

! This is a test program for version 1.4 of module fm\_quad\_int, which contains the interface routines allowing quadruple-length integer variables in the user's program to be used in assignments, arithmetic, and comparisons involving type (fm), (im), and (zm) variables. The same operations are provided as those in the basic module fmzm for single or double precision variables.

! All of the routines in module fm\_quad\_int are tested, and if all tests are completed successfully, this line is printed:

! 280 cases tested. No errors were found.

```
module test_vars

use fmvals
use fmzm
use fm_quad_int

type (fm), save :: m_a, mfm1, mfm2, mfm3, mfm4, mfm6, &
                  mfmv1(3), mfmv2(3), &
                  mfma(3, 3), mfmb(3, 3)

type (im), save :: m_j, mim1, mim2, mim3, mim4, mim5
type (im), save, dimension(3) :: mimv1, mimv2
type (im), save, dimension(3, 3) :: mima2, mimb2

type (zm), save :: m_z, mzm1, mzm2, mzm3, mzm4, mzm5, &
                  mzmv1(3), mzmv2(3), &
                  mzma2(3, 3), mzmb2(3, 3)

integer (quad_int), save :: qi1, qi2, qi3, qi4, qi5, qiv(3), qiv2(3, 3)
real, save :: rv(3), rv2(3, 3)
double precision, save :: dv(3), dv2(3, 3)
complex, save :: cv(3), cv2(3, 3)
complex (kind(0.0d0)), save :: cdv(3), cdv2(3, 3)

integer, save :: klog, kwsave, ncase, nerror
real, save :: time1, time2

end module test_vars

module test_a
use test_vars

contains

subroutine test1

!           Test the = assignment interface.

implicit none

write (kw, "(/' Testing the derived type = interface.')")

ncase = 1
qi4 = mfm1
if (qi4 /= 581) call prtterr(kw)
```

```

ncase = 2
qi4 = mim1
if (qi4 /= 661) call prtterr(kw)

ncase = 3
qi4 = mzm1
if (qi4 /= 731) call prtterr(kw)

ncase = 4
mfm3 = qi2
call fm_st2m('123456789012345678901234567', mfm4)
call fm_sub(mfm3, mfm4, mfm6)
call fm_eq(mfm6, mfm4)
call fm_abs(mfm4, mfm6)
call fm_eq(mfm6, mfm4)
call fm_st2m('0', mfm3)
if (fm_compare(mfm4, 'gt', mfm3)) call prtterr(kw)

ncase = 5
mfm3 = qi2
call fm_st2m('123456789012345678901234567', mfm4)
call fm_sub(mfm3, mfm4, mfm6)
call fm_equ(mfm6, mfm4, ndig, ndig)
call fm_abs(mfm4, mfm6)
call fm_equ(mfm6, mfm4, ndig, ndig)
call fm_st2m('0', mfm3)
if (fm_compare(mfm4, 'gt', mfm3)) call prtterr(kw)

ncase = 6
mfm3 = qi2
call fm_st2m('123456789012345678901234567', mfm4)
call fm_sub_r2(mfm3, mfm4)
call fm_abs(mfm4, mfm6)
call fm_equ(mfm6, mfm4, ndig, ndig)
call fm_st2m('0', mfm3)
if (fm_compare(mfm4, 'gt', mfm3)) call prtterr(kw)

ncase = 7
mfm3 = qi2
mfm4 = to_quad_int(mfm3)
call fm_sub_r2(mfm3, mfm4)
call fm_abs(mfm4, mfm6)
call fm_equ(mfm6, mfm4, ndig, ndig)
call fm_st2m('0', mfm3)
if (fm_compare(mfm4, 'gt', mfm3)) call prtterr(kw)

ncase = 8
mfm3 = qi2
call fm_st2m('123456789012345678901234567', mfm4)
call fm_equ(mfm3, mfm6, ndig, ndig)
call fm_sub_r1(mfm6, mfm4)
call fm_equ(mfm6, mfm4, ndig, ndig)
call fm_abs(mfm4, mfm6)
call fm_equ_r1(mfm6, ndig, ndig)
call fm_eq(mfm6, mfm4)
call fm_st2m('0', mfm3)
if (fm_compare(mfm4, 'gt', mfm3)) call prtterr(kw)

