

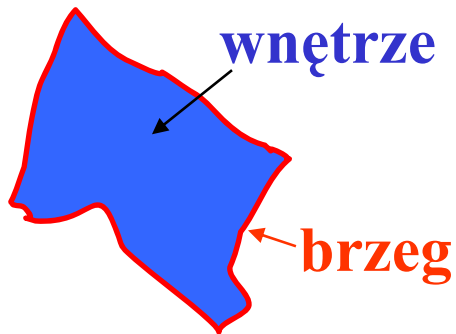
równanie Poissona jako modelowe eliptyczne funkcjonał działania, zbieżność, relaksacje wielosiatkowe

Eliptyczne: opisuje stany stacjonarne

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

- 1) Rozkład potencjału elektrostatycznego
[minimum *działania* w układzie ładunek/pole]
- 2) Rozkład temperatury przy stacjonarnym przepływie ciepła
[granica czasowa problemu parabolicznego]

...



na brzegu musimy określić wartość rozwiązania lub jego pochodnej normalnej lub związek między nimi

Warto wiedzieć: (zasada maximum)
rozwiązanie równanie Laplace'a osiąga wartości
ekstremalne na brzegach (dowód np. u Weinbergera)

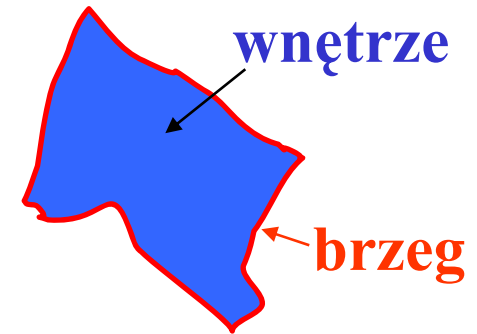
dla metody RS:

$$\nabla^2 \phi = 0$$



$$\Phi(i, j) = (\Phi(i - 1, j) + \Phi(i, j - 1) + \Phi(i + 1, j) + \Phi(i, j + 1))/4$$

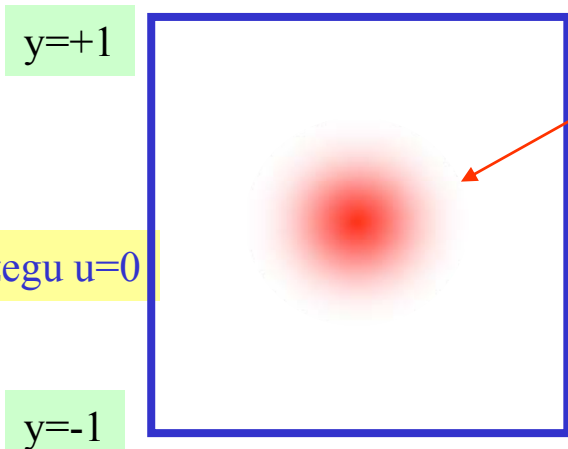
skoro każdy punkt z wewnątrz
obszaru całkowania jest średnią arytmetyczną z sąsiadów
nigdy nie będzie większy od żadnego z nich



Schemat Metropolisisa a równania różniczkowe - wariacyjna metoda MC rozwiązywania równania Poissona

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = -\rho(x, y)$$

przypomnienie



na brzegu $u=0$

$$\rho(x, y) = \exp(-50(x^2 + y^2))$$

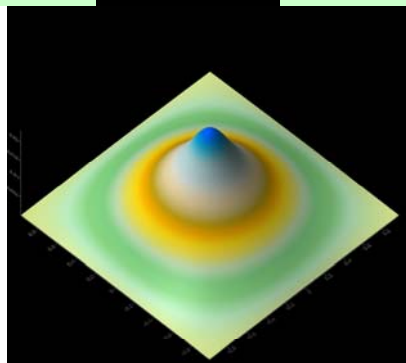
pokażemy (w swoim czasie),
że dla u - rozwiązania r.Poissona
minimalna jest całka:

$$S = \int dx dy \left\{ \frac{1}{2} \left[\left(\frac{\partial u}{\partial x} \right)^2 + \left(\frac{\partial u}{\partial y} \right)^2 \right] - \rho(x, y) u(x, y) \right\}$$

rozwiązanie próbne:

$$\tilde{u} = C (x - 1) (x + 1) (y - 1) (y + 1) e^{-a(x^2 + y^2)}$$

r. dokładne:



C, a - parametry wariacyjne
należy wybrać tak aby S - optymalne

Z elektrostatyki poprzez metodę różnic skończonych do równania Poissona i metod relaksacji i nadrelaksacji .

Działanie dla układu ładunków (ρ) + pole :

$$S = \int \left(\frac{1}{2} \mathbf{E}^2 - \rho \phi \right) dv = \int \left(\frac{1}{2} (\nabla \phi)^2 - \rho \phi \right) dv$$

/ potencjał pola elektrycznego

(to **nie** jest energia układu energia będzie gdy znak przy ρ będzie +)

wektor pola elektrycznego

rozkład ładunku

funkcja podcałkowa:
tzw. lagranżjan układu pole - ładunek

$$\mathbf{E} = -\nabla \phi$$

$$S = \min \longrightarrow \nabla^2 \phi = -\rho$$

Działanie jest najmniejsze dla potencjału, który spełnia równanie Poissona

Zobaczmy to w 1D:



działanie a równanie Poissona w 1D

$$S = \int_{-d/2}^{d/2} \left(\frac{1}{2} \left(\frac{d\phi}{dz} \right)^2 - \rho\phi \right) dz$$

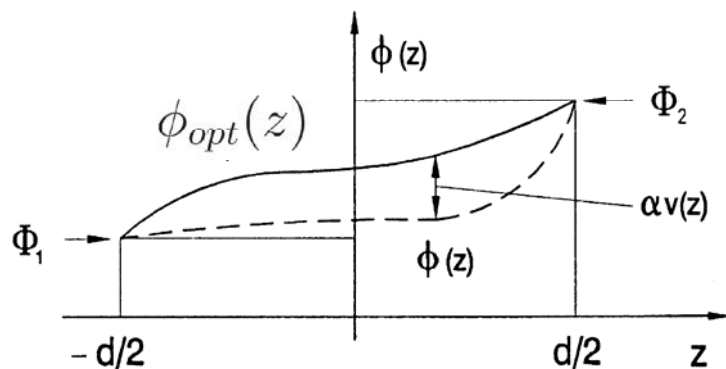
Z warunkami brzegowymi typu Dirichleta

$$\phi(z = -d/2) = \Phi_1, \quad \phi(z = d/2) = \Phi_2$$

Dla jakiego ϕ wartość działania jest ekstremalna? (w praktyce minimalna, bo maksymalna nie istnieje).

Ogólny problem minimum funkcjonału (całki funkcjonalnej)

$$F[\phi(z)] = \int_{-d/2}^{d/2} f[\phi(z), \phi'(z); z] dz$$



$\phi_{opt}(z)$ optymalny potencjał: minimalizuje działanie

$\phi(z)$ „bliski” optymalnemu i spełniający te same warunki brzegowe

$\phi(z) = \phi_{opt}(z) + \alpha v(z)$ ← dowolna funkcja ciągła z pochodną

mały parametr

$$v(z = -d/2) = v(z = d/2) = 0$$

$$\phi(z) = \phi_{opt}(z) + \alpha v(z)$$

z definicji: $F[\phi(z)] \geq F[\phi_{opt}(z)]$

$$F[\phi(z)] = \int_{-d/2}^{d/2} f[\phi_{opt}(z) + \alpha v(z), \phi'_{opt}(z) + \alpha v'(z); z] dz$$

Wartość $\alpha=0$ jest optymalna:

$$\frac{d}{d\alpha} \left(\int_{-d/2}^{d/2} f[\phi_{opt}(z) + \alpha v(z), \phi'_{opt}(z) + \alpha v'(z); z] dz \right)_{\alpha=0} = 0$$

Pochodna pod całką:

$$\int_{-d/2}^{d/2} \left[\frac{\partial f}{\partial \phi} \frac{\partial \phi}{\partial \alpha} + \frac{\partial f}{\partial \phi'} \frac{\partial \phi'}{\partial \alpha} \right]_{\alpha=0} dz = 0$$

\uparrow $v(z)$ \uparrow $v'(z)$

(wstawiamy $\alpha=0$)

$$\int_{-d/2}^{d/2} \left[\frac{\partial f}{\partial \phi_{opt}} v(z) + \frac{\partial f}{\partial \phi'_{opt}} v'(z) \right] dz = 0$$

$$\int_{-d/2}^{d/2} \left[\frac{\partial f}{\partial \phi_{opt}} v(z) + \frac{\partial f}{\partial \phi'_{opt}} v'(z) \right] dz = 0$$

pochođna iloczynu [całkowanie przez części]

$$\frac{\partial f}{\partial \phi'_{opt}} v'(z) = \frac{d}{dz} \left[\frac{\partial f}{\partial \phi'_{opt}} v(z) \right] - \left[\frac{d}{dz} \frac{\partial f}{\partial \phi'_{opt}} \right] v(z)$$

$$\int_{-d/2}^{d/2} dz \left[\frac{\partial f}{\partial \phi_{opt}} - \frac{d}{dz} \frac{\partial f}{\partial \phi'_{opt}} \right] v(z) + \left[\frac{\partial f}{\partial \phi'_{opt}} v(z) \right]_{-d/2}^{d/2} = 0$$

dowolna

$v(-d/2)=v(d/2)=0$

czyli:

$$\frac{\partial f}{\partial \phi_{opt}} - \frac{d}{dz} \frac{\partial f}{\partial \phi'_{opt}} = 0$$

→ równanie Eulera-Lagrange'a
na funkcję dla której całka funkcjonalna
minimalna

Równanie Eulera-Lagrange'a dla energii układu ładunek+pole

$$S = \int_{-d/2}^{d/2} \left(\frac{1}{2} \left(\frac{d\phi}{dz} \right)^2 - \rho\phi \right) dz \longrightarrow f(\phi, \phi'; z) = \frac{1}{2}(\phi')^2 - \rho\phi$$

