Vysoká škola strojní a textilní v Liberci

nositelka Řádu práce

Fakulta strojní

Obor 23 - 40 - 8

Automatizované systémy řízení výrobních procesů ve strojírenství

Katedra technické kybernetiky

V L I V D É L K Y S L O V A P O Č Í T A Č E N A P Ř E S N O S T V Ý S L E D K U

Jarmila Babcová

Vedoucí práce: Doc.Ing. Ján Alaxin, CSc. - - VŠST Liberec

Konzultant: Ing. Miroslav Clehla, CSc. - - VŠST Liberec

Rozsah práce a příloh:

Počet stran	-	54
Počet příloh	-	8
Počet tabulek	-	10
Počet obrázků	-	4
Počet výkresů	-	0

Počet modelů nebo jiných příloh - 0

V Liberci dne 15.9.1982

Vysoká škola: VŠST Liberec Fakulta: strojní

Katedra: technické kybernetiky Školní rok: 1981/82

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

pro s. Jarmilu B a b c o v o u

obor automatizované systémy řízení výrobních procesů ve strojírenství

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: Vliv délky slova počítače na přesnost výsledku

Zásady pro vypracování:

1. Prostudujte z literatury vliv délky slova počítače na přesnost výsledku.
2. Pro zvolené algoritmy ověřte vlivy na přesnost. Uvažujte problémy řešení pro sumu čísel setříděných vzestupně a sestupně, výpočet kovarianční matice, řešení úloh lineární algebry - inverze, determinant, řešení soustavy rovnic.
3. Navrhněte možnosti zvýšené přesnosti výpočtu.

SEVT - 49 395 0 II/81 JČT 8 - 874320-81

Rozsah **grafických prací:**

Rozsah průvodní zprávy: 40 - 50 stran

Seznam odborné literatury:

/1/ Peterka, V.: A Square Root Filter for Real Time Multivariate Regression; Kybernetika č.1, 1975.

/2/ Ralston, A.: Základy numerické matematiky; Academia Praha 1973.

/3/ Děmikovič, B.P., Mason, I.A.: Základy numerické matematiky; Praha SNTL 1966.

/4/ Fadějev, D.K., Fadějevová, V.N.: Numerické metody lineární algebry; Praha SNTL 1964.

/5/ Časopisy a algoritmy CACM.

Vedoucí diplomové práce: Doc.Ing. Ján Alaxin, CSc.
Ing. Miroslav Olehla, CSc.

Datum zadání diplomové práce: 15.9.1981

Termín odevzdání diplomové práce: 4.6.1982

VYSOKÁ ŠKOLA STROJNÍ A TEXTILNÍ LIBEREC fakulta strojního inženýrství

Doc.Ing.Ján Alaxin,CSc.
Vedoucí katedry

Doc.RNDr Bohuslav Stříž,CSc.
Děkan

V Liberci dne 11.9. 19 81

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

V Liberci dne 15.5.1982

O B S A H

Použitá označení:

x^* ..... přesná hodnota čísla x

x ..... přibližná hodnota čísla x

ε_x ..... chyba přibližného čísla x

abs_x ..... absolutní chyba přibližného čísla x

rel_x ..... relativní chyba přibližného čísla x

rel_{1x} ..... relativní chyba přibližného čísla x při použití jednoduché aritmetiky

rel_{2x} ..... relativní chyba přibližného čísla x při použití dvojnásobné aritmetiky

S_{1x} ..... výsledek součtu řady čísel obsahující x sčítanců získaný výpočtem v jednoduché aritmetice

S_{2x} ..... výsledek součtu řady čísel obsahující x sčítanců získaný výpočtem ve dvojnásobné aritmetice

S_{1S} ..... výsledek součtu řady čísel setříděných sestupně / jednoduchá aritmetika /

S_{2V} ..... výsledek součtu řady čísel setříděných vzestupně / dvojnásobná aritmetika /

S_{1V} ..... výsledek součtu řady čísel setříděných vzestupně / jednoduchá aritmetika /

S_{2S} ..... výsledek součtu řady čísel setříděných sestupně / dvojnásobná aritmetika /

1. Ú V O D

V posledních desetiletích pronikly počítače téměř do všech oblastí lidské činnosti. Počítač vznikl v příznivých podmínkách na přelomu první a druhé poloviny našeho století jako jeden z projevů nastupující vědeckotechnické revoluce. Ta otevřela v dnešním životě společnosti nový rozměr jejího vývoje a lidských dějin vůbec. Rozvoj vědy a na ní založené techniky je pro naši dobu a společnost nezbytným předpokladem veškerého dalšího vývoje; člověk zasáhl nebývalým způsobem do života společnosti i přírody, což vyvolává v mnoha případech negativní jevy jako jsou např. poruchy přírodní rovnováhy, informační exploze apod. Společenská podmíněnost vědy a jejího uplatnění má podstatný význam při zabezpečování všech funkcí, které moderní společnost musí mít.

Úspěchy vědy v matematice, atomovém výzkumu, biologii a kybernetice umožnily realizovat takové technické prostředky, které mohou úspěšně nahrazovat nejen lidskou fyzickou práci, ale významnou měrou racionalizovat a umocňovat i práci duševní.

Kybernetická zařízení dnes přebírají logické, řídící a kontrolní funkce člověka v mnoha sférách lidské činnosti. Explozívní růst dat, obklopujících člověka, může racionálně a efektivně zpracovat právě počítač.

V podmínkách ČSSR je hlavním úkolem současné hospodářské politiky zvyšování efektivnosti rozvoje národního hospodářství. Tímto problémem se zabývalo květnové zasedání ÚV KSČ v roce 1974 "K otázkám vědeckotechnického rozvoje čs. národního hospodářství", na němž mezi rozhodujícími prostředky ke splnění cílů byla jmenována kybernetizace výrobních a řídících procesů a byl zdůrazněn význam nových prostředků výpočet-

ní, sdělovací a regulační techniky pro zásadní přestavbu řízení.

Otázkou rozvoje elektroniky a mikroelektroniky se zabýval také XVI.sjezd KSČ v dubnu 1981. Sjezd zdůraznil, že se počítá s předstihem rozvoje elektroniky před ostatními strojírenskými obory, neboť důsledná elektronizace a automatizace uplatněná v rozhodujících odvětvích přináší značné zvýšení společenské produktivity práce, snížení spotřeby surovin, paliv a energie. Je proto nutné zvládnout sériovou výrobu integrovaných obvodů pro mikroprocesorovou techniku, zejména paměti, procesory atd.

V současném období je připravován dlouhodobý program rozvoje elektroniky, který má dosahovat jak řešení základních problémů tohoto odvětví ve výzkumu, v mezinárodní spolupráci, zejména se Sovětským svazem a socialistickými státy, v posílení výrobních kapacit, tak i program zavádění elektroniky do jednotlivých odvětví národního hospodářství. XVI.sjezd KSČ v hlavních směrech hospodářského a sociálního rozvoje ČSSR na léta 1981-85 uložil zvýšením výroby o 40-50% vytvářet podmínky pro elektronizaci národního hospodářství. K tomu je třeba v daleko větší míře využívat mezinárodní dělbu práce, zejména spolupráci v rámci socialistické ekonomické integrace.

Je nesporným faktem, že automatizace a použití počítačů jsou schopny způsobit změny v životě lidstva, že však změny nezpůsobí počítače samy o sobě, ale pouze jako nástroj v rukou člověka. Každý velký technologický pokrok znamenal vedle velkého přínosu i mnohá potenciální nebezpečí při zneužití. Bylo tomu tak u atomové bomby, raket a nejinak je tomu u počítačů. A je jen otázkou etickou, morální a ekonomickou, jak budou počítače využívány.

Vzhledem k řadě problémů, které vznikají při řešení úloh na samočinných počítačích, je tato diplomová práce věnována popisu a ověření některých vlivů na přesnost řešení. Pro tento účel jsou zvoleny často používané principy výpočtů, jako je např. součet řady čísel setříděných vzestupně a sestupně, inverze, řešení soustavy lineárních rovnic apod. Na těchto výpočtech je demonstrováno, jak může být ovlivněn výsledek při různé technice programování a jak lze zabezpečit zlepšení jeho přesnosti. Vliv délky slova počítače nabývá svého významu rovněž vzhledem k širokému zavádění mikroprocesorů do různých oborů a citlivost algoritmu na délku slova počítače a spotřeba paměti je často omezující podmínkou.

tači jsou více méně za[illegible]

[illegible] hodnot numerickou metodou, kon-

2. CHYBY VÝPOČTŮ NA SAMOČINNÝCH POČÍTAČÍCH

2.1 Druhy chyb při numerických výpočtech

Hodnoty získané výpočtem na samočinném číslicovém počítači jsou více méně zatíženy určitou chybou. Je to způsobeno např. nepřesností zadaných hodnot, numerickou metodou, konstrukcí počítače atd.

Chyby vznikající při výpočtu lze rozdělit do několika skupin /1/:

1/ Chyba metody - chyby vzniklé matematickou formulací, která většinou nevyjadřuje přesně skutečný děj; je těžké a někdy i nemožné řešit úlohu v přesné formulaci, úlohu proto zaměníme jinou, přibližnou úlohou, jejíž výsledky jsou blízké výsledkům původním

2/ Chyba zbytková - funkce, které se vyskytují v matematických vzorcích, jsou často určeny nekonečnými posloupnostmi nebo řadami, např. $\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} \frac{x^7}{7!} + \cdots$
Některé matematické problémy lze řešit jen pomocí nekonečného procesu, jehož limita je hledaným řešením. Protože nekonečný proces nemůžeme prakticky realizovat, řešíme jej jako proces s konečným počtem kroků. Vzniká tím vlastně přerušení nebo nedokončení procesu, v důsledku toho se ovšem dopouštíme chyby.

3/ Chyba počáteční /vrozená/ - je způsobena tím, že se v matematických formulích vyskytují číselné parametry, jejichž hodnotu lze určit jen přibližně; tak jsou

určeny např. všechny fyzikální konstanty; dále jsou to čísla, která nemůžeme vyjádřit konečným počtem číslic, jako např. 2, apod.

4/ Chyby operací - vznikají početními operacemi, které provádíme s přibližnými čísly, chyby výchozích čísel se tak přenášejí v jisté míře do výsledku; v tomto smyslu jsou chyby operací neodstranitelné;

5/ Chyby zaokrouhlením - při vyjádření racionálních čísel v desítkové nebo jiné soustavě může být vpravo od desetinné čárky nekonečný počet číslic /např. nekonečný periodický desetinný zlomek/, v počítači však máme k dispozici konečný počet míst pro zobrazení čísla a musíme proto provést zaokrouhlení; z téhož důvodu zaokrouhlujeme konečná čísla, u nichž je počet číslic větší než počet míst, které má k dispozici počítač;
také při převodu konečného racionálního čísla do číselné soustavy, ve které pracuje počítač /obvykle do dvojkové soustavy/, může vzniknout číslo periodické, které je nutno zaokrouhlit.

2.2 Absolutní a relativní chyba

Přibližnou hodnotu čísla x^* označme x /1/. Pak se chybou ε_x přibližného čísla x nazývá zpravidla rozdíl mezi odpovídající přesnou hodnotou x^* a hodnotou přibližnou:

$$\varepsilon_x = x^* - x \qquad /2.2.1/$$

Je-li $x^* > x$, je chyba kladná a $\varepsilon_x > 0$ a naopak.

Abychom určili přesné číslo x^* , tj. přesnou hodnotu, musíme k přibližnému číslu x přičíst jeho chybu ε_x

$$x^* = x + \varepsilon_x \qquad /2.2.2/$$

Znaménko chyby ε_x však většinou neznáme. Proto se zavádí <u>absolutní chyba</u> přibližného čísla x

$$abs_x = |\varepsilon_x| = |x^* - x| \qquad /2.2.3/$$

Vzhledem k tomu, že obvykle x^* odhadujeme, provádíme vlastně odhad absolutní chyby.

<u>Relativní chyba</u> rel_x přibližného čísla x je dána poměrem absolutní chyby k absolutní hodnotě příslušného přesného čísla

$$rel_x = \frac{abs_x}{|x^*|} \qquad /2.2.4/$$

Podobně jako u absolutní chyby budeme provádět odhad relativní chyby podle výrazu

$$rel_x \approx \frac{abs_x}{|x|} \qquad /2.2.5/$$

2.3 Odhad absolutní chyby základních aritmetických operací

Sčítání /1/: $\varepsilon_{x+y} = (x^* + y^*) - (x + y) = (x^* - x) + (y^* - y) = \varepsilon_x + \varepsilon_y$

$$abs_{x+y} = |\varepsilon_x + \varepsilon_y| \leq |\varepsilon_x| + |\varepsilon_y|$$

$$abs_{x+y} \leq abs_x + abs_y$$

Absolutní chyba algebraického součtu několika přibližných čísel je nejvýše rovna součtu absolutních chyb těchto čísel.

Odhad absolutní chyby součtu nemůže být menší než odhad chyby nejméně přesného ze sčítanců, tj. toho ze sčítanců, jehož absolutní chyba je největší. Ani sebevětší přesnost ostatních sčítanců nemůže zlepšit přesnost součtu. Nemá tedy smysl nechávat u přesnějších sčítanců zbytečná desetinná místa.

Existuje proto praktické pravidlo pro součet přibližných čísel s různou absolutní chybou:

1/ Vybereme čísla, jejichž desetinný rozvoj končí dříve než u ostatních /tj. jejichž poslední číslice má nejvyšší řád/ a necháme je beze změny.

2/ Ostatní čísla zaokrouhlíme tak, aby obsahovala o jedno desetinné místo více než čísla z bodu 1/.

3/ Takto upravená čísla sečteme.

4/ Výsledek zaokrouhlíme tak, že poslední číslici vynecháme a předposlední upravíme podle pravidla o zaokrouhlování čísel - viz /4/.

Odčítání: $\varepsilon_{x-y} = (x^* - y^*) - (x - y) = (x^* - x) - (y^* - y) = \varepsilon_x - \varepsilon_y$

$$abs_{x-y} = |\varepsilon_x - \varepsilon_y| \leq |\varepsilon_x| + |\varepsilon_y|$$

Násobení:

$$\varepsilon_{x\cdot y} = (x^* \cdot y^*) - (x \cdot y) = (x + \varepsilon_x)\cdot(y + \varepsilon_y) - (x \cdot y) = \varepsilon_x \cdot y + \varepsilon_y \cdot x \Big/_{\varepsilon_x \varepsilon_y \to 0}$$

$$abs_{x\cdot y} = |\varepsilon_x \cdot y + \varepsilon_y \cdot x| \leq |y| \cdot |\varepsilon_x| + |x| \cdot |\varepsilon_y|$$

$$abs_{x\cdot y} \leq |y| \cdot abs_x + |x| \cdot abs_y$$

Dělení: $\varepsilon_{x/y} = \frac{x^*}{y^*} - \frac{x}{y} = \frac{x^* y - y^* x}{y^* y} = \frac{(x+\varepsilon_x)y - (y+\varepsilon_y)x}{(y+\varepsilon_y)\,y} = \frac{\varepsilon_x \cdot y - \varepsilon_y \cdot x}{y^2}$

$$abs_{x/y} = \left|\frac{\varepsilon_x \cdot y - \varepsilon_y \cdot x}{y^2}\right| = \frac{|\varepsilon_x \cdot y - \varepsilon_y \cdot x|}{|y^2|} \leq \frac{|y| abs_x + |x| abs_y}{|y^2|}$$

Při provádění těchto operací na počítači dochází obvykle k zaokrouhlování výsledku, neboť jeho zapsání je provedeno konečným počtem číslic. Označme postupně absolutní chybu zobrazení výsledku v počítači pro operace sčítání, odčítání, násobení a dělení α , β , μ , δ . Absolutní chyby operací jsou potom dány vztahy:

sčítání $\quad abs_{x+y} \leq abs_x + abs_y + \alpha$

odčítání $\quad abs_{x-y} \leq abs_x + abs_y + \beta$

násobení $abs_{x\cdot y} \leq |y|\cdot abs_x + |x|\cdot abs_y + \mu$

dělení $abs_{x/y} \leq \frac{|y|\cdot abs_x + |x|\cdot abs_y}{|y^2|} + \delta$

2.4 Odhad relativní chyby základních aritmetických operací

Označme /1/ : $r_x = \frac{\varepsilon_x}{x}$, pak $|r_x| = \left|\frac{\varepsilon_x}{x}\right| = \frac{|\varepsilon_x|}{|x|} = \frac{abs_x}{|x|} \approx rel_x$

Použijme pro označení relativních chyb zobrazení výsledků základních operací v důsledku zaokrouhlení postupně $\bar{\alpha}$, $\bar{\beta}$, $\bar{\mu}$, $\bar{\delta}$. O velikosti těchto chyb lze obecně říci, že jejich hodnota nepřesáhne $5\cdot10^{-d}$, kde d je počet desítkových cifer výsledku zobrazeného v počítači po zaokrouhlení.

