ISSN 2078–5097. Вісн. Львів. ун-ту. Сер. прикл. матем. та інф. 2013. Вип. 20. С. 42–48
Visnyk of the Lviv University. Series Appl. Math. and Informatics. Issue 20. P. 42–48

УДК 519.6

ОБЧИСЛЕННЯ ТОЧНИХ ПОХІДНИХ ДЕТЕРМІНАНТА МАТРИЦІ

Б. Подлевський

Львівський національний університет імені Івана Франка,
вул. Університетська, 1, Львів, 79000, e-mail: bpodlev@gmail.com

Запропоновано ефективний чисельний алгоритм обчислення точних похідних детермінанта матриці, який ґрунтується на LU – розкладі. Алгоритм дає змогу обчислити не тільки першу похідну детермінанта матриці, а й другу, що дає змогу застосовувати для обчислення нулів детермінанта (власних значень спектральної задачі) методи, які використовують другі похідні. Алгоритм порівнюють з відомим алгоритмом, основою якого є формула Якобі і показано його переваги за кількістю операцій, потрібних для обчислення першої похідної детермінанта.

Ключові слова: детермінант, LU – розклад , точні похідні детермінанта, кількість операцій.

1. ВСТУП

Класичний підхід до знаходження власних значень нелінійної за спектральним параметром задачі на власні значення $\mathbf{D}(\lambda)x = 0$ (як і до лінійної задачі $\mathbf{D}x = \lambda x$) полягає у тому, що власні значення шукають як розв'язки характеристичного рівняння

$$f(\lambda) \equiv \det \mathbf{D}(\lambda) = 0 . \qquad (1)$$

Для цього спершу треба обчислити коефіцієнти характеристичного полінома (1), а потім застосувати відповідні ітераційні методи для обчислення його нулів, які є власними значеннями матриці $\mathbf{D}$.

Проте відомо [1], що нулі полінома можуть бути дуже чутливими до невеликих змін коефіцієнтів, внаслідок чого ми можемо обчислити нулі полінома, які не будуть відповідати власним значенням матриці $\mathbf{D}$. Тому розв'язують детермінантне рівняння (1) безпосередньо ітераційними методами, наприклад, методом Ньютона, використовуючи LU – розклад матриці для обчислення детермінанта у кожній точці ітераційного процесу, а похідну апроксимують скінченною різницею. Наприклад, у [2], [3] чисельно обчислюють логарифмічну похідну, яка є оберненою величиною до величини поправки у методі Ньютона.

Застосування такого підходу вносить у процес обчислення додаткову похибку, зумовлену апроксимацією похідної скінченною різницею.

Інший, альтернативний до чисельного диференціювання, підхід використовує теорему про слід матриці (формулу Якобі), яка дає точні значення для похідної детермінанта матриці, а саме (див., наприклад, [4]).

Теорема 1. Якщо елементи квадратної матриці $\mathbf{B}(\lambda)$ є диференційовними функціями за параметром λ, тоді для будь-якого λ для похідної детермінанта $\det \mathbf{B}(\lambda)$ матриці $\mathbf{B}(\lambda)$ справджується співвідношення

$$\frac{d(\det \mathbf{B}(\lambda))}{d\lambda} = \operatorname{tr}\left[\operatorname{adj}(\mathbf{B}(\lambda)) \cdot \frac{d\mathbf{B}(\lambda)}{d\lambda} \right],$$

якщо $\det \mathbf{B}(\lambda)$ не перетворюється в нуль, то

$$\frac{d(\det \mathbf{B}(\lambda))}{d\lambda} = \det \mathbf{B}(\lambda) \cdot \mathrm{tr}\left[\mathbf{B}^{-1}(\lambda) \cdot \frac{d\mathbf{B}(\lambda)}{d\lambda} \right] \text{ (формула Якобі).} \qquad (2)$$

Тут adj означає доповнення, tr означає слід, а $d\mathbf{B}(\lambda)/d\lambda = \left(db_{ij}(\lambda)/d\lambda \right)$.

Отже, формула (2) дає явний вираз для похідної детермінанта, тобто ми отримуємо точне значення похідної у потрібних точках, внаслідок чого в процес обчислення уже не вноситься додаткова похибка, зумовлена чисельним диференціюванням.

