Choleski-Banachiewicz Approach to Systems with Non-positive Definite Matrices with *MATHEMATICA*®

Ryszard A. Walentyński

Silesian University of Technology, Faculty of Civil Engineering
ul. Akademicka 5, PL44-101 Gliwice, Poland
rwal@kateko.bud.polsl.gliwice.pl

Abstract. The paper presents the discussion on extension of potential application of the Choleski-Banachiewicz algorithm to the system of linear equations with non-positive definite matrices. It is shown that the method is also stable in case of systems with negative definite matrices and sometimes may be also successful if the matrix is neither positive nor negative definite. The algorithm handles systems with complex symmetric (not Hermitian) matrices, too. This fact has deep physical sense and engineering applications since systems with negative definite matrices are common in tasks of dynamics and post buckling analysis in civil and mechanical engineering. Possibility of utilization of Choleski-Banachiewicz algorithm to such problems can be very practical. The entire analysis has been carried out within *MATHEMATICA*® environment.

1 Introduction

A linear system of equations with a symmetrical matrix can be efficiently solved using the so-called Cholesky decomposition, Weisstein [11]. This routine was not invented by Cholesky (Cholewski, Choleski [8]). The method was presented for the first time by Dolittle [5]. His work was forgotten and the algorithm was reinvented by several mathematicians including Cholesky whose contribution was published by Benoit [2] and by Banachiewicz [1]. Due to many inventors the algorithm has a lot of names: Cholewski m., Cholesky-Croot m., Croot m., Dolittle-Choleski m. and so on. In my country — Poland the routine is called Choleski-Banachiewicz method. However, the procedure can be called "a square root m." to avoid problems with personal names. A brief discussion on the method including its historical context has been presented by Janczura [6].

It is believed that the Choleski-Banachiewicz method is limited to the positive definite matrices, Hermitian in case of complex matrix and symmetrical in case of real matrix, Weisstein [11]. The aim of the paper is to show that this opinion is true as far as it refers to symmetry.

MATHEMATICA® version 5, which is an advanced computer algebra system, has a built-in procedure called **`CholeskyDecomposition[]`**, Wolfram [12, 14]. This routine is limited to Hermitian and positive definite matrices. In other cases it fails.

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3039, pp. 311–318, 2004.

I built my own procedure called **CholeskiBanachiewicz[]** within *MATHEMATICA®* environment. This routine not only decomposes the matrix, but also solves the system of linear equations. Due to its unique features it is used to solve linear systems of equations within Refined Least Squares Method, [10]. Testing the Choleski-Banachiewicz procedure I found that it is not limited to the system with positive definite matrices. It was already mentioned in [9]. Using symbolic computation tools I will try to explain the problem a bit wider in this contribution. It can be shown that the presented procedure works well not only with systems with symmetric non-positive real matrices but also with symmetric complex matrices (symmetric complex matrix is not Hermitian).

The described feature of the Choleski-Banachiewicz method can have important application in computational mechanics and probably in other branches of physics. Therefore a physical sense of the problem and its engineering utilization are discussed.

2 Algorithm and Its Implementation

The algorithm is described in many books and therefore will not be repeated here. The routine has been built according to the recipe given by Burden and Faires [4]. Implementation of the algorithm is straightforward within *MATHEMATICA®*. The code is as long as a procedure. This is one of the advantages of this symbolic computation system.

