Open Sourcing Generalization Tools

Rupert Brooks

GeoAccess Division,
Canada Centre for Remote Sensing
Natural Resources Canada

Outline

- Introduction
- History
- Goals of the project
- Open sourcing
- Four Applications
  - Agg
  - Net
  - Tri
  - Disp
- Conclusions

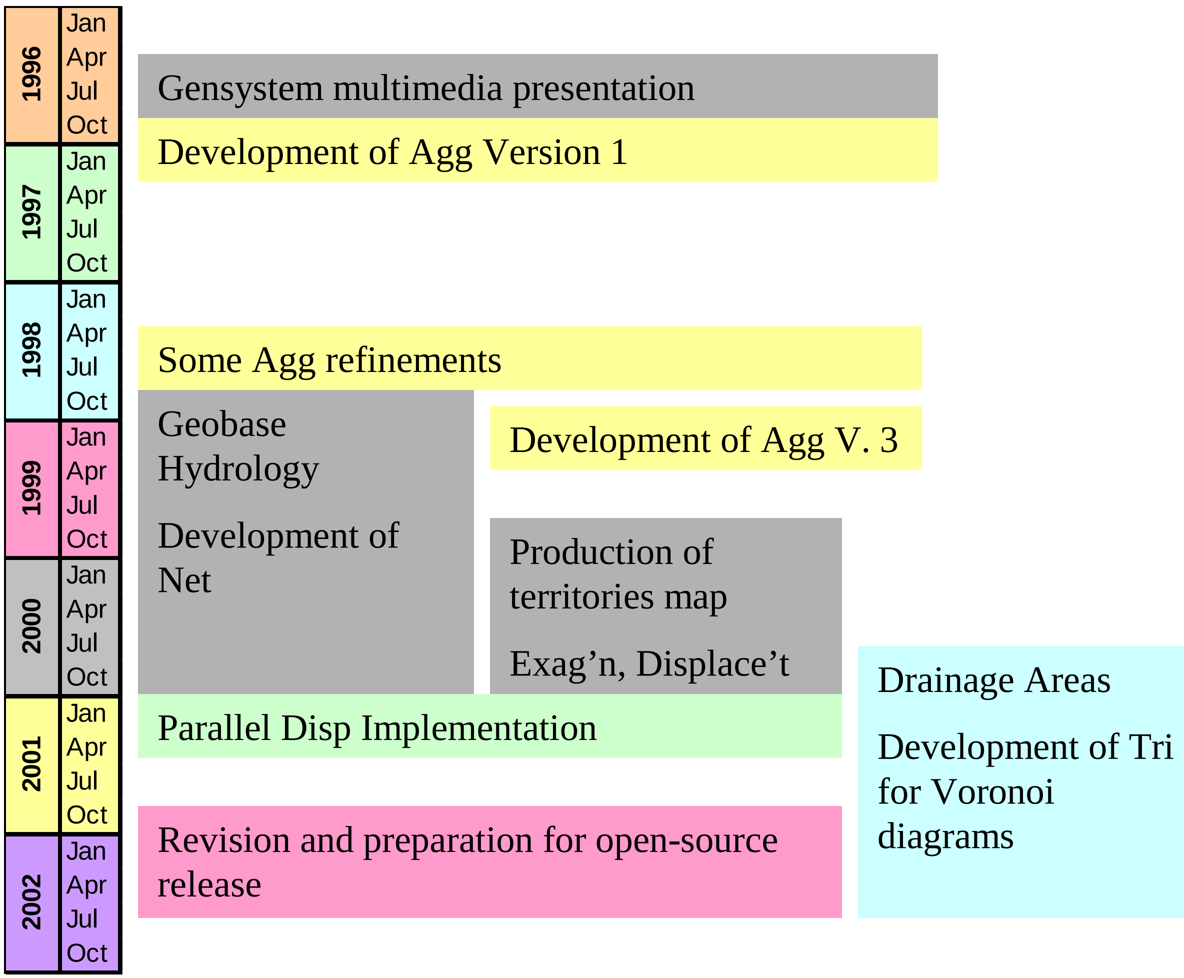
Acknowledgements

- Dianne Richardson developed Gensystem, which Net is partially based on.
- Perceptual strokes based on work by Robert Thomson and Dianne Richardson
- Lesley Chorlton, Catherine Dussault, Mireille Bruel investigated the application of Agg.
- Jorg Sack, Doron Nussbaum and Ji Chen worked on parallelizing Disp
- Rhian Evans built much of the Geobase hydrology
- Ken Arsenault and Terry Williams put up with some really bad cartography as the first map was produced
- Chris Gold gave me the hints I needed to do an effective Voronoi of line segments

Intro

- Atlas of Canada has a requirement for generating maps at various small scales from detailed data
- Naturally, this has led to the development of procedures and software to both build structured data, and to generalize it.
- Until recently, the tools were proprietary, and had many “idiosyncrasies”

History


1996
1997
1998
1999
2000
2001
2002
Jan
Apr
Jul
Oct
Gensystem multimedia presentation
Development of Agg Version 1
Some Agg refinements
Geobase Hydrology
Development of Net
Development of Agg V. 3
Production of territories map
Exag'n, Displace't
Drainage Areas
Development of Tri for Voronoi diagrams
Parallel Disp Implementation
Revision and preparation for open-source release

The Problem

- Tools were working..... But flawed
  - very project specific
  - Intertwined
  - Complex dependencies on external software and specific operating system tools
  - Poorly documented
  - Few resources to maintain or improve them

Open Source Release

- Rewriting the tools, and releasing them to the community under an open-source license.
  - Why?
  - We need the tools, but don't have the resources to build and maintain them in isolation
  - Our interest (and mandate) is in the results of using these tools, not in creating or marketing them
  - Hope is that others will extend, adopt or replace with better

Licensing

- A custom NRCan license
- Basically,
  - Copyright remains with NRCan
  - Do whatever you want with it (except claim you wrote it), no fees, no royalties
  - No liability whatsoever

Design Problem

- Previously, getting the software to work at all was considered success.
- For this release, issues of maintenance, documentation, and usability had to be addressed.
- Collection of tools was broken down into four parts, and dependencies on other software were reduced where possible

Design

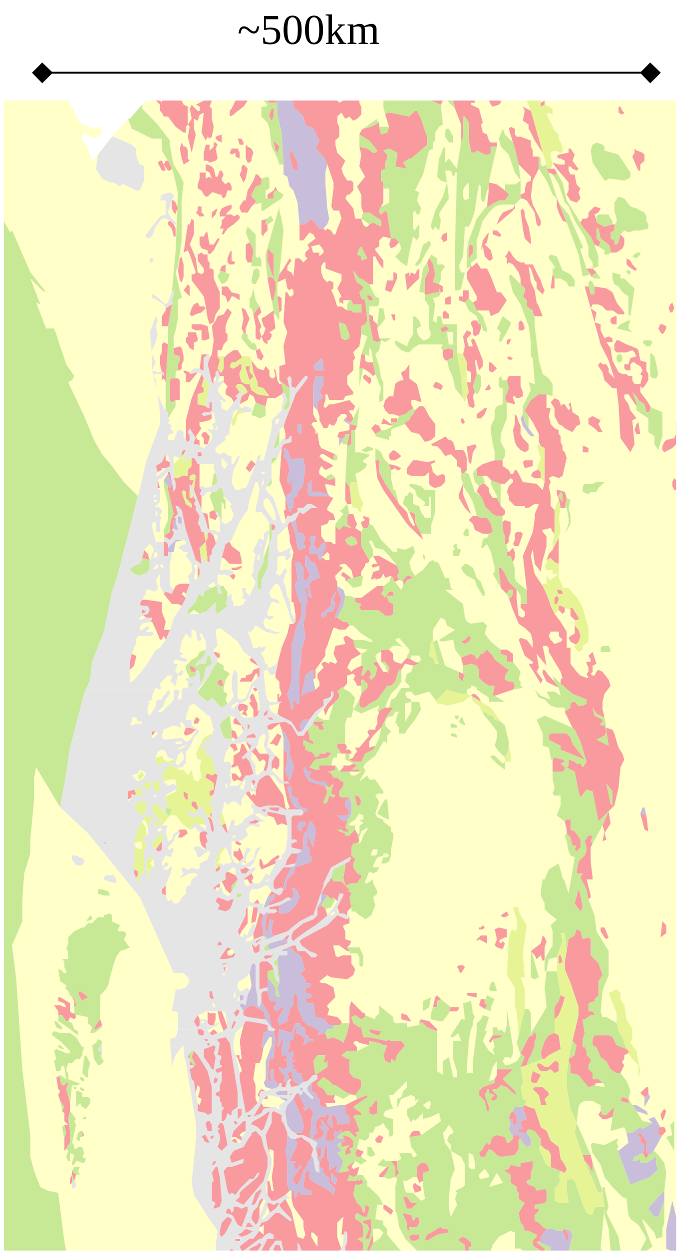
- Minimalism
  - If possible, standard tools and libraries were used
  - Superfluous parts (eg Guis) thrown away
- Focus
  - Tools were divided into logical units that did one task, or a set of related tasks
- Reduce dependencies
  - Instead of using, for example, code in C, Awk, Perl and AML, pick a language and stick with it
- Use standards
  - Whereever possible, well known formats (eg SHP) and well known libraries (eg Triangle) would be used.

Four Applications

- Agg – generalisation of categorical data and areas
  - Runs in Arc/Info Workstation
- Net – Network analysis and generalisation
  - Runs on any Perl platform
  - Needs topological graph as input
- Tri – Triangulations and Voronoi diagrams
  - Ansi C – should compile nearly anywhere
  - Accepts shapefiles as input
- Disp – Displacement
  - Perl and Arc/Info workstation, but being extended at Carleton University.

Agg

- Requires: Arc/Info Workstation
- Primarily a tool which uses buffering to generalize categorical data
- Can also be used to generalize the shape of area features
- Was extended to allow the maintenance of square corners



Original

1 : metamorphic rocks

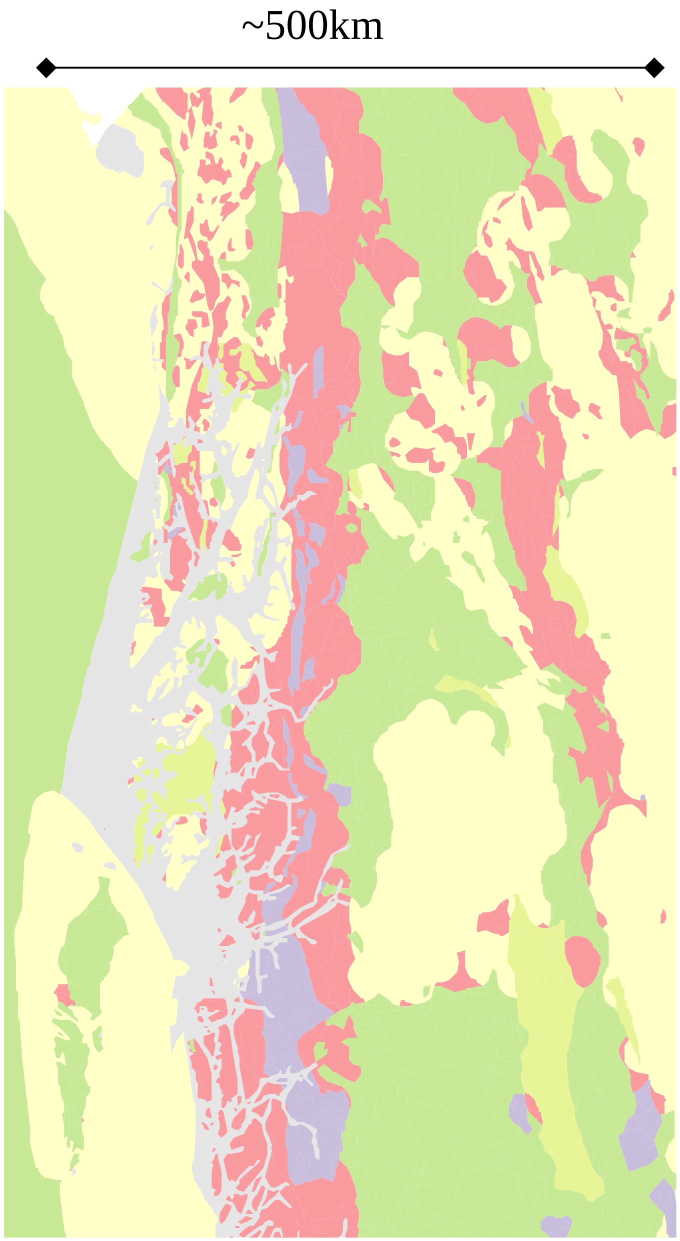
2 : sedimentary and volcanic rocks

3 : volcanic rocks

4 : intrusive rocks

5 : sedimentary rocks

6 : unknown



Result

- 1 : metamorphic rocks
- 2 : sedimentary and volcanic rocks
- 3 : volcanic rocks
- 4 : intrusive rocks
- 5 : sedimentary rocks
- 6 : unknown

Agg – Other Apps

- Generalization of Area features
- We often treat coastlines as a line generalisation problem – but the sidedness of the line is rarely considered in line generalisation
- Nevertheless, one would never generalize a fjord the same way as a peninsula

Cartographic Area Generalization

Islands, the mainland and lakes are *area* features and are generalized together.