Для обчислення точних похідних детермінанта матриці використано підхід, який ґрунтується на LU – розкладі матриці. Пропонований підхід дає змогу обчислити не тільки першу похідну детермінанта матриці, а й другу, що дає змогу застосовувати для обчислення нулів детермінанта (власних значень спектральної задачі) методи, які використовують другі похідні, зокрема, двосторонні методи (див., наприклад, [5-8]).

2. ОБЧИСЛЕННЯ ТОЧНИХ ПОХІДНИХ ДЕТЕРМІНАНТА МАТРИЦІ

Теорема 2. Якщо елементи квадратної матриці $\mathbf{D}(\lambda)$ є диференційовними функціями за параметром λ, тоді для будь-якого λ для похідних детермінанта $\det \mathbf{D}(\lambda) \equiv f(\lambda)$ матриці $\mathbf{D}(\lambda)$ справджуються співвідношення

$$f'(\lambda) \equiv \left[\det \mathbf{D}(\lambda) \right]' = \sum_{k=1}^{n} v_{kk}(\lambda) \prod_{i=1, i\neq k}^{n} u_{ii}(\lambda),$$

$$f''(\lambda) \equiv \left[\det \mathbf{D}(\lambda) \right]'' = \sum_{k=1}^{n} w_{kk}(\lambda) \prod_{i=1, i\neq k}^{n} u_{ii}(\lambda) + \sum_{k=1}^{n} v_{kk}(\lambda) \left(\sum_{j=1, j\neq k}^{n} v_{jj}(\lambda) \prod_{i=1, i\neq k, i\neq j}^{n} u_{ii}(\lambda) \right),$$

де $u_{ii}(\lambda)$, $v_{ii}(\lambda)$ та $w_{ii}(\lambda)$ – елементи верхніх трикутних матриць, відповідно, $\mathbf{U}(\lambda)$, $\mathbf{V}(\lambda)$ та $\mathbf{W}(\lambda)$ у розкладах

$$\mathbf{D}(\lambda) = \mathbf{L}(\lambda)\mathbf{U}(\lambda),$$

$$\mathbf{B}(\lambda) = \mathbf{M}(\lambda)\mathbf{U}(\lambda) + \mathbf{L}(\lambda)\mathbf{V}(\lambda),$$

$$\mathbf{C}(\lambda) = \mathbf{N}(\lambda)\mathbf{U}(\lambda) + 2\mathbf{M}(\lambda)\mathbf{V}(\lambda) + \mathbf{L}(\lambda)\mathbf{W}(\lambda),$$

а $\mathbf{L}(\lambda)$ – нижня трикутна матриця з одиничними діагональними елементами.

Доведення. Відомо, що матриця $\mathbf{D}(\lambda)$ порядку n, в якої за будь-якого фіксованого значення $\lambda = \lambda_m$ головні мінори всіх порядків від 1 до $(n-1)$ відмінні від нуля, за допомогою LU – розкладу може бути записана у вигляді

$$\mathbf{D}(\lambda) = \mathbf{L}(\lambda)\mathbf{U}(\lambda). \qquad (3)$$

Тут $\mathbf{L}(\lambda)$ – нижня трикутна матриця з одиничними діагональними елементами, а $\mathbf{U}(\lambda)$ – верхня трикутна матриця. Тоді

$$f(\lambda) = \det \mathbf{L}(\lambda) \det \mathbf{U}(\lambda) = \prod_{i=1}^{n} u_{ii}(\lambda).$$

Оскільки елементи квадратної матриці $\mathbf{D}(\lambda)$ (а отже, і $\mathbf{U}(\lambda)$) є диференційовними функціями за λ, то для будь-яких λ отримуємо, що

$$f'(\lambda) = \sum_{k=1}^{n} v_{kk}(\lambda) \prod_{i=1, i\neq k}^{n} u_{ii}(\lambda),$$

$$f''(\lambda)=\sum_{k=1}^{n} w_{kk}(\lambda)\prod_{i=1,i\neq k}^{n} u_{ii}(\lambda)+\sum_{k=1}^{n} v_{kk}(\lambda)\left(\sum_{j=1,j\neq k}^{n} v_{jj}(\lambda)\prod_{i=1,i\neq k,i\neq j}^{n} u_{ii}(\lambda)\right),$$

