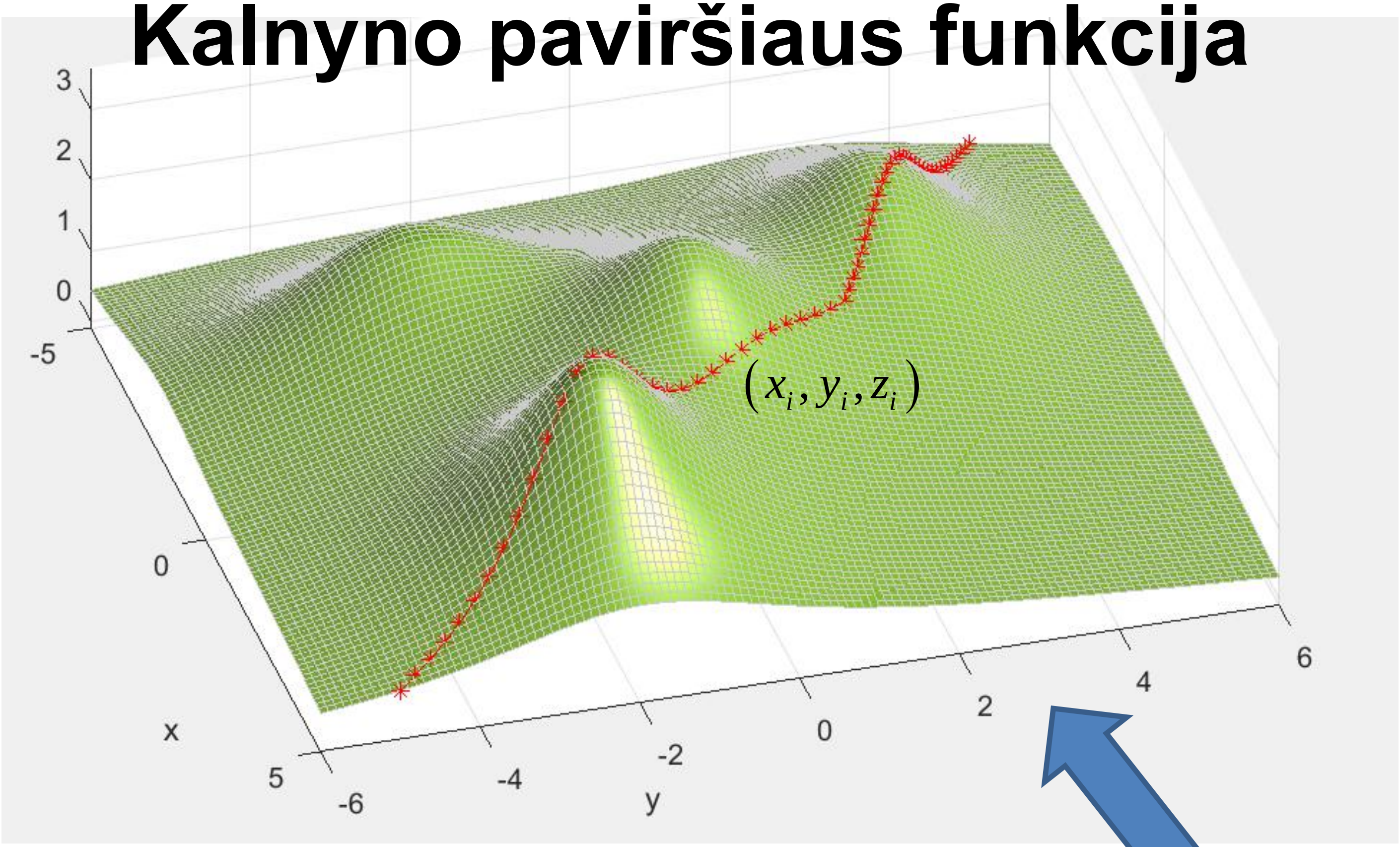
Reikia parinkti optimalią trasą plentui kalnuotoje vietovėje:

- Kelias tarp dviejų duotų taškų turi būti neilgas;
- Tuo pat metu siekiama, kad įkalnių-nuokalnių statumas būtų nedidelis



Ši trasa yra pradinis artinys. Nors ji neilga, tačiau statumas kai kur yra labai didelis

Kalnyno paviršiaus funkcija



Viršūnių skaičius

Viršūnės aukštis

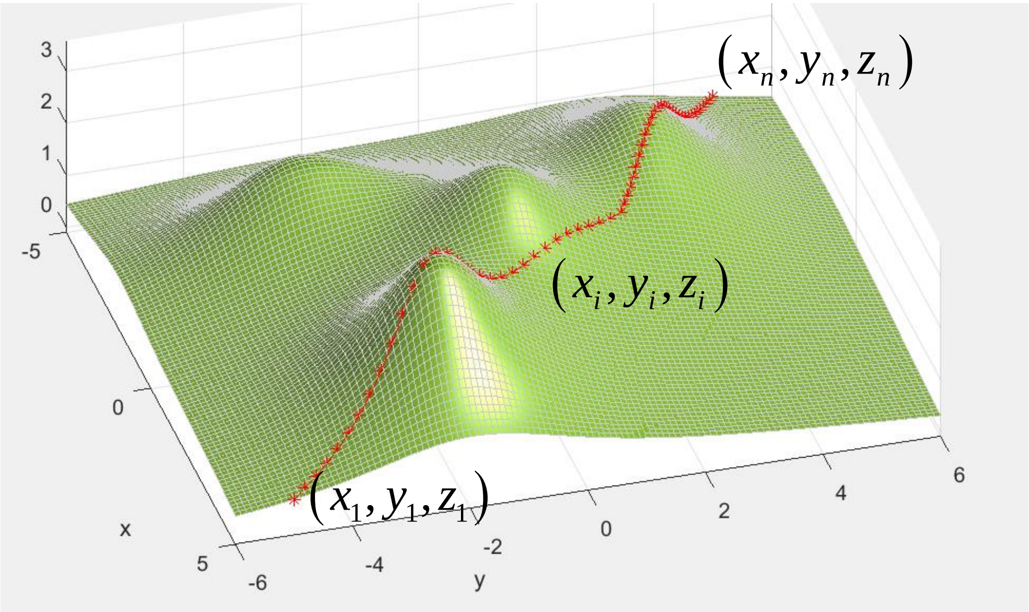
$$z(x,y)=\sum_{j=1}^{m}\frac{h_j}{1+\alpha_j\left(x-a_j\right)^2+\beta_j\left(y-b_j\right)^2}$$

Šlaitų statumo koeficientai

Viršūnės koordinatės xOy plokštumoje

```
height=[  2    3    2   2]
alfax= [ 0.5 0.5 0.25 0.5];
ax=    [  0   3   -2   -2];
alfay= [ 0.5 0.5 0.5 0.25];
ay=    [  0   -2   3  -3 ];
```

Tikslo funkcija, nusakanti trasos kokybę



$$\min_{x_{2:n-1},\, y_{2:n-1}} \Psi(\mathbf{x},\mathbf{y}) = \sum_{i=1}^{n-1} (x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2 + (Kz_{i+1} - Kz_i)^2$$

- Tikslo funkcija yra visų trasos atkarpų ilgių kvadratų suma;
- Optimizavimo kintamieji yra visų taškų, išskyrus pirmąjį ir paskutinį, x ir y koordinatės

- Koeficientas K padidina aukščių skirtumų atkarpų galuose įtaką tikslo funkcijos reikšmei;
- Tikrasis kelio ilgis (kaip atkarpų kvadratų suma) gaunamas, kai K=1

