Edge Detection in Segmented Non-convex Space Polygon

I. Kožar*, D. Lozzi-Kožar ** and Ž. Jeričević***

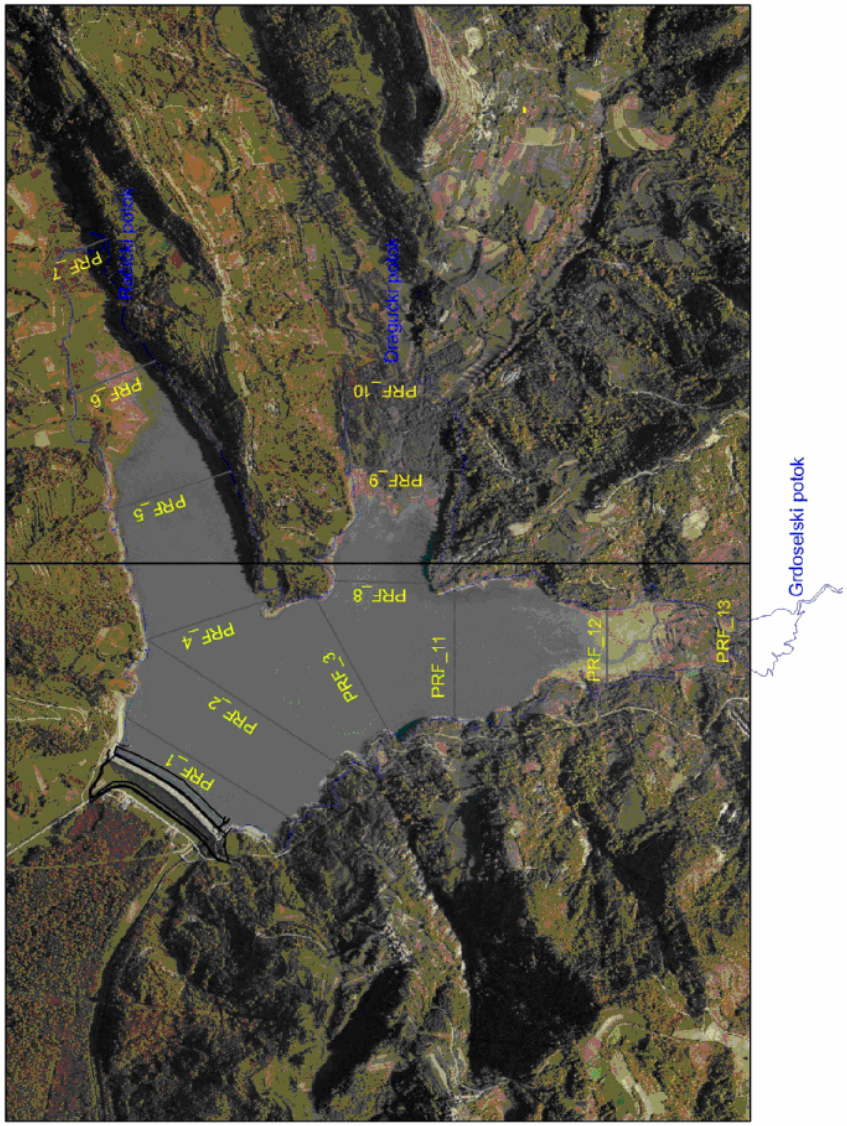
*** Faculty of Civil Engineering /Department of Computer Modeling, Rijeka, Croatia**

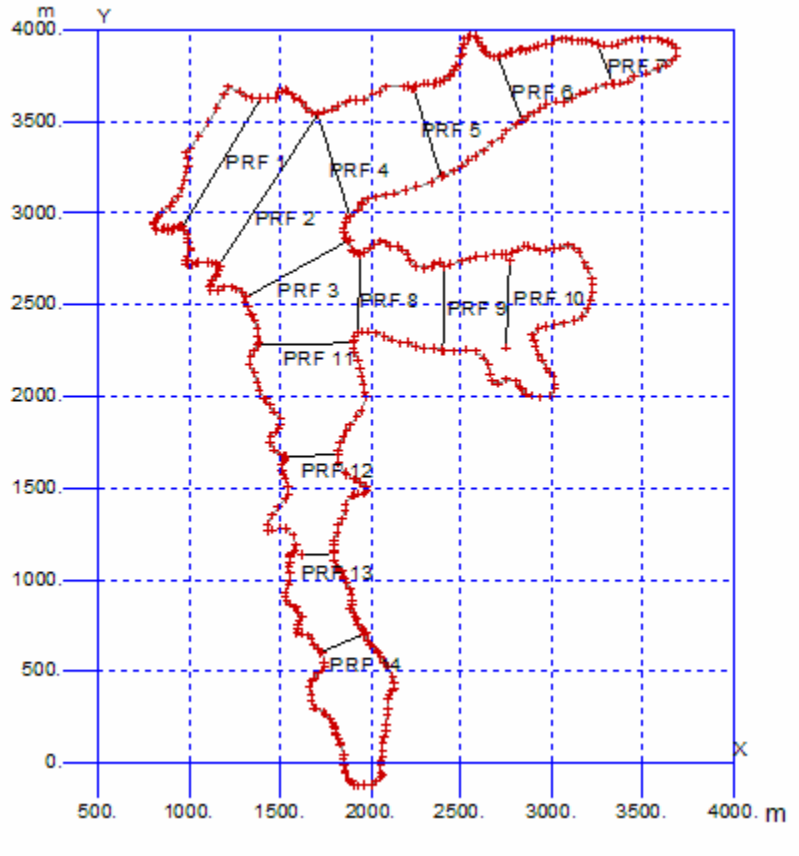
**** Croatian Water Authority, Rijeka, Croatia**

