Vysoká škola strojní a textilní v Liberci

nositelka Řádu práce

Fakulta textilní

Obor 31-20-8

Katedra technické kybernetiky

OPTIMÁLNÍ SEŘIZOVÁNÍ DISKRÉTNÍCH REGULÁTORŮ DLE MINIMA KVADRATICKÉ REGULAČNÍ PLOCHY UZAVŘENÉHO REGULAČNÍHO OBVODU NA SAMOČINNÉM POČÍTAČI

Helena Kantorová

Vedoucí práce: Ing. Osvald Modrlák, KTK VŠST Liberec

KTK ASŘ TF-031

Rozsah práce a příloh

Počet stran: 72

Počet příloh

a tabulek: 5+1

Počet obrázků: 14

Počet výkresů: -

Počet modelů

nebo jiných příloh: -

Vysoká škola: VŠST Liberec Fakulta: textilní

Katedra: technické kybernetiky Školní rok: 1981/82

ZADÁNÍ DIPLOMOVÉ PRÁCE
(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

pro s. Helenu K a n t o r o v o u

obor automatizované systémy řízení ve spotřebním průmyslu

Vedoucí katedry Vám ve smyslu nařízení vlády ČSSR č. 90/1980 Sb., o státních závěrečných zkouškách a státních rigorozních zkouškách, určuje tuto diplomovou práci:

Název tématu: Optimální seřizování diskrétních regulátorů dle minima kvadratické regulační plochy uzavřeného regulačního obvodu na samočinném počítači

Zásady pro vypracování:

1) Seznamte se v literatuře se způsobem optimálního seřizování diskrétních regulátorů dle minima kvadratické regulační plochy v Z-transformaci.

2) Popište optimalizační metody vhodné pro řešení dané úlohy.

3) Vypracujte program umožňující optimální seřízení číslicových regulátorů dle minima kvadratické regulační plochy, jsou-li zadány Z-přenosy jednotlivých členů regulačního obvodu a typy poruch.

4) Dosažené výsůedky ověřte na jednoduchých příkladech.

5) Vypracujte zprávu o programu.

Rozsah grafických prací:

Rozsah průvodní zprávy: 50 - 60 stran

Seznam odborné literatury:

1. Hanuš, B.: Optimalizace systému řízení, VŠST Liberec, 1978.
2. Strejc, V.: Teorie automatického řízení diskrétních systémů, ČVUT-Praha, 1976.
3. CHimmelbau, D.: Príkladnoe nelinejnoe programirovanie. Moskva, 1975.(Překlad).

Časopisy: Regelungstechnik

Vedoucí diplomové práce: Ing. Osvald Modrlák

Datum zadání diplomové práce: 15.9.1981

Termín odevzdání diplomové práce: 4.6.1982

L. S.

Doc.Ing.Ján Alaxin, CSc.
Vedoucí katedry

Doc.Ing.Jáchym Novák, CSc.
Děkan

V Liberci dne 11.9. 1981

P r o h l á š e n í

Místopřísežně prohlašuji, že jsem diplomovou práci vypracovala samostatně s použitím uvedené literatury.

Kantorová Helena

V Liberci, dne 4.6. 1982

Seznam zkratek a použitých symbolů

$\underline{x}$ - vektor

$\underline{A}$ - matice

$A(z)$ - polynom

$f(\underline{x})$ - funkční hodnota v bodě $\underline{x}$

U - úspěšný krok /kap. 2./

N - neúspěšný krok /kap. 2./

O b s a h

1. Ú v o d

Podmínky, v nichž se bude národní hospodářství v osmdesátých letech rozvíjet, vyžadují pohotověji a ve větším rozsahu uplatňovat výsledky vědy a techniky v praxi. Přestože bylo dosaženo některých dobrých výsledků ve zvýšení technické úrovně výroby, je současný stav využívání poznatků vědy a techniky stále nedostatečný. Abychom tento nepříznivý stav změnili, musíme postupně urychlovat rozvoj odvětví, oborů i výrobků méně náročných na surovinové a energetické zdroje, ale o to náročnější na vklad vysoce kvalifikované vědecké a tvůrčí práce. Proto se v Hlavních směrech klade důraz na urychlení rozvoje elektroniky, zvláště mikroelektroniky, jejíž aplikace povede k automatizaci v průmyslu, dopravě, spojích atd.

To však není možné bez soustavného seznamování se s nejnovějšími poznatky vědy a techniky. V současné době se v literatuře objevuje celá řada moderních algoritmů řízení. Vzhledem k tomu, že u soustav vyšších řádů roste pracnost, využívá se k jejich řešení samočinných počítačů. Nasazení číslicového počítače pro řízení představuje dnes nejdokonalejší způsob řízení procesu, neboť dovoluje velmi pružně měnit algoritmy řízení i strukturu řídících obvodů v souvislosti se změnami kriterií řízení, vyvolanými vnějšími okolnostmi. Jedním z nejčastějších úkolů technické praxe je optimální seřízení daného regulátoru vzhledem ke zvolenému kriteriu.

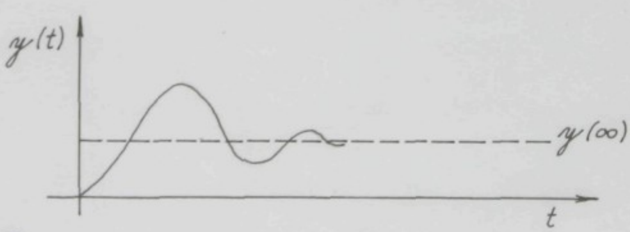
Předložená práce se zabývá optimálním seřizováním diskrétních regulátorů podle minima kvadratické regulační plochy uzavřeného regulačního obvodu na samočinném počítači, jsou-li zadány Z-přenosy jednotlivých členů regulačního obvodu a typy poruch. Dále popsané řešení tohoto problému využívá optimalizační metodu Rosenbrocka rozšířenou Swannem a numerický výpočet integrálu s kontrolou stability. Případ nestability při hledání v daném směru je ošetřen zpětným chodem v tomto směru.

2. Optimální seřízení podle minima kvadratické regulační plochy

Pro optimální řízení technologického a výrobního procesu je nutné mít prostředky pro kvantitativní srovnávání různých postupů. Nejvodnějším takovým prostředkem je matematicky formulované kritérium, které svou hodnotou přímo udává jakost řízení. Optimálním řízením se pak rozumí takové řízení, které dává extrémní hodnotu kritéria jakosti řízení /minimum nebo maximum/.

2.1. Formulace kvadratického kritéria jakosti řízení

Jedním z možných a velmi často používaných kritérií kvality řízení je seřízení podle minima kvadratické regulační plochy. Podle tohoto kritéria má být minimální integrál čtverce odchylek regulované veličiny od její nové ustálené hodnoty.



$$I_S = \int_0^\infty [y(t) - y(\infty)]^2 \, dt = \int_0^\infty e^2(t) \, dt \qquad /2.1.-1/$$

Jeho diskrétní verzí je součet

$$I = \sum_{k=0}^{\infty} [w(k) - y(k)]^2 = \sum_{k=0}^{\infty} e^2(k) \qquad /2.1.-2/$$

Součet /2.1.-2/ lze v komplexní rovině vyjádřit integrálem

$$I = \frac{1}{2\pi i} \oint_{\Gamma_0} E(z)E(z^{-1}) \frac{dz}{z} , \qquad /2.1.-3/$$

kde Γ_0 je integrační dráha podél jednotkové kružnice se středem v počátku komplexní roviny z .

Pokud existuje minimum /2.1.-1/ je toto řešení vždy stabilní. Tento výrok však neplatí o diskrétní verzi /2.1.-2/. Protože nás však zajímá jen takové minimum /2.1.-2/, které je současně stabilní, je třeba stabilitu vhodným způsobem zajistit.

Určování minima integrálu /2.1.-3/ je ekvivalentní s určováním minima součtu /2.1.-2/.

2.2. Výpočet kvadratické regulační plochy v Z-transformaci

Pro určování minima integrálu /2.1.-3/ je nutné najít jeho odhad.

V praxi při analýze dynamického systému získáme integrál obsahující přenosovou funkci $B(z)/A(z)$. Pak /2.1.-3/ lze vyjádřit pomocí polynomů $A(z)$ a $B(z)$.

$$I = \frac{1}{2\pi i} \oint E(z)E(z^{-1}) \frac{dz}{z} = \frac{1}{2\pi i} \oint \frac{B(z)B(z^{-1})}{A(z)A(z^{-1})} \cdot \frac{dz}{z} \qquad /2.2.-1/$$

kde $A(z) = a_0 z^n + a_1 z^{n-1} + \dots + a_n \, , \; a_0 > 0$ /2.2.-2/

$B(z) = b_0 z^n + b_1 z^{n-1} + \dots + b_n$ /2.2.-3/

Je důležité prověřit, má-li jmenovatel $A(z)$ všechny nulové body uvnitř jednotkové kružnice, neboť v opačném případě je dynamický systém nestabilní.

Pro získání výsledků v jednoduché formě zavedeme některá označení. Nechť A^* je polynom vyjádřený formulí

$$A^*(z) = z^n A(z^{-1}) = a_0 + a_1 z + \cdots + a_n z^n . \qquad /2.2.-4/$$

Zavedeme mnohočleny

$$A_k(z) = a_0^k z^k + a_1^k z^{k-1} + \cdots + a_k^k , \qquad /2.2.-5/$$

$$B_k(z) = b_0^k z^k + b_1^k z^{k-1} + \cdots + b_k^k , \qquad /2.2.-6/$$

které se vyjadřují rekurentními formulemi

$$A_{k-1}(z) = z^{-1}\{A_k(z) - \alpha_k A_k^*(z)\} , \qquad /2.2.-7/$$

$$B_{k-1}(z) = z^{-1}\{B_k(z) - \beta_k A_k^*(z)\} , \qquad /2.2.-8/$$

kde

$$\alpha_k = a_k^k / a_0^k , \qquad /2.2.-9/$$

$$\beta_k = b_k^k / a_0^k , \qquad /2.2.-10/$$

$$A_n(z) = A(z) , \qquad /2.2.-11/$$

$$B_n(z) = B(z) . \qquad /2.2.-12/$$

Koeficienty mnohočlenů A_k a B_k jsou zadány rekurentními formulemi

$$a_i^{k-1} = a_i^k - \alpha_k a_{k-i}^k , \quad i = 0, 1, \ldots, k-1 , \qquad /2.2.-13/$$

$$b_i^{k-1} = b_i^k - \beta_k a_{k-i}^k , \quad i = 0, 1, \ldots, k-1 , \qquad /2.2.-14/$$

s počátečními podmínkami

$$a_i^n = a_i , \qquad /2.2.-15/$$

$$b_i^n = b_i . \qquad /2.2.-16/$$

Aby tyto rovnice měly smysl, je nezbytné požadovat pro všechna a_0^k splnění podmínky $a_0^k \neq 0$. Koeficient a_0^n je vždy možné vybrat nenulový. V teorému 2.2.-1 jsou formulovány nutné a postačující podmínky.

Teorém 2.2.-1: Nechť $a_0^k > 0$, tehdy jsou ekvivalentní následující podmínky.

1/ Polynom $A_k(z)$ je stabilní.

2/ Mnohočlen $A_{k-1}(z)$ je stabilní a $a_0^{k-1} > 0$.

Několikanásobným použitím teorému je možné dokázat: Je-li polynom $A_{k-1}(z)$ stabilní, pak všechny koeficienty a_0^{k-1} jsou kladné. Pro důkaz tohoto teorému 2.2.-1 rozebereme následující lemma.

Lemma 2.2.-1: Nechť polynom $f(z)$ s reálnými koeficienty má všechny kořeny uvnitř jednotkové kružnice. Pak

$$|f(z)| < |f^*(z)| \quad \text{pro} \quad |z| < 1$$
$$|f(z)| = |f^*(z)| \quad \text{pro} \quad |z| = 1$$
$$|f(z)| > |f^*(z)| \quad \text{pro} \quad |z| > 1$$

Důkaz: Nechť

$$f(z) = \beta \prod_{k=1}^{n} (z - \alpha_i), \quad |\alpha_i| < 1,$$

pak

$$f^*(z) = \beta \prod_{i=1}^{n} (1 - \alpha_i z).$$

Zavedeme

$$w(z) = \frac{f(z)}{f^*(z)} = \prod_{i=1}^{n} \frac{z - \alpha_i}{1 - \alpha_i z} = \prod_{i=1}^{n} \frac{z - \alpha_i}{1 - \bar{\alpha}_i z},$$

kde $\bar{\alpha}_i$ – konstantní, komplexně sdružené k α_i.

Poslední rovnost vyplývá z toho, že f má proměnné koeficienty. Je-li α_i nulovým bodem polynomu f, pak $\bar{\alpha}_i$ bude také jeho nulovým bodem. Všimneme si transformace

$$w_i(z) = (z - \alpha_i)/(1 - \bar{\alpha}_i z).$$

Ta v sobě zahrnuje vnitřek jednotkové kružnice, to znamená, že jednotková kružnice je invariantou transformace. Transformace

$$w(z) = \prod_{i=1}^{n} w_i(z) = \prod_{i=1}^{n} \frac{z - \alpha_i}{1 - \bar{\alpha}_i z} = \frac{f(z)}{f^*(z)}$$

má také tyto vlastnosti a lemma je dokázáno.

<u>Důkaz teorému 2.2.-1:</u> Nejdříve ukážeme, že z podmínky 1/ vyplývá podmínka 2/. Je-li dodržena podmínka 1/, pak z lemma 2.2.-1 plyne, že

$$|A_k(0)| < |A_k^*(0)| .$$

Ale $A_k(0) = a_k^k$ a $A_k^*(0) = a_0^k$. Odtud

$$|\alpha_k| = |a_k^k / a_0^k| < 1 . \qquad /2.2.-17/$$

Z rovnice /2.2.-13/ získáme

$$a_0^{k-1} = a_0^k - (a_k^k)^2 / a_0^k = [(a_0^k)^2 - (a_k^k)^2] / a_0^k > 0$$

Podle předpokladu teorému je $a_0^k > 0$. Jestliže je $A_k(z)$ stabilní, pak z lemma 2.2.-1 plyne

$$|A_k(z)| \geq |A_k^*(z)| \quad pro \quad |z| \geq 1 .$$

Vezmeme-li do úvahy podmínku /2.2.-17/, získáme

$$|A_k(z)| > |\alpha_k| \cdot |A_k^*(z)| \quad pro \quad |z| \geq 1 .$$

Z formule /2.2.-7/ plyne, že

$$|z| \cdot |A_{k-1}(z)| = |A_k(z) - \alpha_k A_k^*(z)| \geq |A_k(z)| - |\alpha_k| \cdot |A_k^*(z)| > 0$$

pro $|z| \geq 1$.

Toto znamená, že $A_{k-1}(z)$ nemá kořeny vně jednotkové kružnice. Tímto způsobem je 1. část teorému dokázána.

Nyní předpokládejme, že je splněna podmínka 2/. Pak

$$a_0^{k-1} = a_0^k - (a_k^k)^2 / a_0^k = [(a_0^k)^2 - (a_k^k)^2] / a_0^k > 0 .$$

Je-li podle předpokladu $a_0^k > 0$, pak

$$|\alpha_k| = |a_k^k / a_0^k| < 1$$

Z formule /2.2.-7/ plyne, že

$$A_k(z) - \alpha_k A_k^*(z) = z A_{k-1}(z). \qquad /2.2.-18/$$

Odtud

$$z^k A_k(z^{-1}) - \alpha_k z^k A_k^*(z^{-1}) = z^{k-1} A_{k-1}(z^{-1})$$

nebo

$$A_k^*(z) - \alpha_k A_k(z) = A_{k-1}^*(z). \qquad /2.2.-19/$$

Vyloučením $A_k^*(z)$ z výrazů /2.2.-18/ a /2.2.-19/, získáme

$$A_k(z) = \frac{z}{1-\alpha_k^2} A_{k-1}(z) + \frac{\alpha_k}{1-\alpha_k^2} A_{k-1}^*(z).$$

Je-li $|\alpha_k| < 1$, pak takovéto vyloučení je možné vždy.

Pro $|z| \geq 1$ získáme /lemma 2.2.-1/

$$|A_{k-1}(z)| \geq |A_{k-1}^*(z)|.$$

Je-li $|\alpha_k| < 1$, pak pro libovolné z, $|z| \geq 1$, získáme

$$|A_k(z)| \geq \left|\frac{z}{1-\alpha_k^2}\right| \cdot |A_{k-1}(z)| - \left|\frac{\alpha_k}{1-\alpha_k^2}\right| \cdot |A_{k-1}^*(z)| > 0 .$$

Polynom $A_k(z)$ nemá nulové body vně jednotkové kružnice a teorém je dokázán.

Výše je ukázáno, že podmínka $a_0^k > 0$ pro všechna k je nutná pro stabilitu $A(z)$. Dokážeme, že je pravdivé i obrácené tvrzení. Připusťme, že $a_0^k > 0$ pro všechna k. Triviál-

ní polynom je stabilní, je-li . Z

ní polynom A_0 je stabilní, je-li $a_0^k > 0$. Z teorému 2.2.-1 plyne, že A_1 je stabilní. Použijeme-li několikrát teorém 2.2.-1, dokážeme, že polynom A_k je stabilní. Odtud plyne: Má-li polynom A_n všechny nulové body uvnitř jednotkové kružnice, pak $a_0^k > 0$ pro všechna k. Je-li alespoň jeden z koeficientů a_0^k nekladný, je systém s přenosovou funkcí $B(z)/A(z)$ nestabilní. Výsledky obsahuje teorém 2.2.-2.

Teorém 2.2.-2: Nechť $a_0^n > 0$, pak jsou ekvivalentní následující podmínky

1/ $A_n(z)$ je stabilní,

2/ $a_0^k > 0$ pro všechna $k = 0, 1, \ldots, n-1$.

Hlavní výsledek

Ukážeme, že integrál /2.2.-1/ je možné vyčíslit rekurentně. Proto za vedeme integrál

$$I_k = \frac{1}{2\pi i} \oint \frac{B_k(z)\,B_k(z^{-1})}{A_k(z)\,A_k(z^{-1})} \cdot \frac{dz}{z} \qquad /2.2.-20/$$

Z výrazu /2.2.-1/ plyne, že $I = I_n$. Dokážeme teorém 2.2.-3.

Teorém 2.2.-3: Nechť všechny kořeny polynomu $A(z)$ leží uvnitř jednotkové kružnice, pak integrál I_k odpovídá rekurentní rovnici

$$[1 - \alpha_k^2]\, I_{k-1} = I_k - \beta_k^2, \qquad /2.2.-21/$$

$$I_0 = \beta_0^2. \qquad /2.2.-22/$$

Důkaz: Jestliže $A(z)$ má všechny nulové body uvnitř jednotkové kružnice, pak z teorému 2.2.-2 víme, že všechny a_0^k jsou různé od nuly. Takto z formulí /2.2.-7/ a /2.2.-8/ plyne, že všechny polynomy A_k a B_k je možné určit. Dále z teorému 2.2.-2 plyne, že všechny polynomy A_k mají všechny nulové body uvnitř jednotkové kružnice. Tudíž všechny integrály existují.

Pro důkaz teorému použijeme teorém o analytických funkcích. Nejdříve předpokládejme, že polynom $A(z)$ má prosté kořeny, různé od nuly. Integrál /2.2.-20/ je roven součtu hodnot odečtených v pólech funkce $B_k(z)B_k(z^{-1})/\{z A_k(z) A_k(z^{-1})\}$ uvnitř jednotkové kružnice. Jestliže integrál je invariantní vztaženo ke změně proměnných $z \to 1/z$, pak integrál bude roven také součtu hodnot odečtených v bodech vně jednotkové kružnice.

Všimněme si nyní integrálu

$$I_{k-1} = \frac{1}{2\pi i} \oint \frac{B_{k-1}(z) B_{k-1}(z^{-1})}{A_{k-1}(z) A_{k-1}(z^{-1})} \cdot \frac{dz}{z} .$$

Póly výrazu uvnitř integrálu se nachází uvnitř jednotkové kružnice v bodě $z = 0$ a v nulových bodech z_i polynomu $A_{k-1}(z)$. Je-li $A_{k-1}(z_i) = 0$, pak z formulí /2.2.-7/ a /2.2.-4/ vychází

$$A_k(z_i) = \alpha_k A_k^*(z_i) = \alpha_k z_i^k A_k(z_i^{-1}) .$$

Prohlédneme-li si tuto rovnici současně s výrazy /2.2.-7/ a /2.2.-4/ získáme

$$\begin{aligned} A_{k-1}(z_i^{-1}) &= z_i [A_k(z_i^{-1}) - \alpha_k A_k^*(z_i^{-1})] = \\ &= z_i [A_k(z_i^{-1}) - \alpha_k z_i^{-k} A_k(z_i)] = \\ &= (1 - \alpha_k^2) z_i A_k(z_i^{-1}) . \end{aligned}$$

Z formulí /2.2.-4/ a /2.2.-7/ také vychází

$$A_{k-1}^*(z) = A_k^*(z) - \alpha_k A_k(z) .$$

Tedy

$$A^*_{k-1}(0) = A^*_k(0) - \alpha_k A_k(0) = a_0^k - \alpha_k a_k^k = a_0^k(1-\alpha_k^2).$$

Funkce

$$\frac{B_{k-1}(z)B_{k-1}(z^{-1})}{A_{k-1}(z)A_{k-1}(z^{-1})}\cdot\frac{1}{z} = \frac{B_{k-1}(z)B^*_{k-1}(z)}{A_{k-1}(z)A^*_{k-1}(z)}\cdot\frac{1}{z}$$

a

$$\frac{B_{k-1}(z)B_{k-1}(z^{-1})}{A_{k-1}(z)[z(1-\alpha_k^2)A_k(z^{-1})]}\cdot\frac{1}{z} = \frac{B_{k-1}(z)B^*_{k-1}(z)}{A_{k-1}(z)[(1-\alpha_k^2)A^*_k(z)]}\cdot\frac{1}{z}$$

mají uvnitř jednotkové kružnice stejné póly a stejné odečtené hodnoty v těchto pólech. Proto

$$I_{k-1} = \frac{1}{1-\alpha_k^2}\cdot\frac{1}{2\pi i}\oint\frac{B_{k-1}(z)B_{k-1}(z^{-1})}{A_{k-1}(z)A_k(z^{-1})}\cdot\frac{dz}{z^2} =$$

$$= \frac{1}{1-\alpha_k^2}\cdot\frac{1}{2\pi i}\oint\frac{B_{k-1}(z)B_{k-1}(z^{-1})}{A_k(z)A_{k-1}(z^{-1})}\,dz, \qquad /2.2.-23/$$

kde druhou rovnici získáme jako výsledek změny proměnné $z \to 1/z$. Výraz uvnitř integrálu má póly v nulových bodech a z formule /2.2.-7/ vychází

$$A_{k-1}(z^{-1}) = z\{A_k(z^{-1}) - \alpha_k A^*_k(z^{-1})\} = z\{A_k(z^{-1}) - \alpha_k z^{-k} A_k(z)\}.$$

Pro kořeny z_i polynomu $A_k(z)$ získáme

$$A_{k-1}(z_i^{-1}) = z_i A_k(z_i^{-1}).$$

Funkce

$$\frac{B_{k-1}(z)B_{k-1}(z^{-1})}{A_k(z)A_{k-1}(z^{-1})}$$

a

$$\frac{B_{k-1}(z)B_{k-1}(z^{-1})}{A_k(z)A_k(z^{-1})}\cdot\frac{1}{z}=\frac{B_{k-1}(z)B^*_{k-1}(z)}{A_k(z)A^*_k(z)},$$

mají jedny a tytéž póly uvnitř jednotkové kružnice a stejné odečtené hodnoty v těchto pólech. Proto integrály těchto funkcí podél jednotkové kružnice jsou si rovny. Z rovnice /2.2.-23/ získáme

$$I_{k-1}=\frac{1}{1-\alpha_k^2}\cdot\frac{1}{2\pi i}\oint\frac{B_{k-1}(z)B_{k-1}(z^{-1})}{A_k(z)A_k(z^{-1})}\cdot\frac{dz}{z}\ .$$

S přihlédnutím k /2.2.-8/ vychází

$$(1-\alpha_k^2)I_{k-1}=\frac{1}{2\pi i}\oint\frac{[B_k(z)-\beta_k A^*_k(z)]\cdot[B_k(z^{-1})-\beta_k A^*_k(z^{-1})]}{A_k(z)A_k(z^{-1})}\times$$

$$\times\frac{dz}{z}=\frac{1}{2\pi i}\oint\frac{B_k(z)B_k(z^{-1})}{A_k(z)A_k(z^{-1})}\cdot\frac{dz}{z}-$$

$$-\frac{\beta_k}{2\pi i}\oint\frac{B_k(z)A^*_k(z^{-1})}{A_k(z)A_k(z^{-1})}\cdot\frac{dz}{z}-$$

$$-\frac{\beta_k}{2\pi i}\oint\frac{A^*_k(z)B_k(z^{-1})}{A_k(z)A_k(z^{-1})}\cdot\frac{dz}{z}+$$

$$+\frac{\beta_k^2}{2\pi i}\oint\frac{A^*_k(z)A^*_k(z^{-1})}{A_k(z)A_k(z^{-1})}\cdot\frac{dz}{z}\ . \qquad /2.2.-24/$$

První integrál je roven I_k. Druhý integrálje možné přepsat následujícím způsobem:

$$\frac{\beta_k}{2\pi i}\oint\frac{B_k(z)A^*_k(z^{-1})}{A_k(z)A_k(z^{-1})}\cdot\frac{dz}{z}=\frac{\beta_k}{2\pi i}\oint\frac{B_k(z)A_k(z)}{A_k(z)A^*_k(z)}\cdot\frac{dz}{z}=$$

$$= \frac{\beta_k}{2\pi i} \oint \frac{B_k(z)}{A_k^*(z)} \cdot \frac{dz}{z} = \beta_k \frac{B_k(0)}{A_k^*(0)} = \beta_k \frac{b_k^k}{a_0^k} = \beta_k^2 ,$$

kde první rovnost plyne z formule /2.2.-4/, třetí z teorému o odečtených hodnotách a pátá z formule /2.2.-10/. Analogicky lze dokázat, že třetí integrál v pravé části výrazu /2.2.-24/ je také roven β_k^2.

Použitím formule /2.2.-4/ přepíšeme čtvrtý člen pravé části výrazu /2.2.-24/:

$$\frac{\beta_k^2}{2\pi i} \oint \frac{A_k^*(z)\, A_k^*(z^{-1})}{A_k(z)\, A_k(z^{-1})} \cdot \frac{dz}{z} = \frac{\beta_k^2}{2\pi i} \oint \frac{dz}{z} = \beta_k^2 .$$

Jako výsledek získáme výraz /2.2.-21/. Pro $k=0$ z výrazu /2.2.-20/ dostaneme

$$I_0 = \frac{1}{2\pi i} \oint \left(\frac{b_0^0}{a_0^0} \right)^2 \cdot \frac{dz}{z} = \beta_0^2 .$$

Tím jsou dokázány formule /2.2.-21/ a /2.2.-22/ pro případ, kdy $A(z)$ má prosté kořeny. Má-li A násobné nebo nulové ko kořeny, pak je vždy možné změnit koeficienty tak, aby nový polynom měl jednoduché nenulové kořeny. V tom případě rovnice /2.2.-21/ a /2.2.-22/ budou také správné. Jestliže α_k a β_k jsou spojité funkce parametrů, pak výrazy /2.2.-20/ a /2.2.-21/ mají uplatnění i tehdy, má-li A násobné kořeny.