Note that river connections are maintained at point A

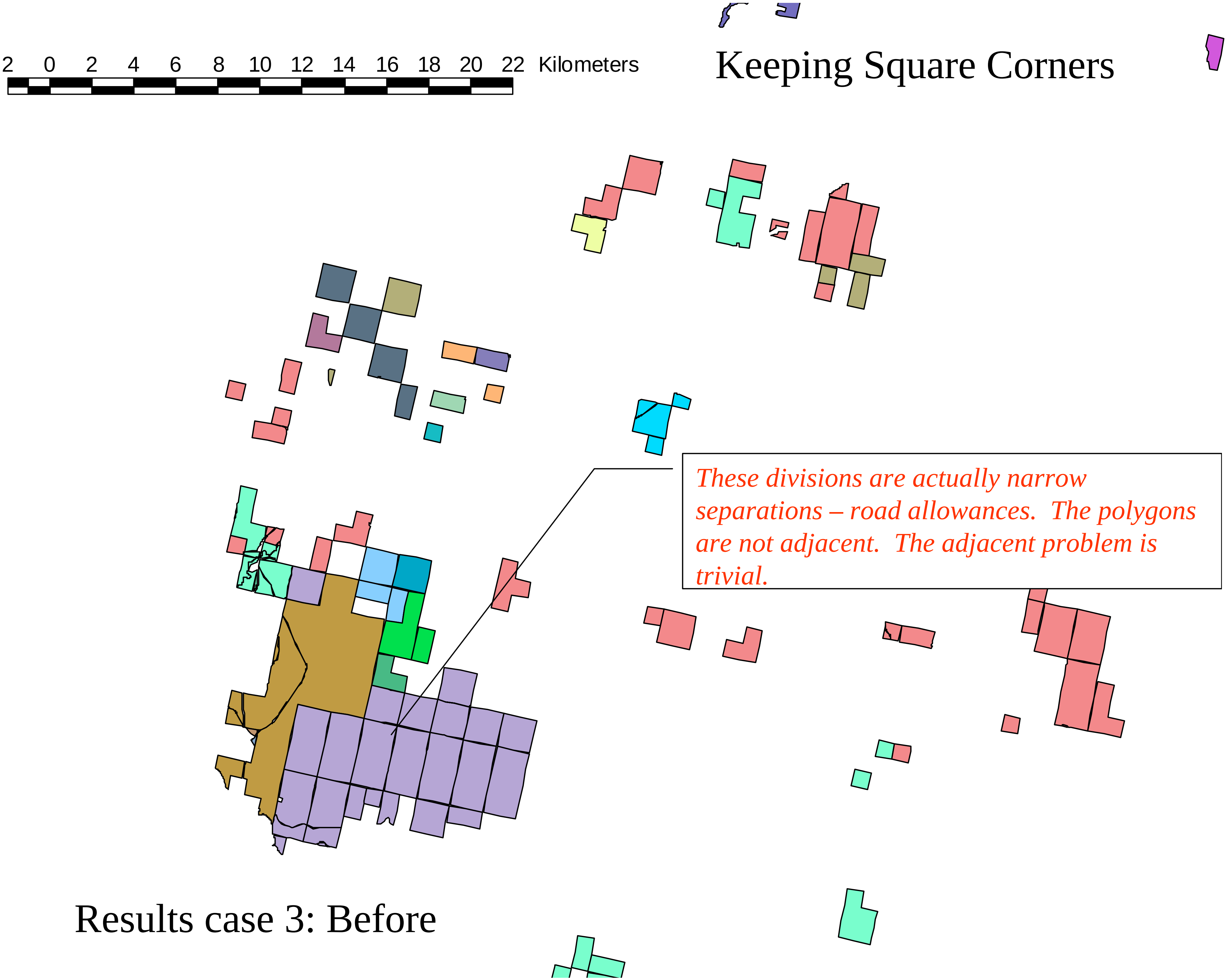
The National Atlas of Canada
L'Atlas national du Canada

Cartographic Area Generalization

Islands, the mainland and lakes are *area* features and are generalized together.

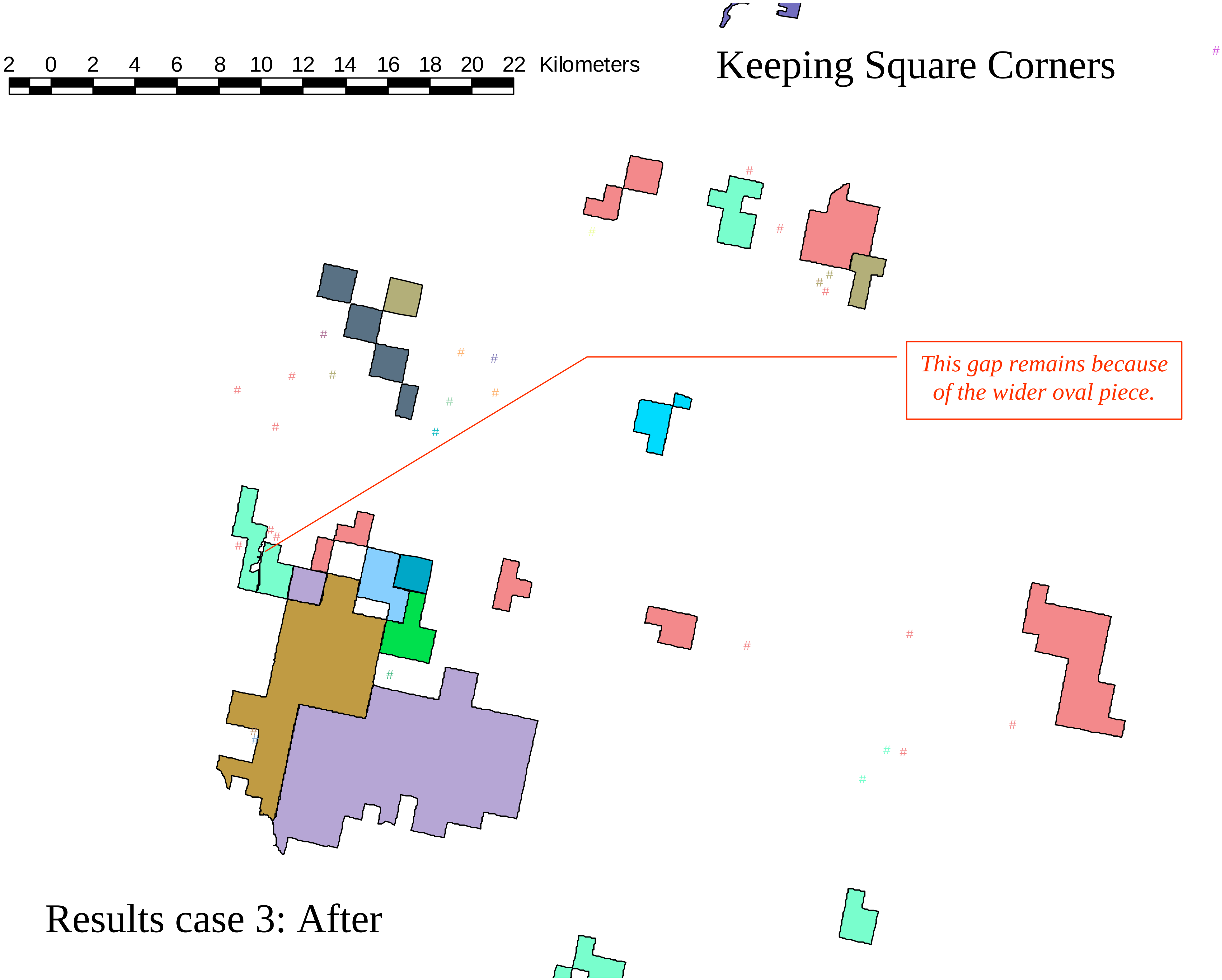
Note that river connections are maintained at point A

Keeping Square Corners

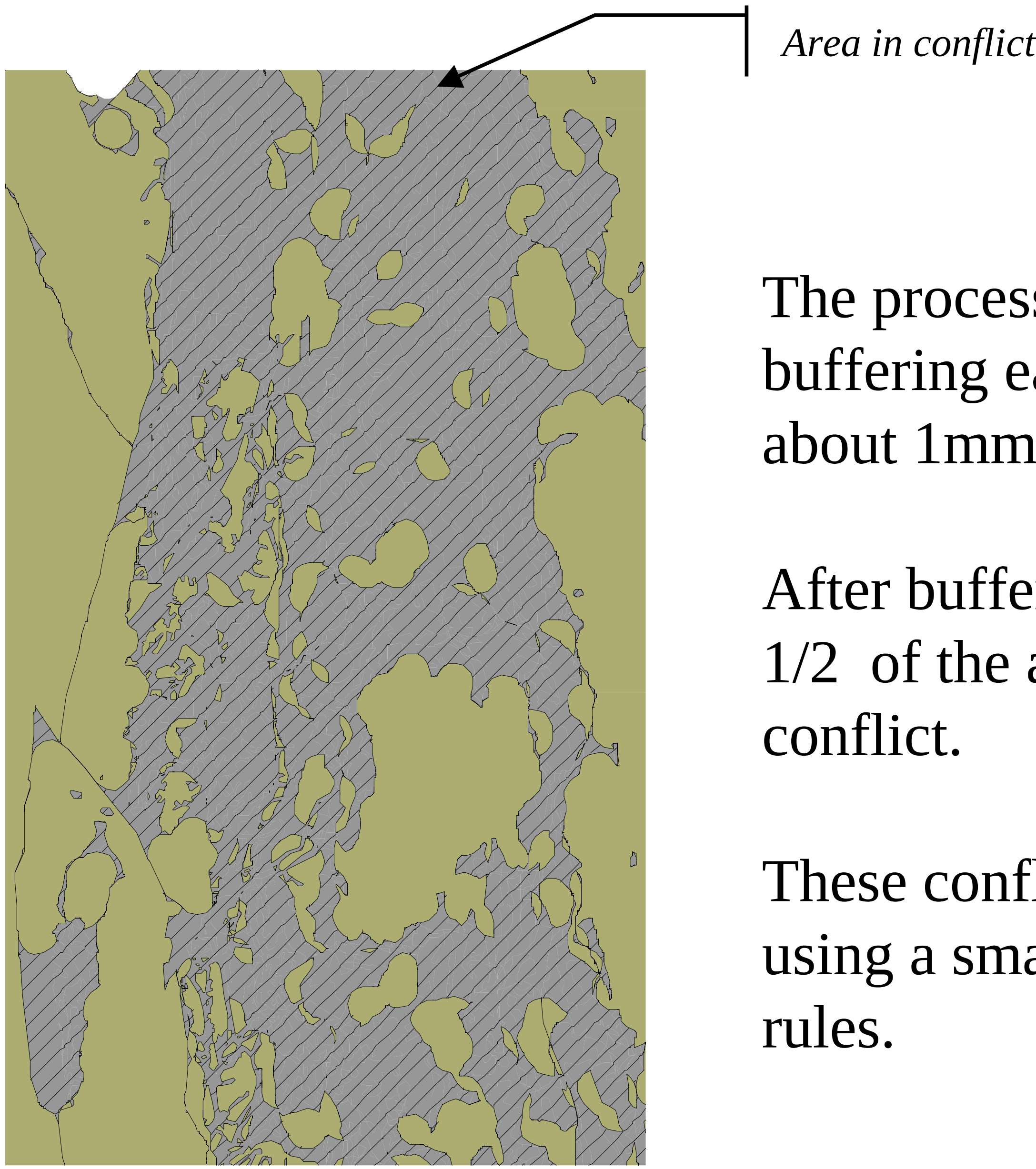


Results case 3: Before

Keeping Square Corners



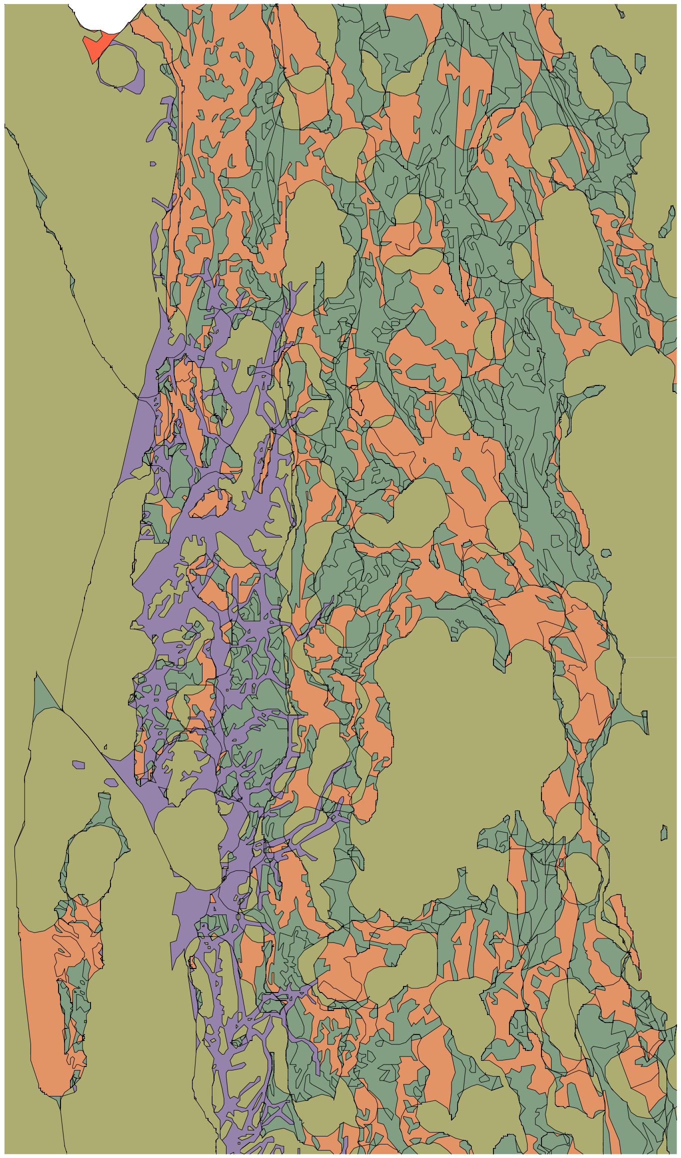
Results case 3: After



The process begins by buffering each class by about 1mm at map scale

After buffering, more than 1/2 of the area is in conflict.

