

Elementy trójkątne

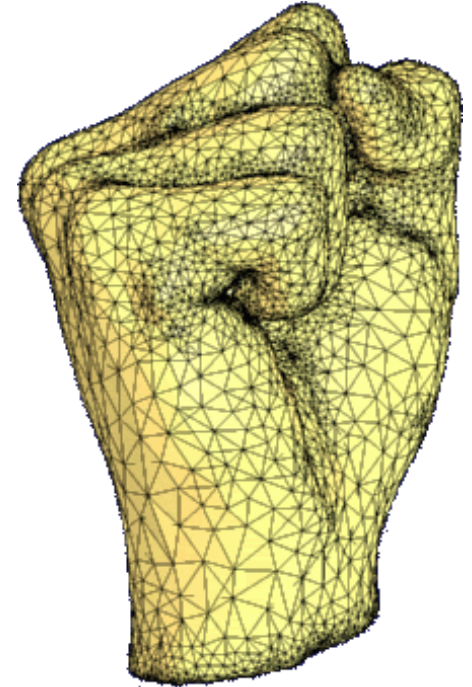
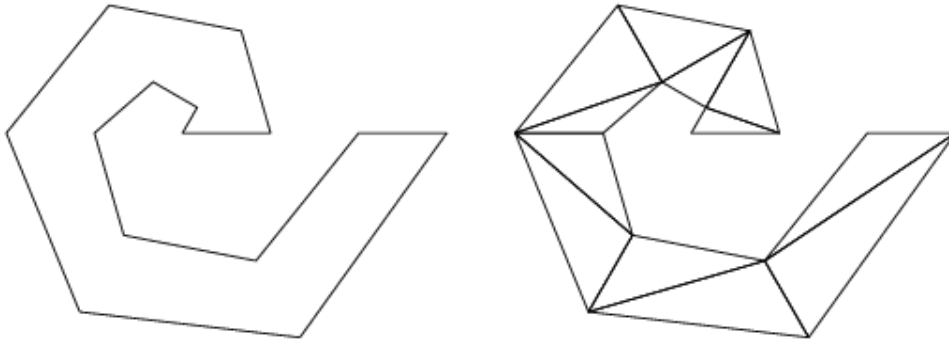
elementy czworokątne: bezpośrednie uogólnienie 1D na 2D, ale:

to trójkąt jest najprostszym elementem w 2D (w 3D tetraedr),

Ponadto: Każdy podział na elementy trójkątne jest odwracalny (poza wierzchołkami współliniowymi, dla czworokątów trzeba dbać o wypukłość) łatwiejszy do zautomatyzowania podział na elementy. lepiej nadaje się do opisu złożonego brzegu.

poza tym: transformacja element odniesienia - fizyczny: ściśle liniowa (nie biliniowa), jacobian transformacji zawsze niezależny od położenia w przestrzeni odniesienia (niezależnie od kształtu elementu fizycznego)

algorytmy triangulacyjne z geometrii i grafiki komp)



<http://www.cs.berkeley.edu/~jrs/mesh/>

Elementy trójkątne: współrzędne polowe

elementy czworokątne: funkcje kształtu – bezpośrednie uogólnienie 1D.

współrzędne ξ_1 i ξ_2 - również bezpośrednie uogólnienie 1D

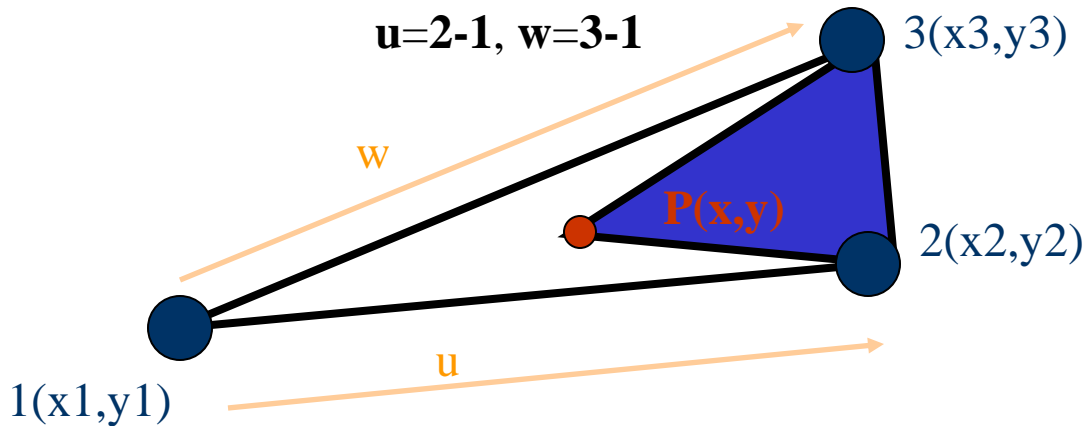
elementy trójkątne: - współrzędne polowe
do generacji funkcji kształtu

Δ – powierzchnia trójkąta (1,2,3)

$$\Delta = \frac{1}{2} |\mathbf{u} \times \mathbf{w}| \quad (\text{bez } \frac{1}{2} - \text{równoległobok})$$

albo

$$\Delta = \frac{1}{2} \begin{vmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{vmatrix}$$

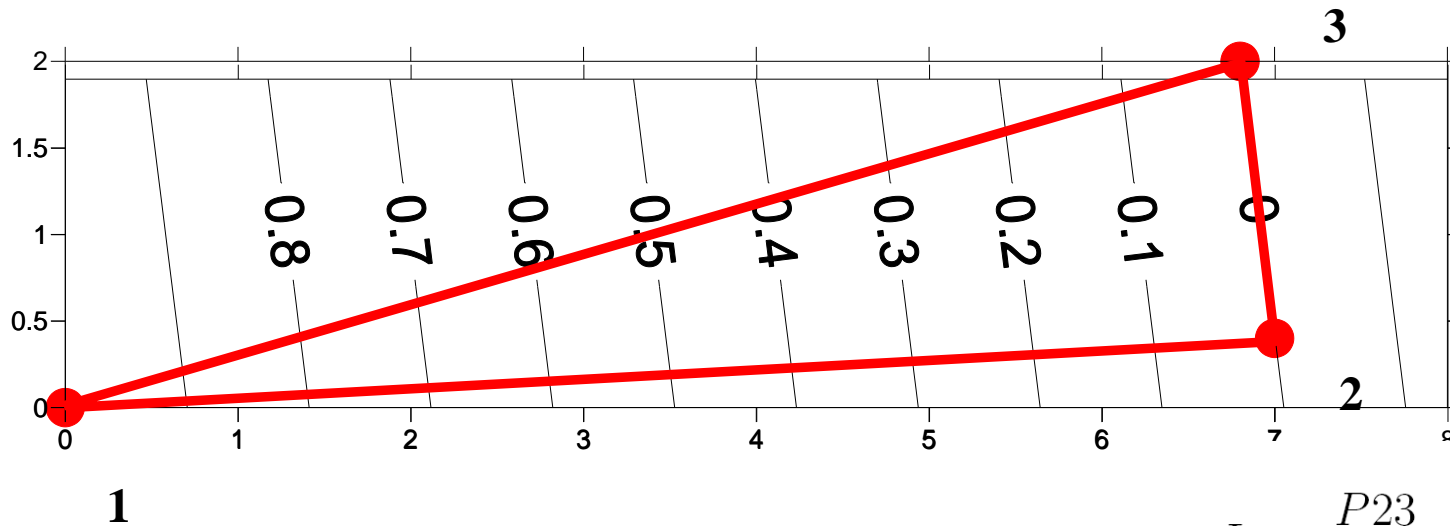


stosunek powierzchni P23 / 123

$$L_1 = \frac{P_{23}}{123} = \frac{1}{2\Delta} \begin{vmatrix} 1 & x & y \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{vmatrix}$$

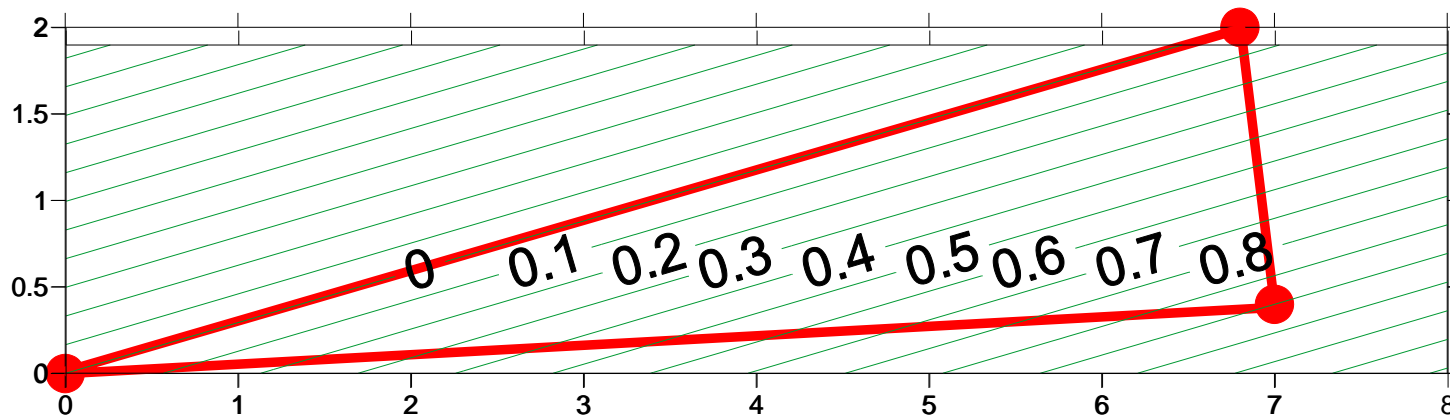
- ← 1) wartość liniowa w x i w y
- 2) $L_1=1$, gdy $P=1$
- 3) $L_1=0$ gdy $P=3$, $P=2$, lub gdy P na linii łączącej 2 i 3 (liniowa zależność wierszy macierzy)

Współrzędna polowa L_1 (równoległa do bo 2-3)



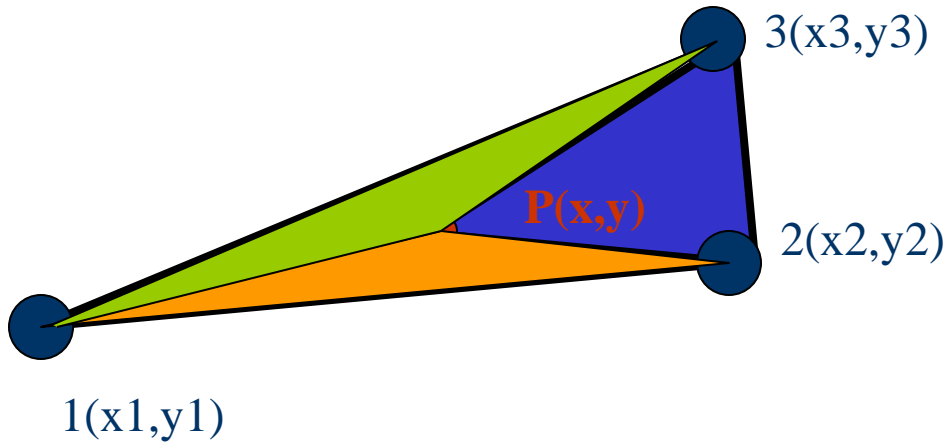
$$L_1 = \frac{P_{23}}{123} = \frac{1}{2\Delta} \begin{vmatrix} 1 & x & y \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{vmatrix}$$

Współrzędna polowa L_2



$$L_2 = \frac{P_{13}}{123} = \frac{1}{2\Delta} \begin{vmatrix} 1 & x & y \\ 1 & x_1 & y_1 \\ 1 & x_3 & y_3 \end{vmatrix}$$

Elementy trójkątne: współrzędne polowe (powierzchniowe *area*)



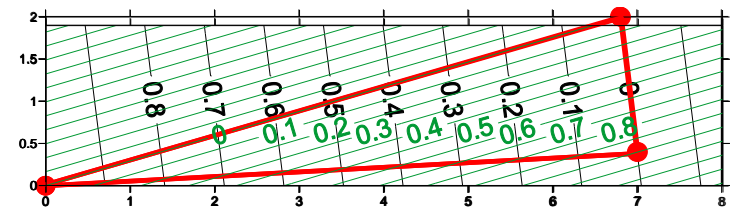
$$L_1 = \frac{P_{23}}{123} = \frac{1}{2\Delta} \begin{vmatrix} 1 & x & y \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{vmatrix}$$

$$L_3 = \frac{P_{12}}{123} = \frac{1}{2\Delta} \begin{vmatrix} 1 & x & y \\ 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \end{vmatrix}$$

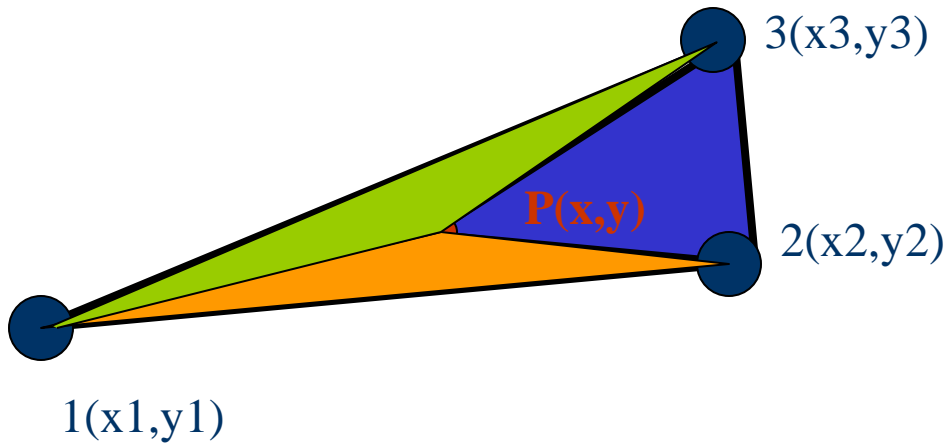
$$L_2 = \frac{P_{13}}{123} = \frac{1}{2\Delta} \begin{vmatrix} 1 & x & y \\ 1 & x_1 & y_1 \\ 1 & x_3 & y_3 \end{vmatrix}$$

$L_1 + L_2 + L_3 = 1$ -tylko 2 współrzędne niezależne

L_1, L_2



Elementy trójkątne: współrzędne polowe (powierzchniowe *area*)



$$L_1 = \frac{P_{23}}{123} = \frac{1}{2\Delta} \begin{vmatrix} 1 & x & y \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{vmatrix}$$

$$L_3 = \frac{P_{12}}{123} = \frac{1}{2\Delta} \begin{vmatrix} 1 & x & y \\ 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \end{vmatrix}$$

$$L_2 = \frac{P_{13}}{123} = \frac{1}{2\Delta} \begin{vmatrix} 1 & x & y \\ 1 & x_1 & y_1 \\ 1 & x_3 & y_3 \end{vmatrix}$$

$L_1 =$ jeden w $(x,y)=1$, zero w pozostałych punktach: warunki stawiane funkcjom kształtu Lagrange'a najniższego rzędu

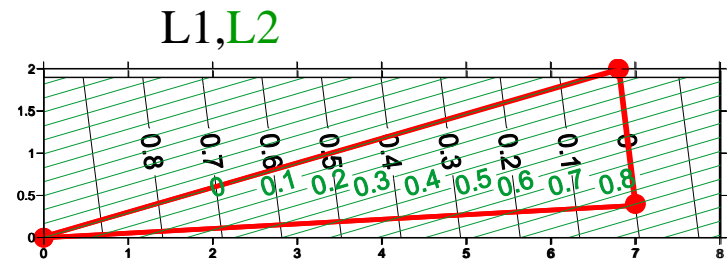
$$u(x, y) = u_1\phi_1(x, y) + u_2\phi_2(x, y) + u_3\phi_3(x, y)$$

$$\phi_1(x, y) = L_1$$

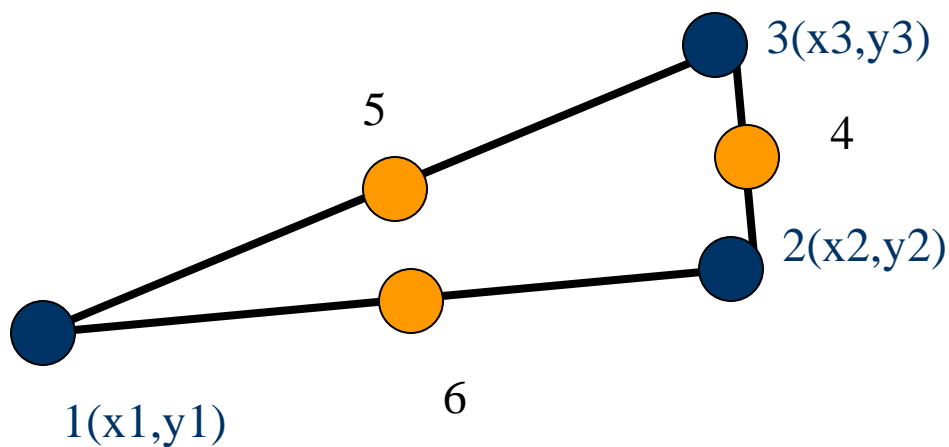
$$\phi_2(x, y) = L_2$$

$$\phi_3(x, y) = L_3 = 1 - L_1 - L_2$$

$L_1 + L_2 + L_3 = 1$ - para współrzędnych

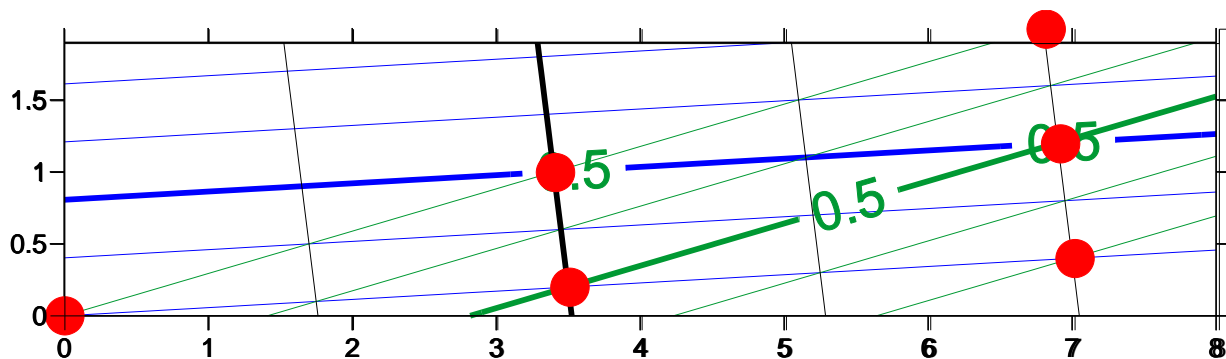


Elementy trójkątne: kwadratowe funkcje kształtu



L_1, L_2, L_3

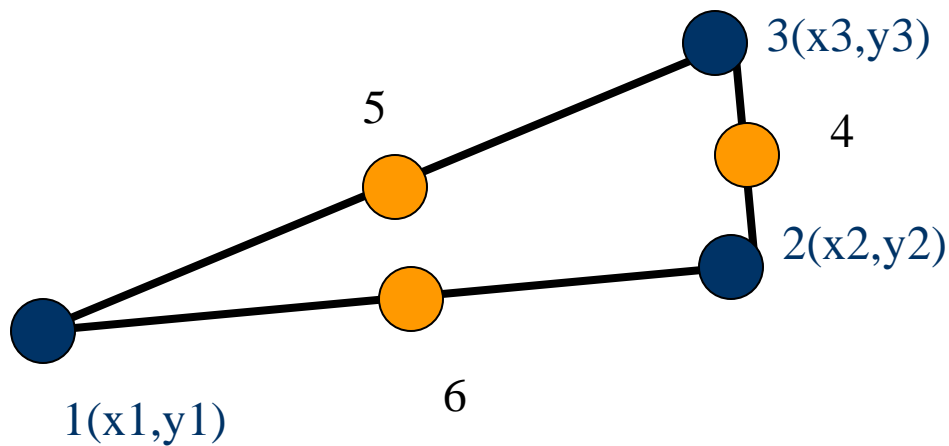
środkie boków trójkąta: dla współrzędnych połowych 0.5



sprawdzić łatwo, że

$$\begin{vmatrix} 1 & (x_1 + x_3)/2 & (y_1 + y_3)/2 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{vmatrix} = \frac{1}{2} \begin{vmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{vmatrix}$$

Elementy trójkątne: kwadratowe funkcje kształtu



funkcje kształtu dla węzłów narożnych:

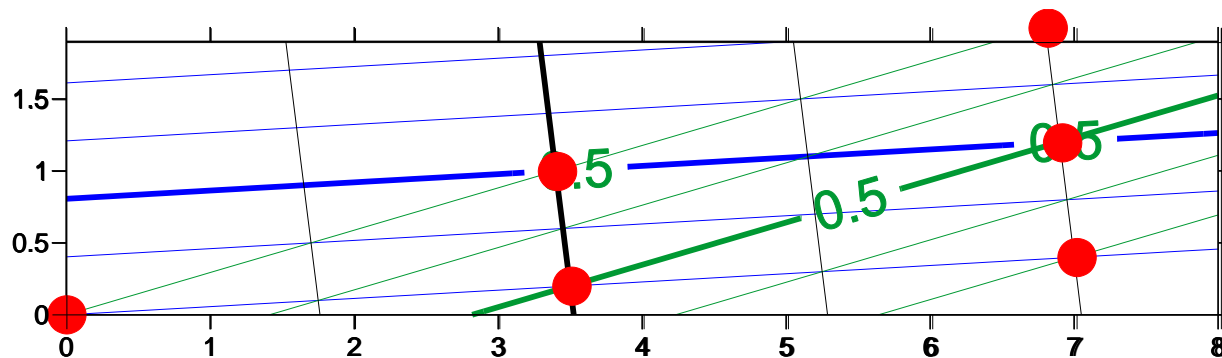
$$\phi_1(x, y) = L_1 (2L_1 - 1)$$

$$\phi_2(x, y) = L_2 (2L_2 - 1)$$

$$\phi_3(x, y) = L_3 (2L_3 - 1)$$

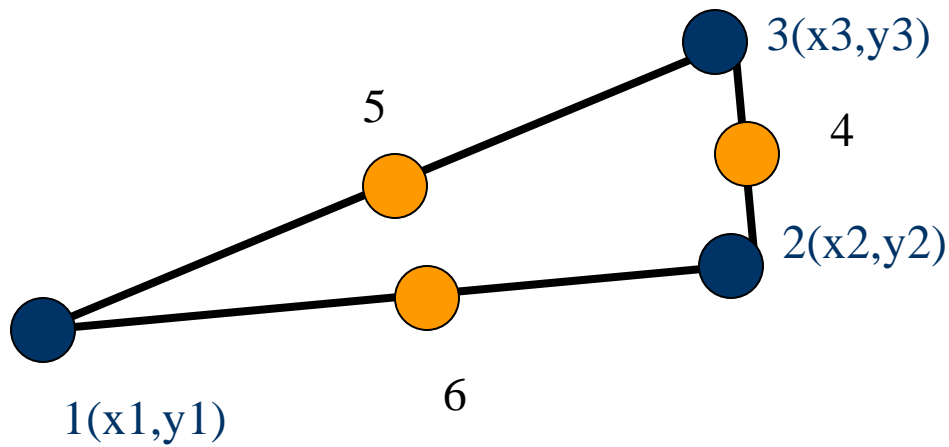
L_1, L_2, L_3

środky boków trójkąta: dla współrzędnych połowych 0.5



$$\begin{vmatrix} 1 & (x_1 + x_3)/2 & (y_1 + y_3)/2 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{vmatrix} = \frac{1}{2} \begin{vmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{vmatrix}$$

Elementy trójkątne: kwadratowe funkcje kształtu cd.

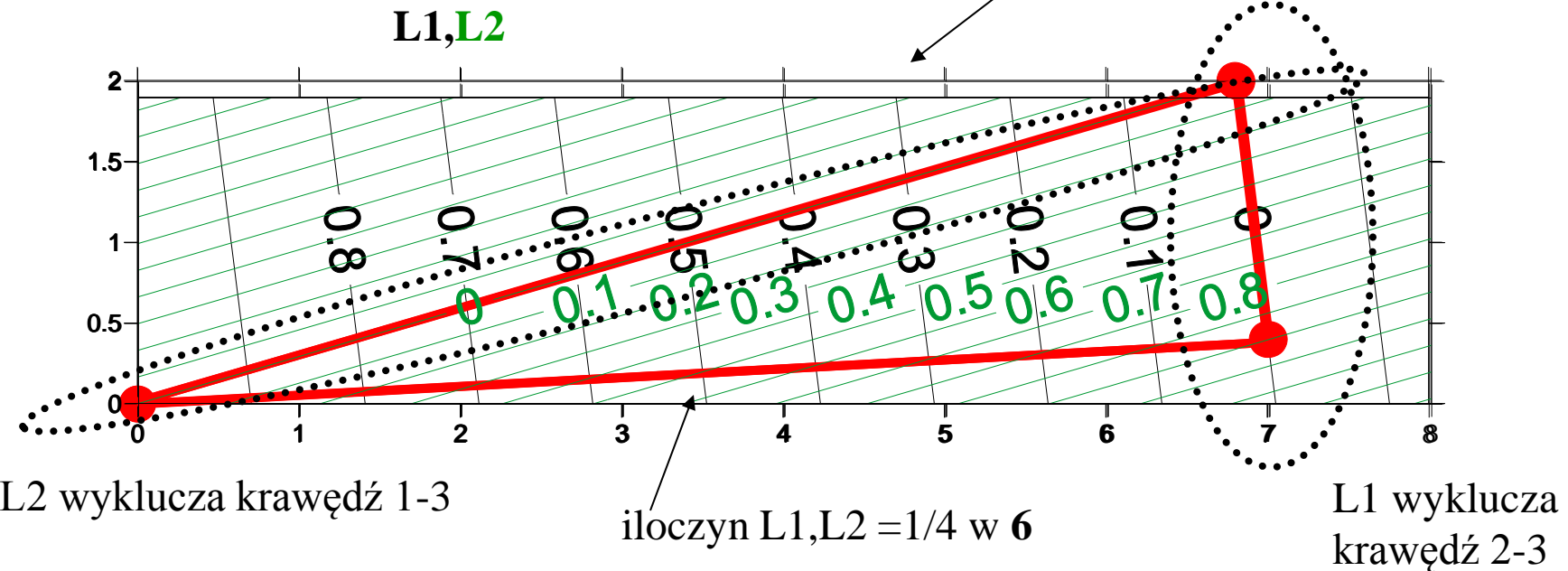


Węzły na środkach ścian:

$$\phi_4 = 4L_2L_3$$

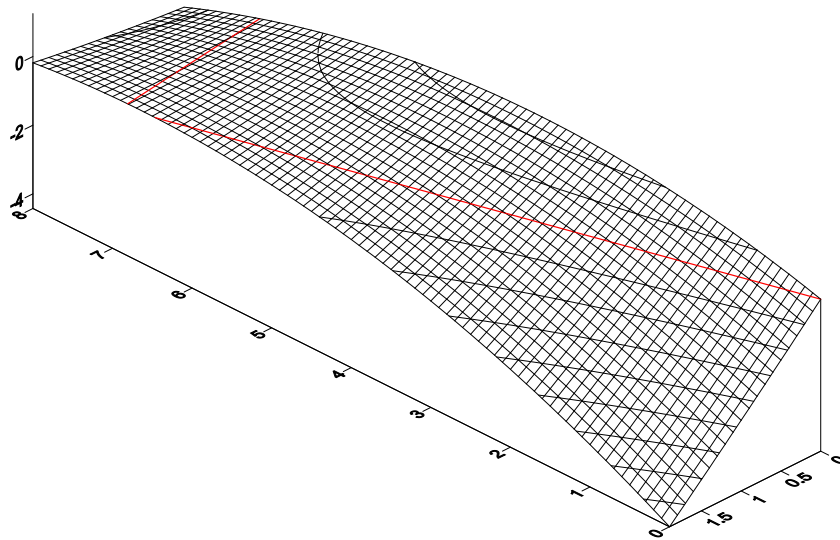
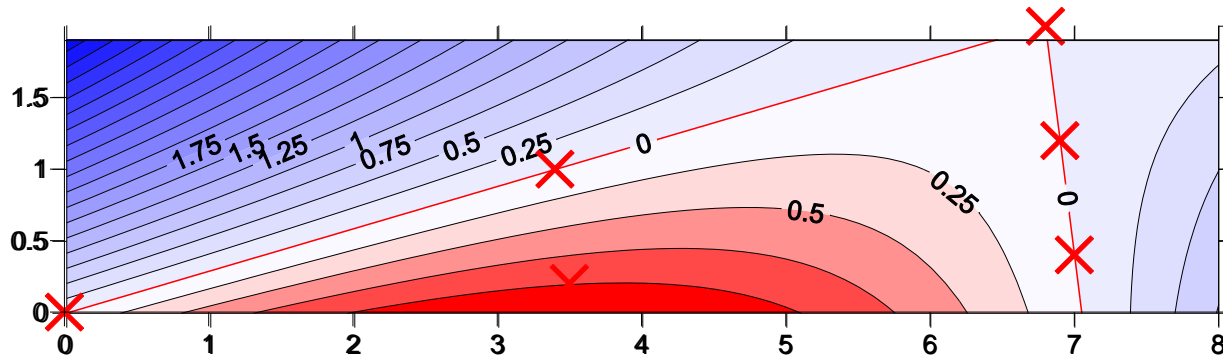
$$\phi_5 = 4L_1L_3$$

$$\phi_6 = 4L_1L_2$$

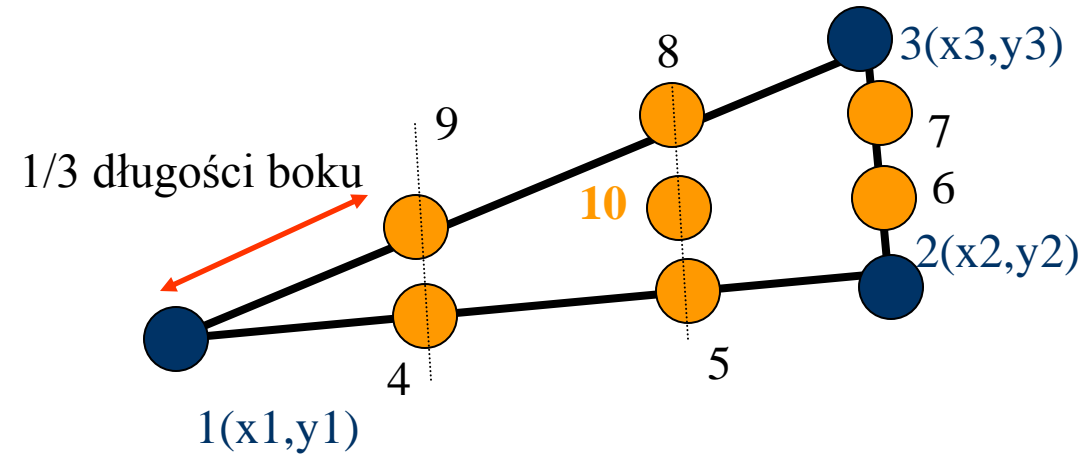


Elementy trójkątne: kwadratowe funkcje kształtu cd.

$$\phi_6 = 4L_1L_2$$



Elementy trójkątne: kubiczne funkcje kształtu



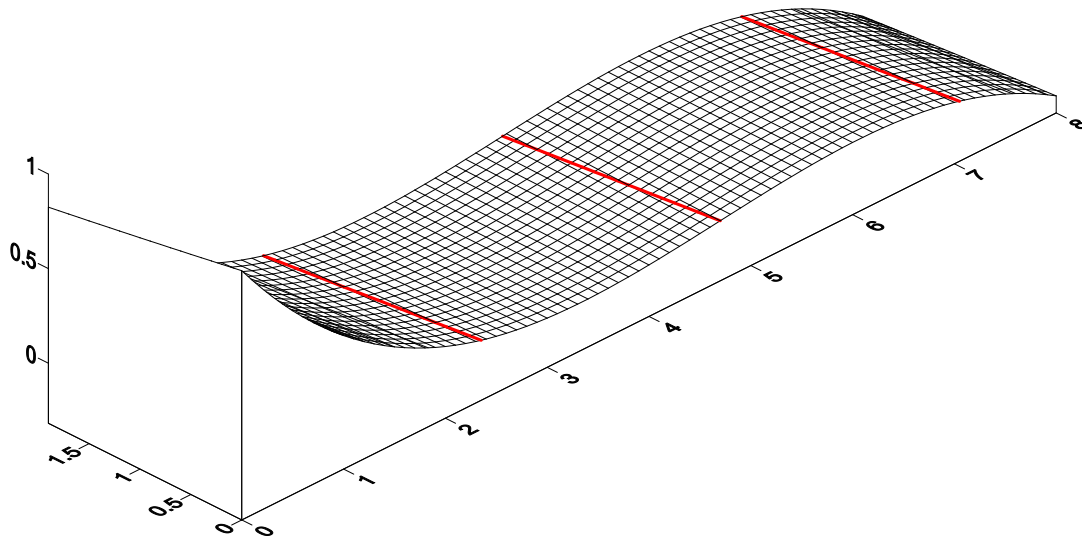
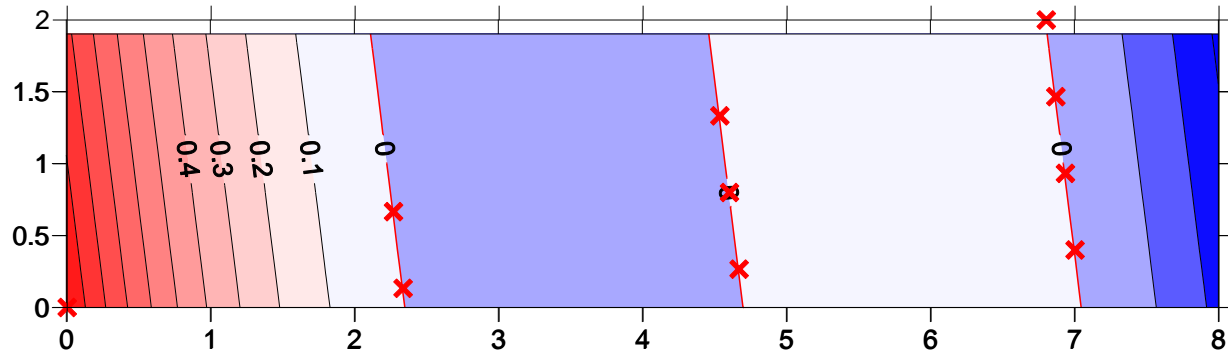
10 – środek ciężkości
 $[(x_1+x_2+x_3)/3, (y_1+y_2+y_3)/3]$

węzły narożne: $\phi_1 = \frac{1}{2}(3L_1 - 1)(3L_1 - 2)L_1$

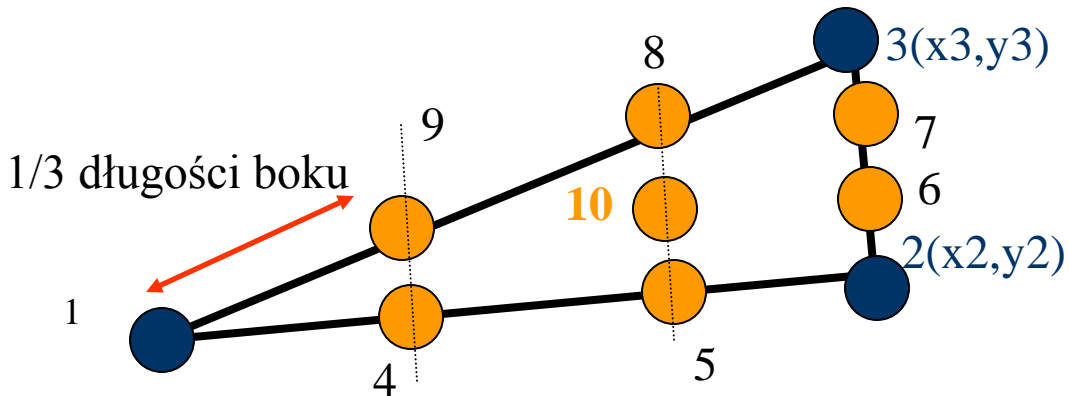
Elementy trójkątne: kubiczne funkcje kształtu

węzły narożne:

$$\phi_1 = \frac{1}{2}(3L_1 - 1)(3L_1 - 2)L_1$$

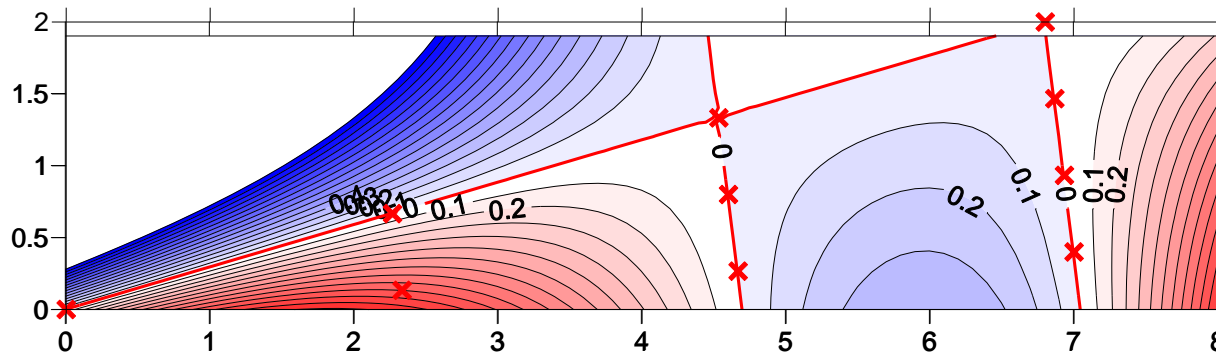


Elementy trójkątne: kubiczne funkcje kształtu

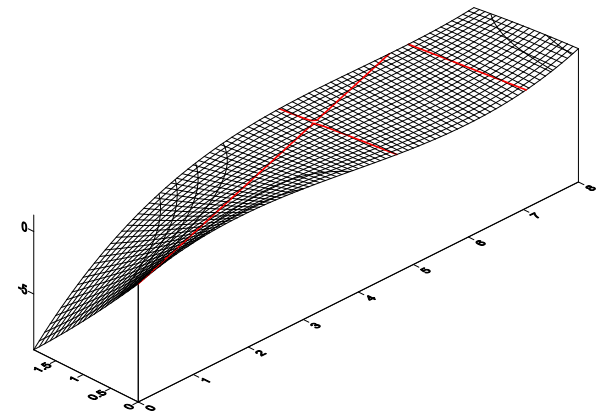


węzły na bokach

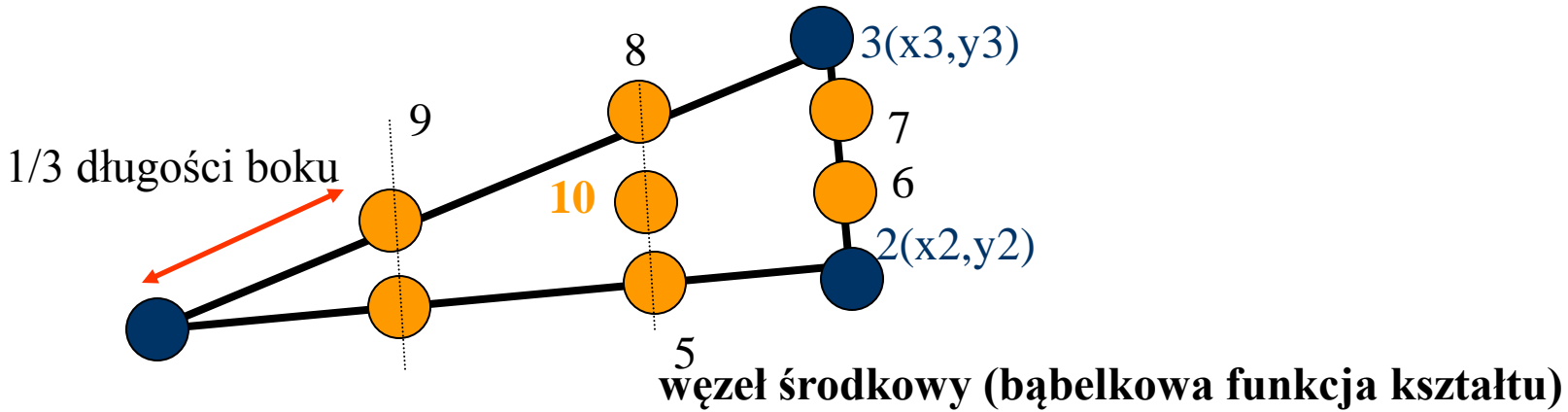
$$\phi_4 = L_2 L_1 (3L_1 - 1) * 9/2$$



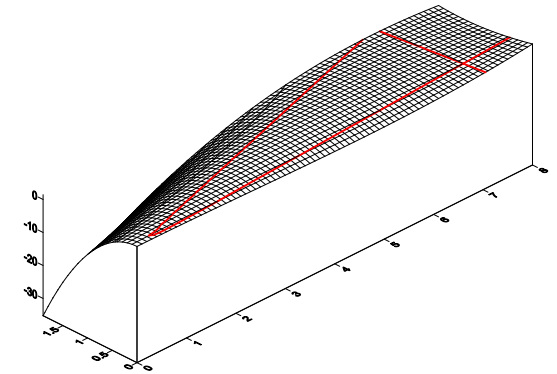
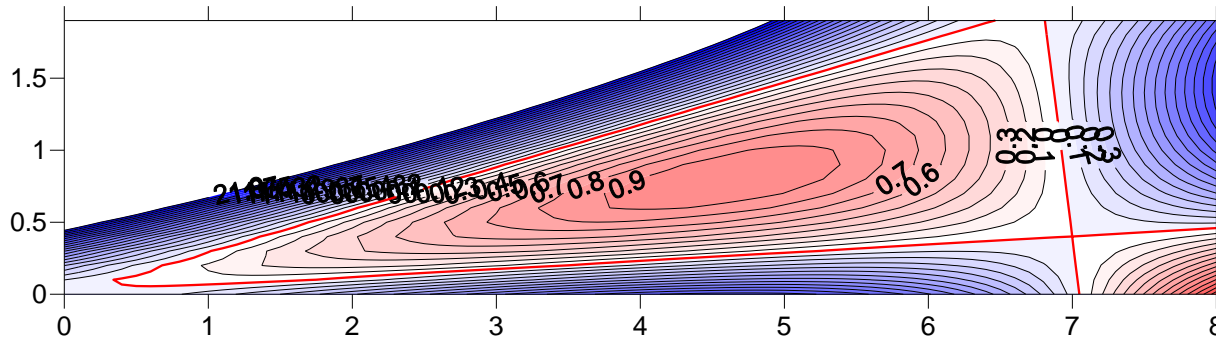
itd



