Netiesinių lygčių sistemų sprendimas

http://oras.if.ktu.lt/moduliai/T170B115/Skaitiniai inzinerijos metodai su MATLAB (KP RG RB)2009

F6.pdf - 3.2 poskyris

http://oras.if.ktu.lt/moduliai/T170B115 /Skaitiniai inzinerijos metodai(RB) 2006

I.1.2.pdf

Netiesinių algebrinių lygčių sprendimas. Matematinė formuluotė

$$f(x) = 0$$

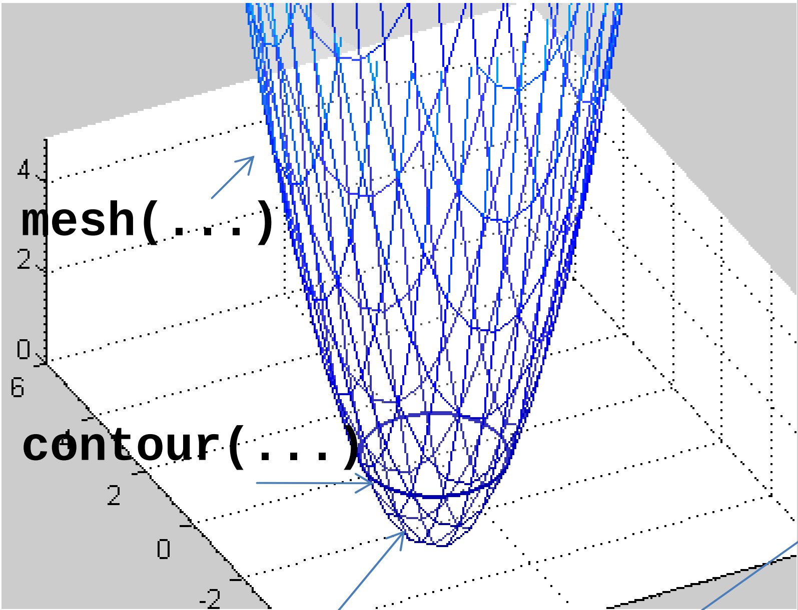
Vieno kintamojo lygtis (skaliarinė skaliarinio argumento funkcija)

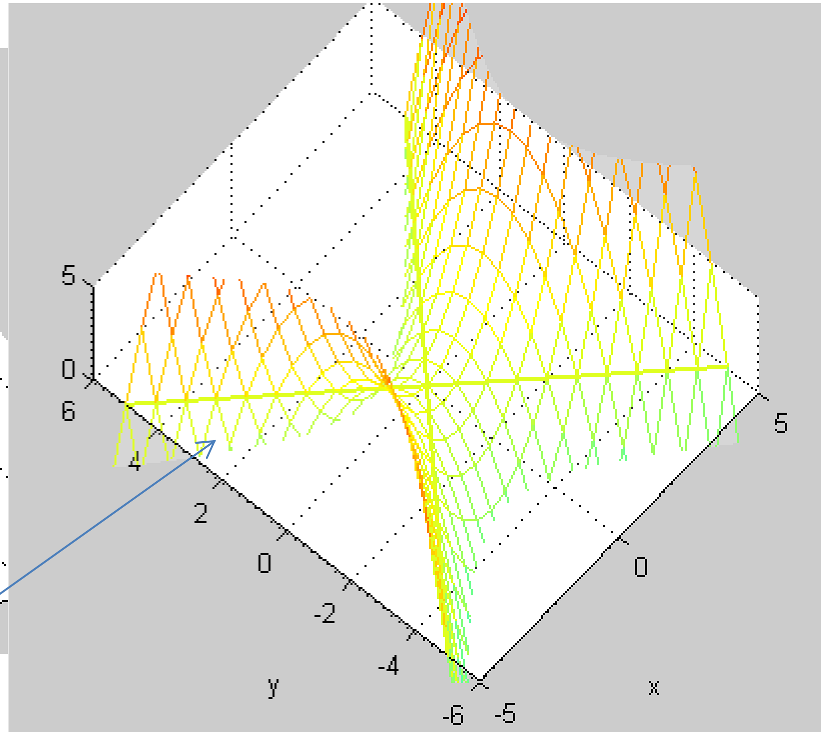
$$\{\mathbf{f}(\{\mathbf{x}\})\} = \begin{cases} \mathbf{f}_1(x_1, x_2, \ldots, x_n) = 0; \\ \mathbf{f}_2(x_1, x_2, \ldots, x_n) = 0; \\ \vdots \\ \mathbf{f}_n(x_1, x_2, \ldots, x_n) = 0 \end{cases}$$

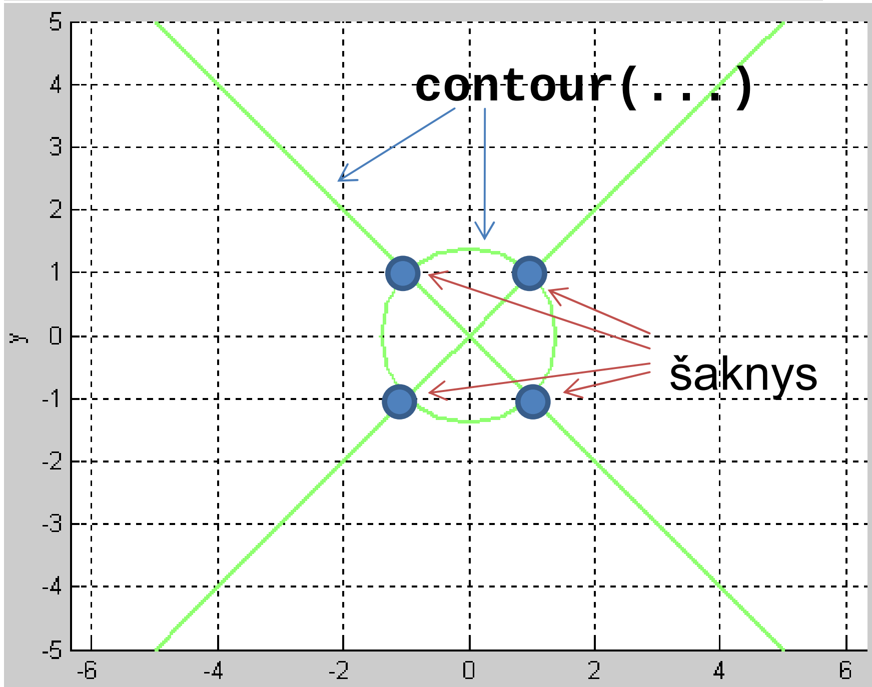
Daugelio kintamųjų lygčių sistema (vektorinė vektorinio argumento argumento funkcija)

Netiesinių algebrinių lygčių sistemų sprendimas.
Pradinis šaknų lokalizavimas

- Daugiamatėje kintamųjų ir funkcijų erdvėje universalių metodų pradiniam artiniui rasti nėra;
- Atskiroms lygčių klasėms dažnai pavyksta parinkti neblogą pradinį priartėjimą remiantis išankstinėmis žiniomis apie nagrinėjamą objektą;
- Dviejų kintamųjų erdvėje lygčių sistemai ištirti galima panaudoti funkcijų grafinį vaizdavimą







