1. Identification of the proposed change

1.1. Title

Library Functions − Complex Mathematics

1.2. MDC Proposer and Sponsor

This proposal originates from Ed de Moel.
Motions regarding the status of this document will be made by Taskgroup 5 (Mathematics) of Subcommittee 13 (Data Management and Manipulation).

Ed de Moel can be reached at:
· 3950 Mahaila Avenue, apartment K12, San Diego, California 92122
· home phone: 619 455 7107
· office phone: 619 535 7566
· telefax: 619 535 7627
· email: demoel@fwva.saic.com
· email: demoel.ed@forum.va.gov

1.3. Motion

None.

1.4. History of MDC actions

Date	Document	Action
February 1995	X11/95-14	Final write-up.
January 1995	X11/SC13/94-40	Typographical errors corrected. Presented for promotion to MDC Type A. Approved 25:0:2
June 1994	X11/SC13/94-26	No changes other than administrative. Presented for promotion to MDC Type A. Too many typographical problems encountered. Document not voted on.
February 1994	X11/SC13/94-8	MUMPS representation of some functions improved. Changed all references from $&MATH.function to $&function^MATH. Presented for promotion to SC#13 Type A. Editorial amendment: passed 17:0:5. Amended document accepted as SC#13 Type A 15:1:5
October 1993	X11/SC13/93-57	MUMPS representation of some functions improved. Presented to replace current SC#13 Type B. Passed 7:1:3.
June 1992	X11/SC13/92-6	Proposal elevated to SC#13 Type B. All names changed from $&LIB.function^COMPLEX to $&MATH.function.
February 1992	no number	Draft library document discussed in taskgroup
June 1991	X11/SC13/91-24	Collection of mathematical functions elevated to Type B of SC#8.

1.5. Dependencies

(Non)numeric values proposal may redefine representation of infinite and indefinite.

2. Justification of Proposed Change

2.1. Needs

Many people have expressed a concern that the language of MUMPS should be extended with at least the basic mathematical functions. Several years ago, a Library Subcommittee was formed, and within this subcommittee, a taskgroup was formed to produce proposals to add mathematical functions to the language. Initially, the plan of this taskgroup was to add only the most needed functions, after comparison of MUMPS with other programming environments, this plan was changed to offer a library that would be at least as complete as the libraries of Fortran and Ada.

2.2. Existing Practice in Area of the Proposed Change

Since no intrinsic functions are available for mathematical functions, Z-extensions, external call extensions, and MUMPS based approximations are currently in use.

3. Description of the proposed change

3.1. General Description of the Proposed Change

The proposed change adds a number of library functions to the MUMPS function library. Although this proposal uses the 'external library call format', the proposer forsees that some or all of these functions may become intrinsic functions in the future. This proposal, however, deals only with the functionality, not with the method of implementation.

3.2. Annotated Examples of Use

Function calls like

```
SET A=$%CABS^MATH(Z)
SET Z=$%CADD^MATH(A,B)
SET Z=$%CCOS^MATH(ZETA)
SET Z=$%CDIV^MATH(A,B)
SET A=$%CEXP^MATH(Z)
SET A=$%CLOG^MATH(Z)
SET Z=$%CMUL^MATH(A,B)
SET Z=$%COMPLEX^MATH(X)
SET A=$%CONJUG^MATH(Z)
SET A=$%POWER^MATH(Z,N)
SET A=$%CSIN^MATH(Z)
SET Z=$%CSUB^MATH(A,B)
```

would become available.

3.3. Formalization

In the MUMPS Library Specification, add the definition of the library functions below (at the time that this document is written, no document number or section numbering is known for the MUMPS Library Specification). As soon as reasonably possible after establishment of a MUMPS Library Specification, the document editor is directed to update this document to conform with that specification. This proposal does not intend to impinge upon the default package name space.

3.3.1. CABS Function

3.3.1.1. Library Element Description

CABS, complex absolute value function.

3.3.1.2. Definition

CABS^MATH:REAL(Z:COMPLEX)

The function returns the absolute value of the complex number Z.

3.3.1.3. Domain

Standard.

3.3.1.4. Range

Standard.

3.3.1.5. Side effects

None.