```

```

ncase = 9
mim3 = qi2
call im_st2m('123456789012345678901234567', mim4)
call im_sub(mim3, mim4, mim5)
call im_eq(mim5, mim4)
call im_st2m('0', mim3)
if (im_compare(mim4, 'gt', mim3)) call prtterr(kw)

ncase = 10
mim3 = to_im(qi2)
call im_st2m('123456789012345678901234567', mim4)
call im_sub(mim3, mim4, mim5)
call im_eq(mim5, mim4)
call im_st2m('0', mim3)
if (im_compare(mim4, 'gt', mim3)) call prtterr(kw)

ncase = 11
mim3 = to_im(qi2)
mim4 = to_quad_int(mim3)
call im_sub(mim3, mim4, mim5)
call im_eq(mim5, mim4)
call im_st2m('0', mim3)
if (im_compare(mim4, 'gt', mim3)) call prtterr(kw)

ncase = 12
mzm3 = qi2
call zm_st2m('123456789012345678901234567', mzm4)
call zm_sub(mzm3, mzm4, mzm5)
call zm_eq(mzm5, mzm4)
call zm_abs(mzm4, mfm4)
call fm_st2m('0', mfm3)
if (fm_comp(mfm4, 'gt', mfm3)) call prtterr(kw)

ncase = 13
mzm3 = to_zm(qi2)
call zm_st2m('123456789012345678901234567', mzm4)
call zm_sub(mzm3, mzm4, mzm5)
call zm_eq(mzm5, mzm4)
call zm_abs(mzm4, mfm4)
call fm_st2m('0', mfm3)
if (fm_comp(mfm4, 'gt', mfm3)) call prtterr(kw)

ncase = 14
mzm3 = to_zm(qi2)
mzm4 = to_quad_int(mzm3)
call zm_sub(mzm3, mzm4, mzm5)
call zm_eq(mzm5, mzm4)
call zm_abs(mzm4, mfm4)
call fm_st2m('0', mfm3)
if (fm_comp(mfm4, 'gt', mfm3)) call prtterr(kw)

ncase = 15
qi1 = 123
mzm3 = to_zm(qi1, qi2)
call zm_st2m('123 + 123456789012345678901234567 i', mzm4)
call zm_sub(mzm3, mzm4, mzm5)
call zm_eq(mzm5, mzm4)

```

```

call zm_abs(mzm4, mfm4)
call fm_st2m('0', mfm3)
if (fm_comp(mfm4, 'gt', mfm3)) call prtterr(kw)

end subroutine test1

subroutine test2

```

! Test the derived type == interface.

```

implicit none

write (kw, "(/' Testing the derived type == interface.')"

ncase = 16
qi1 = 123
m_a = qi1
if (.not.(m_a == qi1)) then
    call errprt_fm(' == ', m_a, 'm_a', m_a, 'm_a', m_a, 'm_a')
endif

ncase = 17
qi1 = 123
m_a = qi1
if (.not.(qi1 == m_a)) then
    call errprt_fm(' == ', m_a, 'm_a', m_a, 'm_a', m_a, 'm_a')
endif

ncase = 18
qi1 = 123
m_j = qi1
if (.not.(m_j == qi1)) then
    call errprt_fm(' == ', m_j, 'm_j', m_j, 'm_j')
endif

ncase = 19
qi1 = 123
m_j = qi1
if (.not.(qi1 == m_j)) then
    call errprt_fm(' == ', m_j, 'm_j', m_j, 'm_j')
endif

ncase = 20
qi1 = 123
m_z = qi1
if (.not.(m_z == qi1)) then
    call errprt_fm(' == ', m_z, 'm_z', m_z, 'm_z', m_z, 'm_z')
endif

ncase = 21
qi1 = 123
m_z = ( 123.0 , 34.5 )
if (m_z == qi1) then
    call errprt_fm(' == ', m_z, 'm_z', m_z, 'm_z', m_z, 'm_z')
endif

ncase = 22
qi1 = 123

```

```

m_z = qi1
if (.not.(qi1 == m_z)) then
    call errprtz_m(' == ', m_z, 'm_z', m_z, 'm_z', m_z, 'm_z')
endif

ncase = 23
qi1 = 123
m_z = ( 123.0 , 34.5 )
if (qi1 == m_z) then
    call errprtz_m(' == ', m_z, 'm_z', m_z, 'm_z', m_z, 'm_z')
endif

return
end subroutine test2

subroutine test3

