

! This is a test program for the FM 1.4 multiple-precision interval arithmetic package.

! All of the interval arithmetic routines are tested.

! Precision is set to 30 significant digits and the results are checked to that accuracy.

! If all tests are completed successfully, this line is printed:

! 1044 cases tested. No errors were found.

```
module test_vars_ia
```

```
use fmvals
use fmzm
use fm_interval_arithmetic
```

! Declare the derived type variables of type (fm_interval), (fm), (im), and (zm).
! These are in the form that would be found in a user program.

```
type (fm_interval), save :: a, b, c, d, result, correct,          &
                        avec(3), bvec(3), cvec(3), dvec(3),      &
                        amat(3, 3), bmat(3, 3), cmatrix(3, 3), dmat(3, 3)
```

```
type (fm), save :: error, mfm1, mfmv(3), mfmv2(3, 3), mv4(4), mv8(8)
type (im), save :: mim1, mimv(3), mimv2(3, 3)
type (zm), save :: mzm1, mzmv(3), mzmv2(3, 3)
```

! These are the variables that are not multiple precision.

```
integer, save :: j1, jv(3), jv2(3, 3)
real, save :: r1, rsmall, rv(3), rv2(3, 3)
double precision, save :: d1, dsmall, dv(3), dv2(3, 3)
complex, save :: c1, cv(3), cv2(3, 3)
complex (kind(0.0d0)), save :: cd1, cdv(3), cdv2(3, 3)
```

```
character(100), save :: st1, st2, string, stv2(3, 3)
integer, save :: klog, kwsave, ncase, nerror
real, save :: time1, time2
logical, external :: fmcompare
```

```
interface sum_ivl
  module procedure sum0
  module procedure sum1
  module procedure sum2
end interface
```

```
contains
```

```
function sum0(a, b) result (return_value)
```

! Function that returns an interval result.

```
implicit none
type (fm_interval) :: a, b, return_value
intent (in) :: a, b
return_value = a + b
end function sum0
```

```
function sum1(a, b)      result (return_value)
```

```
! Function that returns a interval vector result.
```

```
implicit none
type (fm_interval) :: a(3), b(3), return_value(3)
intent (in) :: a, b
integer :: j
do j = 1, 3
    return_value(j) = a(j) + b(j)
enddo
end function sum1
```

```
function sum2(a, b)      result (return_value)
```

```
! Function that returns a interval matrix result.
```

```
implicit none
type (fm_interval) :: a(3, 3), b(3, 3), return_value(3, 3)
intent (in) :: a, b
integer :: j, k
do j = 1, 3
    do k = 1, 3
        return_value(j, k) = a(j, k) + b(j, k)
    enddo
enddo
end function sum2
```

```
subroutine test1
```

```
! Input and output testing.
```

```
implicit none
integer :: j, k, l

write (kw, ("/ Testing input and output conversion for intervals. "))
```

```
!           ncase is the number of cases tested.
```

```
ncase = 1
result = to_fm_interval( to_fm(14) )
write (klog,*) ' '
write (klog, "(a, i6)") ' ncase = ', ncase
write (klog,*) ' to_fm_interval( to_fm(14) ) '
call fmform_interval('f40.35', result, string)
write (klog,*) ' ', trim(string)
write (klog,*) ' '
correct = 14
mfm1 = 14
```

```
!           Use the .not. because fmcompare returns false for special cases like md = unknown,
!           and these should be treated as errors for these tests.
```

```
if ( (.not. fmcompare(result%left, '==', mfm1%mfm)) .or. &
     (.not. fmcompare(result%right, '==', mfm1%mfm)) ) then
    call errprtfm(' to_fm_interval', 0, a, 'a', b, 'b', b, 'b')
endif
```

```

ncase = 2
result = to_fm_interval( 15 )
write (klog,*) ' '
write (klog, "(a, i6)") ' ncase = ', ncase
write (klog,*) ' to_fm_interval( 15 ) '
call fmform_interval('f40.35', result, string)
write (klog,*) ' ', trim(string)
write (klog,*) ' '
correct = 15
if (.not.(result == correct)) then
    call errprtfm(' to_fm_interval', 0, a, 'a', b, 'b', b, 'b')
endif

```

```

ncase = 3
result = to_fm_interval( 16.0 )
write (klog,*) ' '
write (klog, "(a, i6)") ' ncase = ', ncase
write (klog,*) ' to_fm_interval( 16.0 ) '
call fmform_interval('f40.35', result, string)
write (klog,*) ' ', trim(string)
write (klog,*) ' '
correct = 16
if (.not.(result == correct)) then
    call errprtfm(' to_fm_interval', 0, a, 'a', b, 'b', b, 'b')
endif

```

```

ncase = 4
call fm_interval_sp2m(16.0, result)
write (klog,*) ' '
write (klog, "(a, i6)") ' ncase = ', ncase
write (klog,*) ' to_fm_interval( 16.0 ) '
call fmform_interval('f40.35', result, string)
write (klog,*) ' ', trim(string)
write (klog,*) ' '
correct = 16
if (.not.(result == correct)) then
    call errprtfm(' to_fm_interval', 0, a, 'a', b, 'b', b, 'b')
endif

