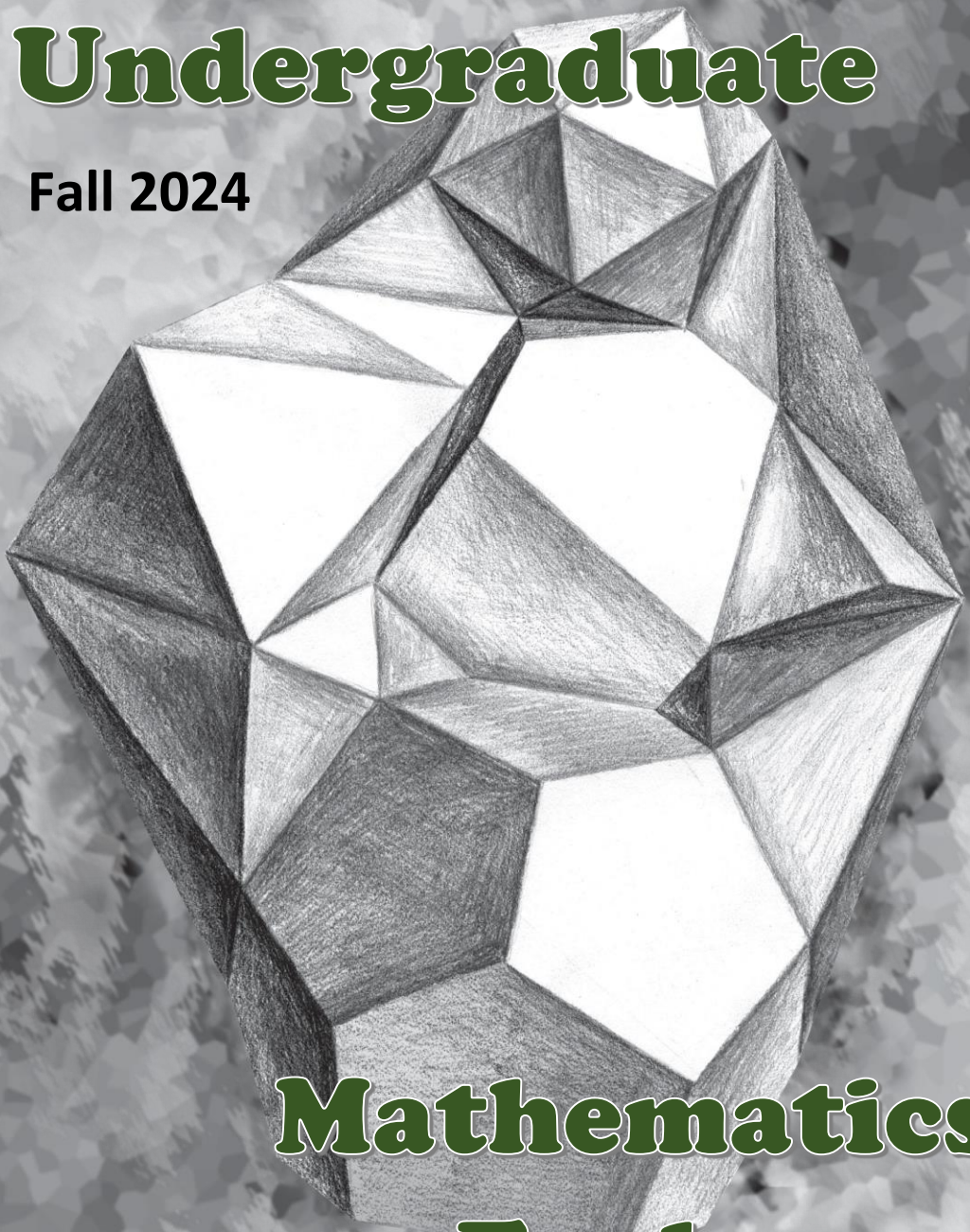
Ball State Undergraduate Mathematics Exchange



Fall 2024

Volume 18, No. 1

Online at https://openjournals.bsu.edu/mathexchange

ISSN 1550-1736

Cover design by Patrick Foley.

A Word from the Editor

The editorial board of the *Mathematics Exchange* is pleased to present this latest issue, featuring six insightful articles that explore a diverse array of mathematical topics tailored to engage and inspire our undergraduate audience. As always, we are inspired by the enthusiasm and dedication of our authors and appreciate their commitment to sharing their discoveries with our readers. We hope that this collection will spark curiosity, provoke thought, and encourage readers to pursue their own explorations in mathematics.

The first article considers solution methods for the nonlinear reactive transport equation used to model protein adsorption, an important process for developing adsorptive chromatography media in biotherapeutics. The authors explore various finite difference schemes and compare implicit and explicit methods, with results indicating optimal convergence and effectiveness in matching experimental data.

The second article addresses the Concept Reinforcement Problem, a graph optimization problem where the goal is to build a graph vertex by vertex with minimal cost. The authors investigate this problem for a range of graph types, including trees, cycles, and grid graphs, providing valuable insights into the cost function behavior across these different structures.

The third article extends two-dimensional rook theory to three and higher dimensions, exploring generalized triangular boards and rook numbers. The paper provides a combinatorial interpretation of generalized Genocchi numbers and proves new triangle generations for these numbers in higher dimensions, further advancing the theory of rook numbers.

In the fourth article, the authors study the variance of the distance to the boundary for planar triangles. Their main result demonstrates that the variance restricted to a line segment joining a vertex to the opposite side is a convex function, providing new insights into geometric properties of triangles.

The fifth article presents a generalization of a continued fraction problem that appeared in *The College Mathematics Journal* and the *PME* Journal. The authors introduce a new continued fraction with a Fibonacci structure and explore how the middle term noise impacts the continued fractions with repeated terms, offering a deeper understanding of this fascinating sequence.

The sixth article explores action graphs, which are closely related to Catalan numbers and their generalizations. The authors examine several sequences, including the (a, b)-Catalan numbers and super Catalan numbers, and present

a conjecture for constructing action graphs for the super Catalan numbers, expanding the theoretical framework of this area of combinatorics.

We are excited to share these contributions and hope they inspire further study and exploration. We trust you will enjoy this issue of *Mathematics Exchange* and look forward to your feedback and suggestions on how we can continue to improve our service to the readership.

Yayuan Xiao

09.25.2024

Call for Papers

We are always soliciting contributions for future issues of this journal. Contributions are accepted from all undergraduate students who have worked on a project beyond the classroom in any mathematical area (e.g., pure, applied, actuarial, and education). Appropriate papers from other departments and other institutions are also welcome. Often the articles are written by undergraduates individually, working in teams, or working with faculty. On occasion we also include articles written solely by faculty or graduate students as long as they are accessible to undergraduates.

To submit an article, please select ONE member from the editorial board, and forward your material in PDF form, usually prepared by LaTeX (preferred) or Microsoft Word, to the editor you selected. We use double anonymized peer review, the identities of both reviewers and authors are concealed from each other throughout the review. To facilitate this, please remove any identifying information, such as authors names or affiliations, from your manuscript before submission. Please ensure that the title page (that include all authors names and affiliations, a complete address of the corresponding author including an email address, acknowledgements, and conflict of interest statement) is present in your submission as a separate file. If authors are undergraduate students, please include your advisors name and contact information in the title page. Review and selection of articles is handled by the editorial committee. Editorial changes of accepted articles are communicated through students advisors, when appropriate.

More information, including links to all previous issues, are available online at `https://openjournals.bsu.edu/mathexchange`.

Contents

A Word from the Editor

Editorial Committee and Call for Papers

Articles

A Comparison of First- and Second Order-in-Time Finite Difference Methods Applied to Nonlinear Reactive Transport
Karen Lawrence, Carson Morris, Kayli Hendricks, Anastasia Wilson 2

Finding the Cheapest Way to Build a Graph
Shu Qian, Samiha Rao 26

A New Triangle Generation of Some Generalized Genocchi Numbers
Feryal Alayont, Stephanie Loewen, Vasily Zadorozhnyy . 51

Triangles and variance of the distance to the boundary
Alastair Fletcher, Katherine Fletcher, Joseph Wasiqi . . 65

Continued Fractions, a-Fibonacci numbers, and the middle b-noise
Aakash Gurung, Cheng-Han Pan 77

Catalan Number Sequences and Generalized Action Graphs
Drew Caldwell, Ali Cochran, Nathan Glisson, Bryce Jennings, Katy McDicken, Luke James Proctor, Sarah Klanderman, Amelia Tebbe 88

Ball State Undergraduate Mathematics Exchange, 18(1), Fall 2024, 2-25
https://openjournals.bsu.edu/mathexchange
DOI: 10.33043/28xy39zc

Pages 2 – 25

A Comparison of First- and Second Order-in-Time Finite Difference Methods Applied to Nonlinear Reactive Transport

Anastasia Wilson , Carson Morris, Kayli Hendricks, Karen Lawrence*

Karen Lawrence graduated from Augusta University in 2022 with a Bachelors Degree of Science in Mathematics, having worked on this paper from her Junior to Senior year. Karen currently works full-time in the legal field and plans to attend law school in 2025.

Carson Morris graduated with a bachelors degree in mathematics at Augusta University in 2022, and he is scheduled to receive his masters degree in mathematics in May of 2024. He has a wide variety of research interests, including but not limited to probability, statistics, numerical analysis, and differential equations.

Kayli Hendricks contributed to this paper as an undergraduate at Augusta University where, in 2023, she graduated with a bachelors degree in mathematics and a Minor in Accounting. Since graduating, she has been promoted to Office Manager at a local tax and accounting firm in Augusta, Georgia.

***Corresponding author**: anawilson@augusta.edu

Anastasia Wilson is an Associate Professor of Mathematics at Augusta University with a Ph.D. in Computational Mathematics from Clemson University and an M.S. in Applied Mathematics from NC State. Formerly at NASA Langley Research Center, her research focuses on mathematical modeling, computational methods, numerical analysis, and optimization, particularly in interdisciplinary.

Abstract

In this paper, we consider solution methods for the nonlinear reactive transport equation used to model the protein adsorption process. Efficient methods for simulating this process are necessary to aid in the development of novel adsorptive chromatography media to ensure high-volume production of purified product for the purposes of biotherapeutics. Using MATLAB®, we compare four finite difference schemes used to solve the nonlinear reactive transport equation, focusing on the differences of efficacy between implicit and explicit methods. As such, two of the methods are semi-implicit and two are explicit with one of each kind using a first-order temporal scheme and one of each using a second-order temporal scheme. The semi-implicit methods evaluate almost all terms implicitly while lagging the nonlinear coefficient function in time to linearize the equations. We include numerical results that indicate optimal convergence of the schemes, and we compare the effectiveness of the schemes in matching experimental data using two different boundary conditions.

1 Introduction

Recent decades have seen a large increase in demand for next-generation biologics, that is, therapies derived from biological sources [1, 2, 3]. In particular, the use of protein therapeutics such as monoclonal antibodies for treatment of COVID-19 and other viruses has led to a need for fast production of such biotherapeutics. Meeting these needs requires the development of novel adsorptive chromatography media to ensure high-volume throughput of purified product.

Resin-bead chromatography, which is used for many bioseparation neads because of its reliability, is relatively slow with low efficiency and, as a result, is considered a “must be addressed" factor to meet the rising demand [4]. Membrane chromatography, which uses a porous, adsorptive membrane instead of small resin beads, is being studied as an alternative to resin bead chromatography (see for example [5, 6, 7, 8, 9, 10, 11] and references therein). Membranes enable higher productivity because they maintain high protein binding capacities at higher flow rates [9]. Efficient methods for simulating the chromatography process using membranes could aid in the development of these media to reduce cost and production time.

