Journal of Interdisciplinary Mathematics
ISSN: 0972-0502 (Print), ISSN: 2169-012X (Online)
Vol. 27 (2024), No. 3, pp. 601–617
DOI : 10.47974/JIM-1860

A general solution for a special tri-diagonal linear system

Peerayuth Charnsethikul [§]
Chusana Nuntanart [†]
Department of Industrial Engineering
Kasetsart University
Bangkok 10900
Thailand

Aphisak Witthayapraphakorn *
Department of Industrial Engineering
Faculty of Engineering University of Phayao
Phayao 56000
Thailand

Abstract

The tri-diagonal linear system with coefficients of (–1, 2, –1) along the bandwidth and the transposed vector of (1^k, 2^k, …., n^k) as the right hand sides is solved in a general form linking with the classical Faulhaber's Formula. We derive the general solution for any k and conduct computational experiments compared with using the direct Thomas's algorithm. The result shows that in the case where k = 1 to 16, the general solution clearly performs more efficiently. For cases where k = 1, the results also indicate that as the problem size continuously expands, there will always be a point at which the general solution processes faster.

Subject Classification: *65F05, 15A06.*

Keywords: *Faulhaber's formula, Tri-diagonal linear system, Thomas's algorithm, General solution.*

1. Introduction

This work is aimed to derive a general solution for the following mathematical problem.

[§] *E-mail:* **fengprc@ku.ac.th**
[†] *E-mail:* **dew_159@hotmail.com**
* *E-mail:* **aphisak.wi@up.ac.th (Corresponding Author)**

$$\begin{gathered} 2x_1 - x_2 = 1^k \\ -x_{j-1} + 2x_j - x_{j+1} = j^k \ , j = 2,3,4,\ldots,n-1, \text{ (P)} \\ -x_{n-1} + 2x_n = n^k \\ (k = 0,1,2,3,\ldots\ldots) \end{gathered}$$

It is well known that the left-hand sides of the above equation system have mainly been motivated by finite difference approximation of a second order derivative term normally existed in physical governing equations. While the right-hand sides are normalized from k^{th} order polynomial function as system input representation. Newton's second law is such an example in this case (see [1]). Also, some finite element model [2] is in the form of the above system. The proposed problem can be classified as a class of tri-diagonal linear system where there exists the well-known Thomas's algorithm [3] as one of the oldest and most efficient technique. However, the problem can be very large and sparse with some numerical instability ask grows. Therefore, the purpose of this work is to derive a general solution form of the above problem and it was found that the key result is surprisingly related to the classical Faulhaber's formula [4, 5]. A preliminary computational experience will also be presented.

2. Literature review

The tri-diagonal linear system (TDLS) has been a focal point of mathematical research due to its critical role in a wide range of applications, including neural networks. This interest stems from the necessity to solve systems of linear equations, particularly in the context of computational fluid dynamics (CFD) [6] and neural network models with delays [7]. Over time, a plethora of solutions have been proposed, each characterized by its unique methodology and application. For instance, Wang et al. investigated the stability and Hopf bifurcation of a general tri-diagonal bidirectional associative memory (BAM) neural network model with 2n-neurons, considering forward transmission delay and feedback delay [7]. The Quasi-Linearization technique and an implicit finite difference scheme were employed by Patil and Kulkarni to solve a block TDLS using Verga's algorithm [8]. El-Mikkawy introduced a novel computational algorithm, PERTRI, for solving periodic TDLS, which was a generalization of Thomas's algorithm [9]. Yambangwai et al. proposed a new weight tri-diagonal iterative method that uses a local weight parameter to accelerate the iteration process [10]. Ke et al. introduced a fast-direct method for

solving a specific type of linear system using a divide-and-conquer strategy and fast Fourier transforms [11]. Warne et al. presented a novel method for solving TDLS using Field Programmable Gate Arrays (FPGAs) [12]. Zhao and Yu introduced a novel GPU shared memory chunked cyclic reduction algorithm for solving TDLS [13]. Krishna Prasad and Patil employed the Thomas algorithm, a simplified form of Gaussian elimination, to solve these systems [14]. Mohanty presented a three-level implicit unconditionally stable difference scheme for the difference solution of second order linear hyperbolic equations [15]. Huang et al. utilized a parallelized Gaussian elimination method to solve these systems [16]. Guo et al. presented two algorithms for solving the Karush-Kuhn-Tucker conditions for optimality in a quadratic program [17]. Mohanty introduced a novel method for solving second order boundary value problems using a three-point arithmetic average discretization scheme on a non-uniform mesh, applied to a TDLS [18]. Patchkovskii and Muller proposed a method for solving the TDLS in the context of the time-dependent Schrödinger equation for one-electron atomic systems [19]. Amidst these diverse solutions, an intriguing proposition is the potential application of Faulhaber's formula, known for its ability to express the sum of the p-th powers of the first n positive integers, in solving the TDLS.

