

# Porównania detektorów i metod aproksymujących

## Parametry detektorów

W przeprowadzonych symulacjach wykorzystano różne parametry detektorów (tabela 1).

Tabela 1. Parametry detektorów.

Parametr	Detektor 1	Detektor 2	Detektor 3	Detektor 4
Wielkość piksela [ $\mu\text{m}$ ]	75	100	50	50
Sigma chmury ładunku [ $\mu\text{m}$ ]	6.31	16	6.31	16
Liczba ładunków [ $e^-$ ]	2200	4970	2200	4970
Sigma szumu [ $e^-$ RMS]	50	200	50	200

## Parametry metod aproksymujących

Wszystkie metody aproksymujące wykorzystywały zmienne typu float.

Przybliżenie Taylora wykorzystywało przybliżenie rzędu 10 (chyba, że stwierdzono inaczej).

Przybliżenie z wykorzystaniem tablicy LUT wykorzystywało tablicę wielkości 50 (chyba, że stwierdzono inaczej).

## Porównanie z podziałem na oś X i Y

Porównanie z podziałem na oś X i Y zostało przeprowadzone dla przekątnej piksela [pozycje (0, 0), (1, 1), (2, 2), ..., (n-1, n-1), (n, n), gdzie n to wielkość piksela, a jednostką jest 1  $\mu\text{m}$ ].

Dla każdej pozycji symulowano 10000 razy uderzenie fotonu w detektor i aproksymowano pozycję uderzenia oraz liczone błąd względny według wzoru:

$$error = |r - c|,$$

gdzie:

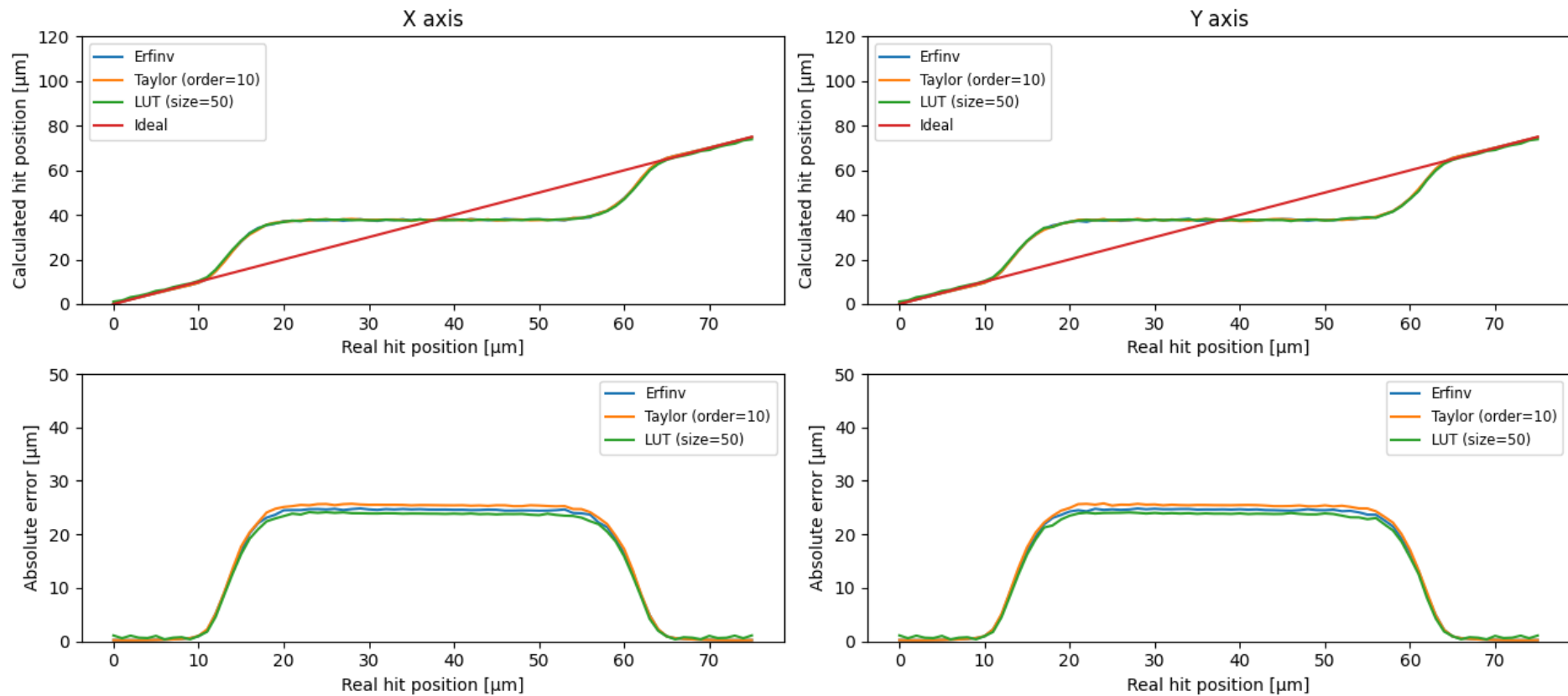
r – idealna pozycja uderzenia w detektor w danej osi,