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

Netiesinių algebrinių lygčių sprendimas.
Paprastųjų iteracijų metodas

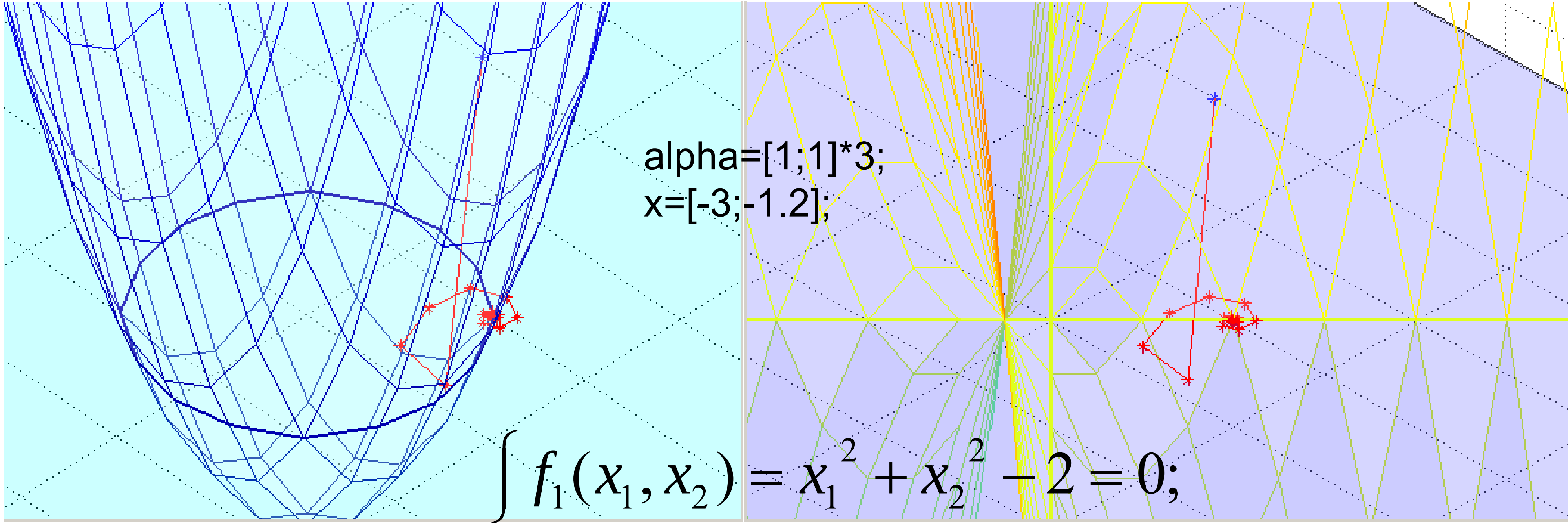
$$\left\{\mathbf{f}\left(\left\{\mathbf{x}\right\}\right)\right\} = 0 \quad \Rightarrow \quad \left\{\mathbf{x}\right\} = \frac{1}{\alpha}\left\{\hat{\mathbf{f}}\left(\left\{\mathbf{x}\right\}\right)\right\}$$

$$\hat{\mathbf{f}}(\mathbf{x}) = \mathbf{f}(\mathbf{x}) + \alpha\mathbf{x}$$

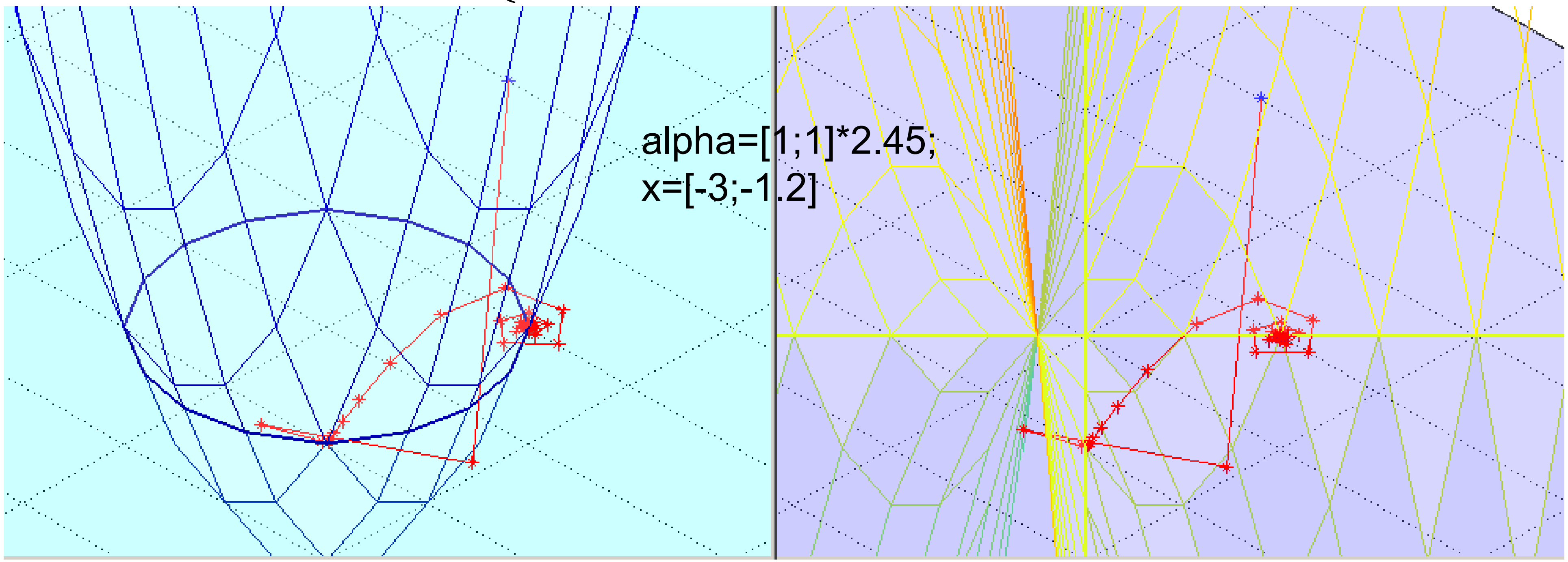
Gali būti įstrižaininė matrica

$\left\{\mathbf{x}^{0}\right\}$ - pradinis artinys

$$\boxed{\left\{\mathbf{x}^{i+1}\right\} = \left[\boldsymbol{\alpha}\right]^{-1}\left\{\hat{\mathbf{f}}\left(\mathbf{x}^{(i)}\right)\right\}, \quad i = 0,1,2,3,\ldots}$$



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



Netiesinių algebrinių lygčių sprendimas.
Niutono metodas

$$\mathbf{f}(\mathbf{x}) = 0 \ , \qquad \mathbf{x}^{(0)}$$ - pradinis artinys

$$f_k(\mathbf{x}^{i+1}) \approx f_k(\mathbf{x}^i) + \Delta x_1 \left.\frac{\partial f_k}{\partial x_1}\right|_{\mathbf{x}^i} + \Delta x_2 \left.\frac{\partial f_k}{\partial x_2}\right|_{\mathbf{x}^i} + \ldots + \Delta x_n \left.\frac{\partial f_k}{\partial x_n}\right|_{\mathbf{x}^i} , \qquad k = 1:n$$

Atkirsta Teiloro eilutė

$$\begin{Bmatrix} f_1(\mathbf{x}^{i+1}) \\ f_2(\mathbf{x}^{i+1}) \\ \vdots \\ f_n(\mathbf{x}^{i+1}) \end{Bmatrix} = \begin{Bmatrix} f_1(\mathbf{x}^{i}) \\ f_2(\mathbf{x}^{i}) \\ \vdots \\ f_n(\mathbf{x}^{i}) \end{Bmatrix} + \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \cdots & \frac{\partial f_n}{\partial x_n} \end{bmatrix} \begin{Bmatrix} \Delta x_1 \\ \Delta x_2 \\ \vdots \\ \Delta x_n \end{Bmatrix}$$