These conflicts are resolved using a small, simple set of rules.



Rules

- Outside Edge
- NOSUB
- Only one choice
- Priority
- Islands

The various rules are applied to resolve the conflicts

Net

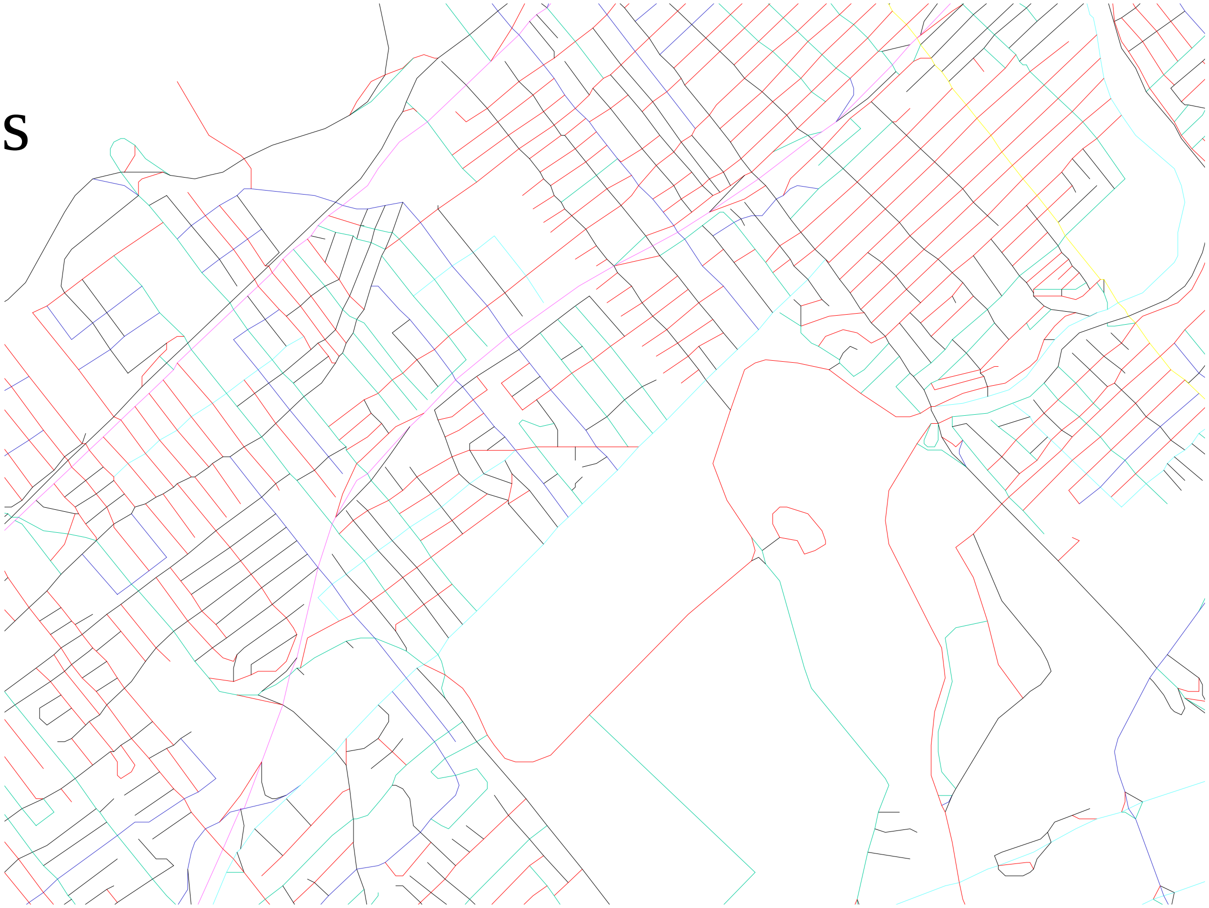
- Requires: Perl, and the ability to create a topological graph in Xbase (dbf) files

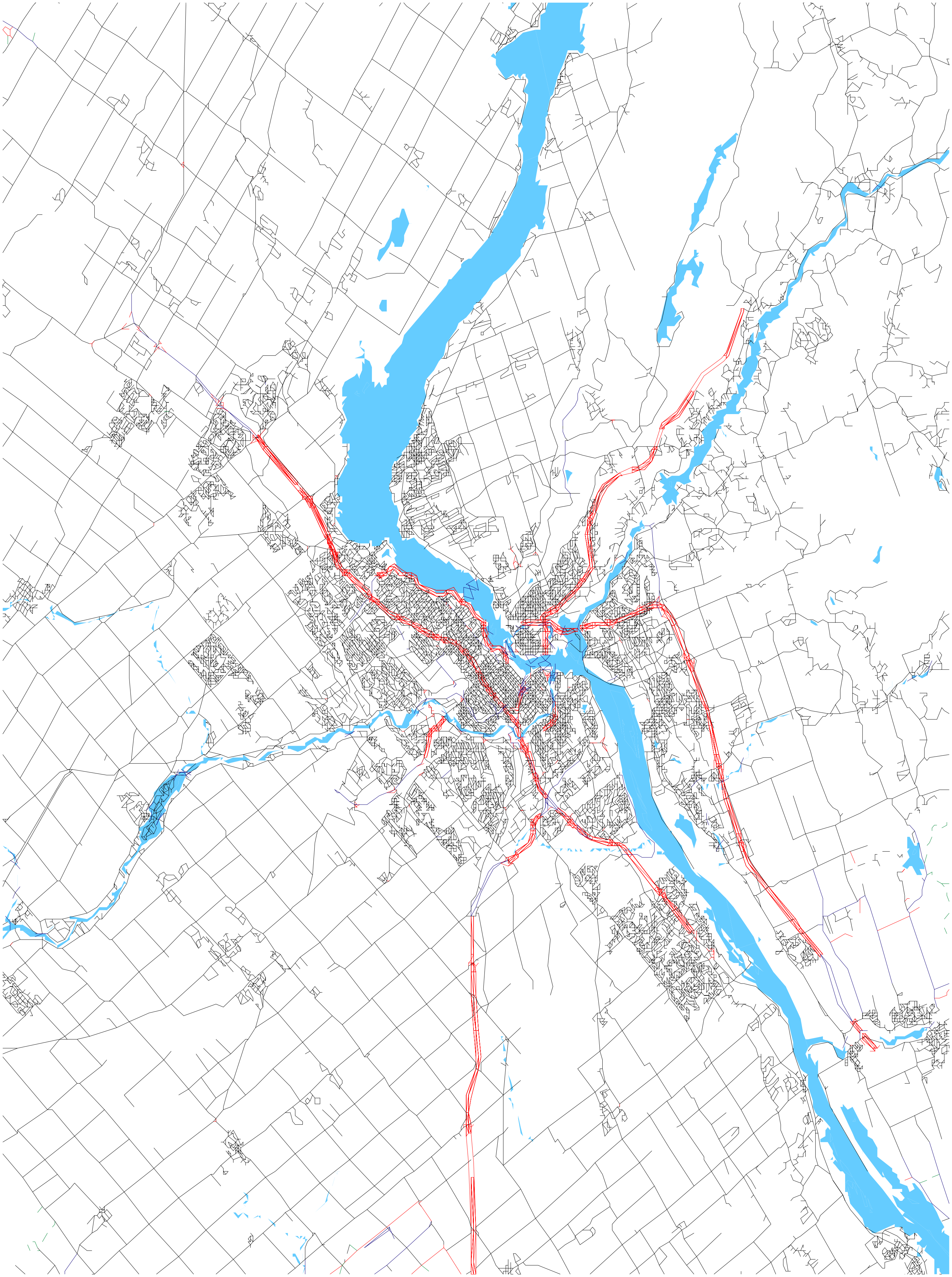
- Net consists of a set of small command line tools

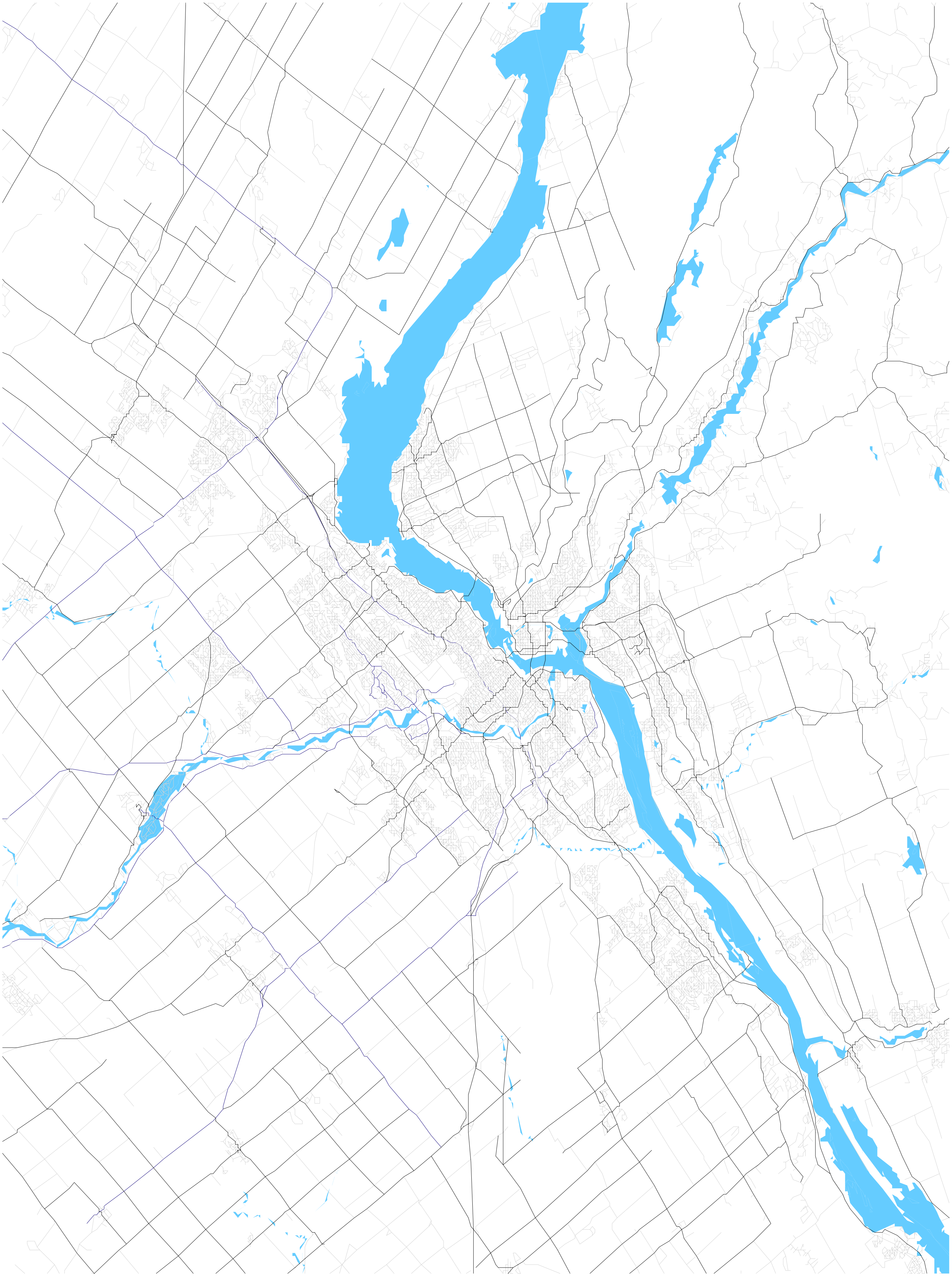
  - Net_color
  - Net_classify
  - Net_hydro
  - Net_genhydro
  - Net_describe
  - Net_genroads
  - Net_strokes
  - Net_color
  - Net_write

