

Vysoká škola strojní a textilní v Liberci
nositelka Řádu práce

Fakulta textilní

Obor 31-20-8

Katedra technické kybernetiky

GRADIENTOVÉ METODY NÁVRHU ALGORITMU ŘÍZENÍ

Alena Nečvěřová

Vešoucí práce: Prof. Ing. Bořivoj Hanuš, CSc
/ KTK VŠST Liberec /

KTK ASŘ TF - 031

Rozsah práce a příloh

Počes stran 73

Počes příloh

a tabulek 8+23

Počes obrázků 22

Počes výkresů 0

Počes modelů

nebo jiných příloh ... 0

4.6. 1982

Vysoká škola: VŠST Liberec
Katedra: technické kybernetiky

Fakulta: textilní
Školní rok: 1981/82

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

pro s. Alena Nedvědová
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: Gradientové metody návrhu algoritmu řízení

Zásady pro vypracování:

- 1) Prostudujte potřebné metody z doporučené literatury.
- 2) Vypracujte program pro výpočet algoritmu pro vyrovnání determinované poruchy na soustavě prvního až pátého řádu.
- 3) Ověřte vypočítané algoritmy simulací regulačních pochodů na číslicovém event. hybridním počítači.
- 4) Proveďte srovnání výpočtu algoritmu metodou gradientovou a jiným přímým způsobem z hlediska náročnosti na počítač, přesnost výsledku, dobu výpočtu.

VYSOKÁ ŠKOLA STROJNÍ A TEXTILNÍ
Ústřední knihovna
LIBEREC 1, STUDENTSKÁ 5
PSČ 461 17

Autorské právo se řídí směrnicemi
MŠK pro státní záv. zkoušky č.j. 31
727/62-III/2 ze dne 13. července
1962-Věstník MŠK XVIII, sešit 24 ze
dne 31.8.1962 a 19 aut. z č. 115/53 Sb.

✓ 16 / 82 T
+ PŘÍLOHA

Rozsah grafických prací:

Rozsah průvodní zprávy: 40 - 50 stran

Seznam odborné literatury:

- /1/ Hanuš, B.: Gradientové metody v automatizaci. /Výzkumná zpráva KTK 0078./ Liberec, VŠST 1981.
- /2/ Hanuš, B.: Optimalizace systému řízení. /Skripta./ Liberec, VŠST 1978.
- /3/ Hanuš, B. - Balda, M. a kolektiv: Základy technické kybernetiky. /Skripta./ Liberec, VŠST 1979.
- /4/ Hanuš, B. - Janeček, B. - Kracík, Vl. - Mikulecký, Bl.: Metoda návrhu víceparametrových číslicových regulačních systémů. /Metodická příručka./ Praha, ÚVVTR 1981.

Vedoucí diplomové práce:

Prof. Ing. Beřivoj Hanuš, DrSc.

Konzultant:

Ing. Bedřich Janeček

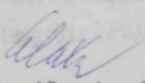
Datum zadání diplomové práce:

15.9.1981

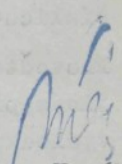
Termín odevzdání diplomové práce:

4.6.1982




Doc. Ing. Ján Alaxin, CSc.

Vedoucí katedry


Doc. Ing. Jáchym Novák, CSc.

Děkan

v Liberci 11.9. 81
dne 10

Prohlášení

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

Olena Nedžadová

V Liberci, dne 4.6. 1982

OBSAH	strana
Úvod	1
1. Základní pojmy	3
2. Gradientové metody v automatizaci	5
2.1. Kaczmarzův algoritmus	5
2.2. Výpočet řídicího algoritmu	8
3. Praktické použití Kaczmarzova gradientního algoritmu	13
3.1. Výpočet akčních zásahů	17
3.2. Výpočet koeficientů algoritmu řízení	33
4. Ověření vypočtených algoritmů simulací regulačních pochodů	45
5. Urychlení konvergence gradientního Kaczmarzova algoritmu	56
5.1. Algoritmus ortogonalizace vstupních vektorů	58
5.2. Zobecněný Kaczmarzův algoritmus	59
5.3. Algoritmus relaxace	60
5.4. Optimální volba koeficientu konvergence	60
5.5. Doplnění Kaczmarzova algoritmu s využitím hledání ve směru	61
5.6. Využití konvergence geometrické řady	63
5.7. Volba počátečního odhadu	65

VŠST LIBEREC Fakulta textilní	Gradientové metody návrhu algoritmu řízení	Strana
----------------------------------	---	--------

6.	Porovnání metod	68
6.1.	Náročnost na paměť počítače	68
6.2.	Náročnost na programování	68
6.3.	Doba výpočtu	69
6.4.	Přesnost	70
6.5.	Použití	70
Závěr		71
Literatura		72

POUŽITÁ OZNAČENÍ

$\underline{X}$	vektor X
$\underline{A}$	matice A
$\nabla F(\underline{x})$	gradient funkce $F(\underline{x})$
$\nabla^2 F(\underline{x}), \underline{H}(\underline{x})$	Hessova čtvercová matice druhých parciálních derivací $F(\underline{x})$
$\Delta \underline{u}$	vektor akčních zásahů
$\Delta \hat{\underline{u}}$	odhad vektoru akčních zásahů
y	vektor optimálních výstupů ze soustavy
n	řád soustavy
d	skoková porucha na vstupu do soustavy
w	skoková změna na řízení

OZNAČENÍ V TABULKÁCH

ŘS	řád soustavy
PP	počet podmínkových rovnic
ČP	číslo podmínkové rovnice
PI	počet iterací
ČI	číslo iterace
LS	levá strana
$\delta(\Delta u_i)$	přírůstek i -tého akčního zásahu

ÚVOD

Základem hospodářského a sociálního programu KSČ je pro období let 1981 - 1985 generální linie výstavby rozvinuté socialistické společnosti.

Splnění záměrů sedmého pětiletého plánu vyžaduje v mnohém urychlení a maximální využití výsledků vědeckotechnického rozvoje jako podmínky ke zvyšování efektivnosti společenské výroby, růstu intenzifikace ekonomiky, zvýšení efektivnosti a kvality veškeré práce.

Jedním z účelných prostředků realizace stanovených úloh je uplatňování vyššího stupně automatizace výrobních procesů při současném využívání automatizovaných systémů řízení technologických procesů v průmyslu, dopravě, stavebnictví, spojích a v zemědělství.

Také v oblasti textilních strojů je nutno vytvářet podmínky pro zavádění automatizovaných systémů řízení, neboť zabezpečení optimálního řízení procesů pomocí výpočetní techniky umožní zvýšit produktivitu zařízení, zlepšit kvalitu výroby snižováním ztrát surovin a materiálů.

Složitost problémů ASŘ technologických procesů vyplývá už z obecného chápání pojmu řízení. Řízení lze definovat jako funkční vztah mezi řídícím subjektem a řízeným objektem. Řídící subjekt působí na řízený objekt a usměrňuje jeho chování. Chceme-li účelně působit na jakýkoliv objekt, musíme znát nejen cíle řízení, ale i samotný objekt. Musíme tedy stanovit logické a kvantitativní závislosti mezi stavovými proměnnými objektů a vnějšími vlivy, tj. musíme provést identifikaci objektu. Informace získaná o stavu objektu a

přenesená do řídicího systému musí být transformována v sou-
ladu s dynamickými vlastnostmi řízeného objektu a předepsa-
ným kriteriem efektivnosti. Tuto transformaci provádí
algoritmus řízení. Algoritmy řízení lze navrhovat různými
způsoby.

Kaczmarz ve své práci / 9 / uvádí gradientní iterační
algoritmus pro výpočet koeficientů v lineárním vztahu.

Úkolem diplomové práce je ověřit vhodnost této metody
pro její použití k návrhu řídicích algoritmů.

Byla sestavena řada programů, sloužících k výpočtu a ově-
ření algoritmů na počítači RPP - 16S.

1. Základní pojmy

Předpokládejme, že máme nalézt bod minima $\underline{x}^*$ funkce $F(\underline{x})$, tedy takový vektor $\underline{x}$, pro který platí

$$F(\underline{x}^*) \leq F(\underline{x})$$

$\underline{x}$ má složky $(x_1, x_2, \dots, x_n)$

Úloha minimalizace se dá jednoduchou úpravou

$$-\max F(\underline{x}) = \min(-F(\underline{x}))$$

převést na hledání maxima. Proto se lze bez újmy na obecnosti zaměřit jen na problém určení minima funkce $F(\underline{x})$.

Minimalizační úlohy definované rovnicí

$$\min F(\underline{x}) \quad \text{pro} \quad \forall \underline{x} \in E^n \quad (1-1)$$

se v praxi vyskytují při hledání extrémů funkcí nebo při řešení soustav rovnic.

V případě řešení soustavy rovnic

$$h_j(\underline{x}) = 0$$

můžeme řešení převést na problém hledání minima funkce

$$F(\underline{x}) = \sum_{j=1}^m |h_j(\underline{x})|$$

Gradient funkce

$$\nabla^T F(\underline{x}) = \left[\frac{\partial F(\underline{x})}{\partial x_1}, \frac{\partial F(\underline{x})}{\partial x_2}, \dots, \frac{\partial F(\underline{x})}{\partial x_n} \right]$$

udává směr, ve kterém funkce $F(\underline{x})$ v daném bodu $\underline{x}$ nejrychleji roste. Gradient je kolmý k tečné nadrovině sestrojené v témže bodu k nadploše konstantní hodnoty funkce $F(\underline{x})$.

Vektor $P(\underline{x})$ obrácený ke gradientu má směr největšího spádu /nejstrmějšího sestupu/.

$$P(\underline{x}) = -\nabla^T F(\underline{x})$$

Gradientové metody se používají pro určování volného extrému funkcí více proměnných. / formulace úlohy- vztah (1-1)/

Nutné a postačující podmínky jsou při hledání ostrého minima

$$\nabla F(\underline{x}) = 0$$

$$\nabla^2 F(\underline{x}) - \text{pozitivně definitní}$$

Pro určení hodnot souřadnic extrému se používá iteračního postupu

$$\underline{x}_{k+1} = \underline{x}_k + \Delta \underline{x}_k = \underline{x}_k + \alpha_k \underline{s}_k^* = \underline{x}_k + \alpha_k^* \underline{g}_k$$

kde jsou

$\underline{x}_k$ hodnota předchozího bodu řešení

$\Delta \underline{x}_k$ přírůstkový vektor pro přechod na následující bod řešení

$\underline{s}_k^*$ jednotkový směrový vektor ve směru $\Delta \underline{x}_k$

$\underline{g}_k$ libovolný vektor ve směru

α_k, α_k^* skaláry

2. Gradientové metody v automatizaci

2.1. Kaczmarzův algoritmus

Kaczmarz ve své práci / 9 / popsal algoritmus pro určení neznámých koeficientů v lineárním vztahu. Řadu úloh automatizace lze převést na problém určení parametrů v lineárním vztahu mezi vstupem a výstupem.

$$y = \underline{a}^T \cdot \underline{u} + e = y^s + e \quad (2.1-1)$$

kde je

y^s skalární výstup ze soustavy odpovídající jejímu přenosu $y^s = \underline{a}^T \cdot \underline{u}$

y skalární výstup soustavy zatížený aditivní náhodnou chybou e

$\underline{u}$ vektor $[n, 1]$ vstupů soustavy

$\underline{a}$ vektor $[n, 1]$ parametrů soustavy

e skalární náhodná chyba s nulovou střední hodnotou $E\{e\}$ a nekorelovaná s $\underline{u}$ a y^s

Pro určení parametrů $\underline{a}$ je k dispozici N vztahů (2.1-1), získaných například měřeními ($N \geq n$)

Střední hodnota kvadrátu rozdílu $y - \underline{a}^T \underline{u}$ je rovna při $N \rightarrow \infty$

$$\begin{aligned} E\{(y - \underline{a}^T \underline{u})^2\} &= E\{(y^s + e - \underline{a}^T \underline{u})^2\} = E\{(y^s - \underline{a}^T \underline{u})^2\} + \\ &+ 2E\{e(y^s - \underline{a}^T \underline{u})\} + E\{e^2\} = E\{e^2\} + E\{(y^s - \underline{a}^T \underline{u})^2\} \end{aligned}$$

Odtud vyplývá, že optimální odhad parametrů $\hat{\underline{a}} = \underline{a}$ soustavy minimalizuje střední hodnotu kvadrátu rozdílu $y - \underline{a}^T \underline{u}$

$$\min E\{(y - \hat{\underline{a}}^T \underline{u})^2\}$$

kde $\hat{\underline{a}}$ je odhad vektoru $[n, 1]$ parametrů soustavy.

Pro konečné N je možno určit optimální odhad $\hat{\underline{a}}$ z podmínky minima funkcionálu J

$$\min J = \min \sum_{i=1}^N (y(i) - \hat{a}^T \underline{w}(i))^2$$

$$\frac{\partial J}{\partial \hat{a}} = -2 \sum_{i=1}^N (y(i) - \hat{a}^T \underline{w}(i)) \underline{w}(i) = \underline{0}$$

$$\hat{a} = \sum_{i=1}^N (\underline{w}(i) \underline{w}(i)^T)^{-1} \cdot \sum y(i) \underline{w}(i) \quad (2.1-2)$$

Výpočet odhadu $\hat{a}$ podle vzorce (2.1-2) není často dost vhodný.

Je potřeba provádět inverzi matice a ukládat do paměti velké množství dat. Vhodnější pro zpracování na počítači jsou metody iterační. Podle Kaczmarze, gradientová metoda minimalizuje postupně účelové funkce

$$J = (y(i) - \underline{a}^T \underline{w}(i))^2 \quad i = 1 \text{ až } N \quad (2.1-3)$$

Výchozí odhad parametrů nechť je $\hat{a}_0$

Pro k -tou iteraci pak platí ($k = i$)

$$\hat{a}(k) = \hat{a}(k-1) - \mu_k \frac{\partial J}{\partial \underline{a}} \Big|_{k-1} = \hat{a}(k-1) + 2\mu_k [y(k) - \hat{a}(k-1)^T \underline{w}(k)] \underline{w}(k) \quad (2.1-4)$$

kde je

μ_k koeficient / zesílení / konvergence

$$\mu_k > 0$$

Protože účelové funkce (2.1-3) je kvadratická, existuje konečný Taylorův rozvoj / 1 /

$$J(\underline{a}(k)) = J(\underline{a}(k-1)) + \frac{\partial J}{\partial \underline{a}} \Delta \underline{a} + \frac{1}{2} \Delta \underline{a}^T \underline{H}(k-1) \Delta \underline{a} \quad (2.1-5)$$

kde je

$\underline{H}(k-1)$ Hessova matice účelové funkce

$$\underline{H}(k-1) = 2 \underline{w}(k) \underline{w}(k)^T$$

Volíme-li přírůstek $\Delta \underline{a}$ ve směru obráceného gradientu

$$\Delta \underline{a} = -\mu_k \frac{\partial J}{\partial \underline{a}} \Big|_{k-1} \quad (2.1-6)$$

po dosazení (2.1-6) do (2.1-5) a po derivaci podle μ_k

získáme minimum funkcionálu J v bodě $\mathcal{N}_k / 2 /$.

$$\mathcal{N}_k = \frac{1}{2\mathbf{w}^T(k)\mathbf{w}(k)} \quad (2.1-7)$$

po dosazení (2.1-7) do (2.1-4) obdržíme iterační vzorce

$$\hat{\mathbf{a}}(k) = \hat{\mathbf{a}}(k-1) + \mathcal{H}_k \frac{\mathbf{y}(k) - \hat{\mathbf{a}}^T(k-1)\mathbf{w}(k)}{\mathbf{w}^T(k)\mathbf{w}(k)} \mathbf{w}(k) \quad (2.1-8)$$

koeficientem $\mathcal{H}_k$ lze kontrolovat rychlost konvergence.

Nulové body účelové funkce (2.1-3) vytvářejí nadrovinu

$$\mathbf{y}(k) - \hat{\mathbf{a}}^T \mathbf{w}(k) = 0 \quad (2.1-9)$$

Volbou $\mathcal{H}$ se určuje další bod iterace $\hat{\mathbf{a}}_k$ na normále spuštěné z $\hat{\mathbf{a}}(k-1)$ na nadrovinu (2.1-9) mezi bodem $\hat{\mathbf{a}}(k-1)$ ($\mathcal{H}_k=0$) a patou normály ($\mathcal{H}_k=1$).

Tím se vlastně s nejmenšími změnami parametrů $\hat{\mathbf{a}}(k-1)$ přiblížíme k minimální hodnotě účelové funkce (2.1-3). Matematicky lze formulovat přechod do paty kolmice jako minimum účelové funkce

$$\min J = \min (\hat{\mathbf{a}}(k) - \hat{\mathbf{a}}(k-1))^T (\hat{\mathbf{a}}(k) - \hat{\mathbf{a}}(k-1))$$

při vedlejší podmínce

$$\mathbf{y}(k) - \hat{\mathbf{a}}^T(k)\mathbf{w}(k) = 0$$

Pomocí Lagrangeových multiplikátorů se převádí problém na minimum funkcionálu

$$\min J = \min \left\{ (\hat{\mathbf{a}}(k) - \hat{\mathbf{a}}(k-1))^T (\hat{\mathbf{a}}(k) - \hat{\mathbf{a}}(k-1)) + p(\mathbf{y}(k) - \hat{\mathbf{a}}^T(k)\mathbf{w}(k)) \right\}$$

kde je

p Lagrangeův multiplikátor

pro minimum platí

$$\frac{\partial J}{\partial \hat{\mathbf{a}}(k)} = 2(\hat{\mathbf{a}}(k) - \hat{\mathbf{a}}(k-1)) - p\mathbf{w}(k) = 0 \quad (2.1-10)$$

$$\mathbf{y}(k) - \hat{\mathbf{a}}^T(k)\mathbf{w}(k) = 0$$

(2.1-11)

Vynásobením rovnice (2.1-10) členem $\underline{w}^T(k)$ a s použitím rovnice (2.1-11) dostáváme

$$p = 2 \frac{\underline{y}(k) - \hat{\underline{a}}^T(k-1) \underline{w}(k)}{\underline{w}^T(k) \underline{w}(k)} \quad (2.1-12)$$

po dosazení tohoto vztahu (2.1-12) zpět do (2.1-10) a s použitím koeficientu konvergence $\mathcal{H}_k$

$$\hat{\underline{a}}(k) = \hat{\underline{a}}(k-1) + \mathcal{H}_k \frac{\underline{y}(k) - \hat{\underline{a}}^T(k-1) \underline{w}(k)}{\underline{w}^T(k) \underline{w}(k)} \underline{w}(k) \quad (2.1-13)$$

2.2. Výpočet řídicího algoritmu

Řídicí algoritmy pro číslicový počítač jsou navrhovány podle nejrozumnějších kritérií. Velmi obecným kritériem jakosti regulace je minimum funkcionálu

$$J = \sum_{i=1}^N (y_i^2 + p^2 \Delta w_{i-1}^2) \rightarrow \min \quad (2.2-1)$$

kde je

y_i hodnota regulované veličiny v i -tém kroku sledovaného regulačního pochodu

Δw_i akční zásah regulátoru v i -tém kroku

$$\Delta w(i) = w(i) - w(i-1)$$

p^2 konstantní váhový koeficient

O regulačním pochodu předpokládáme, že je vyvolán determinovanou poruchou. V našem případě uvažujeme skokovou změnu na vstupu regulované soustavy v čase ($i = -1$) před započítáním nulového kroku regulačního pochodu ($i = 0$).

Určení optimálních akčních
zásahů

Nechť je odezva regulované soustavy na determinovanou poruchu dána posloupností y_0

$$y_0^T = [y_{00}, y_{01}, y_{02}, \dots, y_{0N}] \quad (2.2-2)$$

a odezva na jednotkový akční zásah v čase ($i = 0$) dána posloupností f

$$f^T = [f_1, f_2, \dots, f_N] \quad (2.2-3)$$

Regulační pochod je pak dán posloupností akčních zásahů Δu

$$\Delta u^T = [\Delta u_0, \Delta u_1, \dots, \Delta u_{N-1}] \quad (2.2-4)$$

a regulovaných veličin y

$$y^T = [y_1, y_2, \dots, y_N] \quad (2.2-5)$$

Platí konvolutorní vztah

$$y_i = y_{0i} + \sum_{j=0}^{i-1} f_{i-j} \Delta u_j \quad (2.2-6)$$

Po dosazení (2.2-6) do funkcionálu (2.2-1) je jeho hodnota

$$J = \sum_{i=1}^N \left[\left(y_{0i} + \sum_{j=0}^{i-1} f_{i-j} \Delta u_j \right)^2 + p^2 \Delta u_{i-1}^2 \right] \rightarrow \min \quad (2.2-7)$$

derivováním (2.2-7) podle Δu dostáváme soustavu rovnic pro optimální akční zásahy

$$\frac{\partial J}{\partial \Delta u} = \begin{bmatrix} \sum_{i=1}^N (y_{0i} + \sum_{j=0}^{i-1} f_{i-j} \Delta u_j) f_i + p^2 \Delta u_0 \\ \sum_{i=2}^N (y_{0i} + \sum_{j=0}^{i-1} f_{i-j} \Delta u_j) f_{i-1} + p^2 \Delta u_1 \\ \dots \\ (y_{0N} + \sum_{j=0}^{N-1} f_{N-j} \Delta u_j) f_1 + p^2 \Delta u_{N-1} \end{bmatrix} = \underline{0}$$

Řádky soustavy rovnic (2.2-8) lze upravit na lineární vztah ve smyslu (2.1-1)

$$y = \underline{a}^T \underline{w} = \Delta \underline{w}^T \cdot \underline{w} \quad (2.2-9)$$

kde je

y levá strana

$$y = y(k^*) = - \sum_{i=k^*}^N y_{oi} f_{i+1-k^*} \quad k^* = 1 \text{ až } N \quad (2.2-10)$$

$\underline{a}$ vektor $[N, 1]$ parametrů, což v našem případě představuje neznámý vektor optimálních akčních zásahů $\Delta \underline{w}$

$$\underline{a}^T = \Delta \underline{w}^T = [\Delta w_{01}, \Delta w_{11}, \dots, \Delta w_{N-1}]$$

$\underline{w}$ vektor $[N, 1]$ vstupů

$$\underline{w}^T = \underline{w}^T(k^*) = \left[\left(\sum_{i=k^*}^N f_i f_{i+1-k^*} + p^2 \delta(1, k^*) \right), \left(\sum_{i=k^*}^N f_{i-1} f_{i+1-k^*} + p^2 \delta(2, k^*) \right), \dots, \left(\sum_{i=k^*}^N f_{i-N+1} f_{i+1-k^*} + p^2 \delta(N, k^*) \right) \right] \quad (2.2-11)$$

$$k^* = 1 \text{ až } N$$

$\delta(\ell, k^*)$ Kroneckerovo delta = 1 pro $\ell = k^*$
= 0 pro $\ell \neq k^*$

k^* celé číslo, zbytek po dělení pořadového čísla iterace číslem N

$$k^* = 1 \text{ až } N.$$

Pro odhad vektoru akčních zásahů v k -tém iteračním kroku lze pak podle (2.1-8) psát

$$\Delta \hat{\underline{w}}(k) = \Delta \hat{\underline{w}}(k-1) + \alpha_k \frac{y(k^*) - \Delta \hat{\underline{w}}^T(k-1) \underline{w}(k^*)}{\underline{w}^T(k^*) \underline{w}(k^*)} \underline{w}(k^*) \quad (2.2-12)$$

Za počáteční odhad lze volit například $\Delta \hat{\underline{w}}(0) = \underline{0}$

Při iteraci se podmínky (2.2-8) znovu cyklicky opakují po projití N iteračních kroků

V ý p o č e t k o e f i c i e n t ů a l g o r i t m u ř í z e n í

Řídicí algoritmus předpokládáme ve tvaru / r-rozměrná regula-
ce /

$$\Delta \underline{w}_j = \sum_{i=0}^{n-1} \underline{A}_i^T \underline{y}_{j-i} + \sum_{i=1}^{n-1} \underline{B}_i^T \Delta \underline{w}_{j-i} \quad (2.2-13)$$

kde je

$\Delta \underline{w}_j$ vektor $[n, 1]$ akčních zásahů v j-tém krokovacím inter-
valu optimálního regulačního pochodu, získaný výpoč-
tem pro r různých vstupních poruch

$\underline{y}_{j-i}$ vektor regulovaných veličin v $(j - i)$ -tém kroko-
vacím intervalu optimálního regulačního pochodu

$\underline{A}_i, \underline{B}_i$ matice koeficientů optimálního řídicího algoritmu

Soustavu rovnic (2.2-13) lze rozepsat na podmínky pro každý
akční zásah v i-té řádce samostatně. Tím získáme opět lineár-
ní vztah

$$\underline{y} = \underline{a}^T \underline{w}, \quad \underline{y}(k^*) = \underline{a}^T \underline{w}(k^*) \quad k^* = 1 \text{ až } N$$

kde je

$\underline{y}(k^*)$ velikost optimálního akčního zásahu v k^* -tém kroko-
vacím intervalu regulač. pochodu

$\underline{a}$ vektor všech koeficientů řídicího algoritmu v jedné
řádce

$$\underline{a}^T = {}_l \underline{a} = [{}_l a_1 \dots {}_l a_m]$$

$\underline{w}$ vektor vstupů příslušných krokovacím intervalu k^*

k^* celé číslo, zbytek po dělení pořadového čísla ite-
race k číslem N / místo $k^* = 0$ se však bere $k^* = N$ /

Pro odhad koeficientů algoritmu lze pak psát podle (2.1-8)
vztah

$$\underline{\hat{a}}(k) = \underline{\hat{a}}(k-1) + \mu_k \frac{\Delta u(k^*) - \underline{\hat{a}}^T(k-1) u(k^*)}{\underline{u}^T(k^*) \underline{u}(k^*)} \underline{u}(k^*) \quad (2.2-14)$$

3. Praktické použití Kaczmarzova gradientního algoritmu

Cílem práce je ověřit vhodnost Kaczmarzova algoritmu / 9 /, při výpočtu diskrétních řídicích algoritmů.

