

Rozwiązujemy raczej metodą iteracyjną a nie dokładną:

metody „dokładne”:

- przepis dający rozwiązanie w ściśle określonej liczbie kroków
- złożoność rzędu  $N^3$  [(operacje na macierzy El. Gaussa, LU  $N^3$ )  
podstawienie –  $N^2$ ]
- operacje na macierzy – niszczą jej rzadką, pasmową strukturę (ogólna macierz double  $10^6$  na  $10^6$  - około 1 TB)

metody „iteracyjne”:  $x := Mx + c$  (dla układu  $Ax = b$ ,  
macierz iteracji  $M$ , różna od  $A$ )

- każda iteracja  $N^2$  operacji
- nie zmienia struktury macierzy
- problem zbieżności i strategii prowadzenia iteracji

układ równań:  $Ax=b$

metody iteracyjne, postać ogólna:  $x:=Mx+c$

konstrukcja  $M$ : dokładne rozwiązanie układu musi spełniać  
przepis iteracyjny  $x=Mx+c$   
dobrze gdy  $M$  rzadka (dla Jakobiego jest, ale dla GS – nie)

równanie własne  $Mv_l = \lambda_l v_l$

metody iteracyjne zbieżne wtedy i tylko wtedy, gdy promień  
spektralny macierzy iteracji  $M$  [największy moduł wartości własnej]

$$\rho(M) < 1$$

metody iteracyjne zbieżne wtedy i tylko wtedy, gdy promień spektralny macierzy iteracji  $M$  [największy moduł wartości własnej]

$$\rho(M) < 1$$

uzasadnienie:

$$x^k = x + e^k$$

wektor w  $k$ -tej iteracji      błąd w  $k$ -tej iteracji

dokładne rozwiązanie  $Ax=b$  oraz  $x=Mx+c$

$$x^{k+1} = x + e^{k+1} = M(x + e^k) + c$$

$$e^{k+1} = Me^k$$

$$e^{k+1} = M^{k+1}e^0$$

$$Mv_l = \lambda_l v_l$$

wektory własne = użyjemy jako bazy

$$e^0 = \sum_l c_l v_l$$

$$e^{k+1} = \sum_l c_l \lambda_l^{k+1} v_l$$

index      potęga

błąd znika do zera z  $k$  – nieskończonym wtedy i tylko wtedy, gdy całe widmo mniejsze co do modułu od jedynki

## Jak budujemy macierz iteracji?

tak, żeby

1) dokładne rozwiązanie  $x$  spełniało:  $Ax=b$  oraz  $x=Mx+c$

2)  $\rho(M) < 1$  i tak małe jak to możliwe

asymptotyczny wskaźnik (tempo) zbieżności:  $R_\infty = -\log_{10}[\rho(M)]$

(rate of convergence)

konstrukcja iteracji tak aby  $Ax=b$ ,  $x=Mx+c$

$$A=B+C$$

$$(B+C)x=b$$

$$Bx=b-Cx \quad (B - \text{musi nieosobliwa})$$

$$x := -B^{-1}Cx + B^{-1}b$$

$$M = -B^{-1}C, \quad c = B^{-1}b$$

metoda Jakobiego:  $B=D$ ,  $C=(L+U)$

Metoda Jakobiego dla równań  
produkowanych przez 1D Poissona:

$$A = \begin{pmatrix} -2 & 1 & 0 & 0 & 0 & 0 & \dots & 0 \\ 1 & -2 & 1 & 0 & 0 & 0 & \dots & 0 \\ 0 & 1 & -2 & 1 & 0 & 0 & \dots & 0 \\ & & & \dots & & & & \\ 0 & \dots & 0 & 1 & -2 & 1 & 0 & 0 \\ 0 & 0 & \dots & 0 & 1 & -2 & 1 & 0 \\ 0 & 0 & 0 & \dots & 0 & 1 & -2 & 1 \\ 0 & 0 & 0 & \dots & 0 & 0 & 1 & -2 \end{pmatrix}$$

$$D^{-1} = \begin{pmatrix} -1/2 & 0 & 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & -1/2 & 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & -1/2 & 0 & 0 & 0 & \dots & 0 \\ & & & \dots & & & & \\ 0 & \dots & 0 & 0 & -1/2 & 0 & 0 & 0 \\ 0 & 0 & \dots & 0 & 0 & -1/2 & 0 & 0 \\ 0 & 0 & 0 & \dots & 0 & 0 & -1/2 & 0 \\ 0 & 0 & 0 & \dots & 0 & 0 & 0 & -1/2 \end{pmatrix}$$

$$L+U = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & \dots & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & \dots & 0 \\ & & & \dots & & & & \\ 0 & \dots & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & \dots & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & \dots & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & \dots & 0 & 0 & 1 & 0 \end{pmatrix}$$

$$x := -B^{-1}Cx + B^{-1}b$$

wybór Jakobiego:  $B=D$ ,  $C=(L+U)$ ,

$$D^{-1} = \begin{pmatrix} -1/2 & 0 & 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & -1/2 & 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & -1/2 & 0 & 0 & 0 & \dots & 0 \\ & & & \dots & & & & \\ 0 & \dots & 0 & 0 & -1/2 & 0 & 0 & 0 \\ 0 & 0 & \dots & 0 & 0 & -1/2 & 0 & 0 \\ 0 & 0 & 0 & \dots & 0 & 0 & -1/2 & 0 \\ 0 & 0 & 0 & \dots & 0 & 0 & 0 & -1/2 \end{pmatrix} \quad L+U = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & \dots & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & \dots & 0 \\ & & & \dots & & & & \\ 0 & \dots & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & \dots & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & \dots & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & \dots & 0 & 0 & 1 & 0 \end{pmatrix}$$

$$x := -D^{-1}(L+U)x + D^{-1}b$$

$$M = \begin{pmatrix} 0 & 1/2 & 0 & 0 & 0 & 0 & \dots & 0 \\ 1/2 & 0 & 1/2 & 0 & 0 & 0 & \dots & 0 \\ 0 & 1/2 & 0 & 1/2 & 0 & 0 & \dots & 0 \\ & & & \dots & & & & \\ 0 & \dots & 0 & 1/2 & 0 & 1/2 & 0 & 0 \\ 0 & 0 & \dots & 0 & 1/2 & 0 & 1/2 & 0 \\ 0 & 0 & 0 & \dots & 0 & 1/2 & 0 & 1/2 \\ 0 & 0 & 0 & \dots & 0 & 0 & 1/2 & 0 \end{pmatrix}$$

$$x_i^{k+1} = \frac{x_{i+1}^k + x_{i-1}^k}{2} + \frac{\rho_i \Delta z^2}{2}$$

W równaniu występują dwa wektory (dla iteracji k i k+1)

Metoda Jakobiego to relaksacja globalna

Czy zbieżna? (pierwsze laboratorium : tak)  
aby wykazać: widmo macierzy M.

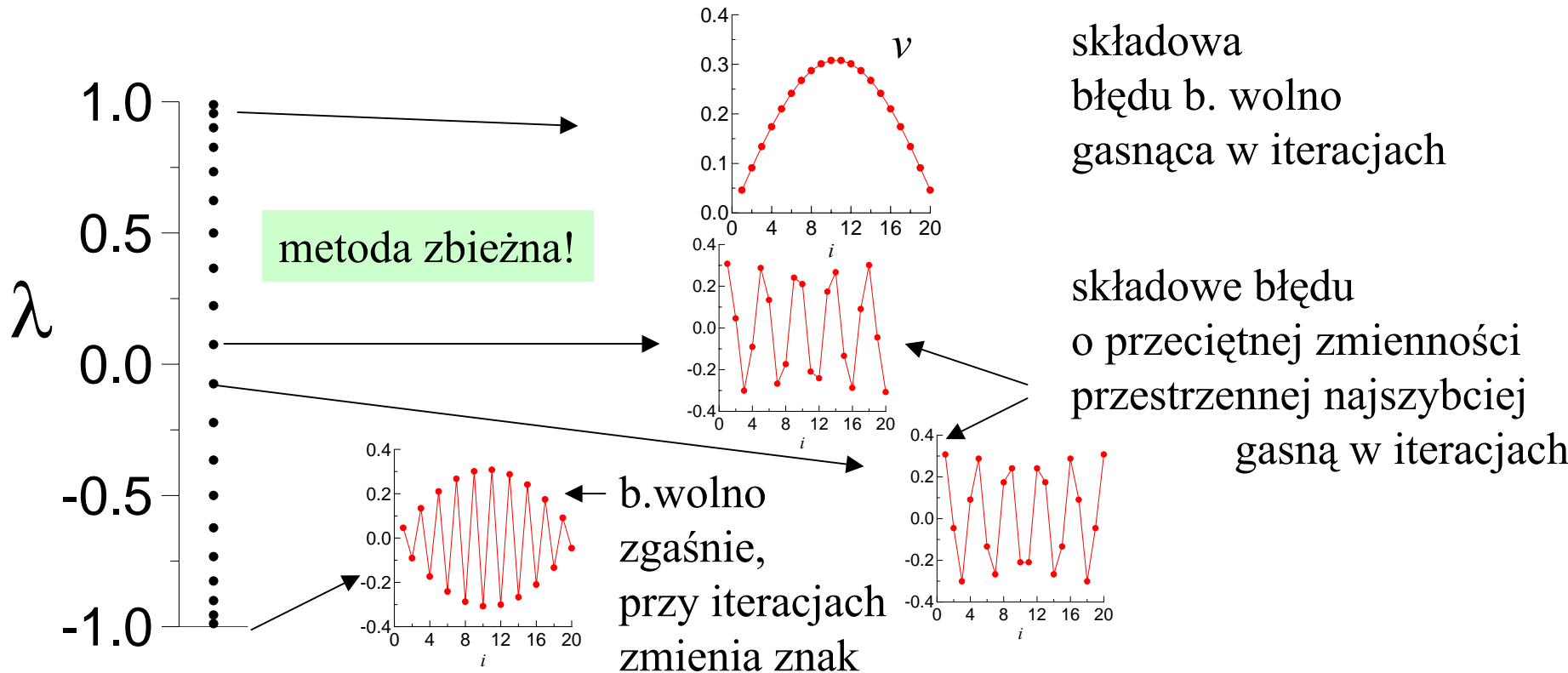
wybór Jakobiego:  $B=D$ ,  $C=(L+U)$

$$M = \begin{pmatrix} 0 & 1/2 & 0 & 0 & 0 & 0 & \dots & 0 \\ 1/2 & 0 & 1/2 & 0 & 0 & 0 & \dots & 0 \\ 0 & 1/2 & 0 & 1/2 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & 1/2 & 0 & 1/2 & 0 & 0 \\ 0 & 0 & \dots & 0 & 1/2 & 0 & 1/2 & 0 \\ 0 & 0 & 0 & \dots & 0 & 1/2 & 0 & 1/2 \\ 0 & 0 & 0 & \dots & 0 & 0 & 1/2 & 0 \end{pmatrix}$$

$$M = -B^{-1}C, \quad c = B^{-1}b$$

$$Mv = \lambda v$$

widmo wartości własnych: (N=20)