The application of Faulhaber's formula in problem-solving is a recurring theme in various research works, including its potential role in solving the tri-diagonal linear system (TDLS). Barbero G. et al. introduced a Faulhaber-like formula involving Bernoulli polynomials, suggesting the potential for generalizing the formula for fractional values of N [20]. Gnewuch et al. derived a generalized Faulhaber inequality, improving known bounds for bracketing numbers of d-dimensional axis-parallel boxes [21]. Ono and Yamamoto extended the refined Kaneko–Zagier conjecture to general integer indices and employed Faulhaber's formula on power sums to prove one of their main theorems, thereby demonstrating its applicability in solving complex mathematical problems [22a]. McGown and Parks extended Faulhaber's formula to apply to non-integral complex powers, providing a corollary that explains the discrepancy between the sum of the a-th powers of the first N positive integers and a certain function of N [23]. Chen et al. expanded upon the classic Faulhaber's theorem regarding sums of odd powers, demonstrating its applicability to any arbitrary arithmetic progression [24]. Kellner and Sondow investigated the denominators of the power sum, proving that such a denominator equals (n + 1) times the square-free product of certain primes 'p' that fulfill a specific condition [25]. Fairlie and Veselov uncovered a relationship

between classical Faulhaber and Bernoulli polynomials and the theory of the Korteweg–de Vries (KdV) equation [26]. Kilar and Simsek explored Faulhaber's 1631 formula for the sums of powers of positive integers, proposing a new formula for the sums of powers of positive integers [27]. Parks proposed an approximation for the sum of non-integral powers of positive integers, expanding Faulhaber's formula [28]. Guo et al. offered a q-analogue of Faulhaber's formula, demonstrating the existence of integer coefficients for which certain formulas hold true [29]. Wang et al. proposed a new algorithm to generate a formula for the sum of integer powers of an arbitrary positive integer, based on a special linear system derived from Faulhaber's formula [30]. These studies collectively underscore the versatility and applicability of Faulhaber's formula in solving complex mathematical problems, including its potential as a basis for solving the TDLS, the main theme of this research.

3. Methodology

To solve for the general solution, firstly consider rearrange problem (P) by identifying x_j, j=2, 3,…., n in term of x_1 as follows.

$$x_2 = 2x_1 - 1^k$$
$$x_3 = 3x_1 - [(1^k + 2^k) + 1^k]$$
$$x_4 = 4x_1 - [(1^k + 2^k + 3^k) + (1^k + 2^k) + 1^k]$$
$$x_5 = 5x_1 - [(1^k + 2^k + 3^k + 4^k) + (1^k + 2^k + 3^k) + (1^k + 2^k) + 1^k]$$
$$x_j = jx_1 - [S_{j-1}^k + S_{j-2}^k + ... + S_3^k + S_2^k + S_1^k] \quad .$$

$$\left(\text{where } S_n^k = 1^k + 2^k + 3^k ... + n^k = \sum_{i=1}^{n} i^k \right)$$

This implies

$$x_j = jx_1 - \sum_{i=1}^{j-1} S_i^k \qquad (1)$$

.

.

$$x_n = nx_1 - \sum_{i=1}^{n-1} S_i^k$$

Therefore,

$$-x_{n-1}+2x_n=-(n-1)x_1+\sum_{i=1}^{n-2}S_i^k+2nx_1-2\sum_{i=1}^{n-1}S_i^k=n^k$$

$$(n+1)x_1=\sum_{i=1}^{n-1}S_i^k+S_{n-1}^k+n^k=\sum_{i=1}^{n}S_i^k$$

$$x_1=\frac{1}{n+1}\sum_{i=1}^{n}S_i^k \tag{2}$$

Consider equations (1) and (2), deriving the general solution of problem (P) is relied on solving the general form of S_n^k belong to the classical Faulhaber's formula which has been studied since 1890. Moreover, solving (P) for an arbitrary k need S_n^{k+1} since S_n^k is in the form of $(k+1)^{\text{th}}$ polynomial function.

$$S_n^{k+1}=1*1^k+2*2^k+3*3^k+\ldots..+(n-1)*(n-1)^k+n*n^k$$

$$=(1^k+2^k+3^k+\ldots+n^k)+(2^k+3^k+\ldots+n^k)+(3^k+\ldots+n^k)+\ldots+((n-1)^k+n^k)+n^k$$

$$=nS_n^k-\sum_{i=1}^{n-1}S_i^k=(n+1)S_n^k-\sum_{i=1}^{n}S_i^k$$

$$\sum_{i=1}^{n}S_i^k=(n+1)S_n^k-S_n^{k+1} \tag{3}$$

$$S_n^k=\frac{1}{k+1}\sum_{j=0}^{k}\binom{k+1}{j}B_j n^{k+1-j} \tag{4}$$

$$x_1=S_n^k-\frac{S_n^{k+1}}{n+1} \tag{5}$$

$$x_j=j(x_1-S_{j-1}^k)+S_{j-1}^{k+1},\quad j=2,3,..,n \tag{6}$$

The coefficient B_j is j^{th} Bernoulli number that can be computed as described in https://mathworld.wolfram.com/BernoulliNumber.html. In this work, the number from $j = 0$ up to 40 is given as the following table where in case of $j = 3, 5, 7, 9, \ldots, 39$, $B_j = 0$.