Tikslo funkcijos gradiento *analizinė išraiška*

$$\Psi(\mathbf{x},\mathbf{y}) = \sum_{i=1}^{n-1} (x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2 + (Kz_{i+1} - Kz_i)^2$$

$$\nabla\Psi = \left(\frac{\partial\Psi}{\partial x_2}, \frac{\partial\Psi}{\partial x_3}, \cdots, \frac{\partial\Psi}{\partial x_{n-1}}, \frac{\partial\Psi}{\partial y_2}, \frac{\partial\Psi}{\partial y_3}, \cdots, \frac{\partial\Psi}{\partial y_{n-1}} \right)$$

$$\frac{\partial\Psi}{\partial x_i} = 2(x_i - x_{i-1}) - 2(x_{i+1} - x_i) + K\left(2(z_i - z_{i-1}) - 2(z_{i+1} - z_i)\right) \left.\frac{\partial z}{\partial x}\right|_i, \; i = 2,3,\ldots,n-1$$

$$\frac{\partial\Psi}{\partial y_i} = 2(y_i - y_{i-1}) - 2(y_{i+1} - y_i) + K\left(2(z_i - z_{i-1}) - 2(z_{i+1} - z_i)\right) \left.\frac{\partial z}{\partial y}\right|_i, \; i = 2,3,\ldots,n-1$$

$$\frac{\partial z(x,y)}{\partial x} = \sum_{j=1}^{m} \frac{-2h_j\alpha_j(x - a_j)}{\left(1 + \alpha_j(x - a_j)^2 + \beta_j(y - b_j)^2\right)^2}$$

$$\frac{\partial z(x,y)}{\partial y} = \sum_{j=1}^{m} \frac{-2h_j\beta_j(y - b_j)}{\left(1 + \alpha_j(x - a_j)^2 + \beta_j(y - b_j)^2\right)^2}$$

Pagal analiziškai gautas *z(x,y)* išvestinių išraiškas reikia apskaičiuoti jų reikšmes taške *(x_i , y_i)*

$$z(x,y) = \sum_{j=1}^{m} \frac{h_j}{1 + \alpha_j(x - a_j)^2 + \beta_j(y - b_j)^2}$$

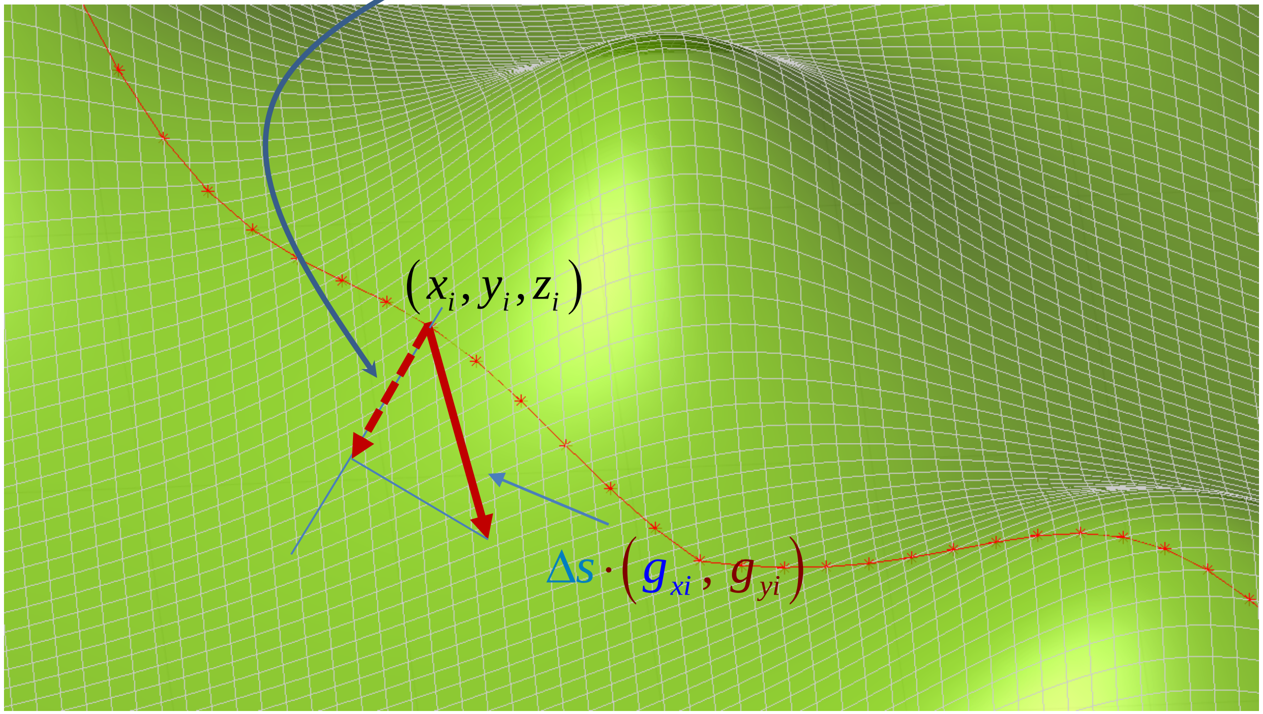
Gradiento projekcijos metodas

- Kiekviename optimizavimo žingsnyje prie visų trasos taškų koordinačių pridedami jų pokyčiai, proporcingi atitinkamoms gradiento komponentėms:

optimizavimo žingsnio ilgis

$$\left(\vec{\mathbf{g}}_x, \vec{\mathbf{g}}_y\right) = \vec{\nabla}\Psi \Big/ \left\|\vec{\nabla}\Psi\right\| \;,\; x_i = x_i - \Delta s \cdot g_{xi} \quad,\quad y_i = y_i - \Delta s \cdot g_{yi}$$

