CERES Software Bulletin 95-09

Fortran 90 Reference_Grid Module, August 24, 1995

1.0 Introduction:

This bulletin announces the reference_grid module, a Fortran 90 interface to the CERES Reference Grid as detailed by Richard Green in the CERES Software Bulletin 95-03. This grid system is made up of 1.25 degree quasi-equal-area regions contained within latitudinal zones which number from 1 to 144 from south to north. Seven layers of subgrids are also defined in the bulletin.

The reference_grid module currently deals only with the 1.25 deg grid, though there are plans to add subgrid capabilities. A bulletin update will be issued when that occurs. Subgrid capabilities will be added in such a way as to not impact the interfaces defined in this bulletin.

The reference_grid module is available through the cereslib.a library module. Use of the routines in this module is encouraged in order to minimize future impact on subsystem code when grid characteristics change, as they are expected to do.

Suggested modifications or additions to this module may be submitted for consideration to the e-mail address below.

2.0 Originator:

Joe Stassi (j.c.stassi@larc.nasa.gov)

3.0 Description:

The reference_grid module contains the following subroutines and functions which are publicly available to any Fortran 90 routine which has USE access to the module:

(1) region_number
(2) zone_number
(3) get_centroid
(4) centroid_distance
(5) get_borders
(6) first_region
(7) last_region
(8) region_width
(9) region_width_dp
(10) region_area

(11) number_of_regions

(12) is_polar

(13) number_of_zones

(14) zone_height.

The defining diagram for the module is given in Figure 1. Detailed descriptions of the module subprograms follow the diagram.