Všimneme si že z výrazu /2.2.-13/ plyne

$$a_0^{k-1} = a_0^k - \alpha_k a_k^k = a_0^k (1-\alpha_k^2) .$$

Rovnici /2.2.-21/ lze zapsat ve tvaru

$$a_0^{k-1} I_{k-1} = a_0^k I_k - a_0^k \beta_k^2$$

nebo

$$a_0^k I_k = a_0^{k-1} I_{k-1} + \beta_k b_k^k = a_0^{k-1} I_{k-1} + (b_k^k)^2 / a_0^k \, .$$

Závěr: Integrál I_k se vyjadřuje vztahem

$$I_k = \frac{1}{a_0^k} \sum_{i=0}^{k} \frac{(b_i^i)^2}{a_0^i} \, . \qquad /2.2.-25/$$

3. Vybrané optimalizační metody

Máme-li nalézt extrémní hodnotu kritéria jakosti řízení, znamená to, že musíme nejdříve zvolit vhodnou optimalizační metodu. Vzhledem k tomu, že při vlastním řešení optimálního seřízení diskrétního regulátoru budeme používat jako účelovou funkci výraz /2.2.-25/, zvolíme některou z metod přímého hledání extrému /nevyžadují derivace účelové funkce/. Tyto metody se spokojí jen s výpočty hodnot funkce. Vyžadují sice větší počet postupných výpočtových kroků, ale výpočty jsou jednodušší a nepotřebují zvláštní analytickou přípravu. Gradientové metody /vyžadují derivace účelové funkce/ vedou k optimu obyčejně v méně krocích než metody přímého hledání. Ovšem z praktického hlediska je třeba dát přednost jednodušší vyhledávací metodě z následujících důvodů :

- Gradientové metody potřebují mnohem delší dobu na přípravu úlohy k řešení
- Derivaci účelové funkce nelze často určit nebo velice nepřesně
- Gradientové metody jsou často citlivé na volbu počátečních hodnot parametrů
- Gradientové metody jsou většinou numericky velmi citlivé a vyžadují proto vysokou přesnost tj. velkou délku slov počítače

Všechny dále popsané metody jsou pro ilustraci a srovnání různých strategií použity při řešení příkladu :

$$f(\underline{x}) = (0.5x_1 - 1)^2 + (x_2 - 1)^2$$

Příklad řešený metodou zvolenou pro optimální seřízení diskrétního regulátoru /metoda Rosenbrocka rozšířená Swannem/ byl vyřešen i na počítači.

3.1. Metoda zkušebního kroku

Strategie hledání je logicky prostá, používá apriorní údaje a zároveň zavrhuje zastaralé informace o uložení účelové funkce v E^n. Princip metody spočívá v postupných změnách všech složek vektoru $\underline{x}$ pro zjišťování, ve kterém směru těchto změn se účelová funkce zmenšuje. V tomto směru se pak provádí hledání minima. Výsledek se ověřuje opět zkušebními kroky. Sledovaný algoritmus se skládá z následujících operací:

1/ Zvolí se výchozí bod $\underline{x}^0$ a velikost přírůstků pro všechny složky $\Delta\underline{x}^0$.

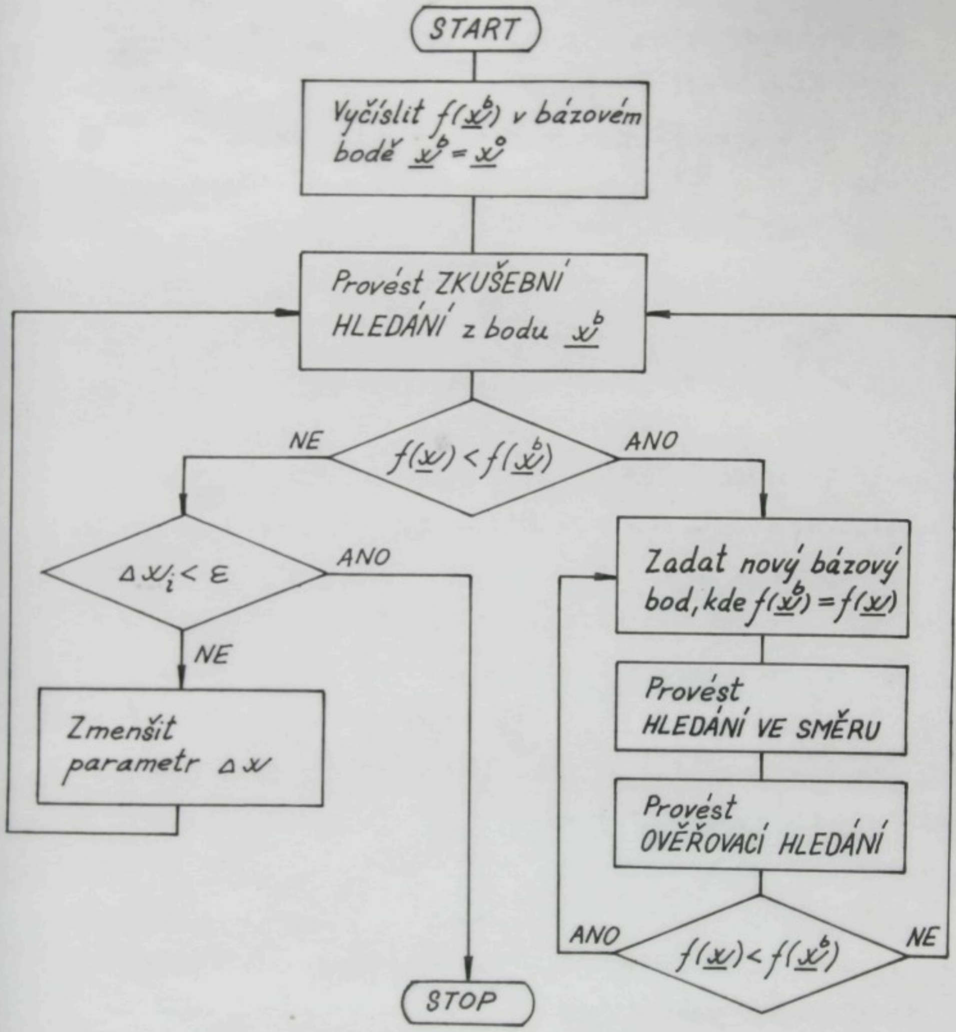
2/ <u>Zkušební hledání.</u> Vyčíslí se hodnota funkce $f(\underline{x}^0)$ v bázovém bodě $\underline{x}^0$. Potom se cyklicky mění vždy jedna složka vektoru $\underline{x}^0$ o přírůstek zvolené velikosti, dokud tímto způsobem nebudou změněny všechny složky vektoru. Konkrétně: x_1^0 se mění v $x_1^1 = x_1^0 - \Delta x_1^0$. Jestliže přírůstek nezlepší hodnotu účelové funkce, x_1^0 se mění o $-\Delta\underline{x}^0$ a prověřuje se hodnota $f(\underline{x})$ jako dříve. Jestliže se hodnota nezlepšila ani nyní, pak x_1^0 zůstane beze změny. Potom x_2^0 změníme o veličinu Δx_2^0 atd., dokud nebudou změněny všechny nezávisle proměnné.
V každém kroku se hodnota účelové funkce srovnává se svou hodnotou v předcházejícím bodě. Jestliže se hodnota účelové funkce v daném kroku zlepšila, pak se její stará hodnota

změní v novou. Jestliže provedená změna $\underline{x}$ je neúspěšná, ponechá se původní hodnota $f(\underline{x})$.

3/ Hledání ve směru. Úspěšné změny proměnných ve zkušebním hledání /tj. změny proměnných, které zmenšily $f(\underline{x})$/ určují vektor v E^n, který ukazuje některé směry jdoucí k minimu, které mohou vést k úspěchu. Hledání ve směru tohoto vektoru se provádí dokud se $f(\underline{x})$ zmenšuje. Pro zrychlení procesu optimalizace se uskutečňuje změna rozměru kroku $\Delta \underline{x}$ zavedením činitele u veličiny $\Delta \underline{x}$, používané v následujících cyklech. Úspěch nebo neúspěch hledání ve směru nelze stanovit, dokud není skončeno ověřovací hledání.

4/ Ověřovací hledání. To je zkušební hledání následující za hledáním ve směru. Jestliže $f(\underline{x})$ se v procesu ověřovacího hledání nezmenšuje, znamená to, že hledání ve směru je neúspěšné a provádí se nové zkušební hledání pro určení nového úspěšného směru. Jestliže zkušební hledání nedá nový úspěšný směr, pak se postupně zmenšuje $\Delta \underline{x}$ dokud: Buď bude možné určit nový úspěšný směr, nebo Δx_i nebude menší než některá dříve stanovená přípustná hodnota. Nemožnost zmenšit $f(\underline{x})$, je-li $\Delta \underline{x}$ dostatečně malé, ukazuje na to, že dosáhnout lokální optimum není možné.

Popsaná posloupnost hledání končí, jestliže se ukazují uspokojujícími podmínky všech základních testů. První test se provádí po každém zkušebním hledání, hledání ve směru a ověřovacím hledání: Změna účelové funkce se srovnává s dříve stanovenou malou veličinou. Jestliže hodnota účelové funkce se neliší od předcházející o hodnotu větší než toto číslo, hledání se počítá jako neúspěšné. V opačném případě


START
Vyčíslit f(x^b) v bázovém bodě x^b = x^0
Provést ZKUŠEBNÍ HLEDÁNÍ z bodu x^b
NE
f(x) < f(x^b)
ANO
Δx_i < ε
ANO
NE
Zmenšit parametr Δx
Zadat nový bázový bod, kde f(x^b) = f(x)
Provést HLEDÁNÍ VE SMĚRU
Provést OVĚŘOVACÍ HLEDÁNÍ
ANO
f(x) < f(x^b)
NE
STOP

se provádí test pro definování zvětšila-li se účelová funkce /neúspěch/ nebo zmenšila /úspěch/. Tento druhý test je nutný proto, abychom si ověřili, jestli se hodnota účelové funkce stále zlepšuje. Třetí test se provádí po neúspěchu ve zkušebním hledání ve stadiu zmenšení $\Delta \underline{x}$. Hledání může být skončeno, jestliže v daném kroku změna každé proměnné Δx_i^k je menší než některé dříve stanovené číslo.

Příklad: Minimalizujte účelovou funkci

$$f(x) = (0.5\,x_1 - 1)^2 + (x_2 - 1)^2$$

Začínáme z bodu $\underline{x}^0 = [4.50 \;\; 2.00]^T$ s počátečním $\Delta \underline{x} = [0.6 \;\; 0.84]^T$

Výchozí hodnota $f(4.50 \;\; 2.00)$ v bázovém bodě $\underline{x}^0$ je rovna 2.5625. Nejdříve se provede zkušební hledání pro určení úspěšného směru.

$$x_1^1 = 4.50 + 0.60 = 5.10 \qquad f(5.10 \;\; 2.00) = 3.4025 \qquad N$$

$$x_1^1 = 4.50 - 0.60 = 3.90 \qquad f(3.90 \;\; 2.00) = 1.9025 \qquad U$$

$$x_2^1 = 2.00 + 0.84 = 2.84 \qquad f(3.90 \;\; 2.84) = 4.2881 \qquad N$$

$$x_2^1 = 2.00 - 0.84 = 1.16 \qquad f(3.90 \;\; 1.16) = 0.9281 \qquad U$$

Zkušební hledání bylo úspěšné. Připomeňme, že při každém hledání se vybírá poslední úspěšný vektor $\underline{x}$. Nový bázový vektor bude $[3.90 \;\; 1.16]^T$. Nyní se z bodu $[3.90 \;\; 1.16]^T$ provede hledání ve směru v souladu s pravidlem

$$x_i^{k+1} = 2\,x_i^k - x_i^b$$

kde x_i^b - předcházející bázový vektor $\underline{x}$. V daném případě je to počáteční vektor $\underline{x}^0$.

$$x_1^2 = 2(3.90) - 4.50 = 3.30$$

$$x_2^2 = 2(1.16) - 2.00 = 0.32 \qquad f(3.30\ 0.32) = 0.8849$$

Nakonec se provede ověřovací hledání, jehož neúspěch či úspěch určíme srovnáním s $f(3.30\ 0.32) = 0.8849$.

$$x_1^3 = 3.30 + 0.60 = 3.90 \qquad f(3.90\ 0.32) = 1.3649 \qquad N$$

$$x_1^3 = 3.30 - 0.60 = 2.70 \qquad f(2.70\ 0.32) = 0.5849 \qquad U$$

$$x_2^3 = 0.32 + 0.84 = 1.16 \qquad f(2.70\ 1.16) = 0.1481 \qquad U$$

Aby bylo možné říci, bylo-li hledání ve směru úspěšné, srovnává se $f(0.70\ 1.16) = 0.1481$ s $f(3.90\ 1.16) = 0.9281$. Protože bylo hledání ve směru úspěšné, bude nový bázový bod $\underline{x}^3 = [2.70\ \ 1.16]^T$; přičemž starý bázový bod je představován vektorem $\underline{x}^1 = [3.90\ \ 1.16]^T$. Nyní znovu provedeme hledání ve směru.

$$x_1^4 = 2(2.70) - 3.90 = 1.50$$

$$x_2^4 = 2(1.16) - 1.16 = 1.16 \qquad f(1.50\ 1.16) = 0.0881$$

Po tomto se provede ověřovací hledání.

$$x_1^5 = 1.50 + 0.60 = 2.10 \qquad f(2.10\ 1.16) = 0.0281 \qquad U$$

$$x_2^5 = 1.16 + 0.84 = 2.00 \qquad f(2.10\ 2.00) = 1.0025 \qquad N$$

$$x_2^5 = 1.16 - 0.84 = 0.32 \qquad f(2.10\ 0.32) = 0.4786 \qquad N$$

Protože $f(2.10\ 1.16) = 0.0281 < f(1.50\ 1.16) = 0.0881$, hledání ve směru je úspěšné a $\underline{x}^5 = [2.10\ 1.16]^T$ bude novýmbázovým bodem a $\underline{x}^3$ starým bázovým bodem.

Tato posloupnost hledání pokračuje do té doby, dokud nebude dosažena situace, ve které na konci ověřovacího hledání hodnota $f(\underline{x})$ bude větší než hodnota $f(\underline{x}^b)$ v posledním bázovém bodě. I když ověřovací hledání bude úspěšné při jedné nebo více změnách, tehdy říkáme, že poslední hledání ve směru je neúspěšné a provádí se z předcházejícího bázového bodu zkušební hledání pro určení nového úspěšného směru. Pro ilustraci pokračujeme ve hledání z bodu $\underline{x}^5 = [2.10 \;\; 1.16]^T$.

Hledání ve směru:

$$x_1^6 = 2(2.10) - 1.50 = 2.70$$

$$x_2^6 = 2(1.16) - 1.16 = 1.16 \qquad f(2.70 \;\; 1.16) = 0.1481$$

Ověřovací hledání:

$$x_1^7 = 2.70 + 0.60 = 3.30 \qquad f(3.30 \;\; 1.16) = 0.4481 \qquad N$$

$$x_1^7 = 2.70 - 0.60 = 2.10 \qquad f(2.10 \;\; 1.16) = 0.0281 \qquad U$$

$$x_2^7 = 1.16 + 0.84 = 2.00 \qquad f(2.10 \;\; 2.00) = 1.0025 \qquad N$$

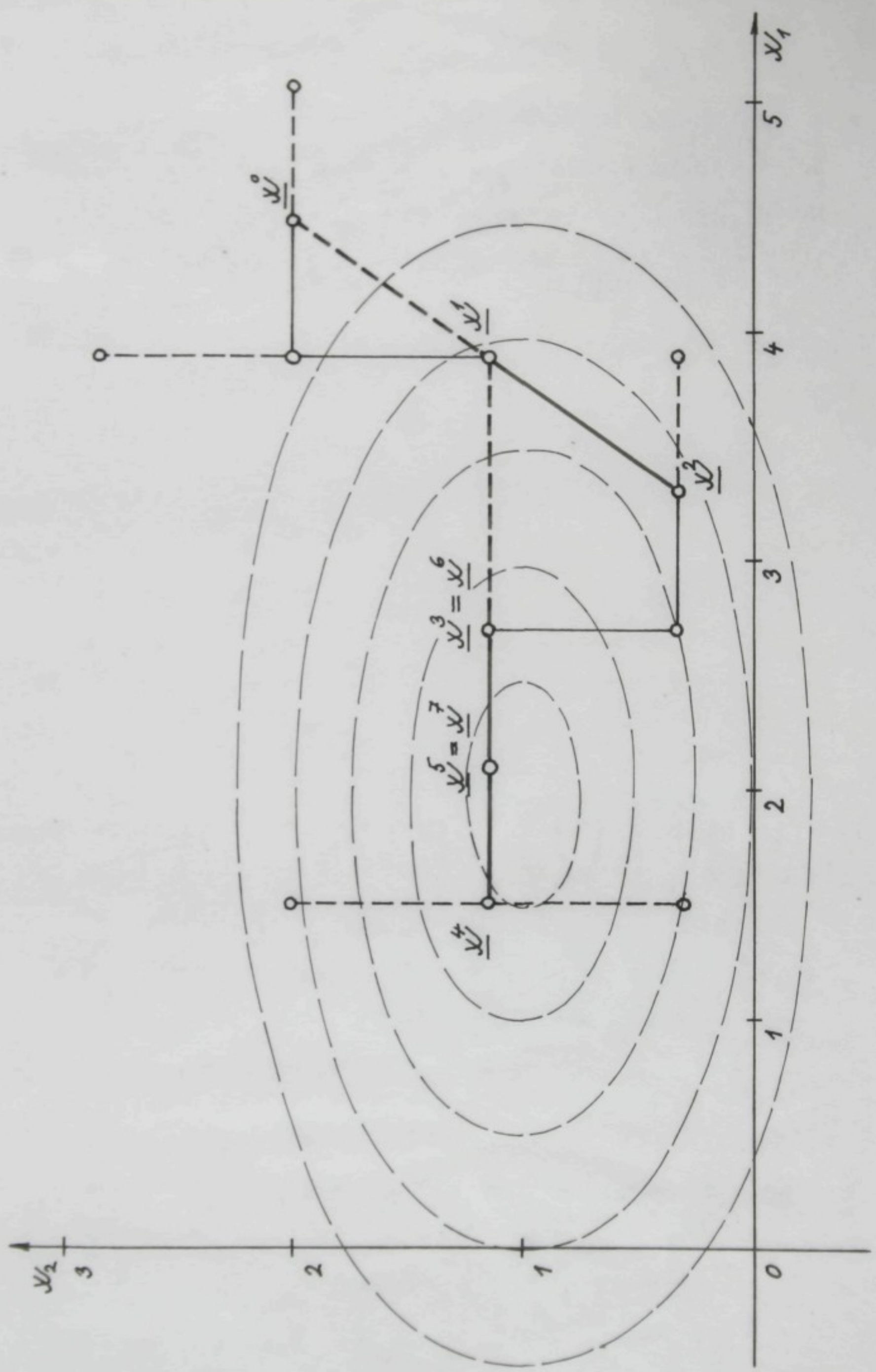
$$x_2^7 = 1.16 - 0.84 = 0.32 \qquad f(2.10 \;\; 0.32) = 0.4786 \qquad N$$

Protože $f(2.10 \;\; 1.16) = 0.0281 = f(2.10 \;\; 1.16) = 0.0281$, nehledě na to, že ověřovací hledání bylo úspěšné, hledání ve směru je neúspěšné a z $\underline{x}^5 = [2.10 \;\; 1.16]^T$ začíná zkušební hledání.

Když dosáhneme stadia, ve kterém ani zkušební hledání, ani hledání ve směru, ani ověřovací hledání nejsou úspěšná v libovolném osovém směru, říkáme, že jsou bezvýsledná a $\Delta \underline{x}$ zmenšíme následujícím způsobem:

$$\Delta x_{i,\text{NOVÉ}} = \Delta x_{i,\text{PŘEDCHÁZEJÍCÍ}} \cdot \Delta x_i^0 / e^{\eta},$$

kde η = počet po sobě následujících neúspěchů ve zkušebních hledáních při dané velikosti kroku, začínajíc od posledního úspěšného zkušebního hledání.



x_1
x_2
0
1
2
3
4
5
$\underline{x}^0$
$\underline{x}^1$
$\underline{x}^2$
$\underline{x}^3 = \underline{x}^6$
$\underline{x}^4$
$\underline{x}^5 = \underline{x}^7$

3.2. Metoda deformovaného polyedru

Tato metoda je poněkud složitější ve srovnání s předcházející metodou, ale ukázala se efektivnější a lehce programovatelná. Připomeňme, že regulární polyedry v E^n jsou simplexy. Na příklad pro případ dvou proměnných regulární simplex představuje rovnostranný trojúhelník /3 body/, v případě třech proměnných představuje čtyřstěn /4 body/ atd.

Při hledání minima účelové funkce $f(\underline{x})$ zkušební vektory $\underline{x}$ mohou být vybrány v bodech E^n, nacházejících se ve vrcholech simplexu. Z analytické geometrie je známé, že souřadnice vrcholů regulárního simplexu se dají vyjádřit následující maticí $\underline{D}$, ve které sloupce představují vrcholy očíslované od 1 do $(n+1)$ a řádky - souřadnice / i nabývá hodnot od 1 do n/.

$$\underline{D} = \begin{bmatrix} 0 & d_1 & d_2 & \dots & d_2 \\ 0 & d_2 & d_1 & \dots & d_2 \\ 0 & d_2 & d_2 & \dots & d_2 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & d_2 & d_2 & \dots & d_1 \end{bmatrix}$$

Kde

$$d_1 = \frac{t}{n\sqrt{2}}\left(\sqrt{n+1} + n - 1\right) \qquad d_2 = \frac{t}{n\sqrt{2}}\left(\sqrt{n+1} - 1\right)$$

t - vzdálenost mezi dvěma vrcholy. Např. pro $n=2$ a $t=1$ /trojúhelník/ má příslušná matice tvar

$$\underline{D} = \begin{bmatrix} 0 & 0{,}965 & 0{,}259 \\ 0 & 0{,}259 & 0{,}965 \end{bmatrix}$$

Účelovou funkci vyčíslíme v každém z vrcholů simplexu. Z vrcholu, kde je účelová funkce maximální, se vede přímka procházející těžištěm simplexu. Potom tento bod vyloučíme a sestrojíme nový simplex, nazvaný "odražený" ze zbylých původních bodů a jednoho nového bodu ležícího v odpovídající vzdálenosti na přímce procházející těžištěm.

Určité praktické obtíže, se kterými se setkáváme při používání regulárních simplexů, a to nedostatečné zrychlení hledání a obtížnost při hledání na zakřivených "výmolech a hřbetech" vedli nutně k některým zlepšeným metodám. V popisované metodě může simplex měnit svou formu, nebude tedy simplexem. Proto budeme používat přiléhavější název "deformovaný polyedr".

Minimalizujeme funkci n nezávisle proměnných s použitím $(n+1)$ vrcholů deformovaného polyedru v E^n. Každý vrchol může být identifikován vektorem $\underline{x}$. Vrchol /bod/ v E^n, ve kterém je hodnota maximální, přechází přes těžiště do zůstávajících vrcholů. Zlepšení hodnoty účelové funkce dosáhneme následnou záměnou bodu s maximální hodnotou $f(\underline{x})$ v "lepší" body dokud nebude nalezeno minimum $f(\underline{x})$. Podrobněji lze tento algoritmus popsat následujícím způsobem.

Nechť $\underline{x}_i^k = [x_{i1}^k, \ldots, x_{ij}^k, \ldots, x_{in}^k]^T$, $i=1,\ldots,n+1$ je i - tým vrcholem /bodem/ v E^n v k - té etapě hledání, $k=0,1,\ldots$ a nechť hodnota účelové funkce v $\underline{x}_i^k$ je rovna $f(\underline{x}_i^k)$. Kromě toho označíme ty vektory $\underline{x}$ polyedru, které dávají maximální a minimální hodnoty $f(\underline{x})$. Určíme

$$f(\underline{x}_h^k) = \max\{f(\underline{x}_1^k), \ldots, f(\underline{x}_{n+1}^k)\}, \text{ kde } \underline{x}_h^k = \underline{x}_i^k$$

$$f(\underline{x}_l^k) = \min\{f(\underline{x}_1^k), \ldots, f(\underline{x}_{n+1}^k)\}, \text{ kde } \underline{x}_l^k = \underline{x}_i^k .$$

Protože polyedr v E^n se skládá z $(n+1)$ vrcholů $\underline{x}_1, \ldots, \underline{x}_{n+1}$, nechť $\underline{x}_{n+2}$ bude těžiště všech vrcholů mimo $\underline{x}_h^k$. Pak souřadnice tohoto těžiště vyjádříme formulí:

$$x_{n+2,j}^k = \frac{1}{n} \cdot \left[\left(\sum_{i=1}^{n+1} x_{ij}^k \right) - x_{hj}^k \right], \quad j = 1, \ldots, n \qquad /3.2.-1/$$

kde index j označuje souřadnicové směry.

Výchozí polyedr obvykle vybíráme ve tvaru regulárního simplexu /není to nutné/ s bodem 1 v počátku souřadnic. Je možné počátek souřadnic umístit do těžiště. Proces vyhledávání vrcholů v E^n, ve kterém má $f(\underline{x})$ nejlepší hodnotu se skládá z následujících operací:

1/ Odraz = projekce $\underline{x}_h^k$ přes těžiště podle výrazu

$$\underline{x}_{n+3}^k = \underline{x}_{n+2}^k + \alpha(\underline{x}_{n+2}^k - \underline{x}_h^k),$$

kde $\alpha > 0$ je koeficient odražení, $\underline{x}_{n+2}^k$ je těžiště vyčíslené podle vzorce /3.2.-1/, $\underline{x}_h^k$ je vrchol, ve kterém má funkce $f(\underline{x})$ nejvyšší z $(n+1)$ svých hodnot v k-té etapě.

2/ Prodloužení. Jestliže $f(\underline{x}_{n+3}^k) \leq f(\underline{x}_l^k)$, pak vektor $(\underline{x}_{n+3}^k - \underline{x}_{n+2}^k)$ se prodlouží na

$$\underline{x}_{n+4}^k = \underline{x}_{n+2}^k + \gamma(\underline{x}_{n+3}^k - \underline{x}_{n+2}^k),$$

kde $\gamma > 1$ představuje koeficient prodloužení. Je-li $f(\underline{x}_{n+4}^k) < f(\underline{x}_l^k)$, pak $\underline{x}_h^k$ se zamění s $\underline{x}_{n+4}^k$ /vznikne nový simplex/ a proces pokračuje znovu od operace 1/ s $k = k+1$. V opačném případě se $\underline{x}_h^k$ zamění s $\underline{x}_{n+3}^k$ a také se provede přechod na operaci 1/ s $k = k+1$.

3/ Zkrácení. Jestliže $f(\underline{x}_{n+3}^k) > f(\underline{x}_i^k)$ pro všechna $i \neq h$,

pak vektor ... se zkrátí na

pak vektor $(\underline{x}_h^k - \underline{x}_{n+2}^k)$ se zkrátí na

$$\underline{x}_{n+5}^k = \underline{x}_{n+2}^k + \beta(\underline{x}_h^k - \underline{x}_{n+2}^k),$$

kde $0<\beta<1$ je koeficient zkrácení. Potom zaměníme $\underline{x}_h^k$ s $\underline{x}_{n+5}^k$ a vrátíme se k operaci 1/ a pokračujeme v hledání v $(k+1)$ -ní etapě.

4/ Zmenšení /redukce/. Jestliže $f(\underline{x}_{n+3}^k) > f(\underline{x}_h^k)$, pak všechny vektory $(\underline{x}_i^k - \underline{x}_l^k)$, $i=1,\dots,(n+1)$, se zmenší podle výrazu:

$$\underline{x}_i^k = \underline{x}_l^k + 0.5(\underline{x}_i^k - \underline{x}_l^k), \quad i=1,\dots,(n+1).$$

Potom se vracíme k operaci 1/ a pokračujeme ve hledání v $(k+1)$ -ní etapě.