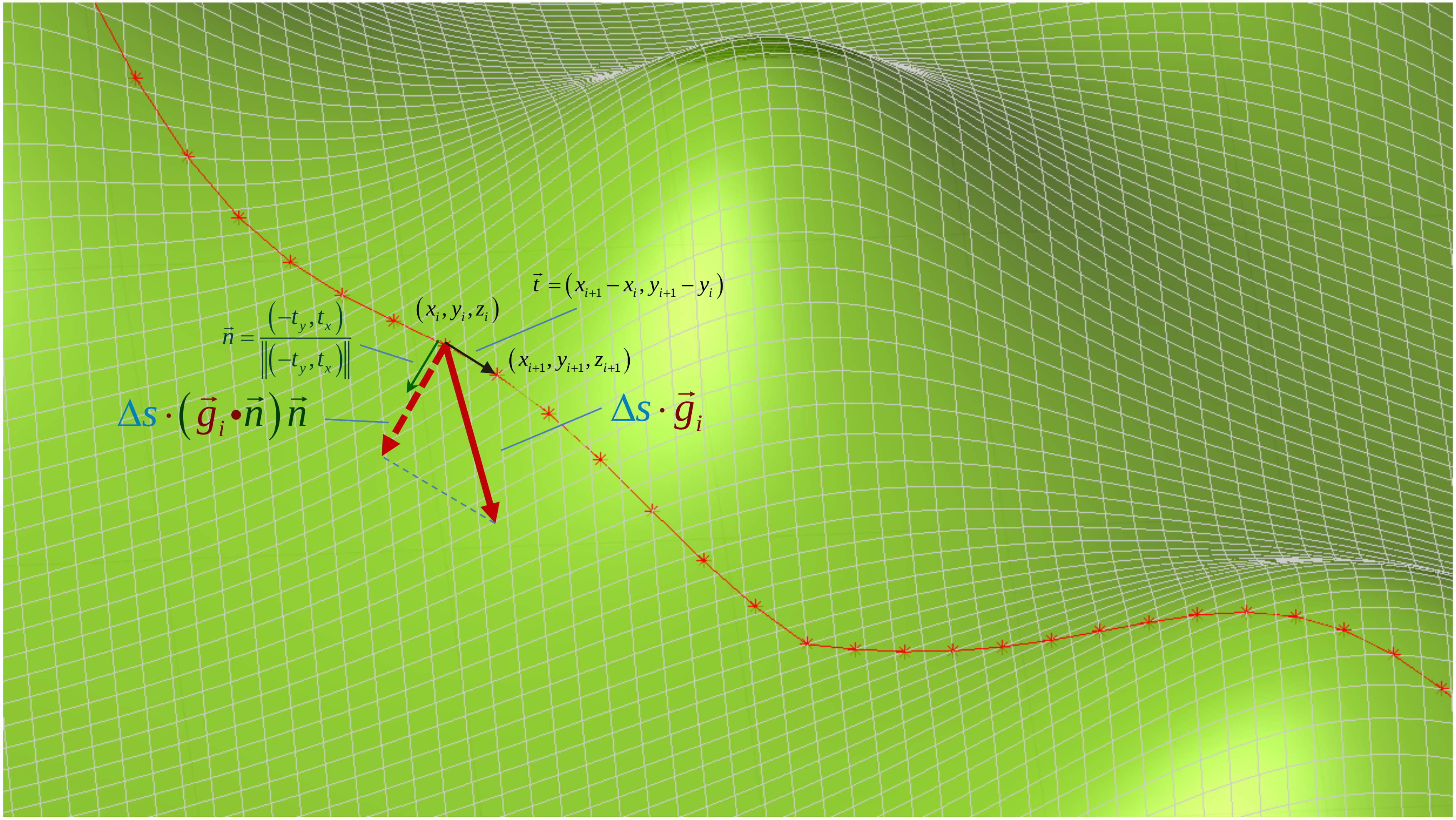
- Šiame uždavinyje tikslinga taikyti ne patį pagal gradientą apskaičiuotą koordinačių pokyčių vektorių, o jo projekciją į statmeną pradinio artinio trajektorijai kryptį. Kitaip kyla pavojus, kad trasos segmentų ilgiai optimizavimo eigoje gal tapti labai skirtingais:



Gradiento projekcijos apskaičiavimas

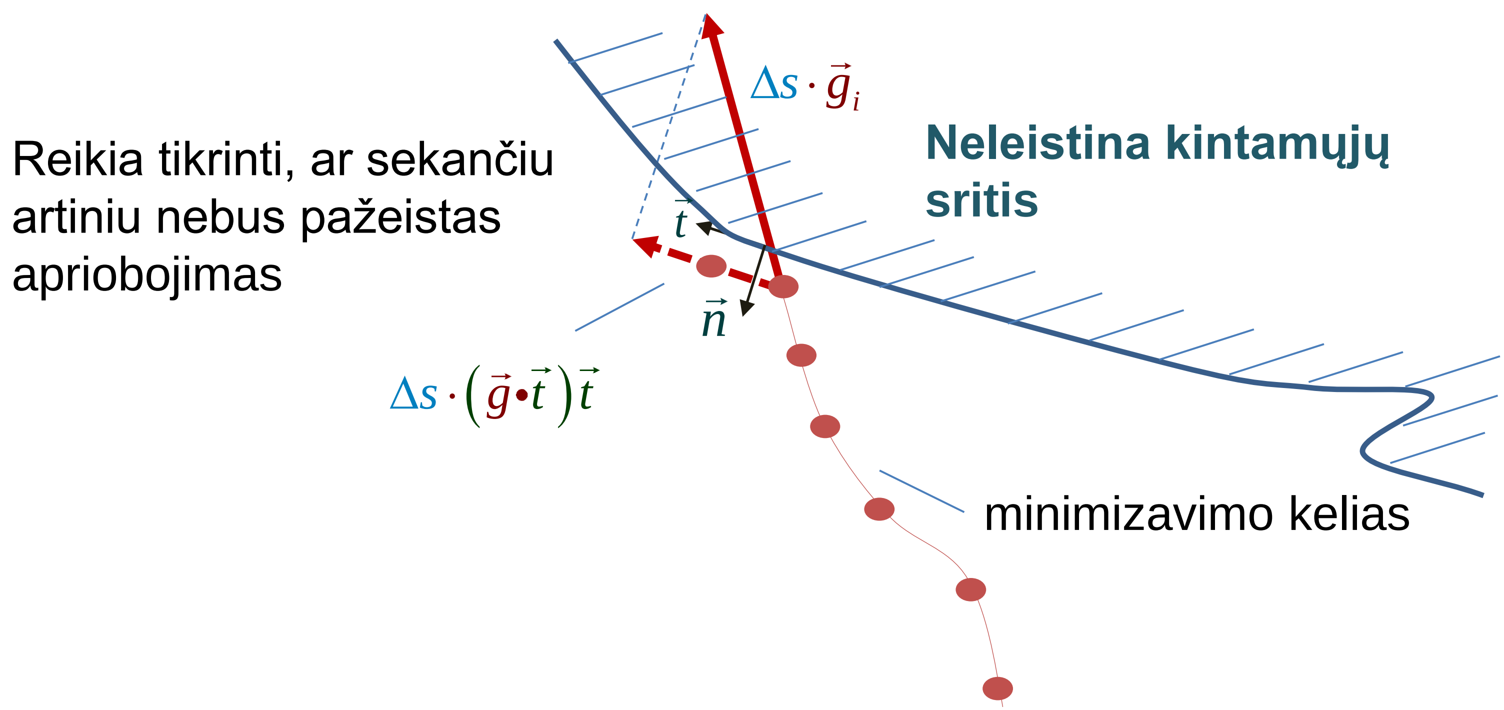
Pagal gradientą apskaičiuota taško koordinačių keitimo kryptis: $\vec{g}_i = \left(g_{xi} \, , \; g_{yi} \right)$

Taško koordinačių keitimo kryptis projektuojant gradientą į normalę: $\left(\vec{g}_i \bullet \vec{n} \right) \vec{n}$