Elementy trójkątne: kubiczne funkcje kształtu

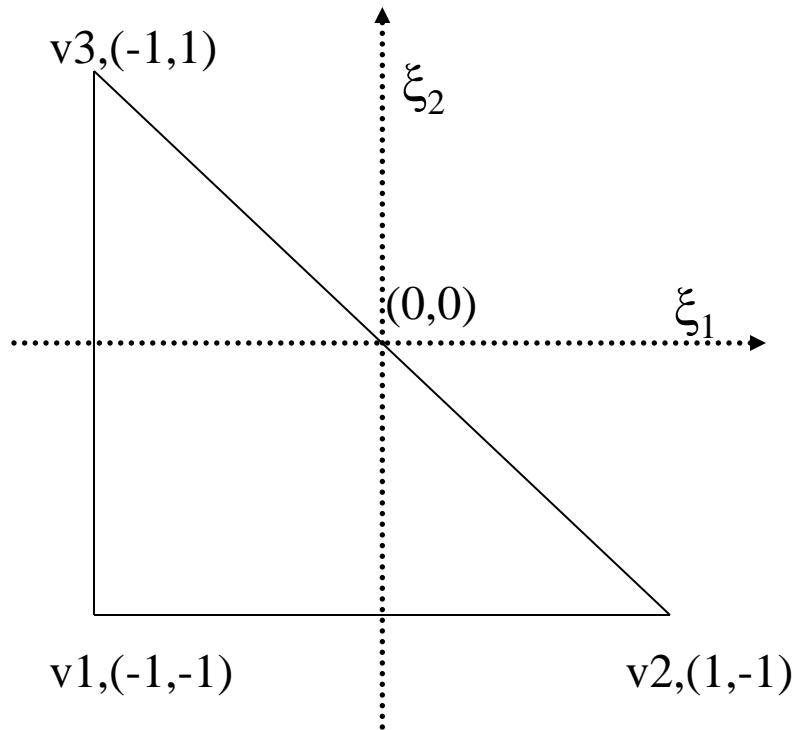


$$\phi_{10} = L_1 L_2 L_3 * 27$$



trójkątny element odniesienia

$$f_3 = \frac{1 + \xi_2}{2}$$



$$f_1 = -\frac{\xi_1 + \xi_2}{2}$$

$$f_2 = \frac{1 + \xi_1}{2}$$

funkcje do mapowania f_1, f_2, f_3

Mapowanie:

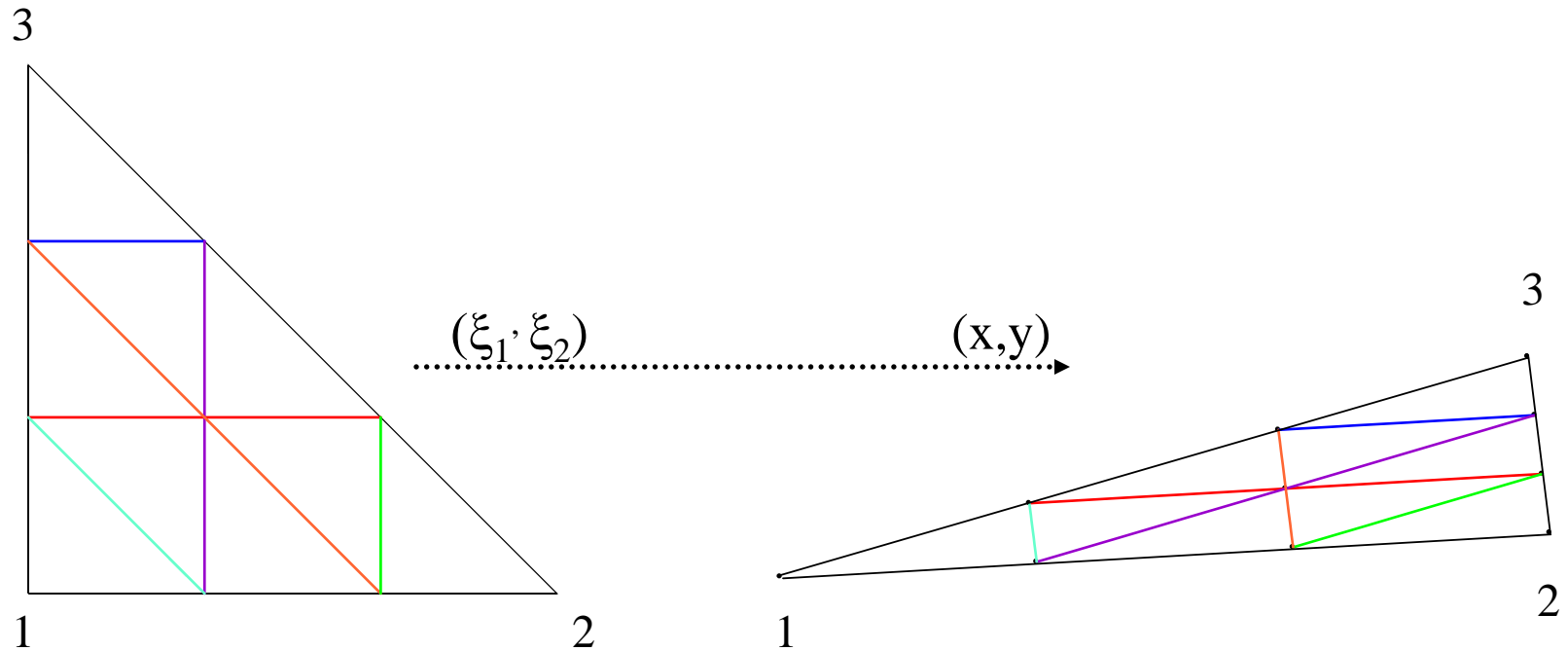
$$\begin{pmatrix} x \\ y \end{pmatrix} = \sum_{i=1}^3 \begin{pmatrix} x_i \\ y_i \end{pmatrix} f_i(\xi_1, \xi_2)$$

odwracalne, o ile wierzchołki niezależne liniowo (nie leżą na jednej linii)

Mapowanie jest *ściśle* liniowe
nie biliniowe jak dla kwadratowego
elementu odniesienia.

Mapowanie: z elementu odniesienia do elementu fizycznego

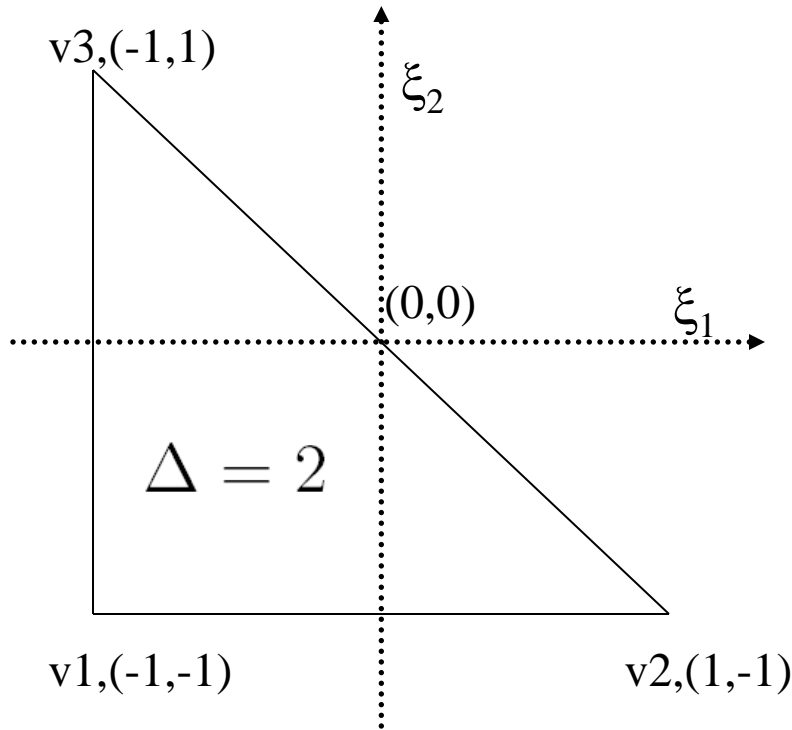
$$\begin{pmatrix} x \\ y \end{pmatrix} = \sum_{i=1}^3 \begin{pmatrix} x_i \\ y_i \end{pmatrix} f_i(\xi_1, \xi_2)$$



mapowanie liniowe: zachowana równoległość ścian
w mapowaniu biliniowym tak nie było

funkcje mapujące f_1, f_2, f_3 to funkcje kształtu najniższego rzędu L1, L2, L3 dla trójkąta odniesienia

$$f_3 = \frac{1 + \xi_2}{2}$$



$$L_1 = \frac{1}{2\Delta} \begin{vmatrix} 1 & \xi_1 & \xi_2 \\ 1 & 1 & -1 \\ 1 & -1 & 1 \end{vmatrix}$$

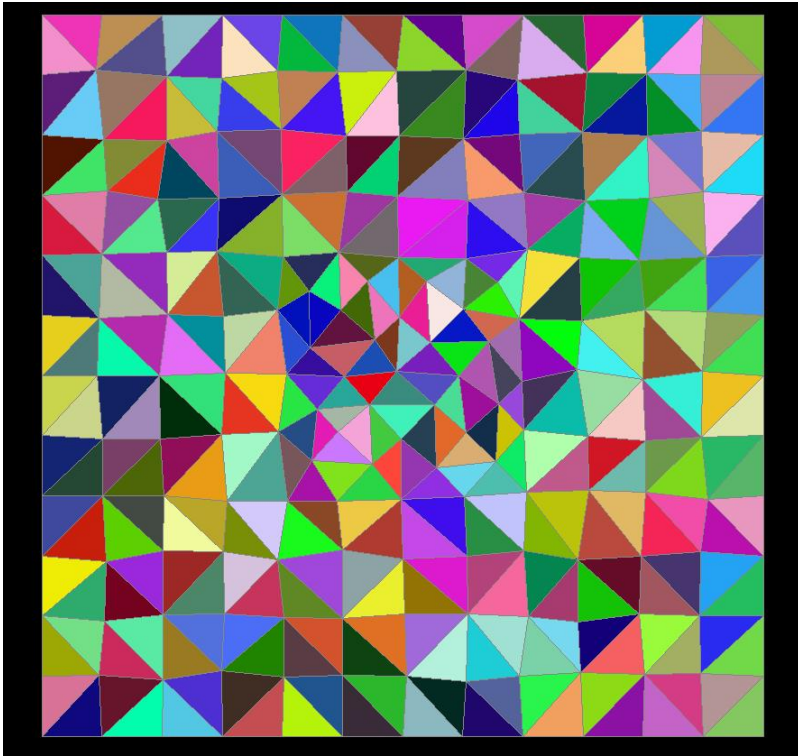
itd.

$$f_1 = -\frac{\xi_1 + \xi_2}{2}$$

$$f_2 = \frac{1 + \xi_1}{2}$$

odwracalne, o ile wierzchołki niezależne liniowo (nie leżą na jednej linii)
dodajmy: **łatwo odwracalne**

Po co odwracać?



Mamy siatkę trójkątną.

i rozwiązanie MES w węzłach.

Chcemy wyprowadzić wartość rozwiązania
w dowolnym punkcie przestrzeni fizycznej.

Najszybciej zaprogramować:

pętle po elementach m

pętle po współrzędnych odniesienia

ξ_1, ξ_2 potem

$$\begin{pmatrix} x \\ y \end{pmatrix} = \sum_{i=1}^3 \begin{pmatrix} x_i^m \\ y_i^m \end{pmatrix} f_i(\xi_1, \xi_2)$$

wypisać x/y : wada jest taka, że

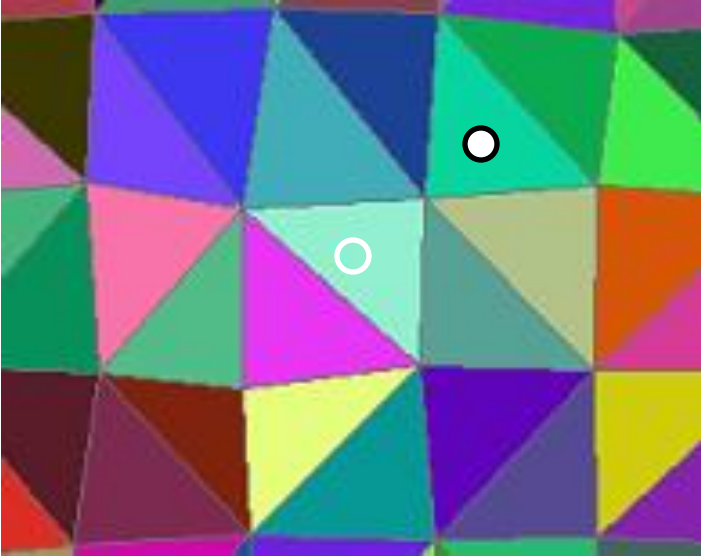
wydruki x/y będą nierówno rozłożone

w przestrzeni fizycznej, problem dla narzędzi
graficznych

Znacznie lepiej sprawę odwrócić: 1) przeglądać punkty w przestrzeni fizycznej

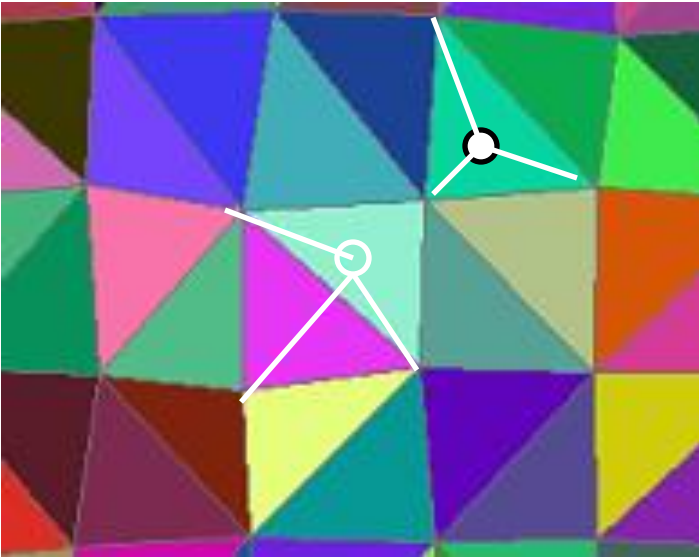
2) identyfikować element 3) wyliczyć ξ_1, ξ_2 które odpowiadają x/y , 4) wyliczyć u

- 1) ustalić x, y
- 2) zidentyfikować element



skąd wiadomo w którym elemencie?

- 1) ustalić x, y
- 2) zidentyfikować element



skąd wiadomo w którym elemencie punkt P ?

tylko we właściwym elemencie
nie będzie kąta $> \pi$

inny pomysł: dla elementu
sprawdzić pole 3 trójkątów jakie można
utworzyć zastępując jeden z wierzchołków
przez P

tylko jeśli P należy do danego elementu
suma pól = pole elementu

3) w elemencie m
wylczyć

$$x_1^m \left(-\frac{\xi_1 + \xi_2}{2} \right) + x_2^m \left(\frac{1 + \xi_1}{2} \right) + x_3^m \left(\frac{1 + \xi_2}{2} \right) = x$$

ξ_1, ξ_2
dla x, y

$$y_1^m \left(-\frac{\xi_1 + \xi_2}{2} \right) + y_2^m \left(\frac{1 + \xi_1}{2} \right) + y_3^m \left(\frac{1 + \xi_2}{2} \right) = y$$

układ równań liniowych na ξ_1, ξ_2

$$\xi_1 = -\frac{-y_2 x_1 - y_3 x_1 + 2 y_1 x_1 - 2 y_1 x + y_2 x_3 + y_1 x_3 - 2 x_3 y + y_1 x_2 + 2 x y_3 - y_3 x_2}{-y_1 x_3 + y_1 x_2 - y_2 x_1 + y_2 x_3 + y_3 x_1 - y_3 x_2}$$

$$\xi_2 := \frac{y_1 x_2 + y_1 x_3 - 2 y_1 x - y_2 x_1 - y_2 x_3 + 2 y_2 x - y_3 x_1 + y_3 x_2 + 2 y_1 x_1 - 2 y_1 x_2}{-y_1 x_3 + y_1 x_2 - y_2 x_1 + y_2 x_3 + y_3 x_1 - y_3 x_2}$$

co pozwala wylczyć u (ξ_1, ξ_2)

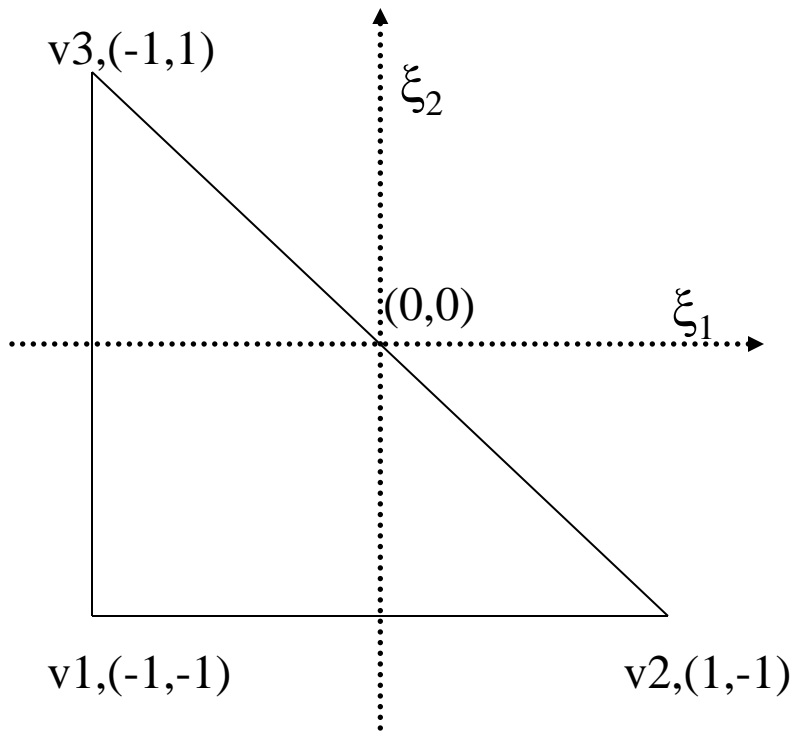
dla biliniowej transformacji (czworokątne elementy)
$$\begin{pmatrix} x \\ y \end{pmatrix} = \sum_{i=1}^4 \begin{pmatrix} x_i \\ y_i \end{pmatrix} \phi_i(\xi_1, \xi_2)$$

$$\phi_1(\xi_1, \xi_2) = \frac{1}{4}(1 + \xi_1)(1 + \xi_2) \quad \phi_3(\xi_1, \xi_2) = \frac{1}{4}(1 - \xi_1)(1 - \xi_2)$$

$$\phi_2(\xi_1, \xi_2) = \frac{1}{4}(1 + \xi_1)(1 - \xi_2) \quad \phi_4(\xi_1, \xi_2) = \frac{1}{4}(1 - \xi_1)(1 + \xi_2)$$

URNL – łatwiej już transformować bazę do p. fizycznej i tam liczyć

**Całkowanie po trójkątnym elemencie odniesienia,
pomysł nr jeden: złożenie 1D kwadratur Gaussa**



$$I = \int_{-1}^1 d\xi_1 \int_{-1}^{-\xi_1} d\xi_2 g(\xi_1, \xi_2)$$

można dwukrotnie zastosować jednowymiarowe całkowanie Gaussa:

$$I = \int_{-1}^1 d\xi_1 \int_{-1}^{-\xi_1} d\xi_2 g(\xi_1, \xi_2) = \int_{-1}^1 d\xi_1 G(\xi_1)$$

$$G(\xi_1) = \int_{-1}^{-\xi_1} d\xi_2 g(\xi_1, \xi_2)$$

$$G(\xi_1) = \int_{-1}^{-\xi_1} d\xi_2 g(\xi_1, \xi_2)$$

przekształcić na całkę od -1 do 1 (w tym przedziale określone punkty Gaussa oraz wagi)

złożenie 1D kwadratur Gaussa

$$t = \frac{2}{1 - \xi_1} \xi_2 + \frac{1 + \xi_1}{1 - \xi_1}$$

nowa zmienna całkowania

$$\xi_2 = \frac{1 - \xi_1}{2} t - \frac{1 + \xi_1}{2}$$

stara współrzędna w nowych

$$G(\xi_1) = \frac{1 - \xi_1}{2} \int_{-1}^1 dt g\left(\xi_1, \frac{1 - \xi_1}{2} t - \frac{1 + \xi_1}{2}\right)$$

punkty Gaussa

$$G(\xi_1) \simeq \frac{1 - \xi_1}{2} \sum_{i=1}^n w_i g\left(\xi_1, \frac{1 - \xi_1}{2} x_i - \frac{1 + \xi_1}{2}\right)$$

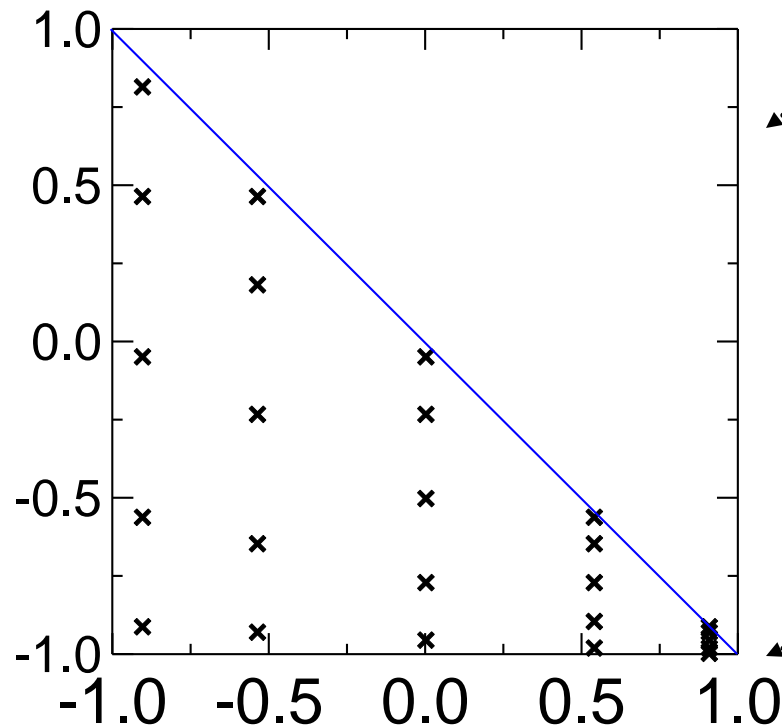
$$I = \int_{-1}^1 d\xi_1 \int_{-1}^{-\xi_1} d\xi_2 g(\xi_1, \xi_2) = \int_{-1}^1 d\xi_1 G(\xi_1)$$

$$I = \sum_{j=1}^n w_j G(x_j) = \sum_{j=1}^n w_j \frac{1 - x_j}{2} \sum_{i=1}^n w_i g\left(x_j, \frac{1 - x_j}{2} x_i - \frac{1 + x_j}{2}\right)$$

modyfikacja wagi

całkowanie po elemencie odniesienia: złożenie 1D kwadratur Gaussa

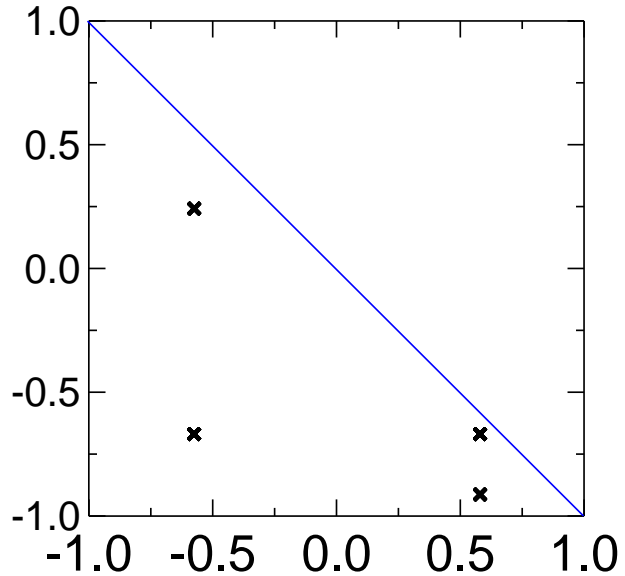
$$I = \sum_{j=1}^n w_j G(x_j) = \sum_{j=1}^n w_j \frac{1-x_j}{2} \sum_{i=1}^n w_i g(x_j, \frac{1-x_j}{2} x_i - \frac{1+x_j}{2})$$



punkty w których
wyliczana funkcja podcałkowa g

nie ma uzasadnienia
asymetrii między liczbą punktów Gaussa
w narożnikach,
w prawym dolnym więcej punktów
a wagi Gaussa mniejsze

całkowanie po elemencie odniesienia: złożenie 1D kwadratur Gaussa



testy:

$$g(\xi_i, \xi_j) = \xi_i^m \xi_j^n \quad (\text{analitycznie można scałkować})$$

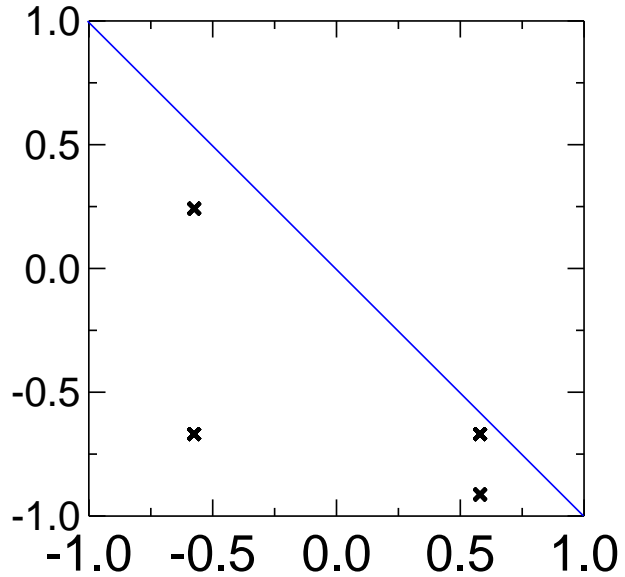
użyto 2-punktowych kwadratur Gaussa 1D
w 1D dokładna dla wielomianów stopnia ≤ 3

jednomiany stopnia $m+n \leq 3$:

m	n	numer.	dokładny	
0	0	2.00000	2.00000	
0	1	-0.66667	-0.66667	
1	0	-0.66667	-0.66667	
0	2	0.66667	0.66667	
2	0	0.66667	0.66667	
1	1	0.00000	0.00000	
.....				
0	3	-0.44444	-0.40000	
3	0	-0.22222	-0.40000	procedura szwankuje.
1	2	-0.07407	-0.13333	numerycznie
2	1	-0.22222	-0.13333	złamana symetria
				ξ_1/ξ_2

dlaczego nie działa
dokładnie dla wielomianów 03/30 ?

całkowanie po elemencie odniesienia: złożenie 1D kwadratur Gaussa



testy:

$$g(\xi_i, \xi_j) = \xi_i^m \xi_j^n \quad (\text{analitycznie można scałkować})$$

użyto 2-punktowych kwadratur Gaussa 1D
w 1D dokładna dla wielomianów stopnia ≤ 3

jednomiany stopnia $m+n \leq 3$:

m	n	numer.	dokładny	
0	0	2.00000	2.00000	
0	1	-0.66667	-0.66667	
1	0	-0.66667	-0.66667	
0	2	0.66667	0.66667	
2	0	0.66667	0.66667	
1	1	0.00000	0.00000	
.....				
0	3	-0.44444	-0.40000	
3	0	-0.22222	-0.40000	procedura szwankuje.
1	2	-0.07407	-0.13333	numerycznie,
2	1	-0.22222	-0.13333	złamana symetria
				ξ_1/ξ_2

dlaczego nie działa
dokładnie dla wielomianów 03/30 ?
bo waga: $(1-x_j)/2$
jest wielomianem
stopnia 1

Ekonomiczne kwadratury Gaussa dla trójkąta odniesienia

$$I = \int_{-1}^1 d\xi_1 \int_{-1}^{-\xi_1} d\xi_2 g(\xi_1, \xi_2) = \sum_k w_k g(\xi_1^k, \xi_2^k)$$

10 niezależnych wielomianów 2D stopnia ≤ 3

1 punkt w całkowaniu Gaussa = 3 stopnie swobody (2-położenie, 1-waga)

Dla dokładnego scałkowania wielomianów stopnia 3: 4 punkty powinny wystarczyć

$$\sum_{k=1}^4 w_k = 2$$

całka z jedynki

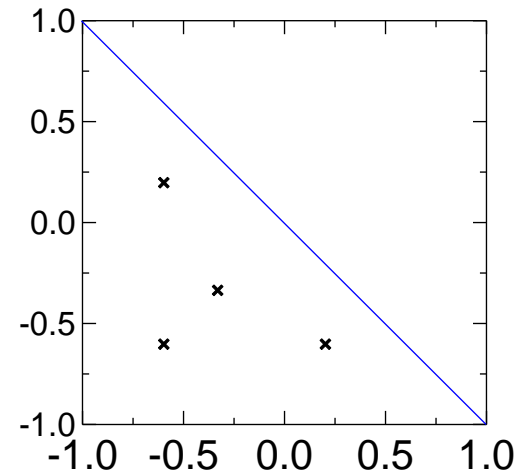
$$\sum_{k=1}^4 w_k \xi_1^{(k)} = -\frac{2}{3}$$

całka z ξ_1

$$\sum_{k=1}^4 w_k \xi_2^{(k)} = -\frac{2}{3}$$

itd.

$$\sum_{k=1}^4 w_k \xi_1^{(k)} \xi_2^{(k)} = 0$$



ξ_1 -Coordinate	ξ_2 -Coordinate	Weight
-0.33333 33333 33333	-0.33333 33333 33333	-1.12500 00000 00000
-0.60000 00000 00000	-0.60000 00000 00000	1.04166 66666 66667
-0.60000 00000 00000	0.20000 00000 00000	1.04166 66666 66667
0.20000 00000 00000	-0.60000 00000 00000	1.04166 66666 66667

za Pawłem Solinem
PDE and FEM
Wiley-Interscience 2006

Ekonomiczne kwadratury Gaussa dla trójkąta odniesienia

wielomiany stopnia 1 [1,x,y]

ξ_1 -Coordinate	ξ_2 -Coordinate	Weight
-0.33333 33333 33333	-0.33333 33333 33333	2.00000 00000 00000

skąd się bierze: (wzory skopiowane z poprzedniej strony)

1
4

$$\sum_{k=1}^4 w_k = 2 \quad \leftarrow \text{jedna waga}$$

1
4

$$\sum_{k=1}^4 w_k \xi_1^{(k)} = -\frac{2}{3} \quad \rightarrow -1/3$$

1
4

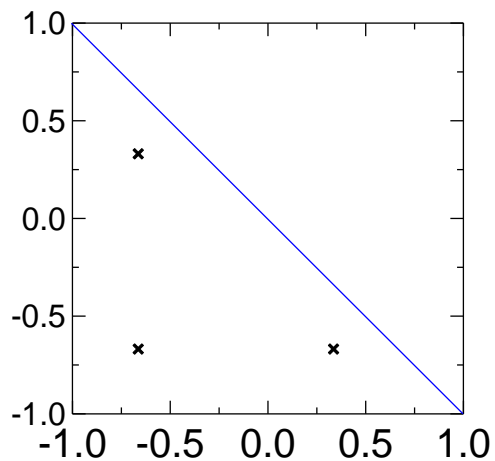
$$\sum_{k=1}^4 w_k \xi_2^{(k)} = -\frac{2}{3} \quad \rightarrow -1/3$$

$$\sum_{k=1}^4 w_k \xi_1^{(k)} \xi_2^{(k)} = 0$$

Ekonomiczne kwadratury Gaussa dla trójkąta odniesienia

wielomiany stopnia 2 [1,x,y,x²,y²,xy]

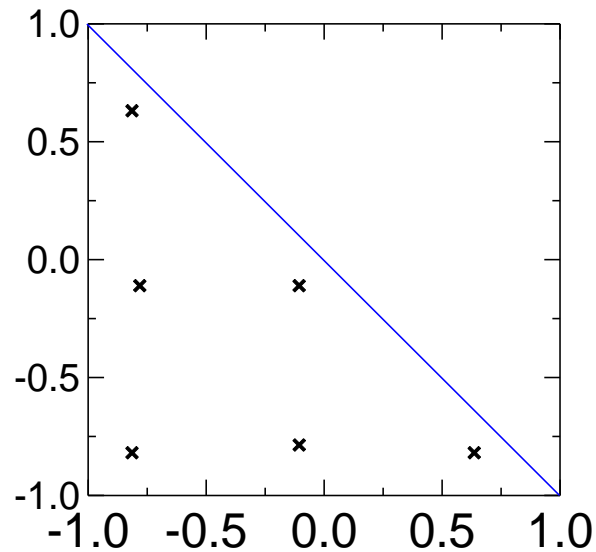
ξ_1 -Coordinate	ξ_2 -Coordinate	Weight
-0.66666 66666 66667	-0.66666 66666 66667	0.66666 66666 66667
-0.66666 66666 66667	0.33333 33333 33333	0.66666 66666 66667
0.33333 33333 33333	-0.66666 66666 66667	0.66666 66666 66667



Ekonomiczne kwadratury Gaussa dla trójkąta odniesienia

wielomiany stopnia 4

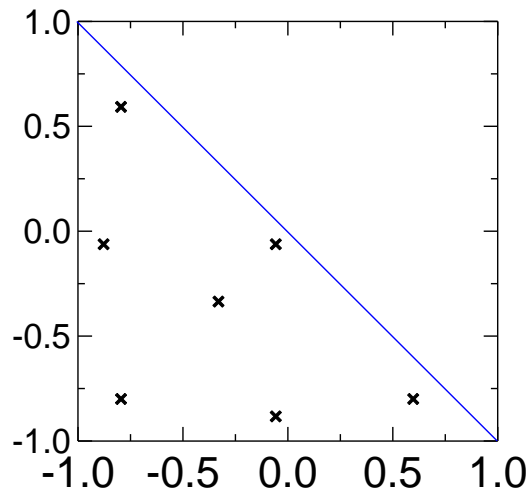
ξ_1 -Coordinate	ξ_2 -Coordinate	Weight
-0.10810 30181 68070	-0.10810 30181 68070	0.44676 31793 56022
-0.10810 30181 68070	-0.78379 39636 63860	0.44676 31793 56022
-0.78379 39636 63860	-0.10810 30181 68070	0.44676 31793 56022
-0.81684 75729 80458	-0.81684 75729 80458	0.21990 34873 10644
-0.81684 75729 80458	0.63369 51459 60918	0.21990 34873 10644
0.63369 51459 60918	-0.81684 75729 80458	0.21990 34873 10644



