

Solving a System of Linear Algebraic Equations with a Tridiagonal Matrix: A New Look at Cramer's Rule

S. K. Kydyraliev^{1*}, S. N. Sklyar^{1**}, and A. B. Urdaletova^{2***}

¹American University of Central Asia, Bishkek, Kyrgyzstan

²Kyrgyz-Turkish Manas University, Bishkek, Kyrgyzstan

Received February 2, 2020; in final form, April 7, 2020; accepted April 14, 2021

Abstract—To numerically solve a system of linear algebraic equations with a tridiagonal matrix, a recursive version of Cramer's rule is proposed. This method requires no additional restrictions on the matrix of the system similar to those formulated for the double-sweep method. The results of numerical experiments on a large set of test problems are presented. A comparative analysis of the efficiency of the method and corresponding algorithms is given.

DOI: 10.1134/S1995423921030058

Keywords: *system of linear algebraic equations with a tridiagonal matrix, Cramer's rule, recursive algorithm, diagonal dominance, double-sweep method.*

1. INTRODUCTION

Solving systems of linear algebraic equations (SLAEs) is one of major problems in computational algebra [1, 2]. SLAEs can be used in mathematical models either directly (for instance, in Leontiev's interindustry balance model) or as stages in numerically solving problems for differential equations and systems describing physical processes [3]. In particular, SLAEs with tridiagonal matrices of large dimension are inherent in approximating boundary value problems for ordinary differential equations of the second order by three-point difference schemes, as well as in difference schemes for time-dependent partial differential equations [4]. In the latter case the SLAEs are solved at each time level, that is, multiple times.

Consider the following SLAE with a tridiagonal matrix:

$$\begin{cases} b_1 y_1 + c_1 y_2 & = f_1, \\ a_k y_{k-1} + b_k y_k + c_k y_{k+1} & = f_k, \quad k = 2, 3, \dots, n-1, \\ a_n y_{n-1} + b_n y_n & = f_n. \end{cases} \quad (1)$$

One of the well-known classical economical methods for solving systems of the form (1) is the double-sweep method: various versions of this method are described, for instance, in the monograph [5]. The double-sweep method is known to be an implementation of Gaussian elimination for the systems of form (1), and the monotonic sweep method is the simplest and most frequently used one. However, the natural condition of unique solvability of system (1) (nonzero determinant of the system) is often insufficient for the method to be correct and stable. The following sufficient conditions of correctness and stability of the monotonic sweep method are typically used [5]:

* E-mail: kydyraliev_s@auca.kg

** E-mail: sklyar_s@auca.kg

*** E-mail: anarkulurdaletova@manas.edu.kg

$$\begin{aligned}
 &1) \ b_1 \neq 0, \quad b_n \neq 0, \quad a_i \neq 0, \quad c_i \neq 0, \quad i = 2, 3, \dots, n-1, \\
 &2) \ |b_1| \geq |c_1|, \quad |b_i| \geq |a_i| + |c_i|, \quad i = 2, 3, \dots, n-1, \quad |b_n| \geq |a_n|,
 \end{aligned}
 \tag{2}$$

and in the second group of conditions (2) at least one inequality is assumed to be strict. These inequalities are commonly referred to as “diagonal dominance conditions.” If system (1) is part of a more complex problem (for instance, in a reservoir hydrodynamics model [6]) and its coefficients are calculated when implementing this model, conditions of type (2) are often difficult to verify. In this case, a more universal method for solving system (1) is needed. The non-monotonic double-sweep method (or Gaussian elimination with partial pivoting [5]) is more universal: it solves system (1) under assumptions that are more general than (2), but uses a rather complicated logic. Despite the above disadvantages, Gaussian elimination is one of the most economical methods: the number of arithmetic operations for its implementation in the case of a dense matrix is proportional to n^3 , while for the double-sweep method it is proportional to n [4].

The most universal method for solving systems of linear algebraic equations with square matrices is Cramer’s rule [7]. It requires only one condition: the system determinant must be nonzero. However, Cramer’s rule is not considered economical: the number of arithmetic operations for implementing it is proportional to $n!$ [4]. Therefore, this method is typically not used in computational practice. It is shown in the present study that since the matrix of system (1) has a band structure, Cramer’s rule can be implemented recursively, retaining its universality and efficiency, like the double-sweep method.