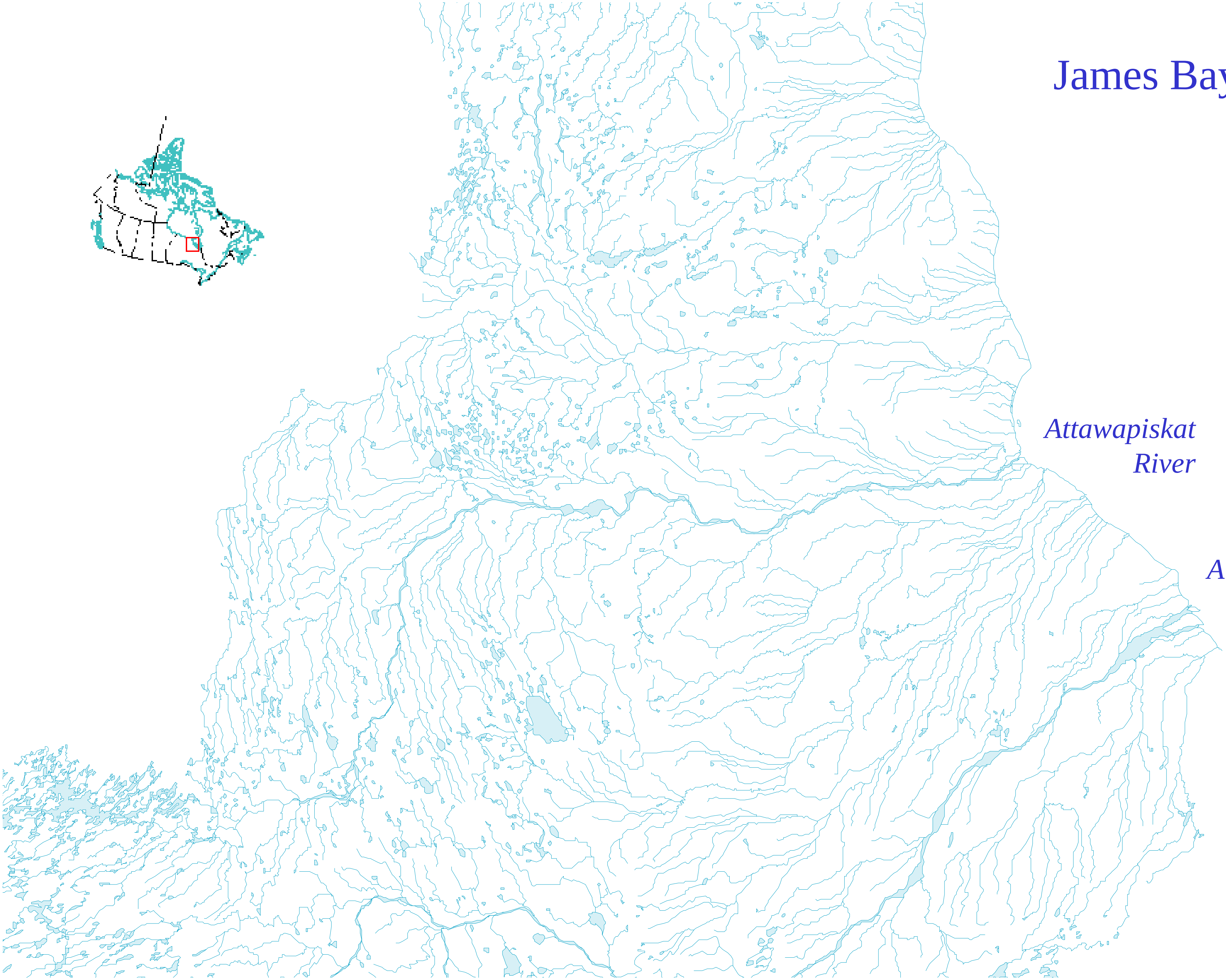
Strokes - Transportation

- Intersections
- Loops









James Bay
Attawapiskat River
Albany River


James Bay
Attawapiskat River
Albany River

The National Atlas of Canada
L'Atlas national du Canada

Net – How it works

Export a graph as Xbase (DBF) files

Net_read.pl
Read it into the Net format
(Perl data structure stored in Sleepycat)

Perform some kind of processing on it

Net_classify.pl
Classify the feature codes into codes net understands using a text lookup table

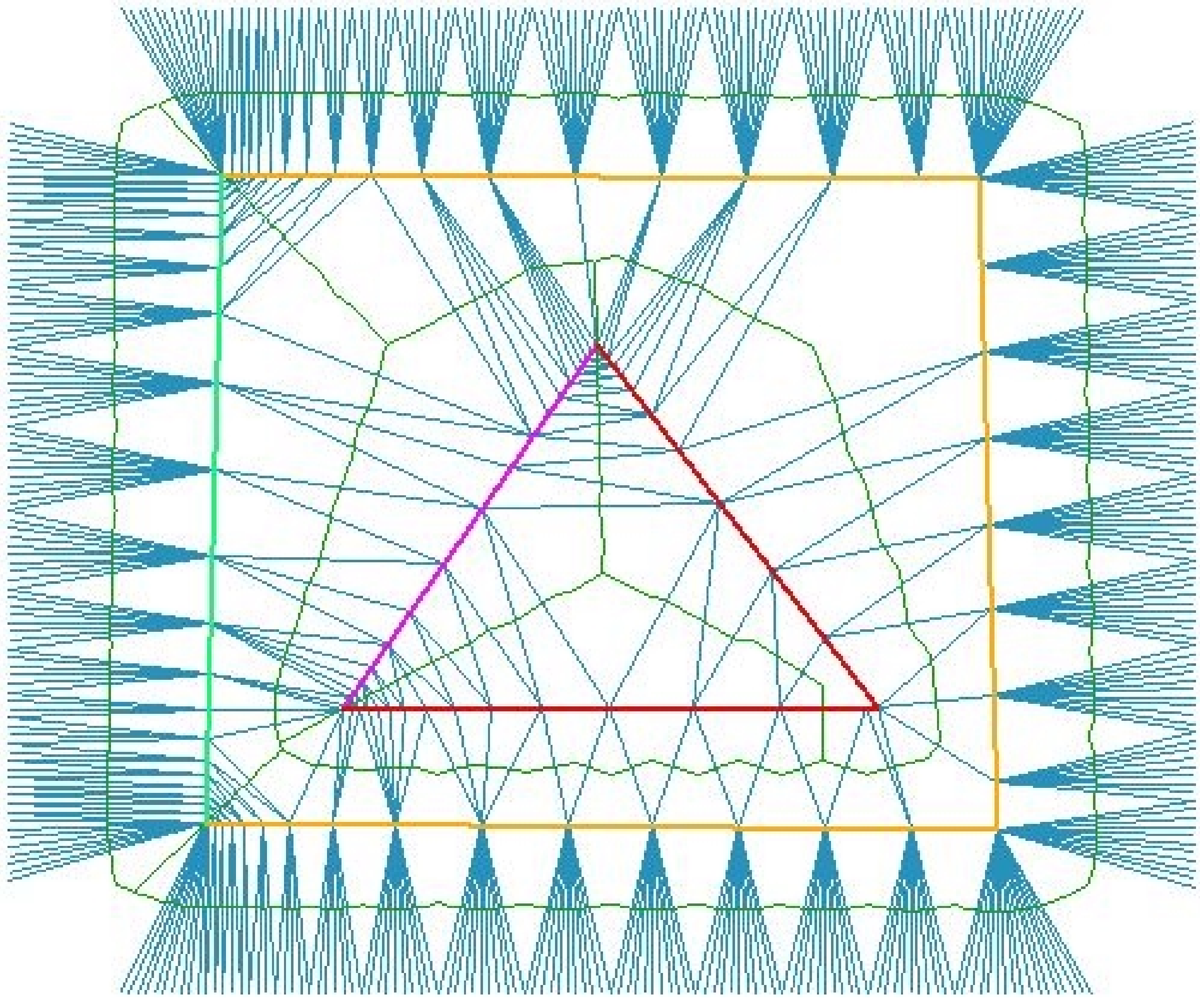
Net_hydro.pl
Use hydrologic analysis to determine directionality, stream order, etc

Net_write.pl
Export the result
(as text, CSV or DBF)

Tri

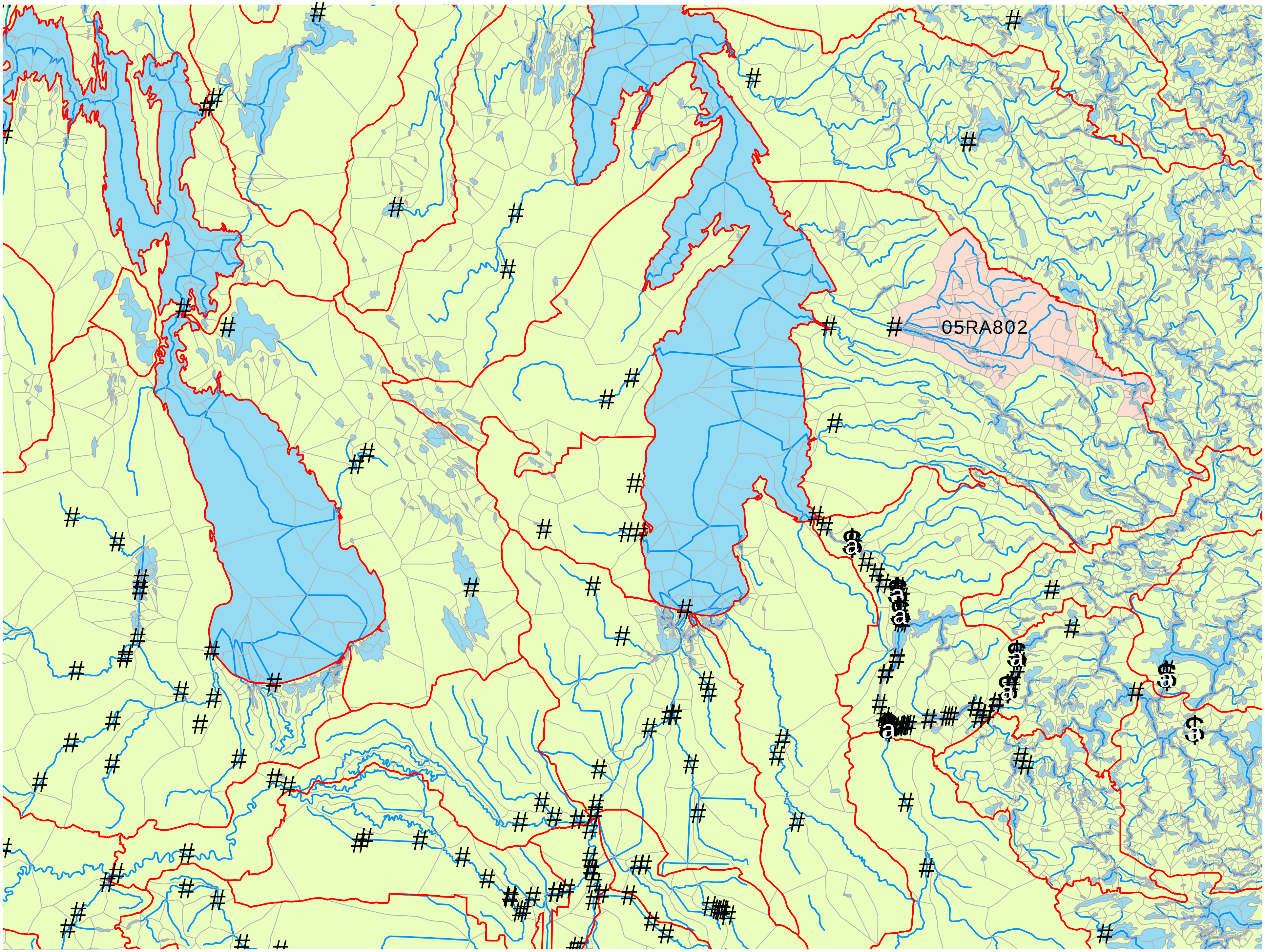
- Requires: Any Ansi C compilation environment. Windows binaries available

- Takes shapefiles as input and generates triangulations / voronoi diagrams using the Triangle library.
- We can then generate approximate Voronoi diagrams of line segments, skeletons, and from the skeleton it is possible to do exaggeration