Ekonomiczne kwadratury Gaussa dla trójkąta odniesienia

wielomiany stopnia 5

ξ_1 -Coordinate	ξ_2 -Coordinate	Weight
-0.33333 33333 33333	-0.33333 33333 33333	0.45000 00000 00000
-0.05971 58717 89770	-0.05971 58717 89770	0.26478 83055 77012
-0.05971 58717 89770	-0.88056 82564 20460	0.26478 83055 77012
-0.88056 82564 20460	-0.05971 58717 89770	0.26478 83055 77012
-0.79742 69853 53088	-0.79742 69853 53088	0.25187 83610 89654
-0.79742 69853 53088	0.59485 39707 06174	0.25187 83610 89654
0.59485 39707 06174	-0.79742 69853 53088	0.25187 83610 89654



elementy trójkątne: różniczkowanie w przestrzeni odniesienia

$$\nabla\phi_i \cdot \nabla\phi_j = \left(\sum_k \frac{\partial\phi_i}{\partial\xi_k} \frac{\partial\xi_k}{\partial x} \right) \times \left(\sum_k \frac{\partial\phi_j}{\partial\xi_k} \frac{\partial\xi_k}{\partial x} \right) + \left(\sum_k \frac{\partial\phi_i}{\partial\xi_k} \frac{\partial\xi_k}{\partial y} \right) \times \left(\sum_k \frac{\partial\phi_j}{\partial\xi_k} \frac{\partial\xi_k}{\partial y} \right)$$

$$\begin{pmatrix} \frac{\partial\xi_1}{\partial x} & \frac{\partial\xi_1}{\partial y} \\ \frac{\partial\xi_2}{\partial x} & \frac{\partial\xi_2}{\partial y} \end{pmatrix} = \frac{1}{\frac{\partial x}{\partial\xi_1} \frac{\partial y}{\partial\xi_2} - \frac{\partial y}{\partial\xi_1} \frac{\partial x}{\partial\xi_2}} \begin{pmatrix} \frac{\partial y}{\partial\xi_2} & -\frac{\partial x}{\partial\xi_2} \\ -\frac{\partial y}{\partial\xi_1} & \frac{\partial x}{\partial\xi_1} \end{pmatrix}$$

Mapowanie:

$$f_1 = -\frac{\xi_1 + \xi_2}{2} \quad f_2 = \frac{1 + \xi_1}{2} \quad f_3 = \frac{1 + \xi_2}{2}$$

$$\begin{pmatrix} x \\ y \end{pmatrix} = \sum_{i=1}^3 \begin{pmatrix} x_i \\ y_i \end{pmatrix} f_i(\xi_1, \xi_2)$$

$$x = \xi_1 \left(-\frac{x_1}{2} + \frac{x_2}{2} \right) + \xi_2 \left(-\frac{x_1}{2} + \frac{x_3}{2} \right) + \frac{x_2}{2} + \frac{x_3}{2}$$

$$y = \xi_1 \left(-\frac{y_1}{2} + \frac{y_2}{2} \right) + \xi_2 \left(-\frac{y_1}{2} + \frac{y_3}{2} \right) + \frac{y_2}{2} + \frac{y_3}{2}$$

$$\frac{dx}{d\xi_1} = -\frac{x_1}{2} + \frac{x_2}{2}$$

$$\frac{dx}{d\xi_2} = -\frac{x_1}{2} + \frac{x_3}{2}$$

$$\frac{dy}{d\xi_1} = -\frac{y_1}{2} + \frac{y_2}{2}$$

$$\frac{dy}{d\xi_2} = -\frac{y_1}{2} + \frac{y_3}{2}$$

elementy trójkątne: Jakobian mapowania elementu odniesienia do elementu w przestrzeni fizycznej

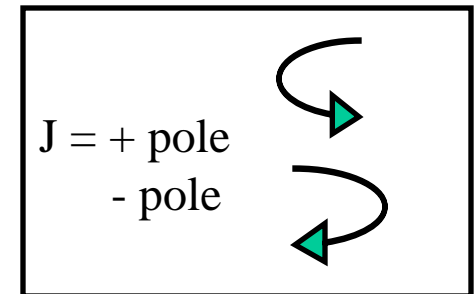
Jakobian:

$$J = \frac{dx}{d\xi_1} \frac{dy}{d\xi_2} - \frac{dy}{d\xi_1} \frac{dx}{d\xi_2} = \frac{1}{4} [(-x_1 + x_2)(-y_1 + y_3) - (-y_1 + y_2)(-x_1 + x_3)]$$

NIGDY nie zależy od położenia jak w ogólnym przypadku elementów czworokątnych

widzimy, że

$$\Delta = \frac{1}{2} \begin{vmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{vmatrix}$$



$$J = \frac{\text{fizyczne pole powierzchni elementu}}{\text{pole powierzchni trójkąta odniesienia}}$$

↗
2

mapowanie odwracalne o ile
3 wierzchołki nie są współliniowe

$$\text{fizyczne pole powierzchni elementu} = \int_{\text{element}} dx dy = \int_{-1}^1 d\xi_1 \int_{-1}^{-\xi_1} d\xi_2 J$$

całkowanie macierzy sztywności:

$$\int_{\text{element}} \nabla \phi_1 \cdot \nabla \phi_2 dx dy = J \int_{-1}^1 d\xi_1 \int_{-1}^{-\xi_1} d\xi_2 \left(\sum_k \frac{\partial \phi_i}{\partial \xi_k} \frac{\partial \xi_k}{\partial x} \right) \times \left(\sum_k \frac{\partial \phi_j}{\partial \xi_k} \frac{\partial \xi_k}{\partial x} \right) + \left(\sum_k \frac{\partial \phi_i}{\partial \xi_k} \frac{\partial \xi_k}{\partial y} \right) \times \left(\sum_k \frac{\partial \phi_j}{\partial \xi_k} \frac{\partial \xi_k}{\partial y} \right)$$

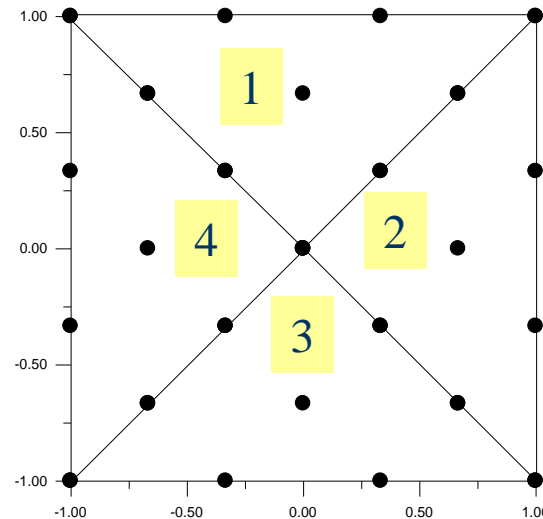
całkowanie wektora obciążeń

$$\int_{\text{element}} \phi_i(x, y) \rho(x, y) dx dy = J \int_{-1}^1 d\xi_1 \int_{-1}^{-\xi_1} d\xi_2 \phi_i(\xi_1, \xi_2) \rho(x(\xi_1), y(\xi_2))$$

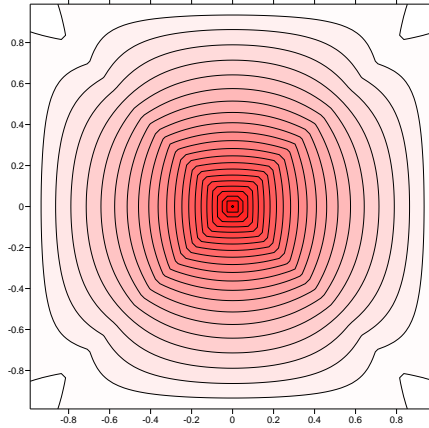
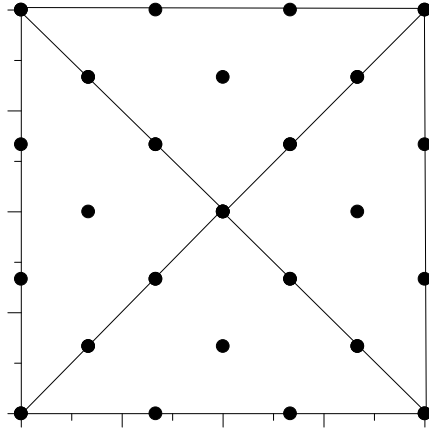
Zastosowanie:

4 elementy,
kubiczne funkcje kształtu
10 węzłów / element
25 różnych węzłów

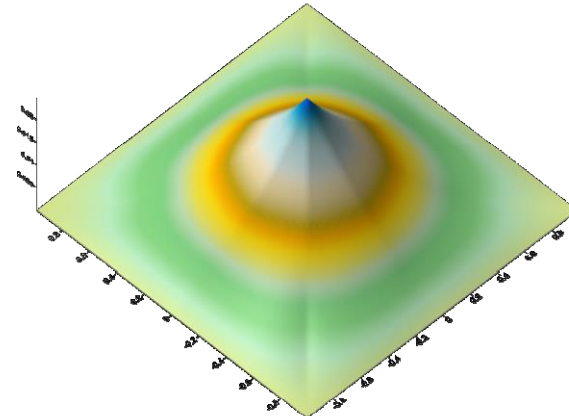
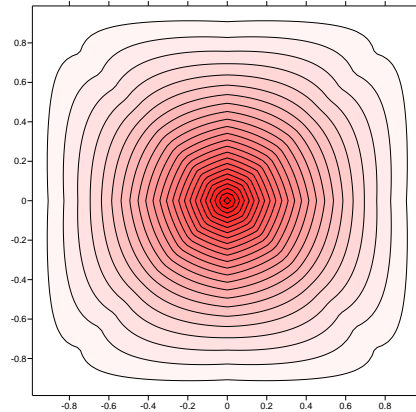
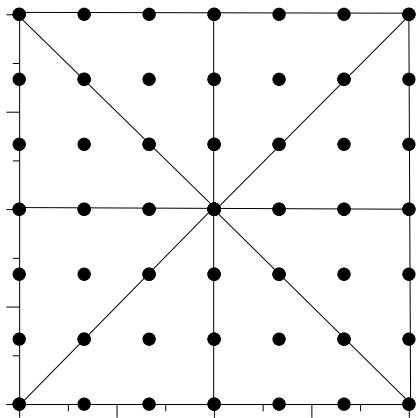
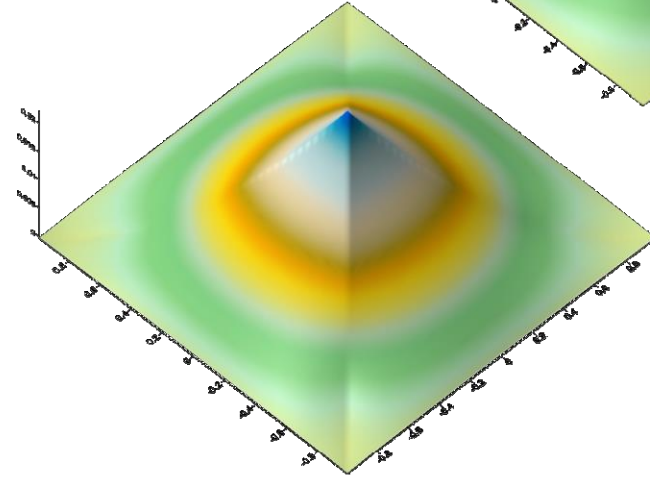
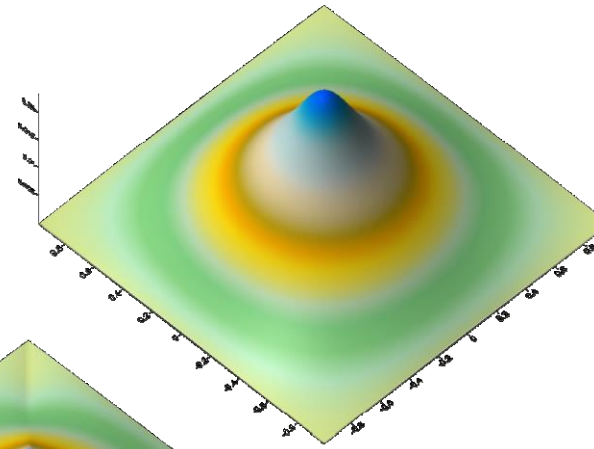
ponumerujemy elementy narożne
przeciwnie do ruchu wskazówek
zegara, wtedy $J=1/2$ dla każdego
elementu



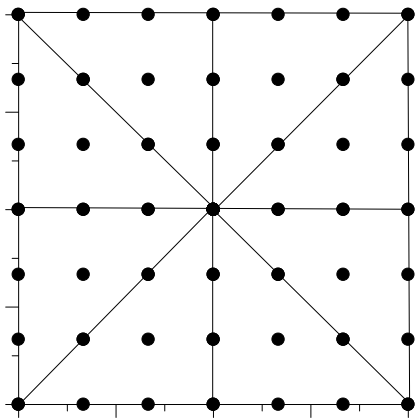
Wyniki



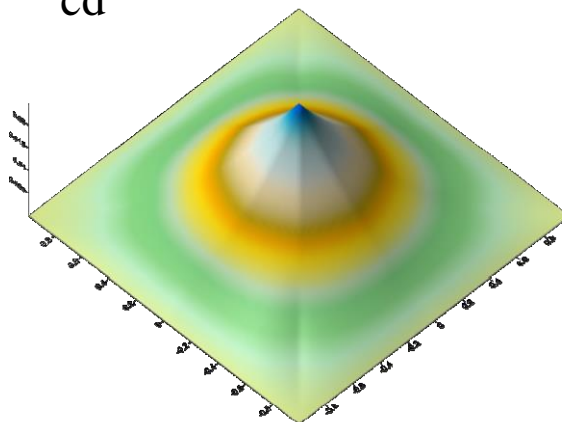
dokładny:



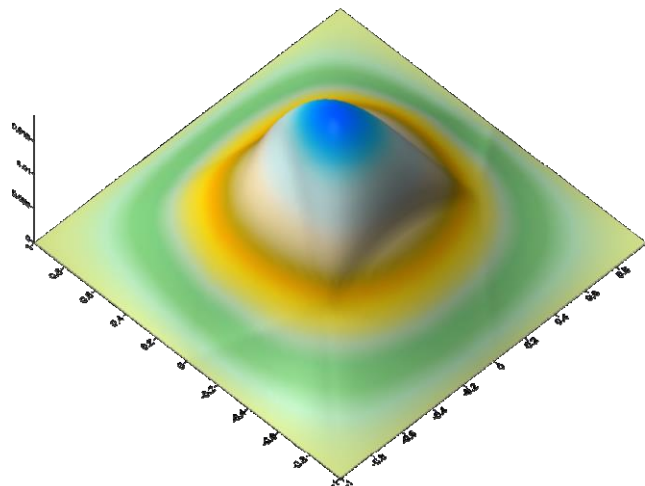
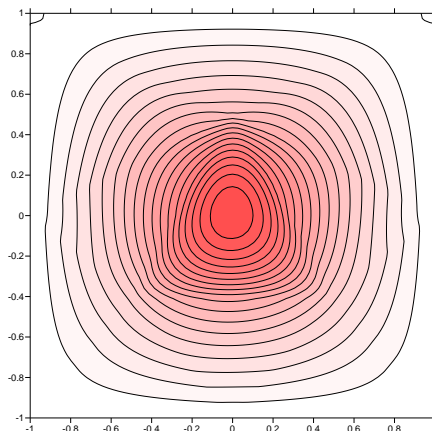
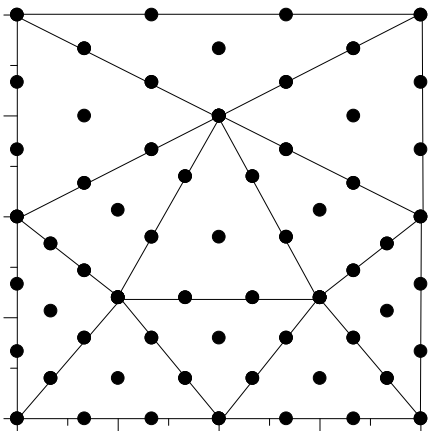
całkowanie: E – wystarczy ekonomiczna k.G. dla rzędu czwartego. W implementacji możliwy problem z wektorem obciążeń – k. zbyt ekonomiczna przy całkowaniu od środka do brzegu



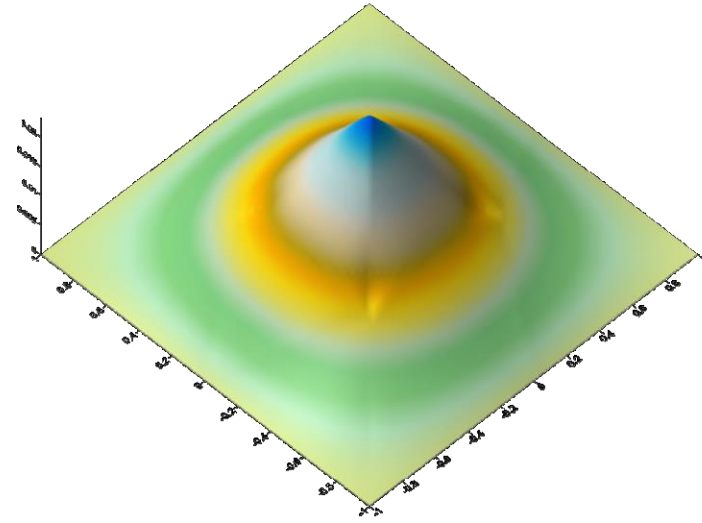
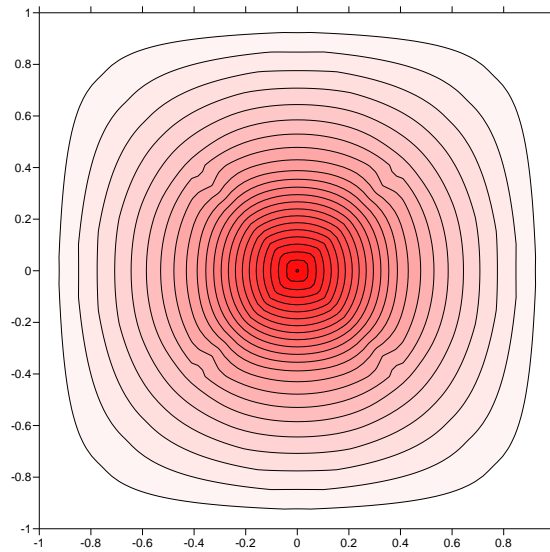
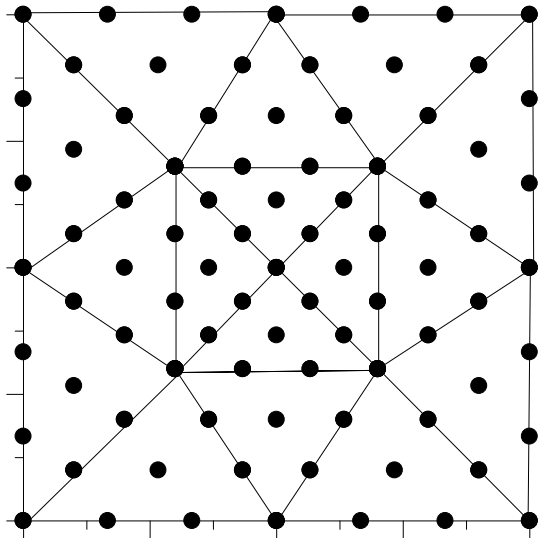
cd



umiarkowany sukces (kant).
 metoda generuje funkcje
 kawałkami kubiczne.
 druga pochodna z rozwiązania
 dokładnego = gaussian.
 na całości: od centrum do brzegu
 słabe szanse.



pomysł na pozbycie się centralnego kantu
 - wykorzystać funkcję bąbelkową centralnego elementu



Widzieliśmy, że w MES podział obszaru całkowania na elementy ma kluczowe znaczenie dla jakości rozwiązania.

Jak dobrać węzły?: np.. korzystając z kryterium funkcjonału działania układu poleładunek była minimalna.

Potrzebny algorytm automatycznego generowania siatki.

Uwaga: w modelowaniu przemysłowym z MES generacja siatki zajmuje około 3/4 całego czasu

Automatyczna generacja siatki

w MES jakość rozwiązania zależy od podziału przestrzeni na elementy.

do tej pory elementy wybieraliśmy ręcznie lub

optymalizowaliśmy na podstawie minimum funkcjonału (w 1D).

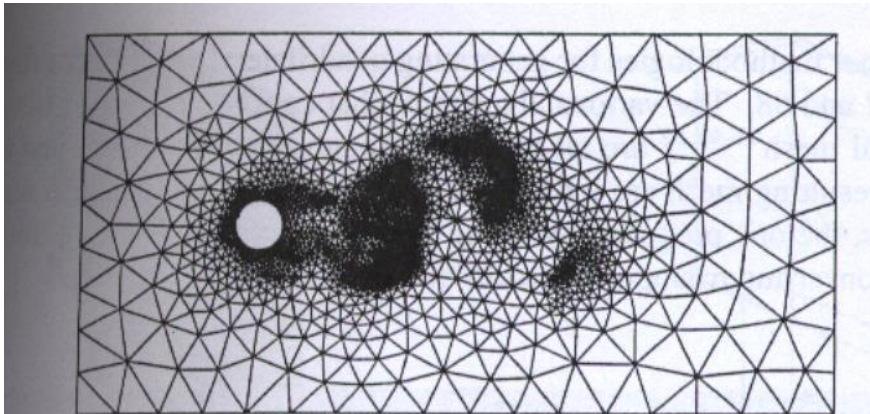
Siatkę można i należy generować automatycznie.

Kryterium doboru siatki: np. wkład do funkcjonału działania od pojedynczego elementu lub lokalny błąd (np. kwadrat residuum).

Automatyczne algorytmy dzielą na trójkąty (2D) i tetraedry (3D)

podjęcia:

- 1) Metoda *advancing front (triangulation)* postępującego frontu (*front* – także jako czoło fali) generacja węzłów w ramach obszaru całkowania.
- 2) Metoda triangulacji Delaunay'a: dla danych węzłów konstrukcja możliwie regularnych elementów
- 3) Rekurencyjne zagęszczanie siatki i inne



Zienkiewicz

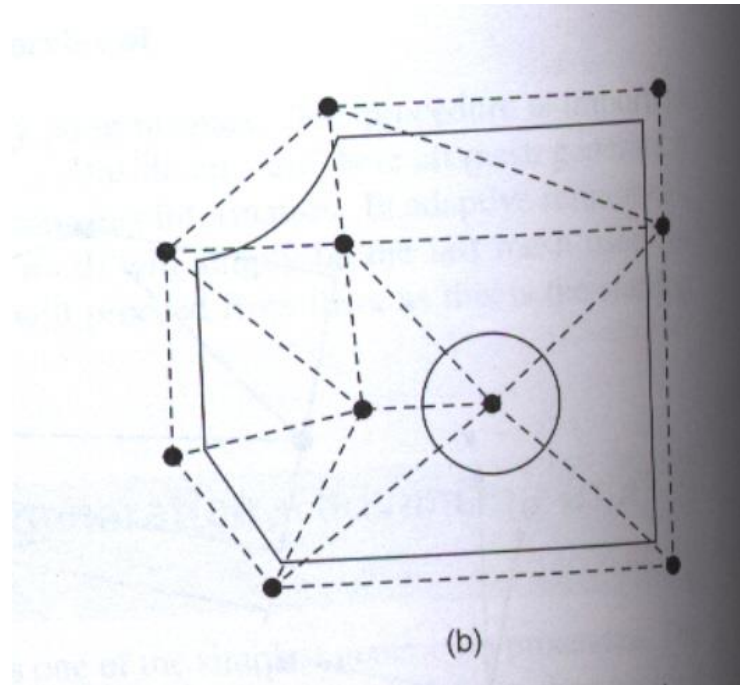
FEM basis & fundamentals

Generacja siatki

3 główne kroki algorytmu

- 1) Generacja węzłów na krawędzi obszaru całkowania (**dyskretyzacja brzegu**)
- 2) Generacja węzłów i elementów wewnątrz obszaru całkowania
(tutaj możliwe różne warianty : triangulacja / postępujący front)
- 3) Rafinacja siatki

zanim wygenerowana zostanie siatka właściwa
zadana jest **siatka tła**: (zawiera naszą wiedzę o układzie: przewidywaną gęstość
i optymalną orientację elementów)



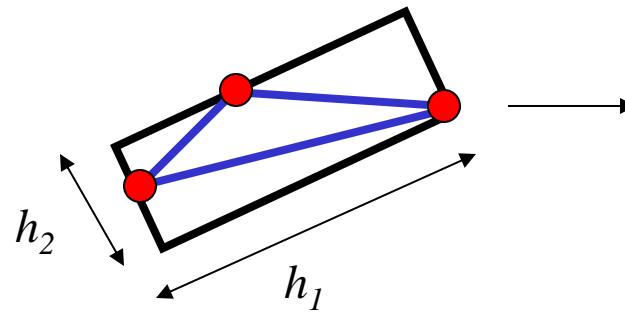
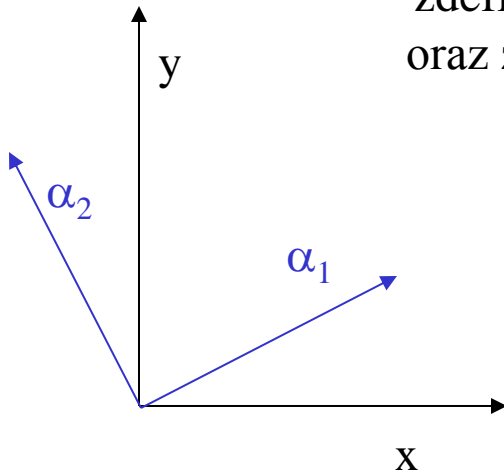
Zienkiewicz, FEM, basis and fundamentals

lokalne rozmiary elementów oraz ich orientacja
informacje zawarte w siatce tła służą do wygenerowania siatki właściwej

siatka tła: lokalne rozmiary i orientacja elementów

Kształt elementów ich rozmiar i orientację opisują *parametry siatki* (zależne od położenia).

Parametry siatki: kierunki rozciągnięcia (*stretching dir.*) 2 ortogonalne kierunki zdefiniowane przez wektory jednostkowe α_1, α_2 oraz związane z nimi rozmiary h_1, h_2



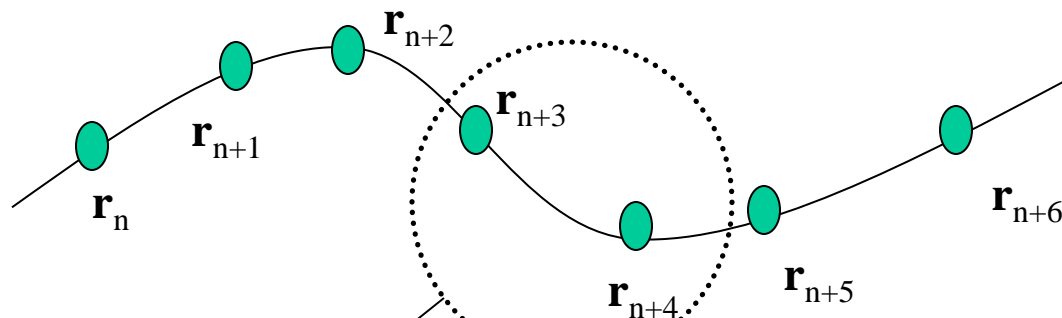
Jeśli nie ma powodu aby elementy były *rozciągnięte* w jednym z kierunków α_1, α_2 – wersory x/y i $h_1=h_2=h$

siatka tła: ma wpływ zarówno na generację węzłów wewnątrz obszaru całkowania jak i na generację węzłów brzegowych (dyskretyzację brzegu)

dyskretyzacja brzegu (generacja węzłów brzegowych)

cegiełka 1: parametryzacja krzywej brzegowej.

Brzeg dzielimy na segmenty przez wybór punktów próbkujących \mathbf{r}_i (*to nie będą węzły MES*).



w ramach jednego

segmentu krzywą brzegową parametryzujemy lokalnie (t wartości z $[-1,1]$)

brzeg ciągły z krzywizną (pochodna x/y)

- funkcje Hermite'a:

$$x(t) = x(-1)\phi_1^0(t) + x(1)\phi_2^0(t) + x'(-1)\phi_1^1(t) + x'(1)\phi_2^1(t)$$

$$y(t) = y(-1)\phi_1^0(t) + y(1)\phi_2^0(t) + y'(-1)\phi_1^1(t) + y'(1)\phi_2^1(t)$$

$$x' = \frac{dx(t)}{dt}$$

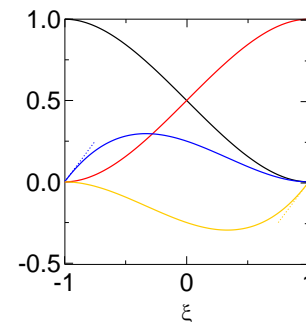
$$y' = \frac{dy(t)}{dt}$$

$$\phi_1^0 = \frac{1}{2} - \frac{3}{4}\xi + \frac{1}{4}\xi^3$$

$$\phi_2^0 = \frac{1}{2} + \frac{3}{4}\xi - \frac{1}{4}\xi^3$$

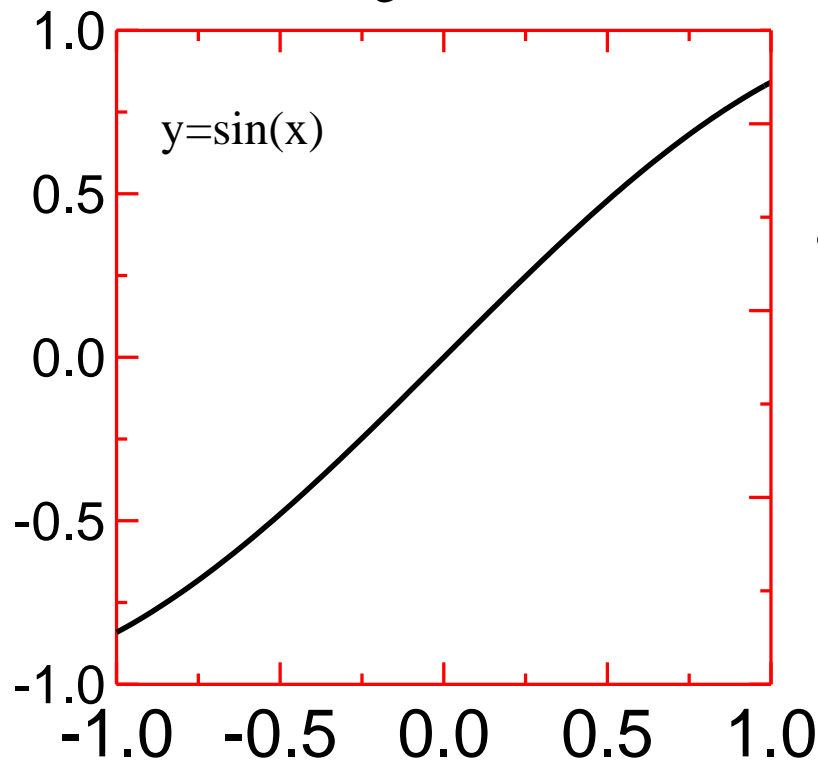
$$\phi_1^1(t) = \frac{1}{4}(1 - t - t^2 + t^3)$$

$$\phi_2^1(t) = \frac{1}{4}(-1 - t + t^2 + t^3)$$



przykład 1:

sinus, segment [-1,1]

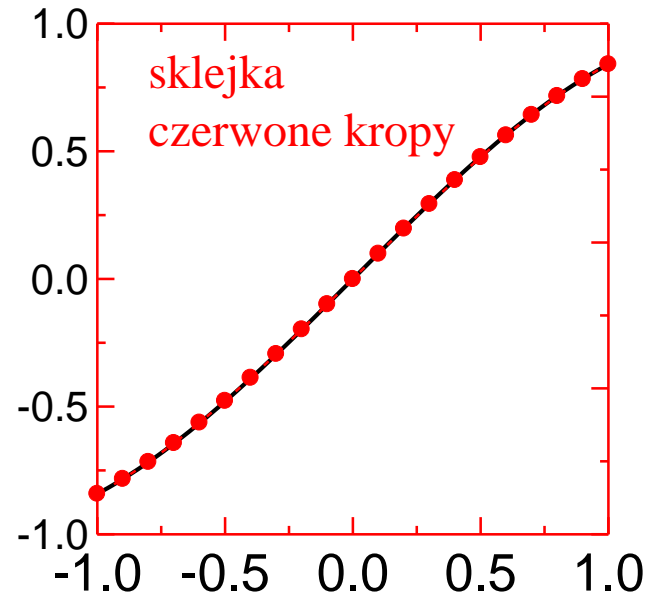


$$\begin{aligned} x &= t \\ y &= \sin(t) \end{aligned}$$



$$-\sin(1) \psi_{10} + \sin(1) \psi_{20} + \cos(1) \psi_{11} + \cos(1) \psi_{21}$$

$$\frac{3}{2} \sin(1) t - \frac{1}{2} \sin(1) t^3 - \frac{1}{2} \cos(1) t + \frac{1}{2} \cos(1) t^3$$

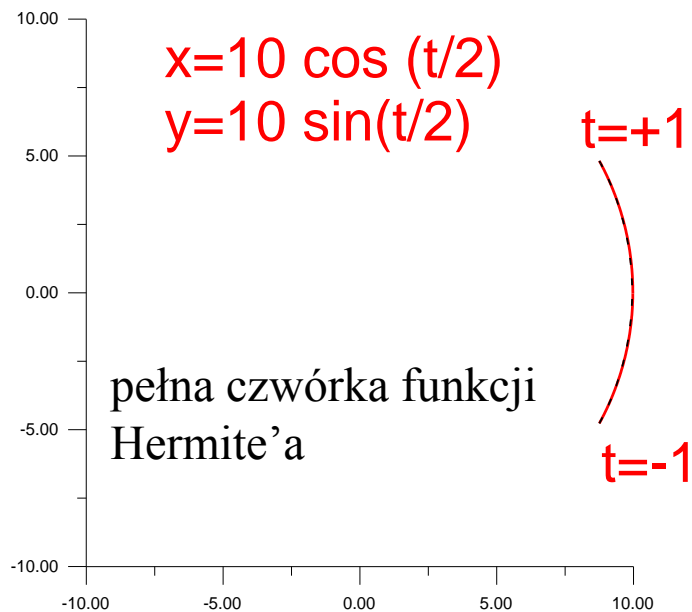
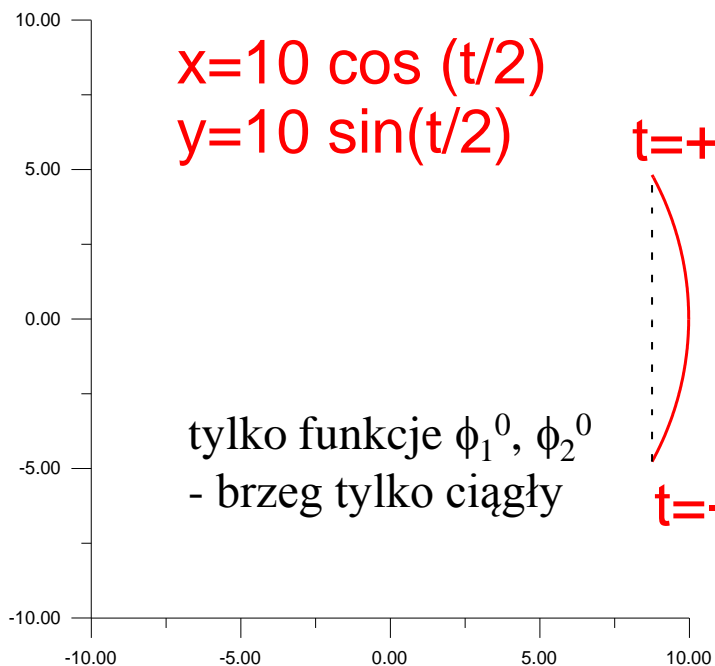


$$x(t) = x(-1)\phi_1^0(t) + x(1)\phi_2^0(t) + x'(-1)\phi_1^1(t) + x'(1)\phi_2^1(t)$$

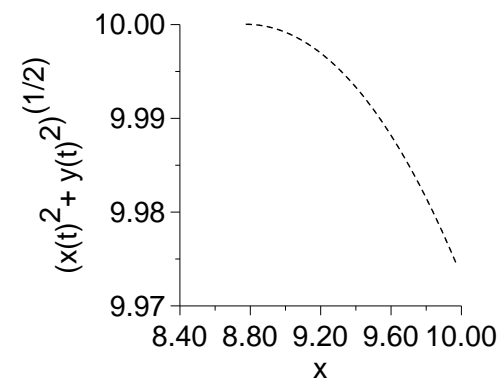
$$y(t) = y(-1)\phi_1^0(t) + y(1)\phi_2^0(t) + y'(-1)\phi_1^1(t) + y'(1)\phi_2^1(t)$$

przykład 2:

segment-fragment koła



precyzja



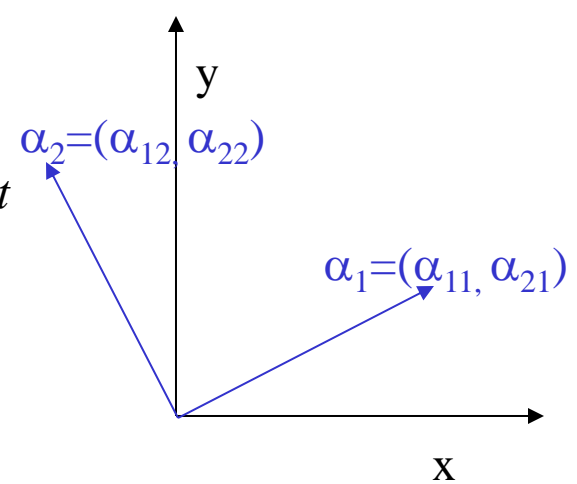
cegiełka 2: transformacja do przestrzeni znormalizowanej

przestrzeń znormalizowana: taka, w której trójkątny element wygenerowany zgodnie z informacją z siatki tła jest prawie równoboczny

wprowadzamy zależną od położenia (zależność podana w siatce tła) transformację T:

$$T = \sum_{i=1}^2 \frac{1}{h_i} \begin{pmatrix} \alpha_{1i} \\ \alpha_{2i} \end{pmatrix} \begin{pmatrix} \alpha_{1i} & \alpha_{2i} \end{pmatrix}$$

transformacja: $\mathbf{r}' = T\mathbf{r}$



transformacja działając na wektor wykonuje 3 operacje:

- 1 mnożenie : rzutowanie na kierunek α_i
- 2 mnożenie: zachowany kierunek α_i
- 3 mnożenie: redukcja długości

narzuca skalę $1/h_i$ na każdy z dwóch kierunków:

efekt – trójkąt [element] (a,b,c) w układzie (x,y) o skalach h_1, h_2

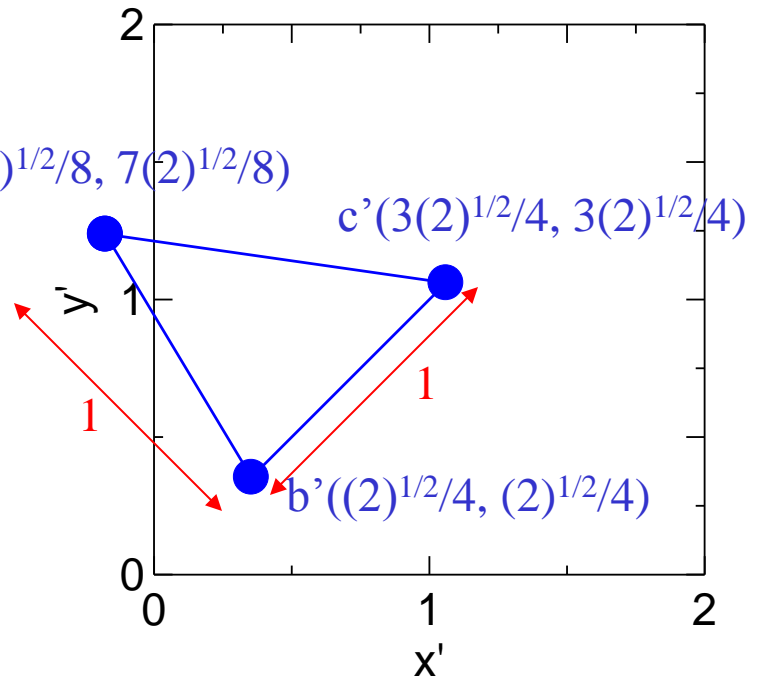
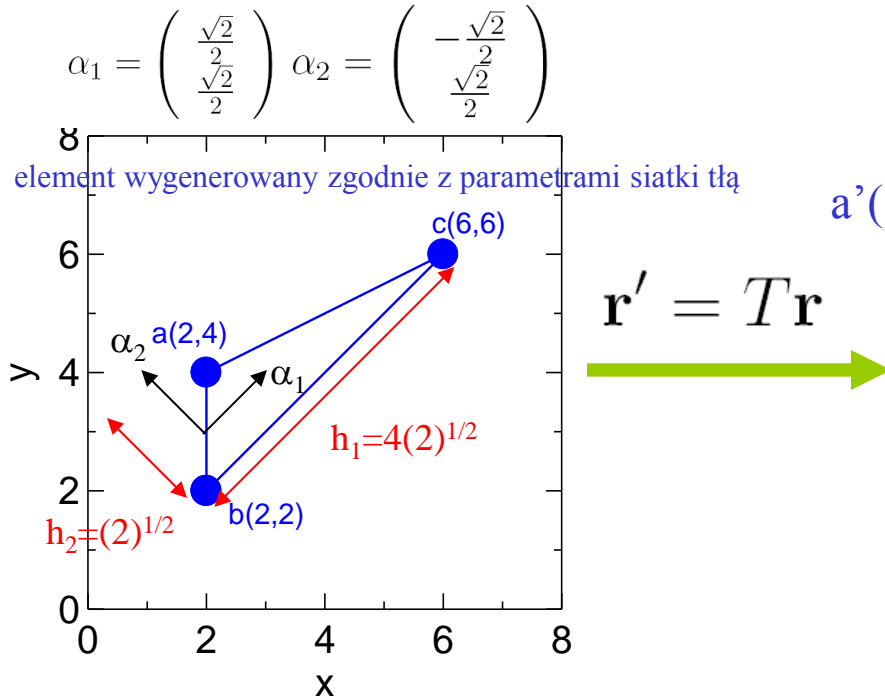
zostaje przekształcony w element o jednostkowych w układzie (x',y') rozmiarach [a więc w miarę równoboczny]

łącznie mnożenie macierzy $T = \sum_{i=1}^2 \frac{1}{h_i} \begin{pmatrix} \alpha_{1i}^2 & \alpha_{1i}\alpha_{2i} \\ \alpha_{1i}\alpha_{2i} & \alpha_{2i}^2 \end{pmatrix}$

przestrzeń fizyczna

Przykład

przestrzeń znormalizowana



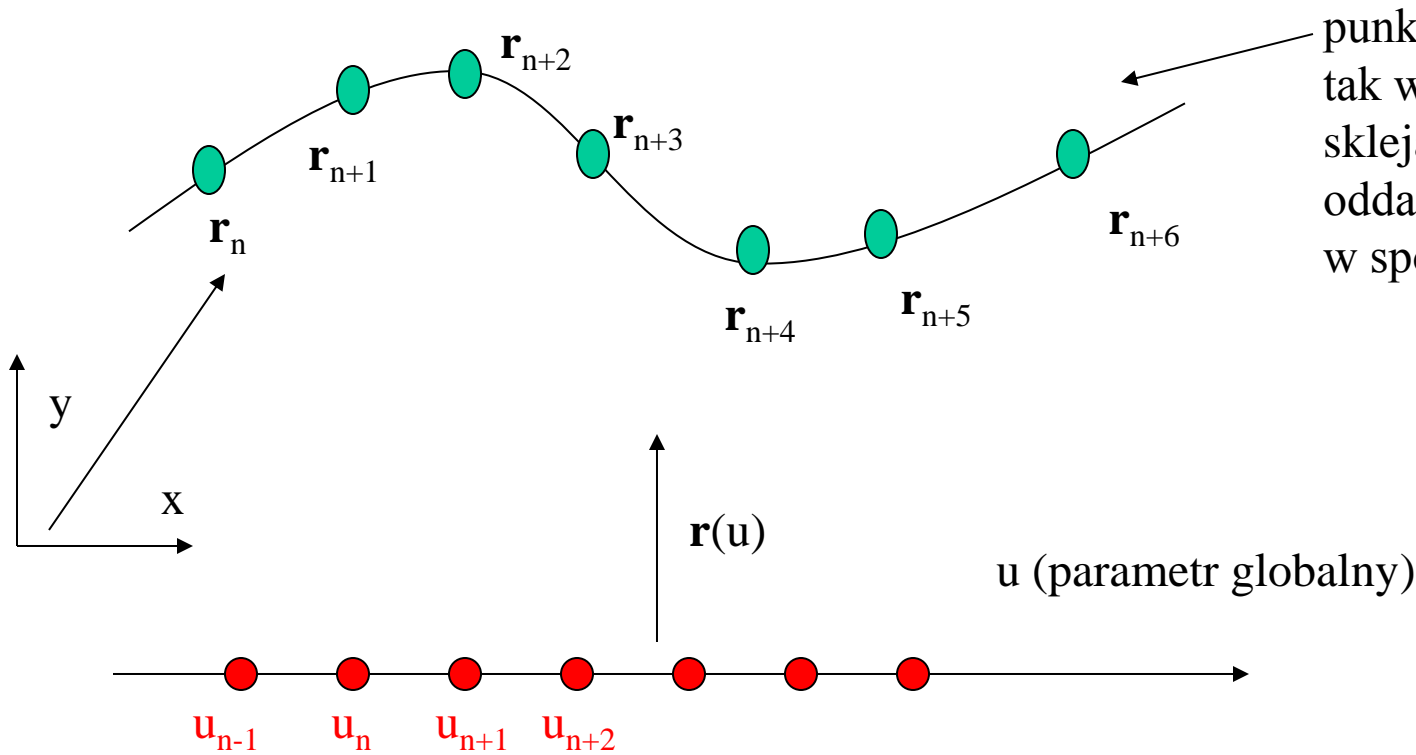
$$T = \sum_{i=1}^2 \frac{1}{h_i} \begin{pmatrix} \alpha_{1i}^2 & \alpha_{1i}\alpha_{2i} \\ \alpha_{1i}\alpha_{2i} & \alpha_{2i}^2 \end{pmatrix}$$

$$T = \frac{\sqrt{2}}{8} \begin{pmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} \end{pmatrix} + \frac{\sqrt{2}}{2} \begin{pmatrix} \frac{1}{2} & -\frac{1}{2} \\ -\frac{1}{2} & \frac{1}{2} \end{pmatrix}$$

rozmiar trójkąta = 1 w każdym z (ortogonalnych) kierunków rozciągania (*stretching*)

trójkąt nie jest idealnie równoboczny
 $|b'c'| > 1$, 1 to wysokość trójkąta z wierzchołka c'

cegielka 2bis: Parametryzacja krzywej brzegowej



można przyjąć taką parametryzację,

że punkty próbkujące równoodległe $u_n - u_{n-1} = \Delta$
w segmencie $[u_{n-1}, u_n]$ odpowiedniość: $u := u_{n-1} + (t+1) \Delta/2$

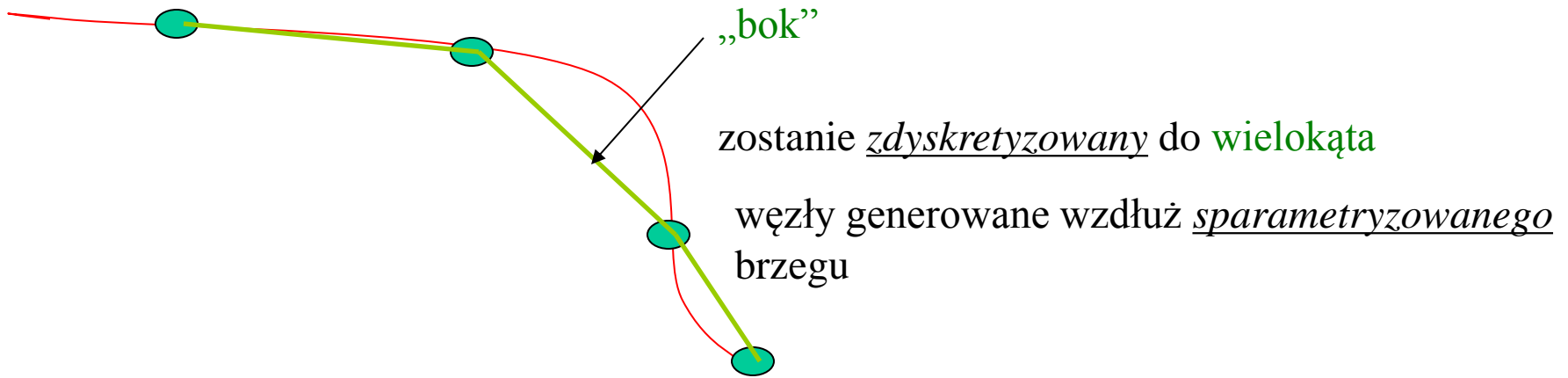
pochodne po t

wtedy:
$$\mathbf{r}(u) = \phi_1^0(t)\mathbf{r}(u_{i-1}) + \phi_2^0(t)\mathbf{r}(u_i) + \phi_1^1(t)\mathbf{r}'(u_{i-1}) + \phi_2^1(t)\mathbf{r}'(u_i)$$

[jeśli zmieniać pochodne z t do u pojawi się $\Delta/2$ przed dwoma ostatnimi wyrazami]

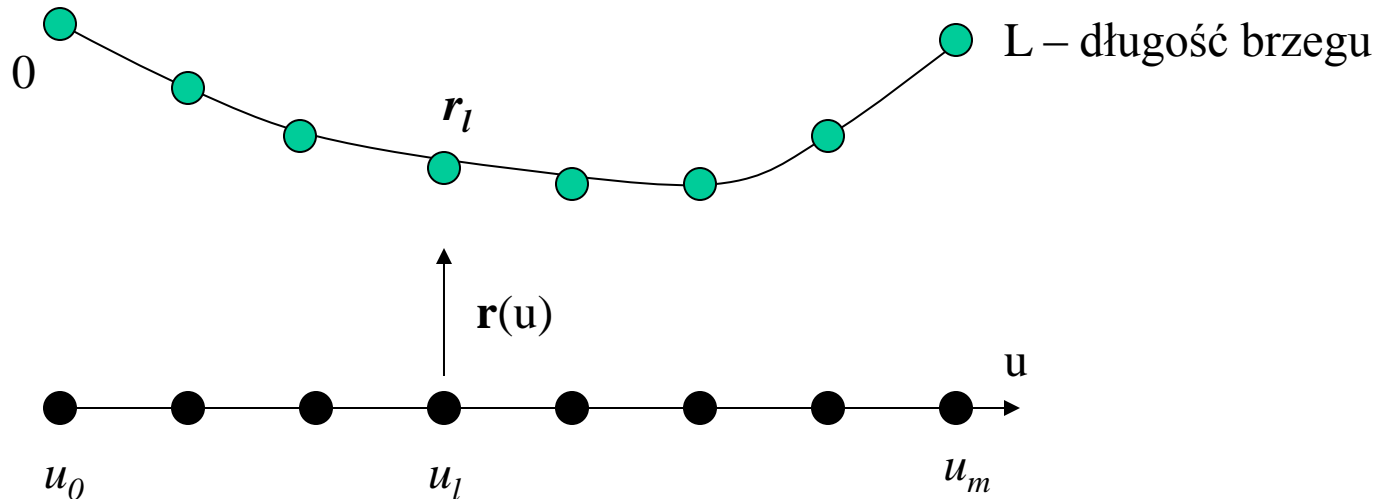
Dyskretyzacja brzegu (wybór węzłów brzegowych dla MES): algorytm

brzeg opisany jedną krzywą (sparametryzowaną sklejką Hermite'a)

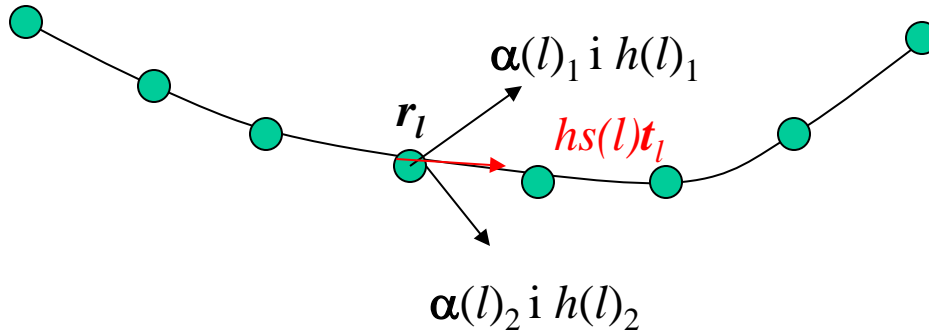


7 – krokowa procedura (najpierw opis, przykład później)

- 1) fragment brzegu sparametryzowany przez u . wybieramy punkty próbkujące $\mathbf{r}_l = \mathbf{r}(u_l)$ ($l=0,1,2,\dots,m$), z parametrami próbkującymi u_l rozmieszczonymi równomiernie



- 2) W każdym z punktów próbkujących: określane parametry siatki a_i, h_i (z interpolacji wielkości zadanych na siatce tła). Macierz transformacji T_l wyliczana w każdym z punktów.



wersor styczny

do krzywej w punkcie l : $\mathbf{t}_l = \begin{pmatrix} t_1(l) \\ t_2(l) \end{pmatrix}$

$$t_1(l) = \frac{x'}{\sqrt{x'^2 + y'^2}}$$

$$t_2(l) = \frac{y'}{\sqrt{x'^2 + y'^2}}$$

tutaj

$$x' = \frac{dx}{du}(u_l)$$

$$y' = \frac{dy}{du}(u_l)$$

- 3) Aby wyznaczyć położenia węzłów: należy **wyznaczyć** funkcję rozkładu rozmiarów elementów wzdłuż krawędzi

$$\boldsymbol{\tau}_l = hs(l)\mathbf{t}_l \quad \text{wektor styczny do krzywej o lokalnej długości łuku}$$

mapujemy wektor $\boldsymbol{\tau}_l$ do przestrzeni znormalizowanej

$$T_l \boldsymbol{\tau}_l = T_l (hs(l)\mathbf{t}_l) = hs(l)T_l \mathbf{t}_l$$

$$T_l \tau_l = T_l (hs(l) \mathbf{t}_l) = hs(l) T_l \mathbf{t}_l$$

żądamy aby w przestrzeni znormalizowanej wektor ten miał długość 1

$$\sqrt{(T_l \tau_l)^T (T_l \tau_l)} = hs(l) \sqrt{(T_l \mathbf{t}_l)^T (T_l \mathbf{t}_l)} = 1$$

stąd,

$$hs(l) = \frac{1}{\sqrt{(T_l \mathbf{t}_l)^T (T_l \mathbf{t}_l)}} \quad (\mathbf{AB})^T = \mathbf{B}^T \mathbf{A}^T$$

$$hs(l) = \frac{1}{\sqrt{t_l^T C_l t_l}}$$

gdzie:

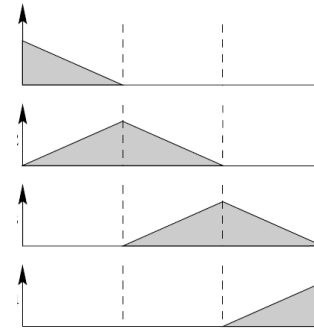
$$C = T^T T = \sum_{i=1}^2 \left(\frac{1}{h_i}\right)^2 \alpha_i \alpha_i^T \quad \begin{array}{l} \text{2D euklidesowy tensor metryczny} \\ \text{w przestrzeni znormalizowanej} \end{array}$$

$$hs(l) = \frac{1}{\sqrt{(T_l t_l)^T (T_l t_l)}} \quad [\text{wyznaczymy na podstawie siatki tła i parametryzacji brzegu}]$$

- 4) Niech $h(u)$ – ciągła funkcja parametru u opisująca rozmiar elementów.
 Zbudujemy ją przy pomocy liniowych funkcji kształtu (Lagrange'a) dla 1D elementów skończonych interpolacyjnych Lagrange'a w węzłach u_l : $hs(l) = h(u)$

pamiętamy:

$$v_i(x) = \begin{cases} \frac{x-x_{i-1}}{x_i-x_{i-1}} & x \in K_i \\ \frac{x_{i+1}-x}{x_{i+1}-x_i} & x \in K_{i+1} \\ 0 & x \notin K_i \cup K_{i+1} \end{cases}$$



$$h(u) = \sum_{l=0}^m hs(l) v_l(u)$$

gęstość wzdłuż linii brzegowe elementów w funkcji parametru u dana przez $1/h(u)$

- 5) A – idealna liczba elementów, która ma zostać rozmieszczona wzdłuż brzegu

$$A = \int_0^L \frac{1}{h(u)} ds = \int_{u_0}^{u_m} \frac{1}{h(u)} \frac{ds}{du} du$$

← A zazwyczaj nie jest całkowita

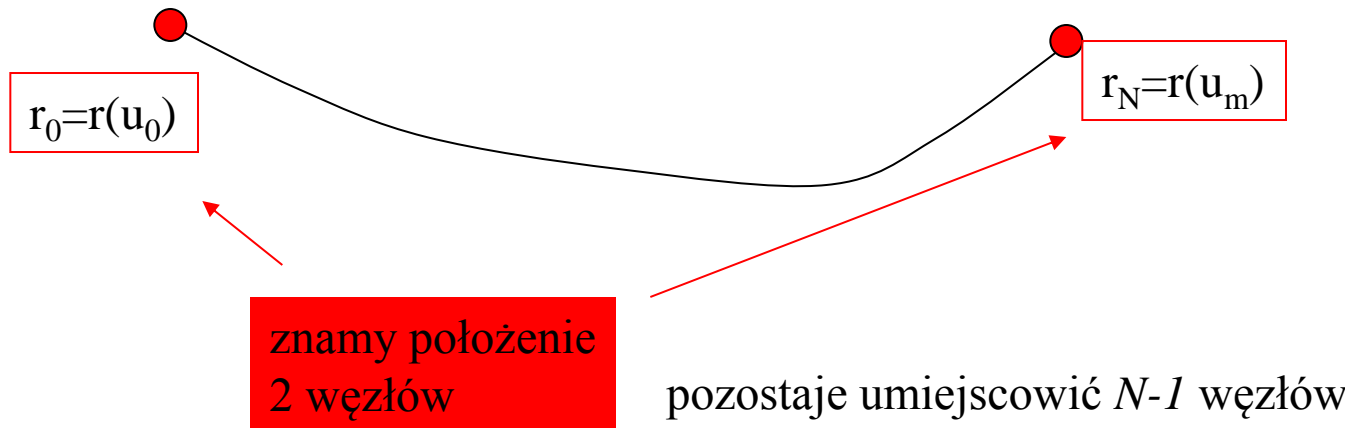
bierzemy $N : l$. całkowita najbliższa A

Indeks konstystencji (zgodności): jak blisko jest N do A:

$$\theta = \frac{A}{N} = \frac{1}{N} \int_{u_0}^{u_m} \frac{1}{h(u)} \frac{ds}{du} du$$

Generacja węzłów brzegowych

wygenerować N elementów ($N+1$ węzłów)



6) Niech każdy węzeł na brzegu będzie generowany z tym samym indeksem konstystencji:
 Węzeł k ($=1,2,\dots,N-1$) – poszukujemy dla niego wartości parametru u_k , takiego, że

$$\theta = \theta_k = \frac{1}{k} \int_{u_0}^{u_k} \frac{1}{h(u)} \frac{ds}{du} du \quad \longrightarrow \quad \text{mając } u_k \text{ wyliczymy } \mathbf{r}(k)$$

następny: u_k


$$\theta = \theta_{k+1} = \frac{1}{k+1} \int_{u_0}^{u_{k+1}} \frac{1}{h(u)} \frac{ds}{du} du$$

itd.

$$\theta_k = \theta_{k+1}$$

$$\frac{1}{k} \int_{u_0}^{u_k} f(u) du = \frac{1}{k+1} \int_{u_0}^{u_{k+1}} f(u) du$$

~~$$(k+1) \int_{u_0}^{u_k} f(u) du = k \left(\int_{u_0}^{u_k} f(u) du + \int_{u_k}^{u_{k+1}} f(u) du \right)$$~~



$$\theta = \int_{u_k}^{u_{k+1}} \frac{1}{h(u)} \frac{ds}{du} du$$

Wyznaczy położenie odpowiadające węzłom $k=1,2,\dots,N-1$

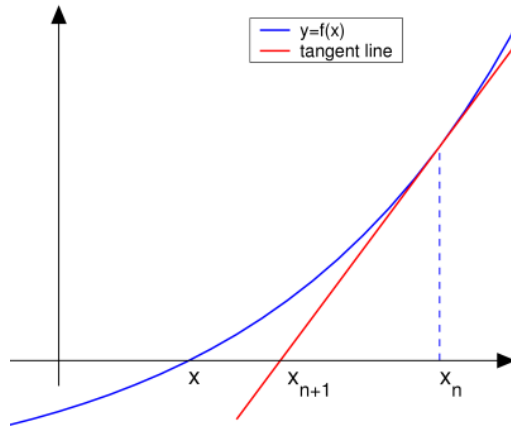
znamy u_k , szukamy następnego u_{k+1}

$$F(u_{k+1}) = \int_{u_k}^{u_{k+1}} \frac{1}{h(u)} \frac{ds}{du} du - \theta = 0$$

- szukamy zera równania nieliniowego

$$F(u_{k+1}) = \int_{u_k}^{u_{k+1}} \frac{1}{h(u)} \frac{ds}{du} du - \theta = 0$$

Metoda Newtona rozwiązywania równań nieliniowych (przypomnienie)



poprawione rozwiązanie:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

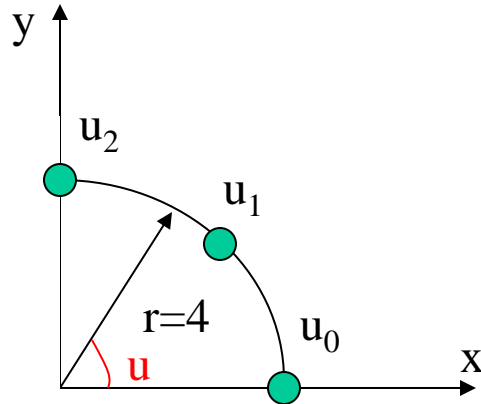
$$u_{k+1}^{j+1} = u_{k+1}^j - F(u_{k+1}^j) \frac{h(u_{k+1}^j)}{\frac{ds}{du} \Big|_{u_{k+1}^j}}$$

do startu iteracji można nawet

$$u_{k+1}^0 = u_k$$

PRZYKŁAD

Element krawędzi: ćwierć okręgu



parametryzacja *dokładna* (przybliżona zbędna)

$$x=4\cos(u), y=4\sin(u), \quad 0 \leq u \leq \pi/2$$

- Wybieramy 3 punkty próbkujące: $u_0=0$, $u_2=\pi/2$ trzeci w środku $u_1=\pi/4$

2. Wyznaczmy wersory styczne do brzegu

$$\mathbf{t}_l = \begin{pmatrix} t_1(l) \\ t_2(l) \end{pmatrix} \quad t_1(l) = \frac{x'}{\sqrt{x'^2 + y'^2}} \quad x' = \frac{dx}{du}(u_l)$$

$$t_2(l) = \frac{y'}{\sqrt{x'^2 + y'^2}} \quad y' = \frac{dy}{du}(u_l)$$

$$x' = -4 \sin(u)$$

$$y' = 4 \cos(u)$$

$$\longrightarrow \mathbf{t} = \begin{pmatrix} -\sin(u) \\ \cos(u) \end{pmatrix} \longrightarrow$$

$$\mathbf{t}(u_0) = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

$$\mathbf{t}(u_1) = \begin{pmatrix} -\frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} \end{pmatrix}$$

$$\mathbf{t}(u_2) = \begin{pmatrix} -1 \\ 0 \end{pmatrix}$$

2cd Kierunki rozciągnięcia – niech nie zależą od położenia i pokrywają się z (x,y).

Generacja węzłów brzegowych

$$\alpha_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad \alpha_2 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

niech idealne rozmiary elementów $h_1=h_2=2$

Macierz transformacji:

$$T = \sum_{i=1}^2 \frac{1}{h_i} \begin{pmatrix} \alpha_{1i} \\ \alpha_{2i} \end{pmatrix} \begin{pmatrix} \alpha_{1i} & \alpha_{2i} \end{pmatrix}$$

nie zależy od położenia i...

$$T = \frac{1}{2} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

3. lokalne długości łuku $hs(l)$

$$hs(l) = \frac{1}{\sqrt{(T_l t_l)^T (T_l t_l)}}$$

Generacja węzłów brzegowych

$$T = \frac{1}{2} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

$$\mathbf{t}(u_0) = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

$$\mathbf{t}(u_1) = \begin{pmatrix} -\frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} \end{pmatrix}$$

$$\mathbf{t}(u_2) = \begin{pmatrix} -1 \\ 0 \end{pmatrix}$$

$$hs(0) = hs(u_0) = 2$$

podobnie

$$hs(1) = hs(2) = 2$$

4. funkcja długości elementu od parametru (u)

$$h(u) = \sum_{l=0}^m hs(l) v_l(u)$$

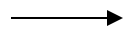
interpolacja liniowa funkcji stałej, czyli $h(u) = 2$

gęstość elementów w przestrzeni parametrycznej $1/h(u) = 1/2$

5. Idealna liczba elementów

$$A = \int_0^L \frac{1}{h(u)} ds = \int_{u_0}^{u_m} \frac{1}{h(u)} \frac{ds}{du} du \quad ds/du=4$$

$$A = \int_0^{\pi/2} \frac{1}{2} 4 du = \pi$$



najbliższy integer $N=3$: będą 3 elementy, 4 węzły, mamy 2 : pierwszy 0 i ostatni $\pi/2$

indeks konsystencji = $\pi/3$

6. Położenie parametryczne węzła nr 1:

$$\theta = \frac{\pi}{3} = \int_0^{u_1} 2 du = 2u_1 \rightarrow u_1 = \pi/6$$

$$\theta = \int_{u_k}^{u_{k+1}} \frac{1}{h(u)} \frac{ds}{du} du$$

węzła nr 2:

$$\theta = \frac{\pi}{3} = \int_{\pi/6}^{u_2} 2 du = 2(u_2 - \pi/6) \rightarrow u_2 = \pi/3$$

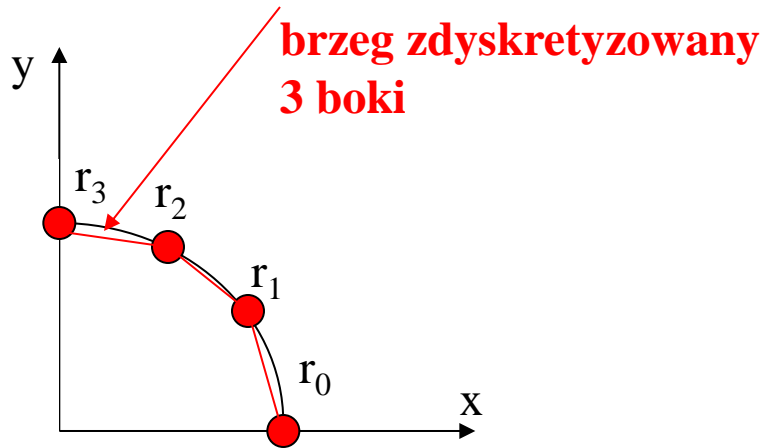
7. Mapowanie węzłów do przestrzeni fizycznej

$$u_0=0$$

$$u_1=\pi/6$$

$$u_2=\pi/3$$

$$u_3=\pi/2$$



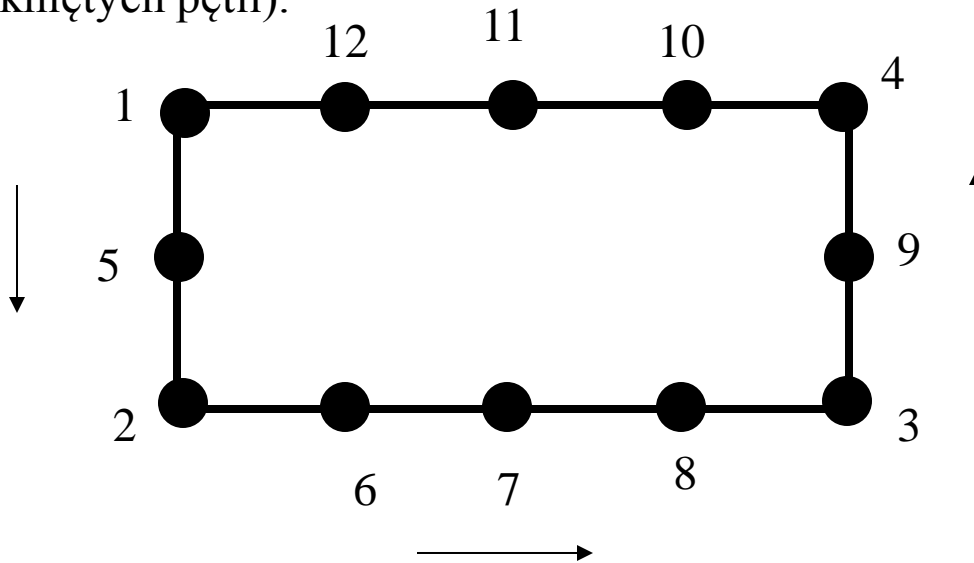
koniec przykładu.

Advancing front method:

Generacja elementów trójkątnych wewnątrz obszaru całkowania

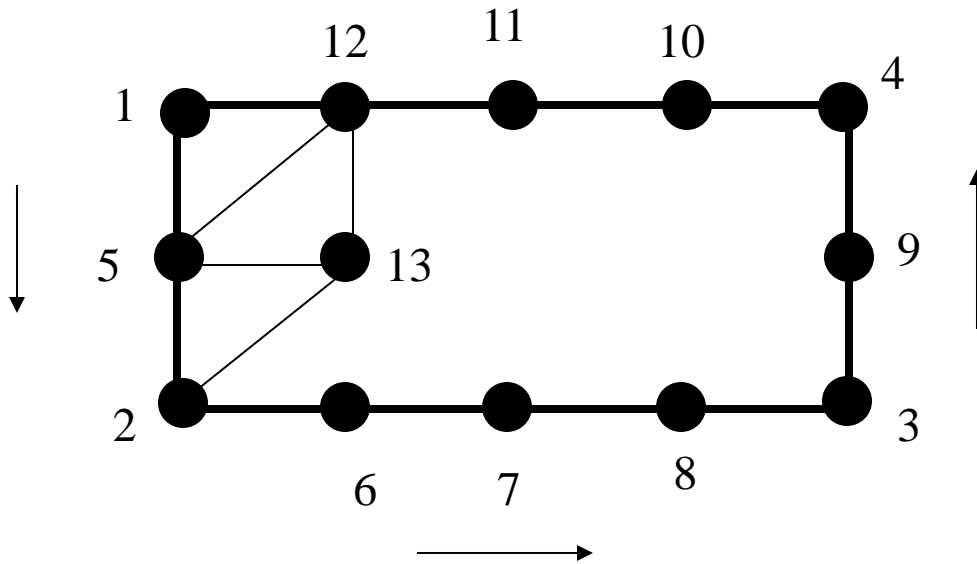
front (*generation front*) – ustalany przed rozpoczęciem triangulacji.

początkowy front g – zbiór wszystkich boków zdyskretyzowanej krawędzi brzegowej (zbiór zamkniętych pętli).



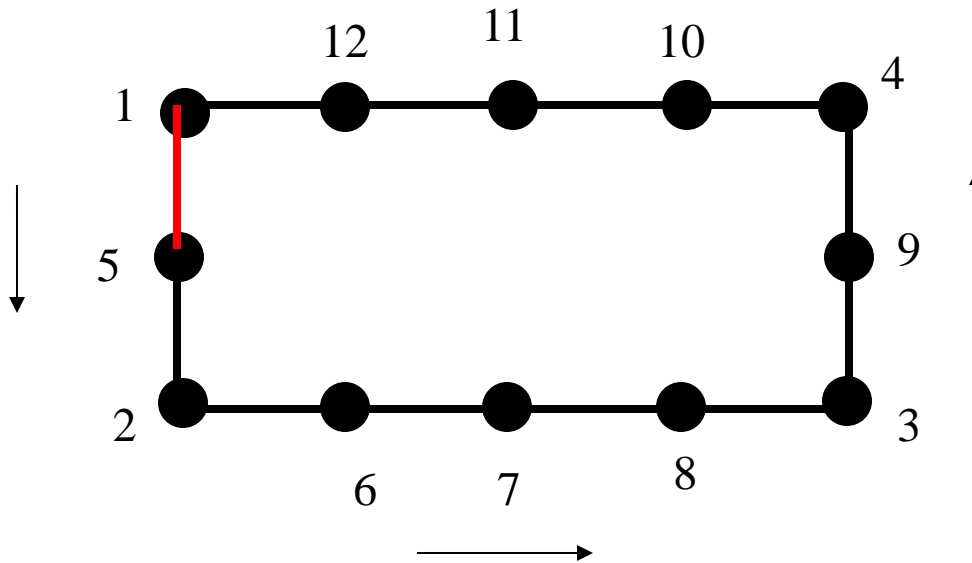
elementy i węzły tak uporządkowane, że
przestrzeń, w której mają być
wygenerowane elementy – na lewo od frontu

Generacja elementów trójkątnych wewnątrz obszaru całkowania



elementy i węzły tak uporządkowane, że
przestrzeń, w której mają być
wygenerowane elementy – na lewo od frontu
- porządek utrzymany w czasie tworzenia nowych elementów

Generacja elementów trójkątnych wewnątrz obszaru całkowania



w procesie generacji – wybierany **aktywny bok** z frontu

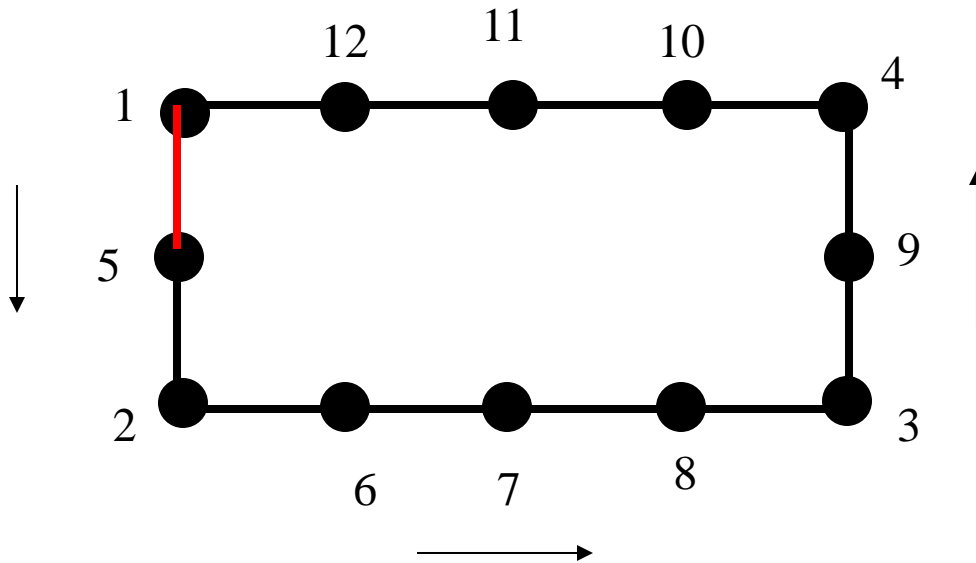
– granica tworzonego elementu

tworzony element : poszukiwany trzeci węzeł: nowy, wygenerowany
lub stary (frontowy)