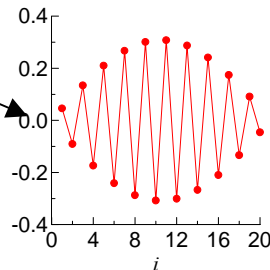
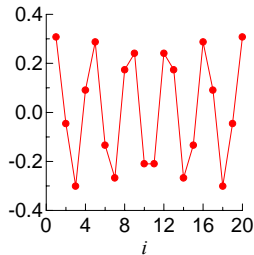
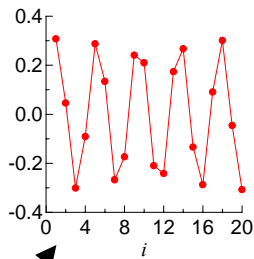
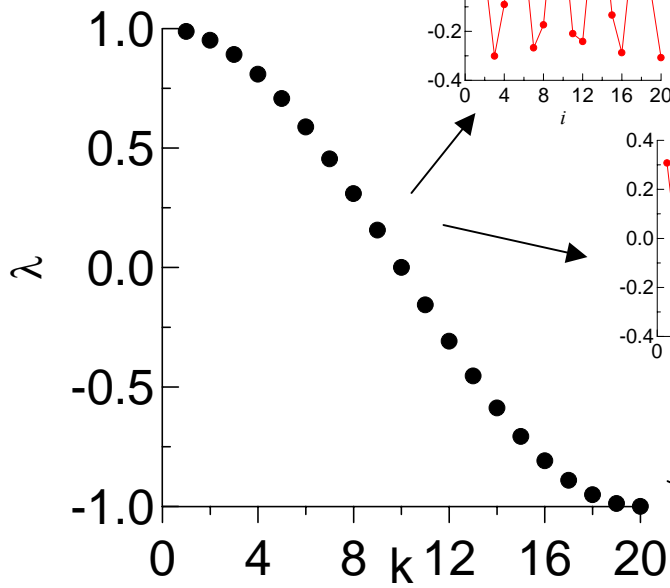
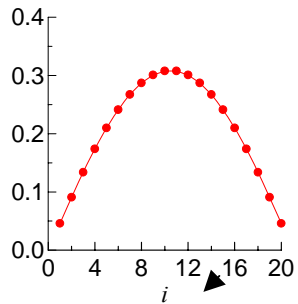
# macierz iteracji Jakobiego: formuły analityczne na wektory i wartości własne

N – liczba oczek siatki

$k=1, N$

$$\lambda_k = \cos\left(\frac{k\pi}{N+1}\right)$$

$$v_k(i) = \sin\left(\frac{ik\pi}{N+1}\right)$$



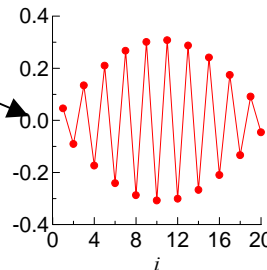
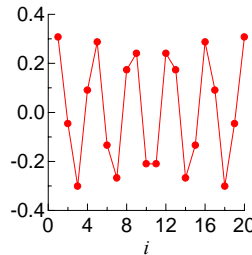
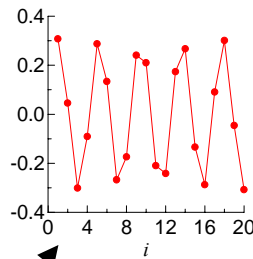
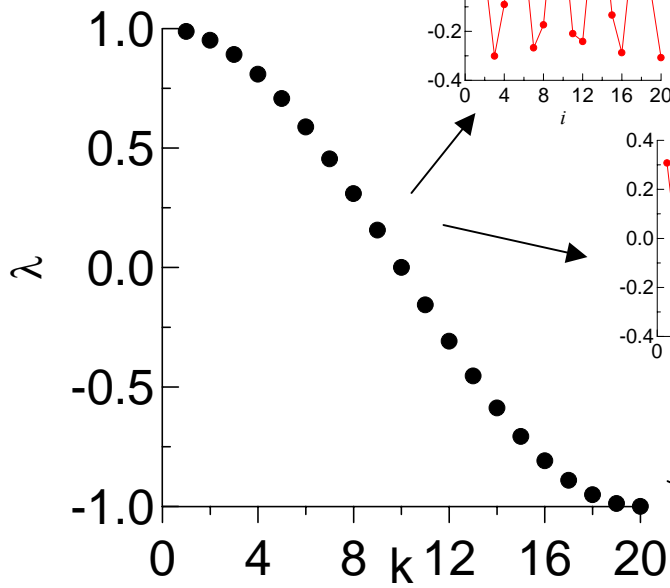
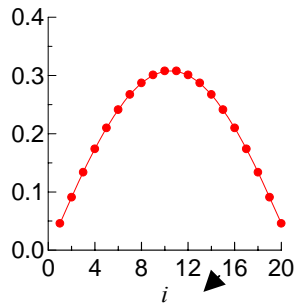
# macierz iteracji Jakobiego: formuły analityczne na wektory i wartości własne

N – liczba oczek siatki

$k=1, N$

$$\lambda_k = \cos\left(\frac{k\pi}{N+1}\right)$$

$$v_k(i) = \sin\left(\frac{ik\pi}{N+1}\right)$$



$$\rho(M) = \cos\left(\frac{\pi}{N+1}\right)$$

im gęstsza siatka, tym argument cosinusa dla  $k=1$  bliższy 0:

$$\cos\left(\frac{\pi}{N+1}\right) = 1 - \left(\frac{\pi}{N+1}\right)^2 + \dots$$

bardzo przykra wiadomość:  
iteracja tym wolniejsza  
im gęstsza siatka!  
(większa liczba punktów  $N$ )



Gauss-Seidel :

$$Ax=b \rightarrow Bx=b-Cx$$

$$B=(L+D), C=U$$

$$(L+D)x^{k+1}=b-Ux^k$$

$$\longrightarrow x^{k+1} = -D^{-1}Ux^k - D^{-1}Lx^{k+1} + D^{-1}b$$

w zastosowaniu do równania z dyskretyzacji Laplasjanu:

$$x^{k+1} = \frac{1}{2}Ux^k + \frac{1}{2}Lx^{k+1} - \frac{1}{2}b \quad (*)$$

$$\begin{pmatrix} x_1^{k+1} \\ x_2^{k+1} \\ x_3^{k+1} \\ x_4^{k+1} \\ x_5^{k+1} \\ x_6^{k+1} \end{pmatrix} = \frac{1}{2} \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} x_1^k \\ x_2^k \\ x_3^k \\ x_4^k \\ x_5^k \\ x_6^k \end{pmatrix}$$

(średnia arytmetyczna:  
z sąsiada z prawej strony  
z poprzedniej iteracji  
oraz z sąsiada z lewej strony  
z iteracji bieżącej)

równość (\*) możemy stosować  
jak przepis iteracyjny bo sąsiad  
z lewej już policzony  
(przełączamy od lewej)

$$+ \frac{1}{2} \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} x_1^{k+1} \\ x_2^{k+1} \\ x_3^{k+1} \\ x_4^{k+1} \\ x_5^{k+1} \\ x_6^{k+1} \end{pmatrix} + \frac{\Delta z^2}{2} \begin{pmatrix} \rho_1 \\ \rho_2 \\ \rho_3 \\ \rho_4 \\ \rho_5 \\ \rho_6 \end{pmatrix}$$

Jacobi:

$$x_i^{k+1} = \frac{x_{i+1}^k + x_{i-1}^k}{2} + \frac{\rho_i \Delta z^2}{2}$$

Jacobi: (relaksacja globalna całego potencjału)  
for i=1,N

$$x_n[i] = (x[i+1] + x[i-1]) / 2$$

x=xn

GS:

$$x_i^{k+1} = \frac{x_{i+1}^k + x_{i-1}^{k+1}}{2} + \frac{\rho_i \Delta z^2}{2}$$

GS: (relaksacja punktowa)

for i=1,N

$$x[i] = (x[i+1] + x[i-1]) / 2$$

mniej pamięci wymaga,  
Zobaczmy na laboratorium, że jest również szybsza  
spójrzmy na rozwiązanie problemu własnego  
dla macierzy iteracji

