2008－HPC－118 (19)
2008／12／17

SR11000モデルK1における多倍長ルーチンによる数値積分の高速化

濱口信行

nobuyuki.hamaguchi.sa@hitachi.com

(株)日立製作所 ソフトウエア事業部

概要

赤外発散を伴うループ積分は,Xeon(3.06GHz) 1CPUで約1週間要していたが,数値積分法の改善,多倍長ルーチンの使用,並列化により,SR11000 モデルK1 64CPUで382秒と非常に実行時間を短縮する事が出来た.

Fast numerical integration with multi-precision library on SR11000/K1

Nobuyuki Hamaguchi

Software Division, Hitachi, Ltd.

Abstract

The loop integrals with the infrared divergence required about one week of elapsed time on Xeon(3.06GHz) one CPU. In order to reduce the elapsed time, We adopt improved numerical integration,Multi-precision library, and parallel computing technique.As a result of evaluation,these techniques and the library are effective toward the high performance calculation of the loop integrals with the infrared divergence.The elapsed time is 382 sec on SR11000/K1 with 64 CPUs.

1. はじめに

赤外発散を伴うループ積分の内,内部特異点をもつ三次元積分をGAUSS型積分法[1]により計算すると,Intel® Xeon ™ 3.06GHz 1CPUで631154秒≒7.2日と非常に時間を要していた.

この積分は極限値を求めるため,補外法を使用し,桁落ちが非常に激しく,各次元で誤差評価をして精度を保持する為に,積分回数が増え,並列化も困難になり,多大な時間を要していた.そこで,積分回数削減と並列化のため,二重指数関数型積分[2]を使用し,桁落ちの影響を少なくするため,Bailey のアルゴリズム[3]による多倍長演算を使用した.

手順としては,一次元積分から多次元積分,内部特異点を持たない積分から内部特異点を持つ積分の順で精度検証と,並列化効果を調べて,目的の内部特異点をもつ三次元積分の高速化を図る様にした.使用した計算機環境を以下に示す.

(1) Intel® Xeon ™ 3.06GHz

OS:	Red Hat Enterprise Linux ES, release 3 (taroon)
Kernel:	2.4.21-4 Elsmp on an i686
コンパイラ:	Intel Fortran Compiler for Linux V8.1 (注-1)

(2) SR11000モデルK1(CPU:POWER5+ 2.1GHz)
OS: AIX 5L V5.2 (注-2)
コンパイラ: 最適化FORTRAN90 01-05

(注-1) IntelおよびIntel Xeon は,アメリカ合衆国,およびその他の国におけるインテルコーポレーションまたはその子会社の商標または登録商標です.
Red Hatは米国およびその他の国でRed Hat,Inc.の登録商標若しくは商標です.
Linuxは,Linus Torvaldsの米国およびその他の国における登録商標若しくは商標です.
(注-2) AIXは,米国における米国International Business Machines Corp.の登録商標です.

2. 一次元積分

一次元の場合は,積分区間を特異点で分割する.一般に二重指数関数型積分法はGAUSS型積分法に比べて端点特異点を持つ積分に強いが,変換後の標本点が元の座標ではx=±1の近傍に集中するため,内部特異点を持つ積分では,特異点が端点になる様に積分区間を分割する.

また目的の積分は特異点が0近傍になる為,変数変換区間が[0,1]になる方法も併用している.これは変数変換区間を[-1,1]にすると,標本点を求める際,桁落ちにより打ち切り誤差が大きくなる為である.[4]

(例-1)

$$\int_0^1 \frac{1}{\sqrt{x}}dx = 2$$

誤差3.366×10^{-8} 変数変換区間[-1,1]

誤差8.881×10^{-16} 変数変換区間[0,1]

主値を分割点にとった計算例を以下に示す.[5]
ともに主値は1/e,4倍精度,分点数496,変数変換区間[-1,1]で計算.

(例-2)

$$\int_{-1}^1 \frac{x}{(x-1/e)}dx = 3.141592653589793$$

計算結果=3.141592653589793

(例-3)

$$\int_0^1 \frac{1}{\ln|\ln x|}dx = -0.154479641320$$

計算結果= -0.154479641320
(注-3):約200年間この値は0と考えられていた.[6]

3. 二次元積分[7]

$I=\int_0^1\int_0^{1-x}\frac{1}{-xys+(x+y)^2m^2+(1-x-y)\lambda^2}dydx$ では, $s>0, a=p^2s,$

$c=p^2m^2+(1-p)\lambda^2$として, $a=4c$となるpをα, $a \geqq 4c, -aq(1-q)+c=0$となる,

qの値を$q1 \leqq q2$とする.

