USGS
science for a changing world

A flexible Python script for stratified random site selection

Curtis Price

U.S. Department of the Interior
U.S. Geological Survey

- This information is preliminary and is subject to revision. It is being provided to meet the need for timely best science. The information is provided on the condition that neither the U.S. Geological Survey nor the U.S. Government may be held liable for any damages resulting from the authorized or unauthorized use of the information.

USGS

Random Site Selection is important

- The design of a sampling network is very important when trying to answer a question
- Removing as much bias as possible is critical to the usefulness of sampling results

USGS

USGS random sampling tool

COMPUTERIZED STRATIFIED RANDOM SITE-SELECTION APPROACHES

FOR DESIGN OF A GROUND-WATER-QUALITY SAMPLING NETWORK

by Jonathon C. Scott

U.S. GEOLOGICAL SURVEY

Water-Resources Investigations Report 90-4101

pubs.er.usgs.gov/publication/wri904101

Oklahoma City, Oklahoma

1990

USGS

USGS random sampling tool

- Designed for a national water-quality assessment
- Used for many national, regional, and local USGS ground-water studies
- Three random sampling methods

USGS

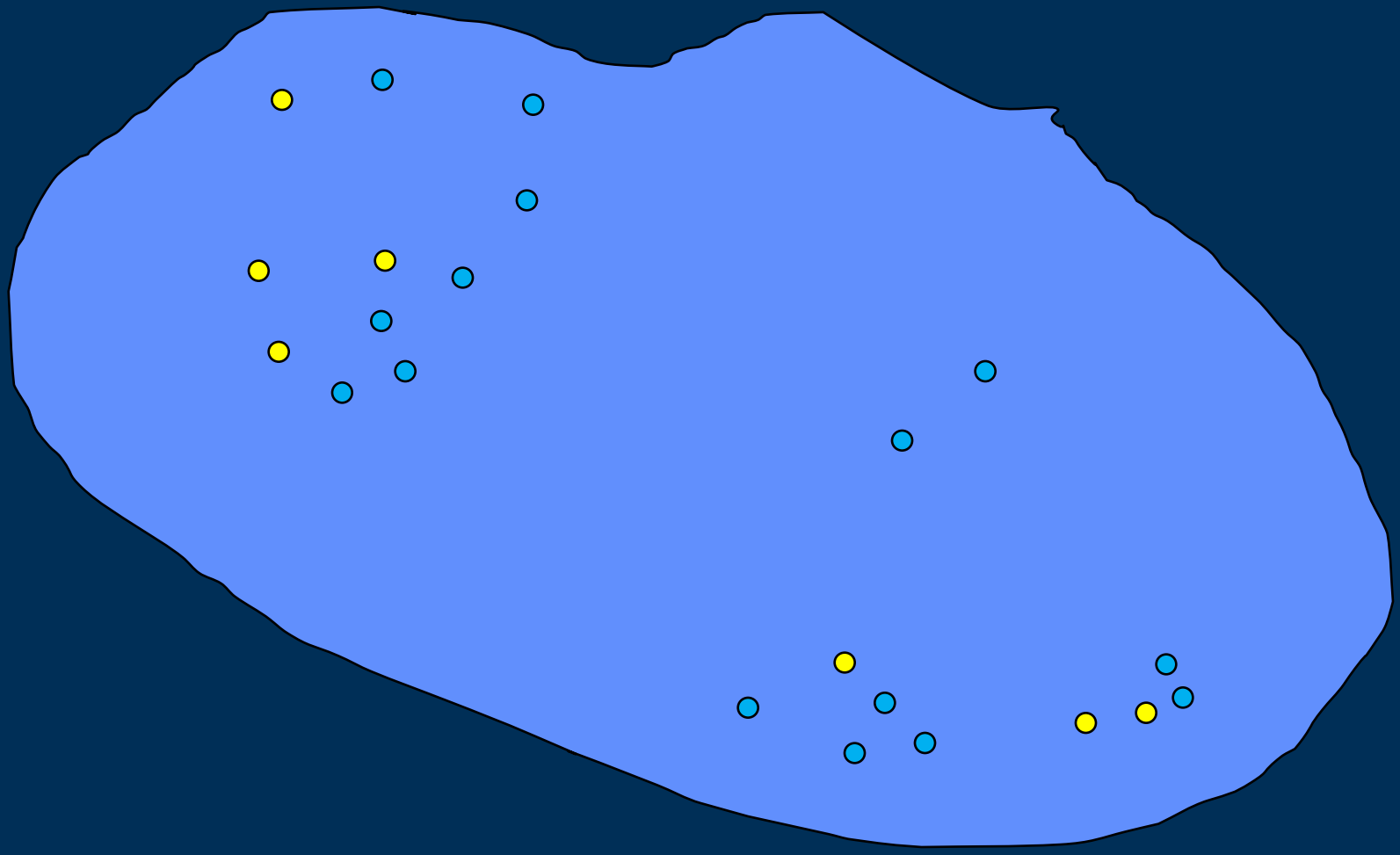
Three methods

1. Simple random selection

   (without replacement)
2. Random selection within cells
3. Random selection with iterating grids