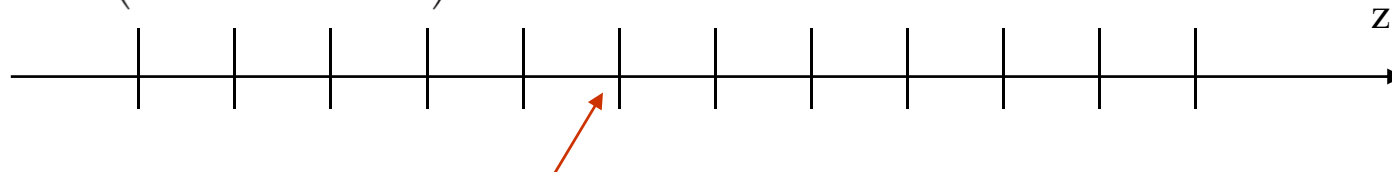
$$\frac{\partial f}{\partial \phi} - \frac{d}{dz} \frac{\partial f}{\partial \phi'} = 0$$

$$-\rho = \frac{d}{dz} \phi' = \phi''$$

minimalne działanie dostajemy
dla potencjału spełniającego
równanie Poissona

Działanie na siatce różnicowej

$$S = \int_{-d/2}^{d/2} \left(\frac{1}{2} \left(\frac{d\phi}{dz} \right)^2 - \rho\phi \right) dz$$



$$\phi_n = \phi(z_n)$$

$$\rho_n = \rho(z_n)$$

$$S = \sum_n \Delta z \left(\frac{1}{2} (\phi'_n)^2 - \rho_n \phi_n \right) \quad \text{Zdyskretyzowane działanie}$$

$$\phi'_n = \frac{\phi_n - \phi_{n-1}}{\Delta z} \quad \text{najprostszy iloraz różnicowy pierwszej pochodnej}$$

Minimum zdyskretyzowanego działania $\frac{\partial S}{\partial \phi_i} = 0$ dla wszystkich oczek siatki i

$$\frac{\partial S}{\partial \phi_i} = \sum_n \left[\frac{1}{2\Delta z^2} \frac{\partial}{\partial \phi_i} (\phi_n - \phi_{n-1})^2 - \rho_n \frac{\partial \phi_n}{\partial \phi_i} \right] = 0$$

$$\frac{\partial S}{\partial \phi_i} = \sum_n \left[\frac{1}{\Delta z^2} \left((\phi_n - \phi_{n-1}) \delta_{ni} - (\phi_n - \phi_{n-1}) \delta_{(n-1)i} \right) - \rho_n \delta_{ni} \right] = 0$$

wysumowane z deltami Kroneckera:

$$\frac{\partial S}{\partial \phi_i} = \frac{1}{\Delta z^2} \left((\phi_i - \phi_{i-1}) - (\phi_{i+1} - \phi_i) \right) - \rho_i = 0$$

$$\frac{1}{\Delta z^2} (-\phi_{i-1} + 2\phi_i - \phi_{i+1}) = \rho_i$$

$$\frac{\partial S}{\partial \phi_i} = \sum_n \left[\frac{1}{2\Delta z^2} \frac{\partial}{\partial \phi_i} (\phi_n - \phi_{n-1})^2 - \rho_n \frac{\partial \phi_n}{\partial \phi_i} \right] = 0$$

$$\frac{\partial S}{\partial \phi_i} = \sum_n \left[\frac{1}{\Delta z^2} \left((\phi_n - \phi_{n-1}) \delta_{ni} - (\phi_n - \phi_{n-1}) \delta_{(n-1)i} \right) - \rho_n \delta_{ni} \right] = 0$$

wysumowane z deltami Kroneckera:

$$\frac{\partial S}{\partial \phi_i} = \frac{1}{\Delta z^2} \left((\phi_i - \phi_{i-1}) - (\phi_{i+1} - \phi_i) \right) - \rho_i = 0$$

z zasady najmniejszego działania na siatce dostaliśmy dokładnie takie samo równanie, jak po bezpośredniej dyskretyzacji równania Poissona:

$$\frac{1}{\Delta z^2} (-\phi_{i-1} + 2\phi_i - \phi_{i+1}) = \rho_i$$

$$-\rho = \frac{d}{dz} \phi' = \phi''$$

z ilorazem różnicowym drugiej pochodnej

$$\phi_i'' = \frac{\phi_{i+1} + \phi_{i-1} - 2\phi_i}{\Delta z^2}$$

wartość działania:

$$S = \int_{-d/2}^{d/2} \left(\frac{1}{2} \left(\frac{d\phi}{dz} \right)^2 - \rho\phi \right) dz$$

pozwała ocenić zbieżność procedur iteracyjnych

ponadto: nieoceniona do kontroli jakości

rozwiązania w metodzie elementów skończonych

(wybór elementów, wybór funkcji kształtu)

Zbieżność procedury relaksacyjnej

Przepis relaksacyjny

$$\frac{1}{\Delta z^2} (-\phi_{i-1} + 2\phi_i - \phi_{i+1}) = \rho_i \rightarrow \phi'_i = \frac{\phi_{i+1} + \phi_{i-1} + \Delta z^2 \rho_i}{2}$$

Działanie:
$$a = \sum_n \left[\frac{1}{2\Delta z^2} (\phi_n - \phi_{n-1})^2 - \rho_n \phi_n \right]$$

Przyczynek do a od punktu i :

$$\begin{aligned} a_i \times 2\Delta z^2 &= (\phi_i - \phi_{i-1})^2 + (\phi_{i+1} - \phi_i)^2 - 2\Delta z^2 \rho_i \phi_i \\ &= 2\phi_i^2 + \phi_{i-1}^2 + \phi_{i+1}^2 - 2\phi_i(\phi_{i-1} + \phi_{i+1}) - 2\Delta z^2 \rho_i \phi_i \end{aligned}$$

Wyliczmy zmianę funkcjonału działania jeśli iteracji poddany zostanie TYLKO i -ty punkt siatki.

$$\begin{aligned} (a'_i - a_i) \times (2\Delta z^2) &= 2\phi_i'^2 - 2\phi_i^2 - 2(\phi'_i - \phi_i)(\phi_{i-1} + \phi_{i+1}) - 2\Delta z^2 \rho_i (\phi'_i - \phi_i) \\ \dots &= -(\phi_{i+1} + \phi_{i-1} + \Delta z^2 \rho_i - 2\phi_i)^2 / 2 \end{aligned}$$

Relaksacja potencjału w każdym z punktów prowadzi do obniżenia wartości działania!
[równanie P. zostanie rozwiązane gdy działanie obniży się do minimalnego]

Nadrelaksacja i podrelaksacja

Uogólnijmy schemat relaksacyjny:
do postaci:

$$\phi'_i = \frac{\phi_{i+1} + \phi_{i-1} + \Delta z^2 \rho_i}{2}$$

$$\phi'_i = (1 - \omega)\phi_i + \omega \frac{\phi_{i+1} + \phi_{i-1} + \Delta z^2 \rho_i}{2}$$

(wygląda jak
część starego rozwiązania + część nowego)

Wracamy do równania na zmianę działania:

$$\begin{aligned} (a'_i - a_i) \times (2\Delta z^2) &= 2(\phi'_i - \phi_i)(\phi'_i + \phi_i - (\phi_{i-1} + \phi_{i+1}) - \Delta z^2 \rho_i) \\ \dots &= 2\omega(\omega - 2) \left[-\phi_i + \frac{1}{2}(\phi_{i+1} + \phi_{i-1} + \Delta z^2 \rho_i) \right]^2 < 0 \text{ dla } \omega \in (0,2) \end{aligned}$$

$\omega = 1$ relaksacja

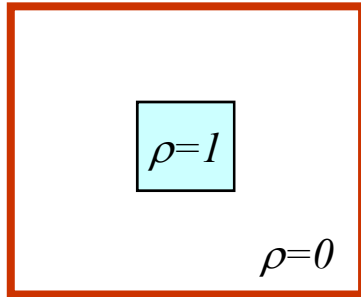
$\omega \in (0,1)$ – podrelaksacja - mniej „nowego” rozwiązania akceptowane w iteracji

$\omega \in (1,2)$ – nadrelaksacja – stare rozwiązanie jest usuwane z funkcji

uwaga: dotyczy relaksacji punktowej (nie globalnej) !

Na laboratorium zobaczymy, że globalna niezbieżna dla $\omega > 1$

nad-, pod- i relaksacja w praktyce (laboratorium)



$$\phi = 0$$

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

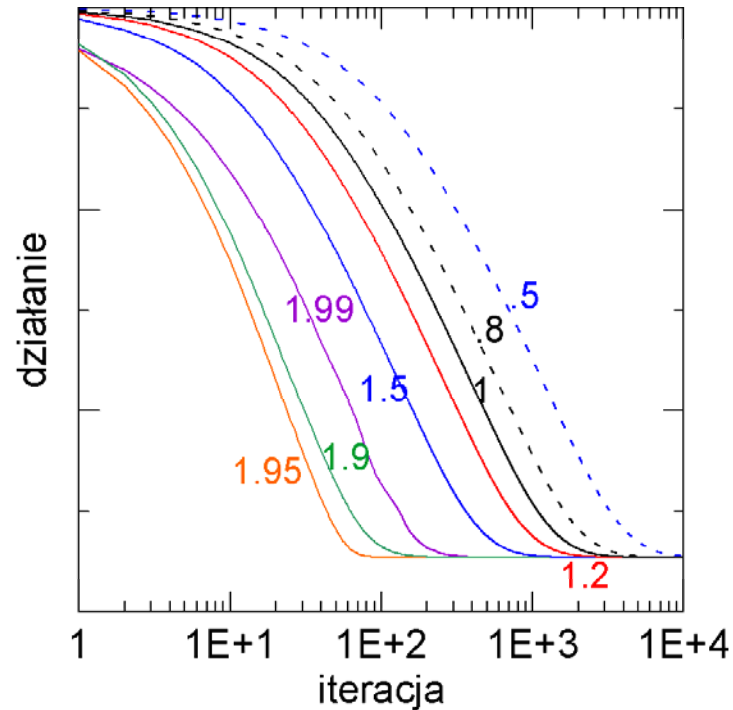
Rozwiązanie:

$$S = \int \left(\frac{1}{2} \mathbf{E}^2 - \rho \phi \right) dv = \int \left(\frac{1}{2} (\nabla \phi)^2 - \rho \phi \right) dv$$

pętla po i
pętla po j

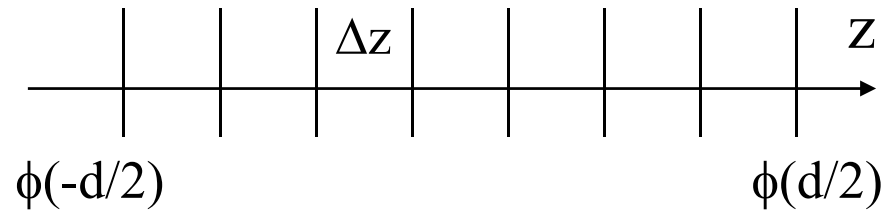
$$\phi'_{ij} = (1 - \omega) \phi_{ij} + \omega \frac{\phi_{(i+1)j} + \phi_{(i-1)j} + \phi_{i(j+1)} + \phi_{i(j-1)} + \Delta x^2 \rho_i}{4}$$

$$\phi_{ij} = \phi'_{ij}$$



Zdyskretyzowane równanie Poissona jako układ równań liniowych

$$\nabla^2 \phi = -\rho \quad \text{1D} \quad \phi(z = -d/2) = \Phi_1, \quad \phi(z = d/2) = \Phi_2$$



