2021 ASEE ANNUAL CONFERENCE
Virtual Meeting | July 26–29, 2021 | Pacific Daylight Time
Paper ID #32388

Lovelace's Program: A Challenging but Achievable Assignment for Undergraduate Students in Engineering and Computer Science

Dr. Erica Haugtvedt, South Dakota School of Mines and Technology

Dr. Erica Haugtvedt is an assistant professor of English and Humanities at South Dakota School of Mines & Technology. She received her PhD in British nineteenth-century literature from Ohio State University in 2015.

Dr. Duane L. Abata, South Dakota School of Mines and Technology

Dr. Abata has worked in academia for over forty years at universities and with the Federal government around the country. He began his career at the University of Wisconsin, served as Associate Dean and Dean at Michigan Technological University and then at the National Science Foundation in Washington, D.C. as program manager in the Engineering Directorate. From 2003 to 2004, Dr. Abata was President of the American Society for Engineering Education. Following his appointment at NSF he served as Dean of Engineering and Engineering Technology at Northern Arizona University and Dean of Engineering at South Dakota School of Mines and Technology. Dr. Abata is currently a tenured full professor in mechanical engineering at South Dakota School of Mines and Technology. His research work focuses in the areas of energy storage and combustion.

Lovelace’s Program: A Challenging but Achievable Assignment for Undergraduate Students in Engineering and Computer Science

Erica Haugtvedt and Duane Abata
South Dakota School of Mines & Technology
Rapid City, SD

Abstract

This paper discusses the world’s first purported program that calculates the first ten Bernoulli numbers written by Countess Ada Lovelace in 1842-43 for a computer, an “Analytical Engine” which was a mechanical system of cogs and gears, never constructed but rather envisioned by Charles Babbage in the 1830s. Lovelace represents an excellent mathematician and has been recognized as a significant female pioneer in computer technology. The program can be written in any modern-day programming languages and provides an excellent educational and pleasing pedagogical assignment for students in engineering and computer science that promotes interdisciplinary knowledge transfer.

Introduction

Interdisciplinary approaches to engineering education are widely recognized as necessary for today’s engineers to meet the societal challenges before them [1]. Among these societal challenges are the opportunities presented by diversity, both demographically and disciplinarily. A key goal of current engineering education practice is to “promote and sustain diversity” [2]. However, despite recruitment efforts, women remain a minority in engineering fields in the U.S. and conceptualizations of interdisciplinary pedagogical practice within engineering education tend to extend only to other science disciplines [3, 1]. Lattuca et. al.’s 2017 engineering education study found that a curricular emphasis on interdisciplinary topics and skills (specifically including participating in non-engineering co-curricular activities and humanitarian engineering projects), “significantly and positively relate to engineering students’ reports of interdisciplinary skills” [4]. In this paper, we suggest an interdisciplinary computer programming exercise based on Ada Lovelace’s 1843 program for Charles Babbage’s Analytical Engine, thus promoting knowledge transfer between the humanities (history) and computer science.

Every student of computer science should be aware of the Bernoulli numbers together with the names of Charles Babbage and Ada Lovelace. The Bernoulli numbers were critical to Victorian mathematics because incorporating these numbers in various algorithms tremendously simplified logarithmic and trigonometric calculations. As these complicated mathematical calculations were done by hand in the nineteenth century, they were subject to a high degree of human error that the introduction of the Bernoulli numbers radically reduced.

But, of course, the Bernoulli numbers did not fully eradicate human error. Frustrated by mistakes he found in tables of logarithms, Charles Babbage famous uttered, "I wish to God these calculations had been executed by steam!" [5]. In trying to do just that, Babbage devised a complex mechanical system of cogs and gears that became one of the world's first machines to perform mathematical manipulations of addition, subtraction, multiplication, and division called the "Difference Engine." The engine was never completed, although portions of it were constructed at great cost to the British government. Later he envisioned an "Analytical Engine" that could be programmed with punched plates, following the mechanical system of the Jacquard loom—a automatic loom that had earlier revolutionized the weaving industry.