2. RECURSIVE ALGORITHM FOR SOLVING SYSTEM (1)

Let us introduce the following notation for determinants ($k = 1, 2, \dots, n-1$):

$$D_k = \begin{bmatrix} b_k & c_k & 0 & \cdots & \cdots & 0 \\ a_{k+1} & b_{k+1} & c_{k+1} & 0 & \cdots & \cdots \\ 0 & \cdots & \cdots & \cdots & 0 & \cdots \\ \cdots & 0 & \cdots & \cdots & \cdots & 0 \\ \cdots & \cdots & 0 & a_{n-1} & b_{n-1} & c_{n-1} \\ 0 & \cdots & \cdots & 0 & a_n & b_n \end{bmatrix}, \quad D_n = b_n, \quad D_{n+1} = 1;$$

$$F_k = \begin{bmatrix} f_k & c_k & 0 & \cdots & \cdots & 0 \\ f_{k+1} & b_{k+1} & c_{k+1} & 0 & \cdots & \cdots \\ f_{k+2} & \cdots & \cdots & \cdots & 0 & \cdots \\ \cdots & 0 & \cdots & \cdots & \cdots & 0 \\ \cdots & \cdots & 0 & a_{n-1} & b_{n-1} & c_{n-1} \\ f_n & \cdots & \cdots & 0 & a_n & b_n \end{bmatrix}, \quad F_n = f_n.$$

It is easy to verify that these determinants satisfy the recurrence relations ($k = 1, 2, \dots, n-1$):

$$D_k = b_k D_{k+1} - c_k a_{k+1} D_{k+2}, \tag{3}$$

$$F_k = f_k D_{k+1} - c_k F_{k+1}. \tag{4}$$

The following simple theorem forms a basis of the method:

Theorem. If $D_1 \neq 0$ then c_k and D_{k+1} are never both zero together for any k .

Proof. Let, on the contrary, $c_k = D_{k+1} = 0$ for some k . Then it follows from (3) that $D_k = 0$. Hence, by virtue of (3), we conclude by induction that $D_{k-1} = 0$ and $D_{k-2} = 0, \dots, D_1 = 0$. The latter relation contradicts the assumption $D_1 \neq 0$, and our theorem is proved. \square

The condition $D_1 \neq 0$ means that the system of equations has a unique solution. It is this condition that we will assume to be satisfied. Let us briefly describe the method: As the double-sweep method (Gaussian elimination), it consists of two steps. The forward sweep of the method is based on the recurrent calculation of the determinants D_k and F_k by formulas (3) and (4) with corresponding initial conditions. The backward sweep, by virtue of the theorem, can be implemented by using various versions of the conditional operator, for instance: Let the quantities y_1, y_2, \dots, y_k be found; if $D_{k+1} \neq 0$, we use Cramer's formula to find y_{k+1} as a solution to a "truncated system" (equations of the system (1) beginning with the $k + 1$ th one):

$$y_{k+1} = \frac{F_{k+1} - a_{k+1}D_{k+2}y_k}{D_{k+1}}, \quad (5)$$

if $D_{k+1} = 0$, the unknown y_{k+1} is determined from the corresponding equation in (1):

$$y_{k+1} = \frac{f_k - a_k y_{k-1} - b_k y_k}{c_k} \quad (6)$$

(in this case $c_k \neq 0$).

Thus, formulas (3)–(6) are used to obtain an algorithm that implements the above-proposed method for solving system (1) (iterative Cramer's algorithm). For an unambiguous interpretation of the commands, it is presented in the form of a Pascal-like pseudocode:

IC-algorithm.

```

▼ INPUT ( $n; \{a_k\}_{k=2}^n; \{b_k\}_{k=1}^n; \{c_k\}_{k=1}^{n-1}; \{f_k\}_{k=1}^n$ );
begin
 $D_{n+1} = 1; D_n = b_n; F_n = f_n;$ 
for k = 1 to n-1 do
  begin
     $D_{n-k} = b_{n-k}D_{n-k+1} - c_{n-k}a_{n-k+1}D_{n-k+2};$ 
     $F_{n-k} = f_{n-k}D_{n-k+1} - c_{n-k}F_{n-k+1};$ 
  end
 $y_1 = F_1/D_1;$ 
if  $abs(c_1) < abs(D_2)$  then  $y_2 = (F_2 - a_2D_3y_1)/D_2$ 
  else  $y_2 = (f_1 - b_1y_1)/c_1;$ 
for k=2 to n-1 do
if  $abs(c_k) < abs(D_{k+1})$  then  $y_{k+1} = (F_{k+1} - a_{k+1}D_{k+2}y_k)/D_{k+1}$ 
  else  $y_{k+1} = (f_k - a_k y_{k-1} - b_k y_k)/c_k;$ 
end
OUTPUT ( $\{y_k\}_{k=1}^n$ )▲