c – obliczona pozycja uderzenia w detektor w danej osi.

Następnie wszystkie dane uśredniono, a wyniki przedstawiono na rysunkach od 1 do 4.

### Calculating methods comparison after 10000 hits

pixel size =  $75\mu\text{m}$ , charge cloud  $\sigma = 6.31\mu\text{m}$   
 number of charges = 2200e, noise  $\sigma = 50\text{e RMS}$

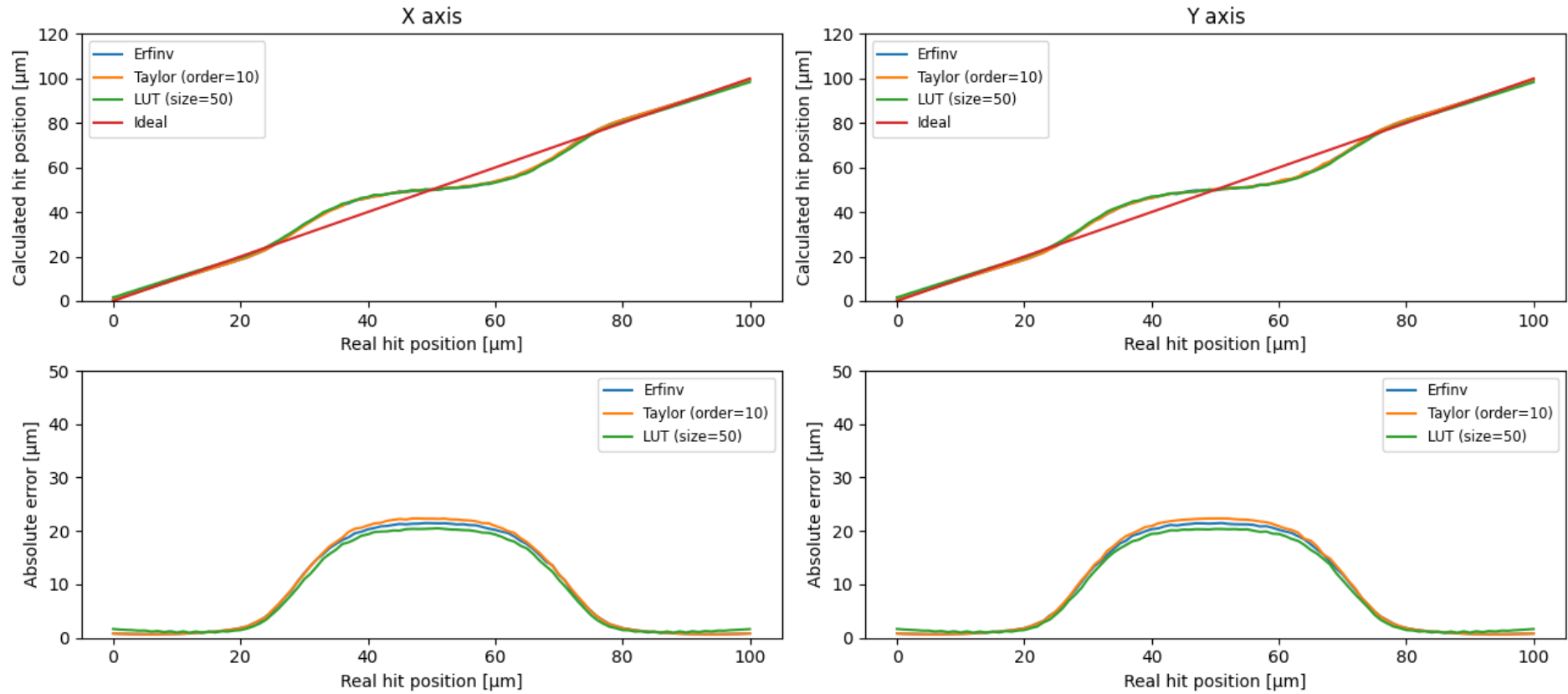


Rysunek 1. Porównanie z podziałem na osie dla detektora 1.

### Calculating methods comparison after 10000 hits

pixel size = 100 $\mu$ m, charge cloud  $\sigma = 16\mu$ m

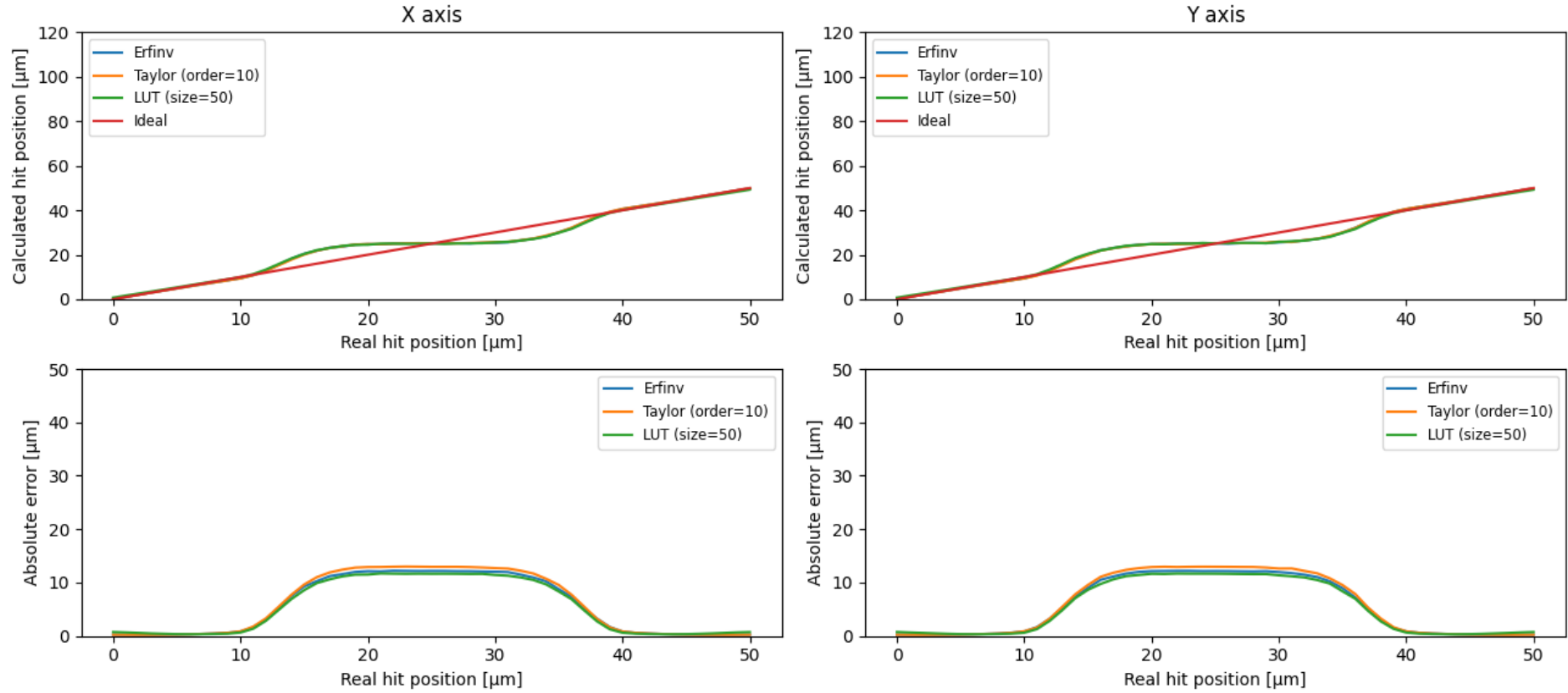
number of charges = 4970e, noise  $\sigma = 200$ e RMS



Rysunek 2. Porównanie z podziałem na osie dla detektora 2.

