MATICOVÉ OPERACE A VĚDECKO-TECHNICKÉ VÝPOČTY

I. Horsák

Laboratoř výpočetní techniky, PF UK, Albertov 2030, Praha 2

Při programování úloh z oblasti vědecko-technických výpočtů je typická dlouhá fáze ladění programů. Mnohdy se stává, že v okamžiku odladění programu je i vyřešena úloha (např. ve formě grafů, tabulky nebo i jediného čísla. A nová úloha vyžaduje zase úpravu programu. V takových případech je s výhodou používán BASIC jako interpretační jazyk. Hlavní nevýhodu interpretru - pomalost, může eliminovat použití softwaru pro maticové operace, pokud je k dispozici.

Maticové operace, nebo jinak nazývané "operace s poli" je programové vybavení umožňující práci s celými numerickými poli, uloženými v paměti počítače. Dvou- a jednorozměrná pole jsou realizace matic a vektorů v počítači. Numerická pole a s nimi související smyčky jsou samozřejmou součástí všech vyšších jazyků, které dovolují manipulovat postupně se všemi prvky polí. Podstatou maticových operací je provádění manipulace se všemi prvky pole na úrovni strojového jazyka. V případě Basicu, jako interpretru to tedy znamená, že se vzdáme možnosti přerušit výpočet uprostřed smyčky, ale až po ukončení operace s celým polem. Programování pomocí maticových operací tak sestává z řazení jednotlivých operací, které představují bloky instrukcí, což přispívá nejen ke zrychlení výpočtu, ale i k přehlednějšímu zápisu programu a ke zmenšení rizika chyb při programování smyček a indexů.

Vzhledem k uvedeným důvodům je logické, že se maticové operace staly součástí Basicu, jako hlavního reprezentanta interpretačních jazyků a dokonce jsou zakotveny v jeho normě. V případě ostatních vyšších jazyků (FORTRAN, PASCAL) je nutno maticové operace řešit pomocí vlastních podprogramů. V případě PASCALU je nezbytné, aby příslušná implementace dovolovala konformní pole.

Podle normy Basicu se pro odlišení maticových operací od ostatních příkazů používá klíčového slova MAT ihned za číslem řádku a vlastní operace je vyznačena pomocí identifikátorů polí. Pojem práce s celým polem znamená od nejnižšího indexu (0 nebo 1 podle OPTION BASE) po nejvyšší index, který je buď deklarován běžným příkazem DIM, případně změněn příkazem REDIM, který již náleží mezi maticové operace.

Jednu skupinu maticových operací představují dosazovací operace:

	MAT A=ZER	(naplnění matice A nulami)
	MAT A=CON	(naplnění matice jedničkami)
	MAT A=IDN	(vytvoření "jednotkové" matice, na diagonále jedničky, jinak nuly)

K nim patří i operace

MAT READ A	(která plní matici hodnotami z ř. DATA)

Pro styk s vnějšími periferiemi jsou určeny operace:

MAT INPUT A	(plnění hodnotami z klávesnice)
MAT PRINT A	(výstup na obrazovku)

Další skupinu tvoří přiřazovací operace:

MAT A=B	(kopírování matice B do A)
MAT A=B+C	(maticové sčítání, ale i A=A+C)
MAT A=B-C	(maticové odečítání)
MAT A=B*C	(maticové násobení)
MAT A=(výraz)*B	(skalarní násobení hodnotou, získanou vyčíslením výrazu)

Speciální skupinu tvoří operace

MAT A=TRN(B) (transpozice matice - záměna indexů)

MAT A=INV(B) (inverze matice)

Výpočet determinantu, který s předchozí operací úzce souvisí je již příkladem takových operací, jejichž výsledkem je skalár:

DET(A) (zde se již nepoužívá MAT)

SIZE(A) (určuje deklarovaný rozměr matice)

DOT(A,B) (skalární součin vektorů A a B)

Redefinici pole lze provádět buď příkazem REDIM nebo připojením nových indexů v závorce za maticovou operací, např.