$$\begin{array}{l}
\mathit{In[1]:=}\ \mathtt{CholeskiBanachiewicz[aa\_List,\ bb\_List,\ prec\_]\ :=} \\
\quad \mathtt{Module}\left[\{\mathtt{a = aa,\ b = bb,\ i,\ j,\ n = Length[bb]}\},\right. \\
\quad\quad \mathtt{Do}\left[\left\{\mathtt{a}\llbracket\mathtt{i,\ i}\rrbracket = \sqrt{\mathtt{N[a}\llbracket\mathtt{i,\ i}\rrbracket\mathtt{,\ prec]} - \sum_{\mathtt{k=1}}^{\mathtt{i-1}} \mathtt{a}\llbracket\mathtt{i,\ k}\rrbracket^{2}},\right.\right. \\
\quad\quad\quad \mathtt{b}\llbracket\mathtt{i}\rrbracket = \dfrac{\mathtt{b}\llbracket\mathtt{i}\rrbracket - \sum_{\mathtt{j=1}}^{\mathtt{i-1}} \mathtt{a}\llbracket\mathtt{i,\ j}\rrbracket\,\mathtt{b}\llbracket\mathtt{j}\rrbracket}{\mathtt{a}\llbracket\mathtt{i,\ i}\rrbracket}, \\
\quad\quad\quad \mathtt{Do}\left[\mathtt{a}\llbracket\mathtt{j,\ i}\rrbracket = \dfrac{\mathtt{a}\llbracket\mathtt{j,\ i}\rrbracket - \sum_{\mathtt{k=1}}^{\mathtt{i-1}} \mathtt{a}\llbracket\mathtt{j,\ k}\rrbracket\,\mathtt{a}\llbracket\mathtt{i,\ k}\rrbracket}{\mathtt{a}\llbracket\mathtt{i,\ i}\rrbracket},\right. \\
\quad\quad\quad \left.\{\mathtt{j,\ i+1,\ n}\}\right], \\
\quad\quad\quad\quad\quad \left.\left.\mathtt{Do[a}\llbracket\mathtt{j,\ i}\rrbracket = \mathtt{0,\ \{j,\ i-1\}]}\right\},\ \{\mathtt{i,\ n}\}\right]; \\
\quad\quad \mathtt{Do}\left[\mathtt{b}\llbracket\mathtt{i}\rrbracket = \dfrac{\mathtt{b}\llbracket\mathtt{i}\rrbracket - \sum_{\mathtt{j=i+1}}^{\mathtt{n}} \mathtt{a}\llbracket\mathtt{j,\ i}\rrbracket\,\mathtt{b}\llbracket\mathtt{j}\rrbracket}{\mathtt{a}\llbracket\mathtt{i,\ i}\rrbracket},\ \{\mathtt{i,\ n,\ 1,\ -1}\}\right]; \\
\quad\quad\quad \left.\{\mathtt{a,\ b}\}\right]
\end{array}$$

The arguments of the procedure are: matrix $\boldsymbol{A}$, vector $\boldsymbol{b}$ and desired working precision. If two first arguments are exact numbers, symbolic or have precision higher than machine precision – the computations can be done with precision up to the infinity. It is important for ill-conditioned tasks. The output of the routine are decomposed matrix $\boldsymbol{L}$ and the vector $\boldsymbol{x}$.

3 Dealing with Non-positive Matrices

As it was mentioned in the introduction, we will show that the algorithm successfully deals with not positive definite matrices. We will discuss two examples of the systems with real and complex matrices.

3.1 Real Matrix

The presented example, for the sake of conciseness, is limited to system of matrices 4 × 4. One can evaluate it with bigger dimensions of matrices.

```
In[2]:= dim = 4;
```

Let us build an arbitrary matrix ***m*** with random elements.

```
In[3]:= m = Table[Random[Integer, {-100, 100}], {i, dim}, {j, dim}];
```

On this basis we can produce a symmetrical matrix $\boldsymbol{m}_1$.

```
In[4]:= MatrixForm[m1 = m + Transpose[m]]
         ( 96    -2   -80  58 )
         ( -2   -142  -40  66 )
Out[4]=  ( -80  -40   162  29 )
         ( 58    66    29  22 )
```

If we try to decompose the matrix $\boldsymbol{m}_1$ with the built-in procedure:

```
In[5]:= CholeskyDecomposition[m1]
```

the attempt finishes with a message announcing: *The matrix is not sufficiently positive definite to complete the Cholesky decomposition to reasonable accuracy*. It is especially odd and conservative behavior since the matrix is exact (integer). This behavior is probably caused by the fact that the matrix is non-positive definite since some of its eigenvalues are negative numbers. Nevertheless the pitfall should be not connected with accuracy.

```
In[6]:= eig = Sort[Eigenvalues[N[m1]]]
Out[6]= {-176.436, -4.60622, 99.5497, 219.493}
```

Let us build a new matrix function $\boldsymbol{m}_2$ of the parameter λ.

```
In[7]:= m2[λ_] := m1 - IdentityMatrix[dim] λ
```