### Calculating methods comparison after 10000 hits

pixel size =  $50\mu\text{m}$ , charge cloud  $\sigma = 6.31\mu\text{m}$   
number of charges = 2200e, noise  $\sigma = 50\text{e RMS}$

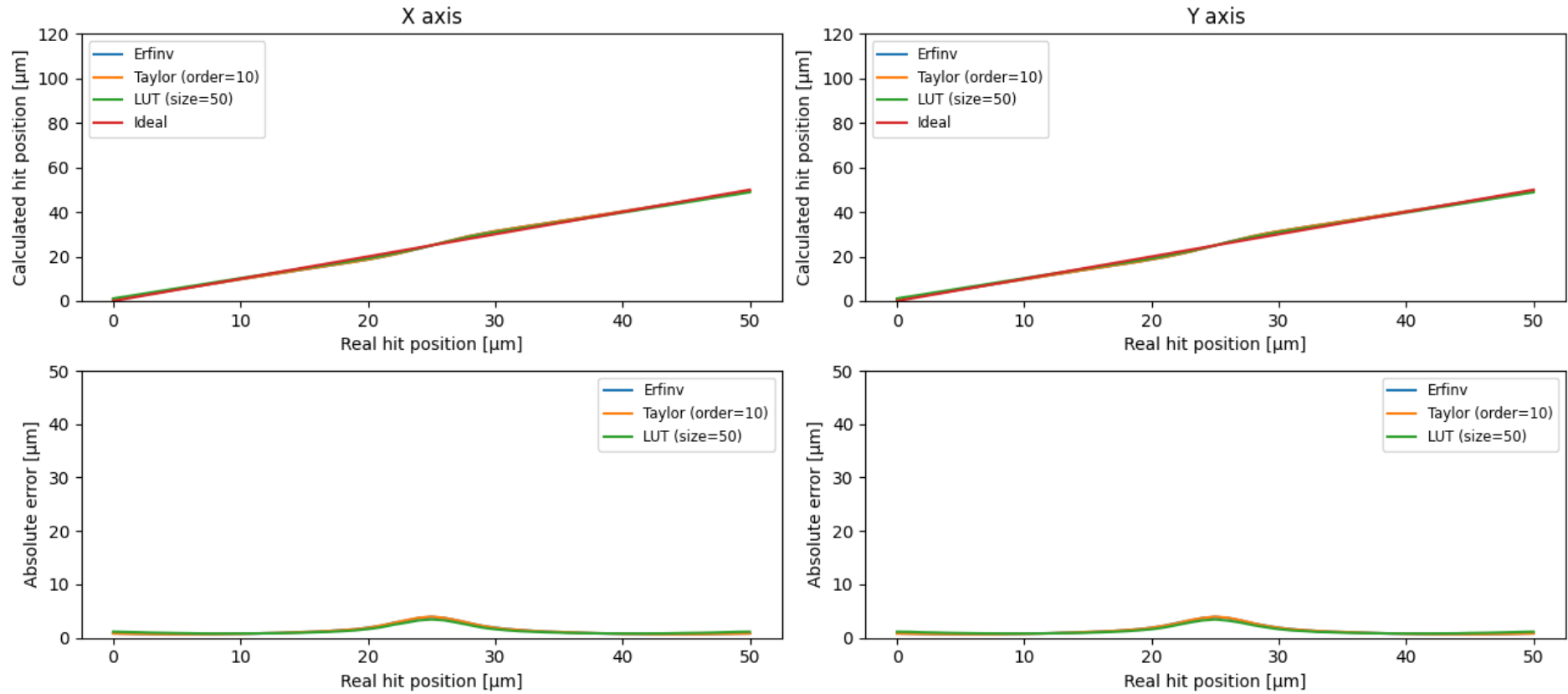


Rysunek 3. Porównanie z podziałem na osie dla detektora 3.