K tomuto účelu byl sestaven soubor programů; programy pro výpočet optimálních akčních zásahů pro vyrovnaní determinované poruchy v konečném / minimálním / počtu kroků a pro vyrovnaní determinované poruchy podle kvadratického kritéria, dále programy pro výpočet koeficientů algoritmu řízení.

K ověření vypočtených hodnot akčních zásahů v minimálním počtu kroků byl sestaven program kontrolní, řešící tento výpočet ne iteračně, ale přímo, jak je odvozeno v / 5 /.

K ověření vypočtených hodnot akčních zásahů podle kvadratického kritéria byl použit program DEP / 6 /.

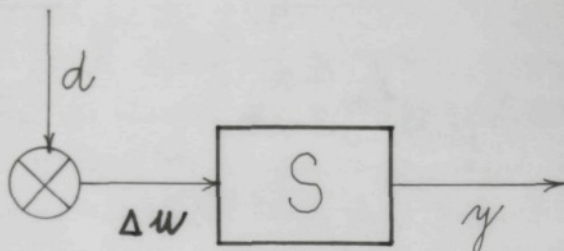
Program na simulaci regulačních pochodů má sloužit k porovnání vypočtených algoritmů; byl sestaven s pomocí / 6 /.

Fáze postupu při práci s Kaczmarzovým algoritmem jsou uvedeny v následujícím schematickém přehledu. Očíslování postupu souhlasí s čísly kapitol, ve kterých je postup zkoumán podrobněji.

Dále jsou v schematu uvedeny názvy jednotlivých programů, které řeší výpočet prakticky. Jednotlivé programy jsou popsány v příloze 1. Výsledky programů jsou uvedeny v přílohách 3-8.

SCHEMATICKÝ PŘEHLED POSTUPU

3.1. Výpočet akčních zásahů



Obr. 3-1

Známe: d ... determinovaná porucha / určena tvarem, místem vstupu /

S ... soustava určena hodnotami přechodové charakteristiky f

Určíme: Δw ... optimální akční zásahy

Způsoby určení Δw :

a/ Algoritmus - minimální počet regulačních kroků

- Kaczmarzův algoritmus výpočtu / kap. 3.1. / , program MING

- přímá metoda výpočtu / 5 / , program MPK

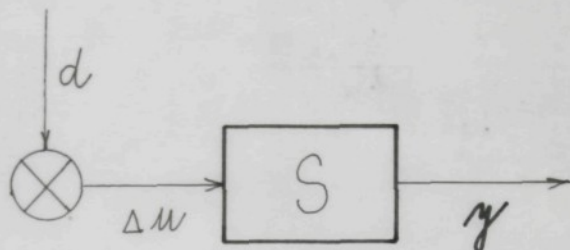
b/ Kvadratické kritérium jakosti regulace

- Kaczmarzův algoritmus výpočtu / kap. 2.2. / , program AKC

- přímá metoda výpočtu / 4 / , program DEP / 6 /

3.2. Výpočet koeficientů algoritmu řízení

a/ Provedení simulace průběhu regulované veličiny



Obr. 3-2

Známe: d ... determinovaná porucha / určena tvarem a místem vstupu /

S ... soustava určena hodnotami přechodové charakteristiky f

Δu ... optimální akční zásahy vypočtené podle /kap. 3.1/

Určíme: y ... optimální průběh regulované veličiny

Způsob určení y :

- pomocí konvolutorního součtu

$$y_i = y_{oi} + \sum_{j=0}^{i-1} f_{i-j} \Delta u_j$$

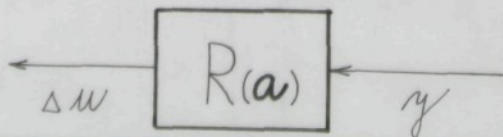
kde je

y_i hodnota regulované veličiny y v i -tém kroku regulačního pochodu

y_{oi} odezva regulované soustavy na determinovanou poruchu v i -tém kroku regulačního pochodu

Tato simulace je provedena v programu RR

b/ Výpočet koeficientů algoritmů řízení



Obr. 3-3

Známe: y ... optimální průběh regulované veličiny
/ viz. 3.2.a /

Δu ... optimální akční zásahy / viz. 3.1.a, b /

$R()$... strukturu regulátoru, tj. algoritmu řízení
/ dána vztahem(2.2-13)/

Určíme: $\mathcal{A}$... koeficienty řídicího algoritmu

Způsob určení $\mathcal{A}$:

a/ Kaczmarzovým algoritmem / kap. 2.2. /, program RR

b/ přímou metodou výpočtu odvozenou v / 4 /, program DEP / 6 /

3.1. Výpočet akčních zásahů

a/ Algoritmus - minimální počet kroků

Při použití tohoto algoritmu se vychází ze znalosti hodnot přechodové charakteristiky / tj. soustava plně popsána /, z konvolutorního součtu.

$$y_i = y_{0i} + \sum_{j=0}^{i-1} f_{i-j} \Delta u_j \quad (3.1-1)$$

kde je

y_i	hodnota regulované veličiny v i-tém kroku regulačního pochodu
Δu_j	akční zásah v j-tém kroku regulačního pochodu
f_{i-j}	hodnota (i - j) - té pořadnice přechodové charakteristiky regulované soustavy při vstupu akční veličiny
y_{0i}	odezva regulované soustavy na determinovanou poruchu v i-tém kroku regulačního pochodu.

Pro vyrovnání determinované poruchy v konečném počtu kroků je třeba $m = n + 1$ akčních zásahů / Obr. 3-4 /

n řád soustavy

Pro hodnotu regulované veličiny v k-tém kroku regulačního pochodu tedy platí

$$y_k = 0 \quad \text{pro } k \geq m+1 \quad (3.1-2)$$

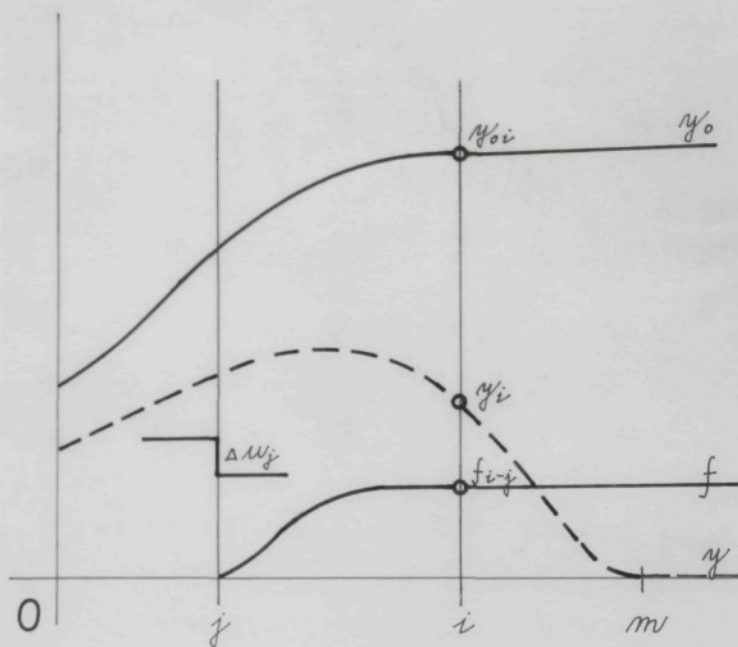
Pro popis algoritmu můžeme použít m rovnic, dostáváme

$$y_m = y_{0m} + f_{m-1} \Delta u_1 + f_{m-2} \Delta u_2 + \dots + f_1 \Delta u_n$$

$$y_{m+1} = y_{0m+1} + f_m \Delta u_1 + f_{m-1} \Delta u_2 + \dots + f_2 \Delta u_n$$

$$y_{2m-1} = y_{02m-1} + f_{2m-2} \Delta u_1 + \dots + f_m \Delta u_n$$

(3.1-3)



Obr.3 -4 : VYROVNÁNÍ V KONEČNÉM POČTU REGULAČNÍCH
KROKŮ

- y_0 odezva soustavy na determinovanou poruchu
- f odezva soustavy na jednotkový akční zásah
- Δu akční zásah
- y výstup ze soustavy
- j, i, m kroky

Pokud je soustava vyregulována, položíme pravou stranu rovnic (3.1-3) rovnu nule, v souladu s (3.1-2).

$$\begin{bmatrix} y_m \\ y_{m+1} \\ \vdots \\ y_{2m-1} \end{bmatrix} = \underline{0}$$

Úpravou rovnic (3.1-3) dostáváme obecnou soustavu podmínkových rovnic

$$\begin{aligned} -y_{0m} &= f_{m-1} \Delta u_1 + f_{m-2} \Delta u_2 + \dots + f_1 \Delta u_n \\ -y_{0m+1} &= f_m \Delta u_1 + f_{m-1} \Delta u_2 + \dots + f_2 \Delta u_n \\ &\vdots \\ -y_{02m-1} &= f_{2m-2} \Delta u_1 + \dots + f_m \Delta u_n \end{aligned} \quad (3.1-4)$$

Tuto soustavu řešíme iteračně Kaczmarzovým algoritmem podle vzorce (2.2-12).

A n a l ý z a v ý s l e d k ů

Byly analyzovány výsledky výpočtů optimálních akčních zásahů vyrovnávajících determinovanou poruchu d ve tvaru jednotkového skoku na vstupu do regulované soustavy / Obr. 3-1 /.

Řešení bylo provedeno pro soustavu 0. - 5. řádu, která má přenos

$$S(p) = \frac{1}{(Tp+1)^n}$$

kde je

n řád soustavy

T časová konstanta / volena ve všech případech $T = 1$ /

Soustava určena hodnotami přechodové charakteristiky. Přesné hodnoty přechodových charakteristik byly zadány z tabulek uvedených v / 5 /.

Počáteční odhad iteračního postupu volen ve všech případech jednotně

$$\Delta w(0) = 0$$

V následující tabulce / Tab. I / je uveden číselný přehled podmínkových rovnic, které mají být pro jednotlivé soustavy splněny. Obecné odvození podmínkových rovnic dáno vztahem (3.1-4).

PODMÍNKOVÉ ROVNICE										
ŘS	PP	ČP	VEKTOR	MATICE PRAVÝCH STRAN						
				LS	ΔW_1	ΔW_2	ΔW_3	ΔW_4	ΔW_5	ΔW_6
0	1	1.	-0.9999996	0.9999996						
1	2	1.	-0.8646645	0.6321203	0.					
		2.	-0.9502125	0.8646645	0.6321203					
2	3	1.	-0.8008499	0.5939895	0.2642398	0.				
		2.	-0.9084194	0.8008499	0.5939895	0.2642398				
		3.	-0.9595688	0.9084194	0.8008499	0.5939895				
3	4	1.	-0.7618959	0.5768005	0.3233233	0.0803014	0.			
		2.	-0.8753479	0.7618959	0.5768005	0.3233233	0.0803014			
		3.	-0.9380310	0.8753479	0.7618959	0.5768005	0.3233233			
		4.	-0.9703636	0.9380310	0.8753479	0.7618959	0.5768005			
5	6	1.	-0.7149429	0.5595064	0.3711628	0.1847366	0.0526530	0.0365984	0.	
		2.	-0.8270084	0.7149429	0.5595064	0.3711628	0.1847366	0.0526530	0.0365984	
		3.	-0.9003674	0.8270084	0.7149429	0.5595064	0.3711628	0.1847366	0.0526530	
		4.	-0.9450363	0.9003674	0.8270084	0.7149429	0.5595064	0.3711628	0.1847366	
		5.	-0.9707474	0.9450363	0.9003674	0.8270084	0.7149429	0.5595064	0.3711628	
		6.	-0.9848953	0.9707474	0.9450363	0.9003674	0.8270084	0.7149429	0.5595064	

Tab. I: Podmínková rovnice pro výpočet akčních zásahů v min. počtu kroků

K porovnání výsledků Kaczmarzova algoritmu byl sestaven program řešící rovněž výpočet optimálních akčních zásahů v minimálním počtu regulačních kroků, ale odlišnou metodou. Tato metoda přímého výpočtu byla navržena s. Hanouskem v práci /5/.

Pro jednotlivé řády soustav / 0. - 5. / bylo zkoumáno, jak dosažené hodnoty akčních zásahů vyhovují podmínkovým rovnicím. Získané hodnoty akčních zásahů byly tedy dosazeny zpět do podmínkových rovnic / matice pravých stran / a porovnáním vektorů levých stran byla zjištěna chyba výpočtu. Totéž bylo provedeno s výsledky obou metod- Kaczmarzovy i přímé.

Řešení spolu se stanovením chyb jsou zaznamenána v tabulce.
/ Tab.II /

Pro případ soustav 3. a 5. řádu Kaczmarzův algoritmus konverguje již velmi pomalu, v tabulce / Tab.II / jsou uvedeny hodnoty získané při 30 000 iteracích. Tyto hodnoty ještě nejsou ustálené, tedy jsou značně nepřesné.

Ustálením hodnot rozumíme jejich neměnnost v následných iteračních krocích nebo jejich periodickou změnu do řádu 10^{-5} . Závislost průměrné chyby splnění podmínkových rovnic na řádu soustavy / resp. počtu neznámých parametrů / je vynesena v grafu / Obr. 3-5 /.

Lze též pozorovat závislost počtu potřebných iterací na řádu soustavy / počtu podmínkových rovnic /.

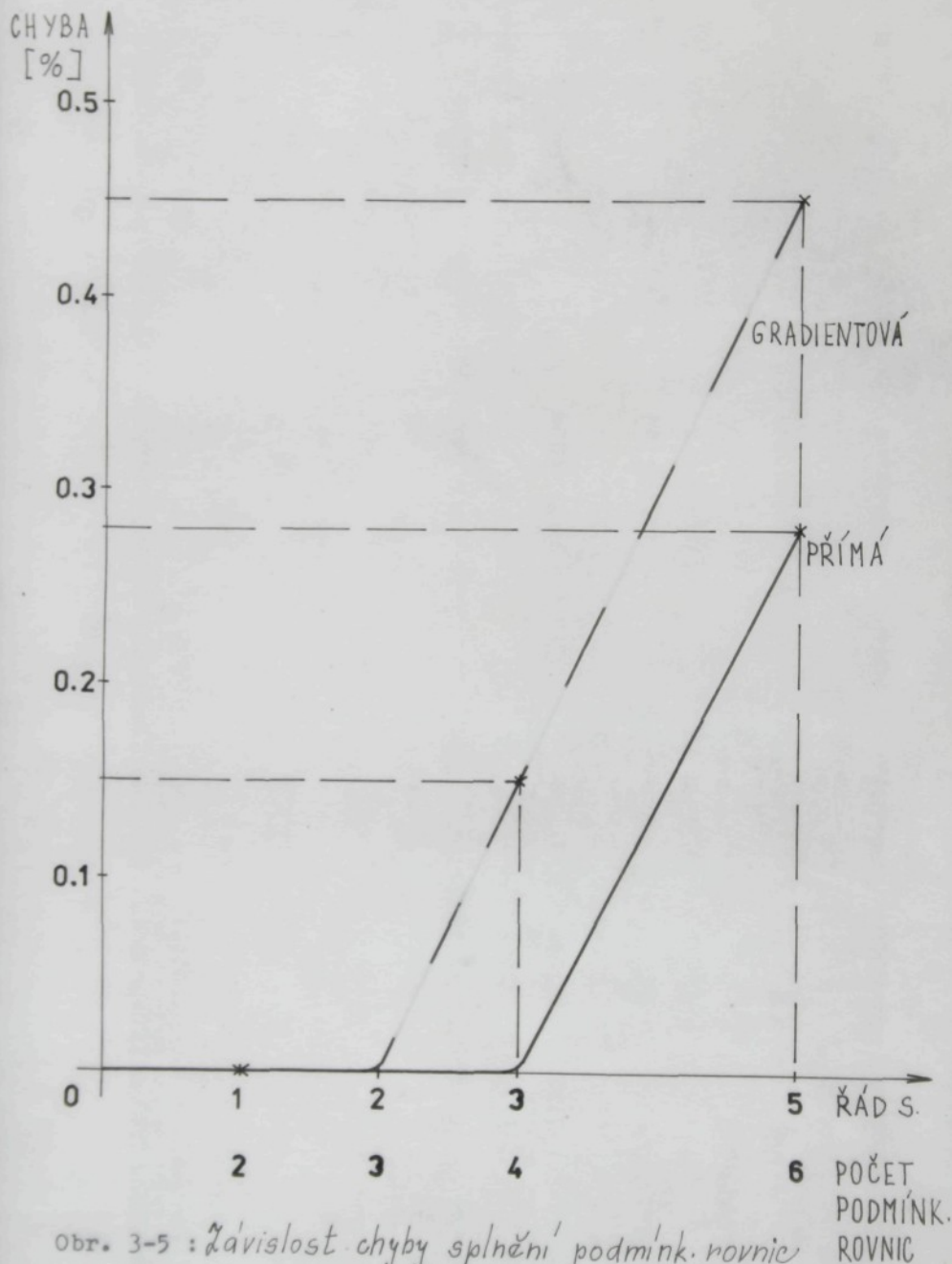
V tabulce / Tab. III / je uvedeno pro každou soustavu, kolik je třeba iterací na dosažení přesnosti výsledku $\varepsilon = 0.001$. Přesnost ε je chápána jako odchylka výsledku od již ustálené hodnoty.

$$\underline{\varepsilon} = |\Delta \underline{w} - \Delta \hat{\underline{w}}|$$

kde je $\Delta \hat{\underline{w}}$ výsledek (odhad), $\Delta \underline{w}$ ustálená hodnota

ŘS	PP	PI	ČP	ŘEŠENÍ KACZMARZOVÝM ALGORITMEM				ŘEŠENÍ PŘÍMOU METODOU			
				HODNOTY AKČ. ZÁSAHU	ABSOLUTNÍ CHYBA $\cdot 10^2$	PRŮMĚRNÁ CHYBA $\cdot 10^2$		HODNOTY AKČ. ZÁSAHU	ABSOLUTNÍ CHYBA $\cdot 10^2$	PRŮMĚRNÁ CHYBA $\cdot 10^2$	
0	1	1	1.	-0.9999996	0.00004	0.00004		—	—	—	
1	2	70	1.	-1.3678790	0.00005	0.00003		-1.3678790	0.00005	0.000025	
			2.	0.3678794	0.00001			0.3678793	0.		
2	3	1100	1.	-1.7357850	0.00005			-1.7357580	0.00055		
			2.	0.8711403	0.00022	0.00011		0.8710938	0.00026	0.000280	
			3.	-0.1353564	0.00005			-0.1353352	0.00003		
3	4	30000 bez ustálení ní	1.	-1.4391410	0.03803			-2.1036380	0.00180		
			2.	0.0465801	0.22894	0.15395		1.5096440	0.00120	0.00083	
			3.	0.6665041	0.34873			-0.4557926	0.00030		
			4.	-0.2929686	0.00008			0.0497870	0.00003		
5	6	30000 bez ustálení ní	1.	-1.4866850	1.62155			-2.8393970	0.09539		
			2.	-0.0289712	0.46422			3.1927480	0.02709		
			3.	0.6488302	0.09074			-1.8512230	0.07122		
			4.	0.1941592	0.05190	0.45547		0.5894485	0.24891	0.28296	
			5.	-0.5104450	0.02816			-0.0983160	0.50119		
			6.	0.1892064	0.47328			-0.0067379	0.75399		

Tab. II: Porovnání metod



Obr. 3-5 : Závislost chyby splnění podmínek rovnice na řádu soustavy.

ŘS	PP	ČP	POTŘEBNÝ POČET ITERACÍ NA DOSAŽENÍ PŘESNOSTI $\varepsilon = 0.001$
0	1	1.	1
1	2	1. 2.	1 28
2	3	1. 2. 3.	6600 7500 6900
3	4	1. 2. 3. 4.	odhad ~ 900000 odhad ~ 900000 odhad ~ 900000 odhad ~ 900000
5	6	1.-6.	nelze stanovit odhadem

Tab. III: Závislost počtu iterací na počtu podmínekových
rovníc

b/ Kvadratické kritérium

Postup výpočtu akčních zásahů pomocí kvadratického kritéria je odvozen v / kap. 2.2. / této práce. Vzorce (2.2-10) a (2.2-11) popisují sestavení vektoru levých a matice pravých stran podmínkových rovnic. K výpočtu je opět použito Kaczmarzova iteračního algoritmu, popsaného vztahem (2.2-12).

Také hodnoty přechodových charakteristik jsou voleny stejně jako u algoritmu v minimálním počtu kroků.

Volbou konstanty $p=0.0$, což je váhový koeficient v kvadratickém kritériu

$$J = \sum_{i=1}^N (y_i^2 + p \Delta u_{i-1}^2) \rightarrow \min$$

neklademe na akční zásahy žádné omezení. Hodnoty akčních zásahů by měly tedy přibližně odpovídat hodnotám akčních zásahů stanoveným v minimálním počtu kroků / pokud $p=0.0$ /. Že tomu tak je, se přesvědčíme porovnáním hodnot, které jsou z výsledků programu vyneseny do tabulky / Tab.IV /.

Číselné vyjádření podmínkových rovnic je uvedeno v následujících tabulkách / Tab.V, VI /.