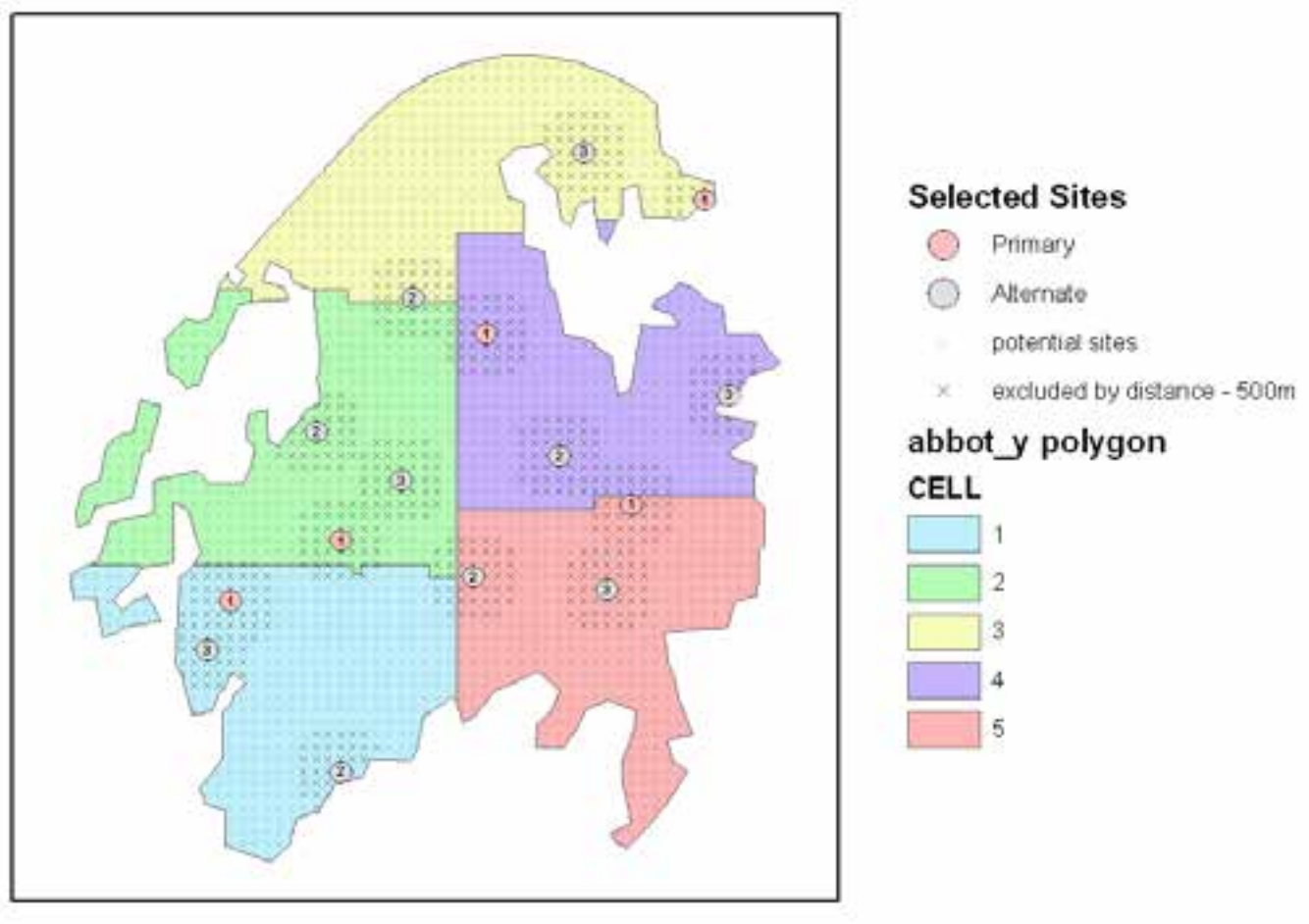
USGS

Simple random selection



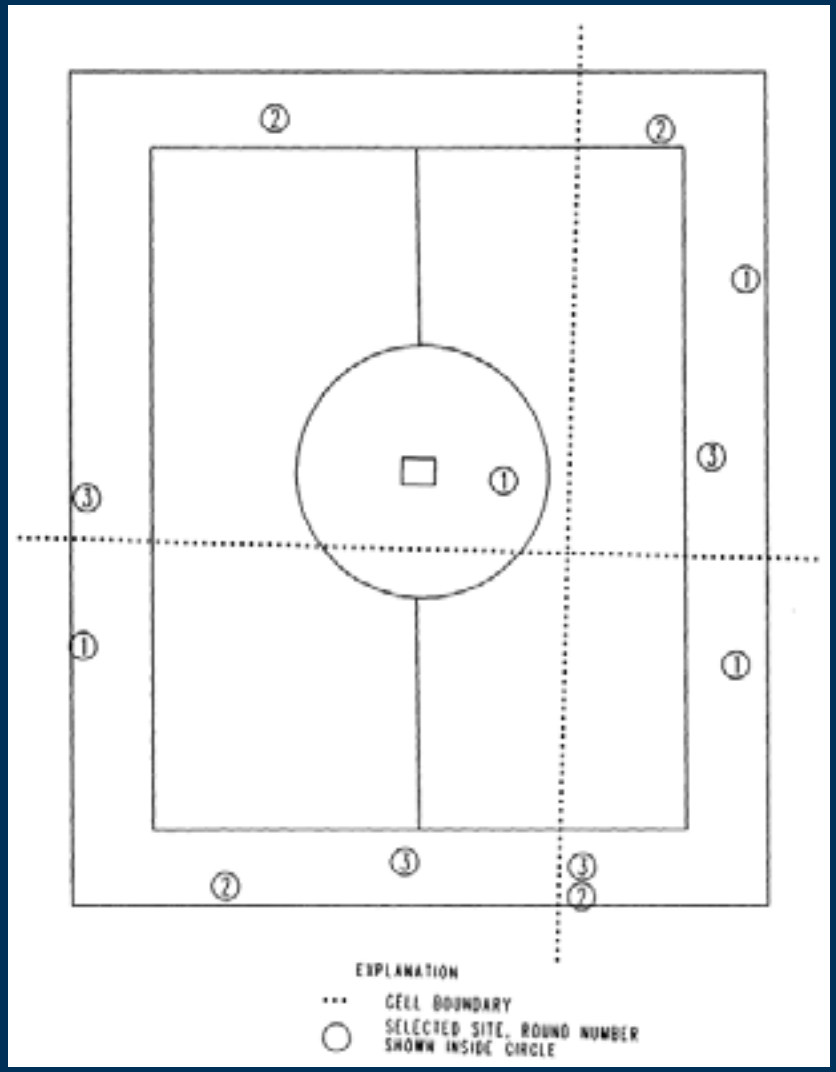
USGS

Random selection within cells



USGS

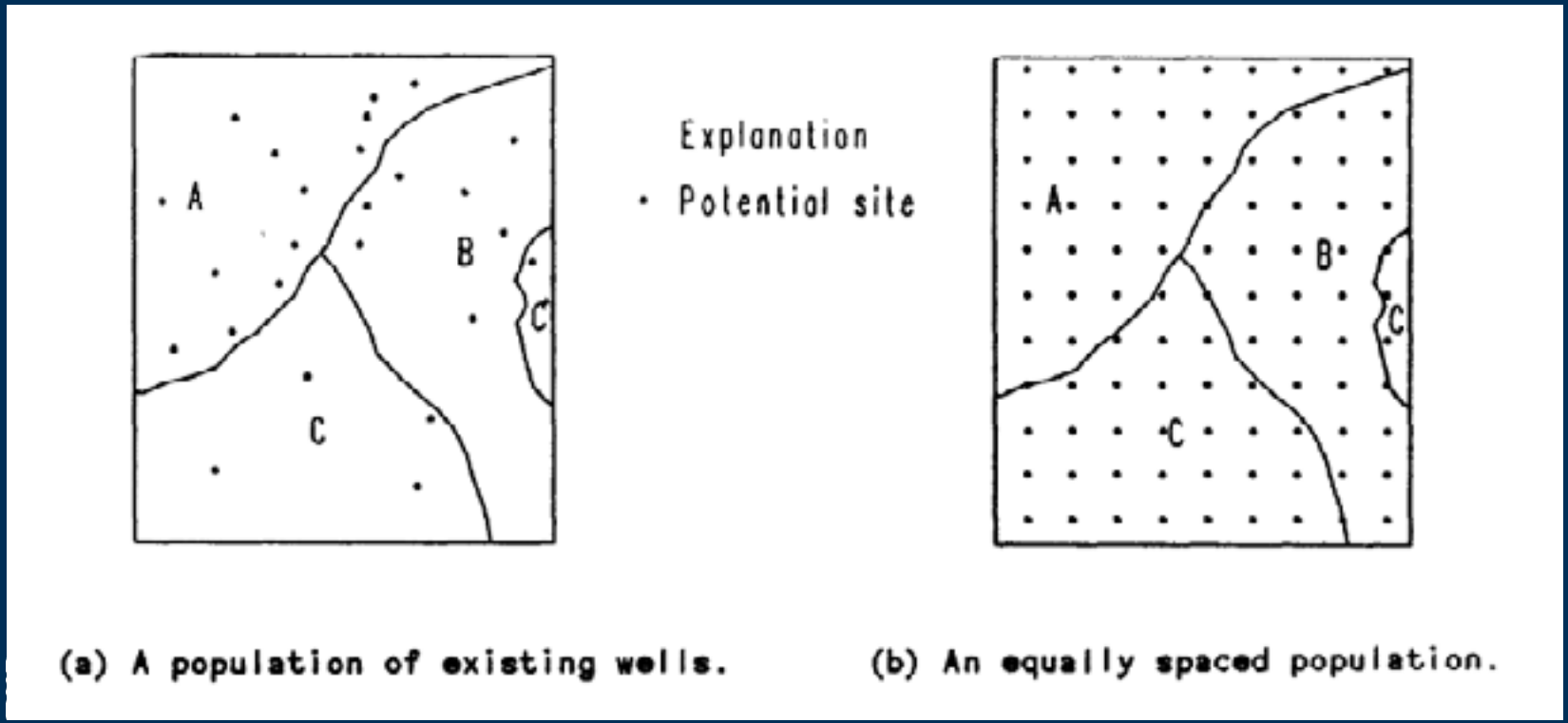
Random selection with iterating grids



USGS

Source points

- All three methods accept existing points or build a lattice of candidate points from a provided study area



(a) A population of existing wells.

(b) An equally spaced population.

Original implemenation

- Developed using
  - ArcInfo Workstation AML
  - ArcSDL FORTRAN (later versions modified to remove ArcSDL libraries)
- GIS polygon overlay is used to delineate and process sample areas
- Command-line interface, user is guided through a series of prompts to complete the analysis

USGS

SAMPLE.AML user interface

```
Arc: &R SAMPLE

Site selection may be performed using one of three methods:

   Code     Method

      1     Simple random selection
      2     Random selection with equal-area distribution
      3     Random selection with iterating grids

Enter method code: 2
Enter name of areal-subset coverage: LUSE
Enter name of category variable: CATEGORY

Checking areal-subset coverage specifications
Submitting command RESUME NAWSOFT>SAMP>FINAL>ITEXST.RUN
Directory of INFO data base is NAWSOFT>SAMP>FINAL>INFO
INFO user name is ARC
INFO file name is LUSE.PAT
```

Sample – arcpy implementation

- Functionality has been preserved as much as possible
- Implemented using raster approach for speed (using NumPy arrays)
- Runs with ArcGIS Basic license
- ArcGIS script tool interface
- Parameter validation code is used to simplify and enhance the user experience