積分計算では, $\beta=q1$として, $[0,\alpha]\times[0,0.5], [\alpha,1]\times[0,\beta], [\alpha,1]\times[\beta,0.5]$の

3つの領域に分割する.

問題の条件は $s=500^2, m=0.0005, \lambda$可変

Xeonで精度検証した結果を表1に示す.演算精度は, λ=1.0q-9を倍精度とし,それ以外は4倍精度としている.以下の表の結果より,十分な精度が得られている.

表 1: 積分結果

λ	実数部解析値	実数部実測値	虚数部解析値	虚数部実測値
10^{-9}	-0.44013021D-02	-0.44013029D-02	0.67702261D-03	0.67702260D-03
10^{-12}	-0.59282487D-02	-0.59282487D-02	0.85063344D-03	0.85063344D-03
10^{-15}	-0.74551954D-02	-0.74551954D-02	0.10242442D-02	0.10242442D-02
10^{-20}	-0.10000106D-01	-0.10000106D-01	0.13135956D-02	0.13135956D-02
10^{-25}	-0.12545017D-01	-0.12545017D-01	0.16029470D-02	0.16029483D-02
10^{-30}	-0.15089928D-01	-0.15089928D-01	0.18922983D-02	0.18922983D-02

4. 積分計算での精度向上

三次元積分で特異点の無い場合の精度を検証した.

$$I=\int_0^1\int_0^{1-x}\int_0^{1-x-y}\frac{1}{D^2}dzdydx$$

$$D=-xys-tz(1-x-y-z)+(x+y)\lambda^2+(1-x-y-z)(1-x-y){m_e}^2+z(1-x-y){m_f}^2$$

問題の条件は $s=-500^2, m_e=0.0005, m_f=150, t=-150^2, \lambda$可変

二重指数関数型積分は分点数512,きざみ幅$0.5^6\times592/512$,変数変換区間[−1,1]

表 2: 積分結果

λ	解析解	4 倍精度	6 倍精度	8 倍精度
10^{-15}	1.92786112E-007	1.86536490E-007	1.92786095E-007	1.92786095E-007
10^{-20}	2.47248635E-007	——-	2.47248396E-007	2.47249044E-007
10^{-25}	3.01711158E-007	——-	2.82773474E-007	3.01710746E-007
10^{-30}	3.56173681E-007	——-	——-	3.53101322E-007

SR11000/K1 1CPUで,実行時間は4倍精度32秒,6倍精度192秒,8倍精度374秒となっている.このことは,4倍精度で分点数を各軸倍(3次元なので演算量は8倍)にするよりは,6倍精度,8倍精度を使用する方が効果的な事を示している.

5. 並列化の性能向上

5.1 二次元積分特異点有りの場合

3章で精度検証した二次元積分特異点有りの場合のSR11000での実行時間を表3に示す.ノード内MPIによる並列化で,16CPUで14.95倍の台数効果である.

問題の条件は $s = 500^2, m = 0.0005, \lambda = 10^{-15}$

二重指数関数型積分は分点数992,きざみ幅$0.5^7 \times 992/1280$,

変数変換区間[0,1],[−1,1]の併用.

表 3: SR11000 実行時間と台数効果

CPU 数	実行時間 (秒)	台数効果
1	25.41272	1.00
2	12.96848	1.96
4	6.528708	3.89
5	5.262908	4.83
8	3.308608	7.68
10	2.660363	9.55
16	1.699701	14.95

5.2 三次元積分特異点無しの場合

4章で精度検証した三次元積分特異点無しの場合のSR11000での実行時間を表4に示す.

6倍精度,8倍精度ともに64CPUで1CPUの63.0倍,62.2倍と並列効果がよくでている.