ŘS	PP	PI	ČP	HODNOTY AKČNÍCH ZÁSAHU ^o		
				ŘEŠENÍ KACZMARZOVÝM ALGORITHMEM		ŘEŠENÍ PŘÍMOU METODOU
				B	A	B
0	5	3000	1.	-0.992188	-0.999999	—
			2.	-0.011043	0.	—
			3.	0.003671	0.	—
			4.	-0.000451	0.	—
			5.	0.000013	0.	—
1	3	9000	1.	-1.367874	-1.367879	-1.367872
			2.	0.367872	0.367879	0.367864
			3.	0.000003	0.	-0.000012
2	5	30000	1.	-1.502574	-1.735785	-2.248239
			2.	0.265880	0.871140	2.023522
			3.	0.396932	-0.135356	-1.173715
			4.	-0.124999	0.	0.602337
			5.	-0.098350	0.	-0.310491

Tab. IV

Význam symbolů:

ŘS ... řád soustavy

PP ... počet podmínkových rovnic

PI ... počet iterací

ČP ... číslo podmínkové rovnice

A algoritmus - minimální počet kroků

B kvadratické kritérium / $p = 0.$ /

ŘS	PP	ČP	P O D M Í N K O V É R O V N I C E					
			VEKTOR LS	MATICE PRÁVÝCH STRAN				
				Δw_1	Δw_2	Δw_3	Δw_4	Δw_5
0	5	1.	-4.9999980	4.9999980	3.9999980	2.9999920	1.9999992	0.9999996
		2.	-3.9999980	3.9999980	3.9999980	2.9999920	1.9999992	0.9999996
		3.	-2.9999992	2.9999992	2.9999992	2.9999992	1.9999992	0.9999996
		4.	-1.9999996	1.9999992	1.9999992	1.9999992	1.9999992	0.9999996
		5.	-0.9999996	0.9999996	0.9999996	0.9999996	0.9999996	0.9999996
1	3	1.	-2.3009967	2.0501251	1.3681879	0.6006488		
		2.	-1.4494768	1.3681879	1.1472210	0.5465721		
		3.	-0.6205423	0.6006488	0.5465721	0.3995761		
2	5	1.	-3.1747999	2.8100376	2.2318782	1.5196943	0.8100238	0.2535585
		2.	-2.4123601	2.2318782	1.8892578	1.3601760	0.7512187	0.2400440
		3.	-1.5969800	1.5196943	1.3601760	1.0640163	0.3626591	0.2116180
		4.	-0.8372455	0.8100238	0.7512187	0.6326591	0.4226522	0.1569576
		5.	-0.2596511	0.2535585	0.2400440	0.2116180	0.1569576	0.0698233

Tab. V: Podmínková rovnice pro výpočet akč. zásahu podle kvadnat. kritéria
(Itenační výpočet)

ŘS	PP	ČP	PODMÍNKOVÉ ROVNICE					
			VEKTOR LS	MATICE PRAVÝCH STRAN				
				ΔW_1	ΔW_2	ΔW_3	ΔW_4	ΔW_5
1	3	1.	-2.3009967	2.0501251	1.3681879	0.6006488		
		2.	-1.4494768	1.3681879	1.1472210	0.5465721		
		3.	-0.6205423	0.6006488	0.5465721	0.3995761		
1	5	1.	-4.2668647	4.0003967	3.2760658	2.3932867	1.4793811	0.6278608
		2.	-3.3725366	3.2760658	3.0138282	2.3009967	1.4494768	0.6205423
		3.	-2.4272384	2.3932867	2.3009967	2.0501251	1.3681879	0.6006488
		4.	-1.4903821	1.4793811	1.4494768	1.3681879	1.1472210	0.5465721
		5.	-0.6305534	0.6278608	0.6205423	0.6006488	0.5465721	0.3995761

Tab. VI: Podmínkové rovnice pro výpočet akč. zásahů podle kvadrat. kriteria
(Iterační výpočet)

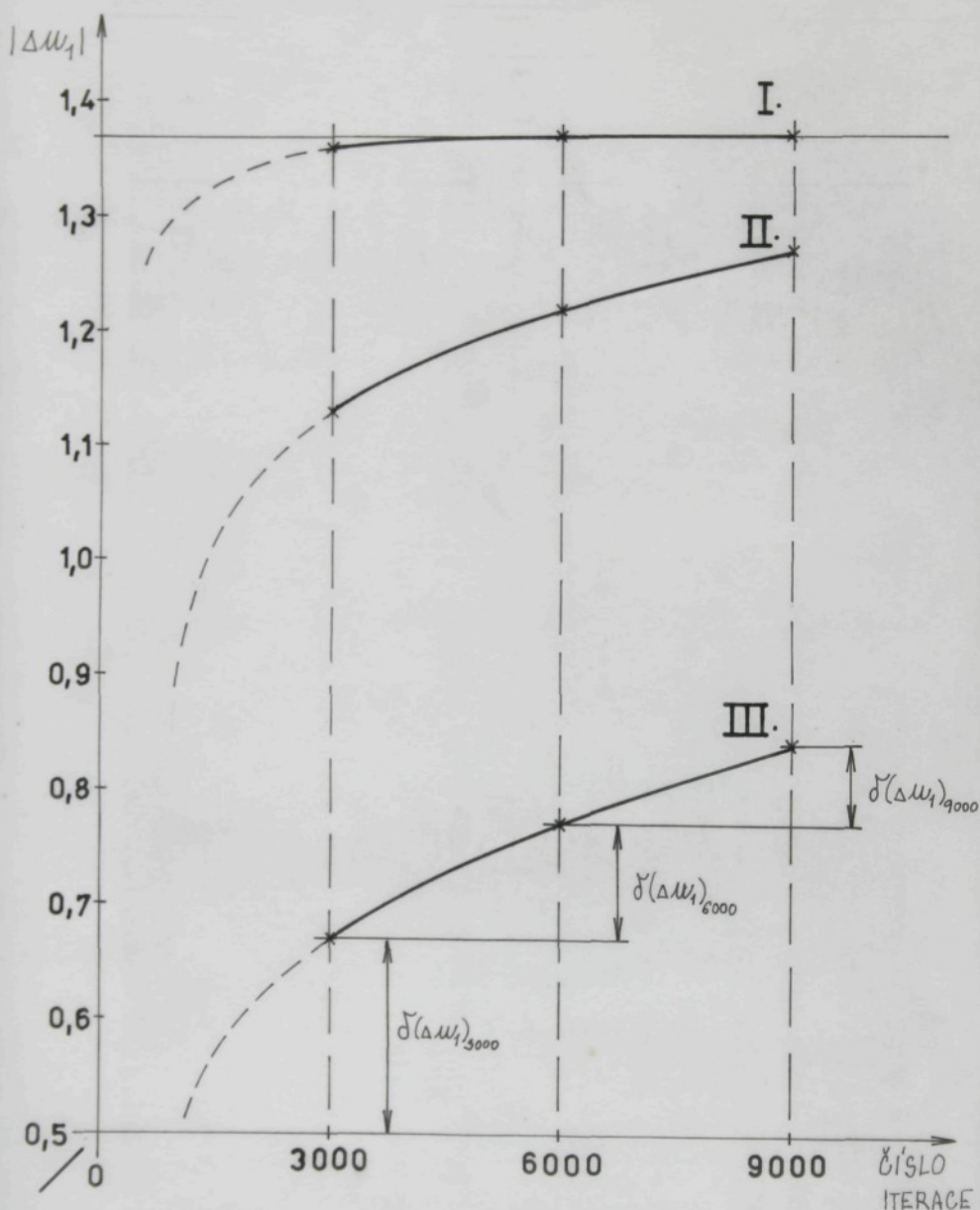
A n a l ý z v ý s l e ě k ů

Proveďme nyní podrobnější analýzu dosažených výsledků. Ukazuje se, že přesnost výpočtu / resp. rychlost konvergence iteračního postupu / je také závislá na počtu podmínkových rovnic / resp. neznámých parametrů /. Ověření bylo provedeno na soustavě 1. řádu. Postupně byl snižován počet podmínkových rovnic / PP = 10, 5, 3 /. Výsledky a grafické znázornění jsou uvedeny v tabulce a grafu / Tab. VII, Obr. 3-6 /.

Pro soustavu 2. řádu byl zvolen počáteční počet podmínkových rovnic - 10; při zjištění nedostatečné přesnosti výsledků, byl tento počet snížen na 5. Ale výpočet ani po 30 000 iteracích / při 5ti neznámých parametrech / nedává ještě ustálené nebo předpokládané hodnoty / viz příloha 3 - výsledky programu AKC /.

SOUSTAVA 1. ŘÁDU	PP	HODNOTY AKČNÍCH ZÁSAHU			PŘÍRUSTEK ZA 3 000 ITERACÍ		
		Δu_1	Δu_2	Δu_3	$ \delta(\Delta u_1) $	$ \delta(\Delta u_2) $	$ \delta(\Delta u_3) $
3 000	3	-1.364016	0.361576	0.002827	1.364016	0.361576	0.002827
	5	-1.127681	-0.018801	0.149455	1.127681	0.018801	0.149455
	10	-0.672521	-0.323699	-0.083862	0.672521	0.323699	0.083862
6 000	3	-1.367840	0.367816	0.000029	0.003824	0.006240	0.002798
	5	-1.223077	0.113542	0.133166	0.095396	0.094741	0.130339
	10	-0.773876	-0.298565	-0.023626	0.101355	0.025134	0.060236
9 000	3	-1.367874	0.367872	0.000003	0.000034	0.000056	0.000026
	5	-1.270675	0.188379	0.107584	0.047598	0.074837	0.025582
	10	-0.835871	-0.272109	0.014792	0.061995	0.026456	0.008834

Tab. VII: Závislost rychlosti konvergence na počtu podmínekových rovníc



Obr. 3-6 : Graf závislosti rychlosti konvergence
na počtu podmínkových rovnic
řádky I-3, II-5, III-10 podmínkových rovnic

3.2. Výpočet koeficientů algoritmu řízení

Kaczmarzův algoritmus lze použít i pro výpočet koeficientů algoritmu řízení, jak je odvozeno v kap.2.2. této práce.

K výpočtu využíváme stále stejného iteračního vzorce

$$\hat{\underline{a}}(k) = \hat{\underline{a}}(k-1) + \mathcal{H}_k \frac{\Delta \underline{u}(k) - \hat{\underline{a}}^T(k-1) \underline{u}(k)}{\underline{u}^T(k) \cdot \underline{u}(k)} \cdot \underline{u}(k)$$

kde je

$\hat{\underline{a}}(k)$ odhad vektoru koeficientů algoritmu řízení v k-tém iteračním kroku

$\hat{\underline{a}}(k-1)$ odhad vektoru koeficientů algoritmu řízení v k-1 iteračním kroku

$\mathcal{H}_k$ koeficient konvergence / voleno stále $\mathcal{H}_k = 1$ /

$\Delta \underline{u}$ akční zásah příslušející vektoru levých stran podmínkových rovnic

k^* číslo podmínkové rovnice

$\underline{u}(k^*)$ vektor z matice pravých stran podmínkových rovnic.

Máme zadánu jednoparametrovou soustavu, přenos soustavy jako v předchozích případech

$$S(p) = \frac{1}{(Tp+1)^n}$$

n řád soustavy / voleno $n = 1, 2, 3, 5$ /

Známe hodnoty přechodových charakteristik / z tabulek práce /5/, a tvar poruchy.

Pro výpočet koeficientů algoritmu řízení bylo použito algoritmů na vyrovnaní těchto poruch:

1/ - vyrovnaní determinované poruchy ve tvaru jednotkového skoku na vstupu do regulované soustavy, před započítím regulace

A- algoritmus řízení v minimálním počtu regulačních kroků / volen u soustav 1, 2, řádu /.

B- algoritmus podle kvadratického kritéria jakosti regulace

/ volen u soustav 1,2 řádu /.

2/ - vyrovnnání skokové změny žádané hodnoty podle kvadratického kritéria jakosti regulace

/ volen u soustav 3,5 řádu /.

Před výpočtem koeficientů algoritmu řízení byla provedena simulace optimálního průběhu regulované veličiny . Vycházelo se ze znalosti konvolutorního součtu (3.1-1), typu a místa vstupu poruchy , z vypočtených hodnot akčních zásahů / kap.3.1. , schema- Obr.3-2 /

Simulací byl určen optimální průběh regulované veličiny y' . Kaczmarzovým algoritmem byly pak vypočteny hodnoty koeficientů algoritmu řízení / viz schema- Obr.3-3 /.

A n a l ý z a v ý s l e d k ů

1/ - vyrovnnání determinované poruchy ve tvaru jednotkového skoku na vstupu do regulované soustavy, před započítáním regulace

Při praktickém ověřování se ukázalo nejvýhodnější volit počet podmínkových rovnic Kaczmarzova algoritmu roven počtu neznámých koeficientů. Nadbytečný počet rovnic, které mají být splněny s týmiž koeficienty, zanáší do Kaczmarzova algoritmu nepřesnosti a zhoršuje výpočet / příloha 5 -výsledky programu RR /.

Př. podmínkové rovnice pro soustavy 1. a 2. řádu mají tvar:

/ algoritmy A, B /

$$\Delta u(j) = a_1 y(j) + a_2 y(j-1) + a_3 \Delta u(j-1)$$

$$\Delta u(j) = a_1 y(j) + a_2 y(j-1) + a_3 y(j-2) + a_4 \Delta u(j-1) + a_5 \Delta u(j-2)$$

Číselné vyjádření podmínkových rovnic pro vyrovnání poruchy 1 podle A, B a pro jednotlivé řády soustav, je v následujících tabulkách / Tab. VIII, IX /.

Bylo ověřováno, jak vypočtené hodnoty koeficientů algoritmu řízení vyhovují podmínkovým rovnicím.

Porovnání algoritmů 1A, 1B nalezneme v tabulce / Tab. X /.

V / Tab. X / je uveden též počet potřebných iterací k dosažení ustálených hodnot iteračního postupu. U soustavy 2. řádu / pouze u algoritmu 1B / jsou uvedeny hodnoty při 20 000 iteracích, které ještě nejsou ustálené ani přesné. Pojem ustálení chápeme ve smyslu definice uvedené v předcházející kapitole / kap. 3.1. / .

ALGORITMUS 1.A									
ŘS	PP	ČP	PODMÍNKOVÉ ROVNICE						
			VEKTOR	MATICE PRAVÝCH STRAN					
				a_1	a_2	a_3	a_4	a_5	
1	3	1.	LS	-1.3678784	0.6321203	0.	0.		
		2.		0.3678789	0.6321203	-1.3678784			
		3.		0.	0.0000002	0.3678789			
2	5	1.		-1.7357574	0.2642409	0.	0.	0.	0.
		2.		0.8710941	0.1353351	0.2642409	-1.7357574	0.	0.
		3.		-0.1353351	0.0000011	0.1353351	0.8710941	0.8710941	-1.7357574
		4.		0.	-0.0000002	0.0000011	-0.1353351	0.1353351	0.8710941
		5.		0.	0.0000014	-0.0000002	0.0000011	0.	-0.1353351

Tab. VIII. Podmínkové rovnice pro výpočet koeficientů algoritmu řízení podle 1A
(Itanačtní výpočet)

ALGORITHMUS 1 B									
ŘS	PP	ČP	PODMÍNKOVÉ ROVNICE						
			VEKTOR LS	MATICE PRAVÝCH STRAN					
				a_1	a_2	a_3	a_4	a_5	
1	3	1.	-1.3678720	0.6321203	0.	0.			
		2.	0.3678637	0.0000045	0.6321203	-1.3678720			
		3.	-0.0000124	-0.0000038	0.0000045	0.3678636			
2	5	1.	-2.2482390	0.2642409	0.	0.	0.	0.	
		2.	2.0235200	-0.0000832	0.2642409	0.	-2.2482390	0.	
		3.	-1.1737140	0.0001087	-0.0000832	0.2642409	2.0235200	-2.2482390	
		4.	0.6023378	-0.0002689	0.0001087	-0.0000832	-1.1737140	2.0235200	
		5.	-0.3104913	-0.0002524	-0.0002689	0.0001087	0.6023378	-1.1737140	

Tab. IX: Podmínkové rovnice pro výpočet koeficientů algoritmu řízení podle 1B
(Iterační výpočet)

ŘS	PP	PI	ČP	ŘEŠENÍ - ALGORITHMUS 1A			ŘEŠENÍ - ALGORITHMUS 1B		
				HODNOTY KOEFCIENTU	ABSOLUTNÍ CHYBA $\cdot 10^6$	PRŮMĚRNÁ CHYBA $\cdot 10^6$	HODNOTY KOEFCIENTU	ABSOLUTNÍ CHYBA $\cdot 10^6$	PRŮMĚRNÁ CHYBA $\cdot 10^6$
1	3	200	1.	-2.163951	0.700		-2.163940	1.600	
			2.	0.581975	0.460	0.430	0.581829	0.480	0.720
			3.	-0.000001	0.116		-0.000063	0.081	
2	5	4000 1A	1.	-6.568840	0.900		-8.508278	4.00	
			2.	3.295669	4.310		0.872597	121000.0	
		40000 1B	3.	-0.511765	7.810	4.327	0.461606	3700.0	251940.9
			4.	-0.512303	8.250		-0.796127	46000.0	
			5.	-0.000079	0.367		-0.142353	0.7	

Tab.X: Porovnání algoritmů

Rozbor výsledků algoritmu 1B na soustavě 2. řádu

Jak již bylo uvedeno při výpočtu koeficientů algoritmu řízení na soustavě 2. řádu, algoritmem vyrovnávajícím determinovanou poruchu na vstupu do regulované soustavy podle kvadratického kritéria / 1B /, nedošlo s výjimkou prvního koeficientu k ustálení výsledků ani po 20 000 iteracích. Proto byly tyto koeficienty stanoveny odhadem, ze zjištěné skutečnosti, že hodnoty koeficientů algoritmu, získané v jednotlivých iteračních krocích, tvoří geometrickou řadu s kvocientem q . Výpočet odhadu je proveden v následujících tabulkách / Tab. XI, XII /. Byl použit vzorec pro součet geometrické řady

$$S_k = a_0 \cdot \frac{1-q^k}{1-q}$$

kde

$$\lim_{k \rightarrow \infty} q^k = 0 \Rightarrow \lim_{k \rightarrow \infty} S_k = S = \frac{a_0}{1-q}$$

a_0 první člen geometrické řady

q kvocient geometrické řady

S součet geometrické řady

Koeficient algoritmu $a_1 = -8.508278$ je již od prvního iteračního kroku ustálený, a proto v Tab. XI není uveden. Podmínkové rovnice vycházejí ze znalosti struktury algoritmu řízení:

$$\Delta u(j) = a_1 y(j) + a_2 y(j-1) + a_3 y(j-2) + a_4 \Delta u(j-1) + a_5 \Delta u(j-2)$$

Odhad koeficientů algoritmu řízení je porovnán se skutečnými hodnotami koeficientů algoritmu řízení, dosaženými po 20 000 iteracích. Porovnání je provedeno z hlediska toho, jak hodnoty koeficientů vyhovují podmínkovým rovnicím. Z Tab. XIII je zřejmé, že chyba výpočtu Kaczmarzovým algoritmem při 20 000 iteracích je značná oproti chybě s vypočteným odhadem.

ODHAD KOEFCIENTŮ ALGORTMU ŘÍZENÍ		ALGORTMUS 1B				soustava 2. řádu			
		a_2			a_3				
i		HODNOTA KOEFCIENTU	ROZDÍL po 1000 iteracích	$q = \frac{\partial a_{2i}}{\partial a_{2i-1}}$	HODNOTA KOEFCIENTU	ROZDÍL po 1000 iteracích	$q = \frac{\partial a_{3i}}{\partial a_{3i-1}}$		
ČI		a_2	$\Delta a_{2i} = a_{2i} - a_{2i-1}$		a_3	$\Delta a_{3i} = a_{3i} - a_{3i-1}$			
1. 16 000		0.7714115			0.5013411				
2. 17 000		0.7970436	0.0256321		0.4912706	-0.0100705		0.99045	
3. 18 000		0.8224430	0.0253994	0.99092	0.4812963	-0.0099743	0.99118	0.99166	
4. 19 000		0.8476257	0.0251827	0.99147	0.4714100	-0.0098863			
5. 20 000		0.8725971	0.0249714	0.99161	0.4616062	-0.0098038			
$a + \frac{\delta}{1-q}$		$0.8725971 + \frac{0.0249714}{0.0083907}$			$0.4616062 - \frac{0.0098038}{0.0083449}$				
ODHAD KOEFCIENTŮ		3.84863						-0.71322	

Tab. XI: Odhad koeficientů algoritmu řízení

ODHAD KOEFCIENTU ALGORITHMU ŘÍZENÍ		ALGORITHMUS 1B				soustava 2. řádu			
		a_4		a_5					
		HODNOTA KOEFCIENTU a_4	ROZDÍL pro 1000 iterací $\Delta a_{4i} = a_{4i} - a_{4i-1}$	$q = \frac{\Delta a_{4i}}{\Delta a_{4i-1}}$	HODNOTA KOEFCIENTU a_5	ROZDÍL pro 1000 iterací $\Delta a_{5i} = a_{5i} - a_{5i-1}$	$q = \frac{\Delta a_{5i}}{\Delta a_{5i-1}}$		
i	ČI								
1.	16 000	-0.8079834			-0.1484114				
2.	17 000	-0.8049797	0.0030037		-0.1468769	0.0015345			
3.	18 000	-0.8020039	0.0029758	0.99071	-0.1453563	0.0015206	0.99094		
4.	19 000	-0.7990534	0.0029505	0.99149	-0.1438489	0.0015074	0.99132		
5.	20 000	-0.7961277	0.0029257	0.99159	-0.1423539	0.0014950	0.99177		
$a + \frac{\delta}{1-q}$		-0.7961277 + $\frac{0.0029257}{0.0084054}$			-0.1423539 + $\frac{0.0014950}{0.0082261}$				
ODHAD KOEFCIENTU		-0.44805				0.03939			

Tab. XII: Odhad koeficientů algoritmu řízení

ALGORITHMUS 1B								
ŘS	PP	ČP	ŘEŠENÍ S ODHADEM			ŘEŠENÍ PO 20 000 ITERACÍCH		
			HODNOTY KOEFCIENTU	ABSOLUTNÍ CHYBA .10 ³	PRŮMĚRNÁ CHYBA .10 ³	HODNOTY KOEFCIENTU	ABSOLUTNÍ CHYBA .10 ³	PRŮMĚRNÁ CHYBA .10 ³
2	5	1.	-8.5082780	0.004		-8.5082780	0.004	
		2.	3.8486300	1.480		0.8725971	1210.0	
		3.	-0.7132188	11.0	4.59	0.4616062	3.7	251.94
		4.	-0.4480537	5.9		-0.7961277	46.0	
		5.	0.0393869	4.58		-0.1423539	0.0007	

Tab. XIII Porovnání skutečných hodnot koeficientů algoritmu řízení po 20 000 iteracích s jejich odhadem.

2/ - vyrovnání skokové změny žádané hodnoty podle kvadratického kritéria jakosti regulace.

Pro soustavu 3. a 5. řádu jsou koeficienty algoritmu řízení vypočítávány Kaczmarzovým algoritmem pro vyrovnání skokové změny žádané hodnoty.

Hodnoty akčních zásahů i hodnoty optimálního průběhu regulované veličiny jsou do Kaczmarzova algoritmu dosazeny ve tvaru vstupních dat z výsledků programu DEP, který používá kvadratického kritéria jakosti regulace a přímého výpočtu s inverzí matic. / Popis programu DEP / 6 / , odvození metody přímého výpočtu podle kvadratického kritéria / 4 / /.

Podmínkové rovnice Kaczmarzova algoritmu jsou uvedeny v Tab. XIV pro soustavu 3. řádu. Pro soustavu 5. řádu nejsou již podmínkové rovnice pro velký rozsah uvedeny v tabulce, nalezneme je v příloze(5).

Podmínkové rovnice vycházejí opět ze znalosti struktury algoritmu řízení.