The Difference and Analytical Engines were not a single physical machine, but rather a succession of designs that Babbage tinkered with until his death in 1871. The "Difference Engine" was only recently constructed and is functional. It was on display at the Computer History Museum in Mountain View, California from 2008 until 2016 [6]. Babbage presented his visionary work at a technical conference in Italy where scribes took notes which were then documented in French by the statesman and mathematician Luigi Menabrea [7]. Menabrea had played a key role in organizing the conference as professor of mechanics and construction at the military academy and at the University of Turin. Ada Lovelace was tasked to translate these notes from French to English. In the footnotes to her translation, Lovelace developed a program to calculate the first ten Bernoulli numbers. She included an outline of her program as an appendix to the translation [8] [see appendix A for the program translated into C++].

Who Was Ada Lovelace?

Ada Lovelace, daughter of the famed poet Lord Byron and known as Ada Byron, first met Charles Babbage at a party in 1833 when she was 17 and he was 41. Lovelace was fascinated with Babbage's Difference Engine. With her analytical mind she could understand how it worked as few people could since she had been extensively tutored in mathematics throughout her childhood. Her mother had decided that a solid grounding in mathematics would ward off the wild and romantic flair that possessed Lovelace's estranged father. After meeting in 1833, Lovelace and Babbage remained a part of the same social circle and wrote to each other frequently [9].

Ada Byron married William King in 1835 and King later became the Earl of Lovelace, making Ada the Countess of Lovelace. She continued her education, even after having three children, in mathematics. She employed Augustus de Morgan, who discovered De Morgan's laws, as her tutor. A member of the English aristocracy and an active member of elite English society, Ada's education in mathematics and social engagements brought her into contact with leading intellectuals of the day including Charles Babbage, Michael Faraday and Charles Dickens [10].

Lovelace saw the potential of Babbage’s envisioned Analytical Engine and was eager to work with Babbage. A friend suggested that she translate Menabrea’s paper from French into English for an English audience [9].

Between 1842 and 1843, Lovelace translated the Menabrea publication, adding an elaborate set of footnotes, called Notes [8]. These notes contain what many consider to be the first computer program—that is, an algorithm designed to be carried out by a machine. These notes are of historical value because of Lovelace’s insight into the capabilities of the engine and decision-making steps in the algorithm. She presented a vision of the capability of computers to go beyond calculating or number-crunching, while Babbage and others focused only on those capabilities [11].

Lovelace died of uterine cancer in 1852, at only 36 years of age before her contributions to the “poetical science” that would become computer science were recognized [9]. See Figure 1 below for an image of Lovelace from 1838, and figure 2 below for a rough timeline of events during Lovelace and Babbage’s lifetimes.

