Równania różniczkowe cząstkowe (RRCz)

→ równanie eliptyczne

→ równanie Poissona

1. Klasyfikacja RRCz, przykłady

2. Metody numerycznego rozwiązywania równania Poissona

a) FFT (met. bezpośrednia)

b) metoda różnic skończonych

c) układ równań z jawnie zdefiniowaną macierzą

d) metody iteracyjne bez konstrukcji macierzy:
- metoda Jakobiego, Gaussa-Seidla, zbieżność
- całka działania
- relaksacja, nadrelaksacja
- metoda zagęszczania siatki
- relaksacja wielosiatkowa

Równania różniczkowe cząstkowe – klasyfikacja równań

Ogólna postać RRCz 2 rzędu zależnego od n zmiennych

$$\boldsymbol{z} \in R^n, \quad \text{np.:} \quad \boldsymbol{z} = (x, y, z, t)^T$$

$$-\sum_{i,j=1}^{n} \frac{\partial}{\partial z_i}\left(a_{i,j}\frac{\partial u}{\partial z_j}\right) + \sum_{i=1}^{n}\left(\frac{\partial b_i u}{\partial z_i} + c_i\frac{\partial u}{\partial z_i}\right) + a_0 u = f$$

i zakładamy

$$\begin{aligned} u &= u(\boldsymbol{z}) \in C^2 & \qquad c_i &= c_i(\boldsymbol{z}) \in C^1 \\ a_{i,j} &= a_{i,j}(\boldsymbol{z}) \in C^1 & \qquad a_0 &= a_0(\boldsymbol{z}) \in C \\ b_i &= b_i(\boldsymbol{z}) \in C^1 & \qquad f &= f(\boldsymbol{z}) \in C \end{aligned}$$

Zakładając, że: $\boldsymbol{v} \in R^n, \quad \boldsymbol{v} \neq 0$ wówczas RRCz jest:

- eliptyczne jeśli $v^T A v > 0$ (A jest dodatniookreślona)
- paraboliczne jeśli: $v^T A v \geq 0 \quad \wedge \quad rank\{A, b, c\} = n$
- hiperboliczne jeśli wartości własne A:

$$\lambda_1, \lambda_2, \ldots, \lambda_{n-1} > 0 \quad \wedge \quad \lambda_n < 0$$

Eliptyczne - równanie Poissona

$$-\nabla^2 = f$$

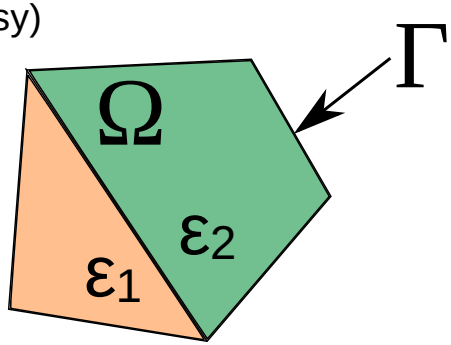
$$\boldsymbol{z} = (x, y, z)^T \qquad -\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2}\right) = f \qquad A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Opisuje rozwiązania stacjonarne. Gdzie się pojawia?

- Rozkład potencjału elektrycznego (grawitacyjnego)

$$-\nabla \cdot \varepsilon \nabla u = f$$

ε - „stała” dielektryczna (w określonych podobszarach Ω), f -gęstość ładunku (masy)



- stacjonarne rozwiązania równania dyfuzji (s – źródło ciepła)

$$\frac{\partial T}{\partial t} = \nabla^2 T - S \quad \Longrightarrow \quad \frac{\partial T}{\partial t} = 0$$

- Jedno z równań Naviera-Stokesa (stacjonarny przepływ cieczy lepkiej)

$$\nabla^2 \Psi = \zeta$$

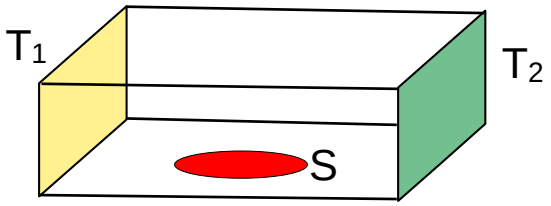
$$\mu \nabla^2 \zeta = \rho \left(\frac{\partial \psi}{\partial y}\frac{\partial \zeta}{\partial x} - \frac{\partial \psi}{\partial x}\frac{\partial \zeta}{\partial y}\right)$$

to równaniem Poissona nie jest

Paraboliczne – równanie dyfuzji ciepła

(k – wsp. przewodności cieplnej, S – źródło ciepła)

$$\frac{\partial T}{\partial t} - \nabla \cdot k \nabla T = -S$$



$$A = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad \lambda_4 = 0$$

4 zmienne niezależne - ale mamy tylko 3 pochodne 2 rzędu w równaniu

---

Hiperboliczne – równanie falowe

$$\frac{\partial^2 u}{\partial t^2} - \frac{1}{v^2}\nabla^2 u = \beta u + f$$

Opisują drgania mechaniczne np. membrany w 2D (f – wymuszenie, β - tłumienie)

lub rozchodzenie się fal elektromagnetycznych np. w światłowodzie

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} \quad \lambda_{1,2,3} = 1 \quad \lambda_4 = -1$$

Eliptyczne RRCz na przykładzie równania Poissona

Aby rozwiązanie równania eliptycznego było jednoznaczne musi ono spełniać określone warunki na brzegu obszaru
– rozwiązujemy tzw. **problem brzegowy** RRCz.

Typy warunków brzegowych (Γ - brzeg):

1. **Dirichleta**

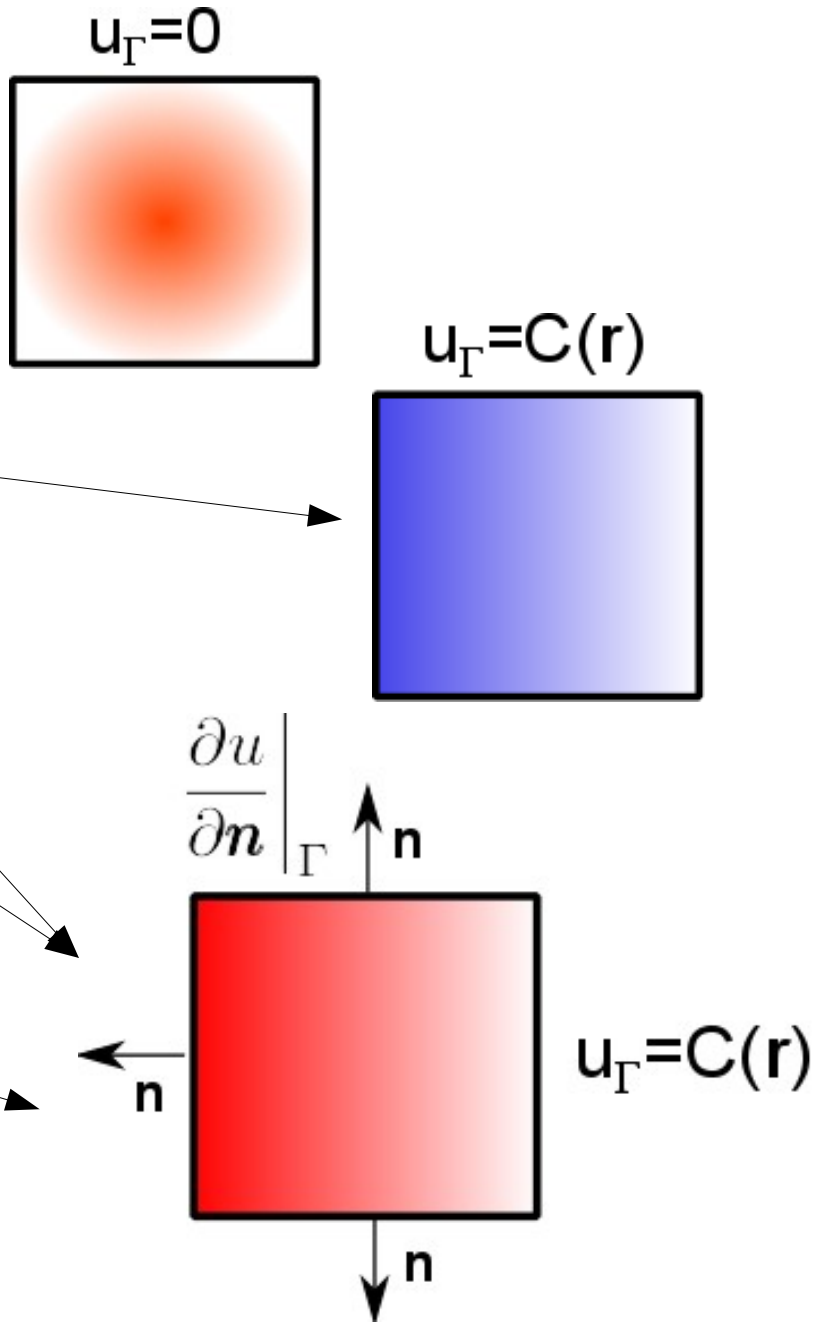
- jednorodne $u|_\Gamma = 0$
- niejednorodne $u|_\Gamma = C(\boldsymbol{r})$

2. **Neumanna**

$$\left.\frac{\partial u}{\partial \boldsymbol{n}}\right|_\Gamma = D(\boldsymbol{r})$$

3. **mieszane (Robina)**

$$\left(u + \frac{\partial u}{\partial \boldsymbol{n}}\right)_\Gamma = W(\boldsymbol{r})$$

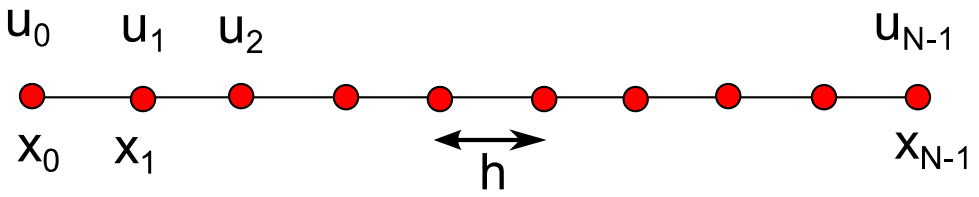


Metoda różnic skończonych w 1D/2D

Wprowadzamy siatkę przestrzenną
z równoodległymi węzłami

Rozwiązania u(x) poszukujemy
w położeniach węzłowych.