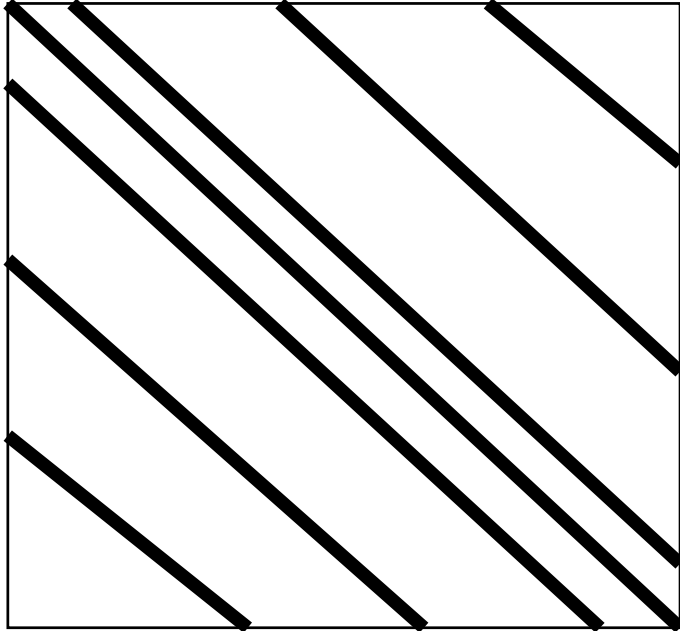
Metoda różnic skończonych

$$\phi_i'' = \frac{\phi_{i+1} + \phi_{i-1} - 2\phi_i}{\Delta z^2}$$

Układ równań $Au=b$

$$A = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ 1 & -2 & 1 & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & 1 & -2 & 1 & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & 1 & -2 & 1 & \dots & 0 & 0 & 0 \\ \dots & & & & & & & & \\ 0 & 0 & 0 & 0 & \dots & 0 & 1 & -2 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad u = \begin{pmatrix} \phi(-d/2) \\ \phi(-d/2 + \Delta z) \\ \phi(-d/2 + 2\Delta z) \\ \phi(-d/2 + 3\Delta z) \\ \phi(-d/2 + 4\Delta z) \\ \dots \\ \phi(d/2 - \Delta z) \\ \phi(d/2) \end{pmatrix} \quad b = \begin{pmatrix} \Phi_1 \\ -\Delta z^2 \rho(-d/2 + \Delta z) \\ -\Delta z^2 \rho(-d/2 + 2\Delta z) \\ -\Delta z^2 \rho(-d/2 + 3\Delta z) \\ -\Delta z^2 \rho(-d/2 + 4\Delta z) \\ \dots \\ -\Delta z^2 \rho(d/2 - \Delta z) \\ \Phi_2 \end{pmatrix}$$

niezerowe elementy w macierzy A (N na N), dla 3D równania Laplace'a



macierze są duże: rozmiar dla 3D: $N = 10^6$ już
w najprostszycy zastosowaniach
(100 na 100 na 100),
macierz $10^6 \times 10^6$ - około 1 TB,
ale do zapamiętania tylko 7 przekątnych
czyli najwyżej $7 \times 10^6 \times 8$ bajtów = 55 MB

faktycznie dla równania Laplace'a
wystarczy pamiętać strukturę (kilka liczb)

Rozwiązać metodą dokładną czy iteracyjną?

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

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

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

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

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

W równaniu występują dwa wektory (dla iteracji k i k+1)
Metoda Jakobiego to relaksacja globalna

Czy zbieżna? (na laboratorium – zobaczymy, że tak)
aby wykazać zbieżność – należy zbadać widmo macierzy M.

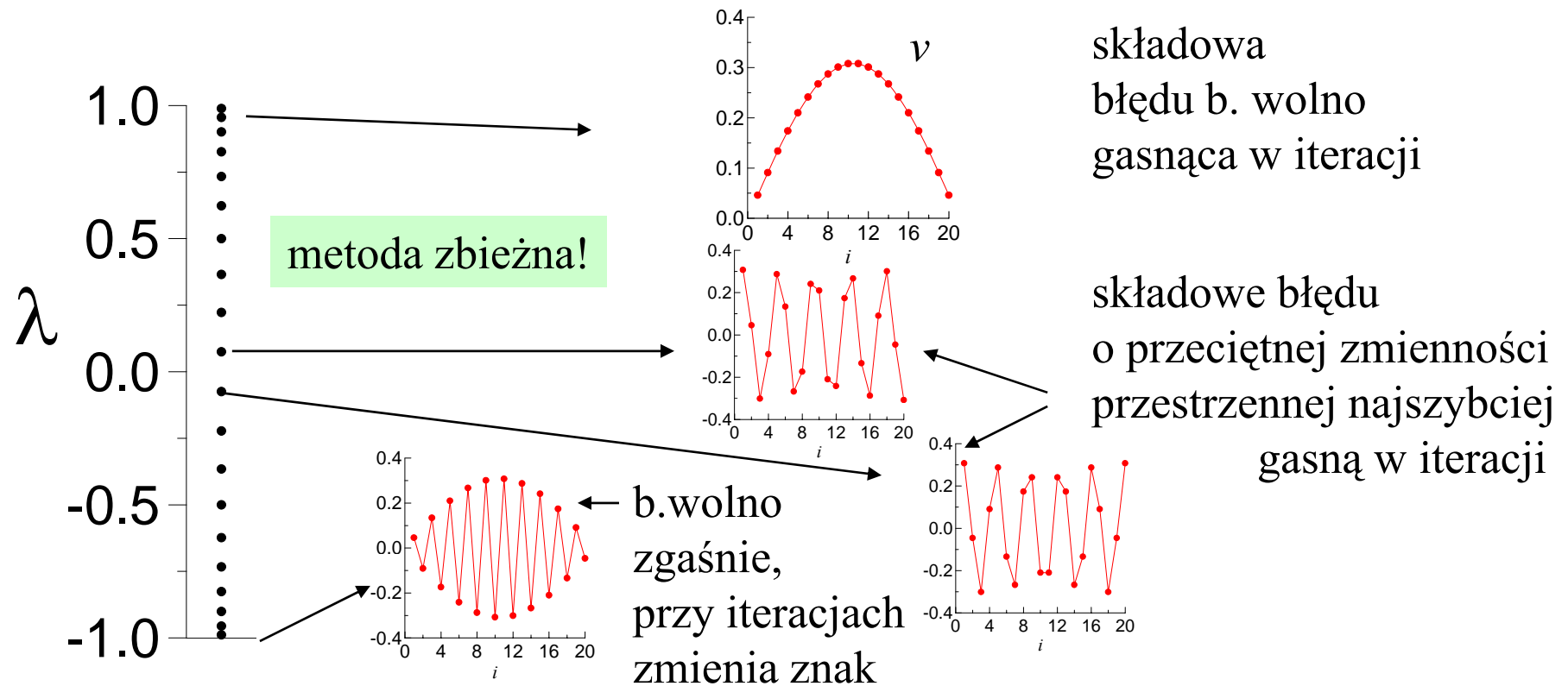
wybór Jakobiego: $B=D$, $C=(L+U)$

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

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

$$Mv = \lambda v$$

widmo wartości własnych: (N=20)



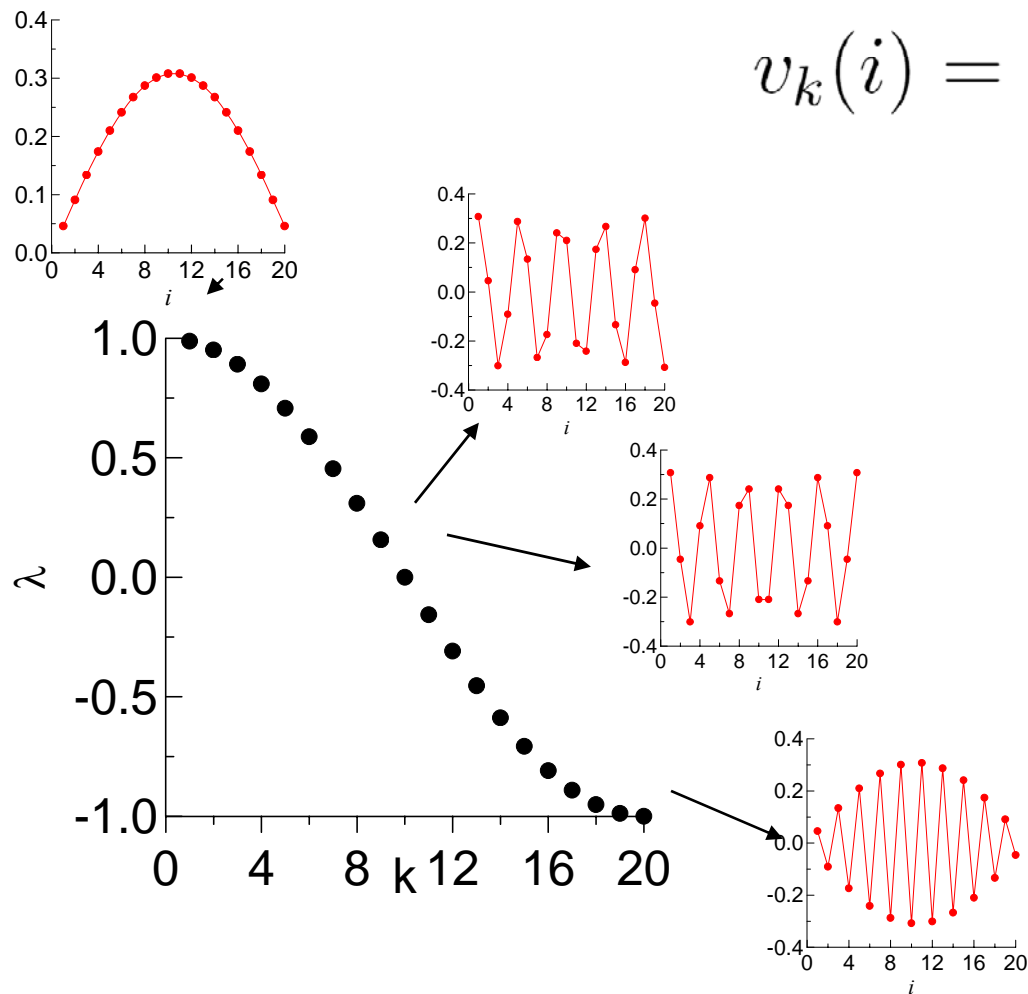
macierz iteracji Jakobiego: formuły analityczne na wektory i wartości własne