Sčítání: $r_{x+y} = \frac{\varepsilon_{x+y}}{x+y}$, po odvození : $rel_{x+y} \leq \frac{rel_x\cdot|x| + rel_y\cdot|y|}{|x+y|} + \bar{\alpha}$

Odčítání: $r_{x-y} = \frac{\varepsilon_{x-y}}{x-y}$, po odvození : $rel_{x-y} \leq \frac{rel_x\cdot|x| + rel_y\cdot|y|}{|x-y|} + \bar{\beta}$

Násobení: $r_{x\cdot y} = \frac{\varepsilon_{x\cdot y}}{x\cdot y}$, po odvození : $rel_{x\cdot y} \leq rel_x + rel_y + \bar{\mu}$

Dělení: $r_{x/y} = \frac{\varepsilon_{x/y}}{x/y}$, po odvození : $rel_{x/y} \leq rel_x + rel_y + \bar{\delta}$

Uveďme příklad, který objasní, proč je nutné brát v úvahu chyby vyskytující se u samočinného počítače.

Máme odečíst čísla: a = 48,243

b = 48,222

předpokládejme, že $abs_a \leq 0{,}0005$

$abs_b \leq 0{,}0005$

potom c = a - b = 48,243 - 48,222 = 0,021

$abs_c \leq abs_a + abs_b = 0{,}0005 + 0{,}0005 = 0{,}001$

$rel_a \leq \frac{abs_a}{a} = \frac{0{,}0005}{48{,}243} = 0{,}00001$

$rel_b \leq \frac{abs_b}{b} = \frac{0{,}0005}{48{,}222} = 0{,}00001$

$rel_c \leq \frac{abs_c}{c} = \frac{0{,}001}{0{,}021} = 0{,}05$

Odhad relativní chyby rozdílu je tedy asi 5000krát větší než odhad relativní chyby menšence a menšitele. Došlo tedy ke ztrátě přesnosti. Tomu lze zabránit dvěma způsoby:

1/ vyčíslit hodnoty menšence a menšitele na větší počet platných cifer, ovšem jsme omezeni délkou slova počítače;

2/ vhodně upravit výpočetní postup /např. nahrazením rozdílu dvou blízkých čísel jejich součtem/.

2.5 Zobrazování čísel v počítači

Konstanty mohou být čísla celá nebo racionální. V případě zobrazení celých čísel je přesnost daného čísla dána délkou slova tj. počtem bitů použitých k zobrazení. Minimální a maximální zobrazitelné číslo je

- v přímém kódu: $-(2^{\alpha-1}-1), \ldots, -0, +0, \ldots, (2^{\alpha-1}-1)$
- v doplňkovém kódu: $-2^{\alpha-1}, \ldots, -0, +0, \ldots, (2^{\alpha-1}-1)$

V přímém zobrazení racionálních čísel je situace komplikovanější. Čísla bývají ve většině případů v semilogaritmickém tvaru normalizována, čímž se zvýší přesnost zobrazení mantisy. Tuto výhodu však ztrácíme při aritmetických operacích sčítání a odčítání.

Pro čísla s velkým rozdílem polohy desetinné čárky a dále pro čísla, která převodem z desítkové soustavy do dvojkové mohou být obrazy ryzých zlomků periodickými zlomky, např.:

$$0,1_{10} = 0,000\overline{1100}_2 = 0,\overline{19}_{16} = 0,0\overline{6314}_8$$

Mantisa zobrazuje přesně jen čísla, která jsou vyjádřena platnými číslicemi lineárního rozvoje, tj. zobrazitelná v počtu bitů vyhražených pro mantisu včetně dané normalizace, která může být dvojková, osmičková nebo šestnáctková.

Pro náš příklad je zobrazení pro počet bitů mantisy d_m =8

$0{,}11001100 * 2^3$ resp.

$0{,}19 * 16^0$ resp.

$0{,}630 * 8^1$ /ne však $0{,}631 * 8^1$/

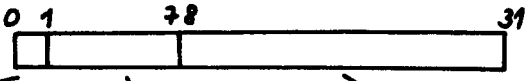
U počítačů řady EC je možné ve FORTRANu vyjádřit celá čísla pro d = 32 v doplňkovém kódu, čísla reálná jsou uložena buď v jednoduché délce slova počítače nebo jako dvojnásobná délka /2 slova/.

čísla celá:

0000 0000 0000 0000 0000 0000 0000 0001 = 2^0 = 1

1111 1111 1111 1111 1111 1111 1111 1111 = $-2^{31} + 2^{31}-1$ = -1

čísla racionální:



znaménko mantisy; 0 - záporné, 1 - kladné

exponent; /zvětšený o 64/

normalizovaná mantisa; /6 hexadecimálních míst/

0000 0000 0001 0000 0000 0000 0000 0000 = $\frac{1}{16} \cdot 16^{-64}$

0111 1111 1111 1111 1111 1111 1111 1111 = $1-16^{-6} \cdot 16^{64}$

1111 1111 1111 1111 1111 1111 1111 1111 = $-(1 - 16^{-6}) \cdot 16^{64}$

Rozsah reálných čísel je m:

$$\frac{1}{16} \cdot 16^{-64} \leq m \leq (1 - 16^{-6}) \cdot 16^{63}$$

$$\sim 5{,}4.10^{-79} \leq m \leq 7.10^{75}$$

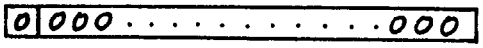
V případě zobrazení dvojnásobné délky / DOUBLE PRECISION / se prodlouží mantisa i do druhého slova, čímž se zvýší přesnost zobrazení /na 14 hexadecimálních míst/, nerozšíří se však interval zobrazitelných čísel.

Příklad 1.:

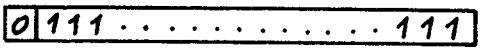
$2^{31} - 1$	0	111.....111
$+ 2^{31} - 1$	0	111.....111
$= -2$	1	111.....110

Příklad 2.: 2^{31} * 

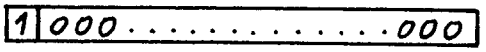
$+ 2^{31}$ * 

= 0 1 

Příklad 3.: 2^{32}* = 0 1 

Příklad 4.: $2^{31} - 1$ 

\+ 1 

2^{31} 

* ... označené výrazy nelze zadat v programu jako konstanty

2.6 Vliv pořadí aritmetických operací na přesnost výsledku

V případě aritmetických operací je nutné si uvědomit, že nejen čísla, ale také výsledky jsou ukládány do slova počítače, který má omezený rozsah. Rozdílné výsledky mohou vzniknout změnou pořadí prováděných operací.

Např. a - b + c je možno vyčíslit buď zleva doprava, tj. (a - b) + c nebo zprava doleva jako a + (- b + c). Vzhledem k platnosti asociativního a komutativního zákona bychom očekávali výsledek v obou případech stejný.

Např. pro sedm platných míst však získáme:

a = 36 165,14 ; b = 36 165,11 ; c = 0,1042785

1/ (a - b) + c : 36 165,14 0,0300000

\- 36 165,11 0,1042785

00 000,03 0,1342785

2/ a + (- b + c) :

\- 36 165,11 36 165,14

\+ 00 000,1042785 -36 165,01

\- 36 165,01 00 000,13

Obdobné problémy vznikají součtem součinů čísel různé velikosti: $\sum a_i.b_i$

$a_1 = 1000$ $b_1 = 1000$ $a_1 \cdot b_1 = 1000000$

$a_2 = 0,001$ $b_2 = 0,001$ $a_2 \cdot b_2 = 0,000001$

V případě sedmi platných míst získáme výsledek 1.10^6 .

3. OVĚŘENÍ VLIVU DÉLKY SLOVA POČÍTAČE NA PŘESNOST VÝSLEDKU

3.1 Součet řady čísel /programovací jazyk - RPP-BASIC/

Vznik chyb při algebraických výpočtech prováděných s malými čísly je možné ověřit sestavením a výpočtem několika jednoduchých programů v programovacím jazyku RPP-BASIC na počítači RPP-16S.

Programy jsou sestavené pro sčítání řady:

1/ 0,1 + 0,2 + 0,3 + ... + 0,9 + 1,0 = 5,5

2/ 0,01 + 0,02 + 0,03 + ... + 0,99 + 1,00 = 50,5

3/ 0,001 + 0,002 + 0,003 + ... + 0,999 + 1,000 = 500,5

Sčítání řady čísel setříděných vzestupně

ad 1/
```
5 LET S = 0.0
10 FOR I = 1 TO 10
20 LET S = S + I  0.1
30 NEXT I
40 PRINT S
50 END
```
výsledek: S = 5.5

ad 2/
```
5 LET S = 0.0
10 FOR I = 1 TO 100
20 LET S = S + I  0.01
30 NEXT I
40 PRINT S
50 END
```
výsledek: S = 50.5

ad 3/
```
5 LET S= 0.0
10 FOR I = 1 TO 1000
20 LET S = S + I  0.001
30 NEXT I
40 PRINT S
50 END
```
výsledek: S = 500.5

Získané výsledky jsou ve všech případech přesné.

Sčítání řady čísel setříděných sestupně

Jedná se o součet stejných řad čísel jako v předchozím případě, ovšem jsou opačně setříděné:

ad 1/
```
 5 LET S = 0.0
10 LET X = 1.0
20 FOR I = 1 TO 10
30 LET S = S + X
40 LET X = X - 0.1
50 NEXT I
60 PRINT S
70 END
```
výsledek: S = 5.5

ad 2/
```
 5 LET S = 0.0
10 LET X = 1.0
20 FOR I = 1 TO 100
30 LET S = S + X
40 LET X = X - 0.01
50 NEXT I
60 PRINT S
70 END
```
výsledek: S = 50.5

ad 3/
```
 5 LET S = 0.0
10 LET X = 1.0
20 FOR I = 1 TO 1000
30 LET S = S + X
40 LET X = X - 0.001
50 NEXT I
60 PRINT S
70 END
```
výsledek: S = 500.523

Výsledek třetího programu je již zatížen určitou chybou. Přesnější výsledky tedy získáme při použití řady čísel setříděných vzestupně.

Sčítání řady stejně velkých čísel

V programech je sčítáno číslo 0,1 , přičemž počet sčítanců se mění: 10 ; 100; 1000; 10000

```
 5 LET S = 0.0
10 FOR I = 1 TO 10 / 100; 1000; 10000/
20 LET S = S + 0.1 / 0.01; 0.001; 0.0001/
30 NEXT I
40 END
```

Získané výsledky jsou zachyceny v tab.3.1.1:

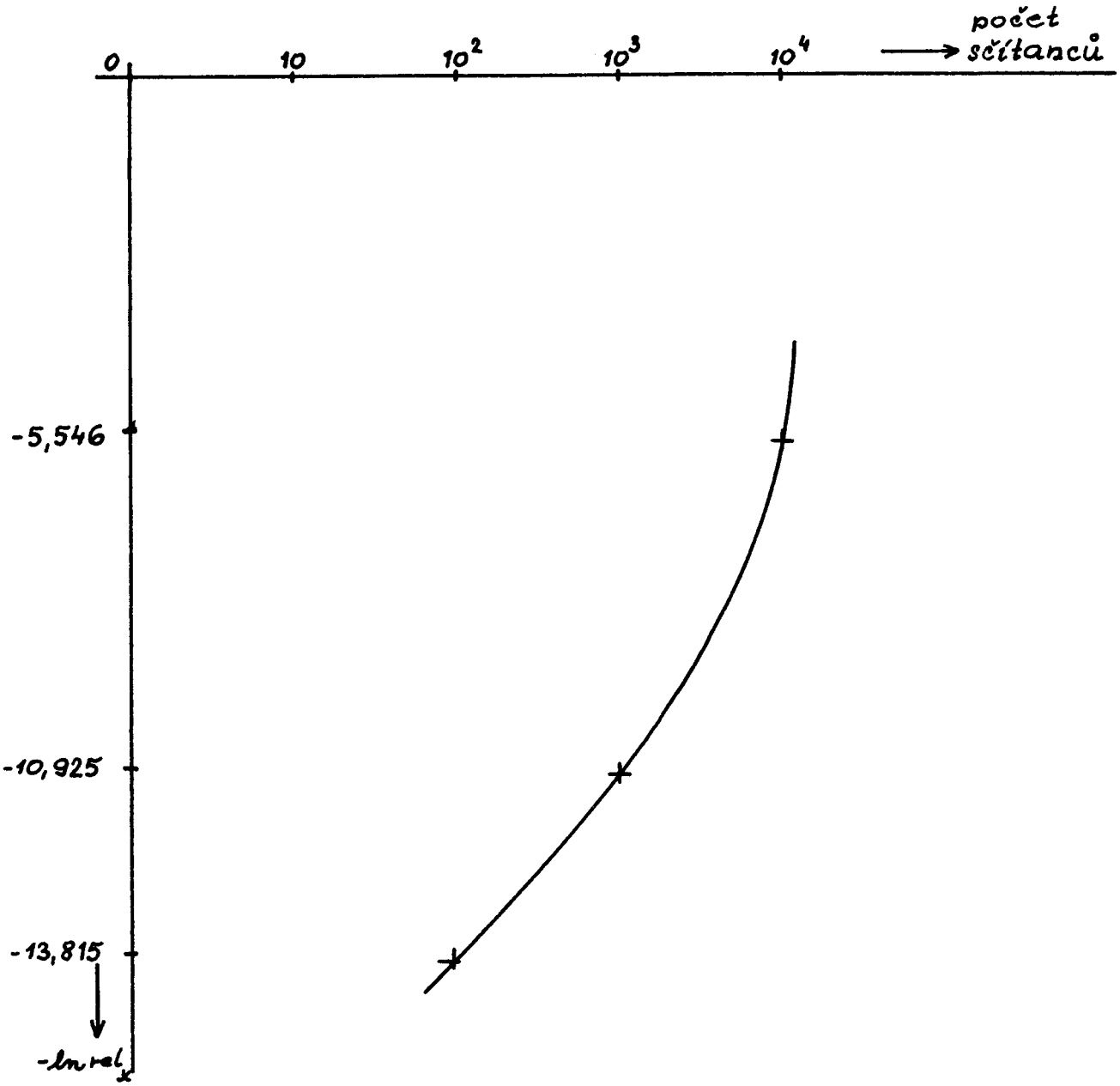
velikost sčítanců	počet sčítanců	výsledek	relativní chyba
0,1	10	1,0	0
0,1	10^2	9,99999	$0,1.10^{-5}$
0,1	10^3	100,004	$0,4.10^{-4}$
0,1	10^4	1000,39	$0,39.10^{-3}$
0,01	10	0,1	0
0,01	10^2	0,999999	$0,1.10^{-5}$
0,01	10^3	9,99982	$0,18.10^{-4}$
0,01	10^4	100,39	$0,39.10^{-2}$
0,001	10	0,01	0
0,001	10^2	0,1	0
0,001	10^3	0,999973	$0,27.10^{-4}$
0,001	10^4	9,99741	$0,26.10^{-3}$
0,0001	10	0,001	0
0,0001	10^2	0,01	0
0,0001	10^3	0,099996	$0,4.10^{-4}$
0,0001	10^4	0,999542	$0,46.10^{-3}$

tab.3.1.1

Relativní chyby výsledků jsou počítané podle vztahů /2.2.1/, /2.2.3/, /2.2.4/.