```

! Test the derived type /= interface.

```

implicit none

write (kw, "(/' Testing the derived type /= interface.')")

ncase = 24
qi1 = 123
m_a = 1 + qi1
if (.not.(m_a /= qi1)) then
    call errprtf_m(' /= ', m_a, 'm_a', m_a, 'm_a', m_a, 'm_a')
endif

ncase = 25
qi1 = 123
m_a = 1 + qi1
if (.not.(qi1 /= m_a)) then
    call errprtf_m(' /= ', m_a, 'm_a', m_a, 'm_a', m_a, 'm_a')
endif

ncase = 26
qi1 = 123
m_j = 1 + qi1
if (.not.(m_j /= qi1)) then
    call errprt_im(' /= ', m_j, 'm_j', m_j, 'm_j')
endif

ncase = 27
qi1 = 123
m_j = 1 + qi1
if (.not.(qi1 /= m_j)) then
    call errprt_im(' /= ', m_j, 'm_j', m_j, 'm_j')
endif

ncase = 28
qi1 = 123
m_z = 1 + qi1
if (.not.(m_z /= qi1)) then
    call errprtz_m(' /= ', m_z, 'm_z', m_z, 'm_z', m_z, 'm_z')
endif

```

```

ncase = 29
qi1 = 123
m_z = ( 123.0 , 34.5 )
if (.not.(m_z /= qi1)) then
    call errprt_zm(' /= ', m_z, 'm_z', m_z, 'm_z', m_z, 'm_z')
endif

ncase = 30
qi1 = 123
m_z = 1 + qi1
if (.not.(qi1 /= m_z)) then
    call errprt_zm(' /= ', m_z, 'm_z', m_z, 'm_z', m_z, 'm_z')
endif

ncase = 31
qi1 = 123
m_z = ( 123.0 , 34.5 )
if (.not.(qi1 /= m_z)) then
    call errprt_zm(' /= ', m_z, 'm_z', m_z, 'm_z', m_z, 'm_z')
endif

return
end subroutine test3

subroutine test4

```

! Test the derived type > interface.

```

implicit none

write (kw, "(/' Testing the derived type > interface.')")

ncase = 32
qi1 = 123
m_a = qi1 + 1
if (.not.(m_a > qi1)) then
    call errprt_fm(' > ', m_a, 'm_a', m_a, 'm_a', m_a, 'm_a')
endif

ncase = 33
qi1 = 123
m_a = qi1 - 1
if (.not.(qi1 > m_a)) then
    call errprt_fm(' > ', m_a, 'm_a', m_a, 'm_a', m_a, 'm_a')
endif

ncase = 34
qi1 = 123
m_j = qi1 + 1
if (.not.(m_j > qi1)) then
    call errprt_im(' > ', m_j, 'm_j', m_j, 'm_j')
endif

ncase = 35
qi1 = 123
m_j = qi1 - 1
if (.not.(qi1 > m_j)) then
    call errprt_im(' > ', m_j, 'm_j', m_j, 'm_j')
endif

```

```

endif

return
end subroutine test4

subroutine test5

```

! Test the derived type >= interface.

```

implicit none

write (kw, "(/' Testing the derived type >= interface.')")

ncase = 36
qi1 = 123
m_a = qi1 + 1
if (.not.(m_a >= qi1)) then
    call errprt_fm(' >= ', m_a, 'm_a', m_a, 'm_a', m_a, 'm_a')
endif

ncase = 37
qi1 = 123
m_a = qi1 - 1
if (.not.(qi1 >= m_a)) then
    call errprt_fm(' >= ', m_a, 'm_a', m_a, 'm_a', m_a, 'm_a')
endif

ncase = 38
qi1 = 123
m_j = qi1 + 1
if (.not.(m_j >= qi1)) then
    call errprt_im(' >= ', m_j, 'm_j', m_j, 'm_j')
endif

ncase = 39
qi1 = 123
m_j = qi1 - 1
if (.not.(qi1 >= m_j)) then
    call errprt_im(' >= ', m_j, 'm_j', m_j, 'm_j')
endif

return
end subroutine test5

subroutine test6

```

! Test the derived type < interface.

```

implicit none

write (kw, "(/' Testing the derived type < interface.')")

ncase = 40
qi1 = 123
m_a = qi1 - 2
if (.not.(m_a < qi1)) then
    call errprt_fm(' < ', m_a, 'm_a', m_a, 'm_a', m_a, 'm_a')
endif

```

```

ncase = 41
qi1 = 123
m_a = qi1 + 2
if (.not.(qi1 < m_a)) then
    call errprt_fm(' < ', m_a, 'm_a', m_a, 'm_a', m_a, 'm_a')
endif

ncase = 42
qi1 = 123
m_j = qi1 - 2
if (.not.(m_j < qi1)) then
    call errprt_im(' < ', m_j, 'm_j', m_j, 'm_j')
endif

ncase = 43
qi1 = 123
m_j = qi1 + 2
if (.not.(qi1 < m_j)) then
    call errprt_im(' < ', m_j, 'm_j', m_j, 'm_j')
endif

return
end subroutine test6

subroutine test7