```

```

ncase = 5
result = to_fm_interval( 17.0d0 )
write (klog,*) ' '
write (klog, "(a, i6)") ' ncase = ', ncase
write (klog,*) ' to_fm_interval( 17.0d0 ) '
call fmform_interval('f40.35', result, string)
write (klog,*) ' ', trim(string)
write (klog,*) ' '
correct = 17
if (.not.(result == correct)) then
    call errprtfm(' to_fm_interval', 0, a, 'a', b, 'b', b, 'b')
endif

```

```

ncase = 6
result = to_fm_interval( cmplx( 18.0 , 28.0 ) )
write (klog,*) ' '
write (klog, "(a, i6)") ' ncase = ', ncase
write (klog,*) ' to_fm_interval( cmplx( 18.0 , 28.0 ) ) '
call fmform_interval('f40.35', result, string)

```

```

write (klog,*) ' ', trim(string)
write (klog,*) ' '
correct = 18
if (.not.(result == correct)) then
    call errprtfm(' to_fm_interval', 0, a, 'a', b, 'b', b, 'b')
endif

ncase = 7
result = to_fm_interval( cmplx( 19.0d0 , 29.0d0 , kind(1.0d0) ) )
write (klog,*) ' '
write (klog, "(a, i6)") ' ncase = ', ncase
write (klog,*) ' to_fm_interval( cmplx( 19.0d0 , 29.0d0 , kind(1.0d0) ) ) '
call fmform_interval('f40.35', result, string)
write (klog,*) ' ', trim(string)
write (klog,*) ' '
correct = 19
if (.not.(result == correct)) then
    call errprtfm(' to_fm_interval', 0, a, 'a', b, 'b', b, 'b')
endif

ncase = 8
a = to_fm_interval( 20 , 21 )
result = to_fm_interval( a )
write (klog,*) ' '
write (klog, "(a, i6)") ' ncase = ', ncase
write (klog,*) ' to_fm_interval( a ) '
call fmform_interval('f40.35', result, string)
write (klog,*) ' ', trim(string)
write (klog,*) ' '
correct = a
if (.not.(result == correct)) then
    call errprtfm(' to_fm_interval', 0, a, 'a', b, 'b', b, 'b')
endif

ncase = 9
result = to_fm_interval( to_im( 22 ) )
write (klog,*) ' '
write (klog, "(a, i6)") ' ncase = ', ncase
write (klog,*) ' to_fm_interval( to_im( 22 ) ) '
call fmform_interval('f40.35', result, string)
write (klog,*) ' ', trim(string)
write (klog,*) ' '
correct = 22
if (.not.(result == correct)) then
    call errprtfm(' to_fm_interval', 0, a, 'a', b, 'b', b, 'b')
endif

ncase = 10
mim1 = 22
call im_interval_i2fm(mim1, result)
write (klog,*) ' '
write (klog, "(a, i6)") ' ncase = ', ncase
write (klog,*) ' to_fm_interval( to_im( 22 ) ) '
call fmform_interval('f40.35', result, string)
write (klog,*) ' ', trim(string)
write (klog,*) ' '
correct = 22
if (.not.(result == correct)) then

```

```
    call errprtfm(' to_fm_interval', 0, a, 'a', b, 'b', b, 'b')
endif
```

```
ncase = 11
result = to_fm_interval( to_zm( " 23 + 33 i " ) )
write (klog,*) ' '
write (klog, "(a, i6)") ' ncase = ', ncase
write (klog,*) ' to_fm_interval( to_zm( " 23 + 33 i " ) ) '
call fmform_interval('f40.35', result, string)
write (klog,*) ' ', trim(string)
write (klog,*) ' '
correct = 23
if (.not.(result == correct)) then
    call errprtfm(' to_fm_interval', 0, a, 'a', b, 'b', b, 'b')
endif
```

```
ncase = 12
result = to_fm_interval( " 24 " )
write (klog,*) ' '
write (klog, "(a, i6)") ' ncase = ', ncase
write (klog,*) ' to_fm_interval( " 24 " ) '
call fmform_interval('f40.35', result, string)
write (klog,*) ' ', trim(string)
write (klog,*) ' '
correct = 24
if (.not.(result == correct)) then
    call errprtfm(' to_fm_interval', 0, a, 'a', b, 'b', b, 'b')
endif
```

```
ncase = 13
jv = (/ 25, 26, 27 /)
avec = to_fm_interval( jv )
write (klog,*) ' '
write (klog, "(a, i6)") ' ncase = ', ncase
write (klog,*) ' to_fm_interval( (/ 25, 26, 27 /) ) '
do j = 25, 27
    result = avec(j-24)
    correct = j
    call fmform_interval('f40.35', result, string)
    write (klog,*) ' ', trim(string)
    if (.not.(result == correct)) then
        call errprtfm(' to_fm_interval', 0, a, 'a', b, 'b', b, 'b')
        exit
    endif
enddo
write (klog,*) ' '

```

```
ncase = 14
rv = (/ 25.0, 26.0, 27.0 /)
avec = to_fm_interval( rv )
write (klog,*) ' '
write (klog, "(a, i6)") ' ncase = ', ncase
write (klog,*) ' to_fm_interval( (/ 25.0, 26.0, 27.0 /) ) '
do j = 25, 27
    result = avec(j-24)
    correct = j
    call fmform_interval('f40.35', result, string)
    write (klog,*) ' ', trim(string)