$$x_l = h \cdot l, \quad l = 0, N-1$$



Dyskretyzujemy rów. Poissona zastępując pochodne ciągłe ilorazami różnicowymi (np. 2 pochodna → iloraz trójpunktowy symetryczny)

$$\nabla^2 u(x) = -\rho(x)$$

1D

$$u(x) = u(x_l) = u_l$$

$$\rho(x) = \rho(x_l) = \rho_l$$

$$\nabla^2 u(x) = \frac{\partial^2 u}{\partial x^2} = -\rho(x)$$

$$\frac{u_{l-1} - 2 \cdot u_l + u_{l+1}}{h^2} = -\rho_l$$

2D

$$u(x,y) = u(x_l, y_j) = u_{l,j}$$

$$\rho(x,y) = \rho(x_l, y_j) = \rho_{l,j}$$

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

$$\frac{u_{l-1,j} - 2 \cdot u_{l,j} + u_{l+1,j}}{\Delta x^2} + \frac{u_{l,j-1} - 2 \cdot u_{l,j} + u_{l,j+1}}{\Delta y^2} = -\rho_{l,j}$$

Rozwiązywanie równania Poissona przy użyciu FFT (fast Poisson solver)

Przypomnienie → FFT w 1D:

u_l - wartości potencjału w węzłach sieci prostej

$\widehat{u}_l$ -wartości potencjału w węzłach sieci odwrotnej (**transformata**)

$$k_m = \frac{2\pi}{N \cdot h} m \qquad \begin{matrix} x_l = h \cdot l \\ y_j = h \cdot j \end{matrix} \qquad \begin{matrix} l, n = 0, 1, \ldots, N-1 \\ j, m = 0, 1, \ldots, M-1 \end{matrix}$$

$$1D \Longrightarrow u_l = \frac{1}{N} \sum_{n=0}^{N-1} \widehat{u}_n e^{ik_n x_l} = \frac{1}{N} \sum_{n=0}^{N-1} \widehat{u}_n e^{i \frac{n \cdot 2\pi}{Nh} hl} = \frac{1}{N} \sum_{n=0}^{N-1} \widehat{u}_n e^{i \frac{2\pi nl}{N}}$$

$$2D \Longrightarrow u_{l,j} = \frac{1}{NM} \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} \widehat{u}_{n,m} e^{i \frac{2\pi nl}{N}} e^{i \frac{2\pi mj}{M}}$$

$$\rho_{l,j} = \frac{1}{NM} \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} \widehat{\rho}_{n,m} e^{i \frac{2\pi nl}{N}} e^{i \frac{2\pi mj}{M}}$$

Po wstawieniu transformat odwrotnych do zdyskretyzowanego równania Poissona:

$$\frac{u_{l-1,j} - 2 \cdot u_{l,j} + u_{l+1,j}}{\Delta x^2} + \frac{u_{l,j-1} - 2 \cdot u_{l,j} + u_{l,j+1}}{\Delta y^2} = -\rho_{l,j}$$

dla $\Delta x = \Delta y = h$ otrzymujemy

$$\widehat{u}_{n,m} \left(e^{i\frac{2\pi n(-1)}{N}} - 2 + e^{i\frac{2\pi n(+1)}{N}} + e^{i\frac{2\pi m(-1)}{M}} - 2 + e^{i\frac{2\pi m(+1)}{M}} \right) = \widehat{\rho}_{n,m} \cdot h^2$$

korzystając z relacji $e^{ix} + e^{-x} = 2cos(x)$

otrzymamy

$$\widehat{u}_{nm} = \frac{\widehat{\rho}_{n,m} h^2}{2\left(cos\frac{2\pi n}{N} + cos\frac{2\pi m}{M} - 2\right)}$$

→ rozwiązanie uzyskamy wykonując transformację odwrotną (poprzedni slajd)
→ jest ono periodyczne

$$u_{l,j} = u_{l+N,j} = u_{l,j+M}$$

→ czy istnieje problem dla m=n=0? (zero w mianowniku) –
nie ponieważ przyczynek od k_m=k_n=0 pochodzi od funkcji stałej,
którą możemy dodać do naszego rozwiązania (tak aby spełniało np. WB)

Warunki brzegowe:

→ **jednorodne WB Dirichleta** $u_{l,j\Gamma} = 0$ wówczas stosujemy transformację sinusową
(u=0 przy l=0,N oraz j=0,M)

$$\hat{\rho}_{n,m} = \sum_{l=1}^{N-1} \sum_{j=1}^{M-1} \rho_{l,j} \sin\left(\frac{\pi n l}{N}\right) \sin\left(\frac{\pi m j}{M}\right)$$

Stosując zdyskretyzowaną postać rów. Poissona korzystamy z relacji

$$sin(x+y) + sin(x-y) = 2sin(x)cos(y)$$

co prowadzi do wyrażenia

$$\widehat{u}_{nm} = \frac{\widehat{\rho}_{n,m} h^2}{2\left(cos\frac{\pi n}{N} + cos\frac{\pi m}{M} - 2\right)}$$

Po obliczeniu wszystkich $\widehat{u}_{n,m}$ dokonujemy transformacji odwrotnej potencjału

$$u_{l,j} = \frac{2}{N}\frac{2}{M} \sum_{n=1}^{N-1} \sum_{m=1}^{M-1} \hat{u}_{n,m} \sin\left(\frac{\pi n l}{N}\right) \sin\left(\frac{\pi m j}{M}\right)$$

Warunki brzegowe cd.:

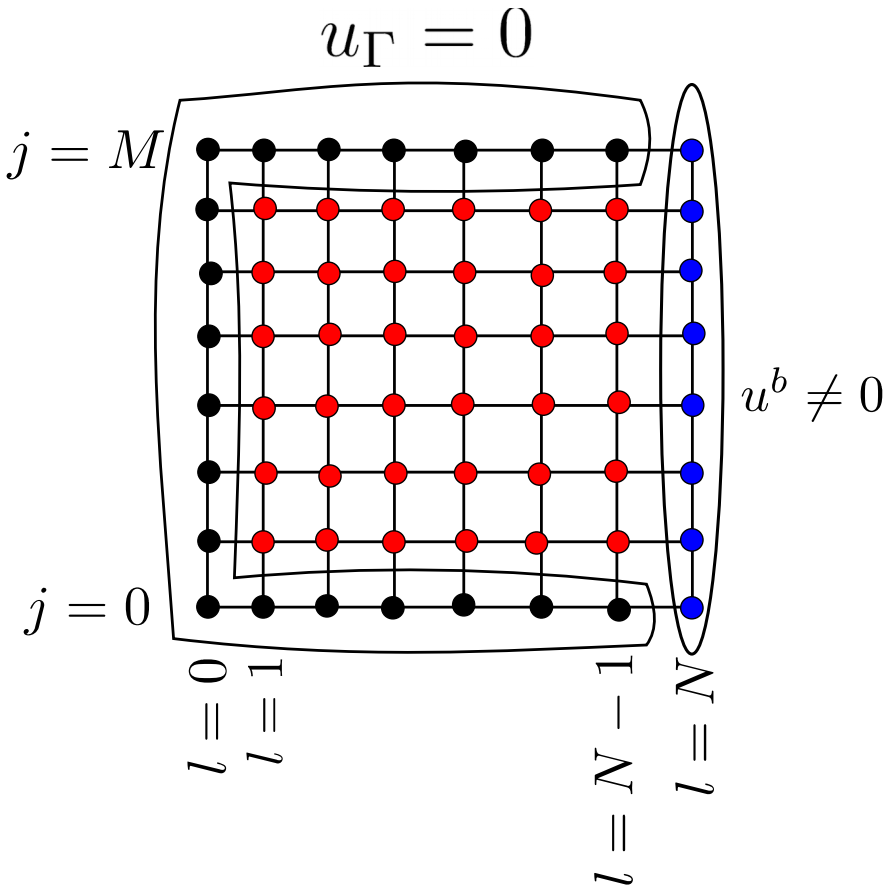
→ **niejednorodne Dirichleta** np. $u(x_{max}, y) = u^b(y)$ i jednorodne w pozostałejczęści brzegu

Rozwiązanie można zapisać w postaci (zasada superpozycji)

$$u = u' + u^b$$

Z warunkami

$$u'_\Gamma = 0 \qquad u^b_\Gamma = u^b(y)|_{x=x_{max}}$$
$$u'_\Omega \neq 0 \qquad u'_\Omega = 0$$



Po wstawieniu u do zdyskretyzowanego r. Poissona dostaniemy

$$\nabla^2 u' = -\nabla^2 u^b - \rho$$

$$u'_{l-1,j} - 2u'_{l,j} + u'_{l+1,j} + u'_{l,j-1} - 2u'_{l,j} + u'_{l,j+1} = \dots$$
$$\dots = -\left(u^b_{l-1,j} - 2u^b_{l,j} + u^b_{l+1,j} + u^b_{l,j-1} - 2u^b_{l,j} + u^b_{l,j+1}\right) - h^2\rho_{l,j}$$

Wkład od u_b pochodzi tylko od węzłów l=N-1 leżących na prawym brzegu
(zakładamy że poza brzegiem u_b=0)

$$u^b_{l,j} = u^b_j \cdot \delta_{l,N-1}$$

$$u'_{l-1,j} + u'_{l+1,j} + u'_{l,j-1} + u'_{l,j+1} - 4u'_{l,j} = -u^b_j \cdot \delta_{l,N-1} - h^2\rho_{l,j}$$

Czyli modyfikujemy tylko prawą stronę równania (gęstość) – dalej postępujemy jak w przypadku warunków jednorodnych Dirichleta.

