Ⅳ. 점 데이터의 처리

1. 점 데이터 처리 기초

환경에 대한 연구에 있어서 지형도에서 추출한 점고(點高, spot height)나 시추 데이터, 기상관측점의 강우 정보 등 단순히 특정 위치나 장소에 대한 정보만을 나타낼 경우가 많다. GRASS에서는 이러한 정보를 분석하고 변환시킬 수 있는 모듈을 제공하고 있다. GRASS에서는 점을 'site'라 칭한다. 벡터(vector)형식으로 저장되어 있는 벡터형 점 데이터와는 달리 site는 Grass만의 독자적인 형식을 취하고 있어 내부적으로 GRASS데이터 베이스에 ASCII 형식의 표가 만들어지는 특징이 있다.

1.1 site(점 데이터) 입력(Import)과 변환(export)

입력과 변환은 ASCII형식을 통해서만 가능한데, 이 형식은 대부분의 스프레드시트(Spread sheet)에서 이용 가능한 가장 일반적인 것이다. 점 데이터 파일을 만들기 위해서는 스프레드시트 프로그램에서 다음과 같은 ASCII 구조(desc.는 옵션이며, 머리글은 필요 없다)로 전환하면 된다.

일반적 형식:

```
X-value1 Y-value1 Z-value1 desc1
X-value2 Y-value2 Z-value2 desc1
X-value3 Y-value3 Z-value3 desc1
.....
```

예:

```
3570321 5776988 102.4 Adyer
3571987.3 5776876 110 Airport
3574987 5777987.5 132.2 TTKRoad
```

점 데이터가 스프레드시트 프로그램에 저장되어 있지 않은 경우, 문서 편집기를 통해서도 이와 같은 ASCII 파일을 만들 수 있다. 앞부분의 두 가지 필드(x와 y)는 좌표를 나타내는 것이고, 세 번째 필드는 Z값으로 표고 값 등을 나타내는 것이다. 여기에 단위를 넣어서는 않된다. x값은 동경을 나타내고 y값은 북위를 나타내는 것이다. 각 필드 간의 분리자로는 '스페이스 키', '탭 키', 특수문자가 이용 가능한데, 기본 분리자는 '스페이스 키'이다. 서술문(Description, 네 번째 필드)은 빈칸이 없는 한 어떠한 문자(빈칸은 밑줄로 대신해야함)라도 가능하지만, 없어도 실행하는데 문제는 없다. 이와 같은 서술문을 GRASS에서는 Label이라고 한다.

입력(Import)

점 데이터 입력 모듈은 다음과 같이 시작된다.

```
$s.in.ascii
* 용법 : s.in.ascii sites=name [input=name] [fs=character|space|tab]
```

다른 것과 마찬가지로 이 모듈도 대화식으로 수행 가능하며, 필요한 정보를 매개변수로 입력하여 수행할 수도 있다. 입력될 ASCII파일을 현재의 GRASS 작업 디렉토리에 저장하는 것이 가장 바람직하다. 우선 GRASS데이터베이스 내에 저장될 파일의 명칭 (Sites 이름)을

입력해야 하며, 다음으로 입력될 ASCII파일(스프래드시트 등에서 변환한 파일)의 이름을 입력해야 한다. 마지막으로 필드분리자(fs)를 지정하면 자료는 입력된다. DOS의 편집기를 이용하여 원 파일을 만들었을 경우에는 DOS-ASCII파일을 미리 UNIX-ASCII로 변환해야 한다. 변환을 위해서 다음 프로그램을 이용하면 된다 :

```
$dos2unix file.asc
```

이 프로그램을 이용할 수 없을 경우에는 프로그램의 이름이 다를 수 있기 때문에 시스템 관리자에게 문의하기 바란다.

GRASS 데이터베이스 내에 이미 만들어져 있는 파일의 목록은 다음 명령을 사용하여 볼 수 있다.

```
$g.list.sites
```

이 명령은 현재의 mapset내에 존재하는 모든 점 데이터 목록을 열거한다.