Pro soustavu 3. řádu:

$$\Delta u(j) = a_1 y'(j) + a_2 y'(j-1) + a_3 y'(j-2) + a_4 y'(j-3) + \\ + a_5 \Delta u(j-1) + a_6 \Delta u(j-2) + a_7 \Delta u(j-3)$$

Pro soustavu 5. řádu:

$$\Delta u(j) = a_1 y'(j) + a_2 y'(j-1) + a_3 y'(j-2) + a_4 y'(j-3) + \\ + a_5 y'(j-4) + a_6 y'(j-5) + a_7 \Delta u(j-1) + a_8 \Delta u(j-2) + \\ + a_9 \Delta u(j-3) + a_{10} \Delta u(j-4) + a_{11} \Delta u(j-5)$$

$$y'(j-k) = y(j-k) - w(j) \quad k=0, \dots, m$$

ALGORTHMUS 2										
ŘS	PP	ČP	PODMÍNKOVÉ ROVNICE							
			VEKTOR	MATICE PRAVÝCH STRAN						
				LS	a_1	a_2	a_3	a_4	a_5	a_6
3	7	1.	-1.71377	0.99999	0.99999	0.99999	0.99999	0.	0.	0.
		2.	0.14025	0.86238	0.99999	0.99999	0.99999	-1.71377	0.	0.
		3.	0.45806	0.45716	0.86238	0.99999	0.99999	0.14025	-1.71377	0.
		4.	0.21456	-0.09361	0.45716	0.86238	0.99999	0.45806	0.14025	-1.71377
		5.	-0.00517	-0.05948	-0.09361	0.45716	0.86238	0.21456	0.45806	0.14025
		6.	-0.04868	-0.06011	-0.05948	-0.09361	0.45716	-0.00517	0.21456	0.45806
		7.	-0.03809	-0.01892	-0.06011	-0.05948	-0.09361	-0.04868	-0.00517	0.21456

Tab. XIV: Podmínkové rovnice pro výpočet koeficientů algoritmu řízení podle 2
(Iterační výpočet)

4. Ověření vypočtených algoritmů simulací regulačních pochodů

Byla provedena simulace regulačních pochodů při zapojeném regulátoru pro následující algoritmy:

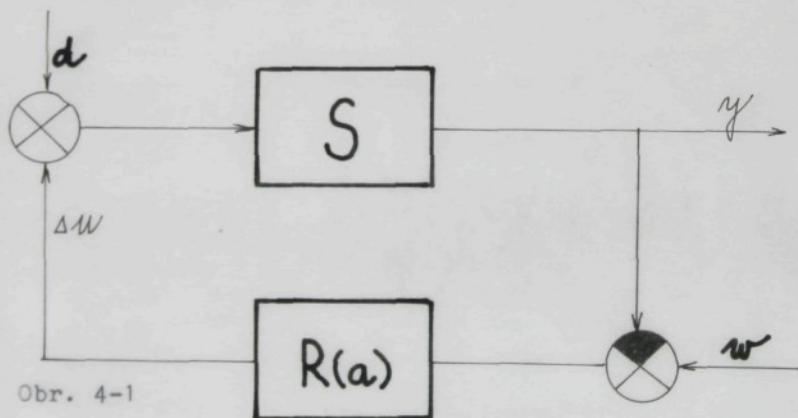
1/ - vyrovnaní determinované poruchy ve tvaru jednotkového skoku na vstupu do regulované soustavy, před započítím regulace.

A- algoritmus řízení v minimálním počtu regulačních kroků

B- algoritmus podle kvadratického kritéria jakosti regulace

2/ - vyrovnaní skokové změny žádané hodnoty podle kvadratického kritéria jakosti regulace

Schema postupu je následující



Obr. 4-1

Známe: d, w ... determinovaná porucha / určena tvarem a místem vstupu /

S ... soustava určena hodnotami přechodové charakteristiky

$R(a)$... algoritmus řízení vypočtený podle / kap. 3.2. /

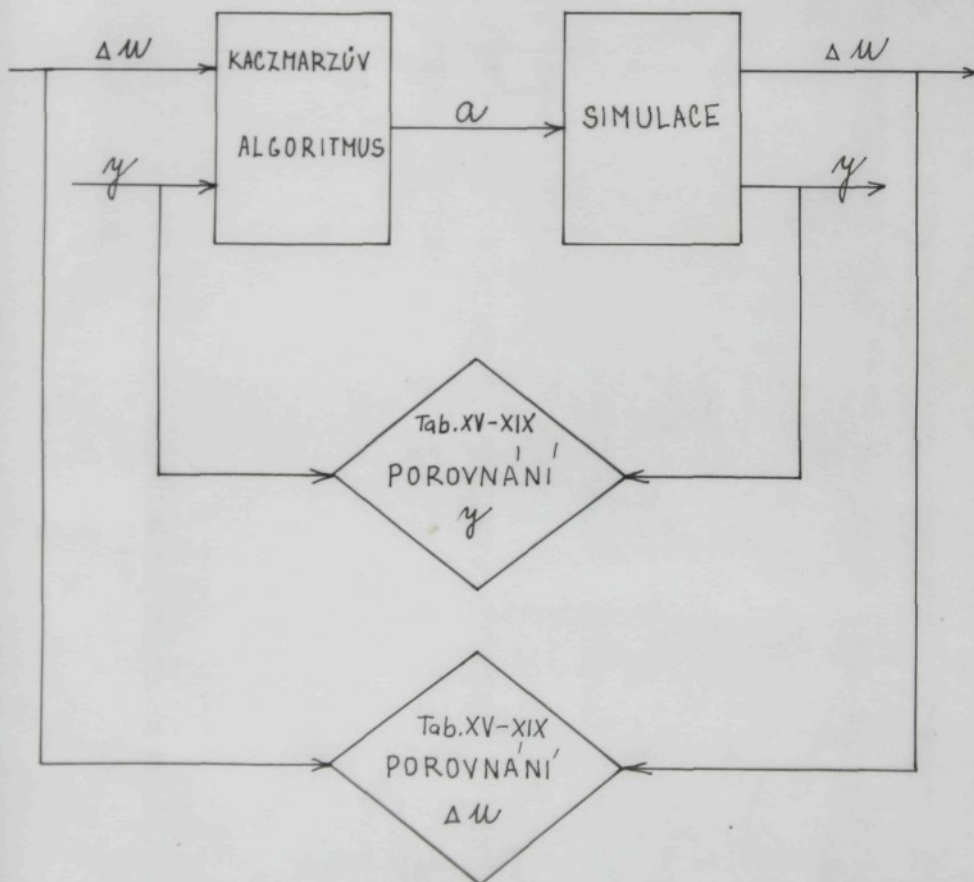
Určíme: y ... optimální průběh regulované veličiny / simulací /

Δw ... optimální akční zásahy / simulací /

A n a l ý z a v ý s l e d k ů

1/ Algoritmy 1A, 1B

Porovnání hodnot akčních zásahů ΔW a optimálního výstupu ze soustavy y , tak jak byly získány simulací regulačních pochodů pro soustavy řízené vypočteným algoritmem řízení podle Kaczmarzova algoritmu, s hodnotami výchozími, je v následujících tabulkách / Tab. XV - XIX/. Pro snazší orientaci v tabulkách je uvedeno schema postupu porovnání:



POROVNÁNÍ VÝCHOZÍCH A SIMULOVANÝCH REGULAČNÍCH POCHODŮ						soustava 1. řádu
ČÍSLO KROKU	HODNOTY AKČNÍCH ZÁSAD		HODNOTY REGUL. VELIČINY		SIMULOVANÉ	ALGORITMUS 1A
	VÝCHOZÍ	SIMULOVANÉ	VÝCHOZÍ	SIMULOVANÉ		
1.	-1.3678780	-1.3678770	0.6321203	0.6321203	0.6321203	ŘÍDÍCÍ ALGORITMUS: $\Delta u(j) = -2.163951 y(j) +$ $+ 0.581975 y(j-1) -$ $- 0.3760 \cdot 10^{-6} \Delta u(j-1)$
2.	0.3678789	0.3678722	0.0000002	0.0000002	0.0000009	
3.	0.	0.0000071	0.	0.	-0.0000031	
4.	0.	-0.0000023	0.0000002	0.0000002	0.0000002	
5.	0.	0.0000012	0.0000005	0.0000005	-0.0000005	

Tab. XV.

POROVNÁNÍ VÝCHOZÍCH A SIMULOVANÝCH REGULAČNÍCH POCHODŮ						soustava 1. řádu
ČÍSLO KROKU	HODNOTY AKČNÍCH ZÁSAD		HODNOTY REGUL. VELIČINY		SIMULOVANÉ	ALGORITMUS 1B
	VÝCHOZÍ	SIMULOVANÉ	VÝCHOZÍ	SIMULOVANÉ		
1.	-1.3678720	-1.3678720	0.6321203	0.6321203	0.6321203	ŘÍDÍCÍ ALGORITMUS: $\Delta u(j) = -2.163942 y(j) +$ $+ 0.581829 y(j-1) +$ $+ 0.6336 \cdot 10^{-6} \Delta u(j-1)$
2.	0.3678637	0.3678635	0.0000045	0.0000043	0.0000043	
3.	-0.0000124	-0.0000115	-0.0000038	-0.0000041	-0.0000041	
4.	0.0000068	0.0000291	-0.0000145	-0.0000145	-0.0000145	
5.	0.0000181	-0.0000085	-0.0000143	0.	0.	

Tab. XVI.

POROVNÁNÍ VÝCHOZÍCH A SIMULOVANÝCH REGULAČNÍCH POCHODU ⁰					soustava 2. řádu	
ČÍSLO KROKU	HODNOTY AKČNÍCH ZÁSAHU ⁰		HODNOTY REGUL. VELIČINY		ALGORITMUS 1A	
	VÝCHOZÍ	SIMULOVANÉ	VÝCHOZÍ	SIMULOVANÉ	ŘÍDÍCÍ ALGORITMUS:	
1.	-1.7357570	-1.7357570	0.2642409	0.2642409	$\Delta u(j) = -6.568840 y(j) +$	
2.	0.8710941	0.8710869	0.1353353	0.1353355	+ 3.295669 $y(j-1) -$	
3.	-0.1353351	-0.1353210	0.0000012	-0.0000014	- 0.511766 $y(j-2) -$	
4.	0.	-0.0000151	-0.0000002	0.0000009	- 0.512303 $\Delta u(j-1) -$	
5.	0.	0.0000255	0.0000014	-0.0000005	- 0.7974 $\cdot 10^{-4} \Delta u(j-2)$	

Tab. XVII

POROVNÁNÍ VÝCHOZÍCH A SIMULOVANÝCH REGULAČNÍCH POCHODU ⁰					soustava 2. řádu	
ČÍSLO KROKU	HODNOTY AKČNÍCH ZÁSAHU ⁰		HODNOTY REGUL. VELIČINY		ALGORITMUS 1B	
	VÝCHOZÍ	SIMULOVANÉ	VÝCHOZÍ	SIMULOVANÉ	ŘÍDÍCÍ ALGORITMUS:	
1.	-2.2482390	-2.2482370	0.2642409	0.2642409	$\Delta u(j) = -8.508278 y(j) +$	
2.	2.0235200	2.0250010	-0.0000832	-0.0000827	+ 3.848630 $y(j-1) -$	
3.	-1.1737140	-1.1888940	0.0001087	0.0004500	- 0.713218 $y(j-2) -$	
4.	0.6023378	0.6433466	-0.0002680	-0.0033986	- 0.448054 $\Delta u(j-1) +$	
5.	-0.3104913	-0.3719348	-0.0025200	0.0027523	+ 0.039387 $\Delta u(j-2)$	

Tab. XVIII

POROVNÁNÍ VÝCHOZÍCH A SIMULOVANÝCH REGULAČNÍCH POCHODU ⁰						soustava 3. řádu
ČÍSLO KROKU	HODNOTY AKČNÍCH ZÁSAHU ⁰		HODNOTY REGUL. VELIČINY		ALGORITHMUS 2	
	VÝCHOZÍ	SIMULOVANÉ	VÝCHOZÍ	SIMULOVANÉ	ŘÍDÍCÍ ALGORITHMUS:	
1.	-1.7137710	-1.7137990	0.9999996	0.9999992	$\Delta u(j) = 0.328071' y(j) -$	
2.	0.1402555	0.1402592	0.8623810	0.8623787	$-2.704748 y'(j-1) - 0.288734 y'(j-2) +$	
3.	0.4580660	0.4580734	0.4571589	0.4571507	$+0.951639 y'(j-3) - 0.011082 \Delta u(j-1) -$	
4.	0.2145595	0.2760065	-0.0936095	-0.0935957	$-1.244701 \Delta u(j-2) - 0.852703 \Delta u(j-3)$	
5.	-0.0051687	-0.5780073	-0.0594837	-0.0545656		

Tab. XIX

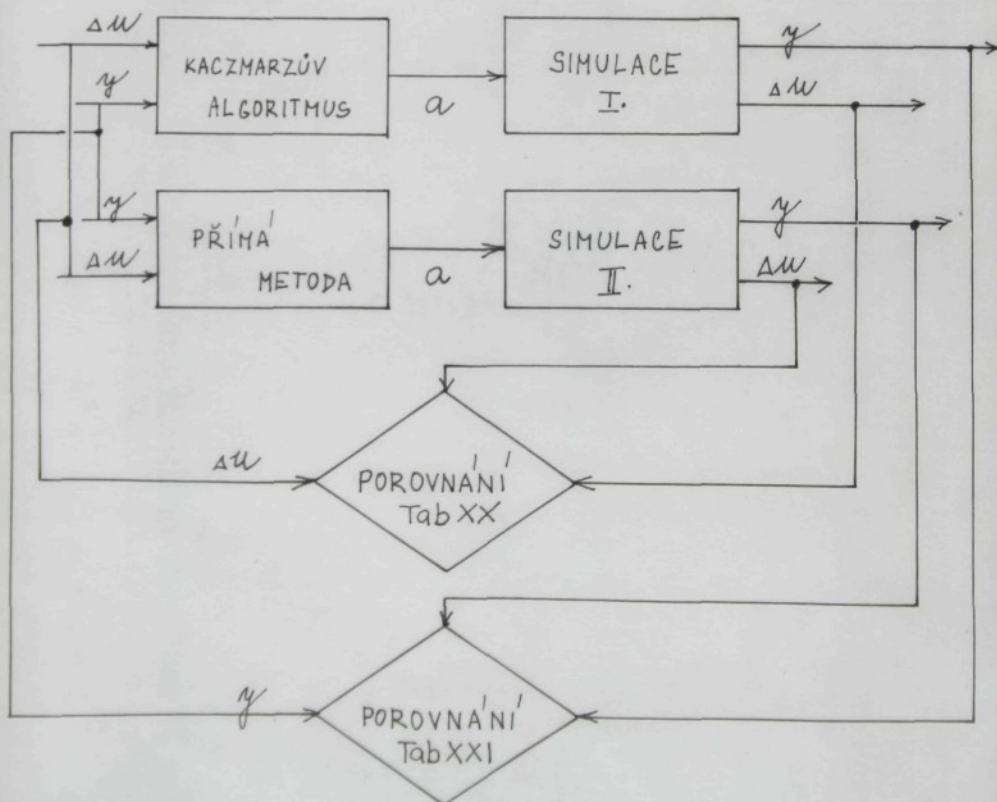
2/ Algoritmus 2

Při výpočtu koeficientů algoritmu řízení podle Kaczmarze se hodnoty vypočtených koeficientů algoritmu řízení liší od hodnot získaných přímým výpočtem / program DEP / 6 /, při použití stejných vstupů - hodnot akčních zásahů

- hodnot optimálních výstupů ze soustavy

Pro řídicí algoritmy získané oběma metodami byla provedena simulace regulačních pochodů. Porovnání simulací je v následujících tabulkách / Tab.XX, Tab.XXI/. Je patrné, že simulace algoritmu řízení navrženého podle Kaczmarze je přesnější.

Schema porovnání obou metod / Obr. 4-3 /



Obr. 4-3

POROVNÁNÍ ŘÍDÍCÍCH ALGORITMŮ					soustava 3. řádu
ČÍSLO KROKU	HODNOTY AKČNÍCH ZÁSAHU		SIMULACE		ALGORITMUS 2
	DU	I.	I.	II.	ŘÍDÍCÍ ALGORITMUS PODLE KACZMARZOVA ALGORITMU:
1.	-1.713771	-1.713799	-1.426025	-1.426025	$\Delta u(j) = 0.328071 y'(j) -$
2.	0.140255	0.140259	0.102777	0.102777	$- 2.704748 y'(j-1) - 0.288734 y'(j-2) +$
3.	0.458066	0.458073	0.354147	0.354147	$+ 0.951639 y'(j-3) - 0.011082 \Delta u(j-1) -$
4.	0.214559	0.276006	0.096682	0.096682	$- 1.244701 \Delta u(j-2) - 0.852703 \Delta u(j-3)$
5.	-0.005168	-0.578007	-0.064862	-0.064862	

Tab. XX

POROVNÁNÍ ŘÍDÍCÍCH ALGORITMŮ					soustava 3. řádu
ČÍSLO KROKU	HODNOTY REGULOVANÉ VELIČINY		SIMULACE		ALGORITMUS 2
	Y	I.	I.	II.	ŘÍDÍCÍ ALGORITMUS PODLE PŘÍMÉ METODY:
1.	0.999999	0.999999	0.999999	0.999999	$\Delta u(j) = -3.848143 y'(j) +$
2.	0.862381	0.862378	0.885487	0.885487	$+ 2.602538 y'(j-1) - 0.058594 y'(j-2) -$
3.	0.457178	0.457151	0.547185	0.547185	$- 0.121826 y'(j-3) - 0.763061 \Delta u(j-1) -$
4.	-0.093609	-0.093596	0.239123	0.239123	$- 0.290405 \Delta u(j-2) - 0.046692 \Delta u(j-3)$
5.	-0.059484	-0.054566	0.095067	0.095067	

Tab. XXI

Dále jsou uvedeny ukázky průběhů regulačních pochodů získaných simulací řídicích algoritmů navržených pomocí Kaczmarzova iteračního postupu.

Jedná se o tyto pochody:

Obr.	zobrazovaná veličina	soustava	porucha	algoritmus	kvaadratické kritérium
4-4	Δw	1. řádu	d	1B	$p^2 = 0$
4-5	y	1. řádu	d	1B	$p^2 = 0$
4-6	Δw	2. řádu	d	1B	$p^2 = 0$
4-7	y	2. řádu	d	1B	$p^2 = 0$
4-8	Δw	3. řádu	w	2	$p^2 = 0.1$
4-9	y	3. řádu	w	2	$p^2 = 0.1$

$$J = \sum_{i=1}^N y_i^2 + p^2 \Delta w_{i-1}$$

SEZNAM OZNACENI * F1 X F2 0 F3 +F4 =F5 .OSA
 MIN 1-0.1367870E 1
 MAX 1 0

LINE123456789012345678901234567890123456789012345678901234567890
 0 1 *
 2 *
 3 *
 4 *
 5 *
 6 *
 7 *
 8 *
 9 *
 10 *
 11 *
 12 *
 13 *
 14 *
 15 *
 16 *
 17 *
 18 *
 19 *
 20 *
 21 *

SEZNAM OZNACENI * F1 X F2 0 F3 +F4 =F5 .OSA
 MIN 1-0.1454352E- 4
 MAX 1 0.6321203

LINE1234567890123456789012345678901234567890123456789012345678901234567890

0 *
 1 *
 2 *
 3 *
 4 *
 5 *
 6 *
 7 *
 8 *
 9 *
 10 *
 11 *
 12 *
 13 *
 14 *
 15 *
 16 *
 17 *
 18 *
 19 *
 20 *
 21 *


```
SEZ/NAM UZNAULN1 * F1      X F2      0 F3      + F4      = F5      .OSA
MIN 1-0.340652L-2
MAX 1 0.2642409
```

L1NL123456789012345678901234567890123456789012345678901234567890

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21

Obr. 4-7

5. Urychlení konvergence gradientního Kaczmarzova algoritmu

Zabýváme-li se ověřením vhodnosti Kaczmarzova algoritmu k použití pro návrh algoritmu řízení, není možno se při tom nezbývat problematikou konvergence.

Zmíněný algoritmus / jak již bylo uvedeno v předcházejících kapitolách / , slouží pro odhad parametrů lineární soustavy

$$y = \underline{a}^T \underline{w} \quad (5-1)$$

a lze jej napsat ve tvaru

$$\hat{\underline{a}}(k) = \hat{\underline{a}}(k-1) + \mathcal{H}_k \frac{y(k) - \hat{\underline{a}}^T(k-1) \underline{w}(k)}{\underline{w}^T(k) \underline{w}(k)} \underline{w}(k) \quad (5-2)$$

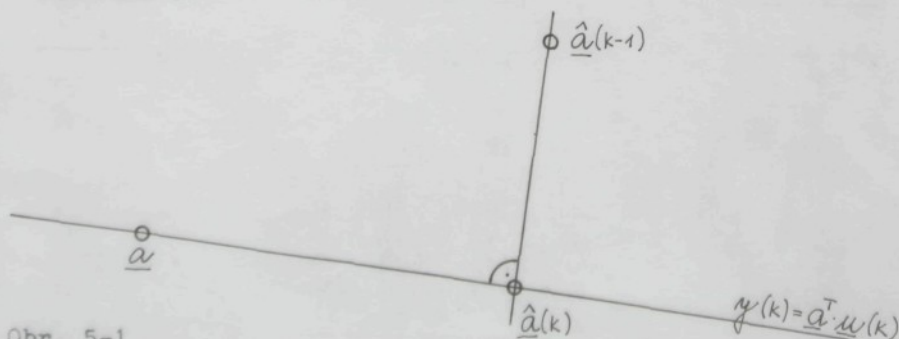
kde je

- y skalární výstup soustavy
- $\hat{\underline{a}}$ n -rozměrný vektor odhadů neznámých parametrů soustavy
- $\underline{w}$ n -rozměrný vektor vstupů soustavy
- $\mathcal{H}_k$ koeficient konvergence
- k krok

Pojem konvergence chápeme jako neustálé směřování posloupnosti odhadů $\hat{\underline{a}}(k)$ ke správné hodnotě $\underline{a}$.

Formule má následující geometrický význam: nový odhad $\hat{\underline{a}}(k)$ je ortogonální projekcí předchozího odhadu $\hat{\underline{a}}(k-1)$ do nadroviny (5-1)

/ Obr. 5-1 /.

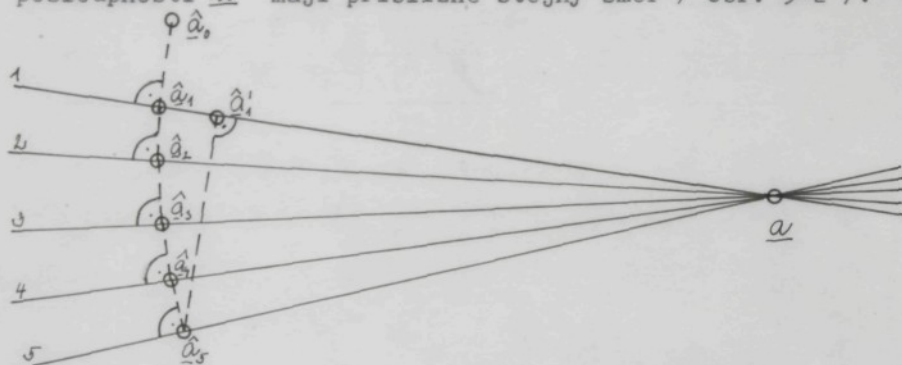


Obr. 5-1

Této geometrické interpretace se využívá i v návrzích na urychlení konvergence algoritmu (5-2).

Zaměříme se nyní na ty vlastnosti, které ovlivňují rychlost konvergence algoritmu (5-2) a ukažme, jakým způsobem lze konvergenci urychlit a tím i zpřesnit výsledné hodnoty odhadů parametrů $\hat{a}$.

Jednou z nepříznivých vlastností, která ovlivňuje rychlost konvergence, je ta skutečnost, kdy všechny vektory vstupní posloupnosti $\underline{u}$ mají přibližně stejný směr / Obr. 5-2 /.