Warunki brzegowe cd.:

→ **warunki von Neumanna**

$$\left.\frac{\partial u}{\partial \boldsymbol{n}}\right|_{\Gamma} = C_{\Gamma}(\boldsymbol{r})$$

→ dla **jednorodnych warunków v.N.** stosujemy transformację kosinusową, która automatycznie spełnia WB

$$\left.\frac{\partial u}{\partial \boldsymbol{n}}\right|_{\Gamma} = 0$$

$$\widehat{u}_{nm} = \frac{\widehat{\rho}_{n,m} h^2}{2\left(cos\frac{2\pi n}{N} + cos\frac{2\pi m}{M} - 2\right)}$$

$$u_{l,j} = \frac{2}{NM} \sum_{n=0}^{N} \sum_{m=0}^{M} \widehat{u}_{n,m} cos\left(\frac{\pi n l}{N}\right) cos\left(\frac{\pi m j}{M}\right) \cdot \boxed{C_n \cdot C_m}$$

$$\mathrm{C}_p = \begin{cases} 1 & p \in \Omega \\ \frac{1}{2} & p \in \Gamma \end{cases}$$

→ **niejednorodne WB. Neumana**

$$\left.\frac{\partial u}{\partial \boldsymbol{n}}\right|_{\Gamma} \neq 0$$

np. na prawym brzegu

$$\left.\frac{\partial u}{\partial x}\right|_{x_{N-1}} = C(y)$$

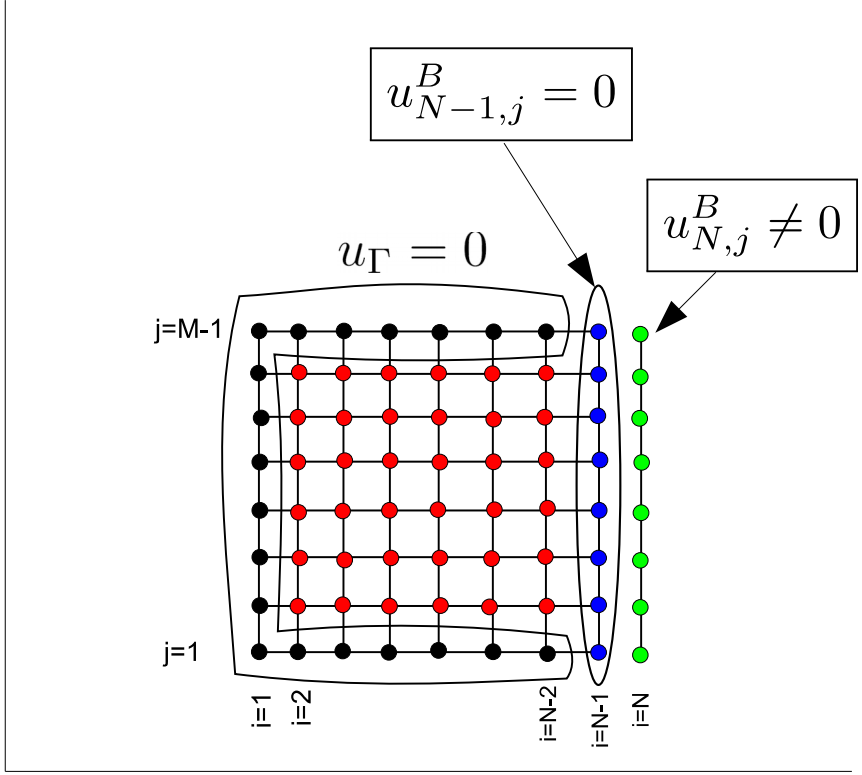
Znowu zakładamy postać rozwiązania na brzegu

$$u = u' + u^B$$

z warunkami:

$$\begin{cases} \left.\frac{\partial u'}{\partial x}\right|_{x_{N-1}} = 0 \\ u'_{N-1} \neq 0 \end{cases}$$

niezerowy wkład do wartości
+ znikanie pochodnej
= jednorodne WB v. Neumanna
= **transformacja kosinusowa**



Znika wartość na brzegu, ale daje wkład do pochodnej

$$\left.\frac{\partial u^B}{\partial x}\right|_{x_{N-1}} = C_j$$

$$u^B_N \neq 0$$

$$u^B_{N-2}, u^B_{N-1} = 0$$

$$\frac{u^B_{N,j} - u^B_{N-2,j}}{2h} = \frac{u^B_{N,j}}{2h} \rightarrow \boxed{u^B_{N,j} = 2hC_j}$$

Do równania różnicowego

$$u'_{l-1,j} - 2u'_{l,j} + u'_{l+1,j} + u'_{l,j-1} - 2u'_{l,j} + u'_{l,j+1} = \dots$$
$$\dots = -\left(u^b_{l-1,j} - 2u^b_{l,j} + u^b_{l+1,j} + u^b_{l,j-1} - 2u^b_{l,j} + u^b_{l,j+1}\right) - h^2\rho_{l,j}$$

wstawiamy dla **l=N-1**
(pozostałe wyrazy =0)

$$u^B_{N,j} = 2hC_j$$

I otrzymujemy równanie ze zmodyfikowaną prawą stroną
dla **transformacji kosinusowej:**

$$u'_{l-1,j} + u'_{l+1,j} + u'_{l,j-1} + u'_{l,j+1} - 4u'_{l,j} = \underbrace{2\,h\,C_j\,\delta_{N-1,l} - h^2\rho_{l,j}}_{f'_{l,j}}$$

Zalety:

→ metoda szybka (jedynie FMG może być szybsza)
i bezpośrednia (daje wynik po skończonej liczbie operacji)

→ dobrze działa na siatce prostokątnej

→ transformacje wykonywane na jednej tablicy potencjału
(potrzebujemy jeszcze 2 tablicę dla prawej strony RRCz)

Wady:

→ zastosowanie tylko do obszarów o regularnych kształtach:
prostopadłościan, sfera, cylinder

→ nie uwzględnia powierzchni brzegowej w środku np. elektrody włożone do obszaru, w którym szukamy potencjału

Biblioteki: Math Kernel Library (Fortran, C/C++) → Fast Poisson Solver
(wykorzystuje pakiet FFTW)

Jeszcze o metodzie bezpośredniej rozwiązywania r. Poissona.

Skorzystajmy ze zdyskretyzowanej postaci równania w 1D

$$\frac{u_{l-1} - 2 \cdot u_l + u_{l+1}}{h^2} = -\rho_l$$

→ generuje ono jedno równanie dla ustalonej wartości l.

Zapisując je dla l=0,2,3,...,N-1 otrzymamy układ równań

przykład

$$A\boldsymbol{u} = \boldsymbol{b}$$

$$\begin{bmatrix} 2 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 2 \end{bmatrix} \cdot \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} = \begin{bmatrix} \Delta^2\rho_1 \\ \Delta^2\rho_2 \\ \Delta^2\rho_3 \\ \Delta^2\rho_4 \end{bmatrix}$$

→ **Jak wprowadzić warunki brzegowe?**

Uwzględniamy warunki brzegowe:

$$u_i = C_i$$

→ **Dirichleta** (3 kroki)

$$1)\; a_{i,j} = 0, \;\text{dla}\; j = 1, \ldots, N$$

$$2)\; a_{i,i} = 1$$

$$3)\; b_i = C_i$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 1 \end{bmatrix} \cdot \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} = \begin{bmatrix} C_1 \\ \Delta^2\rho_2 \\ \Delta^2\rho_3 \\ h\,D_N \end{bmatrix}$$

→ **von Neumanna**

np.: na prawym brzegu

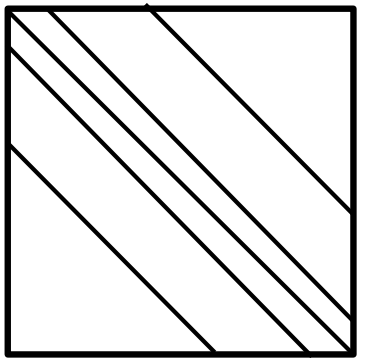
$$\left.\frac{du}{dx}\right|_{x=x_N} = \frac{u_N - u_{N-1}}{h} = D_N$$

$$-u_{N-1} + u_N = h\,D_N$$

Postać macierzowa operatora Laplace'a w 2D i 3D

$$\nabla^2 u(x,y) = -\rho(x,y)$$

A=



2D

$$x = x_i, \quad i = 1, 2, \ldots, n_x$$
$$y = y_i, \quad j = 1, 2, \ldots, n_y$$

$$u_{i-1,j} + u_{i+1,j} + u_{i,j-1} + u_{i,j+1} - 4u_{i,j} = -\Delta^2 \rho_{i,j}$$

reindeksacja węzłów

$$l = i + (j-1) \cdot n_x, \quad l = 1, 2, \ldots, N, \quad N = n_x \cdot n_y$$

5 przekątnych:

$$u_{l-n_x} + u_{l-1} - 4\,u_l + u_{l+1} + u_{l+n_x} = -\Delta^2 \rho_l$$

3D (analogicznie)

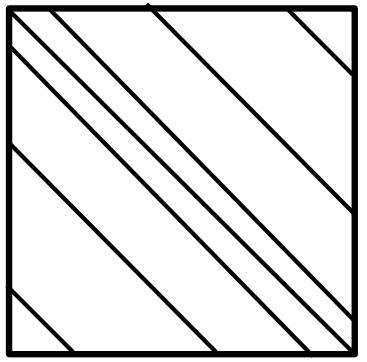
$$z = z_i, \quad k = 1, 2, \ldots, n_z$$

$$l = i + (j-1) \cdot n_x + (k-1) \cdot n_x \cdot n_y, \quad l = 1, 2, \ldots, N, \quad N = n_x \cdot n_y \cdot n_z$$

7 przekątnych:

$$u_{l-n_x n_y} + u_{l-n_x} + u_{l-1} - 6\,u_l + u_{l+1} + \\ + u_{l+n_x} + u_{l+n_x n_y} = -\Delta^2 \rho_l$$

A=



Kiedy opłaca się rozwiązywać rów. Poissona znajdując rozwiązanie Ax=b?

→ tylko wtedy gdy używamy algorytmów dla macierzy rzadkich

→ w 1D UARL rozwiążemy przy użyciu LU dla macierzy trójdiagonalnych
z nakładem ~O(n)

→ w 2D UARL też rozwiążemy stosując LU ale dla macierzy pięcioprzekatniowych

→ w 3D można stosować

i) LU jeśli liczba węzłów jest mała

ii) iLU jako preconditioner dla metod iteracyjnych (BiCGStab,GMRES, TFQMR itp.)