Separation processes using adsorptive membranes can be modeled using the

reactive transport equation [12]

$$\omega \frac{\partial C}{\partial t} + \rho_s (1-\omega) \frac{\partial q}{\partial t} + \nabla \cdot (\mathbf{u}C) - \nabla \cdot (\mathbf{D}\nabla C) = f, \ \mathbf{x} \in \Omega, \ t > 0 \tag{1}$$

where ω is the porosity of the media or solid phase, ρ_s is the density of the solid phase, C is the liquid phase concentration, q is the solid phase concentration (or adsorbent), $\mathbf{u}$ is the Darcy velocity, and $\mathbf{D}$ is the hydrodynamic dispersion tensor. The function f represents sources and/or sinks within the domain Ω.

Some novel chromatography membranes being studied use multiple modes of interaction with the product to recover it selectively from impurities in the solution, leading to mathematically complex, nonlinear models to describe the adsorption relationship between C and q [10, 11, 13, 14, 15]. Since the nonlinearity is connected to a time-derivative term, handling this term is non-trivial particularly considering the fact that stability issues commonly arise with explicit numerical solutions of the transport equation.

Due to the nonlinearity arising from the adsorption model, fully explicit methods are easier to implement and are faster to solve in each time iteration. However, fully explicit methods introduce a time step restriction that is not always practical when simulating experimental data. In that case, implicit methods may be faster in the long run since larger time steps are allowed with implicit methods. Still, implicit methods are more complex to implement and care needs to be taken when solving the nonlinearity arising from the adsorption model which is associated with a time-derivative term. While this paper focuses on solution methods with an explicitly-defined adsorption isotherm, having a thorough understanding of methods applied to explicitly-defined adsorption models will provide a basis for future work considering an implicitly-defined isotherm as in [13] which more accurately models multi-modal adsorption.

In general, solution algorithms for these types of transport problems coupled with adsorption models have been well documented. In many cases, finite elements are used to spatially discretize the equation [16, 17, 18, 19, 20, 33, 22, 23, 24, 25]. Because of the potential nonlinearity arising from the adsorption model, the use of finite elements leads to a complex standard form causing higher-order temporal integration methods to be hard to implement. Using finite differences [19, 26, 27] for the spatial discretization greatly simplifies the standard form, and hence higher-order temporal schemes are easier to use with finite differences.

Although finite differences have been used to solve similar transport problems with adsorption, the research team involving the authors has focused on finite element methods in the past [23, 24, 25]. Recent work by our group [25] has compared different higher order temporal discretization schemes using finite elements applied to these types of problems in the case of membrane adsorption and nonlinear adsorption models. Hence the work presented in this paper is meant as a comparison of the same higher order temporal discretization algorithms using finite difference methods applied to the same problem. This paper focuses on using finite differences to solve the one-dimensional reactive transport equation (1) and compares implicit and explicit finite difference schemes using both a first- and second-order temporal integration scheme for

each case of finite difference scheme. MATLAB® is used to implement the finite difference schemes. In this paper, we use an explicit function to describe the relationship between the solid phase concentration q and the liquid phase concentration C, i.e., $q = q(C)$, as is the case for the Langmuir and Freundlich isotherms. These explicit models represent separation processes defined by a single interaction mode, e.g., ion exchange or hydrophobic interaction [28]. We hope to extend the solution methods studied in this paper to account for multiple modes of adsorption by using an implicitly defined isotherm as in [13].

2 Model Equations

In this paper, we restrict our attention to the one-dimensional problem so that the domain Ω is a bounded domain in $\mathbb{R}^1$, ie. $\Omega = [a,b]$ with $|a| < \infty, |b| < \infty$. The inflow boundary of Ω corresponds to $x = a$ and the outflow boundary corresponds to $x = b$.

The transport and adsorption of product at a concentration C in a porous medium can be modeled in one dimension using the reactive transport (advection-diffusion-reaction) equation [12, 33, 26]

$$\omega \frac{\partial C}{\partial t} + \rho_s (1-\omega) \frac{\partial q}{\partial t} + u \frac{\partial C}{\partial x} - D \frac{\partial^2 C}{\partial x^2} = f, \qquad x \in [a,b],\ t \geq 0, \tag{2}$$

$$q = g(q,C), \qquad x \in [a,b], t \geq 0, \tag{3}$$

with the initial and boundary conditions

$$C(x,0) = C_0, \qquad x \in [a,b], \tag{4}$$

$$q(x,0) = q_0, \qquad x \in [a,b], \tag{5}$$

$$C(a,t) = C_{in}, \qquad t \geq 0, \tag{6}$$

$$\left.\frac{\partial C}{\partial x}\right|_{x=b} = 0, \qquad t \geq 0. \tag{7}$$

Here C is the liquid phase concentration, q is the solid phase concentration (or the adsorbed concentration), ω and ρ_s are the porosity and density of the membrane respectively, u is the (Darcy) velocity through the membrane, D is the hydrodynamic dispersion tensor, and $f = f(x,t)$ is a forcing function normally taken to be 0 in the case of protein chromatography. We note that (6) uses a Dirichlet boundary condition on the inflow since the inflow concentration is known, and (7) uses a homogeneous Neumann condition to model continuity of concentration at the outflow. An alternate inflow condition using a Robin-type condition, specifically

$$uC - D \frac{\partial C}{\partial x} = uC_{in}, \qquad x = a, t > 0, \tag{8}$$

can be used to model continuity of flux at the inflow. The two conditions (7) and (8) are the so-called Danckwerts boundary conditions [27, 29]. These are considered in the simulations as a point of comparison.

Equation (2) describes the flow of the carrying fluid through the membrane,

where we assume instantaneous adsorption of the desired product from solution. The system is closed by the isotherm relationship, equation (3), which gives the isotherm relationship which describes the ability of the membrane to bind the protein. This relationship can be explicit linear or nonlinear functions of concentration [12], or it can be defined as a nonlinear, implicitly function of q and C [13, 30].

The solution methods in this paper are applied to the modified system in which the chain rule is used to rewrite the reactive transport equation (2). Rearranging (3) and differentiating in time, we have

$$\frac{\partial q}{\partial t} - \frac{\partial g(q,C)}{\partial t} = 0 \tag{9}$$

$$\Rightarrow \frac{\partial q}{\partial t} - \frac{\partial g}{\partial q}\frac{\partial q}{\partial t} - \frac{\partial g}{\partial C}\frac{\partial C}{\partial t} = 0 \tag{10}$$

$$\Rightarrow \frac{\partial q}{\partial t} = \frac{\partial g/\partial C}{1-\partial g/\partial q}\frac{\partial C}{\partial t} = \tilde{g}(q,C)\frac{\partial C}{\partial t} \tag{11}$$

where

$$\tilde{g}(q,C) = \frac{\partial g/\partial C}{1-\partial g/\partial q}. \tag{12}$$

Substituting (11) in (2) leads to the modified modeling equations:

$$(\omega + (1-\omega)\rho_s \tilde{g}(q,C))\frac{\partial C}{\partial t} + \mathbf{u}\cdot\nabla C - \nabla\cdot(\mathbf{D}\nabla C) = f, \tag{13}$$

$$q = g(q,C). \tag{14}$$

Note that in the case of an explicit isotherm equation, $\tilde{g}$ simplifies to $\tilde{g}(C) = \partial g/\partial C = q'(C)$, and hence the system reduces to one equation with one unknown:

$$(\omega + (1-\omega)\rho_s \tilde{g}(C))\frac{\partial C}{\partial t} + \mathbf{u}\cdot\nabla C - \nabla\cdot(\mathbf{D}\nabla C) = f. \tag{15}$$

3 Numerical Solution Schemes

For the numerical solution schemes, we assume an explicit isotherm model and apply finite difference methods to spatially discretize the one-dimensional reactive transport equation given in (15). We consider both explicit and semi-implicit temporal solution methods.

The semi-implicit methods evaluate almost all terms implicitly while lagging the nonlinear coefficient function $\tilde{g}(C)$ in time to linearize the equations as detailed in Section 3.1.1. For both the semi-implicit case and the explicit case, we apply a first-order and a second-order discretization method. Respectively for semi-implicit methods, we use lagged Backward Euler and lagged Implicit Trapezoid methods; and for explicit methods, we use Forward Euler and Improved Euler (Explicit Trapezoid).

The fully discrete problem partitions the time interval $[t_0, T]$ and spatial domain $[x_0, x_S]$ as

$$0 = t_0 < t_1 < \cdots < t_N = T \quad \text{and} \quad a = x_0 < x_1 < \cdots < x_S = b$$

respectively where $\Delta t = t_{n+1} - t_n$ with $t_n = n\Delta t$ and $\Delta x = x_{i+1} - x_i$ with $x_i = i\Delta x$. Defining $f_i^n := f(t_n, x_i)$ we have $C^n = C(t_n, x)$ and $q^n = q(t_n, x)$ for the continuous solutions and $C_i^n = C(t_n, x_i)$ and $q_i^n = q(t_n, x_i)$ for the discrete solutions. We will denote the discrete solutions in their entirety by $\hat{C}$ and $\hat{q}$.

3.1 Semi-Implicit Methods

In this section, we derive the systems of equations needed to solve the reactive transport equation (15) beginning with two different implicit methods: Backward Euler and Implicit Trapezoid. With both methods, we lag the coefficient function in equation (15) in order to linearize the equation and avoid an otherwise necessary root-finding method, such as Newton's method. This lagging changes our methods to semi-implicit.

3.1.1 First-Order in Time: Backward Euler

From equation (15), we define $g(C) = \omega + (1-\omega)\rho_s q'(C)$ and apply finite difference approximations to $\frac{\partial C}{\partial x}$ and $\frac{\partial^2 C}{\partial x^2}$ and a Backward Euler temporal discretization to $\frac{\partial C}{\partial t}$ to get

$$g(C_i^{n-1})\left(\frac{C_i^n - C_i^{n-1}}{\Delta t}\right) + \mathbf{u}_i^n\left(\frac{C_{i+1}^n - C_i^n}{\Delta x}\right) - \mathbf{D}_i^n\left(\frac{C_{i+1}^n - 2C_i^n + C_{i-1}^n}{(\Delta x)^2}\right) = f_i^n.$$

where $g(C_i^{n-1})$ is lagged in time to linearize the equation. Rearranging we find

$$\left(-\frac{\mathbf{D}_i^n \Delta t}{g(C_i^{n-1})(\Delta x)^2}\right)C_{i-1}^n + \left(1 - \frac{\mathbf{u}_i^n \Delta t}{g(C_i^{n-1})\Delta x} + \frac{2\mathbf{D}_i^n \Delta t}{g(C_i^{n-1})(\Delta x)^2}\right)C_i^n + \left(\frac{\mathbf{u}_i^n \Delta t}{g(C_i^{n-1})\Delta x} - \frac{\mathbf{D}_i^n \Delta t}{g(C_i^{n-1})(\Delta x)^2}\right)C_{i+1}^n = C_i^{n-1} + \left(\frac{\Delta t}{g(C_i^{n-1})}\right)f_i^n. \quad (16)$$

Denoting the coefficients of C_{i-1}^n, C_i^n, and C_{i+1}^n by X_i^n, Y_i^n, and Z_i^n respectively, and renaming the right hand side of (16) as R_i^{n-1}, we have simply

$$X_i^n C_{i-1}^n + Y_i^n C_i^n + Z_i^n C_{i+1}^n = R_i^{n-1}. \quad (17)$$

A system of equations is then formed using (17) and the boundary conditions described in Section 2.

