Metody iteracyjne

Na tych laboratoriach skupimy się na rozwiązaniu układu:

$$Ax = b$$

Naszym celem będzie więc napisanie funkcji `Solve` zastępującej funkcję `Gauss`. Nie będziemy jednak tego układu rozwiązywać metodą bezpośrednią, taką jak eliminacja Gaussa, ale metodą iteracyjną. Tzn: będziemy konstruować kolejne przybliżenia $x^{(n)}$ dokładnego x, takie że $b - Ax^{(n)}$ będzie coraz bliższe zeru.

$r = b - Ax^{(n)}$ nazywamy **residual'em**.

Zadanie

Policz residual. Następnie policz i wyświetl jego normę: $\|r\| = \sqrt{r^T r}$ (napisz funkcję liczącą normę wektora `norm(double *,int)`). Ile wynosi ta norma przed i po rozwiązaniu układu metodą eliminacji Gaussa?

Na głupa

Pierwszym pomysłem na iteracyjne rozwiązywanie byłoby postawienie:

$$x^{(n+1)} = x^{(n)} + p$$

Gdzie p jest „poprawką" w iteracji. Łatwo sprawdzić, że idealne p byłoby równe:

$$p = A^{-1} r$$

Jednak nie mamy A^{-1} (w tym rzecz). Zamiast niej użyjemy M^{-1}, gdzie M będzie przybliżeniem A. Macierz M^{-1} nazywamy **preconditioner'em**. Na początek zamiast rozwiązywać pełen układ, pominiemy większość jego elementów:

$$\begin{array}{lllll}
A_{11}p_1 & +A_{12}p_2 & +A_{13}p_1 & +\cdots & +A_{1n}p_n = r_1 \\
A_{21}p_1 & +A_{22}p_2 & +A_{23}p_1 & +\cdots & +A_{2n}p_n = r_2 \\
A_{31}p_1 & +A_{32}p_2 & +A_{33}p_1 & +\cdots & +A_{3n}p_n = r_3 \\
\cdots & & & & \\
A_{n1}p_1 & +A_{n2}p_2 & +A_{n3}p_1 & +\cdots & +A_{nn}p_n = r_n
\end{array}$$

Co daje nam prosty wzór na p:

$$p_i = \frac{1}{A_{ii}} r_i$$

Jest to równoważne z wzięciem za M diagonalnej części A. Ten prosty schemat iteracji, z powyższą poprawką nazywamy **metodą Jacobiego**.

Zadanie

Zaczynając od $x = 0$ powtarzaj tą prostą iterację (np. 1000 razy). W każdej iteracji wyświetlaj normę residualu, a także wywołaj funkcję `draw\_residual(double)` by wykonać wykres zbieżności.

Tak wykonana iteracja się nie zbiega. Wprowadźmy współczynnik, który „przytłumi" wykonywane iteracje:

$$x^{(n+1)} = x^{(n)} + \alpha p$$

Zadanie

Sprawdź zbieżność tego schematu przy różnych α. Sprawdź 0.5, 0.9, 1.1 i 2.

Zadanie

Wydziel z funkcji `Solve` część odpowiedzialną za mnożenie przez A: `Mult(double** A, double*x, double* r)` i preconditioner: `Precond(double** A, double*x, double* p)`

Spróbujmy poprawić nasz schemat biorąc lepszy preconditioner. Zauważmy, że licząc p_2 mamy już obliczone p_1 i możemy go użyć. Tak więc nie musimy pomijać elementów układu „pod diagonalą":

$$\begin{array}{lllll}
A_{11}p_1 & +A_{12}p_2 & +A_{13}p_1 & +\cdots & +A_{1n}p_n = r_1 \\
A_{21}p_1 & +A_{22}p_2 & +A_{23}p_1 & +\cdots & +A_{2n}p_n = r_2 \\
A_{31}p_1 & +A_{32}p_2 & +A_{33}p_1 & +\cdots & +A_{3n}p_n = r_3 \\
\cdots & & & & \\
A_{n1}p_1 & +A_{n2}p_2 & +A_{n3}p_1 & +\cdots & +A_{nn}p_n = r_n
\end{array}$$

Co daje nam prosty wzór na p:

$$p_i = \frac{1}{A_{ii}} (r_i - \sum_{j=1}^{i-1} A_{ij} p_j)$$

Gdy $\alpha = 1$ schemat taki nazywamy **Metodą Gaussa-Seidla**.