Metody iteracyjne

Punktem wyjścia jest układ równań
(jeszcze w postaci macierzowej)

$$A\boldsymbol{x} = \boldsymbol{b}$$

Który rozwiązywać będziemy iteracyjnie

$$\boldsymbol{x}^{k+1} = M\boldsymbol{x}^{k} + \boldsymbol{c}$$

M jest macierzą iteracji, która posiada wartości i wektory własne

$$M\boldsymbol{v}_i = \lambda_l \boldsymbol{v}_i$$

$$\rho(M) = max\{|\lambda_1|, |\lambda_2|, \ldots, |\lambda_n|\}$$

Metoda iteracyjna jest zbieżna wtedy i tylko wtedy,
gdy promień spektralny spełnia warunek

$$\rho(M) < 1$$

Co dają metody iteracyjne:

→ jeśli M jest macierzą rzadką to koszt jednej iteracji jest rzędu O(n), dla pełnej macierzy $O(n^2)$

→ jeśli rozwiązanie startowe jest „bliskie” dokładnemu to ilość iteracji może być mała (rel. wielosiatkowe)

→ zazwyczaj prosta konstrukcja macierzy M, a w przypadku rów. Poissona nie trzeba jej nawet tworzyć (zysk w postaci ograniczenia zajętej pamięci)

→ jak w każdej procedurze iteracyjnej mogą jednak wystąpić problemy ze zbieżnością, zbieżność silnie zależy od **postaci macierzy iterującej/schematu relaksacyjnego**

Zbieżność metody iteracyjnej

Założenia:

$$A\boldsymbol{x}_{dok} = \boldsymbol{b}$$

$$\boldsymbol{x}^{k+1} = M\boldsymbol{x}^{k} + \boldsymbol{c}$$

$$\rho(M) < 1$$

Dla przepisu iteracyjnego mamy

$$\boldsymbol{x}^{k} = \boldsymbol{x}_{dok} + \boldsymbol{e}^{k}$$

błąd w k-tej iteracji

$$\boldsymbol{x}^{k+1} = \boldsymbol{x}_{dok} + \boldsymbol{e}^{k+1} = \underbrace{M(\boldsymbol{x}_{dok} + \boldsymbol{e}^{k})}_{(M\boldsymbol{x}_{dok} + \boldsymbol{c}) + M\boldsymbol{e}^{k}} + c$$

$$\boldsymbol{e}^{k+1} = M\boldsymbol{e}^{k}$$

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

Mamy dostępną bazę w postaci wektorów własnych M:

$$M\boldsymbol{v}_i = \lambda_l \boldsymbol{v}_i \qquad \boldsymbol{e}^{0} = \sum_i c_i \boldsymbol{v}_i$$

$$\boldsymbol{e}^{k+1} = M^{k+1}\boldsymbol{e}^{0} = \sum_i c_i \lambda_i^{k+1} \boldsymbol{v}_i \quad \Longrightarrow \quad \lim_{k\to\infty} \boldsymbol{e}^{k+1} = \boldsymbol{0}$$

Metoda powinna być zbieżna, ale jaka będzie szybkość zbieżności do rozwiązania dokładnego? → zależy od postaci M

Konstrukcja macierzy iterującej

Założenia: 1) $A\boldsymbol{x} = \boldsymbol{b} \quad \rightarrow \quad \boldsymbol{x}^{k+1} = M\boldsymbol{x}^{k} + \boldsymbol{c}$

2) zbieżność: $\rho(M) < 1$

3) tempo zbieżności $R_{\infty} = -log_{10}\left(\rho(M)\right)$

$$A = B + C$$
$$(B + C)\boldsymbol{x} = b$$
$$B\boldsymbol{x} = \boldsymbol{b} - C\boldsymbol{x} \quad B^{-1} \cdot /$$
$$\boldsymbol{x} := -B^{-1}C\boldsymbol{x} + B^{-1}\boldsymbol{b}$$
$$M = -B^{-1}C, \quad \boldsymbol{c} = B^{-1}\boldsymbol{b}$$

Metoda Jakobiego: $A = D + L + U$

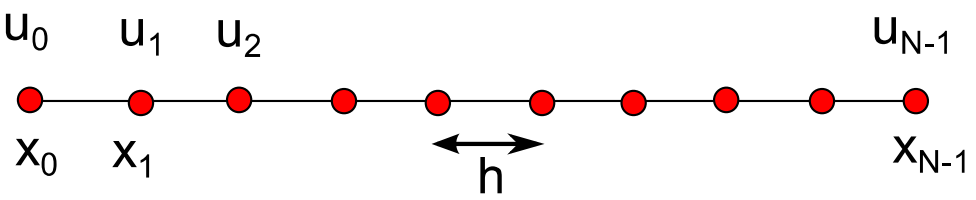
diagonala

macierz trójkątna dolna

macierz trójkątna górna

$$\boxed{B = D, \quad C = L + U}$$

$$u_{i-1} - 2 \cdot u_i + u_{i+1} = -\rho_i \cdot h^2$$



$$A_{6\times 6} = \begin{bmatrix} 2 & -1 & 0 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 & 0 \\ 0 & 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & 0 & -1 & 2 \end{bmatrix}$$

$$\text{M=-D}^{-1}(L+U) = \begin{bmatrix} 0 & 1/2 & 0 & 0 & 0 & 0 \\ 1/2 & 0 & 1/2 & 0 & 0 & 0 \\ 0 & 1/2 & 0 & 1/2 & 0 & 0 \\ 0 & 0 & 1/2 & 0 & 1/2 & 0 \\ 0 & 0 & 0 & 1/2 & 0 & 1/2 \\ 0 & 0 & 0 & 0 & 1/2 & 0 \end{bmatrix}$$

$$\text{B=D}^{-1} = \begin{bmatrix} -\frac{1}{2} & 0 & 0 & 0 & 0 & 0 \\ 0 & -\frac{1}{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{1}{2} & 0 & 0 & 0 \\ 0 & 0 & 0 & -\frac{1}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & -\frac{1}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & -\frac{1}{2} \end{bmatrix}$$

Przepis iteracyjny (macierzowy):

$$\boldsymbol{x}^{k+1} = -D^{-1}(L+U)\boldsymbol{x}^k + D^{-1}\boldsymbol{b}$$

$$\text{C=L+U=} \begin{bmatrix} 0 & -1 & 0 & 0 & 0 & 0 \\ -1 & 0 & -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 & 0 & -1 \\ 0 & 0 & 0 & 0 & -1 & 0 \end{bmatrix}$$

Przepis dla **pojedynczego węzła siatki**
→ to **wzór relaksacyjny**:

$$x_i^{k+1} = \frac{1}{2}\left(x_{i-1}^k + x_{i+1}^k\right) + \frac{1}{2}\rho_i \cdot h^2$$

Otrzymaliśmy wzór relaksacyjny dla metody Jakobiego

$$x_i^{k+1} = \frac{1}{2}\left(x_{i-1}^k + x_{i+1}^k\right) + \frac{1}{2}\rho_i \cdot h^2$$

→ nie musimy już konstruować macierzy
(ta ma prostą konstrukcję, a jej współczynniki są zawarte w równaniu)

→ w obliczeniach **musimy** użyć dwóch tablic/wektorów, dla starego i nowego rozwiązania,
metoda Jakobiego to relaksacja globalna

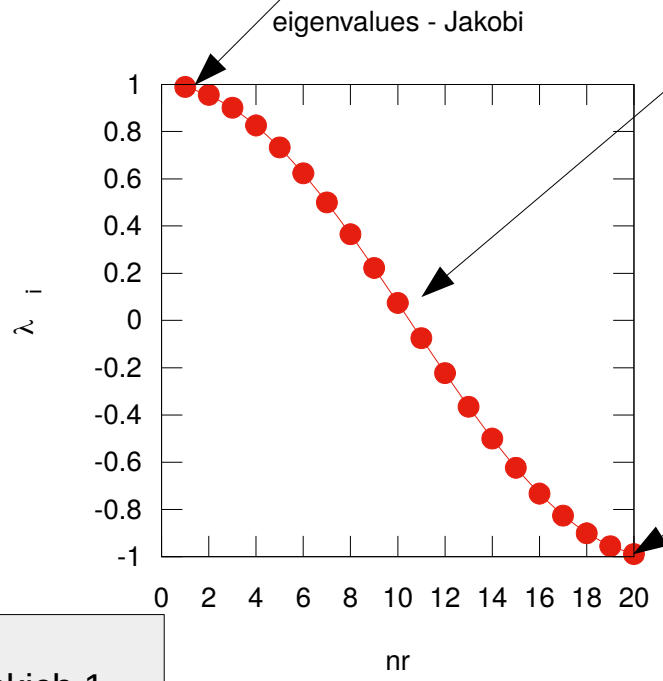
→ jaka jest zbieżność tej metody?