3.3.1.6. Example of MUMPS code to implement

```
CABS(Z) ;
    New ZRE,ZIM
    Set ZRE=+Z,ZIM=+$Piece(Z,"%",2)
    Quit $%SQRT^MATH(ZRE*ZRE+(ZIM*ZIM))
```

3.3.1.7. Note(s)

The code in the previous section is an example of a possible implementation of this library function. Vendors are encouraged to provide implementations that offer a better efficiency as well as a greater accuracy.

3.3.2. CADD Function

3.3.2.1. Library Element Description

CADD, complex addition function.

3.3.2.2. Definition

CADD^MATH:COMPLEX(X:COMPLEX,Y:COMPLEX)

The function returns the sum of X + Y, where X and Y are complex numbers.

3.3.2.3. Domain

Standard.

3.3.2.4. Range

Standard.

3.3.2.5. Side effects

None.

3.3.2.6. Example of MUMPS code to implement

```
CADD(X,Y) ;
    New XRE,XIM,YRE,YIM
    Set XRE=+X,XIM=+$Piece(X,"%",2)
    Set YRE=+Y,YIM=+$Piece(Y,"%",2)
    Quit XRE+YRE_"%"_(XIM+YIM)
```

3.3.2.7. Note(s)

The code in the previous section is an example of a possible implementation of this library function. Vendors are encouraged to provide implementations that offer a better efficiency as well as a greater accuracy.

3.3.3. CCOS Function

3.3.3.1. Library Element Description

CCOS, trigonometric complex cosine function.

3.3.3.2. Definition

`CCOS^MATH:COMPLEX(Z:COMPLEX,PREC:INTEGER:O)`

The function returns the value of the trigonometric cosine cos(Z) of the angle Z in radians. Z is interpreted as a complex number. The number of significant digits in the complex cosine is specified by the optional parameter PREC. If not specified, a default value of 11 digits is assumed for PREC.

3.3.3.3. Domain

Standard.

3.3.3.4. Range

–1 ≤ $%CCOS^MATH(Z) ≤ 1.

3.3.3.5. Side effects

None.

3.3.3.6. Example of MUMPS code to implement

```
CCOS(Z,PREC) ;
    New E1,E2,IA
    Set IA=$%CMUL^MATH(Z,"0%1")
    Set E1=$%CEXP^MATH(IA,PREC)
    Set IA=-IA_"%"_(-$Piece(IA,"%",2))
    Set E2=$%CEXP^MATH(IA,PREC)
    Set IA=$%CADD^MATH(E1,E2)
    Quit $%CMUL^MATH(IA,"0.5%0")
```

3.3.3.7. Note(s)

The code in the previous section is an example of a possible implementation of this library function. Vendors are encouraged to provide implementations that offer a better efficiency as well as a greater accuracy.

3.3.4. CDIV Function

3.3.4.1. Library Element Description

CDIV, complex division function.

3.3.4.2. Definition

`CDIV^MATH:COMPLEX(X:COMPLEX,Y:COMPLEX)`

The function returns the value X / Y, where X and Y are complex numbers.

3.3.4.3. Domain

Standard. If the complex numeric interpretation of Y is equal to "0%0", the usual error for division by zero will occur.

3.3.4.4. Range
Standard.

3.3.4.5. Side effects
None.

3.3.4.6. Example of MUMPS code to implement
```
CDIV(X,Y) ;
    New D,IM,RE,XIM,XRE,YIM,YRE
    Set XRE=+X,XIM=+$Piece(X,"%",2)
    Set YRE=+Y,YIM=+$Piece(Y,"%",2)
    Set D=YRE*YRE+(YIM*YIM)
    Set RE=XRE*YRE+(XIM*YIM)/D
    Set IM=XIM*YRE-(XRE*YIM)/D
    Quit RE_"%"_IM
```

3.3.4.7. Note(s)
The code in the previous section is an example of a possible implementation of this library function. Vendors are encouraged to provide implementations that offer a better efficiency as well as a greater accuracy.

3.3.5. CEXP Function
3.3.5.1. Library Element Description
CEXP, complex exponential function.

3.3.5.2. Definition
`CEXP^MATH:COMPLEX(Z:COMPLEX,PREC:INTEGER:O)`
The function returns the value of *e* raised to the power of the complex number Z. The number of significant digits in the complex exponent is specified by the optional parameter PREC. If not specified, a default value of 11 digits is assumed for PREC.