The parameter λ modifies elements of the matrix diagonal. It is easy to check that the matrix $\boldsymbol{m}_2$ is positive definite for any λ smaller than minimal eigenvalue of the original matrix $\boldsymbol{m}_1$:

```
In[8]:= Sort[Eigenvalues[N[m2[-400]]]]
Out[8]= {223.564, 395.394, 499.55, 619.493}
```

and negative definite for any value of the parameter λ grater than maximal eigenvalue of the matrix $\boldsymbol{m}_1$.

```
In[9]:= Sort[Eigenvalues[N[m2[400]]]]
Out[9]= {-576.436, -404.606, -300.45, -180.507}
```

Let us create a vector $\boldsymbol{b}$, which will be a vector of free elements of the system of linear equations.

```
In[10]:= b = Table[Random[Integer, {-100, 100}], {i, dim}]
Out[10]= {16, -59, -62, 5}
```

The l_2-norm of this vector $\|\boldsymbol{b}\|_2$ is equal to:

```
In[11]:= normb2 = Norm[b, 2]
Out[11]= √7606
```

Since both matrix $\boldsymbol{m}_1$ and vector $\boldsymbol{b}$ are exact numbers we can solve the system exactly, to enforce it the third argument that represents working precision is set to ∞. This command returns the decomposed matrix $\boldsymbol{L}$, only. Solution of the system is omitted by setting the argument **⟦1⟧** in the command.

```
In[12]:= MatrixForm[CholeskiBanachiewicz[m1, b, ∞]⟦1⟧]
```

$$\text{Out[12]=}\begin{pmatrix} 4\sqrt{6} & 0 & 0 & 0 \\ -\frac{1}{2\sqrt{6}} & \frac{1}{2}\,\mathrm{i}\sqrt{\frac{3409}{6}} & 0 & 0 \\ -10\sqrt{\frac{2}{3}} & 250\,\mathrm{i}\sqrt{\frac{2}{10227}} & \sqrt{\frac{366658}{3409}} & 0 \\ \frac{29}{2\sqrt{6}} & -\frac{1613\,\mathrm{i}}{2\sqrt{20454}} & \frac{196421}{\sqrt{1249937122}} & \mathrm{i}\sqrt{\frac{4439473}{366658}} \end{pmatrix}$$

The decomposed matrix is complex. It is not a problem for contemporary computer algebra systems and numerical systems like MATLAB since all of them can handle complex numbers.

In spite of the fact that the input matrix is non-positive definite one we can check that $\boldsymbol{L}\cdot\boldsymbol{L}^T \equiv \boldsymbol{m}_1$.

```
In[13]:= %.Transpose[%] === m1
Out[13]= True
```

We can also test if really *the matrix is not sufficiently positive definite to complete the Cholesky decomposition to reasonable accuracy*. Let us decompose the matrix $\boldsymbol{m}_1$ with PC machine precision that is equal to 16.

```
In[14]:= MatrixForm[CholeskiBanachiewicz[m1, b, 16]⟦1⟧]
          ( 9.79796     0           0          0        )
Out[14]=  ( -0.204124   11.9181 i   0          0        )
          ( -8.16497    3.49608 i   10.3709    0        )
          ( 5.9196      -5.63917 i  5.55576    3.47965 i)
```

This test shows that the round-off error is not very big.

```
In[15]:= MatrixForm[%.Transpose[%] - m1]
```

$$\mathit{Out[15]}= \begin{pmatrix} 0.\times 10^{-14} & -0.\times 10^{-16} & -0.\times 10^{-15} & 0.\times 10^{-15} \\ -0.\times 10^{-16} & -0.\times 10^{-14} & 0.\times 10^{-15} & -0.\times 10^{-15} \\ -0.\times 10^{-15} & 0.\times 10^{-15} & -0.\times 10^{-14} & 0.\times 10^{-14} \\ 0.\times 10^{-15} & -0.\times 10^{-15} & 0.\times 10^{-14} & 0.\times 10^{-14} \end{pmatrix}$$

The actual aim of my Choleski-Banachiewicz procedure is not the decomposition of the matrix itself but solution of the system.

