(Let's) Make Fortran spatial again: Re-establishing Fortran as a language for spatial analysis

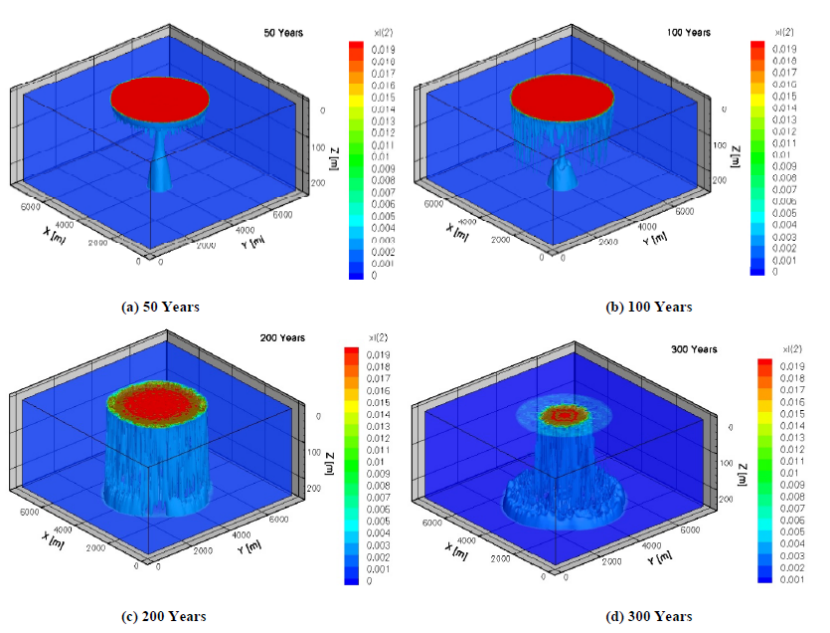
Oscar García-Cabrejo

School of Geological Engineering
Universidad Pedagógica y Tecnológica de Colombia

Contents

Introduction

Spatial Analysis



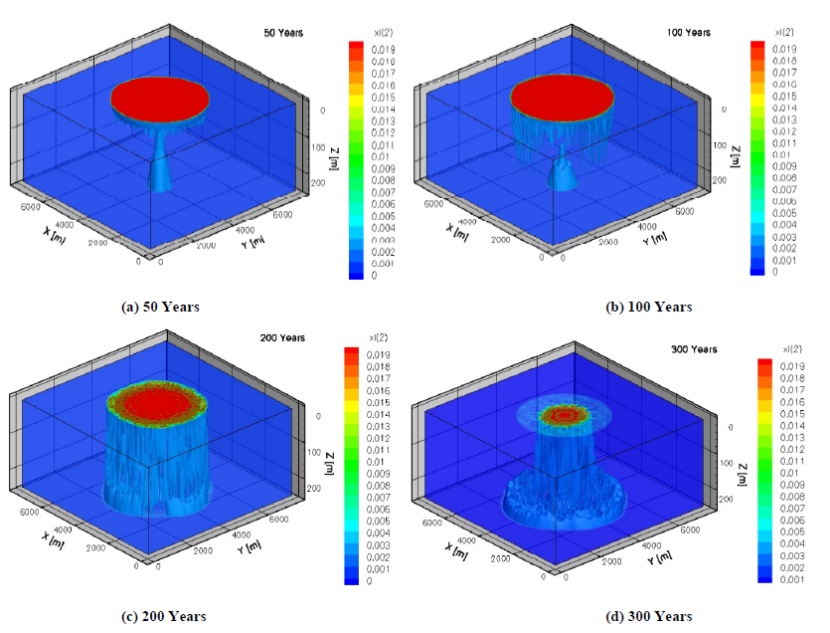
Model of dissolved CO2 in a saline aquifer.

Source: Hammond2012

Context

- High computational demands of modern geospatial data (e.g., climate modeling, reactive transport in porous media)

Spatial Analysis



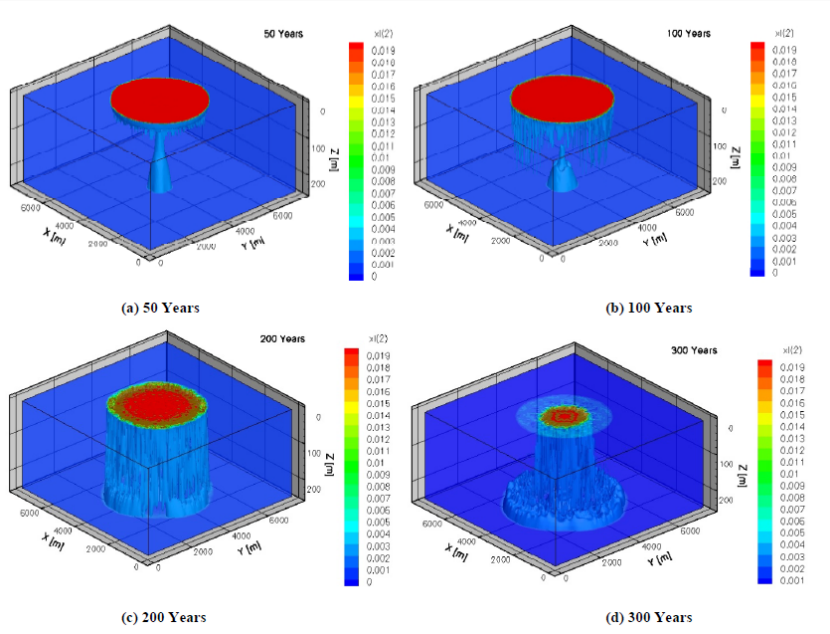
Model of dissolved CO2 in a saline aquifer.