3.3.5.3. Domain
Standard.

3.3.5.4. Range
Standard.

3.3.5.5. Side effects
None.

3.3.5.6. Example of MUMPS code to implement
```
CEXP(Z,PREC) ;
    New R,ZIM,ZRE
    Set ZRE=+Z,ZIM=+$Piece(Z,"%",2)
    Set R=$%EXP^MATH(ZRE,PREC)
    Quit R*$%COS^MATH(ZIM,PREC)_"%"_(R*$%SIN^MATH(ZIM,PREC))
```

3.3.5.7. Note(s)
The code in the previous section is an example of a possible implementation of this library function. Vendors are encouraged to provide implementations that offer a better efficiency as well as a greater accuracy.

3.3.6. CLOG Function

3.3.6.1. Library Element Description

CLOG, complex Naperian logarithm function.

3.3.6.2. Definition

`CLOG^MATH:COMPLEX(Z:COMPLEX,PREC:INTEGER:O)`

The function returns the logarithm of the complex number Z. The number of significant digits in the complex logarithm is specified by the optional parameter PREC. If not specified, a default value of 11 digits is assumed for PREC.

3.3.6.3. Domain

If Im Z = 0, then Re Z > 0..

3.3.6.4. Range

Re $%CLOG^MATH(Z) can be any number.

$-\pi \leq$ Im $%CLOG^MATH(Z) $\leq \pi$.

3.3.6.5. Side effects

None.

3.3.6.6. Example of MUMPS code to implement

```
CLOG(Z,PREC) ;
    New ABS,ARG,ZIM,ZRE
    Set ABS=$%CABS^MATH(Z)
    Set ZRE=+Z,ZIM=+$Piece(Z,"%",2)
    Set ARG=$%ARCTAN^MATH(ZIM,ZRE,PREC)
    Quit $%LOG^MATH(ABS,PREC)_"%"_ARG
```

3.3.6.7. Note(s)

The code in the previous section is an example of a possible implementation of this library function. Vendors are encouraged to provide implementations that offer a better efficiency as well as a greater accuracy.

3.3.7. CMUL Function

3.3.7.1. Library Element Description

CMUL, complex multiplication function.

3.3.7.2. Definition

`CMUL^MATH:COMPLEX(X:COMPLEX,Y:COMPLEX)`

The function returns the value of X * Y, where X and Y are complex numbers.

3.3.7.3. Domain

Standard.

3.3.7.4. Range

Standard.

3.3.7.5. Side effects

None.

3.3.7.6. Example of MUMPS code to implement

```
CMUL(X,Y) ;
```

```
New XIM,XRE,YIM,YRE
Set XRE=+X,XIM=+$Piece(X,"%",2)
Set YRE=+Y,YIM=+$Piece(Y,"%",2)
Quit XRE*YRE-(XIM*YIM)_"%"_(XRE*YIM+(XIM*YRE))
```

3.3.7.7. Note(s)

The code in the previous section is an example of a possible implementation of this library function. Vendors are encouraged to provide implementations that offer a better efficiency as well as a greater accuracy.

3.3.8. COMPLEX Function

3.3.8.1. Library Element Description

COMPLEX, convert number to complex representation.

3.3.8.2. Definition

`COMPLEX^MATH:COMPLEX(X:REAL)`

The function returns the complex representation of the number specified in X.

3.3.8.3. Domain

Standard.

3.3.8.4. Range

Standard.

3.3.8.5. Side effects

None.

3.3.8.6. Example of MUMPS code to implement

```
COMPLEX(X) Quit +X_"%0"
```

3.3.8.7. Note(s)

The code in the previous section is an example of a possible implementation of this library function. Vendors are encouraged to provide implementations that offer a better efficiency as well as a greater accuracy.

3.3.9. CONJUG Function

3.3.9.1. Library Element Description

CONJUG, conjugate of complex number function.

3.3.9.2. Definition

`CONJUG^MATH:COMPLEX(Z:COMPLEX)`

The function returns the value of the conjugate of the the complex number Z.

3.3.9.3. Domain

Standard.