To enforce the nonhomogeneous Dirichlet boundary condition at the inflow given by (6) and the homogeneous Neumann boundary condition at the outflow given by (7), we respectively set

$$C_0^n = C_{in} \quad \text{and} \quad C_S^n = C_{S-1}^n. \quad (18)$$

3.1.2 Second-Order in Time: Implicit Trapezoid

For the Implicit Trapezoid method we isolate $\frac{\partial C}{\partial t}$ to get

$$\frac{\partial C}{\partial t} = \frac{1}{g(C)}\left[-\mathbf{u}\frac{\partial C}{\partial x} + \mathbf{D}\frac{\partial^2 C}{\partial x^2} + f\right] = F(t, C). \quad (19)$$

Applying finite difference methods to $\frac{\partial C}{\partial x}$ and $\frac{\partial^2 C}{\partial x^2}$, we now arrive at

$$F(t,C_i) = \frac{1}{g(C_i)}\left[-\mathbf{u}_i\left(\frac{C_{i+1}-C_i}{\Delta x}\right) + \mathbf{D}_i\left(\frac{C_{i+1}-2C_i+C_{i-1}}{(\Delta x)^2}\right) + f_i\right]. \tag{20}$$

Note that we still use a first order approximation for the first derivative even though Implicit Trapezoid is a second-order method in time. Second order, centered-difference approximations for the first derivative term are known to be less accurate near a concentration front caused by an inflow flux as we are considering in this case. In the case of protein chromatography, accuracy near the concentration front is of vital importance. Forward or backward difference approximations are more accurate in this case. However, second-order, one-sided approximations cannot be used near the boundaries. Initial experimentation using a different approximation (either a centered-difference or first-order) near the boundary showed a reduction in the accuracy of the approximation so that using a second-order, one-sided approximation did not pay off. Hence we chose to use the simpler first-order approximation.

Evaluating at the nth time step gives

$$F(t^n,C_i^n) = \frac{1}{g(C_i^{n-1})}\left[-\mathbf{u}_i^n\left(\frac{C_{i+1}^n-C_i^n}{\Delta x}\right) + \mathbf{D}_i^n\left(\frac{C_{i+1}^n-2C_i^n+C_{i-1}^n}{(\Delta x)^2}\right) + f_i^n\right],$$

where we lag $g(C)$ in order to avoid root-finding methods. $F(C_i^{n-1},t^{n-1})$ is obtained similarly by evaluating (20) at the $n-1$st time step. Then applying an Implicit Trapezoid approximation, we obtain

$$\begin{aligned}\frac{C_i^n-C_i^{n-1}}{\Delta t} = \frac{1}{2g(C_i^{n-1})}\Bigg[&-\mathbf{u}_i^n\left(\frac{C_{i+1}^n-C_i^n}{\Delta x}+\frac{C_{i+1}^{n-1}-C_i^{n-1}}{\Delta x}\right)+\\ &\mathbf{D}_i^n\left(\frac{C_{i+1}^n-2C_i^n+C_{i-1}^n}{(\Delta x)^2}+\frac{C_{i+1}^{n-1}-2C_i^{n-1}+C_{i-1}^{n-1}}{(\Delta x)^2}\right)+(f_i^n+f_i^{n-1})\Bigg].\end{aligned}$$

Multiplying by $(\Delta x)^2\Delta t$ on both sides, expanding, and gathering like terms, we obtain

$$\begin{aligned}&\left(\frac{\mathbf{D}_i^n\Delta t}{2g(C_i^{n-1})}\right)C_{i-1}^n+\left(\frac{\mathbf{u}_i^n\Delta t\Delta x}{2g(C_i^{n-1})}-\frac{\mathbf{D}_i^n\Delta t}{g(C_i^{n-1})}-(\Delta x)^2\right)C_i^n\\ &\qquad+\left(-\frac{\mathbf{u}_i^n\Delta t\Delta x}{2g(C_i^{n-1})}+\frac{\mathbf{D}_i^n\Delta t}{2g(C_i^{n-1})}\right)C_{i+1}^n\\ =&\left(-\frac{\mathbf{D}_i^n\Delta t}{2g(C_i^{n-1})}\right)C_{i-1}^{n-1}+\left(-\frac{\mathbf{u}_i^n\Delta t\Delta x}{2g(C_i^{n-1})}+\frac{\mathbf{D}_i^n\Delta t}{g(C_i^{n-1})}-(\Delta x)^2\right)C_i^{n-1}\\ &+\left(\frac{\mathbf{u}_i^n\Delta t\Delta x}{2g(C_i^{n-1})}-\frac{\mathbf{D}_i^n\Delta t}{2g(C_i^{n-1})}\right)C_{i+1}^{n-1}+\left(-\frac{(\Delta x)^2\Delta t}{2g(C_i^{n-1})}\right)(f_i^n+f_i^{n-1}).\end{aligned} \tag{21}$$

Denoting the coefficients of C_{i-1}^n, C_i^n, and C_{i+1}^n for the Implicit Trapezoid approximation by $\hat{X}_i^n$, $\hat{Y}_i^n$, and $\hat{Z}_i^n$ respectively, and renaming the right hand side

of (21) as $\hat{R}_i^{n-1}$, we get

$$\hat{X}_i^n C_{i-1}^n + \hat{Y}_i^n C_i^n + \hat{Z}_i^n C_{i+1}^n = \hat{R}_i^{n-1}.$$

We note that although the system simplifies to a tridiagonal matrix as with Backward Euler, $\hat{X}_i^n$, $\hat{Y}_i^n$, $\hat{Z}_i^n$, and $\hat{R}_i^{n-1}$ are much more complicated so that more care must be taken when programming the equations. The process for calculating the concentration at any given time is now identical to that of Backward Euler.

3.2 Fully Explicit Methods

In this section, we derive the systems of equations needed to solve the reactive transport equation (15) using two different fully explicit methods: Forward Euler and Improved Euler (Explicit Trapezoid). Since explicit methods have known stability issues, a right-sided approximation is used to approximate $\frac{\partial C}{\partial x}$ to increase the stability of the explicit method solutions.

3.2.1 First-Order in Time: Forward Euler

As with the implicit methods, we use finite difference approximations for the spatial derivatives in (15), but now apply a Forward Euler approximation for $\frac{\partial C}{\partial t}$ to get

$$g(C_i^{n-1})\left(\frac{C_i^n - C_i^{n-1}}{\Delta t}\right) + \mathbf{u}_i^{n-1}\left(\frac{C_{i+1}^{n-1} - C_i^{n-1}}{\Delta x}\right)$$

$$-\mathbf{D}_i^{n-1}\left(\frac{C_{i+1}^{n-1} - 2C_i^{n-1} + C_{i-1}^{n-1}}{(\Delta x)^2}\right) = f_i^{n-1}$$

where again $g(C) = \omega + (1-\omega)\rho_s q'(C)$. We then solve for C_i^n getting

$$C_i^n = \left(1 + \frac{\mathbf{u}_i^{n-1}\Delta t}{g(C_i^{n-1})\Delta x} - \frac{2\mathbf{D}_i^{n-1}\Delta t}{g(C_i^{n-1})(\Delta x)^2}\right)C_i^{n-1}$$
$$+\left(\frac{-\mathbf{u}_i^{n-1}\Delta t}{g(C_i^{n-1})\Delta x} + \frac{\mathbf{D}_i^{n-1}\Delta t}{g(C_i^{n-1})(\Delta x)^2}\right)C_{i+1}^{n-1} + \left(\frac{\mathbf{D}_i^{n-1}\Delta t}{g(C_i^{n-1})(\Delta x)^2}\right)C_{i-1}^{n-1} + \frac{f_i^{n-1}\Delta t}{g(C_i^{n-1})}. \quad (22)$$

For simplicity, we let X_i^{n-1}, Y_i^{n-1}, Z_i^{n-1} denote the coefficient functions of C_{i-1}^{n-1}, C_i^{n-1}, C_{i+1}^{n-1} respectively and F_i^{n-1} denote the last term in (22). Then we have simply

$$C_i^n = X_i^{n-1}C_{i-1}^{n-1} + Y_i^{n-1}C_i^{n-1} + Z_i^{n-1}C_{i+1}^{n-1} + F_i^{n-1}$$

for all interior nodes ($i = 1, ..., S-1$). We enforce the boundary conditions on the exterior nodes as described in Section 3.1.1.

3.2.2 Second-Order in Time: Improved Euler (Explicit Trapezoid)

To obtain the Improved Euler formulation, we use an explicit approximation of the Implicit Trapezoid rule:

$$\frac{C_i^n - C_i^{n-1}}{\Delta t} = \frac{1}{2}\left[F(t^n, \bar{C}_i^n) + F(t^{n-1}, C_i^{n-1})\right], \quad (23)$$

where $\bar{C}_i^n = C_i^{n-1} + \Delta t F(t^{n-1}, C_i^{n-1})$ is the Forward Euler approximation of C and $F(t, C_i)$ is given by (20). Our derivation for $\bar{C}$ is the same as our Forward Euler derivation above, but we store it with different variables; namely, we label our coefficients as FX_i^{n-1}, FY_i^{n-1}, and FZ_i^{n-1} and the final term as FF_i^{n-1}:

$$\bar{C}_i^n = FX_i^{n-1} C_{i-1}^{n-1} + FY_i^{n-1} C_i^{n-1} + FZ_i^{n-1} C_{i+1}^{n-1} + FF_i^{n-1}.$$

The F in each term stands for *Forward* as these are the coefficients for the Forward Euler solution.

Next we use $\bar{C}$ to calculate the first term on the right side of (23)

$$F(t^n, \bar{C}_i^n) = \frac{1}{g(\bar{C}_i^n)} \left[f_i^n - \mathbf{u}_i^n \left(\frac{\bar{C}_{i+1}^n - \bar{C}_i^n}{\Delta x} \right) + \mathbf{D}_i^n \left(\frac{\bar{C}_{i+1}^n - 2\bar{C}_i^n + \bar{C}_{i-1}^n}{(\Delta x)^2} \right) \right],$$

which we will denote by PF, with P standing for *Primary*. Rearranging, we get

$$\begin{aligned} PF(t^n, \bar{C}_i^n) = &\left(\frac{\mathbf{D}_i}{g(\bar{C}_i^n)(\Delta x)^2} - \frac{\mathbf{u}_i^n}{g(\bar{C}_i^n)\Delta x} \right) \bar{C}_{i+1}^n \\ &+ \left(\frac{\mathbf{u}_i^n}{g(\bar{C}_i^n)\Delta x} - \frac{2\mathbf{D}_i^n}{g(\bar{C}_i^n)(\Delta x)^2} \right) \bar{C}_i^n \\ &+ \left(\frac{\mathbf{D}_i^n}{g(\bar{C}_i^n)(\Delta x)^2} \right) \bar{C}_{i-1}^n + \frac{f_i^n}{g(\bar{C}_i^n)}, \end{aligned} \quad (24)$$

and for simplicity we rewrite this as

$$PF(t^n, \bar{C}_i^n) = PX_i^{n-1} \bar{C}_{i-1}^n + PY_i^{n-1} \bar{C}_i^n + PZ_i^{n-1} \bar{C}_{i+1}^n + PpF_i^{n-1}.$$

by renaming the coefficients and last term of (24) as PX_i^{n-1}, PY_i^{n-1}, PZ_i^{n-1}, and PpF_i^{n-1} respectively.

The calculations for $F(t^{n-1}, C_i^{n-1})$, the second F in our Improved Euler formula, are similar to the *Primary* calculations above with the only difference being the shift to the $n-1$ time step. We use the label SF, standing for *Secondary F* to distinguish this term. The Improved Euler Formula is then given by

$$C_i^n = C_i^{n-1} + \frac{\Delta t}{2} \left[PF(t^n, \bar{C}_i^n) + SF(t^{n-1}, C_i^{n-1}) \right].$$

We use this equation for all interior nodes ($i = 1, ..., S-1$). Again, we enforce the boundary conditions on the exterior nodes as described in Section 3.1.1.