***** Engineering Faculty/Department of Computer Engineering, Rijeka, Croatia**

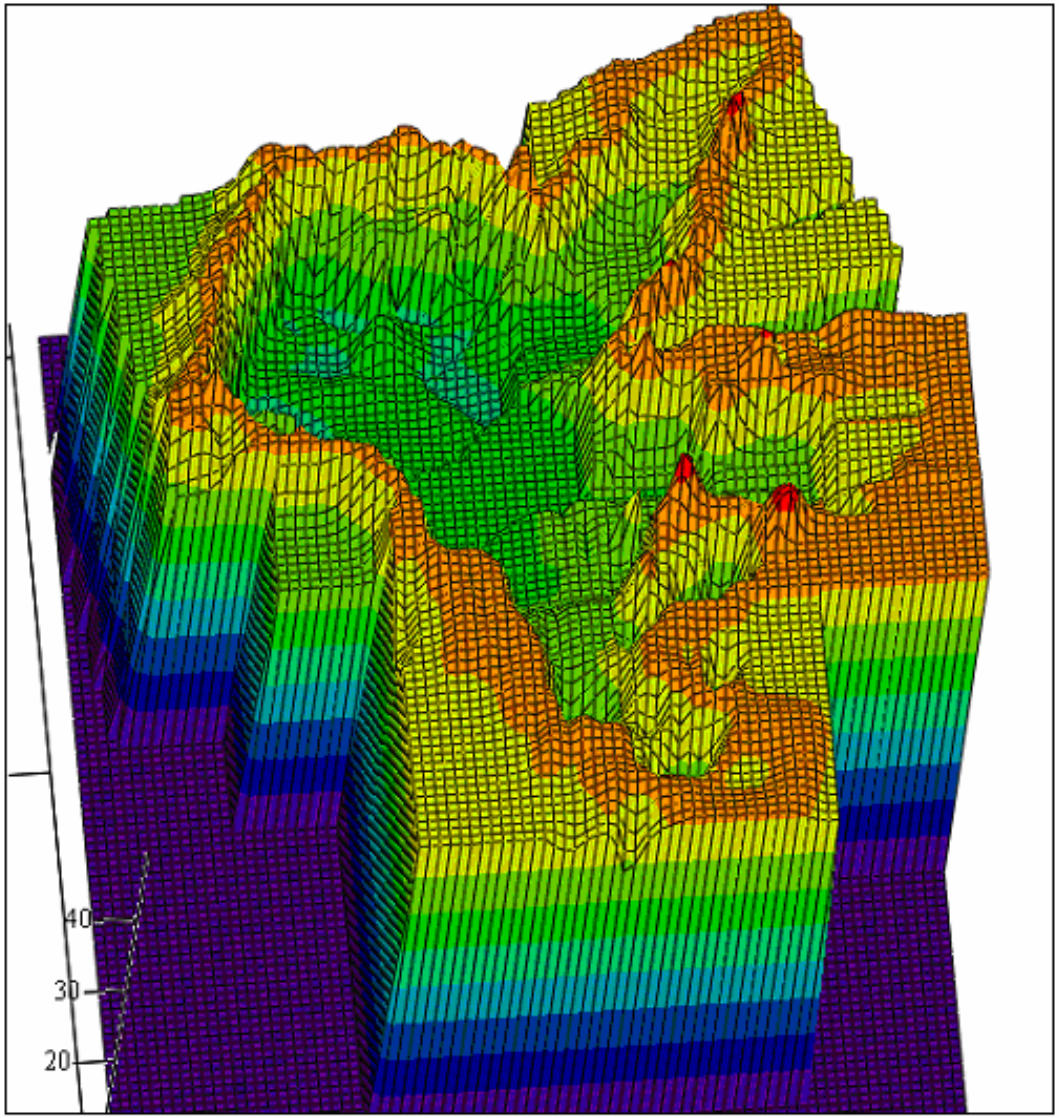
ivicak@gradri.hr

ABOUT EDGE DETECTION



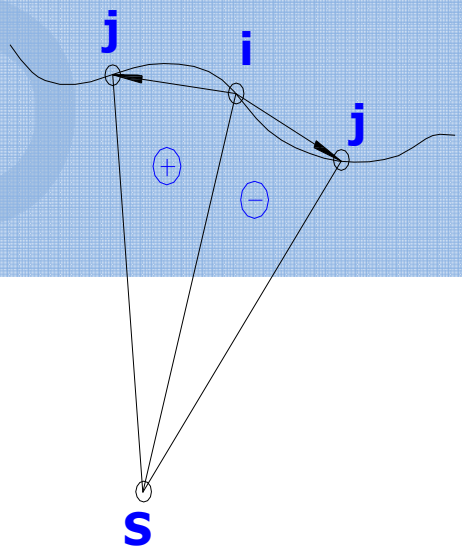


OUR GOAL

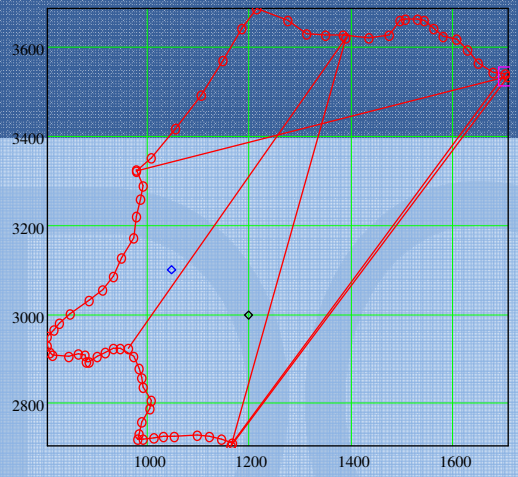


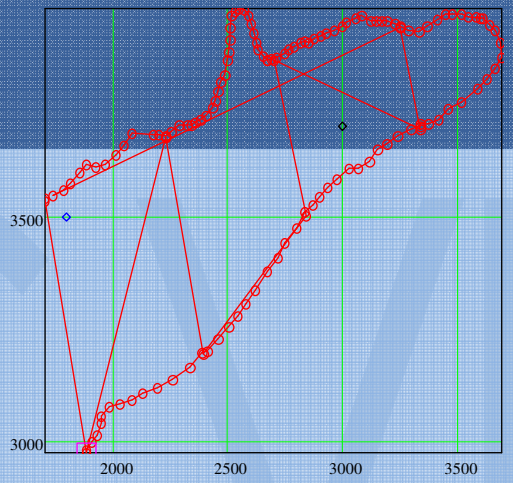
Edge detection serves us in mesh production:
- Inside vs. Outside points

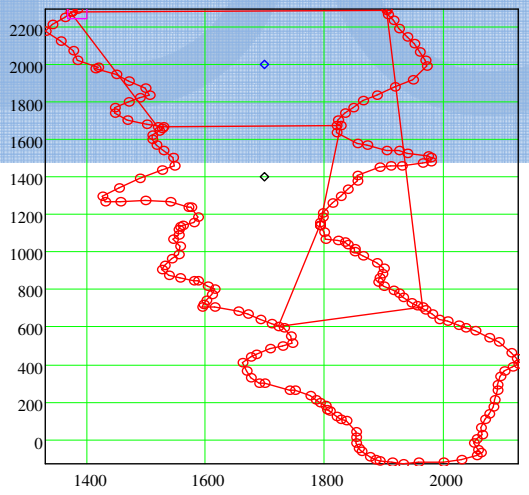
Simple algorithm does not work!