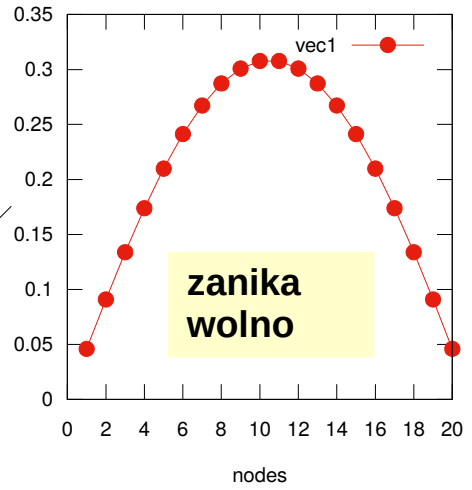
Przykład:

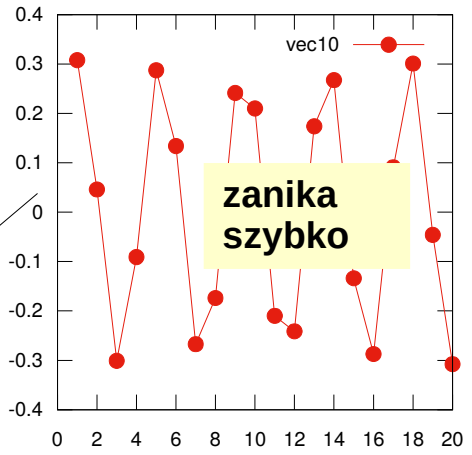
20 węzłów, i=1,2,...,20

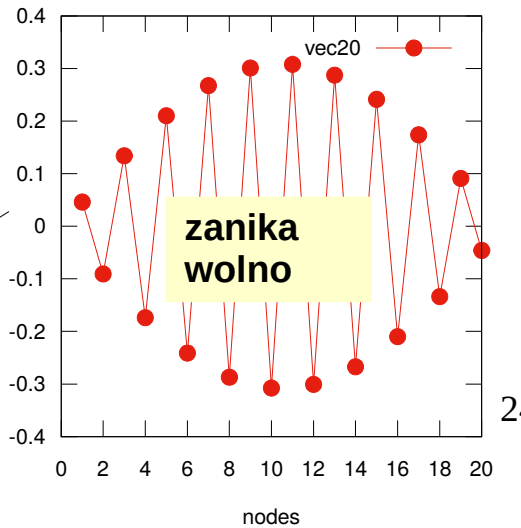
$$\lambda_k = cos\left(\frac{k\pi}{n+1}\right)$$

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









i) metoda Jakobiego jest zbieżna
ii) są wartości własne o modułach bliskich 1
→ metoda jest wolno zbieżna

Zbieżność metody Jakobiego

Tempo zbieżności (eliminacji błędu)

$$\mathrm{e}^{k+1} = \sum_i c_i \lambda_i^{k+1} \boldsymbol{v}_i$$

zależy od promienia spektralnego M

$$k = 1 \Longrightarrow \rho(M) = cos\left(\frac{\pi}{N+1}\right) \approx 1 - \left(\frac{\pi}{N+1}\right)^2 + \ldots$$

→ im więcej węzłów N tym bardziej promień spektralny zbliża się do 1

→ zagęszczanie siatki spowoduje spowolnienie działania metody Jakobiego

Algorytm

$$\begin{aligned}
&for\, k = 1\, to\, K_{max}\, do \\
&\quad for\, i = 2\, to\, N-1\, do \\
&\qquad x_i^{k+1} = \frac{1}{2}(x_{i-1}^k + x_{i-1}^k) + \frac{1}{2}\rho_i h^2 \\
&\quad end\, do \\
&\quad \boldsymbol{x}^k \leftarrow \boldsymbol{x}^{k+1} \\
&end\, do
\end{aligned}$$

Metoda Gaussa-Seidla

$$B = L + D \qquad C = U$$

$$(L+D)\boldsymbol{x}^{k+1} = \boldsymbol{b} - U\boldsymbol{x}^{k} \quad \rightarrow \quad \boldsymbol{x}^{k+1} = -D^{-1}U\boldsymbol{x}^{k} - D^{-1}L\boldsymbol{x}^{k+1} + D^{-1}\boldsymbol{b}$$

$$M = -(L+D)^{-1}U$$

Dla laplasjanu (po wymnożeniu przez D^{-1}) otrzymamy:

$$\boldsymbol{x}^{k+1} = \frac{1}{2}U\boldsymbol{x}^{k} + \frac{1}{2}L\boldsymbol{x}^{k+1} - \frac{1}{2}\boldsymbol{b}$$

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

Średnia arytmetyczna: sąsiada z prawej strony (**k**) oraz z lewej strony (**k+1**)

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

Wzór iteracyjny w metodzie GS

$$\boldsymbol{x}^{k+1} = \frac{1}{2}U\boldsymbol{x}^{k} + \frac{1}{2}L\boldsymbol{x}^{k+1} - \frac{1}{2}\boldsymbol{b}$$

→ Ze względu na jednoprzekątniową postać L i U wzór działa jak podstawienie. Pozbywając się macierzy, dostaniemy wzór relaksacyjny.

→ wyliczając nową wartość w węźle i-tym korzystamy z wartości w węźle (i-1) już zmienionym w bieżącej Iteracji

→ **metoda GS to relaksacja lokalna**
(rozwiązanie znajdziemy dysponując tylko jednym wektorem rozwiązań)

Algorytm GS

$$\begin{aligned}
&for\ k = 1\ to\ K_{max}\ do\\
&\quad for\ i = 2\ to\ N-1\ do\\
&\qquad x_i = \frac{1}{2}(x_{i-1} + x_{i-1}) + \frac{1}{2}\rho_i h^2\\
&\quad end\ do\\
&end\ do
\end{aligned}$$

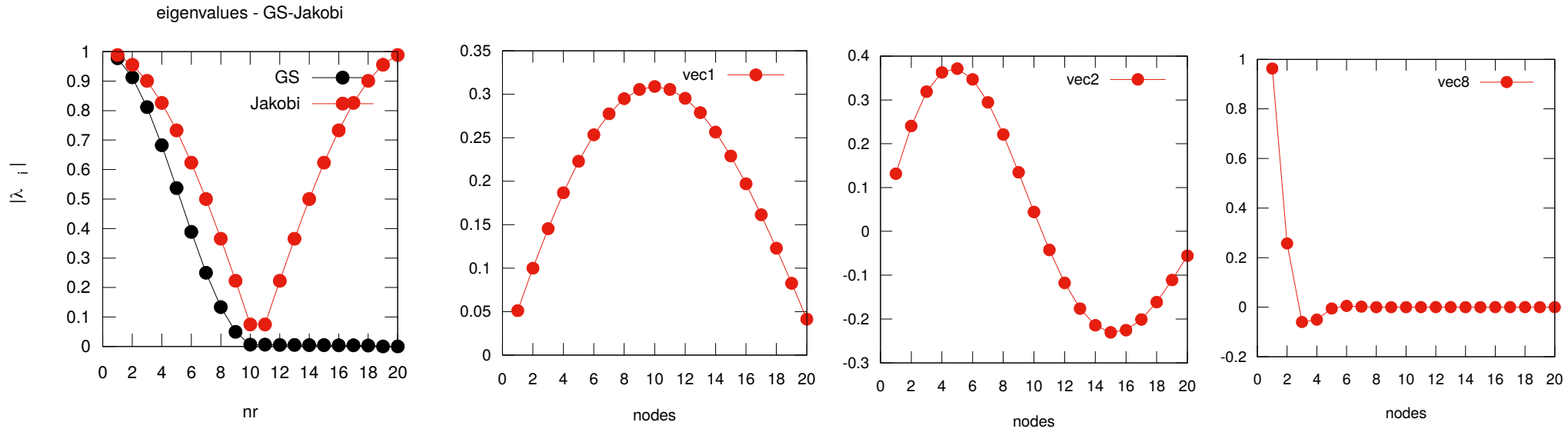
Przykład. Macierz iteracji w metodzie GS

$$M = -(L+D)^{-1}U$$

$$\mathrm{M}_{6\times 6} = \begin{pmatrix}
0 & \frac{1}{2} & 0 & 0 & 0 & 0\\
0 & \frac{1}{4} & \frac{1}{2} & 0 & 0 & 0\\
0 & \frac{1}{8} & \frac{1}{4} & \frac{1}{2} & 0 & 0\\
0 & \frac{1}{16} & \frac{1}{8} & \frac{1}{4} & \frac{1}{2} & 0\\
0 & \frac{1}{32} & \frac{1}{16} & \frac{1}{8} & \frac{1}{4} & \frac{1}{2}\\
0 & \frac{1}{64} & \frac{1}{32} & \frac{1}{16} & \frac{1}{8} & \frac{1}{4}
\end{pmatrix}$$

Jakie ma wartości i wektory własne?

Wartości i wektory własne macierzy iteracji M w metodzie Guassa-Seidla
(dla laplasjanu w 1D), n=20



→ wartości własne są nieujemne

→ wektory własne odpowiadające najwyższym wartościom własnym są wolnozmienne (w m. Jakobiego wektory te były szybko- i wolno-zmienne)

→ **m. Gaussa-Seidla ma własności wygładzające błąd**
(**cechę tę wykorzystujemy w relaksacji wielosiatkowej**)

→ promień spektralny M w metodzie GS jest nieznacznie mniejszy niż w m. Jakobiego, ale dla k=1000 .....

$$\rho_J = 0.98883, \quad \rho_J^{1000} = 1.3 \cdot 10^{-5}$$

$$\rho_{GS} = 0.97779, \quad \rho_j^{1000} = 1.76 \cdot 10^{-10}$$

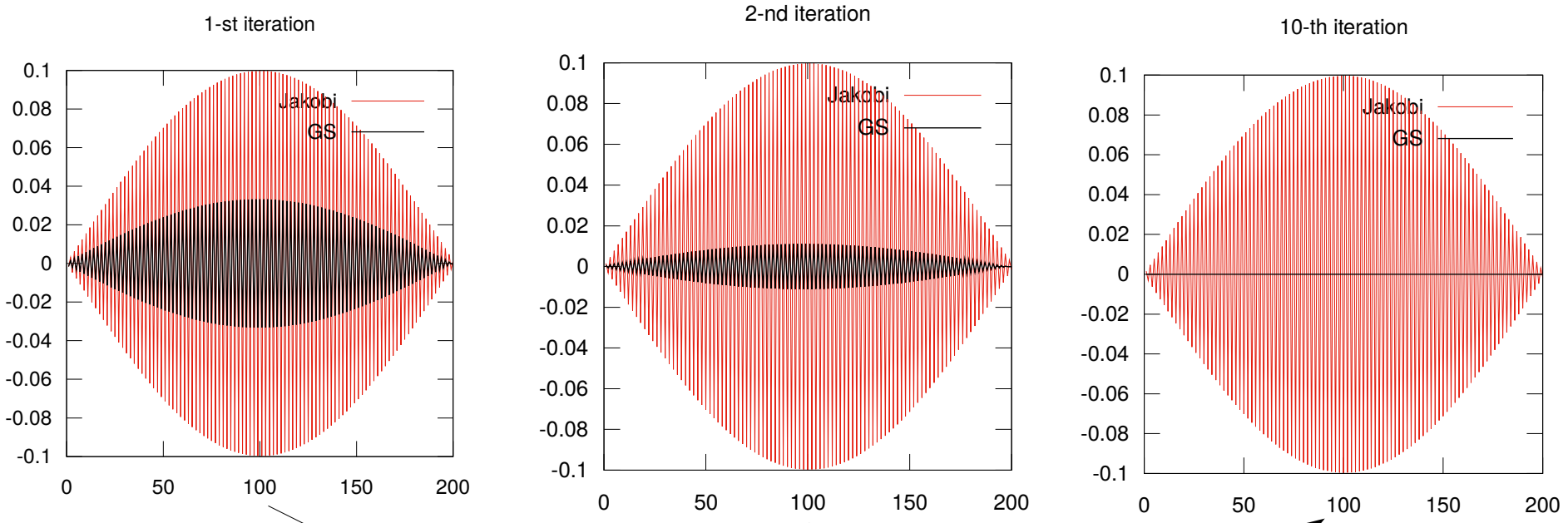
Co oznacza zwrot: „własności wygładzające” w praktyce?