Source: Hammond2012

Context

- High computational demands of modern geospatial data (e.g., climate modeling, reactive transport in porous media)
- These are massive spatial multiphysics problems.

Spatial Analysis



Model of dissolved CO2 in a saline aquifer.

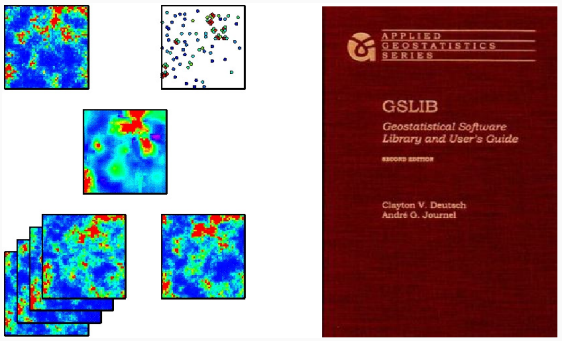
Source: Hammond2012

Context

- Fortran lacks of support for modern geospatial data operations

Spatial Analysis

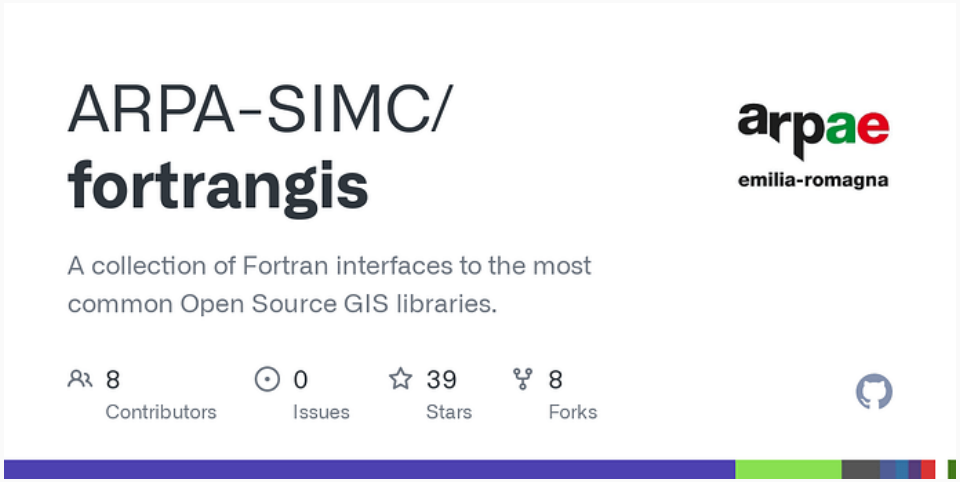
Source: `https://mossgis.org/`



Source: `https://www.gslib.com/`

Previous Work

- 1977: Map Overlay and Statistical System (MOSS) was originally implemented in Fortran77
- 1980-1990: Fortran was the main language for spatial analysis
- 1992: GSLIB (Geostatistical Software Library)

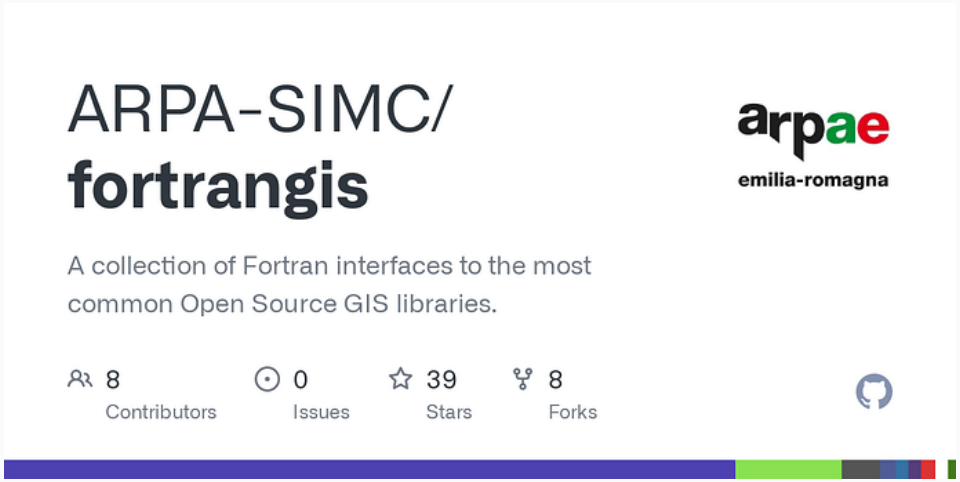


FortranGIS

- FortranGIS collection of Fortran interfaces to some spatial libraries (David Cesari)