Figure 1., "Ada Lovelace, 1838" by Nefi is licensed under CC BY-NC-SA 2.0

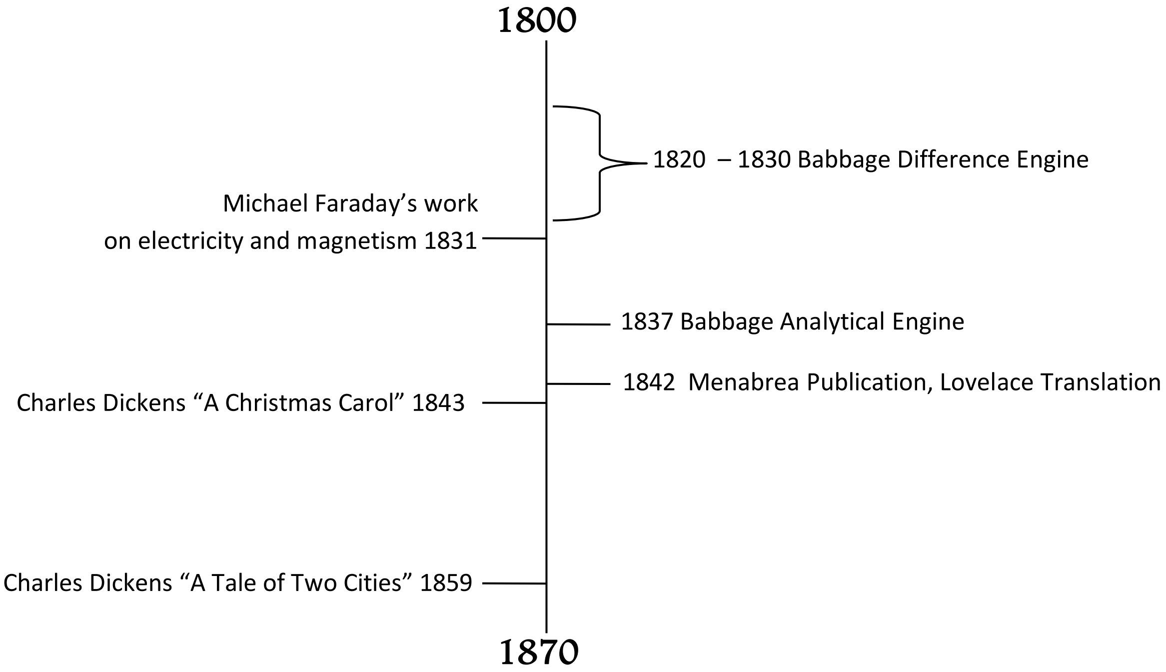

Figure 2. Timeframe of Babbage and Lovelace

What are the Bernoulli Numbers?

The Bernoulli numbers are a curious set of numbers that the engineering or computer science student probably has not encountered in their undergraduate studies. Students of mathematics in graduate school are more likely to be aware of them. This set of rational numbers significantly reduces the computational time of several important functions. In the days prior to high-speed computers, when calculations were essentially done by hand, the Bernoulli numbers reduced the effort and number of mistakes for arduous calculations which could take days. Jacob Bernoulli (1654 – 1705) uncovered these curious numbers a few years before his death [12]. The numbers were subsequently published in 1713 and named the Bernoulli numbers [12]. Interestingly these numbers were also uncovered independently by Japanese mathematician Seki Takakazu, also posthumously published in 1712 [12]. Jacob Bernoulli is not to be confused with Daniel Bernoulli (1700 – 1782) who, as every engineering student knows, is the namesake of the Bernoulli equation used extensively in fluid mechanics. In fact, both Bernoulli's were members of the famous Bernoulli family who excelled in mathematics in the eighteenth century [12]. The Bernoulli family, originally from Antwerp and later Basel, was composed of over twelve scholars through two generations that were gifted in both mathematics and the arts [12]. They truly were academics who, among them, contributed substantially to the development of mathematics and physics during the early modern period.

Their importance lies in the fact that they appear in numerous calculations of mathematical functions and by knowing them the computational effort is considerably reduced.

$$x \to x^p$$

to evaluate the integral of this function the area under the curve must be found. Replace the area under the curve by the rectangular approximations yields the need to compute *$s_p(n)$*

$$s_p(n) = \sum_{k=1}^{n-1} k^p$$

by incorporating

$$B_0(x) = 1$$

$$B_1(x) = x - \frac{1}{2}$$

$$B_2(x) = x^2 - x + \frac{1}{6}$$

$$B_3(x) = x^3 - \frac{3}{2}x^2 + \frac{1}{2}x$$

$$B_4(x) = x^4 - 2x^3 + x^2 - \frac{1}{30}$$

$$B_5(x) = x^5 - \frac{5}{2}x^4 + \frac{5}{3}x^3 - \frac{1}{6}x$$

and

$$s_p(n) = \frac{1}{p+1}\left(B_{p+1}(n) - B_{p+1}\right)$$

Thus, the Bernoulli numbers are found.

B_o	1
B_1	-1/2
B_2	1/6
B_4	-1/30

B_6	1/42
B_8	-1/30
B_{10}	5/66
B_{12}	-691/2730
B_{14}	7/6
	etc.

To illustrate the usefulness of his formula, Bernoulli computed the value of $s_{10}(1000)$ with little effort in less than “half a quarter of an hour” [Smith]. He computed

$$s_{10}(1000) = 91409924241424243424241924242500$$

To achieve this end, he needed to find B_0 up to B_{10}. One can observe that the Bernoulli numbers were at one time of great value because they shortened calculations of several mathematical functions when lengthy computations were done by humans and such lengthy computations were prone to error.