4 Numerical Results

In this section, we present results from implementations of the finite difference schemes described in this paper using MATLAB®. First, we give numerical convergence rates for the numerical approximation to the 1D nonlinear reactive transport equation using the four finite difference methods described in Section 3. Second, we compare numerical simulation data to experimental data in an effort to determine better ways to accurately predict breakthrough. We conclude this section with a comparison of the methods.

4.1 Computation of Convergence Rates

To illustrate the solution methods were performing correctly, we computed the numerical error convergence rates for each of the four methods described above. The error was calculated by computing the discrete $L^2(L^2)$ and $L^\infty(L^2)$ time-space norms of the difference between the theoretical and numerical solutions for each method [31].

For the computation of convergence rates, we take the domain to be $[t_0,T]\times[x_0,x_S]=[0,1]\times[0,1]$. The right hand side function f, the boundary conditions, and the initial condition were determined by the chosen true solution

$$C(t,x)=t\sin(\pi x).$$

Note that this function has homogeneous Dirichlet boundary conditions and an initial function that is zero everywhere. Additionally for simplicity, parameters are chosen to be the following

$$\mathbf{D}=1,\mathbf{u}=1,\rho_s=1,K_{eq}=1,\text{ and }q_{max}=1,\omega=.5$$

where K_{eq} and q_{max} are the parameters in the Langmuir adsorption model

$$q=\frac{K_{eq}q_{max}C}{1+K_{eq}C}. \tag{25}$$

As mentioned above, we make use of the discrete $L^2(L^2)$ and $L^\infty(L^2)$ time-space norms [31] which we denote for simplicity as $\|\cdot\|_{L^2(0,T;L^2(\Omega))}$ and $\|\cdot\|_{L^\infty(0,T;L^2(\Omega))}$ respectively. Specifically, for the exact solution $C(t,x)$ and the approximate solution $\hat{C}$,

$$\|C-\hat{C}\|_{L^2(L^2)}=\left(\sum_{n=0}^{N}\|C(t^n,\cdot)-\hat{C}^n\|^2_{L^2(\Omega)}dt\right)^{1/2}$$

and

$$\|C-\hat{C}\|_{L^\infty(L^2)}=\max_{0\le n\le N}\|C(t^n,\cdot)-\hat{C}^n\|_{L^2(\Omega)}$$

where

$$\|C(t^n,\cdot)-\hat{C}^n\|_{L^2(\Omega)}=\left(\sum_{i=0}^{S}\left(C(t^n,x_i)-\hat{C}_i^n\right)^2\right)^{1/2}.$$

To obtain the order of convergence for the error for both the Forward Euler and Backward Euler solutions, Δx and Δt are each cut in half as the errors are calculated for each pair of discretization values $((\Delta t)_j,(\Delta x)_j)$. The order of convergence is then calculated by

$$\text{Order}_j=\frac{\ln\left|\frac{\text{Err}_j}{\text{Err}_{j-1}}\right|}{\ln\left|\frac{\Delta t_j}{\Delta t_{j-1}}\right|}. \tag{26}$$

where Err_j indicates either the L^2 time-space norm $\|\cdot\|_{L^2(0,T;L^2(\Omega))}$ or the L^∞ time-space norm $\|\cdot\|_{L^\infty(0,T;L^2(\Omega))}$. Since Forward Euler and Backward Euler are both first-order time discretization methods, we expect the order to approach 1 as Δt and Δx are decreased.

For the second order methods (Implicit Trapezoid and Improved Euler), the discretization parameters were treated separately. Specifically, Δt was again cut in half each time while Δx was cut by a fourth; this was to account for the fact that the finite difference approximation used for the spatial derivative $\frac{\partial C}{\partial x}$ is only first order. The order was calculated the same as in (26). We then expect the order to approach 2 as Δt and Δx are decreased indicating that the methods are second order in time.

4.1.1 Fully Implicit Methods

First-Order in Time: Backward Euler Backward Euler, being a first order method, is $\mathscr{O}(\Delta t)$, so in combination with the finite difference formulas this leads to an overall error that is linear in both space and time as in

$$||C-\hat{C}||_{L^2(L^2)} \sim \mathscr{O}(\Delta t+\Delta x)$$

and

$$\|C-\hat{C}\|_{L^\infty(L^2)} \sim \mathscr{O}(\Delta t+\Delta x)$$

where C is the exact solution and $\hat{C}$ is the approximate solution. Therefore, we expect the rate of convergence to converge to 1 as Δt and Δx decrease which is shown by the results in Table 1.

$(\Delta t, \Delta x) \rightarrow$	$\left(\frac{1}{16}, \frac{1}{8}\right)$	$\left(\frac{1}{32}, \frac{1}{16}\right)$	$\left(\frac{1}{64}, \frac{1}{32}\right)$	$\left(\frac{1}{128}, \frac{1}{64}\right)$	$\left(\frac{1}{256}, \frac{1}{128}\right)$
$\|C-\hat{C}\|_{L^\infty(L^2)}$	5.221E-02	2.308E-02	1.082E-02	5.237E-03	2.575E-03
Rate	–	1.1776	1.0926	1.0474	1.0240
$\|C-\hat{C}\|_{L^2(L^2)}$	2.973E-02	1.282E-02	5.936E-03	2.853E-03	1.399-03
Rate	–	1.2131	1.1110	1.0568	1.0288

Table 1: Approximation errors and experimental convergence rates for the Backward Euler approximation. As Δt and Δx are cut in half, the L^2 error is reduced the same amount which is consistent with the theoretical convergence rates.

Second-Order in Time: Implicit Trapezoid Implicit Trapezoid is a second order method in time, $\mathscr{O}(\Delta t^2)$. In combination with the finite difference formulas this leads to an overall error that is linear in space yet quadratic in time as in

$$||C-\hat{C}||_{L^2(L^2)} \sim \mathscr{O}((\Delta t)^2+\Delta x)$$

and

$$\|C-\hat{C}\|_{L^\infty(L^2)} \sim \mathscr{O}((\Delta t)^2+\Delta x)$$

where C is the exact solution and $\hat{C}$ is the approximate solution. As expected, we see the convergence rate approaching 2 with both the L^∞ and L^2 errors as shown in Table 2.

$(\Delta t, \Delta x) \rightarrow$	$\left(\frac{1}{16}, \frac{1}{4}\right)$	$\left(\frac{1}{32}, \frac{1}{16}\right)$	$\left(\frac{1}{64}, \frac{1}{64}\right)$	$\left(\frac{1}{128}, \frac{1}{256}\right)$
$\|C-\hat{C}\|_{L^\infty(L^2)}$	1.334E-01	2.327E-02	5.237E-03	1.256E-03
Rate	–	2.5193	2.1515	2.0596
$\|C-\hat{C}\|_{L^2(L^2)}$	7.687E-02	1.308E-02	2.873E-03	6.688E-04
Rate	–	2.5553	2.1867	2.1029

Table 2: Approximation errors and experimental convergence rates for the Implicit Trapezoid approximation. As Δt is cut in half and Δx is cut into fourths, the L^2 error is reduced by a fourth which indicates quadratic convergence in time. Note that only four columns of data are given because of the small size of Δt and Δx.

4.1.2 Fully Explicit Methods

First-Order in Time: Forward Euler As with Backward Euler, Forwad Euler is a first-order time integration method. Therefore, we expect the rate of convergence to approach to 1 as Δt and Δx decrease which is shown by the results in Table 3. As Forward Euler is a fully explicit method, the stability of the solution depends on Δt being small enough with respect to Δx. Although a full theoretical stability analysis is ongoing, we expect from initial numerical investigations to have a second order stability requirement, i.e. $\Delta t \sim \mathscr{O}((\Delta x)^2)$. Furthermore, this requirement is consistent with what is expected for numerical stability in non-reactive diffusion problems [32]. Therefore to ensure stability of the error results shown in Table 3, we set $\Delta t = \frac{(\Delta x)^2}{50}$ and reduce Δx by half each time.

Δt	$\Delta x \rightarrow$	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{8}$	$\frac{1}{16}$	$\frac{1}{32}$
$4*\frac{(\Delta x)^2}{50}$	$\|C-\hat{C}\|_{L^\infty(L^2)}$	4.18E-01	1.34E-01	5.32E-02	2.35E-02	1.10E-02
	Rate	–	1.64	1.33	1.17	1.09
	$\|C-\hat{C}\|_{L^2(L^2)}$	2.32E-01	7.53E-02	2.98E-02	1.32E-02	6.20E-03
	Rate	–	1.62	1.33	1.17	1.09

Table 3: Approximation errors and experimental convergence rates for the Forward Euler approximation. As Δx and Δt decrease, the convergence rate approaches 1 indicating a linear rate.

Second-Order in Time: Improved Euler (Explicit Trapezoid) As with the Implicit Trapezoid method, Improved Euler is a second-order time integration method resulting in an overall error that is linear in space yet quadratic in time. To account for the slower spatial convergence, we modify the error computations similarly to what was done with Implicit Trapezoid: Δx is reduced by a fourth while Δt is reduced by only a half. We then expect the convergence rate to approach 2 as is shown in Table 4 with both the L^∞ and L^2 errors for Improved Euler.

As with Forward Euler, Improved Euler is a fully explicit method therefore

requiring a stability condition on Δt. Again from initial numerical investigations, we expect $\Delta t \sim \mathscr{O}(\Delta x)^2$. To maintain stability of the solution for the error computations with Δt and Δx being reduced by half independently in each iteration, we set Δt very small to start, specifically $\Delta t = \frac{1}{3000}(\Delta x_0)^2$ where $\Delta x_0 = 0.5$ is the initial Δx value.

$(\Delta t, \Delta x) \rightarrow$	$\left(\frac{1}{3000}\left(\frac{1}{2}\right)^2, \frac{1}{2}\right)$	$\left(\frac{1}{6000}\left(\frac{1}{2}\right)^2, \frac{1}{8}\right)$	$\left(\frac{1}{12000}\left(\frac{1}{2}\right)^2, \frac{1}{32}\right)$	$\left(\frac{1}{24000}\left(\frac{1}{2}\right)^2, \frac{1}{128}\right)$
$\|C-\hat{C}\|_{L^\infty(L^2)}$	4.18E-01	5.32E-02	1.10E-02	2.62E-03
Rate	–	2.974	2.270	2.072
$\|C-\hat{C}\|_{L^2(L^2)}$	2.31E-01	2.99E-02	6.20E-03	1.47E-03
Rate	–	2.952	2.268	2.072

Table 4: Approximation errors and experimental convergence rates for the Improved Euler approximation. As Δx and Δt decrease, the convergence rate approaches 2 indicating a quadratic rate. Note that, as with the Implicit Trapezoid results, only four columns of data are given because of the small size of Δt and Δx.

4.2 Experimental Comparison

To determine the efficacy of our simulation methods, we compare the simulated solution to experimental data. For the comparison with experimental data, we choose parameter values to mimic the associated laboratory conditions. The experimental data was obtained by Juan Wang in the Department of Chemical and Biomolecular Engineering at Clemson University as part of Scott Husson's Bioseparations and Advanced Separation Materials research group.

The multimodal membrane (MMM) was prepared by modification of a commercial regenerated cellulose membrane from Whatman, Inc., which has an average effective pore size of 1 μm and a thickness of 0.7 mm. Details of membrane synthesis were described previously [10]. A stack of 10 MMMs was placed in a Mustang Coin® module (Pall Corporation, Port Washington, NY) with one piece of 25 μm nominal pore diameter filter paper (Whatman 5) placed on each side of the stack. The resulting bed height of the module is approximately 0.7 cm. We orient our one-dimensional model in the vertical direction; hence, the interval for x is $[a,b] = [0,0.7]$. The effective membrane diameter within the module is 1.6 cm, but this does not affect the one-dimensional model. The module was installed in an AKTA purifier. The membrane bed porosity was measured to be $\omega = 0.84$. The interval for t was chosen to simulate the entire adsorption portion of the chromatography process which was found to be $[0,T] = [0,80]$ for the laboratory experiment. An immunoglobulin G (IgG) protein feed solution was pumped vertically through the membrane module using a constant pressure differential providing a constant flow rate of 0.1 mL/min.

