

## Intro

여러 지점들에 걸쳐 산재해있는 지점별 데이터에 대하여 격자화 내삽을 적용하여 규칙 격자 데이터로 변환하는 작업이 필요할 경우들이 종종 있습니다. IDL에서도 이러한 작업을 할 수 있게 해주는 GRIDDATA, SPH\_SCAT, KRIG2D 등의 내장함수들이 지원됩니다. 그런데 지구 구체의 표면에 위치한 지점들인 경우, **내삽의 과정에서 구면의 특성이 반영되도록** 하는 것이 필요할 수도 있습니다. 그래서 오늘은 **GRIDDATA 함수**를 이용하여 구면 상의 지점 데이터를 격자화하는데 있어서 구면의 특성이 반영되도록 하는 방법에 대하여 알아보도록 하겠습니다.

## 예제 데이터의 생성

예제 데이터는 글로벌 맵(Global Map) 즉 세계 전체 맵 상에 분포하는 300개의 지점 데이터로 가정하여 생성합니다. 이 때 RANDOMU 함수를 사용하여 난수로 발생시키는 방식을 사용하여 다음과 같이 data, lons, lats 3종의 배열들을 생성합니다.

```
n = 300
data = RANDOMU(-1, n, /NORMAL)*12+50
lons = RANDOMU(-3, n)*360-180
lats = RANDOMU(-5, n)*180-90
HELP, lons, lats, data
PRINT, MIN(data), MAX(data)
```

여기서 생성된 data, lons, lats 배열들은 각각 300개의 값들로 구성됩니다. 즉 전 세계에 걸쳐 분포하는 300개의 지점들에 대한 데이터로 가정한 것입니다. 지점들이 전 세계에 분포하도록 하기 위하여 lons 배열의 값들은 -180~+180 범위의 경도가 되도록 하였고 lats 배열의 값들은 -90~+90의 범위의 위도가 되도록 하였습니다. 그리고 지점별 데이터인 data 배열의 값들은 대략 0~100의 범위 내에 있도록 하였습니다.

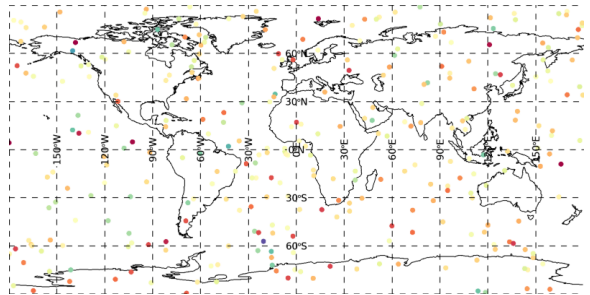
## 지점별 데이터 표출

그러면 먼저 이 지점별 데이터를 있는 그대로 글로벌 맵 상에 표출해봅시다. 다음과 같이 MAP 함수 및 MAPCONTINENTS 함수를 사용하여 글로벌 맵을 표출한 다음, **SCATTERPLOT 함수**를 사용하여 지점별 데이터를 원형 기호로 맵 상에 중첩하여 표출합니다.

```
win1 = WINDOW(DIMENSION=[800, 400], $
/NO_TOOLBAR)
m1 = MAP('geographic', $
LIMIT=[-90, -180, 90, 179.99], $
MARGIN=0, font_size=10, /CURRENT)
m1.MapGrid.GRID_LONGITUDE = 30
m1.MapGrid.GRID_LATITUDE = 30
```

```
m1.MapGrid.LINESTYLE = 2
mcl = MAPCONTINENTS()
ct = COLORTABLE(74, /REVERSE)
spl = SCATTERPLOT(lons, lats, $
RGB_TABLE=ct, $
MAGNITUDE=BYTSCAL(data), $
SYMBOL='circle', /SYM_FILLED, $
/OVERPLOT)
```