Z výpočtů relativních chyb jednotlivých výsledků je zřejmé, že velikost relativní chyby roste se zmenšující se hodnotou sčítanců a s narůstajícím počtem sčítanců.

Závislost relativní chyby na počtu sčítanců je znázorněna graficky na obr.3.1.2, a to pro sčítání čísla 0,01.



obr.3.1.2

Všechny následující programy jsou sestaveny v programovacím jazyku FORTRAN IV. a počítané na číslicovém počítači EC-1033, přičemž každý program je sestaven ve dvou verzích, a to vždy v jednoduché a ve dvojnásobné délce slova počítače.

3.2 Součet řady stejně velkých čísel /viz příloha č.1/

Program pro tento výpočet je obdobný jako předchozí program sestavený v jazyku RPP-BASIC. Je sestaven pro sčítání čísel 0,1 ; 0,01 ; 0,001 ; ... až 1.10^{-15}, přičemž uvedená čísla jsou sčítána vždy 10krát, 100krát atd. až 10^5krát. Součty jsou počítány v jednoduché i ve dvojnásobné délce slova počítače. Při porovnání výsledků získaných v obou těchto případech bylo zjištěno, že k výrazně přesnějšímu výsledku dospějeme při programování ve dvojnásobné délce slova, a to především při sčítání velmi malých čísel. Se zmenšující se velikostí sčítanců narůstá chyba výsledku získaného použitím jednoduché aritmetiky. Výsledky získané výpočtem ve dvojnásobné aritmetice jsou také zatíženy určitou chybou, ta je ovšem mnohonásobně menší nežli v předchozím případě.

Je zajímavé, že v případě sčítání čísla 1.10^{-10} a menších se dospělo ke značně nepřesným výsledkům, a to i při výpočtu v DOUBLE PRECISION. Chyba se v těchto případech se zmenšující se velikostí sčítanců rychle zvětšuje, jelikož všechny získané výsledky jsou přibližně stejné; protože se jedná vždy o stejný počet sčítanců, měly by se správně výsledky zmenšovat. Obdržené výsledky se od správných hodnot liší rozdílem čtyř až šesti řádů.

Výsledky součtů jednotlivých řad a jejich relativní chyby /vypočítané podle vztahů /2.2.1/,/2.2.3/ a /2.2.4/ / jsou zaneseny v tab.3.2.1.

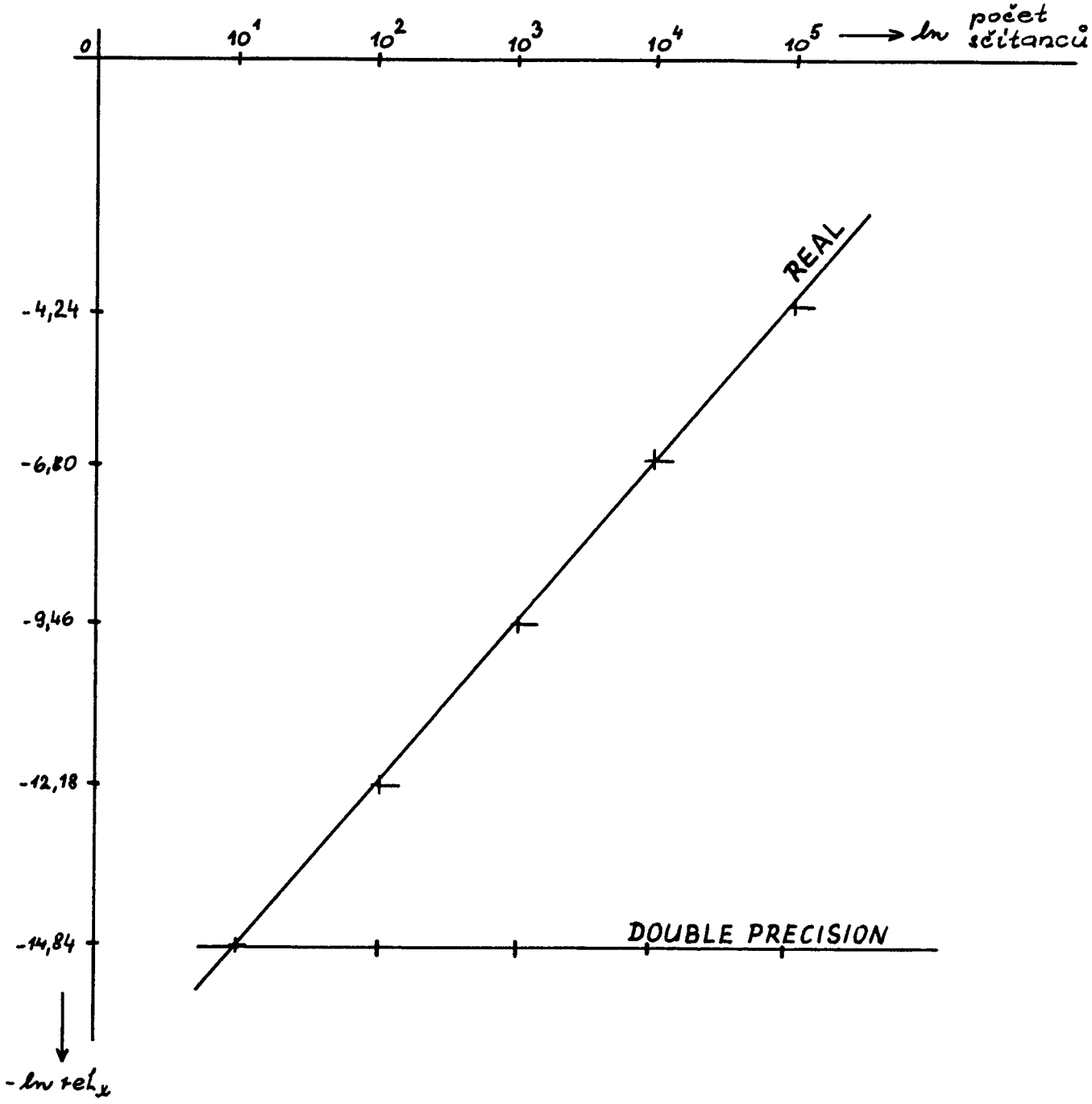
velikost sčítanců	počet sčítanců	výsledek /REAL/	relativní chyba	výsledek /D.P./	relativní chyba
10^{-1}	10	0,9999996	$3,576.10^{-7}$	0,9999996	$3,576.10^{-7}$
	10^2	9,9999485	$5,149.10^{-6}$	9,9999964	$3,576.10^{-7}$
	10^3	99,9922180	$7,782.10^{-5}$	99,9999642	$3,576.10^{-7}$
	10^4	998,8881836	$1,112.10^{-3}$	999,9996424	$3,576.10^{-7}$
	10^5	9856,101563	$1,439.10^{-2}$	9999,996424	$3,576.10^{-7}$
10^{-2}	10	$0,9999996.10^{-1}$	$9,537.10^{-7}$	$0,9999997.10^{-1}$	$2,086.10^{-7}$
	10^2	0,999999	$9,537.10^{-6}$	0,9999997	- " -
	10^3	9,9993467	$6,533.10^{-5}$	$0,9999997.10^{1}$	- " -
	10^4	99,9527741	$4,723.10^{-4}$	$0,9999997.10^{2}$	- " -
	10^5	982,4326172	$1,757.10^{-2}$	$0,9999997.10^{3}$	- " -
10^{-3}	10	$0,9999986.10^{-2}$	$1,326.10^{-5}$	$0,9999999.10^{-2}$	$0,890.10^{-8}$
	10^2	$0,9999937.10^{-1}$	$6,318.10^{-6}$	$0,9999999.10^{-1}$	- " -
	10^3	0,9999877	$1,222.10^{-5}$	0,9999999	- " -
	10^4	9,9950437	$3,956.10^{-4}$	0,9999993	- " -
	10^5	99,3047943	$6,952.10^{-3}$	99,9999931	- " -
10^{-4}	10	$0,9999987.10^{-3}$	$1,233.10^{-6}$	$0,9999999.10^{-3}$	$9,800.10^{-8}$
	10^2	$0,9999871.10^{-2}$	$1,287.10^{-5}$	$0,9999999.10^{-2}$	- " -
	10^3	$0,9998267.10^{-1}$	$1,732.10^{-4}$	$0,9999999.10^{-1}$	- " -
	10^4	0,9995956	$4,044.10^{-4}$	0,9999999	- " -
	10^5	9,9259911	$7,401.10^{-3}$	9,9999990	- " -
10^{-5}	10	$0,9999990.10^{-4}$	$9,711.10^{-7}$	$0,9999999.10^{-4}$	$2,530.10^{-8}$
	10^2	$0,9999878.10^{-3}$	$1,218.10^{-5}$	$0,9999999.10^{-3}$	- " -
	10^3	$0,9999137.10^{-2}$	$8,626.10^{-5}$	$0,9999999.10^{-2}$	- " -
	10^4	$0,9981960.10^{-1}$	$1,804.10^{-3}$	$0,9999999.10^{-1}$	- " -
	10^5	0,9956774	$4,323.10^{-3}$	0,9999999	- " -
10^{-6}	10	$0,9999994.10^{-5}$	$5,710.10^{-7}$	$0,9999994.10^{-5}$	$5,710.10^{-7}$
	10^2	$0,9999941.10^{-4}$	$5,919.10^{-6}$	$0,9999994.10^{-4}$	- " -
	10^3	$0,9998283.10^{-3}$	$1,717.10^{-4}$	$0,9999994.10^{-3}$	- " -
	10^4	$0,9989287.10^{-2}$	$1,071.10^{-3}$	$0,9999994.10^{-2}$	- " -
	10^5	$0,9817123.10^{-1}$	$1,829.10^{-2}$	$0,9999994.10^{-1}$	- " -
10^{-7}	10	$0,9999994.10^{-6}$	$5,710.10^{-7}$	$0,9999997.10^{-6}$	$3,436.10^{-7}$
	10^2	$0,9999986.10^{-5}$	$1,389.10^{-6}$	$0,9999997.10^{-5}$	- " -
	10^3	$0,9998829.10^{-4}$	$1,171.10^{-4}$	$0,9999997.10^{-4}$	- " -
	10^4	$0,9990942.10^{-3}$	$9,058.10^{-4}$	$0,9999997.10^{-3}$	- " -
	10^5	$0,9804369.10^{-2}$	$1,956.10^{-2}$	$0,9999997.10^{-2}$	- " -

tab.3.2.1

pokrač. tab.3.2.1:

10^{-8}	10	$0,9999974.10^{7}$	$2,617.10^{6}$	$0,9999997.10^{7}$	$2,725.10^{7}$
	10^{2}	$0,9999948.10^{6}$	$5,118.10^{6}$	$0,9999997.10^{6}$	- " -
	10^{3}	$0,9999899.10^{5}$	$1,003.10^{5}$	$0,9999997.10^{5}$	- " -
	10^{4}	$0,9997582.10^{4}$	$2,418.10^{4}$	$0,9999997.10^{4}$	- " -
	10^{5}	$0,9832233.10^{3}$	$1,678.10^{2}$	$0,9999997.10^{3}$	- " -
10^{-9}	10	$0,9999976.10^{8}$	$2,404.10^{6}$	$0,9999999.10^{8}$	$1,393.10^{7}$
	10^{2}	$0,9999934.10^{7}$	$6,596.10^{6}$	$0,9999999.10^{7}$	- " -
	10^{3}	$0,9996851.10^{6}$	$3,149.10^{5}$	$0,9999999.10^{6}$	- " -
	10^{4}	$0,9995781.10^{5}$	$4,219.10^{4}$	$0,9999999.10^{5}$	- " -
	10^{5}	$0,9910618.10^{4}$	$8,938.10^{3}$	$0,9999999.10^{4}$	- " -
10^{-10}	10	$0,7091863.10^{8}$	6,092	$0,7091869.10^{8}$	6,092
	10^{2}	$0,7091841.10^{7}$	6,092	$0,7091869.10^{7}$	- " -
	10^{3}	$0,7091791.10^{6}$	6,092	$0,7091869.10^{7}$	- " -
	10^{4}	$0,7085881.10^{5}$	6,086	$0,7091869.10^{7}$	- " -
	10^{5}	$0,7006556.10^{4}$	6,007	$0,7091869.10^{7}$	- " -

Pro lepší přehled je na obr. 3.2.2 uvedeno grafické znázornění závislosti relativní chyby na počtu sčítanců, jedná se o sčítání čísla 1.10^{-1} /další závislosti jsou uvedeny v příloze č.8 /.



Popis proměnných v programu:

I ... proměnná typu INTEGER - její hodnota se mění v cyklu od 1 do 5 /po jedné/

J ... proměnná typu INTEGER - její hodnota se mění v cyklu od 1 do 10 /po jedné/

M ... proměnná typu INTEGER - její hodnota je 10^I; vyjadřuje počet sčítanců v dané řadě

N ... proměnná typu INTEGER - její hodnota je 10^J; slouží k vyjádření velikosti sčítanců

P,Q ... proměnná typu REAL resp. DOUBLE PRECISION - v programu je jí přiřazena hodnota 1/N a vyjadřuje velikost sčítanců

S,T ... proměnná typu REAL resp. DOUBLE PRECISION - její počáteční hodnota je nulová, postupně je k ní přičítáno dané číslo; její konečná hodnota udává výsledek součtu řady M sčítanců o velikosti proměnné P resp.Q

3.3 Součet řady čísel setříděných sestupně /viz příloha č.2/

Dále byl sestaven program pro součet následující řady čísel:

$$1 + 0{,}5 + 0{,}25 + 0{,}125 + 0{,}0625 + \ldots$$

Počet sčítanců této geometrické řady se v cyklech postupně zvyšuje: 10 ; 10^2; 10^3 ; ... Ovšem již při sčítání 10^3 čísel došlo v počítači k podtečení v důsledku velmi malých hodnot sčítanců. Mohou proto být porovnány pouze výsledky součtu řady deseti a sta čísel.

Popis proměnných v programu:

I ... proměnná typu INTEGER - nabývá v cyklu hodnot 1,2

M ... proměnná typu INTEGER - nabývá v cyklu hodnot 10^I; udává počet sčítanců v řadě

K ... proměnná typu INTEGER - nabývá v cyklu hodnot 1 až M ; udává počet sčítanců v řadě

P,Q ... proměnná typu REAL resp. DOUBLE PRECISION - počáteční hodnota proměnné je 2, jejím dělením dělitelem o velikosti 2 získáme čísla požadované řady

S,T ... proměnná typu REAL resp. DOUBLE PRECISION - její počáteční hodnota je nulová, postupně k ní jsou přičítána jednotlivá čísla řady; její konečná hodnota udává výsledek součtu řady

Při součtu deseti čísel jsou výsledky v jednoduché i ve dvojnásobné aritmetice shodné, ovšem již při součtu sta sčítanců se výsledky liší.