### Calculating methods comparison after 10000 hits

pixel size =  $50\mu\text{m}$ , charge cloud  $\sigma = 16\mu\text{m}$

number of charges = 4970e, noise  $\sigma = 200\text{e RMS}$



Rysunek 4. Porównanie z podziałem na osie dla detektora 4.

## Porównanie dwuwymiarowe

Porównanie dwuwymiarowe zostało przeprowadzone dla każdej pary  $x, y$ , gdzie  $x \in \langle 0, 75 \rangle [\mu m]$  oraz  $y \in \langle 0, 75 \rangle [\mu m]$

Dla każdej pozycji symulowano 100 razy uderzenie fotonu w detektor i aproksymowano pozycję uderzenia oraz liczone błąd jako długość wektora w układzie kartezjańskim według wzoru:

$$error = \sqrt{(r_x - c_x)^2 + (r_y - c_y)^2},$$

gdzie:

$r$  – wektor idealnej pozycji uderzenia w detektor,

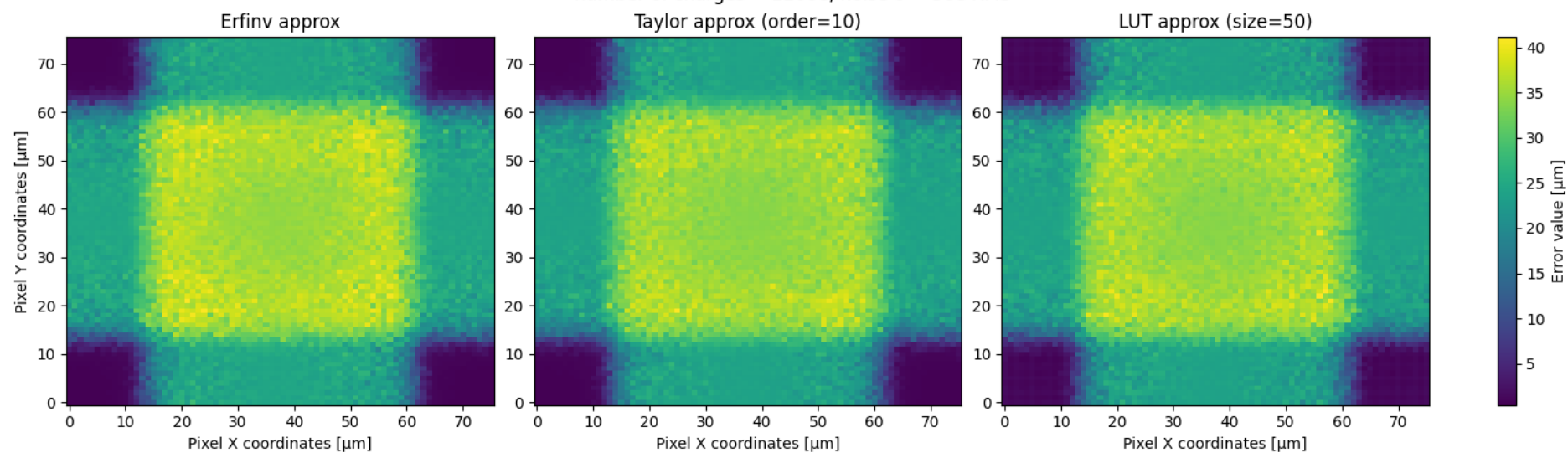
$c$  – wektor obliczonej pozycji uderzenia w detektor.

Następnie wszystkie dane uśredniono, a wyniki przedstawiono na rysunkach od 5 do 8.