The function $e(\lambda)$ finds a normalized relative error of the system solution with Choleski-Banachiewicz method of the system $\boldsymbol{m}_2(\lambda)\cdot\boldsymbol{x} == \boldsymbol{b}$:

$$e(\lambda) = \frac{\|\boldsymbol{m}_2(\lambda)\cdot\boldsymbol{x} - \boldsymbol{b}\|_2}{\|\boldsymbol{b}\|_2}\,, \tag{1}$$

what is implemented with:

$$\mathit{In[16]:=}\ \mathbf{normchol[x\_] :=} \frac{\mathbf{Norm[m2[x].Chop[CholeskiBanachiewicz[m2[x], b, 16]⟦2⟧] - b]}}{\mathbf{normb2}}$$

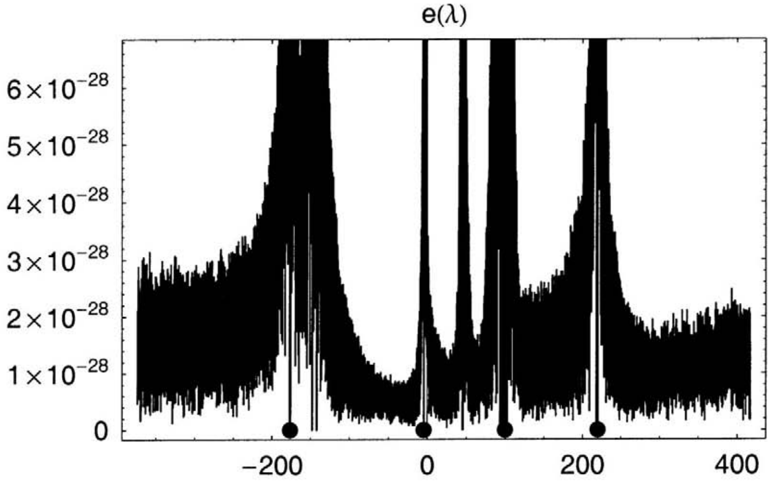

Fig. 1. The relative error of the solution with Choleski-Banachiewicz method $e(\lambda)$ with regard to the parameter λ – description in text

Figure 1 shows the diagram of the function $e(\lambda)$ in a lot of points from the interval $\lambda \in \langle -374, 417\rangle$. Four big points in the bottom of the diagram represents position of the $\boldsymbol{m}_1$ matrix eigenvalues. It can be found that the precision of the solution becomes poor near these points. The matrix $\boldsymbol{m}_2(\lambda)$ for λ equal to eigenvalues of $\boldsymbol{m}_1$ becomes singular and in their neighborhoods badly conditioned.

3.2 Complex Matrix

Let us consider a symmetric complex matrix $\boldsymbol{A}$:

$$\boldsymbol{A} := \begin{pmatrix} 1 & -2+i & -i \\ -2+i & 19 & -3 \\ -i & -3 & 3 \end{pmatrix}. \tag{2}$$

Attempt of use of built-in function **`CholeskyDecomposition[]`** fails with a message: *The matrix is not Hermitian or real and symmetric.*

Let us attempt to solve a system with this matrix with **`CholeskiBanachiewicz[]`** procedure.

$$\mathit{In[17]:=}\ \mathbf{MatrixForm}\Big[\Big(\mathbf{sol = CholeskiBanachiewicz}\Big[\begin{pmatrix} \mathbf{1} & \mathbf{-2+i} & \mathbf{-i} \\ \mathbf{-2+i} & \mathbf{19} & \mathbf{-3} \\ \mathbf{-i} & \mathbf{-3} & \mathbf{3} \end{pmatrix}, \mathbf{\{1, 1, 1\}, \infty}\Big]\Big)[\![\mathbf{2}]\!]\Big]$$

The result is saved in variable **`sol`** and the output presents the decomposed matrix $\boldsymbol{L}$:

$$\mathit{Out[17]=}\ \begin{pmatrix} 1 & 0 & 0 \\ -2+i & 2\sqrt{4+i} & 0 \\ -i & -\frac{2+i}{\sqrt{4+i}} & \sqrt{\frac{52}{17}-\frac{13\,i}{17}} \end{pmatrix}$$