Lovelace’s Program in C Programming Language

First it may be important to discuss the validity of Lovelace’s algorithm, or arguable first computer program in history, for this claim is not without controversy [13]. Babbage was impressed by her mathematical intelligence, and called her “the Enchantress of Numbers” [9]. Although the algorithm is commonly believed to be her work, some suggest that it may be the work of Babbage himself.

A controversy has surfaced for years about the degree of participation of Babbage in the notes of Lovelace. Computer historian Doron Swade, a prominent world expert on the work of Babbage, may have settled the controversy with information presented at a symposium at the University of Oxford [13]. Swade stated, “I confirm that the manuscript evidence clearly shows that Babbage wrote ‘programs’ for his Analytical Engine in 1836-7,” that is, six to seven years before Lovelace’s translation of Menabrea’s notes in 1843 [13]. Swade goes on to say that analysis reveals that are “unarguable” and “do not support, indeed they contradict the claim that Lovelace was the ‘first programmer’” [13]. Swade is specialist and author in the history of computing and is especially known for his work on the computer pioneer Charles Babbage and his Difference Engine [14]. He was a curator at the Science Museum in London, England, and the Computer History Museum in Mountain View, California, United States, where he organized the construction of Charles Babbage's Difference Engine, together with Dr. Allan Bromley. He appeared in the *In Our Time* program on Ada Lovelace, a collaborator with Charles Babbage, broadcast on BBC Radio 4 in 2008 [15]. Swade however emphasized that this in no way undermines Lovelace and the value of her contribution. In fact, he stated, “the obsession with the Bernoulli ‘program’ by those wishing to promote Lovelace may have obscured the much more significant contribution;” that is, in the words of Swade, “an original understanding of where the power and potential of computers lay.” For Swade, Lovelace was ‘remarkably visionary’. According to Betty Alexandra Toole, biographer of Ada Lovelace, “she saw what Babbage did not see: the machine might work on other things beside numbers, e.g. symbols” [16].

The algorithm that appears in Lovelace’s addendum E appears below in Figure 3. It is reproduced here only for illustrative purposes as it is difficult to read in this small format.

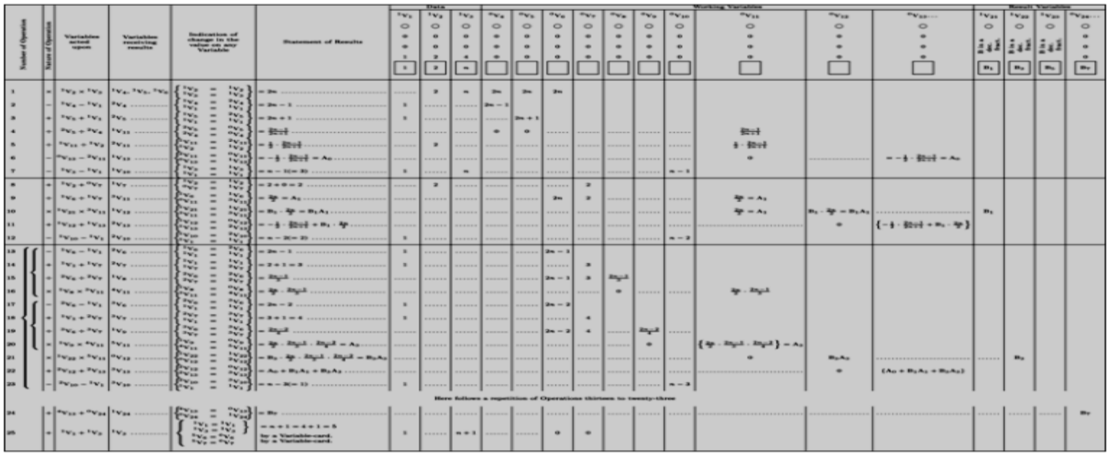

Figure 3. Lovelace’s Algorithm.