3.3.9.4. Range

Standard.

3.3.9.5. Side effects

None.

3.3.9.6. Example of MUMPS code to implement

```
CONJUG(Z) ;
   New ZIM,ZRE
   Set ZRE=+Z,ZIM=+$Piece(Z,"%",2)
   Quit ZRE_"%"_(-ZIM)
```

3.3.9.7. Note(s)

The code in the previous section is an example of a possible implementation of this library function. Vendors are encouraged to provide implementations that offer a better efficiency as well as a greater accuracy.

3.3.10. CPOWER Function

3.3.10.1. Library Element Description

CPOWER, complex exponentiation function.

3.3.10.2. Definition

`CPOWER^MATH:COMPLEX(Z:COMPLEX,X:COMPLEX,PREC:INTEGER:O)`

The function returns the value of the complex number Z raised to the power of the complex number X. The number of significant digits in the complex power is specified by the optional parameter PREC. If not specified, a default value of 11 digits is assumed for PREC.

3.3.10.3. Domain

Standard. If both Z and N are equal to zero (0 or "0%0), an error will occur with <u>ecode</u> "M28".

3.3.10.4. Range

Standard.

3.3.10.5. Side effects

None.

3.3.10.6. Example of MUMPS code to implement

```
CPOWER(Z,N,PREC) ;
   New AR,NIM,NRE,PHI,PI,R,RHO,TH,ZIM,ZRE
   Set ZRE=+Z,ZIM=+$Piece(Z,"%",2)
   Set NRE=+N,NIM=+$Piece(N,"%",2)
   If 'ZRE,'ZIM,'NRE,'NIM Set $Ecode=",M28,"
   If 'ZRE,'ZIM Quit "0%0%
   Set PI=$%PI^MATH()
   Set R=$%SQRT^MATH(ZRE*ZRE+(ZIM*ZIM,PREC))
   If ZRE Set TH=$%ARCTAN^MATH(ZIM,ZRE,PREC)
   Else  Set TH=$SELECT(ZRE>0:PI/2,1:-PI/2)
   Set RHO=$%LOG^MATH(R,PREC)
   Set AR=$%EXP^MATH(RHO*NRE-(TH*NIM),PREC)
   Set PHI=RHO*NIM+(NRE*TH)
   Quit AR*$%COS^MATH(PHI,PREC)_"%"_(AR*$%SIN^MATH(PHI,PREC))
```

3.3.10.7. Note(s)

The code in the previous section is an example of a possible implementation of this library function. Vendors are encouraged to provide implementations that offer a better efficiency as well as a greater accuracy.

3.3.11. CSIN Function

3.3.11.1. Library Element Description

CSIN, trigonometric complex sine function.

3.3.11.2. Definition

`CSIN^MATH:COMPLEX(Z:COMPLEX,PREC:INTEGER:0)`

The function returns the value of the trigonometric sine sin(Z) of the angle Z in radians. Z is interpreted as a complex number. The number of significant digits in the complex sine is specified by the optional parameter PREC. If not specified, a default value of 11 digits is assumed for PREC.

3.3.11.3. Domain

Standard.

3.3.11.4. Range

–1 ≤ Re $%CSIN^MATH(Z) ≤ 1
–1 ≤ Im $%CSIN^MATH(Z) ≤ 1.

3.3.11.5. Side effects

None.

3.3.11.6. Example of MUMPS code to implement

```
CSIN(Z,PREC) ;
    New IA,E1,E2
    Set IA=$%CMUL^MATH(Z,"0%1")
    Set E1=$%CEXP^MATH(IA,PREC)
    Set IA=-IA_"%"_(-$Piece(IA,"%",2))
    Set E2=$%CEXP^MATH(IA,PREC)
    Set IA=$%CSUB^MATH(E1,E2)
    Set IA=$%CMUL^MATH(IA,"0.5%0")
    Quit $%CMUL^MATH("0%-1",IA)
```

3.3.11.7. Note(s)

The code in the previous section is an example of a possible implementation of this library function. Vendors are encouraged to provide implementations that offer a better efficiency as well as a greater accuracy.

3.3.12. CSUB Function

3.3.12.1. Library Element Description

CSUB, complex subtraction function.

