Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics
San Antonio, TX, USA - October 2009

A system on the Web for octuple-precision computation

Takemitsu Hasegawa, Yohsuke Hosoda, Yuuichi Itaya
Department of Information Science
University of Fukui
Fukui, 910-8507, Japan

Ichizo Ninomiya
Emeritus Professor
Nagoya University
Nagoya, Japan

***Abstract*—Online octo-system, an Web system to provide remote users with the facility of octuple-precision computation, i.e. 256-bit floating-point (15-bit exponent and 240-bit mantissa) operations, is described. By simply supplying the required data on the Web page any users with no knowledge of programming can obtain results computed in about 72 significant digits. Also available in the system are more than 70 mathematical functions, say, fundamental functions such as sine and cosine functions as well as special functions like the Bessel function.**

***Index Terms*—Octuple-precision computation, online system, user-friendly computation, C++**

I. INTRODUCTION

Higher (multiple) precision computation than the IEEE 754 standard [6] float- and double-precisions, which are commonly available in most computer hardwares, has been recently required in many problems in science and engineering, say, in high energy physics [3]. Some systems [1], [2], [4], [5], [7], [8], [9] for multiple-precision computation including quadruple- and octuple-precisions have been developed. To perform the multiple-precision computation, however, users are required to have some knowledges in addition to the one on programming language, e.g. Fortran, C, or C++ because the multiple-precision system have not yet been standardized.

In this paper we present 'online octo-system' [10], an online user-friendly system for octuple-precision computation. Remote users can perform the octuple precision computation with the online octo-system by simply inputting the required data on the Web page to get desired results computed in about 72 significant digits without any knowledges on programming and the multiple-precision system. Furthermore, expert users can supply their own program on C language with double-precision into the online octo-system to obtain octuple-precision results.

The online octo-system consists of Web pages that receive data and/or the C program(s) supplied by remote users and give computed results and manual pages describing the use of the octo-system and mathematical functions available as well as the octo-system itself, which is the C++ program for performing octuple-precision computation as shown in the following section.

II. OCTUPLE-PRECISION SYSTEM IN C++

In this section we present the octo-system in the Borland C++ written by Ichizo Ninomiya. The octo-system provides two data types of floating-point number, i.e. octo-real and octo-complex.

A. Octuple-precision real number

Octuple-precision floating-point real number R is expressed by a class constructor consisting of eight 4-byte unsigned integers defined by

```
#typedef unsigned ul8[8]
```

the first bit for the sign S followed by 15 bits for the exponent E in the first element ul8[0] followed by the remaining 16 bits of ul8[0] combined with the remaining 7 elements ul8[1],. . .,ul8[7] for the mantissa M of the total 240 bits, namely

$$R = (-1)^S(1 + M) \times 2^{(E-16383)}.$$

These floating-point number can express real numbers in more than 72 significant digits (by the mantissa of 241 bits including the hidden bit).

B. Declaration and conversion of data type

The octo-system provides the facility of declaring the data type. By declaring

```
octo a,b,c[5];
```

we get octo-real variables a and b and an array $c[5]$. Further the conversion between different data types, i.e. int, float, double and octo is possible. For example, let x be a octo-real variable. Then to convert x into int, float and double types we write

```
int(x); float(x); double(x);
```

respectively. Conversely, declaring

```
oct(i); oct(f); oct(d);
```

converts `int i; float f; double d` into the octo-real type, respectively.

C. The four basic arithmetic operations and mixed mode operation

Octuple-precision addition, subtraction, multiplication and division of two oct-numbers are performed just like the double-precision arithmetic operations. The arithmetic operation of two numbers belonging to different levels of precision is also possible.

To compute the power of a variable, say, x^n or a^b, we write

```
octo opow(octo& x,int n);
octo opow(octo& a,octo& b);
```

respectively.