Jakobio matrica

Netiesinių algebrinių lygčių sprendimas.
Niutono metodas

$$\begin{Bmatrix} f_1(\mathbf{x}^{i+1}) \\ f_2(\mathbf{x}^{i+1}) \\ \vdots \\ f_n(\mathbf{x}^{i+1}) \end{Bmatrix} = \begin{Bmatrix} f_1(\mathbf{x}^{i}) \\ f_2(\mathbf{x}^{i}) \\ \vdots \\ f_n(\mathbf{x}^{i}) \end{Bmatrix} + \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \cdots & \frac{\partial f_n}{\partial x_n} \end{bmatrix} \begin{Bmatrix} \Delta x_1 \\ \Delta x_2 \\ \vdots \\ \Delta x_n \end{Bmatrix}$$

$$\left\{\mathbf{f}(\mathbf{x}^{i+1})\right\} = \left\{\mathbf{f}(\mathbf{x}^{i})\right\} + \left[\left.\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right|_{\mathbf{x}^i}\right]\left\{\Delta \mathbf{x}\right\}$$

$$\left\{\mathbf{f}(\mathbf{x}^{i+1})\right\} = 0 \quad \Rightarrow \quad \left[\left.\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right|_{\mathbf{x}^i}\right]\left\{\Delta \mathbf{x}\right\} = -\left\{\mathbf{f}(\mathbf{x}^{i})\right\}; \quad \Rightarrow \quad \mathbf{x}^{i+1} = \mathbf{x}^{i} + \left\{\Delta \mathbf{x}\right\}$$

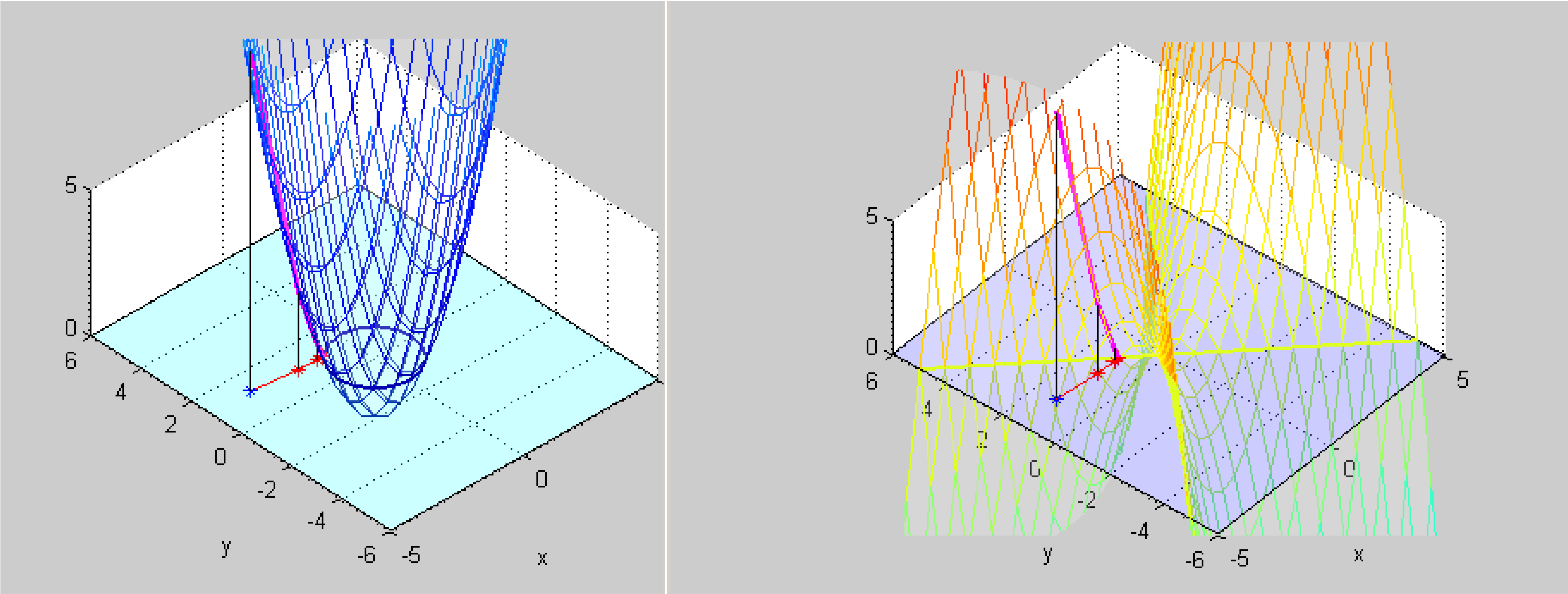
Kiekvienos iteracijos metu reikia išspręsti *tiesinių lygčių sistemą*

Netiesinių algebrinių lygčių sprendimas.
Niutono metodas

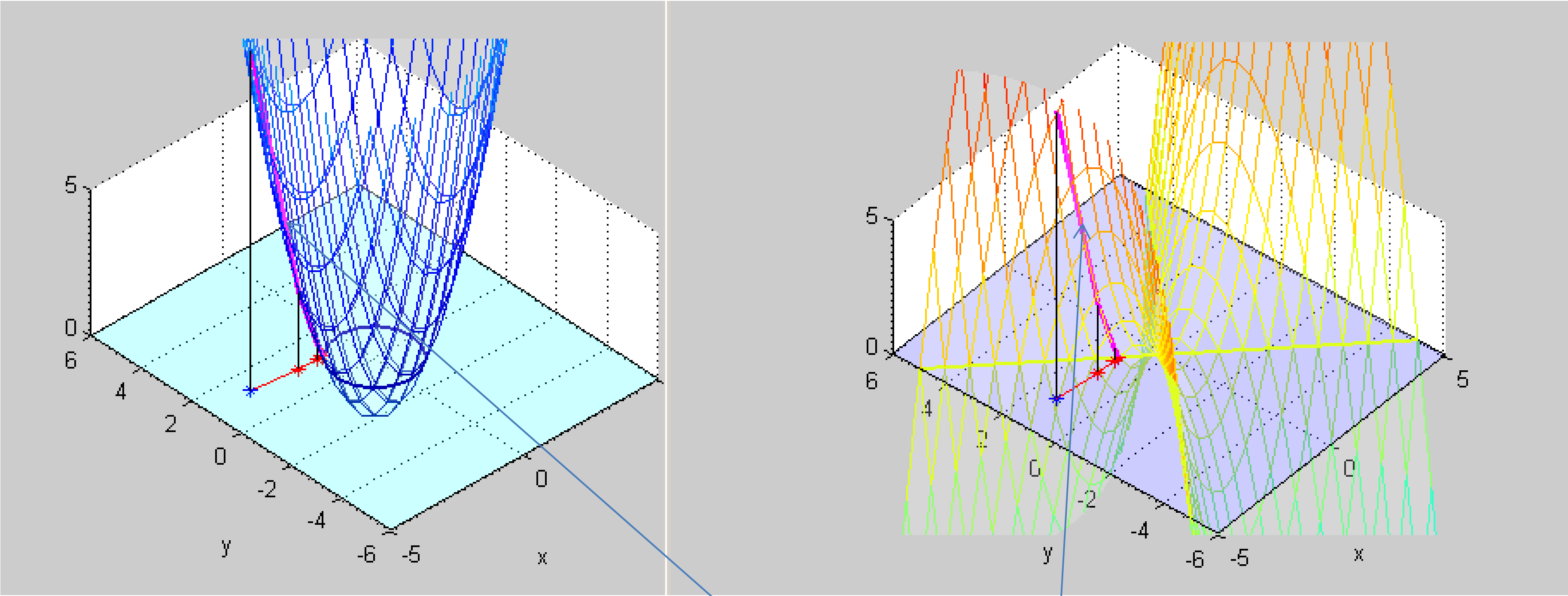
$$\left\{\mathbf{f}\left(\mathbf{x}^{i+1}\right)\right\}=0 \quad \Rightarrow \quad \left[\left.\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right|_{\mathbf{x}^i}\right]\left\{\Delta \mathbf{x}\right\}=-\left\{\mathbf{f}\left(\mathbf{x}^i\right)\right\}; \quad \Rightarrow \quad \mathbf{x}^{i+1}=\mathbf{x}^i+\left\{\Delta \mathbf{x}\right\}$$

$$\left\{\mathbf{x}^{i+1}\right\}=\left\{\mathbf{x}^{i}\right\}-\left[\left.\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right|_{\mathbf{x}^i}\right]^{-1}\left\{\mathbf{f}\left(\mathbf{x}^i\right)\right\}$$