There are, of course, numerous possibilities to represent or translate this algorithm to a modern programming language. The original algorithm in Figure 3 was proposed as a workable program using the punched cards that paralleled the Jacquard loom mechanism [Hobsbawm]. One such program was developed by S. Target, who wrote in C that can be easily adapted to C++ or Python with minor changes [17]. Target’s program appears at the end of this paper in Appendix A along with a logic flowchart (Figure 4) which illustrates the conditional looping arrangement embedded in the original algorithm shown in Figure 3. Target commented that Lovelace’s algorithm or program outline successfully computes the first ten Bernoulli numbers with a single error. This is error is most likely a transcription error (Re: the comment in the programming listing, line 39 in Appendix A of this paper). Target’s work or some similar adaptation could be an excellent assignment for students in engineering and/or computer science that have an intermediate level of programming abilities. Indeed, the website Project Lovelace—an open online platform for learning about science and developing computational thinking through programming and problem solving—lists Lovelace's Note G as an intermediate problem to solve using code written in Python, JavaScript, Julia, and C [18]. Portions of the algorithm could also be adapted to students with beginning skills. These exercises would develop programming skills as well as convey the history of one of the world's first computer programs. The exercise of coding Lovelace's algorithm would bring home the degree of Lovelace's achievement in 1843, helping students envision and understand Lovelace as an important founding figure in computer science through a concrete programming exercise.

Conclusion

In this paper a C++ computer program is presented that follows Lovelace's algorithm to calculate Bernoulli numbers, a tool to simplify calculations of useful mathematical functions when these mathematical functions were laboriously accomplished by hand or 'human computers' prone to error and miscalculations. Unfortunately, the mechanical computers conceptualized by Babbage were never built in his lifetime. The Difference Engine was completely constructed in 2002 by a team of historian/engineers, including organizer Doran Swade, and was operational at the Computer History Museum from 2008 until 2016. The mathematical talent of Lovelace was recognized by Babbage who called her "the Enchantress of Numbers." According to Betty Alexandra Toole, biographer of Ada Lovelace, "she saw what Babbage did not see: the machine might work on other things beside numbers, e.g. symbols" [Toole].

This C++ program, presented in the appendix, can be a useful assignment to both beginning and advanced students in engineering and computer science for developing programming skills and also conveying the historical significance of Lovelace's work. This interdisciplinary exercise would promote interdisciplinary competence and knowledge transfer between science and engineering and the humanities, emphasizing the human aspects of the history of science. Lattuca et. al. observed that engineers "need the strong analytical skills fundamental to engineering practice, but also a number of other attributes, such as creativity; skills in communication, management, and leadership; high ethical standards and professionalism; agility, resilience, and flexibility; and an understanding of the complex societal, global, and professional contexts in which engineering is practiced" [4]. ABET's accreditation standards recognize the fundamental importance of these skills, many of which are learning outcomes tied to the humanities.

Finally, this exercise introduces the story of Ada Lovelace as a female role model and founder of computer science. Psychologist Penelope Lockwood's suggests that women benefit from outstanding female role models more so than men, for whom the gender of the role model does not have an impact [19]. We therefore suggest that using Ada Lovelace as an historical role model in engineering curriculum can help engineering and science students revise dominant origin narratives surrounding science and engineering disciplines that are more inclusive and beneficial to recruiting and retaining women in science and engineering. Thus, this program, in a modern programming language, is an educational and academically pleasing assignment.

References

[1] A. Van den Beemt, M. MacLeod, J. Van der Veen, A. Van de Ven, S. van Baalen, R. Klaasen, and M. Boon. “Interdisciplinary engineering education: A review of vision, teaching, and support.” *Journal of Engineering Education*, vol. 109, January 2020.

[2] C. Baillie, E. Ko, W. Newsletter, D. F. Radcliffe. “Advancing Diverse and Inclusive Engineering Education Practices through Interdisciplinary Research and Scholarship,” *Journal of Engineering Education*, vol. 100, no. 1, January 2011.

[3] K. Beddoes and M. Borrego, “Feminist Theory in Three Engineering Education Journals: 1995-2008,” *Journal of Engineering Education*, vol. 100, no. 2, April 2011.

