

## Intro

바람이나 해류와 같이 유체의 흐름을 나타내는 **벡터장(Vector Field)** 데이터가 있을 때, IDL에서는 이러한 데이터를 벡터들의 격자 배치 형태로 표출할 수 있습니다. 그리고 더 나아가서는 이러한 벡터장 내에서 특정 위치에 있던 입자(Particle)가 유체의 흐름을 타고 이동하게 되는 궤적을 계산하는 **입자 추적(Particle Tracing)**이라는 작업도 가능합니다. IDL에서는 벡터장의 표출 및 입자 추적을 위하여 **VECTOR 함수 및 PARTICLE\_TRACE 프로시저**를 사용하면 됩니다. 오늘은 벡터장의 표출부터 입자 추적으로 이어지는 일련의 작업에 대한 예제를 소개합니다.

## 예제 데이터 준비

예제로 사용할 데이터는 IDL의 설치와 함께 제공되는 globalwinds.dat라는 파일입니다. 이 파일은 .sav 형식의 파일이기 때문에 다음과 같이 RESTORE 명령을 사용하여 내부 정보를 추출하면 됩니다.

```
file = FILEPATH('globalwinds.dat', $
    SUBDIR=['examples', 'data'])
RESTORE, file
HELP, u, v, x, y
```

그러면 파일 내부에 저장되어 있던 u, v, x, y 4개의 배열들을 끄집어내게 됩니다. 실제로 이 배열들은 128x64의 격자구조를 갖는 글로벌 바람 데이터입니다. 그 중 u, v는 각 격자별 u, v 성분값들입니다. 그리고 x, y는 격자들의 X방향 및 Y방향 좌표값들입니다. HELP 명령에 의하여 출력된 내용으로 확인할 수 있습니다.

U	FLOAT	= Array[128, 64]
V	FLOAT	= Array[128, 64]
X	FLOAT	= Array[128]
Y	FLOAT	= Array[64]

## 지도 및 벡터장의 표출

먼저 예제 데이터를 벡터장(Vector Field)의 형태로 표출해봅시다. 이 데이터는 원래 글로벌 바람장(Wind Field) 자료이며, 격자들이 경도 및 위도 전 범위에 걸쳐 분포합니다. 따라서 바람 지도를 먼저 표출하고 그 위에 벡터장을 중첩하는 방식으로 구현해봅시다. 먼저 바람 지도를 표출하는 과정은 다음과 같습니다.

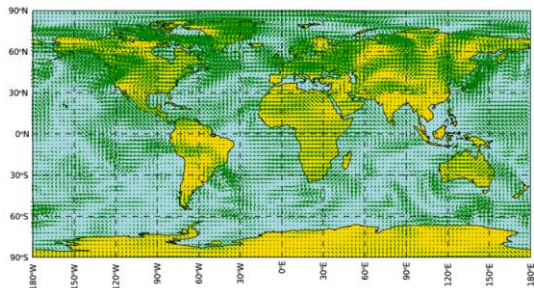
```
win = WINDOW(DIMENSIONS=[800, 480], $
    /NO_TOOLBAR)
m = MAP('Geographic', $
    FILL_COLOR='light blue', $
    MARGIN=0.05, /CURRENT)
```

```
mc = MAPCONTINENTS(FILL_COLOR='gold')
m.MapGrid.LABEL_POSITION = 0
m.MapGrid.GRID_LONGITUDE = 30
m.MapGrid.LINESTYLE = 2
m.MapGrid.HORIZON_LINESTYLE = 0
```