3.3.12.2. Definition

`CSUB^MATH:COMPLEX(X:COMPLEX,Y:COMPLEX)`

The function returns the value of X – Y, where X and Y are complex numbers.

3.3.12.3. Domain

Standard.

3.3.12.4. Range

Standard.

3.3.12.5. Side effects

None.

3.3.12.6. Example of MUMPS code to implement

```
CSUB(X,Y) ;
   New XIM,XRE,YIM,YRE
   Set XRE=+X,XIM=+$Piece(X,"%",2)
   Set YRE=+Y,YIM=+$Piece(Y,"%",2)
   Quit XRE-YRE_"%"_(XIM-YIM)
```

3.3.12.7. Note(s)

The code in the previous section is an example of a possible implementation of this library function. Vendors are encouraged to provide implementations that offer a better efficiency as well as a greater accuracy.

4. Implementation impacts

4.1. Impact on Existing User Practices and Investments

Minimal: existing user-written approximations will continue to be work.
Users may gain precision and performance by using the library functions.

4.2. Impact on Existing Vendor Practices and Investments

None or small. Some vendors already offer some of the proposed functionality.

4.3. Techniques and Costs for Compliance Verification

For each function, a test-suite will have to be developed to check that the function-value is within the limits derived from mathematically correct value and specified error-range (tolerance).

4.4. Legal considerations

None.

5. Closely related standards activities

5.1. Other X11 Proposals (Type A or Type B) Under Consideration

This proposal is one of a series of proposals by the Mathematical Functions taskgroup. The set comprises: trigonometric functions, hyperbolic functions, complex functions, other functions.

5.2. Other Related Standards Efforts

None.

5.3. Recommendations for Co-ordinating Liaison

None.

6. List of Associated Documents

MUMPS/ANSI standard

ISBN 0-201-03809-0 The Art of Computer Programming, Fundamental Algorithms (Knuth)
ISBN 0-486-61272-4 Handbook of mathematical functions (Abramowitz, Stegun)
ISBN 0-06-461019-5 Dictionary of Mathematics (Borowski, Borwein)

7. Issues, Pros and Cons, and Discussion

7.1. 25 October 1993, Dublin Ireland:

Pro 1: Improves existing document.
Con 1: Examples should be verified and annotated with appropriate caveats.
Con 2: Remove MUMPS code in examples.
Sponsor's note: the examples have been verified by the VA offices in Charleston and Hines.
Con 2 is hard to resolve, because the document format requires the presence of the MUMPS code.

7.2. 26 February 1994, Houston Texas:

Pro 1: Makes MUMPS more general use
Pro 2: Removed objection by significant Government Agency against the use of MUMPS.
Con 1: Normative.
Con 2: Mandates numeric precision which may not be available on all platforms.
Con 1: seems to be a matter of taste.
Con 2: The code, as provided and tested, will provide the stated precision, as long as the requested precision falls within the portability limits.

7.3. 12 June 1994, Reno Nevada:

In order to bring the document in synchronization with the latest Library Specification document, several editorial changes need to be made:

- Function type should follow function name: `name^MATH:type(params)` instead of `name^MATH(params):type`
- `$&` should be replaced by `$%`
- Section 3.3.x.6 should be renamed to **Example of MUMPS code to be implemented**
- Section 3.3.x.7 should be added to emphasize that the MUMPS code that is presented is merely an example and that vendors are encouraged to offer implementations that provide a better efficiency and accuracy

A number of enhancements was recommended:

- Do not use O or I as variable names (replaced by K and N where appropriate)
- The value of several trigonometric functions may be 'infinite'. In some cases, the range is described as 'standard'. As yet, infinity is not included in the 'standard' values of numbers.
- The return type for function CEXP was omitted.
- In the definition of the function ARCCOSH, the function name was misspelled twice

(once as ARCSIN, and once as ARCCOS).

- In the definition of the function ARCCSC, the function name was misspelled (as SIN).
- In a number of function definitions, the optional precision parameter was not specified.
- The definition of the functions LOG and LOG10 still contain some text that should have been removed when the function LN was deleted from the proposal.

Individually, the modifications to be made to the document were deemed editorial, in view of the number of modifications, no vote on the document was taken (postponed until an updated document is presented).