N – liczba oczek siatki

$k=1, N$

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

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



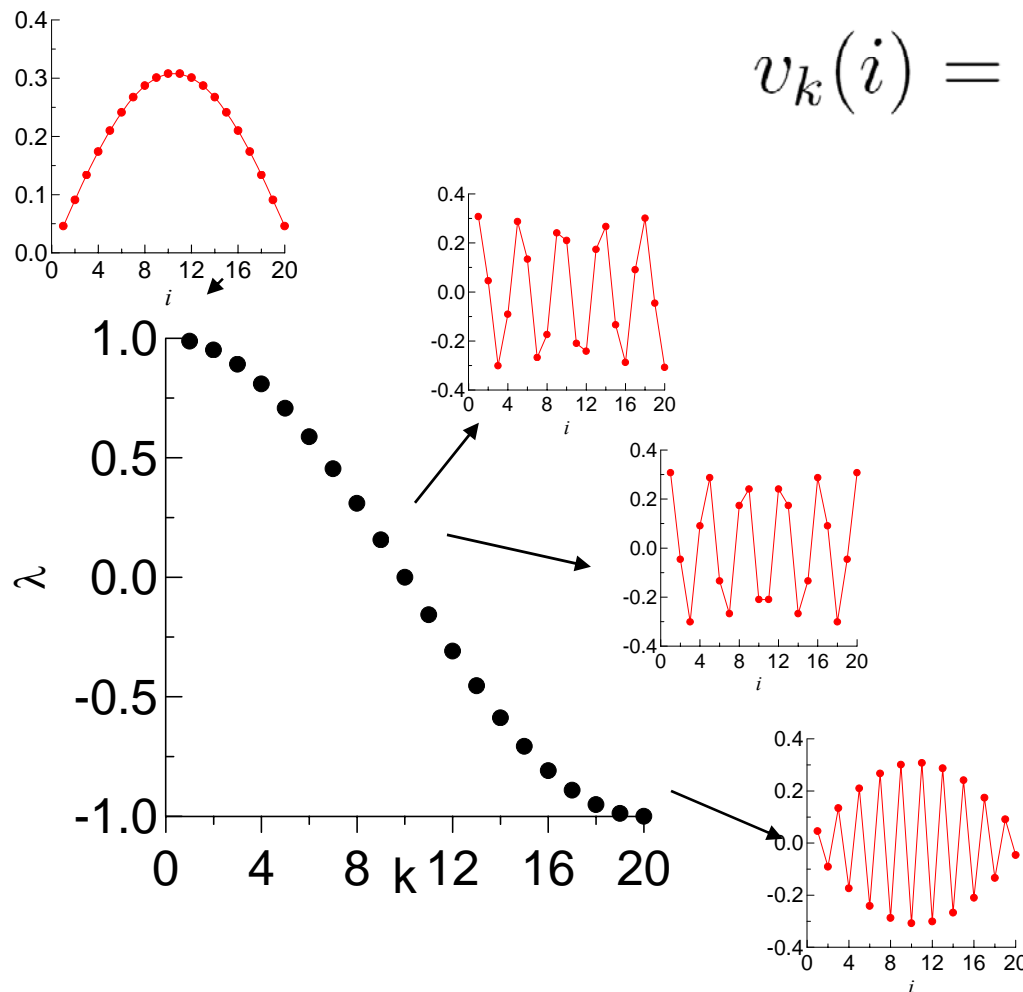
macierz iteracji Jakobiego: formuły analityczne na wektory i wartości własne

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 :

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

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

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

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

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
(przeglądamy 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 $i=1,N$

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

$x = x_n$

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)

$(B+C)x=b$

$Bx=b-Cx$

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

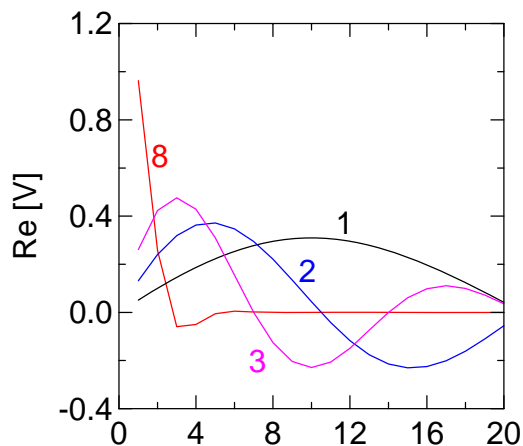
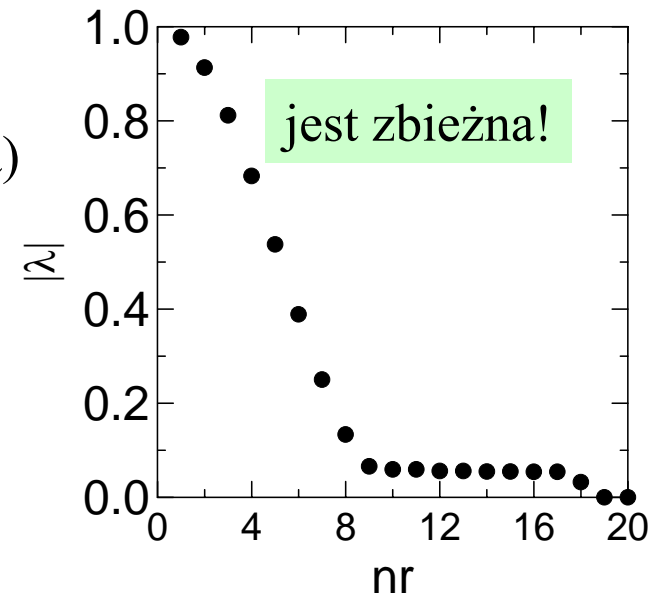
$M = -B^{-1}C, c = B^{-1}b$

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

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

$$M = \begin{pmatrix} 0 & \frac{1}{2} & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{4} & \frac{1}{2} & 0 & 0 & 0 \\ 0 & \frac{1}{8} & \frac{1}{4} & \frac{1}{2} & 0 & 0 \\ 0 & \frac{1}{16} & \frac{1}{8} & \frac{1}{4} & \frac{1}{2} & 0 \\ 0 & \frac{1}{32} & \frac{1}{16} & \frac{1}{8} & \frac{1}{4} & \frac{1}{2} \\ 0 & \frac{1}{64} & \frac{1}{32} & \frac{1}{16} & \frac{1}{8} & \frac{1}{4} \end{pmatrix}$$

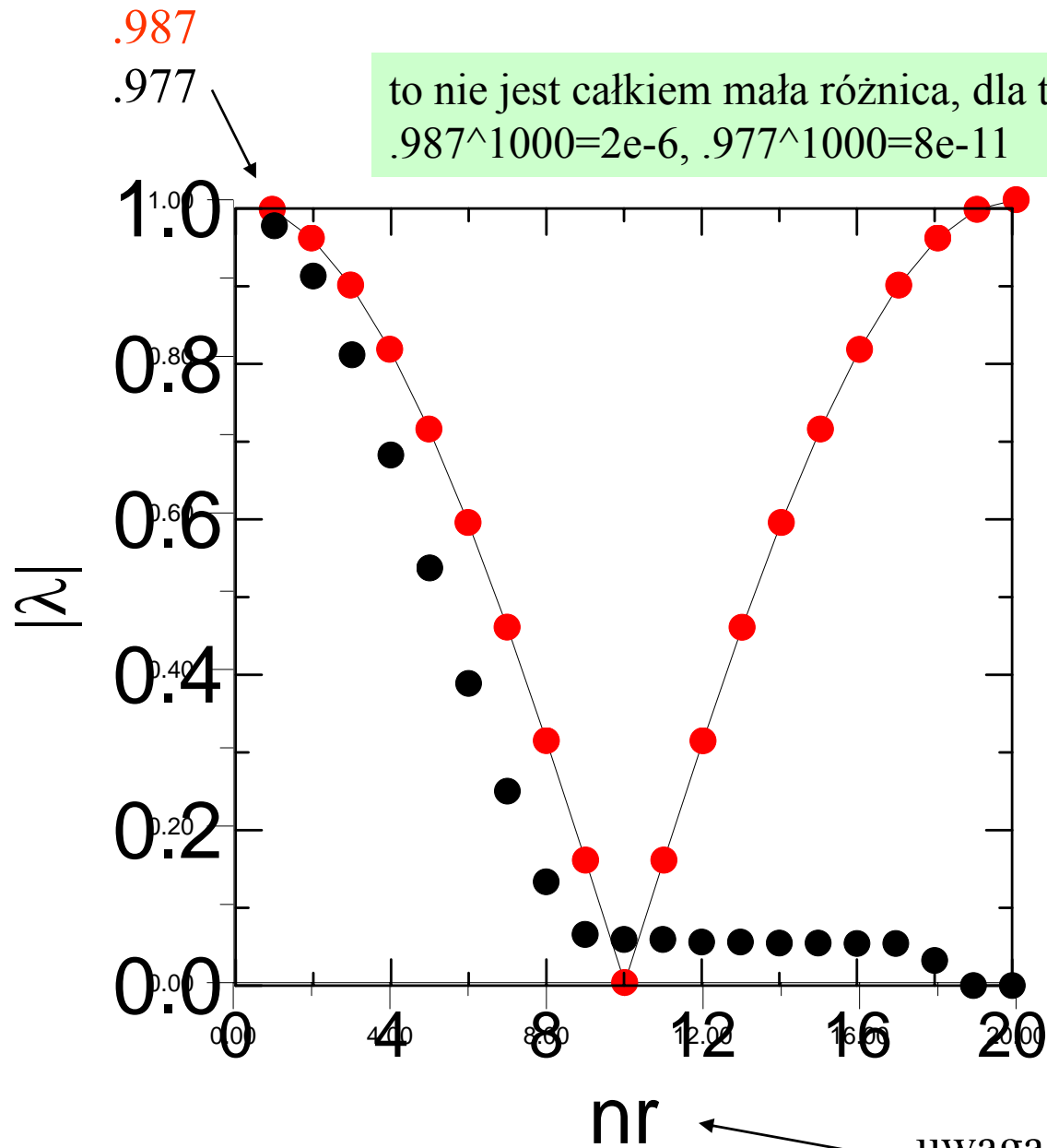
(dla dyskretyzacji
operatora Laplace'a)