де $v_{ii}(\lambda)=u'_{ii}(\lambda)$, а $w_{ii}(\lambda)=v'_{ii}(\lambda)$. Для знаходження значень $v_{ii}(\lambda)$ продиференціюємо (3) за λ. Отримуємо

$$\mathbf{B}(\lambda)=\mathbf{M}(\lambda)\mathbf{U}(\lambda)+\mathbf{L}(\lambda)\mathbf{V}(\lambda), \tag{4}$$

де $\mathbf{B}(\lambda)=\mathbf{D}'(\lambda)$, $\mathbf{M}(\lambda)=\mathbf{L}'(\lambda)$, $\mathbf{V}(\lambda)=\mathbf{U}'(\lambda)$, а $v_{ii}(\lambda)$ – елементи матриці $\mathbf{V}(\lambda)$. Тепер, диференціюючи (4) за λ, отримуємо

$$\mathbf{C}(\lambda)=\mathbf{N}(\lambda)\mathbf{U}(\lambda)+2\mathbf{M}(\lambda)\mathbf{V}(\lambda)+\mathbf{L}(\lambda)\mathbf{W}(\lambda).$$

Тут $\mathbf{C}(\lambda)=\mathbf{B}'(\lambda)$, $\mathbf{N}(\lambda)=\mathbf{M}'(\lambda)$, $\mathbf{W}(\lambda)=\mathbf{V}'(\lambda)$, а $w_{ii}(\lambda)$ – елементи матриці $\mathbf{W}(\lambda)$.

Отже, для обчислення $f(\lambda_m)$, $f'(\lambda_m)$ та $f''(\lambda_m)$ треба при фіксованому $\lambda=\lambda_m$ обчислити

$$\mathbf{D}=\mathbf{LU} \tag{5a}$$

$$\mathbf{B}=\mathbf{MU}+\mathbf{LV} \tag{5b}$$

$$\mathbf{C}=\mathbf{NU}+\mathbf{2MV}+\mathbf{LW}, \tag{5c}$$

звідки

$$f(\lambda_m)=\prod_{i=1}^{n} u_{ii},\quad f'(\lambda_m)=\sum_{k=1}^{n} v_{kk}\prod_{i=1,i\neq k}^{n} u_{ii},$$

$$f''(\lambda_m)=\sum_{k=1}^{n} w_{kk}\prod_{i=1,i\neq k}^{n} u_{ii}+\sum_{k=1}^{n} v_{kk}\left(\sum_{j=1,j\neq k}^{n} v_{jj}\prod_{i=1,i\neq k,i\neq j}^{n} u_{ii}\right). \tag{6}$$

Елементи матриць у розкладах (5) можуть бути обчислені за допомогою рекурентних співвідношень

$$r=1,2,\ldots,n,$$

$$u_{rk}=d_{rk}-\sum_{j=1}^{r-1} l_{rj}u_{jk},\quad k=r,\ldots,n,$$

$$l_{ir}=\left(d_{ir}-\sum_{j=1}^{r-1} l_{ij}u_{jr}\right)/u_{rr},\quad i=r+1,\ldots,n, \tag{7}$$

$$v_{rk}=b_{rk}-\sum_{j=1}^{r-1}(m_{rj}u_{jk}+l_{rj}v_{jk}),\quad k=r,\ldots,n,$$

$$m_{ir}=\left[b_{ir}-\sum_{j=1}^{r-1}(m_{ij}u_{jr}+l_{ij}v_{jr})-l_{ir}v_{rr}\right]/u_{rr},\quad i=r+1,\ldots,n, \tag{8}$$

$$w_{rk}=c_{rk}-\sum_{j=1}^{r-1}(n_{rj}u_{jk}+2m_{rj}v_{jk}+l_{rj}w_{jk}),\quad k=r,\ldots,n,$$

$$n_{ir}=\left[c_{ir}-\sum_{j=1}^{r-1}(n_{ij}u_{jr}+2m_{ij}v_{jr}+l_{ij}w_{jr})-2m_{ir}v_{rr}-l_{ir}w_{rr}\right]/u_{rr},\quad i=r+1,\ldots,n.$$

Якщо деякі головні мінори порядку $j\le n-1$ матриці дорівнюють нулю, то розклад (3) може не існувати або ж, якщо він існує, то він не є однозначним.