```

! Test the derived type <= interface.

```

implicit none

write (kw, "(/' Testing the derived type <= interface.')"

ncase = 44
qi1 = 123
m_a = qi1 - 2
if (.not.(m_a <= qi1)) then
    call errprt_fm(' <= ', m_a, 'm_a', m_a, 'm_a', m_a, 'm_a')
endif

ncase = 45
qi1 = 123
m_a = qi1 + 2
if (.not.(qi1 <= m_a)) then
    call errprt_fm(' <= ', m_a, 'm_a', m_a, 'm_a', m_a, 'm_a')
endif

ncase = 46
qi1 = 123
m_j = qi1 - 2
if (.not.(m_j <= qi1)) then
    call errprt_im(' <= ', m_j, 'm_j', m_j, 'm_j')
endif

ncase = 47
qi1 = 123
m_j = qi1 + 2
if (.not.(qi1 <= m_j)) then

```



```
    call errprt_im(' <= ', m_j, 'm_j', m_j, 'm_j')
endif
```

```
return
end subroutine test7
```

```
subroutine test8
```

!            Test the '+' arithmetic operator.

```
implicit none
```

```
write (kw, "(/' Testing the derived type + interface.')")
```

```
ncase = 48
mfm3 = qi2 + mfm1
call fm_st2m('123456789012345678901234567', mfm4)
call fm_add(mfm4, mfm1, mfm6)
call fm_eq(mfm6, mfm4)
if (.not.(mfm3 == mfm4)) call prtterr(kw)
```

```
ncase = 49
mfm3 = qi2 + mfm1
call fm_st2m('123456789012345678901234567', mfm4)
call fm_add_r1(mfm4, mfm1)
if (.not.(mfm3 == mfm4)) call prtterr(kw)
```

```
ncase = 50
mfm3 = qi2 + mfm1
call fm_st2m('123456789012345678901234567', mfm4)
call fm_add_r2(mfm1, mfm4)
if (.not.(mfm3 == mfm4)) call prtterr(kw)
```

```
ncase = 51
mim3 = qi2 + mim1
call im_st2m('123456789012345678901234567', mim4)
call im_add(mim4, mim1, mim5)
call im_eq(mim5, mim4)
if (.not.(mim3 == mim4)) call prtterr(kw)
```

```
ncase = 52
mzm3 = qi2 + mzm1
call zm_st2m('123456789012345678901234567', mzm4)
call zm_add(mzm4, mzm1, mzm5)
call zm_eq(mzm5, mzm4)
if (.not.(mzm3 == mzm4)) call prtterr(kw)
```

```
ncase = 53
mfm3 = mfm1 + qi2
call fm_st2m('123456789012345678901234567', mfm4)
call fm_add(mfm1, mfm4, mfm6)
call fm_eq(mfm6, mfm4)
if (.not.(mfm3 == mfm4)) call prtterr(kw)
```

```
ncase = 54
mim3 = mim1 + qi2
call im_st2m('123456789012345678901234567', mim4)
call im_add(mim1, mim4, mim5)
```

```
call im_eq(mim5, mim4)
if (.not.(mim3 == mim4)) call prtterr(kw)

ncase = 55
mzm3 = mzm1 + qi2
call zm_st2m('123456789012345678901234567', mzm4)
call zm_add(mzm1, mzm4, mzm5)
call zm_eq(mzm5, mzm4)
if (.not.(mzm3 == mzm4)) call prtterr(kw)
```

```
end subroutine test8
```

```
subroutine test9
```

```
!           Test the '-' arithmetic operator.
```

```
implicit none
```

```
write (kw, "(/' Testing the derived type - interface.')")
```

```
ncase = 56
mfm3 = qi2 - mfm1
call fm_st2m('123456789012345678901234567', mfm4)
call fm_sub(mfm4, mfm1, mfm6)
call fm_eq(mfm6, mfm4)
if (.not.(mfm3 == mfm4)) call prtterr(kw)
```

```
ncase = 57
mim3 = qi2 - mim1
call im_st2m('123456789012345678901234567', mim4)
call im_sub(mim4, mim1, mim5)
call im_eq(mim5, mim4)
if (.not.(mim3 == mim4)) call prtterr(kw)
```

```
ncase = 58
mzm3 = qi2 - mzm1
call zm_st2m('123456789012345678901234567', mzm4)
call zm_sub(mzm4, mzm1, mzm5)
call zm_eq(mzm5, mzm4)
if (.not.(mzm3 == mzm4)) call prtterr(kw)
```

```
ncase = 59
mfm3 = mfm1 - qi2
call fm_st2m('123456789012345678901234567', mfm4)
call fm_sub(mfm1, mfm4, mfm6)
call fm_eq(mfm6, mfm4)
if (.not.(mfm3 == mfm4)) call prtterr(kw)
```

```
ncase = 60
mim3 = mim1 - qi2
call im_st2m('123456789012345678901234567', mim4)
call im_sub(mim1, mim4, mim5)
call im_eq(mim5, mim4)
if (.not.(mim3 == mim4)) call prtterr(kw)
```

```
ncase = 61
mzm3 = mzm1 - qi2
call zm_st2m('123456789012345678901234567', mzm4)