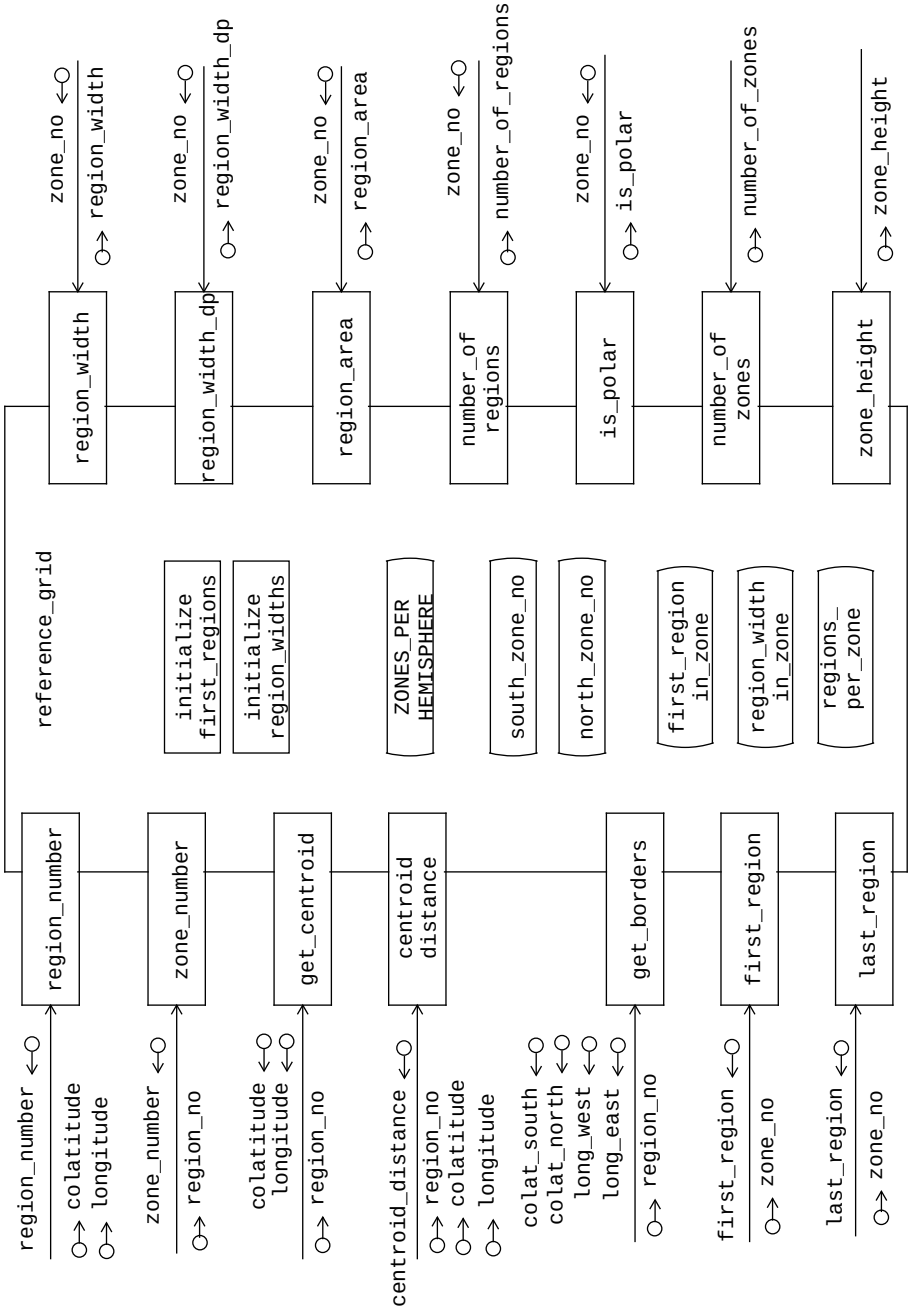

Figure 1. Reference_Grid Module Design, defining diagram

3.1 region_number

3.1.1 Interface

EXAMPLE:

```
USE reference_grid, only : region_number
...
region_no = region_number(colatitude, longitude)
```

where

Routine	Type	Units
region_number	INTEGER (int4) FUNCTION	n/a

and

Argument(s)	I/O	Type	Units	Definition
colatitude	in	real4	degrees	location colatitude
longitude	in	real4	degrees	location longitude

3.1.2 Description

The region_number function returns the reference grid region number for a given colatitude/ longitude location. Reference grid requirements specify that each region contain its southern and western boundaries. Regions in the northern-most zone also contain their northern boundary.

3.1.3 PDL

This function

(1) calculates the zone number of the location using the colatitude, the zone_height function (see 3.14), and the zone-numbering direction (north-to-south or south-to-north, derived from **`north_zone_no`** and **`south_zone_no`**).

(2) calculates the region_number using the longitude, the zone number, and the region_width function (see 3.8).

3.2 zone_number

3.2.1 Interface

EXAMPLE

```
USE reference_grid, only : zone_number
...
zone_no = zone_number(region_no)
```

where

Routine	Type	Units
zone_number	INTEGER (int4) FUNCTION	n/a

and

Argument(s)	I/O	Type	Units	Definition
region_no	in	int4	n/a	reference grid region number

3.2.2 Description

The zone_number function returns the reference grid zone number for a specified region.

3.2.3 PDL

This function uses the **`first_region`** array and the specified region number to calculate the zone number.

3.3 get_centroid

3.3.1 Interface

EXAMPLE:

```
USE reference_grid, only : get_centroid
...
call get_centroid(region_no, colatitude, longitude)
```

where

Routine	Type
get_centroid	SUBROUTINE

and

Argument(s)	I/O	Type	Units	Definition
region_no	in	int4	n/a	reference grid region number
colatitude	out	real4	degrees	centroid colatitude
longitude	out	real4	degrees	centroid longitude

3.3.2 Description

The get_centroid subroutine returns the colatitude/longitude location for the centroid of a specified region.

3.3.3 PDL

This subroutine

(1) determines the zone number for the specified region (see zone_number function, 3.2).

(2) calculates the centroid colatitude using the zone number and the zone-numbering direction (north-to-south or south-to-north) derived from **`north_zone_no`** and **`south_zone_no`**.

(3) calculates the centroid longitude using the zone_number, the region_no, and the first_region function (see 3.6) and region_width function (see 3.8).