[4] L. R. Lattuca, D. B. Knight, H. Kyoung Ro, B. Novoselich. “Supporting the Development of Engineers’ Interdisciplinary Competence.” *Journal of Engineering Education,* vol. 106, no. 1, January 2017.

[5] M.A. Boden, M. A. *Mind as machine: A history of cognitive science.* Oxford University Press, 2006.

[6] M. Moack, “Difference Engine Leaves Computer History Museum,” *Mountain View Voice*, January 28, 2016

[7] J. J. O’Connor, E. F. Robertson, "Luigi Federico Menabrea," MacTutor History of Mathematics archive, University of St Andrews. 2006.

[8] Menabrea, Luigi, “Sketch of the Analytical Engine Invented by Charles Babbage, Esq. with notes by trans. Ada Lovelace,” in *Scientific Memoirs*, Vol. 3, 1842.

[9] W. Isaacson, *The Innovators,* New York: Simon & Schuster Paperbacks, 2015.

[10] B. A. Toole, "Poetical Science", *The Byron Journal,* vol. 15, 1987.

[11] J. Fuegi, J. Francis, "Lovelace & Babbage and the creation of the 1843 'notes'", *Annals of the History of Computing*, vol. 25, no. 4, October–December 2003.

[12] H. Selin, *Encyclopaedia of the History of Science*. Springer Science & Business Media. p. 1356, March 12, 2008.

[13] J. Yanes, "Ada Lovelace: Original and Visionary, but No Programmer, " *Ventana al Conocimiento*, December 9, 2015. OpenMind,

https://www.bbvaopenmind.com/en/technology/visionaries/ada-lovelace-original-and-visionary-but-no-programmer/, (accessed September 30, 2020).

[14] D. Swade, *Computing History,* UK. http://www.computinghistory.org.uk/det/3382/doron-swade/ (accessed February 8, 2021).

[15] BBC Radio 4, *Ada Lovelace*, In Our Time, March 6, 2008.

[16] B. A. Toole, *Ada, The Enchantress of Numbers; Poetical Science*, Strawberry Press, 1998.

[17] S. Target, Two-Bit History, *Computing through the Ages,* a blog about computer history, https://twobithistory.org/2018/08/18/ada-lovelace-note-g.html, (accessed May, 20, 2020).

[18] Project Lovelace. "Ada Lovelace's Note G." https://projectlovelace.net/problems/ada-lovelaces-note-g/ (accessed February 8. 2021).

[19] P. Lockwood. "'Someone Like Me Can Be Successful': Do College Students Need Same-Gender Role Models?" *Psychology of Women Quarterly*, vol. 30, 2006.

[20] A. Lovelace. "Sketch of the Analytical Engine Invented by Charles Babbage . . . with Notes by the Translator," *Scientific Memoirs, Selected from the Transactions of Foreign Academies of Science and Learned Societies* 3 (1843): 666-731.

[21] D. Swade, Archived 26 January 2009 at the Wayback Machine, Computer History Museum, California, USA.

[22] D. Swade, "Phillip's Economic Computer," 1990s, Science Museum, London, UK.

[23] D. Swade, "Two Cultures: Computer Art and the Science Museum," *White Heat Cold Logic*. Paul Brown, Charlie Gere, Nicholas Lambert, and Catherine Mason, eds., The MIT Press, 2008. pp. 203–218.