$A=B+C$  ( $B$  – musi nieosobliwa)

$$(B+C)x=b$$

$$Bx=b-Cx$$

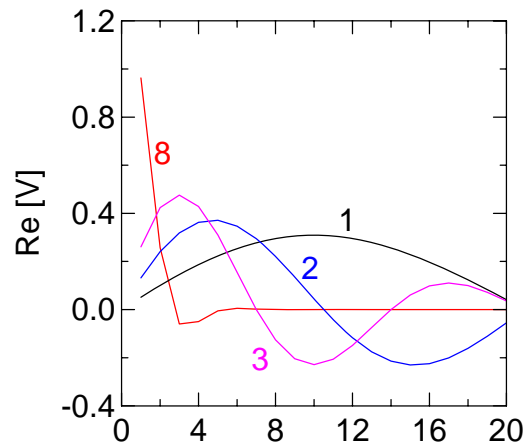
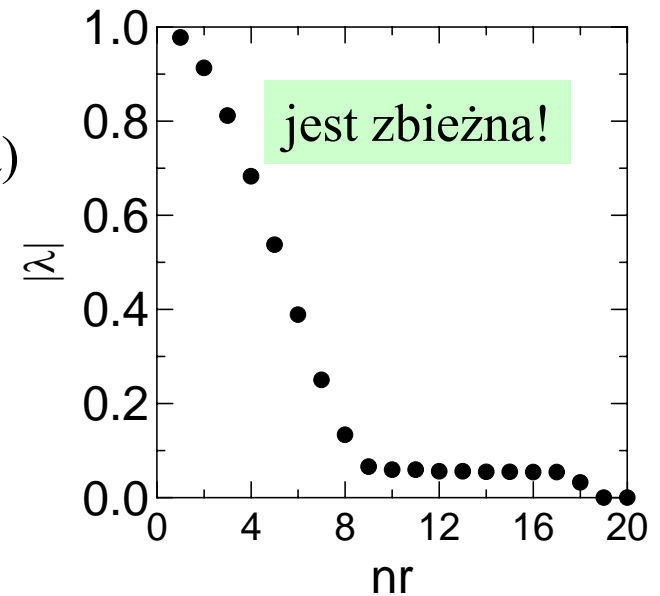
$$x := -B^{-1}Cx + B^{-1}b$$

$$M = -B^{-1}C, c = B^{-1}b$$

w metodzie GS faktycznie  
M się stosuje już rozbitą na składowe  
dla potrzeb analizy musimy  
ją jednak skonstruować

**Gaussa-Seidla:**  $B=(L+D)$ ,  $C=U$

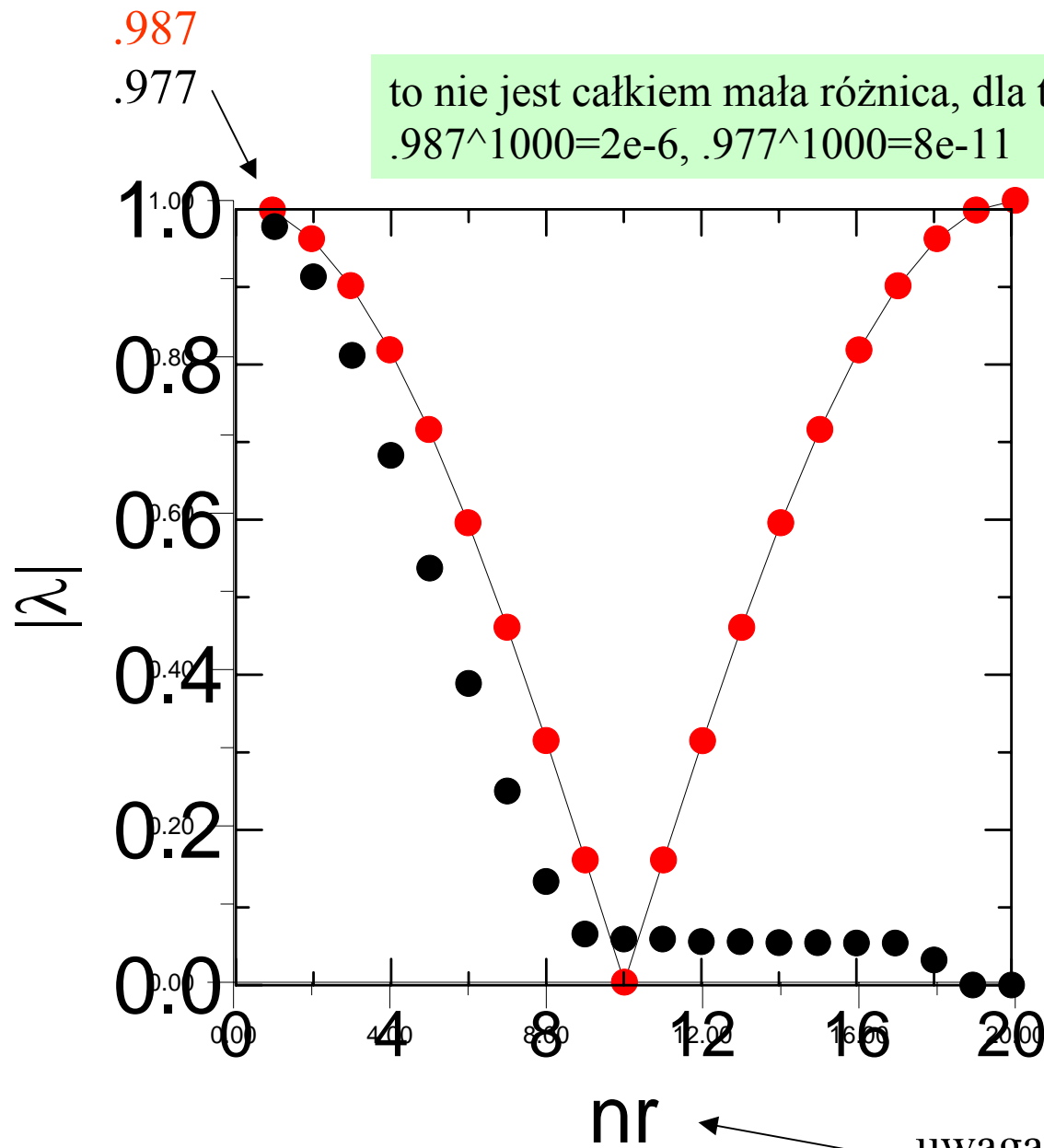
$$M = \begin{pmatrix} 0 & \frac{1}{2} & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{4} & \frac{1}{2} & 0 & 0 & 0 \\ 0 & \frac{1}{8} & \frac{1}{4} & \frac{1}{2} & 0 & 0 \\ 0 & \frac{1}{16} & \frac{1}{8} & \frac{1}{4} & \frac{1}{2} & 0 \\ 0 & \frac{1}{32} & \frac{1}{16} & \frac{1}{8} & \frac{1}{4} & \frac{1}{2} \\ 0 & \frac{1}{64} & \frac{1}{32} & \frac{1}{16} & \frac{1}{8} & \frac{1}{4} \end{pmatrix} \quad \left( \begin{array}{l} \text{dla dyskretyzacji} \\ \text{operatora Laplace'a} \end{array} \right)$$



GS – (im większa wartość własna  
tym wolniejsza zmienność  
wektora własnego)

relaksacja GS ma własność  
wygładzania błędów (*error smoother*)

# metoda Jakobiego vs relaksacja Gaussa Seidla



uwaga:

wektory własne: różne dla J i GS

Jakość „rozwiązania”:  $Ax=b$

pozostałość (residuum):  $r=Ax-b$

przykład numeryczny

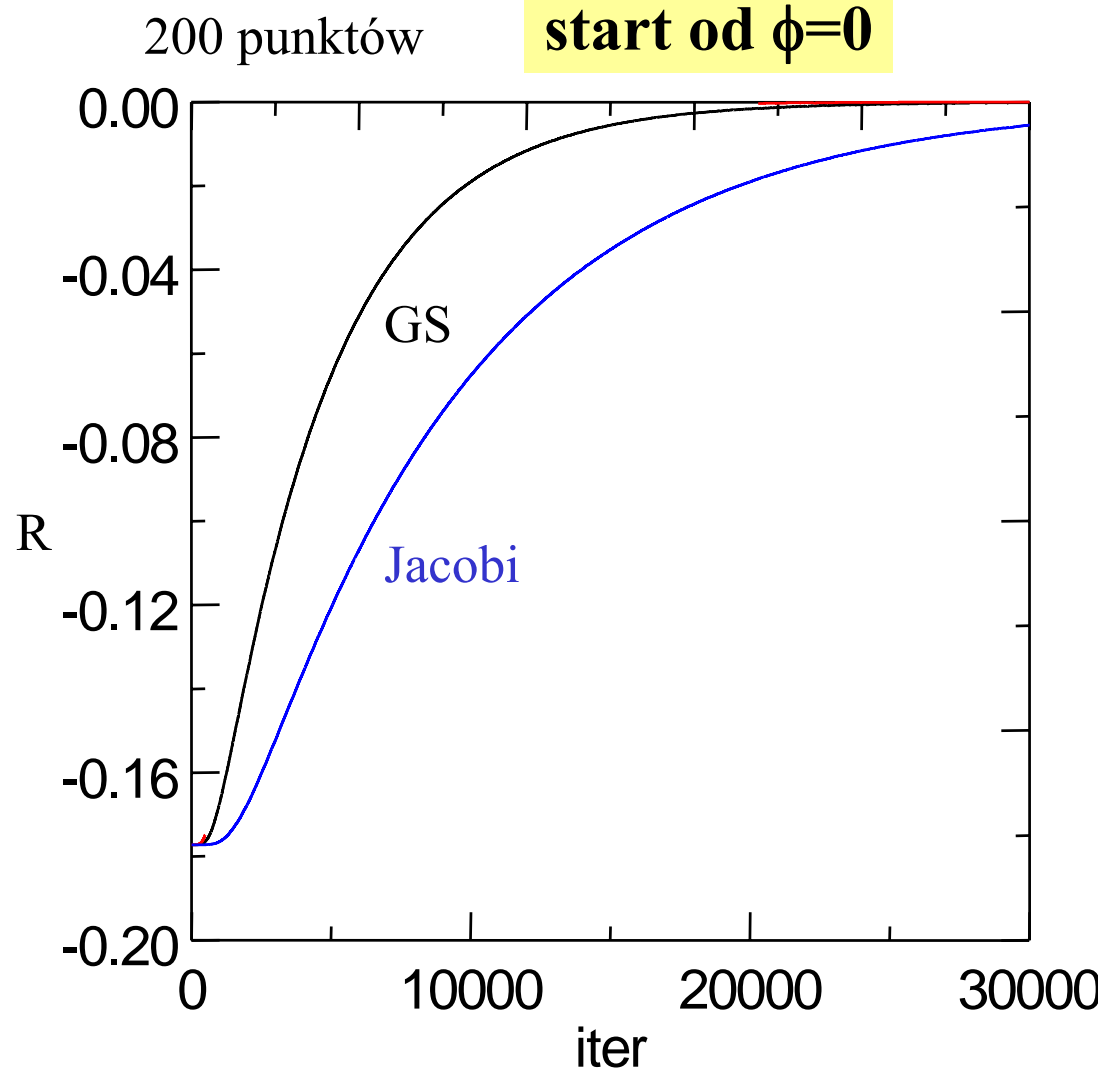
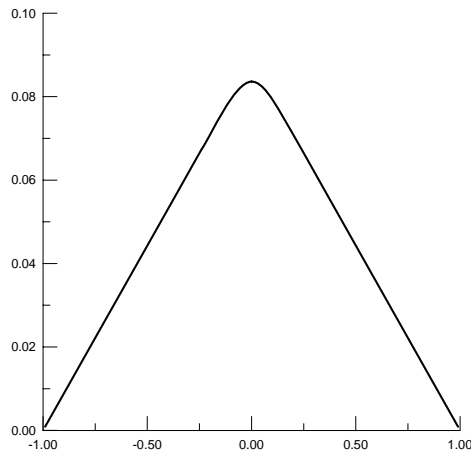
problem modelowy:

$$\nabla^2 \phi = -\rho$$

$$\phi(-1)=\phi(1)=0$$

$$\rho = \exp(-100 x^2)$$

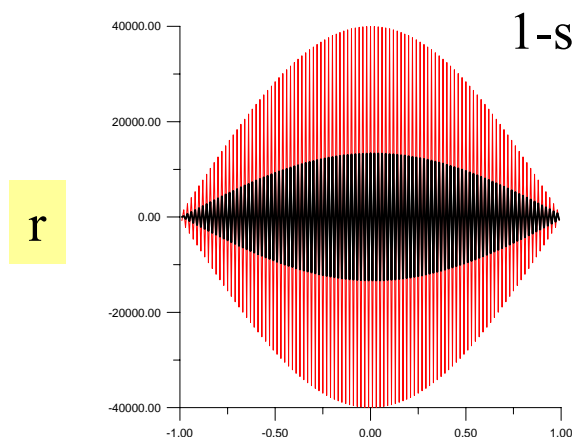
$$R = \text{suma } r_i$$



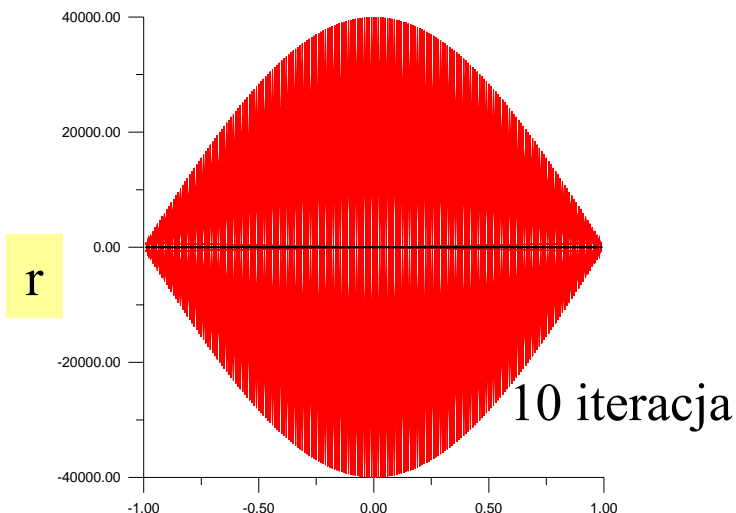
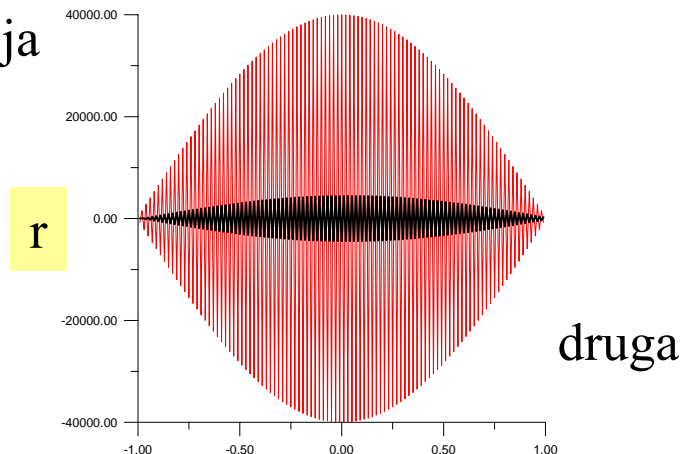
GS vs **Jacobi**: start od wektora *najkrótszej „długości fali”*

$$v_k(i) = \sin\left(\frac{ik\pi}{N+1}\right)$$

$k=N$



200 punktów

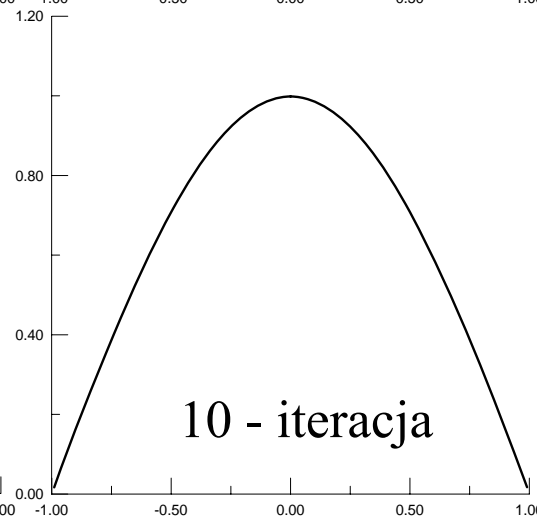
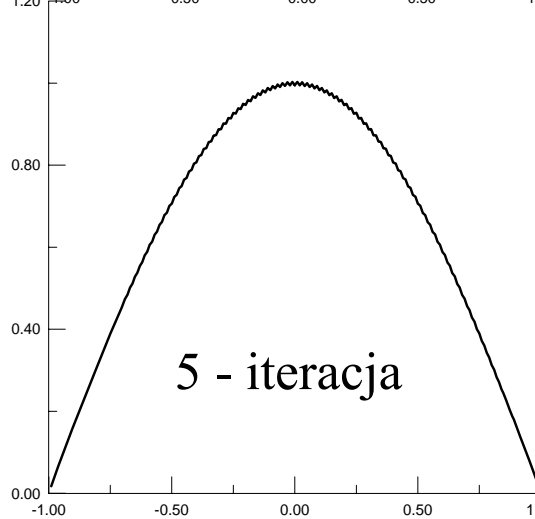
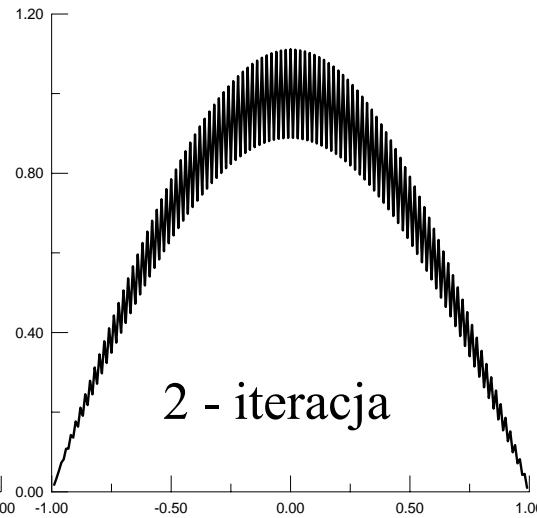
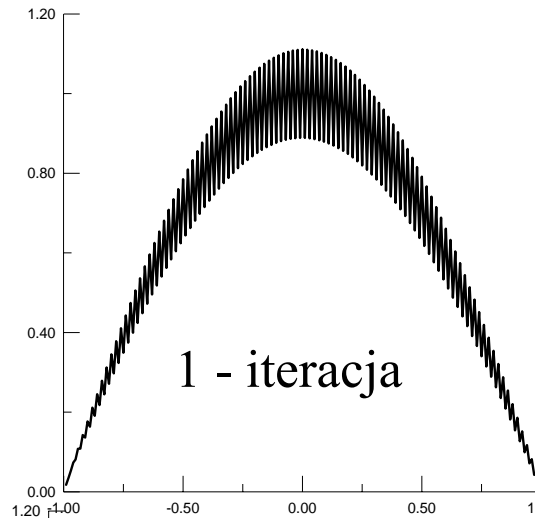


GS: szybkozmienny błąd  
szybko tłum  
*smoother*

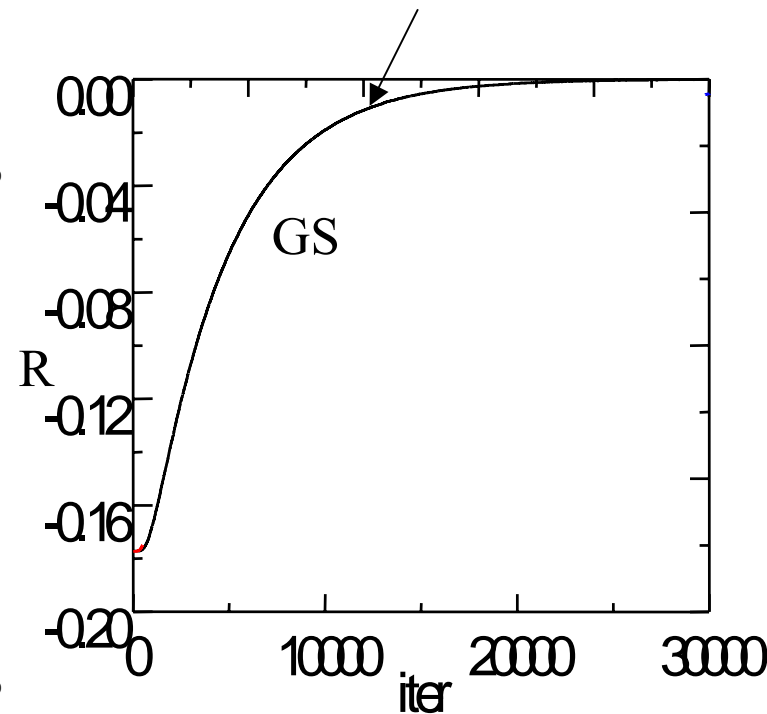
*Jacobi: wolno gasi  
zarówno najszybsze  
jak i najwolniejsze  
błędy*