MAT A=ZER(4,10)

Překladače Basicu, obsahující maticové operace hlídají nejen správnou syntaxi, ale při výpočtu mají několik chybových hlášení pro případy: - kdy není rovnost typů matic

- matice není čtvercová (IDN,INV)
- rozměry neodpovídají pravidlům pro mat. násobení
- invertovaná matice je singulární

Maticové operace jsou většinou omezeny na maximálně dvourozměrná pole. Dvourozměrná pole s druhým rozměrem jednotkovým, bývají vzájemně zastupitelná s jednorozměrnými poli, ale nemusí to být pravidlem. Některé implementace dovolují i maticové řetězcové operace, ale vzhledem k zaměření tohoto příspěvku na vědeckotechnické výpočty byl tento typ operací ignorován.

Ačkoliv jsou tedy maticové operace zakotveny v normě Basicu, jen málo výrobců jimi vybavuje své počítače, respektive jejich programové vybavení. Jednou z firem, která od počátku důsledně vybavuje své počítače maticovými operacemi je americká firma HP (Hewlett-Packard). Měl jsem příležitost seznámit se s 3 typy počítačů této firmy a nechť tyto typy představují sondy do bohatésortimentů počítačů HP a jsou tak ukázkou vývojových tendencí v oblasti maticových operací.

Počítač HP9830A patřil k prvním stolním počítačům s Basicem. Maticové operace byly k němu dodávány jako přídavná ROM paměť o kapacitě asi 2kb; MATRIX ROM stál asi 500$. Umožňoval následující operace: REDIM, READ, PRINT, ZER, CON, IDN, A=B, A=B+C, A=B*C A=(k)*B, TRN, INV a DET. Urychlení výpočtů bylo 5 až 20 násobné. Výhodou byla již zmíněná zastupitelnost vektorů s maticemi s 2. rozměrem jednotkovým. Vadila nemožnost použití operací typu A=B+C s jiným operátorem než + a –. Inverze byla prováděna eliminační Gauusovou metodou s pivotací. Doba výpočtu determinantu byla zjištěna totožná s dobou inverze, takže obě operace používaly pravděpodobně téhož algoritmu. Zdá se, že tento soubor maticových operacích posloužil za základ pro tvorbu normy.

Další typ – HP9845 (mezitím byly ještě typy 9825 a 9835) měl již bohatějši soubor maticových operací. Navíc přistoupily operace: INPUT, A=FCE(B), A=B∩C a A=(k)∩B, kde ∩ představuje různé operátory včetně logických, A=(výraz), CSUM a RSUM pro sumace sloupců nebo řádků a jejich uložení do vektorů . SUM pro sumaci všech prvků pole, DOT(A,B) pro skalární součin dvou vektorů, COL a ROW pro zjištění rozměrů matic. V případě výpočtu determinantů byla kromě klasické operace DET(A) přidána operace DET, kterou je vyjímána hodnota determinantu po předchozí inverzi. Domnívám se, že ke škodě bylo opuštění zastupitelnosti vektorů s maticemi s druhým rozměrem 1. Urychlení výpočtů u tohoto typu počítače bylo řádově 10–30 násobné. U tohoto a dalších typů byly již maticové operace samozřejmou součástí Basicu a ještě později maticové operace přestaly být záležitostí hardwarovou (jako přídavné ROM paměti) nýbrž čistě softwarovou.