For the dispersion, we chose a model correlated with low Peclet numbers [12] to account for the spreading effect due to both diffusion and convective flow. In the case of one-dimension, we consider only the longitudinal dispersivity D_L which is modeled in a low Peclét flow regime as

$$D = D_L = \omega d_0 + \alpha_L |\mathbf{u}|,$$

where ω is the porosity, d_0 is the diffusion coefficient, $|\mathbf{u}|$ is the mean microscopic velocity, and α_L and α_T are the longitudinal and transverse dispersivities, respectively. For an understanding of dispersion models in higher dimensions, one can see for example [12]. As stated above, the flow is purely vertical for the physical experiment being considered so we orient the longitudinal direction of the dispersion in the vertical direction.

Values for the diffusion coefficient of protein molecules in porous membranes range from 6×10^{-4} cm^2/min (for small proteins such as glycine) to 2.28×10^{-5} cm^2/min (for large proteins such as immunoglobulin G) [33]. We chose a diffusion coefficient value of 2.28×10^{-5} cm^2/min as IgG was used in the experiment.

The value for the mean microscopic velocity $|\mathbf{u}|$ was approximated from the constant flow rate. As the experimental flow rate is 0.1 mL/min:

$$|\mathbf{u}| \approx \frac{\text{flowrate}}{\pi r^2} = 0.0497 \text{ cm/min}. \tag{27}$$

The value given in (27) was also used for the coefficient velocity function for the advection term in the reactive transport equation (15) when using a Dirichlet inflow boundary condition. When considering a Danckwerts inflow boundary condition, $\mathbf{u}$ in both the reactive transport equation (15) and the inflow boundary condition (8) was taken to be as large as the flow rate (0.1) in an attempt to increase the accuracy of the simulations.

Values for the longitudinal dispersivity, α_L, are known to vary depending on the physical properties of the membrane. Experimental values for α_L on a small scale range from around a few millimeters to nearly a hundred centimeters depending on the porosity [12, 34], and α_L decreases as the porosity increases [35]. As our porosity is closer to 1, α_L should be on the lower end of this range, i.e., on the order of a few millimeters. The specific value of α_L was chosen to obtain a better match between the experimental and simulated data while still staying in the acceptable range. The specific values used are given with the results below.

For the numerical simulation results shown below, we use Langmuir's adsorption model (equation (25)) and obtained q_{max} and K_{eq} by fitting the model to measured adsorption isotherm data:

$$q_{max} = 150 \text{ mg/mL} \quad \text{and} \quad K_{eq} = 2.06 \text{ mL/mg}.$$

We set Δx and Δt to be small enough for each method to ensure the error in the numerical methods did not affect the results significantly. More specifics are given with the results below.

Productivity is an important metric in evaluating bind-and-elute chromatography processes. Previously, Husson and coworkers [9] defined productivity as the mass of protein that can bind per volume of membrane per time, and provided a convenient expression to calculate it:

$$\text{Productivity} = \frac{\text{B}_{\text{dynamic}}}{\text{t}_{\text{break}}}, \tag{28}$$

where B_{dynamic} is the dynamic binding capacity and t_{break} is the time it takes to reach the point of 10% breakthrough, i.e., the time at which the column effluent concentration reaches 10% of the feed concentration. Focusing on a 10% breakthrough is commonly used in practice as the time at which the protein column would be moved to the next stage in a continuous chromatography process [36, 37]. From a productivity standpoint, differences in dynamic binding capacity do not impact productivity nearly as significantly as differences in residence times [9]. Therefore, high productivities can be achieved when the time to breakthrough is short; that is, the membrane quickly reaches its capacity to bind proteins. Hence accurately predicting the time to 10% breakthrough is of great importance when simulating the chromatography process.

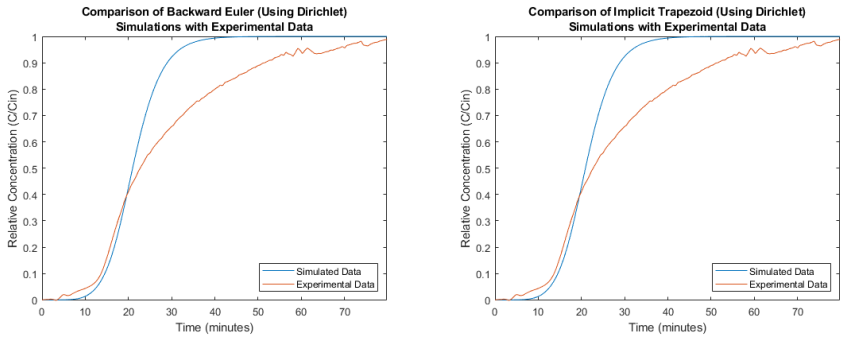
The information on protein breakthrough is visualized using a breakthrough curve, a plot of the relative concentration (C/C_{in}) of the effluent (outflow) as a function of time. In other words, Figures 1 and 2 show $C/C_{in} = C(b)/C_{in}$ as a function of time. We assessed the validity of our approach by comparing our simulated breakthrough curves with the experimental curves generated from laboratory data.

4.2.1 Fully Implicit Methods

In this section, we compare the numerical solutions obtained from Backward Euler and Implicit Trapezoid with the experimental data described above. Figure 1 shows comparisons of the numerical solutions to the experimental data. The discretization parameters were set to $(\Delta t, \Delta x) = (0.1, 0.1)$ to generate the numerical solutions. Both Backward Euler and Implicit Trapezoid using a Dirichlet inflow condition produced results that matched the experimental data very well up through approximately 40% breakthrough (where the curves intersect).

Although the accuracy of the simulations decreases after 40% breakthrough, these methods still provide a high amount of accuracy since predicting the experimental results close to column saturation is known to be difficult due to complex end behavior in the adsorption phase of the chromotographic process. The reasons for this complexity are still unclear but are thought to be cause by protein-protein interactions, non-specific binding, and multilayer adsorption effects [36]. Dimartino, Boi, and Sarti even point out that using the entire range of the concentration during the adsorption process would in fact make the model less accurate initially where it is more important to have higher accuracy [36]. Additionally, the adsorption mechanisms in effect for the experimental data under consideration in this paper are due to multiple modes of adsorption thus making the data more complex and likely more difficult to predict than data obtained with one mode of adsorption (e.g. affinity adsorption in [36]). Consequently, the fact that both solution methods provide accurate results not only for t_{break}, the time to 10% breakthrough, but also up through 40% breakthrough is strong support for the effectiveness of these methods as simulation tools.

Danckwerts Boundary Conditions The Danckwerts boundary conditions, specifically a Robin type boundary condition on the inflow and a Neumann boundary condition on the outflow, were also considered since these boundary conditions are theoretically more realistic. However, for our chosen model and solution methods, the results were much less accurate than those produced by the Dirichlet inflow condition, and hence we chose not to provide a reference figure. Using $(\Delta t, \Delta x) = (0.1, 0.1)$ as with the Dirichlet inflow condition, the data generated by Backward Euler and Implicit Trapezoid using Danckwerts the accuracy of the data was lower for earlier time although the shape of the numerical curve more closely resembles the shape of the experimental data for later t-values. Considering the metrics applied above to the results with Dirichlet inflow condition, accuracy earlier on is more important, and hence this inflow condition does not provide an effective simulation tool for the situation under consideration.



(a) Backward Euler (Dirichlet) (b) Implicit Trapezoid (Dirichlet)

Figure 1: Comparison of Backward Euler and Implicit Trapezoid using Dirichlet inflow boundary condition. Both methods produce accurate results up through 40% breakthrough. Accurately predicting the end behavior is a known issue with simulating adsorption due to complex end behavior of the adsorption phase.

The cause of the inaccuracy with the Danckwerts conditions is yet to be determined. Initial investigations using a higher-order approximation for the derivative term in the inflow condition showed no improvement. The decreased accuracy may be due to the simplifying assumption of a constant velocity affects since the velocity is taken into account in the inflow Danckwerts boundary condition. Further research is necessary to understand why these alternate boundary conditions resulted in less accurate results.

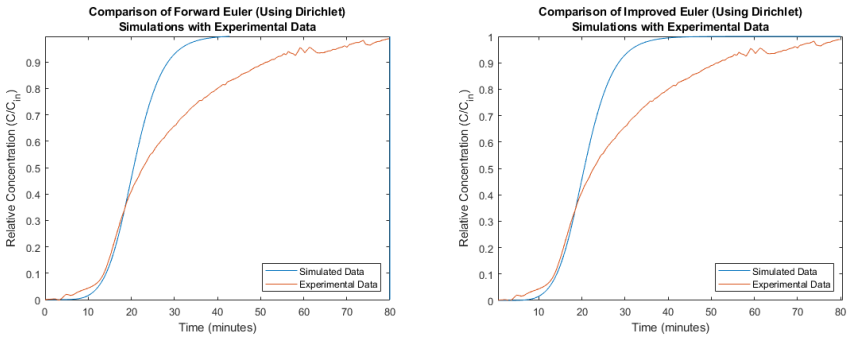
4.2.2 Fully Explicit Methods

In this section, we compare the numerical solutions obtained from Forward Euler and Improved Euler with the experimental data described above. As we can see from Figure 2, both the Forward Euler and the Improved Euler solutions accurately predict the experimental data up through 40% breakthrough similar to the semi-implicit methods, and therefore they both provide effective

simulation tools.

For both simulation results shown in Figure 2, Δx is set to 0.1 and Δt is set to $\frac{1}{50}(\Delta x)^2$. We found that making Δx any bigger (0.125) or smaller (0.0625) resulted in a lower level of accuracy. Changes in the coefficient of our Δt value from 1 to 1/200 seemed to have no noticeable effect; values outside of this range were not investigated. It is not surprising that changes in the Forward Euler code affecting accuracy had the same effect in the Improved Euler code as the Forward Euler method is used to calculate the Improved Euler results. We used this relationship to inform what discretization values were considered for producing the simulation results with Improved Euler.

Danckwerts Boundary Conditions Implementing the Danckwerts inflow boundary condition for Forward Euler and Improved Euler exacerbated the instability inherent in these explicit methods. Through numerical investigations, we found that the Danckwerts inflow condition could be used only if the magnitude of the derivative $\frac{\partial C}{\partial x}$ at the particular time step was less than 17 for Forward Euler and 2.1 for Improved Euler. Consequently, the Danckwerts inflow condition was implemented on a conditional basis: Danckwerts was used as the inflow condition for a particular time step if the magnitude of the derivative at that timestep was less than the given number (17 for Forward Euler, 2.1 for Improved Euler) and the Dirichlet inflow condition was used otherwise. With this implementation, the Forward Euler solution produced simulated results that were visually indistinguishable from the results using a Dirichlet inflow condition as shown in Figure 2a.



(a) Forward Euler (Dirichlet) (b) Improved Euler (Dirichlet)

Figure 2: Comparison of Forward Euler and Improved Euler. With $\Delta x = 0.1$ and $\Delta t = \frac{1}{50}(\Delta x)^2$ both graphs produce similar and accurate results particularly up through 40% breakthrough.

The results from Improved Euler with the Danckwerts inflow boundary conditions contain a notable difference from the graph for Improved Euler with the Dirichlet inflow boundary condition. Using the conditional implementation of the Danckwerts inflow condition described in the previous paragraph, the Improved Euler solution simulated losing some of the mass of the protein

effluent so that in the end the simulated outflow never reached the magnitude of the inflow concentration. In this way, the simulation is not physically accurate, so Improved Euler with a Danckwerts inflow condition cannot be used to predict experimental results. As with the semi-implicit methods, using a higher-order approximation for the derivative term in the inflow condition showed no improvement for either fully explicit method.