여기서는 **MAP 함수와 MAPCONTINENTS 함수**를 이용하여 바탕 지도를 글로벌 맵의 형태로 표출하였습니다. 이 지도의 경도 범위는 -180~+180도이고 위도 범위는 -90~+90도입니다. 그 외에도 자잘한 세부 속성들에 대한 설정도 포함되었습니다. 그리고 바로 이어서 벡터 필드를 지도 위에 중첩하여 표출하려면 다음과 같이 **VECTOR 함수**를 이용하면 됩니다.

```
vec = VECTOR(u, v, x, y, $
    AXIS_STYLE=2, ARROW_THICK=3, $
    HEAD_SIZE=0.5, LENGTH_SCALE=0.5, $
    /HEAD_PROPORTIONAL, $
    COLOR='green', /OVERPLOT)
```

VECTOR 함수에는 이와 같이 u, v, x, y 배열들이 필수 인수로 사용됩니다. 여기서 u는 각 격자별 U 성분값들로 구성된 2차원 배열이고, v는 각 격자별 V 성분값들로 구성된 2차원 배열이어야 합니다. 그리고 x는 X 방향의 격자 좌표값들로 구성된 1차원 배열이고, y는 Y 방향 격자 좌표값들로 구성된 1차원 배열이어야 합니다. 그 외에도 벡터들을 화살표 형태로 표출하는데 있어서 필요한 세부 속성들에 대한 설정도 포함되었습니다. 이렇게 표출된 그림은 다음과 같습니다.



이 그림을 보면 바람장을 구성하는 벡터들이 격자 형태로 중첩 표출된 것을 볼 수 있습니다. 물론 많은 벡터들이 오밀조밀 붙어있으니 개별 벡터에 대한 가독성은 다소 떨어지는 편인데, 이후의 과정에서 경위도 범위를 더 제한하게 되면 벡터들이 더 잘 보일 것입니다.

\* VECTOR 함수의 사용법에 관한 더 자세한 내용은 아래 링크들을 통하여 보실 수 있습니다.

<http://blog.daum.net/swrush/437>

<http://blog.daum.net/swrush/438>

## 입자 추적 결과의 산출

입자 추적을 위해서는 **PARTICLE\_TRACE** 프로시저를 사용하게 됩니다. 그래서 이 프로시저에 투입될 인수들을 정의해야 합니다. 우선 입자의 출발 위치 정보를 생성해야 하는데, 다음과 같이 배열 인덱스로 정의하면 됩니다.

```
x_ind0 = 92
y_ind0 = 32
seed = [x_ind0, y_ind0]
PRINT, 'start (lon/lat) :', $
      x[x_ind0], y[y_ind0]
```

여기서는 [92, 32]라는 XY 인덱스 값들로 구성된 배열을 seed라는 이름으로 생성하였습니다. 실제 좌표값은 x[92], y[32]가 된다는 것을 유의해야 합니다. 실제로 출력된 출발점의 경위도 좌표는 다음과 같이 동경 78.75도, 북위 1.0도입니다.

```
start (lon/lat) :   78.7500   1.00000
```

그리고 **PARTICLE\_TRACE** 프로시저에 투입될 벡터 필드 데이터는 다음과 같이 정의되어야 합니다.

```
data = FLTARR(2, 128, 64)
data[0, *, *] = u
data[1, *, *] = v
```

이제 모든 준비는 끝났으므로 **PARTICLE\_TRACE** 프로시저를 다음과 같이 사용하면 됩니다.

```
PARTICLE_TRACE, data, seed, $
      verts, conn, MAX_ITERATIONS=30
HELP, verts
```

**MAX\_ITERATIONS** 키워드는 입자의 이동 궤적을 몇 회까지 계산할 것인가를 의미합니다. 여기서는 30으로 설정하였습니다. 그리고 최종적인 계산 결과는 **verts**와 **conn**이라는 배열들로 얻어지는데, 실제로 필요한 것은 **verts**입니다. **HELP**에 의하여 출력된 내용을 보면 다음과 같이 2x30의 구조를 갖는 2차원 배열로 얻어집니다.

```
VERTS          FLOAT          = Array[2, 30]
```