Jelikož jde v tomto případě o součet řady utvořené z geometrické posloupnosti, zjistíme přesný výsledek výpočtem podle vzorce pro součet geometrické řady:

$$S_n = a_1 + a_1 q + a_1 q^2 + a_1 q^3 + \ldots + a_1 q^{n-1}$$

v našem případě

$$S_{10} = 1 + 0{,}5 + 0{,}25 + 0{,}125 + \ldots$$

$$\Rightarrow \text{ kvocient } \quad q = 0{,}5$$

Součet prvních n členů pro $q < 1$:

$$S_n = \frac{a_n (1 - q^n)}{1 - q} \qquad /3.3.1/$$

po dosazení:

$$S_{10} = \frac{1(1 - 0{,}5^{10})}{1 - 0{,}5} = 1{,}998046875 \qquad /3.3.2/$$

$$S_{100} = \frac{1(1 - 0{,}5^{100})}{1 - 0{,}5} = 2{,}0 \qquad /3.3.3/$$

Výsledky získané z počítače:

$S_{1,10} = 0{,}1998046875 \cdot 10^1$ $\qquad$ $S_{2,10} = 0{,}1998046875 \cdot 10^1$

$S_{1,100} = 0{,}1999999046 \cdot 10^1$ $\qquad$ $S_{2,100} = 2{,}0$

Podle vztahů /2.2.1/, /2.2.3/, /2.2.4/ lze opět vypočítat relativní chyby výsledků:

$rel_{S_{1,10}} = 0$ $\qquad$ $rel_{S_{2,10}} = 0$

$rel_{S_{1,100}} = 4{,}77 \cdot 10^{-7}$ $\qquad$ $rel_{S_{2,100}} = 0$

Z výpočtů je patrné, že při sčítání menšího počtu čísel vystačíme s jednoduchou délkou slova počítače. Při sčítání většího počtu čísel je ale pro získání přesného výsledku nutné zvolit výpočet ve dvojnásobné aritmetice.

3.4 Součet řady čísel setříděných vzestupně /viz příloha č.3/

Program je sestaven pro součet stejné řady čísel jako v předchozím případě, ale řada je opačně setříděná.

Popis proměnných v programu

I ... proměnná typu INTEGER - nabývá v cyklu hodnot 1 až 5

M ... proměnná typu INTEGER - nabývá v cyklu hodnot 10^I; udává počet sčítanců v řadě

K,L ... proměnné typu INTEGER - jejich hodnota se mění v cyklech od 1 do M

P,Q ... proměnná typu REAL resp. DOUBLE PRECISION - její počáteční hodnota je 2 a snižuje se dělením dělitelem o velikosti 2 /dělení se provádí /M+1/krát /; získaná nejmenší hodnota se v dalším vyklu násobí činitelem 2 a takto získaná čísla se sčítají /vytváří vzestupnou řadu/

S,T ... proměnná typu REAL resp. DOUBLE PRECISION - její počáteční hodnota je nulová, postupně jsou k ní přičítána jednotlivá čísla dané řady; konečná hodnota této proměnné udává výsledek součtu řady získaný v jednoduché resp. dvojnásobné délce slova počítače

Všechny výsledky získané v tomto programu se shodují s výsledky programu předchozího, tj. při součtu řady sestupně setříděné. Tzn., že se neprojevilo očekávané zlepšení přesrosti

Výsledků. Je možné zdůvodnit to tím, že počet řádově stejně velkých sčítanců je nízký /dva až tři/, a nedochází proto při jejich načítání k výraznějšímu posunu platných číslic z nižších do vyšších řádů.

Proto byl zvýšen počet sčítanců tak, že každé číslo dané řady je sčítáno desetkrát. Počet řádově stejných sčítanců se tak zvýšil na dvacet až třicet. Sčítaná řada nyní tedy obsahuje sto a tisíc sčítanců.

Při součtu sta čísel byl výsledek stejný jak v případě řady sestupně tak vzestupně setříděné /přesné výsledky při použití dvojnásobné aritmetiky/.

Při součtu jednoho tisíce čísel jsou již zřejmé rozdíly mezi výsledky, a to nejen rozdíly v přesnosti při použití jednoduché a dvojnásobné délky slova počítače, ale také rozdíly mezi výsledky součtu řady setříděné sestupně a vzestupně:

Výsledky součtu řady setříděné sestupně:

$S_{1S} = 0{,}1999984741.10^2$ $\qquad$ $S_{2S} = 0{,}2.10^2$

Výsledky součtu řady setříděné vzestupně:

$S_{1V} = 0{,}1999998474.10^2$ $\qquad$ $S_{2V} = 0{,}2.10^2$

Přesné výsledky součtů získáme jako desetinásobky výsledků součtů geometrické řady /viz vztahy /3.3.2/, /3.3.3/ /, tj. pro součet tisíce čísel:

$$S^* = 0{,}2.10^2$$

Výsledky získané při použití dvojnásobné aritmetiky jsou tedy v obou případech součtů přesné. Jejich absolutní i relativní chyba je nulová.

Výsledky, které jsme obdrželi při použití jednoduché délky slova počítače, jsou zatíženy určitými chybami. Při porovnání přesnosti obou programů můžeme opět vypočítat

absolutní a relativní chyby jejich výsledků:

Řada setříděná sestupně -

$$\varepsilon_{1S} = S^* - S_{1S} = 0{,}2 \cdot 10^2 - 0{,}1999984741 \cdot 10^2 = 1{,}5259 \cdot 10^{-4}$$

$$|\varepsilon_{1S}| = abs_{1S} = 1{,}5259 \cdot 10^{-4}$$

$$rel_{1S} = \frac{|\varepsilon_{1S}|}{S^*} = \frac{1{,}5259 \cdot 10^{-4}}{0{,}2 \cdot 10^2} = 7{,}6295 \cdot 10^{-6}$$

Řada setříděná vzestupně -

$$\varepsilon_{1V} = S^* - S_{1V} = 0{,}2 \cdot 10^2 - 0{,}1999998474 \cdot 10^2 = 1{,}526 \cdot 10^{-5}$$

$$|\varepsilon_{1V}| = abs_{1V} = 1{,}526 \cdot 10^{-5}$$

$$rel_{1V} = \frac{|\varepsilon_{1V}|}{S^*} = \frac{1{,}526 \cdot 10^{-5}}{0{,}2 \cdot 10^2} = 7{,}63 \cdot 10^{-7}$$

Z porovnání velikosti absolutních a relativních chyb výsledků je zřejmé, že výsledek získaný součtem řady čísel setříděných vzestupně je přesnější /jeho chyba je přibližně desetkrát menší/ nežli výsledek součtu řady setříděné sestupně.

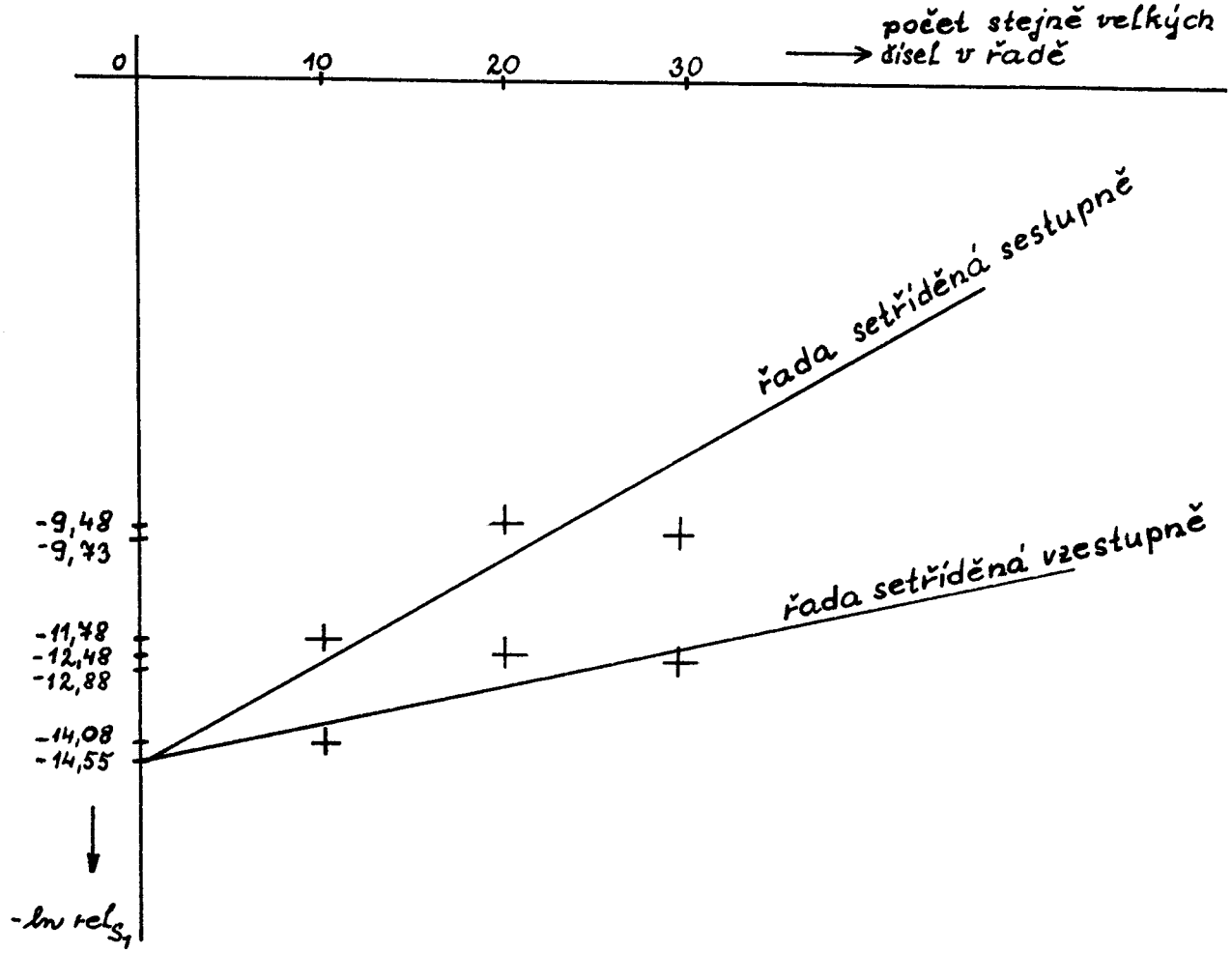
Stejné programy byly použity ještě dvakrát, a to jen s tím rozdílem, že počet řádově stejně velkých sčítanců byl zvýšen z deseti na dvacet a třicet. Výsledky získané při použití dvojnásobné délky slova počítače byly ve všech případech opět zcela přesné. Výsledky součtů obdržené při použití jednoduché aritmetiky a velikosti jejich relativních chyb jsou zachyceny v tab.3.4.1.

výsledek součtu	přesný výsledek	relativní chyba
$0{,}3999998474.10^2$	$0{,}4.10^2$	$3{,}815.10^{-6}$
$0{,}3999969482.10^2$	$0{,}4.10^2$	$7{,}629.10^{-5}$
$0{,}5999998474.10^2$	$0{,}6.10^2$	$2{,}543.10^{-6}$
$0{,}5999954224.10^2$	$0{,}6.10^2$	$5{,}963.10^{-5}$

tab.3.4.1

Pro výpočet relativní chyby bylo použito vztahů /2.2.1/, /2.2.3/ a /2.2.4/.

Pokud opět porovnáme velikosti relativních chyb, vidíme, že jsou i v těchto případech větší chybou zatíženy výsledky součtu řady sestupně setříděné. Rozdíl mezi přesností výsledků obou řad čísel se zvětšuje zároveň s narůstajícím počtem sčítanců, což je patrné z grafického vyjádření /obr.3.4.2/.



obr.3.4.2

3.5 Inverze matice /viz příloha č.4/

Byl použit program pro výpočet inverzní matice eliminační metodou.

Popis hlavního programu - jeho proměnných:

A(I,J) ... proměnná typu REAL /DOUBLE PRECISION/ - originální čtvercová matice o rozměru /N,N/

X(I,J) ... proměnná typu REAL /DOUBLE PRECISION/ - matice o rozměru /N,N/, která vznikne zinvertováním matice

N ... proměnná typu INTEGER - udává počet řádků a počet sloupců používaných matic

DET ... proměnná typu REAL /DOUBLE PRECISION/ - udává hodnotu determinantu matice

Y(I,J) ... proměnná typu REAL /DOUBLE PRECISION/ - matice o rozměru /N,N/, která je výsledkem součinu matice **A(I,J)** a **X(I,J)**

Z(I,J) ... proměnná typu REAL /DOUBLE PRECISION/ - matice o rozměru /N,N/, vznikne dvojnásobným zinvertováním matice **A(I,J)**

P(I,J) ... proměnná typu REAL /DOUBLE PRECISION/ - matice o rozměru /N,N/, vznikne odečtením /rozdílem/ matice **Z(I,J)** a **A(I,J)**

I,J ... proměnné typu INTEGER - udávají počet řádků resp. sloupců matice

Podprogram pro inverzi matice - SUBROUTINE INVMAT/N,AA,CC,DET/

AA(I,J) ... proměnná typu REAL /DOUBLE PRECISION/ - originální čtvercová matice o rozměru /N,N/

CC(I,J) ... proměnná typu REAL /DOUBLE PRECISION/ - matice o rozměru /N,N/, vznikne zinvertováním matice **AA(I,J)**

DET ... proměnná typu REAL /DOUBLE PRECISION/ - udává hodnotu determinantu matice

Podprogram pro součin matic - SUBROUTINE SOUCIN/M,N,L,AA,BB,CC/

AA(I,J) ... proměnná typu REAL /DOUBLE PRECISION/ - matice o rozměru /M,N/

BB(J,K) ... proměnná typu REAL /DOUBLE PRECISION/ - matice o rozměru /N,L/

CC(I,K) ... proměnná typu REAL /DOUBLE PRECISION/ - matice o rozměru /M,L/, vznikne součinem matic **AA(I,J)** a **BB(J,K)**

M ... proměnná typu INTEGER - udává počet řádků matice **AA(I,J)** a **CC(I,J)**

N ... proměnná typu INTEGER - udává počet sloupců matice **AA(I,J)** a počet řádků matice **BB(J,K)**

L ... proměnná typu INTEGER - udává počet sloupců matice **BB(J,K)** a **CC(I,K)**

Podprogram pro rozdíl matic - SUBROUTINE ROZDIL/M,N,E,B,C/

E(I,J) ... proměnná typu REAL /DOUBLE PRECISION/ - matice o rozměru /M,N/, které od sebe odečítáme
B(I,J)

C(I,J) ... proměnná typu REAL /DOUBLE PRECISION/ - matice o rozměru /M,N/, vznikne rozdílem matic **E(I,J)** a **B(I,J)**

Celý program je sestaven takto:

Prvky originální matice, s níž se dále pracuje, jsou vytvořeny následujícím způsobem:

$$a_{ij} = \frac{1}{(i+j)} \quad , \qquad \begin{matrix} i & \text{je 1 až N} \\ j & \text{je 1 až N} \end{matrix}$$

/N je v našem případě rovno pěti/; tzn., že velikost každého prvku získáme jako převrácenou hodnotu součtu pozice daného

prvku na i-tém řádku a j-tém sloupci matice.

Po vyvolání podprogramu pro inverzi matice A(I,J) dostaneme inverzní matici X(I,J) . Aby bylo možné ověřit přesnost výpočtu, je třeba provést další operace.