Generacja elementów trójkątnych wewnątrz obszaru całkowania

front początkowy

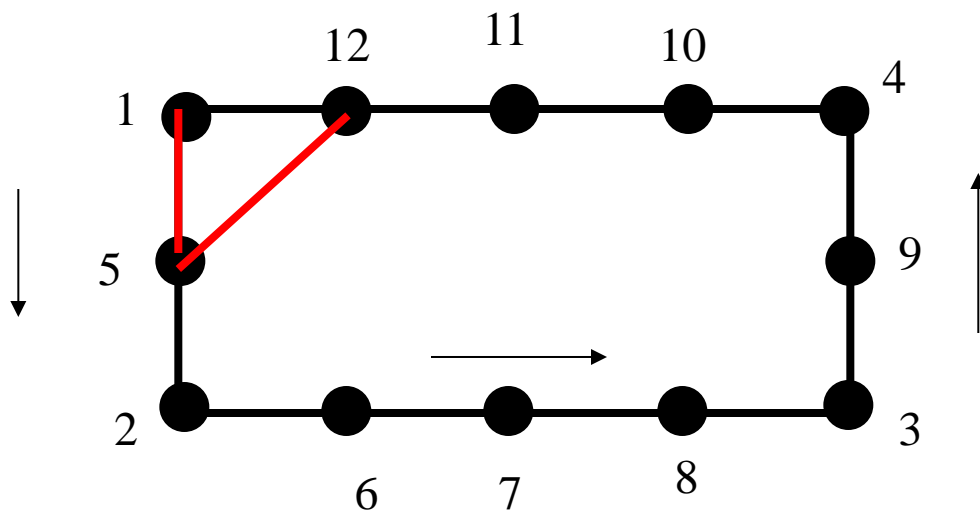
1	5	2	6	7	8	3	9	4	10	11	12
5	2	6	7	8	3	9	4	10	11	12	1



po utworzeniu nowego elementu – uaktualnienie frontu

front początkowy

1	5	2	6	7	8	3	9	4	10	11	12
5	2	6	7	8	3	9	4	10	11	12	1



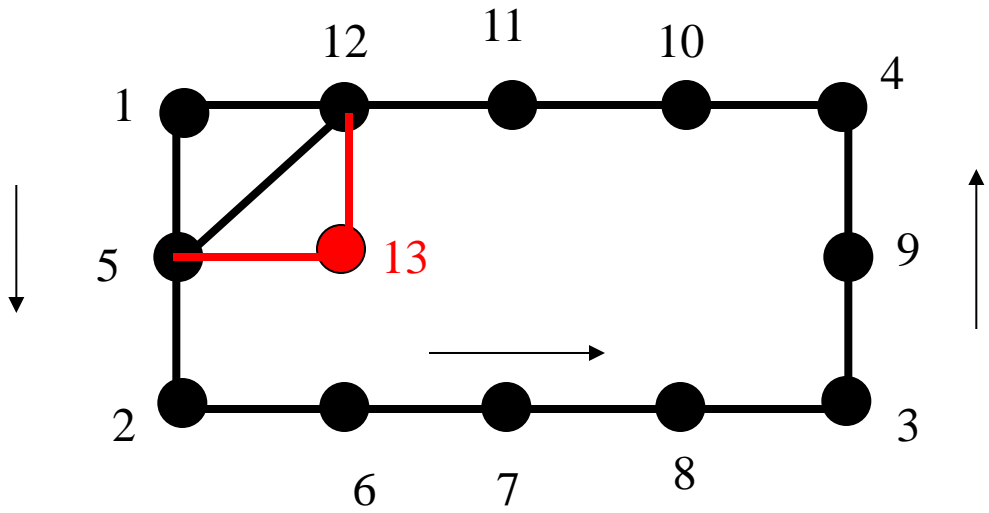
front uaktualniony

1	5	2	6	7	8	3	9	4	10	11	12	12
5	2	6	7	8	3	9	4	10	11	12	1	5

po utworzeniu nowego elementu – uaktualnienie frontu

front

5	2	6	7	8	3	9	4	10	11	12
2	6	7	8	3	9	4	10	11	12	5

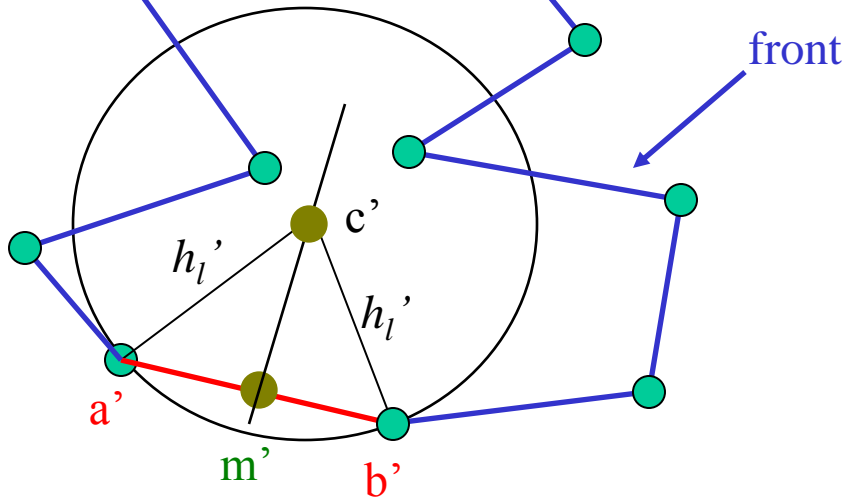


front

5	2	6	7	8	3	9	4	10	11	12	12	13
2	6	7	8	3	9	4	10	11	12	5	13	5

Generacja elementu (zgodnie z siatką tła)

za Zienkiewiczem , FEM its basis & fundamentals



- 1) wybieramy ścianę aktywną
- najkrótszą z frontowej: (a,b)
- 2) środek ściany aktywnej $m=(a+b)/2$
w m liczymy (interpolacja z siatki tła)
parametry siatki:
lokalne kierunki rozciągania $\alpha(m)$
oraz lokalne rozmiary siatki $h(m)$
- 3) dalsze czynności wykonujemy w
przestrzeni znormalizowanej wg parametrów z 2

$$\mathbf{x}' = \mathbf{T}\mathbf{x}$$



$$T = \sum_{i=1}^2 \frac{1}{h_i} \begin{pmatrix} \alpha_{1i} \\ \alpha_{2i} \end{pmatrix} \begin{pmatrix} \alpha_{1i} & \alpha_{2i} \end{pmatrix}$$

w przestrzeni znormalizowanej (primowanej) utworzyć element o tak jednostkowych rozmiarach jak to możliwe

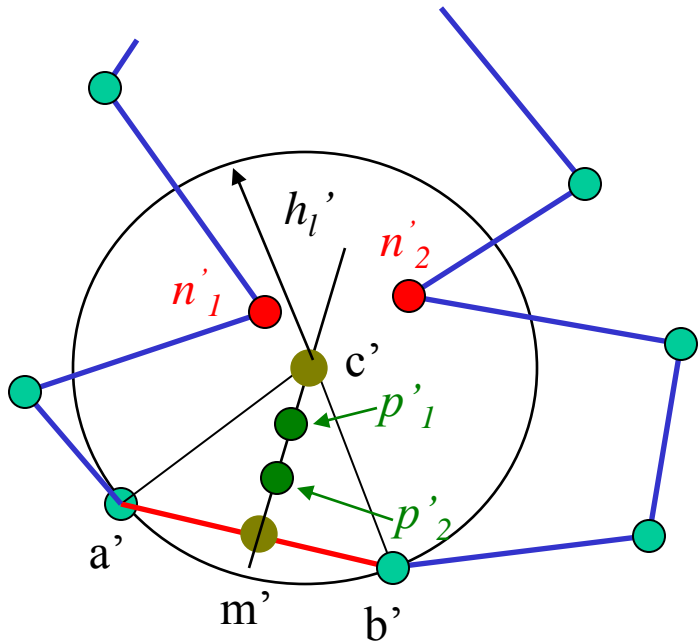
- 4) wyznaczamy **idealne** położenie nowego węzła: c'

c' : w kierunku normalnym do (a',b') [wgląd obszaru całkowania]
w odległości h'_i od a' i b' :

$$h'_i = \begin{cases} 0.55l(a', b') & \text{dla } l(a', b') \geq \frac{1}{0.55} = 1.81 \\ 1 & \text{dla } 0.5 < l(a', b') < 1.81 \\ 2l(a', b') & \text{dla } l(a', b') \leq 0.5 \end{cases}$$

empiryczne, sprawdzone:
nie chcemy elementów
zbyt odkształconych
(poświęcamy idealny rozmiar na
rzecz równomiernego kształtu)

Generacja elementu



5. Dodatkowe punkty – dodawane do listy kandydatów na 3-ci węzeł nowego elementu:
 - wszystkie węzły w promieniu Nh'_l od idealnego położenia c' : notowane n'_1, n'_2 itd. - sortowane względem odległości od c'
[-generacja elementu bez tworzenia nowego węzła] na rysunku $N=1$
 - zbiór L punktów generowanych na linii $(m'c')$ notowane p'_1, p'_2, \dots, p'_L – sortowane wzgl. odległości od c' .
[-gwarancja, że nowy element zawsze może być utworzony] na rysunku $L=2$

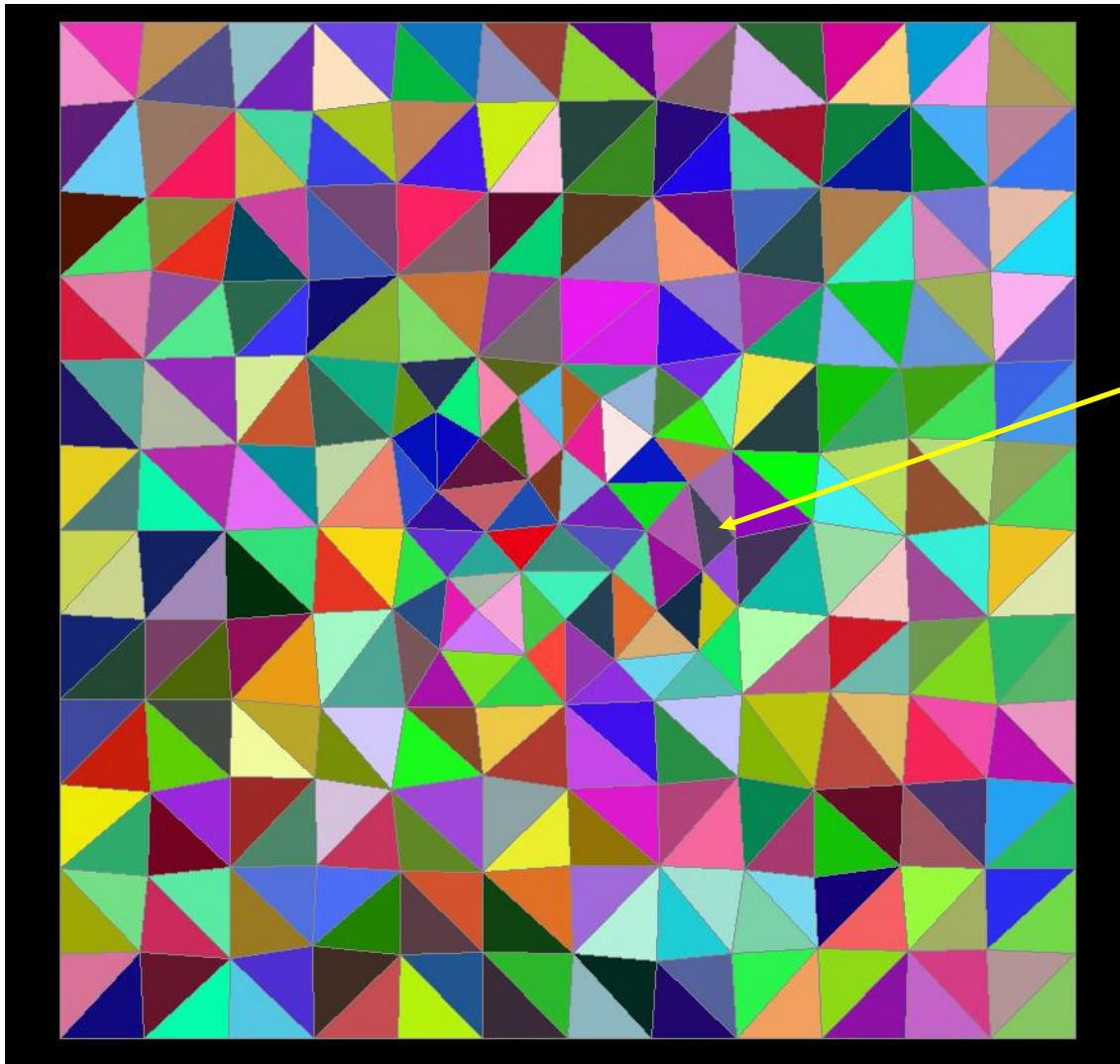
6. przeglądamy sekwencyjnie najpierw punkty z listy n' , następnie punkt c' do utworzenia nowego elementu brany jest q' - pierwszy z punktów, taki że $0.5l(a'b') < l(q',a') < 1.81 l(a'b')$ oraz $0.5l(a'b') < l(q',b') < 1.81 l(a'b')$ i ani (q',a') ani (q',b') nie przecinają żadnego z boków frontu

jeśli żaden z powyższych się nie nadaje – przeglądamy p' i bierzemy pierwszy, który spełnia powyższe kryteria.

7. Nowy element tworzony – współrzędne transformowane c' do przestrzeni fizycznej $c = T^{-1}c'$

8. Front – lista uaktualniana \longrightarrow proces kończy się gdy liczba boków frontu dojdzie do 0

Wygładzanie siatki:

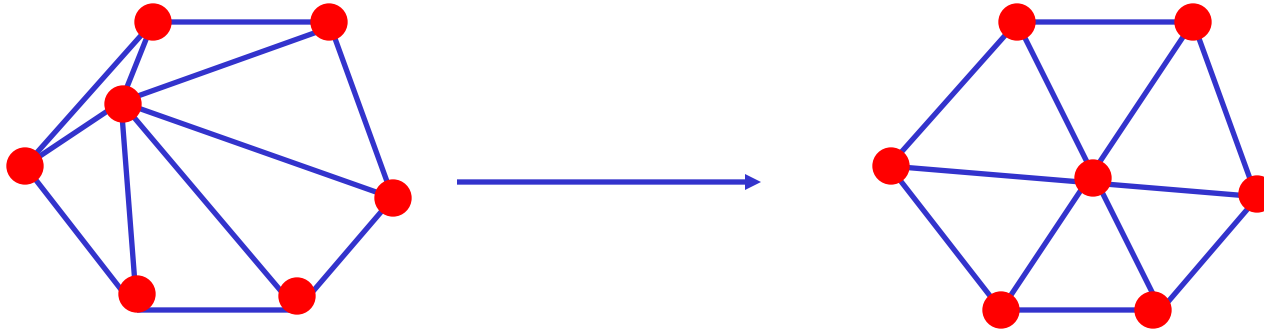


zdarzą się wyraźnie
nierównoboczne elementy

Poprawa jakości siatki trójkątnej

W praktyce, najlepiej sprawdzają się siatki, w których elementy są w miarę możliwości podobnych rozmiarów/kształtów

wygładzanie laplasjańskie [Laplacian smoothing]



zachowana topologia siatki
[przyporządkowanie węzły/elementy,
sąsiedztwo itd.]

$$x_i = \frac{1}{N} \sum_{j=1}^N x_j$$

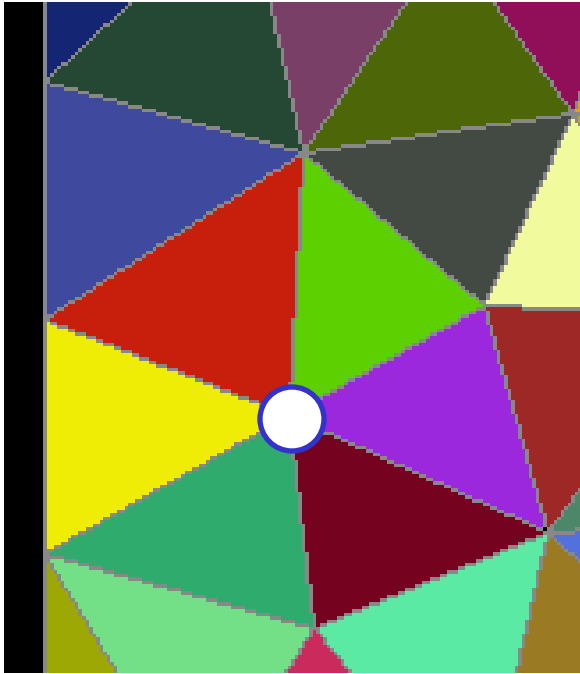
$$y_i = \frac{1}{N} \sum_{j=1}^N y_j$$

5-6 iteracji
po wszystkich
węzłach wewnętrznych

Nazwa inspirowana relaksacją dla równania Laplace'a

$$f(x_i, y_i) = \frac{f(x_i - dx, y_i) + f(x_i + dx, y_i) + f(x_i, y_i - dx) + f(x_i, y_i + dx)}{4}$$

Laplacian smoothing



$$x_i = \frac{1}{N} \sum_{j=1}^N x_j$$

jak zaprogramować:
gdy tworzone węzły
tworzona

lista węzeł – elementy do których należy

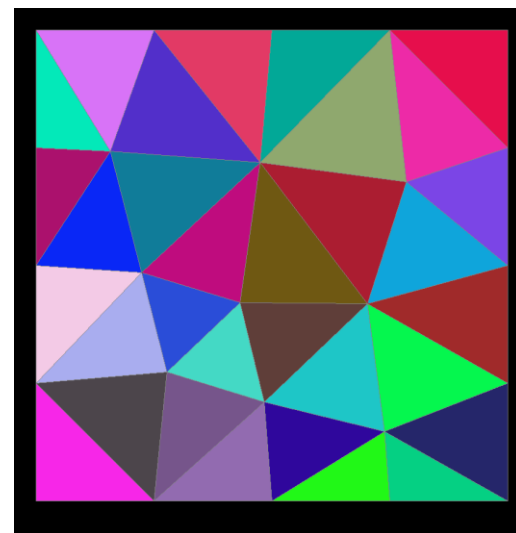
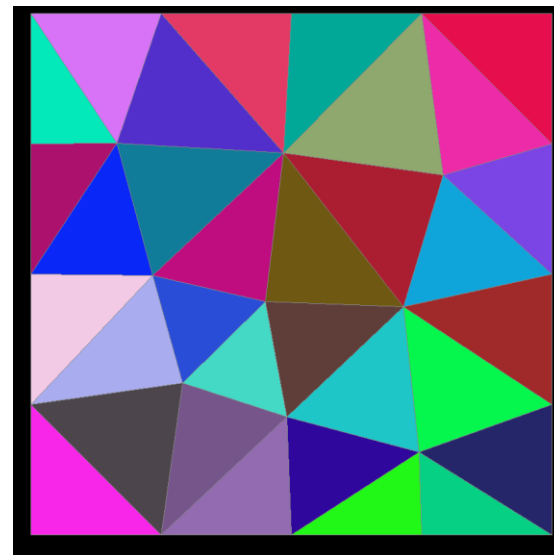
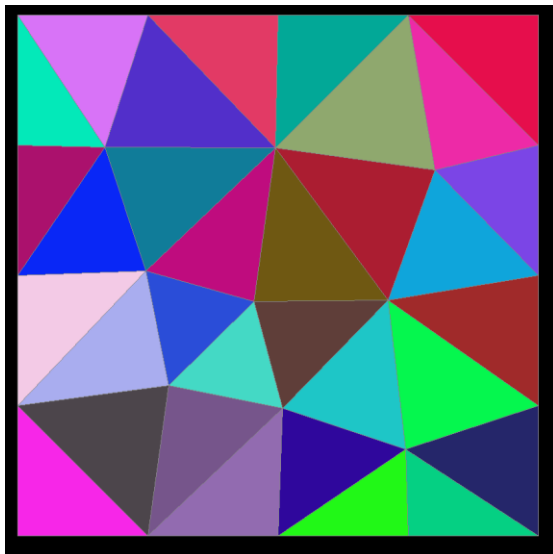
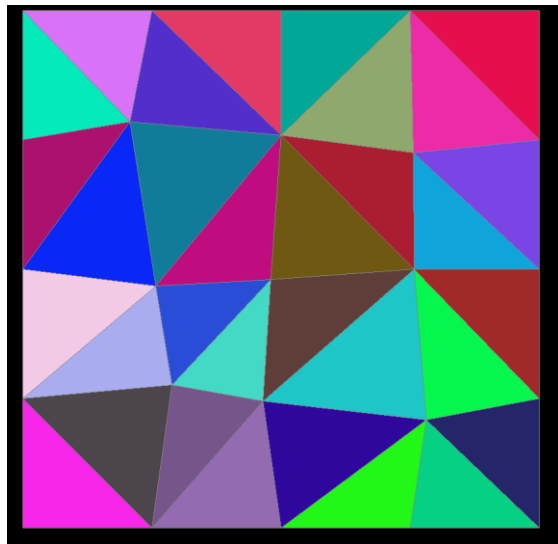
suma s wszystkich współrzędnych x-owych
wszystkich elementów do których węzeł należy

każdy sąsiedni punkt policzony dwukrotnie,
punkt i - tylu krotnie do ilu elementów należy
(na rysunku $M = 6$)

$$s = s - Mx_i$$

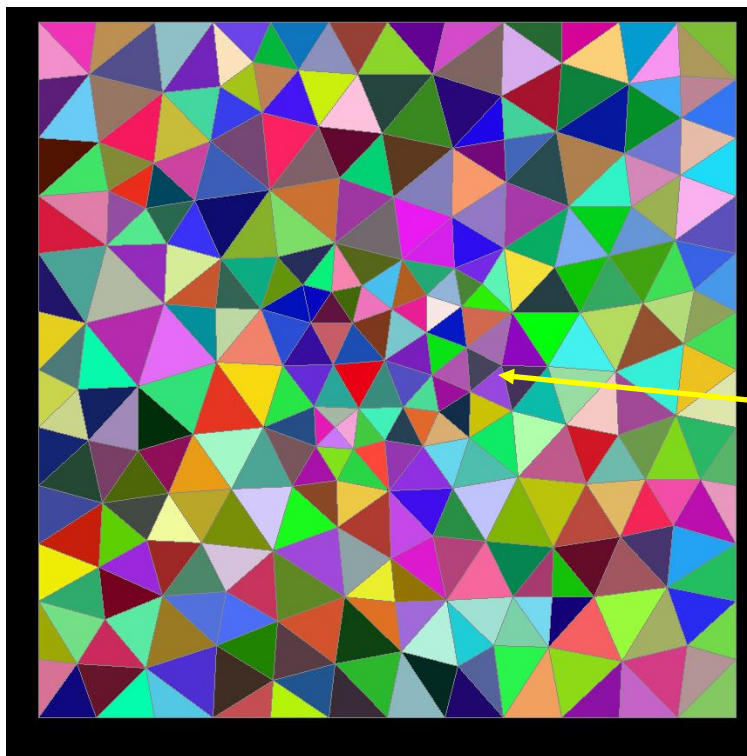
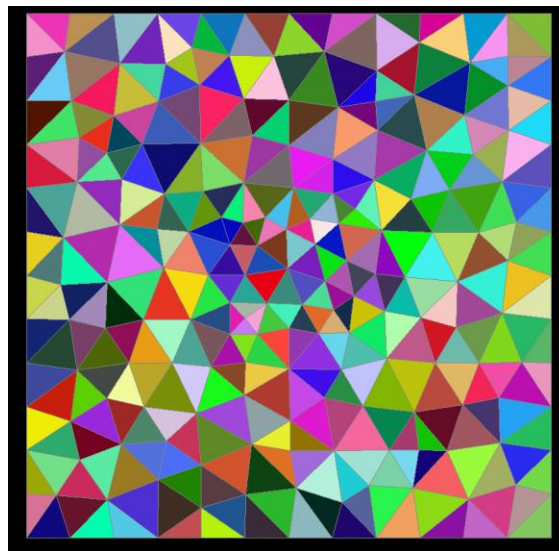
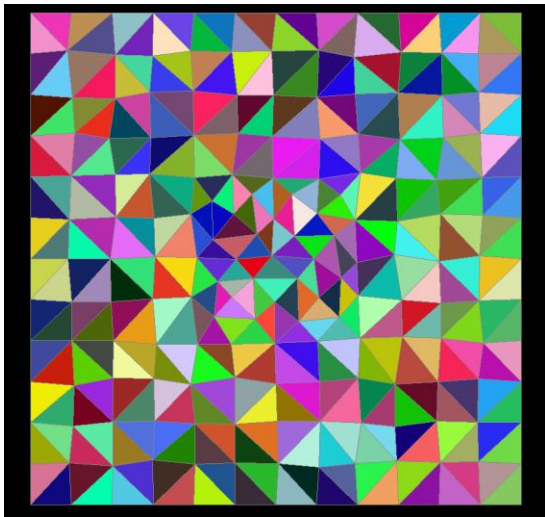
$$s = s / 2M$$

$$x_i = s$$



węzły na brzegu wygładzaniu
nie podlegają

Laplacian smoothing

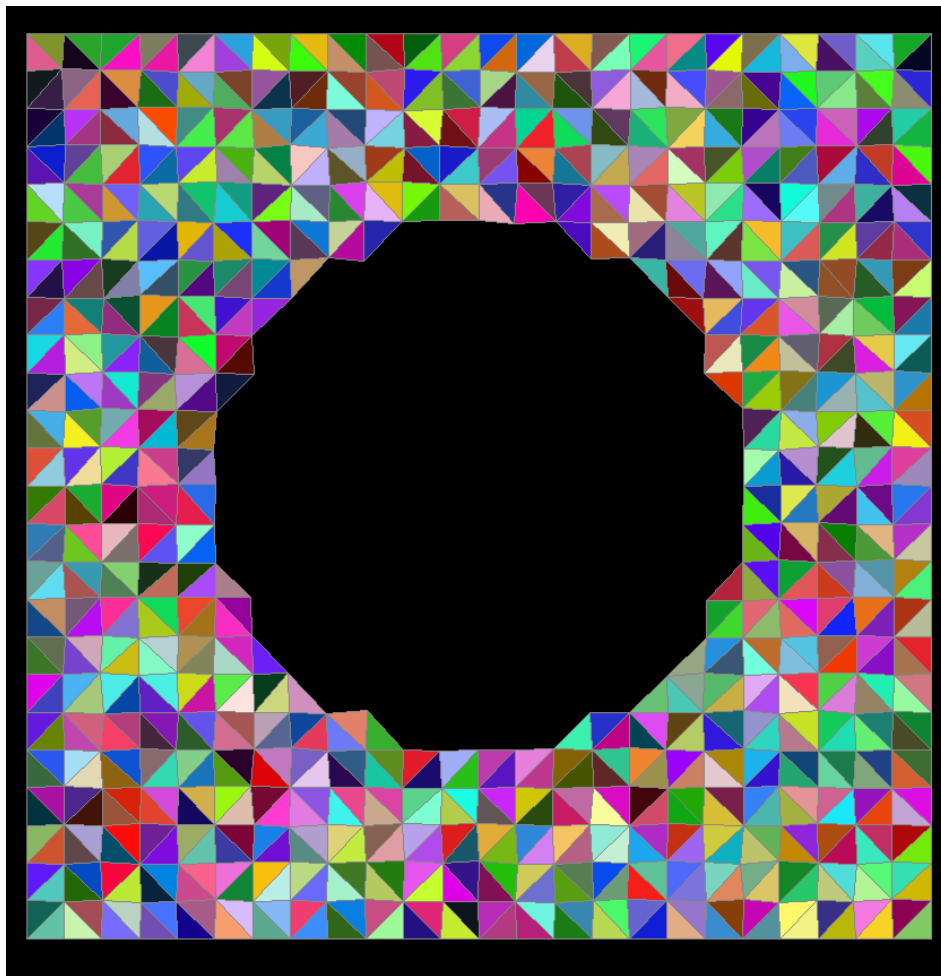


10 iteracji

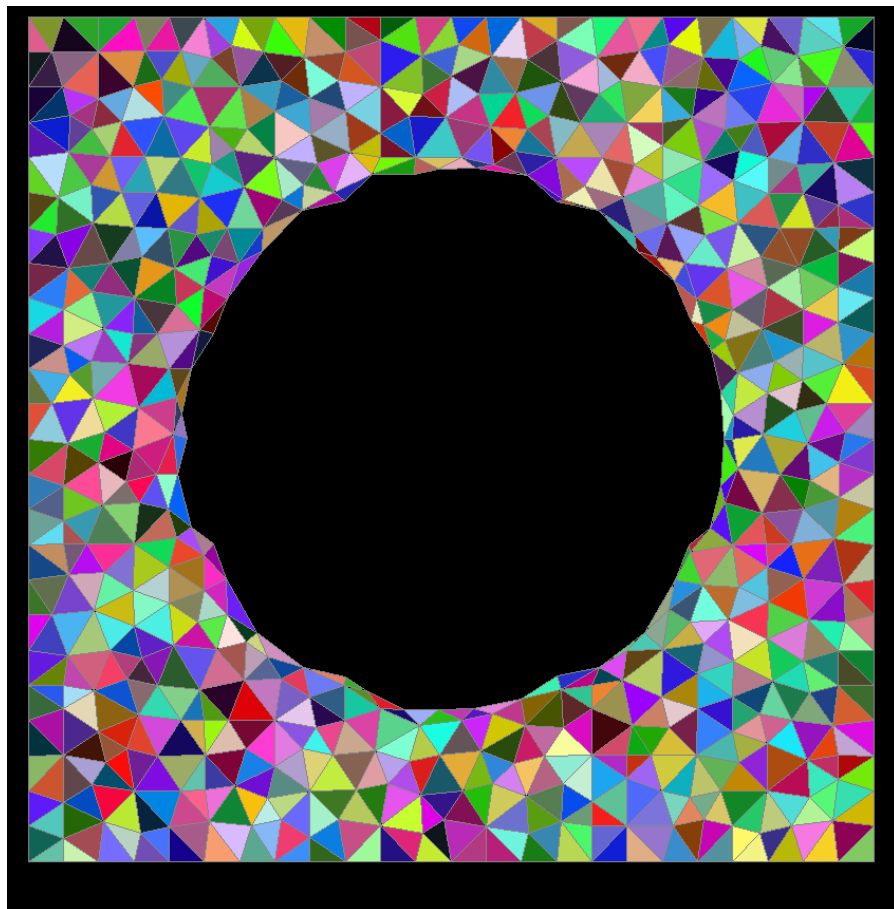


już prawie równoboczny

Laplacian smoothing

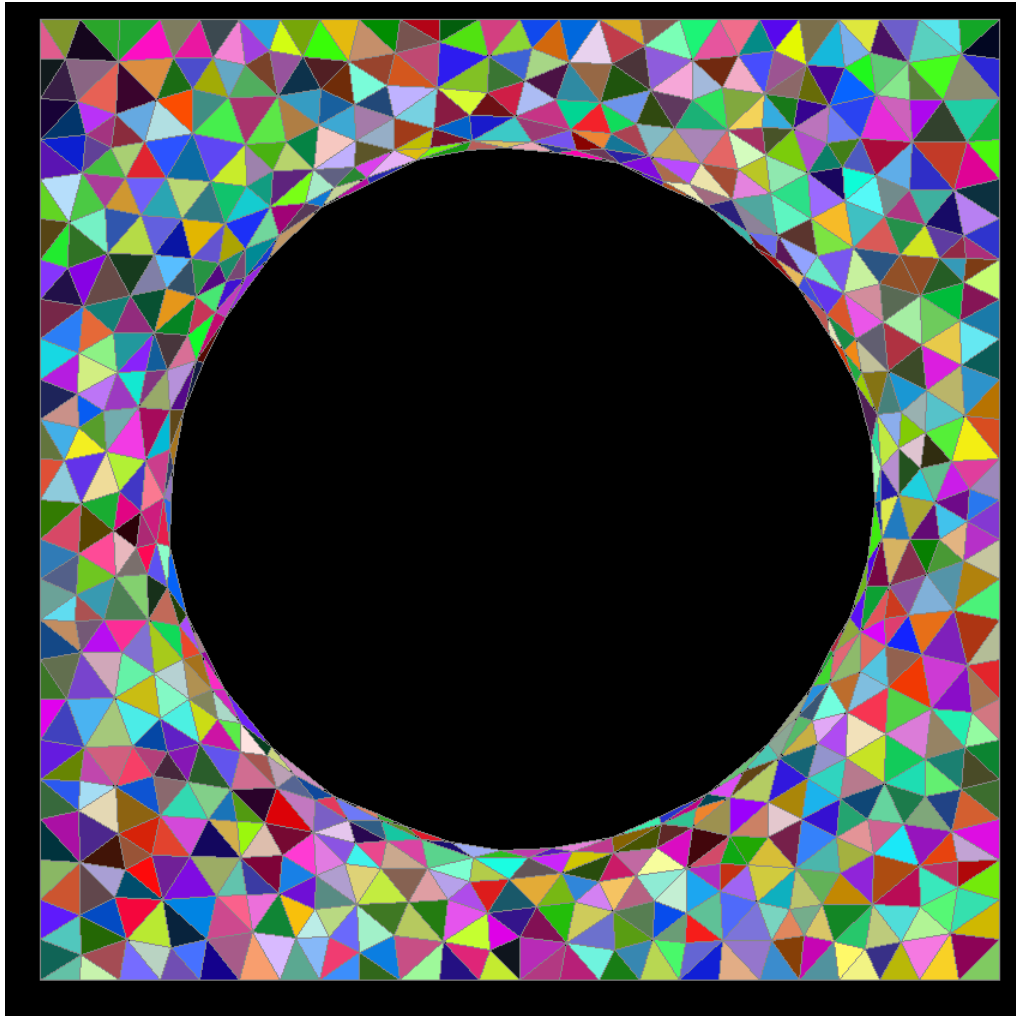


siatka z wyłączonym
kołem o promieniu 0.5



siatka początkowo
z wyłączonym
kołem o promieniu 0.5

1 wygładzanie laplasjańskie

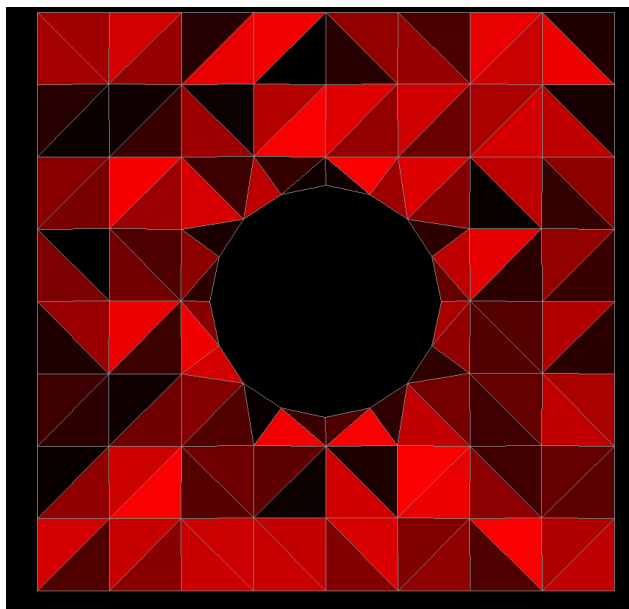


siatka początkowo
z wyłączonym
kołem o promieniu 0.5

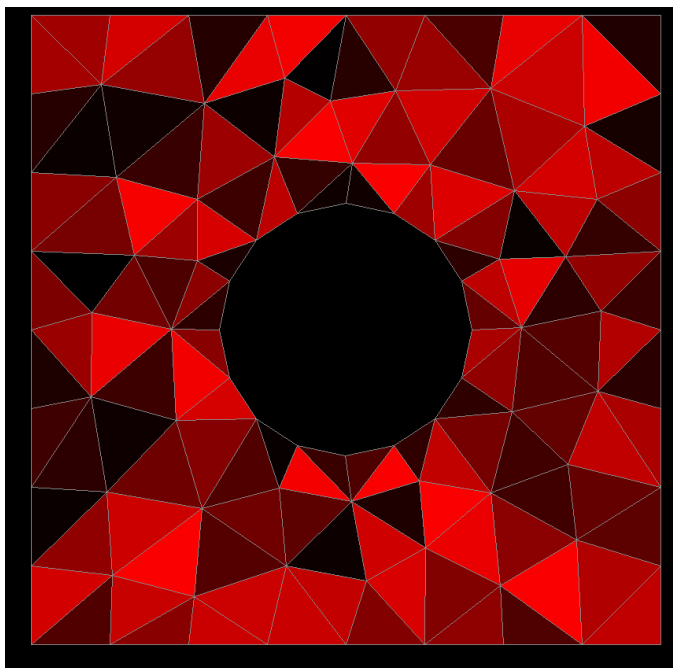
10 wygładzeń laplasjańskich

przezwór się roztwiera

otwór w siatce: rośnie i
wygładza się

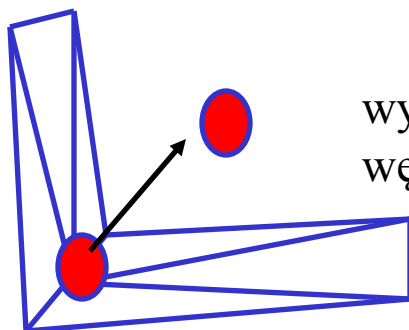


zadane węzły na okręgu
przed wygładzeniem



po wygładzeniu,
z którego węzły środkowe wyłączone

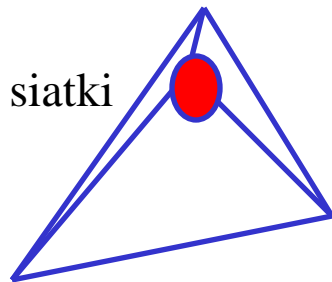
Możliwy problem z wygładzaniem I.



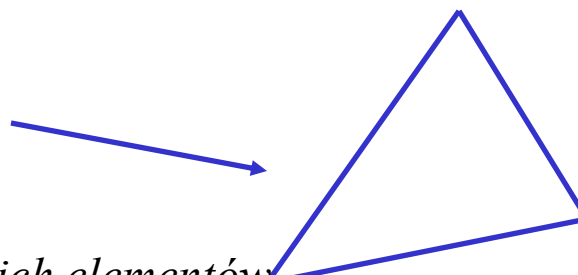
wygładzanie laplasjańskie może zakończyć się wyrzuceniem węzła wewnętrznego poza obszar ograniczony „zewnątrznymi”
- konieczna dodatkowa kontrola

Inne zabiegi:

Modyfikacja topologii siatki

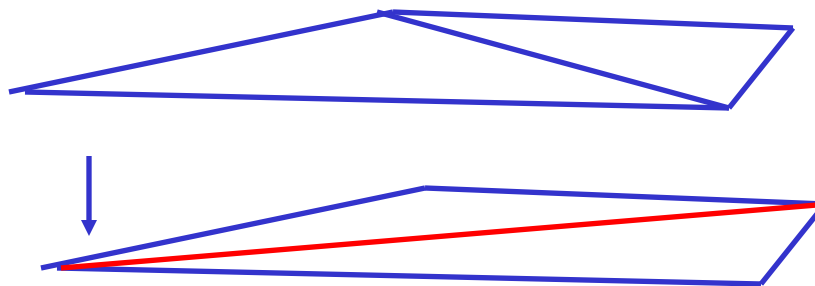


1) usunięcie węzła (i elementów przyległych)



nie chcemy zbyt drobnych elementów, wąskich elementów

2) wymiana przekątnych



Elementy o podobnej powierzchni

Triangulacja Delaunaya (najłatwiej się uogólnia do 3D):

Advancing front method a triangulacja Delaunay'a w generacji MES

- 1) W *advancing front*: zakładamy siatkę tła: rozmiary i kierunki rozciągnięcia elementów i dla nich szukamy położen węzłów
- 2) W triangulacji D.: zakładamy gdzie są węzły (wierzchołki trójkątów) przestrzeń dzielimy na nieprzekrywające się trójkąty o możliwie regularnych kształtach

triangulacja Delauneya: każdy z utworzonych trójkątnych elementów będzie opisany na kole, które w swym wnętrzu nie zawierać będzie żadnego węzła

„mądrzej”: konstruujemy diagram Woronoja a następnie szukamy dla niego diagramu dualnego, który podzieli obszar całkowania na trójkąty.