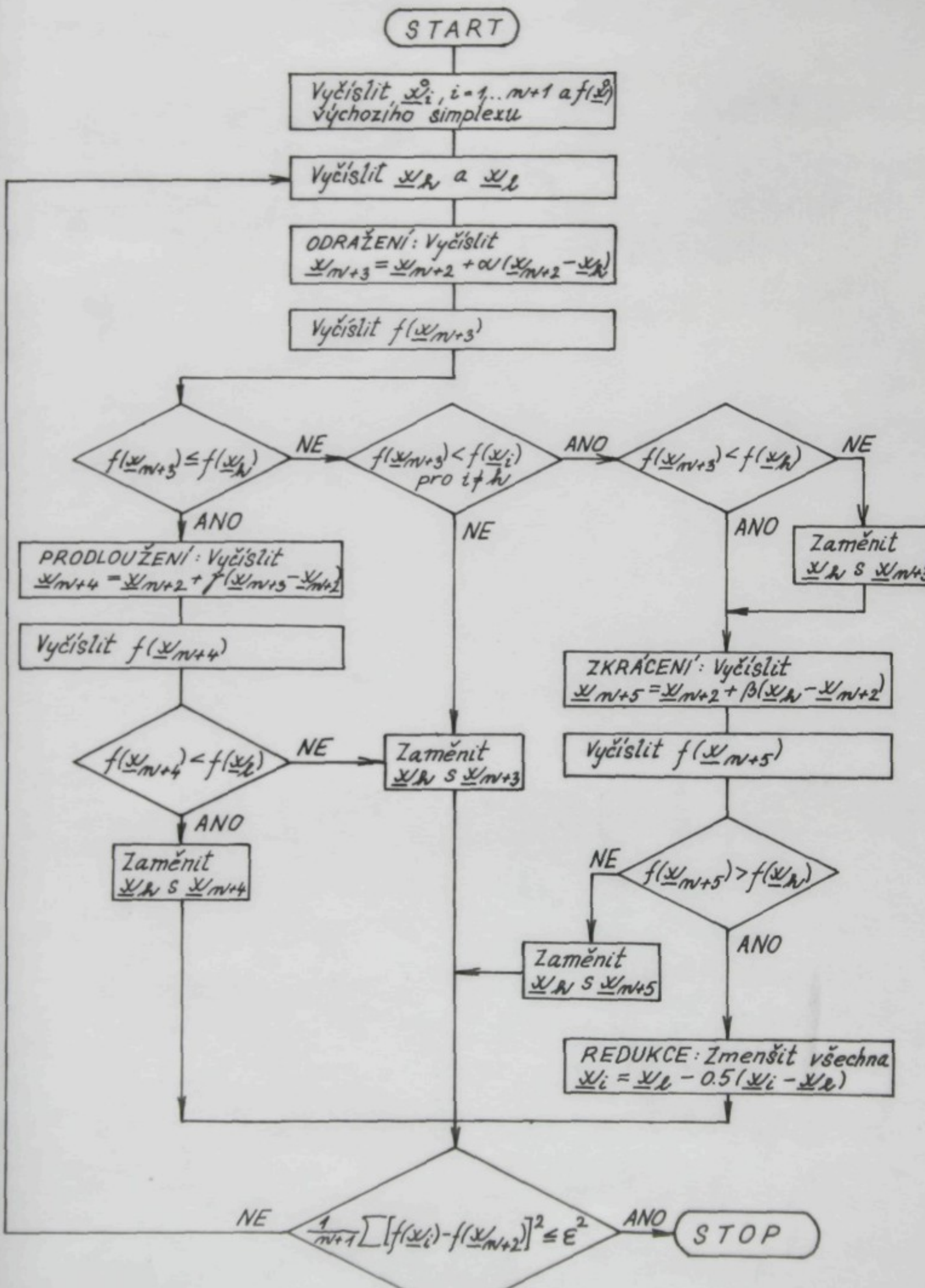
5/ Proces končí, jestliže

$$\left\{1/(n+1)\sum_{i=1}^{n+1}\left[f(\underline{x}_i^k) - f(\underline{x}_{n+2}^k)\right]^2\right\} \leq \varepsilon^2$$

kde ε je libovolné malé číslo a $f(\underline{x}_{n+2}^k)$ je hodnota účelové funkce v těžišti $\underline{x}_{n+2}^k$.

Koeficient odražení α se používá pro projekci vrcholu s nejvyšší hodnotou $f(\underline{x})$ přes těžiště deformovaného polyedru. Koeficient γ se zavádí pro prodloužení vektoru hledání v případě, že odražením vznikl vrchol s hodnotou $f(\underline{x})$ menší než nejmenší hodnota $f(\underline{x})$ získaná do odražení. Koeficient β používáme pro zkrácení vektoru hledání, jestliže odražení nevedlo k vrcholu s hodnotou $f(\underline{x})$ menší než druhá největší hodnota $f(\underline{x})$ získaná do odrazu. J.A. Nelder a R. Mead doporučili volit $\alpha = 1$, $\gamma = 2$, $\beta = 0.5$.

Příklad: Minimalizujte účelovou funkci



START
Vyčíslit $\underline{x}_i^0$, $i = 1 \dots n+1$ a $f(\underline{x}_i^0)$ výchozího simplexu
Vyčíslit $\underline{x}_h$ a $\underline{x}_l$
ODRAŽENÍ: Vyčíslit $\underline{x}_{n+3} = \underline{x}_{n+2} + \alpha(\underline{x}_{n+2} - \underline{x}_h)$
Vyčíslit $f(\underline{x}_{n+3})$
$f(\underline{x}_{n+3}) \leq f(\underline{x}_l)$
NE
$f(\underline{x}_{n+3}) < f(\underline{x}_i)$ pro $i \neq h$
ANO
$f(\underline{x}_{n+3}) < f(\underline{x}_h)$
NE
ANO
NE
ANO
Zaměnit $\underline{x}_h$ s $\underline{x}_{n+3}$
PRODLOUŽENÍ: Vyčíslit $\underline{x}_{n+4} = \underline{x}_{n+2} + \gamma(\underline{x}_{n+3} - \underline{x}_{n+2})$
Vyčíslit $f(\underline{x}_{n+4})$
ZKRÁCENÍ: Vyčíslit $\underline{x}_{n+5} = \underline{x}_{n+2} + \beta(\underline{x}_h - \underline{x}_{n+2})$
Vyčíslit $f(\underline{x}_{n+5})$
$f(\underline{x}_{n+4}) < f(\underline{x}_l)$
NE
Zaměnit $\underline{x}_h$ s $\underline{x}_{n+3}$
ANO
Zaměnit $\underline{x}_h$ s $\underline{x}_{n+4}$
NE
$f(\underline{x}_{n+5}) > f(\underline{x}_h)$
ANO
Zaměnit $\underline{x}_h$ s $\underline{x}_{n+5}$
REDUKCE: Zmenšit všechna $\underline{x}_i = \underline{x}_l - 0.5(\underline{x}_i - \underline{x}_l)$
NE
$\frac{1}{n+1}\sum\left[f(\underline{x}_i) - f(\underline{x}_{n+2})\right]^2 \leq \varepsilon^2$
ANO
STOP

Protože $f(\underline{x})$ je funkce dvou proměnných, použijeme trojúhelník s vrcholy $\underline{x}_1^0=[4\ 2]^T,\ \underline{x}_2^0=[5\ 3]^T,\ \underline{x}_3^0=[4\ 3]^T$. Je možné použít i jinou libovolnou kombinaci tří bodů.

$k=0$ V nulové etapě hledání vyčíslíme hodnotu funkce a dostáváme $f(4,2)=2.0,\ f(5,3)=6.25,\ f(4,3)=5.0$. Potom odrazíme $\underline{x}_2^0\ [5\ 3]^T$ přes těžiště bodů $\underline{x}_1^0$ a $\underline{x}_3^0$ /podle/3.2.-1//, který označíme $\underline{x}_4^0$.

$$x_{4,1}^0 = 0.5[(4+5+4)-5]=4.00$$

$$x_{4,2}^0 = 0.5[(2+3+3)-3]=2.50 \qquad f(4.00,2.50)=3.25$$

s tím, abychom získali $\underline{x}_5^0$.

$$x_{5,1}^0 = 4.0+1(4.0-5.0)=3.00$$

$$x_{5,2}^0 = 2.5+1(2.5-3.0)=2.00 \qquad f(3.00,2.00)=1.25$$

Protože $f(3,2)=1.25<f(4,2)=2.00$, přejdeme k operaci prodloužení.

$$x_{6,1}^0 = 4.0+2(3-4.0)=2.00$$

$$x_{6,2}^0 = 2.5+2(2-2.5)=1.50 \qquad f(2.00,1.50)=0.25$$

Protože $f(2,1.5)=0.25<f(4,2)=2.00$, zaměníme $\underline{x}_2^0$ s $\underline{x}_6^0$ a položíme $\underline{x}_6^0=\underline{x}_2^1$ v následující etapě hledání. Nakonec, protože

$$1/3\,[(2+3.25)^2+(5-3.25)^2+(0.25-3.25)^2]^{1/2}=2.1311>10^{-6},$$

začínáme další etapu hledání. $k=1$

$$x_{4,1}^1 = 0.5[(4+2+4)-4]=3.00$$

$$x_{4,2}^1 = 0.5[(2+1.5+3)-3]=1.75 \qquad f(3.00,1.75)=0.8125$$

$$x^{1}_{5,1} = 3 + 1(3-4) = 2.00$$

$$x^{1}_{5,2} = 1.75 + 1(1.75-3) = 0.50 \qquad f(2.00, 0.50) = 0,25$$

$$x^{1}_{6,1} = 3 + 2(2-3) = 1.00$$

$$x^{1}_{6,2} = 1.75 + 2(0.5-1.75) = -0.75 \qquad f(1.00, -0.75) = 0.3125$$

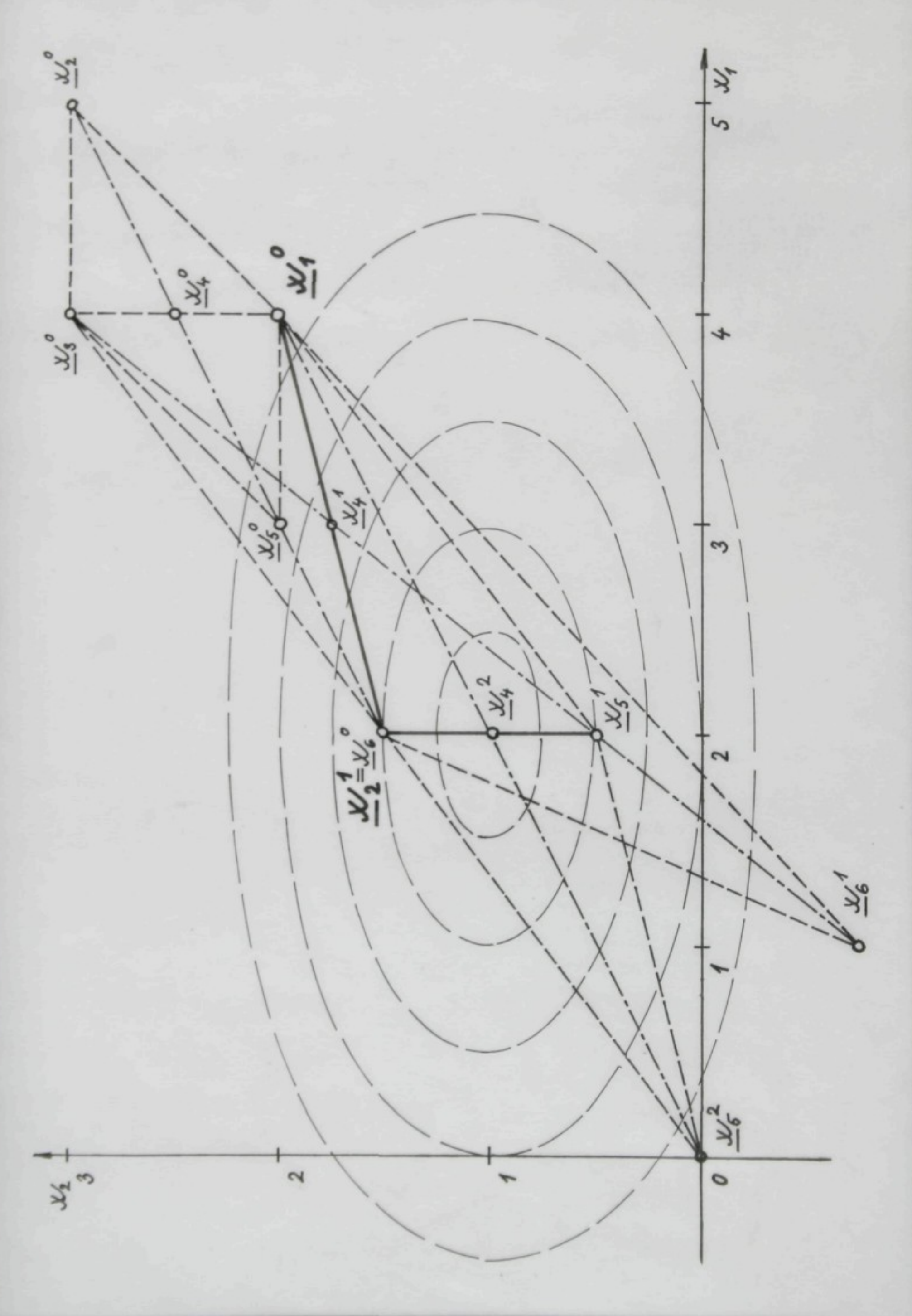
Zaměníme $\underline{x}^{1}_{5}$ s $\underline{x}_{h}$ a začneme další etapu hledání. $k=2$

$$x^{2}_{4,1} = 0.5[(2+4+2)-4] = 2.00$$

$$x^{2}_{4,2} = 0.5[(0.5+2+1.5)-2] = 1.00 \qquad f(2.00, 1.00) = 0$$

$$x^{2}_{5,1} = 2 + 1(2-4) = 0$$

$$x^{2}_{5,2} = 1 + 1(1-2) = 0 \qquad f(0,0) = 2.00$$


x_1
x_2
0
1
2
3
4
5
x_1^0
x_2^0
x_3^0
x_4^0
x_5^0
x_4^1
x_5^1
x_6^1
x_4^2
x_6^2
$x_2^1 = x_6^0$

3.3. Rosenbrockova metoda

Je iterační procedurou, která se shoduje s metodou zkušebního kroku v tom, že se při hledání v ortogonálních směrech podnikají malé kroky. Avšak zde místo nepřetržitého hledání v souřadnicových směrech, které představují směry nezávislých proměnných, se konstruují na základě informací získaných v $(k-1)$ -ní etapě ortogonální vektory $\underline{\hat{s}}_1^k, \dots, \underline{\hat{s}}_n^k$ - jednotkové vektory v E^n, kde index $k = 0, 1, \dots$ označuje etapy hledání. V začínající etapě $k = 0$ směry $\underline{\hat{s}}_1^0, \dots, \underline{\hat{s}}_n^0$ se berou obvykle rovnoběžné s osami $x_1, \dots, x_n$. Ve všeobecném případě mohou být ortogonální směry vyjádřeny jako kombinace souřadnic nezávislých proměnných následujícím způsobem

$$\begin{aligned}
\underline{\hat{s}}_1^k &= a_{11}^k \underline{\delta}_1 + a_{12}^k \underline{\delta}_2 + \dots + a_{1n}^k \underline{\delta}_n , \\
\underline{\hat{s}}_2^k &= a_{21}^k \underline{\delta}_1 + a_{22}^k \underline{\delta}_2 + \dots + a_{2n}^k \underline{\delta}_n , \\
&\dots\dots\dots\dots\dots\dots \\
\underline{\hat{s}}_n^k &= a_{n1}^k \underline{\delta}_1 + a_{n2}^k \underline{\delta}_2 + \dots + a_{nn}^k \underline{\delta}_n
\end{aligned} \qquad /3.3.-1/$$

kde $\underline{\delta}_i$ je jednotkový vektor ve směru x_i a a_{ij} představuje směrové cosiny $\hat{s}_i$. V maticovém označení

$$\underline{\hat{s}}^k = \underline{a}^k \underline{\delta} .$$

Sledujme k -tou etapu. Nechť $\underline{x}_0^k = \underline{x}_n^{k-1}$ představuje bod v E^n, z kterého začíná hledání a $\lambda_1, \dots, \lambda_n$ jsou odpovídající délky kroku spojené se směry $\underline{\hat{s}}_1, \dots, \underline{\hat{s}}_n$.

Hledání začíná z $\underline{x}_0^k$ zavedením přírůstku $\lambda_1^k \underline{\hat{s}}_1^k$ v prvním souřadnicovém směru. Jestliže hodnota $f(\underline{x}_0^k + \lambda_1^k \underline{\hat{s}}_1^k)$ je rovna nebo menší než $f(\underline{x}_0^k)$, krok se počítá jako úspěšný a získaný zkoušený bod vymění $\underline{x}_0^k$, λ_1^k násobíme činitelem $\alpha > 0$ a zavádíme přírůstek ve směru $\hat{\underline{S}}_2^k$. Jestliže je $f(\underline{x}_0^k + \lambda_1^k \hat{\underline{S}}_1^k)$ větší než $f(\underline{x}_0^k)$, pak se krok počítá jako neúspěšný, $\underline{x}_0^k$ se nemění, λ_1^k násobíme činitelem $\beta < 0$ a znovu zkoušíme přírůstek ve směru $\hat{\underline{S}}_2^k$. Rosenbrock navrhl brát v obecném případě hodnoty parametrů $\alpha = 3$ a $\beta = -0.5$.

Pak když je prověřeno všech n směrů $\hat{\underline{S}}_1^k, \dots, \hat{\underline{S}}_n^k$, znovu se vracíme k prvnímu směru $\hat{\underline{S}}_1^k$ a zavádíme přírůstek s délkou kroku rovnou $\alpha\lambda^k$ nebo $\beta\lambda^k$ v závislosti na výsledku předcházejícího přírůstku ve směru $\hat{\underline{S}}_1^k$. Přírůstky ve vybraných směrech hledání se uskutečňují do té doby, dokud v každém směru za úspěchem nenásleduje neúspěch. Potom se k-tá etapa hledání zakončuje. Protože zisk stejných hodnot funkcí představuje úspěch, úspěch konec konců dosáhneme v každém směru. Poslední získaný bod se stává výchozím bodem následující etapy $\underline{x}_0^{k+1} = \underline{x}_n^k$. Normovaný směr $\hat{\underline{S}}_1^{k+1}$ je paralelní s $(\underline{x}_0^{k+1} - \underline{x}_0^k)$ a ostatní směry vybíráme ortogonální jeden ke druhému a k $\hat{\underline{S}}_1^{k+1}$.

Po skončení k-té etapy se v bodě $\underline{x}_0^{k+1} = \underline{x}_n^k$ vypočítávají vektory nových směrů hledání. V podstatě se zde ortogonální směry hledání otáčí ve vztahu k předcházejícím směrům tak, že vedou podél $\cap$ nebo $\cup$, tím se vylučuje vzájemné působení proměnných. Nechť Λ_i^k představuje algebraický součet všech úspěšných kroků ve směru $\hat{\underline{S}}_i^k$ v k-té etapě. Určíme n vektorů $\underline{A}_1, \dots, \underline{A}_n$ následujícím způsobem:

$$\begin{aligned} \underline{A}_1^k &= \Lambda_1^k \underline{S}_1^k + \Lambda_2^k \underline{S}_2^k + \ldots + \Lambda_n^k \underline{S}_n^k, \\ \underline{A}_2^k &= \Lambda_2^k \underline{S}_2^k + \ldots + \Lambda_n^k \underline{S}_n^k, \\ &\ldots \\ \underline{A}_n^k &= \Lambda_n^k \underline{S}_n^k. \end{aligned} \qquad /13.3.-2/$$

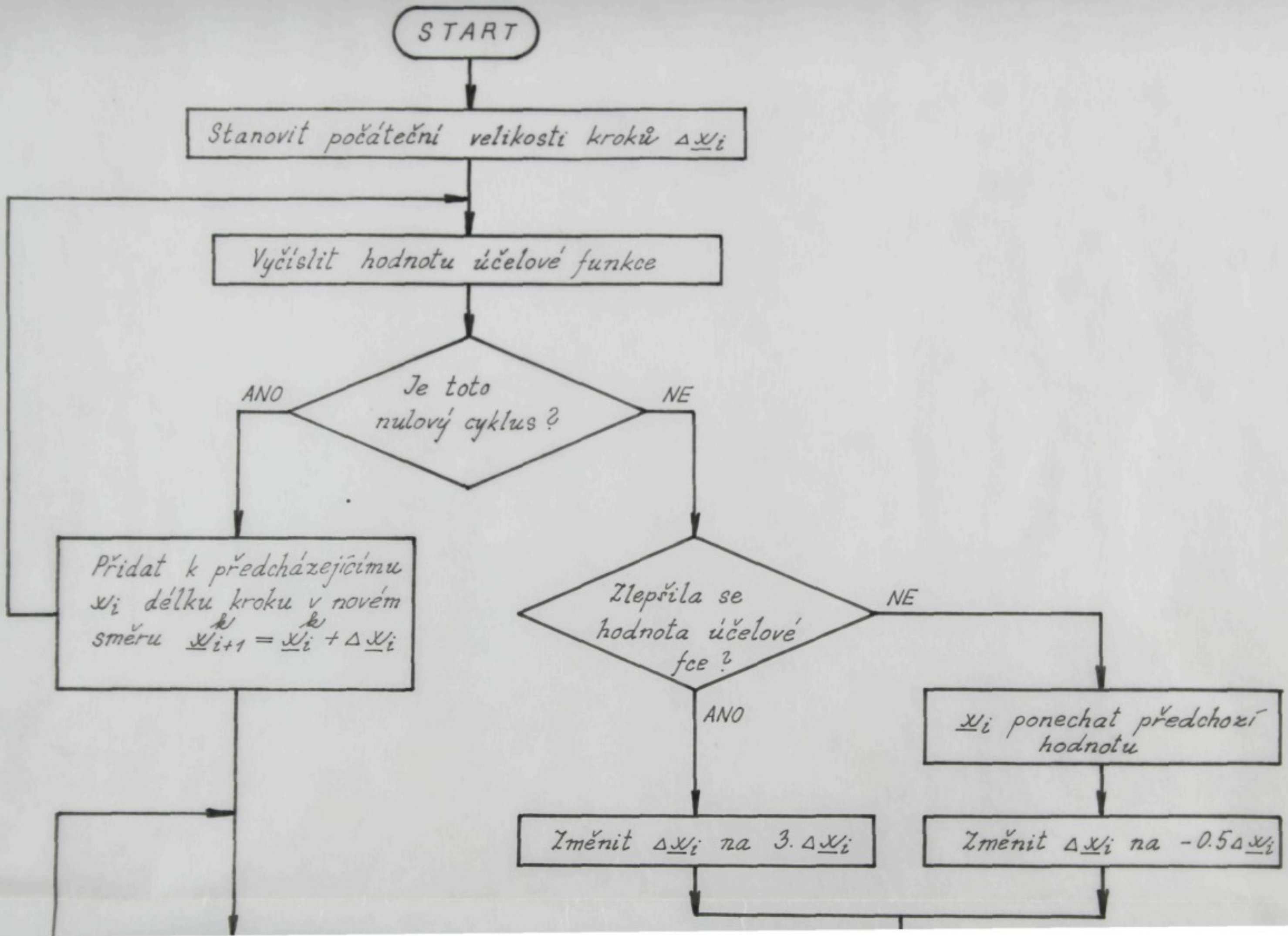
$\underline{A}_1^k$ je vektor přechodu z $\underline{x}_0^k$ v $\underline{x}_0^{k+1}$, $\underline{A}_2^k$ je vektor přechodu z $\underline{x}_1^k$ v $\underline{x}_0^{k+1}$ atd., $\underline{A}_1^k$ představuje úplný přechod z k-té etapy na $(k+1)$-ní etapu atd.

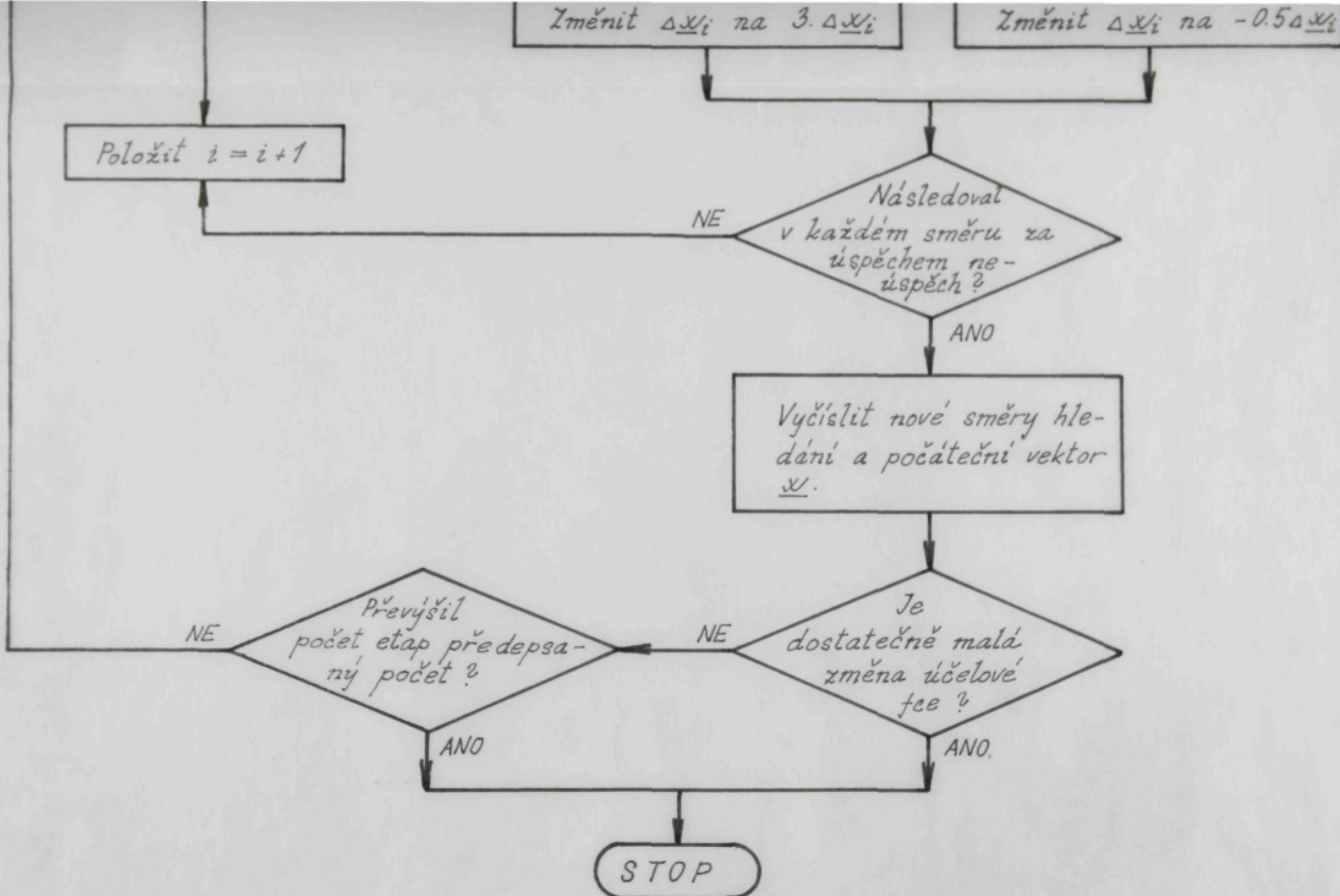
Nové směry se vytváří následujícím způsobem. První jednotkový vektor $\hat{\underline{S}}_1^{k+1}$ z nového souboru směrů v $(k+1)$-ní etapě se vytváří tak, aby byl shodný s výsledným směrem přechodu předcházející etapy $\underline{A}_1^k$. Ostatní směry se konstruují navzájem i k $\underline{A}_1^k$ ortogonální. Podrobně jsou tyto konstrukce popsány v teorii matic a v lineární algebře. Soubor ortogonálních normovaných vektorů $\hat{\underline{S}}_1^{k+1}, \ldots, \hat{\underline{S}}_n^{k+1}$ v $(k+1)$-ní etapě se vyčísluje pomocí následujících vztahů:

$$\begin{aligned} \hat{\underline{S}}_1^{k+1} &= \underline{A}_1^k / \|\underline{A}_1^k\|, \\ \underline{B}_2^k &= \underline{A}_2^k - [(\underline{A}_2^k)^T \hat{\underline{S}}_1^{k+1}] \hat{\underline{S}}_1^{k+1}, \\ \hat{\underline{S}}_2^{k+1} &= \underline{B}_2^k / \|\underline{B}_2^k\|, \\ &\ldots \\ \underline{B}_n^k &= \underline{A}_n^k - \sum_{i=1}^{n-1} [(\underline{A}_n^k)^T \hat{\underline{S}}_i^{k+1}] \hat{\underline{S}}_i^{k+1}, \\ \hat{\underline{S}}^{k+1} &= \underline{B}_n^k / \|\underline{B}_n^k\|, \end{aligned} \qquad /13.3.-3/$$

kde $\|A_i\|$ je norma A_i. V $(k+1)$-ní etapě se pak provádí ta samá procedura hledání, která se prováděla v k-té etapě ; výchozím bodem je bod $\underline{x}_0^{k+1} = \underline{x}_n^k$.