j	B_j	j	B_j
0	1	20	-174611/330
1	±1/2	22	854513/138
2	1/6	24	-236364091/2730
4	-1/30	26	8553103/6
6	1/42	28	-23749461029/870
8	-1/30	30	8615841276005/14322
10	5/66	32	-7709321041217/510
12	-691/2730	34	2577687858367/6
14	7/6	36	-26315271553053477373/1919190
16	-3617/510	38	2929993913841559/6
18	43867/798	40	-261082718496449122051/13530

In this work, the formulas up to $k = 16$ from [5] are derived using (3) and (4) and can be presented as the following rearrangement formulas.

$$\sum_{i=1}^{n} i = \frac{n^2+n}{2}$$

$$\sum_{i=1}^{n} i^2 = \frac{2n^3+3n^2+n}{6}$$

$$\sum_{i=1}^{n} i^3 = \frac{n^4+2n^3+n^2}{4}$$

$$\sum_{i=1}^{n} i^4 = \frac{6n^5+15n^4+10n^3-n}{30}$$

$$\sum_{i=1}^{n} i^5 = \frac{2n^6+6n^5+5n^4-n^2}{12}$$

$$\sum_{i=1}^{n} i^6 = \frac{6n^7+21n^6+21n^5-7n^3+n}{42}$$

$$\sum_{i=1}^{n} i^7 = \frac{3n^8+12n^7+14n^6-7n^4+2n^2}{24}$$

$$\sum_{i=1}^{n} i^8 = \frac{10n^9+45n^8+60n^7-42n^5+20n^3-3n}{90}$$

$$\sum_{i=1}^{n} i^9 = \frac{2n^{10}+10n^9+15n^8-14n^6+10n^4-3n^2}{20}$$

$$\sum_{i=1}^{n} i^{10} = \frac{6n^{11}+33n^{10}+55n^9-66n^7+66n^5-33n^3+5n}{66}$$

$$\sum_{i=1}^{n} i^{11} = \frac{2n^{12}+12n^{11}+22n^{10}-33n^8+44n^6-33n^4+10n^2}{24}$$

$$\sum_{i=1}^{n} i^{12} = \frac{210n^{13}+1365n^{12}+2730n^{11}-5005n^9+8580n^7-9009n^5+4550n^3-691n}{2730}$$

$$\sum_{i=1}^{n} i^{13} = \frac{30n^{14}+210n^{13}+455n^{12}-1001n^{10}+2145n^8-3003n^6+2275n^4-691n^2}{420}$$

$$\sum_{i=1}^{n} i^{14} = \frac{6n^{15}+45n^{14}+105n^{13}-273n^{11}+715n^9-1287n^7+1365n^5-691n^3+105n}{90}$$

$$\sum_{i=1}^{n} i^{15} = \frac{3n^{16}+24n^{15}+60n^{14}-182n^{12}+572n^{10}-1287n^8+1820n^6-1382n^4+420n^2}{48}$$

$$\sum_{i=1}^{n} i^{16} = \frac{30n^{17}+255n^{16}+680n^{15}-2380n^{13}+8840n^{11}-24310n^9+44200n^7-46988n^5+23800n^3-3617n}{510}$$

$$\sum_{i=1}^{n} i^{17} = \frac{10n^{18}+90n^{17}+255n^{16}-1020n^{14}+4420n^{12}-14586n^8+33150n^8-46988n^6+35700n^4-10851n^2}{180}$$

4. Results

The results of S_n^k for $k = 1, 2, 3, \ldots, 17$ were used in equations (1) and (2) and the derived solutions is in the following statements.

If $k = 0$ then

$$x_1 = \frac{n}{2} \text{ and } x_j = jx_1 - \frac{j(j-1)}{2}$$

If $k = 1$ then

$$x_1 = \frac{n(n+2)}{6} \text{ and } x_j = jx_1 - \frac{j(j-1)(j+1)}{6}$$

If $k = 2$ then

$$x_1 = \frac{n(n+1)(n+2)}{12} \text{ and } x_j = jx_1 - \frac{j^2(j-1)(j+1)}{12}$$

If $k = 3$ then

$$x_1 = \frac{n(n+2)(3n^2+6n+1)}{60} \text{ and } x_j = jx_1 - \frac{j(j-1)(j+1)(3j^2-2)}{60}$$

If $k = 4$ then

$$x_1 = \frac{n(n+1)(n+2)(2n^2+4n-1)}{60} \text{ and } x_j = jx_1 - \frac{j^2(j-1)(j+1)(2j^2-3)}{60}$$