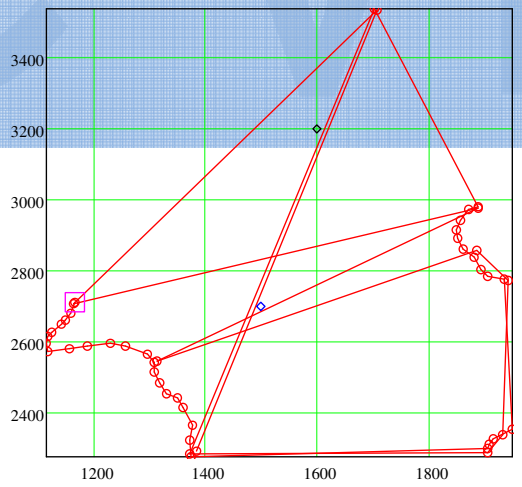


OUR DATA





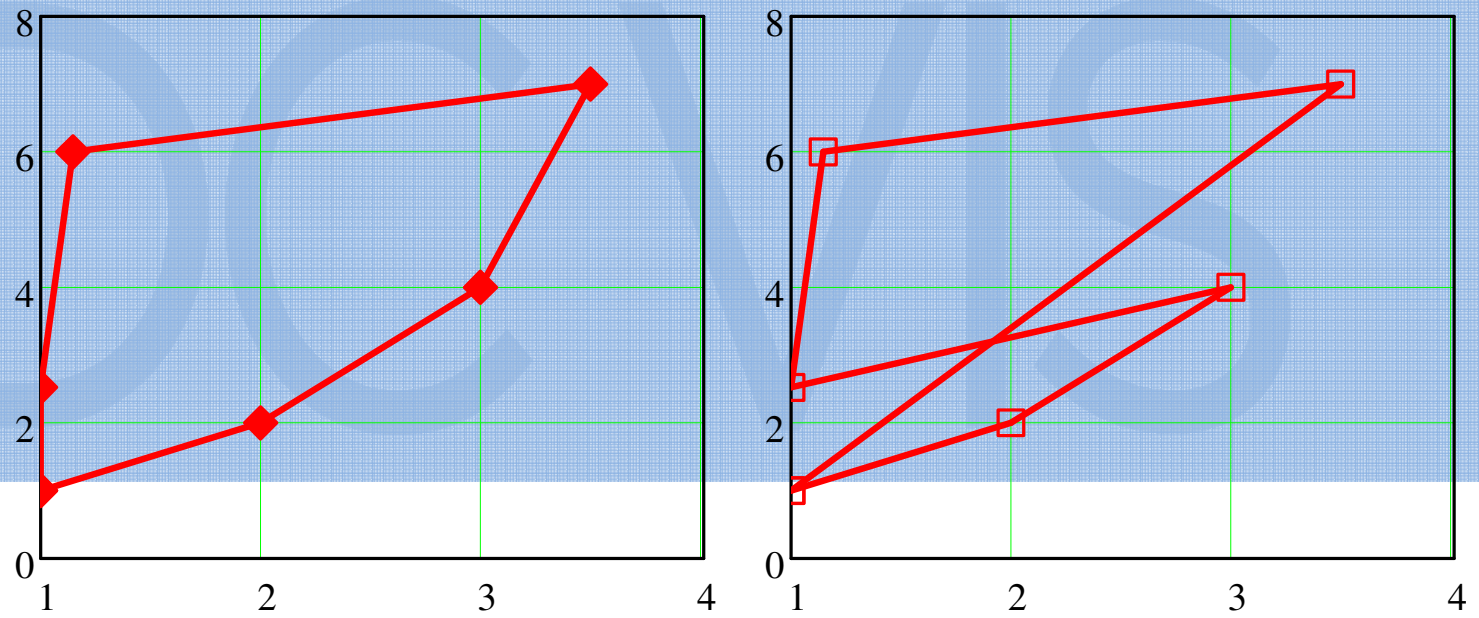




Segmented,
unordered

NON-UNIQUE PROBLEM

- Same points – many shapes

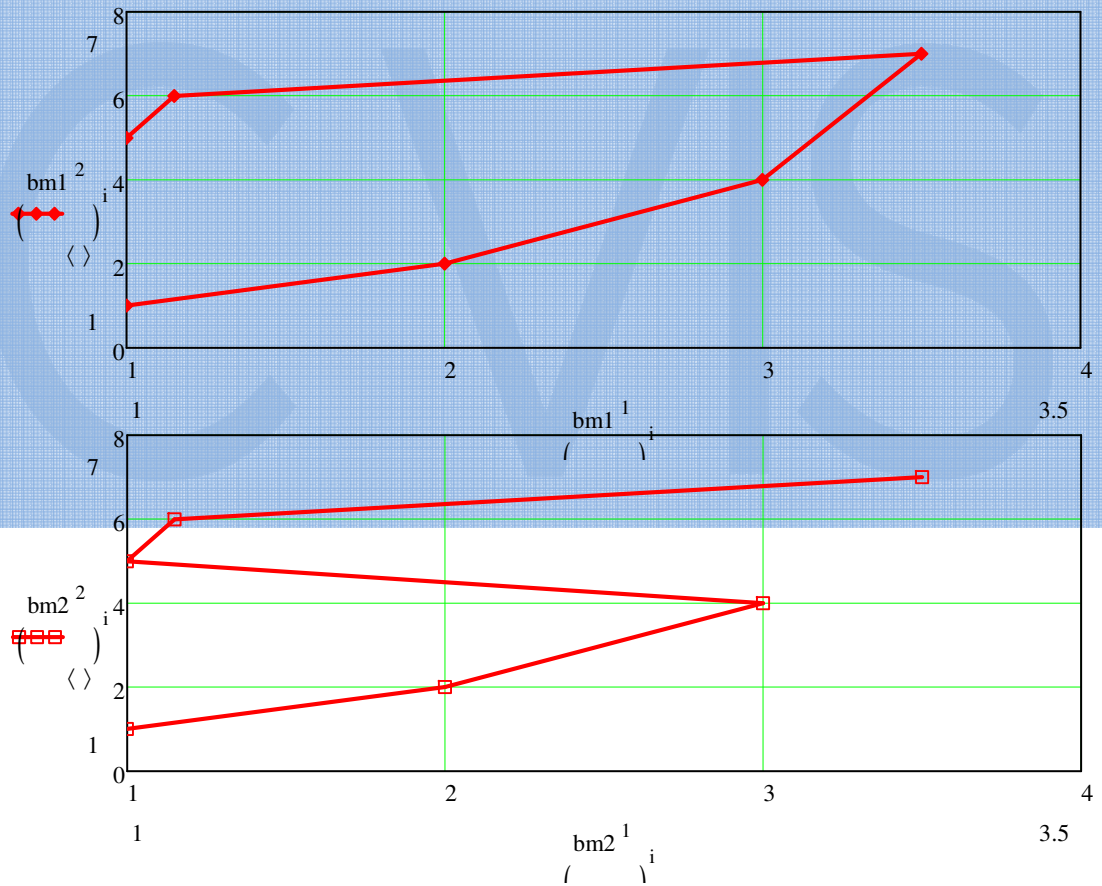


Sorting problem

- Sorting criteria ?

$$bm1 := \begin{pmatrix} 1 & 1 \\ 2 & 2 \\ 3 & 4 \\ 3.5 & 7 \\ 1.15 & 6 \\ 1 & 5 \end{pmatrix}$$

$$bm2 = \begin{pmatrix} 1 & 1 \\ 2 & 2 \\ 3 & 4 \\ 1 & 5 \\ 1.15 & 6 \\ 3.5 & 7 \end{pmatrix}$$



DFT = solution

