

# Aplikace matematiky

---

Antonín Tuzar

O řešení některých úloh na počítači Ural 1

*Aplikace matematiky*, Vol. 6 (1961), No. 5, 406--409

Persistent URL: <http://dml.cz/dmlcz/102772>

## Terms of use:

© Institute of Mathematics AS CR, 1961

Institute of Mathematics of the Academy of Sciences of the Czech Republic provides access to digitized documents strictly for personal use. Each copy of any part of this document must contain these *Terms of use*.



This paper has been digitized, optimized for electronic delivery and stamped with digital signature within the project *DML-CZ: The Czech Digital Mathematics Library* <http://dml.cz>

## O ŘEŠENÍ NĚKTERÝCH ÚLOH NA POČÍTAČI URAL 1

ANTONÍN TUZAR

(Došlo dne 2. května 1961.)

Od června roku 1959 je v početním středisku ÚTIA ČSAV v provozu sovětský samočinný počítač Ural 1. Během dvou let byla na stroji provedena celá řada vědecko-technických výpočtů a byly získány zkušenosti v programování jistých typů úloh. O některých z nich bude pojednáno v tomto článku; přitom bude kladen důraz na matematickou formulaci a na otázky programování. Čtenář tak získá určitý přehled o možnostech využití podobných strojů.

Ural 1 je malý univerzální elektronkový číslicový počítač. Pracuje s pevnou řádovou čárkou (viz např. [3]) ve dvojkové soustavě, vstup a výstup je osmičkový nebo desítkový. Čísla ve stroji mají 35 dvojkových řádů (odpovídajících  $2^{-1}$  až  $2^{-35}$ ) a 1 řád znaménkový. Rychlost výpočtu je 100 operací za vteřinu; výjimkou je dělení, jež je čtyřikrát pomalejší. Systém programování je jednoadresový. Stroj je vybaven operační pamětí na magnetickém bubnu o kapacitě 1024 čísel nebo 2048 instrukcí a vnější pamětí na magnetické páse o kapacitě až 40 000 slov. Vstup je s děrné pásky (filmu), výstupní ústrojí tvoří tiskárna (rychlost tisku je 100 čísel za minutu) a výstupní perforátor (rychlost děrování je 150 čísel za minutu). Kod stroje obsahuje i takové instrukce, jež umožňují (za cenu zkomplikování programu a až třicetinasobného zpomalení výpočtu) počítat s pohyblivou řádovou čárkou. Každé číslo různé od nuly je pak reprezentováno dvojicí  $p, q$ , odpovídající zápisu  $2^p \cdot q$ ,  $\frac{1}{2} \leq q < 1$ .

Programy, jež byly dosud ve středisku sestaveny nebo přejaty odjinud, lze rozdělit podle jejich použitelnosti na tyto tři typy:

a) Podprogramy v obecných adresách, jež umožňují výpočet elementárních funkcí ( $\sin x$ ,  $e^x$ ,  $\sqrt{x}$  apod.), převody mezi dvojkovou a desítkovou soustavou a operace s čísly s pohyblivou čárkou. Tyto podprogramy byly dodány zároveň s ostatní dokumentací ke stroji, některé z nich byly zlepšeny (např. součet v pohyblivé čáře, převod z dvojkové do desítkové soustavy). Používá se jich prakticky ve všech úlohách.

b) Standardní programy v konkrétních adresách, jež umožňují řešit mnoho alternativ určité úlohy (např. inverze matic). Je požadováno, aby manipulace se standardním programem byla jednoduchá a bylo jí možno provést podle příslušného návodu i bez znalosti vnitřní struktury programu, a také aby pro řešení konkrétního případu nebylo třeba děrovat o mnoho více než vstupní údaje (tj. koeficienty soustavy rovnic a její řád).

c) Programy jichž je užito pouze pro jedinou úlohu. Převážná většina programů je tohoto typu. Je přirozené, že tu není třeba věnovat takovou pozornost úsporám strojního času jako u programů standardních.

Řešené úlohy lze rozdělit do několika skupin podle jejich matematické formulace. Především to byly úlohy, jejichž formulace obsahovala pouze elementární funkce. Většinou šlo o sestavení různých technologických tabulek. Programování podobných úloh je snadné a může je dobře provádět absolvent jedenáctiletky, který zná základy programování. Někdy je však třeba i v tako-

vých elementárních úlohách užít určitých obrátů, jež umožní shrnout řadu zdánlivě odlišných případů pod jeden a podobně. Jako příklad uveďme výpočet torsních kmitů volného konce klikového hřídele 16tivalcového Diesela motoru s uspořádáním válců do V v závislosti na pořadí vstříků. V úloze byla počítána výsledná amplituda těchto kmitů, přičemž kmity byly rozloženy do 24 harmonických složek a byly stanoveny součty souřadnic těchto složek v rovině kolmé na osu hřídele. Tyto souřadnice složek byly zadány elementárními vzorci; obtíž byla v tom, že alternativ bylo velmi mnoho. Je zajímavé, že výpočtem byly zjištěny nové souvislosti mezi některými alternativami a to umožnilo zkrátit pozdější podobné výpočty [4].