Užrašas simboliais sutampa su Niutono metodo formule vienai netiesinei lygčiai



iteracija 1 tikslumas 0.304193
iteracija 2 tikslumas 0.241066
iteracija 3 tikslumas 0.109084
iteracija 4 tikslumas 0.0125576
iteracija 5 tikslumas 0.000116431
sprendinys x = -1.00016 1



•Sprendžiant Niutono metodu, kiekvienos lygties funkcijai artinio taške brėžiama liestinė (hiper)plokštuma;

•Liestinių plokštumų susikirtimas yra tiesė (ji yra ta pati abiems(visoms) funkcijoms);

•Sekantis artinys parenkamas ten, kur gautoji tiesė kerta nulinę (hiper)plokštumą

- **Niutono metodas visuomet konverguoja, kai pradedama skaiciuoti nuo gero pradinio artinio;**
- **Kiekvienos iteracijos metu būtina panaudoti ne tik funkcijos, tačiau ir jos išvestinės reikšmę apskaičiuojantį paprogramį;**
- **Kiekvienos iteracijos metu reikia apskaičiuoti funkcijos ir Jakobi matricos reikšmes;**
- **Rafsono modifikacija: Jakobi matricos reikšmes atnaujinti ne kiekvienoje iteracijoje, tačiau kas tam tikrą skaičių iteracijų. Dėl to konvergavimo sparta sumažėja, tačiau kiekvienoje iteracijoje sumažėja atliekamų veiksmų skaičius. Suminis sprendimo laikas dažniausiai gaunamas mažesnis. Toks sprendimo būdas vadinamas *Niutono-Rafsono metodu***

Jakobio matricos reikšmę galima įvertinti skaitiškai, nenaudojant analitinių diferencijavimo formulių(kvazi-Niutono metodai):

$$\left\{\mathbf{x}^{i+1}\right\} = \left\{\mathbf{x}^{i}\right\} - \left[\left.\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right|_{\mathbf{x}^i}\right]^{-1} \left\{\mathbf{f}\left(\mathbf{x}^{i}\right)\right\}$$

$$\Downarrow$$

$$\frac{\partial f_i}{\partial x_j} = \frac{f_i(x_1, x_2, \ldots, x_j + h, \ldots, x_n) - f_i(x_1, x_2, \ldots, x_j, \ldots, x_n)}{h}$$

Lygčių sistemos atveju kiekvienos iteracijos metu tektų n^2 kartų apskaičiuoti funkcijų reikšmes. Tai per brangu skaičiavimų imlumo požiūriu.

Kvazi-Niutono metodai: Broideno metodas

$$\left\{\mathbf{x}^{i+1}\right\} = \left\{\mathbf{x}^{i}\right\} - \left[\left.\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right|_{\mathbf{x}^i}\right]^{-1} \left\{\mathbf{f}\left(\mathbf{x}^{i}\right)\right\}$$

Niutono iteracijų formulė

Formuluojame *apytikslio Jakobio matricos apskaičiavimo* uždavinį, panašiu principu, kaip ir *kirstinių metode* :

$$\left[\left.\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right|_{\mathbf{x}^i}\right] \approx \left[\mathbf{A}_i\right]$$

$$\left[\mathbf{A}_i\right]\left(\left\{\mathbf{x}^{i}\right\} - \left\{\mathbf{x}^{i-1}\right\}\right) = \left\{\mathbf{f}\left(\mathbf{x}^{i}\right)\right\} - \left\{\mathbf{f}\left(\mathbf{x}^{i-1}\right)\right\}$$

$$\left[\mathbf{A}_i\right]\left\{\mathbf{s}\right\} = \left\{\mathbf{y}\right\}$$

Apskaičiuojama nevienareikšmiškai, n lygčių ir n^2 nežinomųjų

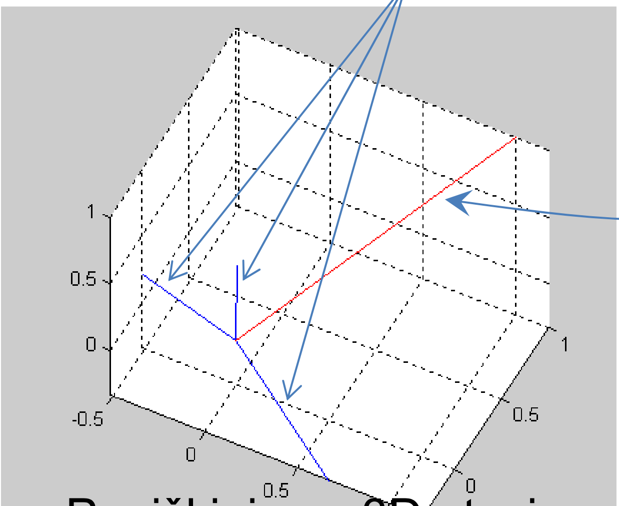
Kvazi-Niutono metodai: Broideno metodas

$$[\mathbf{A}_i]\{\mathbf{s}\} = \{\mathbf{y}\}$$

Papildoma sąlyga: perskaičiuota matrica turi būti galimai artimesnė prieš tai buvusiai. Artumo kriterijų parenkame tokį:

$$[\mathbf{A}_i]\{\mathbf{z}\} = [\mathbf{A}_{i-1}]\{\mathbf{z}\}, \quad visiems\ \{\mathbf{z}\},\ kuriems\ \{\mathbf{s}\}^T\{\mathbf{z}\} = 0$$

parenkame : $\{\mathbf{z}\} = \mathbf{E} - \dfrac{\{\mathbf{s}\}\{\mathbf{s}\}^T}{\{\mathbf{s}\}^T\{\mathbf{s}\}}$ $\quad : \quad \{\mathbf{s}\}^T\{\mathbf{z}\} = \{\mathbf{s}\}^T - \dfrac{\{\mathbf{s}\}^T\{\mathbf{s}\}\{\mathbf{s}\}^T}{\{\mathbf{s}\}^T\{\mathbf{s}\}} = 0$



Paaiškinimas 3D atveju

Tokiu būdu parodome, kad parinkti z ortogonalūs s

Kvazi-Niutono metodai: Broideno metodas

$$\{\mathbf{z}\} = \mathbf{E} - \frac{\{\mathbf{s}\}\{\mathbf{s}\}^T}{\{\mathbf{s}\}^T\{\mathbf{s}\}} \Longrightarrow [\mathbf{A}_i]\{\mathbf{z}\} = [\mathbf{A}_{i-1}]\{\mathbf{z}\}$$