$xm1 := Fm \cdot bm1$ $\quad$ $xm2 := Fm \cdot bm2$

$$F = \begin{bmatrix} 1 & 1 & 1 & . & 1 \\ 1 & w & w^2 & . & w^{n-1} \\ 1 & w^2 & w^4 & . & w^{2(n-1)} \\ . & . & . & . & . \\ 1 & w^{n-1} & w^{2(n-1)} & . & w^{(n-1)^2} \end{bmatrix}$$

$$w = e^{-2\pi i/n} \quad \& \quad w^n = e^{-2\pi i} = 1$$

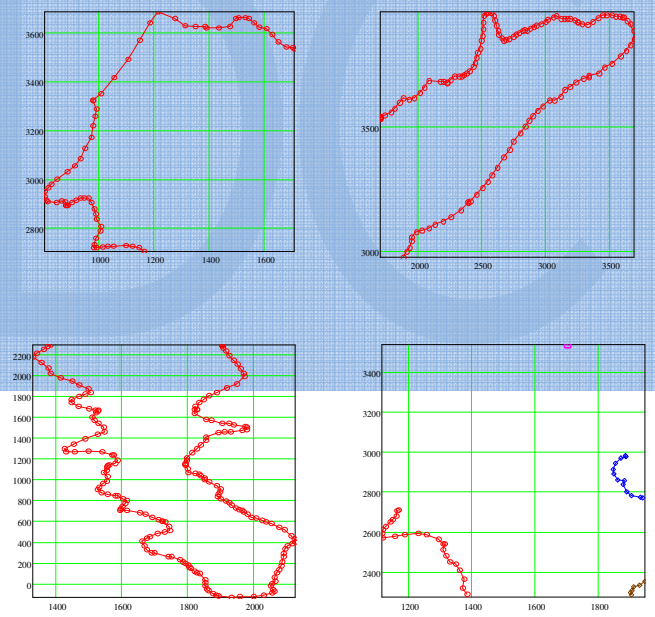
$$P1 := \overrightarrow{\sqrt{Re(xm1)^2 + Im(xm1)^2}} \qquad P2 := \overrightarrow{\sqrt{Re(xm2)^2 + Im(xm2)^2}}$$