Source: `https://github.com/ARPA-SIMC/fortrangis`

Spatial Analysis

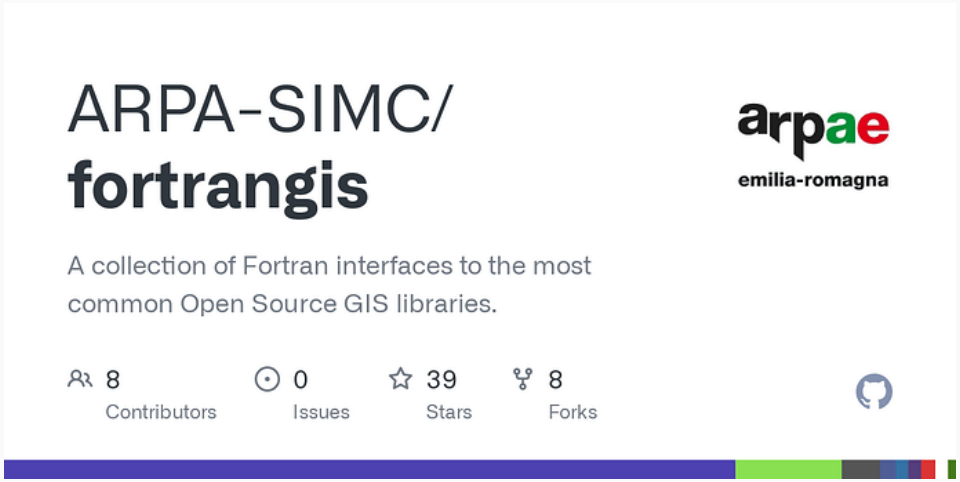


FortranGIS

- FortranGIS collection of Fortran interfaces to some spatial libraries (David Cesari)
- Libraries Coordinate transformation, vector/raster I/O

Source: `https://github.com/ARPA-SIMC/fortrangis`

Spatial Analysis

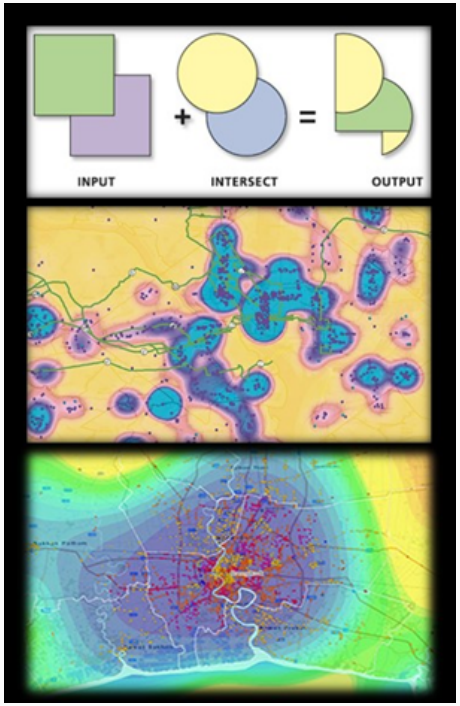


Source: `https://github.com/ARPA-SIMC/fortrangis`

FortranGIS

- FortranGIS collection of Fortran interfaces to some spatial libraries (David Cesari)
- Libraries Coordinate transformation, vector/raster I/O
- Issues Changes in API, library updating, **compilation**