```

```

    if (.not.(result == correct)) then
        call errprtfm(' to_fm_interval', 0, a, 'a', b, 'b', b, 'b')
        exit
    endif
enddo
write (klog,*) ' '

ncase = 15
dv = (/ 25.0d0, 26.0d0, 27.0d0 /)
avec = to_fm_interval( dv )
write (klog,*) ' '
write (klog, "(a, i6)") ' ncase = ', ncase
write (klog,*) ' to_fm_interval( (/ 25.0d0, 26.0d0, 27.0d0 /) ) '
do j = 25, 27
    result = avec(j-24)
    correct = j
    call fmform_interval('f40.35', result, string)
    write (klog,*) ' ', trim(string)
    if (.not.(result == correct)) then
        call errprtfm(' to_fm_interval', 0, a, 'a', b, 'b', b, 'b')
        exit
    endif
enddo
write (klog,*) ' '

ncase = 16
cv = (/ cplx( 25.0 , 35.0 ), cplx( 26.0 , 36.0 ), cplx( 27.0 , 37.0 ) /)
avec = to_fm_interval( cv )
write (klog,*) ' '
write (klog, "(a, i6)") ' ncase = ', ncase
write (klog,*) ' to_fm_interval( cplx( 25.0 , 35.0 ), cplx( 26.0 , 36.0 ),', &
    ' cplx( 27.0 , 37.0 ) ) '
do j = 25, 27
    result = avec(j-24)
    correct = j
    call fmform_interval('f40.35', result, string)
    write (klog,*) ' ', trim(string)
    if (.not.(result == correct)) then
        call errprtfm(' to_fm_interval', 0, a, 'a', b, 'b', b, 'b')
        exit
    endif
enddo
write (klog,*) ' '

ncase = 17
cdv = (/ cplx( 25.0 , 35.0 , kind(1.0d0) ), cplx( 26.0 , 36.0 , kind(1.0d0) ), &
    cplx( 27.0 , 37.0 , kind(1.0d0) ) /)
avec = to_fm_interval( cdv )
write (klog,*) ' '
write (klog, "(a, i6)") ' ncase = ', ncase
write (klog,*) ' to_fm_interval( (/ cplx( 25.0 , 35.0 , kind(1.0d0) ), '
write (klog,*) ' cplx( 26.0 , 36.0 , kind(1.0d0) ), '
write (klog,*) ' cplx( 27.0 , 37.0 , kind(1.0d0) ) /) ) '
do j = 25, 27
    result = avec(j-24)
    correct = j
    call fmform_interval('f40.35', result, string)
    write (klog,*) ' ', trim(string)

```

```

    if (.not.(result == correct)) then
        call errprtfm(' to_fm_interval', 0, a, 'a', b, 'b', b, 'b')
        exit
    endif
enddo
write (klog,*) ' '

ncase = 18
bvec = (/ 25, 26, 27 /)
avec = to_fm_interval( bvec )
write (klog,*) ' '
write (klog, "(a, i6)") ' ncase = ', ncase
write (klog,*) ' to_fm_interval( (/ to_fm_interval(25), to_fm_interval(26),', &
    ' to_fm_interval(27) /) ) '
do j = 25, 27
    result = avec(j-24)
    correct = j
    call fmform_interval('f40.35', result, string)
    write (klog,*) ' ', trim(string)
    if (.not.(result == correct)) then
        call errprtfm(' to_fm_interval', 0, a, 'a', b, 'b', b, 'b')
        exit
    endif
enddo
write (klog,*) ' '

ncase = 19
avec = to_fm_interval( to_fm( (/ 25, 26, 27 /) ) )
write (klog,*) ' '
write (klog, "(a, i6)") ' ncase = ', ncase
write (klog,*) ' to_fm_interval( to_fm( (/ 25, 26, 27 /) ) ) '
do j = 25, 27
    result = avec(j-24)
    correct = j
    call fmform_interval('f40.35', result, string)
    write (klog,*) ' ', trim(string)
    if (.not.(result == correct)) then
        call errprtfm(' to_fm_interval', 0, a, 'a', b, 'b', b, 'b')
        exit
    endif
enddo
write (klog,*) ' '

ncase = 20
avec = to_fm_interval( to_im( (/ 25, 26, 27 /) ) )
write (klog,*) ' '
write (klog, "(a, i6)") ' ncase = ', ncase
write (klog,*) ' to_fm_interval( to_im( (/ 25, 26, 27 /) ) ) '
do j = 25, 27
    result = avec(j-24)
    correct = j
    call fmform_interval('f40.35', result, string)
    write (klog,*) ' ', trim(string)
    if (.not.(result == correct)) then
        call errprtfm(' to_fm_interval', 0, a, 'a', b, 'b', b, 'b')
        exit
    endif
enddo