If $k = 5$ then

$$x_1 = \frac{n(n+2)(2n^4+8n^3+7n^2-2n-1)}{84} \text{ and } x_j = jx_1 - \frac{j(j-1)(j+1)(2j^4-5j^2+2)}{84}$$

If $k = 6$ then

$$x_1 = \frac{n(n+2)(n+1)(3n^4+12n^3+7n^2-10n+2)}{168} \text{ and } x_j = jx_1 - \frac{j^2(j-1)(j+1)(2j^4-5j^2+2)}{168}$$

If $k = 7$ then

$$x_1 = \frac{n(n+2)(5n^6+30n^5+50n^4-37n^2+6n+6)}{360}$$

and

$$x_j = jx_1 - \frac{j(j-1)(j+1)(5j^6-25j^4+38j^2-12)}{360}$$

If $k = 8$ then

$$x_1 = \frac{n(n+2)(n+1)(2n^6+12n^5+17n^4-12n^3-19n^2+18n-3)}{180}$$

and

$$x_j = jx_1 - \frac{j^2(j-1)(j+1)(2j^6-13j^4+29j^2-21)}{180}$$

If $k = 9$ then

$$x_1 = \frac{n(n+2)(6n^8+48n^7+119n^6+42n^5-166n^4-48n^3+146n^2+12n-25)}{660}$$

and

$$x_j = jx_1 - \frac{j(j-1)(j+1)(6j^8-49j^6+149j^4-181j^2+50)}{660}$$

If $k = 10$ then

$$x_1 = \frac{n(n+2)(n+1)(6n^8+48n^7+108n^6-24n^5-243n^4+84n^3+267n^2-210n+30)}{792}$$

and

$$x_j = jx_1 - \frac{j^2(j-1)(j+1)(6j^8-60j^6+237j^4-423j^2+270)}{792}$$

If $k = 11$ then

$$x_1 = \frac{n(n+2)(70n^{10}+700n^9+2310n^8+1680n^7-4665n^6-4410n^5+8240n^4+4120n^3-7819n^2+202n+1382)}{10920}$$

$$\text{and } x_j = jx_1 - \frac{j(j-1)(j+1)(70j^{10}-840j^8+4165j^6-10135j^4+10886j^2-2764)}{10920}$$

If $k = 12$ then

$$x_1 = \frac{n(n+2)(n+1)(30n^{10}+300n^9+925n^8+200n^7-3022n^6-772n^5+7073n^4-1228n^3-7888n^2+5528n-691)}{5460}$$

$$\text{and } x_j = jx_1 - \frac{j^2(j-1)(j+1)(30j^8-425j^8+2578j^6-8174j^4+12874j^2-7601)}{5460}$$

If $k = 13$ then

$$x_1 = \frac{n(n+2)(6n^{12}+72n^{11}+297n^{10}+330n^9-765n^8-1368n^7+2059n^6+2995n^5-4091n^4-2724n^3+4069n^2+66n-735)}{1260}$$

$$\text{and } x_j = jx_1 - \frac{j(j-1)(j+1)(6j^{12}-99j^{10}+720j^8-2855j^6+6154j^4-6131j^2+1470)}{1260}$$

If $k = 14$ then

$$x_1 = \frac{n(n+2)(n+1)(3n^{12}+36n^{11}+141n^{10}+90n^9-591n^8-552n^7+2123n^6+1170n^5-5182n^4+336n^3+5846n^2-3780n+420)}{720}$$

$$\text{and } x_j = jx_1 - \frac{j^2(j-1)(j+1)(3j^{12}-57j^{10}+489j^8-2371j^6+6638j^4-9742j^2+5460)}{720}$$

If k=15 then

$$x_1 = \frac{n(n+2)(15n^{14}+210n^{13}+1040n^{12}+1560n^{11}-3190n^{10}-9020n^9+11340n^8+32640n^7-35515n^6-68130n^5+76520n^4+61400n^3-78806n^2-3852n+14468)}{4080}$$

and

$$x_j = jx_1 - \frac{j(j-1)(j+1)(15j^{14}-325j^{12}+3245j^{10}-18855j^8+66230j^6-132670j^4+125764j^2-28936)}{4080}$$

If k=16 then

$$x_1 = \frac{n(n+2)(n+1)(10n^{14}+140n^{13}+665n^{12}+700n^{11}-3345n^{10}-5730n^9+16145n^8+23800n^7-62363n^6-47698n^5+156657n^4+1788n^3-178173n^2+10851n-10851)}{3060}$$

and

$$x_j = jx_1 - \frac{j^2(j-1)(j+1)(10j^{14}-245j^{12}+2815j^{10}-19285j^8+82817j^6-215533j^4+301335j^2-162765)}{3060}$$

For the results with k > 16, Faulhaber formulas with higher degrees of polynomial are as shown in (5) and (6).