D. Output routine

The output (and no input) routine for octo-real variables is available as follows. The following command

```
void print(int nc,int nb,int np,int n5,
           octo& a);
```

feeds nc lines before printing octo-real a in nb blanks followed by np significant digits with consecutive 5 digits together followed by $n5$ blank and finally followed by the exponent, where $n5 = 0$ or 1. For example, let $a = 10*pi$, $nc = nb = 0$, $np = 72$ and $n5 = 1$, then the output is

$$3.14159\ 26535\ 89793\ 23846\ 26433\ 83279\ 50288$$
$$41971\ 69399\ 37510\ 58209\ 74944\ 59230\ 78164e + 01$$

where we write the above output result in two lines because of the limitation of this manuscript's width. A routine to output the computed results into a given file is also available.

E. Constants and mathematical functions

Twelve constants used often such as π and $\log_2 e$ and table functions like zeta function $\zeta(n)$ and Bernoulli number B_n are available. These constants are stored in the header file "octo.h" in the octo-system in the form like

$$\text{static ul8 pi} =$$
$$\{0x4000921f, 0xb54442d1, 0x8469898c, 0xc51701b8,$$
$$0x39a25204, 0x9c1114cf, 0x98e80417, 0x7d4c7628\}$$

Also available are 57 real octuple-precision functions including standard mathematical functions such as trigonometric functions as well as special functions.

F. Octuple-precision complex number

The octo-system provides 'ocmplx' for octuple-precision floating-point complex numbers consisting of two octo-real numbers for real and imaginary parts and for their arithmetic operations.

1) Declaration of data type and arithmetic operations: To declare w, z and $c[5]$ as octuple-precision complex we express

```
ocmplx w,z,c[5];
```

The four types of arithmetic operation between two octo-complex numbers are performed like those of octo-real numbers. Mixed mode operations between octo-complex numbers, integers and double-precision numbers are possible. To compute the power of an ocmplx variable z we write as folows

```
ocmplx copow(ocmplx& z, int n);
```

2) Constants and mathematical functions: Useful octuple-precision complex constants are provided. Twenty four octuple-precision complex mathematical functions are available, say $coabs$, $coreal$, $coimag$, $coarg$, $ocmplx(octo\&\ x, octo\&\ y)$, $cosqrt$, $colog$, $cosin$, $cocos$, $cosinh$, $cocosh$ as well as parallel functions such as

```
void copscn(ocmplx& z,octo* csn,octo* ccn)
```

which computes `cosin(z)` and `cocos(z)` simultaneously, namely $sin(z)$ and $cos(z)$ of octo-complex z.

III. Online octo-system

The online octo-system is a user-friendly system on the Web for providing remote users with the facility of octuple-precision computation without any knowledges of programming and multiple-precision computations. The online octo-system consists of three Web pages, the CGI program and the octo-system in C++ described in § II.

The Web pages are those for

- the interface for users to input data and to upload programs,
- a manual describing the usage of the system,
- a list of available functions.

The CGI program receives data on the Web pages to transact them on the computer before asking the octo-system to perform the computation.

The original octo-system in Borland C++ on the Windows XP due to Ninomiya is converted to the GNU C++ (Net BSD/64-bit computer/gcc ver. 3.3.3) version for the use in the Linux environment.

A. User-interface for inputting data or mathematical expression

In the user-interface page on the Web one can perform octuple-precision computations by two types of way to upload data.

In th first type one choose a mathematical function in the pull-down menu followed by inputting value(s) of argument(s). Fig 1 depicts the input form for the computation of $osin(\pi/4)$ while Fig 2 shows the function chosen, the input value of the argument and the computed result.

In the second type one can write a simple mathematical expression involving the arithmetic operations and the mathematical functions in a given form of the page to get a computed result. For example, inputting

Fig. 1. Web page to choose a function with an argument

Fig. 2. Web page to show the computed result with input data

`(osin(qp)+ocos(qp))/otan(pi/3)`

gives a computed result as follows

$$8.16496\ 58092\ 77260\ 32732\ 42802\ 49019\ 63797\ 32198\ 24935\ 52223\ 37614\ 42308\ 55750\ 32012e-01$$

where qp means $\pi/4$.

Three data types of numbers are allowed to input on the Web page. One can input integers, fixed-point numbers and floating-point numbers in less than 76 digits sequence for integers, including the period for fixed-point numbers, and followed by the exponent for floating-point numbers, respectively.