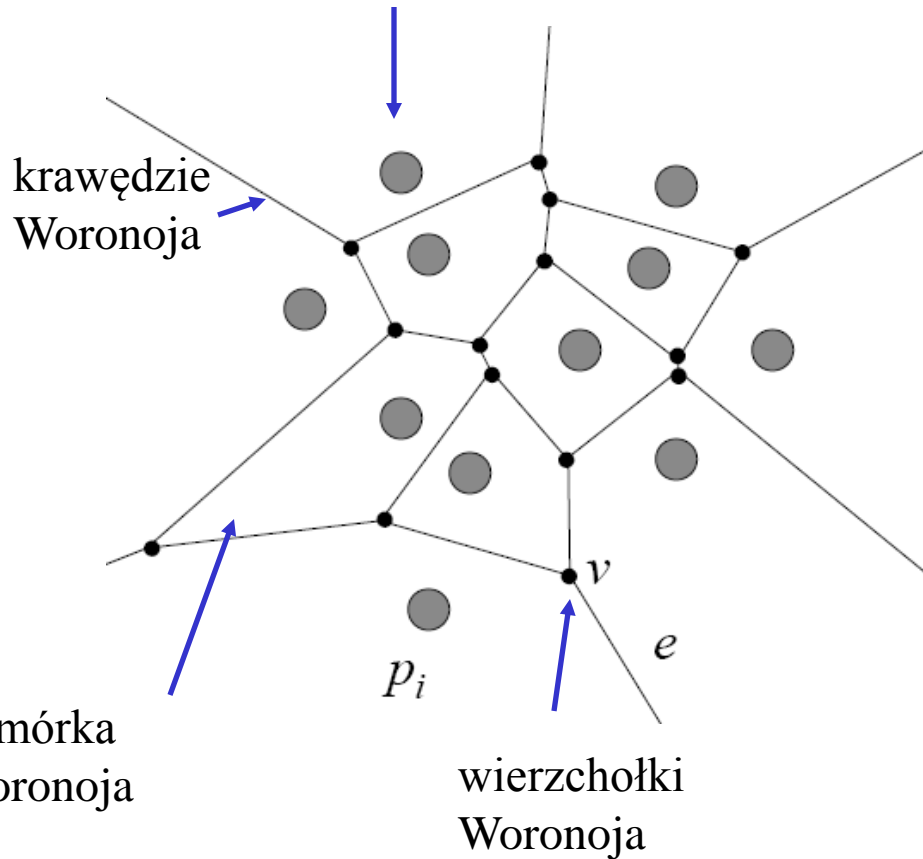
diagram (graf) Woronoja

Ilustracje:

- 1) *własne*
- 2) *Zienkiewicz*
- 3) *Allen Miu*

podział miasta między np. urzędy pocztowe

punkty tworzące (wierzchołki w MES)



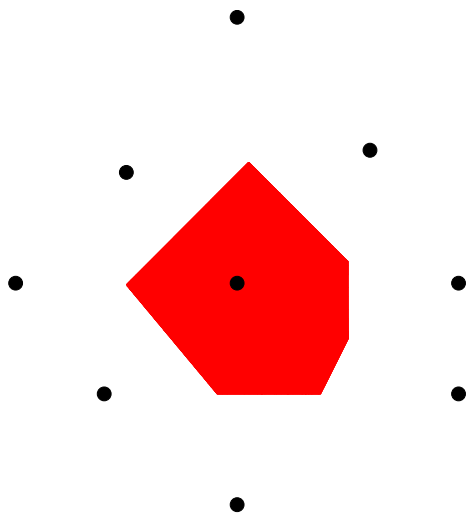
w *advancing front*:
węzły lokowane tak
aby elementy miały
żądany kształt

w TD: wybierane
położenia węzłów
na podstawie naszej wiedzy
elementy generowane
tak aby miały maksymalnie
regularne kształty

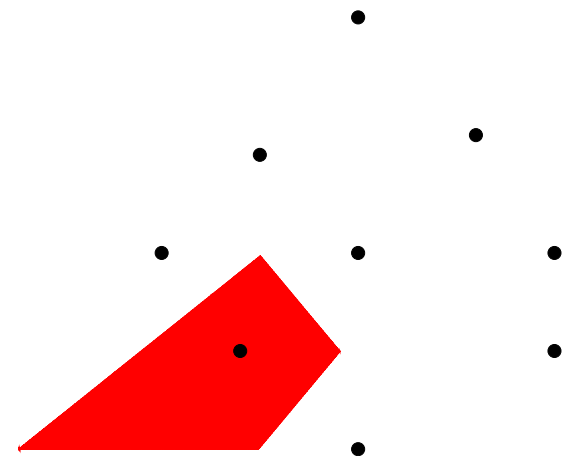
Zbiór N punktów p_1, p_2, \dots, p_N na płaszczyźnie euklidesowej
(tzw. punkty tworzące)

dla każdego z punktów p_i definiujemy **komórkę Woronoja**: $W(p_i)$
jako zbiór punktów, które są bliżej (nie dalej) punktu p_i niż
pozostałych punktów

$$V(p_i) = \{x \in R^2 : |x - p_i| \leq |x - p_j|, \forall j \neq i\}$$



przykłady:



<http://reference.iucr.org/dictionary/Image:W-S-1.gif>

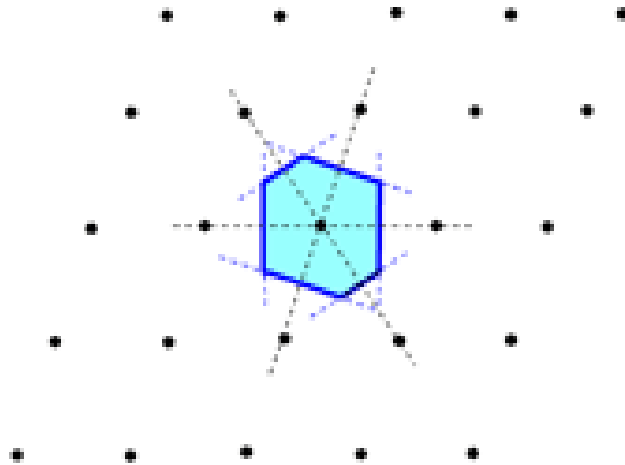


Figure 1 Wigner-Seitz cell.
Shaded: domain of influence.

← kryształ:
układ atomów o symetrii
translacyjnej

komórka elementarna
powtórzona w nieskończoność

każdy obszar zawierający
jeden węzeł sieci nadaje
się na komórkę elementarną

komórka WS jest specjalna,
bo jest obszarem Woronoja
związanym z danym węzłem

Przykłady:

1 punkt tworzący



pusty diagram Woronoja

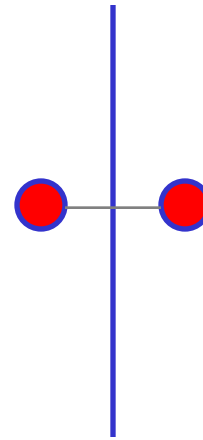
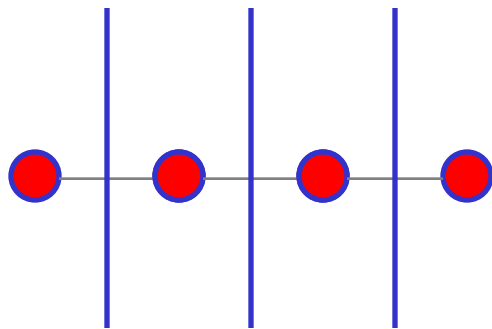


diagram W.: linia prosta, brak wierzchołków



wspólniowe punkty tworzące, brak w.

3 pkty niewspólniowe: środek okręgu
węzeł Woronoja

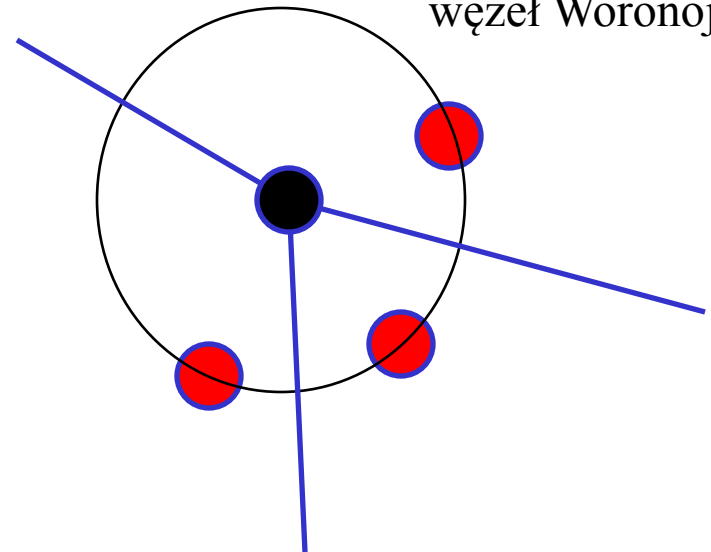
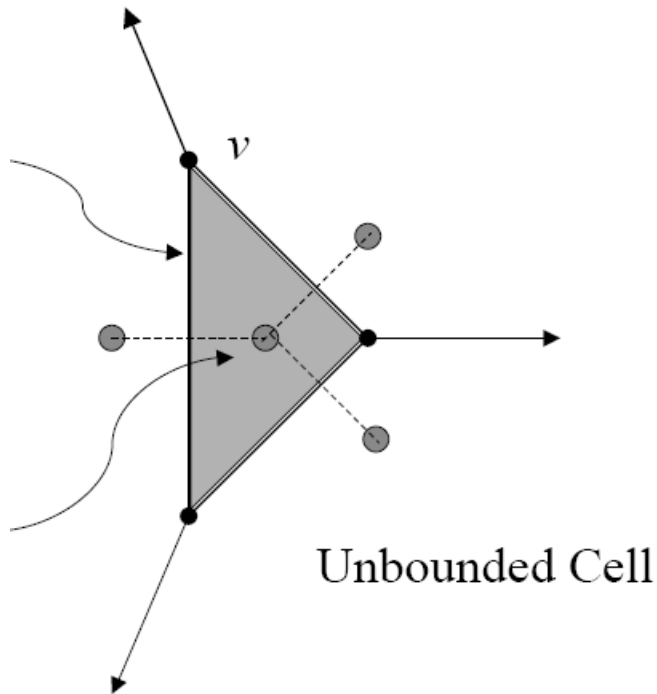


Diagram Woronoja, przykłady cd.

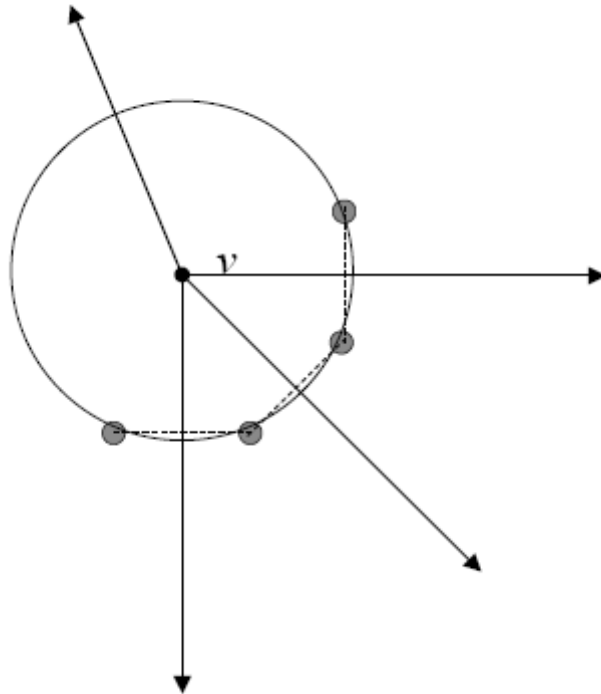
aby komórka W. była zamknięta = potrzebne 4 niewspółliniowe punkty



krawędzie Woronoja:
części wspólne komórek Woronoja

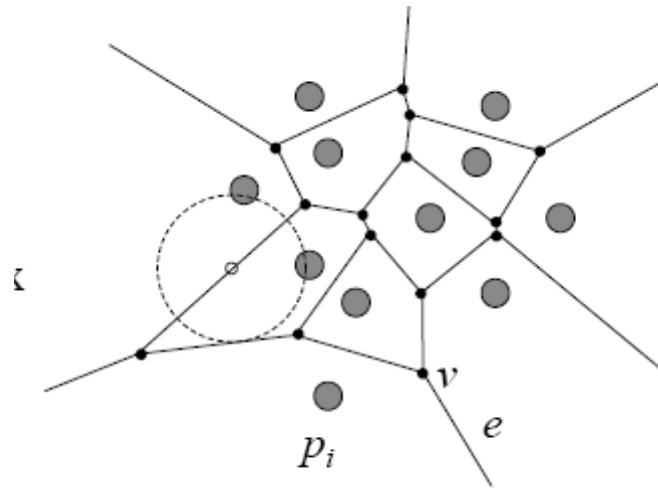
część wspólna krawędzi Woronoja:
węzły Woronoja

potrzebne, ale sama współliniowość 4 punktów nie wystarcza:



tzw. przypadek zdegenerowany

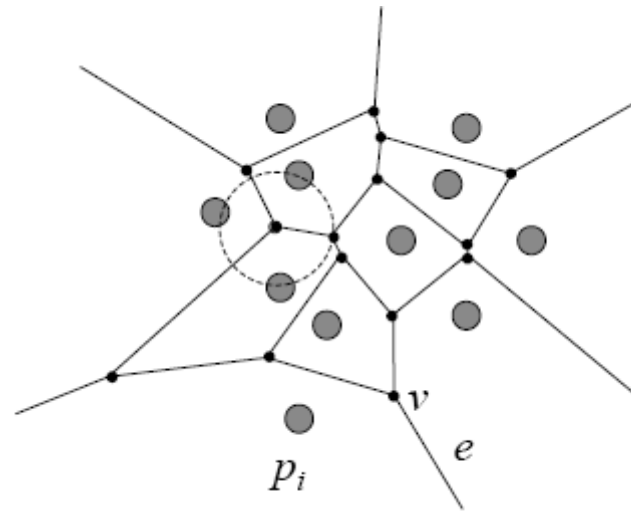
Własności:



punkt leży na krawędzi W. komórek k oraz l
jeśli jest środkiem pustego okręgu dotykającego
punkty tworzące p_k oraz p_l

pusty znaczy nie zawierający wierzchołków w swoim wnętrzu

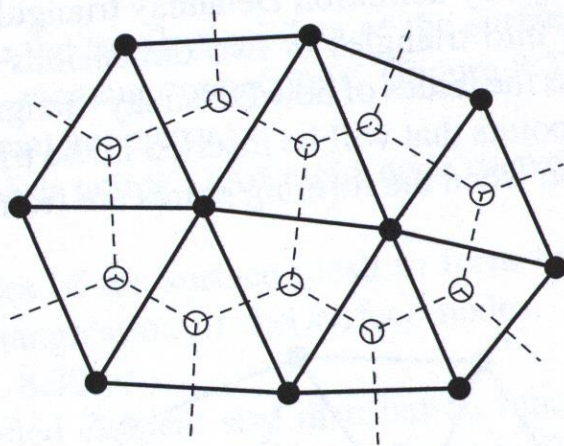
Własności cd.



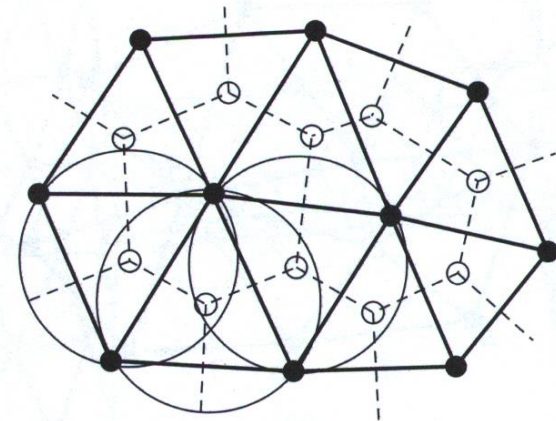
punkt jest wierzchołkiem Woronoja, jeśli środkiem pustego okręgu, który dotyka 3 lub więcej punktów tworzących

triangulacja Delauneya: graf dualny do Woronoja

łączymy krawędziami punkty tworzące sąsiednich obszarów Woronoja, które mają wspólną krawędź. Krawędź diagramu D symetralna krawędzi W.



— Delaunay triangulation
--- Voronoi diagram
(b) Delaunay triangulation



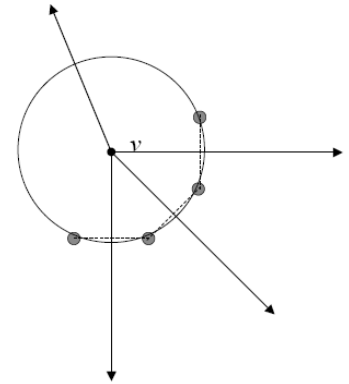
wierzchołki Woronoja:
środki pustego okręgu opisanego
na węzłach jednego elementu

Własności:

1) triangulacja Delaunaya produkuje trójkąty jeśli w P nie ma 4 punktów współśrodkowych [nie ma wierzchołków z 4-ma krawędziami Woronoja]

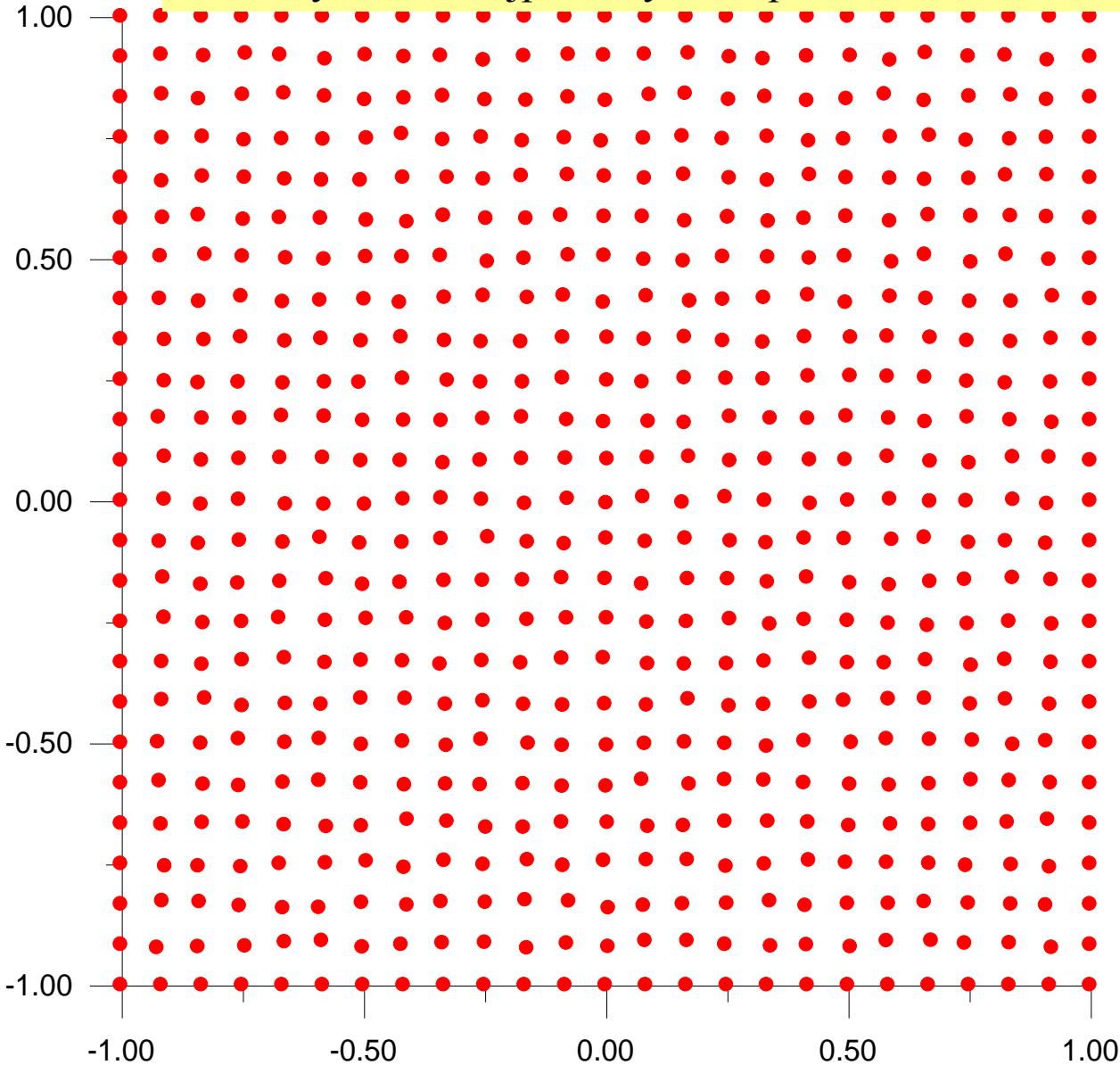
2) Każdą trójkąt D odpowiada jednemu wierzchołkowi Woronoja, który jest środkiem okręgu opisanego na trójkącie D

3) Wnętrze każdego z tych okręgów nie zawiera żadnego punktu tworzącego



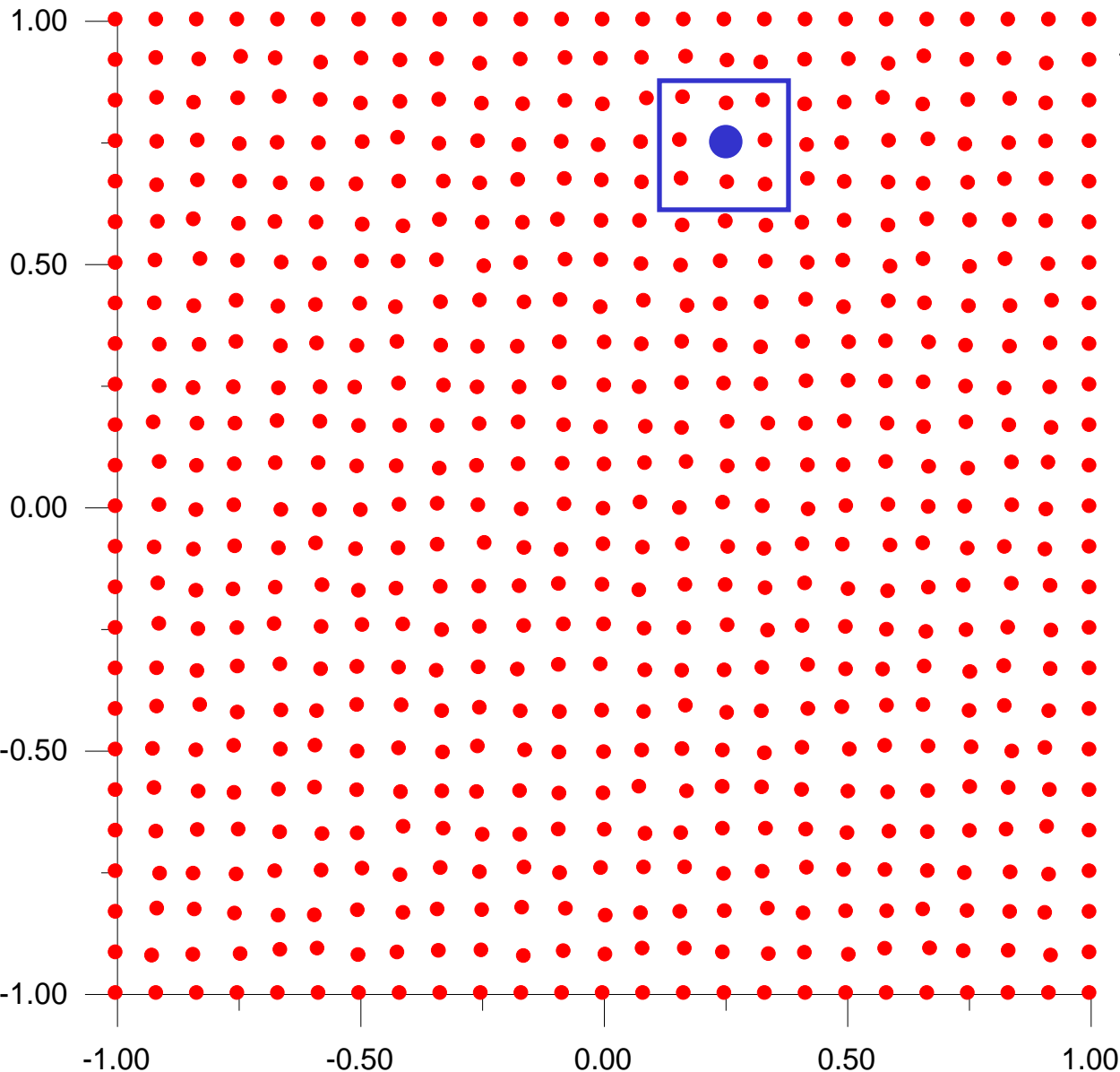
własność 1: używana przy wyborze punktów tworzących, 2- w strukturze danych
3-kryterium Delaunaya (kryterium pustego koła) – sprawdzany czy naruszany przez nowe pty,
również w prostym algorytmie generacji siatki

prosty algorytm triangulacji Delauneya z ominięciem generacji diagramu Woronoja
mało szybki, ale najprostszy do napisania = ten na laboratorium

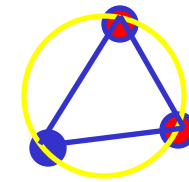


brzeg: $[-1, 1]$ na $[-1, 1]$
węzły rozkładam
ściśle równomiernie,

wewnątrz obszaru
całkowania
daje na siatce nieco
odkształconej
(aby uniknąć
więcej niż 3 punktów
na jednym okręgu)

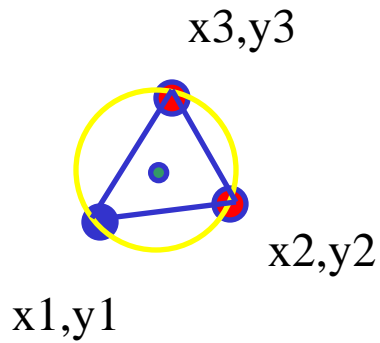


- 1) buduję elementy których wierzchołkiem jest punkt niebieski
- 2) przeglądam okolice tego punktu (niebieski kwadrat)
- 3) dla każdej uporządkowanej trójki punktów (z których jeden to wybrany niebieski konstruuje trójkąt i okrąg na nim opisany



- 4) jeśli okrąg jest *pusty* (wewnątrz nie ma innego węzła) trójkąt staje się nowym elementem

5) Gdy topologia jest ustalona, można przywrócić siatkę równomierną



okrąg opisany na trójkącie:
potrzebny środek: x_a, y_a , oraz promień R

jak sprawdzić, że niewspółliniowe:
iloczyn wektorowy $\mathbf{r}_1 - \mathbf{r}_2$ z $\mathbf{r}_1 - \mathbf{r}_3$ zero jeśli współliniowe

$$(\mathbf{r}_1 - \mathbf{r}_2) \times (\mathbf{r}_1 - \mathbf{r}_3) = \begin{vmatrix} i & j & k \\ (x_1 - x_2) & (y_1 - y_2) & 0 \\ (x_1 - x_3) & (y_1 - y_3) & 0 \end{vmatrix} = (0, 0, \underbrace{(x_1 - x_2)(y_1 - y_3) - (x_1 - x_3)(y_1 - y_2)}_w)$$

$$(x_1 - x_a)^2 + (y_1 - y_a)^2 = R^2$$

$$(x_2 - x_a)^2 + (y_2 - y_a)^2 = R^2$$

$$(x_3 - x_a)^2 + (y_3 - y_a)^2 = R^2$$

$$(1-3) \longrightarrow (x_1 - x_a)^2 + (y_1 - y_a)^2 - (x_2 - x_a)^2 - (y_2 - y_a)^2 = 0$$

$$(1-2) \longrightarrow (x_1 - x_a)^2 + (y_1 - y_a)^2 - (x_3 - x_a)^2 - (y_3 - y_a)^2 = 0$$

$$x_1^2 - 2x_1x_a + y_1^2 - 2y_1y_a - x_2^2 + 2x_2x_a - y_2^2 + 2y_2y_a = 0$$

$$2x_a(x_2 - x_1) + 2y_a(y_2 - y_1) + x_1^2 + y_1^2 - x_2^2 - y_2^2 = 0$$

$$2x_a(x_2 - x_1) + 2y_a(y_2 - y_1) + x_1^2 + y_1^2 - x_2^2 - y_2^2 = 0$$

$$2x_a(x_2 - x_1) + 2y_a(y_2 - y_1) + \mathbf{r}_1^2 - \mathbf{r}_2^2 = 0$$

drugie równanie, jak pierwsze z 3 zamiast 2

$$2x_a(x_3 - x_1) + 2y_a(y_3 - y_1) + \mathbf{r}_1^2 - \mathbf{r}_3^2 = 0$$

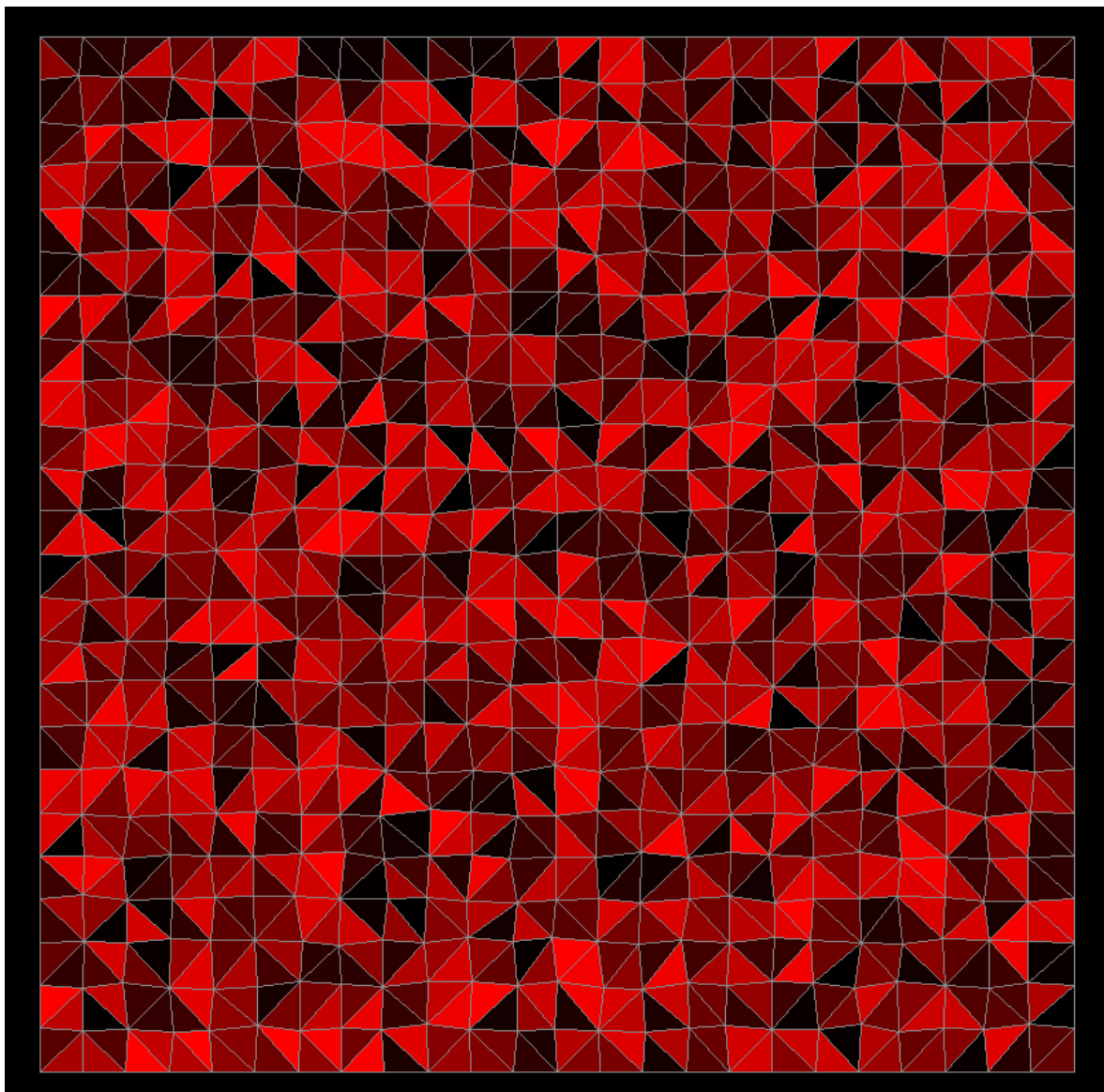
(1)*(x3-x1)-(2)*(x2-x1) [wyeliminować x_a]

$$2y_a(y_2 - y_1)(x_3 - x_1) - 2y_a(y_3 - y_1)(x_2 - x_1) + (\mathbf{r}_1^2 - \mathbf{r}_2^2)(x_3 - x_1) - (\mathbf{r}_1^2 - \mathbf{r}_3^2)(x_2 - x_1) = 0$$

$$y_a = \frac{1}{2w} \left((\mathbf{r}_1^2 - \mathbf{r}_2^2)(x_3 - x_1) - (\mathbf{r}_1^2 - \mathbf{r}_3^2)(x_2 - x_1) \right)$$

$$x_a = -\frac{1}{2w} \left((\mathbf{r}_1^2 - \mathbf{r}_2^2)(y_3 - y_1) - (\mathbf{r}_1^2 - \mathbf{r}_3^2)(y_2 - y_1) \right)$$

$$R = \left((x_1 - x_a)^2 + (y_1 - y_a)^2 \right)^{1/2}$$

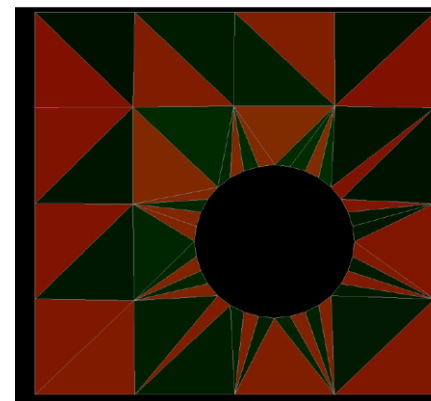
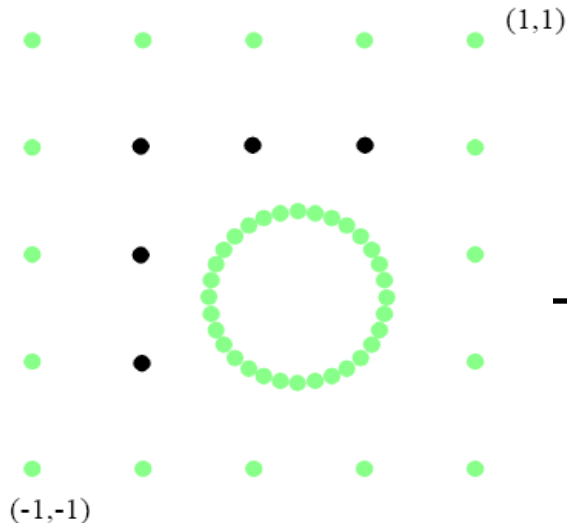
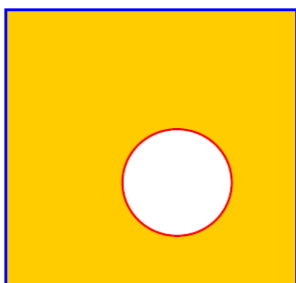


Wynik:
obszar całkowania
podzielony
na trójkątne elementy

random.exe

Laboratorium:

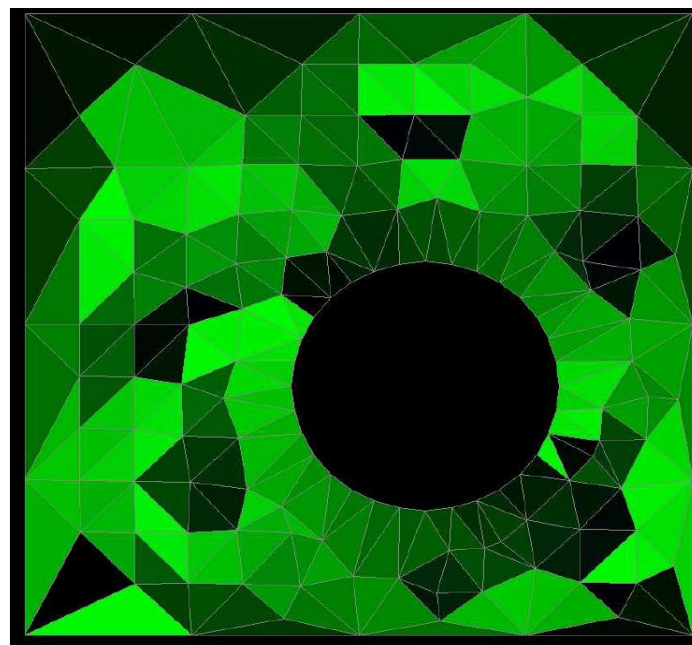
początkowe położenie węzłów
czarne punkty – lekko przesuniemy o losowy wektor



symulacja adaptacji siatki: funkcja „ważności” liczona w środku ciężkości elementu

$$f(x_c, y_c) = \exp(-(x_c - 0.2)^2 - (y_c + 0.2)^2) \longrightarrow$$

razy jego pole ma nie przekraczać 1/80



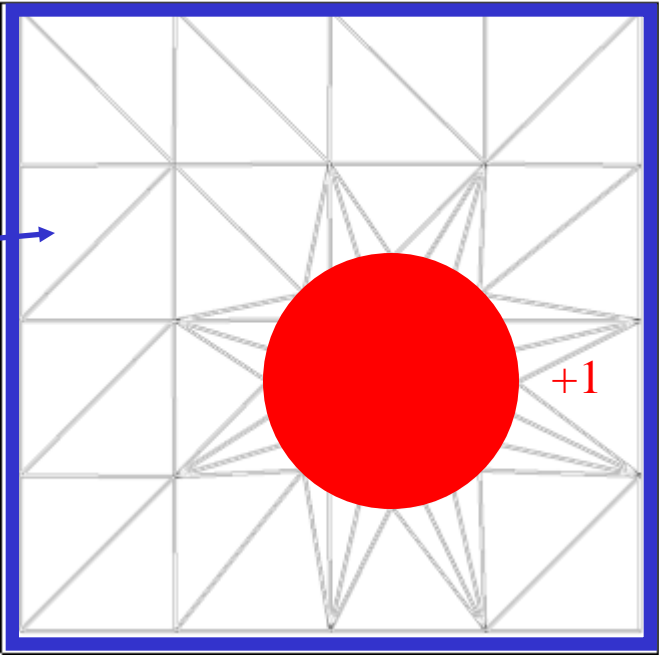
Automatyczna rafinacja siatki

- 1) MES = Galerkin = metoda wariacyjna
- 2) optymalna siatka=minimum funkcjonału
- 3) w każdym z elementów liczymy przyczynek do funkcjonału
- 4) w tym, w którym przyczynek osiąga maksymalną wartość = nowy punkt w środku ciężkości elementu

$$\nabla^2 u = 0$$

siatka początkowa

punkty tworzące na okręgu
(wewnętrzny brzegowy)
na brzegu i na siatce równomiernej
wewnątrz