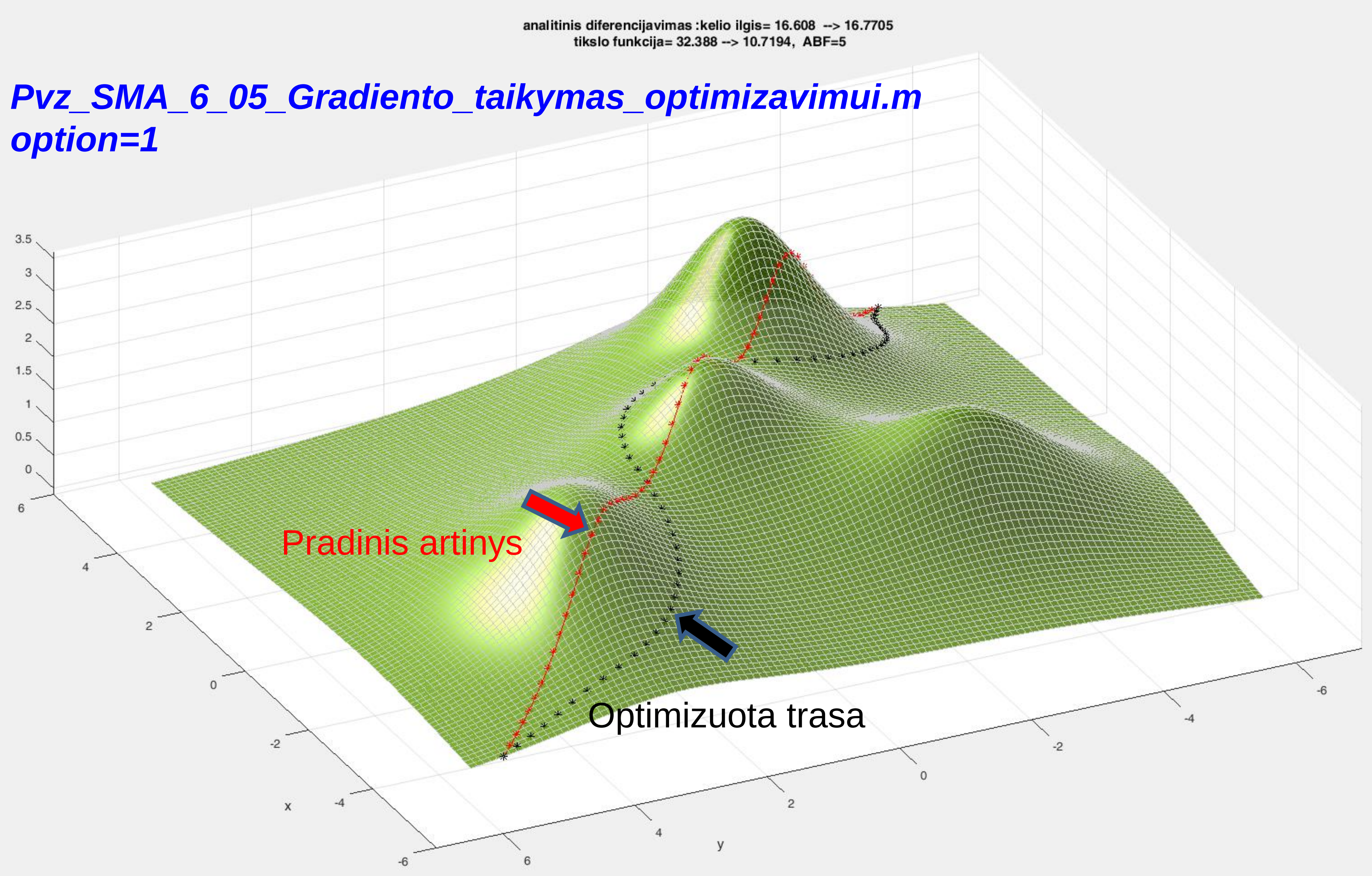


Gradiento projekcijos metodas

- *Sąlyginio optimizavimo* uždaviniuose atvejis, kai minimizuojamos funkcijos gradientas veda į neleistiną kintamųjų reikšmių sritį, pasitaiko dažnai;
- Gradientu nurodoma minizavimo kryptis nukreipiama lygiagrečiai apribojimo hiperplokštumai, vektorizuojant gradiento projekciją į tą hiperplokštumą;
- Toks gradientinės minimumo paieškos būdas vadinamas *gradiento projekcijos* metodu



_Pvz_SMA_6_05_Gradiento_taikymas_optimizavimui.m_
option=1



Gradiento skaitinis įvertis. Kvazi-gradiento apskaičiavimas Broideno metodu

Gradiento vektoriaus reikšmę galima apytiksliai įvertinti *skaitiškai*, t.y. nenaudojant analizinių diferencijavimo formulių:

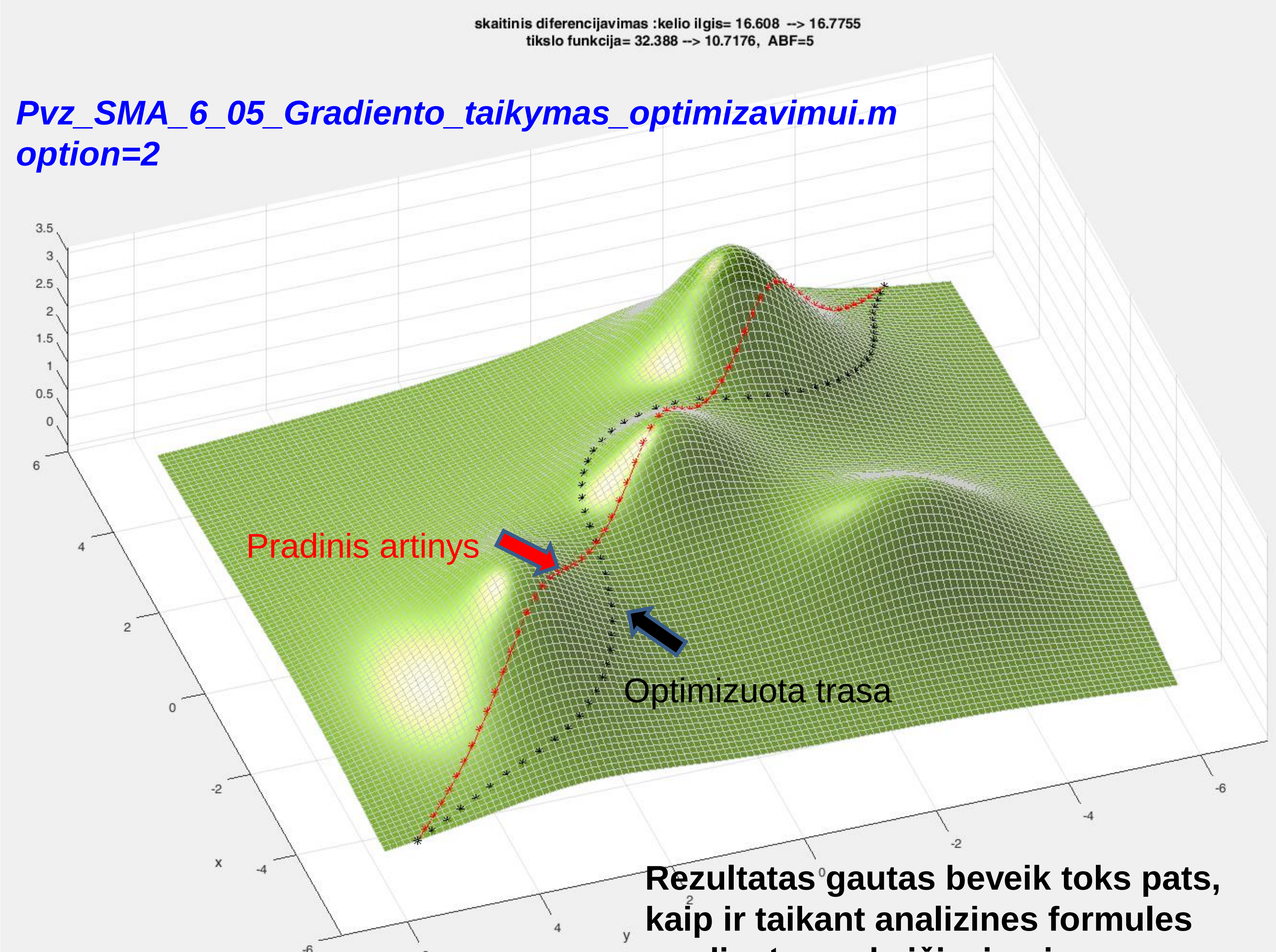
$$\nabla\Psi = \left(\frac{\partial\Psi}{\partial x_2}, \frac{\partial\Psi}{\partial x_3}, \cdots, \frac{\partial\Psi}{\partial x_{n-1}}, \frac{\partial\Psi}{\partial y_2}, \frac{\partial\Psi}{\partial y_3}, \cdots, \frac{\partial\Psi}{\partial y_{n-1}} \right)$$