Metoda Rosenbrocka nezabezpečuje automatické zakončení


START
Stanovit počáteční velikosti kroků $\Delta \underline{x}_i$
Vyčíslit hodnotu účelové funkce
Je toto nulový cyklus?
ANO
NE
Přidat k předcházejícímu x_i délku kroku v novém směru $\underline{x}^k_{i+1} = \underline{x}^k_i + \Delta \underline{x}_i$
Zlepšila se hodnota účelové fce?
NE
ANO
$\underline{x}_i$ ponechat předchozí hodnotu
Změnit $\Delta \underline{x}_i$ na $3 . \Delta \underline{x}_i$
Změnit $\Delta \underline{x}_i$ na $-0.5 \Delta \underline{x}_i$



Blokové schema : Rosenbrockova metoda

hledání potom, co je nalezen extrém. Hledání se může provádět buď ve stanoveném počtu etap nebo když veličina $\underline{A_1}$ bude menší než /v několika následujících etapách/ předem určená hodnota.

Příklad: Minimalizujte účelovou funkci

$$f(\underline{x}) = (0.5x_1 - 1)^2 + (x_2 - 1)^2$$

Začneme z bodu $\underline{x}_0^0 = [4.5 \;\; 2.0]^T$, kde $f(\underline{x}_0^0) = 2.5625$, $\lambda_1 = 0.10$. Směry v počátku hledání jsou souhlasné se souřadnicovými osami x_1 a x_2.

$$\underline{\hat{S}}_1^0 = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \qquad \underline{\hat{S}}_2^0 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

Nejdříve vyčíslíme $f(\underline{x})$ v bodě

$$\underline{x}_1^0 = \begin{bmatrix} 4.5 + (0.1)1 \\ 2.0 + (0.1)0 \end{bmatrix} = \begin{bmatrix} 4.6 \\ 2.0 \end{bmatrix}$$

Zde je $f(\underline{x}_1^0) = 2.69$ tj. místo neúspěchu. Potom vyčíslíme $f(\underline{x})$ v bodě

$$\underline{x}_2^0 = \begin{bmatrix} 4.5 + (0.1)0 \\ 2.0 + (0.1)1 \end{bmatrix} = \begin{bmatrix} 4.5 \\ 2.1 \end{bmatrix}$$

Zde je $f(\underline{x}_2^0) = 2.7725$ a znovu je to místo neúspěchu. Proto v následujícím cyklu λ_1 a λ_2 vynásobíme koeficientem $\beta = -0.5$, a tak $\lambda_3 = \lambda_4 = (-0.5)(0.1) = -0.05$. Hledání provádíme z poslední úspěšné hodnoty $\underline{x}^0$, tj. z bodu $\underline{x}_0^0 = [4.5 \;\; 2]^T$.

Nyní vyčíslíme $f(\underline{x})$ v bodě

$$\underline{x}_3^0 = \begin{bmatrix} 4.5 - (0.05)1 \\ 2.0 - (0.05)0 \end{bmatrix} = \begin{bmatrix} 4.45 \\ 2.00 \end{bmatrix}$$

Zde je $f(\underline{x}_3^0) = 2.5006$, což znamená úspěch. Potom vyčíslíme $f(\underline{x})$ v bodě

$$\underline{x}_4^0 = \begin{bmatrix} 4.45 - (0.05)0 \\ 2.00 - (0.05)1 \end{bmatrix} = \begin{bmatrix} 4.45 \\ 1.95 \end{bmatrix}$$

kde je $f(\underline{x}_4^0) = 2.4031$, což je znovu úspěch. V následujícím cyklu λ zvětšíme třikrát tj. $\lambda_5 = \lambda_6 = 3(-0.05) = -0.15$.

krok	λ	x_1	x_2	$f(x)$	výsledek
5	-0.15	4.30	1.95	2.2250	U
6	-0.15	4.30	1.80	1.9625	U
7	-0.45	3.85	1.80	1.4956	U
8	-0.45	3.85	1.35	0.9781	U
9	-1.35	2.50	1.35	0.1850	U
10	-1.35	2.50	0.00	1.0625	N
11	-4.05	-1.55	1.35	3.2731	N

Nyní v každém souřadnicovém směru za úspěchem následoval neúspěch, a tak skončila nulová etapa. Nyní vyčíslíme nové směry hledání tak, aby $\underline{\hat{S}}_1^1$ měl směr vektoru jdoucího z $\underline{x}_0^0 = [4.5\ 2]^T$ do $\underline{x}_0^1 = [2.5\ \ 1.35]^T$, přičemž poslední bod představuje nejlepší hodnotu $f(\underline{x})$, získanou v nulové etapě; $\underline{\hat{S}}_2^1$ je ortogonální k $\underline{\hat{S}}_1^1$. Vektory $\underline{A}_1$ a $\underline{A}_2$ vypočítáme podle vzorce /3.3.-2/ a $\underline{\hat{S}}_1^1$ a $\underline{\hat{S}}_2^1$ podle vzorce /3.3.-3/.

$$\underline{A}_1^0 = 4\begin{bmatrix} -1 \\ 0 \end{bmatrix} + 3\begin{bmatrix} 0 \\ -1 \end{bmatrix} = \begin{bmatrix} -4 \\ 3 \end{bmatrix} \qquad \underline{A}_2^0 = 3\begin{bmatrix} 0 \\ -1 \end{bmatrix} = \begin{bmatrix} 0 \\ -3 \end{bmatrix}$$

$$\underline{\hat{S}}_1^1 = \frac{[-4\ -3]^T}{[(-4)^2 + (-3)^2]^{1/2}} = \frac{[-4\ -3]^T}{5} = \begin{bmatrix} -4/5 \\ -3/5 \end{bmatrix} = \begin{bmatrix} -0.8 \\ -0.6 \end{bmatrix}$$

$$\underline{B}_2^0 = \begin{bmatrix} 0 \\ -3 \end{bmatrix} - [0\ -3] \cdot \begin{bmatrix} -0.8 \\ -0.6 \end{bmatrix} \cdot \begin{bmatrix} -0.8 \\ -0.6 \end{bmatrix} = \begin{bmatrix} 0 \\ -3 \end{bmatrix} - 1.8\begin{bmatrix} -0.8 \\ -0.6 \end{bmatrix} =$$

$$= \begin{bmatrix} 0 \\ -3 \end{bmatrix} - \begin{bmatrix} -1.44 \\ -1.08 \end{bmatrix} = \begin{bmatrix} 1.44 \\ -1.92 \end{bmatrix}$$

$$\hat{\underline{S}}{}_2^1 = \frac{[1.44 \ -1.92]^T}{[(1.44)^2 + (-1.92)^2]^{1/2}} = \frac{[1.44 \ -1.92]^T}{2.4} = \begin{bmatrix} 0.6 \\ -0.8 \end{bmatrix}$$

V prvním cyklu v první etapě hledání $(k = 1)$ bereme $\lambda_1 = \lambda_2 = = -4.55(-0.5) = 2.025$. Nejdříve vyčíslíme $f(\underline{x})$ ve směru hledání $\hat{\underline{S}}{}_1^1$ v bodě

$$\underline{x}_1^1 = \begin{bmatrix} 2.5 + (2.025)\cdot(-0.8) \\ 1.35 + (2.025)\cdot(-0.6) \end{bmatrix} = \begin{bmatrix} 0.88 \\ 0.135 \end{bmatrix}$$

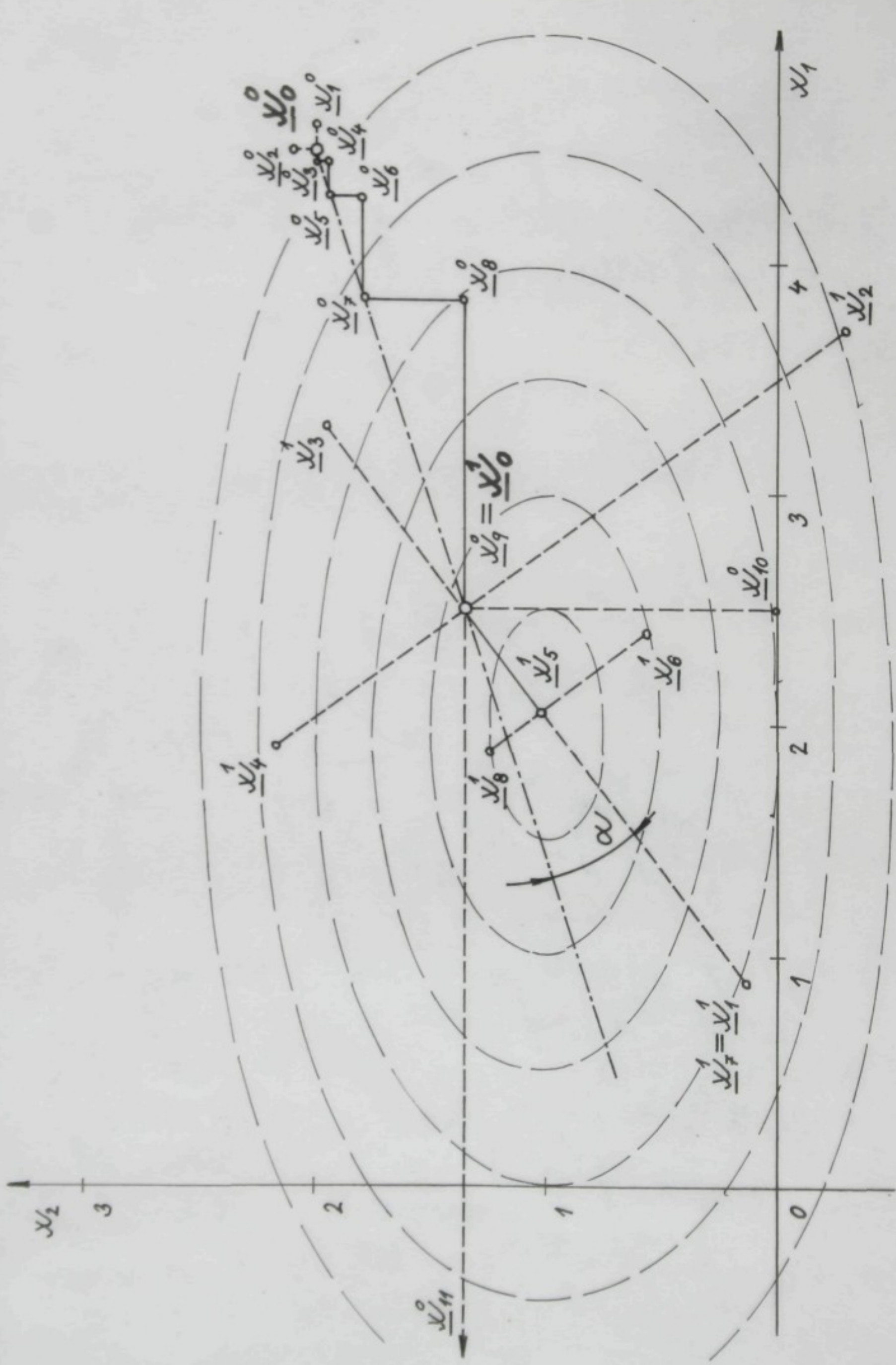
Zde je $f(x) = 1.061825$, což představuje neúspěch.

$$\underline{x}_2^1 = \begin{bmatrix} 2.5 + (2.025)\cdot(0.6) \\ 1.35 + (2.025)\cdot(-0.8) \end{bmatrix} = \begin{bmatrix} 3.715 \\ -0.27 \end{bmatrix}$$

Zde je $f(\underline{x}) = 2.348206$, což je opět neúspěch. Protože 1. a 2. krok hledání nevedly k úspěchu bereme $\lambda_3 = \lambda_4 = (-0.5)(2.025)$

krok	λ	x_1	x_2	$f(x)$	výsledek
3	-1.0125	3.3100	1.9575	1.345831	N
4	-1.0125	1.8925	2.1600	1.348489	N
5	+0.506	2.0952	1.0464	0.004187	U
6	+0.506	2.3988	0.6416	0.168210	N
7	-0.253	0.8808	0.1356	1.060339	N
8	-0.253	1.9434	1.2488	0.062702	N

Vyčíslený nový směr hledání $\hat{\underline{S}}{}_1^1$ by měl mít směr vektoru jdoucího z $\underline{x}_0^0$ do $\underline{x}_0^1$. Toto ovšem platí pouze v případě, kdy je počet úspěšných kroků ve směru $\hat{\underline{S}}{}_1^0$ shodný s počtem úspěšných kroků ve směru $\hat{\underline{S}}{}_2^0$. Není-li tomu tak, odchyluje se vypočítaný nový směr hledání $\hat{\underline{S}}{}_1^1$ od směru vektoru $[\underline{x}_0^1 - \underline{x}_0^0]$ o úhel α vyznačený na následujícím obrázku.

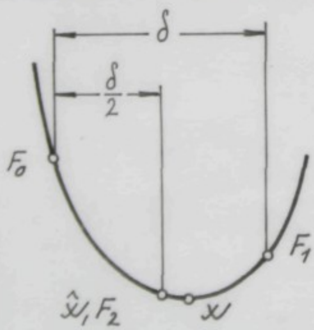

x_1
x_2
0
1
2
3
4
α
$\underline{x}_0^0$
$\underline{x}_9^0 = \underline{x}_0^1$
$\underline{x}_7^1 = \underline{x}_1^1$
$\underline{x}_{10}^0$
$\underline{x}_{11}^0$

3.4. Rosenbrockova metoda rozšířená Swannem

Stejně jako v Rosenbrockově metodě začíná se určením výchozího bodu $\underline{x}_0$ a sadou ortogonálních jednotkových vektorů $\underline{\hat{s}}_i$, $i = 1, \dots n$. Odlišně od způsobu Rosenbrockova se provedou po sobě ve směrech $\underline{\hat{s}}_i$ lineární vyhledávání v těchto směrech pro určení minim $F_{min}(\underline{\hat{s}}_i)$. Výchozím bodem pro stávající vyhledávání je získané minimum z předchozího vyhledávání.

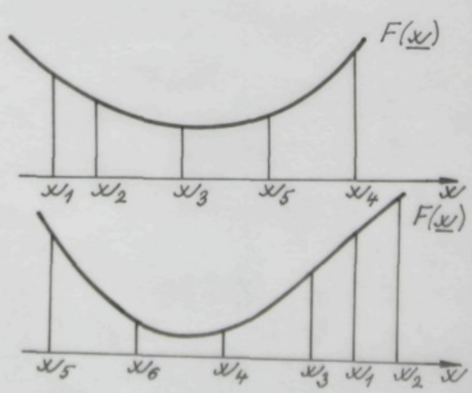
- Jednorozměrné lineární vyhledávání

Pro lineární vyhledávání podél jednoho vyhledávacího směru se použije algoritmus, který odpovídá blokovému schematu /3.4.-1/. Jak je vidět z tohoto schematu stojí na konci každého vyhledávání odhad minima jako vrcholový bod paraboly třemi danými body paraboly s funkčními hodnotami F_0, F_1, F_2 podle obr. /3.4.-1/. Protože se jedná pouze o přibližný výpočet, bude $\underline{x}$ uznáno jako minimum tehdy, když $f(\underline{x}) < F_2$. Jinak zachováme jako minimum $\underline{\hat{x}}$. Na obr. /3.4.-2/ je znázorněna strategie vyhledávání minima.

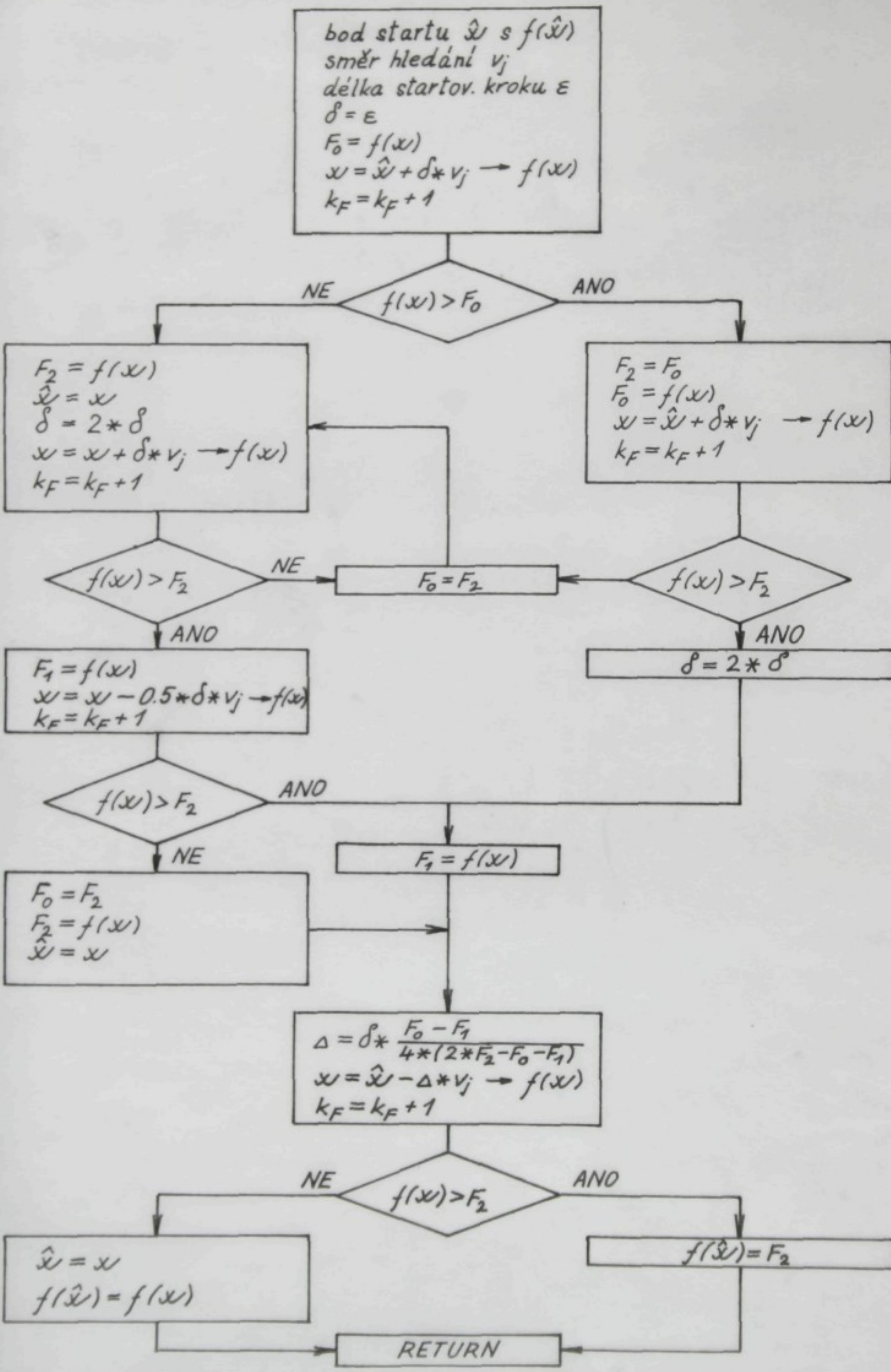


$$\underline{x} = \underline{\hat{x}} - \delta \underline{s}_j \left[\frac{F_0 - F_1}{4(2F_2 - F_0 - F_1)} \right]$$

obr. /3.4.-1/



obr. /3.4.-2/


bod startu $\hat{x}$ s $f(\hat{x})$
směr hledání v_j
délka startov. kroku ε
$\delta = \varepsilon$
$F_0 = f(x)$
$x = \hat{x} + \delta * v_j \rightarrow f(x)$
$k_F = k_F + 1$
NE
$f(x) > F_0$
ANO
$F_2 = f(x)$
$\hat{x} = x$
$\delta = 2 * \delta$
$x = x + \delta * v_j \rightarrow f(x)$
$k_F = k_F + 1$
$F_2 = F_0$
$F_0 = f(x)$
$x = \hat{x} + \delta * v_j \rightarrow f(x)$
$k_F = k_F + 1$
$f(x) > F_2$
NE
$F_0 = F_2$
$f(x) > F_2$
ANO
ANO
$F_1 = f(x)$
$x = x - 0.5 * \delta * v_j \rightarrow f(x)$
$k_F = k_F + 1$
$\delta = 2 * \delta$
$f(x) > F_2$
ANO
NE
$F_1 = f(x)$
$F_0 = F_2$
$F_2 = f(x)$
$\hat{x} = x$
$\Delta = \delta * \frac{F_0 - F_1}{4 * (2 * F_2 - F_0 - F_1)}$
$x = \hat{x} - \Delta * v_j \rightarrow f(x)$
$k_F = k_F + 1$
NE
$f(x) > F_2$
ANO
$\hat{x} = x$
$f(\hat{x}) = f(x)$
$f(\hat{x}) = F_2$
RETURN

Po skončení jednoho vyhledávacího cyklu se určí stejným způsobem jako v Rosenbrockově metodě nový systém ortogonálních vyhledávacích směrů, přičemž první nový vyhledávací směr souhlasí se směrem celkového výsledku z předcházejících směrů.

-Algoritmus ortogonalizace

Budiž Λ_i^k algebraický součet úspěšnýchkroků ve směru $\underline{s}_i^k$. Ortogonalizační proces lze provádět pro všechny směry pouze tehdy, je-li $\Lambda_i^k \neq 0$ pro všechny směry $\underline{s}_i^k$. Při modifikované metodě může na základě popsaného vyhledávání minima přijmout nulovou hodnotu každé Λ_i^k. Aby přesto byl zaručen korektní výpočet nových ortogonálních směrů, přeřadí se staré vyhledávací směry tak, že směry $\underline{s}_i^k$ s $\Lambda_i^k = 0$ se uspořádají vždy nakonec, tedy po $\underline{s}_n^k$. Ortogonalizace se provádí potom ještě jen pro zbývající směry s_i^k s $\Lambda_i^k \neq 0$. Směry s_i^k s $\Lambda_i^k = 0$ jsou převzaty nezměněny, čímž není dotčena ortogonalita nového systému směrů vyhledávání. V praxi se ukázalo výhodnější používat jako kritérium pro přeměnu směru ne $\Lambda_i^k = 0$, nýbrž $|\Lambda_i^k| < \varepsilon$, přičemž ε je přesnost, pomocí které má být určeno minimum.

- Velikost kroku

Velikost startovního kroku se stanoví předem. Při přiblížení k minimu se velikost kroku automaticky redukuje. Redukce následuje potom, když vyhledávání ve směru celkového výsledku iteračního cyklu znamená krok, který je menší než aktuální velikost kroku. Při redukci velikosti kroku se nevypočítávají nové směry vyhledávání, což znamená, že se provede nový iterační cyklus s redukovanou velikostí kroku, ale ve starých vyhledávacích směrech. Jako velikost startovního kroku se vol volí 0.1 .

- Kritérium přerušení

Je-li velikost kroku během optimalizace zredukována na hodnotu, která je menší než požadovaná přesnost proměnných v minimu, pak následuje nový start vyhledávání s parametry, jež byly dosud zjištěny jako optimální /dodatečná iterace/. Kritérium je pak splněno, když

a/ rozdíl funkčních hodnot dvou po sobě jdoucích průběhů je menší než předem stanovený odhad přesnosti

b/ nebo celkový počet výpočtů funkčních hodnot překračuje předem stanovenou hranici.

Dodatečné iterace se provádějí, aby se předešlo předčasnému přerušení vyhledávání minima, neboť mohou při velmi malých změnách funkčních hodnot nastat zvýšené změny hodnot parametrů.

Příklad /3.4.-1/: Minimalizujte účelovou funkci

$$f(\underline{x}) = (0.5x_1 - 1)^2 + (x_2 - 1)^2$$

Začneme z bodu $\underline{x}_0^0 = [5 \quad 3]^T$, kde $f(\underline{x}_0^0) = 6.25$ a volíme $\delta = 0.3$. Směry na počátku hledání jsou souhlasné se souřadnicovými osami x_1 a x_2.

$$\hat{\underline{s}}_1^0 = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \qquad \hat{\underline{s}}_2^0 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

Provedeme jednorozměrné lineární vyhledávání ve směru $\hat{\underline{s}}_1^0$.

krok	δ	x_1^0	x_2^0	$f(\underline{x})$	výsledek
1	+0.3	5.3	3	6.7225	N
2	-0.3	4.7	3	5.8225	U
3	-0.6	4.1	3	5.1025	U
4	-1.2	2.9	3	4.2025	U
5	-2.4	0.5	3	4.5625	N

$$\overset{o}{\underline{x}}_6 = \begin{bmatrix} 0.5 - 0.5(-2.4)1 \\ 3.0 - 0.5(-2.4)0 \end{bmatrix} = \begin{bmatrix} 1.7 \\ 3 \end{bmatrix} \qquad f(1.7, 3) = 4.0225$$

Protože $f(1.7,3)=4.0225 < f(2.9,3)=4.2025$, bude $F_1=4.56$, $F_0=4.2025$ a $F_2=4.0225$.

$$\Delta = -2.4 \left[\frac{4.2025 \quad 4.5625}{4(2(4.0225) \quad 4.2025 \quad 4.5625)} \right] = \frac{0.864}{-2.88} = -0.3$$

$$\overset{o}{\underline{x}}_7 = \begin{bmatrix} 1.7 - (-0.3)1 \\ 3 - (-0.3)0 \end{bmatrix} = \begin{bmatrix} 2 \\ 3 \end{bmatrix} \qquad f(2, 3) = 4.00$$

Protože $f(2,3)=4.00 < F_2=4.0225=f(1.7,3)$, je jako minimum ve směru $\hat{S}_1^o$ uznán bod $\underline{x}=[2 \ 3]^T$ s $f(\underline{x})=4.00$.