Rozwiązujemy równanie $Su=F$

Liczmy działanie:

$$a = \frac{1}{2} \int_{\Omega} (\nabla u)^2 d\Omega,$$

dzielimy na sumę po elementach

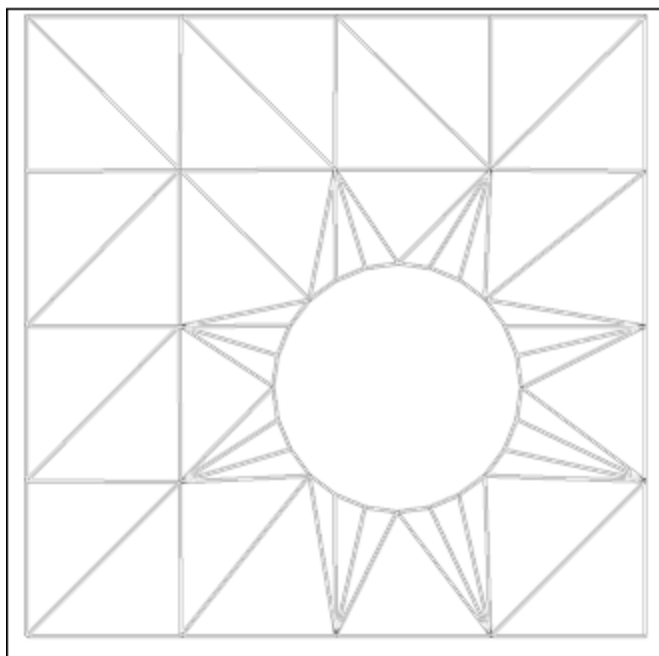
$$a = \sum_m a_m, \quad a_m = \frac{1}{2} \int_{\Omega_m} (\nabla u)^2 d\Omega_m$$

działanie w m-tym elemencie liczymy na podstawie lokalnej macierzy sztywności:

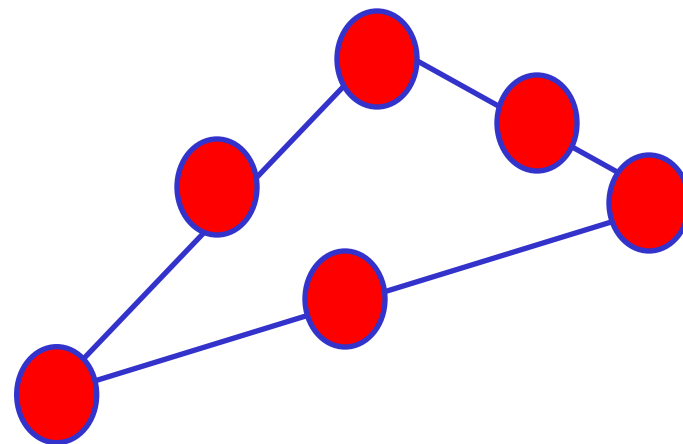
$$a_m = \frac{1}{2} \sum_{kl} u_k^m u_l^m E_{kl}^m$$

Element wnoszący duży przyczynek do funkcjonału:
albo zbyt wielki, albo wewnątrz elementy zbyt duży gradient rozwiązania
(pola elektrycznego lub temperatury)

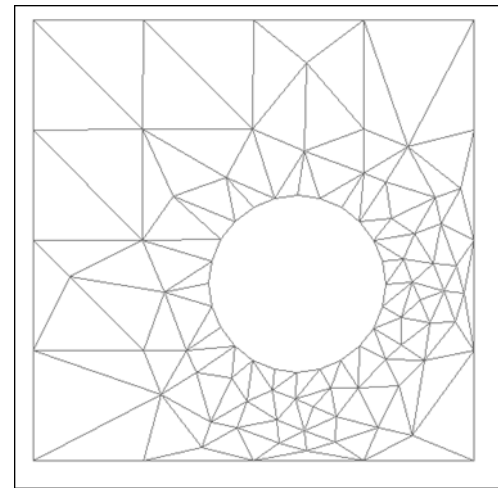
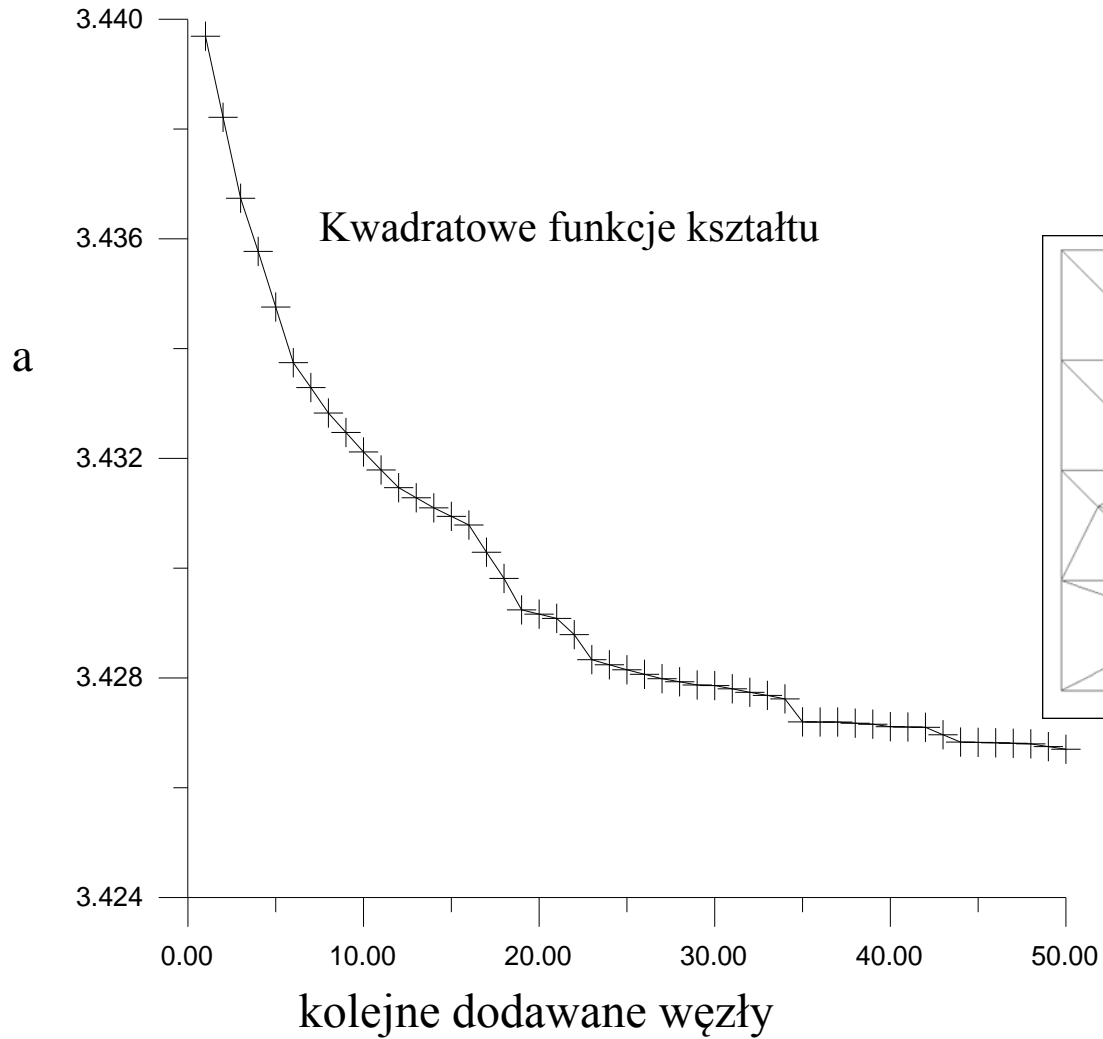
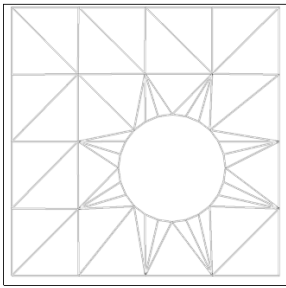
siatka startowa



Wyniki:
dla bazy kwadratowej
(6 węzłów/ element)

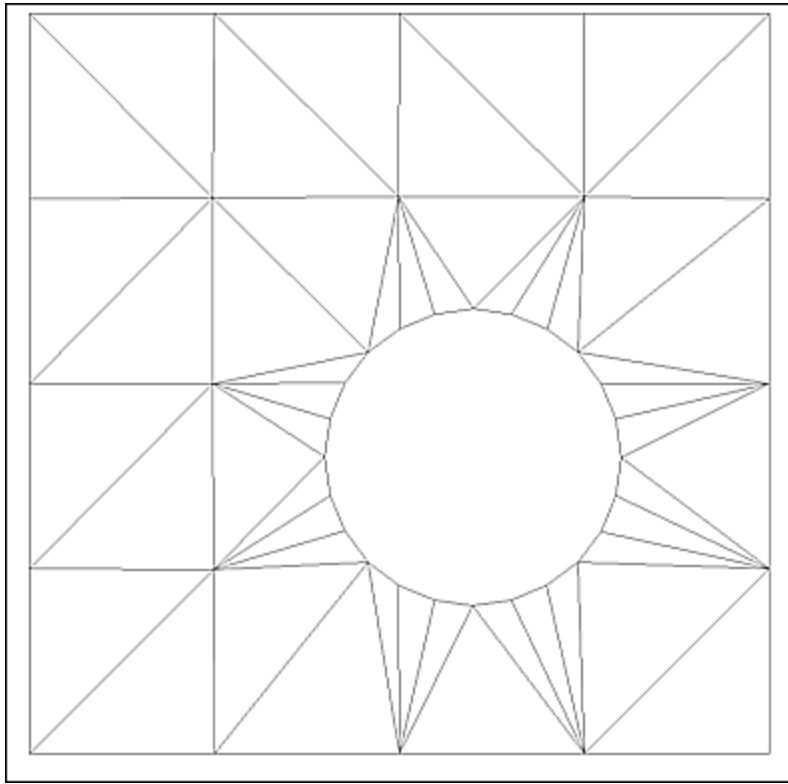


FILM:
0_NOXJ_NOS.EXE

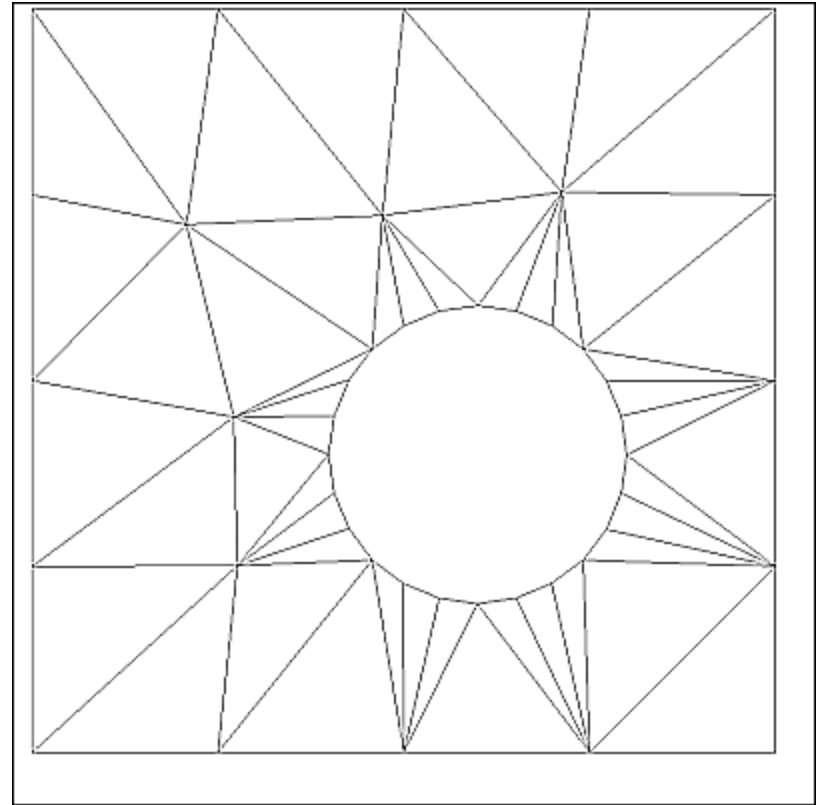


Uniknąć dystorsji:
laplasjańskie wygładzanie

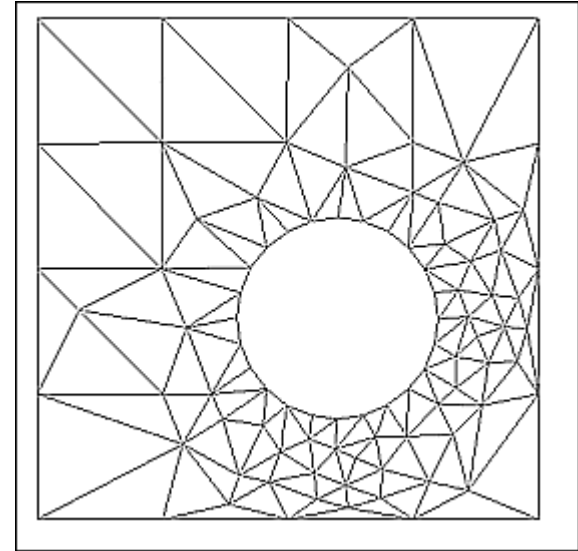
startowa



startowa wygładzona

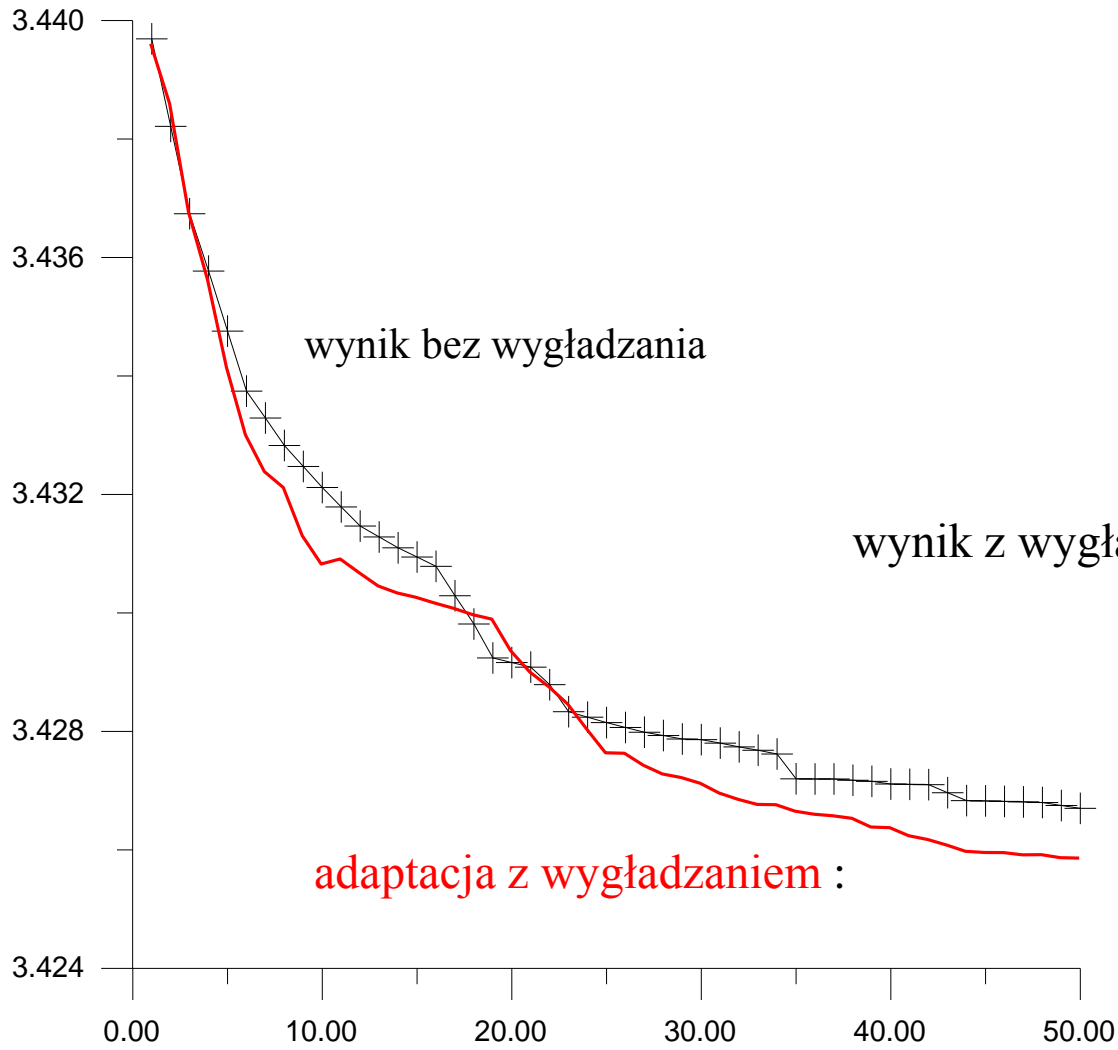


wynik bez wygładzania

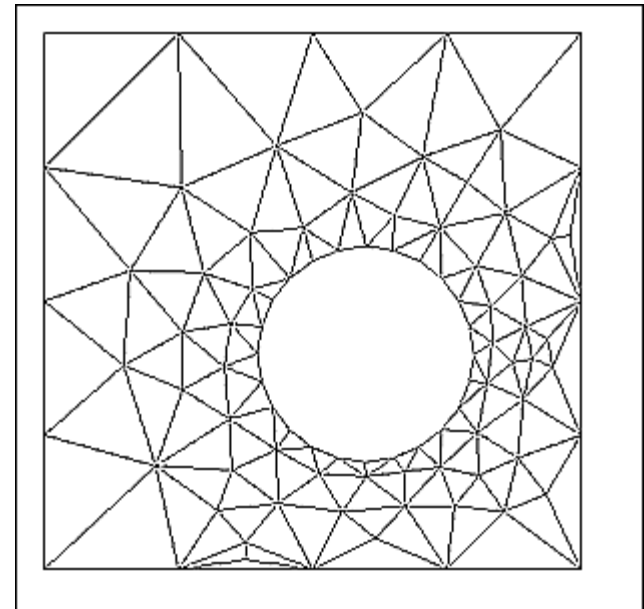


wynik bez wygładzania

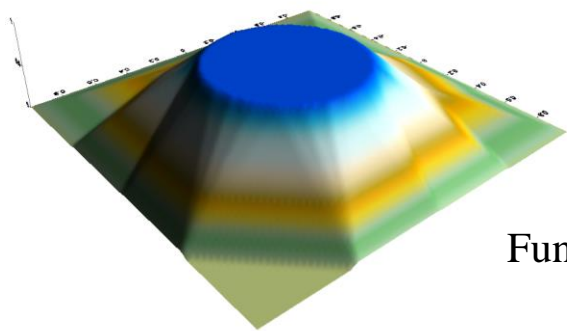
adaptacja z wygładzaniem :



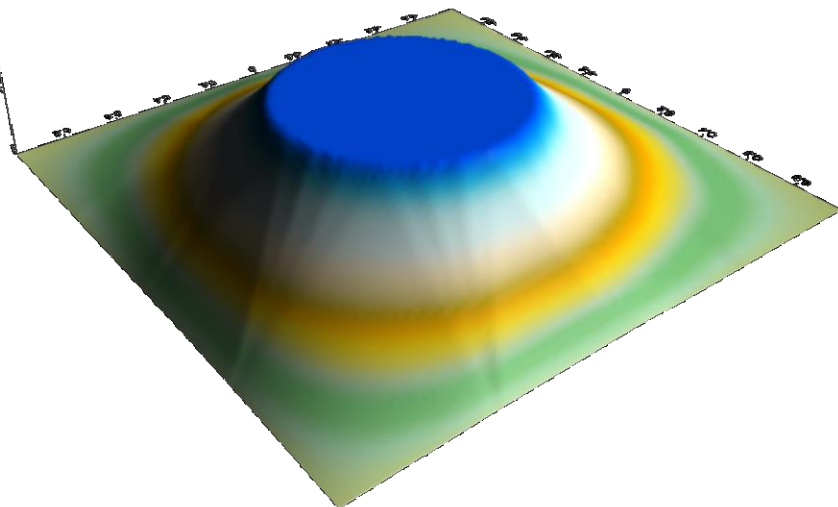
wynik z wygładzaniem



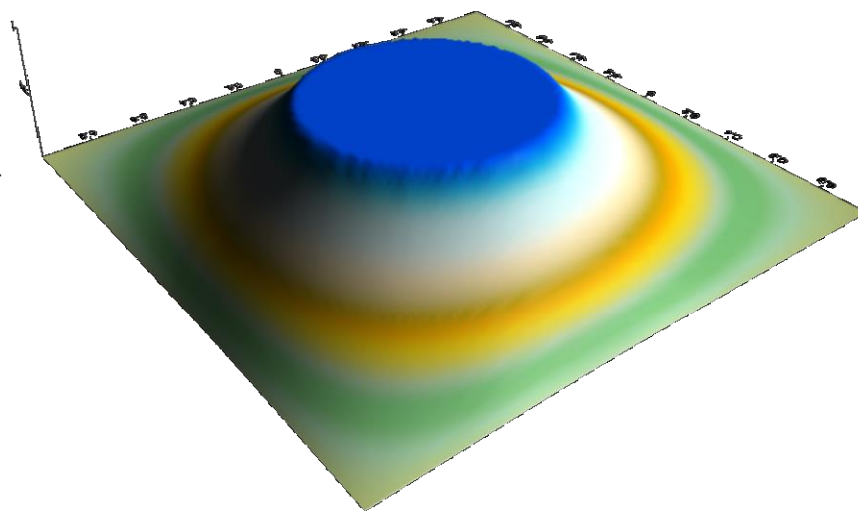
Porównanie różnych funkcji kształtu: wyniki dla siatki startowej



Funkcje liniowe

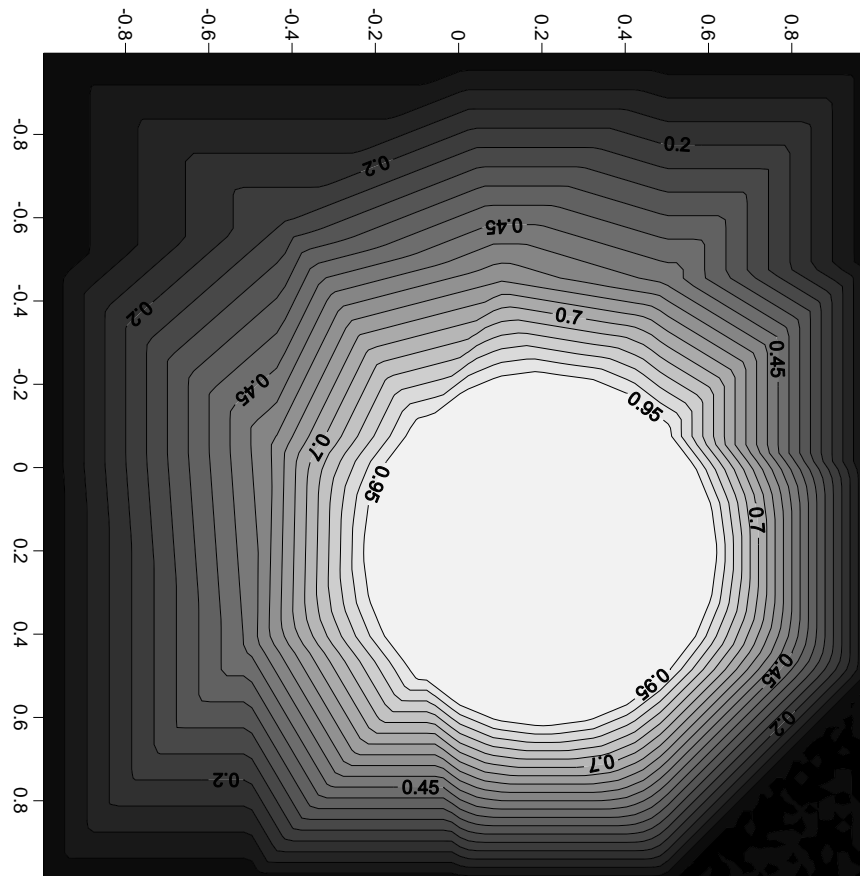
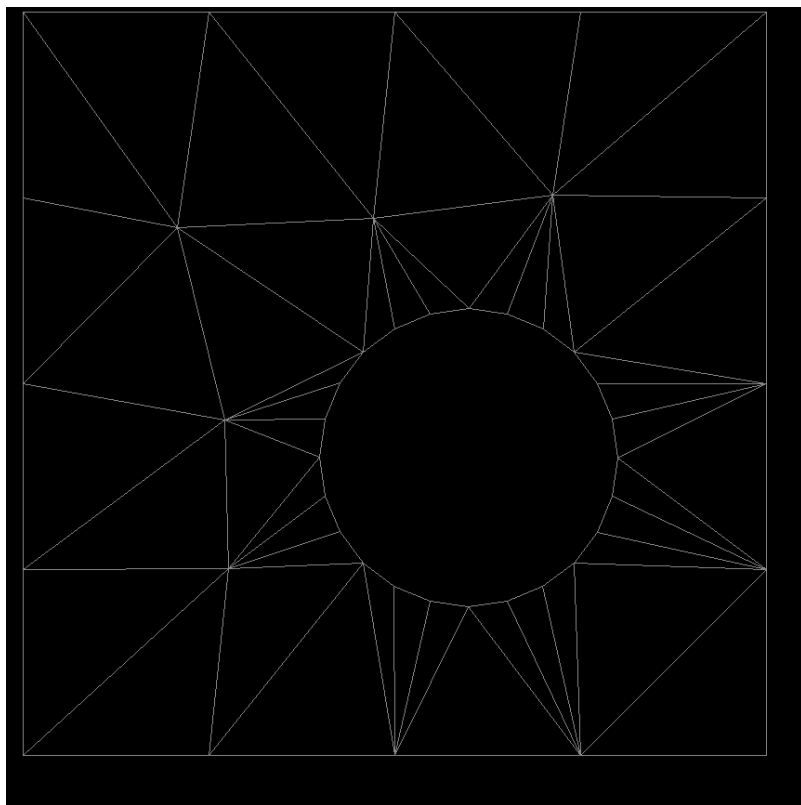


funkcje kwadratowe

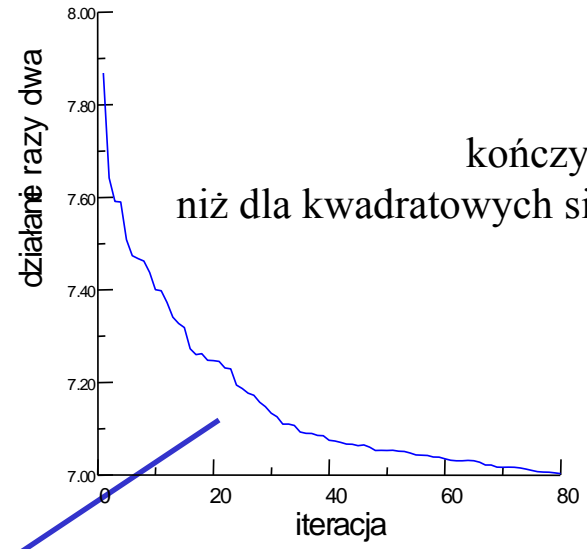
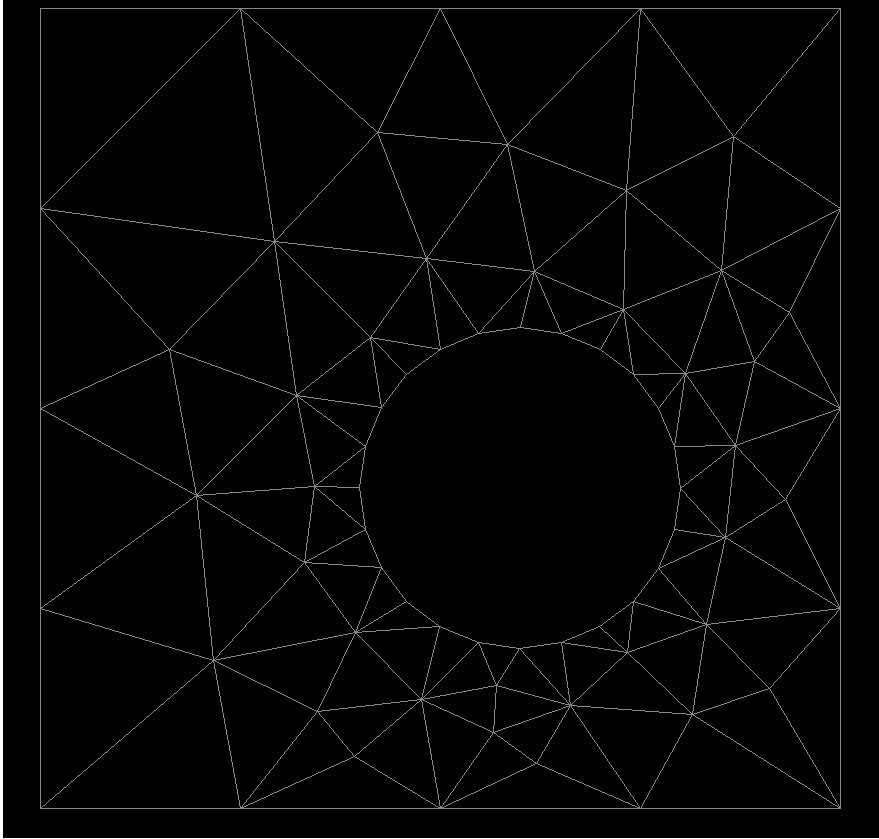


funkcje kubiczne

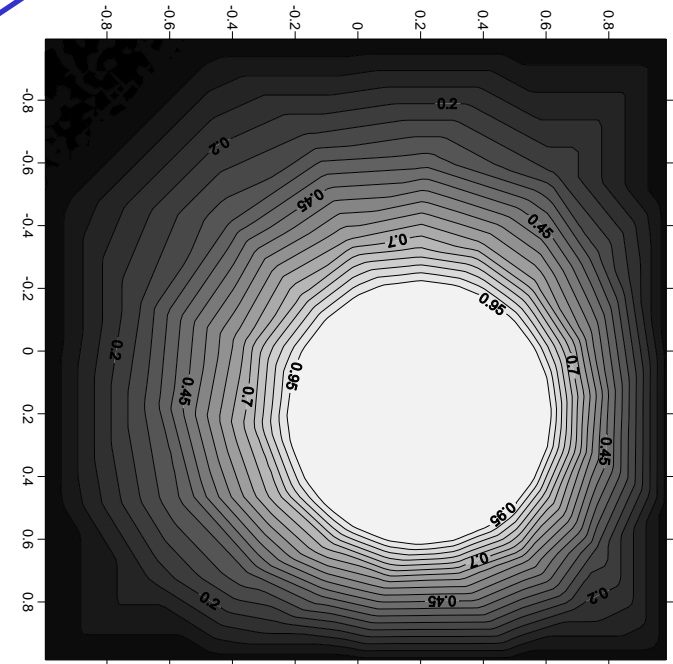
liniowe funkcje kształtu: 3 węzły na element



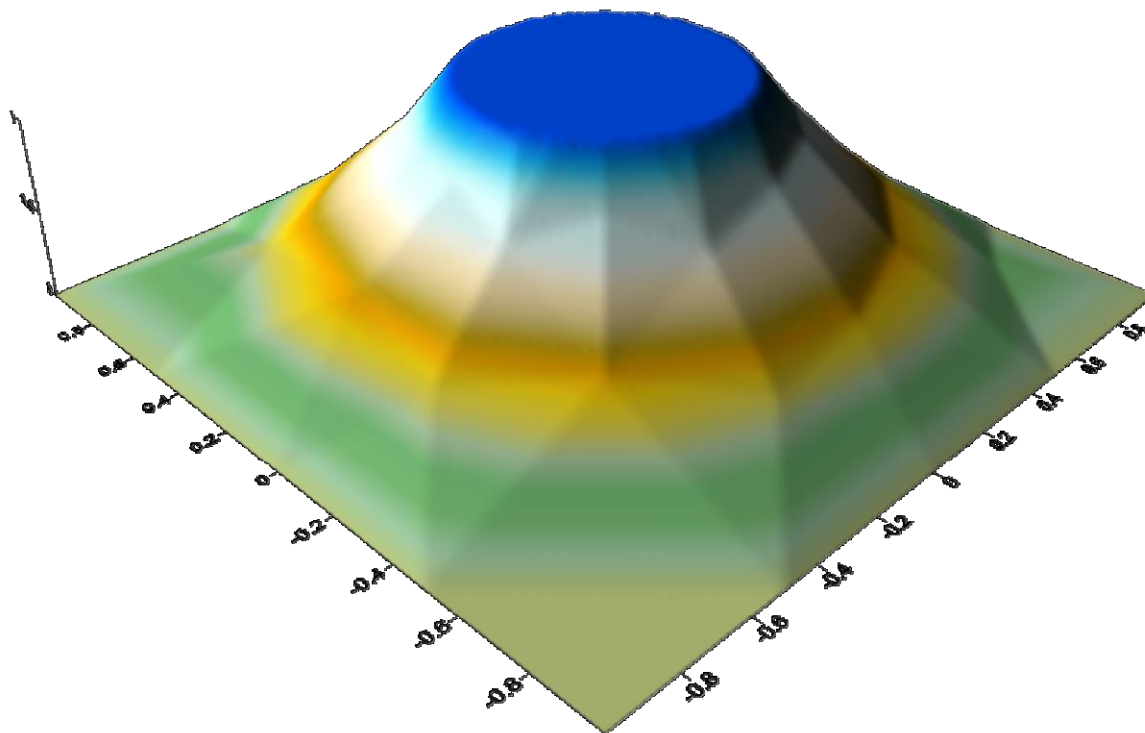
liniowe funkcje kształtu: 3 węzły na element



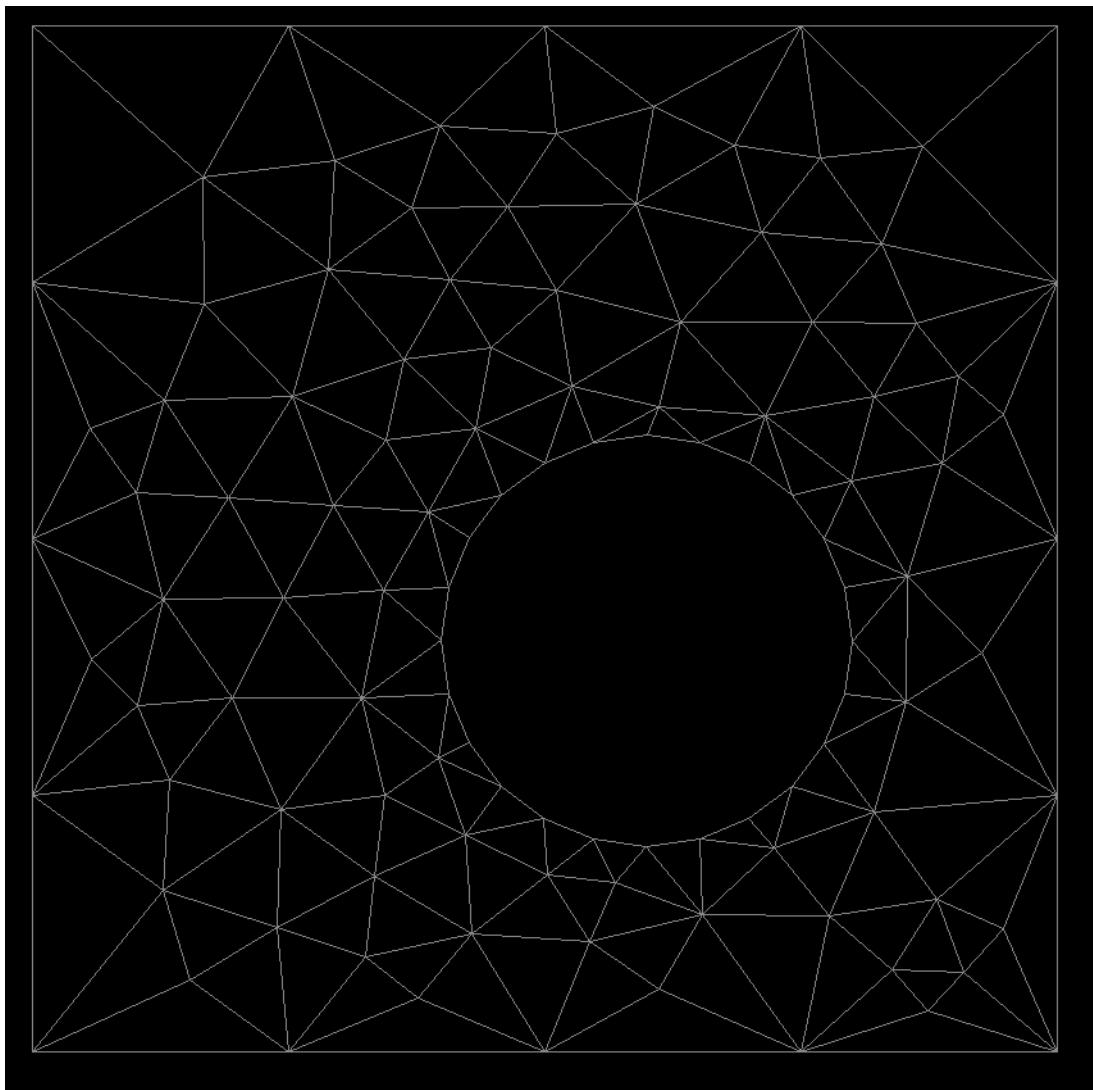
kończy się wyżej niż dla kwadratowych się zaczyna

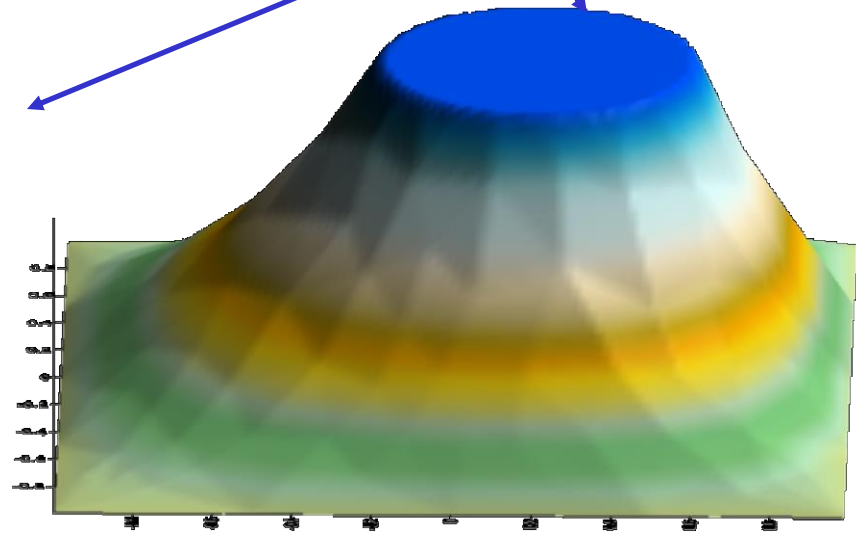
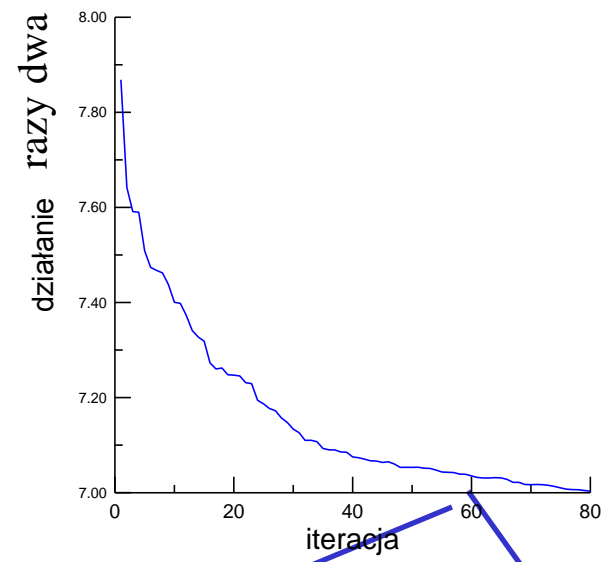
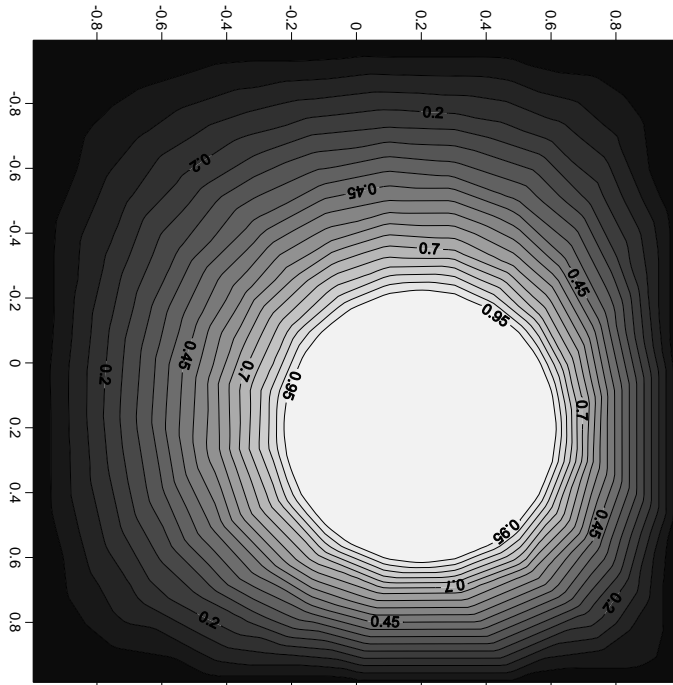


liniowe funkcje kształtu: 3 węzły na element
Po optymalizacji:



liniowe funkcje kształtu: 3 węzły na element





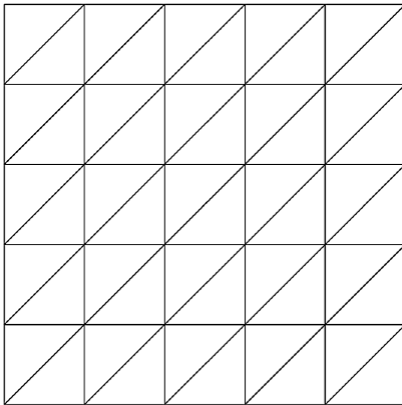
płaskie ściany: kosztują
 Zbiega się do wartości $>$ niż start dla funkcji kwadratowych

Proponowany powyżej sposób:
siatka jest optymalizowana bez żadnej struktury,
dodawane są kolejne węzły i powtarzana triangulacja.