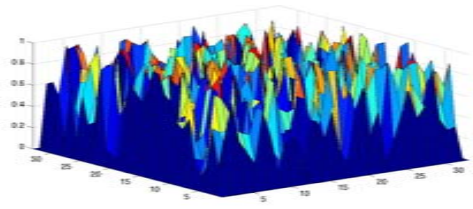
GS - smoothener: start od superpozycji wektorów z  $k=1$  i  $k=N$

$$v_k(i) = \sin\left(\frac{ik\pi}{N+1}\right)$$

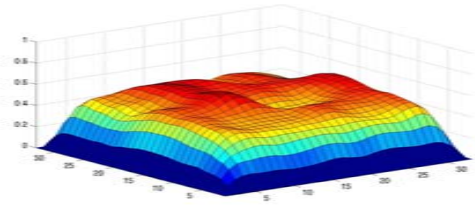


gdy relaksacja zwalnia błąd jest superpozycją wektorów własnych o niskich wartościach własnych

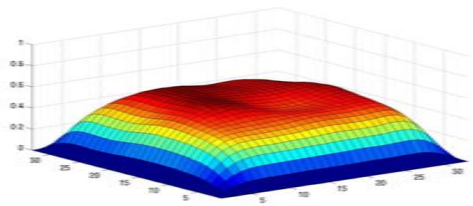




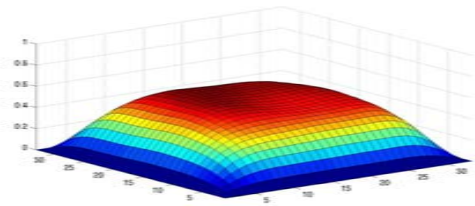
**Error of initial guess**



**Error after 5 relaxation  
sweeps**



**Error after 10 relaxations**



**Error after 15 relaxations**

**Fast error smoothing**  
**slow solution**

Achi Brandt

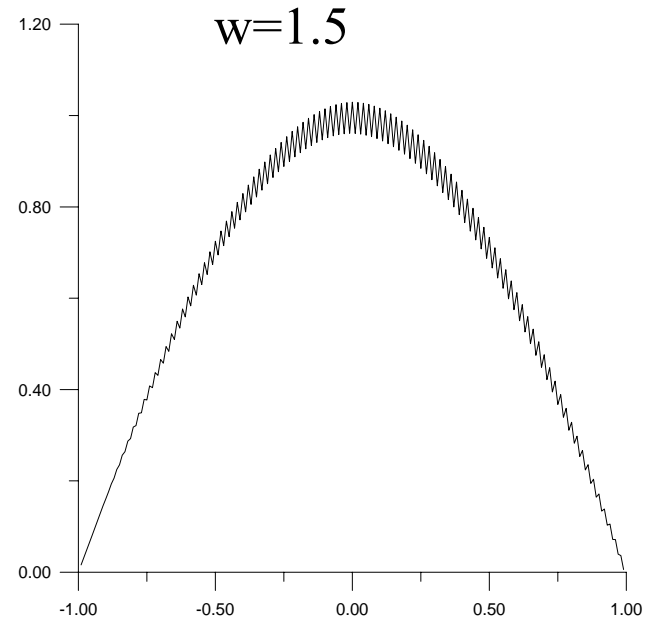
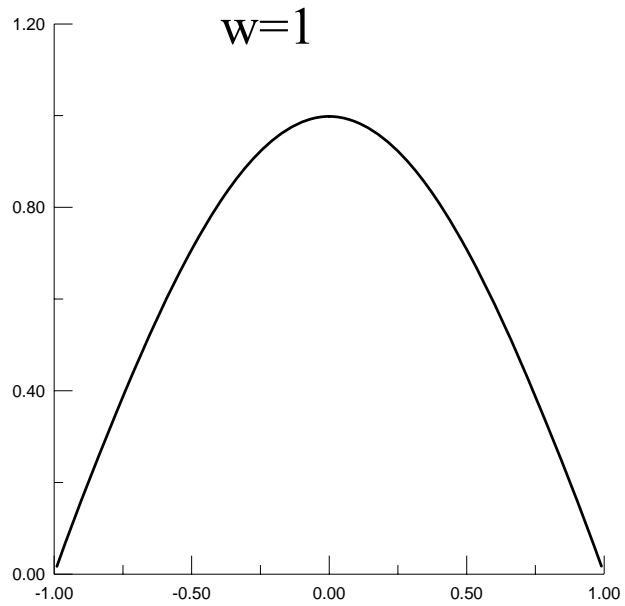
The Weizmann Institute of Science

UCLA

[www.wisdom.weizmann.ac.il/~achi](http://www.wisdom.weizmann.ac.il/~achi)



## 10 iteracja, relaksacja a nadrelaksacja i gaszenie błędów



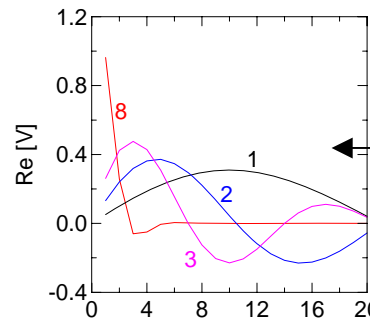
nadrelaksacja nie wygładza błędów

GS i J: im gęstsza siatka tym wynik dokładniejszy  
ale iteracja bardziej kosztowna ( $N^2$ ), co gorzej - wolniej zbieżna

w GS - metodzie wygładzającej błąd szybkozmienny znika najszybciej

### Metoda zagęszczania siatki (najprostsza wielosiatkowa):

Rozwiązanie najpierw na rzadkiej: eliminacja błędu wolnozmiennego, który można opisać na siatce rzadkiej.



wek.w 1 na siatce  
rzadkiej opisujemy  
bez problemu

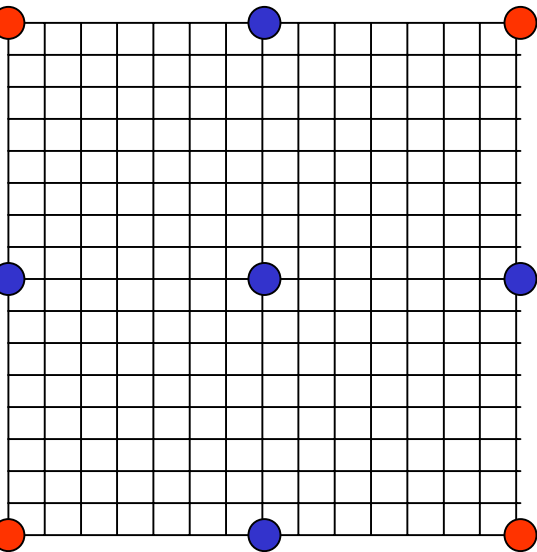
Rozwiązanie przepisane na gęstszą jako punkt startowy do nowego rachunku:  
przy starcie ujawni się błąd szybkozmienny z przepisania. Możemy liczyć, że  
szybko zgaśnie.

## Metoda globalnego zagęszczania siatki:

Najpierw rozwiązać „tanie” równanie na rzadkiej siatce, dopiero następnie na gęstszej unikamy pojawienia się wolnozmiennego i wolnozbieżnego błędu na najgęstszej siatce

### laboratorium

Przykład 2D: siatka ( $N$  na  $N$ ) =  $N^2$  punktów, macierz  $N^2$  na  $N^2$

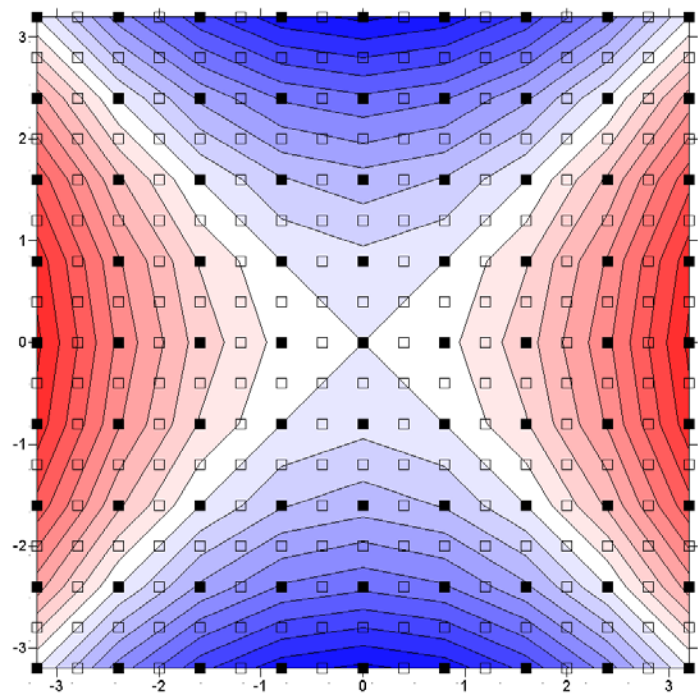


- 1) Rozwiązujemy zdyskretyzowane równanie różniczkowe na siatce  $\Delta x$  (**czzerwone punkty**) (iterujemy do zbieżności)
- 2) Oszacowujemy wartość funkcji na punktach rozłożonych na siatce o skoku  $\Delta x/2$  (**nowe-niebieskie punkty**)
- 3) Rozwiązujemy równanie na nowej siatce (**czzerwone+niebieskie**) (iterujemy do zbieżności)
- 4) itd. aż po  $N$  podziałach dojdziemy do  $\Delta x/2^N$

Im więcej wymiarów, tym strategia bardziej użyteczna.