Obr. 5-2

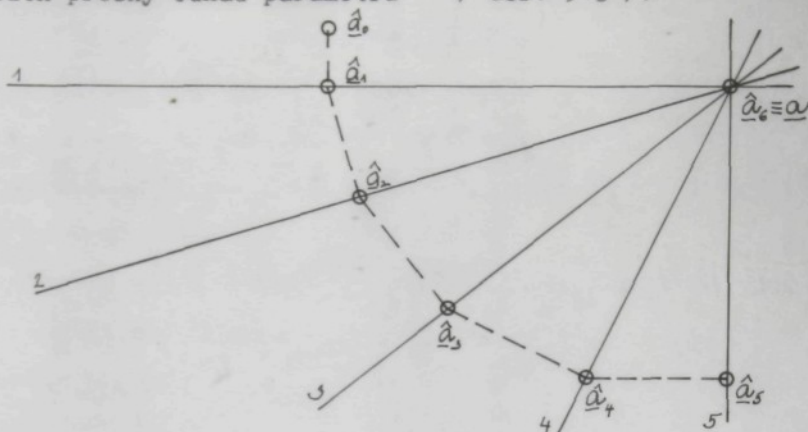
Čísla 1 - 5 jsou označena jednotlivá omezení soustavy rovnic (5-1).

Při iteračním výpočtu Kaczmarzova algoritmu, kterým byly získány hodnoty akčních zásahů i koeficienty algoritmu řízení, byl postup volby podmínkových rovnic^v pevné posloupnosti, jak je naznačeno na Obr. 5-2.

Největší rychlosti konvergence se dosáhne tehdy, tvoří-li vstupní vektory $\underline{u}$ ortogonální systém / Obr. 5-3 /.

Z porovnání Obr. 5-2 a Obr. 5-3 je patrné, jaký vliv má uspořádání vstupních vektorů na kvalitu konvergence. I při střídání podmínkových rovnic v pevné posloupnosti získáme po n kro-

cích přesný odhad parametrů / Obr. 5-3 /.



Obr. 5-3

5.1. Algoritmus ortogonalizace vstupních vektorů

J. Markl / 8 / popisuje algoritmus, který se vyrovnává současně se dvěma jevy:

- vektory vstupní posloupnosti mají přibližně stejný směr

/ Obr. 5-2 /

- změna směru vstupního vektoru probíhá velmi pomalu s časem, tento jev se vyskytuje spíše při identifikaci dynamických soustav.

Jedná se o zobecnění algoritmu(5-2), které lze vyjádřit takto:

$$\hat{f}(k) = \underline{w}(k) - \sum_{j=1}^p \frac{\underline{w}(k) \cdot \hat{f}(k-j)}{\hat{f}(k-j)^T \cdot \hat{f}(k-j)} \cdot \hat{f}(k-j) \quad (5.1-1)$$

$$\hat{a}(k) = \underline{a}(k-1) + \frac{\hat{y}(k) - \hat{a}^T(k-1) \underline{w}(k)}{\hat{f}(k)^T \cdot \hat{f}(k)} \cdot \hat{f}(k) \quad (5.1-2)$$

parametr p ve vzorci (5.1-1) je tzv. stupeň ortogonalizace

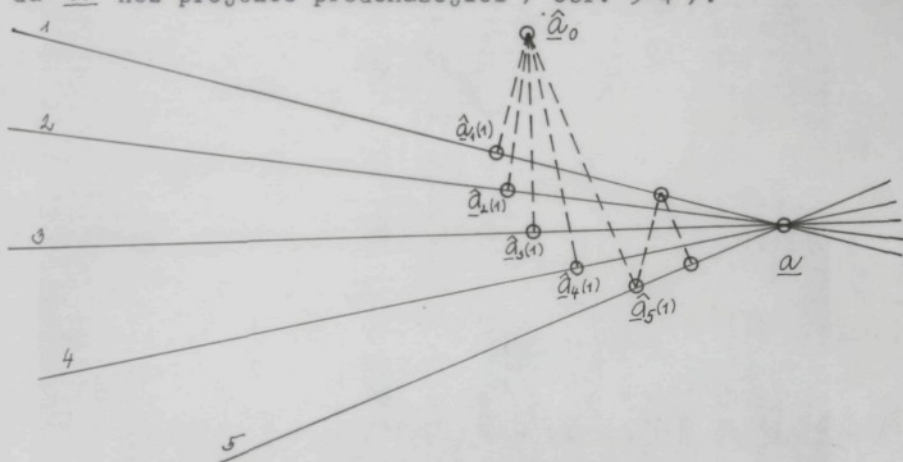
vstupní posloupnosti a může nabývat hodnoty $p = 0, 1, \dots, m-1$

Při $p = 0$ algoritmus (5.1-1), (5.1-2) přechází v původní(5-2).

Podrobnější popis a vlastnosti tohoto algoritmu nalezneme v /8/

5.2. Zobecněný Kaczmarzův algoritmus

Cypkin / 11 / uvádí Kaczmarzův algoritmus doplněný o tzv. algoritmus opravný. Princip zobecnění spočívá v tom, že algoritmus vybírá ze všech možných projekcí vektoru odhadů $\hat{a}^{(k-1)}$, do jednotlivých nadrovin i , jako další odhad tu projekci $\hat{a}_i^{(k)}$, která má od původní $\hat{a}^{(k-1)}$ větší vzdálenost. Je-li toto splněno, pak má také vybraná projekce blíže k hledanému přesnému odhadu $\underline{a}$ než projekce předcházející / Obr. 5-4 /.



Obr. 5-4

Tvar tohoto algoritmu je pak:

$$\hat{a}^{(k)} = \tilde{a}^{(k-1)} + \frac{y(k) - \tilde{a}^{(k-1)T} \underline{u}(k)}{\underline{u}^T(k) \cdot \underline{u}(k)} \underline{u}(k)$$

$$\tilde{a}^{(k-1)} = \hat{a}^{(k-1)} + \alpha(k) [\tilde{a}^{(k-2)} - \hat{a}^{(k-1)}]$$

$$\alpha(k) = \begin{cases} 1 & \text{jestliže } \rho^2[\tilde{a}^{(k-2)}, \underline{a}(k)] > \rho^2[\tilde{a}^{(k-2)}, \tilde{a}^{(k-1)}] \\ 0 & \text{v opačném případě} \end{cases}$$

$$\rho^2[\tilde{a}^{(m)}, \underline{a}(k)] = \frac{(y(k) - \tilde{a}^{(m)T} \underline{u}(k))^2}{\underline{u}^T(k) \underline{u}(k)}$$

kde je

$\hat{a}_{(k-1)}$ odhad parametru v $(k-1)$ -tém iteračním kroku

$\tilde{a}_{(k-1)}$ oprava odhadu $\hat{a}_{(k-1)}$

$\alpha(k)$ proměnný koeficient

$\rho^2[\tilde{a}_{(m)}, a_{(k)}]$ kvadrát vzdálenosti od bodu $\tilde{a}_{(m)}$ do k -té nadroviny

5.3. Algoritmus relaxace

Wernstedt / 10 / uvádí Kaczmarzův algoritmus ve tvaru

$$\hat{a}_{(k)} = \hat{a}_{(k-1)} + \frac{k}{\|\underline{u}_{(k)}\|^2} \cdot \underline{u}_{(k)} (\underline{y}_{(k)} - \underline{u}_{(k)}^T \hat{a}_{(k-1)})$$

pro konvergenci doporučuje volbu konstanty v rozmezí $0 < k < 2$

Zabývá se ve své práci / 10 / rušenými systémy při identifikaci soustav a srovnává algoritmus s tzv. rekurzivní

regresí, která vykazuje rychlejší konvergenci. V práci / 10 / je ukázán příznivý vliv koeficientů k na konvergenci řešení při existenci šumu.

5.4. Optimální volba koeficientu konvergence

Použitím Kaczmarzova algoritmu při identifikaci zašuměných soustav lze odvodit další modifikaci algoritmu (5-2) s optimální volbou koeficientu konvergence $\mathcal{H}_k$.

Největší rychlost konvergence k ustálené hodnotě v systému bez šumu, bude dosaženo při $\mathcal{H}_k = 1$.

Také v této diplomové práci byla ve všech případech iteračního výpočtu pomocí Kaczmarzova algoritmu (5-2) volena $\mathcal{H}_k = 1$.

V práci / 7, 8 / je odvozena rekurentní formule pro hodnotu zesílení $\mathcal{H}_k$, aby byl zabezpečen optimální průběh identifikace zašuměných soustav.

5.5. Doplnění Kaczmarzova algoritmu s využitím hledání ve směru

Tento postup, navržený na KTK, pro urychlení konvergence využívá vzorce

$$\hat{\underline{a}}(k) = \hat{\underline{a}}(k-1) + \mu_k \frac{\underline{y}(k) - \hat{\underline{a}}^T(k-1) \underline{u}(k)}{\underline{S}^T(k) \underline{u}(k)} \underline{S}(k) \quad (5.5-1)$$

kde

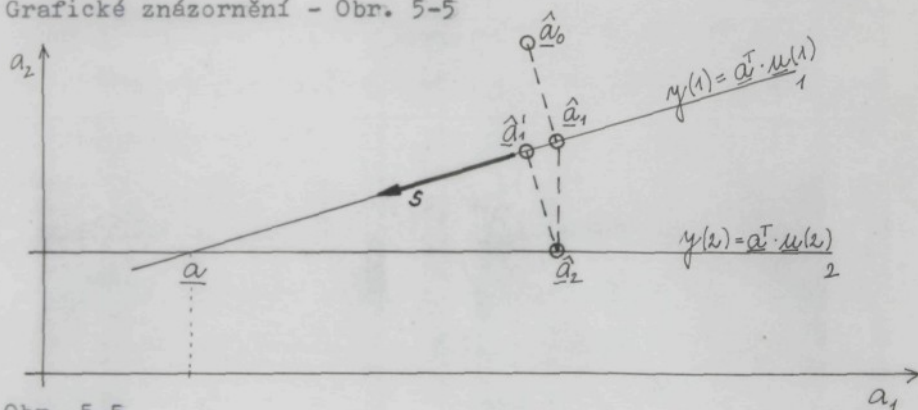
$$\underline{S}(k) = \hat{\underline{a}}(k) - \hat{\underline{a}}(k-1)$$

Praktické použití ukažme na příkladu určení neznámých parametrů $\underline{a}$ 2- rozměrné soustavy.

Podmínka je dána v souladu s (5-1).

$$y = \underline{a}^T \cdot \underline{u}$$

Grafické znázornění - Obr. 5-5



Obr. 5-5

Kolmými projekcemi získáme odhady $\hat{\underline{a}}_1$ a $\hat{\underline{a}}_1'$ splňující podmínku $y(1) = \underline{a}^T \cdot \underline{u}(1)$

a ležící na přímce 1. Z těchto odhadů určíme směr $\underline{S}$.

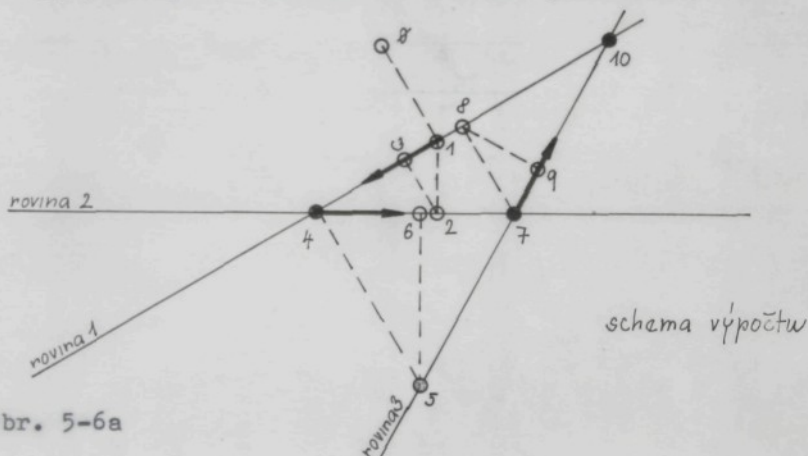
$$\underline{S} = \hat{\underline{a}}_1' - \hat{\underline{a}}_1$$

S použitím vzorce (5.5-1), získáme již ve 4. kroku přesný odhad

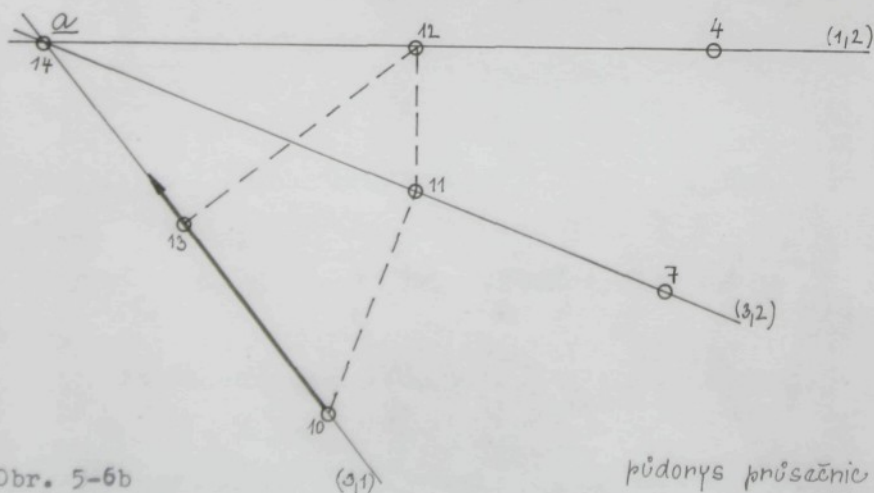
$$\underline{a}$$

Pro vícerozměrné soustavy je celý postup poněkud komplikovanější.

ší, ale v porovnání s klasickým algoritmem (5 - 2) rychlejší. Postup u 3-rozměrné soustavy je schematicky uveden na Obr. 5-6a. Čísly 1-14 jsou označeny kroky, kterými se dostaneme k hledanému přesnému odhadu $\underline{a}$. Body 4, 7, 10 na Obr. 5-6a představují body na průsečnici rovin $(1,2)$, $(3,2)$, $(3,1)$. Na Obr. 5-6b jsou zakresleny průměty všech tří průsečnic $/ (1,2), (2,3), (3,1) /$ do půdorysny. Ze dvou postupně získaných bodů na téže průsečnici 10 a 13 určíme směr do hledaného řešení $\underline{a}$, kterým je průsečík všech tří průsečnic.



Obr. 5-6a



Obr. 5-6b

Algoritmus (5.5-1) byl porovnán s klasickým algoritmem Kaczarze(5-2). Porovnání bylo provedeno pro výpočet optimálních akčních zásahů v minimálním počtu regulačních kroků / viz Tab. XXIII /.

n	0	1	2
p	1	2	3
k	1	28	7500
k*	1	4	14

Tab. XXIII

význam použitých symbolů:

n ... řád soustavy

p ... počet podmínkových rovnic, počet neznámých parametrů

k ... potřebný počet iterací algoritmu (5-2) - prakticky doloženo výpočtem

k* ... minimální teoretická hodnota počtu potřebných iterací navrhovaného algoritmu(5.5-1).

Navrhovaný algoritmus / resp. jeho část uvedená schematicky na Obr. 5-6a /, byl naprogramován v jazyce RPP - FORTRAN k ověření na počítači RPP 16 S. / Popis programu uveden v příloze 1 /.

5.6. Využití konvergence geometrické řady

Jak již bylo uvedeno v kapitole 3.2. , tvoří odhady parametrů soustavy po provedení potřebného počtu iterací, geometrickou řadu s kvocientem q. Odhad posledního členu geometrické řady provedeme podle vzorce

$$\hat{a}(\infty) = \hat{a}(k) + \frac{\Delta \hat{a}}{1-q} \quad \text{kdy} \quad \Delta \hat{a} = \hat{a}(k+1) - \hat{a}(k)$$

S použitím definice o konvergenci geometrické řady:

Geometrická řada konverguje tehdy, jestliže existuje konečná

limita $\lim_{k \rightarrow \infty} S_k = \underline{S}$ posloupnosti jeho částečných součtů.

Částečný součet se vyjádří:

$$\underline{S}_k = \underline{\Delta a} \cdot \frac{1-q^k}{1-q}$$

kde $\underline{\Delta a}$ první člen řady

q kvocient geometrické řady

Jestliže $|q| < 1$, pak $\underline{S} = \lim_{k \rightarrow \infty} S_k = \frac{\underline{\Delta a}}{1-q}$

Tak byly provedeny odhady parametrů $\hat{a}$ jako koeficientů

algoritmu řízení / výpočet uveden v kap. 3.2., Tab. XI, XII /.

S využitím konvergence geometrické řady můžeme například

také stanovit potřebný počet iterací k , na dosažení zvolené přesnosti ε .

Proveďme praktickou ukázkou na soustavě 1. řádu při výpočtu optimálních akčních zásahů v minimálním počtu regulačních kroků Kaczmarzovým algoritmem. (5=2)

1. AKČNÍ ZÁSAH- Δu_1 již 1. iterační krok dává ustálený odhad

$$\Delta u_1 = -1.3678790$$

2. AKČNÍ ZÁSAH- Δu_2 1. iterační krok $\Delta u_2(1) = 0$.

3. iterační krok $\Delta u_2(3) = 0.1281322$

5. iterační krok $\Delta u_2(5) = 0.2116361$

Rozdíl hodnot získaných při iteracích 1,3,5:

$$\Delta u_2(3) - \Delta u_2(1) = 0.1281322$$

$$\Delta u_2(5) - \Delta u_2(3) = 0.0835039$$

Podílem dostáváme kvocient q :

$$q = \frac{0.0835039}{0.1281322} = 0.6514$$

Odhad posledního členu geometrické řady je pak:

$$\hat{\Delta u_2} = 0.1281322 \cdot \frac{1}{1-0.6514} = 0.3678790$$

Zvolme požadovanou přesnost $\varepsilon = 0.001$, určíme pro které k

vyhovuje rovnice:

$$0.1281322 \frac{1}{0.3483} - 0.1281322 \frac{1 - q^k}{0.3483} = 0.001$$

$$k = \frac{\ln \frac{0.001 \cdot 0.3483}{0.1281322}}{\ln 0.6514} \doteq 14$$

Pro dvě podmínkové rovnice je počet potřebných iterací dvojnásobný, tedy:

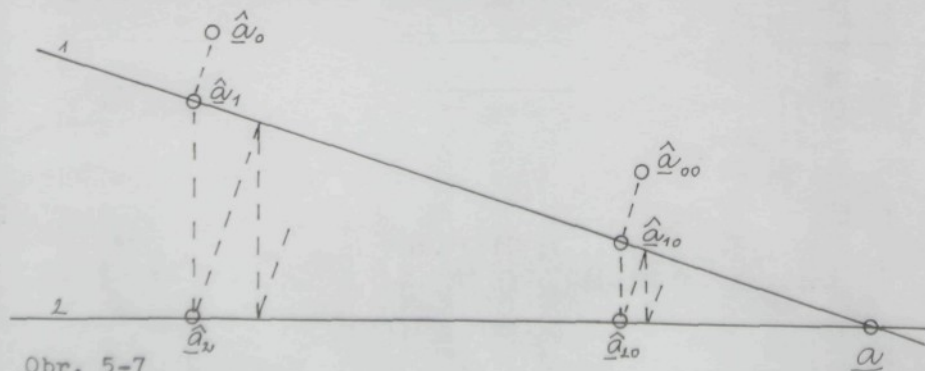
$$k = 28$$

Nyní ověříme, zda je vypočtené k skutečně správné. Ve výsledcích programu / příloha 4 /, nalezneme u 28. iteračního kroku hodnotu akčního zásahu Δu_{28} :

$$\Delta u_{28} = 0.3669627, \text{ která se liší od konečné o } \varepsilon$$

5.7. Volba počátečního odhadu

Počet potřebných iterací gradientové metody je závislý na volbě počátečního odhadu. Vycházíme-li z počátečního odhadu $\hat{a}(0) = \hat{a}_0$, je zajisté počet potřebných iterací větší, než kdybychom vycházeli z hodnoty bližší k hledanému přesnému odhadu $\underline{a}$ - například z $\hat{a}_{10}$ / Obr. 5-7 /.



Obr. 5-7

Při výpočtu optimálních akčních zásahů v minimálním počtu regulačních kroků Kaczmarzovým algoritmem byly místo původního počátečního odhadu $\Delta u(0) = 0$ dosazen jako počáteční odhad vektor přesných hodnot podle přímého výpočtu. / 5 /

Zkoumali jsme, jak se potom ustálené hodnoty z Kaczmarzova algoritmu liší od dosazených a které hodnoty splňují podmínkové rovnice přesněji. Ze srovnání z následující tabulky / Tab. XXIII / je patrné, že hodnoty konečné z Kaczmarzova algoritmu, splňují podmínkové rovnice ještě přesněji.

■ Na základě provedeného rozboru vyplývá, že potřebný počet iterací Kaczmarzova algoritmu ovlivňuje řada faktorů:

- volba počátečního odhadu,
- přesnost zadávaných vstupních vektorů
- počet podmínkových rovnic
- tvar a rozložení vstupních vektorů
- vzájemný sklon vstupních vektorů.

Na urychlení konvergence klasického Kaczmarzova algoritmu byl sestaven program, který byl laděn na soustavě 3. řádu při výpočtu čtyř optimálních akčních zásahů v minimálním počtu regulačních kroků.

ŘS	PP	ČP	ŘEŠENÍ PŘÍMOU METODOU		ŘEŠENÍ KACZMARZOVÝM ALGORITMEM			
			HODNOTY AKČ. ZÁSAHU	ABSOLUTNÍ CHYBA $\cdot 10^2$	PRŮMĚRNÁ CHYBA $\cdot 10^2$	HODNOTY AKČ. ZÁSAHU	ABSOLUTNÍ CHYBA $\cdot 10^2$	PRŮMĚRNÁ CHYBA $\cdot 10^2$
3	4	1.	-2.103638	0.00180	0.00083	VÝCHOZÍ $\hat{u}(0)$	KONEČNÉ	0.00051
		2.	1.509644	0.00120		-2.103638	-2.10372	
		3.	-0.455793	0.00030		1.509644	1.50972	
		4.	0.049787	0.00003		-0.455793	-0.45567	
						0.049787	0.04965	

Tab. XXIII: Vliv volby počátečního odhadu $\hat{u}(0)$ na přesnost splnění
podmínkových rovnic

6. Porovnání metod

Jsou porovnávány metody pro výpočet optimálních akčních zásahů. Porovnání je provedeno na základě praktického ověření algoritmů na číslicovém počítači RPP - 16S. Popisy programů jsou uvedeny v příloze 1.

Porovnávané algoritmy výpočtu:

A/ Kaczmarzův gradientní iterační

B/ Přímá metoda výpočtu optimálních akčních zásahů v minimálním počtu regulačních kroků, odvozená s. Hanouskem v / 5 /.

C/ Přímá metoda výpočtu optimálních akčních zásahů podle kvadratického kritéria jakosti regulace, algoritmus odvozen v / 4 /.

6.1. Náročnost na paměť počítače

Algoritmy A,B jsou velmi nenáročné na kapacitu paměti počítače. K výpočtu není třeba ukládat do paměti velké soubory dat, jak je tomu u algoritmu C. Při použití metody C je nutno provádět inverzi matic, čímž se obsazují místa v paměti. Programy pracují s velkými soubory čísel, což je nevýhoda metody C. Kaczmarzův algoritmus A se pro svou nenáročnost na paměť počítače doporučuje použít při mikroprocesorovém řízení procesů.

6.2. Náročnost na programování

Všechny tři algoritmy / A, B, C / pracují s jednoduchým a snadno interpretovatelným matematickým aparátem, výpočty lze automatizovat pomocí číslicového počítače s poměrně jednoduchými programy. Použití těchto metod nevyžaduje osvojení speciálních teoretických znalostí, a proto je lze

doporučit pro praktické použití.

Potřebný počet instrukcí programu k realizaci algoritmů udává následující tabulka.

ALGORITMUS	POČET INSTRUKCÍ
A	18
B	34
C	110

Do počtu instrukcí je zahrnuta pouze realizace vlastního algoritmu / vyloučeny instrukce čtení, tisku, prázdné příkazy /. U algoritmu A je třeba ještě 20 instrukcí pro přípravu podmínkových rovnic / vektoru levých a matice pravých stran /. Protože tyto instrukce nejsou součástí iteračního algoritmu, nejsou v tabulce uvedeny.