```

```

write (klog,*) ' '

ncase = 21
avec = to_fm_interval( to_zm( (/ 25, 26, 27 /) ) )
write (klog,*) ' '
write (klog, "(a, i6)") ' ncase = ', ncase
write (klog,*) ' to_fm_interval( to_zm( (/ 25, 26, 27 /) ) ) '
do j = 25, 27
  result = avec(j-24)
  correct = j
  call fmform_interval('f40.35', result, string)
  write (klog,*) ' ', trim(string)
  if (.not.(result == correct)) then
    call errprtfm(' to_fm_interval', 0, a, 'a', b, 'b', b, 'b')
    exit
  endif
enddo
write (klog,*) ' '

ncase = 22
avec = to_fm_interval( (/ "25", "26", "27" /) )
write (klog,*) ' '
write (klog, "(a, i6)") ' ncase = ', ncase
write (klog,*) ' to_fm_interval( (/ "25", "26", "27" /) ) '
do j = 25, 27
  result = avec(j-24)
  correct = j
  call fmform_interval('f40.35', result, string)
  write (klog,*) ' ', trim(string)
  if (.not.(result == correct)) then
    call errprtfm(' to_fm_interval', 0, a, 'a', b, 'b', b, 'b')
    exit
  endif
enddo
write (klog,*) ' '

ncase = 23
do j = 1, 3
  do k = 1, 3
    jv2(j, k) = (j+30)*k
  enddo
enddo
amat = to_fm_interval( jv2 )
write (klog,*) ' '
write (klog, "(a, i6)") ' ncase = ', ncase
write (klog,*) ' to_fm_interval( jv2 ) '
do j = 1, 3
  write (klog, "(a, i4)") '          Row ', j
  do k = 1, 3
    result = amat(j, k)
    correct = (j+30)*k
    call fmform_interval('f40.35', result, string)
    write (klog,*) ' ', trim(string)
    if (.not.(result == correct)) then
      call errprtfm(' to_fm_interval', 0, a, 'a', b, 'b', b, 'b')
      exit
    endif
  enddo
enddo

```



```

enddo
write (klog,*) ' '

ncase = 24
do j = 1, 3
  do k = 1, 3
    rv2(j, k) = (j+30)*k
  enddo
enddo
amat = to_fm_interval( rv2 )
write (klog,*) ' '
write (klog, "(a, i6)") ' ncase = ', ncase
write (klog,*) ' to_fm_interval( rv2 ) '
do j = 1, 3
  write (klog, "(a, i4)") '          Row ', j
  do k = 1, 3
    result = amat(j, k)
    correct = (j+30)*k
    call fmform_interval('f40.35', result, string)
    write (klog,*) ' ', trim(string)
    if (.not.(result == correct)) then
      call errprtfm(' to_fm_interval', 0, a, 'a', b, 'b', b, 'b')
      exit
    endif
  enddo
enddo
write (klog,*) ' '

ncase = 25
do j = 1, 3
  do k = 1, 3
    dv2(j, k) = (j+30)*k
  enddo
enddo
amat = to_fm_interval( dv2 )
write (klog,*) ' '
write (klog, "(a, i6)") ' ncase = ', ncase
write (klog,*) ' to_fm_interval( dv2 ) '
do j = 1, 3
  write (klog, "(a, i4)") '          Row ', j
  do k = 1, 3
    result = amat(j, k)
    correct = (j+30)*k
    call fmform_interval('f40.35', result, string)
    write (klog,*) ' ', trim(string)
    if (.not.(result == correct)) then
      call errprtfm(' to_fm_interval', 0, a, 'a', b, 'b', b, 'b')
      exit
    endif
  enddo
enddo
write (klog,*) ' '

ncase = 26
do j = 1, 3
  do k = 1, 3
    cv2(j, k) = (j+30)*k
  enddo

```

```

enddo
amat = to_fm_interval( cv2 )
write (klog,*) ' '
write (klog, "(a, i6)") ' ncase = ', ncase
write (klog,*) ' to_fm_interval( cv2 ) '
do j = 1, 3
  write (klog, "(a, i4)") '          Row ', j
  do k = 1, 3
    result = amat(j, k)
    correct = (j+30)*k
    call fmform_interval('f40.35', result, string)
    write (klog,*) ' ', trim(string)
    if (.not.(result == correct)) then
      call errprtfm(' to_fm_interval', 0, a, 'a', b, 'b', b, 'b')
      exit
    endif
  enddo
enddo
write (klog,*) ' '

ncase = 27
do j = 1, 3
  do k = 1, 3
    cdv2(j, k) = (j+30)*k
  enddo
enddo
amat = to_fm_interval( cdv2 )
write (klog,*) ' '
write (klog, "(a, i6)") ' ncase = ', ncase
write (klog,*) ' to_fm_interval( cdv2 ) '
do j = 1, 3
  write (klog, "(a, i4)") '          Row ', j
  do k = 1, 3
    result = amat(j, k)
    correct = (j+30)*k
    call fmform_interval('f40.35', result, string)
    write (klog,*) ' ', trim(string)
    if (.not.(result == correct)) then
      call errprtfm(' to_fm_interval', 0, a, 'a', b, 'b', b, 'b')
      exit
    endif
  enddo
enddo
write (klog,*) ' '

ncase = 28
do j = 1, 3
  do k = 1, 3
    bmat(j, k) = (j+30)*k
  enddo
enddo
amat = to_fm_interval( bmat )
write (klog,*) ' '
write (klog, "(a, i6)") ' ncase = ', ncase
write (klog,*) ' to_fm_interval( bmat ) '
do j = 1, 3
  write (klog, "(a, i4)") '          Row ', j
  do k = 1, 3