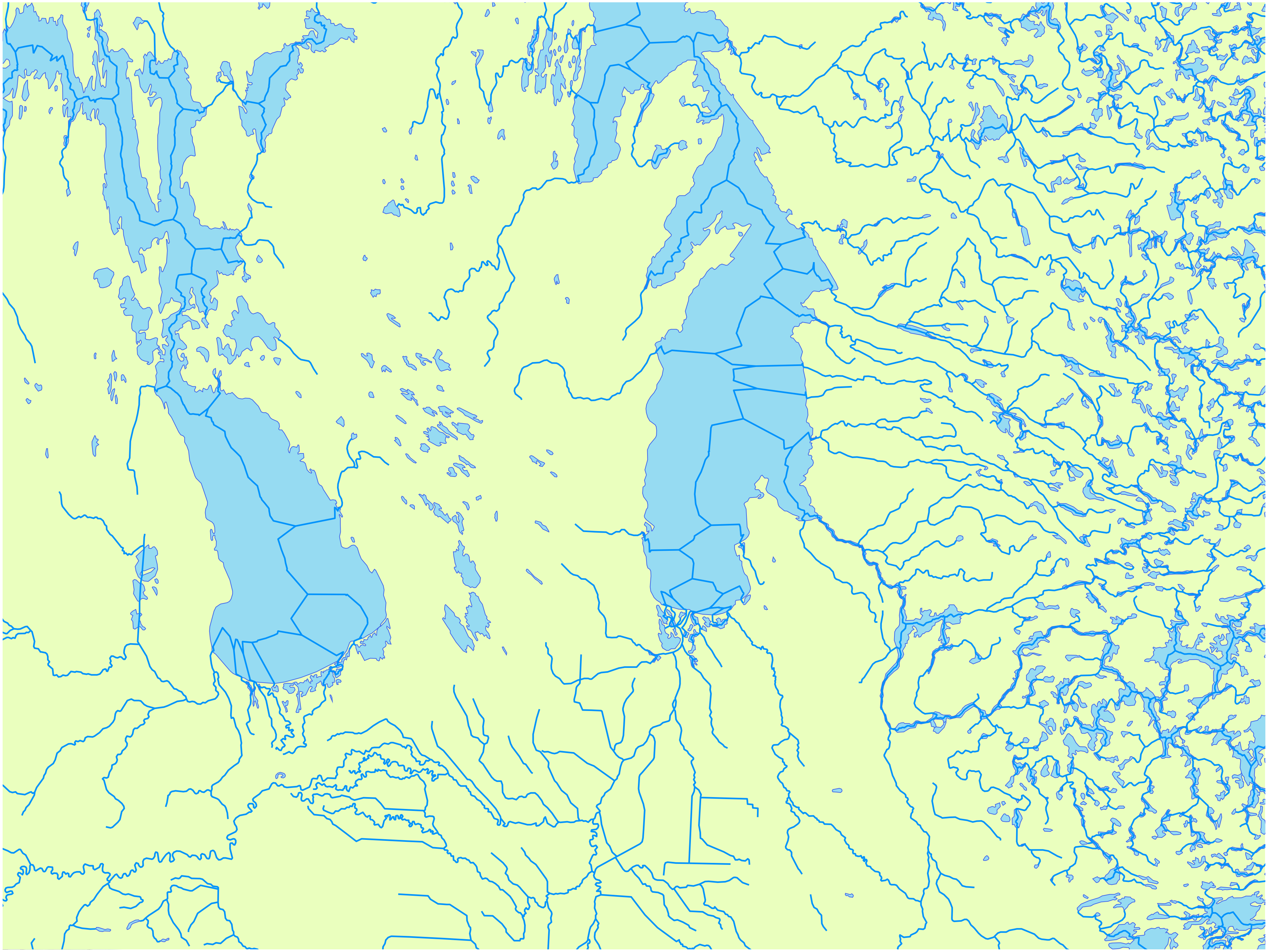
GeoAccess Division
CANADA CENTRE FOR REMOTE SENSING

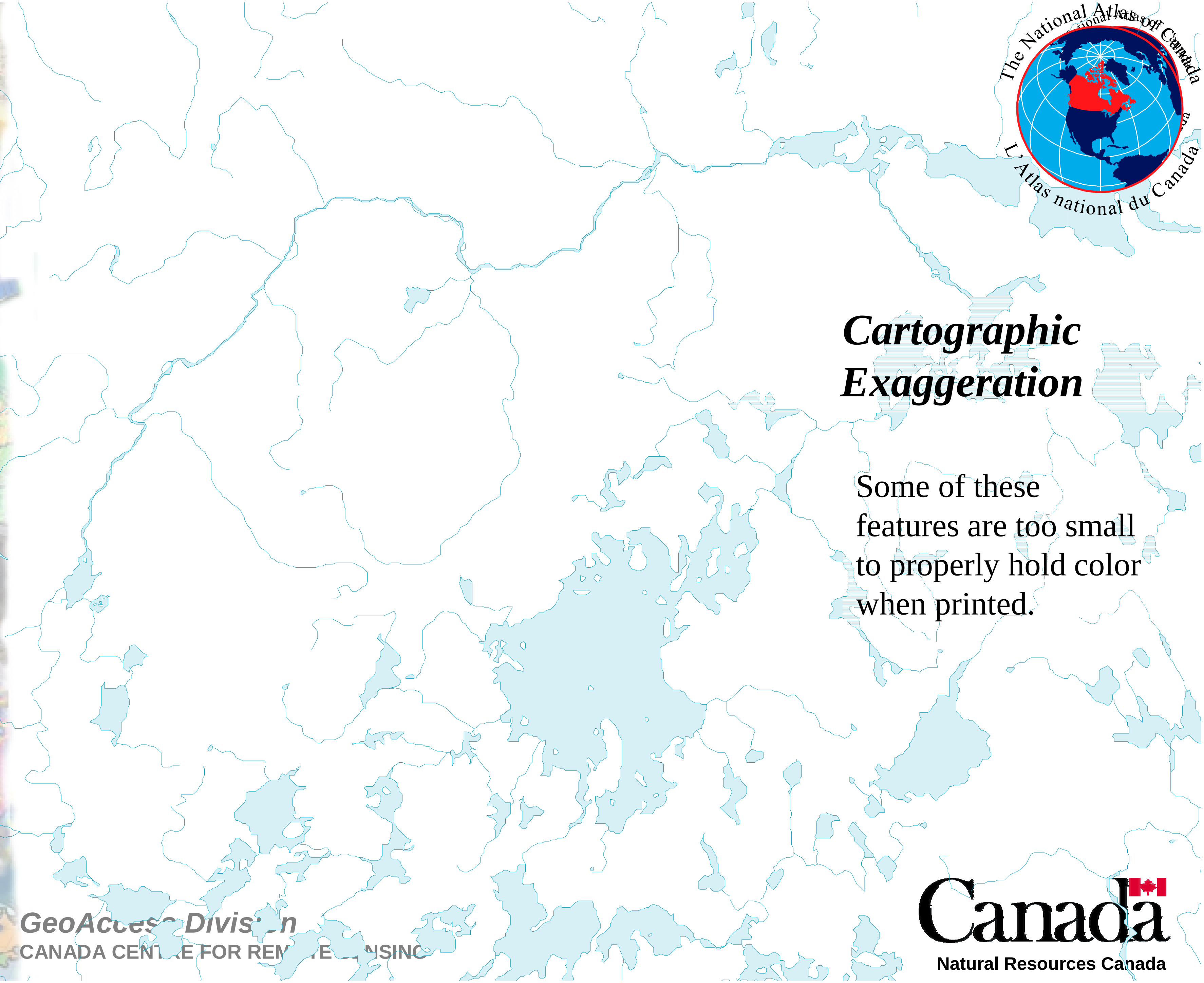
Natural Resources Canada


05RA802

Skeletons

- The Geobase hydrology required the generation of a skeleton to continue the network through lakes
- In that project we used a skeletonization program by Robert Thomson, which is based on the triangulation
- A smoother skeleton may be created from the Voronoi diagram of the elements in the boundary of the polygon



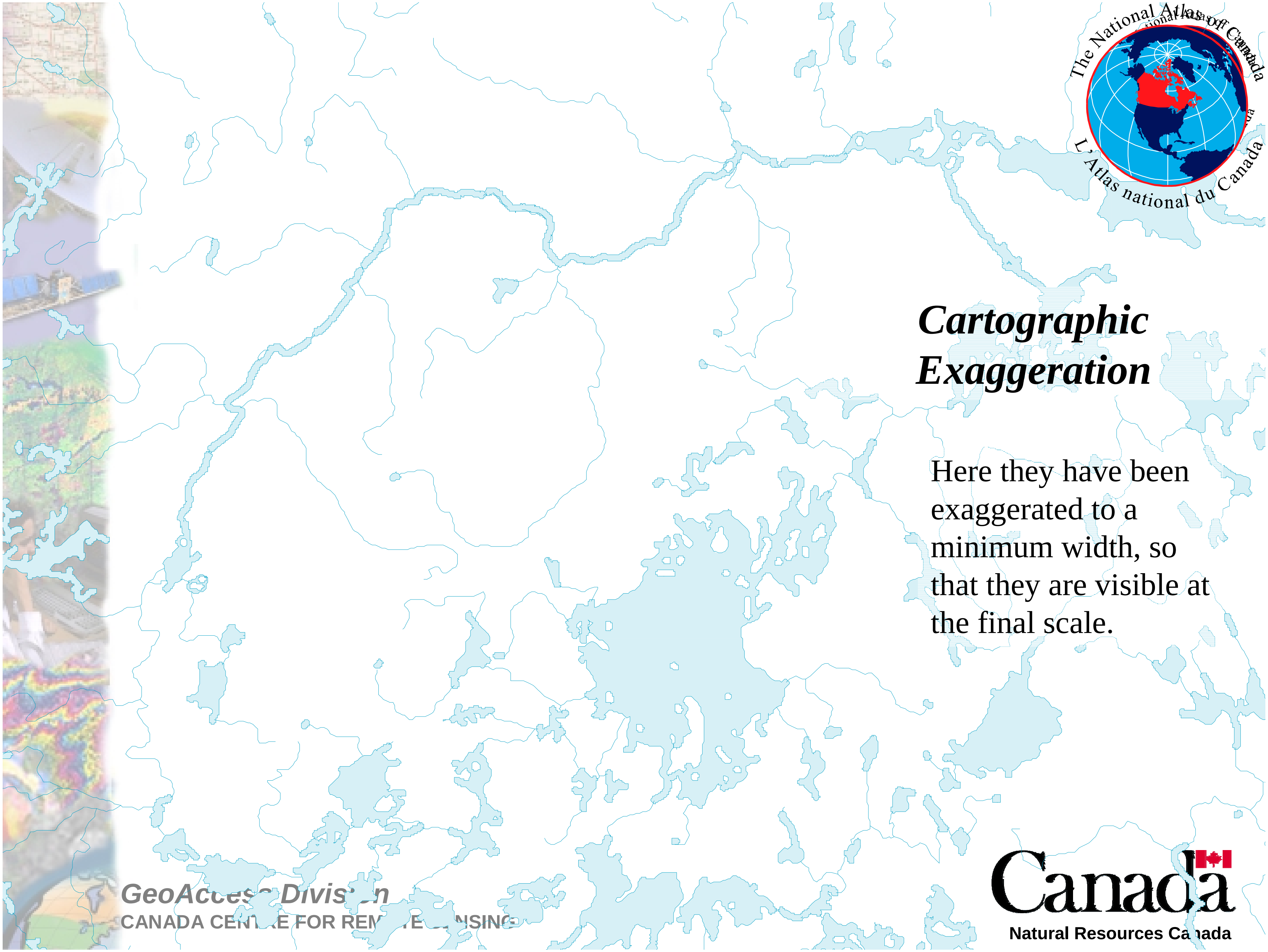


Cartographic Exaggeration

Some of these features are too small to properly hold color when printed.

Cartographic Exaggeration

Here they have been exaggerated to a minimum width, so that they are visible at the final scale.



Tri – how it works

- Basically just links together two existing, very useful libraries
  - Triangle, by Jonathan Shewchuk
  - Shapelib, by Frank Warmerdam
- Creates an approximation of the voronoi diagram of line segments by densifying vertices
- A reasonable skeleton of a feature may be created from the Voronoi diagram of the elements in its outline
- Cartographic exaggeration is possible by buffering the skeleton of a feature and unioning the result with the original feature.

Disp

- Requires: Arc/Info workstation and Perl
- For displacement of rigid small polygons
- Uses a simple “reverse gravity” model
- Computationally intensive, but may be implemented on a parallel machine
- Jorg Sack and Ji Chen continue to work on it at Carleton University.