여기서는 세계 전체 지도가 표출되도록 하기 위하여 경도 및 위도의 범위를 그에 맞게 설정하였습니다. 그리고 각 지점별로 데이터 값의 크기에 따라 색상이 차등적으로 반영되도록 하기 위하여 SCATTERPLOT 함수에서 RGB\_TABLE 및 MAGNITUDE 속성을 사용하였습니다. 이렇게 표출된 그림은 다음과 같습니다



## 격자화 및 표출 [1]

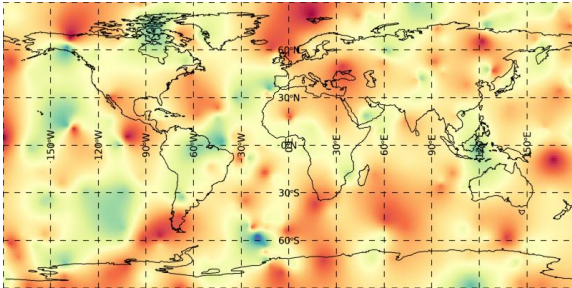
이제는 이와 같이 지점별로 불규칙하게 분포하는 데이터를 규칙 격자 데이터로 변환해봅시다. 여기서는 앞서 언급했듯이 **GRIDDATA 함수**를 사용하고 세부적으로는 **Kriging 기법**이 사용되도록 할 것입니다. 이 과정은 다음과 같습니다.

```
data_gridded = GRIDDATA(lons, lats, $
data, DIMENSION=[360, 180], $
/KRIGING)
```

여기서는 lons, lats, data 배열로 존재하는 지점별 데이터를 GRIDDATA 함수에 투입하여 Kriging 기법을 적용하여 360x180의 격자 구조를 갖는 2차원 데이터로 변환하여 그 결과를 data\_gridded라는 배열로 얻은 것입니다. 이제 data\_gridded 배열을 글로벌 맵 상에 중첩하여 표출해봅시다. 이를 위하여 별도의 그래픽 창을 띄워서 앞서와 동일한 방식으로 글로벌 맵을 표출하되 이번에는 data\_gridded 배열을 이미지의 형태로 중첩하여 표출하면 됩니다. 그 과정 및 표출 그림은 다음과 같습니다.

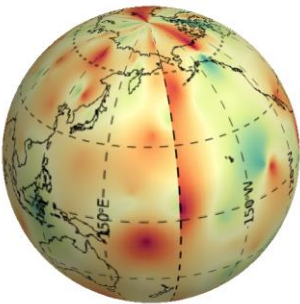
```
win2 = WINDOW(DIMENSION=[800, 400], $
/NO_TOOLBAR)
m2 = MAP('geographic', $
LIMIT=[-90, -180, 90, 179.99], $
MARGIN=0, FONT_SIZE=10, /CURRENT)
```

```
m2.MapGrid.GRID_LONGITUDE = 30
m2.MapGrid.GRID_LATITUDE = 30
m2.MapGrid.LINESTYLE = 2
i = IMAGE(data_gridded, $
  RGB_TABLE=ct, $
  IMAGE_LOCATION=[-180, -90], $
  IMAGE_DIMENSION=[360, 180], $
  /OVERPLOT)
mc2 = MAPCONTINENTS()
```