Počítač HP310 náleží k řadě 3000 a maticové operace jsou součásti verze Basic 5.0. Zde jsou základní maticové operace rozšířeny o příkazy: BASE – vrací dolní hranici indexů, RAND – vrací počet dimenzí pole, MAX a MIN – vracejí největší a nejmenší prvek pole a PLOT – vynáší křivky, jejíž souřadnice jsou na polích. Dále jsou zde součástí maticových operací i operace s komplexními čísly v numerických polích a dále 3 skupiny operací: pro přeuspořádávání polí (MAT REORDER), pro vyhledávání v polích (MAT SEARCH) a pro třídění polí (MAT SORT). Zdá se, že toto je nejdůležitější pokrok, který lze vysledovat ve vývojových tendencích softwaru maticových operací u firmy HP – přechod od maticových operací jako nástroje lineární algebry k obecným operacím s numerickými poli.

Absenci maticových operací u dnešních mikropočítačů jsem se snažil odstranit vytvořením souboru maticových operací ve strojovém kódu pro u nás nejrozšířenější mikropočítač Sinclair ZX Spectrum. Jeho popis byl uveřejněn v časopise Elektronika 1/88. Snažil jsem se zachovat určitou návaznost na normu, avšak jako tvůrce jsem se jí neomezoval. Do souboru maticových operací jsem zahrnul i takové operace, které nejsou ani v normě ani u žádného typu počítačů HP, avšak v praxi jsou velmi užitečné.

Jsou to např. operace dovolující přenášet řádky matic do vektorů a naopak a operace přenosu diagonály (stopy) matice do vektoru:

```
MAT V=A(I)      (I je řádkový index
MAT A(I)=V                 matice)
MAT V=DGN(A)
```

Operace pro přenos sloupců nejsou nutné, stačí předem provést transpozici.

Dále byla vytvořena operace pro potřebu regresních výpočtů:

MAT A=RBS(V)

která plní matici A po řádcích mocninami vektoru V a generuje tak tzv. matici "regresorů" (viz dále).

Konečně jsem vytvořil nový typ dosazovací operace:

MAT V=SEQ

která naplní vektor V posloupností čísel 0,1,2,3,......,n až do rozměru vektoru. Následnými operacemi, přičítáním a násobením lze takovou posloupnost libovolně transformovat na žádaný vektor nezávislé proměnné a pomocí dalších operací na vektor závislé proměnné, kterou pak lze graficky znázornit na obrazovce pomocí operace: MAT PLOT V

Použitím maticových operací na ZX Spectru bylo dosaženo 6-500 násobného urychlení výpočtů. Horní hranice se týká těch operací, které s výhodou používají instrukci blokového přenosu procesoru Z80.

Na základě rozboru, které druhy výpočtů při řešení vědecko-technických úloh jsou časově nejnáročnější a daly by se přesně a jednoznačně definovat, by se daly navrhnout ještě další maticové operace, respektive operace s poli.

VYUŽITÍ MATICOVÝCH OPERACÍ PŘI ŘEŠENÍ REGRESNÍCH ÚLOH

Řešení regresních úloh náleží k jedné z nejdůležitějších úloh vědecko-technických výpočtů, týkajících se zpracování experimentálních dat, zatížených chybami. Při její řešení lze velmi efektivně použít maticové operace.

Zobecněme si regresní úlohu takovým způsobem, že máme daný lineární regresní vztah pro závislou proměnnou y:

$$y = a_1f_1 + a_2f_2 + \ldots\ldots a_Lf_L$$

kde a_1 až a_L jsou hledané regresní parametry a f_1 až f_L jsou tak zvané "regresory", což mohou být např. různé mocniny nezávislé proměnné x v případě polynomické regrese nebo různé nezávislé proměnné v případě vícenásobné regrese.