Nejprve vynásobíme matici inverzní a originální. V případě, že výsledky jsou přesné, bychom měli dostat výslednou matici jednotkovou /je označená Y(I,J) /. Ze získaných výsledků je ale zcela zřejmé, že při výpočtu došlo k nepřesnostem. Pouze v případě jediného prvku je jeho hodnota přesná, a to prvku ve čtvrtém řádku a v prvním sloupci:

$$y_{41} = 0$$

Největší odchylka od přesného výsledku nastala v případě výpočtu prvku v prvním řádku a pátém sloupci:

$$y_{15} = 0,109375$$

Jelikož se jedná o matici jednotkovou, je správná hodnota daného prvku

$$y^*_{15} = 0$$

Vypočítáme absolutní chybu dle vztahů /2.2.1/ a /2.2.3/:

$$\varepsilon_{15} = y^*_{15} - y_{15} = 0,109375$$

$$abs_{max} = abs_{15} = |\varepsilon_{15}| = 0,109375 \qquad /3.5.1/$$

Relativní chybu nelze vypočítat podle vztahu /2.2.4/, neboť bychom dostali neurčitý výraz - číslice dělená nulou. Je tedy možný pouze odhad relativní chyby podle /2.2.5/:

$$rel_{15} \approx \frac{abs_{15}}{|y_{15}|} = 1$$

V případě, že přesná hodnota prvku má být nulová, bude velikost relativní chyby určená podle /2.2.5/ vždy rovna jedné. Velikost absolutní chyby se rovná velikosti příslušného prvku.

Hodnoty dalších prvků matice Y(I,J) jsou již méně nepřesné. Např. prvek ve druhém řádku a prvním sloupci:

$$y_{21} = 0,1220703125 \cdot 10^{-2}$$

V tomto případě je velikost absolutní chyby:

$$\varepsilon_{21} = y_{21}^{*} - y_{21} = -0,1220703125 \cdot 10^{-2}$$

$$abs_{min} = abs_{21} = |\varepsilon_{21}| = 0,1220703125 \cdot 10^{-2} \qquad /3.5.2/$$

Její hodnota je o dva řády menší nežli hodnota největší absolutní chyby /viz vztah /3.5.1//.

V další části programu je opět vyvolán podprogram pro inverzi matice, a to matice $X(I,J)$. Dostaneme tedy dvojnásobnou inverzi originální matice $A(I,J)$ /výsledná matice $-Z(I,J)$/. V případě, že byly dosavadní výpočty přesné, získali bychom opět matici původní. Ze srovnání obou matic však vyplývá, že nastaly rozdíly mezi vzájemně si odpovídajícími prvky. Aby tyto rozdíly byly zcela zřejmé, je dále vyvolán podprogram pro rozdíl matic $A(I,J)$ a $Z(I,J)$. Z výsledků rozdílu jsou již zcela patrné konkrétní odchylky mezi odpovídajícími si prvky, neboť správně bychom měli dostat matici nulovou.

Pro celkový přehled o přesnosti výpočtu je možné opět určit největší a nejmenší absolutní chybu získaných výsledků. K největší chybě došlo u prvku v prvním řádku a třetím sloupci -

$$p_{13} = 0,1859843731 \cdot 10^{-2}$$

$$abs_{max} = abs_{13} = |a_{13} - z_{13}| = 0,1859843731 \cdot 10^{-2} \qquad /3.5.3/$$

Velikost prvku p_{55} vyšla sice rovna nule, ovšem při srovnání příslušných prvků a_{55} a z_{55} je patrný určitý rozdíl mezi nimi. To znamená, že k chybě dochází i při rozdílu matic, kdy v tomto případě byl výsledek rozdílu zaokrouhlen na nulu. Přesná hodnota tohoto rozdílu je:

$$a_{55} - z_{55} = 0,9999996424 \cdot 10^{-1} - 0,9924829006 \cdot 10^{-1} = 0,0007516741 8$$

Velikost absolutní chyby je tedy:

$$abs_{55} = |\varepsilon_{55}| = 7,5167418 \cdot 10^{-4}$$

K nejmenší chybě ovšem došlo u prvku v pátém řádku a prvním sloupci:

$$p_{51} = 0,4290938377 \cdot 10^{-3}$$

jehož absolutní chyba je

$$abs_{min} = abs_{51} = |\varepsilon_{51}| = 4,290938377 \cdot 10^{-4}$$

Tentýž program pro inverzi matice byl použit při výpočtu ve dvojnásobné délce slova, tzn. že všechny proměnné typu REAL byly změněny na DOUBLE PRECISION.

Již z pouhého porovnání prvků inverzní matice vypočítané v jednoduché a ve dvojnásobné aritmetice jsou očividné dost velké rozdíly mezi jejich hodnotami /i když řádově se hodnoty sobě odpovídajících prvků shodují/. Přesnost obou programů můžeme ovšem porovnat a posoudit až po provedení dalších výpočtů, neboť neznáme přesnou hodnotu inverzní matice.

Prvním z nich je opět součin originální a inverzní matice /tj. $A(I,J)$ a $X(I,J)$, výsledná matice je $Y(I,J)$/. Prvky umístěné na hlavní diagonále mají přesnou hodnotu - všechny jsou rovny jedné. Kromě těchto prvků je přesných ještě pět dalších, které jsou nulové. Hodnoty zbývajících prvků jsou již zatíženy určitou, velmi malou chybou.

Pro názornost vypočítáme opět největší a nejmenší absolutní chybu získaných výsledků.

K největší odchylce od správné hodnoty došlo u prvku v prvním řádku a čtvrtém sloupci:

$$y_{14} = 0,1818989404 \cdot 10^{-10}$$

jehož absolutní chyba je

$$abs_{max} = abs_{14} = |y_{14}^{*} - y_{14}| = 0,1818989404 \cdot 10^{-10}$$

Tuto absolutní chybu je možno srovnat s největší hodnotou absolutní chyby, která vznikla při součtu v jednoduché aritme-

tice /viz vztah /2.2.1//. Z porovnání vyplývá, že velikost absolutní chyby výsledku se použitím dvojnásobné aritmetiky snížila o deset řádů. K nejmenší absolutní chybě došlo v případě dvou prvků, a to u prvku ve druhém řádku a prvním sloupci a u prvku v pátém řádku a prvním sloupci matice:

$$y_{21} = y_{51} = 0,5684341886 \cdot 10^{-13}$$

$$y^*_{21} = y^*_{51} = 0$$

absolutní chyba výsledků

$$abs_{min} = abs_{21} = abs_{51} = |\varepsilon_{21}| = |\varepsilon_{51}| = 0,5684341886 \cdot 10^{-13}$$

Její velikost opět porovnáme s velikostí nejmenší absolutní chyby získané při použití jednoduché aritmetiky /viz vztah /3.5.2//. Absolutní chyba se snížila tentokrát o jedenáct řádů.

V další části programu vypočítáme opět hodnotu originální matice dvakrát zinvertované. V tomto případě jsou získané hodnoty prvků opravdu přesné - při vyčíslení na deset platných míst. Tzn. že chyba vzniklá při inverzi matice je natolik malá, že se při dvojnásobné inverzi zcela vykompenzuje. Velikosti všech prvků původní matice $A(I,J)$ se tedy shodují s velikostí odpovídajících si prvků matice $Z(I,J)$.

Další kontrolou přesnosti je, podobně jako v předchozím programu, rozdíl matice původní a matice dvojnásobně zinvertované. Výsledkem by měla být matice nulová $P(I,J)$.

Přesné hodnoty nabývá pouze prvek v pátém řádku a pátém sloupci matice - p_{55} . Ostatní prvky jsou zatíženy určitou chybou. Největší odchylka od přesného výsledku nastala v případě prvku v prvním řádku a prvním sloupci, jehož hodnota je:

$$p_{11} = 0,2743597016 \cdot 10^{-11}$$

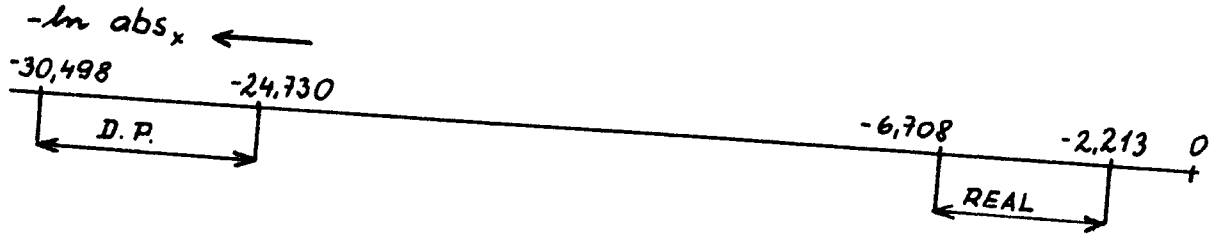
a velikost absolutní chyby

$$abs_{max} = abs_{11} = |\varepsilon_{11}| = 0,2743597016 \cdot 10^{-11}$$

Absolutní chyba se ve srovnání s maximální absolutní chybou získanou při výpočtu v jednoduché aritmetice /viz vztah /3.5.3// snížila o devět řádů.

Intervaly, ve kterých se pohybují hodnoty absolutních chyb výsledků obdržených použitím jak jednoduché tak dvojnásobné aritmetiky jsou znázorněny graficky na obr.3.5.4. /Pro lepší přehled v logaritmických hodnotách z důvodů velkého rozdílu řádů mezi jednotlivými velikostmi chyb./

Součin originální a inverzní matice:



Rozdíl matice originální a dvojnásobně zinvertované:

obr.3.5.4

3.6 Soustava lineárních rovnic /viz příloha č.5/

Jsou použity programy pro průběžný výpočet soustavy lineárních rovnic metodou odmocnin a metodou elementárních rotací. Nejprve si objasníme základní principy obou metod:

Metoda odmocnin /2/

Je dána lineární soustava

$$\underline{\underline{A}}\,\underline{x} = \underline{b} \qquad /3.6.1/$$

kde $\underline{\underline{A}} = (a_{ij})$ je symetrická matice, tj. $\underline{\underline{A}}' = (a_{ji}) = \underline{\underline{A}}$.
V tomto případě lze matici $\underline{\underline{A}}$ vyjádřit ve tvaru součinu dvou navzájem transponovaných trojúhelníkových matic:

$$\underline{\underline{A}} = \underline{\underline{T}}'\underline{\underline{T}} \qquad /3.6.2/$$

kde

$$\underline{\underline{T}} = \begin{pmatrix} t_{11}, t_{12}, \dots, t_{1n} \\ 0, t_{22}, \dots, t_{2n} \\ \dots\dots\dots\dots \\ 0, 0, \dots, t_{nn} \end{pmatrix}; \quad \underline{\underline{T}}' = \begin{pmatrix} t_{11}, 0, \dots, 0 \\ t_{21}, t_{22}, \dots, 0 \\ \dots\dots\dots\dots \\ t_{1n}, t_{2n}, \dots, t_{nn} \end{pmatrix} .$$

Vynásobíme-li $\underline{\underline{T}}'$ a $\underline{\underline{T}}$, dostaneme pro prvky t_{ij} matice $\underline{\underline{T}}$ tyto rovnice:

$$t_{1i}t_{1j} + t_{2i}t_{2j} + \dots + t_{ii}t_{ij} = a_{ij} \quad (i < j)$$
$$t_{1i}^2 + t_{2i}^2 + \dots + t_{ii}^2 = a_{ii}$$

Odtud postupně dostaneme:

$$\left.\begin{aligned} t_{11} &= \sqrt{a_{11}}\,, \quad t_{1j} = \frac{a_{1j}}{t_{11}} \quad (j > 1)\,, \\ t_{ii} &= \sqrt{a_{ii} - \sum_{k=1}^{i-1} a_{ki}^2} \quad (1 < i \le n)\,, \\ t_{ij} &= \frac{a_{ij} - \sum_{k=1}^{i-1} t_{ki}\,t_{kj}}{t_{ii}} \quad (i < j)\,, \\ t_{ij} &= 0 \quad (i > j) \end{aligned}\right\} \qquad /3.6.3/$$

Je-li $t_{ii} \neq 0$ pro všechna i , má soustava právě jedno řešení, neboť

$$\det \underline{\underline{A}} = \det \underline{\underline{T}}' \cdot \det \underline{\underline{T}} = (\det \underline{\underline{T}})^2 = (t_{11} t_{22} \cdots t_{mm})^2 \neq 0$$

Vzhledem ke vztahu /3.6.2/ je maticová rovnice ekvivalentní dvěma maticovým rovnicím

$$\underline{\underline{T}}' \underline{y} = \underline{b} \quad \text{a} \quad \underline{\underline{T}} \underline{x} = \underline{y} \ ;$$

po rozepsání součinů je tedy

$$\begin{aligned} t_{11} y_1 &= b_1 \\ t_{12} y_1 + t_{22} y_2 &= b_2 \\ &\cdots\cdots\cdots\cdots \\ t_{1m} y_1 + t_{2m} y_2 + \cdots + t_{mm} y_m &= b_m \end{aligned} \qquad /3.6.4/$$

a

$$\begin{aligned} t_{11} x_1 + t_{12} x_2 + \cdots + t_{1m} x_m &= y_1 \\ t_{22} x_2 + \cdots + t_{2m} x_m &= y_2 \\ &\cdots\cdots\cdots\cdots \\ t_{mm} x_m &= y_m \end{aligned} \qquad /3.6.5/$$

Odtud postupně dostáváme

$$y_1 = \frac{b_1}{t_{11}}$$

$$y_i = \frac{b_i - \sum_{k=1}^{i-1} t_{ki} y_i}{t_{ii}} \quad (i > 1) \qquad /3.6.6/$$

a

$$x_m = \frac{y_m}{t_{mm}}$$

$$x_i = \frac{y_i - \sum_{k=i+1}^{m} t_{ik} x_k}{t_{ii}} \quad (i < m) \qquad /3.6.7/$$

Pro praktické použití této metody se přímým chodem počítají pomocí vzorců /3.6.3/ a /3.6.6/ postupně koeficienty t_{ij} a y_i $(i = 1, 2, \ldots, m)$ a pak se zpětným chodem počítají pomocí vzorce /3.6.7/ neznámé x_i $(i = m, m-1, \ldots, 1)$.

Metoda elementárních rotací /1/:

Je dána soustava rovnic:

$$a_{11}x_1 + a_{12}x_2 + a_{13}x_3 = a_{14}$$
$$a_{21}x_1 + a_{22}x_2 + a_{23}x_3 = a_{24}$$
$$a_{31}x_1 + a_{32}x_2 + a_{33}x_3 = a_{34}$$

vypočítáme hodnoty konstant C , S podle následujících vzorců:

$$C = \frac{a_{11}}{\sqrt{a_{11}^2 + a_{21}^2}} \quad ; \quad S = -\frac{a_{21}}{\sqrt{a_{11}^2 + a_{21}^2}}$$

když $a_{11} = a_{21} = 0$, potom $C = 1$

$S = 0$

První dvě rovnice soustavy zaměníme rovnicemi:

$$C(a_{11}x_1 + a_{12}x_2 + a_{13}x_3) - S(a_{21}x_1 + a_{22}x_2 + a_{23}x_3) = C \cdot a_{14} - S \cdot a_{24}$$
$$S(a_{11}x_1 + a_{12}x_2 + a_{13}x_3) + C(a_{21}x_1 + a_{22}x_2 + a_{23}x_3) = S \cdot a_{14} + C \cdot a_{24}$$

Tímto způsobem bude koeficient při neznámé x_1 v druhé rovnici roven nule. Podobné operace provedeme s první transformovanou rovnicí a třetí původní rovnicí; získaná druhá rovnice se kombinuje se získanou třetí rovnicí atd.

Takto získáme novou soustavu s trojúhelníkovou maticí, kterou řešíme zpětným chodem.

Tato metoda vyžaduje např. čtyřnásobek výpočetních operací oproti eliminační Gaussově metodě, má však větší stabilitu a je málo citlivá na nepřesnosti vzniklé malými hodnotami determinantů pomocných soustav rovnic.

Obě uvedené metody jsou použity pro průběžné řešení sedmi různých soustav rovnic. Ve všech případech se jedná o soustavu sedmi rovnic, počet neznámých je roven třem

popř. čtyřem.