```

```

    result = amat(j, k)
    correct = (j+30)*k
    call fmform_interval('f40.35', result, string)
    write (klog,*) ' ', trim(string)
    if (.not.(result == correct)) then
        call errprtfm(' to_fm_interval', 0, a, 'a', b, 'b', b, 'b')
        exit
    endif
enddo
enddo
write (klog,*) ' '

ncase = 29
do j = 1, 3
    do k = 1, 3
        mfmv2(j, k) = (j+30)*k
    enddo
enddo
amat = to_fm_interval( mfmv2 )
write (klog,*) ' '
write (klog, "(a, i6)") ' ncase = ', ncase
write (klog,*) ' to_fm_interval( mfmv2 ) '
do j = 1, 3
    write (klog, "(a, i4)") '          Row ', j
    do k = 1, 3
        result = amat(j, k)
        correct = (j+30)*k
        call fmform_interval('f40.35', result, string)
        write (klog,*) ' ', trim(string)
        if (.not.(result == correct)) then
            call errprtfm(' to_fm_interval', 0, a, 'a', b, 'b', b, 'b')
            exit
        endif
    enddo
enddo
write (klog,*) ' '

ncase = 30
do j = 1, 3
    do k = 1, 3
        mimv2(j, k) = (j+30)*k
    enddo
enddo
amat = to_fm_interval( mimv2 )
write (klog,*) ' '
write (klog, "(a, i6)") ' ncase = ', ncase
write (klog,*) ' to_fm_interval( mimv2 ) '
do j = 1, 3
    write (klog, "(a, i4)") '          Row ', j
    do k = 1, 3
        result = amat(j, k)
        correct = (j+30)*k
        call fmform_interval('f40.35', result, string)
        write (klog,*) ' ', trim(string)
        if (.not.(result == correct)) then
            call errprtfm(' to_fm_interval', 0, a, 'a', b, 'b', b, 'b')
            exit
        endif
    enddo
enddo

```

```

        enddo
    enddo
write (klog,*) ' '

ncase = 31
do j = 1, 3
    do k = 1, 3
        mzm2(j, k) = (j+30)*k
    enddo
enddo
amat = to_fm_interval( mzm2 )
write (klog,*) ' '
write (klog, "(a, i6)") ' ncase = ', ncase
write (klog,*) ' to_fm_interval( mzm2 ) '
do j = 1, 3
    write (klog, "(a, i4)") '          Row ', j
    do k = 1, 3
        result = amat(j, k)
        correct = (j+30)*k
        call fmform_interval('f40.35', result, string)
        write (klog,*) ' ', trim(string)
        if (.not.(result == correct)) then
            call errprtfm(' to_fm_interval', 0, a, 'a', b, 'b', b, 'b')
            exit
        endif
    enddo
enddo
write (klog,*) ' '

ncase = 32
do j = 1, 3
    do k = 1, 3
        write ( stv2(j, k) , "(i5)" ) (j+30)*k
    enddo
enddo
amat = to_fm_interval( stv2 )
write (klog,*) ' '
write (klog, "(a, i6)") ' ncase = ', ncase
write (klog,*) ' to_fm_interval( stv2 ) '
do j = 1, 3
    write (klog, "(a, i4)") '          Row ', j
    do k = 1, 3
        result = amat(j, k)
        correct = (j+30)*k
        call fmform_interval('f40.35', result, string)
        write (klog,*) ' ', trim(string)
        if (.not.(result == correct)) then
            call errprtfm(' to_fm_interval', 0, a, 'a', b, 'b', b, 'b')
            exit
        endif
    enddo
enddo
write (klog,*) ' '

ncase = 33
result = to_fm_interval( 41 , 42 )
write (klog,*) ' '
write (klog, "(a, i6)") ' ncase = ', ncase

```

```

write (klog,*) ' to_fm_interval( 41 , 42 ) '
call fmform_interval('f40.35', result, string)
write (klog,*) ' ', trim(string)
write (klog,*) ' '
call fmi2m(41, correct%left)
call fmi2m(42, correct%right)
if (.not.(result == correct)) then
    call errprtfm(' to_fm_interval', 0, a, 'a', b, 'b', b, 'b')
endif

ncase = 34
result = to_fm_interval( 41.0 , 42.0 )
write (klog,*) ' '
write (klog, "(a, i6)") ' ncase = ', ncase
write (klog,*) ' to_fm_interval( 41.0 , 42.0 ) '
call fmform_interval('f40.35', result, string)
write (klog,*) ' ', trim(string)
write (klog,*) ' '
correct = to_fm_interval( 41 , 42 )
if (.not.(result == correct)) then
    call errprtfm(' to_fm_interval', 0, a, 'a', b, 'b', b, 'b')
endif

ncase = 35
result = to_fm_interval( 41.0d0 , 42.0d0 )
write (klog,*) ' '
write (klog, "(a, i6)") ' ncase = ', ncase
write (klog,*) ' to_fm_interval( 41.0d0 , 42.0d0 ) '
call fmform_interval('f40.35', result, string)
write (klog,*) ' ', trim(string)
write (klog,*) ' '
correct = to_fm_interval( 41 , 42 )
if (.not.(result == correct)) then
    call errprtfm(' to_fm_interval', 0, a, 'a', b, 'b', b, 'b')
endif

ncase = 36
result = to_fm_interval( cplx( 41.0 , 51.0 ) , cplx( 42.0 , 52.0 ) )
write (klog,*) ' '
write (klog, "(a, i6)") ' ncase = ', ncase
write (klog,*) ' to_fm_interval( cplx( 41.0 , 51.0 ) , cplx( 42.0 , 52.0 ) ) '
call fmform_interval('f40.35', result, string)
write (klog,*) ' ', trim(string)
write (klog,*) ' '
correct = to_fm_interval( 41 , 42 )
if (.not.(result == correct)) then
    call errprtfm(' to_fm_interval', 0, a, 'a', b, 'b', b, 'b')
endif

ncase = 37
result = to_fm_interval( cplx( 41.0d0 , 51.0d0 , kind(1.0d0) ) , &
                        cplx( 42.0d0 , 52.0d0 , kind(1.0d0) ) )
write (klog,*) ' '
write (klog, "(a, i6)") ' ncase = ', ncase
write (klog,*) ' to_fm_interval( cplx( 41.0d0 , 51.0d0 , kind(1.0d0) )', &
                        ' cplx( 42.0d0 , 52.0d0 , kind(1.0d0) ) ) '
call fmform_interval('f40.35', result, string)
write (klog,*) ' ', trim(string)