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

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

metoda Jakobiego vs relaksacja Gaussa Seidla



to nie jest całkiem mała różnica, dla tysiąca iteracji:
 $.987^{1000}=2e-6$, $.977^{1000}=8e-11$

uwaga:
wektory własne: różne dla J i GS

Jakość „rozwiązania”: $Ax=b$

pozostałość (residuum): $r=Ax-b$

przykład numeryczny

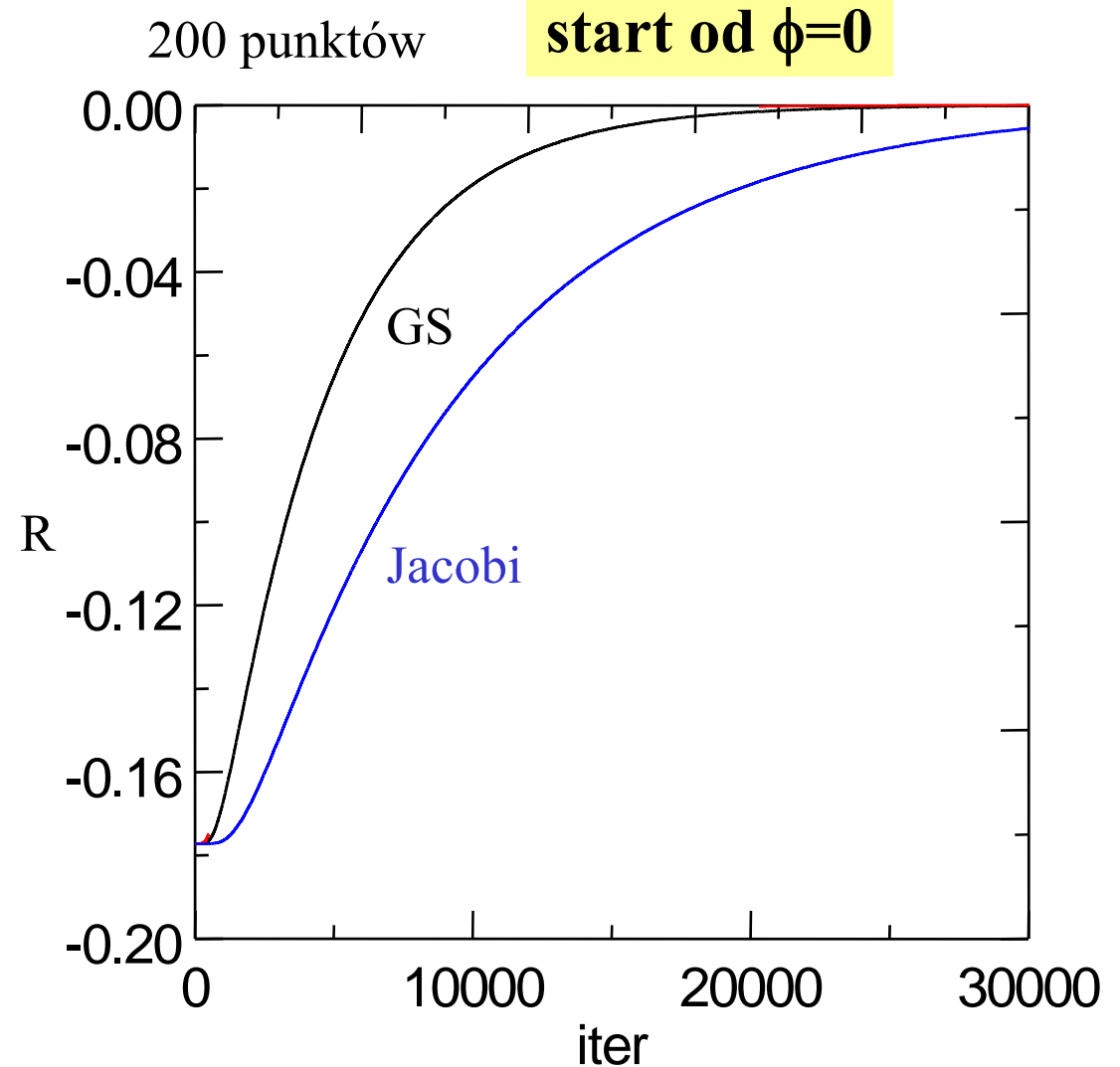
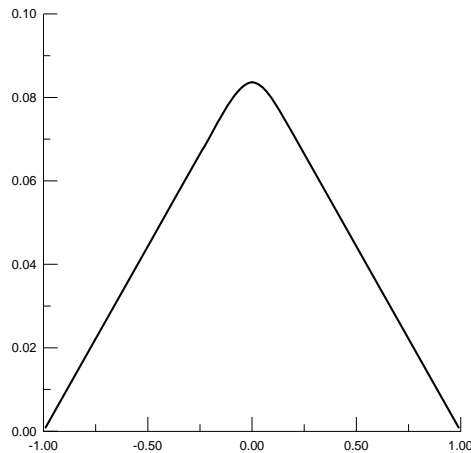
problem modelowy:

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

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

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

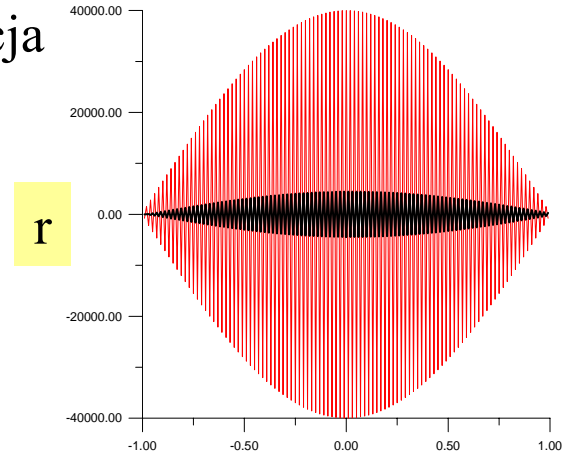
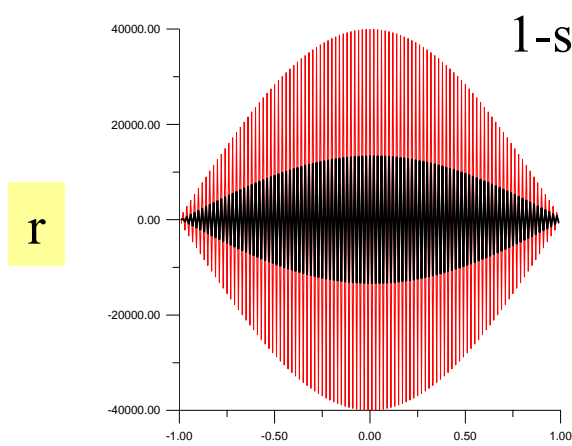
$$R = \text{suma } r_i$$



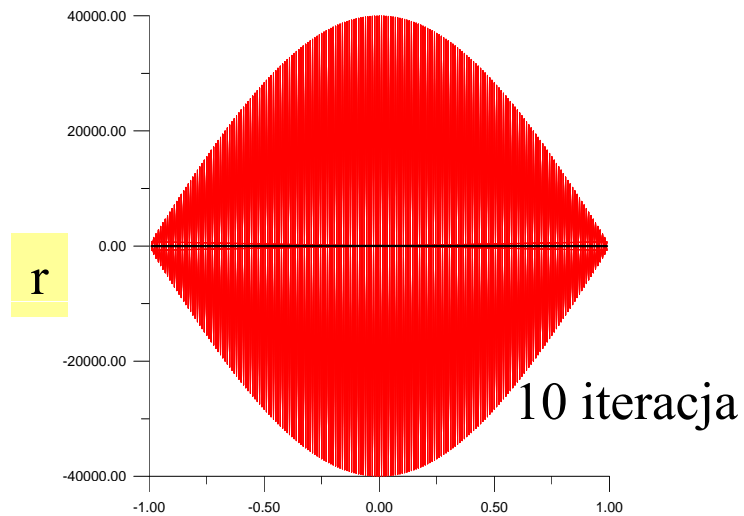
GS vs **Jacobi**: start od wektora *najkrótszej „długości fali”*