Vstupní data programů, tj. koeficienty rovnic znázorníme vždy jednoduchou tabulkou.

1.případ /viz tab.3.6.8/:

x_1	x_2	x_3	x_4	y
1	0	0	0	1
1	1	1	1	4
1	2	4	2	9
1	3	9	3	16
1	4	16	4	25
1	5	25	5	36
1	6	36	6	49

tab.3.6.8

x_1, x_2, x_3, x_4 ... koeficienty u neznámých

y ..... pravá strana rovnice

Jedná se o soustavu rovnic se čtyřmi neznámými, kdy koeficient u neznámé x_2 se vždy rovná koeficientu u neznámé x_4; koeficient na pravé straně rovnice je dán součtem všech koeficientů ležících na straně levé.

Řešení této soustavy rovnic metodou elementárních rotací poskytuje přesné výsledky, pokud použijeme dvojnásobnou délku slova počítače; při výpočtu v jednoduché délce slova jsou výsledky zatíženy chybou.

Tato soustava rovnic má nekonečně mnoho řešení, z nichž uvedeme dvě:

1/ $x_1 = x_2 = x_3 = x_4 = 1$

2/ $x_1 = x_3 = 1$

$x_2 = 2$

$x_4 = 0$

Metoda elementárních rotací řeší danou soustavu druhým způsobem na rozdíl od metody odmocnin, při které dochází

k rovnoměrnému rozložení hodnot mezi neznámé veličiny, jak je uvedeno v prvním případě. Při použití metody odmocnin jsou výsledky získané výpočtem v jednoduché aritmetice přesnější nežli je výpočet v DOUBLE PRECISION.

2.případ /viz tab.3.6.9/:

x_1	x_2	x_3	x_4	y
1	0	0	0	1
1	1	1	0	3
1	2	4	0	7
1	3	9	0	13
1	4	16	0	21
1	5	25	0	31
1	6	36	0	43

tab.3.6.9

Tato soustava rovnic je obdobná jako v předchozím případě, ale s tím rozdílem, že koeficienty u čtvrté neznámé jsou vesměs nulové.

Obě metody řeší danou soustavu rovnic přibližně stejnými výsledky; hodnota čtvrté neznámé je ve všech případech nulová. Výsledky výpočtů ve dvojnásobné aritmetice jsou přesné v případě použití metody elementárních rotací, naopak u metody odmocnin je opět přesnější výsledek získaný výpočtem v jednoduché délce slova počítače.

Ze srovnání výsledků obou metod při použití jednoduché aritmetiky je patrné, že přesnější je v tomto případě metoda odmocnin.

3.případ /viz tab.3.6.10/:

x_1	x_2	x_3	y
1	0	0	1
1	1	1	4
1	2	4	9
1	3	9	16
1	4	16	25
1	5	25	36
1	6	36	49

tab.3.6.10

Jedná se o soustavu sedmi rovnic o třech neznámých. Koeficienty u první neznámé jsou vesměs rovny jedné, koeficienty u třetí neznámé jsou vždy druhou mocninou koeficientů u druhé neznámé.

V tomto případě jsme při použití metody odmocnin obdrželi přesnější výsledky výpočtem ve dvojnásobné délce slova počítače, i když rozdíly v porovnání s jednoduchou aritmetikou jsou velmi malé. Proto zde není téměř třeba prodlužovat slovo počítače.

Výsledky výpočtu metodou elementárních rotací jsou méně přesné nežli při výpočtech metodou odmocnin.

4.případ /viz tab.3.6.11/:

x_1	x_2	x_3	x_4	y
1	0	0	0,1	1
1	1	1	0,9	4
1	2	4	2,06	9
1	3	9	3,03	16
1	4	16	3,91	25
1	5	25	5,1	36
1	6	36	4,9	49

tab.3.6.11

Soustava těchto rovnic je obdobná jako soustava v 1.případě, ovšem koeficienty u čtvrté neznámé jsou zatíženy určitou malou poruchou. Soustavu je také možné porovnat se 3.pří-

padem, nyní ale přibývá neznámá x_4. Výsledky by se měly shodovat s předchozími . Můžeme říci, že relativně nejpřesnější výsledky dává metoda elementárních rotací s použitím dvojnásobné délky slova počítače, kdy

$$x_1 = 1,0$$
$$x_2 = 2,0$$
$$x_3 = 1,0$$
$$x_4 = 0,2777136.10^{-15}$$

Hodnota neznámé x_4 se tedy téměř blíží nule.

Ostatní výsledky jsou podobné, větší rozdíly mezi oběma metodami nastaly pouze v případě neznámé x_4, jejíž hodnota se při výpočtu metodou odmocnin zvýšila o deset řádů:

$$x_4 = 0,157028342.10^{-5}$$

5.případ /viz tab.3.6.12/:

x_1	x_2	x_3	x_4	y
1	0	0	0,1	1
1	1	1	-0,1	4
1	2	4	0,06	9
1	3	9	0,03	16
1	4	16	-0,09	25
1	5	25	0,1	36
1	6	36	-0,1	49

tab.3.6.12

Jedná se o soustavu obdobnou jako ve 3.případě, ovšem přibývá zde čtvrtá neznámá, jejíž koeficienty se blíží nule, jsou ale zatíženy určitou poruchou. Výsledek by se měl shodovat s výsledkem soustavy uvedené ve 3.případě, neboť je patrné, že hodnota neznámé x_4 by měla být nulová.

Získané výsledky jsou velmi podobné výsledkům předchozího případu, tzn. že hodnota neznámé x_4 se opět blíží nule. Relativně nejlepší výsledky dostaneme použitím metody elementárních rotací a dvojnásobné aritmetiky.

6.případ /viz tab.3.6.13/:

x_1	x_2	x_3	y
1	0	0	1,1
1	1	1	2,9
1	2	4	7,06
1	3	9	13,03
1	4	16	20,91
1	5	25	31,1
1	6	36	42,9

tab.3.6.13

Soustava je obdobná jako ve druhém případě, ovšem neexistuje zde neznámá x_4 a pravé strany rovnic jsou opět zatížené určitou poruchou.

Mezi výsledky získanými oběma metodami jak v jednoduché tak ve dvojnásobné aritmetice nejsou podstatné rozdíly; hodnoty výsledků jsou méně přesné.

7.případ /viz tab.3.6.14/:

x_1	x_2	x_3	y
1	0,1	0	1
1	0,9	1	3
1	2,06	4	7
1	3,03	9	13
1	3,91	16	21
1	5,1	25	31
1	4,9	36	43

tab.3.6.14

Soustava je obdobná jako v předchozím případě; tentokrát jsou poruchou zatížené koeficienty u neznámé x_2, přičemž hodnoty pravých stran jsou rovny součtu koeficientů stran levých při neuvažování poruchy.

Výsledky jsou opět vesměs méně přesné.

V prvních třech případech, kdy všechny koeficienty měly přesnou hodnotu, se jednalo o soustavy rovnic, ve kterých

byly rovnice lineárně závislé. Z tohoto důvodu mají tyto soustavy rovnic nekonečně mnoho řešení, přičemž použité metody výpočtu nám poskytují vždy některé z těchto možných řešení.

Vlivem korelace koeficientů u některé z neznámých nebo koeficientů na pravých stranách rovnic se daná soustava změní, rovnice již nejsou lineárně závislé a počet řešení soustavy se redukuje na jedno. Z porovnání získaných výsledků je zřejmé, že v případech, kdy poruchou jsou zatíženy koeficienty u neznámých, nejsou jí výsledky natolik ovlivněny. Hodnota neznámé je vždy přibližně nulová. Větší vliv na konečný výsledek má korelace koeficientů na pravých stranách rovnic.

3.7 Determinant /viz příloha č.6/

Je použit program pro výpočet determinantu pomocí elementárních transformací matice.

Princip metody /1/:

Počítejme determinant

$$D = \begin{vmatrix} a_{11}, a_{12}, \dots, a_{1n} \\ a_{21}, a_{22}, \dots, a_{2n} \\ \dots\dots\dots\dots \\ a_{n1}, a_{n2}, \dots, a_{nn} \end{vmatrix}$$

Předpokládejme, že $a_{11} \neq 0$. Vytkněme prvek a_{11} z prvního řádku a dostaneme

$$D_n = a_{11} \begin{vmatrix} 1, {}^1a_{12}, \dots, {}^1a_{1n} \\ a_{21}, a_{22}, \dots, a_{2n} \\ \dots\dots\dots\dots \\ a_{n1}, a_{n2}, \dots, a_{nn} \end{vmatrix} \quad , \text{ kde } {}^1a_{1j} = a_{1j}/a_{11}$$

Nyní od každého řádku, druhým počínaje, odečteme první řádek násobený vždy prvním prvkem příslušného řádku, tj.

${}^1a_{ij} = a_{ij} - a_{i1} \cdot {}^1a_{1j}$ a pak

$$D_n = a_{11} \begin{vmatrix} 1, {}^1a_{12}, \dots, {}^1a_{1n} \\ 0, {}^1a_{22}, \dots, {}^1a_{2n} \\ \dots\dots\dots\dots \\ 0, {}^1a_{n2}, \dots, {}^1a_{nn} \end{vmatrix} = a_{11} \begin{vmatrix} {}^1a_{22}, \dots, {}^1a_{2n} \\ \dots\dots\dots \\ \dots\dots\dots \\ {}^1a_{n2}, \dots, {}^1a_{nn} \end{vmatrix} = a_{11} \cdot D_{n-1}$$

Dále postupujeme stejným způsobem s výpočtem determinantu D_{n-1} , jestliže ${}^1a_{22} \neq 0$. Pokračujeme-li ve výpočtu tímto způsobem, dostaneme, že hledaný determinant je roven součinu vedoucích prvků

$$D = a_{11} \cdot {}^{1}a_{22} \cdot \ldots \cdot {}^{(n-1)}a_{nn}$$

Ukáže-li se, že v některém kroku ${}^{(i-1)}a_{ii} = 0$ nebo je velmi malé číslo /což by snížilo přesnost výpočtu/, lze vyměnit pořadí řádků a sloupců determinantu tak, aby v levém horním rohu byl dostatečně velký prvek. Tedy nejvýhodnější pro přesnost výpočtu je převést do levého horního rohu ten prvek determinantu, který má největší absolutní hodnotu.

Pro výpočet byl zvolen determinant sestavený ze stejných prvků jako v případě výpočtu inverzní matice, tzn.

$$A_{ij} = \frac{1}{(i+j)} \qquad \text{pro } i,j = 1 \div 5$$

Při výpočtu v jednoduché délce slova počítače je výsledek

$$D = 0{,}1380650829 \cdot 10^{-13}$$

Hodnota determinantu v případě použití dvojnásobné délky slova je

$$D = 0{,}1459931789 \cdot 10^{-13}$$

Zcela přesnou hodnotu determinantu dané matice neznáme. Je možné pouze porovnat oba výsledky, které se vzájemně dosti liší, což opět svědčí o vlivu délky slova počítače na přesnost výsledku; předpokládáme, že výsledek výpočtu ve dvojnásobné aritmetice je přesnější.

3.8 Výpočet kovarianční matice /viz příloha č.7./

Budiž $X = (X_1, \ldots, X_m)'$ náhodný vektor /5/.
Existují-li střední hodnoty $EX_1, \ldots, EX_m$, nazýváme $EX = (EX_1, \ldots, EX_m)$ střední hodnotou náhodného vektoru $X = (X_1, \ldots X_m)$. O vektoru X pak říkáme, že má první momenty. Zcela analogicky se definuje střední hodnota matice, jejímiž prvky jsou náhodné veličiny. Jestliže $EX_k^2 < \infty$ pro $k = 1,2, \ldots, m$, říkáme, že vektor X má konečné druhé momenty a definujeme kovarianci $cov(X_i, X_j)$ náhodných veličin X_i a $X_j (1 \leq i, j \leq m)$ vztahem:

$$cov(X_i, X_j) = E(X_i - EX_i)(X_j - EX_j)$$

Mějme nyní dva náhodné vektory $X = (X_1, \ldots, X_m)$, $Y(Y_1, \ldots, Y_m)'$ s konečnými druhými momenty. Kovarianční matice vektorů X a Y se značí $cov(X,Y)$ a je definována vztahem:

$$cov(X,Y) = E(X-EX)(Y-EY)'.$$

V tomto případě, stejně jako při výpočtu determinantu, neznáme přesné výsledky, proto je možné pouze konstatovat, že mezi výsledky programů v jednoduché a ve dvojnásobné aritmetice jsou opět patrné rozdíly. Předpokládáme opět, že přesnější výsledky dává použití dvojnásobné délky slova počítače.

4. Z Á V Ě R

V této diplomové práci byl na několika výpočtech ověřen vliv délky slova počítače na přesnost výsledku. Pro toto ověření byl zvolen výpočet součtu řady stejně velkých čísel /při změně počtu sčítanců/, součet řady čísel vzestupně a sestupně setříděných, výpočet inverzní matice, výpočet soustavy lineárních rovnic, výpočet determinantu a kovariančí matice. Byly použity programy sestavené v programovacím jazyku FORTRAN IV., a to každý program vždy pro jednoduchou a pro dvojnásobnou délku slova počítače.

Při porovnání výsledků získaných výpočtem v jednoduché a ve dvojnásobné aritmetice je většinou zcela zřejmý rozdíl v jejich přesnosti. Při použití dvojnásobné aritmetiky jsou obdržené výsledky vesměs daleko přesnější. Pouze v některých případech soustav lineárních rovnic je zbytečné prodlužovat délku slova počítače. Pro výpočet soustav lineárních rovnic byly použity dvě metody - metoda odmocnin a metoda elementárních rotací. Z porovnání výsledků získaných výpočtem podle obou uvedených metod vyplývá, že v průměru jsou přesnější výsledky metody elementárních rotací, a to především při použití dvojnásobné aritmetiky. Souvisí to také s tím, že k výrazné kumulaci chyb dochází při součtu součinů, přičemž těchto operací používá metoda odmocnin ve větším měřítku.

Při součtu malých, stejně velkých čísel roste chyba výsledku zároveň se zvyšujícím se počtem sčítanců v řadě. Proto je vhodné, zejména pro větší počet sčítaných čísel, použít prodloužené slovo počítače.

V případě součtu řady různých čísel je vhodné pro získání přesného výsledku rovněž použít dvojnásobnou aritmetiku.

V těchto případech byla chyba součtu vždy nulová. Při výpočtu v jednoduché aritmetice jsou výsledky přesnější, je-li sčítaná řada čísel setříděná vzestupně. To je možno vysvětlit tím, že při součtu čísel vzestupně setříděných dochází k postupnému posuvu platných číslic z nižších do vyšších řádů, a proto se natolik neprojevuje chyba zaokrouhlení v důsledku omezeného rozsahu slova počítače.

Největší rozdíly v přesnosti výsledků jednoduché a dvojnásobné aritmetiky nastaly v případě výpočtu inverzní matice, kdy prodloužením délky slova počítače klesla relativní chyba výsledku až o jedenáct řádů.

Určité rozdíly nastaly i ve výsledcích výpočtu determinantu a kovarianční matice.

Shrneme-li všechny dosažené výsledky, je zřejmé, že ve většině případech výpočtů je možné pro zpřesnění výsledků doporučit použití dvojnásobné délky slova počítače. Neplatí to ovšem ve všech případech, je proto vhodné každý řešený problém předem analyzovat a na základě výsledků analýzy navrhnout optimální délku slova počítače. Analýza by se měla týkat již samotné použité metody, neboť většina numerických metod je zatížena určitou chybou, jak již bylo uvedeno v kapitole 2.1. Dále je třeba analyzovat aritmetické operace používané v dané metodě. Například při součinu dvou čísel s určitým počtem desetinných míst dostaneme výsledek, jehož počet desetinných míst je mnohem vyšší, a v důsledku omezeného rozsahu slova počítače je proto značně zaokrouhlen. V neposlední řadě je nutný rozbor řešeného problému z důvodů samotné omezené délky slova počítače, do kterého jsou ukládána veškerá data, mezivýsledky i konečné výsledky výpočtů.