Spatial Analysis

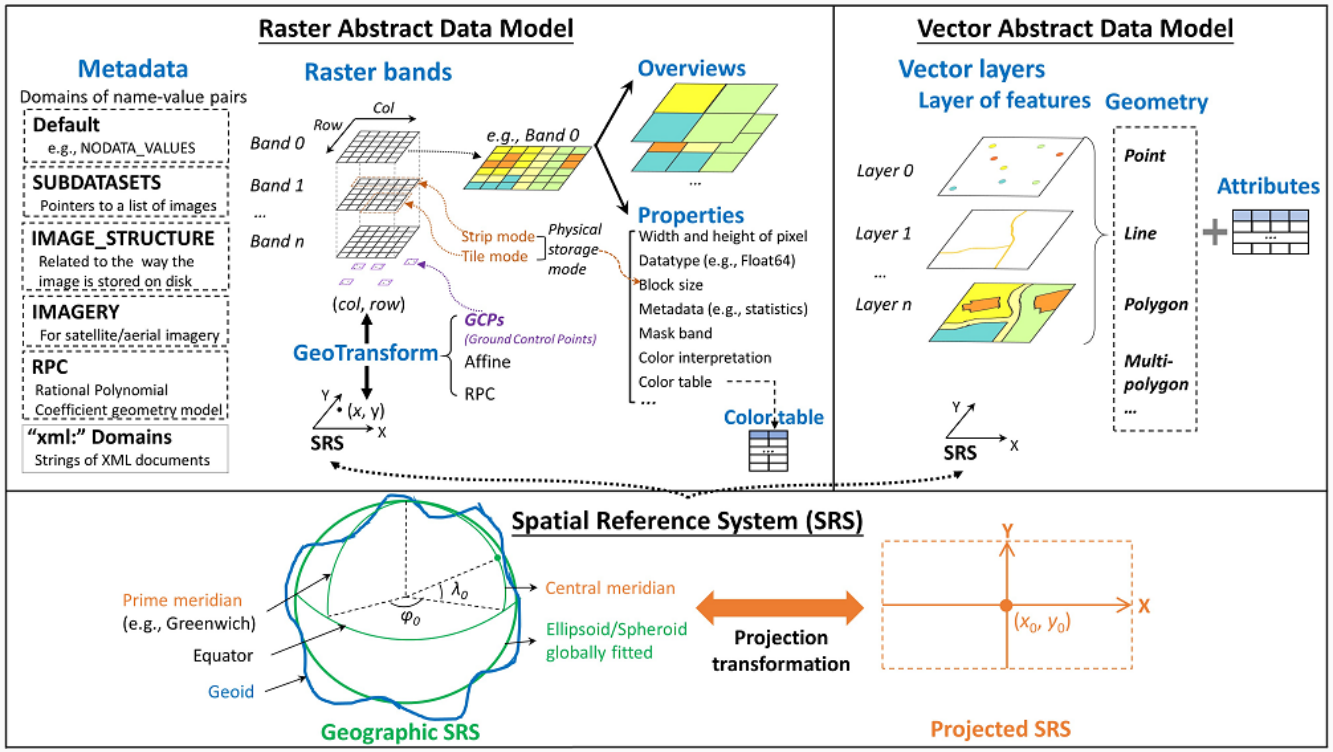


Source: `https://giscenter.sites.clemson.edu/home/basic-spatial-analysis.html`

Problem

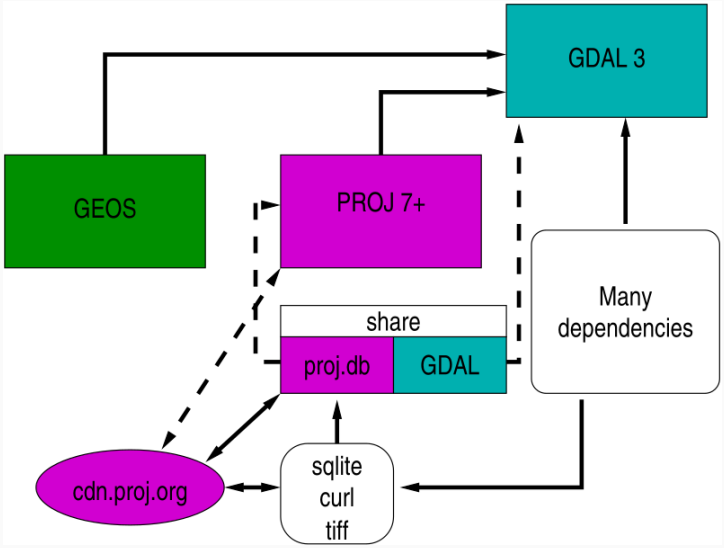
- Develop improved Fortran bindings to geospatial libraries
- Modern Fortran interfaces (Low/High level)
- Permissive licenses (MIT, BSD)
- Use of Fortran Package Manager

Spatial Analysis



Source:

Spatial Analysis

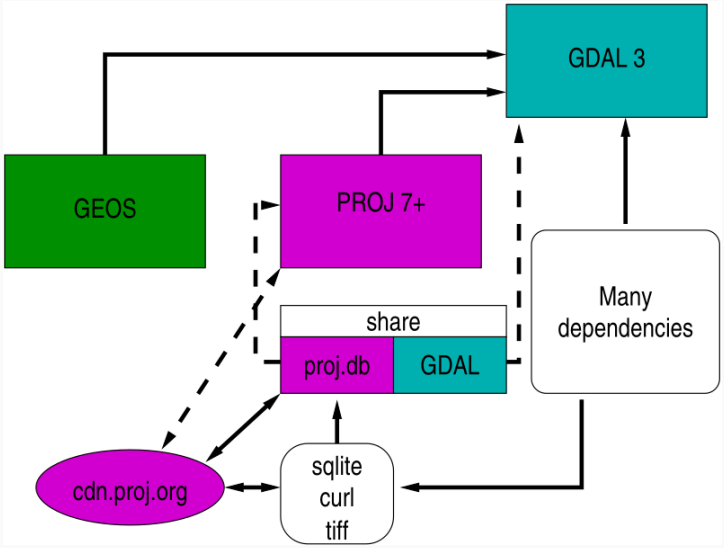


Ecosystem

- PROJ

Source: `https://rsbivand.github.io/foss4g_2022`

Spatial Analysis

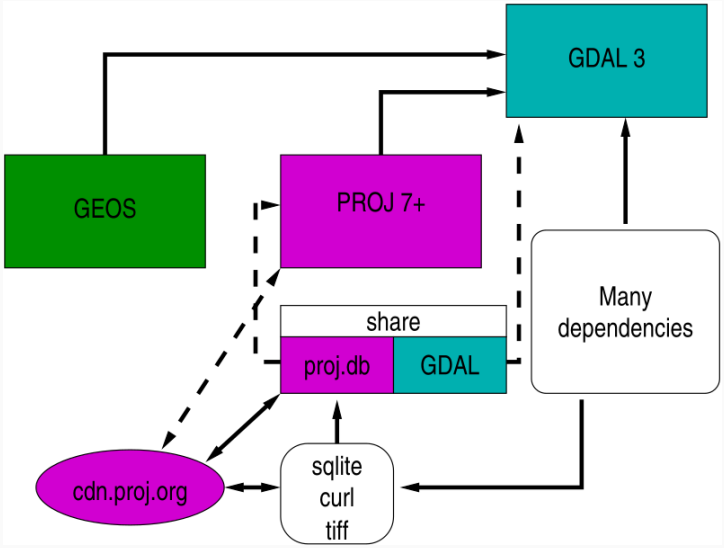


Ecosystem

- PROJ
- GDAL

Source: `https://rsbivand.github.io/foss4g_2022`

Spatial Analysis

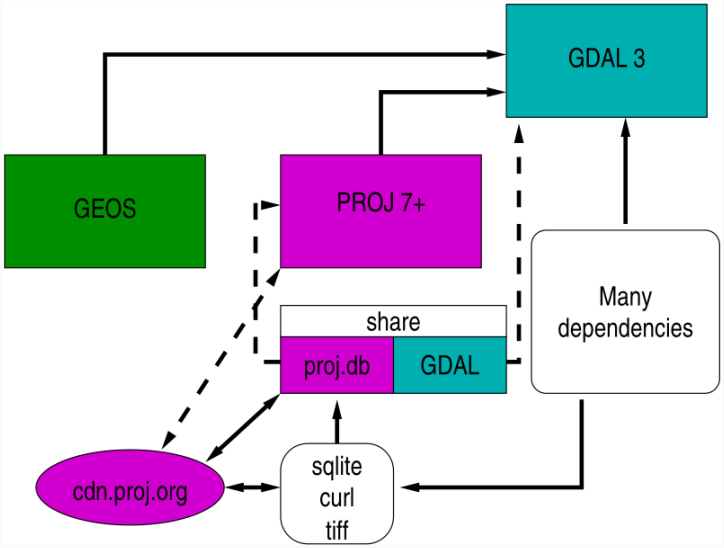


Ecosystem

- PROJ
- GDAL
- GEOS

Source: `https://rsbivand.github.io/foss4g_2022`

Spatial Analysis



Source: `https://rsbivand.github.io/foss4g_2022`

Ecosystem

- PROJ
- GDAL
- GEOS
- Other libraries (shapelib)

FProj

Proj

- PROJ provides coordinate reference system definitions, projections and datum transformations

Source: `https://www.osgeo.org/projects/proj/`

Proj

Source: `https://www.osgeo.org/projects/proj/`

- PROJ provides coordinate reference system definitions, projections and datum transformations
- PROJ stores CRS, projection and transformation info in a SQLite database

Constructors

```
use fproj, only: CRS
type(CRS) :: my_crs,crs1,crs2
! Create Coordinate Reference System objects
call my_crs%create(4326)
call my_crs%create("EPSG:4326")
call my_crs%create("+proj=latlon")
```

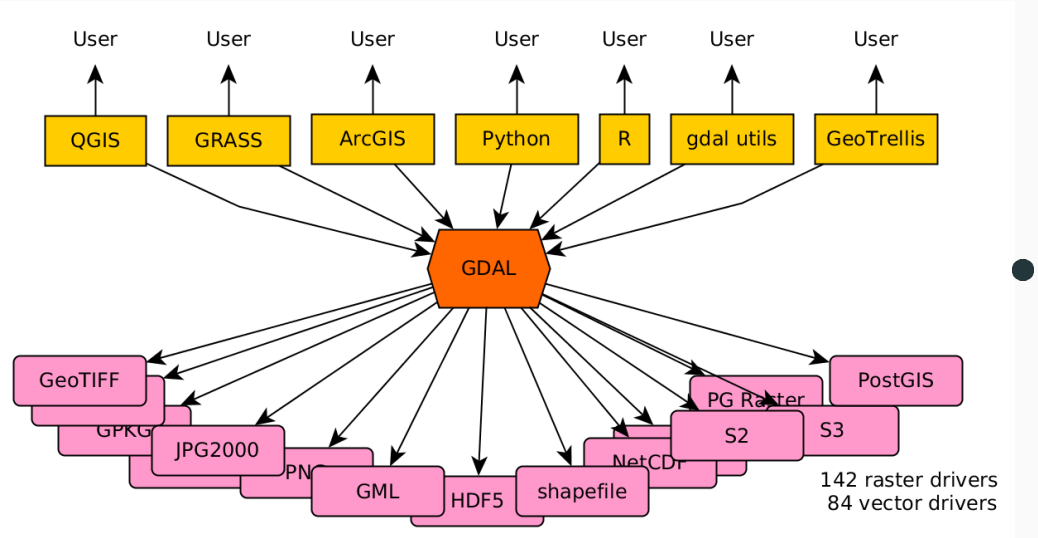
```
type(CRS) :: crs_4326, crs_3116
!
call crs_4326%create_from_epsg(4326)
call crs_3116%create_from_string("EPSG:3116")
call crs_proj%create_from_proj4("+proj=latlon")
```

Coordinate Transformation

```
type(transformer) ::
my_transformer,my_transformer1
crs_4326 = CRS.from_epsg(4326)
crs_26917 = CRS.from_epsg(26917)
call
my_transformer%create_from_crs(crs_4326,crs_26917)
;
my_transformer%create_from_crs("EPSG:4326",&
"EPSG:26917");
!
call my_transformer%transform(50.0, -81.5);
call my_transformer%transform([34.5,67.8,21.5],
[34.5,67.8,21.5]);
!
my_transformer1%create_from_crs("EPSG:4326",&
"EPSG:26917", always_xy=TRUE);
```