$$\Downarrow$$

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

$$\Downarrow$$

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

$$\Downarrow$$

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

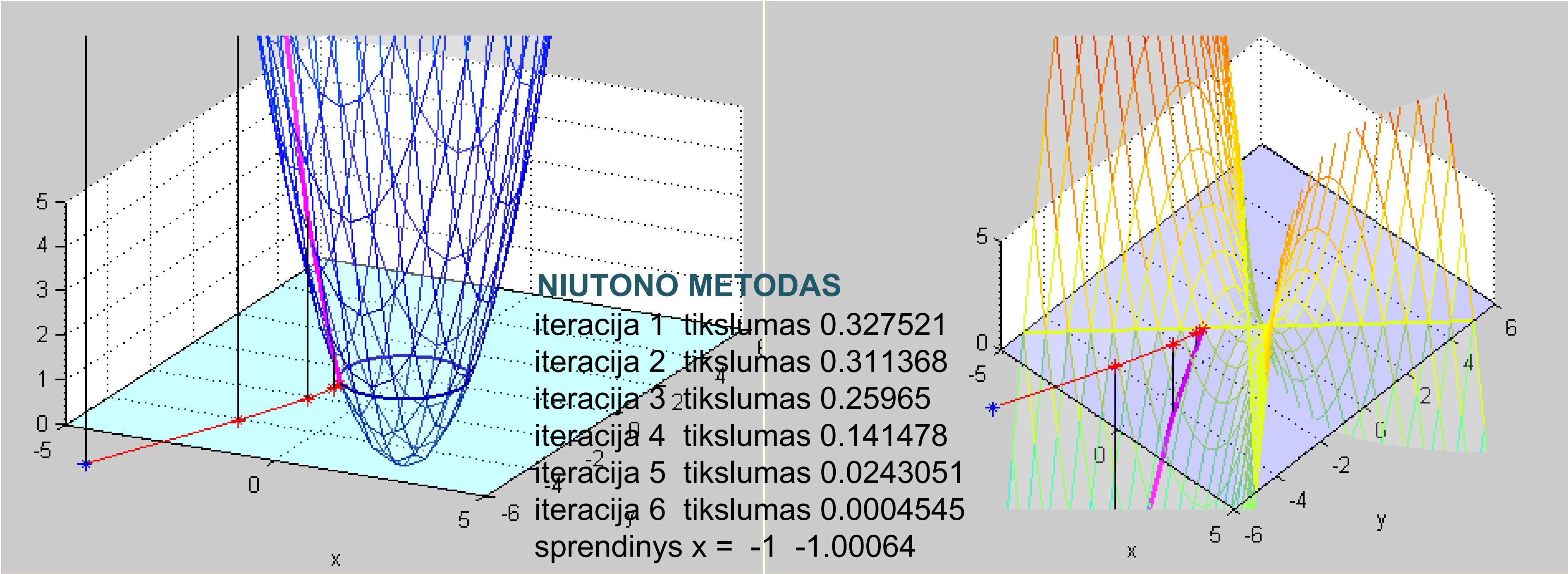
Kvazi-Niutono metodai: Broideno metodas

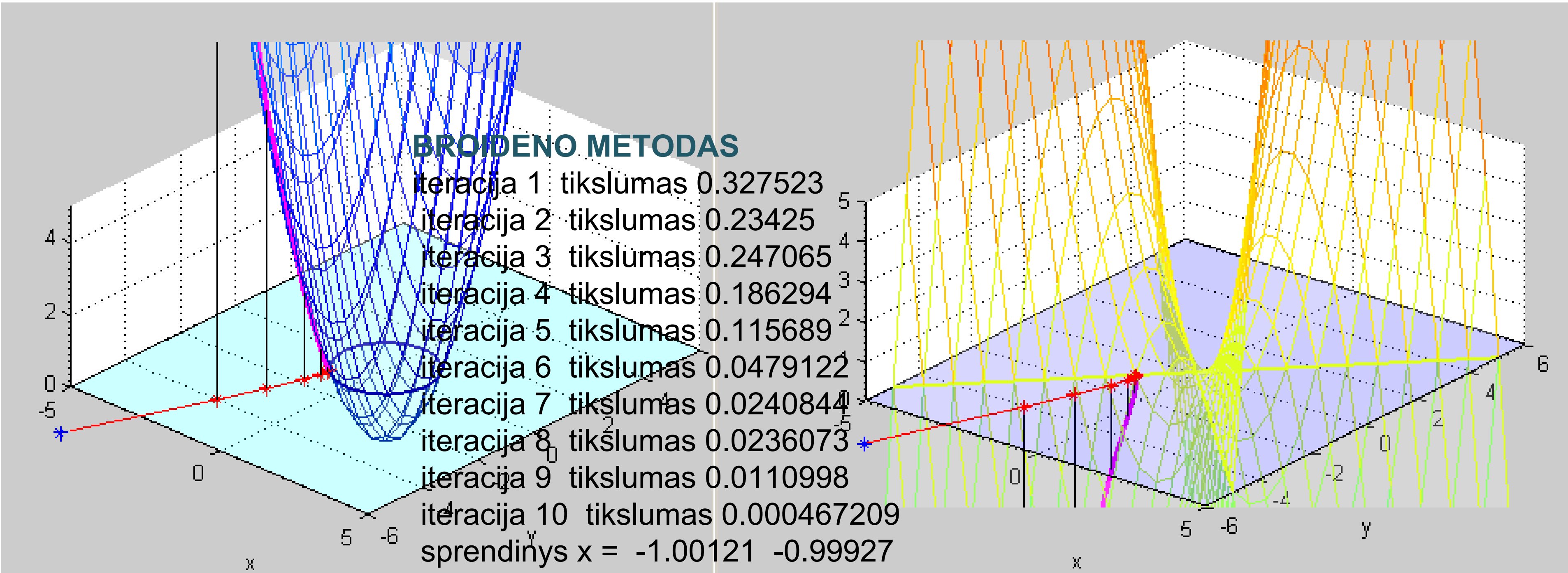
Bendroji formulė, taikoma kiekvienoje iteracijoje:

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

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

Apskaičiavę naują artinį, pagal jo reikšmę atnaujiname ir Jakobio matricą. Vis dėlto, pradinės iteracijos metu A_0 turi būti tam tikru būdu apskaičiuotas (galima pagal analitines formules, arba taikant skaitinio diferencijavimo veiksmą)


NIUTONO METODAS
iteracija 1 tikslumas 0.327521
iteracija 2 tikslumas 0.311368
iteracija 3 tikslumas 0.25965
iteracija 4 tikslumas 0.141478
iteracija 5 tikslumas 0.0243051
iteracija 6 tikslumas 0.0004545
sprendinys x = -1 -1.00064


BROIDENO METODAS
iteracija 1 tikslumas 0.327523
iteracija 2 tikslumas 0.23425
iteracija 3 tikslumas 0.247065
iteracija 4 tikslumas 0.186294
iteracija 5 tikslumas 0.115689
iteracija 6 tikslumas 0.0479122
iteracija 7 tikslumas 0.0240844
iteracija 8 tikslumas 0.0236073
iteracija 9 tikslumas 0.0110998
iteracija 10 tikslumas 0.000467209
sprendinys x = -1.00121 -0.99927