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

$k=N$



200 punktów

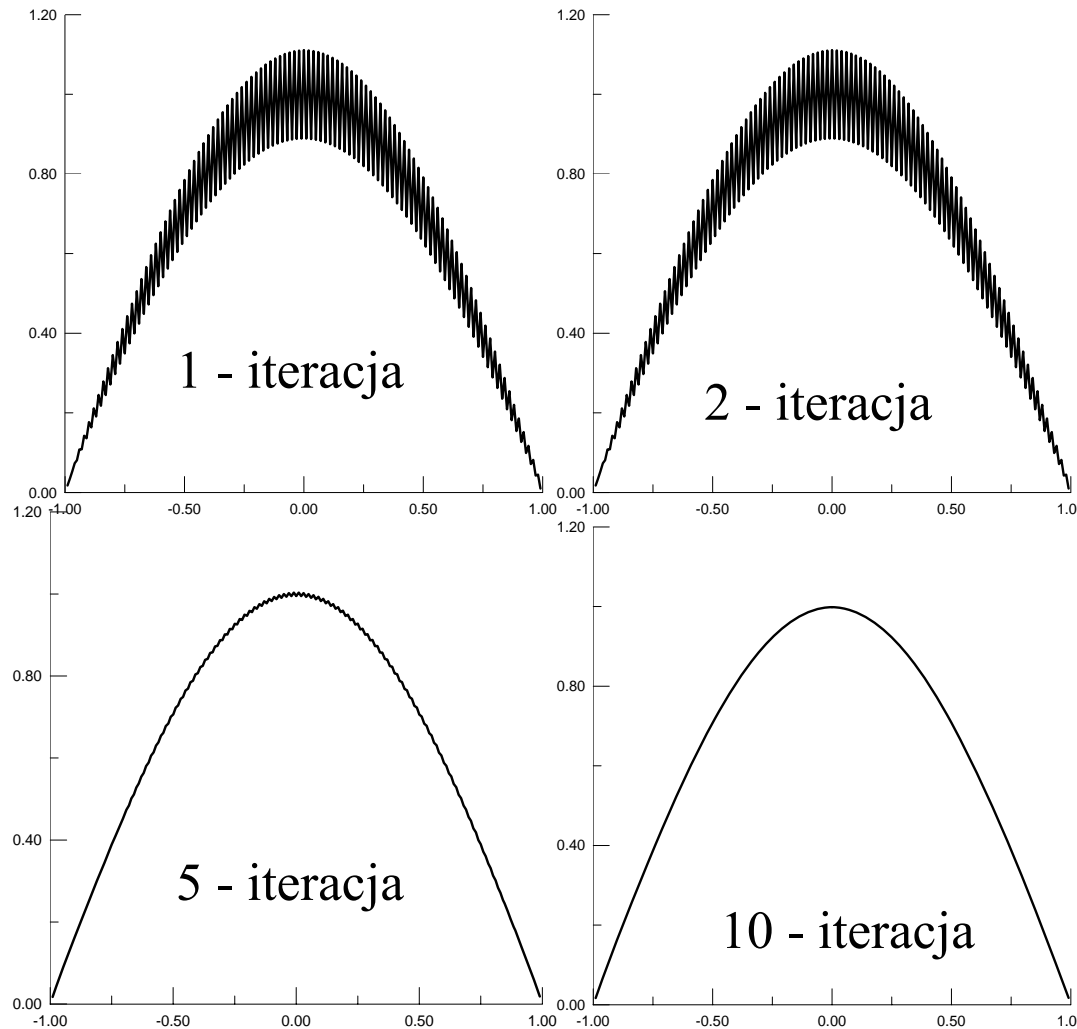


GS: szybkozmienny błąd
szybko tłumii
smoother

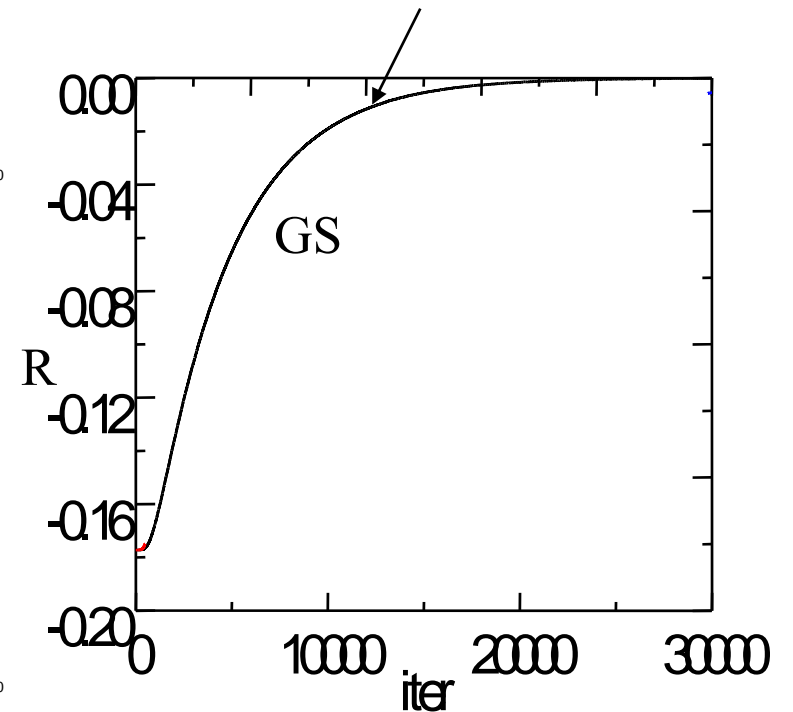
*Jacobi: wolno gasi
zarówno najszybsze
jak i najwolniejsze
błędy*

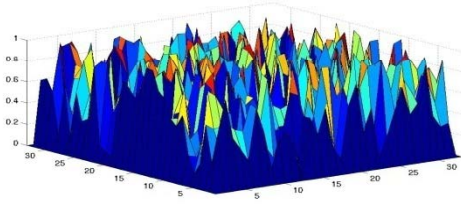
GS - smoothener: start od superpozycji wektorów z $k=1$ i $k=N$

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

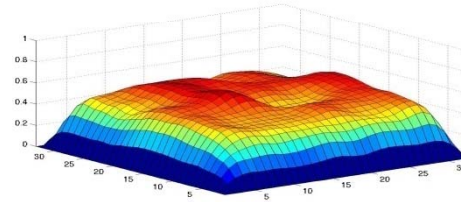


gdy relaksacja zwalnia błąd jest superpozycją wektorów własnych o niskich wartościach własnych

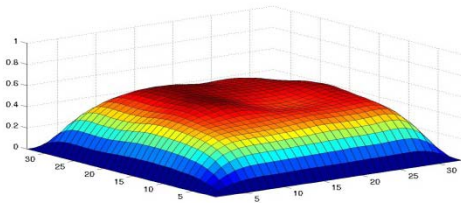




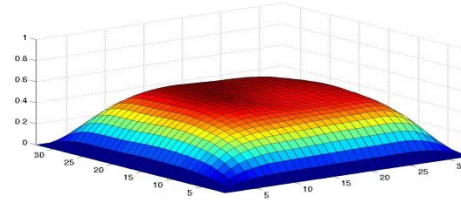
Error of initial guess



**Error after 5 relaxation
sweeps**



Error after 10 relaxations



Error after 15 relaxations

Fast error smoothing
slow solution

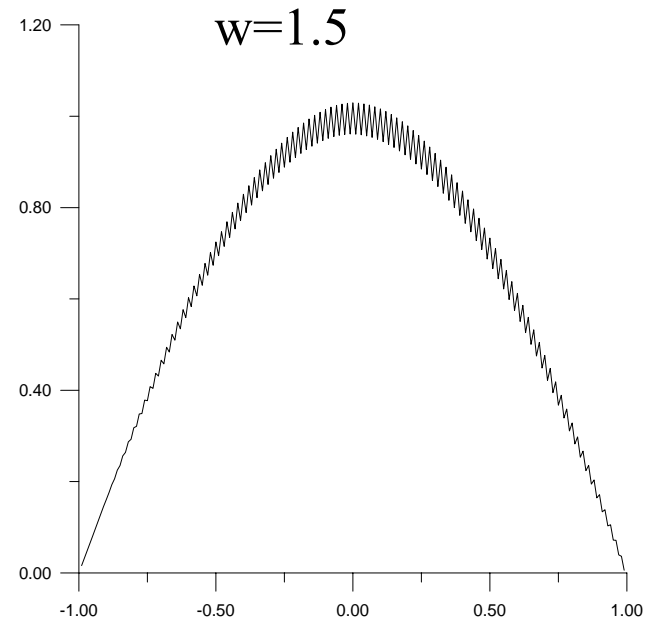
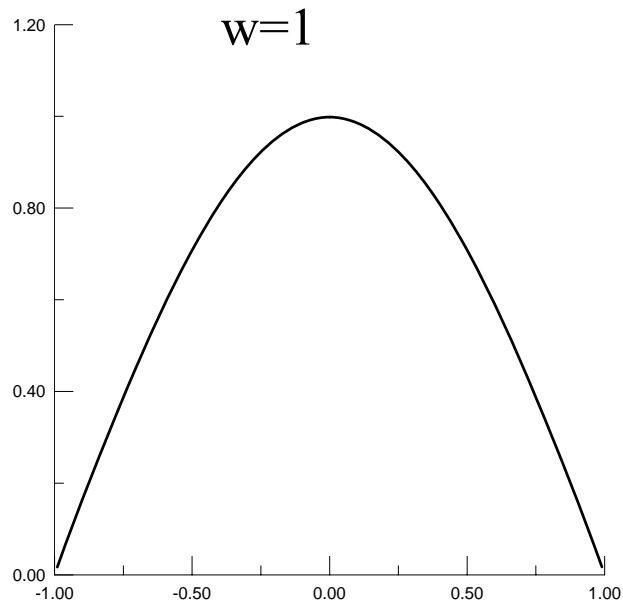
Achi Brandt

The Weizmann Institute of Science

UCLA

www.wisdom.weizmann.ac.il/~achi

10 iteracja, relaksacja a nadrelaksacja i gaszenie błędów



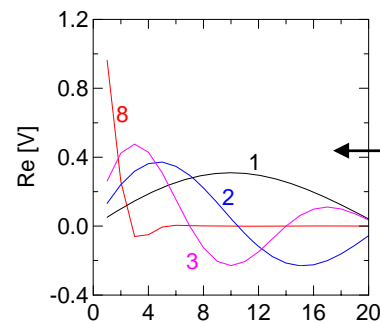
nadrelaksacja nie wygładza błędów

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 opiszemy
bez problemu

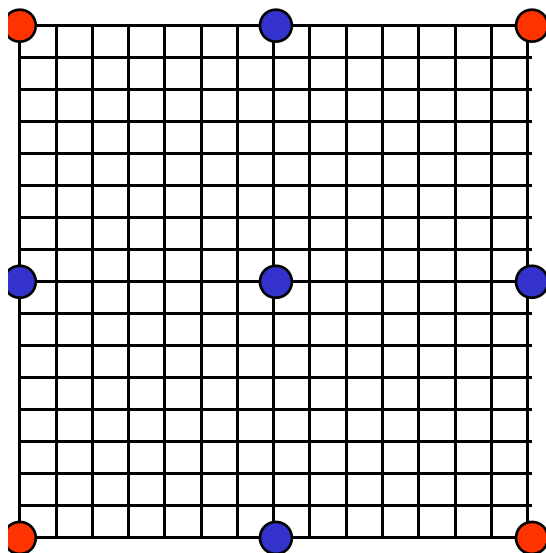
Rozwiązanie przepisane na gęstszą jako punkt startowy do nowego rachunku:
przy starcie ujawni się błąd szybkozmienny z przepisania. Możemy liczyć, że
szybko zgaśnie.

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

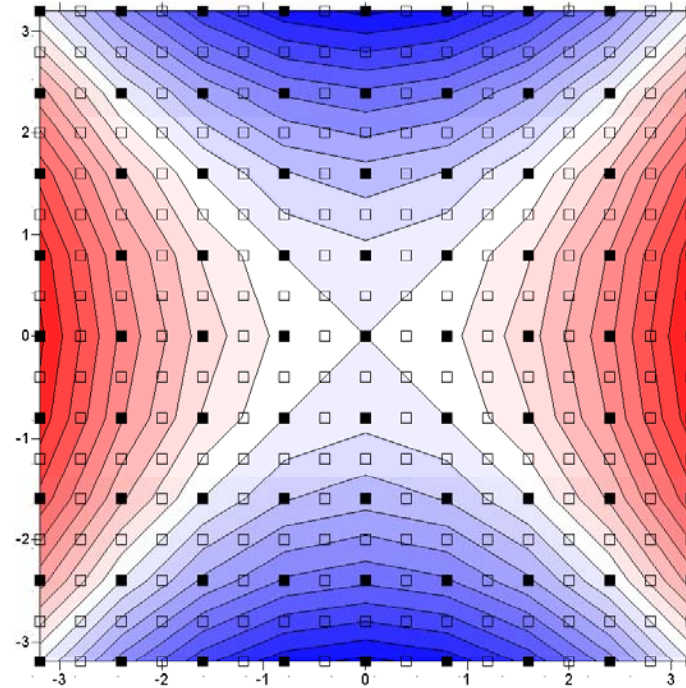


- 1) Rozwiązujemy zdyskretyzowane równanie różniczkowe na siatce Δx (**czerwone punkty**) (iterujemy do zbieżności)
- 2) Oszacowujemy wartość funkcji na punktach rozłożonych na siatce o skoku $\Delta x/2$ (**nowe-niebieskie punkty**)
- 3) Rozwiązujemy równanie na nowej siatce (**czerwone+nieskie**) (iterujemy 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 x 9 (czarne kwadraty)
i nowa siatka 17 x 17 (czarne+puste kwadraty)