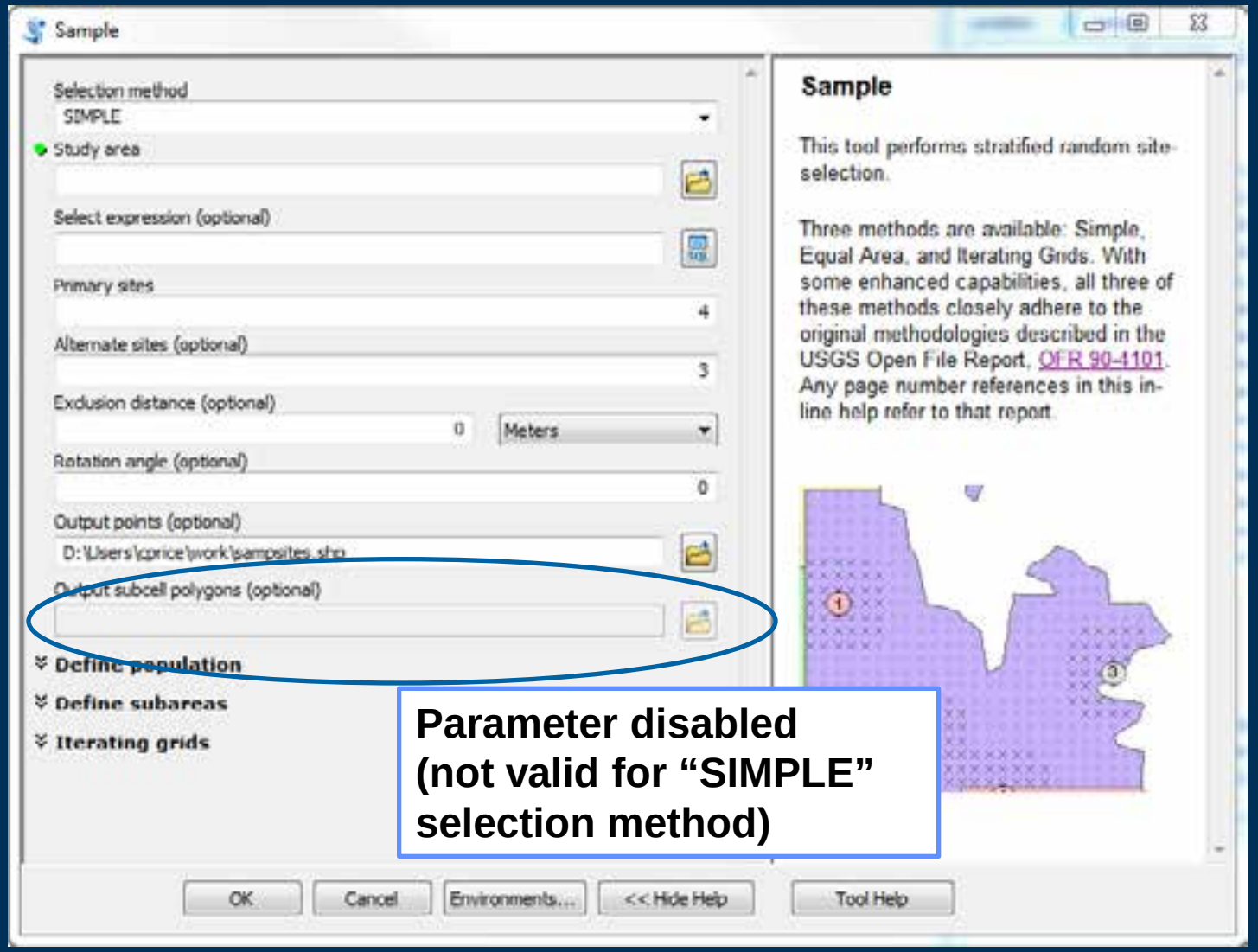
USGS

Sample tool user interface

- ArcGIS tool interface
- All three methods in one tool interface – with many (17!) parameters
- User experience simplified using
  - Parameter defaults
  - Parameter validation to set defaults based on inputs and disable unneeded parameters
  - Parameter categories to ‘hide’ parameters

USGS

Parameter validation



Parameter disabled (not valid for "SIMPLE" selection method)