### Cartesian errors of hit calculations after 100 hits

pixel size =  $75\mu\text{m}$ , charge cloud  $\sigma = 6.31\mu\text{m}$

number of charges = 2200e, noise  $\sigma = 50\text{e RMS}$

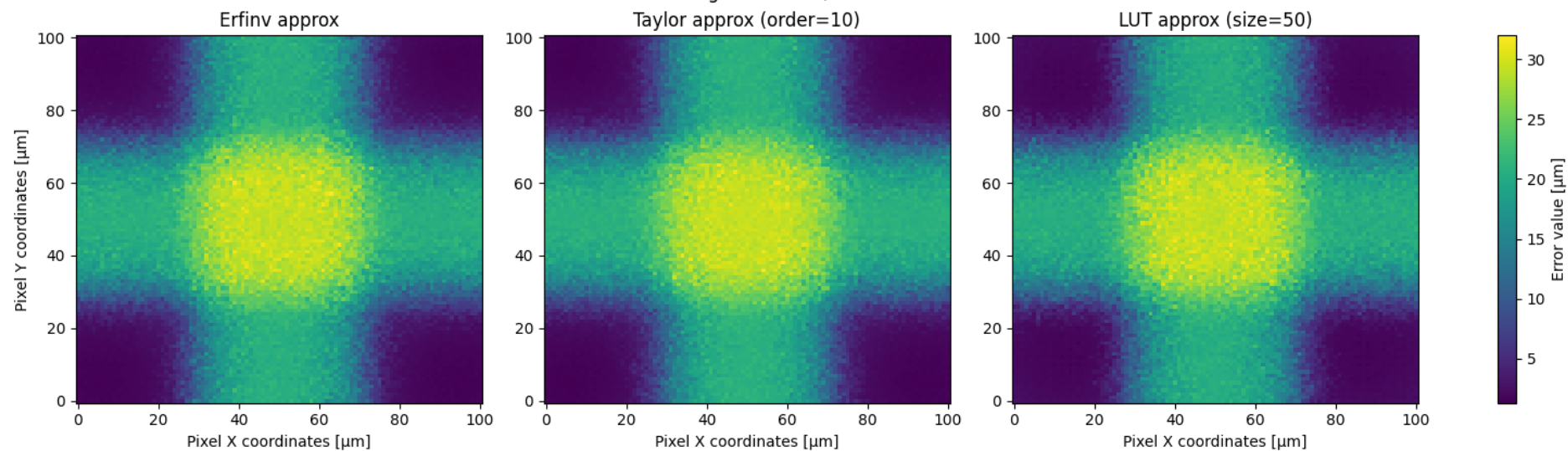


Rysunek 5. Porównanie dwuwymiarowe dla detektora 1.

### Cartesian errors of hit calculations after 100 hits

pixel size =  $100\mu\text{m}$ , charge cloud  $\sigma = 16\mu\text{m}$

number of charges = 4970e, noise  $\sigma = 200\text{e RMS}$



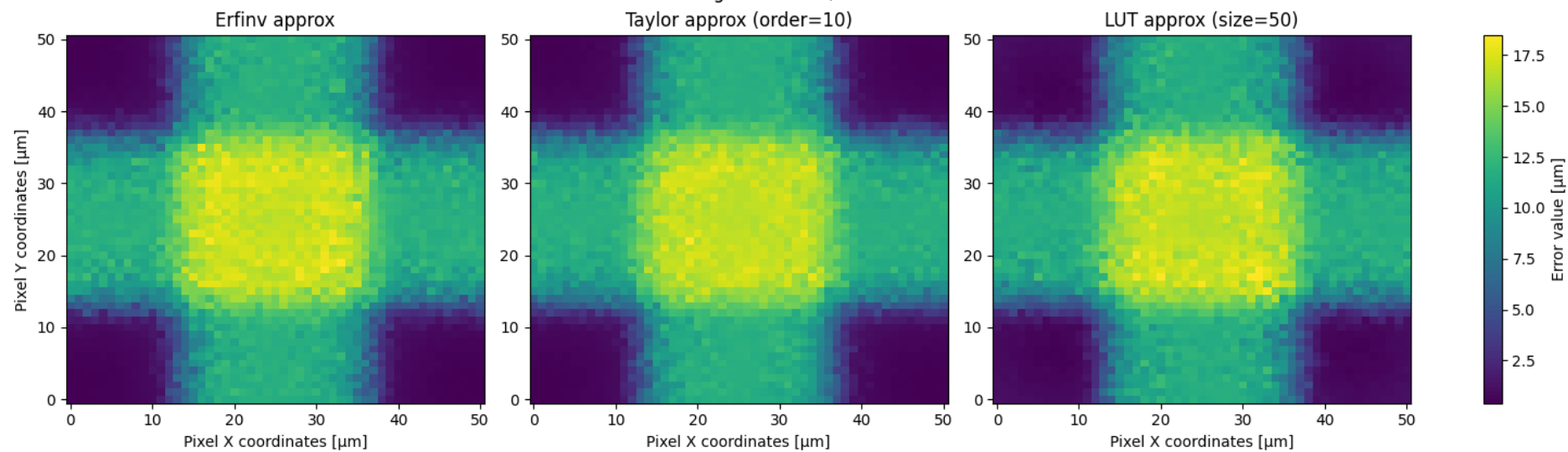
Rysunek 6. Porównanie dwuwymiarowe dla detektora 2.