변환(Export)

변환모듈(export)도 입력(import)과 비슷하게 작동한다. 그러나 출력파일을 지정하지 않으면 스크린에만 출력되기 때문에 변환된 결과를 저장할 파일명을 기록해야 한다는 것을 주의해야 한다.

```
$s.out.ascii>file.asc
```

* 용법 : s.out.ascii sites=*name*>출력파일

```
s.out.ascii sites=name|d.points color=red size=10 type=diamond
```

← 다른 프로그램으로 변환 결과를 보낼 경우의 용법

점 데이터 이름과 필드 분리자를 정의하고 나면, Location내의 모든 sites(점)을 변환할 것인지 mapset내의 sites(점) 변환할 것인지를 결정하게 된다. 마지막 질문은 sites의 서술문(description, Label)의 변환에 관한 사항이다.

변환 결과는 작업중이 디렉토리에 파일형태로 저장된다.

1.2 Sites의 출력- 디스플레이(display) 명령

작업결과를 화면에 출력하기 위해서는 먼저 "GRASS monitor"라는 그래픽 창(graphic window)을 구동시켜야 한다. "x0"에서 "x7"까지 총 8개의 모니터(monitor)를 이용할 수 있다.[1)]

칼라 테이블(color table)이 충돌을 일으킬 수 있기 때문에, 그래픽 창을 구동시키기 전에 웹브라우저(netscape 등)나 이미지 변환프로그램(xv 등)이 수행되고 있어서는 안 된다. 이들 프로그램은 GRASS Monitor를 구동시킨 후에 실행해야 한다.

모듈의 이용법은 다음과 같다.

```
$d.mon                                  ← 대화식 명령
$d.mon start=x0 또는 간단히 $d.mon x0   ← 일괄 명령
$d.mon stop=x0                          ← 그래픽 창 닫기 명령
```

---

1) d.mon 명령를 수행해도 그래픽 창이 나타나지 않을 경우는 대부분 Xdriver가 잘 못 맞추어져 있기 때문이다. 예를 들어 현재 24bit 그래픽을 사용하고 있는 경우, /usr/local/grass42/driver/xdriver가 xdriver_24bit와 link되어 있어야 한다.

점 데이터(sites)는 라벨(각 점에 대한 문자 표시)로 표현될 수도 있으며, 라벨 없이 표현될 수도 있다.

$d.sites ← 각 점(site)은 “X”로 표시된다.

$d.site.labels ← 각 점(site)은 "X"표시 없이 라벨이 표시된다.

매개변수로 점 데이터(site)의 파일명만 입력하면, "x"기호와 라벨이 동시에 표현된다.(?)

스크린에 표현된 내용을 화면에서 지우기 위해서는 다음 명령을 사용한다.

$d.erase

현재 작업중인 Region을 수정한 후 이 명령(d.erase)을 수행하지 않으면, 그래픽 창의 좌표가 틀려지기 때문에, 아주 중요한 명령이다.

1.3 그래픽 창(Grass Monitor) 사용 시 주의 사항

그래픽 창(GRASS Monitor)은 다소 민감하기 때문에, 첫째 마우스로 창의 크기를 변경하지 않도록 주의해야 하며, 작업종료 후에는 “**d.mon stop=x0**”명령어로 그래픽 창을 필히 닫아야 한다.

유닉스 윈도우 메뉴에서 창을 닫게 되면, GRASS는 그래픽 창이 종료되었다는 것을 인식하지 못한다. 이 경우 그래픽 창은 이용자가 종료시키기 전까지는 계속 작업대기상태로 남아 있게 된다.

실수로 윈도우의 크기를 마우스로 조절하였을 경우에는 d.erase 명령으로 그래픽 창을 깨끗이 하거나, 그래픽 창을 종료 한 후 재실행함으로써 문제를 해결할 수 있다.

그래픽창을 닫지 않고 GRASS를 끝낸 경우 GRASS를 다시 시작하고 다음 명령으로 대기중인 그래픽 창을 선택하여 종료시킬 수 있다.

$d.mon select=x0

유닉스 시스템의 모든 사용자는 하나 이상의 그래픽 창(GRASS Monitor)을 사용할 수 있다. 위의 명령으로 그래픽 창 중 하나를 활성화시킬 수 있다. 그러나 동시에 여러 개의 그래픽 창에서 작업할 수는 없다.