```

```

write (klog,*) ' '
correct = to_fm_interval( 41 , 42 )
if (.not.(result == correct)) then
    call errprtfm(' to_fm_interval', 0, a, 'a', b, 'b', b, 'b')
endif

ncase = 38
result = to_fm_interval( to_fm(41) , to_fm(42) )
write (klog,*) ' '
write (klog, "(a, i6)") ' ncase = ', ncase
write (klog,*) ' to_fm_interval( to_fm(41) , to_fm(42) ) '
call fmform_interval('f40.35', result, string)
write (klog,*) ' ', trim(string)
write (klog,*) ' '
correct = to_fm_interval( 41 , 42 )
if (.not.(result == correct)) then
    call errprtfm(' to_fm_interval', 0, a, 'a', b, 'b', b, 'b')
endif

ncase = 39
result = to_fm_interval( to_im(41) , to_im(42) )
write (klog,*) ' '
write (klog, "(a, i6)") ' ncase = ', ncase
write (klog,*) ' to_fm_interval( to_im(41) , to_im(42) ) '
call fmform_interval('f40.35', result, string)
write (klog,*) ' ', trim(string)
write (klog,*) ' '
correct = to_fm_interval( 41 , 42 )
if (.not.(result == correct)) then
    call errprtfm(' to_fm_interval', 0, a, 'a', b, 'b', b, 'b')
endif

ncase = 40
result = to_fm_interval( to_zm(41) , to_zm(42) )
write (klog,*) ' '
write (klog, "(a, i6)") ' ncase = ', ncase
write (klog,*) ' to_fm_interval( to_zm(41) , to_zm(42) ) '
call fmform_interval('f40.35', result, string)
write (klog,*) ' ', trim(string)
write (klog,*) ' '
correct = to_fm_interval( 41 , 42 )
if (.not.(result == correct)) then
    call errprtfm(' to_fm_interval', 0, a, 'a', b, 'b', b, 'b')
endif

ncase = 41
result = to_fm_interval( "41" , "42" )
write (klog,*) ' '
write (klog, "(a, i6)") ' ncase = ', ncase
write (klog,*) ' to_fm_interval( "41" , "42" ) '
call fmform_interval('f40.35', result, string)
write (klog,*) ' ', trim(string)
write (klog,*) ' '
correct = to_fm_interval( 41 , 42 )
if (.not.(result == correct)) then
    call errprtfm(' to_fm_interval', 0, a, 'a', b, 'b', b, 'b')
endif

```

```

ncase = 42
a = to_fm_interval( 43 , 45 )
result = to_fm(a)
write (klog,*) ' '
write (klog, "(a, i6)") ' ncase = ', ncase
write (klog,*) ' to_fm(a) '
call fm_form('f40.35', to_fm(a), string)
write (klog,*) ' ', trim(string)
write (klog,*) ' '
correct = to_fm_interval( 44 , 44 )
if (.not.(result == correct)) then
    call errprtfm(' to_fm', 0, a, 'a', b, 'b', b, 'b')
endif

ncase = 43
mfmv = to_fm( (/ to_fm_interval( 43 , 45 ) , to_fm_interval( 44 , 46 ) , &
                to_fm_interval( 47 , 48 ) /) )
dv = (/ 44.0d0, 45.0d0, 47.5d0 /)
write (klog,*) ' '
write (klog, "(a, i6)") ' ncase = ', ncase
write (klog,*) ' to_fm( (/ to_fm_interval( 43 , 45 ) , to_fm_interval( 44 , 46 ) , ' , &
                ' to_fm_interval( 47 , 48 ) /) ) '

do j = 1, 3
    result = mfmv(j)
    call fm_form('f40.35', mfmv(j), string)
    write (klog,*) ' ', trim(string)
    correct = dv(j)
    if (.not.(result == correct)) then
        call errprtfm(' to_fm', 0, a, 'a', b, 'b', b, 'b')
        exit
    endif
enddo
write (klog,*) ' '

ncase = 44
do j = 1, 3
    do k = 1, 3
        amat(j, k) = to_fm_interval( (j+50)*k , (j+55)*k )
        dv2(j, k) = (2*j+105)*k/2.0d0
    enddo
enddo
mfmv2 = to_fm( amat )
write (klog,*) ' '
write (klog, "(a, i6)") ' ncase = ', ncase
write (klog,*) ' to_fm( amat ) '
do j = 1, 3
    write (klog,*) ' Row ', j
    do k = 1, 3
        result = mfmv2(j, k)
        call fm_form('f40.35', mfmv2(j, k), string)
        write (klog,*) ' ', trim(string)
        correct = dv2(j, k)
        if (.not.(result == correct)) then
            call errprtfm(' to_fm', 0, a, 'a', b, 'b', b, 'b')
            exit
        endif
    enddo
enddo