```

```
call zm_sub(mzm1, mzm4, mzm5)
call zm_eq(mzm5, mzm4)
if (.not.(mzm3 == mzm4)) call prtterr(kw)
```

```
end subroutine test9
```

```
end module test_a
```

```
module test_b
use test_vars
```

```
contains
```

```
subroutine test10
```

```
!           Test the '*' arithmetic operator.
```

```
implicit none
```

```
write (kw, "(/' Testing the derived type * interface.')")
```

```
ncase = 62
mfm3 = qi2 * mfm1
call fm_st2m('123456789012345678901234567', mfm4)
call fm_mpy(mfm4, mfm1, mfm6)
call fm_eq(mfm6, mfm4)
if (.not.(mfm3 == mfm4)) call prtterr(kw)
```

```
ncase = 63
mfm3 = qi2 * mfm1
call fm_st2m('123456789012345678901234567', mfm4)
call fm_mpy_r1(mfm4, mfm1)
if (.not.(mfm3 == mfm4)) call prtterr(kw)
```

```
ncase = 64
mfm3 = qi2 * mfm1
call fm_st2m('123456789012345678901234567', mfm4)
call fm_mpy_r2(mfm1, mfm4)
if (.not.(mfm3 == mfm4)) call prtterr(kw)
```

```
ncase = 65
mfm3 = qi2 * mfm1
mfm4 = mfm1 * to_fm('123456789012345678901234567')
if (.not.(mfm3 == mfm4)) call prtterr(kw)
```

```
ncase = 66
mim3 = qi2 * mim1
call im_st2m('123456789012345678901234567', mim4)
call im_mpy(mim4, mim1, mim5)
call im_eq(mim5, mim4)
if (.not.(mim3 == mim4)) call prtterr(kw)
```

```
ncase = 67
mzm3 = qi2 * mzm1
call zm_st2m('123456789012345678901234567', mzm4)
call zm_mpy(mzm4, mzm1, mzm5)
call zm_eq(mzm5, mzm4)
if (.not.(mzm3 == mzm4)) call prtterr(kw)
```

```
ncase = 68
mfm3 = mfm1 * qi2
call fm_st2m('123456789012345678901234567', mfm4)
call fm_mpy(mfm1, mfm4, mfm6)
call fm_eq(mfm6, mfm4)
if (.not.(mfm3 == mfm4)) call prtterr(kw)
```

```
ncase = 69
mim3 = mim1 * qi2
call im_st2m('123456789012345678901234567', mim4)
call im_mpy(mim1, mim4, mim5)
call im_eq(mim5, mim4)
if (.not.(mim3 == mim4)) call prtterr(kw)
```

```
ncase = 70
mzm3 = mzm1 * qi2
call zm_st2m('123456789012345678901234567', mzm4)
call zm_mpy(mzm1, mzm4, mzm5)
call zm_eq(mzm5, mzm4)
if (.not.(mzm3 == mzm4)) call prtterr(kw)
```

```
end subroutine test10
```

```
subroutine test11
```

```
!           Test the '/' arithmetic operator.
```

```
implicit none
```

```
write (kw, "(/' Testing the derived type / interface.')")
```

```
ncase = 71
mfm3 = qi2 / mfm1
call fm_st2m('123456789012345678901234567', mfm4)
call fm_div(mfm4, mfm1, mfm6)
call fm_eq(mfm6, mfm4)
if (.not.(mfm3 == mfm4)) call prtterr(kw)
```

```
ncase = 72
mfm3 = qi2 / mfm1
call fm_st2m('123456789012345678901234567', mfm4)
call fm_eq(mfm1, mfm6)
call fm_div_r2(mfm4, mfm6)
call fm_eq(mfm6, mfm4)
if (.not.(mfm3 == mfm4)) call prtterr(kw)
```

```
ncase = 73
mim3 = qi2 / mim1
call im_st2m('123456789012345678901234567', mim4)
call im_div(mim4, mim1, mim5)
call im_eq(mim5, mim4)
if (.not.(mim3 == mim4)) call prtterr(kw)
```

```
ncase = 74
mzm3 = qi2 / mzm1
call zm_st2m('123456789012345678901234567', mzm4)
call zm_div(mzm4, mzm1, mzm5)
```

```

call zm_eq(mzm5, mzm4)
if (.not.(mzm3 == mzm4)) call prtterr(kw)

ncase = 75
mfm3 = mfm1 / qi2
call fm_st2m('123456789012345678901234567', mfm4)
call fm_div(mfm1, mfm4, mfm6)
call fm_eq(mfm6, mfm4)
if (.not.(mfm3 == mfm4)) call prtterr(kw)

ncase = 76
mim3 = mim1 / qi2
call im_st2m('123456789012345678901234567', mim4)
call im_div(mim1, mim4, mim5)
call im_eq(mim5, mim4)
if (.not.(mim3 == mim4)) call prtterr(kw)

ncase = 77
mzm3 = mzm1 / qi2
call zm_st2m('123456789012345678901234567', mzm4)
call zm_div(mzm1, mzm4, mzm5)
call zm_eq(mzm5, mzm4)
if (.not.(mzm3 == mzm4)) call prtterr(kw)