서로 다른 투영법을 사용하고 있는 또 다른 Location으로 작업을 전환하기 위해서는 GRASS를 종료하기 전에 그래픽 창(GRASS Monitor)을 필히 종료해야 한다.

2. DTM 분석

새로 입력된 점 데이타는 곧 바로 분석에 이용될 수 있다. 우선 점고(Spot Elevation)와 같은 표고 데이터를 이용한 분석에 대해서 설명하겠고, 다음에 수리학적 데이터 분석에 대해서 설명하겠다. 마지막으로 서로 다른 데이터 포맷으로 전환하는 과정에 대해서 설명하겠다.

2.1 지형 분석

점의 형태로 입력될 ASCII 파일의 구조는 일반적으로 다음과 같다. 즉, 다음과 같은 구조로 점의 정보를 담은 ASCII파일은 GRASS에서 불러들일 수 있으며, 분석에 이용될 수 있다. 또 다른 말로 하면, 다음과 같은 형식의 ASCII파일이 GRASS에서 이용되는 일반적인 포맷이다.

Easting	Northing	HeightValue
Easting	Northing	HeightValue
Easting	Northing	HeightValue
동경값	북위값	높이값
X좌표	Y좌표	Z좌표
.....		

부동소수점이 아닌(non-floatingpoint) 값만 z값으로 이용될 수 있다는 GRASS의 Raster 데이터 포맷이 가지는 한계 때문에, 높이값(z값)을 정수(Whole Number)로 변환해야 한다. 이러한 문제는 높이값에 10을 곱하는 방법으로 해결하고 있는데, 이 과정은 나중에 raster데이타 포맷으로 변환해야 할 필요가 있을 때 상당히 중요한 의미를 가진다.

Unix에서는 데이터베이스의 특정 칼럼(통상 세 번째 칼럼)의 숫자에 10, 100 또는 특정 수를 곱하는 간단한 방법을 제공하고 있다. 방법은 다음과 같다.

```
$ awk '{print $1, $2, $3*10}' file.asc > newfile.asc
```

첫 번째와 두 번째 칼럼 값인 $1, $2에는 변화가 없고, 세 번째 칼럼인 $3에는 10이라는 수자가 곱해졌으며, 이들 값 모두는 새로운 파일에 기록된다. “awk" 유닉스 표준 명령이다. 이 과정에서 남게되는 소수(decimals)는 raster로 변환되는 과정에서 반올림될 것이다.

입력완료된 점고(Spot Height)는 여러 가지 지형 분석에 사용될 수 있다.

- 경사, 향
- 지형 단면 곡선
- 등고선 분석
- 평균 고도 곡선

기존의 raster를 이용하여 분석대상지(관심있는 지역)과 제외지역(비관심지역)을 구분할 수도 있다. 이때 사용되는 GRASS 명령은 다음과 같다.

```
$ s.surf.tps
```

이 명령은 결과값을 raster값으로만 저장하고, 계산 중 내삽(interpolation)과정을 거치게 된다. 또한, 이 명령에서는 정밀한 계산을 위해서 다음과 같은 여러 가지 옵션사항을 설정할 수 있다.

- dmin1 : 점과 점사이의 최소거리를 설정한다. 기본값은 데이터베이스 정의시 설정된 raster 해상도의 각 셀값의 0.5배이다.
- zmult : z값 변환 인자이다. 기본 값은 1이다. 이것은 수평 축적과 수직축적의 비를 나타내는 값으로 수평변화에 비해서 높이 변화량이 너무 작아서 지형의 변화를 쉽게 파악할 수 없을 경우 이 수치를 높여 줌으로써, 수직 변화를 증폭시키는 역할을 한다. 주로 사막이나 평야지대 등에서 미미한 지형변화를 파악하기 위해서 많이 이용하는 옵션이다.
- tension : 지형 탄력 값을 설정하는 인자이다. 기본값은 40이며, 부드러운 지형을 형성하는데 적절하다.
- smooth : 지형의 광택도를 설정하는 변수이다. 기본 값은 0이며, 무광택으로 처리된다.
- segmax : 단위 구획당 점의 최대 개수를 설정하는 옵션이다. 기본값은 40이다.
- npmin : 내삽에 사용될 점의 최소개수를 설정한다. 기본 값은 150이며, 기복이 심한 지형에 적합하다.