Při řešení regresních úloh pomocí maticových operací lze navrhnout dvě různé metody. Základem první je vytvoření matice regresorů F o rozměrech NxL, kde N je počet experimentálních bodů a L počet parametrů. Každý řádek matice je tvořen regresory pro daný experimentální bod. Tuto matici je třeba invertovat: G=TRN(F) a obě matice vynásobit. Zde je okamžitě zřejmá nevýhoda této metody - velká náročnost na paměť v důsledku nutnosti držet v paměti obě matice o rozměru NxL. Vynásobením vznikne matice soustavy lineárních rovnic: B=G*F, obsahující v jednotlivých prvcích sumace regresorů. Vynásobením matice F vektorem závislé proměnné dostaneme vektor pravých stran C soustavy lineárních rovnic, v maticovém zápisu: B*A=C, kterou je třeba řešit. Je výhodnější neřešit ji přímo, ale přes inverzi matice, protože diagonální prvky inverzní matice budeme potřebovat. Řešení je tedy: $A=B^{-1}*C$. Tím však použití maticových operací nekončí. Vynásobením matice G vektorem parametrů A získáme vektor vypočtených hodnot závislé proměnné Z, jehož odečtením od vektoru Y dostaneme vektor reziduí R a skalárním součinem dvou těchto vektorů dostaneme sumu čtverců odchylek: S=DOT(R,R). Z ní lze vypočítat směrodatnou odchylku a vynásobením odmocninami diagonálních prvků inverzní matice dostaneme směrodatné odchylky parametrů.

Vraťme se ještě k matici regresorů. Pro její vytvoření v případě polynomické regrese není soubor maticových operací dostačující, a to ani podle normy, ani u žádného z počítačů HP. Pomocí násobení vektorů s operátorem tečkou (tedy nikoliv matic. násobení) by se daly vytvořit různé mocniny vektorů, ale zase chybí operace pro ukládání vektorů do řádků matice. Z těchto důvodů jsem pro Spectrum navrhl operaci REG pro přímé generování matice regresorů F. Bez ní je nutno naplnit matici F ve dvojité smyčce, procházející jak všechny exp. body, tak všechny parametry. Další postup, který byl v předchozím odstavci komentován, vypadá v zápisu pomocí maticových operací takto:

```
MAT G=TRN(F)
MAT B=G*F
MAT C=F*Y
MAT B=INV(B)
MAT A=B*C
MAT Z=G*A
MAT R=Z-Y
MAT S=DOT(R,R)   nebo   MAT T=TRN(R)
                        MAT S=T*R
```

S použitím uvedeného algoritmu pro řešení polynomické regrese lze dosáhnout výrazného zkrácení doby výpočtu oproti klasickému postupu pomocí příkazů Basicu. V případě již zmíněného mikropočítače ZX Spektrum bylo urychlení až 10x, takže dosažené časy jsou srovnatelné, ne-li lepší než u 16 bitových počítačů.

Druhá metoda řešení regresních úloh spočívá v tom, že ve smyčce, procházející přes všechny experimentální body je vždy generován pouze vektor regresorů F, který je transponován na řádkový vektor G. Oba vektory jsou pak vynásobeny, ale v takovém pořadí, F*G, že vznikne matice D, rozměru LxL, která tvoří příspěvek do matice soustavy B. Jestliže matici B před smyčkou vynulujeme, pak jednotlivé příspěvky jsou do ní sumovány: B=B+D. Podobným způsobem se vytvoří vektor pravých stran C. Po ukončení smyčky jsou naplněny matice B a C a tak je možno již známým způsobem řešit vektor parametrů A. Tato metoda je sice pomalejší, ale má minimální nároky na paměť, takže je předurčena ke zpracovávání velkých souborů dat.

Obě uvedené metody řešení regresních úloh obsahují etapu řešení inverze matice. Z hlediska přesnosti výpočtu je to klíčová etapa. Pro výpočet inverze je běžně používána Gaussova eliminační metoda, která poskytuje pouze přibližné řešení a které by mělo být dále zpřesňováno. Avšak místo tohoto postupu, který je náročný na počet matic držených v paměti byla navržena jiná metoda, jejíž použití v praxi se velmi osvědčilo.