```

Numerous experiments have shown (see Tables 2, 3, and 7) that the IC-algorithm does not always yield the desired result: for systems with a large number of equations it can accumulate errors in the recurrent calculation of determinants by formulas (3) and (4). To avoid this, the algorithm is modified by introducing some normalizing factors. Let us multiply the k th equation of system (1) by a nonzero factor λ_k , thereby changing the coefficients and the right-hand sides of the system but not changing its solution. Let us use the above IC-algorithm to solve the resulting system and set

$$\lambda_k = \frac{1}{|D_{k+1}| + |c_k|}, \quad k = 1, 2, \dots, n-1, \lambda_n = 1.$$

This choice of the normalizing factors is correct by virtue of the theorem we have proved above. However, the question of how to find the optimal normalizing factors in the method remains open and requires further studies. In terms of the initial system (1) the new algorithm (called the modified IC-algorithm) is as follows:

MIC-algorithm.

```

▼ INPUT ( $n$ ;  $\{a_k\}_{k=2}^n$ ;  $\{b_k\}_{k=1}^n$ ;  $\{c_k\}_{k=1}^{n-1}$ ;  $\{f_k\}_{k=1}^n$ );
begin
 $D_{n+1} = 1$ ;  $D_n = b_n$ ;  $F_n = f_n$ ;  $\lambda_n = 1$ ;
for k = 1 to n-1 do
  begin
     $\lambda_{n-k} = 1 / (|D_{n-k+1}| + |c_{n-k}|)$ ;
     $D_{n-k} = (b_{n-k}D_{n-k+1} - c_{n-k}\lambda_{n-k+1}a_{n-k+1}D_{n-k+2})\lambda_{n-k}$ ;
     $F_{n-k} = (f_{n-k}D_{n-k+1} - c_{n-k}F_{n-k+1})\lambda_{n-k}$ ;
  end
 $y_1 = F_1/D_1$ ;
if  $abs(c_1) < abs(D_2)$  then  $y_2 = (F_2 - \lambda_2 a_2 D_3 y_1) / D_2$ 
  else  $y_2 = (f_1 - b_1 y_1) / c_1$ ;
for k=2 to n-1 do
  if  $abs(c_k) < abs(D_{k+1})$  then  $y_{k+1} = (F_{k+1} - \lambda_{k+1} a_{k+1} D_{k+2} y_k) / D_{k+1}$ 
    else  $y_{k+1} = (f_k - a_k y_{k-1} - b_k y_k) / c_k$ ;
end
OUTPUT ( $\{y_k\}_{k=1}^n$ )▲

```

Note that theoretical issues of stability of the modified algorithm (MIC), which is considered to be the main one in the present paper, are not discussed here. Such a study (as evident from paper [10]) is rather complicated even for the monotonic sweep method. It is of interest in its own right and will be performed in the future.

3. RESULTS OF NUMERICAL EXPERIMENTS

Let us illustrate the computational capabilities of the algorithms proposed above with a series of test problems, comparing them with the monotonic sweep method called DSM (Double-Sweep Method). The numerical experiments have been performed in the MatLab environment. The results are presented in Tables 1–7. The tables present absolute errors calculated by the formula

$$\max_{1 \leq k \leq n} |y_k - \bar{y}_k|,$$

where y_k is an exact solution and \bar{y}_k is an approximate solution of the corresponding problem.