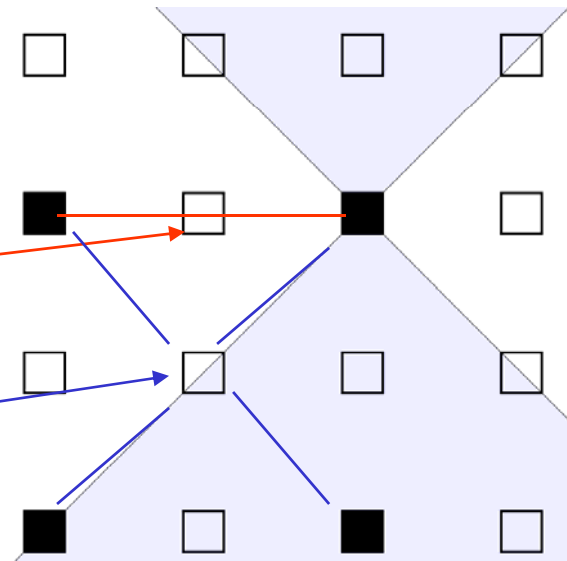


Start do iteracji na gęstszej siatce:

Punkty stare – zostawiamy bez zmian

Nowe punkty między dwoma starymi
średnia arytmetyczna z
dwóch czarnych sąsiadów

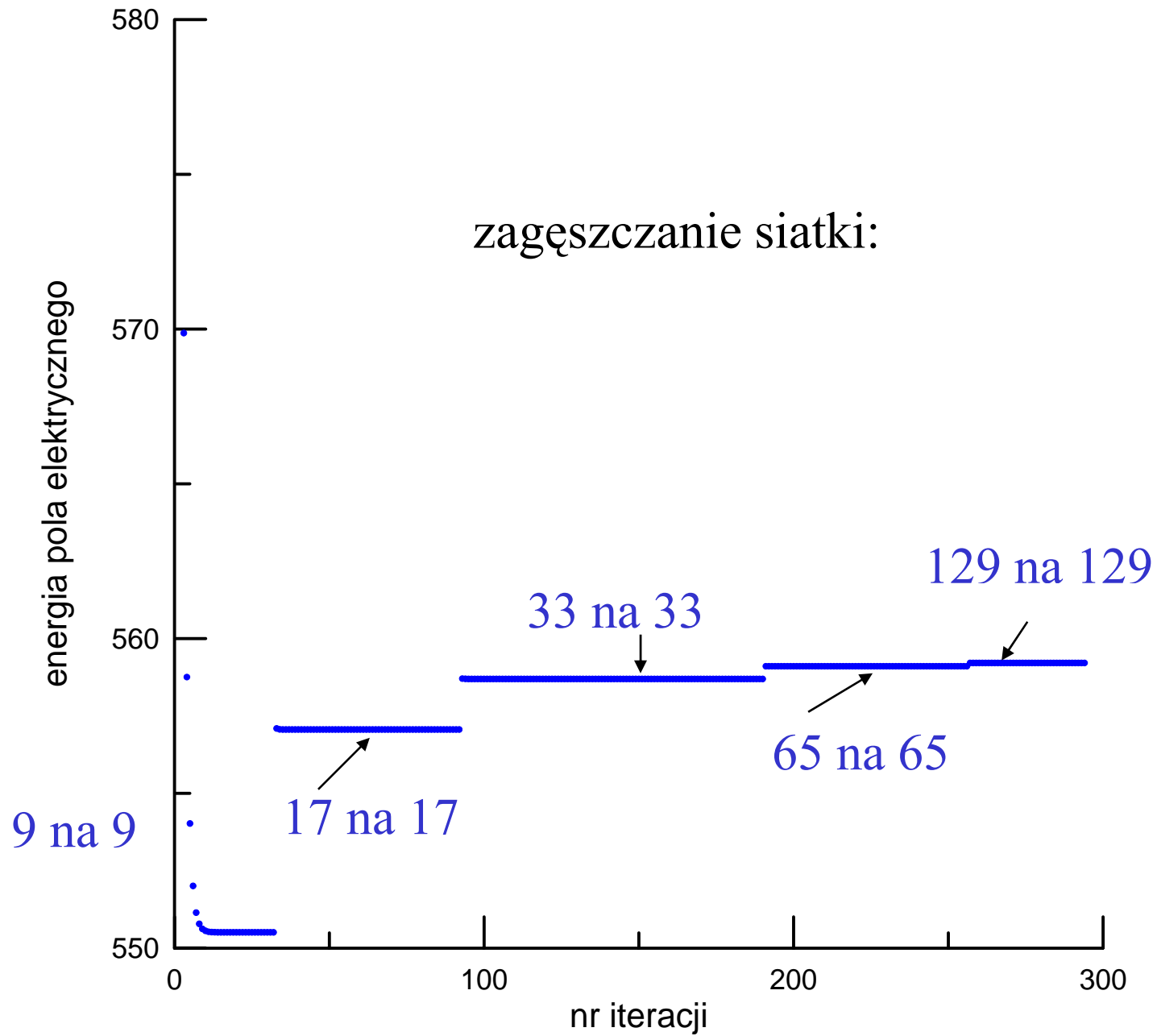
Pozostałe nowe – średnia z czterech
sąsiadów leżących w narożnikach
kwadratu



laboratorium

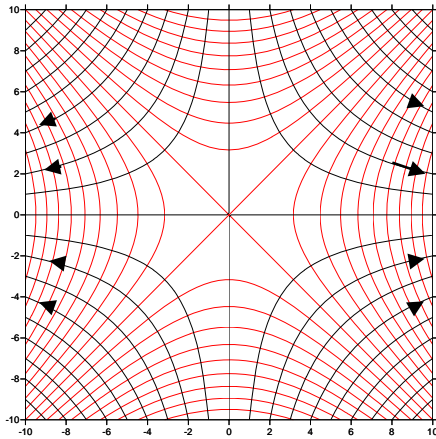






na najgęstszej siatce wykonujemy około 40 iteracji
zamiast 6500

zagęszczanie siatki dla równania Laplace'a 2D (przepływ stagnacyjny)



$$\phi = x^2 - y^2$$

start od zera wewnątrz pudła:

Chcemy rozwiązać równanie Laplace'a na potencjał przepływu dla $(x,y) \in [-3.2, 3.2]$ na siatce o skoku $\Delta x = \Delta y = 0.05$ (siatka 129 na 129 punktów).

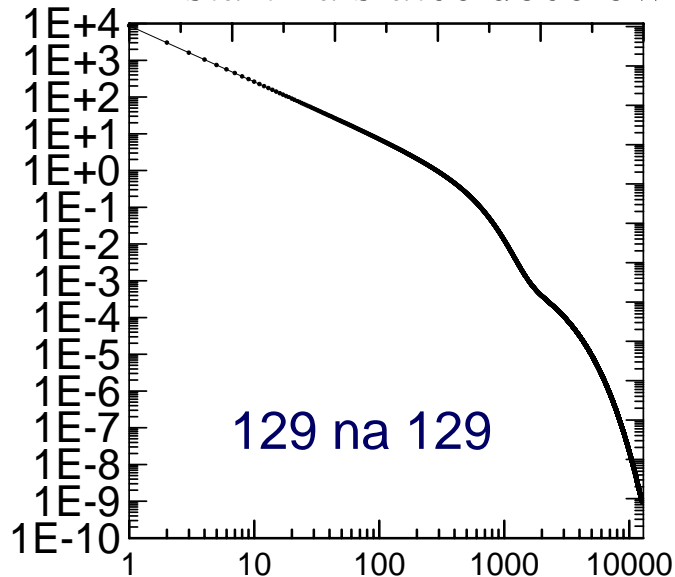
Monitorujemy sumę kwadratów pozostałości ($\nabla^2 \phi$)

Uznajemy, że rozwiązanie jest osiągnięte gdy $R^2 < 10^{-9}$.

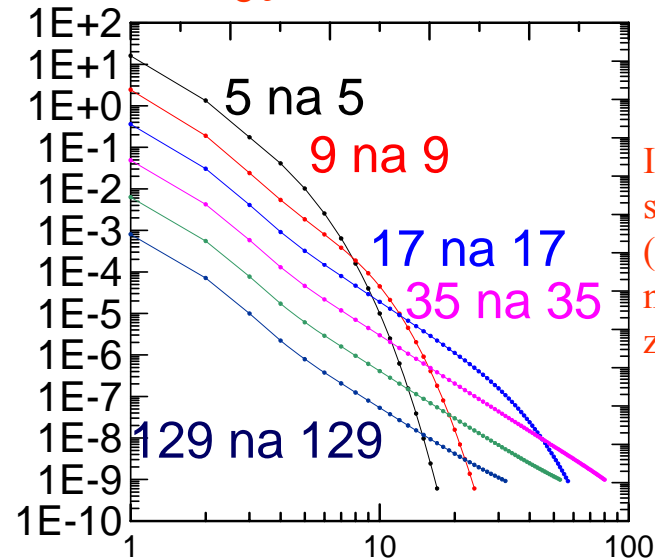
GS

błąd w funkcji
numeru iteracji

start na siatce docelowej



Zbieżność przy zastosowaniu
zagęszczania



Iteracja na pełnej
siatce 400 razy szybsza.
(czas spędzony
na rzadszych siatkach
zaniedbywalny)

Metody wielosiatkowe - idea:

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

- 1) rachunek na gęstej siatce przy pomocy iteracji wygładzającej
= błąd szybkozmienny szybko gaśnie (wykonujemy kilka iteracji,
nie walczymy o zbieżność)
- 2) błąd wolnozmienny – rzutowany na rzadką siatkę
(udaje się go prawie dokładnie wyrzutować bo jest wolnozmienny)
i eliminowany przez iterację
na rzadkiej 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 Δx i $2\Delta x$.