Jedná se o iterační řešení regresních úloh, podobně jako v případě nelineární regrese. Ačkoliv je tedy řešena úloha lineární regrese, je získaný vektor parametrů považován pouze za odhad. Získaný vektor reziduí je proto znovu podroben regresnímu zpracování. Jako řešení je nyní získán vektor oprav parametrů, který je k původnímu odhadu přičten. Tento postup se opakuje tak dlouho dokud významně klesá suma čtverců odchylek. Ukazuje se, že ve většině případů je vektor oprav téměř nulový a iterace tak po dvou krocích končí. Časové zdržení není tak velké a je vyváženo jistotou, že byla provedena kontrola. Avšak v případě špatně zvoleného regresního vztahu vzhledem k experimentálním datům, kdy vznikají tzv. špatně podmíněné matice, navržený iterační postup velmi rychle konverguje a poskytuje správné řešení, které se v hodnotě sumy čtverců odchylek může lišit až o několik řádů od prvního odhadu. V případě divergence iteračního postupu to signalizuje hrubé chyby v počátečních předpokladech, není poskytnuto žádné řešení a nutí tak uživatele provést přehodnocení zadání. To je také jeden z nedostatků firemních regresních programů, že tuto možnost neposkytují a dovolí, aby uživatel odešel s nesprávným řešením a ani o tom nevěděl.

Pomocí tohoto postupu se také podařilo uspokojivě řešit regresní úlohy na našich mikropočítačích, které mají 6 místnou aritmetiku a která je pro vědecko-technické výpočty absolutně nevhodná. Avšak praktické pokusy ukázaly, že otázkou přesnosti při inverzi matice se musíme zabývat i u takových počítačů, které počítají s 12 místnou přesností, jako např. HP9845. Jakmile je řešeno více jak 6 parametrů, začínají se i zde projevovat zaokrouhlovací chyby a iterační postup je opět na místě.

Zmíněný iterační algoritmus je ve své podstatě velmi podobný řešení úlohy nelineární regrese pomocí Gauss-Newtonovy metody. Zde však jsou regresory derivace regresního vztahu podle jednotlivých parametrů a v každém iteračním kroku je třeba se vracet až na začátek k výpočtu jednotlivých derivací, které jsou díky nelineárnímu regresnímu vztahu vždy jiné v závislosti na právě uvažovaném odhadu parametrů. Tento algoritmus je dobře známý a nebude zde proto rozváděn. Avšak zmíněná metoda má jednu nevýhodu, že potřebuje poměrně dobrý odhad parametrů, jinak snadno diverguje. Existuje známá modifikace této metody - Marquartova, která tuto nevýhodu odstraňuje. Její princip spočívá v přičtení konstanty k diagonálním prvkům matice soustavy před její inverzí Tato konstanta působí jako tlumící faktor, který zajišťuje konvergenci iteračního procesu, avšak prodlužuje výpočet. V praxi se mi osvědčila následující metoda pro automatické řízení tlumícího faktoru, která důsledně využívá maticových operací. Lze ji stručně popsat následujícím způsobem. Tlumící faktor je na počátku nastaven na hodnotu 1E-10, je tedy téměř nulový. Je přičten na diagonálu matice B, provedena inverze a vypočten vektor oprav parametrů. Pokud oprava kteréhokoliv parametru přesahuje v absolutní hodnotě polovinu hodnoty parametru, je tlumící faktor zvýšen o řád (vynásoben 10) a postup se opakuje. Jakmile všechny opravy splňují uvedenou podmínku, jsou přičteny a iterační algoritmus jde k dalšímu kroku. Důsledkem toho, je-li počáteční odhad vzdálený a normální GN metoda by divergovala, je tlumící faktor držen na vysoké hodnotě. Jakmile se odhad přiblíží konečnému řešení a přestane hrozit riziko divergence, tlumení ustoupí a výpočet rychle zkonverguje. Tato metoda vyžaduje správný odhad co do znaménka, protože nedovoluje, aby parametry během iterací změnily znaménko. Zápis zmíněného úseku iteračního algoritmu za pomoci maticových operací vypadá následujícím způsobem:

```
1000 H=1E-10                    nastavení tlum. faktoru
1010 MAT D=IDN                  jednotková matice
1020 MAT D=(H)*D                faktor na diagonále
1030 MAT D=D+B                  přičtení faktoru k diag.
1040 MAT D=INV(D)               inverze matice
```

```
1050 MAT P=D*C                          výpočet oprav parametrů
1060 FOR I=1 TO L                       testování,je-li oprava
1070 IF ABS(P(I)/A(I))>0.5 THEN 1100    menší než 50% parametru
1080 NEXT I
1090 GOTO 1120                          když ano skok na 1120
1100 H=H*10                             když ne zvětšení h a
1110 GOTO 1010                          opakovánani postupu
1120 MAT A=A+P                          přičtení oprav k paramet.
1130 GOTO 500                           a skok na další iteraci
```