Problem 1:

$$\begin{cases} y_1 = \phi, \\ -y_{k-1} + 2y_k - y_{k+1} = 0, \quad k = 2, 3, \dots, n-1, \\ y_n = \psi. \end{cases} \quad (7)$$

Table 1. Problem 1, $\phi = -1$, $\psi = 1$

n	10	10^2	10^3	10^4
DSM	$2 \cdot 10^{-16}$	$5 \cdot 10^{-15}$	10^{-13}	$3 \cdot 10^{-12}$
IC	$2 \cdot 10^{-16}$	$3 \cdot 10^{-15}$	10^{-15}	$2 \cdot 10^{-15}$
MIC	$2 \cdot 10^{-16}$	10^{-14}	10^{-12}	$2 \cdot 10^{-11}$

The system of equations (7) is obtained by approximating with a three-point difference scheme the following simple boundary-value problem for a second-order equation:

$$\begin{cases} \frac{d^2 y}{dx^2} = 0, & x \in (0, 1), \\ y(0) = \phi, & y(1) = \psi. \end{cases}$$

The solution to system (7) has the following form:

$$y_k = \frac{\phi(n-k) + \psi(k-1)}{n-1}, \quad k = 1, 2, \dots, n.$$

It is easy to see that conditions (2) for system (7) are satisfied. Hence, the monotonic sweep method is correct and stable, which is confirmed by calculations whose results are presented in Table 1. The IC and MIC algorithms are also efficient for this problem.

Problem 2:

$$\begin{cases} y_1 = 0, \\ (\operatorname{cth}(R) - 1)y_{k-1} - 2\operatorname{cth}(R)y_k + (\operatorname{cth}(R) + 1)y_{k+1} = 0, & k = 2, 3, \dots, n-1, \\ y_n = 1; \end{cases} \quad (8)$$

$$R = \frac{1}{2\varepsilon(n-1)}.$$

System (8) is a difference scheme proposed by A.M. Il'in (see [8, 9]) to approximate the following singularly perturbed boundary value problem:

$$\begin{cases} \varepsilon \frac{d^2 y}{dx^2} + \frac{dy}{dx} = 0, & x \in (0, 1); \\ y(0) = 0, & y(1) = 1. \end{cases}$$

The solution to system (8) has the form

$$y_k = \frac{1 - e^{-(k-1)/\varepsilon(n-1)}}{1 - e^{-1\varepsilon}}, \quad k = 1, 2, \dots, n.$$

Conditions (2) for system (8) are satisfied, and the monotonic sweep method is correct and stable. This is confirmed (as in the previous example) by calculation with results presented in Table 2. The MIC algorithm also works well for this problem. However, the IC algorithm accumulates errors in case of a sufficiently large number of equations in the system.

Table 2. Problem 2, $\varepsilon = 10^{-2}$

n	10	10^2	10^3	10^4
DSM	$2 \cdot 10^{-15}$	$7 \cdot 10^{-15}$	$8 \cdot 10^{-13}$	$8 \cdot 10^{-11}$
IC	10^{-15}	$2 \cdot 10^{-15}$	<i>NaN</i>	<i>NaN</i>
MIC	10^{-15}	$6 \cdot 10^{-15}$	$2 \cdot 10^{-13}$	$3 \cdot 10^{-13}$

Table 3. Problem 3, $\varepsilon = 10^{-3}$

n	10	10^2	10^3	10^4
DSM	$2 \cdot 10^{-16}$	$6 \cdot 10^{-15}$	$2 \cdot 10^{-13}$	$8 \cdot 10^{-12}$
IC	10^{-15}	$2 \cdot 10^{-15}$	<i>NaN</i>	<i>NaN</i>
MIC	$2 \cdot 10^{-16}$	$2 \cdot 10^{-15}$	$7 \cdot 10^{-14}$	$2 \cdot 10^{-12}$

Problem 3:

$$\left\{ \begin{array}{l} -y_1 + y_2 = \frac{1}{n-1}, \\ \left[\varepsilon - \frac{(k-1)}{2(n-1)^2} \right] y_{k-1} - 2\varepsilon y_k + \left[\varepsilon + \frac{(k-1)}{2(n-1)^2} \right] y_{k+1} = \frac{k-1}{(n-1)^3}, \quad k = 2, 3, \dots, n-1, \\ y_n = 2. \end{array} \right. \quad (9)$$

The system of equations (9) is taken from [10] and is a difference scheme for solving the following problem:

$$\left\{ \begin{array}{l} \varepsilon \frac{d^2y}{dx^2} + x \frac{dy}{dx} = x, \quad x \in (0, 1); \\ \frac{dy}{dx}(0) = 1, \quad y(1) = 2. \end{array} \right.$$

The solution to system (9) has the form

$$y_k = 1 + \frac{k-1}{n-1}, \quad k = 1, 2, \dots, n.$$

Conditions (2) for system (9) are, generally speaking, not satisfied: in the equations with numbers such that $2\varepsilon(n-1)^2 + 1 < k$ the conditions of diagonal dominance are not satisfied. However, the monotonic sweep method works well (as shown in Table 3). The MIC method is also efficient, in contrast to the IC algorithm, which does not work at sufficiently large n .