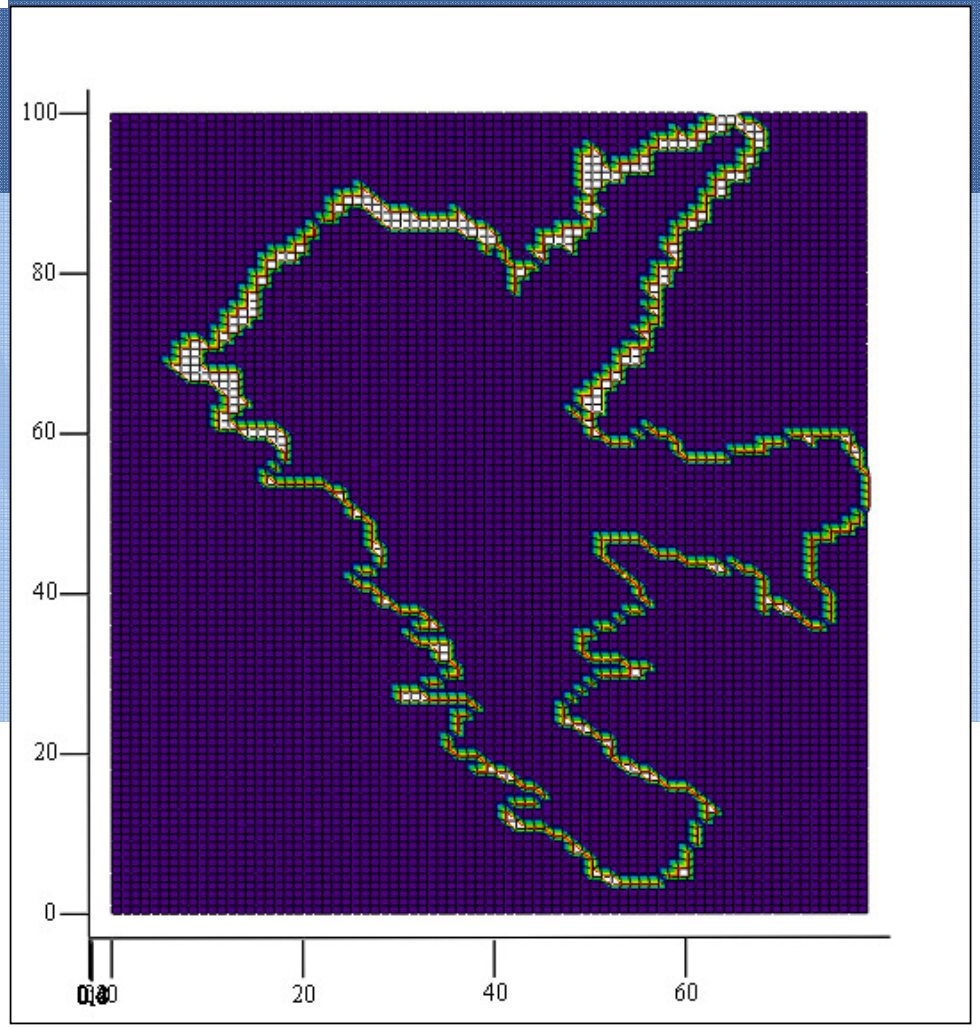
$$\sum_i \left[\left(P1^{\langle 1 \rangle} \right)_i \cdot \left(P1^{\langle 2 \rangle} \right)_i \right] = 57.218 \qquad \text{always} > \qquad \sum_i \left[\left(P2^{\langle 1 \rangle} \right)_i \cdot \left(P2^{\langle 2 \rangle} \right)_i \right] = 53.915$$

Parseval's theorem (Power theorem)!