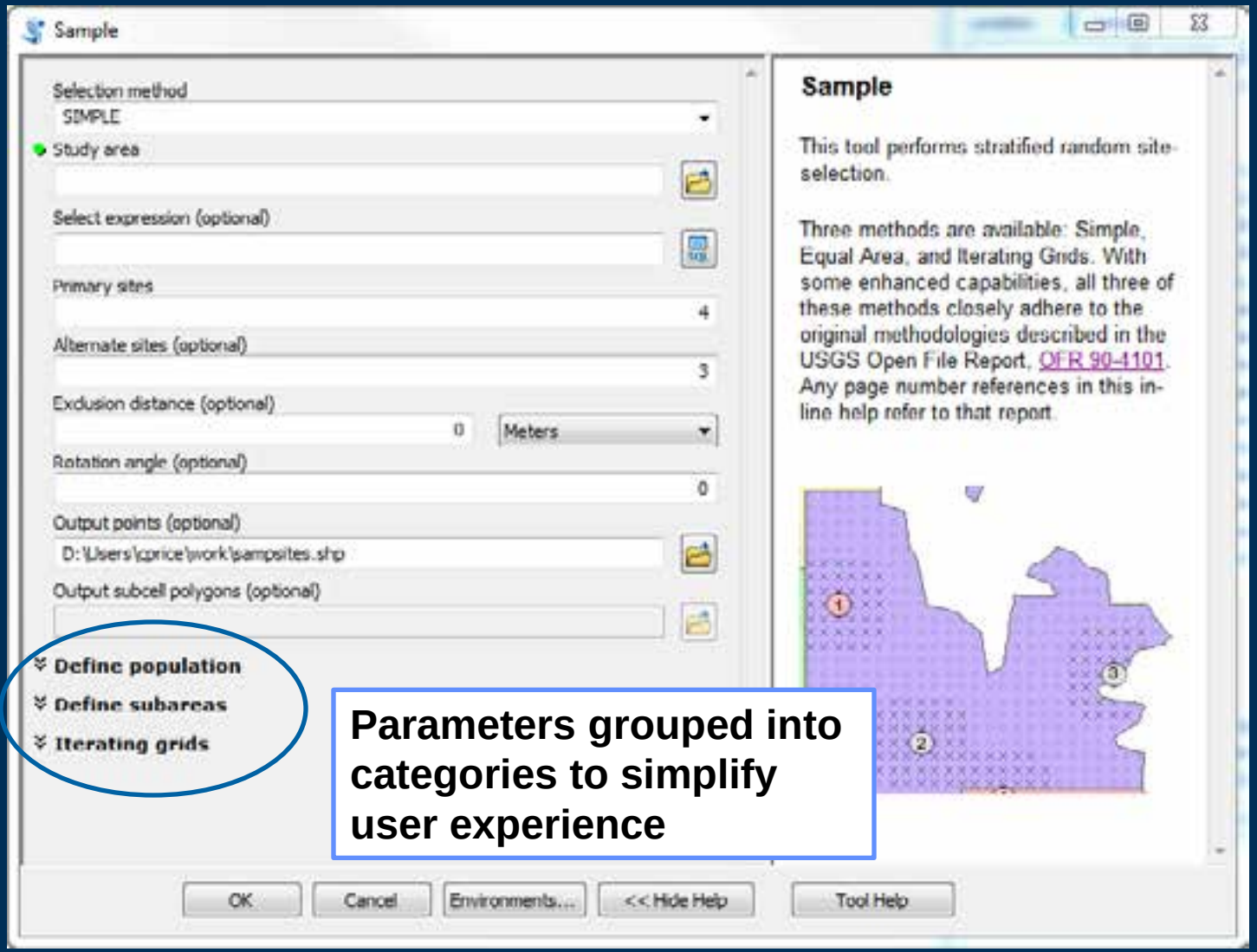
USGS

Parameter validation code

```
# determine method
meth = self.params[0].value
# ...
if m_simple:
  # SIMPLE method:
  #   no rotation
  #   no sample cell output
  #   no subarea parameters
  for p in [8, 12, 13, 14, 15, 16]:
    self.params[p].value = None
    self.params[p].enabled = False
```

USGS

Parameter categories



Parameters grouped into categories to simplify user experience

USGS

Parameter categories

- Tool validation code:

```
def initializeParameters(self):
  for p in [9, 10, 11]:
    self.params[p].category = \
       "Define population"
```