$$\frac{\partial\Psi}{\partial x_j} = \frac{\Psi(x_2, x_3, \ldots, x_j + h, \ldots, x_{n-1}, y_2, \ldots y_{n-1}) - \Psi(x_2, x_3, \ldots, x_j, \ldots, x_{n-1}, y_2, \ldots y_{n-1})}{h};$$

$$\frac{\partial\Psi}{\partial y_j} = \frac{\Psi(x_2, x_3, \ldots, x_{n-1}, y_2, \ldots, y_j + h, \ldots, y_{n-1}) - \Psi(x_2, x_3, \ldots, x_j, \ldots, x_{n-1}, y_2, \ldots y_{n-1})}{h}$$

Parinktas mažas argumentų prieaugis

_Pvz_SMA_6_05_Gradiento_taikymas_optimizavimui.m_
option=2



Rezultatas gautas beveik toks pats, kaip ir taikant analizines formules gradiento apskaičiavimui

Kvazi-gradiento apskaičiavimas *Broideno metodu*(1)

- Jakobio matricos eilutėse būna įrašyti visų lygčių sistemą sudarančių funkcijų gradientai;
- Jeigu turime tik vieną funkciją (t.y. optimizavimo uždavinio tikslo funkciją), ***anksčiau išvestos Broideno formulės taikomos vienai matricos A eilutei***:

Gradiento aproksimacija yra vektorius-eilutė

Vektoriniai dydžiai

$$\mathbf{s} = \mathbf{x}^{i} - \mathbf{x}^{i-1}$$

$$y = \Psi\left(\mathbf{x}^{i}\right) - \Psi\left(\mathbf{x}^{i-1}\right)$$

Skaliariniai dydžiai

$$\begin{cases} \mathbf{a}_i \mathbf{s} = y; \\ \mathbf{a}_i \left(\mathbf{E} - \dfrac{\mathbf{s}\mathbf{s}^T}{\mathbf{s}^T\mathbf{s}} \right) = \mathbf{a}_{i-1} \left(\mathbf{E} - \dfrac{\mathbf{s}\mathbf{s}^T}{\mathbf{s}^T\mathbf{s}} \right) \end{cases}$$

$$\Downarrow$$

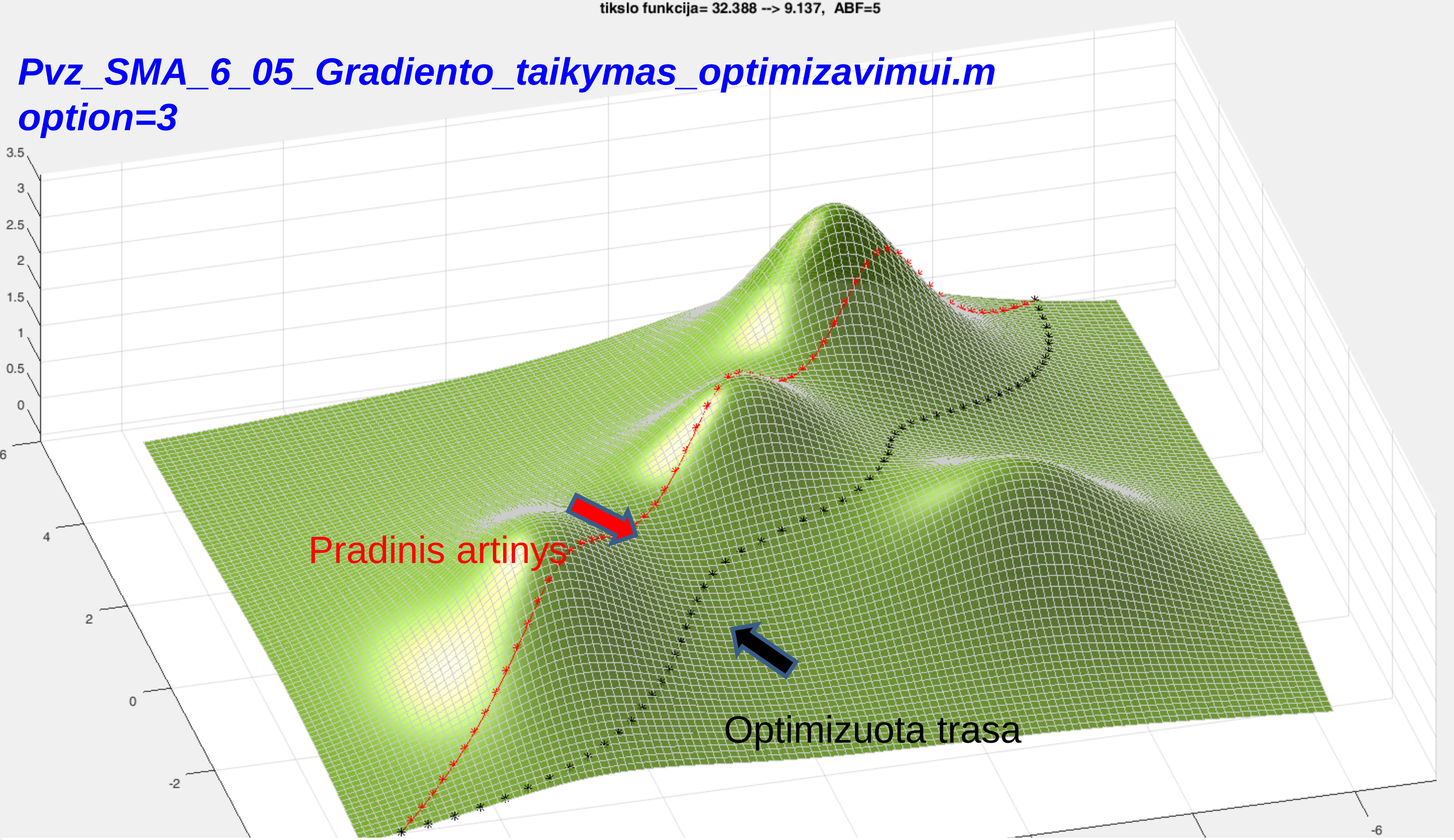
$$\mathbf{a}_i = \mathbf{a}_{i-1} + \frac{\left(y - \mathbf{a}_{i-1}\mathbf{s} \right)\mathbf{s}^T}{\mathbf{s}^T\mathbf{s}}$$

Kvazi-gradiento apskaičiavimas *Broideno metodu*(2)

- Apskaičiuodami gradientą tiek skaitinio diferencijavimo, tiek ir Broideno būdu, nenaudojame analizinių diferencijavimo formulių;
- Broideno metodu gradientui apskaičiuoti reikia mažiau aritmetinių veiksmų, lyginant su skaitiniu diferencijavimu;
- Geriausi optimizavimo rezultatai gaunami, gradiento apskaičiavimui ***kombinuojant Broideno ir skaitinio diferencijavimo metodus***:

1) Pradinis gradiento artinys gaunamas skaitiniu diferencijavimu;
2) Optimizuojant greičiausio nusileidimo metodu, paieškos krypčių pakeitimui naudojamas Broideno metodas;
3) Žingsnių skaičius pagal vieną kryptį imamas nedidelis, pavyzdžiui, 3 arba 4. T.y., nesiekiama pilno nusileidimo pagal kryptį. Geresni rezultatai gaunami, kai Broideno metodu apskaičiuotos kryptys dažniau keičiamos;
4) Kai funkcija jau nebemažėja, gradientas patikslinamas skaitiniu diferencijavimu ir taip gauta kryptimi bandoma optimizuoti toliau;
5) Po patikslinimo skaitiniu diferencijavimu, sekančioms kryptims gauti vėl taikomas Broideno metodas

Pvz_SMA_6_05_Gradiento_taikymas_optimizavimui.m option=3



- **Kombinuojant *Broideno ir skaitinio diferencijavimo metodus* pavyko gauti geresnį sprendinį. Jis yra kokybiškai kitoks, nei gautieji diferencijuojant vien analitiškai arba vien skaitiškai;**
- **Prisiminkime, kad *Broideno metodas* yra *kirstinių metodo* plėtinys. Gautas rezultatas patvirtina kirstinių metodo privalumus, kai gradiento įvertis gaunamas remiantis gana nutolusiais vienas nuo kito artiniais**

Funkcijos minimizavimu paremti netiesinių lygčių sistemų sprendimo metodai

Funkcijos minimizavimu paremti netiesinių lygčių sistemų sprendimo metodai

$$\mathbf{f}(\mathbf{x}) = 0 \quad ,$$

$$\min_{\{\mathbf{x}\}} \Psi(\mathbf{f}(\mathbf{x})) = \frac{1}{2}\mathbf{f}(\mathbf{x})^T \mathbf{f}(\mathbf{x}) \quad ,$$

$$\min_{\{\mathbf{x}\}} \Psi(\mathbf{f}(\mathbf{x})) = \frac{1}{2}\sum_{i=1}^{n} f^2{}_i(\mathbf{x}) \quad ,$$

- Funkcijos Ψ minimumas yra 0 reikšmė, kuri galima tik esant $\mathbf{f}(\mathbf{x}) = 0$;
- Argumentų vektoriaus x reikšmės, kurioms esant gaunama minimali Ψ reikšmė, yra lygčių sistemos sprendinys;
- Skirtumas nuo lygčių sistemos sprendimo toks, kad reikia suteikti 0 reikšmę tik vienai funkcijai, o ne funkcijų vektoriui;
- Optimizavimu grįsti metodai kartais veikia stabiliau, nei Niutono arba kvazi-Niutono metodai. Jie ne tiek jautrūs pradinio artinio parinkimui

Skaliarinės funkcijos, aprašančios netiesinių lygčių sistemą, gradientas (1)

$$\mathbf{f}(\mathbf{x}) = 0$$

Ψ yra sudėtinė funkcija, kuri tiesiogiai priklauso tik nuo *f*, kuri savo ruožtu priklauso nuo *x*

$$\Psi(\mathbf{f}(\mathbf{x})) = \frac{1}{2}\mathbf{f}(\mathbf{x})^T \mathbf{f}(\mathbf{x}) = \frac{1}{2}\sum_{i=1}^{n} f_i^{\,2}(\mathbf{x})\ ;$$

$$\vec{\nabla}\Psi = \frac{\partial \Psi(\mathbf{f}(\mathbf{x}))}{\partial \mathbf{x}} = \left\{ \frac{\partial \Psi(\mathbf{f}(\mathbf{x}))}{\partial x_1} \quad \frac{\partial \Psi(\mathbf{f}(\mathbf{x}))}{\partial x_2} \quad \cdots \quad \frac{\partial \Psi(\mathbf{f}(\mathbf{x}))}{\partial x_n} \right\} =$$

$$= \frac{\partial \Psi(\mathbf{f}(\mathbf{x}))}{\partial \mathbf{x}} = \underbrace{\frac{\partial \Psi(\mathbf{f})}{\partial \mathbf{f}} \frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}}}$$

Vektorinės funkcijos ***f*** *Jakobio matrica*, kai diferencijuojama pagal vektorinį argumentą ***x***

Šis vektorius-eilutė yra funkcijos Ψ *gradientas*, vektorinio argumento ***f*** erdvėje

Sandauga yra vektorius-eilutė. Tai funkcijos Ψ gradientas vektorinio argumento ***x*** erdvėje

Skaliarinės funkcijos, aprašančios netiesinių lygčių sistemą, gradientas (2)

$$\mathbf{f}(\mathbf{x}) = 0$$

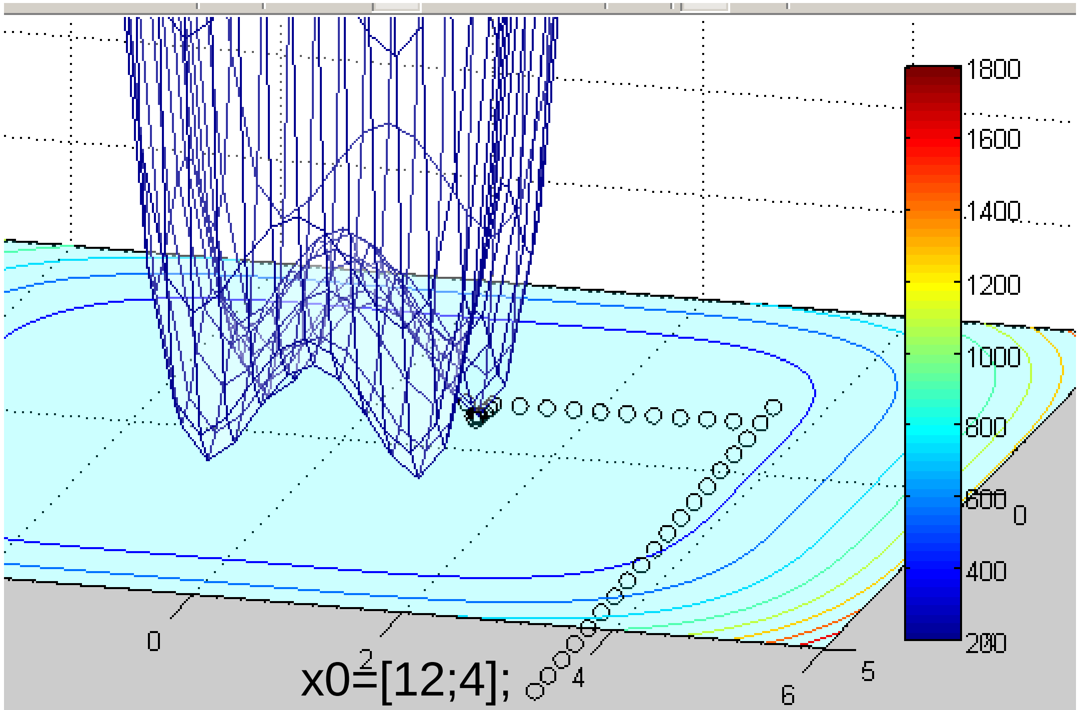
$$\Psi(\mathbf{f}) = \frac{1}{2}\mathbf{f}^T\mathbf{f} = \frac{1}{2}\sum_{i=1}^{n} f^2{}_i;$$

$$\left[\frac{\partial\Psi(\mathbf{f})}{\partial\mathbf{f}}\right] = \left\{\frac{\partial\Psi}{\partial f_1} \quad \frac{\partial\Psi}{\partial f_2} \quad \cdots \quad \frac{\partial\Psi}{\partial f_n}\right\} = \left\{f_1 \quad f_2 \quad \cdots \quad f_n\right\} = \mathbf{f}^T;$$

$$\boxed{[\nabla\Psi] = \mathbf{f}^T\frac{\partial\mathbf{f}(\mathbf{x})}{\partial\mathbf{x}} = \mathbf{f}^T[\mathbf{J}]}$$

$$\frac{\partial\left(\mathbf{f}^T\mathbf{f}\right)}{\partial\mathbf{f}} = 2\mathbf{f}^T$$