이렇게 얻어진 **verts**는 입자의 이동 궤적에 대한 30개의 X, Y 좌표 인덱스들로 구성됩니다. 그래서 다음과 같이 X, Y 각각에 대한 인덱스 배열로 분리하는 것이 좋습니다.

```
x_inds = REFORM(verts[0, *])
y_inds = REFORM(verts[1, *])
PRINT, 'end (lon/lat) :', $
      x[x_inds[-1]], y[y_inds[-1]]
```

그리고 입자 이동 궤적의 최종점의 경위도 좌표값들도 출력하게 했는데, 그 내용을 보면 다음과 서경 5.6도, 남위 4.6도가 됩니다.

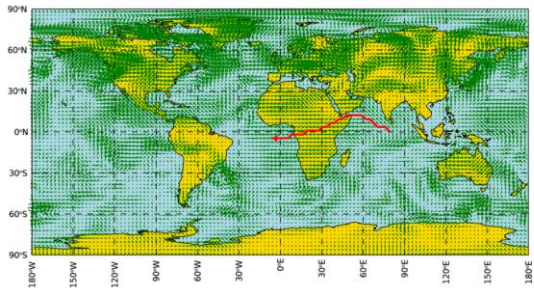
```
end (lon/lat) :   -5.62500   -4.62500
```

## 입자 추적 결과의 표출

이제 입자 추적의 결과인 입자의 이동 궤적을 앞서 표출했던 지도 상에 중첩하여 표출해봅시다. 이를 위하여 다음과 같이 **POLYLINE** 및 **ARROW** 함수를 사용하였습니다.

```
p = POLYLINE(x[x_inds], y[y_inds], $
      THICK=3, COLOR='red', /DATA)
ar = ARROW(x[x_inds[-2:-1]], $
      y[y_inds[-2:-1]], THICK=3, $
      HEAD_SIZE=0.5, COLOR='red', /DATA)
```

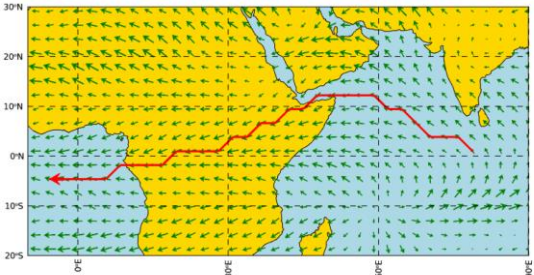
여기서는 **POLYLINE** 함수를 사용하여 이동 궤적 전체를 선으로 연결하여 표출하고, **ARROW** 함수를 사용하여 궤적의 마지막 부분에 화살표 표시를 한 것입니다. 표출 결과는 다음과 같습니다.



그러면 이번에는 이동 궤적이 위치한 영역을 좀 더 확대하여 표출해봅시다. 이를 위해서는 앞서 바탕 지도의 표출을 위하여 사용했던 **MAP** 함수의 내용만 다음과 같이 수정하면 됩니다.

```
m = MAP('Geographic', $
      LIMIT=[-20, -10, 30, 90], $
      FILL_COLOR='light blue', $
      MARGIN=0.05, /CURRENT)
```

즉 이와 같이 **LIMIT** 키워드를 사용하여 지도의 경위도 영역을 제한하면 됩니다. 이렇게 하여 다시 표출해보면 그 모습은 다음과 같습니다.



이와 같이 지도의 경위도 범위를 국지 영역으로 제한하면 바람장의 벡터들과 입자 이동 궤적의 모습이 더욱 잘 보입니다. 따라서 위와 같은 방법을 사용하면 벡터장 내의 특정한 위치에서 출발한 입자가 유체의 흐름을 타고 이동하게 될 경로를 계산하고 그 궤적을 도식적으로 나타낼 수 있습니다. 오늘 소개한 과정 전반에 대한 더 자세한 내용은 아래 링크에서 보실 수 있습니다.

<http://blog.daum.net/swrush/575>