USGS

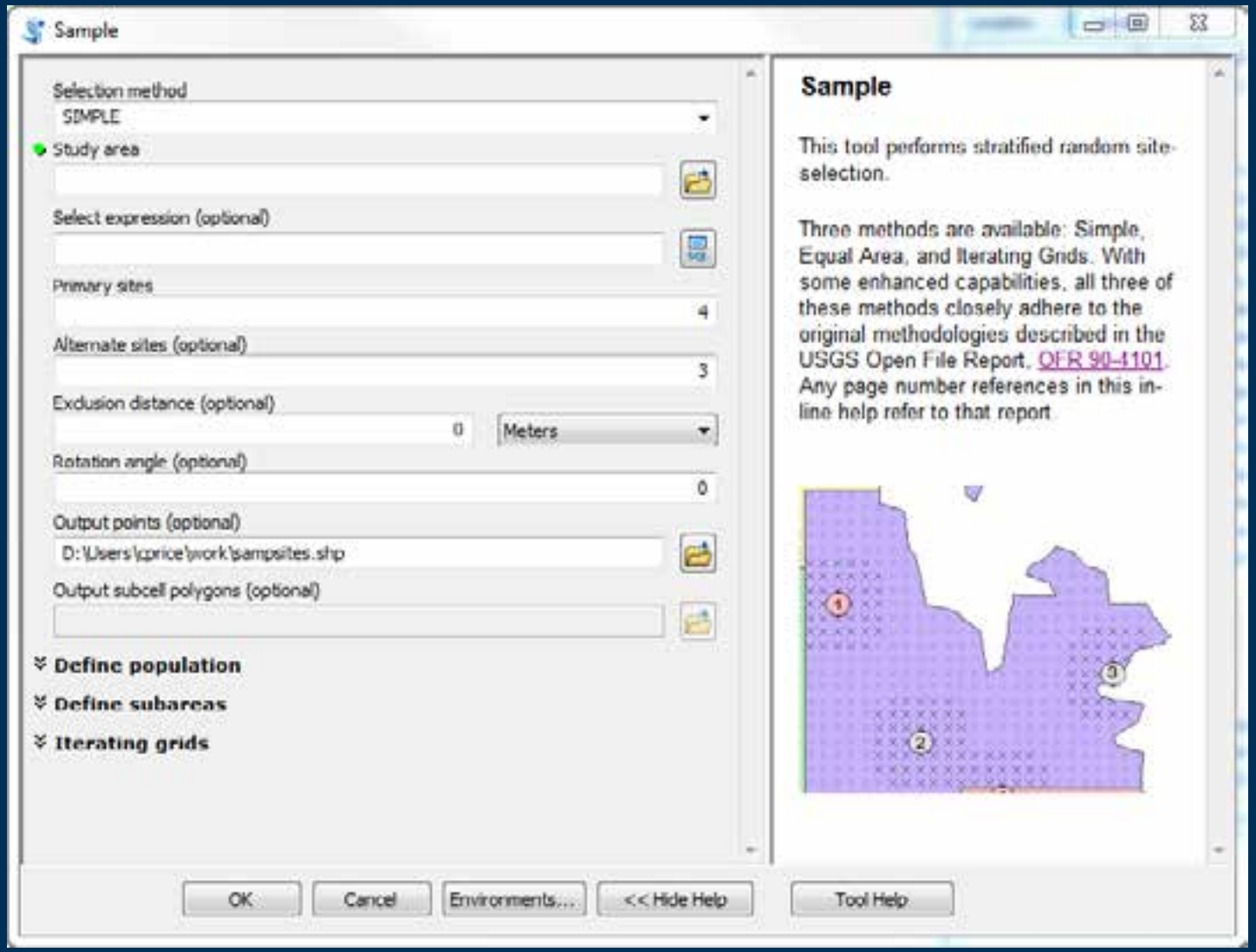
Using NumPy arrays

- RasterToNumPyArray, NumPyArrayToRaster

```
numpyNoData = 0
numpyData = 1
R = arcpy.RasterToNumPyArray(procRas,
                nodata_to_value=numpyNoData)
R = R.astype('int32') # force integer type
J = len(R)      # number of rows in studyarea raster
I = len(R[0])   # number of cols in studyarea raster
```

USGS

Sample tool demonstration



USGS

Questions?

USGS