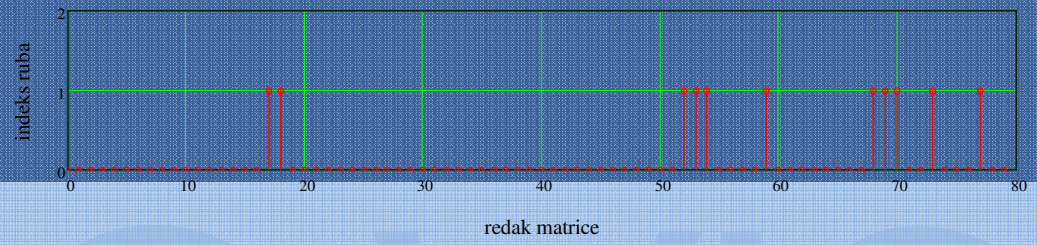
Problems with edge

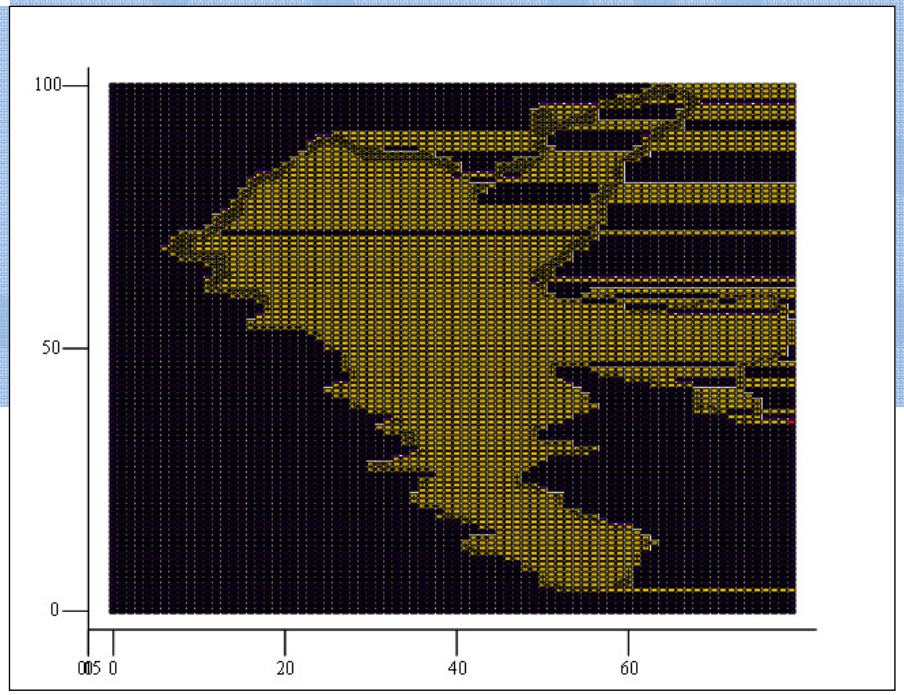
- After normalising, scaling, kriging :