На практиці найкращий спосіб виявити можливість LU – розкладу – це спробувати обчислити його. Може виникнути ситуація, коли $u_{rr} = 0$ (r – порядок головного мінору матриці, який дорівнює нулю). Щоб уникнути цього, в процесі розкладу застосовують низку перестановок рядків (і / або стовпців) матриці **D** з вибором головного елемента (див., наприклад, [9]). У цьому випадку розклади (5) запишемо у вигляді

$$\mathbf{PD} = \mathbf{LU},$$
$$\mathbf{PB} = \mathbf{MU} + \mathbf{LV},$$
$$\mathbf{PC} = \mathbf{NU} + 2\mathbf{MV} + \mathbf{LW},$$

де $\mathbf{P}$ – матриця перестановок; причому $\det \mathbf{P} = (-1)^q$, де q – кількість перестановок, (наприклад, рядків). У такому випадку співвідношення (6) набудуть вигляду

$$f(\lambda_m) = (-1)^q \prod_{i=1}^{n} u_{ii}\ ,\ f'(\lambda_m) = (-1)^q \sum_{k=1}^{n} v_{kk} \prod_{i=1, i\neq k}^{n} u_{ii}\ ,$$

$$f''(\lambda_m) = (-1)^q \sum_{k=1}^{n} w_{kk} \prod_{i=1, i\neq k}^{n} u_{ii} + (-1)^q \sum_{k=1}^{n} v_{kk} \left(\sum_{j=1, j\neq k}^{n} v_{jj} \prod_{i=1, i\neq k, i\neq j}^{n} u_{ii} \right).$$

3. ПІДРАХУНОК КІЛЬКОСТІ ОПЕРАЦІЙ

Коштовність LU – розкладу матриці **D** розмірності $n \times n$ можна оцінити безпосередньо шляхом підрахунку арифметичних операцій. Це можна зробити так. Матриці **L** та **U** на r-му кроці набувають вигляду

$$\mathbf{L} \qquad\qquad \mathbf{U}$$

$$\begin{pmatrix} 1 & & & & & & \\ * & 1 & & & & & \\ * & * & 1 & & & & \\ * & * & * & 1 & & & \\ * & * & * & & 1 & & \\ \vdots & \vdots & \vdots & & & \ddots & \\ * & * & * & & & & 1 \\ * & * & * & & & & & 1 \end{pmatrix} \qquad \begin{pmatrix} * & * & * & * & * & \cdots & * & * \\ & * & * & * & * & \cdots & * & * \\ & & * & * & * & \cdots & * & * \\ & & & & & & & \\ & & & & & & & \\ \vdots & \vdots & \vdots & & & \ddots & & \\ & & & & & & & \\ & & & & & & & \end{pmatrix}$$

Рис. 1. Структура матриць **L** та **U**

Коштовність наступного кроку розкладу визначаємо з (7) у два етапи:

1) елементи верхньої трикутної матриці **U**

$$u_{rk} = d_{rk} - l_{rj} u_{jk}, \quad j = 1, \ldots, r-1, \quad k = r, \ldots, n;$$

2) елементи нижньої трикутної матриці **L**

$$l_{ir} = \left(d_{ir} - l_{ij} u_{jr} \right) / u_{rr}\ , \quad j = 1, \ldots, r-1, \quad i = r+1, \ldots, n\ .$$

На першому етапі цього кроку отримаємо $(r-1)(n-(r-1))$ операцій (1-ша операція – це множення або ділення), на другому етапі – $r(n-r)$ операцій.

Підсумовуючи для всіх значень r кількість операцій на кожному етапі, отримуємо: на першому етапі:

$$\sum_{r=2}^{n} (r-1)(n-(r-1)) = n \sum_{r=2}^{n} (r-1) - \sum_{r=2}^{n} (r-1)^2 = n \sum_{r=1}^{n-1} r - \sum_{r=1}^{n-1} r^2 =$$

$$n\frac{n(n-1)}{2}-\frac{n(n-1)(2n-1)}{6}=\frac{n(n-1)[3n-2n+1]}{6}=\frac{n(n-1)(n+1)}{6}=\frac{n(n^2-1)}{6};$$

на другому етапі:

$$\sum_{r=1}^{n-1}r(n-r)=n\sum_{r=1}^{n-1}r-\sum_{r=1}^{n-1}r^2=\frac{n(n^2-1)}{6}.$$