4.3 Comparison of Numerical Schemes

4.3.1 Semi-Implicit Methods: Backward Euler vs. Implicit Trapezoid

Although different size reductions for $\Delta t, \Delta x$ in the error analysis for Backward Euler and Implicit Trapezoid make it difficult to judge their effectiveness, we can make use of the common $(\Delta t, \Delta x)$ between the two methods, namely $(\frac{1}{32}, \frac{1}{16})$. The errors shown for both methods in Tables 1 and 2 are on the same order (10^{-2}) although the Backward Euler error is marginally smaller; however, reducing Δt while holding Δx constant would result in a larger reduction in the error with Implicit Trapezoid since it is of higher order.

For both Backward Euler and Implicit Trapezoid, simulations were done using a Neumann outflow while changing the inflow boundary condition from Dirichlet to Danckwerts. Using the Dirichlet boundary condition, both graphs shown in Figure 1 appear to be nearly identical. However, it is possible that with higher dimensions, the difference between the two results would be more noticeable. In the case of higher dimensions, the higher-order Implicit Trapezoid method may prove more useful in allowing larger step sizes while preserving accuracy, but further research is necessary to determine if this is the case.

We also compared computation time in regards to the simulations. The computation times were calculated with MATLAB's `tic` and `toc` commands. With $\Delta t = \Delta x = 0.1$ using the Dirichlet inflow boundary condition, Backward Euler has a computation time in the range of 0.083 to 0.097 seconds, while Implicit Trapezoid has a computation time in the range of 0.091 to 0.120 seconds. Considering the Danckwerts inflow boundary condition, Backward Euler has a computation time in the range of 0.078 to 0.096 seconds and Implicit Trapezoid has a computation time in the range of 0.109 to 0.131 seconds. Considering the similarity in the simulated results for the chosen discretization parameters, Backward Euler was more efficient than Implicit Trapezoid in producing the comparative data shown in this paper.

4.3.2 Explicit Methods: Forward Euler vs. Improved Euler

In order to compare the error results for Improved Euler with the error results for Forward Euler, we computed the errors associated with the Forward Euler solution for the same Δt and Δx values used for the Improved Euler error results given in Table 4; these new error calculations are given in Table 5. The L^∞ and L^2 errors for both methods are very similar. The Forward Euler error is occasionally slightly smaller than the Improved Euler error although the largest

difference is on the order of 10^{-3} and this difference shrinks as Δt and Δx shrink. Additionally, we would expect reducing Δt while holding Δx constant to result in a larger reduction in the error with Improved Euler since it is of higher order.

$(\Delta t, \Delta x) \rightarrow$	$\left(\frac{1}{3000}\left(\frac{1}{2}\right)^2, \frac{1}{2}\right)$	$\left(\frac{1}{6000}\left(\frac{1}{2}\right)^2, \frac{1}{8}\right)$	$\left(\frac{1}{12000}\left(\frac{1}{2}\right)^2, \frac{1}{32}\right)$	$\left(\frac{1}{24000}\left(\frac{1}{2}\right)^2, \frac{1}{128}\right)$
$\|C-\hat{C}\|_{L^\infty(L^2)}$	4.18E-01	5.30E-02	1.10E-02	2.62E-03
$\|C-\hat{C}\|_{L^2(L^2)}$	2.27E-01	2.94E-02	6.10E-03	1.45E-03

Table 5: Approximation errors and convergence rates for Forward Euler to compare to Table 5.

With a Dirichlet inflow boundary condition both codes produced accurate simulation results as shown in Figure 2. However, a Danckwerts inflow boundary condition could only be implemented conditionally depending on the magnitude of the first derivative which produced results that were unrealistic with Improved Euler and indistinguishable from the Dirichlet inflow condition with Forward Euler. Consequently, neither of the explicit methods were capable of fully utilizing a Danckwerts inflow boundary condition for simulations.

For our time analysis, we ran both codes with $\Delta x = \Delta t = 0.1$ and used our Dirichlet and Homogeneous Neumann boundary conditions; although this choice of Δt does not follow the stability requirements discussed previously, we found that the time step restriction was not necessary in practice with the chosen parameter values. We did not do a time analysis with Danckwerts due to the unrealistic output discussed in Section 4.2.2. The Forward Euler run time ranged from 0.008 to 0.032 seconds, and the Improved Euler run time ranged from 0.017 to 0.053 seconds. Hence, in general Forward Euler produced results faster than Improved Euler. However as with Implicit Trapezoid, the higher-order Improved Euler method may prove more useful in allowing larger step sizes in the case of higher dimension. It remains to be seen whether the time step restriction will negate the potential for larger time steps with Improved Euler in higher dimensions. Thus for one dimension, Forward Euler was the more efficient explicit simulation method.

4.3.3 Implicit vs. Explicit Methods

In comparing the implicit and explicit methods, the differences between errors were negligible so we will focus on two topics: efficacy with different boundary conditions and computation time.

With the implicit methods, both types of inflow boundary conditions that were considered (Dirichlet and Danckwerts) were easy to implement and produced reasonably accurate results in comparison to the experimental data. In comparison of the simulation results with the two boundary conditions, the Dirichlet condition was better for one dimensional simulations, but the Danckwerts condition may provide more accurate results in higher dimensions. For the explicit methods, the Danckwerts inflow condition exacerbated the instability, so a conditional implementation was necessary. Consequently, it did not noticeably improve the Forward Euler results, and it made the Improved Euler results less

accurate (in fact, unrealistic). Overall, the implicit methods were much better when considering more complex inflow boundary conditions.

For this comparison, we use the computation times described in Sections 4.3.1 and 4.3.2. Between the two implicit methods, Backward Euler had a slightly shorter computation time as expected since there are more function evaluations for Implicit Trapezoid. The explicit methods were much faster (as much as an order of magnitude faster) when considering the same discretization parameters. Between the two explicit methods, Forward Euler had faster computation times as expected since it has fewer function evaluations. However, the faster computation time can only be achieved if stability does not require the timestep to be extremely small. Therefore, the explicit methods are conditionally better in terms of computation time.

5 Conclusion

In this paper, we developed and compared four finite difference schemes, two semi-implicit and two explicit, for solving the nonlinear reactive transport equation. A summary of some of our findings is provided in Table 6. No strategy was a clear winner over the others, but the implicit methods held a slight advantage because they had no time step restriction. Between the two implicit methods, the Backward Euler solution scheme was moderately better in one-dimension because of the shorter computation times and simpler derivation and computation. However, the explicit methods had extremely short computation times when a larger timestep could be taken as was the case with the simulation comparison to experimental data. Between the two explicit methods, Forward Euler was better in one dimension as it ran more quickly and was more versatile than Improved Euler. Thus, while Backward Euler is best overall for the purpose in this paper, Forward Euler is the better of the two Explicit Methods. In general, which strategy is best for a given situation depends on the needs of the end user as is illustrated in Table 6.

If the user is looking for...	**Then use...**
Less complex to implement	Forward Euler
Less complex to calculate converging error	Backward Euler
Fastest computation time	Forward Euler* or Backward Euler
Increased accuracy with faster computation time	Improved Euler* or Implicit Trapezoid
Increased accuracy with larger time steps	Implicit Trapezoid
More robust with boundary conditions	Backward Euler or Implicit Trapezoid

Table 6: A comparison of the benefits of each method. The asterisk (*) indicates the benefit holds only if stability does not require a very small timestep.

Our next steps include investigating more complex kinematics, variable velocity, higher dimensions, adaptive time-stepping schemes and other linearization techniques in an effort to produce more accurate and yet efficient simulation methods. Incorporating more complex kinematic equations such as non-instantaneous adsorption and an isotherm that incorporates multiple modes of adsorption may

result in more accurate predictions of the end behavior of the experimental data. Additionally, a variable velocity may increase the accuracy of the simulation predictions, particularly with a Danckwerts inflow condition.

With higher dimensions, the higher-order schemes may prove more useful in allowing larger time steps while preserving accuracy. Incorporating an adaptive time-stepping scheme with Improved Euler could also help the simulations achieve larger time steps while still maintaining accuracy. Given our use of semi-implicit methods in this paper, we intend to explore fully implicit methods in the future considering both built-in MATLAB® nonlinear solvers and custom programmed linearization techniques such as Newton iteration or a fixed point with Anderson Acceleration. It remains to be seen whether the increased computational cost of the more complex solution schemes will cancel out the benefit of the increased accuracy. As Dimartino, Boi, and Sarti [36] point out, an effective simulation tool for chromatography must have a model that is accurate while being as simple as possible.

Initial numerical investigations indicate that the explicit methods are stable only for small enough Δt whereas the semi-implicit methods have no requirements on Δt for stability. A theoretical stability analysis is ongoing and will be included in a subsequent paper.

6 Acknowledgements

This research was supported by the National Science Foundation under Grant No. NSF-DMS-2011911. The authors are grateful to Professor Scott Husson and Dr. Juan Wang for their work in producing the experimental data.

Bibliography

[1] Davies, N. The future of biologics. *Thepharmaletter*. (2017)

[2] Research, T. Global Biological Drugs Market to be Worth US 287,139.7 Million by 2020. (http://globenewswire.com/news-release/2014/10/20/674317/10103285/en/Global-Biological-Drugs-Market-to-be-Worth-US-287-139-7-Million-by-2020-Transparency -Market-Research.html,2014)

[3] Hiller, A. Fast Growth Foreseen for Protein Therapeutics. *Genet. Eng. Biotechn.*. **29** pp. 153-155 (2009), http://www.genengnews.com/gen-articles/fast-growth-foreseen-for-protein-therapeutics/2722/

[4] Langer, E. Focus on Efficiency: Single-use, analytical methods and downstream processing at the forefront. *Pharm. Manuf.*. **March** pp. 3-11 (2013)

[5] Bhut, B., Wickramasinghe, S. & Husson, S. Preparation of high-capacity, weak anion-exchange membranes for protein separations using surface-initiated atom transfer radical polymerization. *J. Membr. Sci.*. **325** pp. 176-183 (2008)

[6] Bhut, B. & Husson, S. Dramatic performance improvement of weak anion-exchange membranes for chromatographic bioseparations. *J. Membr. Sci.*. **337** pp. 215-233 (2009)

[7] Bhut, B., Christensen, K. & Husson, S. Membrane chromatography: Protein purification from E.colilysate using newly designed and commercial anion-exchange stationary phases. *J. Chromatogr. A*. **1217**, 4946-4957 (2010)

[8] Bhut, B., Christensen, K. & Husson, S. Membrane chromatography: Protein purification from E. Colilysate using newly designed and commercial anion-exchange stationary phases. *J. Chromatogr. A*. **1217**, 4946-3957 (2010)

[9] Chenette, H., Robinson, J., Hobley, E. & Husson, S. Development of high-productivity, strong cation-exchange adsorbers for protein capture by graft polymerization from membranes with different pore sizes. *J. Membrane Sci.*. **423424** pp. 43-52 (2012)

[10] Wang, J., Sproul, R., Anderson, L. & Husson, S. Development of multimodal membrane adsorbers for antibody purification using atom transfer radical polymerization. *Polymer*. **55**, 1404-1411 (2014)

[11] Wang, J., Wilson, A., Robinson, J., Jenkins, E. & Husson, S. A new multimodal membrane adsorber for monoclonal antibody purifications. *J. Membr. Sci.*. **492** pp. 137-146 (2015)

[12] Marsily, G. Quantitative Hydrogeology: Groundwater Hydrology for Engineers. (Academic Press,1986)