Firemní programy pro řešení regresních úloh se vyznačují některými nedostatky, jako např.

- vložená data se neuchovávají, ale přímo se zabudují do sum
- v případě pol. regrese nedovolují vynechat některé členy
- neposkytují směrodatné odchylky parametrů
- nedovolují transformovat vložená data

Zmíněné nedostatky lze nalézt i v softwaru takové firmy,jako HP. Důsledkem toho uživatelé počítačů používají koupený drahý software je krátce, než jsou schopni vytvořit si vlastní, který by lépe vyhověl jejich požadavkům.

Během dlouholeté praxe s regresními výpočty jsem navrhl prototyp univerzálního regresního programu, který byl postupně adaptován na 7 typech počítačů (HP9830A, HP9825A, ZX81, ZX Spectrum, IQ151, HP9845A, TNS) podle možností jednotlivých počítačů. Tento program se vyznačuje několika rysy:

1. je důsledně klíčován; Pokud počitač nemá možnost rozdělení programu do klíčů,je řízen prostřed. asi 10 kláves (INKEY$)
2. při vstupu dat se dotazuje, je-li začátek nebo pokračování, čímž dovoluje rozšířit již existující soubor dat v paměti

3. dovoluje transformaci dat vložením 3 transformačních vztahů: pro x, y a zpětnou y. Výpočet regrese probíhá s použitím transformovaných proměnných. Pro výstup tabulky na tiskárnu nebo obrázku na plotter je možno cyklicky měnit režim původních nebo transformovaných proměnných
4. typ polynomu je zadáván počtem parametrů a jednotlivými exponenty u proměnné x, případně počtem parametrů a prvním exponentem u tzv. pravidelného polynomu
5. vlastní výpočet používá důsledně výše popsaný iterační postup
6. při výstupu parametrů jsou uváděny i směrodatné odchylky parametrů, dovolující statistické testování jejich významnosti
7. při výstupu tabulky jsou kromě vložených hodnot zobrazeny i vypočtené, rozdíly a rozdíly v %. Dovoluje-li to šířka papíru je užitečné i grafické zobrazení reziduí přímo vedle tabulky
8. výstup všech možných statistických charakteristik, sumy čtverců odchylek, směrodatné odchylky a střední relativní odchylky
9. grafický výstup na obrazovce nebo na plotteru sestává z dotazu na SCALE a zvlášť kreslení os, popisu os, zakreslení bodů ve zvolené velikosti a zvoleným druhem značky a zakreslení křivky zvoleným druhem čáry a zvoleným počtem přímk. úseků
10. program dovoluje vylučování odlehlých bodů (pomocí vah, které jsou nastaveny na 0 z původní hodnoty 1) což dovoluje i opětovné vrácení vyloučeného bodu.

Takový typ programu pro polynomickou regresi lze považovat za určitý náznak expertního systému v oboru regresního zpracování experimentálních dat, protože poskytnutím všech numerických a grafických výstupů dovoluje v dialogovém režimu tvůrčím způsobem zpracovávat experimentální data z nejrůznějších oborů.