3.4 centroid_distance

3.4.1 Interface

EXAMPLE:

```
USE reference_grid, only : centroid_distance
...
distance = centroid_distance(region_no, colatitude, longitude)
```

where

Routine	Type	Units
centroid_distance	REAL (real4) FUNCTION	degrees

and

Argument(s)	I/O	Type	Units	Definition
region_no	in	int4	n/a	reference grid region number
colatitude	in	real4	degrees	location colatitude
longitude	in	real4	degrees	location longitude

3.4.2 Description

The centroid_distance function returns an approximation to the earth central angle distance between a specified colatitude/longitude location and the centroid of a specified region. This function is intended to be used to help identify the “key footprint” within a region (i.e. the footprint closest to the centroid of the region), and should be used only for colatitude/longitude locations in or near the specified region. It will not return accurate measurements for large distances.

3.4.3 PDL

This function

(1) uses the get_centroid subroutine (see 3.3) to find the centroid colatitude/longitude location in the specified region

(2) calculates an approximate angular distance (assuming a “flat surface” region) between the centroid colatitude/longitude and the specified colatitude/longitude location.

3.5 get_borders

3.5.1 Interface

EXAMPLE:

```
USE reference_grid, only : get_borders
...
call get_borders(region_no,                &
  &              colat_north, colat_south, &
  &              long_west, long_east)
```

where

Routine	Type	Units
get_borders	SUBROUTINE	n/a

and

Argument(s)	I/O	Type	Units	Definition
region_no	in	int4	n/a	reference grid region number
colat_north	out	real4	degrees	northern colatitude boundary
colat_south	out	real4	degrees	southern colatitude boundary
long_west	out	real4	degrees	western longitude boundary
long_east	out	real4	degrees	eastern longitude boundary

3.5.2 Description

The get_borders subroutine returns the 4 boundaries of a specified reference grid region.

3.5.3 PDL

This subroutine calls the get_centroid subroutine (see 3.3) and then calculates the 4 boundaries by using the region_width_dp (see 3.9) and the zone_height (see 3.14) functions.

3.6 first_region

3.6.1 Interface

EXAMPLE:

```
USE reference_grid, only : first_region, last_region
...
DO region = first_region(zone_no), last_region(zone_no)
   ...
END DO
```

where

Routine	Type	Units
first_region	INTEGER (int4) FUNCTION	n/a

and

Argument(s)	I/O	Type	Units	Definition
zone_no	in	int4	n/a	reference grid zone number

3.6.2 Description

The first_region function returns the reference grid region number of the first region within a specified latitudinal zone.

3.6.3 PDL

This function accesses and returns the value of the module array **`first_region_in_zone`** for the specified zone number.

3.7 last_region

3.7.1 Interface

EXAMPLE:

```
USE reference_grid, only : first_region, last_region
...
DO region = first_region(zone_no), last_region(zone_no)
   ...
END DO
```

where

Routine	Type	Units
last_region	INTEGER (int4) FUNCTION	n/a

and

Argument(s)	I/O	Type	Units	Definition
zone_no	in	int4	n/a	reference grid zone number

3.7.2 Description

The last_region function returns the reference grid region number of the last region within a specified latitudinal zone.

3.7.3 PDL

This function uses the zone number and the module array **`first_region_in_zone`** to calculate the last region number in the specified zone.

3.8 region_width

3.8.1 Interface

EXAMPLE:

```
USE reference_grid, only : region_width
...
width = region_width(zone_no)
```

where

Routine	Type	Units
region_width	REAL (real4) FUNCTION	degrees

and

Argument(s)	I/O	Type	Units	Definition
zone_no	in	int4	n/a	reference grid zone number

3.8.2 Description

The region_width function returns an angular width of the regions within a specified latitudinal zone. The returned value has 4-byte precision, which is generally the precision which longitude values are stored in. If greater precision is necessary, for instance if region widths are being added to calculate region boundaries, then the 8-byte region_width_dp function should be used instead.

3.8.3 PDL

This function accesses, converts to 4-byte precision, and returns the region width information contained within the module array **`region_width_in_zone.`**

3.9 region_width_dp

3.9.1 Interface

EXAMPLE:

```
USE reference_grid, only : number_of_regions, region_width_dp
REAL (real8) :: longitude_dp
REAL (real4) :: west_longitude
...
longitude_dp = 0.0D0
DO region_no = 1, number_of_regions(zone_no)
   longitude_dp = longitude_dp + region_width_dp(zone_no)
   west_longitude = REAL(longitude_dp, real4)
   ...
END DO
```

where

Routine	Type	Units
region_width_dp	REAL (real8) FUNCTION	degrees

and

Argument(s)	I/O	Type	Units	Definition
zone_no	in	int4	n/a	reference grid zone number

3.9.2 Description

The region_width_dp function is the 8-byte version of the region_width function. This function returns the angular width of the regions within a specified latitudinal zone. It should be used particularly in cases where the region width is being summed to calculate the longitude boundary of the regions within a zone.