6.3. Doba výpočtu

Potřebný strojový čas k výpočtu podle jednotlivých algoritmů je uveden v následujícím přehledu:

ALGORITMUS	DOBA VÝPOČTU / 5 akč. zásahů za 1 sec./
A	0.04
B	1
C	5

Zdálo by se, že nejnáročnější z hlediska spotřeby strojového času je algoritmus C. Ve skutečnosti tomu tak není; nejrychlejší je algoritmus B. U Kaczmarzova algoritmu A je problematika časové náročnosti spojena s rychlostí konvergence. V tabulce je u algoritmu A uveden čas potřebný k provedení výpočtu v 1 iteračním kroku, ale výpočet 1000 iterací trvá 30-60 sec, 30 000 iterací již 30-45 minut podle

počtu podmínkových rovnic. Pro výpočet v reálném čase je třeba algoritmus A doplnit ještě algoritmem na urychlení konvergence. Návrhy na urychlení algoritmu A jsou uvedeny v kapitole 5.

6.4. Přesnost

Přesnost dosažených výsledků a porovnání metod je uvedeno v tabulce a grafu.

POROVNÁVANÉ ALGORITMY	Tab, Obr.	strana
A, B	TabII	23
A, B	Obr.3-5	24

Připomeňme, že bylo zkoumáno jak výsledky obou algoritmů A, B vyhovují podmínkovým rovnicím / Tab.I, str.21 /.

6.5. Použití

Výhodou Kaczmarzova algoritmu A je jeho široká aplikovatelnost, lze jej použít pro výpočet optimálních akčních zásahů, pro výpočet koeficientů algoritmu řízení. Díky své jednoduchosti našel uplatnění při identifikaci lineárních objektů a při řešení celé řady úloh lineární algebry.

ZÁVĚR

V diplomové práci byl prakticky ověřen Kaczmarzův gradientní iterační algoritmus při návrhu řídicích algoritmů. Celkem bylo sestaveno šest programů, které řeší výpočet Kaczmarzovým algoritmem nebo slouží k porovnání získaných hodnot.

Z výsledků programů a teoretického rozboru vyplývá, že klasický Kaczmarzův algoritmus lze pro návrh řídicích algoritmů použít v případě soustav nižších řádů. Jedná-li se o soustavy vyšších řádů, je třeba použít ještě algoritmy doplňující - na urychlení konvergence, tak jak jsou navrženy v kapitole 5. této práce.

Jinak má Kaczmarzův algoritmus řadu výhod - je velmi přesný, nenáročný na kapacitu paměti počítače i na programování.

V jednom iteračním kroku má nepatrné nároky na čas počítače. Byl by proto vhodný k použití na mikroprocesorech.

Problém použití tohoto algoritmu v praxi je však spojen s problémem rychlosti konvergence iteračního postupu. Podaří-li se najít takový algoritmus, který by konvergenci iteračního výpočtu urychlil natolik, že by výsledků bylo dosahováno řádově do 10^3 iteračních kroků, pak lze tímto algoritmem navrhovat řízení v reálném čase - řízení technologických procesů. Zatím se používá při řešení úloh lineární algebry a při identifikaci dynamických soustav.

Závěrem děkuji vedoucímu diplomové práce Prof. Ing. Bořivoji Hanušovi, CSc za cenné rady a připomínky.

SEZNAM POUŽITÉ LITERATURY

- / 1 / Hanuš, B.: Optimalizace systému řízení
/ skripta /
Liberec, VŠST 1978
- / 2 / Hanuš, B.: Gradientové metody v automatizaci
/ výzkumná zpráva KTK 0078 /
Liberec, VŠST 1981
- / 3 / Himmelblau, D. M.: Příkladnoje nelinejnoje programirovaniye
/ překlad z angličtiny /
Moskva, Mir 1975
- / 4 / Hanuš, B., - Janeček, B., - Kracík, V., - Mikulecký, B.:
Metoda návrhu víceparametrových číslicových
regulačních systémů
/ metodická příručka /
Praha, ÚVVTŘ 1981
- / 5 / Hanousek, V.: Etalony přechodových procesů
/ výzkumná zpráva KTK 0070 /
Liberec, VŠST 1980
- / 6 / Janeček, B.: Program pro výpočet algoritmu řízení pro
determinované vstupní poruchy a ověření
algoritmu simulací regulačních pochodů při
zapojeném regulátoru
/ zpráva P - 0285 /
Liberec, VŠST 1981

- / 7 / Markl, J.: Průběžná identifikace jako optimálně
řízený proces
Automatizace 24 1981 , č. 10, str. 252 -256
- / 8 / Markl, J.: Urychlení konvergence a zvýšení přesnosti
odhadu jednoduchého gradientního algorit-
mu průběžné identifikace
Zborník prednášok IV. celoštátnej konferen-
cie ASR TP'80, ČSVTS - ÚRAP, Žilina, 1980
- / 9 / Kaczmarz, S.: Angenäherte Auflösung von Systemen
Linear Gleichungen.
Bul. Inter. Acad. Pol. Cl. Sci. Math.
Natur., Ser. A., 1937
- / 10 / Wernstedt, J.: Zur Parameterchätzung statischer
Systeme mittels rekursiver Verfahren
Messen - Steuern- Regeln, 1973, No 1
- / 11 / Avedjan, E. D., - Cypkin, Ja. Z.: Obobščenyj algoritm
Kačmaža
Avtomatika i telemekhanika, 1979, No 1

SEZNAM PŘÍLOH

č. přílohy	název
1	Popisy programů
2	Programy
3 - 9	Výsledky programů

PŘÍLOHA 1

^o
POPISY PROGRAMU

počet stran 13

1	Popis programu AKC	1 - 2
2	Popis programu MING	3 - 4
	Blokové schema	5
3	Popis programu RR	6
	Blokové schema	7
4	Popis programu MPK	8
5	Popis programu SIM	9
6	Popis programu RTZ	10 - 12
	Blokové schema logické ná vaznosti programů	13

① PROGRAM - VÝPOČET OPTIMÁLNÍCH AKČNÍCH ZÁSAHŮ GRADIENTOVOU
METODOU S VYUŽITÍM KVADRATICKÉHO KRITÉRIA - AKC

Program řeší výpočet optimálních akčních zásahů jednorozměrné soustavy, známe-li typ a místo vstupu poruchy. V našem případě jde o poruchu ve tvaru jednotkového skoku na vstupu do regulované soustavy.

Popis vstupních dat do programu AKC:

Data jsou do programu načítána v tomto pořadí:

READ READ, NR, N, P, KAPA, T, KO, NI;

DO 1 I = 1, N;

1 READ READ, YD(I);

DO 22 I = 1, N;

22 READ READ, F(I);

Význam jednotlivých dat:

NR / INTEGER / ... řád soustavy

N / INTEGER / ... horní mez součtu v kvadratickém kritériu

P / REAL / váhový koeficient v kvadratickém kritériu

KAPA / REAL / koeficient rychlosti konvergence

T / REAL / časová konstanta

KO / INTEGER / ... konstanta určující po kolika iteracích
bude prováděn tisk výsledků

NI / INTEGER / ... konstanta určující celkový počet iterací

YD(I) / REAL / ... vektor hodnot přechodové charakteristiky
dané soustavy - odezva na determinovanou
poruchu ve tvaru jednotkového skoku

F(I) / REAL / vektor hodnot přechodové charakteristiky
dané soustavy - odezva soustavy na jednot-
kový akční zásah

Další důležité identifikátory programu AKC:

Y(KL) / REAL / vektor levých stran podmínkových rovnic

U(KL, K) / REAL / .. matice pravých stran podmínkových rovnic

DU(1, K) / REAL / .. výchozí odhad akčních zásahů

DU(2, K) / REAL / .. následující odhad akčních zásahů

KL / INTEGER / číslo podmínkové rovnice

② PROGRAM - VÝPOČET AKČNÍCH ZÁSAHŮ GRADIENTOVOU METODOU
V MINIMÁLNÍM POČTU KROKŮ - MING

Data jsou do programu načítána instrukcemi:

READ READ, NR, P, KAPA, T, MK, KO, NI;

DO 22 I = 1, MK;

22 READ READ, F(I);

Význam jednotlivých identifikátorů je stejný, jako v programu AKC.

MK / INTEGER / ... konstanta udávající počet načítaných
hodnot přechodové charakteristiky.

Varianty programu:

1/ při použití klíče č. 3 se iteračním postupem nevypočítávají hodnoty akčních zásahů, ale hodnoty oprav akčních zásahů. Předem je nutno vypočítat opravy podmínkových rovnic a zavést do programu ve formě vstupních dat následující instrukcí: READ READ, Y(KL);

2/ použitím klíče č. 5 měníme výchozí odhad $\Delta \underline{u}(0) = 0$ / na libovolný počáteční odhad, který načteme z děrné pásky:
READ READ, DU(1, K);

3/ stlačením klíče č. 8 provádíme iterační výpočet s řízením řešení jednotlivých podmínkových rovnic. K řízení jsou potřeba tyto parametry:

READ READ, KN;

DO 83 IT = 1, KN;

READ READ, KLO(IT);

KN / INTEGER / ... konstanta určující počet prvků v cyklu
řízení

KLO(IT) / INTEGER / ... vektor posloupnosti čísel podmínko-

vých rovnic.

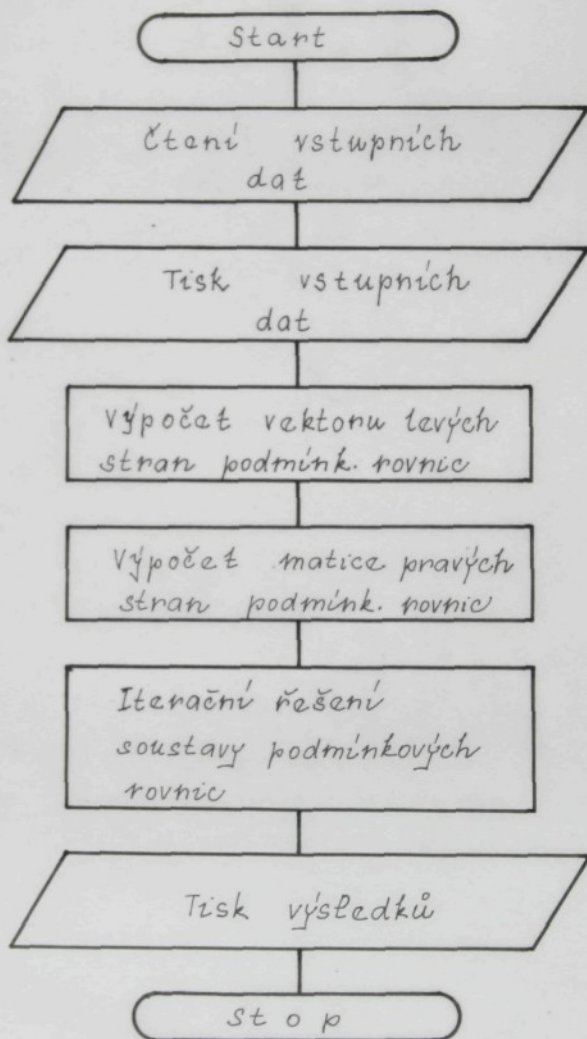
K vygenerování čísla podmínkové rovnice, která se má zapojit do řešení, slouží podprogram:

SUBROUTINE PODM (KA, KN, KLO, KL) ;

kde:

KA / INTEGER / ... udává číslo právě probíhající iterace

KL / INTEGER / ... určuje číslo podmínkové rovnice, která se má zapojit do iteračního výpočtu .

O
OBECNÉ BLOKOVÉ SCHEMA PROGRAMU AKC, MING

③ PROGRAM - VÝPOČET K⁰EFICIENTŮ ALGORITMU ŘÍZENÍ - RR

Data jsou do programu načítána instrukcemi:

READ READ, NR, N, KAPA, KO, NI, EPS;

DO 2 I = 1, N;

2 READ READ, F(I);

DO 3 I = 1, N;

3 READ READ DU(I) ;

Význam jednotlivých identifikátorů je stejný jako v programu AKC.

N / INTEGER / ... počet načtených hodnot přechodové charakteristiky

EPS / REAL / konstanta

DU(I) / REAL / .. vektor hodnot optimálních akčních zásahů

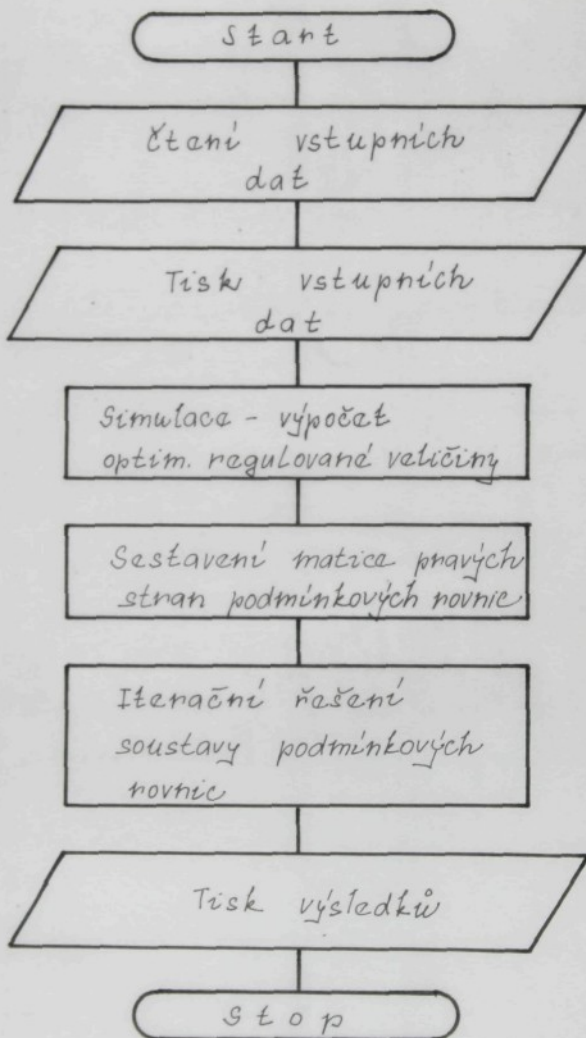
Další důležité identifikátory:

DU(KL) / REAL / . vektor hodnot optimálních akčních zásahů,
tj. vektor levých stran podmínkových rovnic

O(KL,K) / REAL / . matice pravých stran podmínkových rovnic

YO(J) / REAL / vektor optimálních regulovaných veličin

OBEZNÉ BLOKOVÉ SCHEMA PROGRAMU RR



④ PROGRAM - VÝPOČET AKČNÍCH ZÁSAHŮ V MINIMÁLNÍM POČTU
REGULAČNÍCH KROKŮ - MPK

Odvození algoritmu je v / 5 /. Pro objasnění uvádíme vzorec na výpočet akčních zásahů v konečném počtu kroků:

$$\Delta W_i = (-1)^i \left[\binom{N}{i-1} \bar{e}^{-i+1} + \binom{N}{i} \bar{e}^{-i} \right] \quad i=1 \text{ až } N-1$$

$$\Delta W_N = (-1)^N \cdot \bar{e}^{-N+1}$$

kde je

N řád soustavy

M počet akčních zásahů

i pořadí akčních zásahů

Popis dat do programu:

READ READ, N;

N / INTEGER / řád soustavy

Popis podprogramu a vnější funkce:

SUBROUTINE FACT (NA, NL);

podprogram na výpočet faktoriálu z čísla NL

NA / INTEGER / číslo ze kterého se faktoriál vypočítává

NL / INTEGER / faktoriál z čísla NA

FUNCTION KLC (KA, KB)

vnější funkce pro výpočet kombinačního čísla z čísel:

$$\binom{KA}{KB}$$

⑤ PROGRAM - SIMULACE REGULAČNÍHO POCHODU - SIM

Program slouží k ověření správnosti vypočtených řídicích algoritmů.

Způsob zadávání dat je shodný s programem / 6 /.

Navíc jsou do programu načítány koeficienty řídicího algoritmu-instrukcemi:

DO 36 I= 1, MR;

DO 36 I= 1, NXY;

36 READ READ, AB(I,J);

⑥ PROGRAM - URYCHLENÍ KONVERGENCE PŘI VÝPOČTU AKČNÍCH
 ZÁSAHŮ V MINIMÁLNÍM POČTU KROKŮ GRADIENTOVOU
 METODOU - RTZ

Popis vstupních dat i označení je stejné jako u programu MING.
 Budou uvedeny jen odlišnosti a význam jednotlivých podprogramů.
 Vhodnou úpravou lze tento doplňující algoritmus použít v kterémkoliv z uváděných programů, řešící výpočet Kaczmarzovým algoritmem, tj. MING, RR, AKC.

Do programu jsou navíc zavedena tato data:

DO 39 K= 1,M;

39 READ READ, NC(K);

READ INTYP, KS, KPS;

NC(K) / INTEGER /posloupnost čísel podmínkových rovnic

M / INTEGER /celkový počet podmínkových rovnic

KS / INTEGER /počet skupin

KPS / INTEGER /počet prvků ve skupině

Př. Chceme postupně dosazovat podmínkové rovnice za sebou
 v tomto pořadí. Střídání podmínkových rovnic zadáme
 posloupností NC(K), konstantami KS, KPS, takto:

12121212//32323232//34343434//14141414//.....

KS= 4 NC(1) = 1

KPS= 8 NC(2) = 2

NC(3) = 3

NC(4) = 4

Charakteristika jednotlivých podprogramů:

SUBROUTINE GENER (IB, KS, NC, KL, KA, IQ);

podprogram slouží k vygenerování čísla podmínkové rovnice
KL z čísla právě probíhající iterace KA.

SUBROUTINE TEST (IB, IC, KPS, KLC);

slouží k testování, zda má dojít k výpočtu ve směru či nikoliv
Výstupem je proměnná, KLC, která nabývá dvou hodnot:

pokud $KLC = 1$ - vyvolá se podprogram GRADS

pokud $KLC \neq 1$ - vyvolá se podprogram GRAD

SUBROUTINE GRAD(D1, D2, U, Y, N, KL, KAPA, KA);

provádí výpočet podle vzorce, odvozeného v kap. 2.2, vzorec
má tvar:

$$\Delta \underline{w}(k) = \Delta \underline{w}(k-1) + \mu_k \cdot \frac{y(k) - \Delta \underline{w}^T(k-1) \cdot \underline{w}(k)}{\underline{w}^T(k) \cdot \underline{w}(k)} \cdot \underline{w}(k)$$

formální parametry podprogramu mají tento význam:

D1 $\sim \Delta \underline{w}(k-1)$

D2 $\sim \Delta \underline{w}(k)$

KAPA $\sim \mu_k$

Y(KL) $\sim y(k)$

T1 $\sim$ skalární součin $\Delta \underline{w}^T(k-1) \cdot \underline{w}(k)$

T2 $\sim$ skalární součin $\underline{w}^T(k) \cdot \underline{w}(k)$

U(KL, K)

SUBROUTINE GRADS(D1, D2, U, Y, N, KL, APA, KA, PAM);

provádí výpočet podle vzorce, uvedeného v kap. 5, vzorec
má tvar:

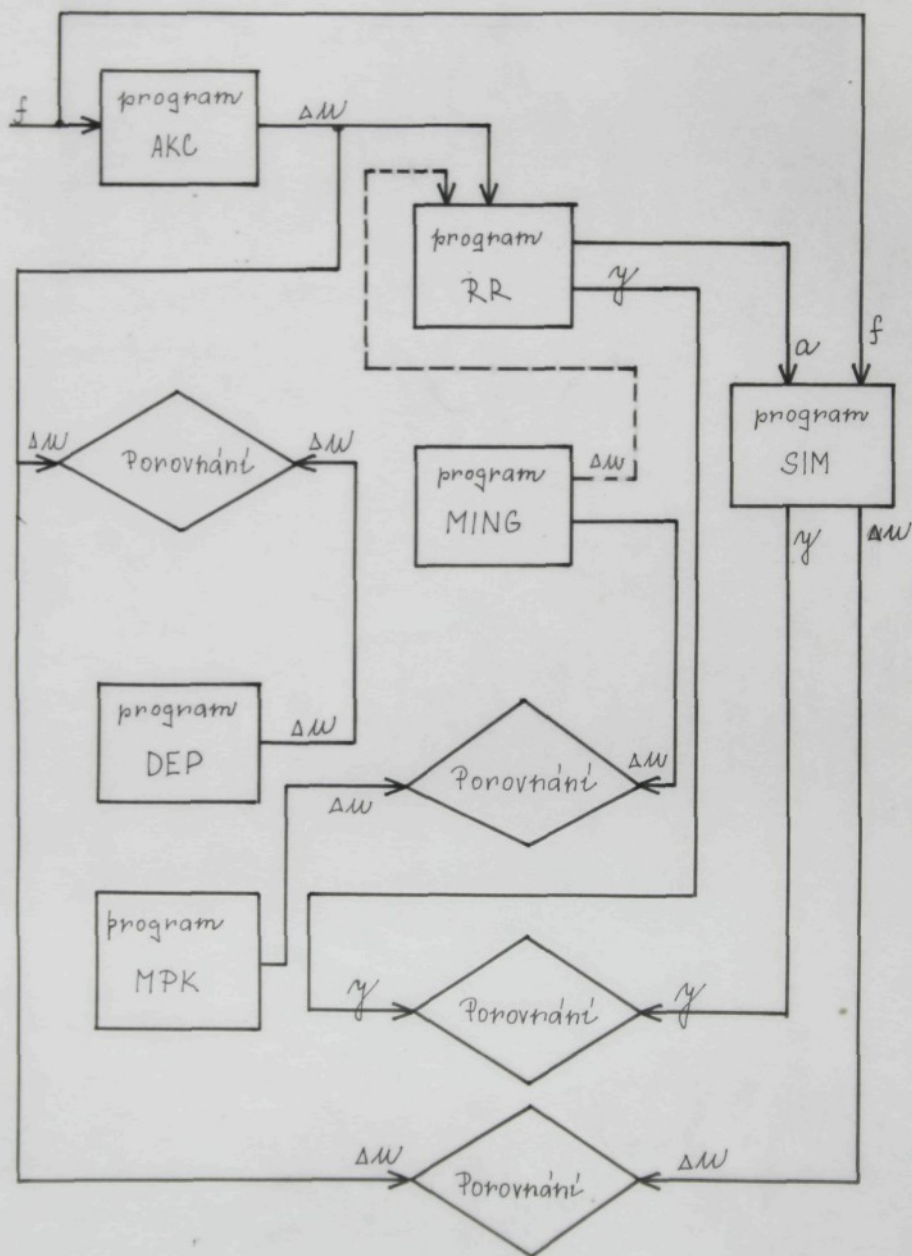
$$\Delta \underline{w}(k) = \Delta \underline{w}(k-1) + \mu_k \cdot \frac{y(k) - \Delta \underline{w}^T(k-1) \cdot \underline{w}(k)}{\underline{w}^T(k) \cdot \underline{w}(k)} \cdot \underline{w}(k)$$

SUBROUTINE PAMET(IB, D2, IC, PAM, N);

určuje, zda je nutno si vypočtenou hodnotu vektoru akčních
zásahů pamatovat. Jestliže ano, pak provádí uložení této

hodnoty $D2(K)$ na proměnnou $PAM(K)$:

BLOKOVÉ SCHEMA LOGICKÉ NÁVAZNOSTI PROGRAMU^o



PŘÍLOHA 2

PROGRAMY

počet stran 23

Program AKC	1 - 3
Program MING	4 - 6
Program RR	7 - 9
Program MPK	10 - 11
Program SIM	12 - 19
Program RTZ	20 - 23