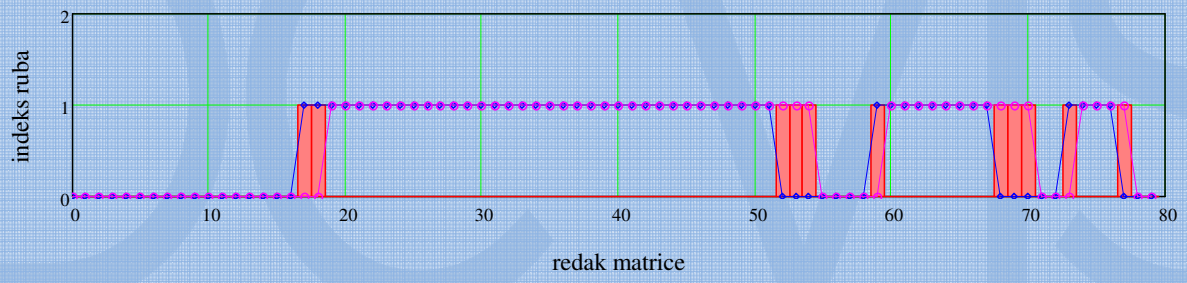
Cross section and edge-detection

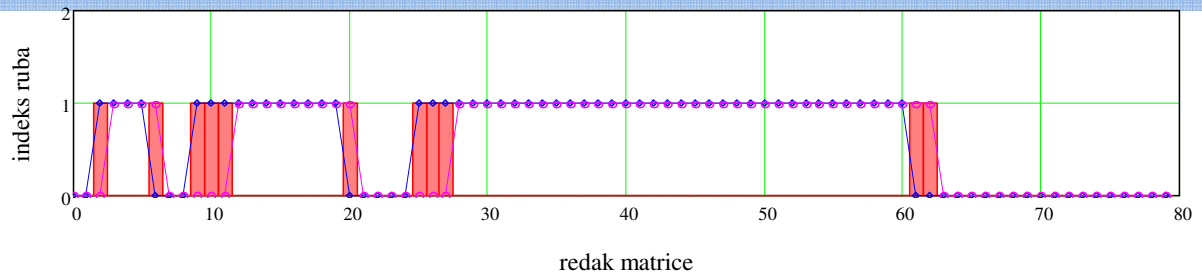




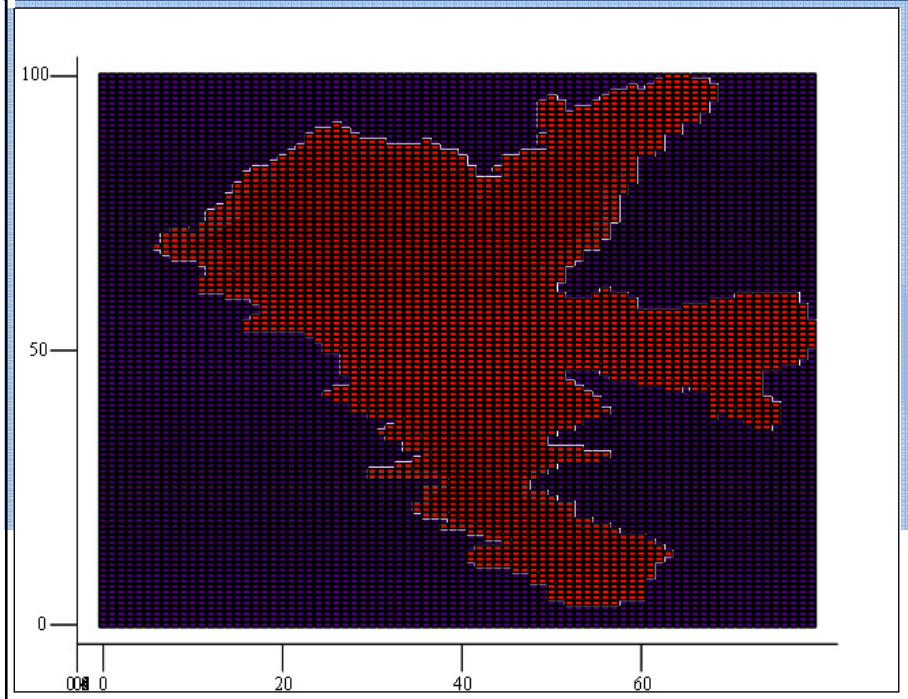
Improved cross section description