```

```

write (klog,*) ' '

ncase = 45
a = to_fm_interval( 43 , 45 )
result = to_im(a)
write (klog,*) ' '
write (klog, "(a, i6)") ' ncase = ', ncase
write (klog,*) ' to_im(a) '
call im_form('i10', to_im(a), string)
write (klog,*) ' ', trim(string)
write (klog,*) ' '
correct = to_fm_interval( 44 )
if (.not.(result == correct)) then
    call errprtfm(' to_im', 0, a, 'a', b, 'b', b, 'b')
endif

ncase = 46
mimv = to_im( (/ to_fm_interval( 43 , 45 ) , to_fm_interval( 44 , 46 ) , &
                to_fm_interval( 47 , 48 ) /) )
jv = (/ 44, 45, 47 /)
write (klog,*) ' '
write (klog, "(a, i6)") ' ncase = ', ncase
write (klog,*) ' to_im( (/ to_fm_interval( 43 , 45 ) , to_fm_interval( 44 , 46 ) , ' , &
                    ' to_fm_interval( 47 , 48 ) /) ) '

do j = 1, 3
    result = mimv(j)
    call im_form('i10', mimv(j), string)
    write (klog,*) ' ', trim(string)
    correct = jv(j)
    if (.not.(result == correct)) then
        call errprtfm(' to_im', 0, a, 'a', b, 'b', b, 'b')
        exit
    endif
enddo
write (klog,*) ' '

ncase = 47
do j = 1, 3
    do k = 1, 3
        amat(j, k) = to_fm_interval( (j+50)*k , (j+55)*k )
        jv2(j, k) = (2*j+105)*k/2
    enddo
enddo
mimv2 = to_im( amat )
write (klog,*) ' '
write (klog, "(a, i6)") ' ncase = ', ncase
write (klog,*) ' to_im( amat ) '
do j = 1, 3
    write (klog,*) ' Row ', j
    do k = 1, 3
        result = mimv2(j, k)
        call im_form('i10', mimv2(j, k), string)
        write (klog,*) ' ', trim(string)
        correct = jv2(j, k)
        if (.not.(result == correct)) then
            call errprtfm(' to_im', 0, a, 'a', b, 'b', b, 'b')
            exit
        endif
    enddo
enddo

```



```

    enddo
enddo
write (klog,*) ' '

ncase = 48
a = to_fm_interval( 43 , 45 )
result = to_zm(a)
write (klog,*) ' '
write (klog, "(a, i6)") ' ncase = ', ncase
write (klog,*) ' to_zm(a) '
call zm_form('f30.25', 'f30.25', to_zm(a), string)
write (klog,*) ' ', trim(string)
write (klog,*) ' '
correct = to_fm_interval( 44 , 44 )
if (.not.(result == correct)) then
    call errprtfm(' to_zm', 0, a, 'a', b, 'b', b, 'b')
endif

ncase = 49
mzmv = to_zm( (/ to_fm_interval( 43 , 45 ) , to_fm_interval( 44 , 46 ) , &
                to_fm_interval( 47 , 48 ) /) )
rv = (/ 44.0, 45.0, 47.5 /)
write (klog,*) ' '
write (klog, "(a, i6)") ' ncase = ', ncase
write (klog,*) ' to_zm( (/ to_fm_interval( 43 , 45 ) , to_fm_interval( 44 , 46 ) ,' , &
                ' to_fm_interval( 47 , 48 ) /) ) '

do j = 1, 3
    result = mzmv(j)
    call zm_form('f30.25', 'f30.25', mzmv(j), string)
    write (klog,*) ' ', trim(string)
    correct = rv(j)
    if (.not.(result == correct)) then
        call errprtfm(' to_zm', 0, a, 'a', b, 'b', b, 'b')
        exit
    endif
enddo
write (klog,*) ' '

ncase = 50
do j = 1, 3
    do k = 1, 3
        amat(j, k) = to_fm_interval( (j+50)*k , (j+55)*k )
        rv2(j, k) = ( (j+50)*k + (j+55)*k ) / 2.0
    enddo
enddo
mzmv2 = to_zm( amat )
write (klog,*) ' '
write (klog, "(a, i6)") ' ncase = ', ncase
write (klog,*) ' to_zm( amat ) '
do j = 1, 3
    write (klog,*) ' Row ', j
    do k = 1, 3
        result = mzmv2(j, k)
        call zm_form('f30.25', 'f30.25', mzmv2(j, k), string)
        write (klog,*) ' ', trim(string)
        correct = rv2(j, k)
        if (.not.(result == correct)) then
            call errprtfm(' to_zm', 0, a, 'a', b, 'b', b, 'b')
        endif
    enddo
enddo