Mnohé problémy, přicházející z různých pracovišť, lze řešit metodami lineární algebry. Proto byly vypracovány standardní programy, o nichž nyní pojednáme. — Program *inverse matric* (sestavený E. Kindlerem) pracuje s pevnou řádovou čárkou. Lze jím invertovat čtvercové matice do 29. řádu včetně. Program sám provádí volbu měřítek pro jednotlivé řádky; pokud tato měřítká jsou různá, je *inverse* možná jen do řádu matice 28. Je použito eliminační metody s výběrem maximálního prvku v řádku. Jestliže během úprav matice vznikne nulový řádek, stroj se samočinně zastaví. *Inverse matice* 10. řádu trvá asi 1 hodinu, matice 25. řádu necelé tři hodiny. Tímto programem bylo řešeno např. 60 matice řádu 24 a pro většinu z nich byla namátkově počítána residua, jež vyšla menší než  $10^{-5}$ . Poznamenejme, že k těmto maticím vedlo řešení úlohy o namáhání ve vodorovné desce tvaru kosodélníka metodou sítí. Program *inverse matric* je doplněn programem pro násobení matice do 20. řádu včetně. (Sestavil J. Zezula.) Násobení matice 10. řádu trvá asi 25 minut, 20. řádu asi 1,5 hodiny.

Eliminační metody používá také program pro řešení soustav lineárních rovnic (do 23 rovnic včetně, sestavil M. Fendrych). Tento program pracuje s pohyblivou čárkou. Řešení osmi rovnic trvá asi 20 minut, dvaceti tří rovnic asi 1,5 hodiny. Mimo tento byl sestaven (Z. Drahoňovským) ještě program pro současně řešení soustav lineárních rovnic o  $n$  neznámých pro  $r$  pravých stran, kde  $n + r \leq 56$ ; speciálně jím lze řešit soustavu o 55 neznámých. Výpočet soustavy o 50 rovnicích a šesti pravých stranách včetně výpočtu a tisku residuí (rozdíly pravých a levých stran po dosazení) trval asi 2 hodiny.

V současné době je zkoušen program (J. Král, J. Žáček) pro vyhledání vlastních čísel matice. Program se hodí na matice do 27. řádu včetně, jejichž všechna vlastní čísla jsou reálná a pokud jsou různá, není jejich poměr větší než 0,90. V programu je použito tohoto iterativního postupu: Na počátku zvolíme libovolně vektor  $u^{(0)}$  (např.  $u^{(0)} = (1, 1, \dots, 1)$ ). Postupně tvoříme vektory  $u^{(n)} = A \cdot u^{(n-1)}$ , kde  $A$  je daná matice. Když poměr  $\frac{u_i^{(n)}}{u_i^{(n-1)}} = \lambda$  je stejný pro všechny nenulové složky, považujeme jej za největší vlastní číslo. Vektor  $u^{(n)} - \lambda u$  je pak příslušný charakteristický vektor. Stejně najdeme modální řádek  $v$  a proces opakujeme pro matici  $(a_{ij} - \lambda_1 u_i v_j)$ . [7]. Výpočet jednoho vlastního čísla a příslušného charakteristického vektoru bude trvat asi 20 minut.

Ve středisku se také pracuje na programu Jakobioho metody diagonalisace, která je obecnou metodou pro výpočet spektra a charakteristických vektorů Hermiteových matic (tj. v reálném oboru matic symetrických). Jakobioho metoda má být pro speciální typ matic blízkých matic jednotkové srovnána s metodou diagonalisace, navrženou M. Fiedlerem a V. Ptákem. Jejich iterační metoda byla již zkoušena na řadě matic 16-tého řádu a v nejméně příznivém případě trvala 1,5 hodiny.