## laboratorium

rozwiązanie na siatce  $9 \times 9$  (czarne kwadraty)  
i nowa siatka  $17 \times 17$  (czarne+puste kwadraty)

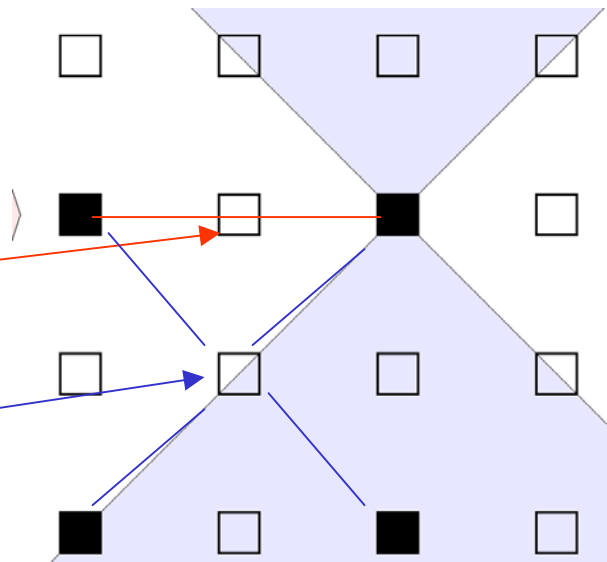


Start do iteracji na gęstszej siatce:

Punkty stare – zostawiamy bez zmian

Nowe punkty między dwoma starymi  
średnia arytmetyczna z  
dwóch czarnych sąsiadów

Pozostałe nowe – średnia z czterech  
sąsiadów leżących w narożnikach  
kwadratu



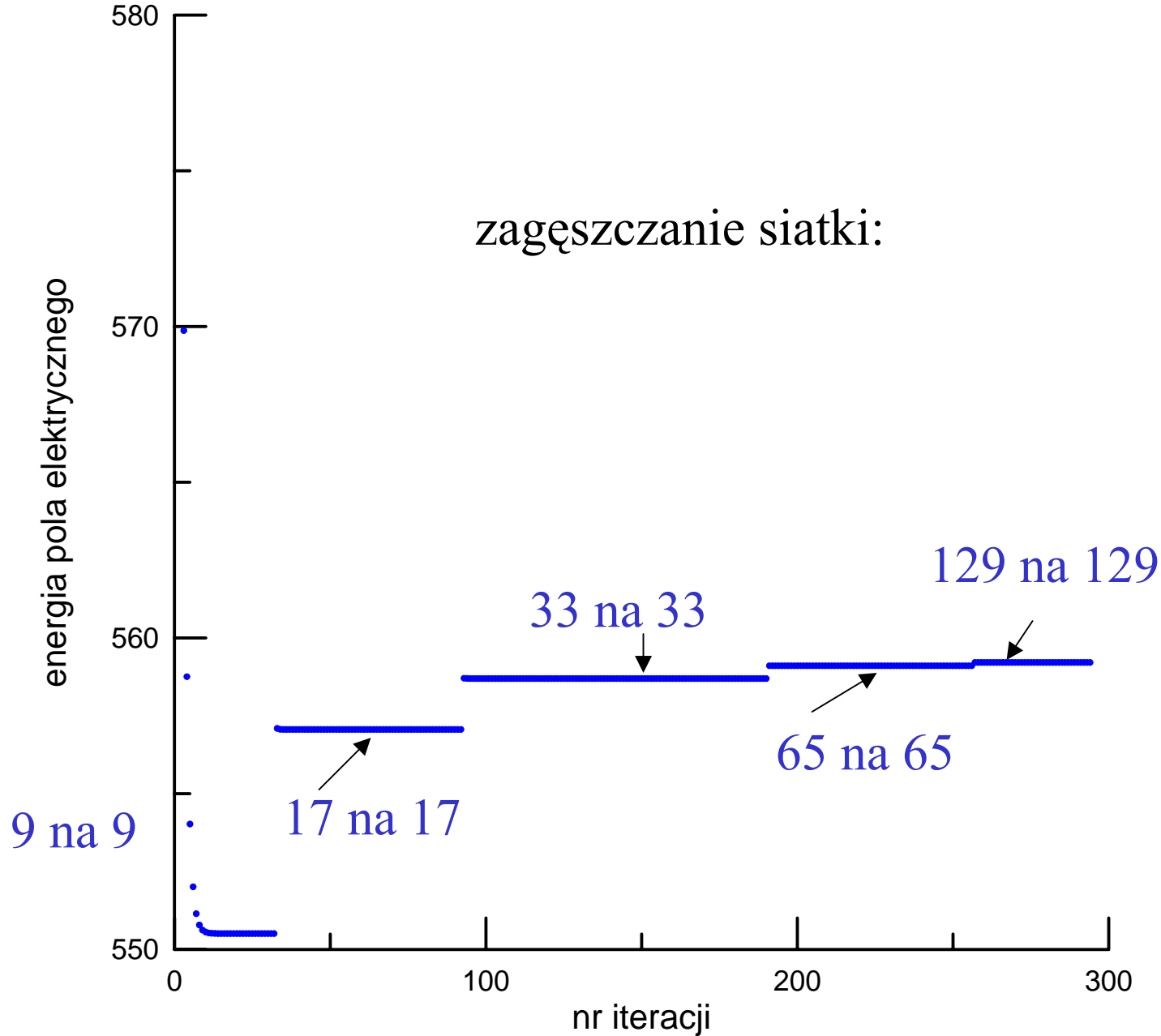




uznamy że zbieżność  
osiągnięta gdy energia  
pola z iteracji na iterację  
zmienia się mniej niż  $10^{-9}$

start na siatce docelowej

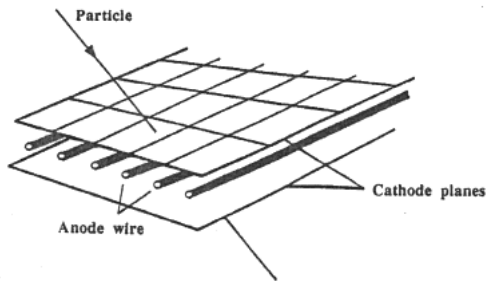
zagęszczanie siatki



na najgęstszej siatce wykonujemy około 40 iteracji  
zamiast 6500

## lokalne zagęszczanie siatki dla równania Laplace'a 2D

Komora drutowa: detektor cząstek – cienkie druty (anody) w dużym pudle (katodzie)



Co zrobić jeśli interesuje nas tylko bliskie otoczenie jednego z drutów?

Można dać bardzo gęstą siatkę. Może nie starczyć pamięci.

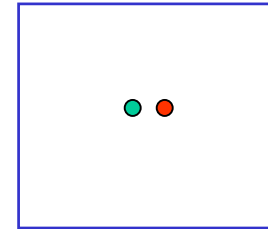
**Lepszy pomysł: wyliczyć potencjał metodą kolejnych powiększeń**

WB dla każdego powiększenia wyliczyć zgodnie z metodą zagęszczania siatki.

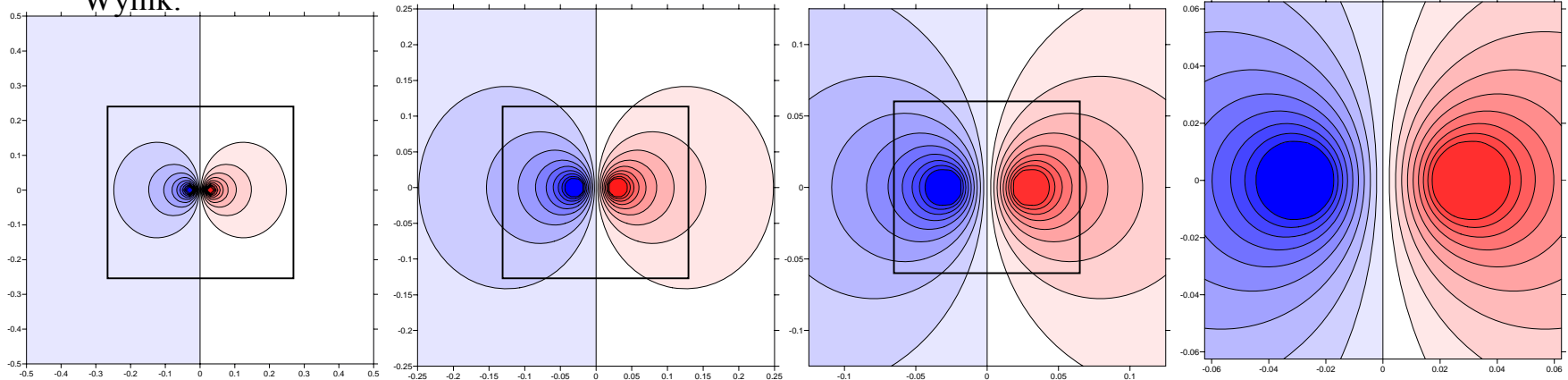
Mamy uziemione pudło metalowe o rozmiarach 1cm na 1cm.

Dwa druty o promieniu 1.2 mm w środku pudła.

Odległe o 6 mm względem siebie. Na lewy podajemy potencjał +1, na prawy +1. Wyznaczyć rozkład potencjału w środku pudła.



Wynik:



Każde powiększenie liczone na siatce 50 x 50 punktów.



## Metody wielosiatkowe - idea:

rachunki na 2 lub więcej siatkach (różnej gęstości,  $\Delta x$ ,  $2\Delta x$ )

- 1) rachunek na gęstej siatce przy pomocy iteracji wygładzającej  
= błąd szybkozmienny szybko gaśnie (wykonujemy kilka kroków,  
nie walczymy o zbieżność)
- 2) błąd wolnozmienny – rzutowany na rzadką siatkę  
(udaje się bo wolnozmienny) i eliminowany przez iterację  
na tej siatce [iterowanie do zbieżności, najlepiej nadrelaksacją]
- 3) wartości na siatce gęstej – poprawione o błąd wolnozmienny  
wyliczony na siatce rzadkiej – idziemy do (1)

## Metoda dwusiatkowa $Ax=c$ (V – cycle)

Dwie siatki, ze skokiem  $\Delta x$  i  $2\Delta x$ .