It can be checked that $\boldsymbol{L} \cdot \boldsymbol{L}^T \equiv \boldsymbol{A}$.

```
In[18]:= MatrixForm[%.Transpose[%]]
```

$$\mathit{Out[18]=}\ \begin{pmatrix} 1 & -2+i & -i \\ -2+i & 19 & -3 \\ -i & -3 & 3 \end{pmatrix}$$

The solution with **`CholeskiBanachiewicz[]`** procedure is exact since both matrix $\boldsymbol{A}$ and free vector are exact numbers and working precision has been set to ∞.

```
In[19]:= sol[[2]]
```

$$\mathit{Out[19]=}\ \left\{\frac{15}{13}+\frac{4\,i}{13}, \frac{7}{26}+\frac{i}{26}, \frac{1}{2}+\frac{11\,i}{26}\right\}$$

Let us check if numerical computations with machine precision will finish with satisfactory precision. The result of Choleski-Banachiewicz procedure:

$$\mathit{In[20]:=}\ \mathbf{CholeskiBanachiewicz}\Big[\begin{pmatrix} \mathbf{1} & \mathbf{-2+i} & \mathbf{-i} \\ \mathbf{-2+i} & \mathbf{19} & \mathbf{-3} \\ \mathbf{-i} & \mathbf{-3} & \mathbf{3} \end{pmatrix}, \mathbf{\{1, 1, 1\}, 16}\Big][\![\mathbf{2}]\!]$$

```
Out[20]= {1.15385+0.307692 i, 0.269231+0.0384615 i, 0.5+0.423077 i}
```

One can easily check that a built-in function **`LinearSolve[]`** returns the same result.

4 Physical Sense and Engineering Applications

Let us consider a mechanical, dynamical, discrete system described with the following equation of motion, Langer [7].

$$\boldsymbol{B} \cdot \ddot{\boldsymbol{q}} + \boldsymbol{C} \cdot \dot{\boldsymbol{q}} + \boldsymbol{K} \cdot \boldsymbol{q} = \boldsymbol{P}\,, \tag{3}$$

where $\boldsymbol{B}$ is a matrix of inertia, $\boldsymbol{C}$ is a matrix of viscotic dumping, $\boldsymbol{K}$ is a stiffness matrix, $\boldsymbol{P}$ is a vector of external load and $\boldsymbol{q}$ is a vector of generalized Lagrangian displacements.

If the system is loaded with stationary periodic load $\boldsymbol{P} := \boldsymbol{P}_s \sin(2\pi\, p t)$, where p is a frequency of the load, $[p] = Hz$, and dumping is negligible and we can set $\boldsymbol{C} := 0$ in the equation of motion (3) we can derive the following system of linear algebraic equations.

$$\left(\boldsymbol{K} - 4\pi^2 p^2 \boldsymbol{B}\right) \boldsymbol{q}_s = \boldsymbol{P}_s\,. \tag{4}$$

The matrix $\boldsymbol{K}$ is positive definite but the matrix $\left(\boldsymbol{K} - 4\pi^2 p^2 \boldsymbol{B}\right)$ becomes non-positive definite if frequency p becomes bigger than the first frequency of free vibration of the system.

There are two rational engineering approach to design of the dynamical systems, both of them assume that the frequency of external load to be far from the resonance (eigenvalues) zone. The first approach consists in such design of the dynamical system that the free vibration frequencies become higher than frequency of excitations. It results in very stiff and heavy (expensive) structures. Another approach consists in design of very flexible structure with frequencies of free vibrations much smaller than the frequency of excitation. The disadvantage of such approach is that the dynamical system has to go through "resonance zone" during acceleration. Such phenomenon can be observed in many facilities like domestic washing machines, computer processor fans and turbo-generators in power stations. This problem may be overcome by decreasing the time of acceleration during turning on and off. On the other hand the mathematical model of the dynamical system excited with frequencies close to free vibrations has to consider dumping. In this case equation (4) is not valid. Some more complicated formulas have to be used, Langer [7]. A shape of upper envelope of the diagram of the normalized error in Figure 1 has some analogy to the diagram of the dynamical coefficient used in engineering dynamical analysis.