Przykład.
Porównanie szybkości tłumienia błędu w metodach GS i Jakobiego w przypadku równania Laplace'a (brak niejednorodności).
Generujemy wektor szybko-zmienny (najmniejsza wartość własna macierzy iterującej Jakobiego), które stanowi rozwiązanie początkowe

$$\nabla^2 x = 0$$

$$n = 0, \quad WB: \; x_0 = x_{N+1} = 0$$

$$v_k(i) = sin\left(\frac{ik\pi}{N+1}\right) \Longrightarrow \boxed{\boldsymbol{x}^0 = \boldsymbol{v}^N}$$



→ widać że m. GS bardzo szybko tłumi wysokozmienny błąd

→ M Jakobiego tej własności nie posiada

Tempo zbieżności relaksacji GS możemy kontrolować dodając do równania

$$\boldsymbol{x}^{k+1} = \frac{1}{2}U\boldsymbol{x}^{k} + \frac{1}{2}L\boldsymbol{x}^{k+1} - \frac{1}{2}\boldsymbol{b}$$

parametr zbieżności $\omega \in (0, 2)$

$$\boldsymbol{x}^{k+1} = (1-\omega)\cdot\boldsymbol{x}^{k} + \omega\cdot\left(\frac{1}{2}U\boldsymbol{x}^{k} + \frac{1}{2}L\boldsymbol{x}^{k+1} - \frac{1}{2}\boldsymbol{b}\right)$$

Inaczej: mieszamy stare i nowe rozwiązanie

→ ω<1 **podrelaksacja** (obniżamy tempo zbieżności)

→ ω=1 zwykła relaksacja

→ ω>1 **nadrelaksacja** (tu powinniśmy uzyskać przyśpieszenie obliczeń)

(dowód zbieżności metody GS dla $\omega \in (0, 2)$ pokazany na wykładzie z Metod Numerycznych dla układu równań liniowych)

Pytanie: kiedy zatrzymać relaksację?

Kiedy zatrzymać relaksację?

Jakość rozwiązania równania Poissona możemy sprawdzić na dwa sposoby:

1) badając normę euklidesową wektora reszt

$$A\boldsymbol{x} = \boldsymbol{b}$$
$$\boldsymbol{r} = \boldsymbol{b} - A\boldsymbol{x}$$

2) licząc całkę działania dla pola elektrycznego

$$S = \int_\Omega \frac{1}{2} (\nabla u)^2 - \rho \cdot u \, dx$$

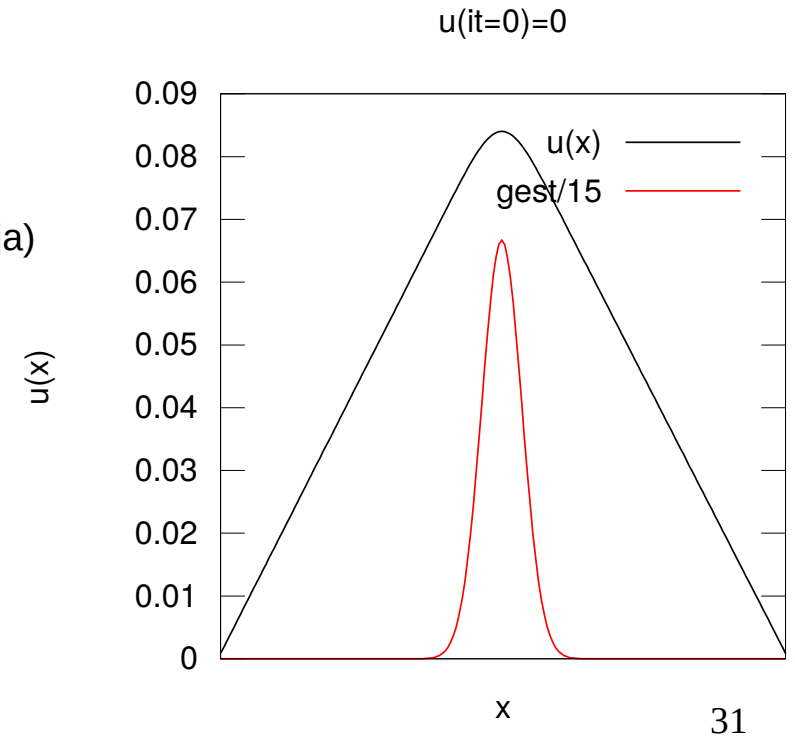
→ **całka działania osiąga minimum dla rozwiązania dokładnego (zasada wariacyjna Rayleigha-Ritza)**

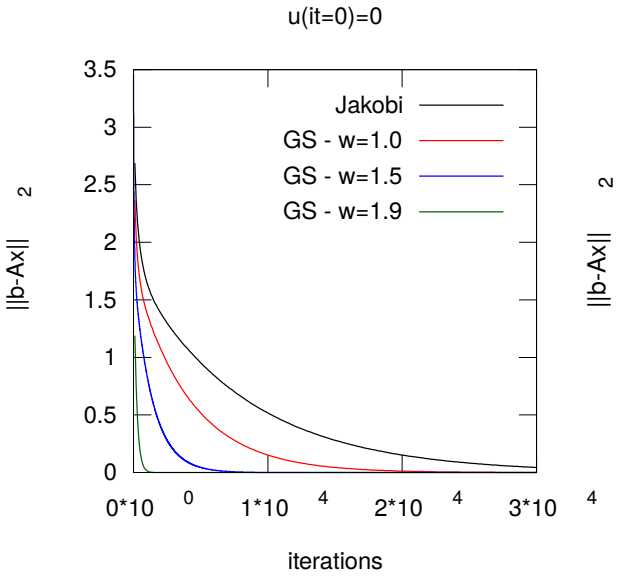
Przykład. Rozwiązujemy metodą relaksacji (Jakobi,GS, nadrelaksacja) równanie Poissona w 1D

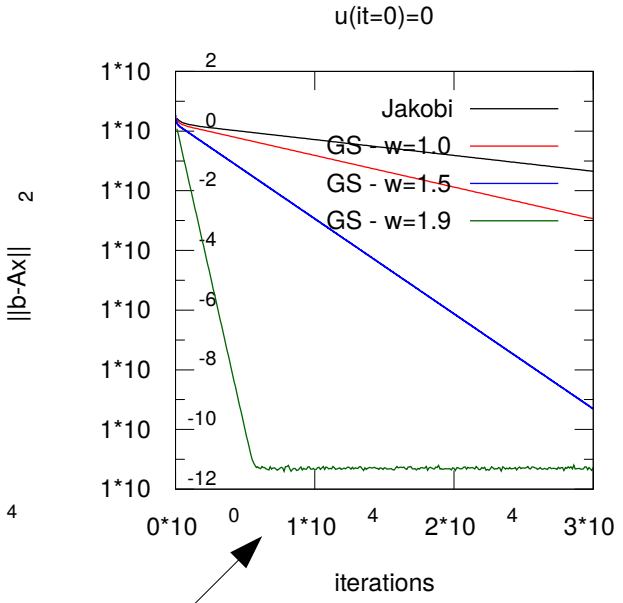
$$\nabla^2 u(x) = -\rho(x), \quad x \in [-1, 1]$$

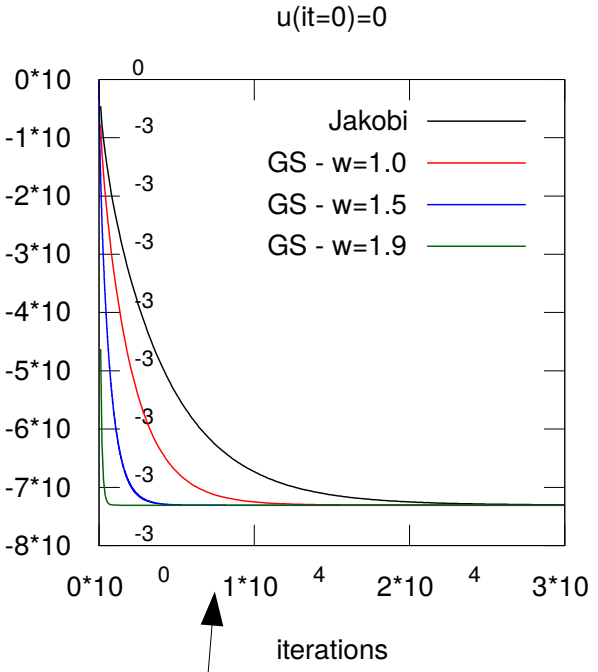
$$WB.: \quad u(-1) = u(1) = 0$$

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









Dla normy wektora reszt pojawia się problem, który jest wynikiem dokładności rozwiązania numerycznego i precyzji obliczeń

→ **trudno określić warunek zakończenia relaksacji**

Jeśli całka działania osiąga minimum to w kolejnych iteracjach już się nie zmieni

→ **to wygodny warunek zatrzymania relaksacji**.