Отже, загальна кількість операцій для отримання $LU-$розкладу матриці дорівнює

$$\frac{n(n^2-1)}{3},$$

а для обчислення детермінанта матриці

$$\frac{n(n^2-1)}{3}+n-1=\frac{n^3}{3}+\frac{2n-3}{3}.$$

Аналогічно підраховуємо кількість операцій, які потрібні для розкладу (5b). Матриці $\mathbf{L}$, $\mathbf{U}$, $\mathbf{M}$, та $\mathbf{V}$ на r-му кроці набувають вигляду

$$\mathbf{L}\quad\begin{pmatrix}1&&&&&&\\ *&1&&&&&\\ *&*&1&&&&\\ *&*&*&1&&&\\ *&*&*&&1&&\\ \vdots&\vdots&\vdots&&&\ddots&\\ *&*&*&&&&1\\ *&*&*&&&&&1\end{pmatrix}\quad\mathbf{U}\quad\begin{pmatrix}*&*&*&*&*&\cdots&*&*\\ &*&*&*&*&\cdots&*&*\\ &&*&*&*&\cdots&*&*\\ &&&&&&&\\ &&&&&&&\\ \vdots&\vdots&\vdots&&&\ddots&&\\ &&&&&&&\\ &&&&&&&\end{pmatrix}\quad\mathbf{M}\quad\begin{pmatrix}0&&&&&&\\ *&0&&&&&\\ *&*&0&&&&\\ *&*&*&0&&&\\ *&*&*&&0&&\\ \vdots&\vdots&\vdots&&&\ddots&\\ *&*&*&&&&0\\ *&*&*&&&&&0\end{pmatrix}\quad\mathbf{V}\quad\begin{pmatrix}*&*&*&*&*&\cdots&*&*\\ &*&*&*&*&\cdots&*&*\\ &&*&*&*&\cdots&*&*\\ &&&&&&&\\ &&&&&&&\\ \vdots&\vdots&\vdots&&&\ddots&&\\ &&&&&&&\\ &&&&&&&\end{pmatrix}$$

Рис. 2. Структура матриць $\mathbf{L}$, $\mathbf{U}$, $\mathbf{M}$ та $\mathbf{V}$

Коштовність наступного кроку розкладу визначаємо з (7), (8) у чотири етапи:

1) елементи верхньої трикутної матриці $\mathbf{U}$

$$u_{rk}=d_{rk}-l_{rj}u_{jk},\quad j=1,\dots,r-1,\quad k=r,\dots,n;$$

2) елементи нижньої трикутної матриці $\mathbf{L}$

$$l_{ir}=\left(d_{ir}-l_{ij}u_{jr}\right)/u_{rr},\quad j=1,\dots,r-1,\quad i=r+1,\dots,n;$$

3) елементи верхньої трикутної матриці $\mathbf{V}$

$$v_{rk}=b_{rk}-(m_{rj}u_{jk}+l_{rj}v_{jk}),\quad j=1,\dots,r-1,\quad k=r,\dots,n;$$

4) елементи нижньої трикутної матриці $\mathbf{M}$

$$m_{ir}=\left[b_{ir}-(m_{ij}u_{jr}+l_{ij}v_{jr})-l_{ir}v_{rr}\right]/u_{rr},\quad j=1,\dots,r-1,\quad i=r+1,\dots,n.$$

На перших двох етапах кількість операцій обчислили вище, а на третьому та четвертому етапах одного кроку отримуємо $2(r-1)(n-(r-1))$ операцій та $2r(n-r)$ операцій, відповідно.

Підсумовуючи для всіх значень r кількість операцій на кожному етапі, отримуємо: на першому та другому етапах кількість операцій обчислили вище і дорівнює

$$\frac{n(n^2-1)}{3},$$

а на третьому та четвертому етапах кількість операцій дорівнює $\frac{n(n^2-1)}{3}$ на кожному.

Отже, загальна кількість операцій для отримання розкладу (5a), (5b) дорівнює

$$n(n^2-1),$$

а для обчислення похідної від детермінанта матриці –

$$n(n^2-1)+n(n-1)=n^3+n^2-n. \tag{9}$$

4. ВИСНОВКИ

Якщо використовувати для обчислення першої похідної детермінанта формулу Якобі (теорема 1), то відомо (див., наприклад, [4], [9]), що кількість операцій для обчислення похідної детермінанта матриці дорівнює

$$\frac{4}{3}n^3+n^2. \tag{10}$$

Якщо тепер порівняти (9) та (10), то доходимо висновку, що з погляду кількості операцій, потрібних для обчислення точної похідної детермінанта матриці, чисельний алгоритм, який ґрунтується на LU – розкладі матриці, ефективніший, ніж чисельний алгоритм, в основі якого є формула Якобі (2).