Zadanie

Wypróbuj nowy wzór na p, znów sprawdzając różne α.

Schematy z $\alpha > 1$ nazywamy metodami **Successive Over-Relaxation** (SOR).

Dobieramy α

Widać wyraźnie, że zbieżność bardzo zależy od α i jasnym jest, że najlepiej byłoby dobierać ten współczynnik w każdej iteracji. Zauważmy że residual po iteracji wynosi:

$$\hat{r} = r - \alpha Ap$$

Spróbujmy zminimalizować kwadrat normy tego residualu:

$$\hat{r}^T\hat{r} = (r - \alpha Ap)^T(r - \alpha Ap) = r^Tr - 2\alpha r^TAp + \alpha^2(Ap)^TAp$$

Licząc pochodną po α mamy:

$$-r^TAp + 2\alpha(Ap)^TAp = 0$$

Ostatecznie:

$$\alpha = \frac{r^TAp}{(Ap)^TAp}$$

Schemat z takim α nazywamy metodą **MINRES**.

Zadanie

Oblicz wektor Ap. Zauważ, że wyrażenie a^Tb to iloczyn skalarny dwóch wektorów $a^Tb = a \cdot b$. Napisz funkcję liczącą iloczyn skalarny `skal(double*, double*, int)` i oblicz α z powyższego wzoru. Sprawdź zbieżność przy takim α.

Wycinamy nadmiary

Przez q oznaczmy poprawkę z poprzedniej iteracji. Można powiedzieć, że w następnej iteracji nie chcemy „stracić''tego co „zyskaliśmy" w poprzedniej. Dlatego za nową poprawkę weźmiemy $p - \beta q$. Teraz wzór na nowy residual będzie:

$$\hat{r} = r - \alpha A(p - \beta q)$$

Zadanie

Wypisz wzór na $\hat{r}^T\hat{r}$ i zróżniczkuj go po β. Wylicz β przyjmując, że $r^TAq = 0$ (to wynika z poprzedniej iteracji).

Zadanie

Zmodyfikuj iterację wg. schematu: - oblicz residual - oblicz $p = M^{-1}r$ - jeżeli to nie pierwsza iteracja: oblicz β i nową poprawkę: $p = p - \beta q$ - oblicz α - wylicz nowe rozwiązanie $x = x + \alpha p$ - zachowaj poprawkę $q = p$ (opłaca się też zachować Ap)

A jeśli A jest symetryczna i dodatnio określona ...

W naszym przypadku możemy wykorzystać fakt, że macierz A jest symetryczna i dodatnio określona. Wtedy zamiast minimalizować r^Tr możemy minimalizować pewien specjalny funkcjonał:

$$\frac{1}{2}x^TAx - b^Tx$$

Pytanie: Jakie fizyczne wyjaśnienie mają następujące rzeczy w naszym przypadku: - Czym jest powyższy funkcjonał? - Dlaczego A jest symetryczna? - Dlaczego A jest dodatnio określona?

Zadanie

Podstaw w powyższym wzorze $x = x^{(n)} + \alpha p$, zróżniczkuj i wylicz α. Zauważ, że $\frac{1}{2}x^TAx - b^Tx = \text{const} + \frac{1}{2}(\alpha p)^TA(\alpha p) - r^T(\alpha p)$.

Zadanie

Analogicznie jak poprzednio, podstaw $x = x^{(n)} + \alpha(p - \beta q)$, zróżniczkuj i wylicz β.(tym razem $q^Tr = 0$)

Zadanie

Zastosuj dokładnie identyczną iterację zamieniając jedynie α i β i zbadaj zbieżność.

Schemat taki nazywamy metodą **gradientu sprzężonego** — Conjugate Gradient Method (**CG**).

Uwaga: Aktualnie zbieżność jest bardzo słaba. Wynika to z faktu, że choć A jest symetryczna to preconditioner z metody Gaussa-Seidla M^{-1} już nie jest.

Zadanie

Zbadaj zbieżność z preconditionerem diagonalnym, lub wyrażeniem $p = r$ (brakiem preconditionera).

Uwaga: Metodę Conjugate Gradient można zaimplementować w bardziej „zwartej" formie. Taki schemat można znaleść na wikipedii, bądz w notatkach z wykładu.