Computationally, a preliminary test is conducted using the case of k = 1, 5, 9, 13 with very large n. Thomas's algorithm (TA) is coded as a Matlab

Table 1

Demonstrates a comparison of processing efficiency between GS and TA

Size (n x10^6)	Time (Sec.)									
	k=1		k=5		k=9		k=13		k=16	
	GS	TA	GS	TA	GS	TA	GS	TA	GS	TA
100	0.42	1.93	15.58	1.93	44.37	2.00	73.69	2.15	88.58	1.90
200	0.82	3.80	31.31	4.04	89.36	4.01	150.62	4.71	177.36	3.94
300	1.20	5.80	46.54	6.00	134.20	5.82	221.88	6.30	265.98	6.02
400	1.70	8.06	63.18	8.21	178.61	8.18	295.82	8.29	362.71	8.94
500	2.17	11.18	78.87	11.34	223.22	9.66	369.37	9.94	454.83	10.63
600	3.26	13.45	94.40	11.74	268.35	12.18	453.41	12.84	542.92	14.66
700	5.49	26.75	109.67	16.08	316.00	30.18	529.67	45.16	626.63	34.74
800	7.26	79.08	125.62	73.73	362.68	69.08	603.20	102.75	707.51	92.32
900	9.97	110.13	141.96	117.02	402.38	112.37	668.92	138.95	797.34	115.19
1,000	11.98	207.17	158.11	164.10	447.78	166.64	743.48	199.73	898.67	205.71
1,100	13.84	-	172.33	-	497.53	-	850.31	-	1,004.28	-

program [3] and run comparing with the use of results (the general solution (GS) in case of k = 1, 5, 9, 13 and 16) in the previous section also coded as another Matlab code. Both programs were run on a desktop microcomputer with CPU i7-7700 CPU @ 3.60GHz and 32 Gb RAM. The results can be presented in the following Table 1.

From the experimental results in Table 1, it can be observed that when k=1, the processing time of GS is less than that of TA for all problem sizes. However, when k is increased, such as in the case of k=5, it can be observed that GS takes more processing time than TA in almost every instance. The only exception is that the rate of increase in processing time according to problem size in the same k scenario is slower for GS than TA, a trend that is distinctly noticeable. Furthermore, in cases where the problem size is large, as in the instance where n=1,100,000,000, using TA method in Matlab results in an error message stating, 'render process terminated: null.' This error message usually indicates that the process has run out of memory and is not able to allocate more. To illustrate the relationship of the rate of increase in processing time for each k scenario, a graph representing the correlation between processing time and problem size is created as shown in Figure 1.

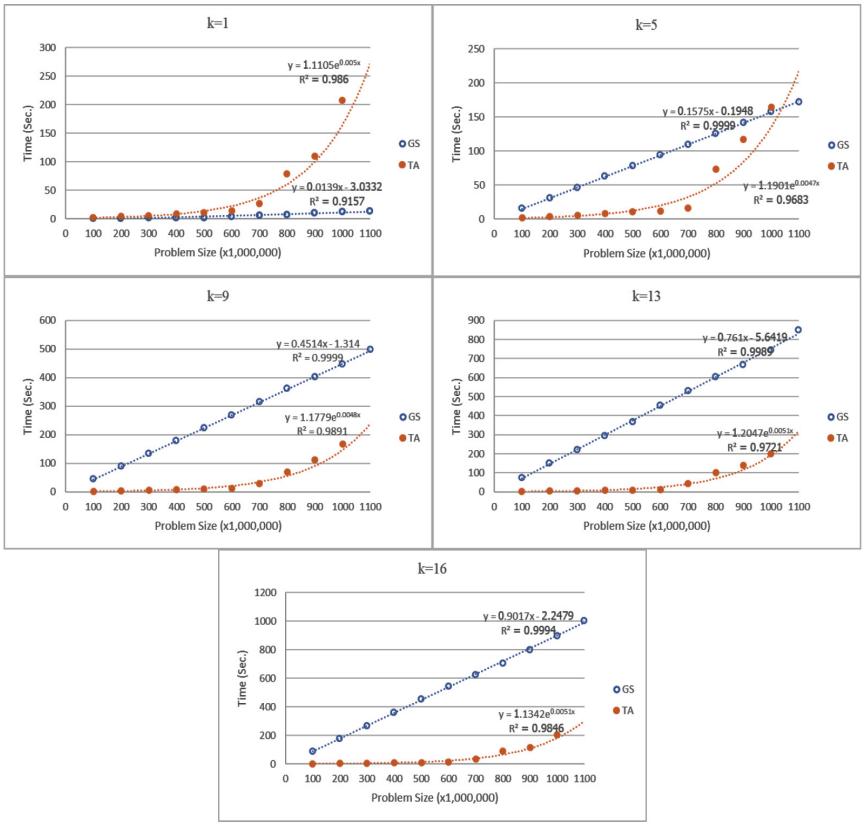

Figure 1

Graph depicting the relationship between time and problem-size.