Зазначимо також, що запропонований підхід дає змогу обчислити не тільки першу та другу похідні детермінанта матриці, а й похідні вищих порядків, модифікувавши для цього відповідні рекурентні формули.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. *Wilkinson J.H*. Rounding errors in algebraic processes / J.H. Wilkinson. – New York: Dover Publication, 1994. – 161 p.
2. *Arbenz P*. Solving Nonlinear Eigenvalue Problems by Algorithmic differentiation / P. Arbenz, W. Gander // Computing. – 1986. – Vol. 36, No. 3. – P. 205–215.
3. *Gander W*. Zeros of Determinants of λ-Matrices / W. Gander // PAMM. – 2007. – Vol. 8, No. 1. – P. 10811–10814.
4. *Lancaster P*. Lambda-Matrices and Vibrating Systems. 2-nd Edition / P. Lancaster. – New York: Dover Publication, 2002. – 208 p.
5. *Podlevskyi B*. On the Bilateral Convergence of Halley's Method / B. Podlevskyi // ZAMM. – 2003. – Vol. 83, No. 4. – P. 282–286.
6. *Подлевский Б.М*. О некоторых двусторонних аналогах метода Ньютона решения нелинейной спектральной задачи / Б.М. Подлевский // Журн. вычисл. матем. и матем. физ. – 2007. – Т. 47, № 11. – С. 1819–1829.
7. *Podlevskyi B.M*. Some computational aspects of algorithms for solving nonlinear two-parameter eigenvalue problems / B.M. Podlevskyi // Журнал обчисл. і прикл. матем. – 2010. – № 1 (100). – С. 100–110.
8. *Podlevskyi B.M.* One approach to construction of bilateral approximations methods for solution of nonlinear eigenvalue problems / B.M. Podlevskyi // American Journal of Computational Mathematics. – 2012. – Vol. 2, No. 2. – P. 118–124.
9. *Райс Дж*. Матричные вычисления и математическое обеспечение / Дж. Райс; пер. с англ. О.Б. Арушаняна; под ред. В.В. Воеводина. – М.: Мир, 1984. – 264 с.

Стаття: надійшла до редколегії 14.11.2012
доопрацьована 26.12.2012
прийнята до друку 24.01.2013

ВЫЧИСЛЕНИЕ ТОЧНЫХ ПРОИЗВОДНЫХ ДЕТЕРМИНАНТА МАТРИЦЫ

Б. Подлевский

Львовский национальный университет имени Ивана Франко,
ул. Университетская, 1, Львов, 79000, e-mail: bpodlev@gmail.com

Предложено эффективный численный алгоритм вычисления точных производных детерминанта матрицы, который основан на $LU-$разложении. Алгоритм позволяет вычислить не только первую производную детерминанта матрицы, но и вторую, что в свою очередь позволяет применить для вычисления нулей детерминанта (собственных значений спектральной задачи) методы, использующие вторые производные. Алгоритм сравнивается с известным алгоритмом, в основании которого есть формула Якоби и показано его преимущество по количеству операций, необходимых для вычисления первой производной детерминанта.

Ключевые слова: детерминант, $LU-$разложение, точные производные детерминанта, количество операций.

CALCULATING THE EXACT DERIVATIVES OF MATRIX DETERMINANT

B. Podlevskyi

Ivan Franko National University of Lviv,
Universytetska Str., 1, Lviv, 79000, e-mail: bpodlev@gmail.com

An efficient numerical algorithm for computing exact derivatives of the determinant of the matrix which is based on the $LU-$decomposition is proposed. The algorithm allows us to compute not only the first derivative of the determinant of the matrix, but and the second derivative, which in turn allow us to apply for compute the zeros of determinant (eigenvalues of the spectral problem) methods that use the second derivatives. The algorithm is compared with the known algorithm, which is based on Jacobi formula and is shown its advantages by the number of operations required to compute the first derivative of determinant.

Key words: determinant, $LU-$decomposition, the exact derivatives of determinant, the number of operations.