- Two directional screening
- left -> right / right -> left

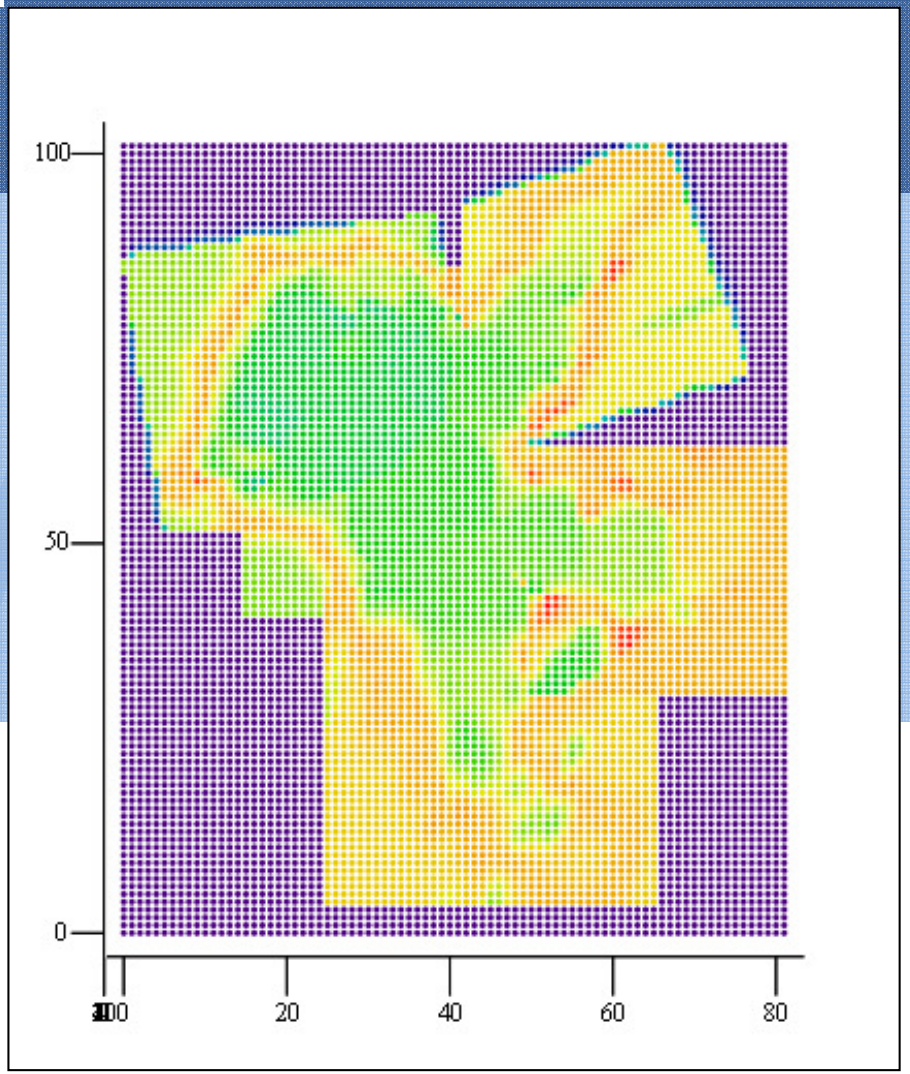




Result of an improved edge detection



After kriging ->



Result is a successful FE model