Seznam použité literatury:

/1/ Olehla,M. - Tišer,J.: Praktické použití Fortranu.
Praha, NADAS, 1979

/2/ Děmidovič,B.P. - Maron,J.A.: Základy numerické matematiky.
Praha, SNTL, 1966

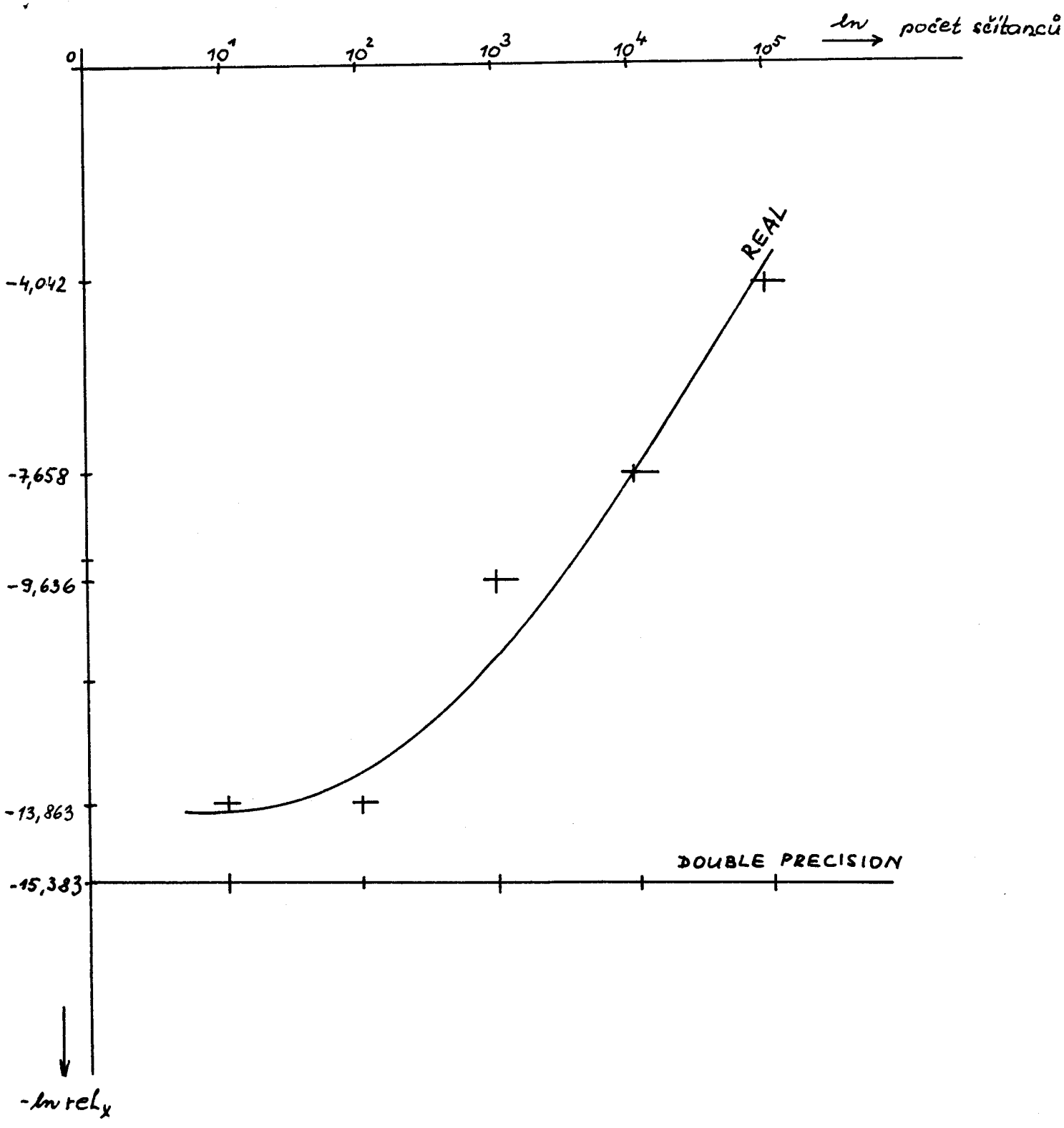
/3/ Faddějev,D.K. - Faddějevová,V.N.: Numerické metody
lineární algebry. Praha, SNTL, 1964

/4/ Ralston,A.: Základy numerické matematiky.
Praha, Academia, 1973

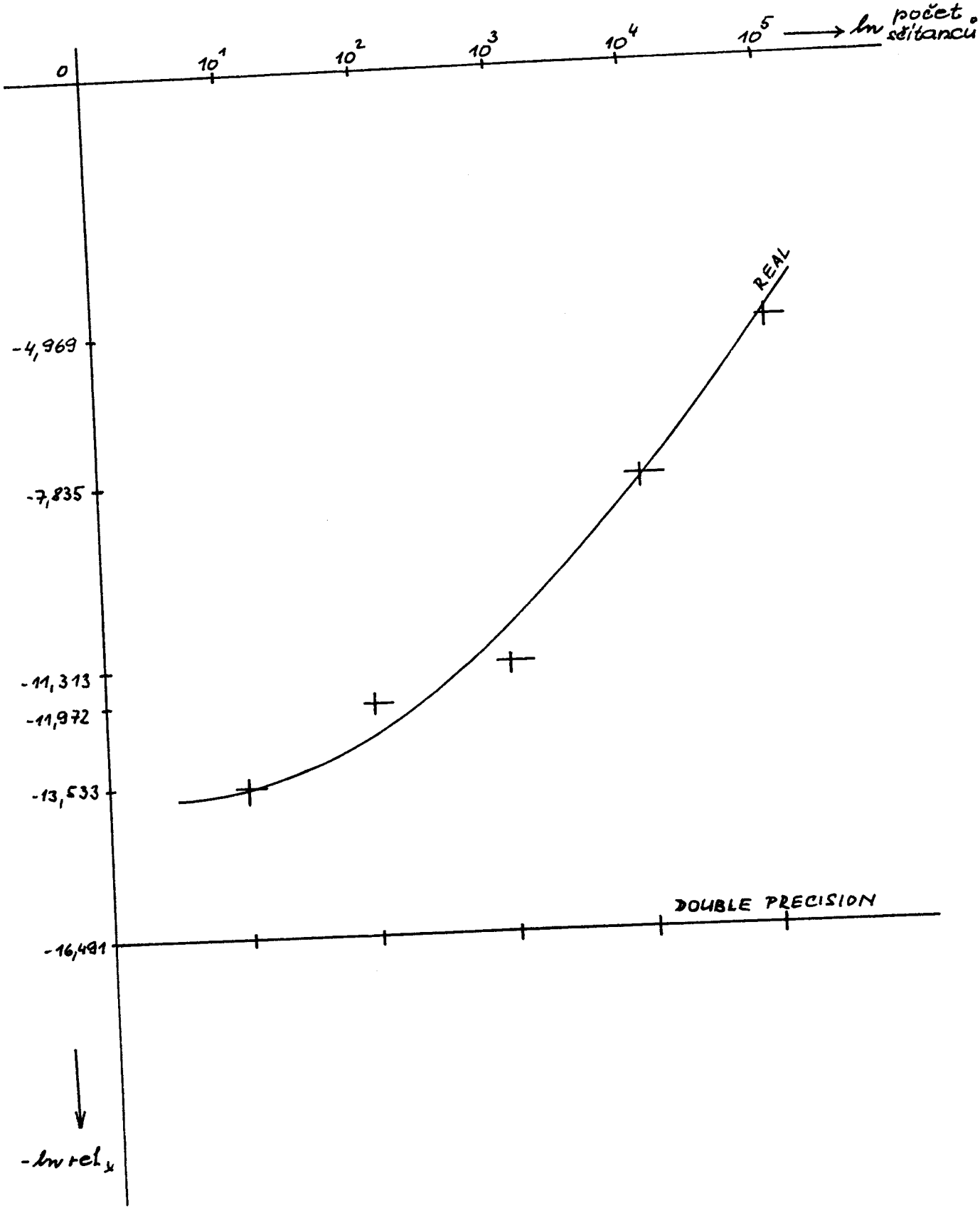
/5/ Anděl,J.: Matematická statistika.
Praha, SNTL, 1978