Nyní provedeme jednorozměrné lineární vyhledávání ve směru $\hat{S}_2^o$

krok	δ	x_1	x_2	$f(\underline{x})$	*výsledek*
1	+0.1	2	3.1	4.41	N
2	-0.1	2	2.9	3.61	U
3	-0.2	2	2.7	2.89	U
4	-0.4	2	2.3	1.69	U
5	-0.8	2	1.5	0.25	U
6	-1.6	2	-0.1	1.21	N

$$\overset{o}{\underline{x}}_7 = \begin{bmatrix} 2 - 0.5(-1.6)0 \\ -0.1 - 0.5(-1.6)1 \end{bmatrix} = \begin{bmatrix} 2 \\ 0.7 \end{bmatrix} \qquad f(2, 0.7) = 0.09$$

Protože $f(2,0.7)=0.09 < f(2,1.5)=0.25$, bude $F_1=1.21=$ $=f(2,-0.1)$, $F_0=0.25=f(2,1.5)$, $F_2=0.09=f(2,0.7)$.

$$\Delta = -1.6 \left[\frac{0.25 - 1.21}{4(2(0.09) - 0.25 - 1.21)} \right] = \frac{1.536}{-5.12} = -0.3$$

$$\overset{o}{\underline{x}}_8 = \begin{bmatrix} 2 - (-0.3)0 \\ 0.7 - (-0.3)1 \end{bmatrix} = \begin{bmatrix} 2 \\ 1 \end{bmatrix} \qquad f(2, 1) = 0$$

Nyní vyčíslíme nové směry hledání

$$\underline{A}_1^0 = -3\begin{bmatrix}1\\0\end{bmatrix} - 4\begin{bmatrix}0\\1\end{bmatrix} = \begin{bmatrix}-3\\-4\end{bmatrix} \qquad \underline{A}_2^0 = -4\begin{bmatrix}0\\1\end{bmatrix} = \begin{bmatrix}0\\-4\end{bmatrix}$$

$$\underline{\hat{S}}_1^1 = \frac{[-3\ \ -4]^T}{[(-3)^2+(-4)^2]^{1/2}} = \frac{[-3\ \ -4]^T}{5} = \begin{bmatrix}-0.6\\-0.8\end{bmatrix}$$

$$B_2^0 = \begin{bmatrix}0\\4\end{bmatrix} - [0\ -4]\begin{bmatrix}-0.6\\-0.8\end{bmatrix}\begin{bmatrix}-0.6\\-0.8\end{bmatrix} = \begin{bmatrix}0\\-4\end{bmatrix} - 3.2\begin{bmatrix}-0.6\\-0.8\end{bmatrix} = \begin{bmatrix}1.92\\-1.44\end{bmatrix}$$

$$\underline{\hat{S}}_2^1 = \frac{[1.92\ -1.44]^T}{[(1.92)^2+(-1.44)^2]^{1/2}} = \frac{[1.92\ -1.44]^T}{2.4} = \begin{bmatrix}0.8\\-0.6\end{bmatrix}$$

Hledání ve směru $\underline{\hat{S}}_1^1$:

$$\underline{x}_1^1 = \begin{bmatrix}2+(0.1)(-0.6)\\1+(0.1)(-0.8)\end{bmatrix} = \begin{bmatrix}1.94\\0.92\end{bmatrix} \qquad f(\underline{x}_1^1) = 0.0073 \qquad N$$

$$\underline{x}_2^1 = \begin{bmatrix}2-(0.1)(-0.6)\\1-(0.1)(-0.8)\end{bmatrix} = \begin{bmatrix}2.06\\1.08\end{bmatrix} \qquad f(\underline{x}_2^1) = 0.0073 \qquad N$$

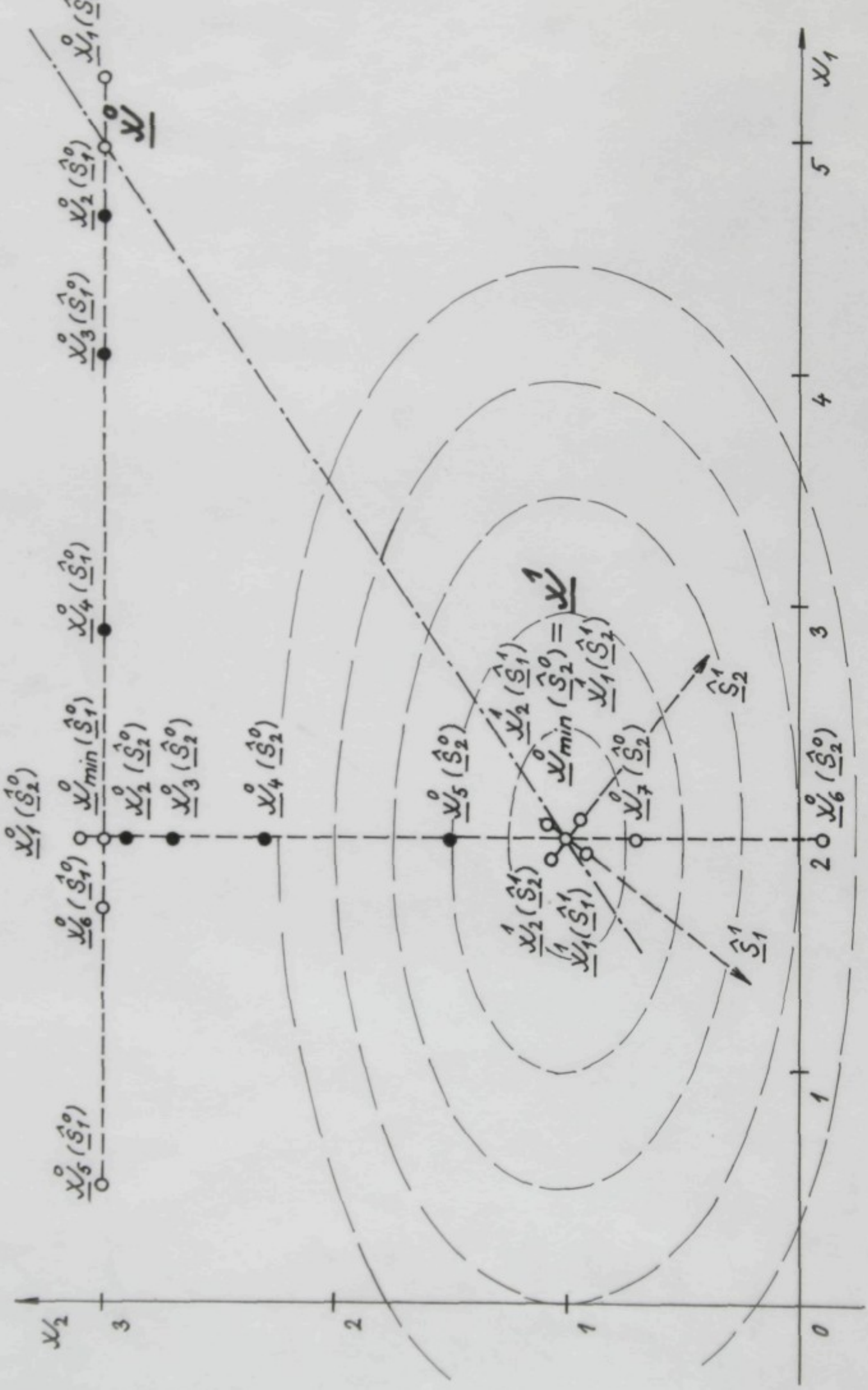
Hledání ve směru $\underline{\hat{S}}_2^1$:

$$\underline{x}_1^1 = \begin{bmatrix}2+(0.1)(0.8)\\1+(0.1)(-0.6)\end{bmatrix} = \begin{bmatrix}2.08\\0.94\end{bmatrix} \qquad f(\underline{x}_1^1) = 0.0052 \qquad N$$

$$\underline{x}_2^1 = \begin{bmatrix}2-(0.1)(0.8)\\1-(0.1)(-0.6)\end{bmatrix} = \begin{bmatrix}1.92\\1.06\end{bmatrix} \qquad f(\underline{x}_2^1) = 0.0052 \qquad N$$

Poznámka: U = úspěch

N = neúspěch


x_1
x_2
0
1
2
3
4
5
$\underline{x}^0$
$\underline{x}^1$
$\hat{\underline{s}}_1^1$
$\hat{\underline{s}}_2^1$

Příklad /3.4.-2/: Minimalitujte účelovou funkci

$$f(\underline{x}) = (0.5\,x_1 - 1)^2 + (x_2 - 1)^2$$

Začneme z bodu $\underline{x}_0^0 = [8 \;\; 9]^T$, kde je $f(\underline{x}_0^0) = 73$ a volíme $\delta_1^0 = 0.1$.

Směry na počátku hledání $\hat{\underline{S}}_1^0 = [1 \;\; 0]^T$, $\hat{\underline{S}}_2^0 = [0 \;\; 1]^T$.

Provedeme vyhledávání ve směru $\hat{\underline{S}}_1^0$:

krok	δ	x_1	x_2	$f(\underline{x})$	výsledek
1	+0.1	8.1	9	73.302	N
2	-0.1	7.9	9	72.702	U
3	-0.2	7.7	9	72.122	U
4	-0.4	7.3	9	71.022	U
5	-0.8	6.5	9	69.062	U
6	-1.6	4.9	9	66.102	U = F0
7	-3.2	1.7	9	64.022	U = F2
8	-6.4	-4.7	9	75.222	N
9	-6.4	-1.5	9	67.062	= F1
minimum		2.0	9	64.000	

Provedeme vyhledávání ve směru $\hat{\underline{S}}_2^0$:

krok	δ	x_1	x_2	$f(\underline{x})$	výsledek
1	+0.1	2	9.1	65.61	N
2	-0.1	2	8.9	62.41	U
3	-0.2	2	8.7	59.29	U
4	-0.4	2	8.3	53.29	U
5	-0.8	2	7.5	42.25	U
6	-1.6	2	5.9	24.01	U
7	-3.2	2	2.7	2.89	U = F0
8	-6.4	2	-3.7	22.09	N = F1
9	-6.4	2	-0.5	2.25	= F2
minimum		2	1.00	0.00	

Vyčíslíme nové směry hledání: $\hat{\underline{S}}_1^1 = [-0.65079 \;\; -0.75926]^T$

$\hat{\underline{S}}_2^1 = [+0.75926 \;\; -0.65079]^T$

Provedeme vyhledávání ve směru $\hat{\underline{S}}_1^1$:

krok	δ	x_1	x_2	$f(\underline{x})$	výsledek
1	+0.1	1.935	0.924	0.0068322	N
2	-0.1	2.065	1.076	0.0068322	N

Provedeme vyhledávání ve směru $\underline{\hat{S}}_2^1$:

krok	δ	x_1	x_2	$f(x)$	výsledek
1	+0.1	2.076	0.935	0.005669	N
2	-0.1	1.924	1.065	0.005669	N

Nalezené minimum je tedy v bodě $\underline{x}_{MIN} = [2\ 1]$ s $f(\underline{x}_{MIN}) = 0$.

Řešení na počítači:

```
            X(1)        X(2)        F(X)        DELTA       KF
            8.10        9.00       73.302       0.100        1
            7.90        9.00       72.702      -0.100        2
            7.70        9.00       72.122      -0.200        3
            7.30        9.00       71.022      -0.400        4
            6.50        9.00       69.062      -0.800        5
            4.90        9.00       66.102      -1.600        6
            1.70        9.00       64.022      -3.200        7
           -4.70        9.00       75.222      -6.400        8
           -1.50        9.00       67.062      -6.400        9
            2.00        9.00       64.000      -6.400       10
         MINX(1)     MINX(2)     FMINX
          2.00        9.00       64.000
            X(1)        X(2)        F(X)        DELTA       KF
            2.00        9.10       65.610       0.100        1
            2.00        8.90       62.410      -0.100        2
            2.00        8.70       59.290      -0.200        3
            2.00        8.30       53.290      -0.400        4
            2.00        7.50       42.250      -0.800        5
            2.00        5.90       24.010      -1.600        6
            2.00        2.70        2.890      -3.200        7
            2.00       -3.70       22.090      -6.400        8
            2.00       -0.50        2.250      -6.400        9
            2.00        1.00        0.000      -6.400       10
         MINX(1)     MINX(2)     FMINX
          2.00        1.00        0.000
S1=   -0.651  -0.759  S2=    0.759  -0.651
            X(1)        X(2)        F(X)        DELTA       KF
            1.93        0.92        0.007       0.100        1
            2.07        1.08        0.007      -0.100        2
            2.00        1.00        0.000      -0.200        3
         MINX(1)     MINX(2)     FMINX
          2.00        1.00        0.000
            X(1)        X(2)        F(X)        DELTA       KF
            2.08        0.93        0.006       0.100        1
            1.92        1.07        0.006      -0.100        2
            2.00        1.00        0.000      -0.200        3
         MINX(1)     MINX(2)     FMINX
          2.00        1.00        0.000
```

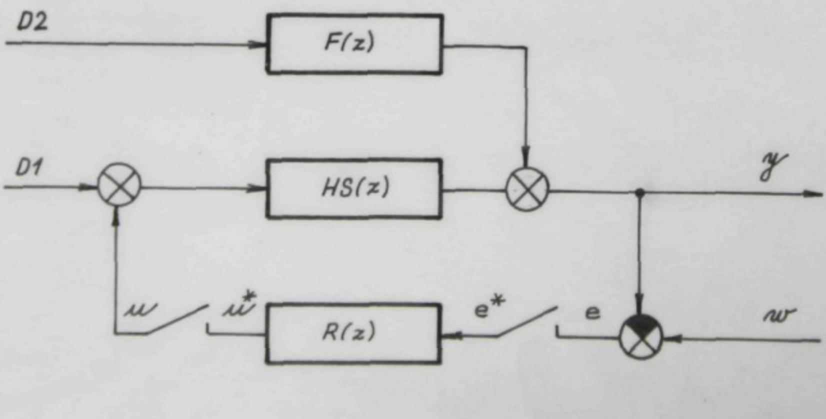
4. Optimální seřízení diskrétních regulátorů podle minima kvadratické regulační plochy

V této kapitole je popsán praktický způsob určení parametrů diskrétního regulátoru podle minima kvadratické regulační plochy, přičemž je předem dána struktura regulátoru a přenosy jednotlivých členů regulačního obvodu. Celá úloha v podstatě obsahuje

- určení výsledného přenosu uzavřeného regulačního obvodu
- výpočet kvadratického kritéria a kontrolu stability
- ošetření nestability
- optimalizaci

4.1. Popis regulačního obvodu

Předpokládejme regulační obvod podle obr. /4.1.-1/.



obr. /4.1.-1/

$HS(z)$ = zadaný obrazový přenos regulované soustavy a tvarovače

$HS(z,1) = z \cdot HS(z)$

$R(z)$ = obrazový přenos diskrétního regulátoru dané struktury

$F(z)$ = zadaný obrazový přenos poruchy

y = regulovaná veličina /skutečná hodnota/

u = akční veličina

w = žádaná hodnota regulované veličiny

e = regulační odchylka

$D1, D2$ = poruchové veličiny

e^*, u^* = vzorkované veličiny

Při působení regulátoru vznikne z rozdílu mezi řídící veličinou w a regulovanou veličinou y regulační odchylka e, která je vstupní hodnotou regulátoru. Regulátor ji zpracuje podle svých dynamických a statických vlastností na akční veličinu u, působící na vstupu soustavy tak, aby byla odstraněna nebo alespoň zmenšena regulační odchylka e.

Výsledné obrazové přenosy k regulační odchylce e od jednotlivých poruch w, $D1$ a $D2$ jsou dány vztahy:

1/ Porucha je ve tvaru jednotkového skoku

$$E_w(z) = \frac{E(z)}{W(z)} = \frac{1}{1+R(z)HS(z,1)} \cdot \frac{z}{z-1} \qquad /4.1.-1/$$

$$E_{D1}(z) = \frac{E(z)}{D1(z)} = \frac{-HS(z)}{1+R(z)HS(z,1)} \cdot \frac{z}{z-1} \qquad /4.1.-2/$$

$$E_{D2}(z) = \frac{E(z)}{D2(z)} = \frac{-F(z)}{1+R(z)HS(z,1)} \cdot \frac{z}{z-1} \qquad /4.1.-3/$$

2/ Porucha je ve tvaru jednotkového impulzu

$$E_w(z) = \frac{E(z)}{W(z)} = \frac{1}{1 + R(z)HS(z,1)} \cdot 1 \qquad /4.1.-4/$$

$$E_{D1}(z) = \frac{E(z)}{D1(z)} = \frac{-HS(z)}{1 + R(z)HS(z,1)} \cdot 1 \qquad /4.1.-5/$$

$$E_{D2}(z) = \frac{E(z)}{D2(z)} = \frac{-F(z)}{1 + R(z)HS(z,1)} \cdot 1 \qquad /4.1.-6/$$

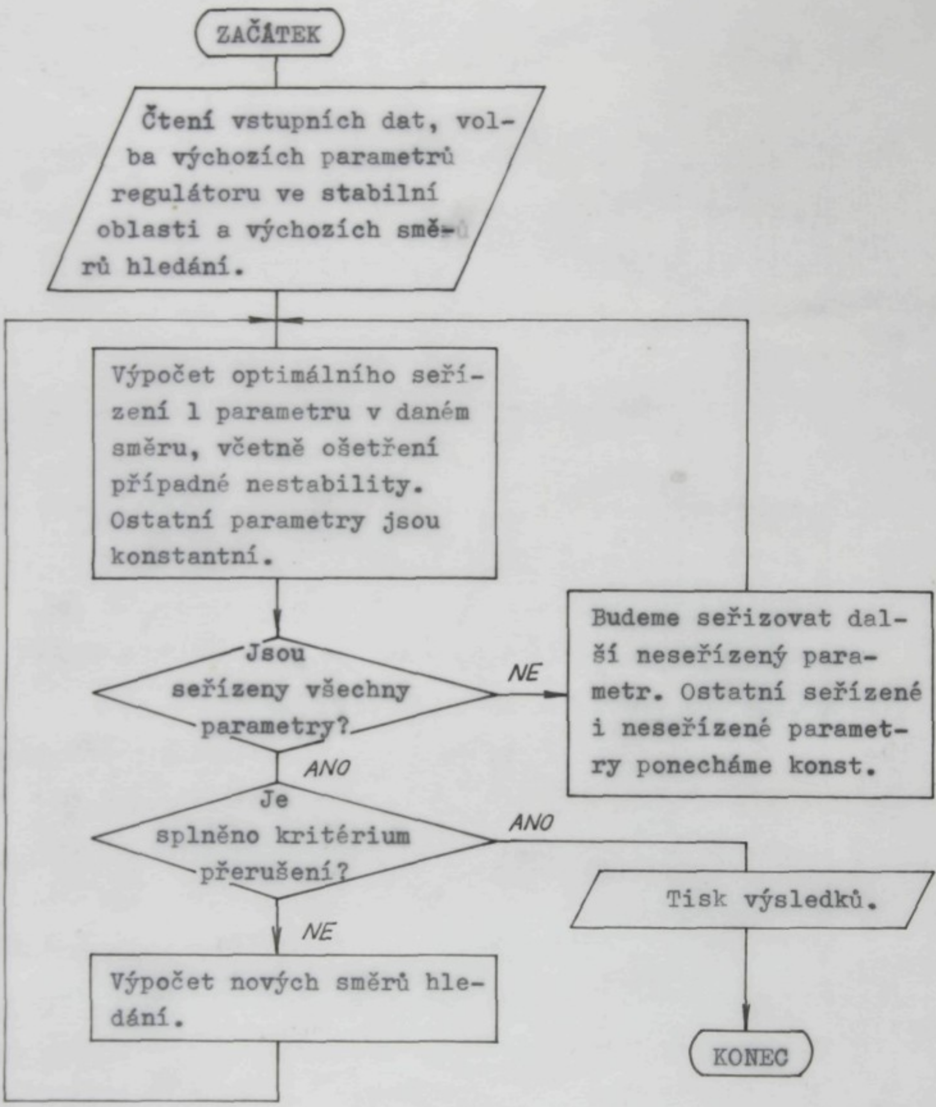
4.2. Hrubý vývojový diagram s popisem

Při řešení úlohy: Nalezení optimálního seřízení diskrétního regulátoru na samočinném počítači; budeme postupovat podle hrubého vývojového diagramu /4.2.-1/. V podstatě se jedná o aplikaci jedné z metod přímého hledání extrému /Rosenbrockova metoda rozšířená Swannem - viz kapitola /3.4//. na řešení dané úlohy.

Popis hrubého vývojového diagramu /4.2.-1/:

Při výpočtu optimálního seřízení 1 parametru v daném směru /přičemž ostatní parametry jsou konstantní/ je použito strategie jednodimensionálního vyhledávání extrému funkce /viz blokové schema /3.4.-1//. S tím rozdílem, že nehledáme extrém účelové funkce $F(\hat{x})$, nýbrž extrémní /minimální/ hodnotu kritéria, to znamená, že "účelovou funkcí" je hodnota integrálu /2.2.-25/. Postup výpočtu:

1/ Ze zadaných přenosů jednotlivých členů regulačního obvodu, výchozího seřízení regulátoru a typu poruchy určíme pomocí



Hrubý vývojový diagram /4.2.-1/

podprogramu PRENOS výsledný přenos uzavřeného regulačního obvodu.

2/ Podprogramy KRATIT a UPRAVP upraví polynomy tohoto přenosu na tvar vstupních parametrů podprogramu KRITER.

3/ Podprogram KRITER pak vypočítá hodnotu integrálu /2.2.-25/ = "hodnota účelové funkce" a zkontroluje stabilitu pro zvolené výchozí seřízení.

4/ V duchu zvolené optimalizační metody změníme 1 parametr o předem stanovenou velikost startovního kroku /ostatní parametry necháme konstantní/ a opakujeme cyklus

PRENOS → KRATIT → UPRAVP → KRITER

tak dlouho, dokud není nalezeno minimum v daném směru. Minimum ve směru pak nalezneme pro všechny seřizované parametry regulátoru.

Kritérium přerušení je splněno, když rozdíl hodnot integrálu dvou po sobě jdoucích cyklů je menší než předem stanovený odhad přesnosti.

Výpočet nových směrů hledání je proveden pomocí formulí /3.3.-2/ a /3.3.-3/.

4.3. Výpočet kritéria jakosti regulace a kontrola stability

Výpočet integrálu /2.2.-25/ /kritérium jakosti regulace/ včetně kontroly stability obsahuje podprogram KRITER. Abychom získali integrál 2.2.-25/, vyčíslíme nejdříve koeficienty polynomů $A_k(z)$ a $B_k(z)$. To můžeme provést pomocí následujících tabulek:

$$
\begin{array}{llll}
\boldsymbol{a_0} & a_1 \ldots\ldots a_{n-1} & a_n \\
a_n & a_{n-1} \ldots a_1 & a_0 \\
\boldsymbol{a_0^{n-1}} & a_1^{n-1} \ldots\ldots a_{n-1}^{n-1} & \\
a_{n-1}^{n-1} & a_{n-2}^{n-1} \ldots a_0^{n-1} & \\
\ldots\ldots\ldots & & \\
\ldots\ldots\ldots & & \\
\boldsymbol{a_0^{1}} & a_1^{1} & \\
a_1^{1} & a_0^{1} & \\
\boldsymbol{a_0^{0}} & &
\end{array}
\qquad\qquad
\begin{array}{llll}
b_0 & b_1 \ldots\ldots b_{n-1} & b_n \\
a_n & a_{n-1} \ldots\ldots a_1 & a_0 \\
b_0^{n-1} & b_1^{n-1} \ldots\ldots b_{n-1}^{n-1} & \\
a_{n-1}^{n-1} & a_{n-2}^{n-1} \ldots a_0^{n-1} & \\
\ldots\ldots\ldots & & \\
\ldots\ldots\ldots & & \\
b_0^{1} & b_1^{1} & \\
a_1^{1} & a_0^{1} & \\
b_0^{0} & &
\end{array}
$$

Každý sudý řádek tabulky koeficientů A získáme zápisem koeficientů předcházející řady v opačném pořadí. Sudé řádky A- a B- tabulek mají stejné koeficienty. Elementy lichých řádků obou tabulek získáme pomocí výrazů

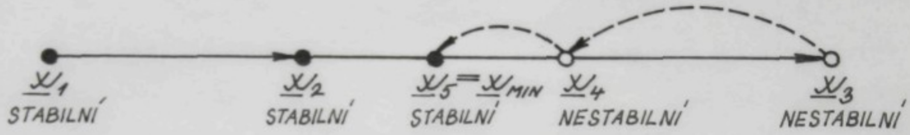
$$
a_i^{k-1} = a_i^{k} - \alpha_k \, a_{k-1}^{k} \qquad \alpha_k = a_k^{k} / a_0^{k}
$$

$$
b_i^{k-1} = b_i^{k} - \beta_k \, a_{k-1}^{k} \qquad \beta_k = b_k^{k} / a_0^{k}
$$

Podle kritéria stability z teorému /2.2.-2/:
Všechny nulové body polynomu $A(z)$ leží uvnitř jednotkové kružnice, jestliže všechny koeficienty a_0^{k} jsou kladné /v tabulce jsou zvýrazněny/. Po získání koeficientů α_k a β_k není těžké vyčíslit hodnotu integrálu podle formule /2.2.-25/.

4.4. Ošetření nestability

Předpokládáme, že při optimalizaci parametrů regulátoru leží zvolené výchozí seřízení parametrů ve stabilní oblasti. V případě, že bychom se při hledání extrému dosta-li do nestabilní oblasti použijeme postup graficky znázorněný na obr. /4.4.-1/. První stabilní bod nalezený při zpětném chodu bude uznán jako minimum v daném směru.



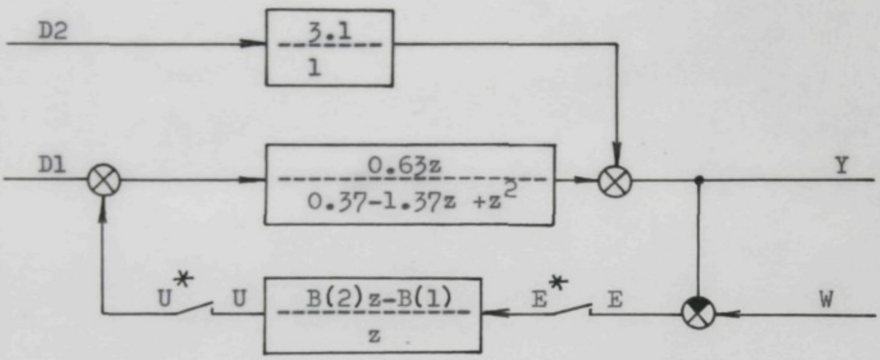
obr. /4.4.-1/

Kontrola stability je prováděna v podprogramu KRITER /viz kap. 5.4./.

Leží-li hodnoty startu v nestabilní oblasti, je i přesto možné úspěšné vyhledávání.[6] Aby se zcela využilo předností analytického výpočtu funkční hodnoty, především ve vztahu k době výpočtu, je třeba, aby startovní parametry byly dány ve stabilní oblasti předem. Takové parametry lze často hrubým odhadem určit.

5. Ověření činnosti programu

Mějme uzavřený regulační obvod:



Zvolíme výchozí seřízení B(1) = -0.50 a B(2) = +2.00.

5.1. Porucha typu jednotkový skok

Jmenovatel výsledného obrazového přenosu:

$Q = -0.37 + 2.055z - 3.945z^2 + 2.26z^3$

Pomocí diskrétní verze Routhova-Shurova algoritmu zkontrolujeme, je-li výchozí seřízení pro tento typ poruchy stabilní.

$\boxed{2.26}$	-3.945	2.055	-0.37
-0.37	2.055	-3.945	2.26
$\boxed{2.20}$	-3.61	1.41	
1.41	-3.61	2.20	
$\boxed{1.30}$	-1.30		
-1.30	1.30		
$\boxed{0.00}$			

Vidíme, že α_0 není větší než nula, a proto Q je nestabilní polynom.

5.2. Porucha typu jednotkový pulz

Jmenovatel výsledného obrazového přenosu:

$Q = +0.37 - 1.685z + 2.26z^2$

Opět zkontrolujeme, je-li Q stabilní polynom.

$\boxed{2.26}$	-1.685	0.37
0.37	-1.685	2.26
$\boxed{2.20}$	-1.41	
-1.41	2.20	
$\boxed{1.30}$		

Vidíme, že všechny členy α_0 jsou kladné a tedy polynom Q je stabilní.

Vzhledem k tomu, že velikost parametru $B(2) \rightarrow \infty$ pro hodnotu kritéria $\rightarrow 0$, zvolila jsem na velikost parametrů omezení. Dosáhne-li tedy parametr hodnoty vyšší než je toto omezení, je toto seřízení považováno za nestabilní a výpočet probíhá dále podle optimalizační metody.

Výsledky příkladů jsou uvedeny v tabulce 5.2.-1 a v příloze č. 1 - 3 .

Tabulka 5.2.-1

místo vstupu poruchy	B 1	B 2	hodnota kritéria výchozího seřízení	hodnota kritéria výsledného seřízení
porucha na řízení L = 1	-1.7287655	10.0000744	0.287042 E 00	0.388487 E-01
porucha na vstupu soustavy L = 2	2.1276208	10.0001345	0.135445 E 00	0.746705 E-02
porucha D2 L = 3	-8.0802946	10.0001364	0.275847 E 01	0.227809 E 00

6. Z p r á v a   o   p r o g r a m u

Programovací jazyk: FORTRAN IV

Typ počítače: EC 1033

6.1. Popis vstupních dat

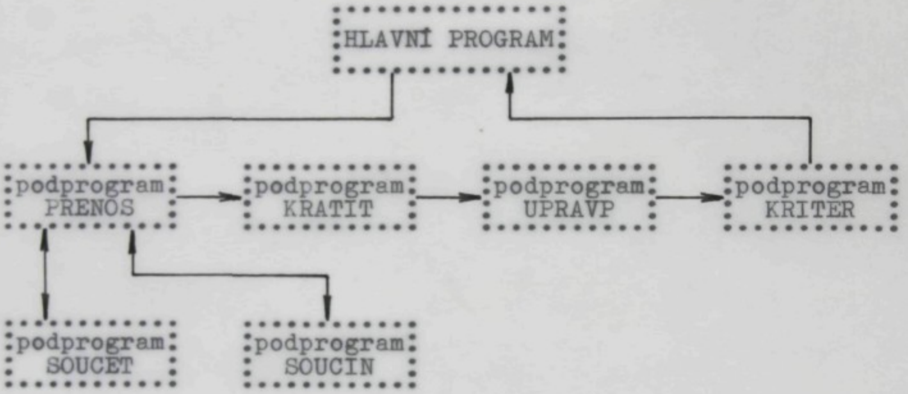
Data jsou načítána instrukcemi:

```
READ(5,500)EPSIL,N,MA,MB,MU,MV,ME,MF,L,K,THETA,OMEGA
READ(5,530)(B(I),I=1,MB),(A(I),I=1,MA)
READ(5,520)(U(I),I=1,MU),(V(I),I=1,MV)
READ(5,540)(E(I),I=1,ME),(F(I),I=1,MF)
READ(5,510)(RMINX(I),I=1,N)
READ(5,550)(VJ(I),I=1,N),(VK(I),I=1,N),(VL(I),I=1,N)
```

identifikátor	formát	význam identifikátoru
EPSIL	F10.7	délka startovního kroku optimalizační metody
N	I1	počet složek vektoru X
MA,MB,MU,MV,	I2	délky polynomů A,B,U,V
ME,MF	I2	délky polynomů E,F
L	I1	typ přenosu /přepínač/
K	I1	typ poruchy /přepínač/
B/A	F5.2	polynomy přenosu regulátoru
U/V	F5.2	polynomy přenosu HS(Z)
E/F	F5,2	polynomy přenosu poruchy D2
RMINX	F5.2	výchozí seřízení regulátoru
VJ,VK,VL	F4.1	výchozí směry hledání $VJ=\begin{bmatrix}1\\0\\0\end{bmatrix}$ $VK=\begin{bmatrix}0\\1\\0\end{bmatrix}$ $VL=\begin{bmatrix}0\\0\\1\end{bmatrix}$
THETA	F6.3	přesnost použitá v kritériu přerušení
OMEGA	F10.7	minimální velikost aktuální délky kroku DELTA

6.2. Popis struktury programu a podprogramů

Program je sestaven z hlavního programu a šesti podprogramů.