```

```
end subroutine test11
```

```
subroutine test12
```

!            Test the '\*\*' arithmetic operator.

```
implicit none
```

```
write (kw, "(/' Testing the derived type ** interface.')")
```

```

ncase = 78
mfm3 = qi2 ** mfm1
call fm_st2m('123456789012345678901234567', mfm4)
call fm_power(mfm4, mfm1, mfm6)
call fm_eq(mfm6, mfm4)
if (.not.(mfm3 == mfm4)) call prtterr(kw)

```

```

ncase = 79
qi4 = 2
mim3 = qi4 ** mim1
call im_st2m('2', mim4)
call im_power(mim4, mim1, mim5)
call im_eq(mim5, mim4)
if (.not.(mim3 == mim4)) call prtterr(kw)

```

```

ncase = 80
qi4 = 23
mzm3 = qi4 ** mzm1
mzm4 = qi4
call zm_power(mzm4, mzm1, mzm5)
call zm_eq(mzm5, mzm4)
if (.not.(mzm3 == mzm4)) call prtterr(kw)

```

```
ncase = 81
```

```

qi4 = 2345
mfm3 = mfm1 ** qi4
mfm4 = qi4
call fm_power(mfm1, mfm4, mfm6)
call fm_eq(mfm6, mfm4)
if (.not.(mfm3 == mfm4)) call prtterr(kw)

```

```

ncase = 82
qi4 = 17
mim3 = mim1 ** qi4
call im_st2m('17', mim4)
call im_power(mim1, mim4, mim5)
call im_eq(mim5, mim4)
if (.not.(mim3 == mim4)) call prtterr(kw)

```

```

ncase = 83
qi4 = 179
mzm3 = mzm1 ** qi4
mzm4 = qi4
call zm_power(mzm1, mzm4, mzm5)
call zm_eq(mzm5, mzm4)
if (.not.(mzm3 == mzm4)) call prtterr(kw)

```

```
end subroutine test12
```

```
subroutine test13
```

```
!           Test functions to_fm, to_im, to_zm, ..., to_quad_int.
```

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

```
write (kw, "(/' Testing the derived type to_fm, ..., to_quad_int interfaces.')")
```

```

ncase = 84
qiv = (/ 123, -432, 567 /)
mfmv1 = to_fm(qiv)
mfmv2 = qiv
do j = 1, 3
    if (.not.(mfmv1(j) == mfmv2(j))) then
        call prtterr(kw)
        exit
    endif
enddo

```

```

ncase = 85
qi3 = 123
mfmv1 = qi3
mfmv2 = 123
do j = 1, 3
    if (.not.(mfmv1(j) == mfmv2(j))) then
        call prtterr(kw)
        exit
    endif
enddo

```

```

ncase = 86
qiv = (/ 123, -432, 567 /)