PŘÍLOHA č. 8

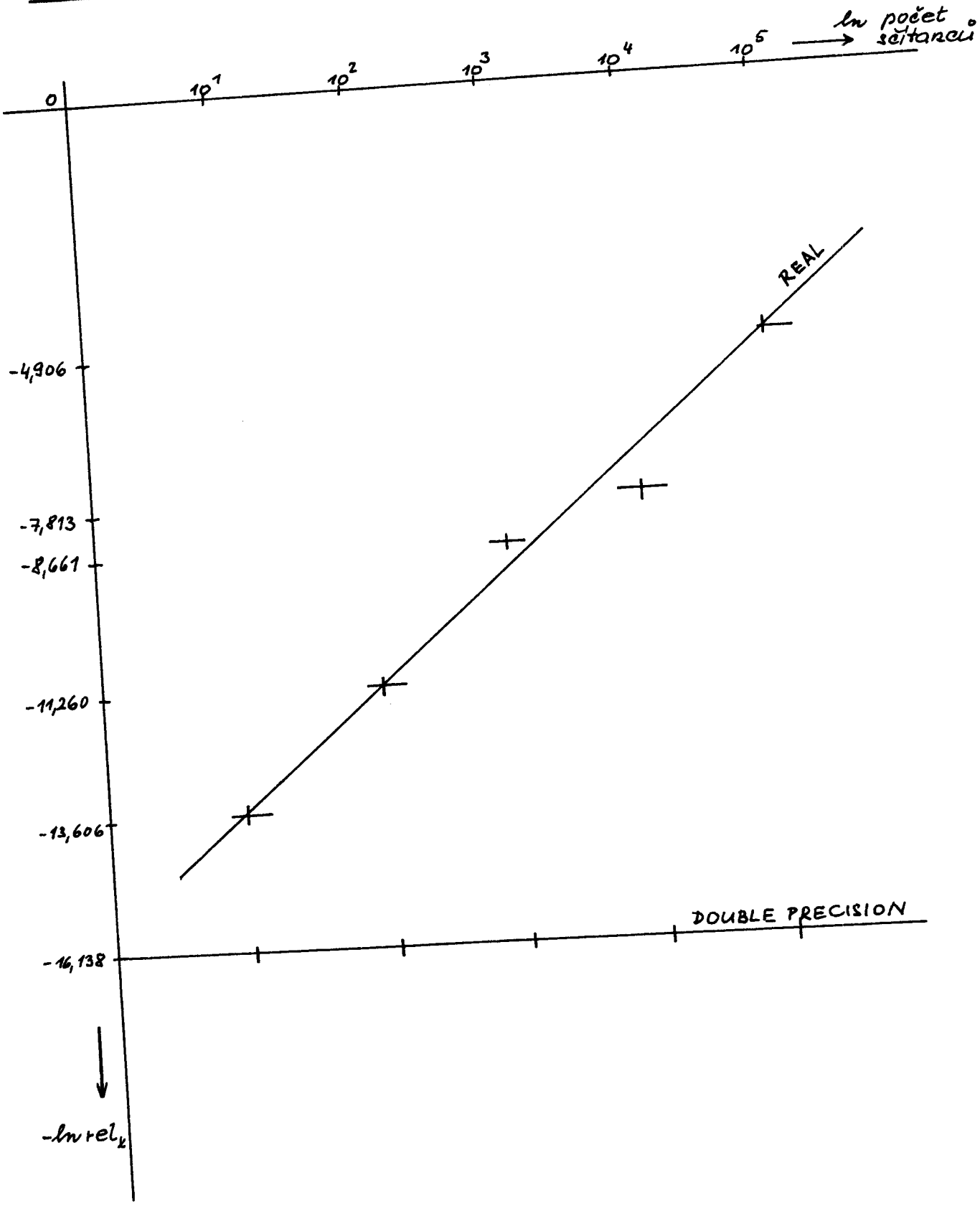
Sčítání čísla 1.10^{-2}



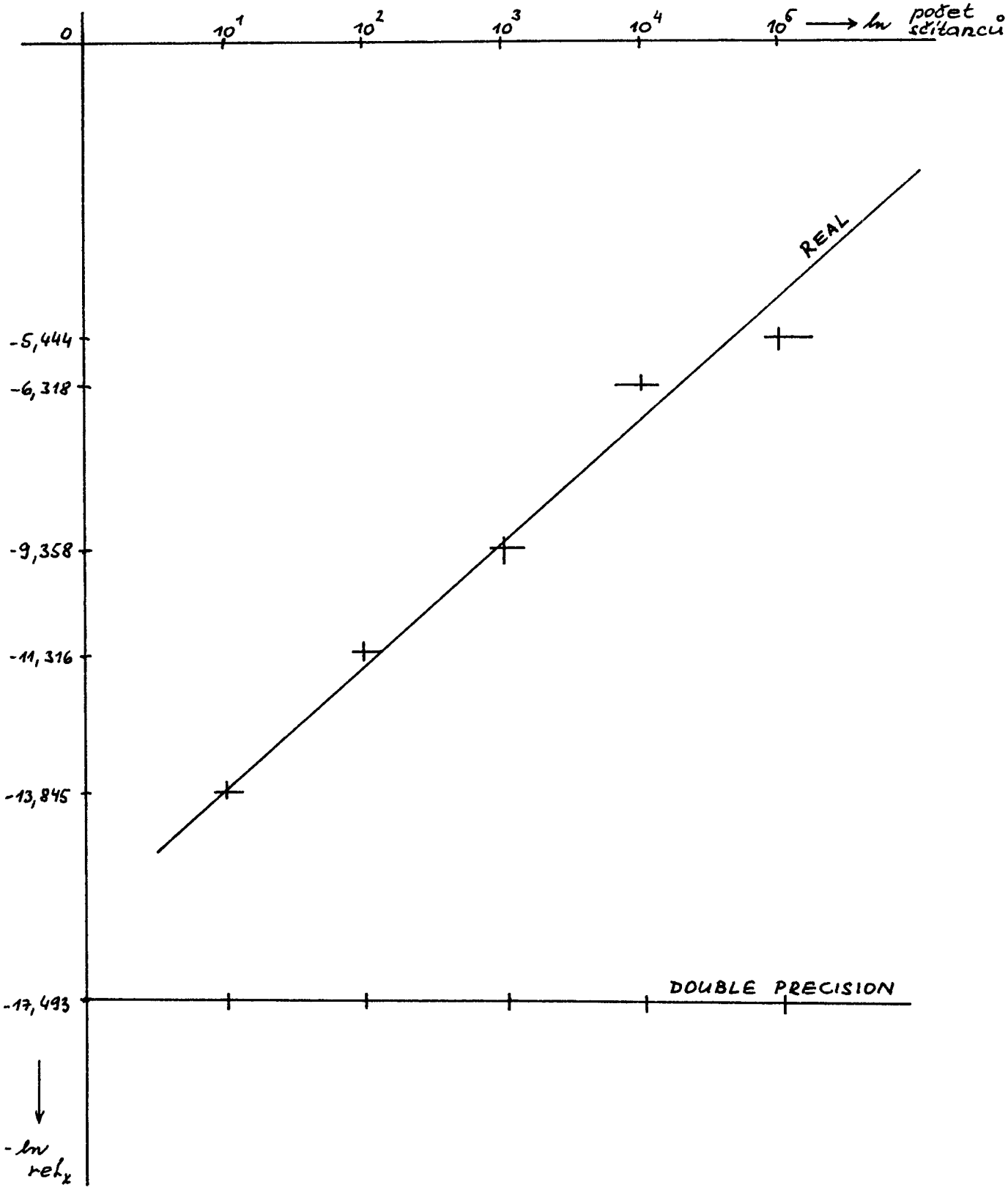
Sčítání čísla 1.10^{-3}



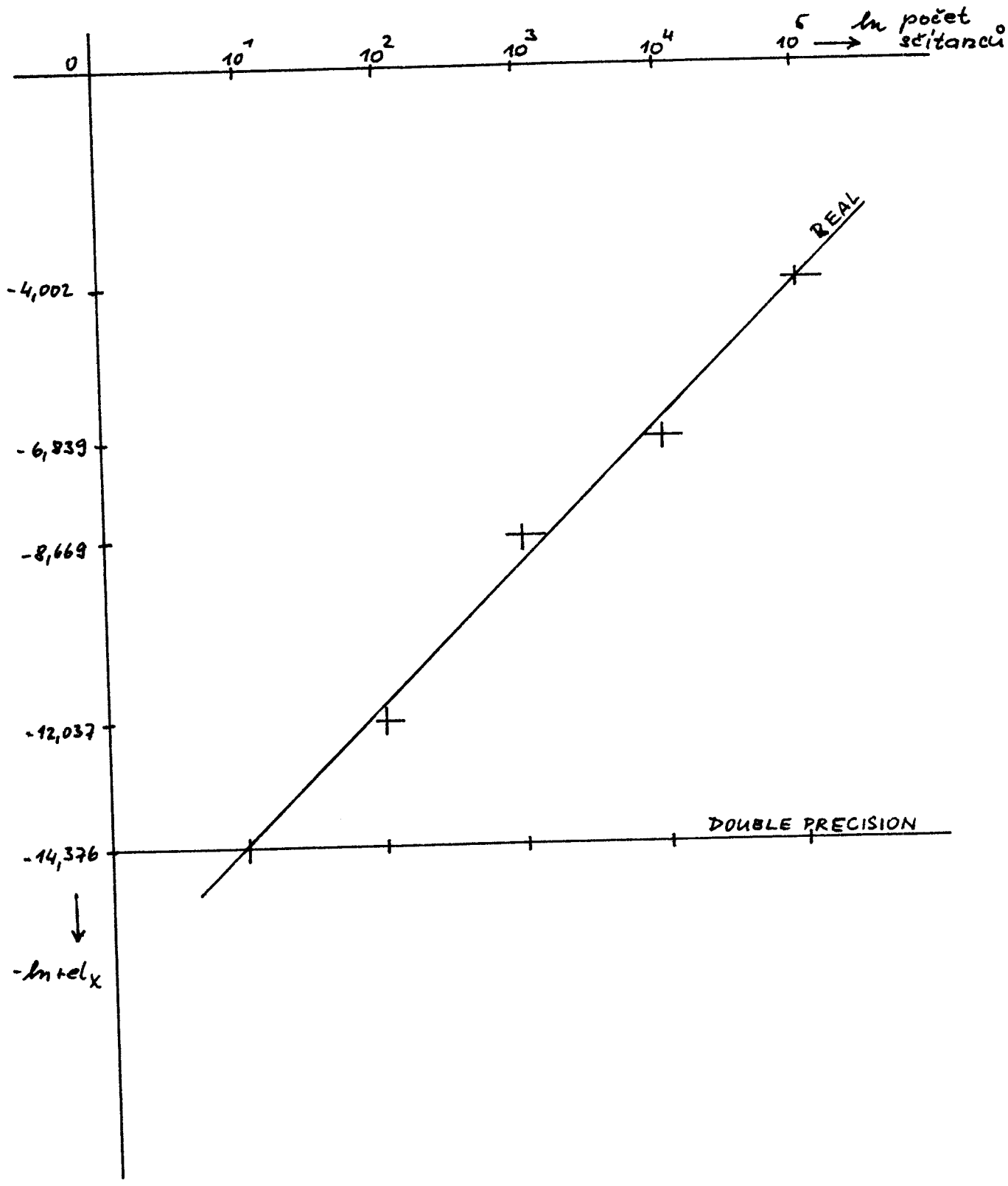
Sčítání čísla 1.10^{-4}



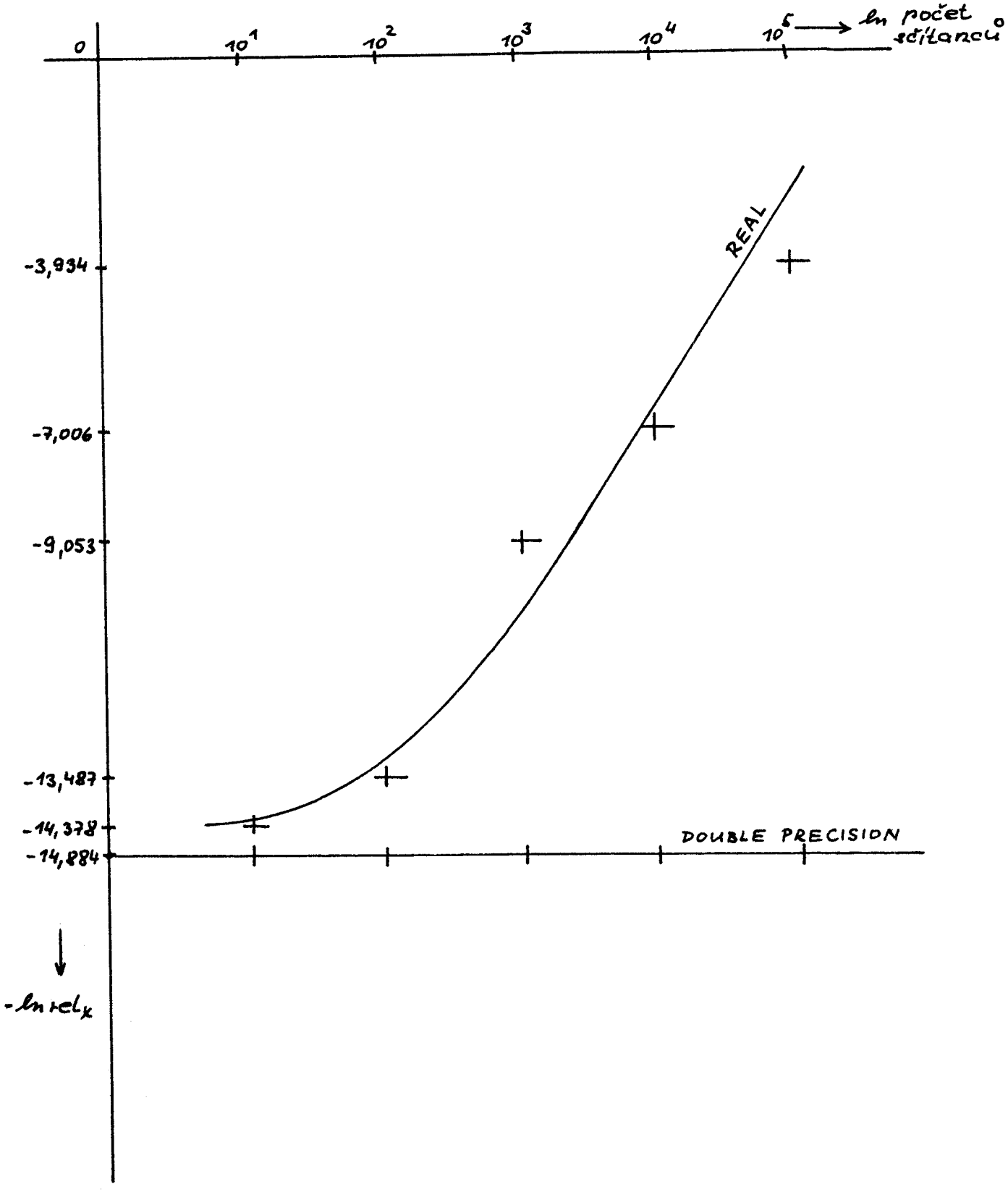
Sčítání čísla 1.10^{-5}



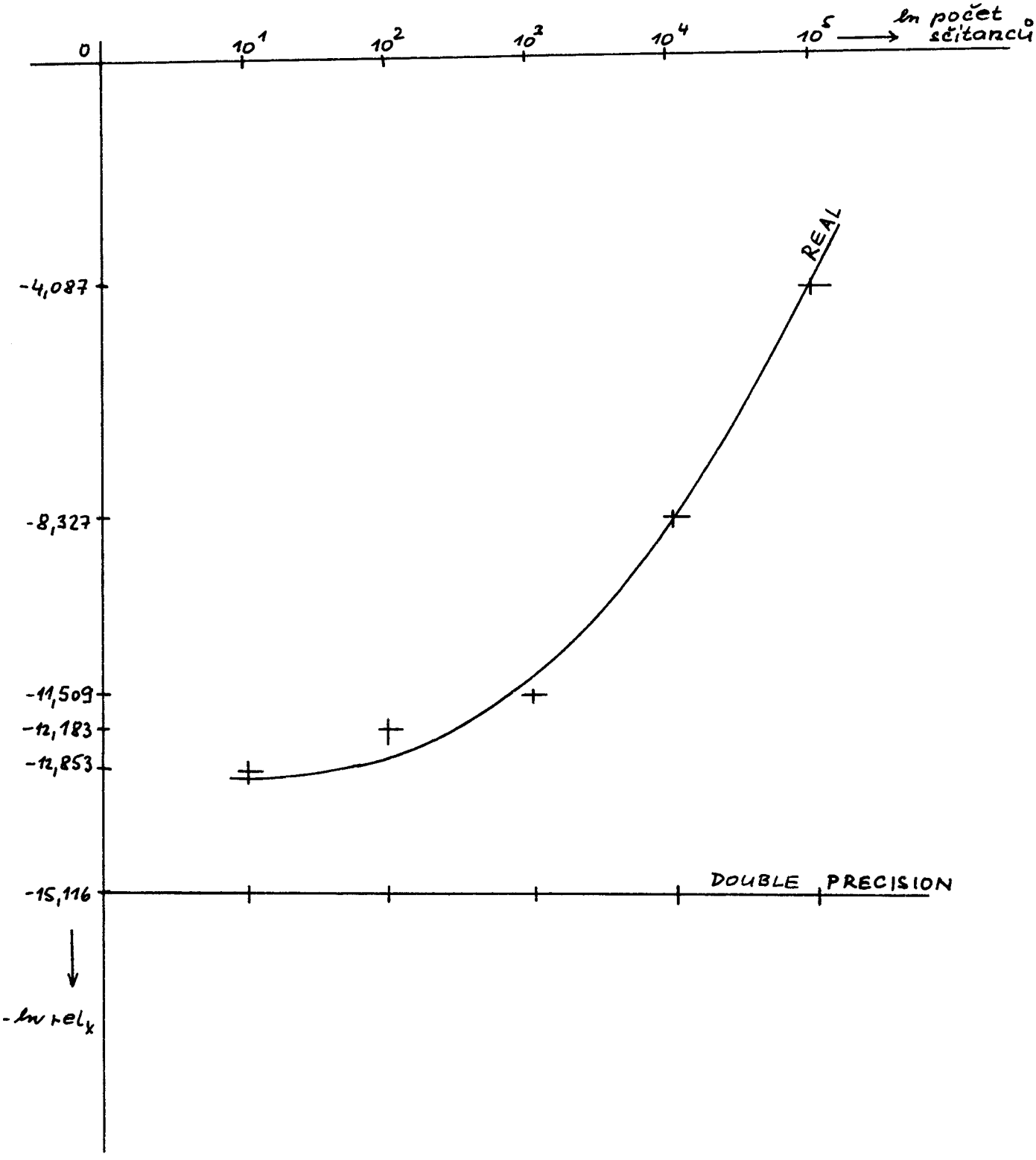
Sčítání čísla 1.10^{-6}



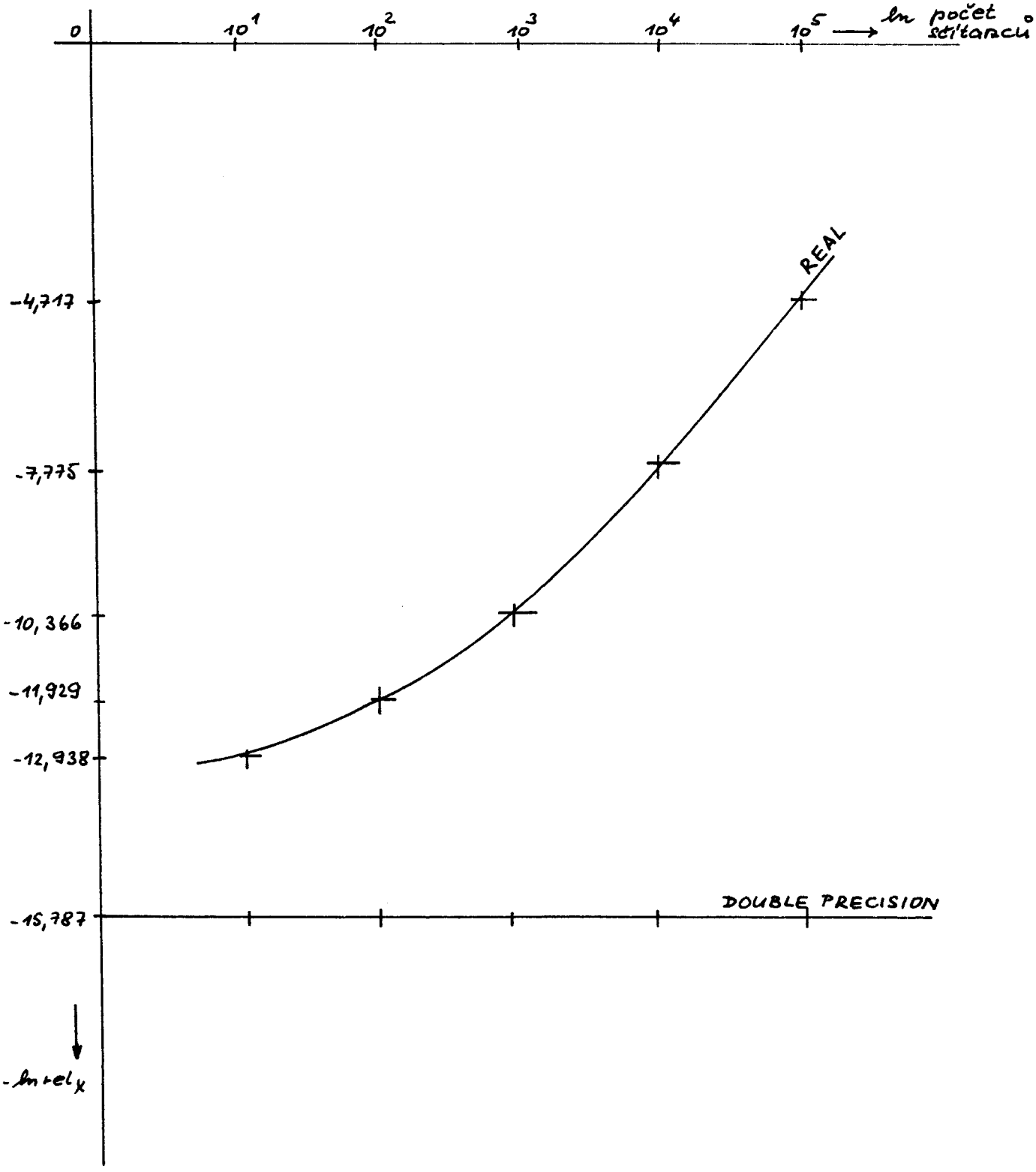
Sčítání čísla 1.10^{-7}



Sčítání čísla 1.10^{-8}



Sčítání čísla 1.10^{-9}



Sčítání čísla 1.10^{-10}