SUBROUTINE PRENOS(L,A,B,U,V,P,Q,MA,MB,MU,MV,MP,MQ,K)

Program pro výpočet přenosu uzavřeného regulačního obvodu.

Vstupní parametry:

B(Z)/A(Z) - polynomy přenosu regulátoru

A = A(1) + A(2)*Z + A(3)*Z**2 + ... + A(MA)*Z**(MA-1)
B = B(1) + B(2)*Z + B(3)*Z**2 + ... + B(MB)*Z**(MB-1)

U(Z)/V(Z) - polynomy přenosu HS Z

U = U(1) + U(2)*Z + U(3)*Z**2 + ... + U(MU)*Z**(MU-1)
V = V(1) + V(2)*Z + V(3)*Z**2 + ... + V(MV)*Z**(MV-1)

E(Z)/F(Z) - polynomy přenosu poruchy D2

E = E(1) + E(2)*Z + E(3)*Z**2 + ... + E(ME)*Z**(ME-1)
F = F(1) + F(2)*Z + F(3)*Z**2 + ... + F(MF)*Z**(MF-1)

MA,MB,MU,MV,ME,MF - délky polynomů A,B,U,V,E,F

L - přepínač

L = 1 /návěští 116/ přenos od poruchy W
L = 2 /návěští 117/ přenos od poruchy D1
L = 3 /návěští 118/ přenos od poruchy D2

K - přepínač
K = 1 /návěští 122/ porucha typu jednotkový pulz
K = 2 /návěští 121/ porucha typu jednotkový skok

Výstupní parametry:

P(Z)/Q(Z) - polynomy výsledného přenosu
P = P(1) + P(2)*Z + P(3)*Z**2 + ... + P(MP)*Z**(MP-1)
Q = Q(1) + Q(2)*Z + Q(3)*Z**2 + ... + Q(MQ)*Z**(MQ-1)

MP,MQ - délky polynomů P,Q

SUBROUTINE PRENOS využívá SUBROUTINE SOUCET
SUBROUTINE SOUCIN

SUBROUTINE SOUCET(A,B,C,MA,MB,MC)

Program pro součet dvou polynomů.

Vstupní parametry:

A = A(1) + A(2)*Z + A(3)*Z**2 + ... + A(MA)*Z**(MA-1)
B = B(1) + B(2)*Z + B(3)*Z**2 + ... + B(MB)*Z**(MB-1)

MA,MB - délky polynomů A,B

Výstupní parametry:

C - výsledný polynom C = A + B
C = C(1) + C(2)*Z + C(3)*Z**2 + ... + C(MC)*Z**(MC-1)

MC - délka polynomu C

SUBROUTINE NASOB(A,B,C,MA,MB,MC)

Program pro násobení polynomu polynomem.

Vstupní parametry:

A = A(1) + A(2)*Z + A(3)*Z**2 + ... + A(MA)*Z**(MA-1)
B = B(1) + B(2)*Z + B(3)*Z**2 + ... + B(MB)*Z**(MB-1)

MA,MB - délky polynomů A,B

Výstupní parametry:

C - výsledný polynom C = A * B
C = C(1) + C(2)*Z + C(3)*Z**2 + ... + C(MC)*Z**(MC-1)

MC - délka polynomu C

SUBROUTINE KRATIT(P,Q,MP,MQ,KRATP,KRATQ,KP,KQ)

Program pro krácení polynomů přenosu regulačního obvodu.

Vstupní parametry:

P(Z)/Q(Z) - polynomy přenosu
P = P(1) + P(2)*Z + P(3)*Z**2 + ... + P(MP)*Z**(MP-1)
Q = Q(1) + Q(2)*Z + Q(3)*Z**2 + ... + Q(MQ)*Z**(MQ-1)

MP,MQ - délky polynomů P,Q

Výstupní parametry:

KRATP(Z)/KRATQ(Z) - vykrácené polynomy P(Z)/Q(Z)
KRATP = KRATP(1) + KRATP(2)*Z + ...+ KRATP(KP)*Z**(KP-1)
KRATQ = KRATQ(1) + KRATQ(2)*Z + ... + KRATQ(KQ)*Z**(KQ-1)

KP,KQ - délky polynomů KRATP,KRATQ

SUBROUTINE UPRAVP(MP,MQ,P,Q,PP,QP,MPP)

Program pro úpravu polynomů P a Q na tvar vstupních parametrů SUBROUTINE KRITER.

Vstupní parametry:

P = P(1) + P(2)*Z + P(3)*Z**2 + ... + P(MP)*Z**(MP-1)
Q = Q(1) + Q(2)*Z + Q(3)*Z**2 + ... + Q(MQ)*Z**(MQ-1)

MP,MQ - délky polynomů P,Q

Výstupní parametry:

PP - upravený polynom P

PP = PP(1)*Z**MPP + PP(2)*Z**(MPP-1) + ... + PP(MPP+1)

QP - upravený polynom Q

QP = QP(1)*Z**MPP + QP(2)*Z**(MPP-1)+ ... + QP(MPP+1)

MPP - stupeň polynomů PP,QP

SUBROUTINE KRITER(A,B,N,VK,IERR)

Program pro vyčíslení integrálu racionální funkce

1/(2*PI*I)*B(Z)*B(1/Z)/(A(Z)*A(1/Z))*(1/Z)

podél jednotkové kružnice.

Vstupní parametry:

A = A(1)*Z**N + A(2)*Z**(N-1) + ... + A(N+1)

B = B(1)*Z**N + B(2)*Z**(N-1) + ... + B(N+1)

N - stupeň polynomů A a B /max. 10/

Výstupní parametry:

VK - vyčíslená hodnota integrálu

IERR - testovací identifikátor stability

IERR = 1 /stabilní/, A má všechny kořeny uvnitř jednotkové kružnice

IERR = 0 /nestabilní/, A má nějaký kořen vně nebo na hranici jednotkové kružnice nebo A(1) není kladné.

6.3. Použité identifikátory, návěští a přepínače

identifikátor	význam
KC	číslo cyklu
KF	číslo kroku
DELTA	délka kroku
UK	počet úspěšných kroků v prověřovaném směru
IN	počet prověřovaných směrů

F0,F1,F2 - body sloužící k odhadu minima pomocí náhradní paraboly

S1,S2,S3 - vypočítané nové směry hledání

ROZDIL - rozdíl funkčních hodnot dvou po sobě jdoucích cyklů musí být menší než předem stanovené THETA /tj. kritérium přerušení/

návěští	bude následovat operace
777	tisk hlavičky
500	čtení dat
1	tisk zadaných hodnot
1000	optimalizace v jednom směru
1001	prověřování 1 souboru směrů /1 cyklus/
932	výpočet nových směrů hledání
21	tisk vypočtených nových směrů hledání
600	tisk mezivýsledků
611	tisk minima v daném směru

přepínače

IN
IN = 1 /návěští 920/ - je prověřen 1 směr
IN = 2 /návěští 940/ - je prověřen 2. směr
IN = 3 /návěští 930/ - jsou prověřeny všechny tři směry

6.4. Popis výstupních dat

Výstupní data jsou tištěna na širokou tiskárnu. Za hlavičkou programu se vytisknou zadané obrazové přenosy jednotlivých členů regulačního obvodu, místo vstupu poruchy, typ poruchy a abrázek znázorňující strukturu obvodu. Z vypočtených hodnot se nejdříve vytisknou koeficienty obrazového přenosu uzavřeného regulačního obvodu. Dále se tiskne tabulka,,která obsahuje číslo kroku, hodnoty seřizovaných parametrů regulátoru, hodnotu kritéria a identifikátor stability. Během výpočtu je tabulka přerušena tiskem minima v daném směru a tiskem vypočtených nových směrů hledání.

6.5. Omezení programu

Program umožňuje seřizovat 1, 2 nebo 3 parametry regulátoru /B(1),B(2),A(1)/. Seřízení jiné kombinace tří parametrů vyžaduje celkem jednoduchý zásah do hlavního programu. Poněkud složitější úpravou lze zvýšit počet seřizovaných parametrů. Avšak při více jak 4 parametrech roste pracnost programování výpočtu nových směrů hledání.

7. Závěr

Sestaveným programem bylo ověřeno, že i tento postup řešení optimálního seřízení diskrétního regulátoru je schůdný.

Použitá metoda je numericky nenáročná, má malou spotřebu výpočtového času /3 - 4 s strojového času počítače/ a je nenáročná na paměť počítače /44 - 46 K/. Proto je tato metoda vhodná především pro malé počítače při nasazení "on-line". Kromě toho se může použít k určování parametrů při adaptivní regulaci.

Z časových důvodů nebylo možné odsimulovat více příkladů a ověřit vlastnosti regulátorů, které jsou seřízeny tímto způsobem.

Uvedený postup se týká minimalizace součtu čtverců odchylky.

$$I = \sum_{k=0}^{\infty} e^2(k)$$

Zásahem do podprogramu PRENOS by bylo možné přejít na kombinované kritérium s omezením na velikost akčních zásahů.

$$I = \sum_{k=0}^{\infty} [\, e^2(k) + \varkappa\, u^2(k)]$$

kde $\varkappa \geq 0$ je váhová konstanta.

Při určování obrazového přenosu uzavřeného regulačního obvodu jsem narazila na problém krácení. Případ, kdy lze vytknout mocninu z z čitatele i jmenovatele je vyřešen v podprogramu KRATIT. Případ, kdy lze vytknout např. $(z-a)$, program neřeší.

Seznam použité literatury

[1] ÅSTRÖM KARL J. - Vvedenie v stochastičeskuju teoriju upravlenija. /Překlad z angličtiny./
Moskva, Izdat. Mir 1973

[2] HANUŠ B. - Optimalizace systému řízení. /Skripta./
Liberec, VŠST 1978

[3] HANUŠ B. - Základy teorie lineárního impulsního regulačního obvodu, díl I., II. /Skripta./
Liberec, VŠST 1972

[4] HIMMELBLAU DAVID M. - Prikladnoe nelinejnoe programmirovanie. / Překlad z angličtiny./
Moskva, Izdat. Mir 1975

[5] STREJC V. - Teorie automatického řízení diskrétních systémů. /Skripta./
Praha, ČVUT 1979

[6] SCHENK R. - Lineares Suchverfahren zur Parameteroptimierung digitaler Regler. Regelungstechnik /28/ 1980, sešit 9, str. 299-304

[7] VOGEL J. - Programování v jazyku FORTRAN.
Praha, SNTL 1981

S e z n a m p ř í l o h

```
*******************************************************************************************************************************
          O P T I M A L N I   S E R I Z E N I      D I S K R E T N I H O   R E G U L A T O R U                           ***
                  PODLE MINIMA KVADRATICKE REGULA   CNI PLOCHY UZAVRENEHO REGULACNIHO OBVODU                              ***
*******************************************************************************************************************************

NE OBRAZOVE PRENOSY JEDNOTLIVYCH CLENU REGULACNIHO OBVODU
*********************************************************

   PRENOS SOUSTAVY HS(Z)=U(Z)/V(Z)
   U=  0.0 *Z**0 0.63*Z**1
   V=  0.37*Z**0-1.37*Z**1 1.00*Z**2

   PRENOS PORUCHY D2 =E(Z)/F(Z)
   E= 0
   F= 0

   PRENOS REGULATORU R(Z)=B(Z)/A(Z) -VYCHOZI SERIZENI
   B= -0.50*Z**0 2.00*Z**1
   A=  0.0 *Z**0 1.00*Z**1

   PORUCHA       L =1
   TYP PORUCHY   K =1
 0.0         0.3699999  -1.3699999   1.0000000
 0.0         0.3699699  -1.6849995   2.2599993
```

```
 D2                 *******
******************* E/F *******
                    *******     *
                                *
                                *
                                *
                                * +
 D1    ***          *******    ***        Y
*******   ********* U/V *****    ***********
       ***          *******    ***    *
        *                             *
        *                             *
        *                             *
        *   *                 *       * -
        *   U     U ******* E    E    ***   W
        *****/ **** B/A *****/ *****     ******
                    *******           ***
```

KF	B(1)	B(2)	A(1)	KRITERIUM	IERR
0	-0.5000000	2.0000000	0.0	0.287042E 00	1
1	-0.4999000	2.0000000	0.0	0.287048E 00	1
2	-0.5001000	2.0000000	0.0	0.287035E 00	1
3	-0.5002999	2.0000000	0.0	0.287021E 00	1
4	-0.5006999	2.0000000	0.0	0.286995E 00	1
5	-0.5014998	2.0000000	0.0	0.286941E 00	1
6	-0.5030998	2.0000000	0.0	0.286833E 00	1
7	-0.5062998	2.0000000	0.0	0.286618E 00	1
8	-0.5126998	2.0000000	0.0	0.286190E 00	1
9	-0.5254998	2.0000000	0.0	0.285334E 00	1
10	-0.5510997	2.0000000	0.0	0.283633E 00	1
11	-0.6022997	2.0000000	0.0	0.280268E 00	1
12	-0.7046996	2.0000000	0.0	0.273684E 00	1
13	-0.9094995	2.0000000	0.0	0.261074E 00	1
14	-1.3190985	2.0000000	0.0	0.237876E 00	1
15	-2.1382980	2.0000000	0.0	0.449733E 00	0
16	-1.7286978	2.0000000	0.0	0.217029E 00	1
17	-1.5575743	2.0000000	0.0	0.225476E 00	1

NIMUM V DANEM SMERU

-1.7286978	2.0000000	0.0	0.217029E 00

1	-1.7286978	2.0000992	0.0	0.217022E 00	1
2	-1.7286978	2.0002985	0.0	0.217008E 00	1
3	-1.7286978	2.0006981	0.0	0.216979E 00	1
4	-1.7286978	2.0014973	0.0	0.216921E 00	1
5	-1.7286978	2.0030966	0.0	0.216805E 00	1
6	-1.7286978	2.0062962	0.0	0.216574E 00	1
7	-1.7286978	2.0126953	0.0	0.216113E 00	1
8	-1.7286978	2.0254946	0.0	0.215195E 00	1
9	-1.7286978	2.0510941	0.0	0.213375E 00	1
10	-1.7286978	2.1022940	0.0	0.209798E 00	1
11	-1.7286978	2.2046938	0.0	0.202892E 00	1
12	-1.7286978	2.4094934	0.0	0.190013E 00	1
13	-1.7286978	2.8190928	0.0	0.167555E 00	1
14	-1.7286978	3.6382923	0.0	0.132940E 00	1
15	-1.7286978	5.2766914	0.0	0.894295E-01	1
16	-1.7286978	8.5534897	0.0	0.484166E-01	1
17	-1.7286978	15.1070862	0.0	0.207832E-01	0
18	-1.7286978	11.8302879	0.0	0.303322E-01	0
19	-1.7286978	10.1918888	0.0	0.377976E-01	0
20	-1.7286978	9.3726892	0.0	0.426136E-01	1
21	-1.7286978	11.0110884	0.0	0.337565E-01	0
22	-1.7286978	10.1918888	0.0	0.377976E-01	0
23	-1.7286978	9.7822886	0.0	0.400970E-01	1
24	-1.7286978	10.6014881	0.0	0.356911E-01	0
25	-1.7286978	10.1918879	0.0	0.377976E-01	0
26	-1.7286978	9.9870872	0.0	0.389217E-01	1
27	-1.7286978	10.3966866	0.0	0.367217E-01	0
28	-1.7286978	10.1918859	0.0	0.377976E-01	0
29	-1.7286978	10.0894852	0.0	0.383535E-01	0
30	-1.7286978	10.0382843	0.0	0.386361E-01	0
31	-1.7286978	10.0126839	0.0	0.387785E-01	0
32	-1.7286978	9.9998837	0.0	0.388501E-01	1
33	-1.7286978	10.0254831	0.0	0.387072E-01	0
34	-1.7286978	10.0126829	0.0	0.387785E-01	0
35	-1.7286978	10.0062828	0.0	0.388143E-01	0

39	-1.7286978	1Ø.ØØØ2813	Ø.Ø	Ø.388478E-Ø1	Ø
4Ø	-1.7286978	1Ø.ØØØØ811	Ø.Ø	Ø.38849ØE-Ø1	Ø
41	-1.7286978	9.99998Ø9	Ø.Ø	Ø.388495E-Ø1	1
42	-1.7286978	1Ø.ØØØ18Ø2	Ø.Ø	Ø.388484E-Ø1	Ø
43	-1.7286978	1Ø.ØØØØ8Ø1	Ø.Ø	Ø.38849ØE-Ø1	Ø
44	-1.7286978	1Ø.ØØØØ296	Ø.Ø	Ø.388492E-Ø1	Ø
45	-1.7286978	1Ø.ØØØØØ38	Ø.Ø	Ø.388494E-Ø1	Ø
46	-1.7286978	9.99999Ø5	Ø.Ø	Ø.388495E-Ø1	1
47	-1.7286978	1Ø.ØØØØ153	Ø.Ø	Ø.388493E-Ø1	Ø
48	-1.7286978	1Ø.ØØØØØ19	Ø.Ø	Ø.388494E-Ø1	Ø
49	-1.7286978	9.9999952	Ø.Ø	Ø.388494E-Ø1	1
5Ø	-1.7286978	1Ø.ØØØØØ76	Ø.Ø	Ø.388494E-Ø1	Ø
51	-1.7286978	1Ø.ØØØØØ1Ø	Ø.Ø	Ø.388494E-Ø1	Ø
52	-1.7286978	9.9999971	Ø.Ø	Ø.388494E-Ø1	1
53	-1.7286978	1Ø.ØØØØØ29	Ø.Ø	Ø.388494E-Ø1	Ø
54	-1.7286978	9.999999Ø	Ø.Ø	Ø.388494E-Ø1	1
55	-1.7286978	1Ø.ØØØØØ48	Ø.Ø	Ø.388494E-Ø1	Ø
56	-1.7286978	1Ø.ØØØØØ1Ø	Ø.Ø	Ø.388494E-Ø1	Ø
57	-1.7286978	9.999999Ø	Ø.Ø	Ø.388494E-Ø1	1
58	-1.7286978	9.9999981	Ø.Ø	Ø.388494E-Ø1	1
59	-1.7286978	9.999999Ø	Ø.Ø	Ø.388494E-Ø1	1

INIMUM V DANEM SMERU

-1.7286978	9.999999Ø	Ø.Ø	Ø.388494E-Ø1

1	-1.7286978	9.999999Ø	Ø.Ø	Ø.388494E-Ø1	1
2	-1.7286978	9.999999Ø	Ø.Ø	Ø.388494E-Ø1	1
3	-1.7286978	9.999999Ø	Ø.Ø	Ø.388494E-Ø1	1
4	-1.7286978	9.999999Ø	Ø.Ø	Ø.388494E-Ø1	1

INIMUM V DANEM SMERU

-1.7286978	9.999999Ø	Ø.Ø	Ø.388494E-Ø1

YPOCITANE NOVE SMERY HLEDANI

1= -Ø.489 Ø.873 Ø.Ø S2= Ø.873 Ø.489 Ø.Ø S3= Ø.Ø Ø.Ø Ø.Ø

1	-1.7287464	1Ø.ØØØØ858	Ø.Ø	Ø.388487E-Ø1	1
2	-1.7288437	1Ø.ØØØ2594	Ø.Ø	Ø.388475E-Ø1	Ø
3	-1.7287941	1Ø.ØØØ1717	Ø.Ø	Ø.388481E-Ø1	Ø
4	-1.7287693	1Ø.ØØØ1278	Ø.Ø	Ø.388484E-Ø1	Ø
5	-1.7287569	1Ø.ØØØ1Ø59	Ø.Ø	Ø.388486E-Ø1	Ø
6	-1.72875Ø2	1Ø.ØØØØ944	Ø.Ø	Ø.388487E-Ø1	Ø
7	-1.7287464	1Ø.ØØØØ887	Ø.Ø	Ø.388487E-Ø1	Ø
8	-1.7287445	1Ø.ØØØØ858	Ø.Ø	Ø.388487E-Ø1	Ø
9	-1.7287436	1Ø.ØØØØ839	Ø.Ø	Ø.388488E-Ø1	1
1Ø	-1.7287464	1Ø.ØØØØ858	Ø.Ø	Ø.388487E-Ø1	1

INIMUM V DANEM SMERU

-1.7287464	1Ø.ØØØØ858	Ø.Ø	Ø.388487E-Ø1

1	[illegible]	[illegible]			1
2	-1.7286978	9.9999990	0.0	0.388494E-01	1
3	-1.7286978	9.9999990	0.0	0.388494E-01	1
4	-1.7286978	9.9999990	0.0	0.388494E-01	1

IMUM V DANEM SMERU

-1.7286978 9.9999990 0.0 0.388494E-01

OCITANE NOVE SMERY HLEDANI

= -0.489 0.873 0.0 S2= 0.873 0.489 0.0 S3= 0.0 0.0 0.0

1	-1.7287464	10.0000858	0.0	0.388487E-01	1
2	-1.7288437	10.0002594	0.0	0.388475E-01	0
3	-1.7287941	10.0001717	0.0	0.388481E-01	0
4	-1.7287693	10.0001278	0.0	0.388484E-01	0
5	-1.7287569	10.0001059	0.0	0.388486E-01	0
6	-1.7287502	10.0000944	0.0	0.388487E-01	0
7	-1.7287464	10.0000887	0.0	0.388487E-01	0
8	-1.7287445	10.0000858	0.0	0.388487E-01	0
9	-1.7287436	10.0000839	0.0	0.388488E-01	1
10	-1.7287464	10.0000858	0.0	0.388487E-01	1

IMUM V DANEM SMERU

-1.7287464 10.0000858 0.0 0.388487E-01

1	-1.7286587	10.0001345	0.0	0.388488E-01	1
2	-1.7288332	10.0000362	0.0	0.388488E-01	1
3	-1.7287655	10.0000744	0.0	0.388487E-01	1

IMUM V DANEM SMERU

-1.7287655 10.0000744 0.0 0.388487E-01

1	-1.7287655	10.0000744	0.0	0.388487E-01	1
2	-1.7287655	10.0000744	0.0	0.388487E-01	0
3	-1.7287655	10.0000744	0.0	0.388487E-01	1
4	-1.7287655	10.0000744	0.0	0.388487E-01	1

IMUM V DANEM SMERU

-1.7287655 10.0000744 0.0 0.388487E-01

```
*************************************************************************************************************
        O P T I M A L N I   S E R I Z E N I     D I S K R E T N I H O   R E G U L A T O R U                **
              PODLE MINIMA KVADRATICKE REGULA   CNI PLOCHY UZAVRENEHO REGULACNIHO OBVODU                     **
*************************************************************************************************************

ANE OBRAZOVE PRENOSY JEDNOTLIVYCH CLENU REGULACNIHO OBVODU          D2            *******
**********************************************************    ***************** E/F *******
                                                                                  *******      *
   PRENOS SOUSTAVY HS(Z)=U(Z)/V(Z)                                                             *
   U=  0.0 *Z**0 0.63*Z**1                                                                     *
   V=  0.37*Z**0-1.37*Z**1 1.00*Z**2                                                           *
                                                                                               * +
   PRENOS PORUCHY D2 =E(Z)/F(Z)                                      D1    ***       *******   ***        Y
   E= 0                                                           *******  ********* U/V ******  ************
   F= 0                                                                    ***       *******   ***    *
                                                                           *                          *
   PRENOS REGULATORU R(Z)=B(Z)/A(Z) -VYCHOZI SERIZENI                      *                          *
   B= -0.50*Z**0 2.00*Z**1                                                 *                          *
   A=  0.0 *Z**0 1.00*Z**1                                                 *   *             *        * -
                                                                           *  U     U ******* E    E  ***   W
   PORUCHA        L =2                                                     *****/ **** B/A *****/ *****   ******
   TYP PORUCHY    K =1                                                               *******          ***
 0.0           0.0        -0.6300000
 0.0           0.3699999  -1.6849995   2.2599993
```

KF	B(1)	B(2)	A(1)	KRITERIUM
0	-0.5000000	2.0000000	0.0	0.135445E 00
1	-0.4999000	2.0000000	0.0	0.135438E 00
2	-0.4997000	2.0000000	0.0	0.135424E 00
3	-0.4992999	2.0000000	0.0	0.135395E 00
4	-0.4984999	2.0000000	0.0	0.135339E 00
5	-0.4968999	2.0000000	0.0	0.135227E 00
6	-0.4936998	2.0000000	0.0	0.135003E 00
7	-0.4872998	2.0000000	0.0	0.134557E 00
8	-0.4744998	2.0000000	0.0	0.133679E 00
9	-0.4488997	2.0000000	0.0	0.131967E 00
10	-0.3976998	2.0000000	0.0	0.128720E 00
11	-0.2952998	2.0000000	0.0	0.122852E 00
12	-0.0904999	2.0000000	0.0	0.113164E 00
13	0.3190998	2.0000000	0.0	0.995061E-01
14	1.1382990	2.0000000	0.0	0.850916E-01
15	2.7766981	2.0000000	0.0	0.815442E-01
16	6.0534964	2.0000000	0.0	0.584309E 00
17	4.4150972	2.0000000	0.0	0.112153E 00
18	2.1276598	2.0000000	0.0	0.798581E-01

MINIMUM V DANEM SMERU

2.1276598	2.0000000	0.0	0.798581E-01

1	2.1276598	2.0000992	0.0	0.798534E-01
2	2.1276598	2.0002985	0.0	0.798444E-01
3	2.1276598	2.0006981	0.0	0.798261E-01
4	2.1276598	2.0014973	0.0	0.797896E-01
5	2.1276598	2.0030966	0.0	0.797165E-01
6	2.1276598	2.0062962	0.0	0.795707E-01
7	2.1276598	2.0126953	0.0	0.792803E-01
8	2.1276598	2.0254946	0.0	0.787043E-01
9	2.1276598	2.0510941	0.0	0.775710E-01
10	2.1276598	2.1022940	0.0	0.753773E-01
11	2.1276598	2.2046938	0.0	0.712621E-01
12	2.1276598	2.4094934	0.0	0.639884E-01
13	2.1276598	2.8190928	0.0	0.524392E-01
14	2.1276598	3.6382923	0.0	0.370918E-01
15	2.1276598	5.2766914	0.0	0.213823E-01
16	2.1276598	8.5534897	0.0	0.975717E-02
17	2.1276598	15.1070862	0.0	0.359253E-02
18	2.1276598	11.8302879	0.0	0.556530E-02
19	2.1276598	10.1918888	0.0	0.722530E-02
20	2.1276598	9.3726892	0.0	0.834902E-02
21	2.1276598	11.0110884	0.0	0.631423E-02
22	2.1276598	10.1918888	0.0	0.722530E-02
23	2.1276598	9.7822886	0.0	0.775671E-02
24	2.1276598	10.6014881	0.0	0.674674E-02
25	2.1276598	10.1918879	0.0	0.722530E-02
26	2.1276598	9.9870872	0.0	0.748393E-02
27	2.1276598	10.3966866	0.0	0.697987E-02
28	2.1276598	10.1918859	0.0	0.722531E-02
29	2.1276598	10.0894852	0.0	0.735292E-02
30	2.1276598	10.0382843	0.0	0.741799E-02
31	2.1276598	10.0126839	0.0	0.745086E-02
32	2.1276598	9.9998837	0.0	0.746737E-02
33	2.1276598	10.0254831	0.0	0.743440E-02
34	2.1276598	10.0126829	0.0	0.745086E-02
35	2.1276598	[illegible]	0.0	[illegible]