問題の条件は $s = 500^2, m_e = 0.0005, m_f = 150, t = -150^2, \lambda = 10^{-15}$

表 4: SR11000 実行時間一覧表

CPU 数	6 倍精度実行時間 (秒)	8 倍精度実行時間 (秒)
1	572.82	1233.64
2	287.918	625.752
4	143.915	313.06
8	72.034	155.575
16	36.089	76.139
32	18.392	38.593
64	9.082	19.824

6.三次元積分[7]

$m = m_e^2$, $\mu = m_f^2 = -t$ とし, $A(z) = -\mu z^2 + (2\mu - m)z + m$ とおき, $a = x^2 s, c = (1-x)^2 A(z) + x\lambda^2$ とする. $\alpha = \frac{2}{s-4m}(\sqrt{(2m-\lambda^2)^2 + m(s-4m)} - (2m-\lambda^2))$. x 軸は $[0,\alpha], [\alpha,1]$ の 2 つの領域に分ける.

x が後者の領域にある場合, $a = x^2 s, c = (1-x)^2 A(z) + x\lambda^2$ とし, $a < 4c$ なら $D \neq 0$ で y 軸で分割せず, $a \geqq 4c$ なら, y 軸では被積分関数が 0.5 で対称より, $[0, \frac{1-\sqrt{1-\frac{4c}{a}}}{2}], [\frac{1-\sqrt{1-\frac{4c}{a}}}{2}, 0.5]$ と 2 分割する.以上から, 積分領域の分割数は 4 になる.並列実行時は, 最外側並列を行う.

Xeon 1CPUで約7.2日を要した下記問題を, SR11000で実測した結果を示す.
問題の条件は $s = 500^2, m_e = 0.0005, m_f = 150, t = -150^2, \lambda = 10^{-30}$

6.1 4倍精度

二重指数関数型積分は分点数1280,きざみ幅 $0.5^7 \times 1188/1280$, 変数変換区間[0,1]
計算結果の精度は以下のとおりである。結果はSR11000でXeonより良くなっている.

解析解 =-0.356173681291824D-06
Xeonの結果 =-0.3562207190019720D-06
SR11000の結果 =-0.3561984205196539D-06

SR11000の実行時間を表5に示す.64CPUでの実行時間は681秒であり,Xeon 1CPUの927倍の時間短縮となっている. 8CPUを基準とした64CPUの台数効果は7.88倍である,

表 5: 実行時間一覧表

CPU 数	実行時間 (秒)	台数効果 (対 8CPU)
8	5363	1
16	2687	2.00
32	1349	3.98
64	681	7.88

6.2 8倍精度

二重指数関数型積分は分点数768,きざみ幅 $0.5^6 \times 594/768$,, 変数変換区間[0,1]
計算結果の精度は以下のとおりである。結果はSR11000でXeonより良くなっている.

解析解 =-0.356173681291824D-06
Xeonの結果 =-0.3562207190019720D-06
SR11000の結果 =-0.3561964069834390D-06

SR11000の実行時間を表6に示す.64CPUでの実行時間は382秒であり,Xeon 1CPUの1652倍の時間短縮となっている. 8CPUを基準とした64CPUの台数効果は7.51倍である,

表 6: 実行時間一覧表

CPU 数	実行時間 (秒)	台数効果 (対 8CPU)
8	2867	1
16	1442	1.99
32	728	3.94
64	382	7.51

7. まとめ

昨今まで多くの計算時間を必要としていた赤外発散を伴うループ積分のうち,簡単なケースで三次元までは非常に高速化できる事が分かった.今後はより多次元でかつ複雑なケースの場合の高速化を検討して行く.

8. 参考文献

[1] QUADPACK A Subroutine Package for Automatic Integration, R. Piessens, E. de, Doncker-Kapenga, C. W. Uberhuber, D. K. Kahaner

[2] 杉原正顯,室田一雄,数値計算法の数理,岩波書店,2003

[3] Yozo Hida, Xiaoye S. Li, and Davit H. bailey, Algorithms for quad-double precision floating-point arithmetic. In Neil Burgess and Luigi Ciminiera, editors, 15th IEEE symposium on Computer Arithmetic, pages 155-162, Vail, Colorado, June 2001.

[4] 濱口信行,理論物理学計算への多倍長ライブラリの適用,情報処理学会研究報告IPSJ SIG Technical Report,2007-HPC-113(4),2007/12/7.

[5] 岩波数学公式集I ,2005年9月26日第25刷

[6] 200年続いた孫引きの誤まり発見顛末記---斉藤基彦:数学セミナー 1988年02月号

[7] J. Fujimoto, M.Igarashi, N. Nakazawa, Y. Shimizu and K. Tobimatsu Progress of Theoretical Physics, Supplement No.100(1990) 1-379

9. 謝辞

精度検証にあたり御指導いただきました高エネルギー加速器研究機構の石川正准教授および湯浅富久子准教授に感謝致します.