A Parallel Displacement Operator

The displacement operator separates these tightly packed features so they can be read

A Parallel Displacement Operator

The displacement operator separates these tightly packed features so they can be read

⇐ *Like this*

A Parallel Displacement Operator

The displacement operator separates these tightly packed features so they can be read

⇐ *Like this*

A Parallel Displacement Operator

The displacement operator separates these tightly packed features so they can be read

⇐ *Like this*

A Parallel Displacement Operator

The displacement operator separates these tightly packed features so they can be read

⇐ *Like this*

A Parallel Displacement Operator

The displacement operator separates these tightly packed features so they can be read

⇐ *Like this*

A Parallel Displacement Operator

The displacement operator separates these tightly packed features so they can be read

⇐ *Like this*

Disp – How it works

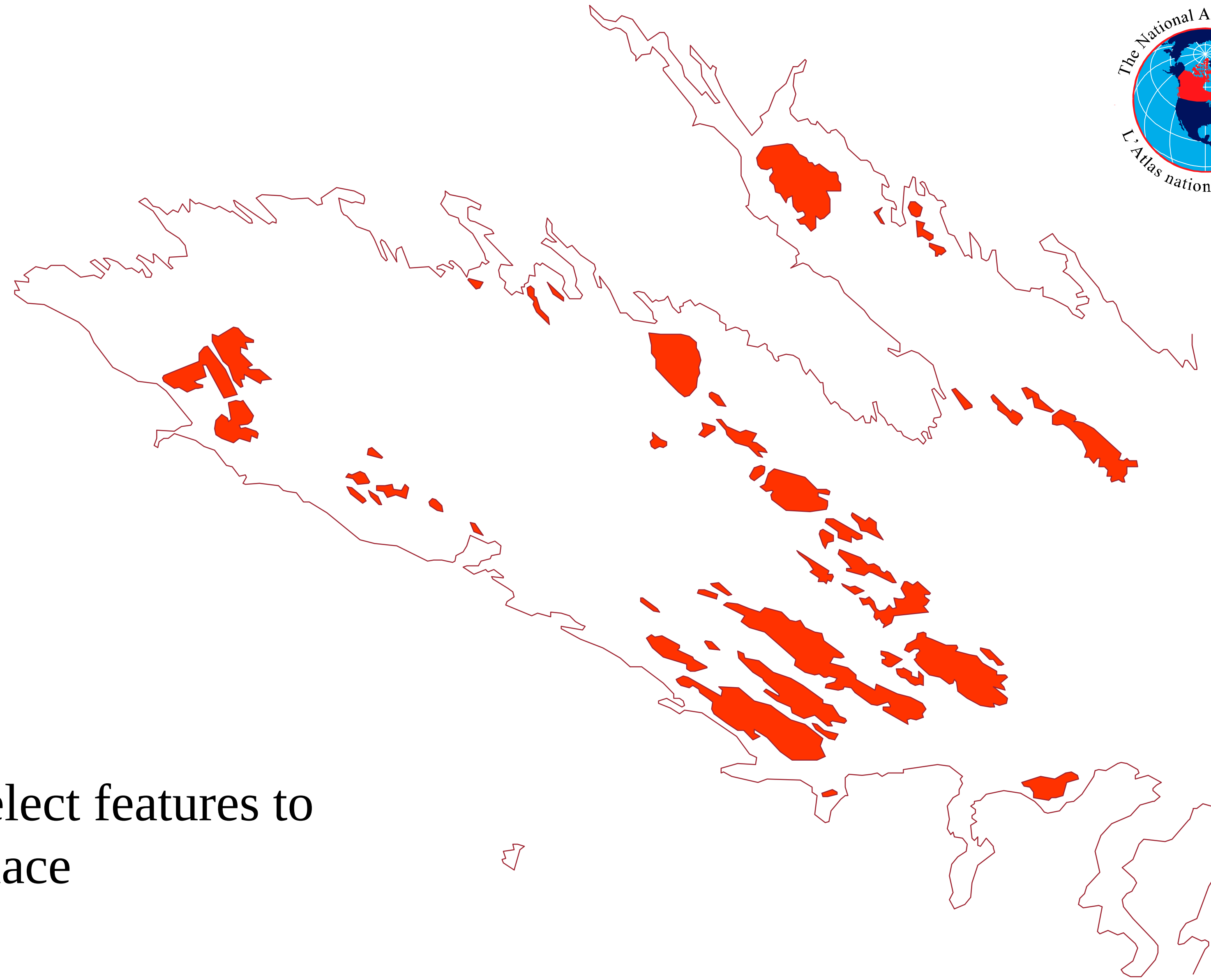
1. Select features which may be displaced
2. Buffer (width d) those features to identify interacting clusters
3. Determine distance and bearing between each object
   based on an average of distance and bearing between closest 5 vertices
4. Compute the force exerted by object A on B as:

$$\overline{F} = \frac{10^6 \cdot Area(A)}{d_{AB}^{\ 3}} \cdot \overline{e}$$

5. Sum Forces on each object
6. Compute the movement of each object as

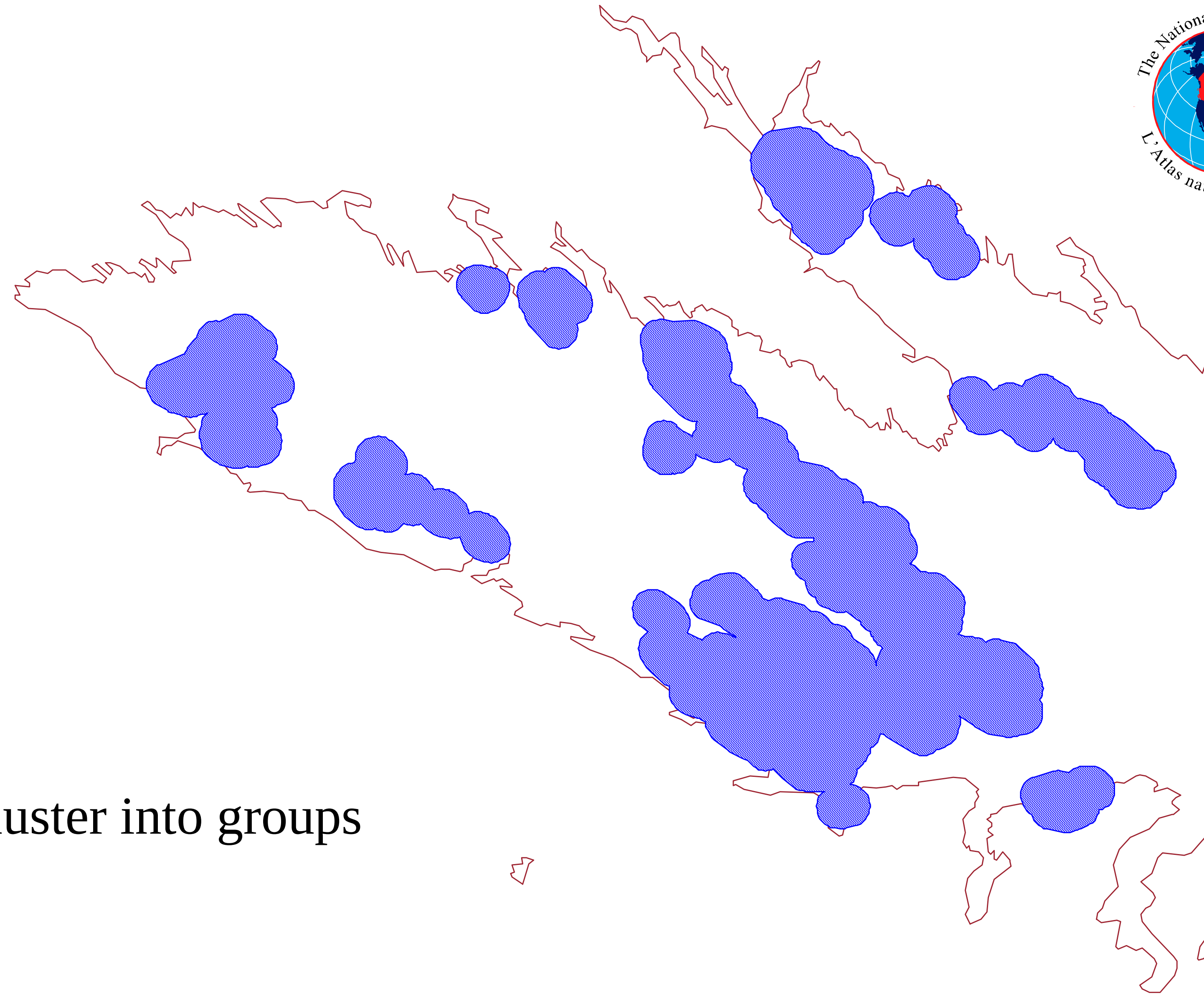
$$disp = \frac{\overline{F}}{\ln(Area(B))}$$

7. Move the objects. (Limit move to d/2)
8. Check for interference.
9. While interference exists
   9.1 Move one of the interfering objects to its original position
   9.2 Check for interference again