Appendix A: Program to Calculate the 8th Bernoulli Number as Proposed by Ada Lovelace

```
---------------begin program-----------------
#include <stdio.h>

/*
 * Calculates what Ada Lovelace labeled "B7", which today we would call the 8th
 * Bernoulli number.
 */
int main(int argc, char* argv[])
{
   // ------------------------------------------------------------------------
   // Data
   // ------------------------------------------------------------------------
   float v1 = 1; // 1
   float v2 = 2; // 2
   float v3 = 4; // n

   // ------------------------------------------------------------------------
   // Working Variables
   // ------------------------------------------------------------------------
   float v4 = 0;
   float v5 = 0;
   float v6 = 0;                    // Factors in the numerator
   float v7 = 0;                    // Factors in the denominator
   float v8 = 0;
   float v10 = 0;                    // Terms remaining count, basically
   float v11 = 0;                    // Accumulates v6 / v7
   float v12 = 0;                    // Stores most recent calculated term
   float v13 = 0;                    // Accumulates the whole result

   // ------------------------------------------------------------------------
   // Result Variables
   // ------------------------------------------------------------------------
   float v21 = 1.0f / 6.0f;           // B1
   float v22 = -1.0f / 30.0f;          // B3
   float v23 = 1.0f / 42.0f;           // B5
   float v24 = 0;                    // B7, not yet calculated
```

```
// ----------------------------------------------------------------------
// Calculation
// ----------------------------------------------------------------------
// ------- A0 -------
/* 01 */ v4 = v5 = v6 = v2 * v3;      // 2n
/* 02 */ v4 = v4 - v1;                // 2n - 1
/* 03 */ v5 = v5 + v1;                // 2n + 1

// In Lovelace's diagram, the below appears as v5 / v4, which is incorrect.
/* 04 */ v11 = v4 / v5;               // (2n - 1) / (2n + 1)

/* 05 */ v11 = v11 / v2;              // (1 / 2) * ((2n - 1) / (2n + 1))
/* 06 */ v13 = v13 - v11;             // -(1 / 2) * ((2n - 1) / (2n + 1))
/* 07 */ v10 = v3 - v1;               // (n - 1), set counter?

// A0 = -(1 / 2) * ((2n - 1) / (2n + 1))

// ------- B1A1 -------
/* 08 */ v7 = v2 + v7;                // 2 + 0, basically a MOV instruction
/* 09 */ v11 = v6 / v7;               // 2n / 2
/* 10 */ v12 = v21 * v11;             // B1 * (2n / 2)

// A1 = (2n / 2)
// B1A1 = B1 * (2n / 2)

// ------- A0 + B1A1 -------
/* 11 */ v13 = v12 + v13;             // A0 + B1A1
/* 12 */ v10 = v10 - v1;              // (n - 2)

// On the first loop this calculates B3A3 and adds it on to v13.
// On the second loop this calculates B5A5 and adds it on.
while (v10 > 0)
{
   // ------- B3A3, B5A5 -------
   while (v6 > 2 * v3 - (2 * (v3 - v10) - 2))
   {                                  // First Loop:
      /* 13 */ v6 = v6 - v1;          // 2n - 1
```

```
        /* 14 */ v7 = v1 + v7;        // 2 + 1
        /* 15 */ v8 = v6 / v7;        // (2n - 1) / 3
        /* 16 */ v11 = v8 * v11;      // (2n / 2) * ((2n - 1) / 3)

                            // Second Loop:
        // 17   v6 = v6 - v1;         2n - 2
        // 18   v7 = v1 + v7;          3 + 1
        // 19   v8 = v6 / v7;         (2n - 2) / 4
        // 20   v11 = v8 * v11;         (2n / 2) * ((2n - 1) / 3) * ((2n - 2) / 4)
    }

    // A better way to do this might be to use an array for all of the
    // "Working Variables" and then index into it based on some calculated
    // index. Lovelace might have intended v14-v20 to be used on the
    // second iteration of this loop.
    //
    // Lovelace's program only has the version of the below line using v22
    // in the multiplication.
    if (v10 == 2)
    {
    /* 21 */ v12 = v22 * v11;         // B3 * A3
    }
    else
    {
    /* 21 */ v12 = v23 * v11;         // B5 * A5
    }

    // B3A3 = B3 * (2n / 2) * ((2n - 1) / 3) * ((2n - 2) / 4)

    // ------- A0 + B1A1 + B3A3, A0 + B1A1 + B3A3 + B5A5 -------
    /* 22 */ v13 = v12 + v13;         // A0 + B1A1 + B3A3 (+ B5A5)
    /* 23 */ v10 = v10 - v1;        // (n - 3), (n - 4)
}

/* 24 */ v24 = v13 + v24; // Store the final result in v24
/* 25 */ v3 = v1 + v3;    // Move on to the next Bernoulli number!

// This outputs a positive number, but really the answer should be
```

```
    // negative. There is some hand-waving in Lovelace's notes about the
    // Analytical Engine sorting out the proper sign.
    printf("A0 + B1A1 + B3A3 + B5A5: %.2f\n", v24);
}
```