[13] Nfor, B., Noverraz, M., Chilamkurthi, S., Verhaert, P., Wielen, L. & Ottens, M. High-throughput Isotherm Determination and Thermodynamic Modeling of Protein Adsorption on Mixed Mode Adsorbents. *J. Chromatogr. A*. **1217** pp. 6829-6850 (2010)

[14] Singh, N., Husson, S., Zdyrko, B. & Luzinov, I. Surface modification of microporous PVDF membranes by ATRP. *J. Membrane Sci.*. **262** pp. 81-90 (2005)

[15] Singh, N., Wang, J., Ulbrict, M., Wickramasinghe, S. & Husson, S. Surface-initiated atom transfer radical polymerization: A new method for the preparation of polymeric membrane adsorbers. *J. Membrane Sci.*. **309** pp. 64-72 (2008)

[16] Boyer, T., Miller, C. & Singer, P. Modeling the Removal of Dissolved Organic Carbon by Ion Exchange in a Completely Mixed Flow Reactor. *Water Res.*. **42** pp. 1897-1906 (2008)

[17] Farthing, M., Kees, C., Russell, T. & Miller, C. An ELLAM Approximation for Advective-Dispersive Transport with Nonlinear Sorption. *Adv. Water Resour.*. **29** pp. 657-675 (2006)

[18] Kaur, J., Malengier, B. & Remesíková, M. Convergence of an operator splitting method on a bounded domain for a convection-diffusion-reaction system. *J. Math. Anal. Appl.*. **348** pp. 894-914 (2008)

[19] Poulain, C. & Finlayson, B. A Comparison of Numerical Methods Applied to Non-linear Adsorption Columns. *Int. J. Numer. Meth. Fl.*. **17** pp. 839-859 (1993)

[20] Remesíková, M. Solution of Convection-Diffusion Problems with Nonequilibrium Adsorption. *J. Comput. Appl. Math.*. **169** pp. 101-116 (2004)

[21] Suen, S. & Etzel, M. A mathematical analysis of affinity membrane bioseparations. *Chem. Eng. Sci.*. **47**, 1355-1364 (1992)

[22] Arbogast, T., Wheeler, M. & Zhang, N. A nonlinear mixed finite element method for a degenerate parabolic equation arising in flow in porous media. *SIAM J. Numer. Anal.*. **33** pp. 1669-1687 (1996)

[23] Wilson, A. & Jenkins, E. Numerical Simulation of Solid Phase Adsorption Models Using Time-Integrated, Up-winded Finite Element Strategies. *Comput. Sci. Eng.*. (2019)

[24] Wilson, A. & Jenkins, E. Analysis of a fully implicit SUPG scheme for a filtration and separation model. *Comp. Appl. Math.*. (2020)

[25] Wilson, A. & Jenkins, E. Towards Higher Order Methods for Nonlinear Adsorption Problems.

[26] Tarafder, A. Modeling and Multi-Objective Optimization of a Chromatographic System. *Multi-Objective Optimization In Chemical Engineering: Developments And Applications*. (2013)

[27] Yang, H., Bitzer, M. & Etzel, M. Analysis of Protein Purification Using Ion-Exchange Membranes. *Ind. Eng. Chem. Res.*. **38** pp. 4044-4050 (1999)

[28] Scopes, R. Protein Purification: Principles and Practice. (Springer-Verlag,1994)

[29] Mott, H. & Green, Z. On Danckwerts' Boundary Conditions for the Plug-Flow with Disperson/Reaction Model. *Chem. Eng. Comm.*. **202**, 739-745 (2015)

[30] Mollerup, J. A review of the thermodynamics of protein association to ligands, protein adsorption and adsorption isotherm. *Chem. Eng. Technol.*. **31**, 864-874 (2008)

[31] Strikwerda, J. Finite Difference Schemes and Partial Differential Equations. (SIAM,2004)

[32] Cheney, E. & Kincaid, D. Numerical Mathematics and Computing. (Brooks Cole,2007)

[33] Suen, S. & Etzel, M. A mathematical analysis of affinity membrane bioseparations. *Chem. Eng. Sci.*. **47**, 1355-1364 (1992)

[34] Agarwal, N., Semmens, M., Novak, P. & Hozalski, R. Zone of influence of a gas permeable membrane system for delivery of gases to groundwater. *Water Resour. Res.*. **41** (2005)

[35] Xu, M. & Eckstein, Y. Statistical analysis of the relationships between dispersivity and other physical properties of porous media. *Hydrogeol. J.*. **5**, 4-20 (1997)

[36] Dimartino, S., Boi, C. & Sarti, G. A validated model for the simulation of protein purification through affinity membrane chromatography. *J Chrom A.* **1218** pp. 1677-1690 (2022)

[37] Riske, F. & Ransohoff, T. Development of Continuous Capture Steps in Bioprocess Applications. *Preparative Chromatography For Separation Of Proteins.* (2017)

Ball State Undergraduate Mathematics Exchange, 18(1), Fall 2024, 26-50
https://openjournals.bsu.edu/mathexchange
DOI: 10.33043/92ra679d

Pages 26 – 50

Finding the Cheapest Way to Build a Graph

*Shu Qian, Samiha Rao**

Shu Qian was a senior at George Washington University majoring in Pure Mathematics when working on this paper. She is currently a master student in the Mathematics program at New York University. She is interested in Number Theory and Combinatorics but is still exploring other interesting fields.

Samiha Rao worked on this paper while a senior at George Washington University where she pursued a double major in Applied Mathematics and International Affairs. She is currently working at Bristol Myers Squibb in Information Technology, specializing in analytics. She is passionate about working in spaces where math and analytics can help bring social change.

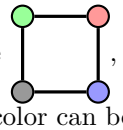
Abstract

The Concept Reinforcement Problem is a graph optimization problem introduced by Novikoff. One seeks to build a graph G vertex by vertex in the cheapest way possible for that graph. The cost function for each vertex is a positive, decreasing, convex function where the input is determined by the number of neighbors already built. We solve this problem for a variety of different graphs such as simple connected small-sized graphs, trees, cycles, wheels, grid graphs, ladder graphs, and complete bipartite graphs.

1 Introduction

1.1 Background Consider building a network G vertex by vertex. The cost of building this graph is the sum of the costs of vertices, and the cost of building each vertex v is determined by the number of neighbors of v that have already been built. For an arbitrary graph $G = (V, E)$, there are different

***Corresponding author**:samiharao1002@gmail.com

ways to build it. For example, for the four-vertex cycle  , if the green vertex is installed first, any other vertex in a different color can be built next. Installing a vertex v_i costs $f(k)$ where k is the number of vertices built before v_i and adjacent to v_i. So in the case of the four-vertex square, if we build the vertices with permutation $\{\text{green}, \text{red}, \text{blue}, \text{black}\}$, the total cost of this graph is $f(0)+f(1)+f(1)+f(2)$; but if the permutation is $\{\text{green}, \text{blue}, \text{red}, \text{black}\}$, the cost is changed to $f(0)+f(0)+f(2)+f(2)$. Which permutation gives a cheaper cost? How about other graphs? Is there a patterned way to build an arbitrary graph to obtain the cheapest installing cost?

In this paper, we consider a graph-theoretic optimization problem introduced by Novikoff in his study of the Concept Reinforcement Problem. We will use standard terminology and notations from graph theory as found in Bonin's *Introduction to Graph Theory* [2].

Novikoff's thesis gives an introduction to our problem discussed in this paper. He showed that for arbitrary cost functions, the problem is computationally intractable in general [1, Section 5.3.1]. However, for a linear cost function $f(k) = ak + b$, the problem is simple: as described in [1, Section 5.3.2], the optimal cost is $am + bn$ where m is the number of edges and n is the number of vertices in G, and it is achieved for any building order. This led Novikoff to mention the Concept Reinforcement Problem, where he discusses what happens when f is convex and decreasing, which is the case we consider in this paper. One consequence of convexity that will be of great use to us is the following formula: if $b \leq a-2$ then

$$f(a)+f(b) > f(a-1)+f(b+1). \tag{1. A}$$

Novikoff applies this formula in the case of two adjacent vertices that are built consecutively, where the first vertex gives a cost of $f(a)$ and the second vertex gives a cost of $f(b)$ such that $b \leq a-2$; in this case, it will be cheaper to switch the order of these vertices in this permutation. Once these vertices are switched, Novikoff remarks that the costs become $f(a-1)$ and $f(b+1)$, since the earlier vertex would have lost an already built neighbor. Then, the cost of the two vertices combined would be cheaper than before due to convexity, while the cost of all other vertices would be unchanged.

Our problem is to find the cheapest permutation of a graph, using inequality 1. A to analyze different families of graphs. Some families of graphs we will explore include graphs on less than five vertices, trees, cycles, wheels, and some bipartite graphs.

1.1 Our Results

Since there are not many simple connected graphs with one, two, three, or four vertices, we list all graphs and all possible cost expressions for each graph in Section 2, and we compare their costs to find the optimal cost. In this paper, we also check and prove the optimal cost for paths, stars (which are two kinds of trees), general trees, cycles, wheels, and some bipartite graphs including ladder graphs, grid graphs, and complete bipartite graphs.

In Section 3, we consider trees with n vertices, including paths (Section 3.1), stars Section 3.2, and general trees Section 3.3. We show that the cheapest cost for a tree is $f(0)+(n-1)f(1)$, which we get when we first build an arbitrary vertex and then keep building vertices that have a neighbor already built.

For cycles with n vertices, we build an arbitrary vertex first and then keep building vertices that have a neighbor already built. Then we achieve the cheapest cost $f(0)+(n-2)f(1)+f(2)$, as shown in Section 4.

For wheels with n vertices, we show in Section 5 that the cheapest cost is $f(0)+f(1)+(n-3)f(2)+f(3)$. We can get this cost when we start by building two vertices, where one of them must be the central vertex and the other is arbitrary. Then we keep building the other vertices such that each of them have at least two pre-existing neighbors.

In Section 6, we look at bipartite graphs, focusing on ladder graphs, grid graphs, and complete bipartite graphs. The ladder graph L_n is formed by taking two paths of n vertices and connecting corresponding vertices by an edge, as illustrated in Figure 1. In Section 6.1, we prove that the optimal cost for L_n is $f(0)+nf(1)+(n-1)f(2)$, which can be achieved when we first build the vertices of one path in order, then build the vertices of the other path in order. Then in Section 6.2, we generalize this for the cheapest cost of a grid graph $G_{m\times n}$. Similar to the construction of the ladder graph, we conclude that the cheapest cost for the graph $G_{m\times n}$ with mn vertices is $f(0)+(m+n-2)f(1)+(mn-m-n+1)f(2)$. In Section 6.3, we show that for a complete bipartite graph $K_{m,n}$ where $n\leq m$, the optimal cost is $f(0)+2[f(1)+f(2)+\cdots+f(n-1)]+(m-n+1)f(n)$. To accomplish this, we alternate between the two vertex sets until all vertices in one set are built, and then build any remaining vertices.

We mainly use the inequality 1. A to prove these results. In some proofs, we also use the fact that the sum of all inputs of the cost function for a graph is equal to the number of edges of the whole graph. Take the four-vertex cycle above as an example. Its two possible cost expressions are $f(0)+f(1)+f(1)+f(2)$ and $f(0)+f(0)+f(2)+f(2)$. The sum of the inputs $0+1+1+2=0+0+2+2=4$ is the number of edges in this four-vertex cycle.

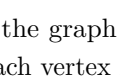
Lemma 0.1. *For any graph G, for any order on the vertices, the sum of the inputs to the cost function when building G is the number of edges of G.*

Proof. Each input of the cost function of a vertex v is the number of its neighbors built previously, and each of these neighbors is connected by exactly one edge to the vertex v. Thus, when we sum the inputs, each edge is counted exactly

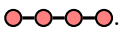
once, for the second of its endpoints that we build. Therefore, the sum of all inputs is the total number of edges. □

2 Simple connected graphs with fewer than five vertices

Obviously, there is only one possible cost for $n = 1$ and $n = 2$. When $n = 1$, the cost is $f(0)$, and when $n = 2$, the cost is $f(0) + f(1)$.