### Cartesian errors of hit calculations after 100 hits

pixel size =  $50\mu\text{m}$ , charge cloud  $\sigma = 6.31\mu\text{m}$

number of charges = 2200e, noise  $\sigma = 50\text{e RMS}$

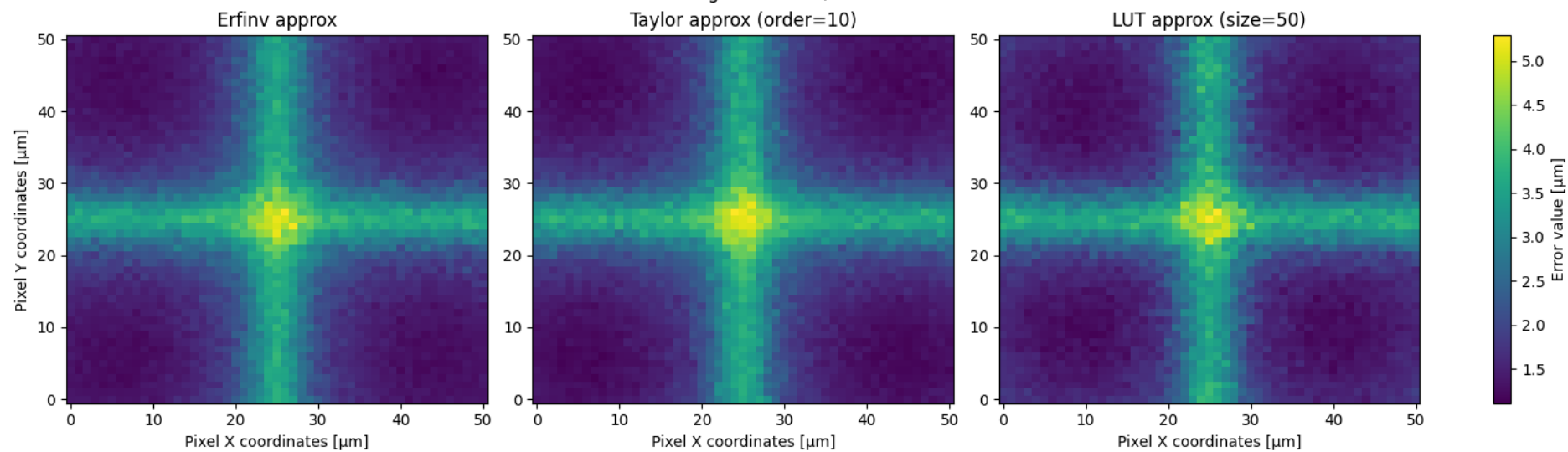


Rysunek 7. Porównanie dwuwymiarowe dla detektora 3.

### Cartesian errors of hit calculations after 100 hits

pixel size =  $50\mu\text{m}$ , charge cloud  $\sigma = 16\mu\text{m}$

number of charges = 4970e, noise  $\sigma = 200\text{e RMS}$



Rysunek 8. Porównanie dwuwymiarowe dla detektora 4.