1. Iterujemy równanie  $x:=Mx+b$  na gęstej siatce  $v_1$  razy dla wygładzenia błędów.

$$\tilde{x}(\Delta x) = x(\Delta x) + e(\Delta x)$$

Diagram illustrating the error equation with arrows pointing to the terms:

- $\tilde{x}(\Delta x)$ : uzyskane przybliżenie
- $x(\Delta x)$ : wynik dokładny
- $e(\Delta x)$ : błąd

$$A\tilde{x}(\Delta x) = Ax(\Delta x) + Ae(\Delta x)$$

2. Liczymy pozostałość  $r(\Delta x) = A\tilde{x}(\Delta x) - c$

równanie na błąd  $Ae(\Delta x) = r(\Delta x)$

gdy znamy rozwiązanie  $Ae=r$ , znamy również rozwiązanie  $Ax=c$ .

**ALE:**  $Ae=r$  wystarczy rozwiązać na rzadszej siatce, bo  $e$  jest gładki

3. Rzutujemy pozostałość na siatkę rzadką

$$r(2\Delta x) = I(\Delta x \rightarrow 2\Delta x)r(\Delta x)$$

operator restrymkcji (*restriction*)

3. Rzutujemy pozostałość na siatkę  $2\Delta x$

$$r(2\Delta x) = I(\Delta x \rightarrow 2\Delta x)r(\Delta x)$$

$$Ae(2\Delta x) = r(2\Delta x) \Big| \text{rozwiązujemy (iterujemy aż do zbieżności)}$$

4. Przenosimy uzyskany błąd na siatkę gęstą

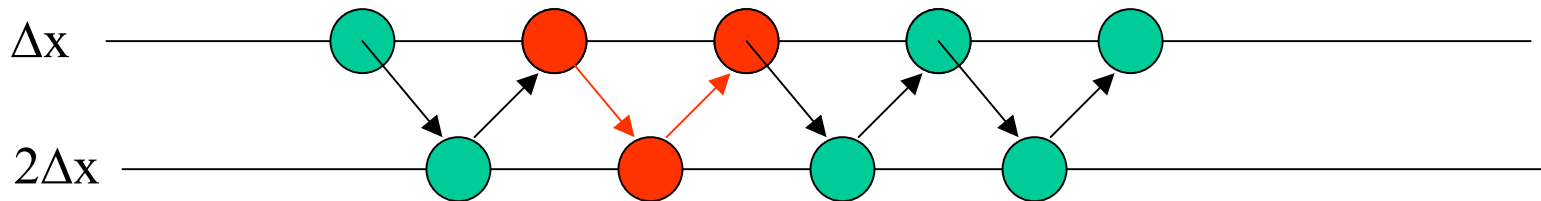
$$e(\Delta x) = I(2\Delta x \rightarrow \Delta x)e(2\Delta x)$$

operator przedłużenia (*prolongation*)

i poprawiamy rozwiązanie

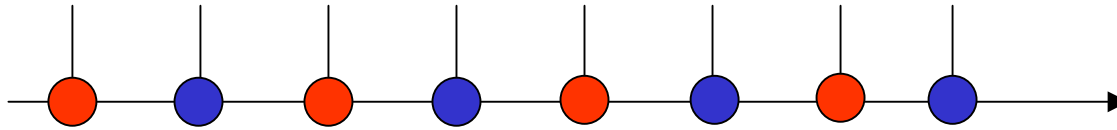
$$\tilde{x}(\Delta x) := \tilde{x}(\Delta x) - e(\Delta x) \Big|$$

5. Powyższy zabieg wprowadza nowy błąd (rzutowania  $I$  nie są dokładne)  
- iterujemy równanie  $v_2$  razy na gęstrzej siatce dla usunięcia szybkozmiennej części błędu



V-cycle

macierzowa forma najprostszych operatorów restrykcji i przedłużenia:  
 przybliżenie liniowe (równanie Laplace'a 1D)



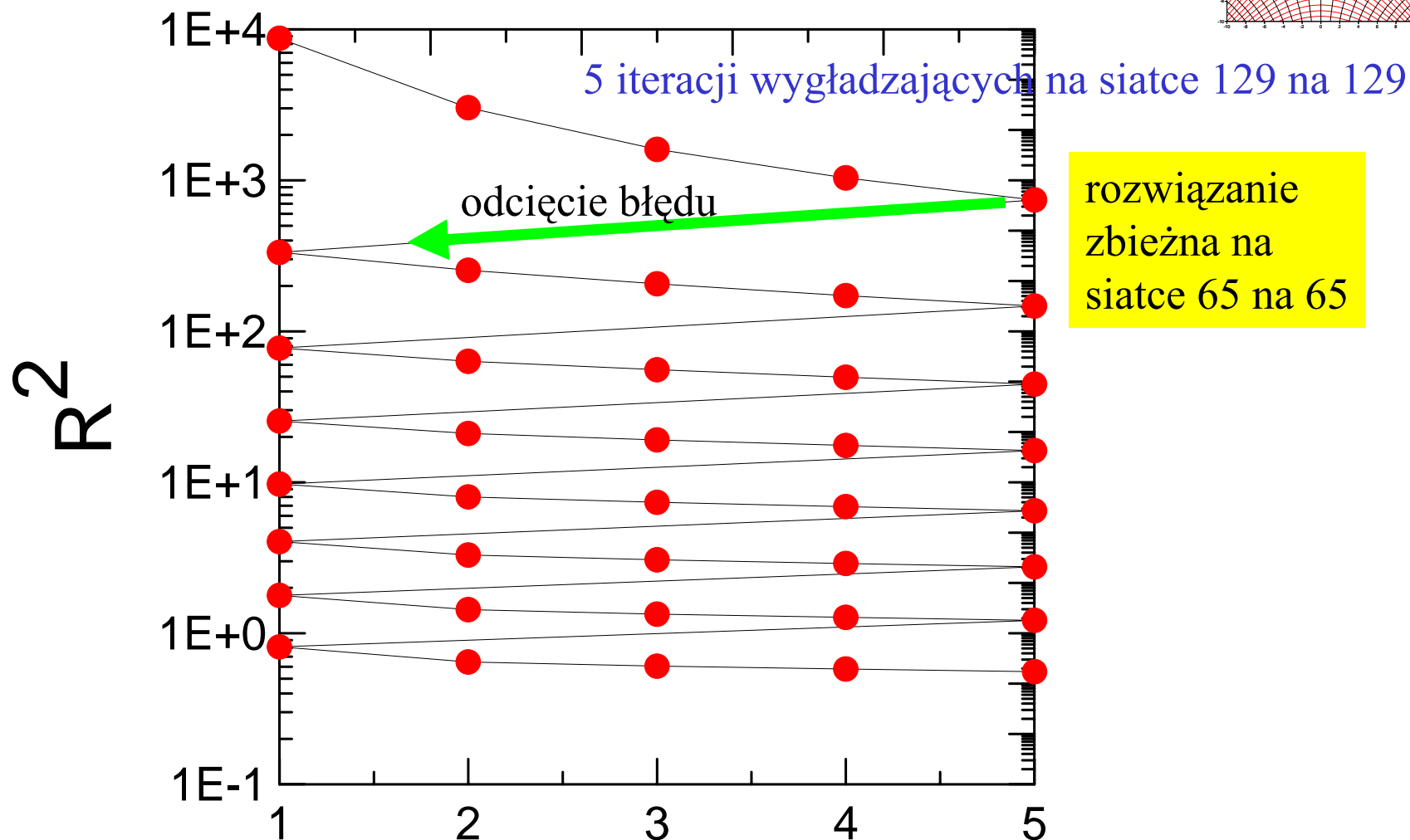
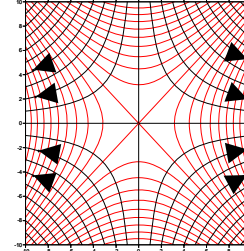
$I(\Delta x \rightarrow 2\Delta x)$  z czerwonych i niebieskich do czerwonych

$$\begin{pmatrix} x_2 \\ x_4 \\ x_6 \\ x_8 \\ x_{10} \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \\ x_{10} \end{pmatrix}$$

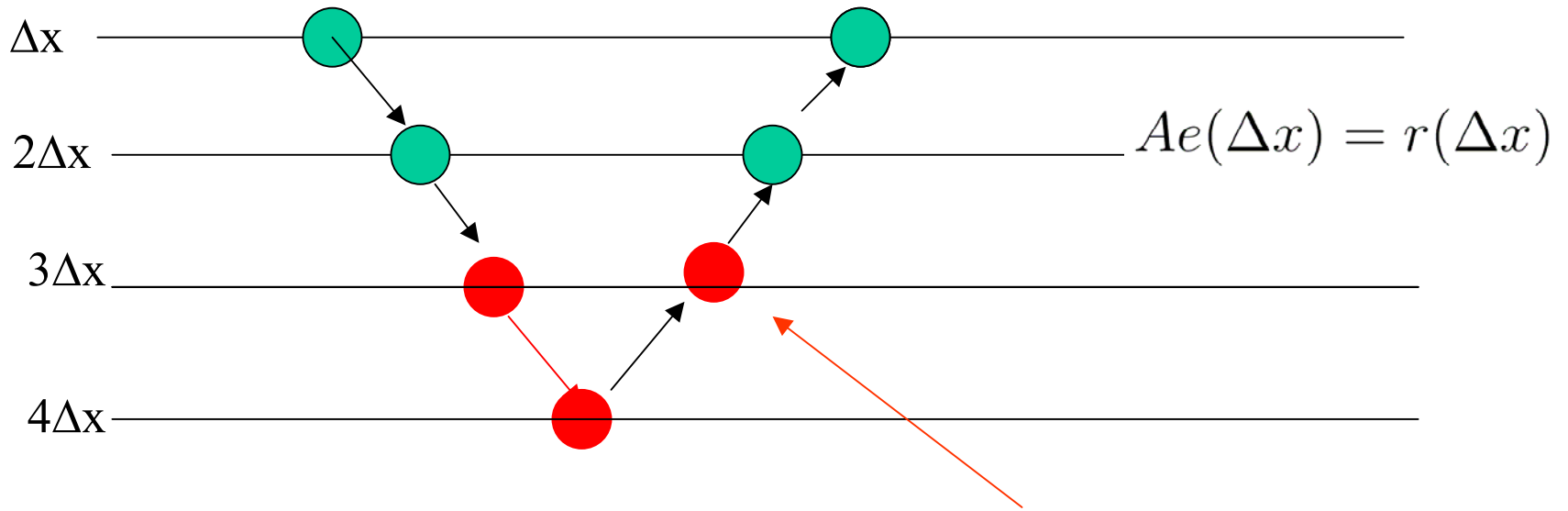
$I(2\Delta x \rightarrow \Delta x)$  z czerwonych do niebieskich i czerwonych