→ m. Jakobiego najwolniejsza

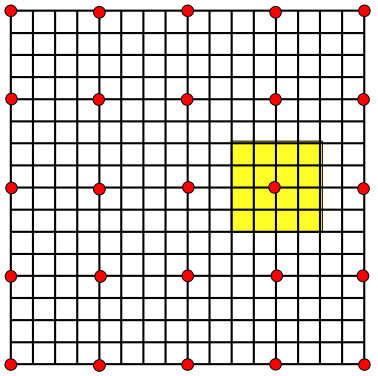
→ nadrelaksacja tym szybsza im większa wartość parametru zbieżności (ale nie jest to regułą – zależy często od WB)

→ co lepsze do warunku zakończenia?: $||r||_2$ czy całka działania?

Jak jeszcze można przyśpieszyć (zoptymalizować) proces relaksacji?
Np. stosując:

1) metodę zagęszczania siatki

2) relaksację wielosiatkową

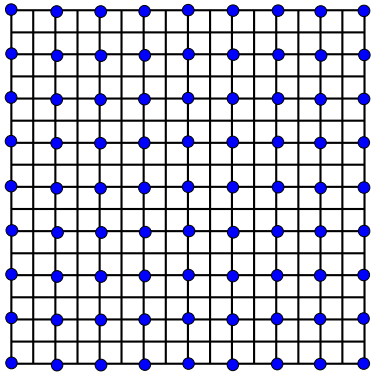


Zagęszczanie siatki – najprostsze.

1) Znajdujemy rozwiązanie na najrzadszej siatce (czerwone węzły, krok na siatce k=4)

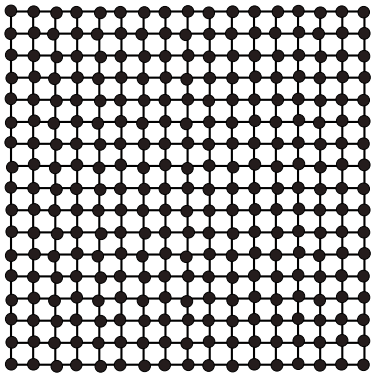
stosując wzór relaksacyjny

$$x_{i,j} = (1-\omega)x_{i,j} + \frac{\omega}{4}\left(x_{i+k,j} + x_{i-k,j} + x_{i,j+k} + x_{i,j-k} + k^2 \cdot h^2 \cdot \tilde{\rho}_{i,j}\right)$$



oraz uśredniając gęstość w komórce otaczającej aktualny węzeł (**żółty obszar**)

$$c_\alpha = \begin{cases} 0.5, & \alpha = \pm k/2 \\ 1, & -k/2 < \alpha < k/2 \end{cases}$$



$$w = \sum_{\alpha=-k/2}^{k/2} \sum_{\beta=-k/2}^{k/2} c_\alpha c_\beta$$

$$\tilde{\rho}_{i,j} = \frac{1}{w} \sum_{\alpha=-k/2}^{k/2} \sum_{\beta=-k/2}^{k/2} c_\alpha c_\beta \rho_{i+\alpha,j+\beta}$$

2) Przechodzimy na siatkę o dwukrotnie mniejszym oczku siatki, wartości w nowych węzłach (niebieskie) interpolujemy liniowo wartościami z węzłów sąsiednich

Siatka rzadka: **czerwone węzły**

Siatka gęsta: **niebieskie węzły**

$$u_{i,j}$$

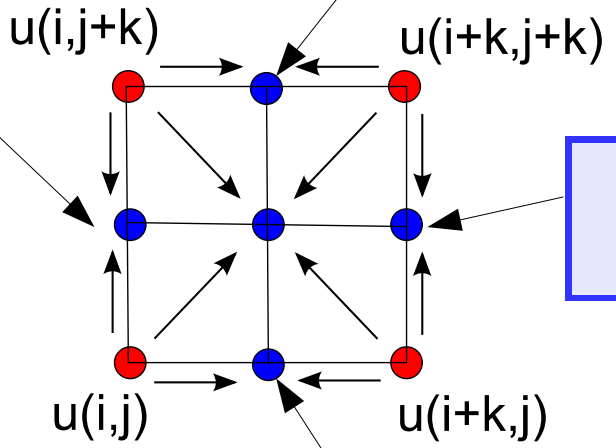
$$u_{i+k,j}$$

$$u_{i,j+k}$$

$$u_{i+k,j+k}$$

$$u_{i+\frac{k}{2},j+k} = \frac{1}{2}\left(u_{i,j+k} + u_{i+k,j+k}\right)$$

$$u_{i,j+\frac{k}{2}} = \frac{1}{2}\left(u_{i,j} + u_{i,j+k}\right)$$



$$u_{i+k,j+\frac{k}{2}} = \frac{1}{2}\left(u_{i+k,j} + u_{i+k,j+k}\right)$$

$$u_{i+\frac{k}{2},j} = \frac{1}{2}\left(u_{i,j} + u_{i+k,j}\right)$$

węzeł środkowy:

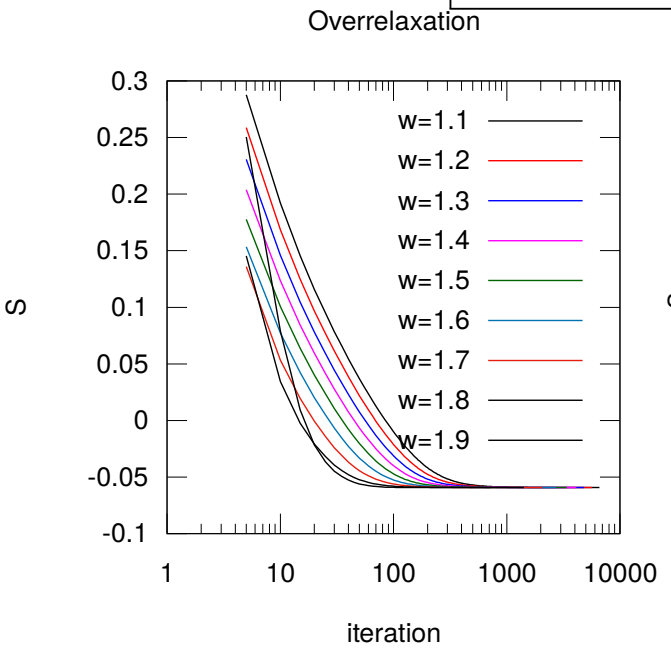
$$u_{i+\frac{k}{2},j+\frac{k}{2}} = \frac{1}{4}\left(u_{i,j} + u_{i+k,j} + u_{i,j+k} + u_{i+k,j+k}\right)$$

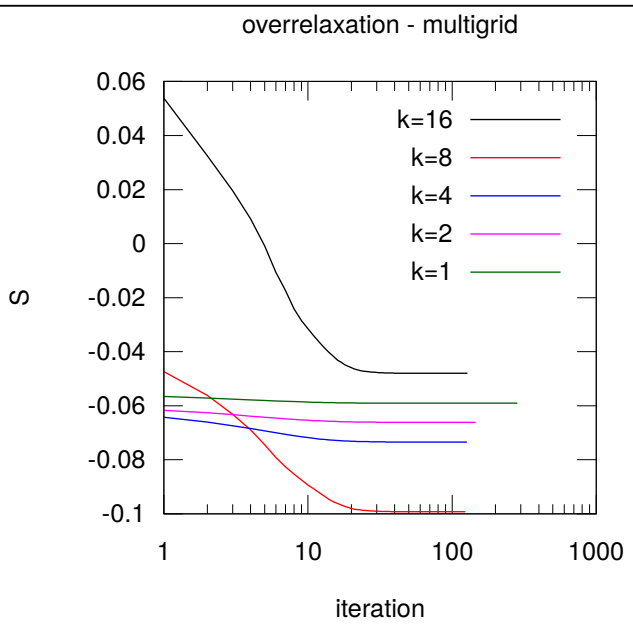
3) na rzadszej siatce relaksujemy równanie

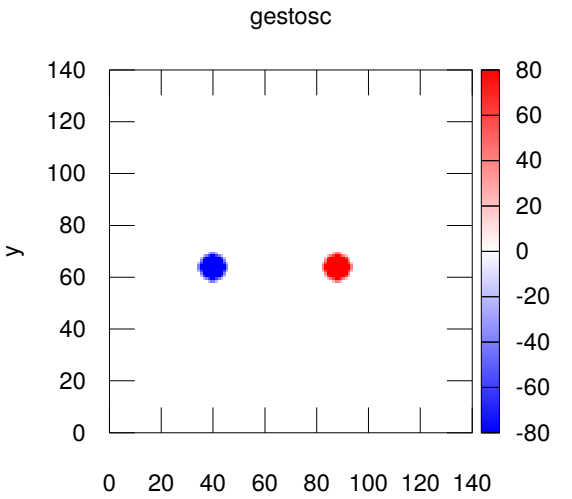
4) powtarzamy operacje (2) i (3) na kolejnych gęstszych siatkach

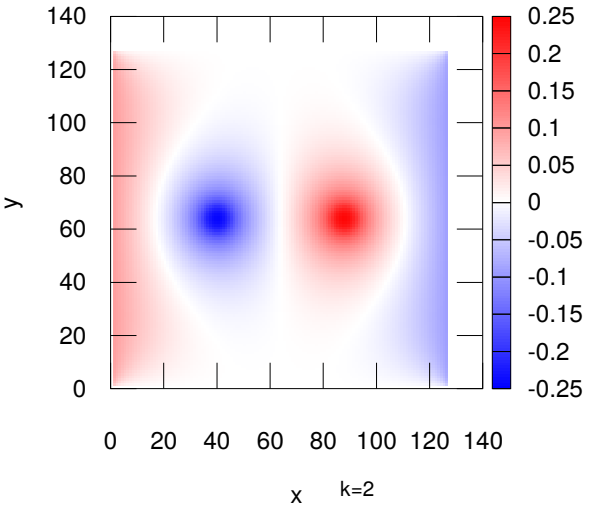
Przykład. $\nabla^2 u(,y) = -\rho(x,y)$

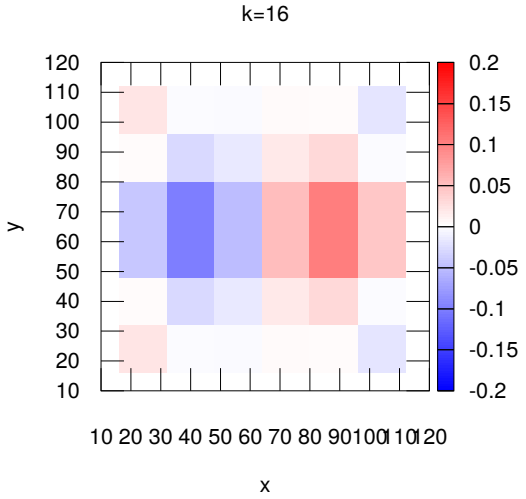
$$u_{0,j} = +0.1,\ u_{N,j} = -0.1,\ u(i,0) = 0,\ u_{i,N} = 0$$