그러나, 대부분 기본값으로 분석을 해도 별 무리가 없다. 분석 알고리즘은 tension값을 이용한 Spline 내삽이다.

결과의 출력

점 데이터를 이용한 내삽의 결과는 대부분 raster로 출력되기 때문에, 여기서는 raster 이미지의 출력에 대해서 잠시 살펴보겠다. 결과 출력을 위해서는 우선 앞에서 설명한 것과 같이 GRASS 모니터를 작동시켜야 한다.

raster 이미지를 출력하기 위한 명령어는 다음과 같다.

```
$ d.rast rasterfile
```

어떤 raster파일이 있는지를 확인하기 위해서 사용하는 명령은 다음과 같다.

```
$ g.list rast
```

위의 두 명령을 이용해 관심있는 래스터 이미지를 검색하고 화면에 출력할 수 있을 것이다. 여기서 우리는 raster 이미지의 특정 부분을 확대해서 살펴봐야 할 경우가 있는데, 이때 사용하는 명령은 다음과 같다.

```
$ d.rast.zoom
```

이 명령을 입력한 후 GRASS 모니터 위로 마우스를 움직여서 관심있는 특정 부분의 한쪽 모서리 지점을 선택하고, 마우스를 계속 움직이면 확대영역 범위를 보여주는 사각형 박스가 형성될 것이다. 이때 오른쪽 마우스 단추를 클릭함으로써, 선택된 영역을 확대할 수 있다. 불행하게도 GRASS에서는 마우스만을 이용한 zoom-out(축소)기능을 지원하지 못하고 있다(즉, 확대 전의 상태로 복원할 수 없다). "d.rast.zoom"명령은 마지막 단계의 zoom 상태로 돌아가는 unzoom 기능을 지원하고 있을 뿐이다. mapset을 raster이미지 최대 범위영역인 기본 범위 값으로 설정하게 되면(즉, 래스터 이미지 전체를 모니터에 출력한 후), 특정 부분을 다시 확대해 볼 수 있다. mapset을 기본 좌표영역으로 설정하는 명령은 다음과 같다.

```
$ g.region -d
$ d.erase
```

d.erase 명령은 새로운 좌표값을 GRASS 모니터에 전달하기 위해, 좌표 값 변환 후에는 항상 입력되어야 한다. 그렇지 않으면, 이전의 좌표값 상태의 화면 출력이 그대로 유지되며, 변화된 내용을 화면상에서 볼 수가 없을 것이다. "g.region"명령을 이용하면 좌표 값을 수동으로 설정하는 것도 가능하다.

2.2 점고(Spot Heights)를 이용한 지표면 내삽(Intrepoation)

GRASS에는 지표면 내삽을 위한 세 가지 모듈(module)이 있으며, 생성된 지표면 모델은 raster 포맷으로 저장된다. 우선 중요한 것은 점고(spot height)의 숫자를 파악하는 것이다. 정확한 개수를 모르면 내삽을 제대로 수행할 수 없을 것이다. 그래서, 우선 각 점의 좌표와 높이 값을 포함하고 있는 ASCII 파일에 포함된 점의 개수를 파악해야 한다. 이것을 위해서 우리가 사용할 수 있는 Unix 명령은 "wc"이다. 이 명령은 단어의 개수를 파악하는 명령이며, -l 옵션으로 문자열의 수도 파악할 수 있다.

```
$ wc -l spotheights.asc
```

이 명령을 수행하면 점의 개수를 파악할 수 있고, 이 정보를 이용해서 내삽을 시작할 수 있다.

```
$ s.surf.idw in=spotheight out=elevation npoints=<number>
```

입력(input)은 점 데이터 파일이름(GRASS에서 인식하고 있는 이름, g.list sites명령으로 확인할 수 있음)이며, <number>는 위에서 파악된 점의 개수이다. 또한, 출력(out)은 새로 생성될 래스터의 이름이다.

```
$ d.rast elevation
```

이 명령을 이용해서 새로 생성된 DEM을 화면에 출력할 수 있다.