From Figure 1, in the same k case, as the problem size increases, the relationship between processing time and problem size for TA tends to be exponential, while for GS, it tends to be linear. This trend implies that once the problem size reaches a certain level, GS always outperforms TA, in cases where TA can still process and find a solution. At the same time, in cases where the problem size is constant but k increases, GS shows a notable increase in processing time, while TA's processing time trend remains relatively stable. To illustrate the relationship of processing time and the rate of increase of k value, a comparison is made using problem sizes n=300, 600, 900 (x10^6), providing a more distinct illustration as shown in Figure 2.

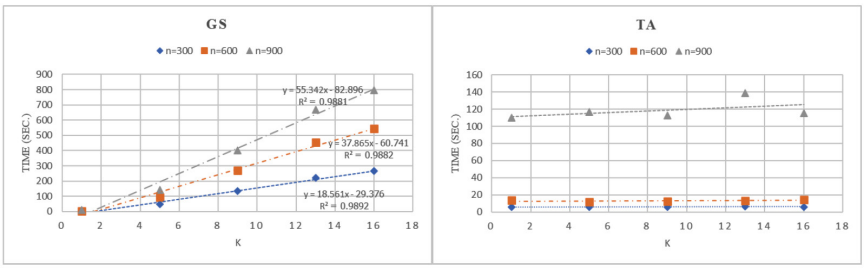

Figure 2

The graph illustrates the relationship between processing time and the value of *k*.

From Figure 2, it is clearly illustrated that an increase in the value of *k* results in a distinct increase in the processing time for GS for the same problem size, following a linear trend. In contrast, for TA, an increase in the value of *k* barely affects the processing time for the same problem size. This phenomenon is likely because as the value of *k* increases, the formula to find the solution for the GS method becomes longer, requiring more computational steps. This is different from TA, where despite an increase in *k*, the computational steps remain the same, resulting in the increase of *k* having little effect on TA.

From all the past experimental results, it can be concluded that an increase in the problem size (n) has a greater impact on the processing time of TA than GS. Conversely, an increase in the value of *k* significantly affects the processing time of GS, unlike TA, which is barely affected. Therefore, it can be stated that GS is a suitable choice when *k* is small. However, when *k* is large and the problem size is not significantly large, TA might be a more appropriate choice.

In the case where $k>16$, if a mathematical formula is needed, the formula would be too lengthy. However, we can construct an algorithm (hereinafter referred to as the general solution algorithm; GSA) using equations (3), (4), (5), and (6) to find the solution from $k=2$ to $k=39$. The variable definitions are as follows: S_{j-1}^{k} = sk(j-1), S_{j-1}^{k+1} = skplusone(j-1), x_j = x(j), x_1 = x (1), and the solution process is as illustrated in Figure 3. The procedures for finding various values will follow the commands of MATLAB.

To test the efficiency of GSA in finding solutions, an experiment was conducted to compare the processing speed with TA. The size of the problem had to be reduced for GS case, as the GSA takes longer to process. The results of the experiment are presented in Table 2.

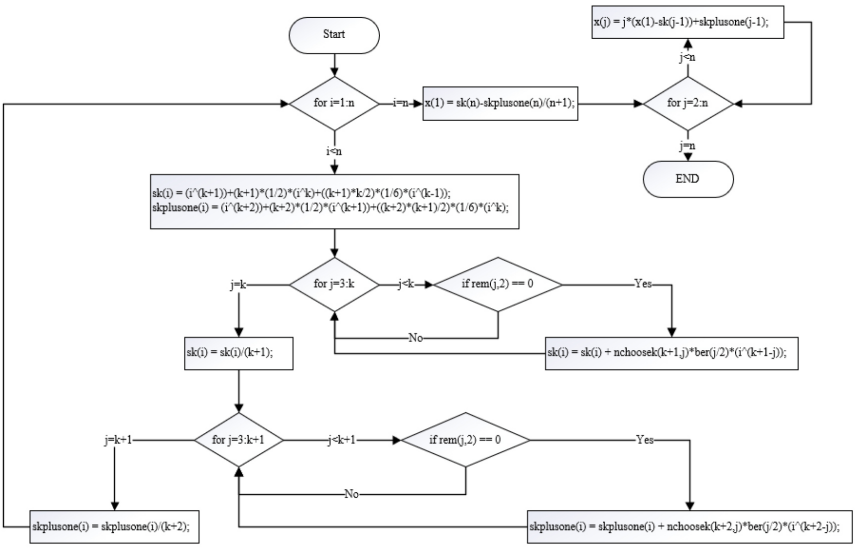

Figure 3

The process of the general solution algorithm.