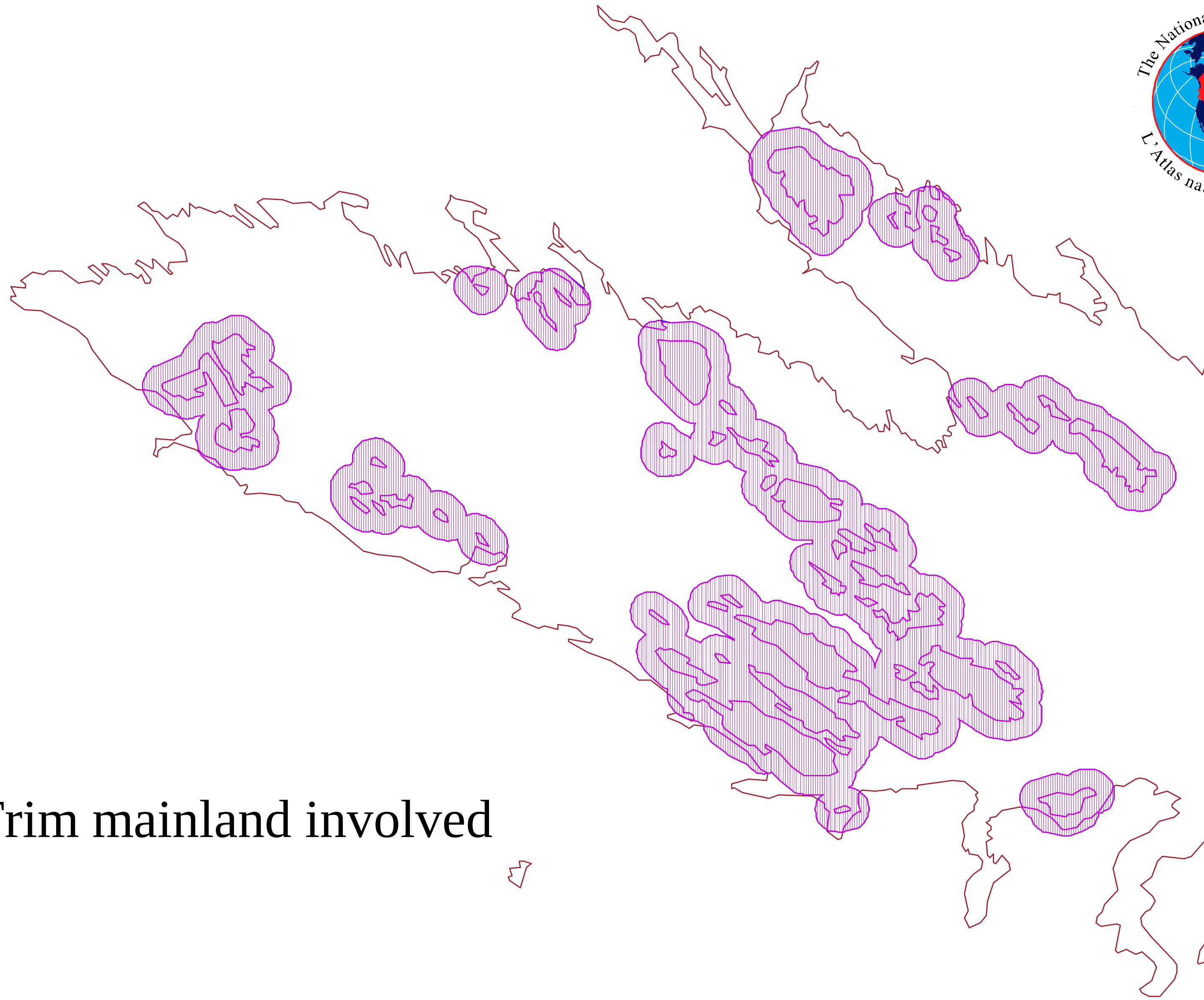
1. Select features to displace

The National Atlas of Canada
L'Atlas national du Canada



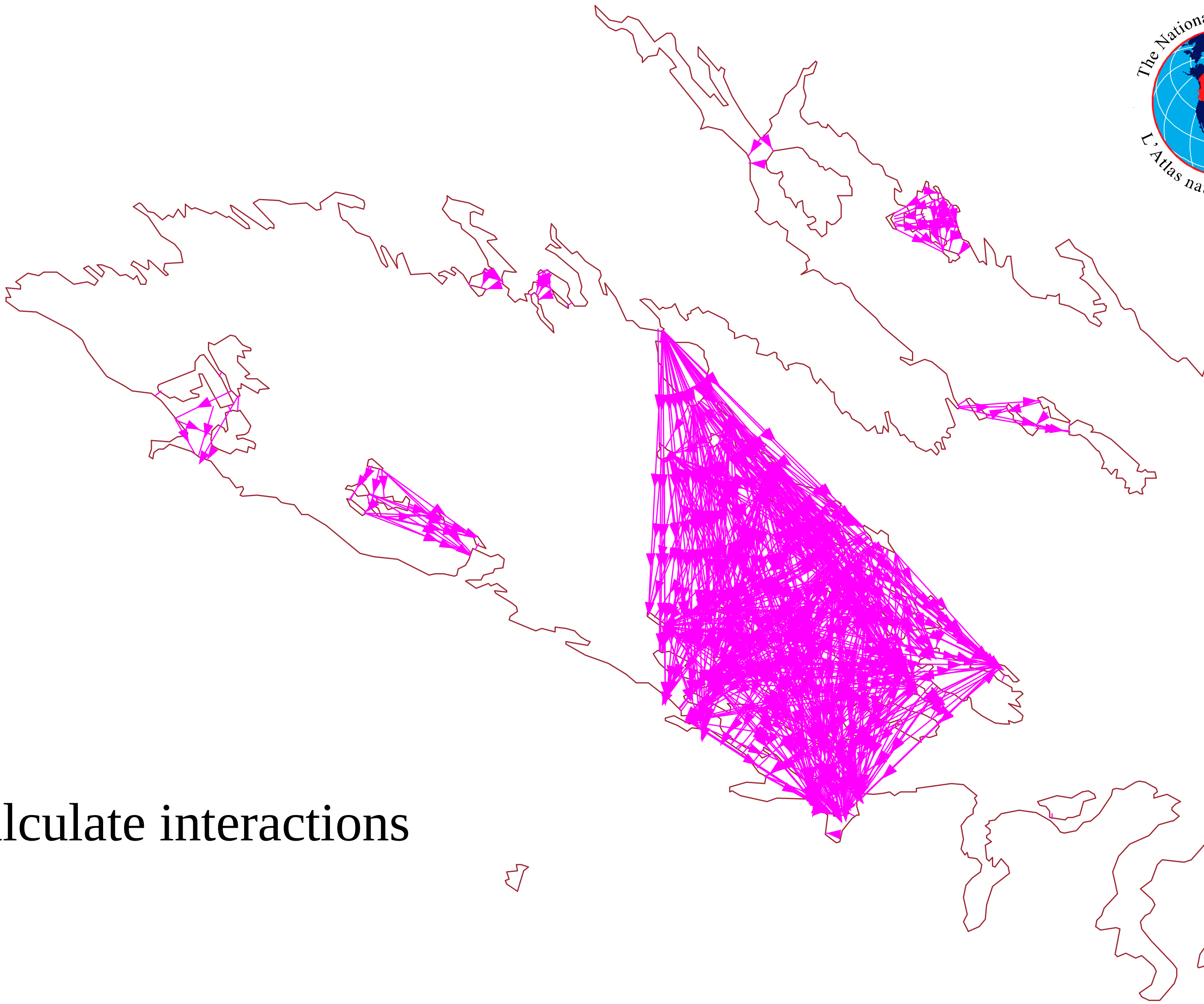
2. Cluster into groups

2a. Trim mainland involved



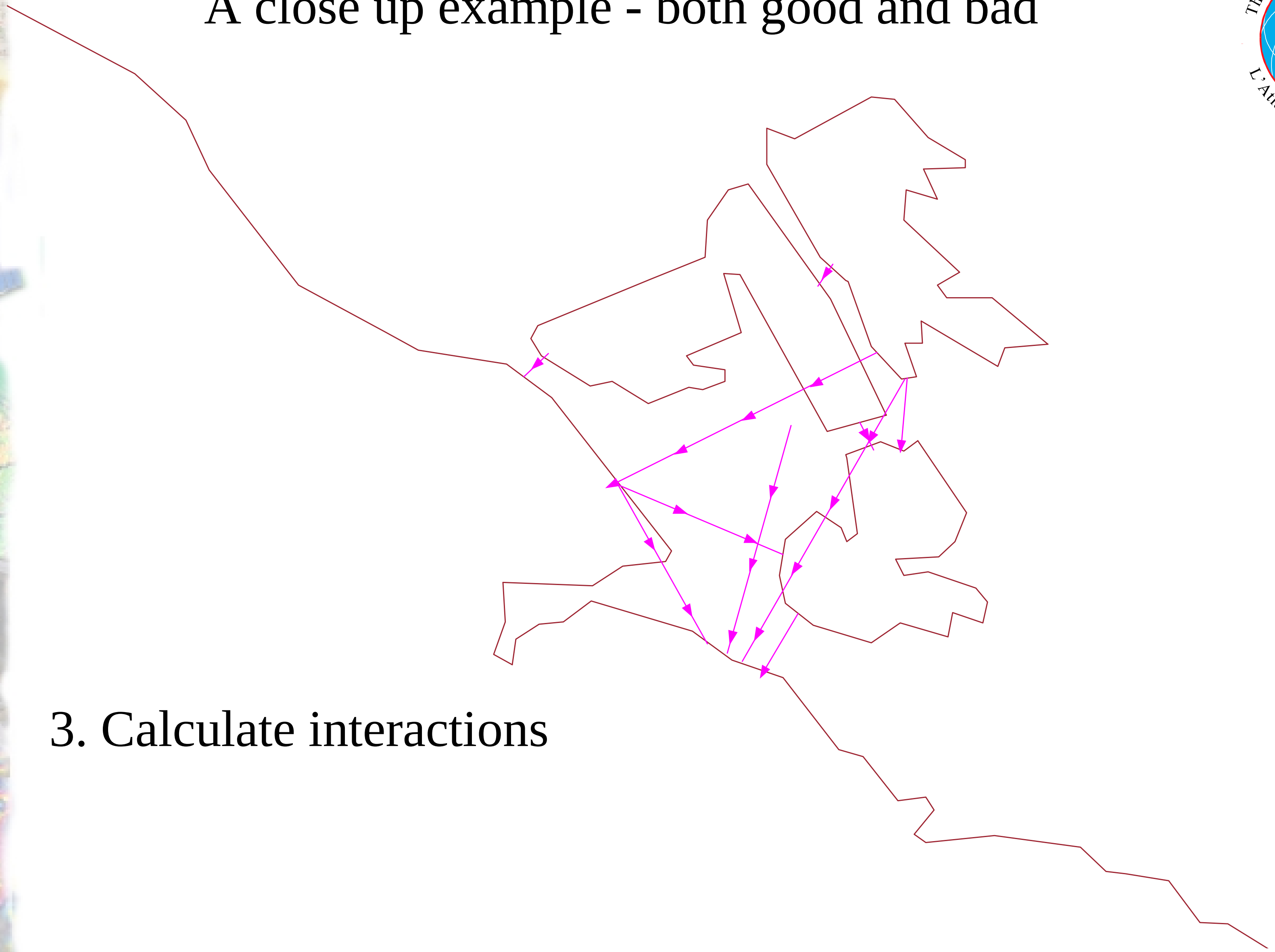
GeoAccess Division
CANADA CENTRE FOR REMOTE SENSING

