Algorytm faktoryzacji Cholesky'ego

Jeśli A jest macierzą symetryczną i dodatnio określoną (jest to częsty przypadek w zadaniach optymalizacji i sterowania optymalnego), a równanie nasze dotyczy nie jednego wektora x tylko macierzy o dużej liczbie kolumn, wygodniej jest skorzystać z faktoryzacji Cholesky'ego. Polega ona na poszukiwaniu najpierw dolnej trójkątnej macierzy L, takiej że

$$A = LL^T$$

a potem uzyskaniu rozwiązania układu równań $Ax = b$, czyli $LL^T x = b$, w dwóch etapach:

1. Metodą podstawiania w przód rozwiązuje się równanie:

$$Lu = b \tag{27}$$

2. Metodą podstawiania wstecznego rozwiązuje się równanie:

$$L^T x = u \tag{28}$$

W metodzie Cholesky'ego elementy macierzy L spełniają zależności:

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \dots & a_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \dots & a_{nn} \end{bmatrix} = \begin{bmatrix} l_{11} & 0 & 0 & \dots & 0 \\ l_{21} & l_{22} & 0 & \dots & 0 \\ l_{31} & l_{32} & l_{33} & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ l_{n1} & l_{n2} & l_{n3} & \dots & l_{nn} \end{bmatrix} \begin{bmatrix} l_{11} & l_{21} & l_{31} & \dots & l_{n1} \\ 0 & l_{22} & l_{32} & \dots & l_{n2} \\ 0 & 0 & l_{33} & \dots & l_{n3} \\ \vdots & \vdots & \ddots & \vdots & \\ 0 & 0 & 0 & \dots & l_{nn} \end{bmatrix} =$$

$$= \begin{bmatrix} l_{11}^2 & l_{21}l_{11} & l_{31}l_{11} & \dots & l_{n1}l_{11} \\ l_{21}l_{11} & l_{21}^2 + l_{22}^2 & l_{21}l_{31} + l_{22}l_{32} & \dots & l_{n1}l_{21} + l_{22}l_{n2} \\ l_{31}l_{11} & l_{31}l_{21} + l_{32}l_{22} & l_{31}^2 + l_{32}^2 + l_{33}^2 & \dots & l_{31}l_{n1} + l_{32}l_{n2} + \dots + l_{33}l_{n3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ l_{n1}l_{11} & l_{n1}l_{21} + l_{n2}l_{22} & l_{n1}l_{31} + l_{n2}l_{32} + l_{n3}l_{33} & \dots & l_{n1}^2 + l_{n2}^2 + \dots + l_{nn}^2 \end{bmatrix} \tag{29}$$

Zrównoleglenie kodu: wersja wierszowa

Widać, że elementy kolejnych kolumn macierzy L można wyznaczyć w następujący sposób:

for *k = 1 to n /* pętla kolejnych kolumn wyliczanych */* **do**
- $l_{kk} = \sqrt{a_{kk} - \sum_{j=1}^{k-1} l_{kj}^2}$;
- **for** *i=k+1 to n /* pętla elementów w k-tej kolumnie */* **do**
  - $l_{ik} = \frac{a_{ik} - \sum_{j=1}^{k-1} l_{ij} l_{kj}}{l_{kk}}$
- **end**

end

Zrównoleglenie metody: wersja kolumnowa

W powyższym algorytmie iteracje wewnętrznej pętli, w której najwięcej pracy wymaga obliczenie iloczynu skalarnego dotychczas wyznaczonej części wiersza k-tego oraz i-tego, mogą być realizowane niezależnie od siebie.

Algorytm ten można zapisać w nieco inny sposób, w którym zmieniane są elementy macierzy A:

for *k = 1 to n-1 /* pętla kolejnych kolumn wyliczanych */* **do**
  $l_{kk} = \sqrt{a_{kk}}$;
  for *i=k+1 to n /* pętla elementów w k-tej kolumnie */* **do**
    $l_{ik} = \frac{a_{ik}}{l_{kk}}$
  end
  for *j=k+1 to n/* pętla modyfikacji kolumn na prawo od k-tej w macierzy A*/* **do**
    for *i=j to n /* pętla elementów w j-tej kolumnie */* **do**
      $a_{ij} := a_{ij} - l_{ik}l_{jk}$
    end
  end
end
$l_{nn} = \sqrt{a_{nn}}$;

W wersji tej, zwanej kolumnową, albo z produktem zewnętrznym, kolumny macierzy A mogą być modyfikowane niezależnie od siebie, także równolegle. Dzięki uniknięciu liczenia iloczynu skalarnego, czyli redukcji, lepiej się ona nadaje na architektury wektorowe.