MENU PROGRAMU: AKC
KOMENTAR:

```
1
2      PROGRAM VYPOCET OPTIMALNICH AKCNICH ZASAHU GRADIENT. METODOU;
3      DIMENSION YD(20),F(20),Y(20),U(20,20),DU(3,20);
4      REAL KAPA;
5      READ READ;
6      NR=N,P,KAPA,T,KO,NI;
7      WRITE LINE,200;
8      WRITE LINE,201,NR;
9      WRITE LINE,100,N,P,KAPA,T;
10     WRITE LINE,108;
11     WRITE LINE,103;
12     WRITE LINE,108;
13     DO 1 I=1,N;
14     IREAD READ,YD(I);
15     DO 22 I=1,N;
16     22READ READ,F(I);
17     DO 23 I=1,N;
18     WRITE LINE,104,YD(I);
19     WRITE LINE,111,F(I);
20     23 WRITE LINE,108;
21     WRITE LINE,108;
22     WRITE LINE,109;
23     WRITE LINE,108;
24     IF KEY(3),81,82;
25     81 DO 35 KL=1,N;
26     READ READ,Y(KL);
27     35 CONTINUE;
28     COMMENT Y= VEKTOR LEVYCH STRAN
29     PODMINKOVYCH ROVNIC;
30     GOTO 36;
31     82 DO 2 KL=1,N;
32     S=0.0E0;
33     DO 3 I=KL,N;
34     IA=I+1-KL;
35     3 S=S+(YD(I)*F(IA));
36     Y(KL)=-S;
37     WRITE LINE,104,Y(KL);
38     WRITE LINE,108;
39     2 CONTINUE;
40     36 CONTINUE;
41     COMMENT U= MATICE PRAVYCH STRAN
42     PODMINKOVYCH ROVNIC;
43
44     M=0;
45     DO 7 KL=1,N;
46     DO 8 K=1,N;
47     SU=0.0E0;
48     IF (KL-K),9,9,10;
49     9L=K;
50     GOTO 110;
51     10 L=KL;
52     110 CONTINUE;
53     DO 11 I=L,N;
54     IA=I+1-K;
55     IB=I-M;
```



```

56 SU=SU+F(1B)*F(1A);
57 11 CONTINUE;
58 SA=SU;
59 IF (KL-K),14,15,14;
60 14 U(KL,K)=SA;
61 GOTO 8;
62 15 U(KL,K)=SA+P;
63 8 CONTINUE;
64 M=M+1;
65 7 CONTINUE;
66 WRITE LINE,108;
67 WRITE LINE,107;
68 WRITE LINE,108;
69 IF KEY(5),68,69;
70 68 DO 34 K=1,N;
71 READ READ,DU(1,K);
72 34 CONTINUE;
73 GOTO 500;
74 69 DU 16 K=1,N;
75 DU(1,K)=0.0;
76 16 CONTINUE;
77 500 CONTINUE;
78 COMMENT ITERACNI VYPOCET;
79 DU 17 KA=1,NI;
80 T1=0.0E0;
81 T2=0.0E0;
82 KL=KA-(KA/N*N);
83 IF ( KL),48,49,48;
84 49 KL=N;
85 48 CONTINUE;
86 DO 19 K=1,N;
87 T1=T1+DU(1,K)*U(KL,K);
88 19 T2=T2+U(KL,K)*U(KL,K);
89 KK=KA/KO*KO;
90 DO 18 K=1,N;
91 IF (N-5),93,94,94;
92 93 NC=N;
93 GOTO 98;
94 94 NC=5;
95 98KP=K/NC*NC;
96 IF (T2),85,86,85;
97 85DU(2,K)=DU(1,K)+(KAPA*(Y(KL)-T1)/T2)*U(KL,K);
98 86IF (KA-KK),55,54,55;
99 54 CONTINUE;
100 IF (K-1),56,57,56;
101 57 WRITE LINE,105,KA;
102 56WRITE LINE,102,DU(2,K);
103 IF (K-KP),59,58,59;
104 58 WRITE LINE,108;
105 WRITE LINE,113;
106 IF (K-N),18,60,18;
107 60 WRITE LINE,108;
108 59 CONTINUE;
109 55 CONTINUE;
110 18 CONTINUE;
111 DO 13 K=1,N;
112 DU(1,K)=DU(2,K);
113 13 CONTINUE;
114 17 CONTINUE;
115 101 FORMAT (E14.7);
116 102 FORMAT (5X,E14.7);
117 200 FORMAT ('VYPOCET OPTIM. AKCNICH ZASAHU GRADIENTOVOU METODOU
118 *****');
119 201 FORMAT ('
120 RAD SOUSTAVY ',I3);

```

```

121 100 FORMAT('
122 N=','13','P=','F5.2','KAPA=','F5.2','T=','F5.2');
123 103 FORMAT ('DATA');
124 105 FORMAT (I7);
125 111 FORMAT (10X,F14.7);
126 104 FORMAT(F14.7);
127 113 FORMAT(7X);
128 106 FORMAT('MEZIVYPUCET U(KA,K)');

129 107 FORMAT('C.ITER. KROK                OPTIMAL.AKCNI ZASAHY
130 =====');
131 109FORMAT('MEZIVYPUCET Y');
132 108 FORMAT('
133 ');
134 END;
135 END OF PROGRAM;
136

```

```

137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

 MENU PROGRAMU: MING
 KOMENTAR:

```

1
2      PROGRAM VYPOCET OPTIMALNICH AKCNICH ZASAHU
3  V MINIMALNIM POCTU KROKU GRADIENT.METODOU;
4  DIMENSION YD(20),F(20),Y(20),U(20,20),DU(3,20);
5  DIMENSION KLO(50);
6  REAL KAPA, KONS;
7  READ READ;
8  NR,P,KAPA,Y,MK,KO,NI;
9  WRITE LINE, 108;
10 WRITE LINE, 100;
11 WRITE LINE,101,NR,P,KAPA,T;
12 WRITE LINE,108;
13 WRITE LINE,103;
14 WRITE LINE,108;
15 DO 22 I=1,MK;
16 22READ READ,F(I);
17 DO 23 I=1,MK;
18 WRITE LINE,111,F(I);
19 23 WRITE LINE,108;
20 WRITE LINE,108;
21 M=NR+1;
22 IF KEY (3),81,82;
23 81 DO 35 KL=1,M;
24 READ READ ,Y(KL);
25 35 CONTINUE;
26 WRITE LINE,205;
27 WRITE LINE,108;
28 205 FORMAT('
29 VYPOCET UPRAV AKC.ZASAHU');
30 GOTO 36;
31 COMMENT Y= VEKTOR LEVYCH STRAN
32 PODMINKOVYCH ROVNIC;
33 82 KF=M;
34 DO 2 KL=1,M;
35 Y(KL) =-F(KF);
36 KF=KF+1;
37 2 CONTINUE;
38 36 NA=NR;
39 DO 3 KL=1,M;
40 KB=NA;
41 COMMENT U= MATICE PRAVYCH STRAN
42 PODMINKOVYCH ROVNIC;
43 DO 4 K=1,M;
44 IF (NR),62,62,61;
45 62 U(KL,K)=F(M);
46 GOTO 4;
47 61IF (KB),26,26,25;
48 26 U(KL,K)=0.0;
49 GOTO 4;
50 25U(KL,K)=F(KB);
51 KB=KB-1;
52 4 CONTINUE;
53 NA=NA+1;
54 3 CONTINUE;
55 IF KEY(15),98,99;

```

```

56 98 WRITE LINE,201;
57 WRITE LINE,108;
58 DO 87 KL=1,M;
59 87 WRITE LINE,200,Y(KL);
60 WRITE LINE,108;
61 WRITE LINE,108;
62 DO 88 K=1,M;
63 DO 89 KL=1,M;
64 89 WRITE LINE,202,U(KL,K);
65 88 WRITE LINE,108;
66 200 FORMAT('
67 ',E14.7);
68 201 FORMAT('
69 MEZIVYSLEDKY');
70 202 FORMAT(2X,F10.7);
71 99WRITE LINE,108;
72 WRITE LINE,107;
73 WRITE LINE,108;
74 IF KEY(5),68,69;
75 68 DO 34 K=1,M;
76 READ READ,DU(1,K);
77 34 CONTINUE;
78 WRITE LINE,206;
79 WRITE LINE,108;
80 206 FORMAT('
81 VYPOCET S ODHADEM');
82 GOTO 500;
83 69 DO 16 K=1,M;
84 DU(1,K)=0.0;
85 16 CONTINUE;
86 500 CONTINUE;
87 IF KEY(8),86,90;
88 86 READ READ,KN;
89 DO 83 IT=1,KN;
90 READ READ,KLO(IT);
91 83 CONTINUE;
92 WRITE LINE,208;
93 WRITE LINE,108;
94 90 CONTINUE;
95 COMMENT ITERACNI VYPOCET;
96 DO 17 KA=1,NI;
97 T1=0.0E0;
98 T2=0.0E0;
99 IF KEY(8),84,85;
100 84 CALL PODM(KA,KN,KLO,KL);
101 208 FORMAT('
102 VYPOCET S RIZENIM PODMINKOVYCH ROVNIC');
103 GOTO 48;
104 85 KL=KA-(KA/M*M);
105 IF (KL),48,49,48;
106 49 KL=M;
107 48 CONTINUE;
108 DO 19 K=1,M;
109 T1=T1+DU(1,K)*U(KL,K);
110 T2=T2+U(KL,K)**2;
111 19 CONTINUE;
112 KONS =KAPA*(Y(KL)-T1)/T2;
113 KK=KA/KO*KO;
114 DO 18 K=1,M;
115 DU(2,K)=DU(1,K)+KONS*U(KL,K);
116 COMMENT RIZENI TISKU PO KO-ITERACICH;
117 IF (KA-KK),55,54,55;
118 54 CONTINUE;
119 IF (KA-2500),71,72,71;
120 72 KO=100;

```



```

121 71 IF (K-1),56,57,56;
122 57 WRITE LINE,105,KA;
123 WRITE LINE,220,KL;
124 220 FORMAT(I2);
125 56 WRITE LINE,102,DU(2,K);
126 IF (K-M),59,58,59;
127 58 WRITE LINE,108;
128 WRITE LINE,108;
129 59 CONTINUE;
130 55 CONTINUE;
131 18 CONTINUE;
132 DO 13 K=1,M;
133 DU(1,K)=DU(2,K);
134 13 CONTINUE;
135 17 CONTINUE;
136 102 FORMAT(5X,E14.7);
137 100 FORMAT (
138 VYPOCET OPTIM. AKCNICH ZASAHU V MIN. POCTU KROKU GRAD. METODOU
139 *****);
140 101 FURMAT (
141 RAD SOUSTAVY N=',I3,
142 P =',F5.2,' KAPA=',F5.2,' T=',F5.2);
143 103 FORMAT (
144 HODNOTY PŘECHODOVÉ CHARAKTERISTIKY);
145 105 FORMAT(I7);
146 111 FORMAT (25X,E14.7);
147 107 FORMAT(' CÍSLA KROKŮ          AKCNÍ ZASAHY
148 =====');
149 108 FORMAT('
150 ');
151 END;
152 SUBROUTINE PODM(KA,KN,KLO,KL);
153 DIMENSION KLO(50);
154 I=KA-(KA/KN*KN);
155 IF (I),2,2,3;
156 2 I=KN;
157 3 KL=KLO(I);
158 RETURN;
159 END;
160 END OF PROGRAM;
161

```

HENO PROGRAMU: RR
KUMENIAR:

```
1  PROGRAM RIDICI ALGORITMUS;  
2  DIMENSION YD(20),F(20),DU(20),U(20,20),A(3,20),YD(20);  
3  REAL KAPA;  
4  READ READ,NR,N,KAPA,KU,NI,EPS;  
5  N1=N+1;  
6  DO 2 I=1,N;  
7    2 READ READ,F(I);  
8    DO 3 I=1,N;  
9      3 READ READ,DU(I);  
10   WRITE LINE,100;  
11   WRITE LINE,250,NR,KAPA;  
12   WRITE LINE,108;  
13   N4=2*NR+1;  
14   WRITE LINE,101,N4;  
15   WRITE LINE,108;  
16   DO 4 I=1,N;  
17     WRITE LINE,103,F(I);  
18     WRITE LINE,104,DU(I);  
19     4 WRITE LINE,108;  
20     WRITE LINE,108;  
21     COMMENT VYPUCET YO OPTIMALNI;  
22     COMMENT DU= VEKTOR LEVYCH STRAN  
23     PODMINKOVYCH RUVNIC;  
24     WRITE LINE,106;  
25     WRITE LINE,108;  
26     DO 6 J=1,N;  
27       N2=J-1;  
28       IF (N2),32,32,33;  
29       32 N2=1;  
30       33S=0.0;  
31       DO 7 K=1,N2;  
32         J1=J-K;  
33         IF (J1),30,30,31;  
34         30 S=0.0;  
35         GOTO 7;  
36         31 S=S+F(J1)*DU(K);  
37       7 CONTINUE;  
38       B=S+F(J);  
39       YD(J)=B;  
40       WRITE LINE,102,YD(J);  
41       WRITE LINE,108;  
42       6 CONTINUE;  
43       COMMENT U= MATICE PRAVYCH STRAN  
44       PODMINKOVYCH RUVNIC;  
45       N3=NR+1;  
46       DO 8 KL=1,N4;  
47         DO 9 K= 1,N4;  
48         IF (K-N3),10,10,11;  
49         11 KD=K-N3;  
50         K2=KL-KD;  
51         IF (K2),72,72,73;  
52         72 U(KL,K)=0.0;  
53         GOTO 9;  
54         73 U(KL,K)=DU(K2);  
55         GOTO 9;
```

```

56 10 KY=K-1;
57 K1=KL-KY;
58 IF (K1),74,74,75;
59 74 O(KL,K)=0.0;
60 GOTO 9;
61 75 O(KL,K)=Y0(K1);
62 9 CONTINUE;
63 8 CONTINUE;
64 WRITE LINE,201;
65 WRITE LINE,108;
66 DO 81 K=1,N4;
67 DO 82 KL=1,N4;
68 82 WRITE LINE,200,O(KL,K);
69 81 WRITE LINE,108;
70 200 FORMAT(2X,F10.7);
71 201 FORMAT('
72 VEKTORY O');
73 COMMENT A= VEKTOR KOEF.RIDICIHO ALGORITHMU;
74 DO 15 K=1,N4;
75 15 A(1,K)=0.0;
76 WRITE LINE,107;
77 WRITE LINE,108;
78
79 COMMENT ITERACNI VYPOCET;
80 DO 12 KA=1,NI;
81 T1=0.0;
82 T2=0.0;
83 KL=KA-(KA/N4*N4);
84 IF (KL),48,49,48;
85 49 KL=N4;
86 48 CONTINUE;
87 DO 13 K=1,N4;
88 T1=T1+A(1,K)*O(KL,K);
89 T2=T2+O(KL,K)*O(KL,K);
90 13 CONTINUE;
91 IF (T2-EPS),83,83,84;
92 83 T2=EPS;
93 84 KK=KA/KO*KO;
94 DO 14 K=1,N4;
95 IF (N4-5),93,94,94;
96 93 NC=N4;
97 GOTO 98;
98 94 NC=5;
99 98 KP=K/NC*NC;
100 A(2,K)=A(1,K)+(KAPA*(DO(KL)-T1)/T2)*O(KL,K);
101 COMMENT RIZENI TISKU;
102 IF (KA-KK),55,54,55;
103 54 CONTINUE;
104 IF (K-1),56,57,56;
105 57 WRITE LINE,109,KA;
106 WRITE LINE,109,KL;
107 56WRITE LINE,103,A(2,K);
108 IF (K-KP),59,58,59;
109 58 WRITE LINE,108;
110 WRITE LINE,113;
111 59 IF (K-N4),14,60,14;
112 60 WRITE LINE,108;
113 55 CONTINUE;
114 14 CONTINUE;
115 DO 20 K=1,N;
116 A(1,K)=A(2,K);
117 20 CONTINUE;
118 12 CONTINUE;
119 100 FORMAT ('VYPOCET KOEF.RIDICIHO ALGORITHMU GRADIENT. METODOU
120 *****') ;

```

```

121 250 FORMAT('
122 SOUSTAVA NR=',I3,5X,'KAPA=',F5.2,'
123 ');
124 101 FORMAT('POCET KOEFICIENTU=',I3,'
125 ');
126 113 FORMAT(12X);
127 102 FORMAT(E 14.7);
128 103 FORMAT(5X,E14.7);
129 104 FORMAT(50X,E14.7);
130 105 FORMAT (2I5);
131 106 FORMAT ('VYPOCET Y OPTIMALNI ');
132 107 FORMAT ( 'CISLO ITER.          KOEF.ALGORITMU
133 =====');
134 109 FORMAT (I6);
135 108 FORMAT('
136 ');
137 END;
138 END OF PROGRAM;
139

```

MENU PROGRAMU: MPK

KOMENTAR:

```
1 PROGRAM VYPOCET AKC.ZASAHU V MIN.POCTU KROKU;
2 DIMENSION U(20);
3 WRITE LINE,103;
4 READ READ,N;
5 WRITE LINE,102,N;
6 WRITE LINE,104;
7 M=N+1;
8 DO 1 I=1,N;
9   R1=-I+1;
10  R2=-I;
11  RC=KCL(N,I-1);
12  RL=KCL(N,I);
13  R5=(-1)**I;
14  U(I)=R5*(RC*EXP(R1)+RL*EXP(R2));
15  WRITE LINE,100,U(I);
16  1 CONTINUE;
17  R3=-M+1;
18  R4=(-1)**M;
19  U(M)=R4*EXP(R3);
20  WRITE LINE,100,U(M);
21  S=0.0;
22  DO 2 I=1,M;
23    S=S+U(I);
24    WRITE LINE,108;
25    WRITE LINE,101,S;
26    WRITE LINE,108;
27    100 FORMAT('
28    ',E14.7);
29    104 FORMAT('
30    *****');
31    101 FORMAT ('SUMA=',F5.2);
32    102 FORMAT ('
33    SOUSTAVA RADU N=',I5);
34    103 FORMAT('
35    OPTIMALNI AKCNI ZASAHY V MIN. POCTU KROKU
36    =====');
37    108 FORMAT ('
38    ');
39  END;
40
41
42
43
44
45
46 COMMENT VYPOCET KOMB.CISLA;
47 FUNCTION KCL(KA,KB);
48 KC=KA-KB;
49 CALL FACT (KC,LC);
50 CALL FACT (KA,LA);
51 CALL FACT (KB,LB);
52 KCL=LA/(LC*LB);
53 RETURN;
54 END;
```

```

56 SUBROUTINE FACT (NA,NL);
57 IF (NA) 2,3,3;
58 3 NL=1;
59 4 IF (NA-1) 5,5,6;
60 6 NL=NL*NA;
61 NA=NA-1;
62 GOTO 4;
63 5 RETURN;
64 2 WRITE LINE,101;
65 101 FORMAT('FACTOR,NENI DEF. ');
66 END;
67 END OF PROGRAM;
68
69

```

```

70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

MENO PROGRAMU: SIM22

KOMENTAR:

```
1
2 PROGRAM SIMULACE REGULACNIHO POCHODU;
3 DIMENSION F(3,40),SF(40,3),SFF(40,40),FF(40,40),A(40,40),Y(40,3),
4      W(3,1),AW(40,3),YW(40,1),NX(2,40),AB(3,40),FR(3,120),
5      YR(120,1),G(3,3),UK(3),YK(40),FU(3,120),GW(3,3);
6 EQUIVALENCE (SFF(1),FR(1)),(FF(1),YR(1)),(AB(1),AW(1),GW(1));
7
8
9      JR=3;
10     NR=40;
11     NR3=3*NR;
12     READ READ,DT,FK,KR,MR,MZ,MX,NXY,IR,IT;
13
14 IF KEY(15),37,44;
15 44 I1=0;
16 DO 70 I=1,80;
17 70 WRITE PUNCH,201,I1;
18 201 FORMAT(A1);
19 WRITE PUNCH,200,DT,FK,KR,MR,MZ,MX,NXY,IR,IT;
20 200 FORMAT('
21 ',E14.7,', ',E14.7,7(' ',I3));
22
23 37IRK=IR*KR+KR+1;
24 IF (IRK-NR),33,33,34;
25 34 IR=(NR-KR-1)/KR;
26 GOTO 37;
27 33 IF((NU+2)*KR-IRK),42,42,43;
28 43 NU=IRK/KR-2;
29 42WRITELINE,100,DT,FK,KR,MR,MZ,MX,NXY,IR;
30 100 FORMAT('
31 VYPOCET SIMULACE REG. POCHODU
32
33 DT/1=',E14.7,', FK=',E14.7,', KR=',I3,', MR=',I3,', MZ=',I3,', MX=',I3,
34 ', NXY=',I3,', IR=',I3);
35 UK(1)=KR;
36 DTK=DT/UK(1);
37 WRITE LINE,111;
38 111 FORMAT('
39 NX
40 ');
41 DO 21 I=1,2;
42 110 FORMAT('
43 ');
44 DO 20 J=1,NXY;
45 READ READ,NX(I,J);
46 20 WRITE LINE,112,NX(I,J);
47 112 FORMAT(1X,I3);
48 21 WRITE LINE,110;
49
50 DO 36 I=1,MR;
51 DO 36 J=1,NXY;
52 36 READ READ,AB(I,J);
53 WRITE LINE,116;
54 116 FORMAT('
55 MATICE ALGORITHMU A,B');
```

```

56 CALL TIS(B,MR,NXY,JR,NR);
57 WRITELINE,110;
58 COMMENT VYPOCET PRECH.CHAR.;
59 DO 68 I=1,IRK;
60 I1=I-1;
61 CALL VYT(G,I1,DTK,JR,I5);
62 I1=I1*MZ;
63 DO 68 I2=1,MX;
64 J=(I2-1)*MZ;
65 DO 68 I3=1,MZ;
66 I4=I1+I3;
67 I5=J+I3;
68 FF(I,I5)=G(I2,I3)*FK;
69 68 FU(I2,I4)=FF(I,I5);
70 WRITE LINE,144;
71 144 FORMAT('
72 PRECH.CHAR. ');
73 I1=MX*MZ;
74 CALL TIST(FF,IRK,I1,NR,NR);
75 CALL GRAFT(FF,NR,NR,IRK,I1,0,0,0,0,110);
76 WRITELINE,122;
77 122 FORMAT('
78 REGULACE ');
79 IR=IRK-1;
80 I1R=IR+1;
81 MRZ=MR;
82 DO 39 I6=1,1;
83 IF (I6-1),73,73,74;
84 74 MRZ=MZ;
85 73 DO 39 I1=1,MRZ;
86
87
88 CALL VYTG(G,0,I6,1,JR,MX,MR,MZ,FU);
89 DO 22 I=1,I1R;
90 DO 72 J=1,MR;
91 72 Y(I,J)=0.0;
92 DO 22 J=1,MX;
93 22 SF(I,J)=G(J,I1);
94
95
96 I8=KR-1;
97 DO 19 I=1,IR;
98 I4=(I-1)*MR;
99 I7=I*MR;
100 I8=I8+1;
101 DO 65 J=1,MR;
102 65 W(J,1)=0.0;
103 IF (I8-KR),60,61,61;
104 61 I8=0;
105
106
107 DO 24 J=1,NXY;
108 I2=NX(1,J);
109 I3=NX(2,J)*KR+I1R;
110 IF (I2),25,26,26;
111 25 I2=-I2;
112 YW(J,1)=Y(I3,I2);
113 GOTO 24;
114 26 YW(J,1)=SF(I3,I2);
115 24 CONTINUE;
116 CALL NAS(AB,YW,W,MR,NXY,1,JR,NR,1);
117 IF (IT),59,60,60;
118 59 WRITELINE,117;
119 117 FORMAT('
120 (X,Y)T ');