그러나 이 결과에는 **심각한 문제가 존재합니다**. 그림에서 경도 +180인 부분과 -180인 부분은 구체(Sphere) 형태의 지구 표면 상에서는 서로 동일한 자오선이기 때문에 실제로도 구면 상에서 서로 이어집니다. 그런데 중첩된 격자 이미지를 보면 그 형태가 **불연속적**입니다. 다음과 같은 과정에 의하여 위의 그림을 구체 폴리곤 상에 입혀서 표출해보면 확실하게 확인할 수 있습니다.

```
img = win2.CopyWindow(WIDTH=800)
MESH_OBJ, 4, vertices, polygons, $
  REPLICATE(0.5, 101, 101)
oImage = OBJ_NEW('IDLGrImage', img)
vector = FINDGEN(101)/100.
texture_coordinates = $
  FLTARR(2, 101, 101)
texture_coordinates[0, *, *] = $
  vector # REPLICATE(1., 101)
texture_coordinates[1, *, *] = $
  REPLICATE(1., 101) # vector
oSphere = OBJ_NEW('IDLGrPolygon', $
  SHADING=1, DATA=vertices, $
  POLYGONS=polygons, $
  COLOR=[255, 255, 255], $
  TEXTURE_COORD=texture_coordinates, $
  TEXTURE_MAP=oImage, /TEXTURE_INTERP)
XOBJVIEW, oSphere, XSIZE=600, $
  YSIZE=600, SCALE=0.8
```

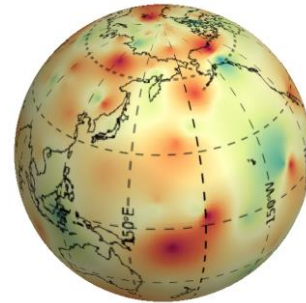
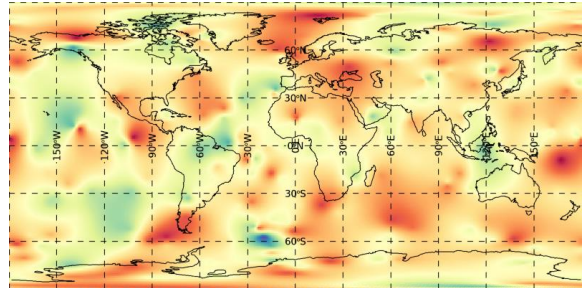


### 격자화 및 표출 [2]

이러한 문제는 결국 격자화 내삽의 과정에서 경도 +180과 -180이 서로 동일한 지점이라는 **구체(Sphere) 표면의 특수성이 전혀 반영되지 않았기 때문에 발생한 것**입니다. 사실 데이터의 지점들이 국지적인 영역(예 : 한반도 영역) 내에 제한적으로 분포할 경우에는 이러한 문제를 굳이 신경쓰지 않아도 되지만, 글로벌 맵 전체에 분포하는 지점들에 대한 데이터를 다룰 경우에는 분명히 큰 문제가 됩니다. 물론 다행히도 해결책은 존재합니다. 앞서 **GRIDDATA** 함수가 사용된 내용에서 다음과 같이 **/SPHERE 및 /DEGREES** 두 속성을 추가적으로 사용하는 것입니다

```
data_gridded = GRIDDATA(lons, lats, $
  data, DIMENSION=[360, 180], $
  /KRIGING, /DEGREES, /SPHERE)
```

이러한 변경 사항만 반영하여 모든 과정을 다시 실행해보면 결과는 다음 그림들과 같습니다.



이 결과를 보면 경도 +180 및 -180인 부분에서 발생하던 이미지 패턴의 불연속 현상 뿐 아니라 북극 지역에서 보이던 이상한 패턴들도 완전히 사라지고 **구체 표면이라는 특성이 잘 반영된 것**을 확인할 수 있습니다. 여기서 사용된 /DEGREES 및 /SPHERES 속성은 GRIDDATA 함수에서 Kriging 외에 다른 세부 기법을 적용할 경우에도 얼마든지 사용이 가능합니다. 따라서 글로벌 맵 상에 분포한 지점 데이터를 규칙 격자 데이터로 변환해야 하는 경우에는 이러한 방법을 사용하는 것이 좋습니다. 그리고 글로벌 맵의 규모까지는 아니더라도 지점들의 분포 영역이 상당히 넓을 경우, 지점들의 분포 위치가 주로 고위도 지역일 경우, 그리고 지점들의 분포 위치가 경도 +180(즉 -180)인 부분과 가까운 경우에도 이 방법을 적용하는 것이 필요할 수도 있습니다.