Table 2

Demonstrates a comparison of processing efficiency between GSA and TA

Size (n x10^6)	Time (Sec.)							
	k=27		k=31		k=35		k=39	
	GSA	TA	GSA	TA	GSA	TA	GSA	TA
1	54.74	0.04	69.18	0.04	79.34	0.04	91.36	0.04
2	116.75	0.06	133.21	0.06	151.95	0.08	175.63	0.07
3	170.29	0.07	204.46	0.08	230.27	0.09	261.21	0.08
4	228.52	0.08	270.42	0.10	301.33	0.10	352.61	0.08
5	285.93	0.10	331.25	0.12	379.85	0.11	439.19	0.10
6	346.37	0.13	404.06	0.13	454.53	0.12	532.42	0.12
7	401.25	0.15	470.32	0.14	533.32	0.17	616.03	0.13
8	456.43	0.19	547.46	0.18	612.87	0.19	701.89	0.17
9	509.48	0.20	613.75	0.18	688.65	0.19	786.79	0.20
10	555.75	0.22	676.00	0.24	768.73	0.20	877.83	0.21

Table 2 clearly shows that the processing time of GSA is higher than that of TA. Therefore, in terms of processing efficiency, it would be preferable, if possible, to transform the GSA into the form of GS. However,

Table 3
Compares the answers of GSA and TA for the case where k=39 and n=2.

x_j	GSA	TA
x_1	183251933957.463	183251937963.333
x_2	366503867908.449	366503875925.667

there is a minor additional observation from this experiment regarding the accuracy of the answers. In cases where k is not large, the GSA and TA yield identical results. But if k is large, such as when k=39 and n=2, the values of the answers will differ, as shown in Table 2

Table 3 reveals that the answers provided by GSA differ from those of TA. Upon examining the accuracy, it is found that the GSA provides answers that are closer to the true values. This discrepancy occurs because the solutions obtained using TA involve successive divisions to find continuous answers, leading to a series of rounding errors. This iterative rounding can cause the answers to deviate from the true values. In contrast, the GSA involves division by constants such as k+1, k+2, and n+1, resulting in answers that are more closely aligned with the correct values.

From the experimental results concerning the GSA, it can be concluded that this method has the advantage of providing more accurate results than TA, especially when k has higher values. However, its drawback is a clear inefficiency in processing compared to TA. Therefore, if possible, transforming the GSA into the form of GS would be a preferable approach.

5. Conclusion

In this paper, we proposed a general solution form for the special case of tri-diagonal linear system with coefficients of (-1,2,-1) along the bandwidth and the transpose vector of (1^k, 2^k, …., n^k) as the right hand sides with k = 1 to 16. Experimental results indicate that the general solution form is appropriate for solving problems of large size where k is small. Meanwhile, Thomas's algorithm is suitable for problems where k is large, and the problem size is not excessively large (n<=1,000,000,000). Moreover, independent of the value of k, as the problem size continues to grow, there will inevitably be a point where the general solution form surpasses Thomas's algorithm in performance. Additionally, we have developed the general solution algorithm, adhering to the principles of

constructing a general solution form, to find answers for cases where k ranges from 2 to 39. Experimental results indicate that the general solution algorithm is less efficient in processing compared to Thomas's algorithm. However, it has the advantage of providing more accurate answers when k has higher values.

Conflict of Interest

The authors declare that they have no conflict of interest.

Data Availability Statement

The data that support the findings of this study are available from the corresponding author upon reasonable request.

References

[1] Strang, G., Linear Algebra and its Applications, 4th edition, Brook Coles (2005).

[2] Zienkiewicz, O.C., Taylor, R. L. and Zhu, J.Z., The Finite Element Method: Its Basis and Fundamentals, 6th Edition, Butterworth-Heinemann (2005).

[3] http://en.wikipedia.org/wiki/Tridiagonal_matrix_algorithm.

[4] Dubeau, F., "Linear algebra and the sums of powers of integers", *Electronic Journal Linear Algebra*, Vol. 17, pp. 577-596 (2008).

[5] Knuth, D., "Johann Faulhaber and sums of powers", *Mathematics of Computation,* Vol. 61, no. 203, pp.277-294 (1993).

[6] Parag V. Patil et al., "Algorithm: Three Dimensional Finite Volume Numerical Grid Technique", *Global Journal of Pure and Applied Mathematics,* Vol. 13, Number 9, pp. 5655-5671 (2017).

[7] T. Wang, Y. Wang, and Z. Cheng, "Stability and Hopf Bifurcation Analysis of a General Tri-diagonal BAM Neural Network with Delays", *Neural Processing Letters*, vol. 53, pp. 4571–4592, Aug. (2021).

[8] Patil, Kulkarni, "A numerical study on MHD double diffusive nonlinear mixed convective nanofluid flow around a vertical wedge with diffusion of liquid hydrogen", *Journal of the Egyptian Mathematical Society*, Vol. 29, Article number: 24 (2021).

[9] M.E.A. El-Mikkawy, "A new computational algorithm for solving periodic tri-diagonal linear systems", *Applied Mathematics and Computation*, Vol. 161, Issue 2, pp. 691-696 (2005).