37	2.1276598	10.0014820	0.0	[illegible]	
38	2.1276598	10.0006819	0.0	0.746634E-02	0
39	2.1276598	10.0002813	0.0	0.746686E-02	0
40	2.1276598	10.0000811	0.0	0.746711E-02	0
41	2.1276598	9.9999809	0.0	0.746724E-02	1
42	2.1276598	10.0001802	0.0	0.746699E-02	0
43	2.1276598	10.0000801	0.0	0.746712E-02	0
44	2.1276598	10.0000296	0.0	0.746718E-02	0
45	2.1276598	10.0000038	0.0	0.746721E-02	0
46	2.1276598	9.9999905	0.0	0.746723E-02	1
47	2.1276598	10.0000153	0.0	0.746720E-02	0
48	2.1276598	10.0000019	0.0	0.746721E-02	0
49	2.1276598	9.9999952	0.0	0.746723E-02	1
50	2.1276598	10.0000076	0.0	0.746721E-02	0
51	2.1276598	10.0000010	0.0	0.746722E-02	0
52	2.1276598	9.9999971	0.0	0.746722E-02	1
53	2.1276598	10.0000029	0.0	0.746721E-02	0
54	2.1276598	9.9999990	0.0	0.746722E-02	1
55	2.1276598	10.0000048	0.0	0.746721E-02	0
56	2.1276598	10.0000010	0.0	0.746722E-02	0
57	2.1276598	9.9999990	0.0	0.746722E-02	1
58	2.1276598	9.9999990	0.0	0.746722E-02	1

INIMUM V DANEM SMERU

2.1276598 9.9999990 0.0 0.746722E-02

1	2.1276598	9.9999990	0.0	0.746722E-02	1
2	2.1276598	9.9999990	0.0	0.746722E-02	1
3	2.1276598	9.9999990	0.0	0.746722E-02	1
4	2.1276598	9.9999990	0.0	0.746722E-02	1

INIMUM V DANEM SMERU

2.1276598 9.9999990 0.0 0.746722E-02

YPOCITANE NOVE SMERY HLEDANI

1= 0.500 0.866 0.0 S2= -0.866 0.500 0.0 S3= 0.0 0.0 0.0

1	2.1277094	10.0000849	0.0	0.746711E-02	1
2	2.1278086	10.0002575	0.0	0.746689E-02	0
3	2.1277580	10.0001707	0.0	0.746700E-02	0
4	2.1277323	10.0001268	0.0	0.746705E-02	0
5	2.1277189	10.0001049	0.0	0.746708E-02	0
6	2.1277122	10.0000935	0.0	0.746710E-02	0
7	2.1277084	10.0000877	0.0	0.746711E-02	0
8	2.1277065	10.0000849	0.0	0.746711E-02	0
9	2.1277056	10.0000830	0.0	0.746711E-02	1
10	2.1277094	10.0000849	0.0	0.746711E-02	1

INIMUM V DANEM SMERU

2.1277094 10.0000849 0.0 0.746711E-02

[illegible]	[illegible]	[illegible]	[illegible]	0.746722E-02
2	2.1276598	9.9999990	0.0	0.746722E-02
3	2.1276598	9.9999990	0.0	0.746722E-02
4	2.1276598	9.9999990	0.0	0.746722E-02

MINIMUM V DANEM SMERU

2.1276598	9.9999990	0.0	0.746722E-02

VYPOCITANE NOVE SMERY HLEDANI

S1= 0.500 0.866 0.0 S2= -0.866 0.500 0.0 S3= 0.0 0.0

1	2.1277094	10.0000849	0.0	0.746711E-02
2	2.1278086	10.0002575	0.0	0.746689E-02
3	2.1277580	10.0001707	0.0	0.746700E-02
4	2.1277323	10.0001268	0.0	0.746705E-02
5	2.1277189	10.0001049	0.0	0.746708E-02
6	2.1277122	10.0000935	0.0	0.746710E-02
7	2.1277084	10.0000877	0.0	0.746711E-02
8	2.1277065	10.0000849	0.0	0.746711E-02
9	2.1277056	10.0000830	0.0	0.746711E-02
10	2.1277094	10.0000849	0.0	0.746711E-02

MINIMUM V DANEM SMERU

2.1277094	10.0000849	0.0	0.746711E-02

1	2.1276226	10.0001345	0.0	0.746705E-02
2	2.1274490	10.0002337	0.0	0.746692E-02
3	2.1275349	10.0001831	0.0	0.746698E-02
4	2.1275778	10.0001574	0.0	0.746702E-02
5	2.1275988	10.0001440	0.0	0.746704E-02
6	2.1276093	10.0001373	0.0	0.746704E-02
7	2.1276140	10.0001335	0.0	0.746705E-02
8	2.1276159	10.0001316	0.0	0.746705E-02
9	2.1276207	10.0001345	0.0	0.746705E-02

MINIMUM V DANEM SMERU

2.1276207	10.0001345	0.0	0.746705E-02

1	2.1276207	10.0001345	0.0	0.746705E-02
2	2.1276207	10.0001345	0.0	0.746705E-02
3	2.1276207	10.0001345	0.0	0.746705E-02
4	2.1276207	10.0001345	0.0	0.746705E-02

MINIMUM V DANEM SMERU

2.1276207	10.0001345	0.0	0.746705E-02

```
*****************************************************************************************************
        O P T I M A L N I   S E R I Z E N I    D I S K R E T N I H O   R E G U L A T O R U        **
              PODLE MINIMA KVADRATICKE REGULA   CNI PLOCHY UZAVRENEHO REGULACNIHO OBVODU            **
*****************************************************************************************************

NE OBRAZOVE PRENOSY JEDNOTLIVYCH CLENU REGULACNIHO OBVODU
*********************************************************

   PRENOS SOUSTAVY HS(Z)=U(Z)/V(Z)
   U=  0.0 *Z**0 0.63*Z**1
   V=  0.37*Z**0-1.37*Z**1 1.00*Z**2

   PRENOS PORUCHY D2 =E(Z)/F(Z)
   E= *
   F= *

   PRENOS REGULATORU R(Z)=B(Z)/A(Z) -VYCHOZI SERIZENI
   B= -0.5[illegible]*Z**0 2.00*Z**1
   A=  0.0 *Z**0 1.00*Z**1

   PORUCHA       L =3
   TYP PORUCHY   K =1
 0.0          -1.1469994   4.2469988  -3.0999994
 0.0           0.3699999  -1.6849995   2.2599993
```

```
   D2                  *******
   ****************** E/F *******
                       *******    *
                                  *
                                  *
                                  *
                                  * +
   D1    ***           *******   ***          Y
   *******   ********* U/V *****   ***********
         ***           *******   ***     *
          *                              *
          *                              *
          *                              *
          *   *               *          * -
          *  U    U ******* E    E  ***   W
          *****/ **** B/A *****/ *****   ******
                       *******          ***
```

KF	B(1)	B(2)	A(1)	KRITERIUM	IERR
0	-0.5000000	2.0000000	0.0	0.275847E 01	1
1	-0.4999000	2.0000000	0.0	0.275853E 01	1
2	-0.5001000	2.0000000	0.0	0.275840E 01	1
3	-0.5002999	2.0000000	0.0	0.275827E 01	1
4	-0.5006999	2.0000000	0.0	0.275802E 01	1
5	-0.5014998	2.0000000	0.0	0.275750E 01	1
6	-0.5030998	2.0000000	0.0	0.275647E 01	1
7	-0.5062998	2.0000000	0.0	0.275440E 01	1
8	-0.5126998	2.0000000	0.0	0.275028E 01	1
9	-0.5254998	2.0000000	0.0	0.274206E 01	1
10	-0.5510997	2.0000000	0.0	0.272571E 01	1
11	-0.6022997	2.0000000	0.0	0.269337E 01	1
12	-0.7046996	2.0000000	0.0	0.263011E 01	1
13	-0.9094995	2.0000000	0.0	0.250892E 01	1
14	-1.3190985	2.0000000	0.0	0.228598E 01	1
15	-2.1382980	2.0000000	0.0	0.432193E 01	0
16	-1.7286978	2.0000000	0.0	0.208565E 01	1
17	-1.5575743	2.0000000	0.0	0.216682E 01	1

INIMUM V DANEM SMERU

-1,7286978   2.0000000   0.0   0.208565E 01

1	-1.7286978	2.0000992	0.0	0.208558E 01	1
2	-1.7286978	2.0002985	0.0	0.208544E 01	1
3	-1.7286978	2.0006981	0.0	0.208517E 01	1
4	-1.7286978	2.0014973	0.0	0.208461E 01	1
5	-1.7286978	2.0030966	0.0	0.208350E 01	1
6	-1.7286978	2.0062962	0.0	0.208128E 01	1
7	-1.7286978	2.0126953	0.0	0.207685E 01	1
8	-1.7286978	2.0254946	0.0	0.206802E 01	1
9	-1.7286978	2.0510941	0.0	0.205053E 01	1
10	-1.7286978	2.1022940	0.0	0.201616E 01	1
11	-1.7286978	2.2046938	0.0	0.194979E 01	1
12	-1.7286978	2.4094934	0.0	0.182603E 01	1
13	-1.7286978	2.8190928	0.0	0.161021E 01	1
14	-1.7286978	3.6382923	0.0	0.127755E 01	1
15	-1.7286978	5.2766914	0.0	0.859417E 00	1
16	-1.7286978	8.5534897	0.0	0.465284E 00	1
17	-1.7286978	15.1070862	0.0	0.199727E 00	0
18	-1.7286978	11.8302879	0.0	0.291492E 00	0
19	-1.7286978	10.1918888	0.0	0.363235E 00	0
20	-1.7286978	9.3726892	0.0	0.409516E 00	1
21	-1.7286978	11.0110884	0.0	0.324400E 00	0
22	-1.7286978	10.1918888	0.0	0.363235E 00	0
23	-1.7286978	9.7822886	0.0	0.385332E 00	1
24	-1.7286978	10.6014881	0.0	0.342992E 00	0
25	-1.7286978	10.1918879	0.0	0.363235E 00	0
26	-1.7286978	9.9870872	0.0	0.374038E 00	1
27	-1.7286978	10.3966866	0.0	0.352895E 00	0
28	-1.7286978	10.1918859	0.0	0.363235E 00	0
29	-1.7286978	10.0894852	0.0	0.368577E 00	0
30	-1.7286978	10.0382843	0.0	0.371293E 00	0
31	-1.7286978	10.0126839	0.0	0.372662E 00	0
32	-1.7286978	9.9998837	0.0	0.373349E 00	1
33	-1.7286978	10.0254831	0.0	0.371976E 00	0
34	-1.7286978	10.0126829	0.0	0.372662E 00	0
35	-1.7286978	10.0062828	0.0	0.373005E 00	0
36	-1.7286978	10.0030823	0.0	[illegible]	[illegible]

```
35    -1.7286978    10.0062828    0.0    [illegible]       0
36    -1.7286978    10.0030823    0.0    0.373177E 00      0
37    -1.7286978    10.0014820    0.0    0.373263E 00      0
38    -1.7286978    10.0006819    0.0    0.373306E 00      0
39    -1.7286978    10.0002813    0.0    0.373327E 00      0
40    -1.7286978    10.0000811    0.0    0.373338E 00      0
41    -1.7286978     9.9999809    0.0    0.373344E 00      1
42    -1.7286978    10.0001802    0.0    0.373333E 00      0
43    -1.7286978    10.0000801    0.0    0.373338E 00      0
44    -1.7286978    10.0000296    0.0    0.373341E 00      0
45    -1.7286978    10.0000038    0.0    0.373342E 00      0
46    -1.7286978     9.9999905    0.0    0.373343E 00      1
47    -1.7286978    10.0000153    0.0    0.373342E 00      0
48    -1.7286978    10.0000019    0.0    0.373343E 00      0
49    -1.7286978     9.9999952    0.0    0.373343E 00      1
50    -1.7286978    10.0000076    0.0    0.373342E 00      0
51    -1.7286978    10.0000010    0.0    0.373343E 00      0
52    -1.7286978     9.9999971    0.0    0.373343E 00      1
53    -1.7286978    10.0000029    0.0    0.373343E 00      0
54    -1.7286978     9.9999990    0.0    0.373342E 00      1
55    -1.7286978    10.0000048    0.0    0.373342E 00      0
56    -1.7286978    10.0000010    0.0    0.373343E 00      1
57    -1.7286978     9.9999990    0.0    0.373342E 00      1


INIMUM V DANEM SMERU
      -1.7286978     9.9999990    0.0    0.373342E 00


 1    -1.7286978     9.9999990    0.0    0.373342E 00      1
 2    -1.7286978     9.9999990    0.0    0.373342E 00      1
 3    -1.7286978     9.9999990    0.0    0.373342E 00      1
 4    -1.7286978     9.9999990    0.0    0.373342E 00      1


INIMUM V DANEM SMERU
      -1.7286978     9.9999990    0.0    0.373342E 00



YPOCITANE NOVE SMERY HLEDANI

1=  -0.489  0.873  0.0   S2=  0.873  0.489  0.0   S3=  0.0   0.0   0.0


 1    -1.7287464    10.0000858    0.0    0.373336E 00      1
 2    -1.7288437    10.0002594    0.0    0.373324E 00      0
 3    -1.7287941    10.0001717    0.0    0.373330E 00      0
 4    -1.7287693    10.0001278    0.0    0.373334E 00      0
 5    -1.7287569    10.0001059    0.0    0.373335E 00      0
 6    -1.7287502    10.0000944    0.0    0.373336E 00      0
 7    -1.7287464    10.0000887    0.0    0.373336E 00      0
 8    -1.7287445    10.0000858    0.0    0.373336E 00      0
 9    -1.7287436    10.0000839    0.0    0.373337E 00      1
10    -1.7287464    10.0000858    0.0    0.373336E 00      1


INIMUM V DANEM SMERU
      [illegible]
```

INIMUM V DANEM SMERU

-1.7286978	9.9999990	0.0	0.373342E 00

1	-1.7286978	9.9999990	0.0	0.373342E 00	1
2	-1.7286978	9.9999990	0.0	0.373342E 00	1
3	-1.7286978	9.9999990	0.0	0.373342E 00	1
4	-1.7286978	9.9999990	0.0	0.373342E 00	1

INIMUM V DANEM SMERU

-1.7286978	9.9999990	0.0	0.373342E 00

YPOCITANE NOVE SMERY HLEDANI

1= -0.489 0.873 0.0 S2= 0.873 0.489 0.0 S3= 0.0 0.0 0.0

1	-1.7287464	10.0000858	0.0	0.373336E 00	1
2	-1.7288437	10.0002594	0.0	0.373324E 00	0
3	-1.7287941	10.0001717	0.0	0.373330E 00	0
4	-1.7287693	10.0001278	0.0	0.373334E 00	0
5	-1.7287569	10.0001059	0.0	0.373335E 00	0
6	-1.7287502	10.0000944	0.0	0.373336E 00	0
7	-1.7287464	10.0000887	0.0	0.373336E 00	0
8	-1.7287445	10.0000858	0.0	0.373336E 00	0
9	-1.7287436	10.0000839	0.0	0.373337E 00	1
10	-1.7287464	10.0000858	0.0	0.373336E 00	1

INIMUM V DANEM SMERU

-1.7287464	10.0000858	0.0	0.373336E 00

1	-1.7286587	10.0001345	0.0	0.373337E 00	1
2	-1.7288332	10.0000362	0.0	0.373336E 00	1
3	-1.7290268	9.9999380	0.0	0.373336E 00	1
4	-1.7293549	9.9997425	0.0	0.373335E 00	1
5	-1.7300520	9.9993515	0.0	0.373334E 00	1
6	-1.7314472	9.9985695	0.0	0.373331E 00	1
7	-1.7342386	9.9970055	0.0	0.373326E 00	1
8	-1.7398224	9.9938784	0.0	0.373316E 00	1
9	-1.7509899	9.9876242	0.0	0.373296E 00	1
10	-1.7733259	9.9751158	0.0	0.373254E 00	1
11	-1.8179979	9.9500990	0.0	0.373171E 00	1
12	-1.9073420	9.9000654	0.0	0.373005E 00	1
13	-2.0860310	9.7999992	0.0	0.372669E 00	1
14	-2.4434090	9.5998669	0.0	0.371984E 00	1
15	-3.1581650	9.1996021	0.0	0.370522E 00	1
16	-4.5876770	8.3990736	0.0	0.366913E 00	1
17	-7.4467220	6.7980185	0.0	0.188317E 01	0
18	-6.0171900	7.5985460	0.0	0.361541E 00	1
19	-5.3074627	7.9959927	0.0	0.364509E 00	1

19	-5.3874627	7.7959927	[illegible]		

NIMUM V DANEM SMERU

-6.0171900	7.5985460	0.0	0.361541E 00

1	-6.0171900	7.5985460	0.0	0.361541E 00	1
2	-6.0171900	7.5985460	0.0	0.361541E 00	1
3	-6.0171900	7.5985460	0.0	0.361541E 00	1
4	-6.0171900	7.5985460	0.0	0.361541E 00	1

INIMUM V DANEM SMERU

-6.0171900	7.5985460	0.0	0.361541E 00

YPOCITANE NOVE SMERY HLEDANI

1= -0.901 -0.433 0.0 S2= 0.433 -0.901 0.0 S3= 0.0 0.0 0.0

1	-6.0172796	7.5985022	0.0	0.361540E 00	1
2	-6.0174599	7.5984154	0.0	0.361539E 00	1
3	-6.0178204	7.5982418	0.0	0.361535E 00	1
4	-6.0185413	7.5978947	0.0	0.361529E 00	1
5	-6.0199833	7.5972013	0.0	0.361517E 00	1
6	-6.0228672	7.5958147	0.0	0.361492E 00	1
7	-6.0286350	7.5930414	0.0	0.361442E 00	1
8	-6.0401707	7.5874958	0.0	0.361341E 00	1
9	-6.0632429	7.5764046	0.0	0.361141E 00	1
10	-6.1093884	7.5542231	0.0	0.360736E 00	1
11	-6.2016792	7.5098600	0.0	0.359920E 00	1
12	-6.3862619	7.4211349	0.0	0.358254E 00	1
13	-6.7554274	7.2436848	0.0	0.354773E 00	1
14	-7.4937892	6.8887854	0.0	0.187137E 01	0
15	-7.1245928	7.0662346	0.0	0.191579E 01	0
16	-6.5716658	7.3320141	0.0	0.356532E 00	1

INIMUM V DANEM SMERU

-6.7554274	7.2436848	0.0	0.354773E 00

1	-6.7553835	7.2435942	0.0	0.354779E 00	1
2	-6.7554703	7.2437744	0.0	0.354767E 00	1
	-6.7555561	7.2439547	0.0	0.354756E 00	1
	-6.7557587	7.2443151	0.0	0.354733E 00	1
	-6.7560749	7.2450361	0.0	0.354687E 00	1
6	-6.7567873	7.2464781	0.0	0.354595E 00	1
7	-6.7581530	7.2493620	0.0	0.354412E 00	1
8	-6.7609253	7.2551298	0.0	0.354046E 00	1
9	-6.7664700	7.2666655	0.0	0.353314E 00	1
10	-6.7775602	7.2897377	0.0	0.351859E 00	1
1[illegible]	-6.7997408	7.3358831	0.0	0.348975E 00	1
1[illegible]	-6.8441019	7.4281740	0.0	0.343311E 00	1
[illegible]	[illegible]	7.6127567	0.0	0.332386E 00	1

[illegible]	[illegible]	7.4261740	0.0	[illegible]	[illegible]
1	-6.9328251	7.6127567	0.0	0.332386E 00	1
1	-7.1102715	7.9819231	0.0	0.312032E 00	1
1	-7.4651651	8.7202568	0.0	0.276523E 00	1
1	-8.1749535	10.1969252	0.0	0.221469E 00	0
1[illegible]	-7.8200588	9.4585905	0.0	0.246720E 00	1
1[illegible]	-8.5298471	10.9352589	0.0	0.199893E 00	0
1[illegible]	-8.1749525	10.1969242	0.0	0.221469E 00	0
20	-7.9975052	9.8277569	0.0	0.233585E 00	1
21	-8.3523089	10.5660906	0.0	0.210268E 00	0
22	-8.1749516	10.1969233	0.0	0.221469E 00	0
23	-8.0862274	10.0123396	0.0	0.227407E 00	0
24	-8.0418653	9.9200478	0.0	0.230465E 00	1
25	-8.1305885	10.1046305	0.0	0.224408E 00	0
26	-8.0862265	10.0123386	0.0	0.227407E 00	0
27	-8.0640450	9.9661922	0.0	0.228928E 00	1
28	-8.1084061	10.0584831	0.0	0.225900E 00	0
29	-8.0862246	10.0123367	0.0	0.227407E 00	0
30	-8.0751333	9.9892635	0.0	0.228166E 00	1
31	-8.0973139	10.0354090	0.0	0.226652E 00	0
32	-8.0862226	10.0123358	0.0	0.227407E 00	0
33	-8.0806770	10.0007992	0.0	0.227786E 00	0
34	-8.0779037	9.9950304	0.0	0.227975E 00	1
35	-8.0834484	10.0065660	0.0	0.227596E 00	0
36	-8.0806751	10.0007973	0.0	0.227786E 00	0
37	-8.0792885	9.9979124	0.0	0.227881E 00	1
38	-8.0820608	10.0036802	0.0	0.227691E 00	0
39	-8.0806742	10.0007954	0.0	0.227786E 00	0
40	-8.0799809	9.9993525	0.0	0.227834E 00	1
41	-8.0813665	10.0022364	0.0	0.227738E 00	0
42	-8.0806732	10.0007935	0.0	0.227786E 00	0
43	-8.0803261	10.0000725	0.0	0.227810E 00	0
44	-8.0801525	9.9997120	0.0	0.227822E 00	1
45	-8.0804987	10.0004330	0.0	0.227798E 00	0
46	-8.0803251	10.0000725	0.0	0.227810E 00	0
47	-8.0802383	9.9998922	0.0	0.227816E 00	1
48	-8.0804110	10.0002527	0.0	0.227804E 00	0
49	-8.0803242	10.0000725	0.0	0.227810E 00	0
50	-8.0802803	9.9999819	0.0	0.227813E 00	1
51	-8.0803861	10.0001621	0.0	0.227807E 00	0
52	-8.0803223	10.0000715	0.0	0.227810E 00	0
53	-8.0803003	10.0000257	0.0	0.227811E 00	0
54	-8.0802889	10.0000029	0.0	0.227812E 00	0
55	-8.0802832	9.9999914	0.0	0.227812E 00	1
56	-8.0802937	10.0000134	0.0	0.227812E 00	0
57	-8.0802879	10.0000019	0.0	0.227812E 00	0
58	-8.0802851	9.9999962	0.0	0.227812E 00	1
59	-8.0802898	10.0000067	0.0	0.227812E 00	0
60	-8.0802870	10.0000010	0.0	0.227812E 00	0
61	-8.0802851	9.9999981	0.0	0.227812E 00	1
62	-8.0802870	10.0000029	0.0	0.227812E 00	0
63	-8.0802851	10.0000000	0.0	0.227812E 00	1
64	-8.0802841	9.9999981	0.0	0.227812E 00	1
65	-8.0802841	9.9999981	0.0	0.227812E 00	1

INIMUM V DANEM SMERU

-8.0802841	9.9999981	0.0	0.227812E 00

1	-8.0802841	9.9999981	0.0	0.227812E 00	1
2	-8.0802841	9.9999981	0.0	0.227812E 00	1
3	-8.0802841	9.9999981	0.0	0.227812E 00	1
4	-8.0802841	9.9999981	0.0	0.227812E 00	1