Pvz_SMA_6_06, 6_06x,6_07



$$\begin{cases} f_1(x_1, x_2) = x_1^{\ 2} + x_2^{\ 2} - 2 = 0; \\ f_2(x_1, x_2) = x_1^{\ 2} - x_2^{\ 2} = 0 \end{cases}$$

Kvazi-gradiento metodų taikymas NLS funkcijos minimizavimui

Kvazi-gradiento metodų taikymas NLS sprendimui

- Minimizuojant funkciją gradiento metodu, reikia apskaičiuoti analitines Jakobio matricos visų narių išraiškas. Tai toks pats veiksmas, kuris atliekamas, taikant Niutono metodą;
- Būtų galima taikyti kvazi-gradiento metodus, t.y. apytikslį Jakobio matricos pavidalą;
- Apskaičiuojant minimizuojamos funkcijos kvazi-gradientą, reikalingas lygiai toks pats Jakobio matricos artinys, kaip ir taikant Broideno metodą

Lygčių sistemos sprendimas Broideno metodu

$$\mathbf{x}^0, \mathbf{A}_0$$

$$\mathbf{A}_{i-1}\mathbf{s}_i = -\mathbf{f}(\mathbf{x}^{i-1});$$

$$\mathbf{x}^i = \mathbf{x}^{i-1} + \mathbf{s}_i$$

$$\mathbf{y}_i = \mathbf{f}(\mathbf{x}^i) - \mathbf{f}(\mathbf{x}^{i-1});$$

$$\mathbf{A}_i = \mathbf{A}_{i-1} + \frac{\left(\mathbf{y}_i - \mathbf{A}_{i-1}\mathbf{s}_i\right)\mathbf{s}_i^T}{\mathbf{s}_i^T\mathbf{s}_i};$$

$$i = 1, 2, \ldots$$

$\Rightarrow$

Funkcijos minimizavimas, kvazi-gradiento išraiškoje taikant Jakobio matricos artinį, kaip ir Broideno metode

$$\mathbf{x}^0, \mathbf{A}_0$$

$$\left[\nabla\Psi\right] = \mathbf{f}\left(\mathbf{x}^{i-1}\right)^T \mathbf{A}_{i-1}$$

$$\mathbf{x}^{i-1} \overset{[\nabla\Psi]}{\Rightarrow} \mathbf{x}^i$$

Funkcija minimizuojama nuo $\mathbf{x}^{i-1}$ greičiausio nusileidimo būdu. $\mathbf{x}$ keičiamas mažais žingsniais priešinga gradientui kryptimi, gaunamas x^i

$$\mathbf{s}_i = \mathbf{x}^i - \mathbf{x}^{i-1}$$

$$\mathbf{y}_i = \mathbf{f}(\mathbf{x}^i) - \mathbf{f}(\mathbf{x}^{i-1});$$

$$\mathbf{A}_i = \mathbf{A}_{i-1} + \frac{\left(\mathbf{y}_i - \mathbf{A}_{i-1}\mathbf{s}_i\right)\mathbf{s}_i^T}{\mathbf{s}_i^T\mathbf{s}_i};$$

$$i = 1, 2, \ldots$$

Pvz_SMA_6_08, 6_08x, 6_09

Funkcijų **f** šiame paveiksle nematome

$$\mathbf{f} = \begin{Bmatrix} x_1^{\ 2} + x_2^{\ 2} - 2 \\ x_1^{\ 2} - x_2^{\ 2} \end{Bmatrix}$$

$$\Psi = \frac{1}{2}\mathbf{f}^T\mathbf{f}$$

$$\mathbf{x}^0, \mathbf{A}_0$$

$$[\nabla\Psi] = \mathbf{f}\left(\mathbf{x}^{i-1}\right)^T \mathbf{A}_{i-1}$$

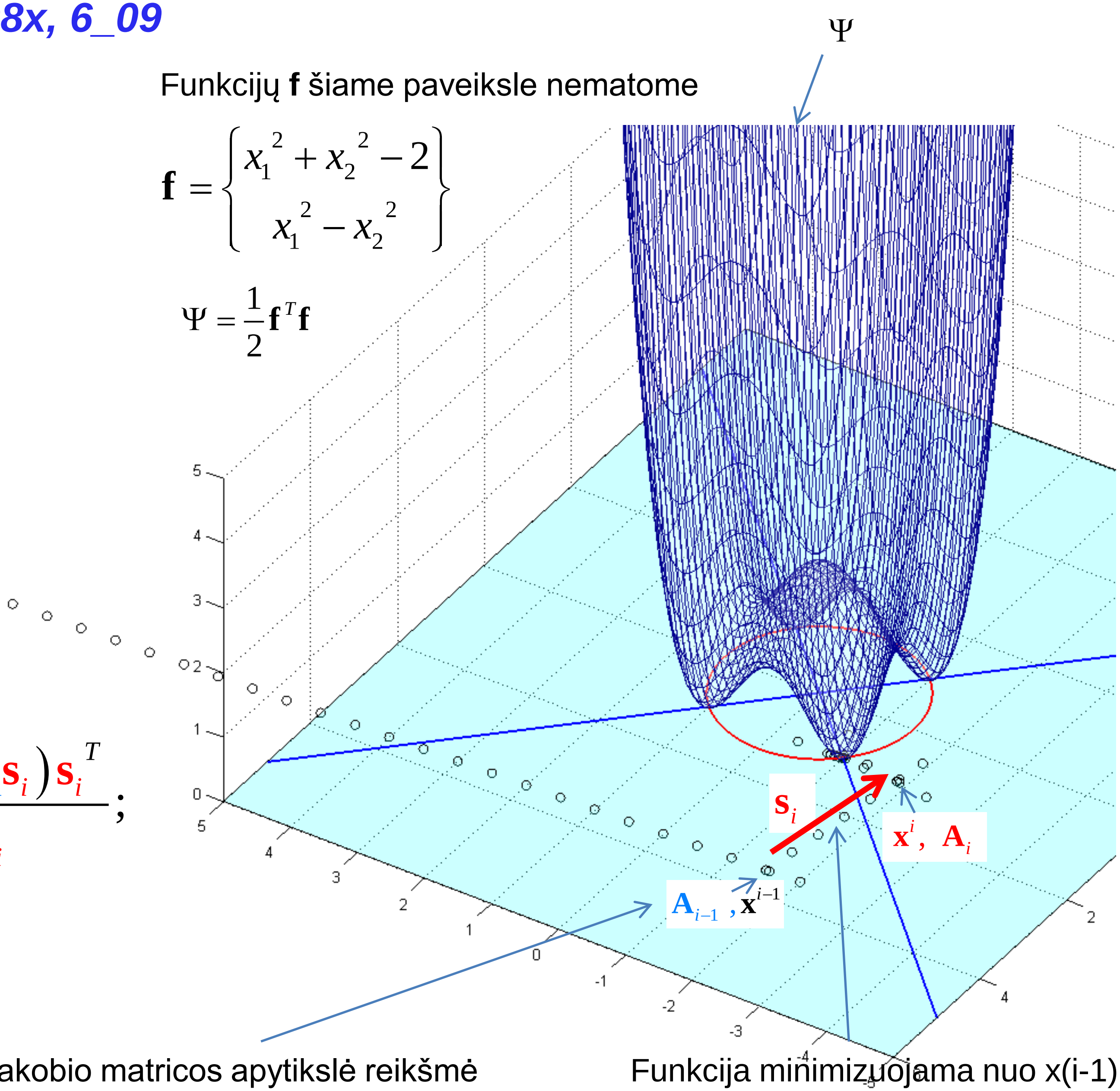
$$\mathbf{x}^{i-1} \overset{[\nabla\Psi]}{\Rightarrow} \mathbf{x}^i$$