FGDAL

GDAL



- GDAL provides abstractions for reading, writing and manipulating GIS raster and vector data

Source:

`https://r-spatial.org/2016/11/29/openeo.html`

Constructor/Accessors

```
use fgdal, only: gdal,dataset,bounding_box,gdal_crs;
!
type(gdal) :: my_gdal
type(dataset) :: current_ds
type(bounding_box) :: bbox
type(gdal_crs) :: current_crs,current_crs1
character(len=NUMCHAR) :: current_name, current_mode
real(kind=wp),allocatable :: band1(:,:),band2(:,:)
!
call my_gdal%open(current_ds,'boyaca.tif');
!
write(*,*) 'Current name= ',&
    trim(current_ds%name())
write(*,*) 'Current mode= ',&
    trim(current_ds%mode())
```

Attributes

```
!
! Dataset attributes
!
write(*,*) 'count= ',current_ds%count();
write(*,*) 'width= ',current_ds%width();
write(*,*) 'height= ',current_ds%height();
write(*,*) 'nrow= ',current_ds%nrow();
write(*,*) 'ncol= ',current_ds%nrow();
!
call current_ds%bounds(bbox);
call bbox%print();
call current_ds%crs(current_crs)
call current_crs%print();
current_crs1.from_epsg(3116)
!
call current_ds%read(1,band1);
call current_ds%read(2,band2);
!
write(*,*) 'band1 allocated? ',allocated(band1);
write(*,*) 'band2 allocated? ',allocated(band2);
!
call my_gdal%close(current_ds);
```

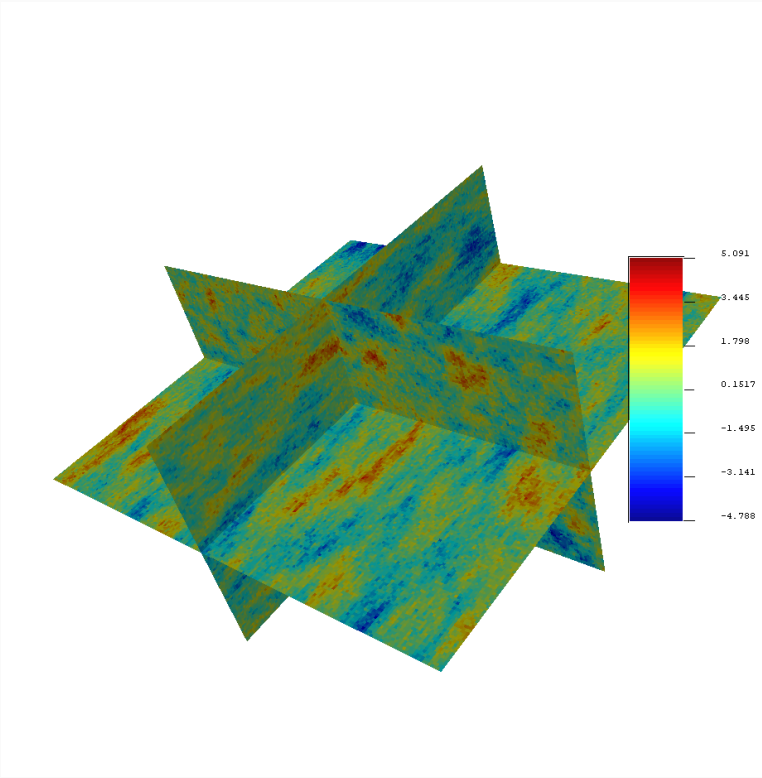
Create a raster file

```
type(dataset) :: current_ds1
real(kind=wp),allocatable :: A(:,:)
!
call current_crs%create(3116)
!
call my_gdal%open(current_ds1,&
    file = 'temporal.tif', &
    mode = 'w', &
    format= 'tiff', &
    nrow = 1730, ncol = 1890, &
    crs = current_crs);
!
allocate(A(1730,1980));
call random_number(A);
call current_ds1%write(A,band=1);
call current_ds1%close();
deallocate(A);
```

Operations

```
!
! Some operations
!
type(datasest) :: current_ds1,current_ds2;
real(kind=wp),allocatable :: coords_polygon(:,:)
!
call my_gdal%open(current_ds1,'head_central.tif');
call my_gdal%read_polygon_ascii('polygon.out',&
    coords_polygon);
!
call my_gdal%open(current_ds2,&
    file = 'head_central_mask.tif', &
    mode = 'w', &
    format= 'tiff', &
    nrow = 835, ncol = 790, &
    crs = current_crs);
!
call my_gdal%mask(current_ds1,coords_poligon,&
    .true.,current_ds2);
call current_ds2%write(1);
call current_ds2%close();
```