"Kriging" 알고리즘을 이용하려면, 다음과 같은 모듈을 이용해야 한다.

```
$ s.surf.krig
```

이 모듈은 GRASS의 외부 확장 모듈의 한 사례이며, 이것은 미국의 한 대학에서 개발된 것이다. Kriging에 대한 정보는 SURFER(TM)의 도움말-페이지(help-page)에서 찾을 수 있다. 적절한 변수를 입력해야만 올바른 결과를 얻을 수 있다. 도움말을 참고 바란다.

위에서 이용된 **$s.surf.tps**모듈은 주어진 점고(spot height) 데이터로부터 DEM을 추출해내는 명령이다. 이때, 표면 탄력값(tension) 알고리즘이 사용된다.

2.3 점 데이터에 대한 통계 분석

점 데이터에 대한 정보를 얻기 위해서는 다음의 명령으로 레포트를 준비할 수 있다.

```
$ s.menu
```

이 명령은 메뉴 선택방식으로 진행되기 때문에, 쉽게 통계 정보를 얻을 수 있다.

3. 삼각망 지형모델 형성 (Triangulation)

GRASS 배포판에서는 삼각망 형성을 위한 두 가지 방식의 모듈을 제공하고 있다. 여러분들은 GRASS4.2.1에서 삼각망과 관련된 두 가지 새로운 패키지 (v.geom과 s.geom)를 찾을 수 있을 것이다. 이들 두 패키지는 삼각망 형성과 관련된 기존의 패키지를 수정 보완한 것이다. 여기서는 삼각망 형성과 관련된 표준적인 모듈에 대해 설명하겠다.

3.1 Delaunay 삼각망 형성

볼록한 껍질 모양한 삼각망을 형성하기 위해서는 다음과 같은 모듈이 사용된다.

```
$ s.delaunay
```

이 모듈의 결과물은 vector 데이터며, 다음의 명령으로 화면에 출력할 수 있다.

```
$ d.vect
```

물론, 생성된 vector 데이터의 목록을 먼저 살펴보기 위해서는 다음의 명령을 이용하면 된다.

```
$ g.list vect
```

3.2 Thiessen 다각형(Polygons)

강우량 데이터를 이용해서 수리학적 분석 중의 Thiessen 다각형을 계산해 내기 위해서 두 가지 방법이 있는데, 각각의 결과는 데이터 형식에서 차이가 있다. 한 방식은 결과물이 벡타(vector)형식의 다각형으로 저장되는 것이고, 또 다른 방식은 래스트(Raster) 형식으로 저장되는데, 선택은 분석결과를 향후 어디에 사용할 것인지에 따라서 결정된다. 분석에 이용되는 입력데이타는 일반적인 ASCII파일이며, 포맷은 각 기

상관측소의 동서 및 남북 좌표와 강유량 수치가 순서대로 입력되어 있는 것이면 된다.

3.2.1 Vector 포맷으로의 출력

다음의 모듈은 Thiessen 다각형을 Vector 포맷으로 생성한다.

```
$ s.voronoi
```

3.2.2 Raster 포맷으로의 출력

계산된 폴리곤을 Raster 포맷으로 저장하기 위해서는 다음과 같은 모듈을 이용한다.

```
$ s.surf.idw in=rainfall out=thiessen npoint=1
```

여기서는 내삽 점의 개수(npoint)를 1로 설정하는 것에 주의해야 한다.

4. 점 데이터를 다른 형식으로 변환하기

4.1 점을 래스터로 변환하기

이러한 형태의 변환에는 내삽 작업이 수행되지 않는다. 점(sites)들은 좌표 변환 없이 곧 바로 래스터 데이터로 변환된다. 값이 없는 cell에는 0이라는 값이 할당된다.

```
$ s.to.rast
```

점들을 특정 vector 형식의 grid(v.mkgrids 명령으로 만들어짐) 위에 위치시키고자 할 때는 다음명령을 사용한다.

```
$ s.medp
```

모든 점은 Vector Grid의 가장 가까운 노드에 위치하게 되며, 결과는 새로운 sitemap으로 생성된다.

4.2 점을 벡타(Vector)로 변환하기

점을 벡타 포맷으로 변환하기 위해서는 다음의 명령을 사용한다.

```
$ s.to.vect
```

이때, 점들은 벡타포맷의 점으로 변환되며, 생성된 벡타 레이어는 다음의 명령으로 화면에 출력할 수 있다.

$ d.vect