B. Interface for uploading C program

An expert user can upload their own C program with double-precision computation in the interface page to implement the program in the octo-system in C++. Fig 3 depicts the Web page for uploading the user's C file. In this case the online octo-system converts the uploaded C file to the C++ program complied with the octo-system presented in § II before implementing the program to perform the octuple-precision computation, see § III-D2 below for details.

C. Manual and list of functions

The online octo-system provides a simple manual page in Japanese illustrating how to use the system, data types, constants and functions. A list of functions available in the octo-system is also provided. Fig 4 depicts a part of the function list.

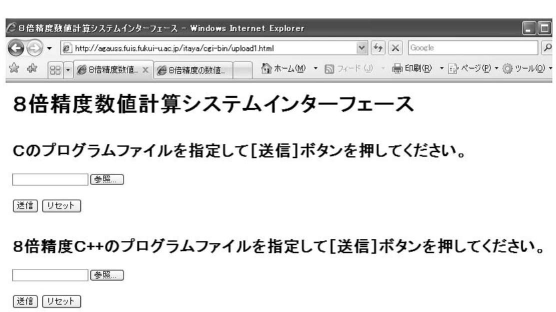


Fig. 3. Web page to upload a program file in C or C++

D. Computational procedure on the online system

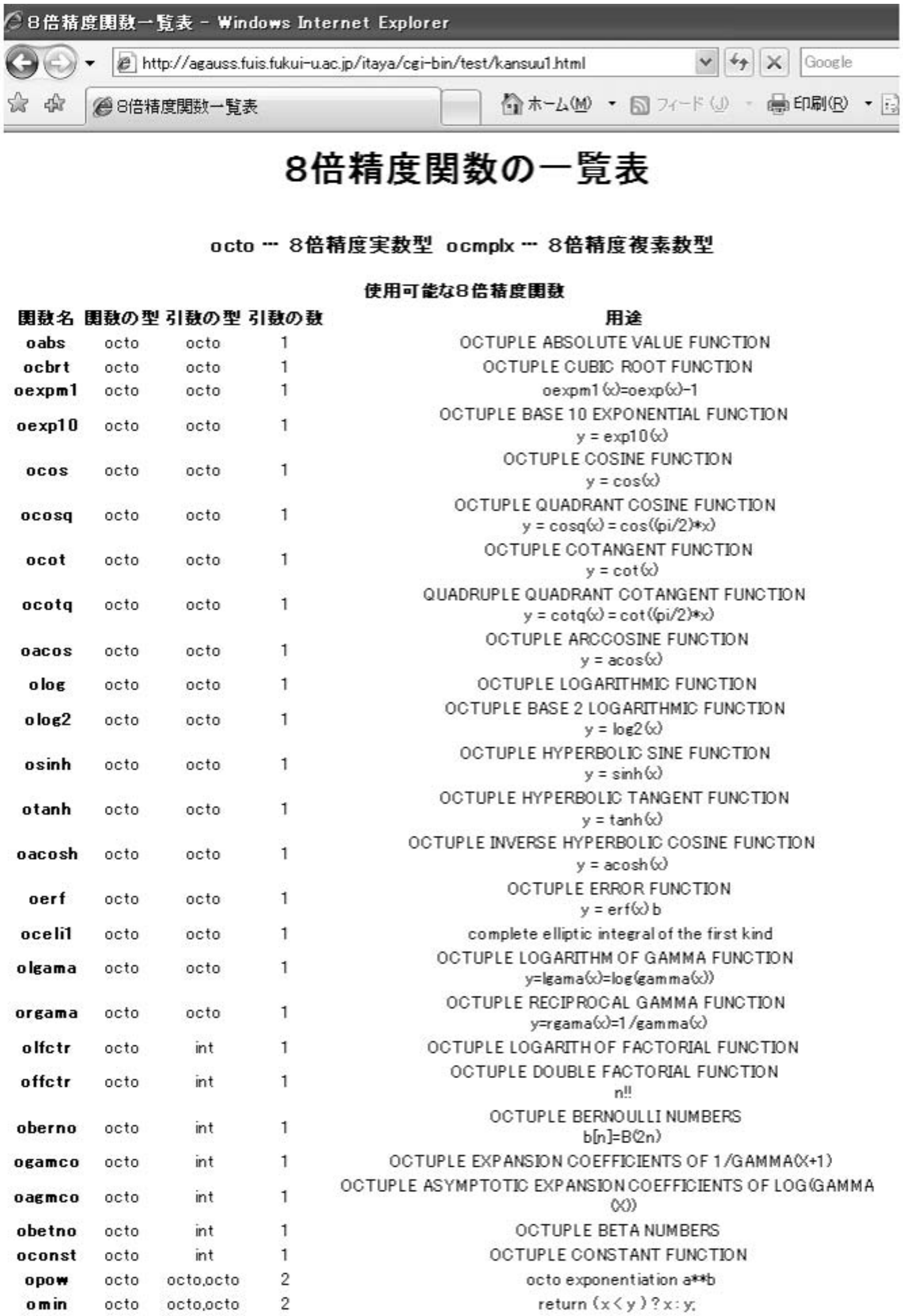
1) procedure from user's data to computed results: Here we show the procedure on the treatment of input data from users to give computed results. When one chooses a function and inputs value(s) in the argument(s) on the interface page the system conveys the information on the selected function and input data to the CGI program, which then checks the accepted data and examines if some errors exist or not. If no error is detected, then the data are transfered to the octo-system in C++ to perform the octuple-precision computation. The computed results are given back to the CGI program to display them on the Web page.