Example: ATALIB

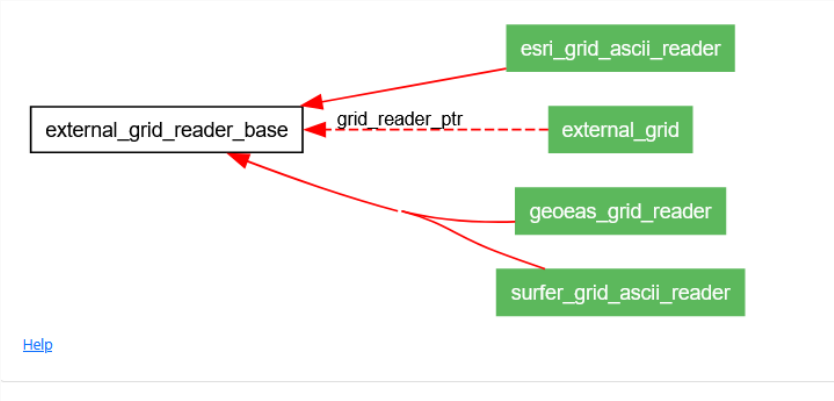


Log_{10} Permeability generated with sequential gaussian simulation

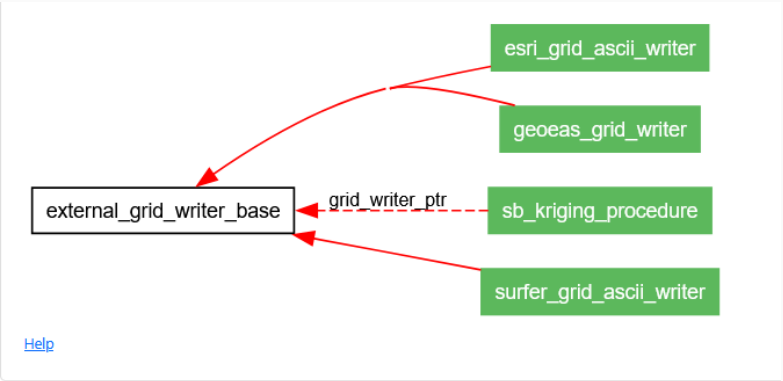
ATALIB

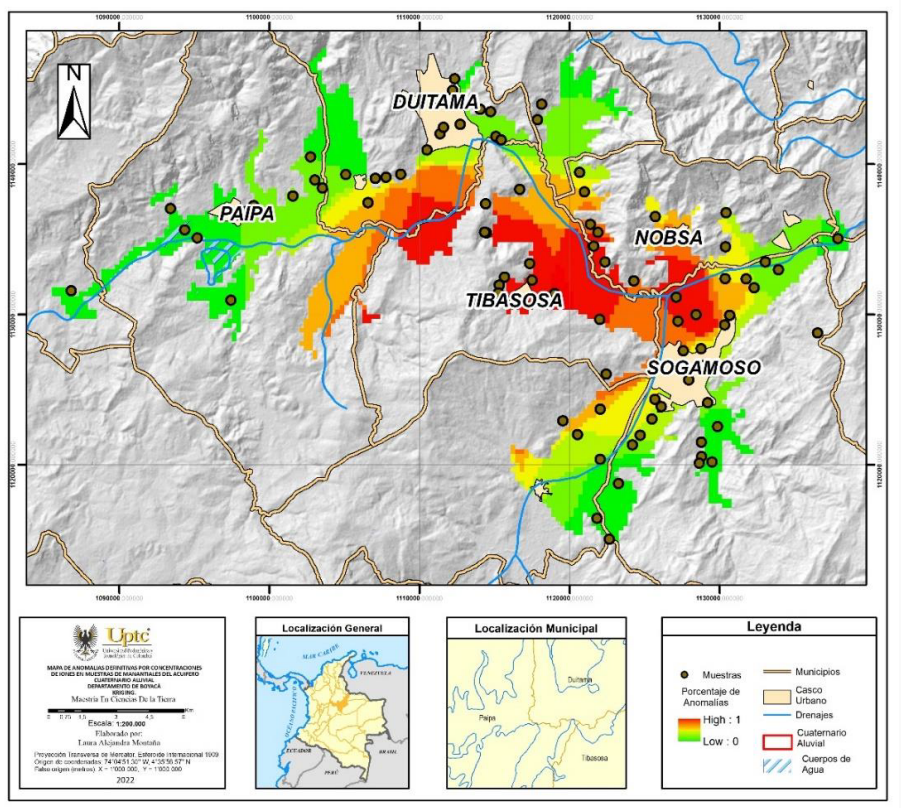
- Library for geostasistical modeling
- Methods
  - Declustering
  - Semivariogram calculation and fitting
  - Estimation (Kriging)
  - Stochastic simulation