Funkcijos minimizavimu paremti netiesinių lygčių sistemų sprendimo metodai

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

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

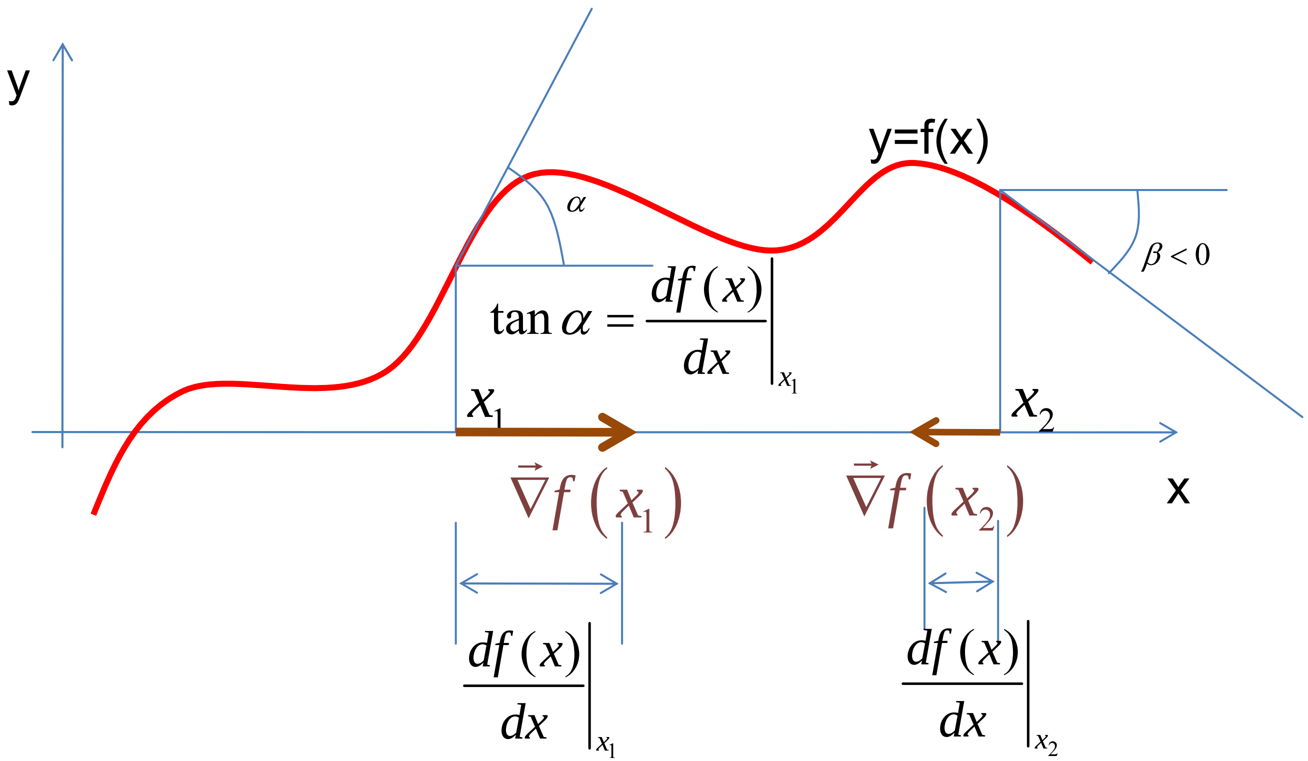
$$\min_{\{\mathbf{x}\}} \Psi\left(\left\{ \mathbf{x} \right\}\right) = \frac{1}{2} \sum_{i=1}^{n} f^2{}_i\left(\left\{ \mathbf{x} \right\}\right) \quad ,$$

Ψ **funkcijos minimumas yra 0 reikšmė, kuri galima tik kai** $\left\{ \mathbf{f}\left(\left\{ \mathbf{x} \right\}\right) \right\} = 0$

Argumentų x reikšmės, kurioms esant gaunama ši minimali Ψ **reikšmė, yra ieškomasis lygčių sistemos sprendinys**

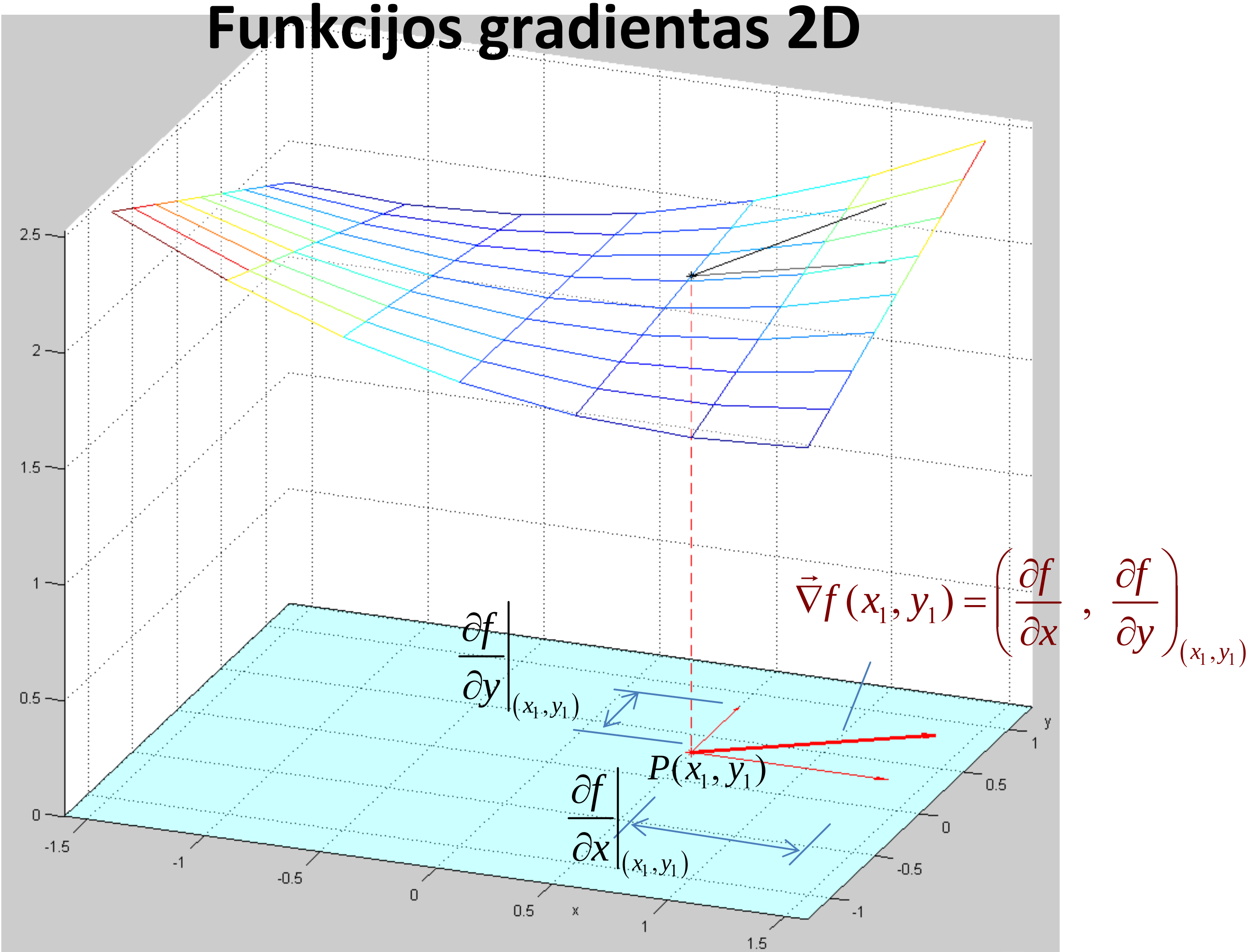
- **Jeigu formuluojant uždavinį visos x reikšmės yra leistinos(t.y. prasmingos), turime *besąlyginio optimizavimo uždavinį;***
- **Jį spręsime taikydami *gradientinius metodus.* Jais nustatome, kaip reikia pakeisti funkcijos argumentų reikšmes, kad funkcijos reikšmė sumažėtų. Minimumo ieškome nuosekliais artiniais (iteracijomis);**
- **Optimizavimu grįsti metodai veikia stabiliau, nei Niutono arba kvazi-Niutono metodai ir dažniausiai nereiklauja labai gero pradinio artinio**