3.9.3 PDL

This function accesses and returns the region width information contained within the module array **`region_width_in_zone`**.

3.10 region_area

3.10.1 Interface

EXAMPLE:

```
USE reference_grid, only : region_area
...
area = region_area(zone_no)
```

where

Routine	Type	Units
region_area	REAL (real4) FUNCTION	km^2

and

Argument(s)	I/O	Type	Units	Definition
zone_no	in	int4	n/a	reference grid zone number

3.10.2 Description

The region_area function returns the approximate area of regions within a specified latitudinal zone.

3.10.3 PDL

This function calculates the region area by multiplying the region_width (see 3.8) by the zone_height (see 3.14) and then multiplying the product by the factors, km_per_degrees_latitude and km_per_degrees_longitude. The km_per_degrees_longitude factor is adjusted to the center colatitude of the zone.

3.11 number_of_regions

3.11.1 Interface

EXAMPLE:

```
USE reference_grid, only : number_of_regions
...

regions_in_zone = number_of_regions(zone_no)
```

or

```
regions_on_globe = number_of_regions()
```

where

Routine	Type	Units
number_of_regions	INTEGER (int4) FUNCTION	n/a

and

Argument(s)	I/O	Type	Units	Definition
* zone_no	in	int4	n/a	reference grid zone number

* optional parameter

3.11.2 Description

The number_of_regions function has two options.

(1) If a zone number is supplied as an input parameter, then the number_of_regions function returns the number of regions within the specified latitudinal zone.

(2) If the function is called without any parameters, then the number_of_regions function returns the number of regions contained within the entire reference grid (i.e. the globe).

3.11.3 PDL

(1) If the zone_no parameter is present, then this function accesses and returns the information contained within the module array **`regions_per_zone`**.

(2) If no parameters are present, then the function uses the **`regions_per_zone`** array and the number_of_zones function (see 3.13) to calculate the total number of regions within the reference grid.

3.12 is_polar

3.12.1 Interface

EXAMPLE:

```
USE reference_grid, only : number_of_zones, is_polar
...
DO zone_no = 1, number_of_zones()
   IF (is_polar(zone_no)) THEN
      ...
   END IF
END DO
```

where

Routine	Type	Units
is_polar	LOGICAL FUNCTION	n/a

and

Argument(s)	I/O	Type	Units	Definition
zone_no	in	int4	n/a	reference grid zone number

3.12.2 Description

The is_polar function returns a .true. value if the specified latitudinal zone is located within either of the two polar circles.

3.12.3 PDL

This function calculates the center colatitude of the zone. If it is located above the Arctic Circle (66.45 deg latitude) or below the Antarctic Circle (-66.45 deg latitude), then the zone is considered a polar zone, and the function returns a .true. value.

3.13 number_of_zones

3.13.1 Interface

EXAMPLE:

```
USE reference_grid, only : number_of_zones
...
DO zone = 1, number_of_zones()
   ...
END DO
```

where

Routine	Type	Units
number_of_zones	INTEGER (int4) FUNCTION	n/a

There are no calling parameters.

3.13.2 Description

The number_of_zones function returns the number of latitudinal zones contained within the reference grid.

3.13.3 PDL

This function returns the maximum value of **`north_zone_no`** and **`south_zone_no`**.

3.14 zone_height

3.14.1 Interface

EXAMPLE:

```
USE reference_grid, only : zone_height
...
height = zone_height()
```

where

Routine	Type	Units
zone_height	REAL (real4) FUNCTION	degrees

There are no calling parameters.

3.14.2 Description

The zone_height function returns the angular height of the latitudinal zones in the reference grid.

3.14.3 PDL

This function calculates zone height by dividing 180.0 degrees by number_of_zones (see 3.13).