```

```

mfmv1 = to_fm(qiv)
qiv = to_quad_int(mfmv1)
mfmv2 = qiv
do j = 1, 3
  if (.not.(mfmv1(j) == mfmv2(j))) then
    call prtterr(kw)
    exit
  endif
enddo

ncase = 87
qiv2 = reshape( (/ (11+3*j, j=1, 9) /), shape = (/ 3, 3 /) )
mfma = to_fm(qiv2)
mfmb = qiv2
do j = 1, 3
  do k = 1, 3
    if (.not.(mfma(j, k) == mfmb(j, k))) then
      call prtterr(kw)
      exit
    endif
  enddo
enddo

ncase = 88
qi3 = 1234
mfma = qi3
mfmb = 1234
do j = 1, 3
  do k = 1, 3
    if (.not.(mfma(j, k) == mfmb(j, k))) then
      call prtterr(kw)
      exit
    endif
  enddo
enddo

ncase = 89
qiv2 = reshape( (/ (11+3*j, j=1, 9) /), shape = (/ 3, 3 /) )
mfma = to_fm(qiv2)
qiv2 = to_quad_int(mfma)
mfmb = qiv2
do j = 1, 3
  do k = 1, 3
    if (.not.(mfma(j, k) == mfmb(j, k))) then
      call prtterr(kw)
      exit
    endif
  enddo
enddo

ncase = 90
qiv = (/ 123, -432, 567 /)
mimv1 = to_im(qiv)
mimv2 = qiv
do j = 1, 3
  if (.not.(mimv1(j) == mimv2(j))) then
    call prtterr(kw)
    exit
  endif
enddo

```

```

    endif
enddo

ncase = 91
qi3 = 1234
mimv1 = qi3
mimv2 = 1234
do j = 1, 3
    if (.not.(mimv1(j) == mimv2(j))) then
        call prtterr(kw)
        exit
    endif
enddo

```

```

ncase = 92
qiv = (/ 123, -432, 567 /)
mimv1 = to_im(qiv)
qiv = to_quad_int(mimv1)
mimv2 = qiv
do j = 1, 3
    if (.not.(mimv1(j) == mimv2(j))) then
        call prtterr(kw)
        exit
    endif
enddo

```

```

ncase = 93
qiv2 = reshape( (/ (11+3*j, j=1, 9) /), shape = (/ 3, 3 /) )
mima2 = to_im(qiv2)
mimb2 = qiv2
do j = 1, 3
    do k = 1, 3
        if (.not.(mima2(j, k) == mimb2(j, k))) then
            call prtterr(kw)
            exit
        endif
    enddo
enddo

```

```

ncase = 94
qi3 = 1234
mima2 = qi3
mimb2 = 1234
do j = 1, 3
    do k = 1, 3
        if (.not.(mima2(j, k) == mimb2(j, k))) then
            call prtterr(kw)
            exit
        endif
    enddo
enddo

```

```

ncase = 95
qiv2 = reshape( (/ (11+3*j, j=1, 9) /), shape = (/ 3, 3 /) )
mima2 = to_im(qiv2)
qiv2 = to_quad_int(mima2)
mimb2 = qiv2
do j = 1, 3

```



```

do k = 1, 3
  if (.not.(mima2(j, k) == mimb2(j, k))) then
    call prtterr(kw)
    exit
  endif
enddo
enddo

```

```

ncase = 96
qiv = (/ 123, -432, 567 /)
mzmv1 = to_zm(qiv)
mzmv2 = qiv
do j = 1, 3
  if (abs(mzmv1(j)-mzmv2(j)) /= 0) then
    call prtterr(kw)
    exit
  endif
enddo

```

```

ncase = 97
qi3 = 1234
mzmv1 = qi3
mzmv2 = 1234
do j = 1, 3
  if (abs(mzmv1(j)-mzmv2(j)) /= 0) then
    call prtterr(kw)
    exit
  endif
enddo

```

```

ncase = 98
qiv = (/ 123, -432, 567 /)
mzmv1 = to_zm(qiv)
qiv = to_quad_int(mzmv1)
mzmv2 = qiv
do j = 1, 3
  if (abs(mzmv1(j)-mzmv2(j)) /= 0) then
    call prtterr(kw)
    exit
  endif
enddo

```

```

ncase = 99
qiv2 = reshape( (/ (11+3*j, j=1, 9) /), shape = (/ 3, 3 /) )
mzma2 = to_zm(qiv2)
mzmb2 = qiv2
do j = 1, 3
  do k = 1, 3
    if (.not.(mzma2(j, k) == mzmb2(j, k))) then
      call prtterr(kw)
      exit
    endif
  enddo
enddo
enddo