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**

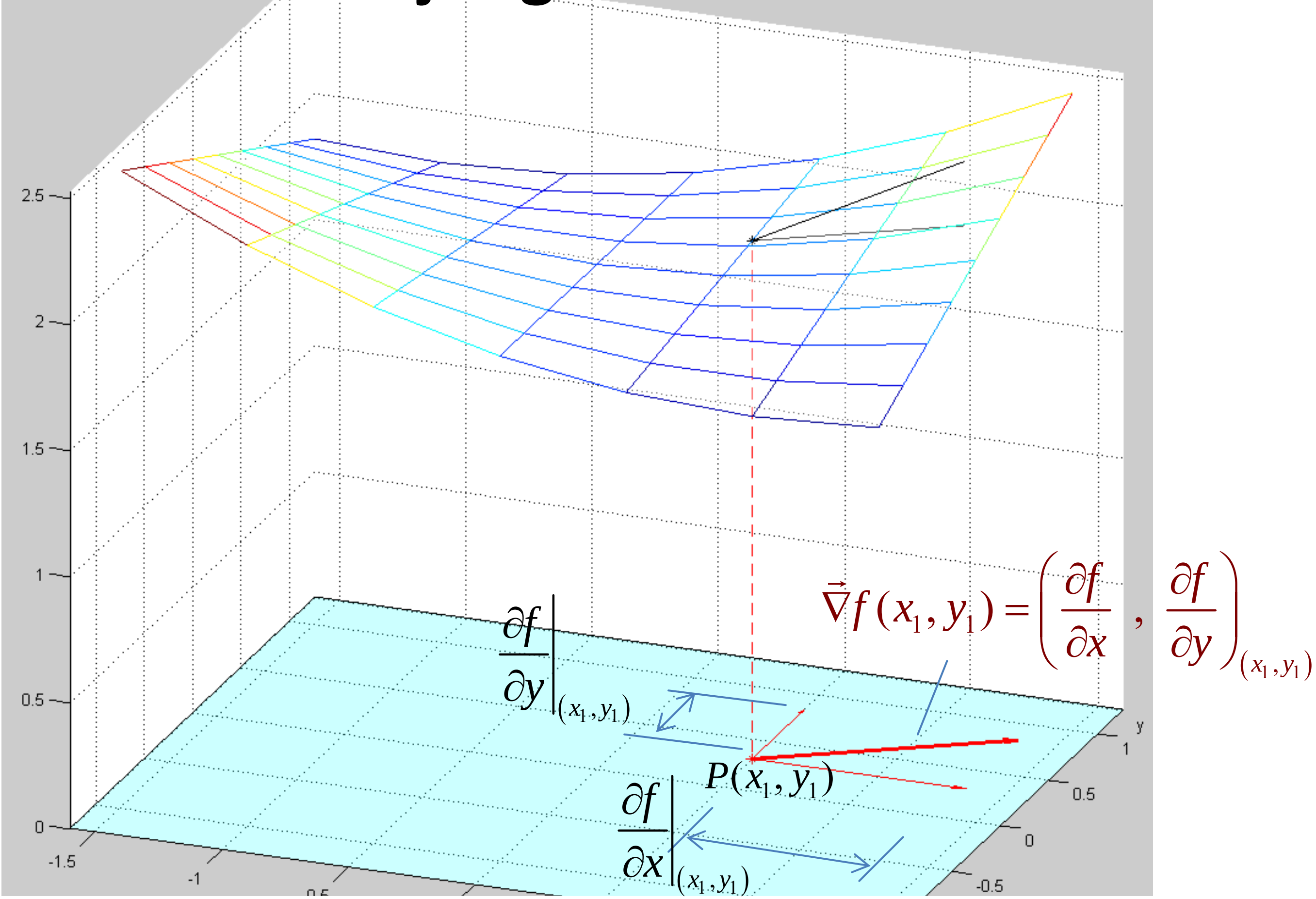
Funkcijos gradientas 2D



•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

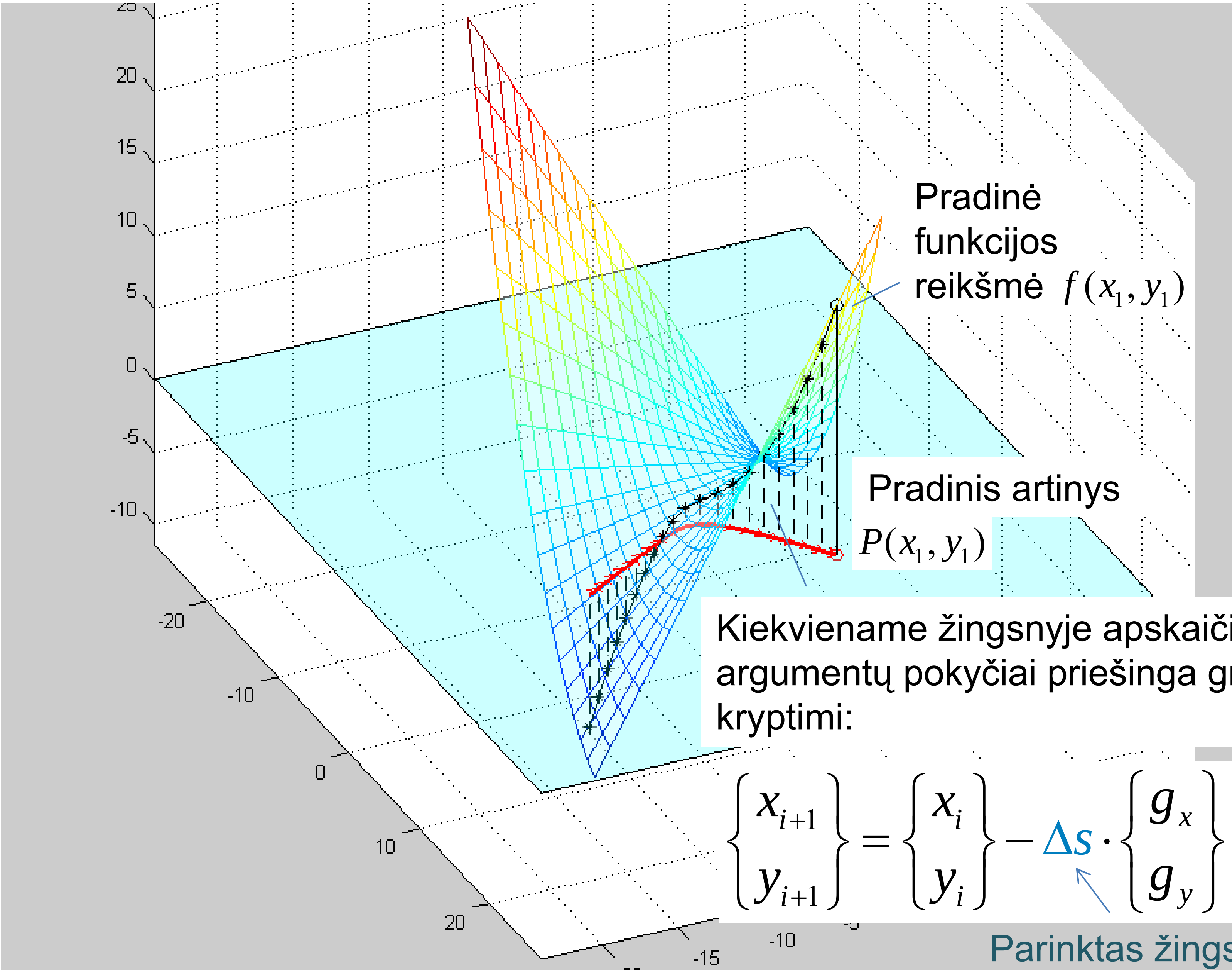


•**Gradiento vektoriaus ilgis *matuojamas skirtingais vienetais, nei funkcijos ir jos argumentų reikšmės. Todėl funkcijos ir jos gradiento vaizdavimo masteliai tame pačiame brėžinyje yra skirtingi;***

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

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

Dviejų argumentų funkcijos minimizavimas, einant priešinga gradientui kryptimi



Pradinė funkcijos reikšmė $f(x_1, y_1)$

Pradinis artinys $P(x_1, y_1)$

Kiekviename žingsnyje apskaičiuojami argumentų pokyčiai priešinga gradientui kryptimi:

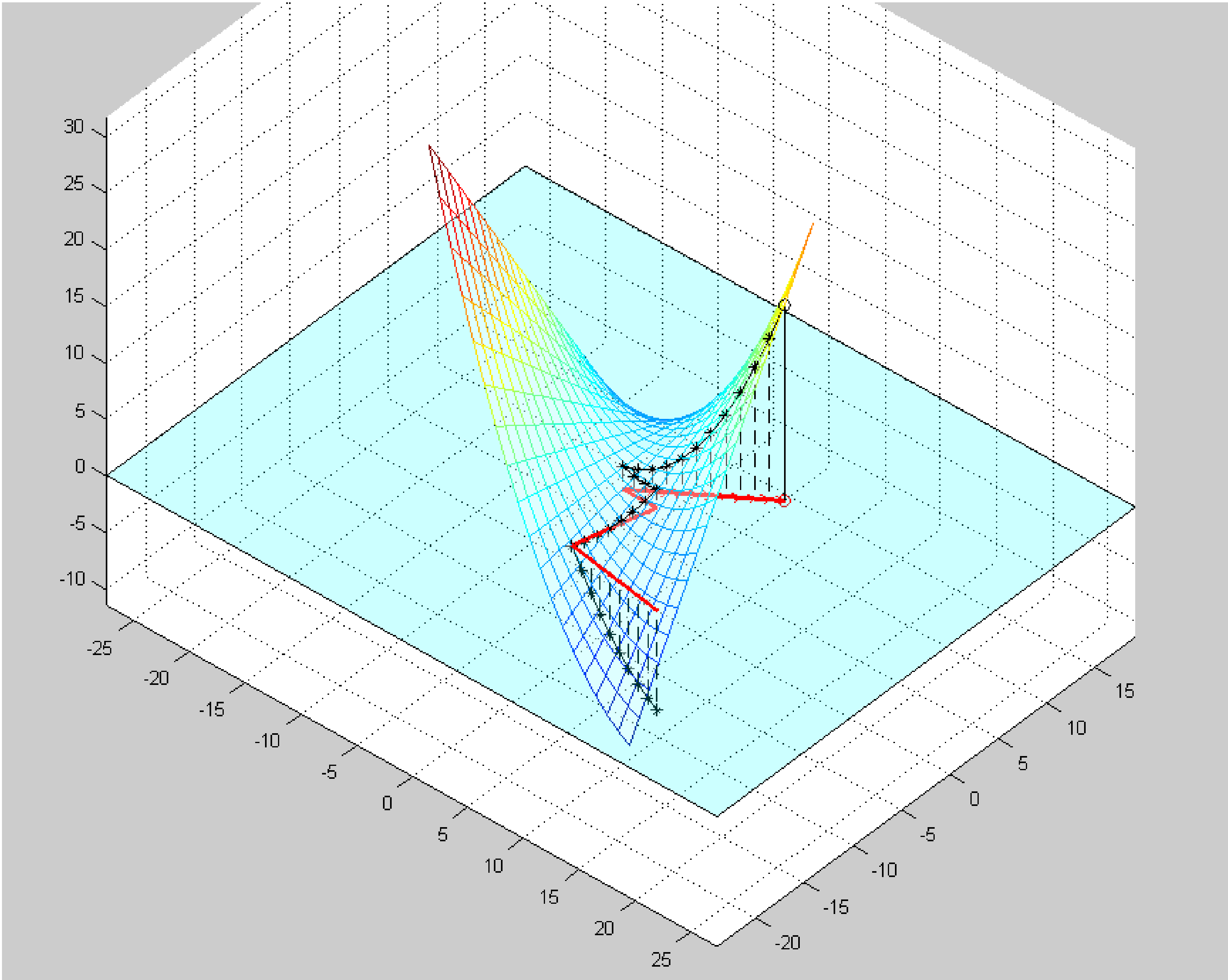
$$\begin{Bmatrix} x_{i+1} \\ y_{i+1} \end{Bmatrix} = \begin{Bmatrix} x_i \\ y_i \end{Bmatrix} - \Delta s \cdot \begin{Bmatrix} g_x \\ g_y \end{Bmatrix}$$

Parinktas žingsnio ilgis

Dviejų argumentų funkcijos minimizavimas

- **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



Daugelio kintamųjų funkcijos minimizavimas

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

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

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

Gradientas yra vektorius-eilutė. Toks užrašas dera su Jakobi 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.

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

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

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

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

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

Gradientas vektorinės funkcijos **f** erdvėje

Vektorinės funkcijos **f** Jakobio matrica vektorinio argumento **x** erdvėje

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

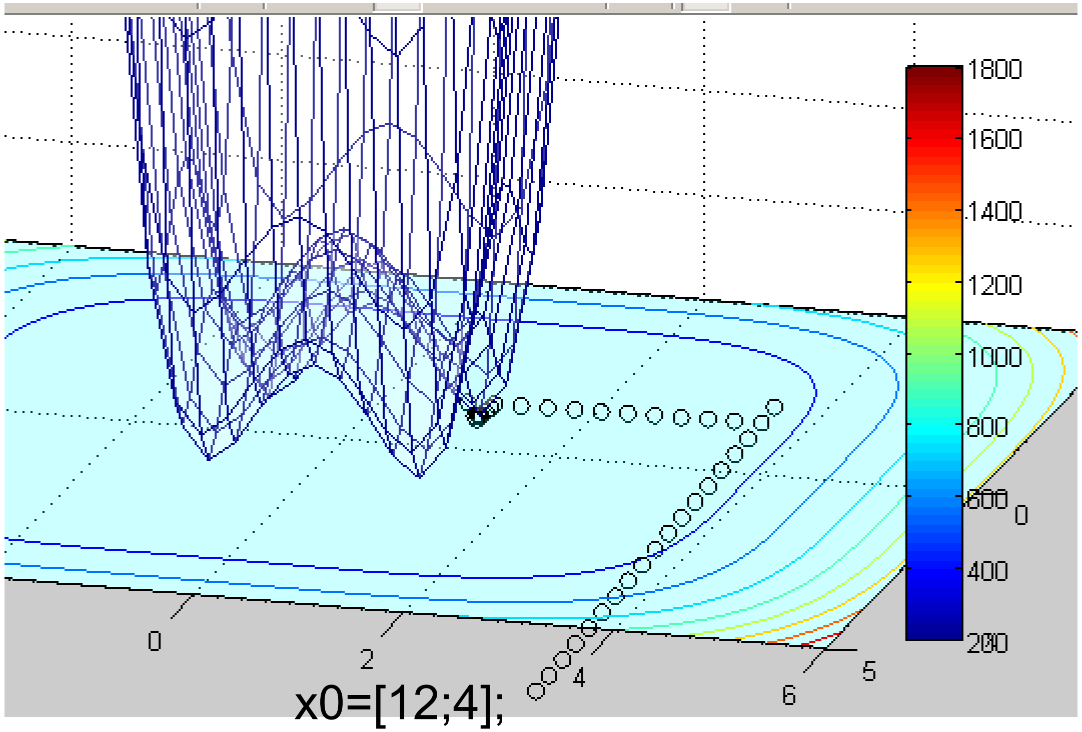
$$\left\{\mathbf{f}\left(\left\{\mathbf{x}\right\}\right)\right\}=0$$

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

$$\frac{\partial\Psi\left(\left\{\mathbf{f}\right\}\right)}{\partial\left\{\mathbf{f}\right\}}=\left\{\begin{matrix}\frac{\partial\Psi}{\partial f_1} & \frac{\partial\Psi}{\partial f_2} & \cdots & \frac{\partial\Psi}{\partial f_n}\end{matrix}\right\}=\left\{\begin{matrix}f_1 & f_2 & \cdots & f_n\end{matrix}\right\}=\left\{\mathbf{f}\right\}^{T};$$

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

Funkcijos minimizavimo metodų yra ir sudėtingesnių bei tobulesnių. Pavyzdžiui, minimizavimo kryptis gali būti apskaičiuota taikant kvazi-Niutono arba kitokius apytikslius priėjimus. Jie nagrinėjami *Optimizavimo metodų* ir jam giminingųose akademiniuose kursuose.



x0=[12;4];

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

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;**
- **Gradientiniai funkcijos minimizavimo metodai gali sustoti, aptikę lokalųjį minimumą. Dviejų kintamųjų funkcijos atveju tai yra įdubimas funkciją vaizduojančiame paviršiuje;**