PŘÍLOHA č.4


```
C PROGRAM PRO OPTIMALNI SERIZENI REGULATORU
C ****************************************
C
C  RMINX-VYCHOZI BOD
C  FMINX-FUNKCNI HODNOTA VYCHOZIHO BODU
C  VJ-SMER HLEDANI
C  EPSIL-DELKA STARTOVNIHO KROKU
C  KF-POCET KROKU
C  N-POCET SLOZEK VEKTORU X
C  UK-POCET USPESNYCH KROKU
C  IN-POCET PROVERENYCH SMERU
C  KC-CISLO CYKLU
C S1,S2,S3,-VYPOCITANE NOVE SMERY HLEDANI
C
      DIMENSION A(10),B(10),U(10),V(10),P(10),Q(10),PP(10),QP(10),E(10)
      DIMENSION X(100),RMINX(50),VJ(20),VK(20),BOD(20),F(10),VL(20)
      DIMENSION V1(30),V2(30),A1(30),A2(30),S1(20),S2(20),B2(20)
      DIMENSION S3(20),A3(30),B3(20),V3(30),KRATP(15),KRATQ(15)
      WRITE(6,7771)
 7771 FORMAT(1H1)
      WRITE(6,777)
  777 FORMAT(1H ,'**************************************************
     /***************************************************')
      WRITE(6,888)
  888 FORMAT(1X,'***          O P T I M A L N I   S E R I Z E N I
     /  D I S K R E T N I H O   R E G U L A T O R U          ***')
      WRITE(6,999)
  999 FORMAT(1X,'***                    PODLE MINIMA KVADRATICKE REGULA
     /CNI PLOCHY UZAVRENEHO REGULACNIHO OBVODU                ***')
      WRITE(6,777)
      READ(5,500)EPSIL,N,MA,MB,MU,MV,ME,MF,L,K,THETA,OMEGA
      READ(5,530)(B(I),I=1,MB),(A(I),I=1,MA)
      READ(5,520)(U(I),I=1,MU),(V(I),I=1,MV)
      READ(5,540)(E(I),I=1,ME),(F(I),I=1,MF)
      READ(5,510)(RMINX(I),I=1,N)
      READ(5,550)(VJ(I),I=1,N),(VK(I),I=1,N),(VL(I),I=1,N)
  500 FORMAT(F10.7,I1,6I2,2I1,F6.3,F10.7)
  530 FORMAT(4F5.2)
  520 FORMAT(5F5.2)
  540 FORMAT(2F5.2)
  510 FORMAT(3F5.2)
  550 FORMAT(9F4.1)
      WRITE(6,1)
      WRITE(6,2)
      WRITE(6,3)
      WRITE(6,4)
      WRITE(6,5)(U(I),I=1,MU)
      WRITE(6,6)(V(I),I=1,MV)
      WRITE(6,7)
      WRITE(6,8)
      WRITE(6,9)(E(I),I=1,ME)
      WRITE(6,10)(F(I),I=1,MF)
      WRITE(6,11)
      WRITE(6,12)
      WRITE(6,13)(B(I),I=1,MB)
      WRITE(6,14)(A(I),I=1,MA)
      WRITE(6,15)
      WRITE(6,16)L
```

```
      WRITE(6,17)K
    1 FORMAT('0ZADANE OBRAZOVE PRENOSY JEDNOTLIVYCH CLENU REGULACNIHO O
     /VODU',7X,'D2',14X,'*******')
    2 FORMAT(1X,'*************************************************************
     /******',5X,'****************** E/F ********')
    3 FORMAT(85X,'*******',6X,'*')
    4 FORMAT(9X,'PRENOS SOUSTAVY HS(Z)=U(Z)/V(Z)',58X,'*')
    5 FORMAT(9X,'U= ',F5.2,'*Z**0',F5.2,'*Z**1',66X,'*')
    6 FORMAT(9X,'V= ',F5.2,'*Z**0',F5.2,'*Z**1',F5.2,'*Z**2',56X,'*')
    7 FORMAT(98X,'* +')
    8 FORMAT(9X,'PRENOS PORUCHY D2 =E(Z)/F(Z)',32X,'D1    ***',8X,'*****
     /*      ***',9X,'Y')
    9 FORMAT(9X,'E= ',I1,54X,'*******    ********* U/V ******    ********
     /****')
   10 FORMAT(9X,'F= ',I1,61X,'***',8X,'*******',5X,'***',4X,'*')
   11 FORMAT(75X,'*',28X,'*')
   12 FORMAT(9X,'PRENOS REGULATORU R(Z)=B(Z)/A(Z) -VYCHOZI SERIZENI',16
     /,'*',28X,'*')
   13 FORMAT(9X,'B= ',F5.2,'*Z**0',F5.2,'*Z**1',43X,'*',28X,'*')
   14 FORMAT(9X,'A= ',F5.2,'*Z**0',F5.2,'*Z**1',43X,'*    *',15X,'*',8X,
     /* -')
   15 FORMAT(75X,'*  U    U *******  E    E   ***   W')
   16 FORMAT(9X,'PORUCHA       L =',I1,49X,'*****/ **** B/A *****/ *****
     /  ******')
   17 FORMAT(9X,'TYP PORUCHY  K =',I1,59X,'*******',11X,'***')
      CALL PRENOS(L,A,B,U,V,E,F,P,Q,MA,MB,MU,MV,ME,MF,MP,MQ,K)
      WRITE(6,610)(P(I),I=1,MP)
      WRITE(6,630)(Q(I),I=1,MQ)
  610 FORMAT(4H P= ,8F12.7)
  630 FORMAT(4H Q= ,8F12.7)
      CALL KRATIT(P,Q,MP,MQ,KRATP,KRATQ,KP,KQ)
      CALL UPRAVP(KP,KQ,KRATP,KRATQ,PP,QP,MPP)
      CALL KRITER(QP,PP,MPP,FMINX,IERR)
      KF=0
      WRITE(6,620)
      WRITE(6,640)
      WRITE(6,600)KF,B(1),B(2),A(1),FMINX,IERR
      KC=1
      BOD(1)=FMINX
 1001 IN=0
 1000 IN=IN+1
      KF=1
      DELTA=EPSIL
      F0=FMINX
      DO 410 I=1,N
      X(I)=RMINX(I)+DELTA*VJ(I)
  410 CONTINUE
  413 B(1)=X(1)
      B(2)=X(2)
      A(1)=X(3)
      CALL PRENOS(L,A,B,U,V,E,F,P,Q,MA,MB,MU,MV,ME,MF,MP,MQ,K)
      CALL KRATIT(P,Q,MP,MQ,KRATP,KRATQ,KP,KQ)
      CALL UPRAVP(KP,KQ,KRATP,KRATQ,PP,QP,MPP)
      CALL KRITER(QP,PP,MPP,AK,IERR)
      WRITE(6,600)KF,B(1),B(2),A(1),AK,IERR
  620 FORMAT('1  KF',7X,'B(1)',12X,'B(2)',8X,'A(1)',12X,'KRITERIUM',6X,
     /'IERR')
  600 FORMAT(2X,I3,4X,F10.7,5X,F10.7,5X,F10.7,5X,E12.6,7X,I1)
```

```
  640 FORMAT(2X,'**************************************************
     /******************')
      KF=KF+1
      IF(AK-F0)115,115,100
  115 IF(IERR-0)113,222,113
  222 IF(DELTA-OMEGA)1050,1050,114
  114 DELTA=0.5*DELTA
      DO 412 I=1,N
  412 X(I)=X(I)-DELTA*VJ(I)
      GO TO 413
  113 F2=AK
      UK=1.
      GO TO 110
  100 F2=F0
      F0=AK
      DELTA=-DELTA
      DO 420 I=1,N
      X(I)=RMINX(I)+DELTA*VJ(I)
  420 CONTINUE
  417 B(1)=X(1)
      B(2)=X(2)
      A(1)=X(3)
      CALL PRENOS(L,A,B,U,V,E,F,P,Q,MA,MB,MU,MV,ME,MF,MP,MQ,K)
      CALL KRATIT(P,Q,MP,MQ,KRATP,KRATQ,KP,KQ)
      CALL UPRAVP(KP,KQ,KRATP,KRATQ,PP,QP,MPP)
      CALL KRITER(QP,PP,MPP,BK,IERR)
      WRITE(6,600)KF,B(1),B(2),A(1),BK,IERR
      KF=KF+1
      IF(BK-F2)280,280,200
  280 IF(IERR-0)416,415,416
  415 DELTA=0.5*DELTA
      DO 414 I=1,N
  414 X(I)=X(I)-DELTA*VJ(I)
      GO TO 417
  416 F2=BK
      UK=1.
      F0=F2
      GO TO 110
  110 DO 430 I=1,N
      RMINX(I)=X(I)
  430 CONTINUE
      DELTA=2.0*DELTA
      DO 440 I=1,N
      X(I)=RMINX(I)+DELTA*VJ(I)
  440 CONTINUE
  422 B(1)=X(1)
      B(2)=X(2)
      A(1)=X(3)
      CALL PRENOS(L,A,B,U,V,E,F,P,Q,MA,MB,MU,MV,ME,MF,MP,MQ,K)
      CALL KRATIT(P,Q,MP,MQ,KRATP,KRATQ,KP,KQ)
      CALL UPRAVP(KP,KQ,KRATP,KRATQ,PP,QP,MPP)
      CALL KRITER(QP,PP,MPP,C,IERR)
      DO 2437 I=1,N
      IF(ABS(X(I)).GT.10.)IERR=0
 2437 CONTINUE
      WRITE(6,600)KF,B(1),B(2),A(1),C,IERR
      KF=KF+1
      IF(C-F2)240,230,230
```

```
24Ø IF(IERR-Ø)423,418,423
418 DELTA=Ø.5*DELTA
    DO 419 I=1,N
419 X(I)=X(I)-DELTA*VJ(I)
    GO TO 422
423 FØ=F2
    UK=UK+1.
    F2=D
    GO TO 11Ø
23Ø F1=D
    DO 45Ø I=1,N
    X(I)=X(I)-Ø.5*DELTA*VJ(I)
45Ø CONTINUE
426 B(1)=X(1)
    B(2)=X(2)
    A(1)=X(3)
    CALL PRENOS(L,A,B,U,V,E,F,P,Q,MA,MB,MU,MV,ME,MF,MP,MQ,K)
    CALL KRATIT(P,Q,MP,MQ,KRATP,KRATQ,KP,KQ)
    CALL UPRAVP(KP,KQ,KRATP,KRATQ,PP,QP,MPP)
    CALL KRITER(QP,PP,MPP,D,IERR)
    WRITE(6,6ØØ)KF,B(1),B(2),A(1),D,IERR
    KF=KF+1
    IF(D-F2)32Ø,32Ø,31Ø
2ØØ DELTA=2.Ø*DELTA
    F1=EK
    GO TO 35Ø
31Ø F1=D
    GO TO 35Ø
32Ø IF(IERR-Ø)427,424,427
424 DELTA=Ø.5*DELTA
    DO 425 I=1,N
425 X(I)=X(I)-DELTA*VJ(I)
    GO TO 426
427 FØ=F2
    F2=D
    UK=UK+1.
    DO 46Ø I=1,N
    RMINX(I)=X(I)
46Ø CONTINUE
    DJ=DJ+Ø.5*DELTA
35Ø POM=4.Ø*(2.Ø*F2-FØ-F1)
    ODHAD=Ø.
    IF(POM.NE.Ø.)ODHAD=DELTA*(FØ-F1)/POM
    DO 47Ø I=1,N
    X(I)=RMINX(I)-ODHAD*VJ(I)
47Ø CONTINUE
    B(1)=X(1)
    B(2)=X(2)
    A(1)=X(3)
    CALL PRENOS(L,A,B,U,V,E,F,P,Q,MA,MB,MU,MV,ME,MF,MP,MQ,K)
    CALL KRATIT(P,Q,MP,MQ,KRATP,KRATQ,KP,KQ)
    CALL UPRAVP(KP,KQ,KRATP,KRATQ,PP,QP,MPP)
    CALL KRITER(QP,PP,MPP,EK,IERR)
    WRITE(6,6ØØ)KF,B(1),B(2),A(1),EK,IERR
    KF=KF+1
    IF(EK-F2)48Ø,48Ø,36Ø
36Ø FMINX=F2
    WRITE(6,631)
```

```
      WRITE(6,611)RMINX(1),RMINX(2),RMINX(3),FMINX
  631 FORMAT(///' MINIMUM V DANEM SMERU')
  611 FORMAT(8X,F10.7,5X,F10.7,5X,F10.7,5X,E12.6///)
      GO TO 1111
  480 DO 490 I=1,N
      RMINX(I)=X(I)
  490 CONTINUE
      FMINX=EK
   19 WRITE(6,631)
      WRITE(6,611)RMINX(1),RMINX(2),RMINX(3),FMINX
 1111 GO TO(920,940,930),IN
  920 UK1=UK
      DELTA1=DELTA
      DO 921 I=1,N
  921 V1(I)=VJ(I)
      DO 907 I=1,N
  907 VJ(I)=VK(I)
      GO TO 1000
  940 UK2=UK
      DELTA2=DELTA
      DO 941 I=1,N
  941 V2(I)=VJ(I)
      DO 908 I=1,N
  908 VJ(I)=VL(I)
      GO TO 1000
  930 UK3=UK
      DELTA3=DELTA
      KC=KC+1
      BOD(KC)=FMINX
      ROZDIL=ABS(BOD(KC)-BOD(KC-1))
      IF(ROZDIL-THETA)1050,1050,932
  932 DO 931 I=1,N
  931 V3(I)=VJ(I)
      DO 900 I=1,N
  900 A1(I)=UK1*(SIGN(1.,DELTA1))*V1(I)+UK2*(SIGN(1.,DELTA2))*V2(I)+UK3
     /(SIGN(1.,DELTA3))*V3(I)
      DO 901 I=1,N
  901 A2(I)=UK2*(SIGN(1.,DELTA2))*V2(I)+UK3*(SIGN(1.,DELTA3))*V3(I)
      DO 902 I=1,N
  902 A3(I)=UK3*(SIGN(1.,DELTA3))*V3(I)
      S=0
      DO 903 I=1,N
  903 S=S+(A1(I)**2)
      S=SQRT(S)
      IF(S.EQ.0.)S=1.
      DO 909 I=1,N
  909 S1(I)=A1(I)/S
      PS=0.
      DO 906 I=1,N
  906 PS=PS+(A2(I)*S1(I))
      DO 977 I=1,N
  977 B2(I)=A2(I)-PS*S1(I)
      S=0
      DO 904 I=1,N
  904 S=S+(B2(I)**2)
      S=SQRT(S)
      IF(S.EQ.0.)S=1.
      DO 905 I=1,N
```

```
  905 S2(I)=B2(I)/S
      DO 24 I=1,N
 24   CONTINUE
   22 PR=0.
      PS=0.
      DO 910 I=1,N
  910 PS=PS+(A3(I)*S1(I))
      PR=PR+(A3(I)*S2(I))
      DO 911 I=1,N
  911 B3(I)=A3(I)-(PS*S1(I)+PR*S2(I))
      S=0.
      DO 912 I=1,N
  912 S=S+(B3(I)**2)
      S=SQRT(S)
      IF(S.EQ.0.)S=1.
      DO 914 I=1,N
  914 S3(I)=B3(I)/S
   21 WRITE(6,716)(S1(I),I=1,N),(S2(I),I=1,N),(S3(I),I=1,N)
  716 FORMAT(///' VYPOCITANE NOVE SMERY HLEDANI'/
     //5H S1= ,3F7.3,5H S2= ,3F7.3,5H S3= ,3F7.3///)
      DO 913 I=1,N
      VJ(I)=S1(I)
      VK(I)=S2(I)
  913 VL(I)=S3(I)
      GO TO 1001
 1050 STOP
      END
```

```
      SUBROUTINE PRENOS(L,A,B,U,V,E,F,P,Q,MA,MB,MU,MV,ME,MF,MP,MQ,K)
C     ****************************************************************
C PROGRAM PRO VYPOCET PRENOSU UZAVRENEHO REGULACNIHO OBVODU
C
C L=1  -PRENOS OD PORUCHY W
C L=2  -PRENOS OD PORUCHY D1
C L=3  -PRENOS OD PORUCHY D2
C K=1  -PORUCHA TYPU DURACUV PULZ
C K=2  -PORUCHA TYPU JEDNOTKOVY SKOK
C B(Z)/A(Z)-PRENOS REGULATORU
C U(Z)/V(Z)-PRENOS HS(Z)
C E(Z)/F(Z)-PRENOS PORUCHY D2
C P(Z)/Q(Z)-VYSLEDNY PRENOS OBVODU
C MA,MB,MU,MV,MP,MQ-DELKY POLYNOMU
      DIMENSION A(10),B(10),U(10),V(10),P(10),Q(10),Z(10),P1(10),P2(10)
      DIMENSION U1(20),BU1(20),AV(20),UA(20),QT(20),E(20),F(20)
      DIMENSION Q2(20)
      MU1=MU+1
      U1(1)=0.
      DO 115 I=2,MU1
  115 U1(I)=U(I-1)
      CALL NASOB(B,U1,BU1,MB,MU1,MBU1)
      CALL NASOB(A,V,AV,MA,MV,MAV)
      GO TO(116,117,118),L
  116 DO 140 I=1,MAV
  140 P(I)=AV(I)
      CALL SOUCET(AV,BU1,Q,MAV,MBU1,MQ)
      GOTO 120
  118 DO 141 I=1,MAV
  141 P2(I)=-AV(I)
  142 MP2=MAV
      CALL SOUCET(AV,BU1,Q2,MP2,MBU1,MQ2)
      CALL NASOB(P2,E,P,MP2,ME,MP)
      CALL NASOB(Q2,F,Q,MQ2,MF,MQ)
      GO TO 120
  117 CALL NASOB(U,A,UA,MU,MA,MUA)
      DO 119 I=1,MUA
  119 P(I)=-UA(I)
      MP=MUA
      CALL SOUCET(AV,BU1,Q,MAV,MBU1,MQ)
  120 GO TO(122,121),K
  121 MP=MP+1
      P1(1)=0.
      DO 123 I=2,MP
  123 P1(I)=P(I-1)
      DO 125 I=1,MP
  125 P(I)=P1(I)
      Z(1)=-1.
      Z(2)=1.
      MZ=2
      CALL NASOB(Q,Z,QT,MQ,MZ,MQT)
      MQ=MQT
      DO 124 I=1,MQ
  124 Q(I)=QT(I)
  122 RETURN
      END
```

```
      SUBROUTINE SOUCET(A,B,C,MA,MB,MC)
C     *******************************
C PROGRAM PRO SOUCET DVOU POLYNOMU A+B=C
C
C A,B,C,-VEKTORY KOEFICIENTU POLYNOMU
C MA,MB,MC-DELKY POLYNOMU
      DIMENSION A(10),B(10),C(10)
      IF(MA-MB)106,106,107
  106 MC=MB
      GO TO 108
  107 MC=MA
  108 DO 113 I=1,MC
      IF(I-MA)109,109,110
  110 A(I)=0.
  109 IF(I-MB)111,111,112
  112 B(I)=0.
  111 C(I)=A(I)+B(I)
  113 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE NASOB(A,B,C,MA,MB,MC)
C     ******************************
C PROGRAM PRO NÁSOBENI POLYNOMU POLYNOMEM A*B=C
C
C A,B,C,-VEKTORY KOEFICIENTU POLYNOMU
C MA,MB,MC-DELKY POLYNOMU
      DIMENSION A(10),B(10),C(10)
      MC=MA+MB-1
      DO 105 I=1,MC
      J=1
      K=I
      C(I)=0.
  104 IF(K-MA)100,100,101
  100 IF(J-MB)102,102,101
  102 C(I)=A(K)*B(J)+C(I)
  101 K=K-1
      J=J+1
      IF(K-1)105,104,104
  105 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE KRATIT(P,Q,MP,MQ,KRATP,KRATQ,KP,KQ)
C     ************************************************
C PROGRAM PRO KRACENI POLYNOMU PRENOSU
C
C P/Q-POLYNOMY KOEFICIENTU PRENOSU OBVODU
C MP,MQ-DELKY POLYNOMU P A Q
C KRATP/KRATQ-VYSTUPNI POLYNOMY
C KP,KQ-DELKY POLYNOMU KRATP A KRATQ
      REAL KRATP,KRATQ
      DIMENSION P(15),Q(15),KRATP(15),KRATQ(15)
      NUL=0
      DO 10 I=1,MQ
      IF(Q(I)-0.)14,17,14
   17 IF(P(I)-0.)14,11,14
   11 NUL=NUL+1
   10 CONTINUE
   14 IF(NUL-0)15,15,16
   16 KQ=MQ-NUL
      DO 12 K=1,KQ
   12 KRATQ(K)=Q(K+NUL)
      KP=MP-NUL
      DO 13 K=1,KP
   13 KRATP(K)=P(K+NUL)
      GO TO 30
   15 KP=MP
      KQ=MQ
      DO 18 K=1,KP
   18 KRATP(K)=P(K)
      DO 19 K=1,KQ
   19 KRATQ(K)=Q(K)
      WRITE(6,50)
   50 FORMAT(' NELZE KRATIT')
   30 RETURN
      END
```

```
      SUBROUTINE UPRAVP(MP,MQ,P,Q,PP,QP,MPP)
C     ***********************************
C PROGRAM PRO UPRAVU KOEFICIENTU POLYNOMU P A Q
C
C P,Q-KOEFICIENTY VSTUPNICH POLYNOMU
C MP,MQ-DELKY POLYNOMU
C PP,QP-KOEFICIENTY VYSTUPNICH POLYNOMU
      DIMENSION P(10),Q(10),PP(10),QP(10)
      J=1
      K=MP
   10 PP(J)=P(K)
      J=J+1
      K=K-1
      IF(J-MP)10,10,11
   11 J=1
      K=MQ
   12 QP(J)=Q(K)
      J=J+1
      K=K-1
      IF(J-MQ)12,12,13
   13 MP=MP-1
      MQ=MQ-1
      IF(MP-MQ)3,6,2
    3 L=MP+2
      DO 22 I=L,MQ
      PP(I)=0.
   22 MPP=MQ
      GO TO 5
    2 LL=MQ+2
      DO 33 I=LL,MP
   33 QP(I)=0.
    6 MPP=MP
    5 RETURN
      END
```

```
      SUBROUTINE KRITER(A,B,N,VK,IERR)
C     ******************************
C PROGRAM PRO VYCISLENI INTEGRALU RACIONALNI FUNKCE
C     1/(2*PI*I)*B(Z)*B(1/Z)/(A(Z)*A(1/Z))*(1/Z)
C PODEL JEDNOTKOVE KRUZNICE
C
C A-VEKTOR KOEFICIENTU POLYNOMU
C     A(1)*Z**N+A(2)*Z**(N-1)+... +A(N+1)
C B-VEKTOR KOEFICIENTU POLYNOMU
C     B(1)*Z**N+B(2)*Z**(N-1)+... +B(N+1)
C N-STUPEN POLYNOMU A A B (MAX 10)
C IERR-STABILNI IERR=1 A MA VSECHNY KORENY UVNITR JEDNOTKOVE KRUZNICE
C       NESTABILNI IERR=0 A MA NEJAKY KOREN VNE NEBO NA HRANICI JEDNOTKOV
C                           KRUZNICE NEBO A(1) NENI KLADNE
      DIMENSION A(10),B(10),AS(11)
      A0=A(1)
      IERR=1
      VK=0.0
      DO 10 K=1,N
      L=N+1-K
      L1=L+1
      ALFA=A(L1)/A(1)
      BETA=B(L1)/A(1)
      VK=VK+BETA*B(L1)
      DO 20 I=1,L
      M=L+2-I
      AS(I)=A(I)-ALFA*A(M)
   20 B(I)=B(I)-BETA*A(M)
      IF(AS(1))50,50,30
   30 DO 40 I=1,L
   40 A(I)=AS(I)
   10 CONTINUE
      VK=VK+B(1)**2/A(1)
      VK=VK/A0
      RETURN
   50 IERR=0
      RETURN
      END
```

PŘÍLOHA č.5

```
 JOB   #1107L,KANTOROVA
C FORTGTC,MODULE=DKAN2222,REQU=REP
N      DD DCB=(BLKSIZE=00400,BUFNO=002),
YS82116.T113241.IV000.DKAN2222.S0000015,
(D001),
SDA,01),DISP=(OLD,DELETE),SPACE=(80,000166)

 SKIP TO CHANNEL 01   * * * * * * * * * * * * * * * * * * * * * * * * * * *

C IEBUPDTE 63-09.78                      DATE 26.04.82            TIME 12.3

N                               NEW MASTER

 ADD     NAME=DKAN2222
 NUMBER NEW1=100,INCR=100
BER NAME (DKAN2222) NOT FOUND IN NM DIRECTORY.  STOWED WITH TTR,
HEST CONDITION CODE WAS 00000000
 OF JOB IEBUPDTE.

 SKIP TO CHANNEL 01   * * * * * * * * * * * * * * * * * * * * * * * * * * *

TEP WAS EXECUTED - COND CODE 0000
P /TEXT     / START 82116.1231
P /TEXT     / STOP  82116.1232 CPU   0MIN 02.72SEC MAIN  38K LCS   0K

*****************************************************************************
        S  T  E  P   PROGRAM                                  S T E P S T A R T
AME     NAME     NR.    NAME     PROGRAMMER  NAME               DATE        TIME
2222   TEXT      001  SVS12LSY  KANTOROVA                    26.04.82   12/31/13

UNT   PARTITION  CORE    TIMES IN SECONDS    NR.OF INPUT       L I S T  OF  I
ER     NUMBER    USED     STEP       CPU     STREAM CARDS   DISK  TAPE  READ
7L      P01       38K     73.70      2.72        166          9     1
*****************************************************************************

 SKIP TO CHANNEL 01   * * * * * * * * * * * * * * * * * * * * * * * * * * *

V G LEVEL  21                   MAIN                  DATE = 82116          12/3

    C PROGRAM: ROSENBROKOVA METODA ROZSIRENA SWANNEM
    C ****************************************
    C  RMINX-VYCHZI BOD
    C  FMINX-FUNKCNI HODNOTA VYCHOZIHO BODU
    C  VJ-SMER HLEDANI
    C  EPSIL-DELKA STARTOVNIHO KROKU
    C  KF-POCET KROKU
    C  N-POCET SLOZEK VEKTORU X
    C  UK-POCET USPESNYCH KROKU
    C  IN-POCET PROVERENYCH SMERU
    C  K-CISLO CYKLU
    C
          DIMENSION X(100),RMINX(50),VJ(100),VK(100),BOD(50)
          DIMENSION V1(30),V2(30),A1(30),A2(30),S1(20),S2(20),B2(20)
          PEPA(U,V)=(ABS(0.5*U-1.0))**2+(ABS(V-1.0))**2
          READ(5,500)EPSIL,FMINX,N
      500 FORMAT(F6.3,F7.3,I1)
          READ(5,510)(RMINX(I),I=1,N),(VJ(I),I=1,N),(VK(I),I=1,N)
```

```
1000 IN=IN+1
     KF=1
     DJ=0.0
     DELTA=EPSIL
     F0=FMINX
     DO 410 I=1,N
     X(I)=RMINX(I)+DELTA*VJ(I)
 410 CONTINUE
     X1=X(1)
     X2=X(2)
     A=PEPA(X1,X2)
     WRITE(6,620)
 620 FORMAT(55H                  X(1)       X(2)       F(X)       DELTA      K
     WRITE(6,600)X1,X2,A,DELTA,KF
 600 FORMAT(12X,F5.2,5X,F5.2,3X,F7.3,5X,F6.3,6X,I2)
     KF=KF+1
     IF(A-F0)111,111,100
 111 F2=A
     UK=1.
     GO TO 110
 100 F2=F0
     F0=A
     DELTA=-DELTA
     DO 420 I=1,N
     X(I)=RMINX(I)+DELTA*VJ(I)
 420 CONTINUE
     X1=X(1)
     X2=X(2)
     B=PEPA(X1,X2)
     WRITE(6,600)X1,X2,B,DELTA,KF
     KF=KF+1
     IF(B-F2)280,280,200
 280 F2=B
     UK=1.
     F0=F2
     GO TO 110
SKIP TO CHANNEL 01     * * * * * * * * * * * * * * * * * * * * * * * * * * *
G LEVEL  21                  MAIN                    DATE = 82116          12/33/
 110 DO 430 I=1,N
     RMINX(I)=X(I)
 430 CONTINUE
     DJ=DJ+DELTA
     DELTA=2.0*DELTA
     DO 440 I=1,N
     X(I)=RMINX(I)+DELTA*VJ(I)
 440 CONTINUE
     X1=X(1)
     X2=X(2)
     C=PEPA(X1,X2)
     WRITE(6,600)X1,X2,C,DELTA,KF
     KF=KF+1
     IF(C-F2)240,240,230
 240 F0=F2
     UK=UK+1.
     F2=C
     GO TO 110
 230 F1=C
     DO 450 I=1,N
     X(I)=X(I)-0.5*DELTA*VJ(I)
 450 CONTINUE
     X1=X(1)
     X2=X(2)
```

```
      D=PEPA(X1,X2)
      WRITE(6,600)X1,X2,D,DELTA,KF
      KF=KF+1
      IF(D-F2)320,320,310
  200 DELTA=2.0*DELTA
      F1=R
      GO TO 350
  310 F1=D
      GO TO 350
  320 F0=F2
      F2=D
      UK=UK+1.
      DO 460 I=1,N
      RMINX(I)=X(I)
  460 CONTINUE
      DJ=DJ+0.5*DELTA
  350 ODHAD=DELTA*((F0-F1)/(4.0*(2.0*F2-F0-F1)))
      DO 470 I=1,N
      X(I)=RMINX(I)-ODHAD*VJ(I)
  470 CONTINUE
      X1=X(1)
      X2=X(2)
      E=PEPA(X1,X2)
      WRITE(6,600)X1,X2,E,DELTA,KF
      KF=KF+1
      IF(E-F2)480,480,360
  360 FMINX=F2
      WRITE(6,630)
  630 FORMAT(35H              MINX(1)   MINX(2)   FMINX)
      WRITE(6,610)RMINX(1),RMINX(2),FMINX
  610 FORMAT(10X,F5.2,5X,F5.2,5X,F7.3)
      GO TO 1111
  480 DO 490 I=1,N
      RMINX(I)=X(I)
```

SKIP TO CHANNEL 01 *

G LEVEL 21 MAIN DATE = 82116 12/33/

```
  490 CONTINUE
      DJ=DJ+ODHAD
      FMINX=E
      WRITE(6,630)
      WRITE(6,610)RMINX(1),RMINX(2),FMINX
 1111 GO TO(920,930),IN
  920 UK1=UK
      DELTA1=DELTA
      DO 921 I=1,N
  921 V1(I)=VJ(I)
      DO 907 I=1,N
  907 VJ(I)=VK(I)
      GO TO 1000
  930 UK2=UK
      DELTA2=DELTA
      K=K+1
      BOD(K)=FMINX
      ROZDIL=ABS(BOD(K)-BOD(K-1))
      IF(ROZDIL-0.001)1050,1050,932
  932 DO 931 I=1,N
  931 V2(I)=VJ(I)
      DO 900 I=1,N
  900 A1(I)=UK1*(SIGN(1.,DELTA1))*V1(I)+UK2*(SIGN(1.,DELTA2))*V2(I)
      DO 901 I=1,N
  901 A2(I)=UK2*(SIGN(1.,DELTA2))*V2(I)
      S=0
```

```
      DO 903 I=1,N
  903 S=S+(A1(I)**2)
      S=SQRT(S)
      DO 902 I=1,N
  902 S1(I)=A1(I)/S
      P=0
      DO 906 I=1,N
  906 P=P+(A2(I)*S1(I))
      DO 977 I=1,N
  977 B2(I)=A2(I)-P*S1(I)
      S=0
      DO 904 I=1,N
  904 S=S+(B2(I)**2)
      S=SQRT(S)
      DO 905 I=1,N
  905 S2(I)=B2(I)/S
      WRITE(6,716)(S1(I),I=1,N),(S2(I),I=1,N)
  716 FORMAT(5H S1= ,2F7.3,5H S2= ,2F7.3)
      DO 913 I=1,N
      VJ(I)=S1(I)
  913 VK(I)=S2(I)
      GO TO 1001
 1050 STOP
      END

SKIP TO CHANNEL 01     * * * * * * * * * * * * * * * * * * * * * * * * * * *

 G LEVEL  21                  MAIN               DATE = 82116           12/33/

                     SUBPROGRAMS CALLED
OCATION        SYMBOL    LOCATION        SYMBOL    LOCATION        SYMBOL    LOC
  198          SQRT         19C

                     SCALAR MAP
OCATION        SYMBOL    LOCATION        SYMBOL    LOCATION        SYMBOL    LOC
  1F4          FMINX        1F8          N            1FC          I
  208          KF           20C          DJ           210          DELTA
  21C          X2           220          A            224          F2
  230          C            234          F1           238          D
  244          UK1          248          DELTA1       24C          UK2
  258          S            25C          P            260

                     ARRAY MAP
OCATION        SYMBOL    LOCATION        SYMBOL    LOCATION        SYMBOL    LOC
  264          RMINX        3F4          VJ           4BC          VK
  8A4          V2           910          A1           994          A2
  AD4          B2           B24

                     FORMAT STATEMENT MAP
OCATION        SYMBOL    LOCATION        SYMBOL    LOCATION        SYMBOL    LOC
  B74            510        B7E            620        B8A            600
  C06            716        C17

 IN EFFECT*  ID,EBCDIC,SOURCE,NOLIST,NODECK,LOAD,MAP
 IN EFFECT*  NAME = MAIN    , LINECNT =        60
ICS*    SOURCE STATEMENTS =        154,PROGRAM SIZE =      6076
ICS*  NO DIAGNOSTICS GENERATED

SKIP TO CHANNEL 01     * * * * * * * * * * * * * * * * * * * * * * * * * * *
```