$$\begin{pmatrix} x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \\ x_{10} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 1/2 & 1/2 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1/2 & 1/2 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1/2 & 1/2 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1/2 & 1/2 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_2 \\ x_4 \\ x_6 \\ x_8 \\ x_{10} \end{pmatrix}$$

pozostałość w kwadracie dla iterowanego V-cyklu



## V-cycle z wielosiatkowym rozwiązaniem na siatce o skoku $2\Delta x$



Równanie na błąd (na rzadszej siatce)

ma tę samą formę co problem oryginalny –  
można je rozwiązać w ten sam sposób –  
z rachunkiem na rzadszej siatce

rachunki na siatkach  
gęstszych – po kilka  
iteracji wygładzających  
-do zbieżności relaksowana  
tylko iteracja na najrzadszej siatce

# minimalizacja działania jako metoda: zadanie 3 z projektu 1

- Zbieżność iterowanej funkcji śledzić licząc całkę z lagranżjanu układu ładunek-pole

$$a = \int_S \left[ \frac{1}{2} \left( \frac{\partial u}{\partial x} \right)^2 + \frac{1}{2} \left( \frac{\partial u}{\partial y} \right)^2 - \rho(i, j) u(i, j) \right] dx dy, \quad (8)$$

- w wersji dyskretnej

$$a = + \sum_{i,j=-N+1}^{N-1} \left[ \frac{1}{2} \left( \frac{u(i+1, j) - u(i-1, j)}{2dx} \right)^2 \right. \quad (9)$$

$$+ \left. \frac{1}{2} \left( \frac{u(i, j+1) - u(i, j-1)}{2dy} \right)^2 \right] \quad (10)$$

$$- \rho(i, j) u(i, j) dx^2 dy^2. \quad (11)$$

- Operator pochodnej jest antyhermitowski ( $f(x), \nabla g(x) = -(\nabla f(x), g(x))$ ), więc

$$a = - \int_S \left[ \frac{1}{2} \left( u \frac{\partial^2 u}{\partial x^2} \right) + \frac{1}{2} \left( u \frac{\partial^2 u}{\partial y^2} \right) + \rho(i, j) u(i, j) \right] dx dy, \quad (12)$$

- w wersji dyskretnej

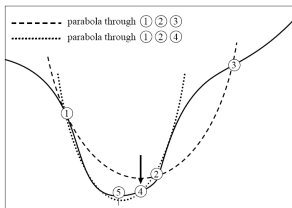
$$a = - \sum_{i,j=-N+1}^{N-1} \left[ \frac{1}{2} u(i, j) \frac{u(i+1, j) + u(i-1, j) - 2u(i, j)}{dx^2} \right] \quad (13)$$

$$+ \frac{1}{2} u(i, j) \frac{u(i, j+1) + u(i, j-1) - 2u(i, j)}{dy^2} \quad (14)$$

$$+ \rho(i, j) u(i, j) dx^2 dy^2. \quad (15)$$

# metoda parabol

- Iteracja, w której będziemy starać się poprawić wartość  $u(i, j)$  w każdym punkcie w pudle poza brzegiem.
- dla  $(i, j)$ 
  - 1 Wyznamy wartość  $a(\delta)$  w zależności od zmiany wartości potencjału w punkcie  $(i, j)$  z  $u(i, j)$  na  $u(i, j) + \delta$ .
  - 2  $a_1 = a(\delta_1 = 0)$
  - 3  $a_2 = a(\delta_2 = 0.5)$
  - 4  $a_3 = a(\delta_3 = 1)$
  - 5 Poprowadzimy parabolę przez te 3 punkty i wyznaczmy  $\delta$  dla którego parabola ma ekstremum  $\delta_4 = \frac{1}{4} \frac{3a_1 - 4a_2 + a_3}{a_1 - 2a_2 + a_3}$ .
  - 6 wyznaczamy indeks  $i_{\min}$  najmniejszego  $a(\delta_i)$  dla  $i = 1, 2, 3, 4$
  - 7 przesuwamy wartość potencjału w punkcie  $u(i, j)$  o  $\delta_{i_{\min}}$ .



- rysunek z Numerical Recipes (metoda Brenta)



# szybkie liczenie energii

- wersja z laplasjanem

$$a = - \sum_{i,j=-N+1}^{N-1} \left[ \frac{1}{2} u(i, j) \frac{u(i+1, j) + u(i-1, j) - 2u(i, j)}{dx^2} \right. \quad (16)$$

$$+ \left. \frac{1}{2} u(i, j) \frac{u(i, j+1) + u(i, j-1) - 2u(i, j)}{dx^2} \right] \quad (17)$$

$$+ \rho(i, j) u(i, j) dx^2. \quad (18)$$

- szybkie wyliczenie  $a$  przy zmianie wartości  $u$  w punkcie  $(i, j)$ :

•

$$a_{\text{loc}}(u_{i', j'}) = - \sum_{i=i'-1}^{i'+1} \sum_{j=j'-1}^{j'+1} \left[ \frac{1}{2} u(i, j) \frac{u(i+1, j) + u(i-1, j) - 2u(i, j)}{dx^2} \right. \quad (19)$$

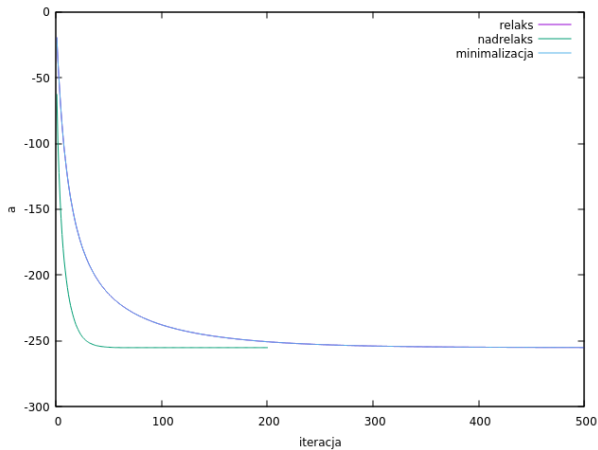
$$+ \left. \frac{1}{2} u(i, j) \frac{u(i, j+1) + u(i, j-1) - 2u(i, j)}{dx^2} \right] \quad (20)$$

$$+ \rho(i, j) u(i, j) dx^2. \quad (21)$$

- Wtedy  $a$  przy zmianie wartości  $u(i', j')$  o  $\delta$  można wyznaczyć jako  $a(\delta) = a(0) - a_{\text{loc}}(u_{i', j'}) + a_{\text{loc}}(u_{i', j'} + \delta)$ .

# wyniki

- film parabol.gif



- 
- tempo zbieżności: jak relaksacja

## problem z wzorem bez laplasjanu

- 

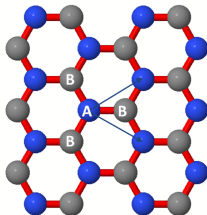
$$a = + \sum_{i,j=-N+1}^{N-1} \left[ \frac{1}{2} \left( \frac{u(i+1,j) - u(i-1,j)}{2dx} \right)^2 \right] \quad (22)$$

$$+ \frac{1}{2} \left( \frac{u(i,j+1) - u(i,j-1)}{2dy} \right)^2 \quad (23)$$

$$- \rho(i,j)u(i,j)]dx^2. \quad (24)$$

- brak związku między sąsiednimi punktami
- start od  $u = 1$  w środku pudła: kwadrat-pierwszej-pochodnej-dzialanie.gif
- start od  $u = 0$  w środku pudła: kwadrat-pierwszej-od-zera.gif

# grafen i stany fałszywe (spurious)

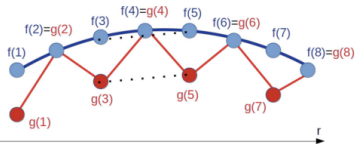


- 
- hamiltonian Diraca dla elektronów w grafenie (liniowa relacja dyspersji, nabla zamiast laplasjanu,  $k_x = -i \frac{\partial}{\partial x}$ )

$$H_\eta = \begin{pmatrix} U_A(\mathbf{r}) & \hbar v_f(k_x - i\eta k_y) \\ \hbar v_f(k_x + i\eta k_y) & U_B(\mathbf{r}) \end{pmatrix},$$

•

# grafen i stany fałszywe (spurious)



- równoważność prawdziwego (niebieska) i fałszywego stanu (czerwona).
- trik ze sztucznym wyrażeniem z laplasjanem (składnik Wilsona)

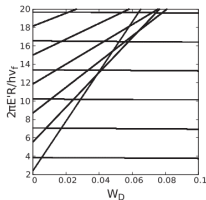


FIG. 2. Energy levels for  $\eta = 1$  and  $m = 0$  shifted by the Wilson term that separates the spurious energy levels from the original ones; see Table II. Results were obtained for  $N = 200$  mesh points.

$$H_D = -W_D \hbar v_f \nabla^2 \sigma_z dr,$$

- PRB 99, 195406 (2019)