Grid reader



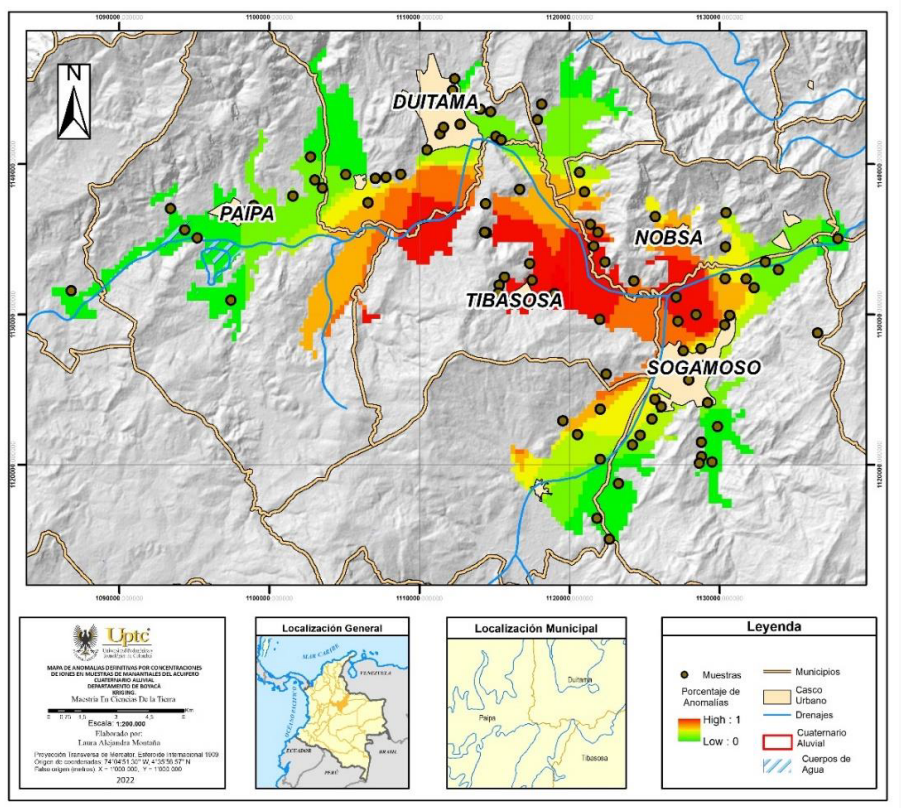
Grid writer





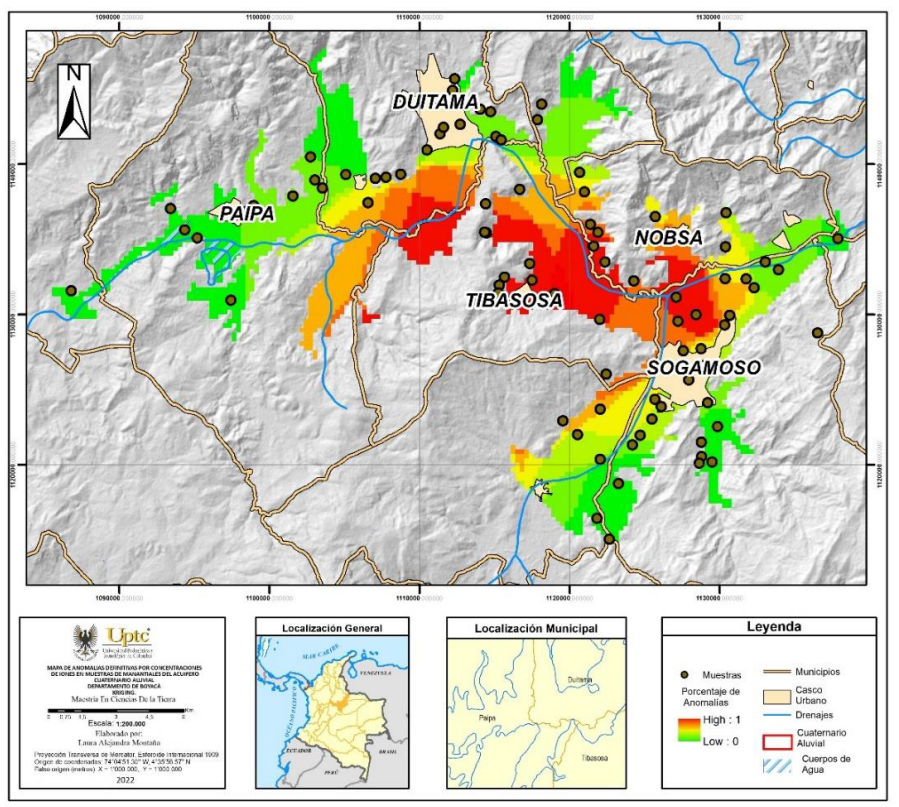
Example

- Anomalies in chemical composition of groundwater



Example

- Anomalies in chemical composition of groundwater
- Method: Local Singularity Analysis (Fractal method)



Example

- Anomalies in chemical composition of groundwater
- Method: Local Singularity Analysis (Fractal method)
- Output

ESRI ASCII Grid	16.4 Mb
GEOEAS Grid	15.9 Mb
Surfer ® Grid	17.7 Mb
Tiff Raster	4.3 Mb

Final Thoughts

Final Thoughts

- Implementation bindings to spatial libraries (Fproj, Fgdal) can improve the interoperability of Fortran code without compromising efficiency

Final Thoughts

- Implementation bindings to spatial libraries (Fproj, Fgdal) can improve the interoperability of Fortran code without compromising efficiency
- High level interfaces to spatial libraries can help in providing native spatial capabilities to existing Fortran applications

Final Thoughts

- Implementation bindings to spatial libraries (Fproj, Fgdal) can improve the interoperability of Fortran code without compromising efficiency
- High level interfaces to spatial libraries can help in providing native spatial capabilities to existing Fortran applications
- Spatial operations can be achieved using additional bindings (GEOS)

Thanks!!!!!

Contact
Oscar García-Cabrejo

`oscar.garcia04@uptc.edu.co`

Fproj `https://github.com/khaors/fproj`

FGDAL `https://github.com/khaors/fgdal`

ATALIB `https://github.com/khaors/atalibf08`