$$\mathbf{s}_i = \mathbf{x}^i - \mathbf{x}^{i-1}$$

$$\mathbf{y}_i = \mathbf{f}(\mathbf{x}^i) - \mathbf{f}(\mathbf{x}^{i-1});$$

$$\mathbf{A}_i = \mathbf{A}_{i-1} + \frac{\left(\mathbf{y}_i - \mathbf{A}_{i-1}\mathbf{s}_i\right)\mathbf{s}_i^{\ T}}{\mathbf{s}_i^{\ T}\mathbf{s}_i};$$

$$i = 1, 2, \ldots$$



Nauja Jakobio matricos apytikslė reikšmė apskaičiuojama tik keičiant minimizavimo kryptį (t.y. radus minimumą anksčiau naudotoje kryptyje)

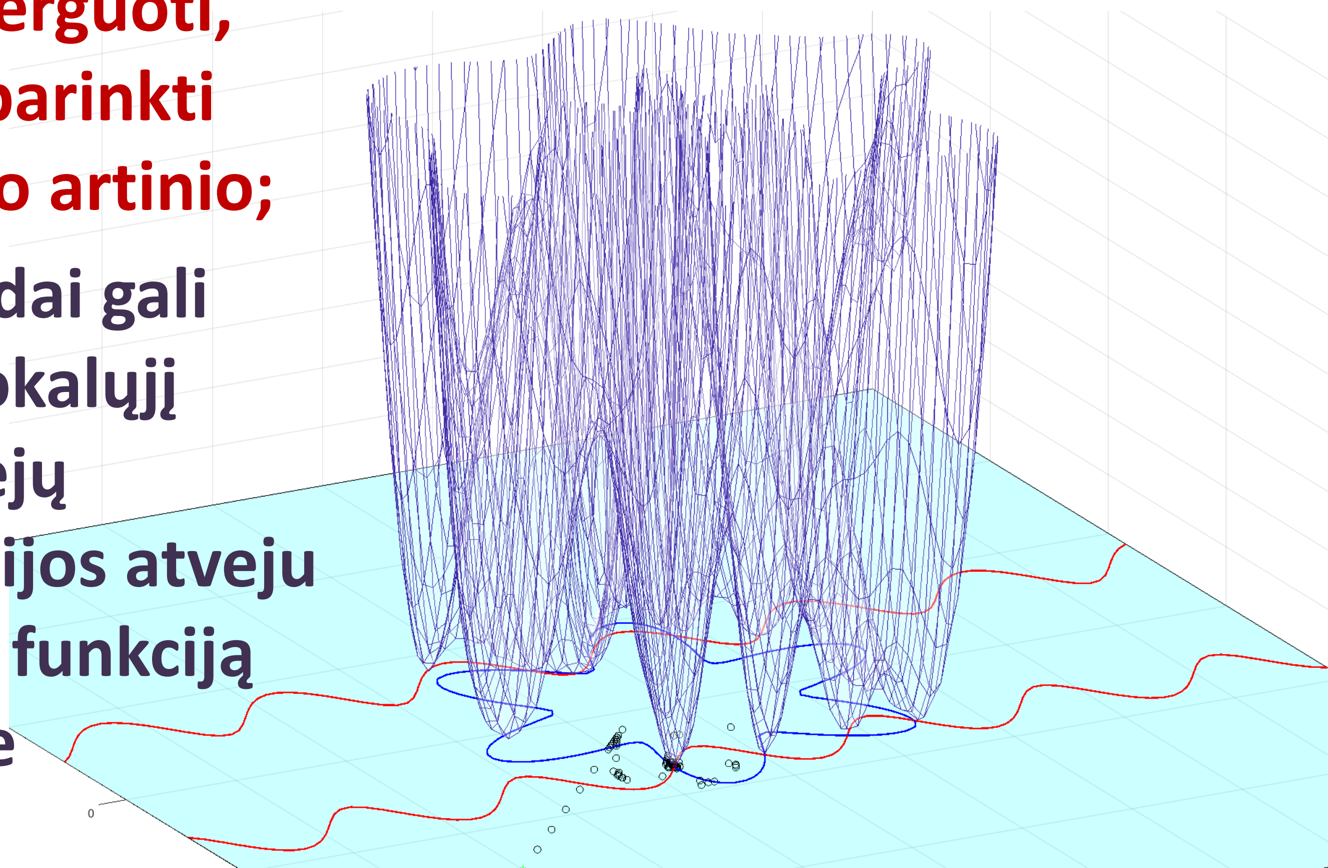
Funkcija minimizuojama nuo x(i-1) greičiausio nusileidimo būdu. **x** keičiamas mažais žingsniais priešinga gradientui kryptimi, gaunamas xi

Problemos, iškylančios sprendžiant netiesines lygčių sistemas

Bendruoju atveju sprendinio galime ir neaptikti:

- **Niutono bei kvazi-Niutono metodai gali diverguoti, kai nepavyksta parinkti tinkamo pradinio artinio;**
- Gradiento metodai gali sustoti, aptikę lokalųjį minimumą. Dviejų kintamųjų funkcijos atveju tai yra įdubimas funkciją vaizduojančiame paviršiuje;

```
sin(4*x(1))+x(2)^2-2=0;
x(1)^2+x(2)^2-(0.5*sin(6*atan(x(2)/x(1)))+1.5)^2=0
```



SMA_06_1_ Klausimai savikontrolei (1):

1. Kuo skiriasi optimizavimo ir minimizavimo sąvokos. Kuri iš jų platesnė pagal prasmę;
2. Suformuluokite daugelio kintamųjų funkcijos minimizavimo uždavinį;
3. Kas yra funkcijos gradientas;
4. Kaip minimizuojama funkcija, einant priešinga gradientui kryptimi;
5. Kaip minimizuojama funkcija greičiausio nusileidimo metodu. Kuo tai skiriasi nuo minimizavimo priešinga gradientui kryptimi;
6. Kaip parenkamas žingsnis, funkcijos minimizavimui taikant gradiento metodus;
7. Kaip lygčių sistemos sprendimo uždavinys pakeičiamas vienos funkcijos minimizavimo uždaviniu;
8. Paaiškinkite, kaip apskaičiuojamas gradientas funkcijos, aprašančios netiesinių lygčių sistemą;
9. Kaip būtų galima gradiento būdu minimizuoti funkciją, neskaičiuojant Jakobio matricos pagal išvestinių formules;
10. Kokiais atvejais NLS sprendinio galima ir neaptikti, taikant Niutono, kvazi-Niutono ir gradiento sprendimo metodus