2) Converter routine for C program: A user can implement their own C or C++ program in the octo-system by uploading the program in the input form on the interface page to perform the octuple-precision computation. If he/she uploads a C program file, then the online octo-system asks the CGI program to store the file in the computer memory before the converter routine (translator) converts the file into the C++ program complied with the octo-system.

The converter translates the C program in the double-precision version into the octuple-precision version. To this end the translator (converter) routine starts by checking if some programming errors in the original C program exist. Further prohibited system functions are checked to protect the online octo-system against the attacks of malicious users. If neither programming error nor prohibited function exists, then the translation starts. Every double-precision constant (and variable) in the original C program is replaced by the corresponding octuple-precision constant (and variable) after the addition of the header lines required in the octo-system. Further every double-precision function is replaced by the octuple-precision function. For example,

$$\text{sqrt(5)} \quad \Rightarrow \quad \text{osqrt(octo(5))}.$$

The output routine is also replaced by using the octuple-precision 'print()' described in § II-D. For example, for $octo(b)$



8倍精度関数の一覧表

octo … 8倍精度実数型 ocmplx … 8倍精度複素数型

使用可能な8倍精度関数

関数名	関数の型	引数の型	引数の数	用途
oabs	octo	octo	1	OCTUPLE ABSOLUTE VALUE FUNCTION
ocbrt	octo	octo	1	OCTUPLE CUBIC ROOT FUNCTION
oexpm1	octo	octo	1	oexpm1(x)=oexp(x)-1
oexp10	octo	octo	1	OCTUPLE BASE 10 EXPONENTIAL FUNCTION y = exp10(x)
ocos	octo	octo	1	OCTUPLE COSINE FUNCTION y = cos(x)
ocosq	octo	octo	1	OCTUPLE QUADRANT COSINE FUNCTION y = cosq(x) = cos((pi/2)*x)
ocot	octo	octo	1	OCTUPLE COTANGENT FUNCTION y = cot(x)
ocotq	octo	octo	1	QUADRUPLE QUADRANT COTANGENT FUNCTION y = cotq(x) = cot((pi/2)*x)
oacos	octo	octo	1	OCTUPLE ARCCOSINE FUNCTION y = acos(x)
olog	octo	octo	1	OCTUPLE LOGARITHMIC FUNCTION
olog2	octo	octo	1	OCTUPLE BASE 2 LOGARITHMIC FUNCTION y = log2(x)
osinh	octo	octo	1	OCTUPLE HYPERBOLIC SINE FUNCTION y = sinh(x)
otanh	octo	octo	1	OCTUPLE HYPERBOLIC TANGENT FUNCTION y = tanh(x)
oacosh	octo	octo	1	OCTUPLE INVERSE HYPERBOLIC COSINE FUNCTION y = acosh(x)
oerf	octo	octo	1	OCTUPLE ERROR FUNCTION y = erf(x) b
oceli1	octo	octo	1	complete elliptic integral of the first kind
olgama	octo	octo	1	OCTUPLE LOGARITHM OF GAMMA FUNCTION y=lgama(x)=log(gamma(x))
orgama	octo	octo	1	OCTUPLE RECIPROCAL GAMMA FUNCTION y=rgama(x)=1/gamma(x)
olfctr	octo	int	1	OCTUPLE LOGARITH OF FACTORIAL FUNCTION
offctr	octo	int	1	OCTUPLE DOUBLE FACTORIAL FUNCTION n!!
oberno	octo	int	1	OCTUPLE BERNOULLI NUMBERS b[n]=B(2n)
ogamco	octo	int	1	OCTUPLE EXPANSION COEFFICIENTS OF 1/GAMMA(X+1)
oagmco	octo	int	1	OCTUPLE ASYMPTOTIC EXPANSION COEFFICIENTS OF LOG(GAMMA(X))
obetno	octo	int	1	OCTUPLE BETA NUMBERS
oconst	octo	int	1	OCTUPLE CONSTANT FUNCTION
opow	octo	octo,octo	2	octo exponentiation a**b
omin	octo	octo,octo	2	return (x<y)?x:y;