[10] D. Yambangwai, W. Cholamjiak, T. Thianwan, H. Dutta, "On a new weight tri-diagonal iterative method and its applications", *Soft Computing,* vol 25, pp.725–740 (2020).

[11] Rihuan Ke, Michael K. Ng, Hai-Wei Sun, "A fast direct method for block triangular Toeplitz-like with tri-diagonal block systems from time-fractional partial differential equations", *Journal of Computational Physics,* Vol. 303, pp. 203-211 (2015).

[12] D. J. Warne, N. A. Kelson, R. F. Hayward, "Solving Tri-Diagonal Linear Systems using Field Programmable Gate Arrays", 4th International Conference on Computational Methods (ICCM 2012) (2012).

[13] Di Zhao, Jinhang Yu, "Efficiently solving tri-diagonal system by chunked cyclic reduction and single-GPU shared memory", *The Journal of Supercomputing*, Vol. 71, pp. 369–390 (2014).

[14] J. S. V. R. Krishna Prasad, Parag V. Patil, "Algorithm for Solving Tridiagonal Finite Volume Discretized Linear Systems", *Applications and Applied Mathematics: An International Journal*, Vol. 10, Issue 2, pp. 995-1006 (2015).

[15] R.K. Mohanty, "An unconditionally stable finite difference formula for a linear second order one space dimensional hyperbolic equation with variable coefficients", *Applied Mathematics and Computation*, Vol. 165, Issue 1, pp. 229-236 (2005).

[16] Yao Huang, Bing-Jia Xiao, Zheng-Ping Luo, "Fast parallel Grad–Shafranov solver for real-time equilibrium reconstruction in EAST tokamak using graphic processing unit", *Chinese Physics B*, Vol. 26, Number 8 (2017).

[17] Meichen Guo, Adair Lang, Michael Cantoni, "Structured Moving Horizon Estimation for Linear System Chains", 18th European Control Conference (ECC) (2019).

[18] R.K. Mohanty, "A class of non-uniform mesh three point arithmetic average discretization for y'' = f(x,y,y') and the estimates of y' ", *Applied Mathematics and Computation*, Vol. 183, Issue 1, pp. 477-485 (2006).

[19] Serguei Patchkovskii, H.G. Muller, "Simple, accurate, and efficient implementation of 1-electron atomic time-dependent Schrödinger

equation in spherical coordinates", *Computer Physics Communications*, Vol. 199, pp. 153-169 (2016).

[20] J. F. Barbero G., J. Margalef-Bentabol, and E. J. S. Villaseñor, "A two-sided Faulhaber-like formula involving Bernoulli polynomials", *Comptes Rendus Mathématique*, Vol. 358, issue 1, pp. 41-44 (2020).

[21] M. Gnewuch, H. Pasing, and C. Weiß, "A Generalized Faulhaber Inequality, Improved Bracketing Covers, and Applications to Discrepancy", *Mathematics of Computation*, Vol. 90, pp. 2873-2898 (2021).

[22] M. Ono and S. Yamamoto, "On the refined Kaneko–Zagier conjecture for general integer indices", *Mathematische Zeitschrift*, vol. 305, no. 1, pp. 1-13, Aug. (2023).

[23] K. J. McGown and H. R. Parks, "The generalization of Faulhaber's formula to sums of non-integral powers", *Journal of Mathematical Analysis and Applications*, Vol. 330, Issue 1, pp. 571-575 (2007).

[24] W. Y. C. Chen, A. M. Fu, and I. F. Zhang, "Faulhaber's theorem on power sums", *Discrete Mathematics*, Vol. 309, Issue 10, pp. 2974-2981 (2009).

[25] B. C. Kellner and J. Sondow, "Power-Sum Denominators", *The American Mathematical Monthly*, Vol. 124, No. 8, pp. 695-709 (2017).

[26] D. B. Fairlie and A. P. Veselov, "Faulhaber and Bernoulli Polynomials and Solitons", *Physica D: Nonlinear Phenomena*, Vol. 152–153, pp. 47-50 (2001).

[27] N. Kilar and Y. Simsek, "Formulas Involving Sums of Powers, Special Numbers and Polynomials Arising from p-Adic Integrals, Trigonometric and Generating Functions", Publications De L'institut Mathématique, Nouvelle série, tome 108(122), pp. 103–120 (2020).

[28] H. R. Parks, "Sums of non-integral powers", *Journal of Mathematical Analysis and Applications*, Vol. 297, Issue 1, pp. 343-349 (2004).

[29] V. J. W. Guo, M. Rubey, and J. Zeng, "Combinatorial interpretations of the q-Faulhaber and q-Salié coefficients", *Journal of Combinatorial Theory,* Series A, Vol. 113, Issue 7, pp. 1501-1515 (2006).

[30] M. J. Wang, S. Goel, and G. Mao, "A New Algorithm to Generate A Formula for the Sum of Integer Powers", Proceedings of the 2014 ACM Southeast Regional Conference, Article No.: 50, pp. 1–4 (2014).

Received October, 2023