Dále byl ve středisku sestaven (autorem tohoto článku podle [8]) standardní program pro řešení dopravního problému do rozsahu  $20 \times 50$ . Program používá simplexové metody a umožňuje řešit (přesně) úlohy, v nichž vstupní hodnoty jsou nejvyšší čtyřciferná čísla. Doba výpočtu nezávisí jen na rozsahu, přesto uveďme pro zajímavost tyto údaje: řešení úlohy  $4 \times 6$  trvalo asi 12 minut,  $5 \times 30$  asi 20 minut,  $10 \times 45$  asi 65 minut,  $20 \times 40$  asi 90 minut. Poznamenejme, že pracovníkům VÚTE při Ministerstvu chem. průmyslu (K. Bláha, M. Veverka, B. Cahová) se podařilo sestavit standardní program pro rozsahy až  $16 \times 210$ . Doba trvání jedné iterace je však u tohoto programu delší.

Početná skupina úloh vedla na hledání kořenů rovnic různých typů. Pro řešení algebraických rovnic do 26. stupně včetně byl sestaven (Z. Drahoňovským) standardní program, umožňující nalézt všechny (reálné i komplexní) kořeny. Je použito Shin-Nge-Linovy metody (viz např. [2]), daný mnohočlen se tak rozloží v součin kvadratických činitelů a v případě rovnice lichého stupně přibude jeden lineární činitel. Vyřešení rovnice 10. stupně tímto programem trvalo asi 23 minuty. — Pro hledání reálných kořenů algebraických i transcendentních rovnic bylo zpravidla použito tohoto postupu: Příslušná funkce byla počítána ve stroji se zvoleným krokem. Jakmile došlo ke změně znaménka, byl zmenšen krok a obrácen postup. Zpravidla nevedla pomalá konvergence této metody, jejíž programování je velmi snadné. Avšak u úloh, v nichž šlo o řešení velkého počtu podobných rovnic, ukázalo se výhodným použít metody Newtonovy nebo jiných iterativních metod. Např. při hledání reálných kořenů rovnic 
$$\sum_{i=1}^n \frac{a_i}{x - b_i} = 0 \quad (a_i, b_i \text{ reálná})$$
 bylo užitím Newtonovy metody dosaženo až trojnásobné úspory času proti výše uvedenému postupu.

Dostí běžným typem úloh, zadávaných středisku, jsou výpočty různých transcendentních funkcí, jejich numerické derivování a integrování, hledání extrémů apod. Zatím bylo v těchto úlohách použito vždy běžných metod numerické analýsy, např. integrály byly počítány většinou podle Simpsonova pravidla. Protože u těchto úloh bývá požadována vysoká přesnost, je často třeba předem provádět velmi podrobný rozbor jednotlivých úseků výpočtu a je nutné volit takové pořadí operací, jež zaručí minimální ztrátu řádových míst. Leckdy je nutno volit různé umělé obraty, např. při násobení velkého čísla číslem velmi malým a podobně.

Je přirozené, že na počítači byly řešeny též diferenciální rovnice. Ve většině případů obyčejných rovnic šlo o počáteční úlohu a vždy se zde vystačilo s Runge-Kuttovou metodou, jež je dostatečně přesná a pro programování jednoduchá. Maximální soustava, řešená na stroji, sestávala z třinácti rovnic prvního řádu. Značně složitější bylo řešení rovnic s tzv. dopravním zpožděním, jež se vyskytují v teorii samočinné regulace. Byly sestaveny (J. Žáčkem) dva programy pro rovnice tvaru

$$\sum_{i=0}^n a_i y^{(i)}(t) + \sum_{j=0}^2 b_j y^{(j)}(t - \tau) = c,$$

kde  $2 \leq n \leq 11$ ,  $\tau > 0$ ,  $a_i, b_j, c$  jsou reálné konstanty, pro  $t = 0$  jsou dány počáteční podmínky. První program používá Runge-Kuttovy metody (viz např. [5]), a ukázalo se, že je vhodný pro nižší řády (do pátého). U těchto rovnic, jak se snadno nahlédne, je třeba si pamatovat výsledky z předchozího intervalu délky  $\tau$ . Hodnoty v polovinách intervalů délky kroku, jež vystupují ve vzorcích Runge-Kuttovy metody, je třeba interpolovat a dále je třeba v bodech, jež jsou celistvými násobky  $\tau$ , počítat nové počáteční podmínky zprava. (Srov. [1], [6]). Tyto nevýhody kompenzuje částečně druhá metoda, používající zpětné Laplaceovy transformace [9]. Při numerickém integrování obrazu vznikají některé komplikace u rovnic nižších řádů, u vyšších řádů (5 až 11) dává tato metoda poměrně přesné výsledky. Parciální rovnice se vyskytly v naší praxi zatím ojedinelé. Z toho byly úplně vyřešeny tyto dvě úlohy: 1. Výpočet dopravní účinnosti dvoutaktního motocyklového motoru. Šlo tu o vyřešení soustavy dvou lineárních parciálních rovnic 1. řádu s danými okrajovými podmínkami. Úloha byla řešena metodou charakteristik. — 2. Exotermické jevy při betonáži přehrad. V této úloze byla řešena metodou sítí rovnice pro vedení tepla na oblasti se „stupňovitou“ hranicí; byla prováděna jistá iterace řešeními na obdélníku. V současné době se připravuje řešení tzv. rovnic vorticity pro barotropní předpověď.