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

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

uzyskane przybliżenie wynik dokładny 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 restrykcji (*restriction*)

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

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

$Ae(2\Delta x) = r(2\Delta x)$ rozwiązujemy (iterujemy aż do zbieżności)

4. Przenosimy uzyskany błąd na siatkę gęstą

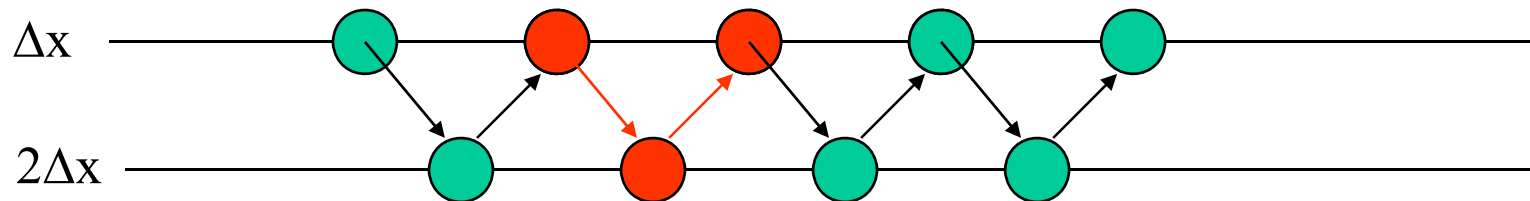
$$e(\Delta x) = I(2\Delta x \rightarrow \Delta x)e(2\Delta x)$$

operator przedłużenia (*prolongation*)

i poprawiamy rozwiązanie

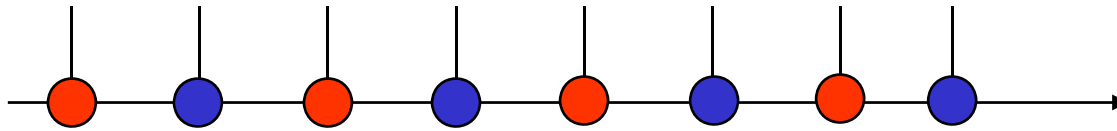
$$\tilde{x}(\Delta x) := \tilde{x}(\Delta x) - e(\Delta x)$$

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



V-cycle

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



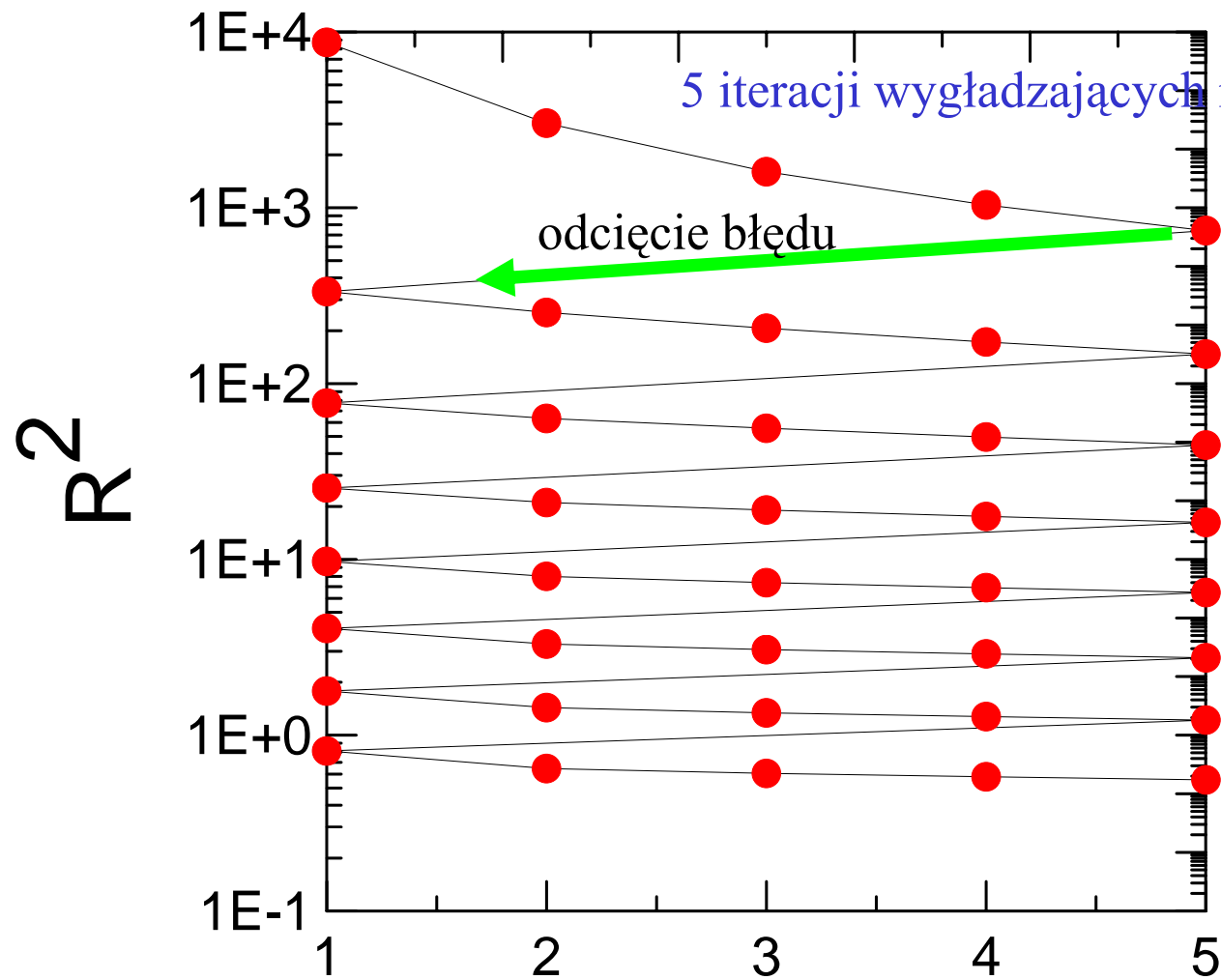
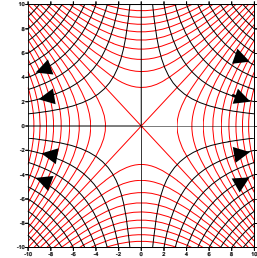
$I(\Delta x \rightarrow 2\Delta x)$ z czerwonych i niebieskich do czerwonych

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

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

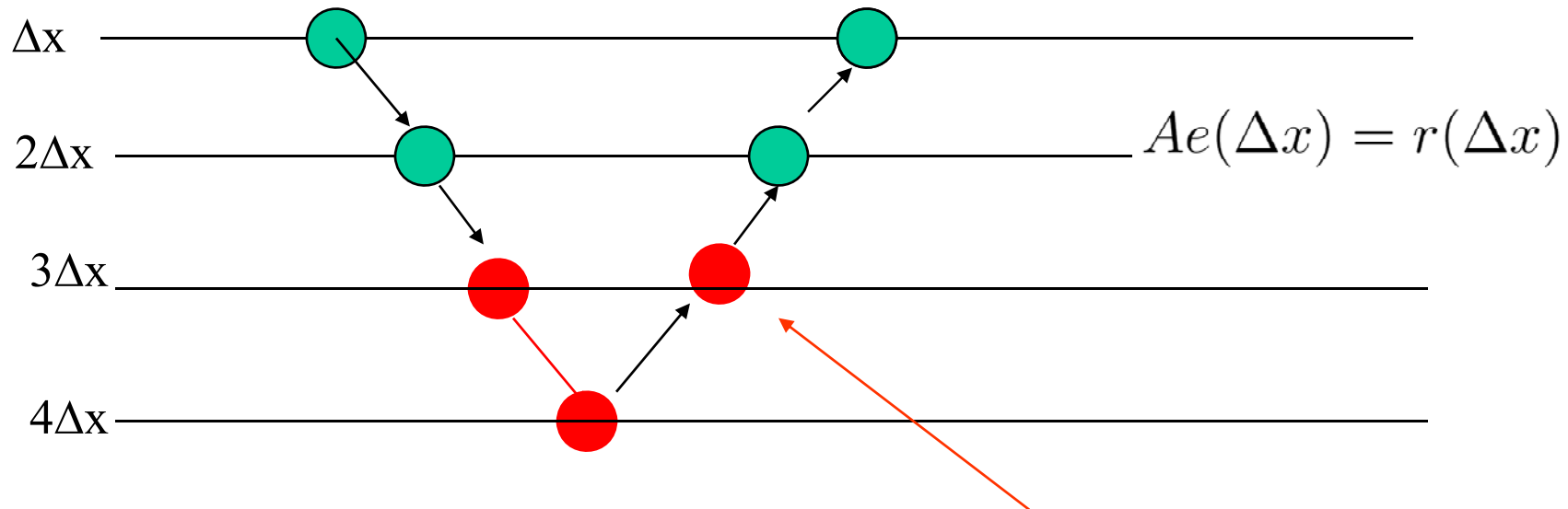
$$\begin{pmatrix} x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \\ x_{10} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 1/2 & 1/2 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1/2 & 1/2 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1/2 & 1/2 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1/2 & 1/2 \\ 0 &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

pozostałość w kwadracie dla iterowanego V-cyklu



rozwiązanie
zbieżne na
siatce 65 na 65

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



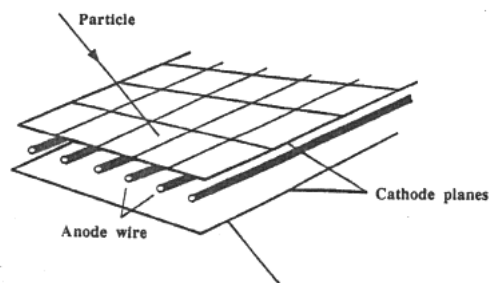
Równanie na błąd (na rzadszej siatce)

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

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?

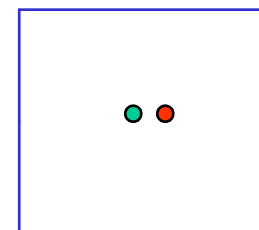
Można dać bardzo gęstą siatkę. Może nie starczyć pamięci.

Lepszy pomysł: wyliczyć potencjał metodą kolejnych powiększeń WB dla każdego powiększenia wyliczyć zgodnie z metodą zagęszczania siatki.

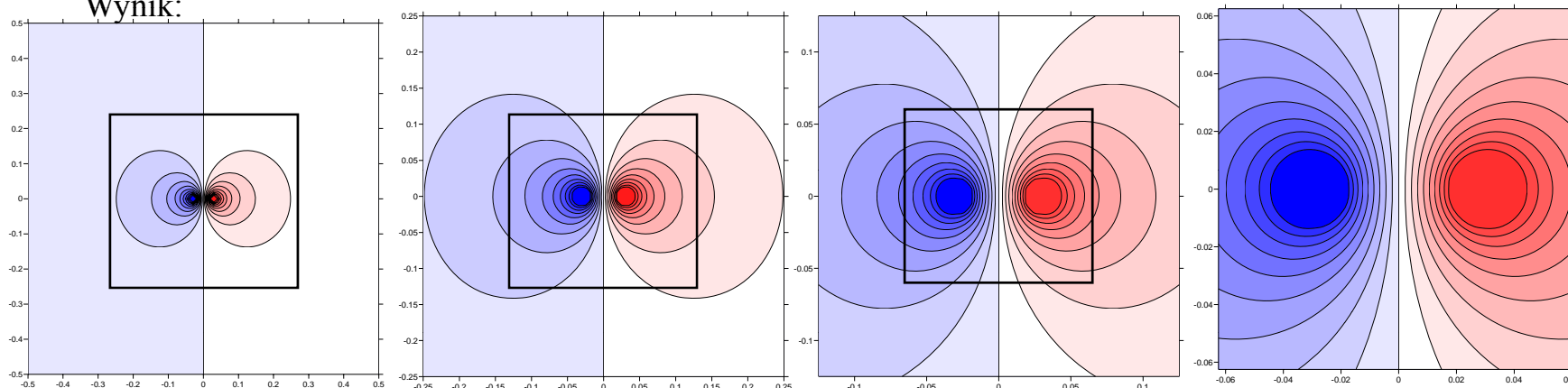
Mamy uziemione pudło metalowe o rozmiarach 1cm na 1cm.

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.