Problem 4:

$$\left\{ \begin{array}{l} y_1 = \phi, \\ -y_{k-1} + y_k - y_{k+1} = 0, \quad k = 2, 3, \dots, n-1, \\ y_n = \psi. \end{array} \right. \quad (10)$$

The system of equations (10) is presented in [5] as an example of a system for which the monotonic sweep algorithm is not suitable. In fact, in calculating the third sweep coefficient the algorithm fails

Table 4. Problem 4, $\phi = -5$, $\psi = 10$

n	30	$3 \cdot 10^2$	$3 \cdot 10^3$	$3 \cdot 10^4$
DSM	<i>NaN</i>	<i>NaN</i>	<i>NaN</i>	<i>NaN</i>
IC	$2 \cdot 10^{-14}$	$6 \cdot 10^{-13}$	$6 \cdot 10^{-12}$	$7 \cdot 10^{-11}$
MIC	$2 \cdot 10^{-14}$	$6 \cdot 10^{-13}$	$6 \cdot 10^{-12}$	$7 \cdot 10^{-11}$

Table 5. Problem 5, $\phi = -1$, $\psi = 10$

n	40	$4 \cdot 10^2$	$4 \cdot 10^3$	$4 \cdot 10^4$
DSM	0.7	0.7	0.7	0.4
IC	$5 \cdot 10^{-14}$	$4 \cdot 10^{-13}$	$5 \cdot 10^{-12}$	$6 \cdot 10^{-11}$
MIC	$2 \cdot 10^{-14}$	$6 \cdot 10^{-13}$	$8 \cdot 10^{-12}$	$4 \cdot 10^{-11}$

because of division by zero. Note that the conditions of diagonal dominance (2) do not hold for this system. At the same time, if $n \neq 3k + 1$ for some integer $k > 0$, a solution to system (10) exists and is determined by the formula

$$y_k = \frac{\phi \sin \frac{\pi(n-k)}{3} + \psi \sin \frac{\pi(k-1)}{3}}{\sin \frac{\pi(n-1)}{3}}, \quad k = 1, 2, \dots, n.$$

The results presented in Table 4 show that the monotonic sweep method is unacceptable for this case, but both IC and MIC are efficient.

Problem 5:

$$\begin{cases} y_1 = \phi, \\ -y_{k-1} + \sqrt{2}y_k - y_{k+1} = 0, \quad k = 2, 3, \dots, n-1, \\ y_n = \psi. \end{cases} \quad (11)$$

The system of equations (11) is presented as an example of a system for which the monotonic sweep algorithm is not correct. In fact, in calculating the fourth sweep coefficient the algorithm fails because of division by zero. Note that the conditions of diagonal dominance (2) do not hold for this system. At the same time, if $n \neq 4k + 1$ for some integer $k > 0$, a solution to the system (11) exists and is determined by the formula

$$y_k = \frac{\phi \sin \frac{\pi(n-k)}{4} + \psi \sin \frac{\pi(k-1)}{4}}{\sin \frac{\pi(n-1)}{4}}, \quad k = 1, 2, \dots, n.$$

The results presented in Table 5 show that the monotonic sweep method is unacceptable in this case, but both IC and MIC work well.