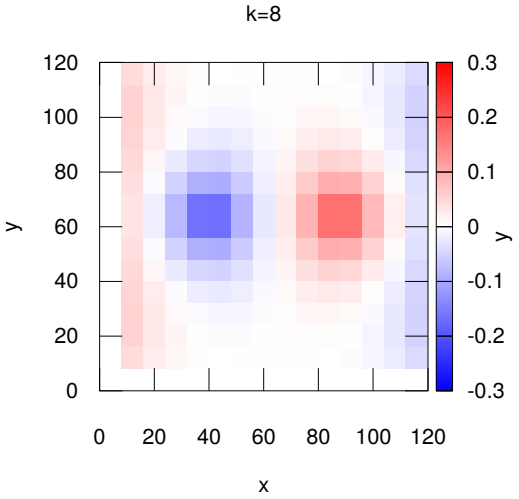


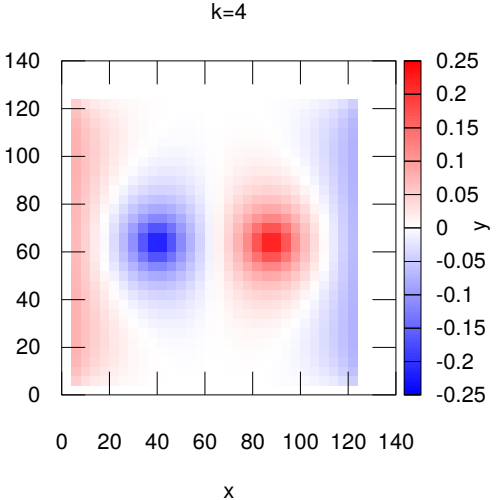


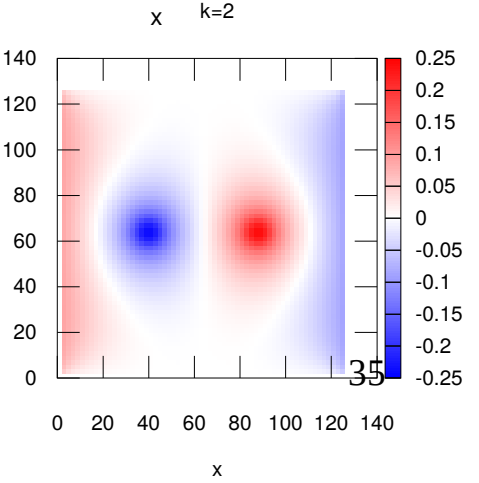




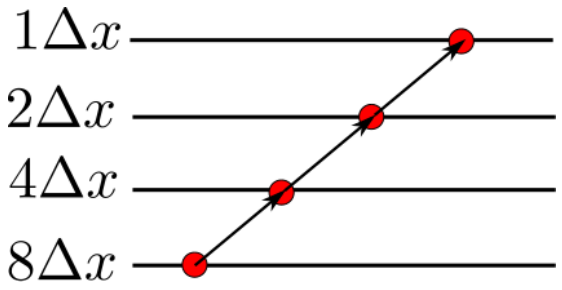








schemat prostego zagęszczania siatki



koniec

na każdej siatce iteracja do pełnej zbieżności

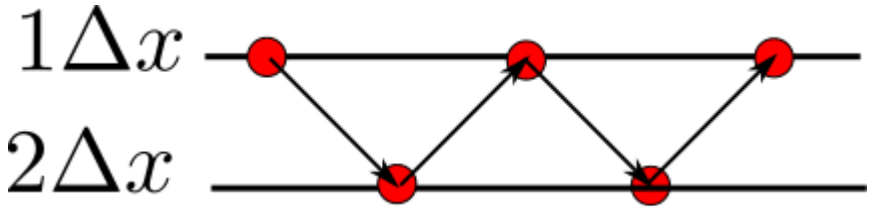
ten czynnik spowalnia metodę

start

Relaksacja wielosiatkowa

Zamiast jednokierunkowego zagęszczania siatki
możemy użyć schematu wielosiatkowego (*multigrid method*)

np. najprostszego dwusiatkowego



tu szukamy rozwiązania

tu szacujemy błąd (wolnozmienny)

na każdej siatce wykonujemy tylko kilka iteracji

Procedura iteracji dwusiatkowej dla równania $A\boldsymbol{x} = \boldsymbol{b} \quad \Longrightarrow \quad \boldsymbol{x}^{k+1} = M\boldsymbol{x}^{k} + \boldsymbol{c}$

1. na siatce (Δx) wykonujemy n_1 iteracji, dostajemy: $\boldsymbol{x}(\Delta x) = \boldsymbol{x}_{dok}(\Delta x) + \boldsymbol{e}(\Delta x)$

2. liczymy wektor reszt (Δx):

$$\boldsymbol{r}(\Delta x) = A\boldsymbol{x}(\Delta x) - \boldsymbol{b} = \underbrace{(A\boldsymbol{x}_{dok} - \boldsymbol{b})}_{=0} + A\boldsymbol{e}(\Delta x)$$

$$A\boldsymbol{e}(\Delta x) = \boldsymbol{r}(\Delta x)$$

Uwaga: wektor błędu e(Δx) zawiera składowe wolnozmienne - znajdźmy go na siatce (2dx)

3. rzutujemy wektor reszt na rzadszą siatkę używając (macierzowego) operatora restrykcji

$$\mathrm{r}(2\Delta x) = R(\Delta x \rightarrow 2\Delta x)\boldsymbol{r}(\Delta x)$$

4. teraz znajdujemy przybliżone rozwiązanie dla e(2Δx) tj. eliminujemy wpływ błędu wolnozmiennego

$$A\boldsymbol{e}(2\Delta x) = \boldsymbol{r}(2\Delta x)$$

- wykonujemy tylko n_2 iteracji

5. Wektor błędu e(2Δx) rzutujemy na siatkę gęstszą używając (macierzowego) operatora prolongacji

$$\mathrm{r}(\Delta x) = P(2\Delta x \rightarrow \Delta x)\boldsymbol{r}(2\Delta x)$$

6. poprawiamy przybliżone rozwiązanie na siatce (Δx)

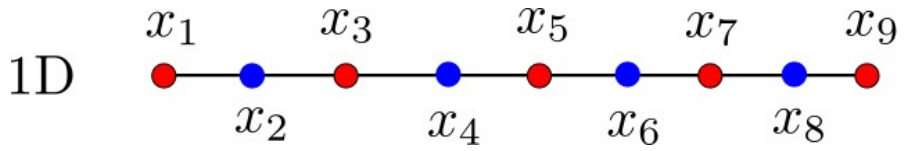
$$\tilde{\boldsymbol{x}}(\Delta x) = \boldsymbol{x}(\Delta x) - e(\Delta x)$$

Uwaga: wektory x(Δx) oraz e(Δx) stanowią tylko przybliżenia ich dokładnych odpowiedników dlatego procedurę (kroki 1-6) powtarzamy aż do uzyskania zbieżności np. funkcjonału dla pola elektrycznego

Operator restrykcji/prolongacji – jak je skonstruować?

Najpierw zobaczmy co chcemy uzyskać

Przykład dla siatki 1D



---

restrykcja:

zostawiamy tylko
węzły niebieskie

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

---

prolongacja:

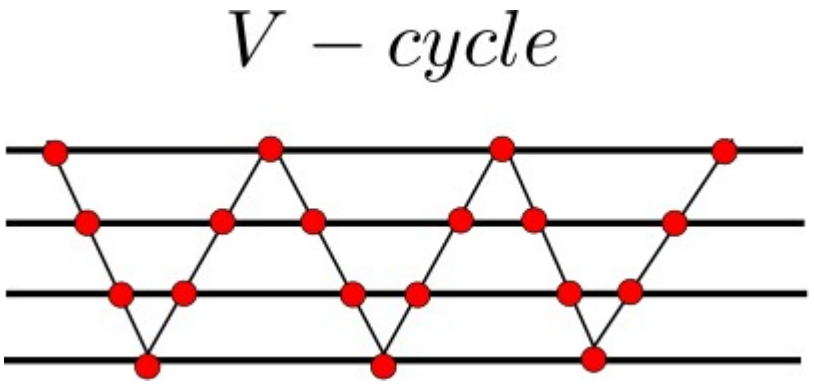
generujemy węzły czerwone x_3 , x_5 i x_7

(węzły brzegowe x_1 i x_9 – bez zmian
np. ze względu na WB Dirichleta)

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1/2 & 1/2 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1/2 & 1/2 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1/2 & 1/2 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_4 \\ x_6 \\ x_8 \\ x_9 \end{bmatrix}$$

W relaksacji wielosiatkowej możmy stosować różne schematy

4-siatkowa relaksacja typu V-cycle



Pełna relaksacja wielosiatkowa

- szukamy rozwiązania na siatce (Δx*k) [1]
- rozwiązania rzutujemy je na siatkę (Δx*k/2) [2]
  i wykonujemy N(k/2) iteracji
- szukamy błedu na siatce (Δx*k) [3]
- poprawiamy rozwiązanie na siatce (Δx*k/2) [4]
- rzutujemy rozwiązanie na siatkę (Δx*k/4) [5]
  i wykonujemy N(k/4) iteracji
- szukamy błędu na siatce (Δx*k/2) [6] oraz (Δx*k) [7]
- poprawiamy błąd na siatce (Δx*k/2) [8] oraz
  rozwiązanie na siatce (Δx*k/4) [9]
- rzutujemy rozwiązanie na siatkę (Δx*k/8) [10]
  i relaksujemy je N(k/8) razy
- wykonujemy jeden do kilku pełnych V-cykli