MPI derived types

Purpose is to provide an easy way to send messages that contain data that are non-contiguous in memory

Constructors

- `mpi_type_dup(oldtype, newtype, ierror)`; returns a value of newtype = oldtype; can be useful with building an MPI support library to avoid type confusion
- `mpi_type_contiguous(count, oldtype, newtype, ierror)`; creates newtype that consists of count copies of oldtype

```
integer, parameter :: n=1024, dp=...
double precision :: A(n)
integer :: newtype
!...
!...
call mpi_type_contiguous(n, dp, newtype, ierror)
call mpi_type_commit(newtype)
!...
!...
call mpi_send(A(1), 1, newtype, dest, comm, ierror)
call mpi_send(A(1), n, dp, dest, comm, ierror) ! same
!...
!...
call mpi_type_free(newtype)
```

- `mpi_type_vector(count, blocklength, stride, oldtype, newtype, ierror)`; creates newtype that consists of count copies of oldtype

```
integer, parameter :: m = ?, n = ?, dp = ?
double precision :: A(n, m)
integer :: newtype
!...
!...
call mpi_type_vector(m, 1, n, dp, rowtype, ierror)
! m = # blocks
! 1 = # of elements in each block
! n = stride
call mpi_type_commit(rowtype, ierror)
!...
!...
call mpi_bcast(A(2,1), 1, rowtype, root, comm, ierror)
```