Problem 6:

$$\begin{cases} -y_1 + y_2 = -1, \\ \cos(\pi k/2) y_{k-1} + \cos(\pi k/2) y_k + \sin(\pi k/2) y_{k+1} = (-1)^k, \quad k = 2, 3, \dots, 4m-1, \\ y_{4m-1} + y_{4m} = 1. \end{cases} \quad (12)$$

Table 6. Problem 6

n	40	$4 \cdot 10^2$	$4 \cdot 10^3$	$4 \cdot 10^4$
DSM	0.5	0.5	0.5	0.5
IC	$4 \cdot 10^{-15}$	$8 \cdot 10^{-14}$	$7 \cdot 10^{-13}$	$6 \cdot 10^{-12}$
MIC	$4 \cdot 10^{-15}$	$8 \cdot 10^{-14}$	$7 \cdot 10^{-13}$	$6 \cdot 10^{-12}$

Table 7. Problem 7

n	12	120	$12 \cdot 10^2$	$12 \cdot 10^3$
DSM	0.04	0.2	0.2	0.3
IC	$2 \cdot 10^{-15}$	10^{-15}	∞	∞
MIC	10^{-15}	$2 \cdot 10^{-15}$	$3 \cdot 10^{-15}$	$3 \cdot 10^{-15}$

The system of equations (12) splits into subsystems of dimension 4×4 and does not satisfy conditions (2). Its solution is determined by the formula

$$y_k = \cos(\pi k/2), \quad k = 1, 2, \dots, 4m.$$

The results of numerical experiments show (Table 6) that system (12) cannot be solved by the monotonic sweep method. Both the IC and MIC algorithms solve the system (12) well.

Problem 7:

$$\begin{aligned} & \cos\left(\pi \frac{k+1}{4}\right) y_{k-1} - \cos\left(\pi \frac{k}{4}\right) y_k + 2 \sin\left(\pi \frac{k}{3}\right) y_{k+1} \\ & = 2 \sin\left(\pi \frac{k}{3}\right) \cos\left(\pi \frac{k+2}{4}\right), \quad k = 1, 2, \dots, 12m. \end{aligned} \quad (13)$$

This is one more splitting system with a solution determined by the formula

$$y_k = \cos\left(\pi \frac{k+1}{4}\right), \quad k = 1, 2, \dots, 12m.$$

System (13) does not satisfy conditions (2) and (as shown in Table 7) cannot be solved by the monotonic sweep method. The IC algorithm in case of a large number of equations in the system is also not efficient, but the MIC method solves the system (13) well.

Thus, the MIC algorithm turned out to be efficient in solving all the problems presented in the present paper, while the monotonic sweep method has a limited area of application. The IC algorithm, as shown by the numerical experiments, may cause accumulation of errors at a large number of equations in the system.

REFERENCES

1. Faddeev, D.K. and Faddeeva, V.N., *Vychislitel'nye metody lineinoi algebry* (Computational Methods of Linear Algebra), Moscow: Fizmatgiz, 1963.
2. Voevodin, V.V., *Vychislitel'nye osnovy lineinoi algebry* (Computational Fundamentals of Linear Algebra), Moscow: Nauka, 1977.
3. Patankar, S., *Chislennyye metody resheniya zadach teploobmena i dinamiki zhidkosti* (Numerical Methods for Solving Problems of Heat Transfer and Fluid Dynamics), Moscow: Energoatomizdat, 1984.
4. Samarskii, A.A. and Gulin, A.V., *Chislennyye metody* (Numerical Methods), Moscow: Nauka, 1989.

5. Samarskii, A.A. and Nikolaev, E.S., *Metody resheniya setochnykh uravnenii* (Numerical Methods for Grid Equations), Moscow: Nauka, 1978.
6. Marchuk, G.I. and Sarkisyan, A.S., *Matematicheskoe modelirovanie tsirkulyatsii okeana* (Mathematical Modelling of Ocean Circulation), Moscow: Nauka, 1988.
7. Kurosh, A.G., *Kurs vysshei algebry* (Course of Higher Algebra), Moscow: Nauka, 1971.
8. Il'in, A.M., Differencing Scheme for a Differential Equation with a Small Parameter Affecting the Highest Derivative, *Mat. Zametki*, 1969, vol. 6, no. 2, pp. 237–248.
9. Doolan, E., Miller, J., and Schilders, W., *Ravnomernye chislennye metody resheniya zadach s pograničnym sloem* (Uniform Numerical Methods for Problems with Initial and Boundary Layers), Moscow: Mir, 1983.
10. Il'in, V.P., Direct Stability Analysis of the Double-Sweep Method, in *Aktual'nye problemy vychislitel'noi matematiki i matematicheskogo modelirovaniya* (Advanced Problems of Computational Mathematics and Mathematical Modeling), Novosibirsk: Nauka, 1985, pp. 189–201.