Systems with negative definite matrices can be stable not only in tasks of civil engineering; example is Bulatović [3]. It seems to be good news that such systems can be effectively solved with Choleski-Banachiewicz method.

5 Conclusions

Choleski-Banachiewicz procedure can be applied not only to positive definite matrices but also to negative definite ones and sometimes to matrices which are neither positive nor positive definite. It has been also shown that the procedure can be successfully employed to systems with symmetrical complex matrices.

Possible application of the method to negative definite matrices can be very useful in analysis of dynamical systems in civil and mechanical engineering. Another field can be post-buckling behavior of structures.

The analysis was carried out thanks to the computer algebra system *MATHEMATICA*®. Despite of conservative behavior of the built-in procedure, the system enabled me to challenge the problem and extend the potential of the procedure.

References

1. Banachiewicz, T.: Methode de Resolution Numerique des Equations Lineaires, Du Calcul des Determinants et des Inverses et de Reduction des Formes Quadratiques. Bull. Ac. Pol. Sci. A. (1938) 393–404
2. Benoit, A.: Note Surface ton Methode, Proce de Du Commendent Cholesky. Bull. Geod. Toulouse (1924) 67–77
3. Bulatović, R.M.: On Stability Criteria for Gyroscopic Systems with Negative Definite Stiffness. Facta Universitatis, series Mechanica, Automatic Control and Robotics **2**(10) (2000) 1081–1087 http://facta.junis.ni.ac.yu/facta/macar/macar2000/macar2000-07.pdf
4. Burden, R.L., Faires, J.D.: Numerical Analysis. 5th edn. PWS Publishing Company, Boston (1993)
5. Dolittle, M.H.: Method Employed in the Solution of Normal Equations and the Adjustment of Transquation. U. S. Survey Report (1878) 115–120
6. Janczura, A.: Solving of Linear Equations with Bounded Right-Hand Side. Mechanika i Komputer, Warsaw **8** (1990) 235–250
7. Langer, J.: Dynamics of Bar Structures. In Structural Mechanics with Elements of Computer Approach, volume 2, chapter 7. Arkady, Warsaw (1984) 76–156
8. Pattison, D.H.: Summary: Pronunciation of 'Cholesky'. S-News Mailing List Archives Division of Biostatistics at Washington University School of Medicine, Saint Louis (2000) http://www.biostat.wustl.edu/archives/html/s-news/2000-10/msg00172.html
9. Walentyński, R.A.: Refined Least Squares Method for Shells Boundary-Value Problems. In: Tazawa, Y. et al. (eds.): Symbolic Computation: New Horizons. Proceedings of the Fourth International *MATHEMATICA*® Symposium. Tokyo Denki University Press, Tokyo (2001) 511–518 (extended version on the proceedings CD-ROM)
10. Walentyński, R.A.: Application of Computer Algebra in Symbolic Computations and Boundary-Value Problems of the Theory of Shells. Silesian University of Technology Press, Gliwice, Civil Engineering **1587**(100) (2003)
11. Weisstein, E.: Cholesky Decomposition. In: World of Mathematics, a Wolfram Web Resource. CRC Press LLC, Wolfram Research, Inc., Boca Raton Champaign (2003) http://mathworld.wolfram.com/CholeskyDecomposition.html
12. Wolfram Research: CholeskyDecomposition. In: Built-in Functions, chapter Lists and Matrices - Matrix Operations. Wolfram Media, Champaign (2003) http://documents.wolfram.com/v5/Built-inFunctions/ListsAndMatrices/MatrixOperations/CholeskyDecomposition.html
13. Wolfram Research: MatrixManipulation. In: Standard Add-on Packages, chapter Linear Algebra. Wolfram Media, Champaign (2003) http://documents.wolfram.com/v5/Add-onsLinks/ StandardPackages/LinearAlgebra/MatrixManipulation.html
14. Wolfram, S.: Advanced Matrix Operations. In: The *MATHEMATICA*® Book – Online, chapter 3.7.10. 5th edn. Wolfram Media, Champaign (2003) http://documents.wolfram.com/v5/TheMathematicaBook/AdvancedMathematicsInMathematica/LinearAlgebra/3.7.10.html