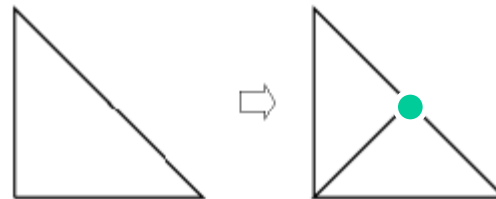
Alternatywa: hierarchiczna optymalizacja siatki.

Przykład: rekurencyjny podział elementów

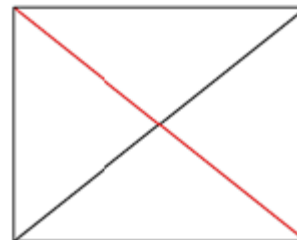
Zaczynamy od siatki równomiernej trójkątów prostokątnych :

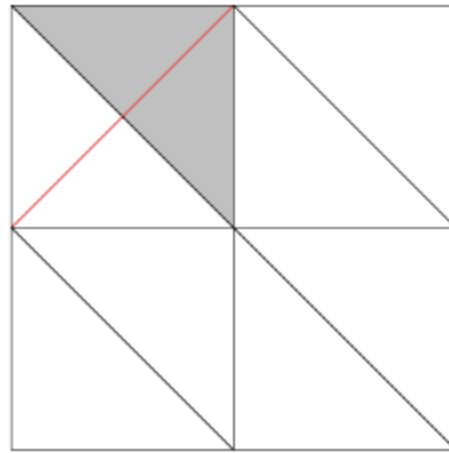
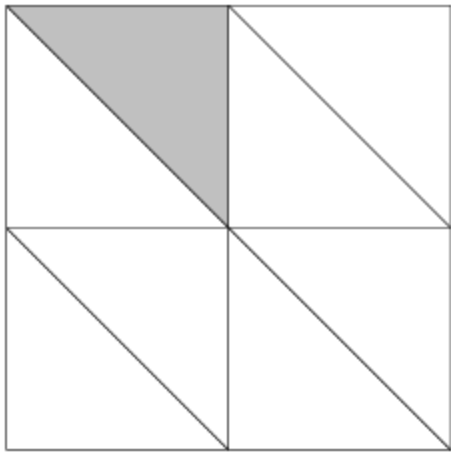


Badamy lokalny wkład do działania lub błędu
dzielimy na pół trójkąty wnoszące największe przyczynki
Dzielimy: znaczy dodajemy nowy węzeł



Z powodu nowego węzła należy podzielić również sąsiada:

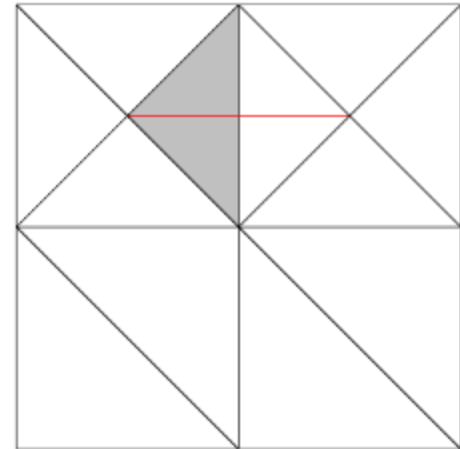
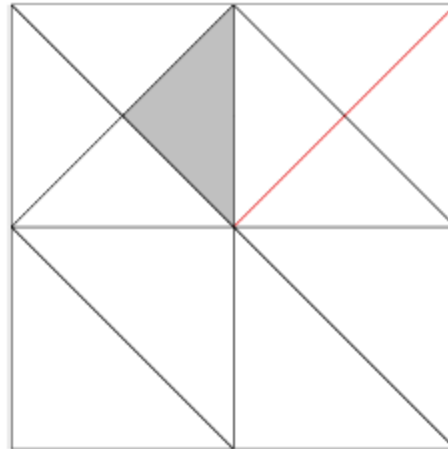
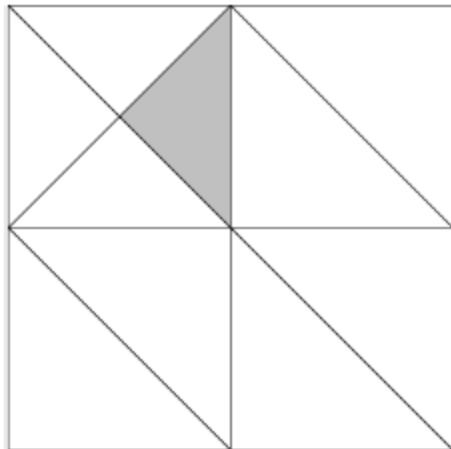




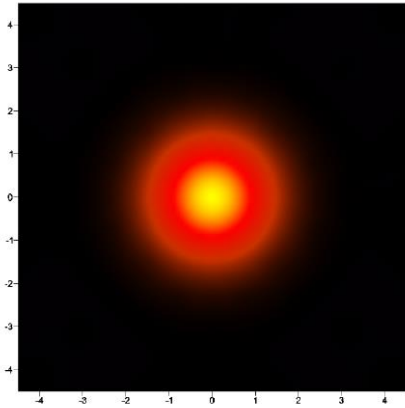
Element do podziału jest tego samego rozmiaru co sąsiad:
wystarczy podzielić sąsiada

Element do podzielenia zaznaczony szarym kolorem

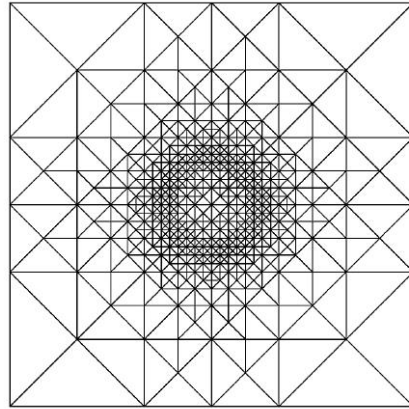
Gdy element do podziału ma mniejsze pole niż sąsiad
trzeba najpierw podzielić sąsiada.
Potrzeba więcej podziałów



Zastosowanie:

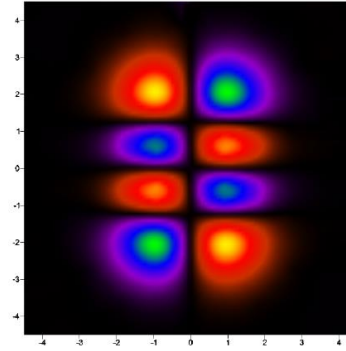


(a) funkcja falowa

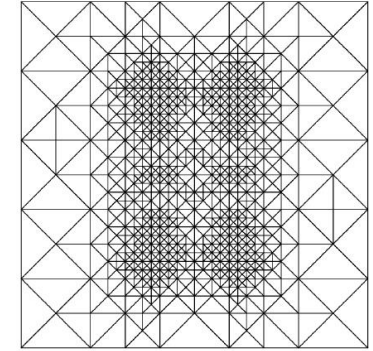


(b) siatka

Rysunek 36: Wyniki dla stanu $n_x = 0, n_y = 0$

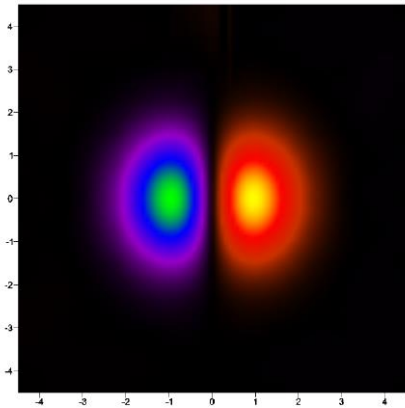


(a) funkcja falowa

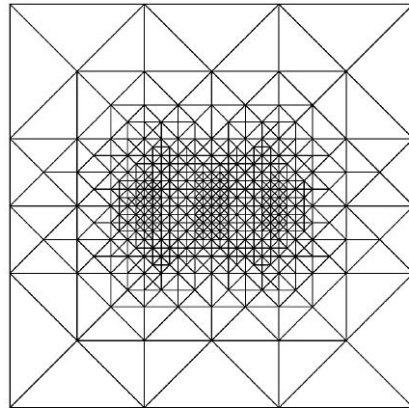


(b) siatka

Rysunek 40: Wyniki dla stanu $n_x = 1, n_y = 3$



(a) funkcja falowa



(b) siatka

Rysunek 37: Wyniki dla stanu $n_x = 1, n_y = 0$

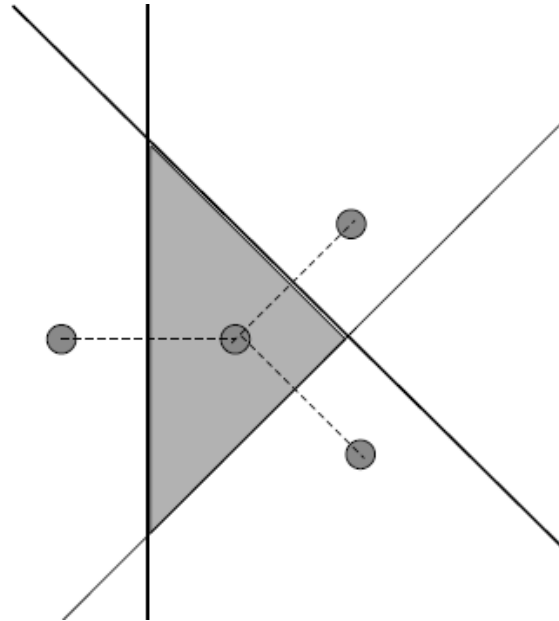
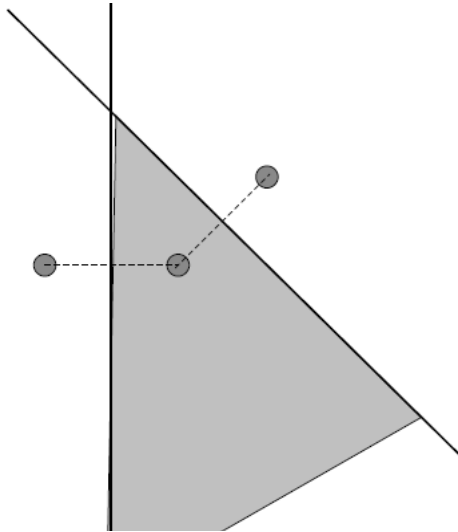
Wyniki z pracy mgr. Michała Stalmacha
(oscylator harmoniczny 2D: kryterium
podziału – maksymalny wkład do energii
z elementu

Triangulacja Delauneya - uzupełnienie

Algorytm omówiony wyżej: łatwy do zaprogramowania lecz wolny, opiera się na przeglądaniu trójek punktów: złożoność: $O(n^3)$

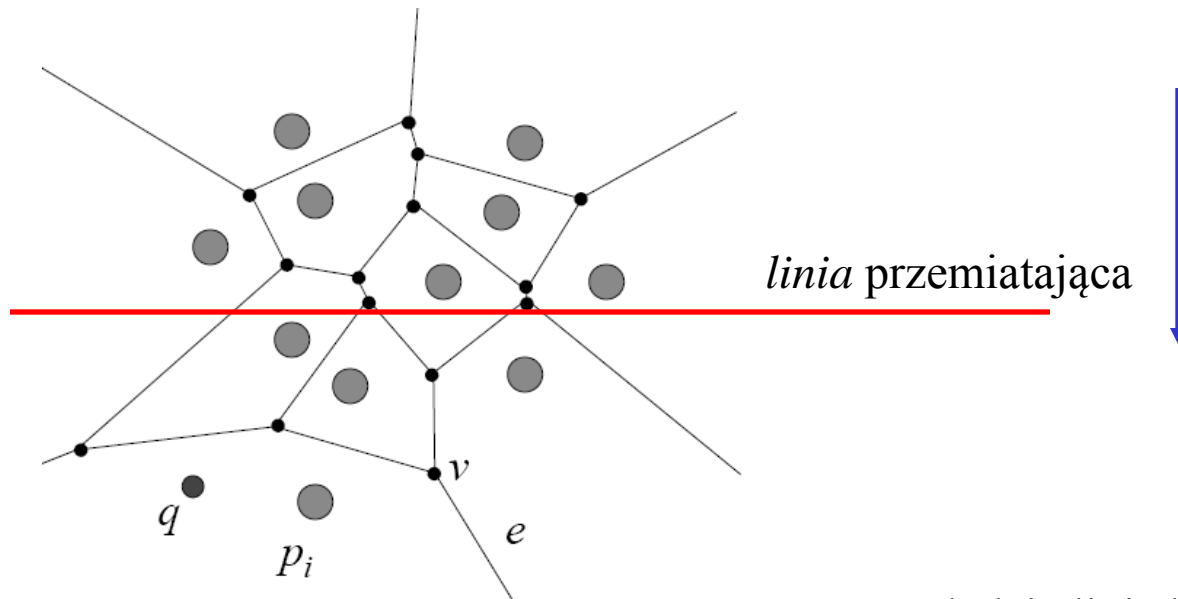
szybszy : $O(n^2 \log n)$ przeglądanie par

– z algorytmem na wyznaczanie przecięć (części wspólnej) półpłaszczyzn

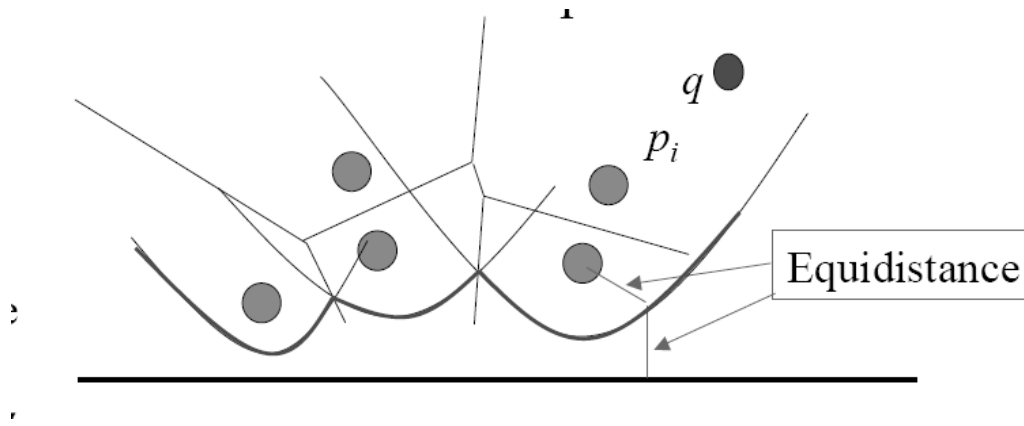


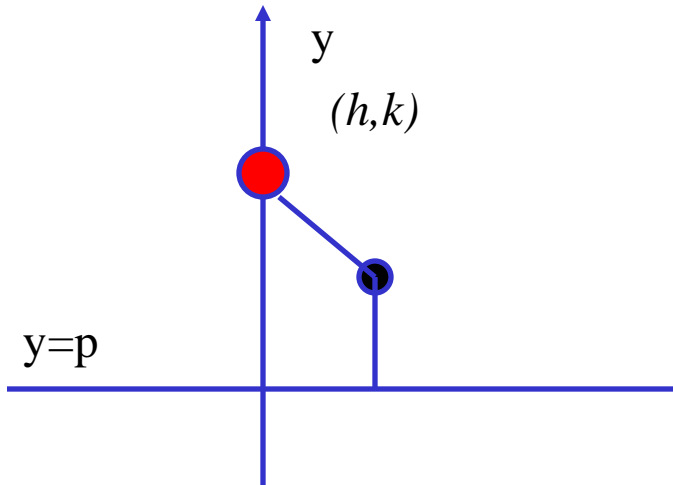
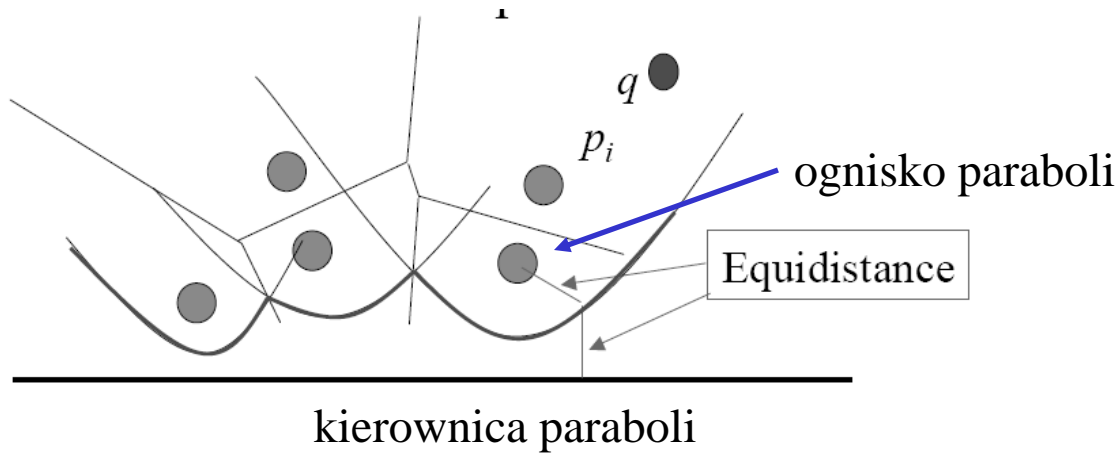
Algorytm linii przemiatającej (Fortune): przeglądanie odległości punktów od linii przemiatającej

Allen Miu



mamy śledzić linię brzegową
powyżej której bliżej jest
do punktów tworzących niż do *linii*

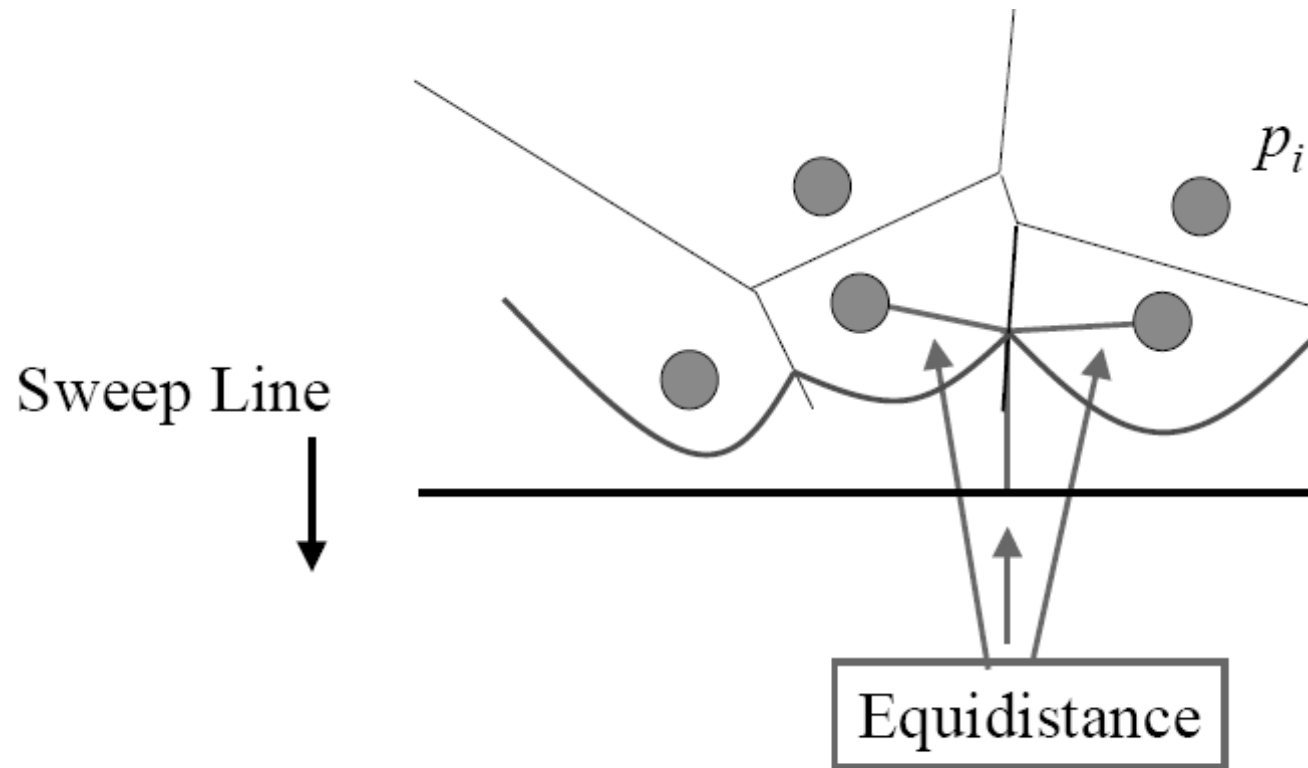




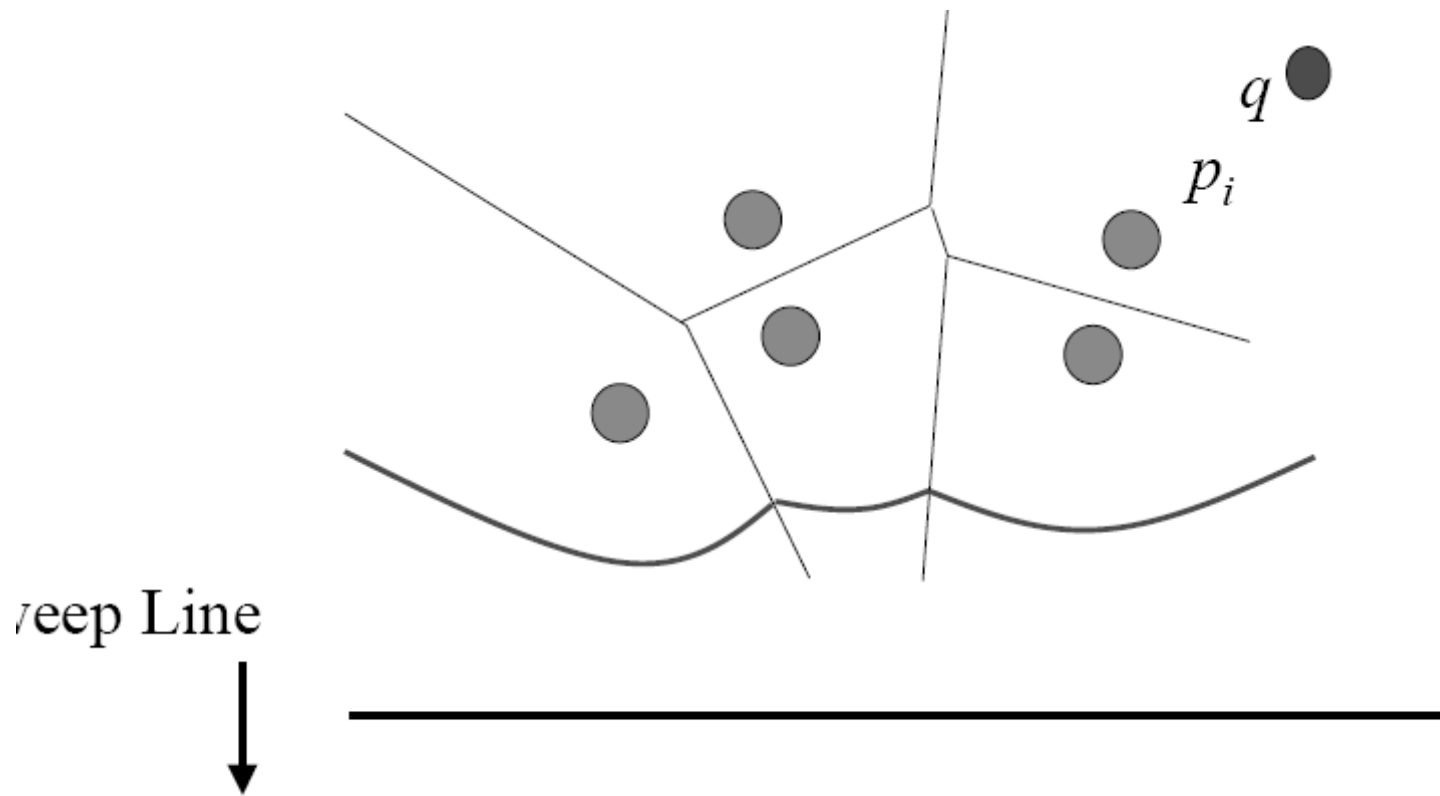
$$(x-h)^2 + (y-k)^2 = (y-p)^2$$

$$(x-h)^2 + 2y(p-k) + k^2 - p^2 = 0$$

$$y = (x-h)^2 / 2(p-k) + (k+p)/2$$

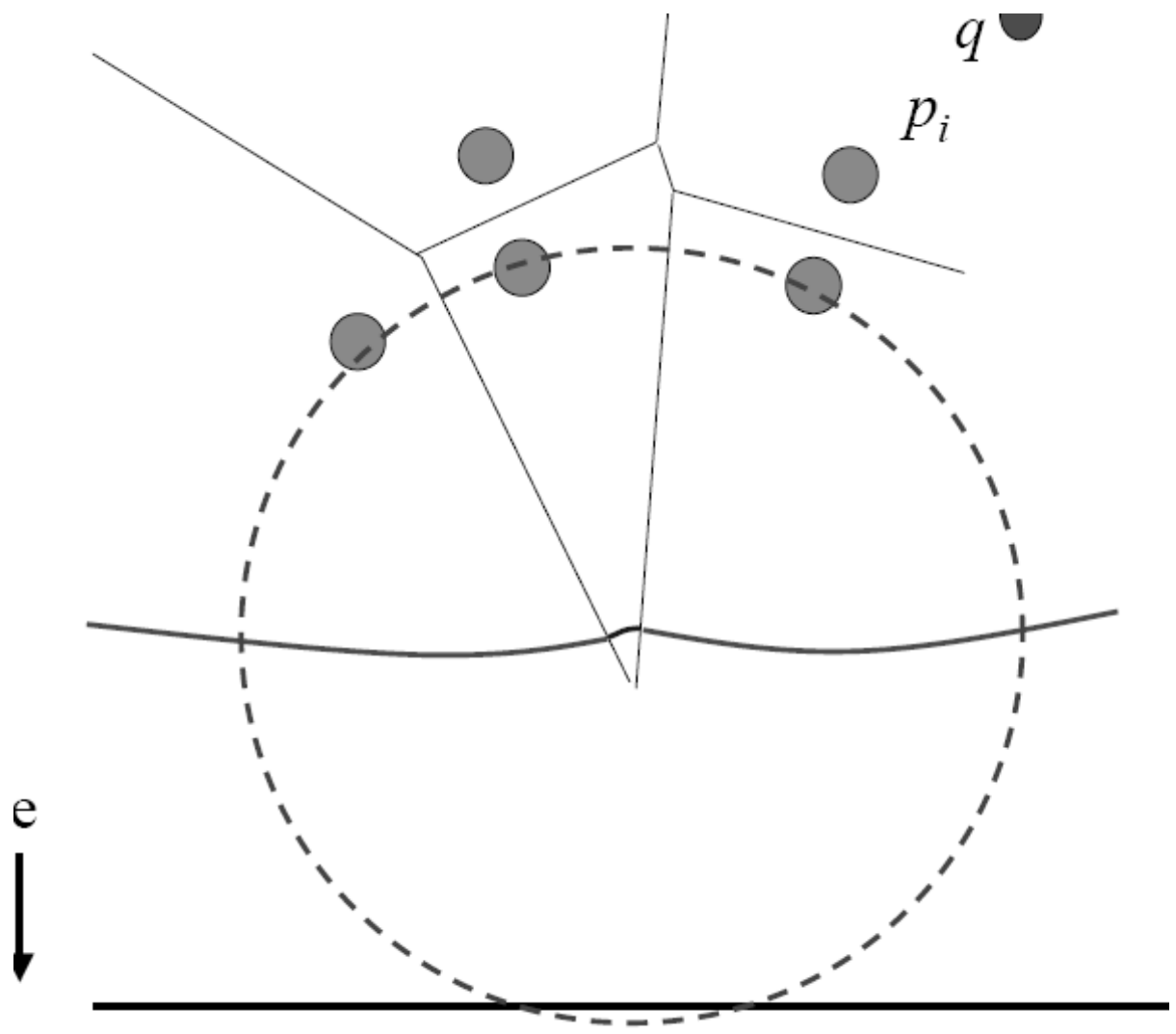


krawędzie Woronoja: przecięcia parabol

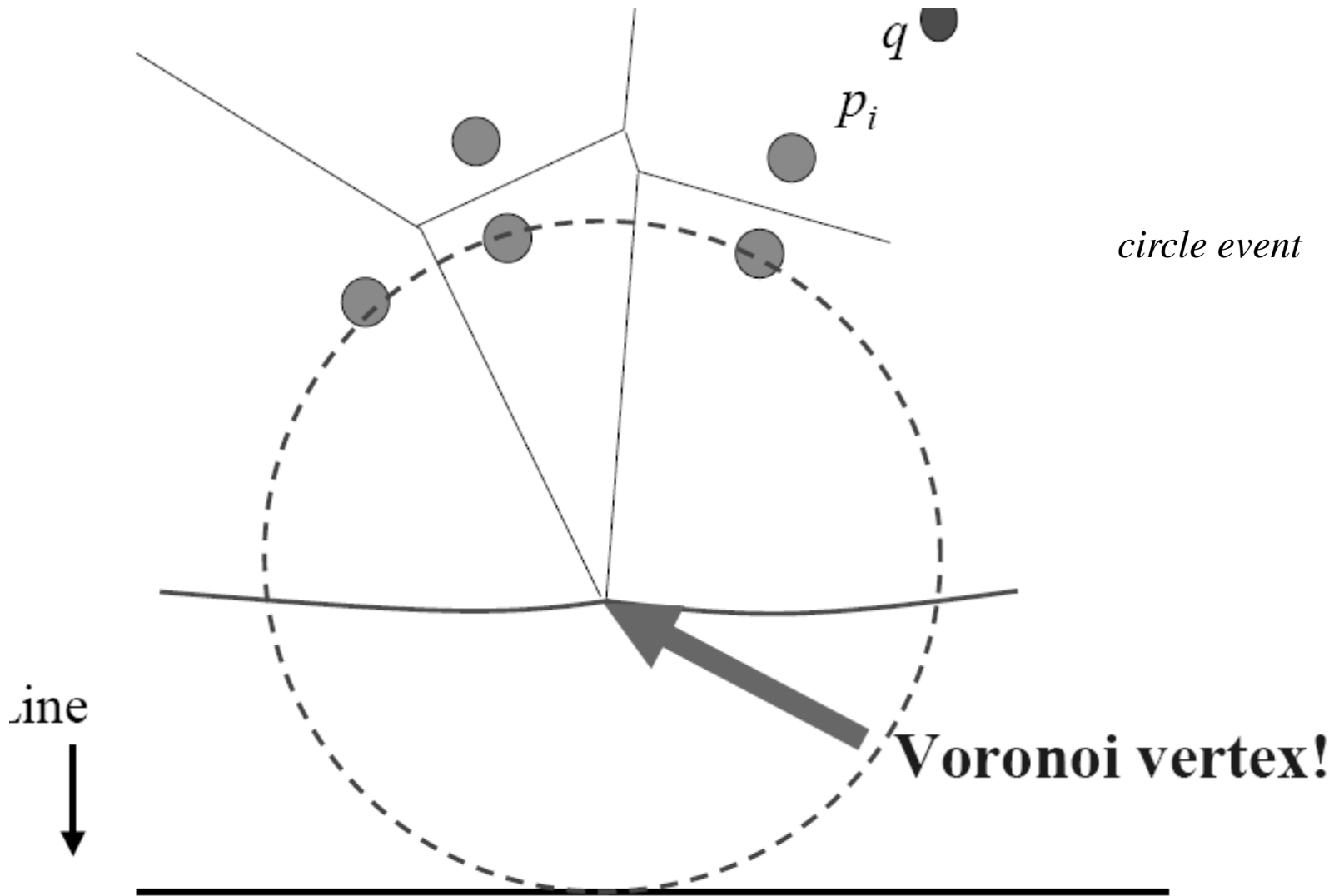


Deep Line

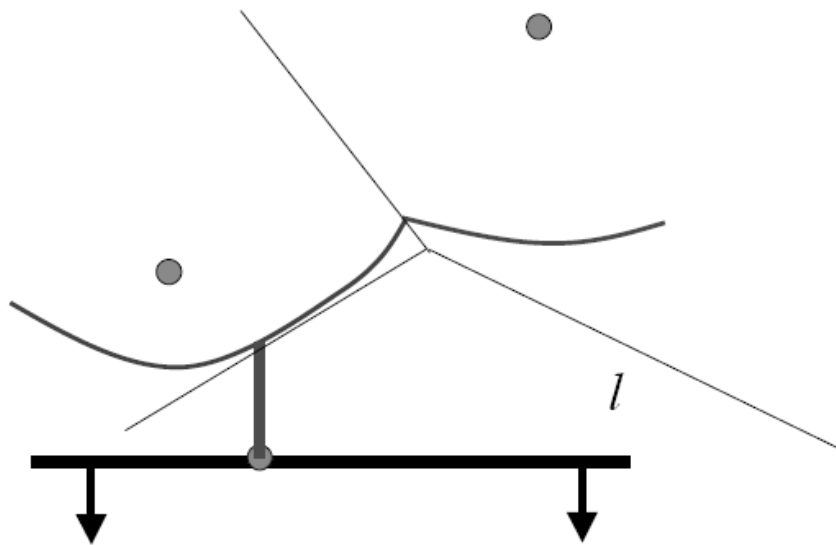
łuki ulegają wygładzeniu



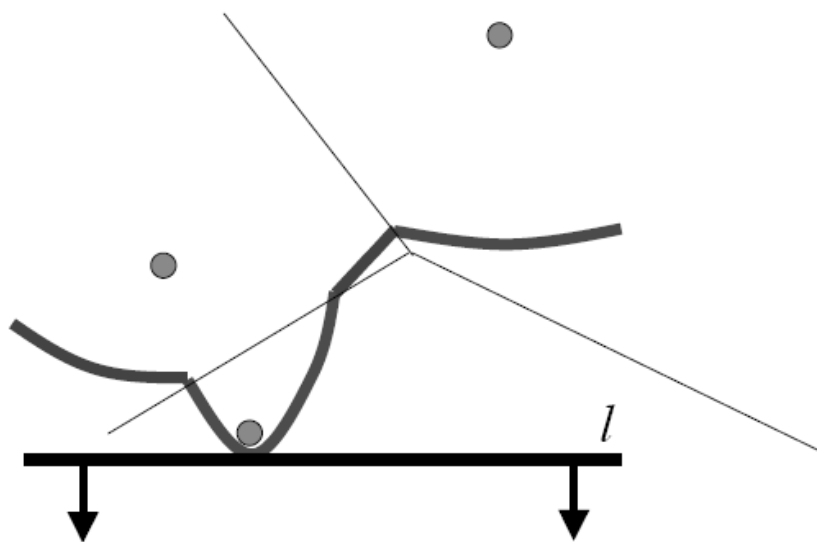
jeden z łuków robi się coraz krótszy



i ostatecznie znika: w wierzchołku Woronoja



site event



nowy łuk pojawi się gdy nowy punkt powyżej linii

$O(n \log(n))$ optymalny

się kończy gdy wszystkie punkty powyżej linii