Fig. 4. Web page for the list of octuple-precision functions available

```
printf("a=%d,b=%f\n",a,b);
```

in the C program is replaced by

```
printf("a=%d",a); printf(",b=");
print(0,5,70,1,b); printf("<BR>");
```

where
 is used to display the result on the Web page. Finally the translated file is complied and executed before displaying the computed results on the Web page via the CGI program. If computing doesn't terminate in a given time interval, the system stops computing without any results.

IV. Conclusion

We have shown the online octo-system to provide any remote users of primitive to professional level with the facility of octuple-precision computations. It doesn't appear that some environments exist to provide higher-precision computation than double-precision without requiring any knowledges on programming language and numerical computation although some systems for multiple-precision computation with Fortran or C language are available.

The online octo-system presented here is constructed both for novice users and for expert users on programming. For novice users the system is so easy to use and similar to a personal pocket calculator in the usage. On the Web page choosing a desired function and supplying value(s) to the argument(s) of the function is enough to get the computed result(s) in about 72 significant digits. It is also easy to perform a simple calculation involving the four basic arithmetic operations and using mathematical functions.

On the other hand, higher level of computation involving programming in C is also possible for expert users. Supplying a C program with double-precision computation to the online octo-system gives the computed results in octuple-precision without any troubles to write a program complied with octuple-precision computation system.

References

[1] D. H. Bailey, *Algorithm 719 Multiprecision translation and execution of FORTRAN program*, ACM Trans. Math. Soft., 19 (1993) 288–319.

[2] D. H. Bailey *A Fortran 90-based multiprecision system*, ACM Trans. Math. Soft., 21 (1995) 379–387.

[3] J. Fujimoto, N. Hamaguchi, et al., *Numerical precision control and GRACE*, Nucl. Instr. Meth. Phys. Res. A, 559 (2006) 269–272.

[4] A. H. Karp, P. Markstein, *High-precision division and square root*, ACM Trans. Math. Soft., 23 (1997) 561–589.

[5] D. Mitchell, S. Noble, *Multiprecision floating-point arithmetic on Apple systems*, http://images.apple.com/acg/pdf/MP_Floating_Point_20070313.pdf.

[6] M. L. Overton, *Numerical Computing with IEEE Floating Point Arithmetic*, Philadelphia, PA: SIAM, 2001.

[7] D. M. Smith, *Algorithm 693 A FORTRAN package for floating-point multiple-precision arithmetic*, ACM Trans. Math. Soft., 17 (1991) 273–283.

[8] D. M. Smith, *A multiple-precision division algorithm* Math. Comp., 65 (1996) 157–163.

[9] D. M. Smith, *Algorithm 786 Multiple-precision complex arithmetic and functions*, ACM Trans. Math. Soft., 24 (1998) 358–367.

[10] http://netnumpac.fuis.fukui-u.ac.jp/˜tanabe/cgi-bin/test/index.html