```

```

        exit
    endif
enddo
enddo
write (klog,*) ' '

ncase = 51
open(42, file="RWtest")
do j = 1, 3
    do k = 1, 3
        amat(j, k) = to_fm_interval( (j+50)*k , (j+55)*k )
        call fm_interval_write(42, amat(j, k))
    enddo
enddo
write (klog,*) ' '
write (klog, "(a, i6)") ' ncase = ', ncase
write (klog,*) ' call fm_interval_write( amat ) and call fm_interval_read( amat ) '
close(42)
open(42, file="RWtest")
do j = 1, 3
    write (klog,*) ' Row ', j
    do k = 1, 3
        call fm_interval_read(42, bmat(j, k))
        result = bmat(j, k)
        call fmform_interval('f40.35', result, string)
        write (klog,*) ' ', trim(string)
        correct = amat(j, k)
        if (.not.(result == correct)) then
            call errprtfm(' RWtest', 0, a, 'a', b, 'b', b, 'b')
            exit
        endif
    enddo
enddo
write (klog,*) ' '
close(42)

ncase = 52
open(42, file="RWtest")
do j = 1, 3
    do k = 1, 3
        amat(j, k) = to_fm_interval( (j+50)*k , (j+55)*k )
        l = kw
        kw = 42
        call fmprint_interval(amat(j, k))
        kw = l
    enddo
enddo
write (klog,*) ' '
write (klog, "(a, i6)") ' ncase = ', ncase
write (klog,*) ' call fm_interval_write( amat ) and call fm_interval_read( amat ) '
close(42)
open(42, file="RWtest")
do j = 1, 3
    write (klog,*) ' Row ', j
    do k = 1, 3
        call fm_interval_read(42, bmat(j, k))
        result = bmat(j, k)
        call fmform_interval('f40.35', result, string)
    enddo
enddo

```

```

write (klog,*) ' ', trim(string)
correct = amat(j, k)
if (.not.(result == correct)) then
    call errprtfm(' RWtest', 0, a, 'a', b, 'b', b, 'b')
    exit
endif
enddo
enddo
write (klog,*) ' '
close(42)

return
end subroutine test1

```

```

subroutine test2

```

! +, -, *, / testing.

```

implicit none

```

```

write (kw, "(/' Testing +, -, *, /.)")

```

! ncase is the number of cases tested.

```

ncase = 53
a = to_fm_interval( 3 , 5 )
b = to_fm_interval( 2 , 7 )
result = a + b
write (klog,*) ' '
write (klog, "(a, i6)") ' ncase = ', ncase
write (klog,*) ' ( 3 , 5 ) + ( 2 , 7 ) = '
call fmform_interval('f40.35', result, string)
write (klog,*) ' ', trim(string)
write (klog,*) ' '
correct = to_fm_interval( 5 , 12 )
if (.not.(result == correct)) then
    call errprtfm(' Addition', 2, a, 'a', b, 'b', b, 'b')
endif

```

```

ncase = 54
a = to_fm_interval( 3 , 5 )
b = to_fm_interval( 2 , 7 )
result = a - b
write (klog,*) ' '
write (klog, "(a, i6)") ' ncase = ', ncase
write (klog,*) ' ( 3 , 5 ) - ( 2 , 7 ) = '
call fmform_interval('f40.35', result, string)
write (klog,*) ' ', trim(string)
write (klog,*) ' '
correct = to_fm_interval( -4 , 3 )
if (.not.(result == correct)) then
    call errprtfm(' Subtraction', 2, a, 'a', b, 'b', b, 'b')
endif

```

```

ncase = 55
a = to_fm_interval( 3 , 5 )
b = to_fm_interval( 2 , 7 )
call fm_interval_sub(a, b, result)

```

```

write (klog,*) ' '
write (klog, "(a, i6)") ' ncase = ', ncase
write (klog,*) ' ( 3 , 5 ) - ( 2 , 7 ) = '
call fmform_interval('f40.35', result, string)
write (klog,*) ' ', trim(string)
write (klog,*) ' '
correct = to_fm_interval( -4 , 3 )
if (.not.(result == correct)) then
    call errprtfm(' Subtraction', 2, a, 'a', b, 'b', b, 'b')
endif

ncase = 56
a = to_fm_interval( 3 , 5 )
b = to_fm_interval( 2 , 7 )
result = a
call fm_interval_sub_r1(result, b)
write (klog,*) ' '
write (klog, "(a, i6)") ' ncase = ', ncase
write (klog,*) ' ( 3 , 5 ) - ( 2 , 7 ) = '
call fmform_interval('f40.35', result, string)
write (klog,*) ' ', trim(string)
write (klog,*) ' '
correct = to_fm_interval( -4 , 3 )
if (.not.(result == correct)) then
    call errprtfm(' Subtraction', 2, a, 'a', b, 'b', b, 'b')
endif

ncase = 57
a = to_fm_interval( 3 , 5 )
b = to_fm_interval( 2 , 7 )
result = b
call fm_interval_sub_r2(a, result)
write (klog,*) ' '
write (klog, "(a, i6)") ' ncase = ', ncase
write (klog,*) ' ( 3 , 5 ) - ( 2 , 7 ) = '
call fmform_interval('f40.35', result, string)
write (klog,*) ' ', trim(string)
write (klog,*) ' '
correct = to_fm_interval( -4 , 3 )
if (.not.(result == correct)) then
    call errprtfm(' Subtraction', 2, a, 'a', b, 'b', b, 'b')
endif

ncase = 58
a = to_fm_interval( 3 , 5 )
b = to_fm_interval( 2 , 7 )
result = a * b
write (klog,*) ' '
write (klog, "(a, i6)") ' ncase = ', ncase
write (klog,*) ' ( 3 , 5 ) * ( 2 , 7 ) = '
call fmform_interval('f40.35', result, string)
write (klog,*) ' ', trim(string)
write (klog,*) ' '
correct = to_fm_interval( 6 , 35 )
if (.not.(result == correct)) then
    call errprtfm(' Multiplication', 2, a, 'a', b, 'b', b, 'b')
endif

```