```

```

ncase = 100
qi3 = 1234
mzma2 = qi3

```

```

mzmb2 = 1234
do j = 1, 3
  do k = 1, 3
    if (.not.(mzma2(j, k) == mzmb2(j, k))) then
      call prtterr(kw)
      exit
    endif
  enddo
enddo

ncase = 101
qiv2 = reshape( (/ (11+3*j, j=1, 9) / ), shape = (/ 3, 3 /) )
mzma2 = to_zm(qiv2)
qiv2 = to_quad_int(mzma2)
mzmb2 = qiv2
do j = 1, 3
  do k = 1, 3
    if (.not.(mzma2(j, k) == mzmb2(j, k))) then
      call prtterr(kw)
      exit
    endif
  enddo
enddo

ncase = 102
call fmm2qi(mfm1%mfm, qi3)
if (to_int(mfm1) /= qi3) call prtterr(kw)

ncase = 103
call imm2qi(mim1%mim, qi3)
if (to_int(mim1) /= qi3) call prtterr(kw)

ncase = 104
call fmm2qi(mzm1%mzm(1), qi3)
if (to_int(mzm1) /= qi3) call prtterr(kw)

end subroutine test13

subroutine test14

```

!            Test the derived-type interface routines that are not used elsewhere in this program.

```

implicit none

ncase = 105
qi4 = mfm1
call fmm2qi(mfm1%mfm, qi5)
if (qi4 /= qi5) call prtterr(kw)

ncase = 106
mim3 = mim1 / 13
qi5 = mim1
qi5 = qi5 / 13
if (.not.(mim3 == qi5)) call prtterr(kw)

ncase = 107
qi4 = mod(mim1, to_im(13))
qi5 = mod(to_quad_int(mim1), 13_quad_int)

```

```

if (qi4 /= qi5) call prtterr(kw)

ncase = 108
qi4 = mim1
call imm2qi(mim1%mim, qi5)
if (qi4 /= qi5) call prtterr(kw)

ncase = 109
qi4 = mzm1
call fmm2qi(mzm1%mzm(1), qi5)
if (qi4 /= qi5) call prtterr(kw)

end subroutine test14

end module test_b

module test_c
use test_vars

contains

subroutine test15

```

! Test type (fm) array equal assignments.

```

implicit none
integer :: j

write (kw, "(/' Testing derived-type array operations.')")

kwsave = kw
call fmsetvar(' kw = 22 ')
call fmsetvar(' ntrace = 0 ')

ncase = 110
mfm3 = to_fm('234.56')
qiv = mfm3
qi5 = 0
do j = 1, 3
    qi5 = qi5 + abs(qiv(j) - 234)
enddo
call fm_st2m(' 1.0e-45 ', mfm4)
if (.not.(to_fm(qi5) <= mfm4)) then
    call prtterr(kwsave)
endif

ncase = 111
qiv = (/ 12, -34, 56 /)
mfmv1 = qiv
mfm3 = 0
do j = 1, 3
    mfm3 = mfm3 + abs(mfmv1(j) - qiv(j))
enddo
call fm_st2m(' 1.0e-45 ', mfm4)
if (.not.(mfm3 <= mfm4)) then
    call prtterr(kwsave)
endif

```

```

ncase = 112
mfmv1 = (/ to_fm('12.1123456789') , to_fm('-34.2123456789') , to_fm('56.3123456789') /)
qiv = mfmv1
mfm3 = 0
do j = 1, 3
    mfm3 = mfm3 + abs(qiv(j) - int(mfmv1(j)))
enddo
call fm_st2m(' 1.0e-45 ', mfm4)
if (.not.(mfm3 <= mfm4)) then
    call prtterr(kwsave)
endif

kw = kwsave
return
end subroutine test15

subroutine test16

```

! Test type (im) array equal assignments.

```

implicit none
integer :: j

kwsave = kw
call fmsetvar(' kw = 22 ')

ncase = 113
mim1 = to_fm('234.56')
qiv = mim1
qi5 = 0
do j = 1, 3
    qi5 = qi5 + abs(qiv(j) - 234)
enddo
call fm_st2m(' 1.0e-45 ', mfm4)
if (.not.(to_fm(qi5) <= mfm4)) then
    call prtterr(kwsave)
endif

ncase = 114
qiv = (/ 12, -34, 56 /)
mimv1 = qiv
mfm3 = 0
do j = 1, 3
    mfm3 = mfm3 + abs(mimv1(j) - qiv(j))
enddo
call fm_st2m(' 1.0e-45 ', mfm4)
if (.not.(mfm3 <= mfm4)) then
    call prtterr(kwsave)
endif

ncase = 115
rv = (/ 12.1, -34.2, 56.3 /)
mimv1 = rv
mfm3 = 0
qiv = rv
do j = 1, 3
    mfm3 = mfm3 + abs(mimv1(j) - qiv(j))
enddo

```