```



```

56 CALL TIS(B,MR,NXY,JR,NR);
57 WRITELINE,110;
58 COMMENT VYPOCET PRECH.CHAR.;
59 DO 68 I=1,IRK;
60 I1=I-1;
61 CALL VYT(G,I1,DTK,JR,I5);
62 I1=I1*MZ;
63 DO 68 I2=1,MX;
64 J=(I2-1)*MZ;
65 DO 68 I3=1,MZ;
66 I4=I1+I3;
67 I5=J+I3;
68 FF(I,I5)=G(I2,I3)*FK;
69 68 FU(I2,I4)=FF(I,I5);
70 WRITE LINE,144;
71 144 FORMAT('
72 PRECH.CHAR. ');
73 I1=MX*MZ;
74 CALL TIST(FF,IRK,I1,NR,NR);
75 CALL GRAFT(FF,NR,NR,IRK,I1,0,0,0,0,110);
76 WRITELINE,122;
77 122 FORMAT('
78 REGULACE ');
79 IR=IRK-1;
80 I1R=IR+1;
81 MRZ=MR;
82 DO 39 I6=1,1;
83 IF (I6-1),73,73,74;
84 74 MRZ=MZ;
85 73 DO 39 I1=1,MRZ;
86
87
88 CALL VYTG(G,0,I6,1,JR,MX,MR,MZ,FU);
89 DO 22 I=1,I1R;
90 DO 72 J=1,MR;
91 72 Y(I,J)=0.0;
92 DO 22 J=1,MX;
93 22 SF(I,J)=G(J,I1);
94
95
96 I8=KR-1;
97 DO 19 I=1,IR;
98 I4=(I-1)*MR;
99 I7=I*MR;
100 I8=I8+1;
101 DO 65 J=1,MR;
102 65 W(J,1)=0.0;
103 IF (I8-KR),60,61,61;
104 61 I8=0;
105
106
107 DO 24 J=1,NXY;
108 I2=NX(1,J);
109 I3=NX(2,J)*KR+I1R;
110 IF (I2),25,26,26;
111 25 I2=-I2;
112 YW(J,1)=Y(I3,I2);
113 GOTO 24;
114 26 YW(J,1)=SF(I3,I2);
115 24 CONTINUE;
116 CALL NAS(AB,YW,W,MR,NXY,1,JR,NR,1);
117 IF (IT),59,60,60;
118 59 WRITELINE,117;
119 117 FORMAT('
120 (X,Y)T ');

```

```

121 CALL TIST(YW,NXY,1,NR,1);
122
123
124 WRITELINE,118;
125 118 FORMAT('
126 YT');
127 CALL TIST(W,MR,1,JR,1);
128 60 DO 27 J=1,IR;
129 DO 67 I2=1,MX;
130 67 SF(J,I2)=SF(J+1,I2);
131 DO 27 I2=1,MR;
132 27 Y(J,I2)=Y(J+1,I2);
133 DO 28 I2=1,MR;
134 I3=I4+I2;
135 YR(I3,1)=W(I2,1);
136 28 Y(IR,I2)=W(I2,1);
137
138
139 CALL VYTF(FR,I,1,JR,NR3,MX,MR,MZ,FU);
140 CALL VYTG(G,I,I6,1,JR,MX,MR,MZ,FU);
141 CALL NAS(FR,YR,W,MX,I7,1,JR,NR3,1);
142 IF(IT),71,62,62;
143 71 I2=I-1;
144 WRITELINE,101,I2;
145 WRITELINE,124;
146 CALL TIS(FR,MX,I7,JR,NR3);
147 WRITELINE,120;
148 120 FORMAT('
149 (F*Y)T');
150 CALL TIST(W,MX,1,JR,1);
151 WRITELINE,119;
152 119 FORMAT('
153 G');
154 CALL TIS(G,MX,MR,JR,JR);
155
156
157 62 DO 19 J=1,MX;
158 19 SF(I1R,J)=W(J,1)+G(J,I1);
159
160
161 WRITELINE,121;
162 121 FORMAT('
163 X');
164 CALL TIST(SF,I1R,MX,NR,JR);
165 WRITE LINE,143;
166 143 FORMAT('
167 DELTA Y');
168 CALL TIST(Y,I1R,MR,NR,JR);
169 DO 77 I=1,IR;
170 DO 77 J=1,MR;
171 77 Y(I+1,J)=Y(I+1,J)+Y(I,J);
172 WRITE LINE,118;
173 CALL TIST(Y,I1R,MR,NR,JR);
174 CALL GRAFT(SF,NR,JR,I1R,MX,0,0,0,0,110);
175 39 CALL GRAFT(Y,NR,JR,I1R,MR,0,0,0,0,110);
176 40 CONTINUE;
177 124 FORMAT('F');
178 101 FORMAT('
179 I=',I3);
180 END;
181
182
183
184 SUBROUTINE NAS(A,B,C,L,M,N,LM,MM,NM);
185 DIMENSION A(LM,MK),B(MM,NM),C(LM,NM);

```

```

186 DO 1 I=1,L;
187 DO 1 J=1,N;
188 W=0.0;
189 DO 2 K=1,M;
190 2 W=W+A(I,K)*B(K,J);
191 1 C(I,J)=W;
192 RETURN;
193 END;
194
195
196 SUBROUTINE TIS(A,MA,NA,M,N);
197 DIMENSION A(M,N);
198 WRITELINE,101;
199 101 FORMAT('
200 ');
201 DO 1I=1,MA;
202 I1=5;
203 DO 2J=1,NA;
204 WRITELINE,100,A(I,J);
205 100FORMAT(E15.7);
206 I1=I1-1;
207 IF (I1),3,3,2;
208 3I1=5;
209 WRITE LINE,101;
210 2CONTINUE;
211 1WRITE LINE,101;
212 RETURN;
213 END;
214
215
216 SUBROUTINE TIST(A,MA,NA,M,N);
217 DIMENSION A(M,N);
218 WRITELINE,101;
219 101 FORMAT('
220 ');
221 DO 1 J=1,NA;
222 I1=5;
223 DO 2 I=1,MA;
224 WRITELINE,100,A(I,J);
225 100 FORMAT(E15.7);
226 I1=I1-1;
227 IF (I1),3,3,2;
228 3 I1=5;
229 WRITELINE,101;
230 2 CONTINUE;
231 1 WRITELINE,101;
232 RETURN;
233 END;
234
235
236 SUBROUTINE NAST(F,V,WF,M,N,MR,NR);
237 DIMENSION F(MR,NR),V(MR,NR),WF(NR,NR);
238 DO 1IR=1,N;
239 DO 1 IS=IR,N;
240 W=0.0;
241 DO 3J=1,M;
242 3W=W+F(J,IR)*F(J,IS)*V(J);
243 WF(IR,IS)=W;
244 1 WF(IS,IR)=W;
245 RETURN;
246 END;
247
248 SUBROUTINE MAX(A,N,NR);
249 DIMENSION A(NR,NR);
250 DO 1I=1,N;

```

```

551 1A(I,I)=A(I,I)-0.1E1;
552 AM=0.0;
553 DO 2I=1,N;
554 DO 2J=1,N;
555 P=ABS(A(I,J));
556 IF (AM-P),3,2,2;
557 3AM=P;
558 2CONTINUE;
559 WRITE LINE,100,AM;
560 100FORMAT('
561 MAX. CHYBA=',E14.7);
562 WRITE LINE,101;
563 101 FORMAT('
564 ');
565 RETURN;
566 END;
567
568
569 SUBROUTINE GRAFT (Y,MM,NN,MMAX,NMAX,L1,L2,T,IRKA);
570 DIMENSION Y(MM,NN);
571 DIMENSION GV(5),POS(5),YN(5),YX(5),IZ(5);
572 IF (NMAX-5),7,7,8;
573 8 NMAX=5;
574 7 WRITE LINE,300;
575 300 FORMAT('
576
577 ');
578 WRITE LINE,302;
579 302 FORMAT('
580 SEZNAM OZNACENI * F1',4X,'X F2',4X,'0 F3',4X,'+F4',4X,'=F5',4X,'.OSA');
581 DO 66 N=1,NMAX;
582 YX(N)=-0.1E9;
583 YN(N)=+0.1E9;
584 DO 66 M=1,MMAX;
585 IF (Y(M,N)-YX(N)),62,62,63;
586 63 YX(N)=Y(M,N);
587 62 IF (Y(M,N)-YN(N)),64,66,66;
588 64 YN(N)=Y(M,N);
589 66 CONTINUE;
590 IF (1-L2),71,72,400;
591 71 WRITE LINE,301,T;
592 301 FORMAT(6X,'OMEZENI MINIMA=',F12.4);
593 DO 40 N=1,NMAX;
594 IF (YN(N)-T),74,40,40;
595 74 YN(N)=T;
596 40 CONTINUE;
597 GOTO 400;
598 72 WRITE LINE,310,T;
599 310 FORMAT(6X,'OMEZENI MAXIMA= ',F12.4);
600 DO 42 N=1,NMAX;
601 IF (YX(N)-T),42,42,75;
602 75 YX(N)=T;
603 42 CONTINUE;
604 400 YMAX=-0.1E9;
605 YMIN=+0.1E9;
606 DO 16 N=1,NMAX;
607 IF (YX(N)-YMAX),5,5,15;
608 15 YMAX=YX(N);
609 5 IF (YN(N)-YMIN),6,16,16;
610 6 YMIN=YN(N);
611 16 CONTINUE;
612 IF (YMIN),51,51,50;
613 50 YMIN=0.0;
614 51 IF (YMAX),52,53,53;
615 52 YMAX=0.0;

```



```

316 S3 W=IRKA-2;
317 GVOC=W/(YMAX-YMIN);
318 POSUN=GVOC*ABS(YMIN)+0.15E1;
319 K=POSUN;
320 IF (1-L1),600,700,600;
321 600 DO 44 N=1,NMAX;
322 GV(N)=GVOC;
323 44 POS(N)=POSUN;
324 GOTO 888;
325 700 DO 46 N=1,NMAX;
326 GV(N)=W/(YX(N)-YN(N));
327 46 POS(N)=0.15E1-GV(N)*YN(N);
328 K=0;
329 888 WRITE LINE,109;
330 109 FORMAT('
331 ');
332 DO 811 N=1,NMAX;
333 811 WRITE LINE,321,N,YN(N);
334 WRITE LINE,109;
335 DO 812 N=1,NMAX;
336 812 WRITE LINE,322,N,YX(N);
337 322 FORMAT(6X,'MAX',I2,E14.7);
338 321 FORMAT(6X,'MIN',I2,E14.7);
339 WRITE LINE,307;
340 307 FORMAT('
341
342
343
344
345 LINE');
346 DO 1000 M=1,IRKA,10;
347 1000 WRITE LINE,303;
348 303 FORMAT('1234567890');
349 DO 93 I=1,5;
350 93 IZ(I)=0;
351 COMMENT TISK GRAFU;
352 IRK=IRKA+2;
353 DO 1 M=1,MMAX;
354 DO 2 N=1,NMAX;
355 2 IZ(N)=GV(N)*Y(M,N)+POS(N);
356 I=M-1;
357 WRITELINE,111,I;
358 DO 3 I=1,IRK;
359 IF (I-IZ(1)),20,21,20;
360 21 WRITE LINE,101;
361 GOTO 3;
362 20 IF (I-IZ(2)),22,23,22;
363 23 WRITE LINE,102;
364 GOTO 3;
365 22 IF (I-IZ(3)),24,25,24;
366 25 WRITE LINE,103;
367 GOTO 3;
368 24 IF (I-IZ(4)),26,27,26;
369 27WRITE LINE,104;
370 GOTO 3;
371 26 IF (I-IZ(5)),9,29,9;
372 29 WRITE LINE,105;
373 GOTO 3;
374 9 IF (I-K),30,31,30;
375 31 WRITE LINE,110;
376 GOTO 3;
377 30 WRITE LINE,100;
378 3 CONTINUE;
379 1 CONTINUE;
380 WRITE LINE,300;

```

```

381 111 FORMAT('
382 ',I4);
383 100 FORMAT(' ');
384 101 FORMAT('*');
385 102 FORMAT('X');
386 103 FORMAT('O');
387 104 FORMAT('+');
388 105 FORMAT('=');
389 110 FORMAT('.'.);
390 RETURN;
391 END;
392
393
394
395 SUBROUTINE VYTF(F,N,KR,JR,NR,MX,MR,MZ,FU);
396 DIMENSION F(JR,NR),FU(3,45);
397 DO 1 I=1,N;
398 I1=(N+1-I)*KR*MZ;
399 I2=(I-1)*MR;
400 DO 1 I3=1,MR;
401 I4=I1+I3;
402 I5=I2+I3;
403 DO 1 I6=1,MX;
404 1 F(I6,I5)=FU(I6,I4);
405 RETURN;
406 END;
407
408
409
410
411 SUBROUTINE VYTG(G,IK,IG,KR,JR,MX,MR,MZ,FU);
412 DIMENSION G(JR,JR),FU(3,45);
413 IF (IG-1)/2,2,1;
414 1 DO 4 I2=1,MR;
415 DO 3 I1=1,MX;
416 3 G(I1,I2)=0.0;
417 4 G(I2,I2)=0.1E1;
418 RETURN;
419 2 I1=IK*KR*MZ;
420 DO 5 I2=1,MZ;
421 I3=I1+I2;
422 DO 5 I4=1,MX;
423 5 G(I4,I2)=FU(I4,I3);
424 RETURN;
425 END;
426
427
428
429 SUBROUTINE VYT(G,I1,DTK,JR,IN);
430 COMMENT DETERMINOVANA PORUCHA;
431 DIMENSION G(JR,JR);
432 IF (IN),1,1,2;
433 1 IN=1;
434 WRITE LINE,100;
435 100 FORMAT('
436 DET. PORUCHA D
437 ');
438 2 T=I1;
439 T=T*DTK;
440 G(1,1)=FN(2,T);
441 G(1,2)=FN(1,T);
442 G(2,1)=G(1,2);
443 G(2,2)=0.0;
444 RETURN;
445 END;

```

```

447
448
449
450 FUNCTION FN(NR,T);
451     IT=T;
452     K=2*IABS(IT)+NR+25;
453     S=0.0;
454     EPS=0.0;
455     DO 201 J=1,K;
456     N=K-J+1;
457     R=N;
458     R=T/R;
459     EPS=EPS*R;
460     S=(S+0.1E1)*R;
461     IF (N-NR),201,200,201;
462 200 EPS=S;
463 201 CONTINUE;
464     FN=EPS/(S+0.1E1);
465     RETURN;
466     END;
467
468
469 END OF PROGRAM;
470

```

```

471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

MENO PROGRAMU: RTZ

KOMENTAR:

```
1
2      PROGRAM NA URYCHLENÍ KONVERGENCE PRI VYPOCTU
3      OPTIMALNICH AKCNICH ZASAHU
4      V MINIMALNIM POCTU KROKU GRADIENT.METODOU;
5      DIMENSION YD(20),F(20),Y(20),U(20,20),D1(20),D2(20);
6      DIMENSION PAM(20),NC(10);
7      REAL KAPA;
8      READ READ,
9      NR,P,KAPA,T,MK,KU,NI;
10     WRITE LINE,108;
11     WRITE LINE,100;
12     WRITE LINE,101,NR,P,KAPA,T;
13     WRITE LINE,108;
14     WRITE LINE,103;
15     WRITE LINE,108;
16     DO 22 I=1,MK;
17     22 READ READ,F(I);
18     DO 23 I=1,MK;
19     WRITE LINE,111,F(I);
20     23 WRITE LINE,108;
21     WRITE LINE,108;
22     COMMENT VYPOCET LEVE STR.PODMINK.ROVNIC;
23     M=NR+1;
24     DO 39 K=1,M;
25     39 READ READ,NC(K);
26     READ READ,KS,KPS;
27     KF=M;
28     DO 24 KL=1,M;
29     Y(KL)=-F(KF);
30     KF=KF+1;
31     24 CONTINUE;
32     COMMENT VYPOCET PRAVEE STR.PODMINK.ROVNIC;
33     NA=NR;
34     DO 38 KL=1,M;
35     KB=NA;
36     DO 4 K=1,M;
37     IF (NR),62,62,61;
38     62 U(KL,K)=F(K);
39     GOTO 4;
40     61 IF (KB),26,26,25;
41     26 U(KL,K)=0.0;
42     GOTO 4;
43     25 U(KL,K)=F(KB);
44     KB=KB-1;
45     4 CONTINUE;
46     NA=NA+1;
47     38 CONTINUE;
48     COMMENT TISK PODLE KLICE 15;
49     IF KEY(15),98,99;
50     98 WRITE LINE,201;
51     WRITE LINE,108;
52     DO 87 KL=1,M;
53     87 WRITE LINE,200,Y(KL);
54     WRITE LINE,108;
55     WRITE LINE,108;
```



```

56 DO 88 KL=1,M;
57 DO 89 K=1,M;
58 89 WRITE LINE,202,U(KL,K);
59 88 WRITE LINE,108;
60 200 FORMAT('
61 ',E14.7);
62 201 FORMAT('
63 MEZIVYSLEDKY');
64 202 FORMAT(2X,F10.7);
65 99WRITE LINE,108;
66 IB=1;
67 IF KEY (5),92,93;
68 92 WRITE LINE ,106;
69 106 FORMAT ('
70 POKRACOVANI VYPOCTU OD VLOZENEHU
71 POCATECNIHU ODHADU ');
72 DO 29 K=1,M;
73 READ READ,D1(K);
74 29 WRITE LINE,109,D1(K);
75 109 FORMAT ('
76 ',E14.7);
77 WRITE LINE,108;
78 GOTO 28;
79 93 DO 16 K=1,M;
80 16 D1(K)=0.0;
81 28 WRITE LINE,107;
82 WRITE LINE,108;
83 DO 17 KA=1,NI;
84 IC=KA-(KA/KPS*KPS);
85 1F(IC),1,2,1;

86 2 IC=KPS;

87 1 IF(KA-IC),11,11,10;
88 10 IF(IC-1),11,12,11;
89 12 IB=1B+1;
90 GOTO 3;
91 11 IB=IB;
92 3 CALL GENER(IB,KS,NC,KL,KA,IQ);
93 CALL TEST(IB,IC,KPS,KLC);
94 IF(KLC-1),56,56,57;
95 57 CALL GRAD(D1,D2,U,Y,M,KL,KAPA,KA);
96 CALL PAMET(1B,D2,IC,PAM,M);
97 GOTO 58;
98 56 CALL GRADS(D1,D2,U,Y,M,KL,0.1E1,KA,PAM);
99 58DO 13 K=1,M;
100 D1(K)=D2(K);
101 13 CONTINUE;
102 17 CONTINUE;
103 102 FORMAT(5X,E14.7);
104 100 FORMAT ('
105 VYPOCET OPTIM. AKCNICH ZASAHU V MIN. POCETU KROKU GRAD. METODOU
106 *****');
107 101 FORMAT ('
108 RAD SOUSTAVY N=',I3,'
109 P =',F5.2,' KAPA=',F5.2,' T=',F5.2);
110 103 FORMAT ('
111 HODNOTY PRECHODOVE CHARAKTERISTIKY');
112 105 FORMAT(I7);
113 111 FORMAT (25X,E14.7);
114 107 FORMAT(' CISLO KROKU AKCNI ZASAHY
115 =====');
116 108 FORMAT('
117 ');
118 END;

```

```

117 SUBROUTINE GRADS(D1,D2,U,Y,N,KL,APA,KA,PAM) ;
120 DIMENSION D1(20),U(20,20),Y(20),D2(20),PAM(20) ;
121 DIMENSION SM(20) ;
122 T1=0.0E0 ;
123 T2=0.0E0 ;
124 WRITE LINE,102 ;
125 WRITE LINE,108 ;
126 108 FORMAT('
127 ') ;
128 DO 19 K=1,N ;
129 101 FURMAT('
130 ',F10.7) ;
131 SM(K)=D1(K)-PAM(K) ;
132 102 FORMAT('
133 HLEDANI VE SMERU') ;
134 T1= T1+D1(K)*U(KL,K) ;
135 19 T2 =T2 + SM(K)*U(KL,K) ;
136 DO 18 K=1,N ;
137 D2(K)=D1(K)+(APA*(Y(KL)-T1)/T2)*SM(K) ;
138 CALL TISK(KL,KA,1,K,D2,N) ;
139 18 CONTINUE ;
140 RETURN ;
141 END ;
142 SUBROUTINE GRAD(D1,D2,U,Y,N,KL,KAPA,KA) ;
143 REAL KAPA ;
144 DIMENSION D1(20),U(20,20),Y(20),D2(20) ;
145 DIMENSION A(20) ;
146 T1=0.0E0 ;
147 T2=0.0E0 ;
148 DO 19 K=1,N ;
149 T1=T1+D1(K)*U(KL,K) ;
150 19 T2= T2+U(KL,K)*U(KL,K) ;
151 DO 18 K=1,N ;
152 A(K)=U(KL,K) ;
153 D2(K)=D1(K)+(KAPA*(Y(KL)-T1)/T2)*A(K) ;
154 CALL TISK(KL,KA,1,K,D2,N) ;
155 18 CONTINUE ;
156 RETURN ;
157 END ;
158 SUBROUTINE TISK(KL,KA,KV,K,D2,N) ;
159 DIMENSION D2(20) ;
160 KK=KA/KV*KV ;
161 IF (N-5),93,94,94 ;
162 93 NC=N ;
163 GO TO 98 ;
164 94 NC=5 ;
165 98 KP=K/NC*NC ;
166 IF (KA-KK),55,54,55 ;
167 54 CONTINUE ;
168 IF (K-1),56,57,56 ;
169 57 WRITE LINE,105,KA ;
170 WRITE LINE,105,KL ;
171 56 WRITE LINE,102,D2(K) ;
172 IF (K-KP),59,58,59 ;
173 58 WRITE LINE,108 ;
174 WRITE LINE,113 ;
175 59 IF (K-N),18,60,18 ;
176 60 WRITE LINE,108 ;
177 55 CONTINUE ;
178 18 CONTINUE ;
179 105 FORMAT(I6) ;
180 102 FORMAT(5X,E14.7) ;
181 113 FORMAT(12X) ;
182 108 FORMAT('

```

```

183      ) ;
184      RETURN ;
185      END ;
186      SUBROUTINE PAMET (IB,D2,IC,PAM,N) ;
187      DIMENSION D2(20),PAM(20) ;
188      KAS=IB/2*2 ;
189      IF (IB-KAS),1,2,1 ;
190      1 IF (IC-3),3,4,3 ;
191      4 DO 5K=1,N ;
192      PAM(K)=D2(K) ;
193      5 CONTINUE ;
194      3 RETURN ;
195      2 IF (IC-2),6,7,6 ;
196      7 DO 8 K=1,N ;
197      PAM(K)=D2(K) ;
198      8 CONTINUE ;
199      6 RETURN ;
200      END ;
201      SUBROUTINE GENER (IB,KS,NC,KL,KA,IQ) ;
202      DIMENSION NC(10) ;
203      IQ=IB-(IB/KS*KS) ;
204      KAS=KA/2*2 ;
205      IBS=IB/2*2 ;
206      IF (IQ),1,2,1 ;
207      2 IQ=1 ;
208      IF (KA-KAS),3,4,3 ;
209      4 KL=NC(KS) ;
210      RETURN ;
211      3 KL=NC(IQ) ;
212      RETURN ;
213      1 IF (IB-IBS),5,6,5 ;
214      6 IF (KA-KAS),7,8,7 ;
215      8 KL=NC(IQ) ;
216      RETURN ;
217      7 KL=NC(IQ+1) ;
218      RETURN ;
219      5 IF (KA-KAS),9,10,9 ;
220      10 KL=NC(IQ+1) ;
221      RETURN ;
222      9 KL=NC(IQ) ;
223      RETURN ;
224      END ;
225      SUBROUTINE TEST (IB,IC,KPS,KLC) ;
226      IBS=IB/2*2 ;
227      IF (IB-IBS),1,9,1 ;
228      1 IF (IB-1),3,7,3 ;
229      3 IF (IC-KPS),6,10,6 ;
230      6 IF (IC-1),9,8,9 ;
231      8 KLC=1 ;
232      RETURN ;
233      7 IF (IC-KPS),9,10,9 ;
234      10 KLC=1 ;
235      RETURN ;
236      9 KLC=2 ;
237      RETURN ;
238      END ;
239      END OF PROGRAM ;
240

```

V 16/82 T
+ příloha