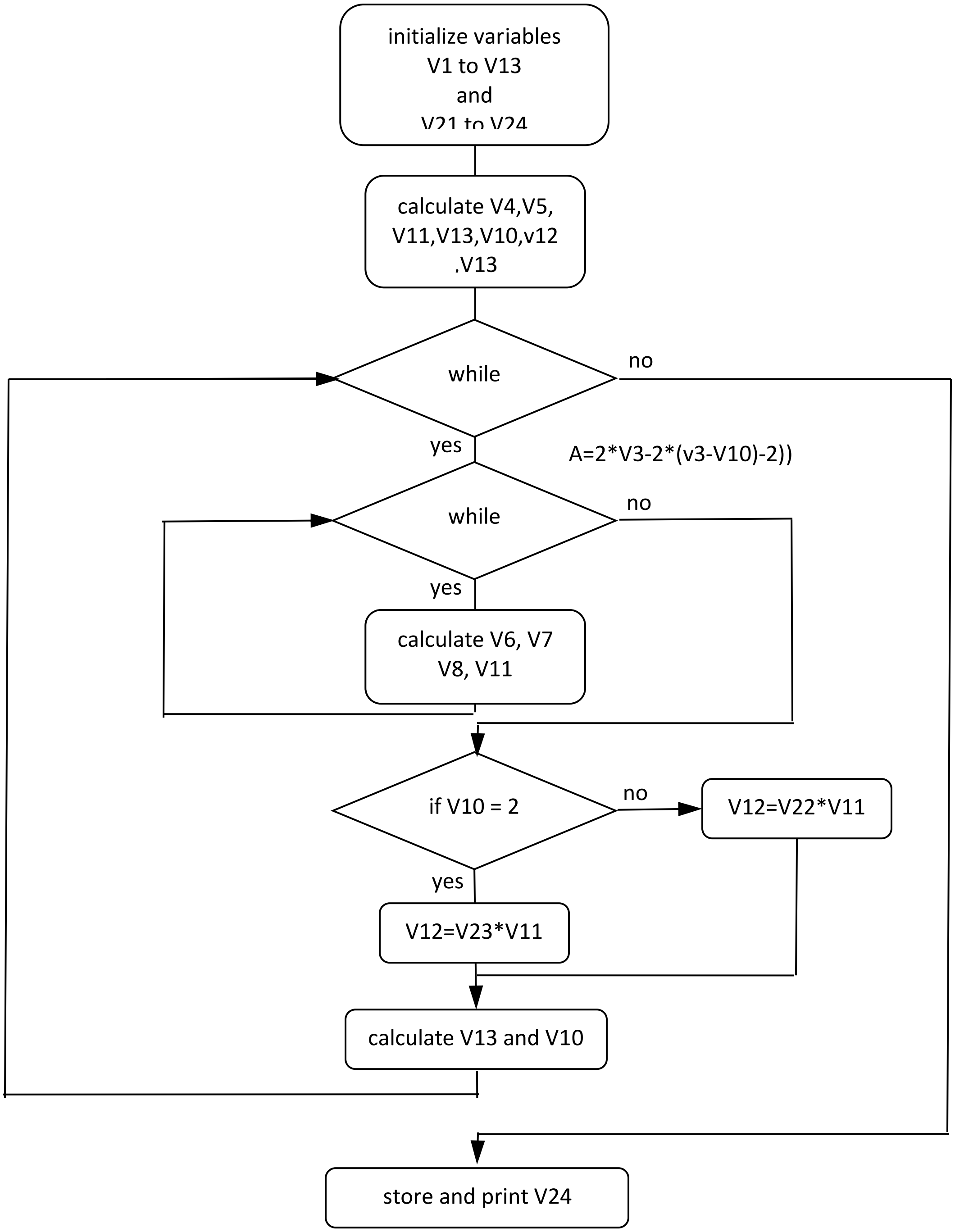

Figure 4. Flowchart of Lovelace's Program