Konečně bylo řešeno několik úloh, jež nelze zařadit do žádné z uvedených skupin. První z nich byla úloha o minimalizaci Booleovy funkce, souvisecí s hledáním optimálních zapojení ve složitých elektrických obvodech. Program sestával z dvou částí, v první z nich byly vyhledány všechny podstatné konjunkce dané Booleovy funkce, druhá část řešila nalezení rozvoje Booleovy funkce čtyř proměnných s minimálním součtem vah konjunkcí, jež v něm vystupují (váhy byly

předem dány). — Dále byly řešeny některé statistické úlohy, např. výpočet periodogramu, výpočet střední kvadratické odchylky metodou redukce polynomů a jiné. Celkem bylo vyřešeno asi 50 externích a přes 30 interních úloh. Z toho asi 30% byly úlohy s elementárním zadáním, kolem 20% diferenciální rovnice a asi 25% úlohy z lineární algebry. Jak čtenář viděl, ve velké většině případů bylo možné vystačit s běžnými numerickými metodami. Převážná většina programů kromě standardních užívala pevné čárky a proměnných měřítek. Vnější paměti bylo užito jen asi v pěti případech pro výpočet. Jinak se jí užívá tehdy, když je třeba přerušit výpočet a zapamatovat současný obsah paměti. Ukazuje se, že v některých úlohách nestačí řádový rozsah stroje, proto byl (J. Králem) sestaven program dvojité aritmetiky. Program pracuje v pseudokodu a umožňuje provádět všechny instrukce stroje se zdvojeným řádovým rozsahem. Program zabírá víc než třetinu paměti a pseudooperace trvají dlouho (sčítání 0,5 vteřiny, násobení a dělení asi 2 vteřiny). Lze jej proto použít jen na nejjednodušší úkoly. — Zajímavý je také program automatické modifikace adres (M. Fendrych). Tento program má za úkol zbavit programátora zdlouhavého rozepisování do konkrétních adres. Na vstup je třeba zadat program v obecných adresách a počty buněk (pracovních, s konstantami, s programem). Rozmístění v paměti a modifikace adres se děje automaticky. Modifikace je hotova během několika minut (v závislosti na délce programu). Používá se ho však málo, protože u většiny programů příliš záleží na optimálním využití poměrně malé paměti. Jeho význam je spíš v tom, že jde o pokus o automatisaci programování.

Dvouletá zkušenost s využitím počítače Ural I prokázala všestrannou použitelnost podobných strojů. Zároveň se však ukazuje, že programování v té formě, jak se dosud provádělo, je časově dosti náročné (i u velmi jednoduchých úloh málokdy trvá kratší dobu než týden, u složitých úloh i několik měsíců!) a rozepisování programů je častým zdrojem chyb. Mnoho zbytečné práce by se ušetřilo, kdyby napříště byly zaváděny počítače vybavené rozsáhlými knihovnami standardních programů a takové, u nichž je možné automatické programování.

#### *Literatura*

- [1] *Айзерман М. А.*: Лекции по теории автоматического регулирования, Физматгиз, Москва 1958.
- [2] *Березин И. С. - Жидков Н. П.*: Методы вычислений, I, II, Физматгиз, Москва 1959.
- [3] *Бондаренко В. Н., Плотников И. Т., Полозов П. П.*: Программирование для цифровой вычислительной машины „Урал“, Москва 1957.
- [4] *Brandt I.*: Torsní kmitý - numerická metoda výpočtu, závěrečná zpráva TIZ 2341, VÚ 18/59, ČKD Praha.
- [5] *Collatz L.*: Numerische Behandlung von Differentialgleichungen, Springer Verlag, Berlin - Göttingen - Heidelberg 1951.
- [6] *Эльсольц Л. Э.*: Качественные методы в математическом анализе, Гостехиздат, Москва 1955.
- [7] *Фаддеев Д. К. - Фаддеева В. Н.*: Вычислительные методы линейной алгебры, Физматгиз, Москва 1960.
- [8] *Dennis J. B.*: A high-speed computer technique for the transportation problem, Journal of the Association for Computing Machinery, vol. 5, No. 2., 1958.
- [9] *Солодовников В. В.*: Основы автоматического регулирования, Машгиз, Москва 1954.