Natural Resources Canada

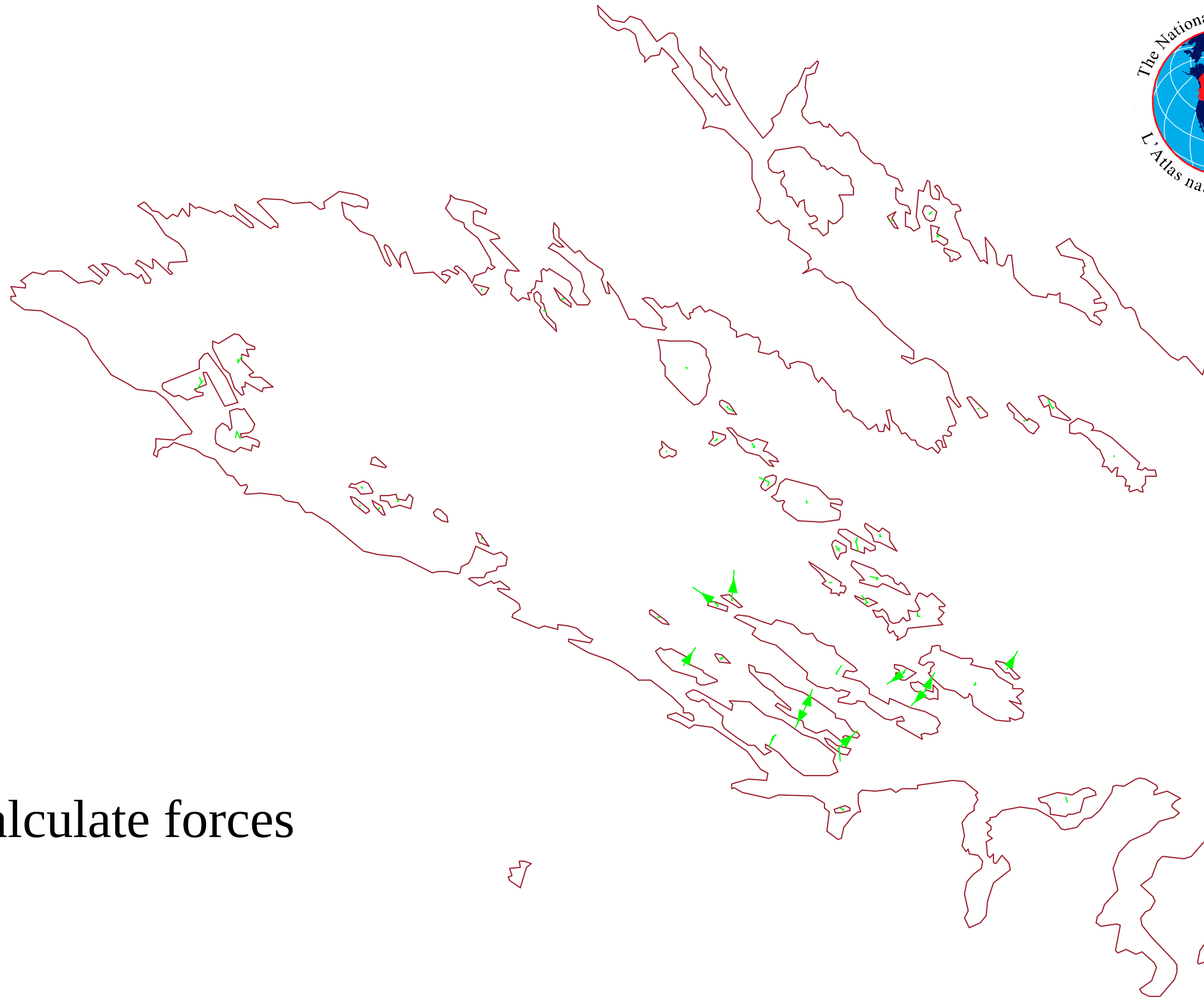


3. Calculate interactions

A close up example - both good and bad

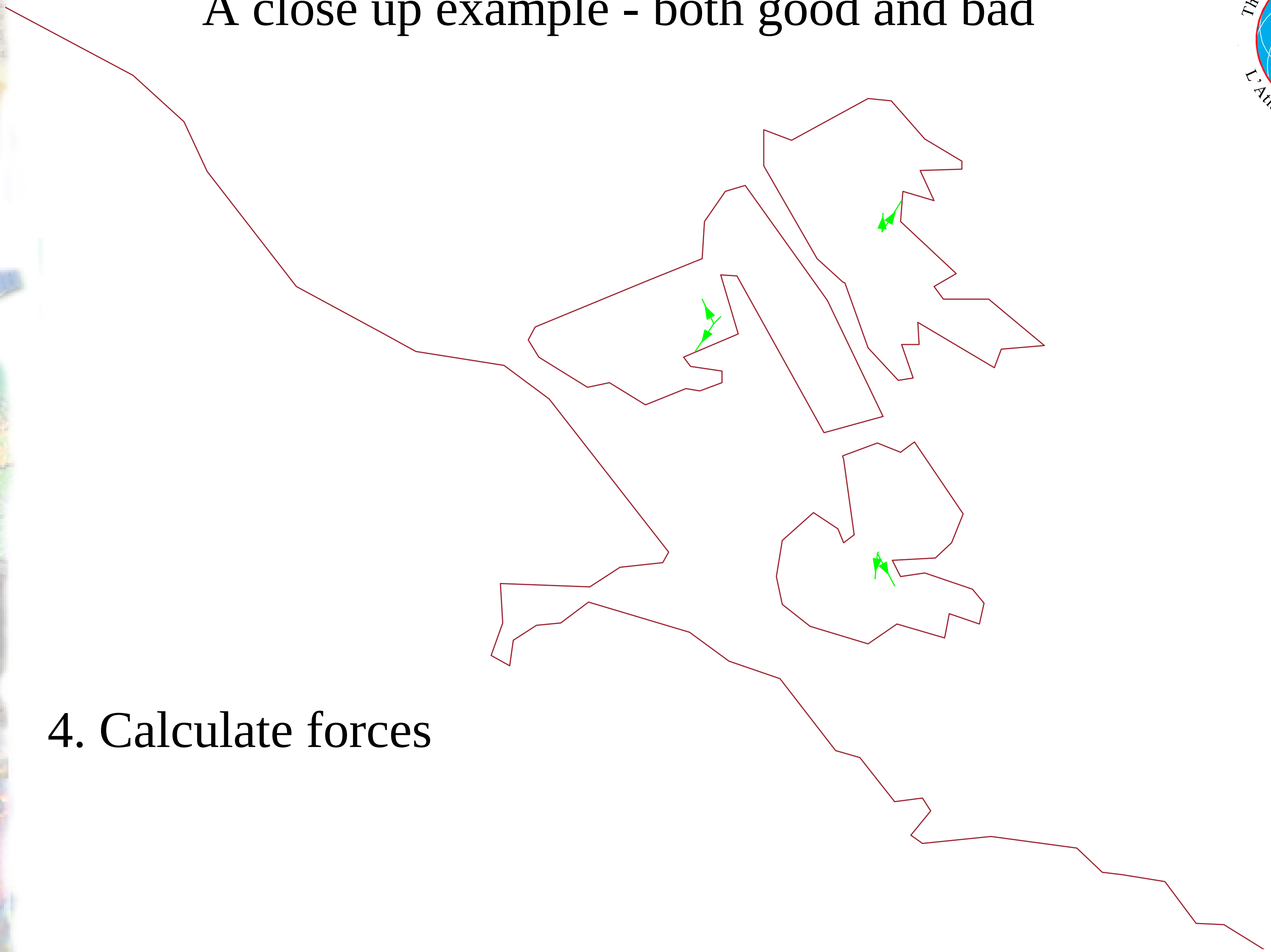


3. Calculate interactions

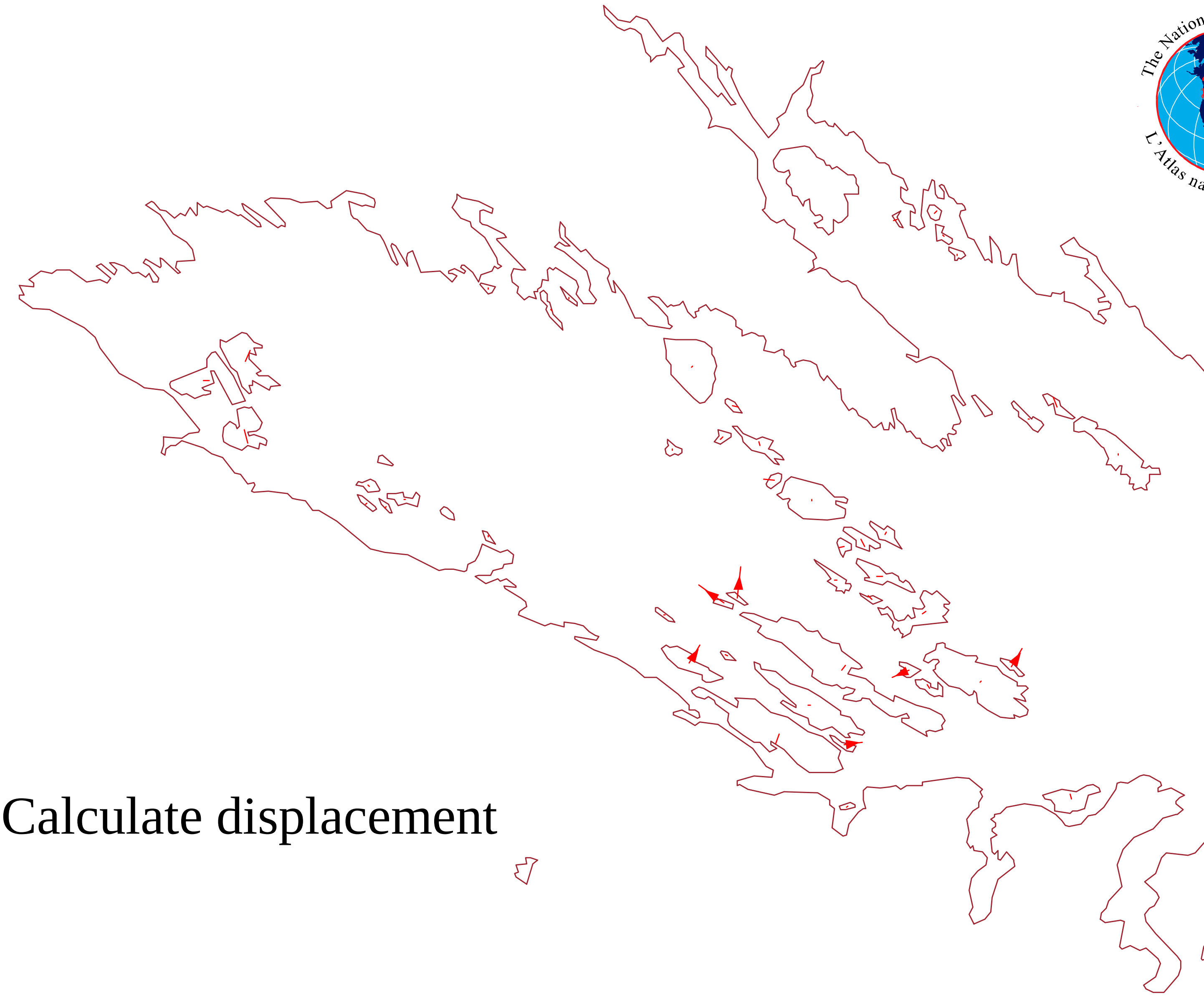


4. Calculate forces

A close up example - both good and bad

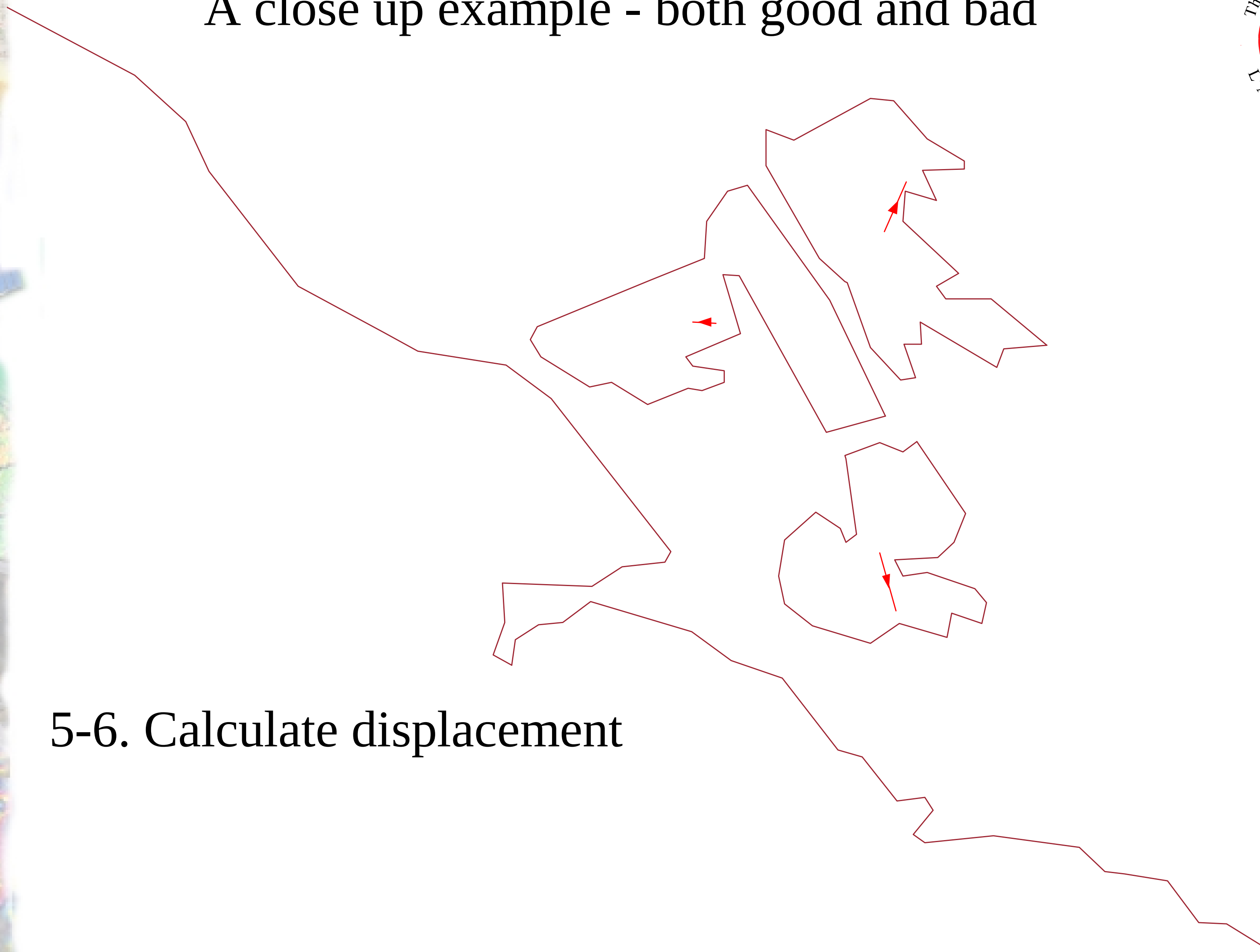


4. Calculate forces



5-6. Calculate displacement

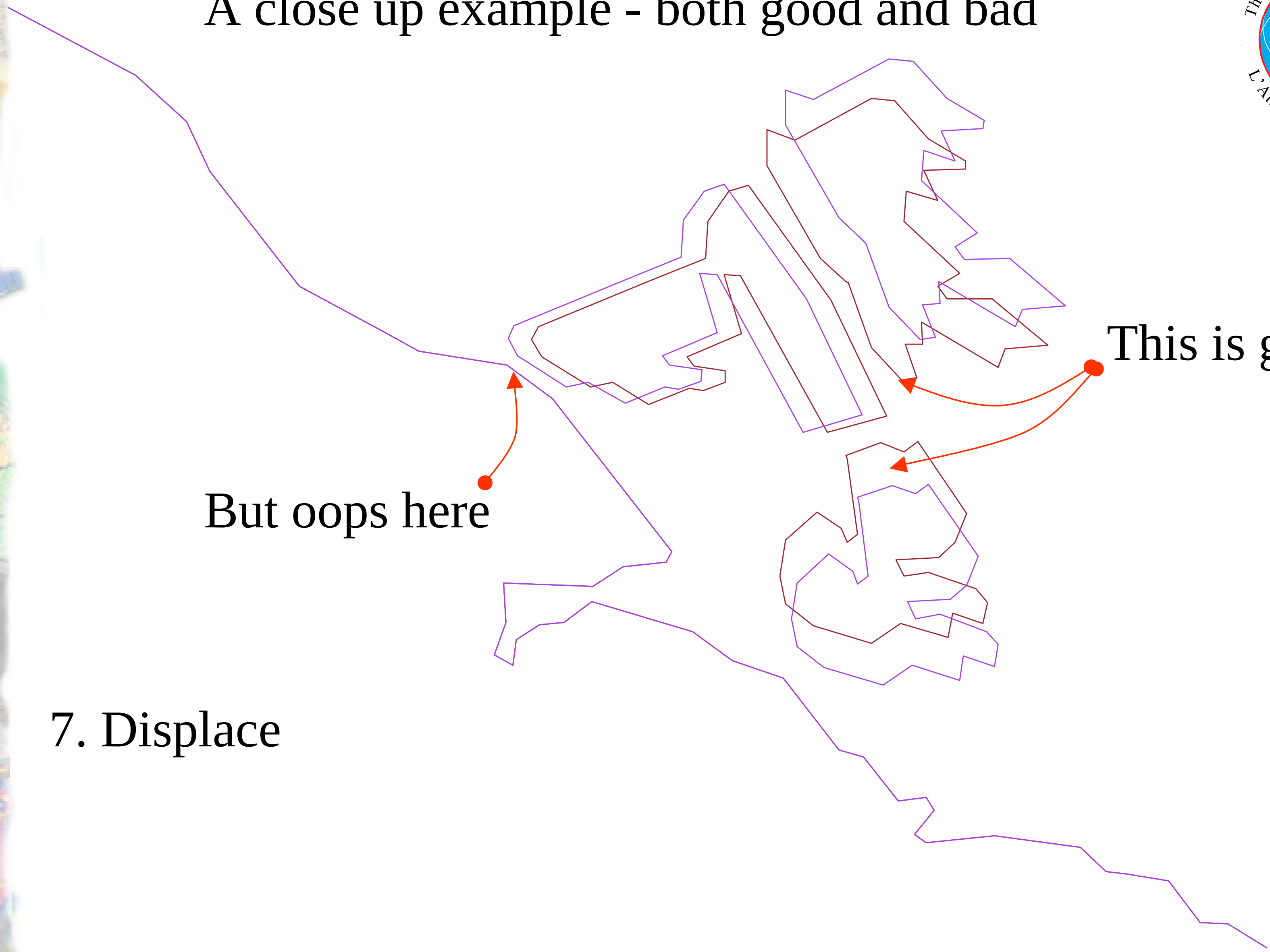
A close up example - both good and bad



5-6. Calculate displacement

7. Displace

A close up example - both good and bad



7. Displace

8-9. Clean up topology

Conclusions

- Our tools, though imperfect, have been useful in production for us in the past.
- We hope by releasing them publically, to both provide something useful to the community, and to encourage others to help in their maintenance and development
- To that end, the tools have been rewritten and repackaged and released online.