When $n = 3$, the graph can be the path  or the cycle  . For the cycle, since each vertex is adjacent to all of other vertices, the cost can only be $f(0) + f(1) + f(2)$. For the path, when we pick one of these three vertices as our first vertex, it is either (1) one of the leaves, or (2) the central vertex, and the cost of this vertex is $f(0)$ because no vertex has been built before. If the first vertex we build is one of the leaves (case (1)), the second vertex can either be (a) the central vertex or (b) the other leaf. If the central vertex is the second vertex, its cost is $f(1)$, and the cost for the path is $f(0) + f(1) + f(1)$ for case (1a). If the second vertex is another leaf, its cost is $f(0)$ because it is not adjacent to the first built leaf, then the cost of the central vertex is $f(2)$. So the cost in case (1b) is $f(0) + f(0) + f(2)$. In the case (2), when the central vertex is our first vertex, the cost is $f(0) + f(1) + f(1)$ because it is the same no matter which leaf is our second vertex. Therefore, there are two possible costs for the path: $\text{Cost}_1 = f(0) + f(1) + f(1)$ and $\text{Cost}_2 = f(0) + f(0) + f(2)$. By the convexity of f, since we have $0 \leq 2 - 2$, we get the inequality $f(1) + f(1) < f(0) + f(2)$, so $\text{cost}_1 < \text{cost}_2$.

When $n = 4$, there are six different simple connected graphs [3]. We will discuss them separately below. Since we can pick any of four vertices to be the first vertex, any of the remaining three to be the second, and either of the remaining two to be the third one, for each graph with four vertices, there are $4! = 24$ orders to build it. However, depending on the different properties of the graph, such as its symmetries or its greatest degree, each graph does not give 24 distinct cases.

The first graph discuss is the path $G_1 =$  . The cost of the first vertex in any permutation is $f(0)$, so the first term in any expression is $f(0)$. Since the second vertex can either be a vertex that is not adjacent to the first or be a vertex adjacent, the cost of the second term can be $f(0)$ or $f(1)$. If the first two are not adjacent, the third vertex has to be adjacent to at least one of them, so the cost is either $f(1)$ or $f(2)$. After knowing the first three vertices, the cost of the fourth vertex is also decided, and it is the other of $f(1)$ and $f(2)$. If the first two vertices are adjacent, the cost of the third vertex can be $f(0)$, when the first two are not the central ones and the third is the degree-one vertex that is none of the first two (and in this case the fourth vertex costs $f(2)$), or $f(1)$ (and in this case the fourth vertex costs $f(1)$). So there are only four possible

cost expressions for G_1:

$$f(0)+\begin{cases} f(0)+\begin{cases} f(1)+f(2) & \text{(2. B)} \\ f(2)+f(1) & \text{(2. C)} \end{cases} \\ f(1)+\begin{cases} f(0)+f(2) & \text{(2. D)} \\ f(1)+f(1) & \text{(2. E)} \end{cases} \end{cases}$$

Since the value of the first three are equal, we only need to compare one of these expressions and the expression 2. E. By the inequality 1. A, we have the inequality $f(a)+f(b) > f(a-1)+f(b+1)$ for all $b \le a-2$. Since $0 \le 2-2$, we have that $f(0)+f(2) > f(1)+f(1)$. Therefore, the term 2. E is the cheapest cost for the graph G_1.

By a similar case-analysis, the possible costs of the graph $G_2 =$  (discussed in the introduction) are

$$f(0)+\begin{cases} f(0)+f(2)+f(2) & \text{(2. F)} \\ f(1)+f(1)+f(2) & \text{(2. G)} \end{cases}$$

Using the inequality $f(a)+f(b) \ge f(a-1)+f(b+1)$ for all $b \le a-2$, we have $f(0)+f(2) > f(1)+f(1)$. Therefore, the term 2. G is cheaper than the term 2. F.

There are three fundamentally different orders in which to build the graph $G_3 =$  . Their costs are

$$f(0)+\begin{cases} f(0)+\quad f(2)+f(3) & \text{(2. H)} \\ f(1)+\begin{cases} f(1)+f(3) & \text{(2. I)} \\ f(2)+f(2) & \text{(2. J)} \end{cases} \end{cases}$$

Invoking inequality 1. A, and that $0 \le 3-2$ and $1 \le 3-2$, we get

$$f(0)+f(3) > f(1)+f(2) \qquad \text{(2. K)}$$

and

$$f(1)+f(3) > f(2)+f(2). \qquad \text{(2. L)}$$

Adding $f(0)+f(2)$ to the both sides of the inequality 2. K, we have that the term 2. H is greater than the term 2. J. Similarly, adding $f(0)+f(1)$ to both sides of the inequality 2. L, we get that term 2. I is greater than term 2. J. Therefore, the term 2. J is the cheapest one for the graph G_3.

Let G_4 be the graph  . The cost can be

$$f(0)+\begin{cases} f(0)+\begin{cases} f(1)+f(3) & \text{(2. M)} \\ f(2)+f(2) & \text{(2. N)} \end{cases} \\ f(1)+\begin{cases} f(0)+f(3) & \text{(2. O)} \\ f(1)+f(2) & \text{(2. P)} \\ f(2)+f(1) & \text{(2. Q)} \end{cases} \end{cases}$$

The term 2. P is equal to the term 2. Q, and the term 2. M is equal to the term 2. O. Compare the term 2. M and the term 2. N, using the inequality 1. A, since $f(1)+f(3) > f(2)+f(2)$, we get that the term 2. N is less than the term 2. M. Similarly, we have $f(0)+f(2) > f(1)+f(1)$ by the inequality 1. A. Adding $f(0)+f(2)$ to both sides, we can get that the terms 2. P and 2. Q are

the cheapest cost for G_4.

Let G_5 denote the star  . The possible costs are

$$f(0)+\begin{cases} f(0)+\begin{cases} f(0)+f(3) & \text{(2. R)} \\ f(2)+f(1) & \text{(2. S)} \end{cases} \\ f(1)+\quad f(1)+f(1) & \text{(2. T)} \end{cases}$$

As above, we have $f(0)+f(2) > f(1)+(1)$. So the term 2. S is greater than the term 2. T. Since $f(0)+f(3) > f(2)+f(1)$, we get the term 2. R is greater than the term 2. S. Thus, the term 2. R is greater than the term 2. T. In conclusion, the term 2. T is the cheapest one for G_5.

Let G_6 be the graph  . The cost for G_6 is $f(0)+f(1)+f(2)+f(3)$ in any order.

In this section for graphs on 4 vertices or fewer, we use the brute-force approach to find the cheapest order to build the graphs. Our reasoning for this is that once we start to look at graphs on vertices of 5 or more, the number of connected graphs increase drastically. For example, the number of connected graphs on 5 vertices is 21, and the number of connected graphs on 6 vertices is 112. If we jump to 7 vertices, there are 853 possible connected graphs [4]. Although these graphs would be interesting to study, this section only analyzes graphs up to 4 vertices for simplicity's sake. However, we will look at graphs on 5 and 6 vertices later in this paper, in Section 7, using a computer to do a cost analysis. In later sections, we also discuss cheapest orders for infinite families of graphs.

3 Trees

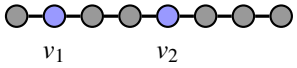
Trees are graphs where any two vertices are connected by exactly one path, wherein there are no cycles. In this section, we start with simplest form of trees such as paths and stars. Then we move on to the case of general trees.

3.1 Paths

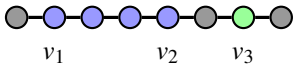
A path is an alternating sequence of vertices and edges with no repeating vertices. Thus, all vertices in a path will have degree two except the first and last vertex, which will have degree one.

Theorem 1. *The cost* $f(0)+f(1)+f(1)+\cdots+f(1)$ *is the cheapest cost for a path.*

Proof. If we start to build a path from one leaf and every vertex we build next is the vertex next to the vertex we built before, then we end by building the other leaf. This proves that the cost $f(0)+f(1)+\cdots+f(1)$ is achievable.
In general, the degree of any vertex in a connected path is either 1 or 2, so the input of a cost function for a vertex is at most 2. The first vertex v_1 that is built will always cost $f(0)$. After the first vertex, if we build a new isolated vertex v_2, that would add an $f(0)$ to the cost expression. Then, there has to be another vertex, built later, between these two vertices and adjacent to two vertices built before in order to create a path from v_1 to v_2. So this vertex costs $f(2)$.



After forming this path, we assign another isolated vertex, v_3. Again, there must be a vertex, in the later construction, between v_2 and v_3 that connects both to make a path.



The first vertex, v_1 has cost $f(0)$, but whenever there is another $f(0)$, there also must be an $f(2)$ in the later part of the expression. Thus, the general expression is equal to

$$f(0)+k[f(0)+f(2)]+mf(1)$$

where $0 \le k \le (n-1)/2$ when n is odd or $0 \le k \le (n-2)/2$ when n is even, and $m = n-(2k+1)$. Now, we can use the inequality 1. A to compare 0 and 2 with the inputs 1 and 1. Since $f(0)+f(2) > f(1)+f(1)$, we see that $f(0)+f(1)+f(1)+\cdots+f(1)$ is the cheapest cost for a path. □

3.2 Stars

A star is a tree that has one vertex with degree $n-1$ and all other vertices have degree one.

Theorem 2. *The cost* $f(0)+f(1)+\cdots+f(1)$ *is a cheapest cost for a star.*

Proof. We can first pick either the central vertex or a non-central vertex, and the cost of this vertex is $f(0)$. If the central vertex is our first vertex, we get our expression $f(0)+f(1)+\cdots+f(1)$. In the case when a non-central vertex is the first vertex, the cost of a non-central vertex before the central vertex is $f(0)$, the

cost of the central vertex is $f(m)$ where m is the number of vertices built before the central vertex, and the cost of any vertex after the central vertex is $f(1)$. For example, if a non-central vertex is the first vertex, and it is followed by the central vertex, then the expression will be $f(0)+f(1)+\cdots+f(1)$. Therefore, the general expression for a star with n vertices in this case is

$$\text{Cost}_{n\text{-star}}(m) = \underbrace{f(0)+\cdots+f(0)}_{m}+f(m)+\underbrace{f(1)+\cdots+f(1)}_{n-(m+1)}$$

Thus, we have Thus, we have

$$\begin{aligned}\text{Cost}_{n\text{-star}}(i) &= \underbrace{f(0)+\cdots+f(0)}_{i}+f(i)+\underbrace{f(1)+\cdots+f(1)}_{n-(i+1)}\\ &= \underbrace{f(0)+\cdots+f(0)}_{i}+f(i)+f(1)+\underbrace{f(1)+\cdots+f(1)}_{n-(i+2)}.\end{aligned}$$

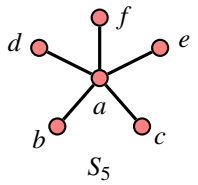
and

$$\begin{aligned}\text{Cost}_{n\text{-star}}(i+1) &= \underbrace{f(0)+\cdots+f(0)}_{i+1}+f(i+1)+\underbrace{f(1)+\cdots+f(1)}_{n-(i+2)}\\ &= \underbrace{f(0)+\cdots+f(0)}_{i}+f(0)+f(i+1)+\underbrace{f(1)+\cdots+f(1)}_{n-(i+2)}.\end{aligned}$$

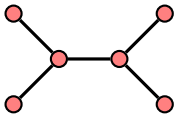
where $1 \le i \le n-2$. By inequality 1. A, we get that that $f(1)+f(i) < f(0)+f(i+1)$ when $0 \le (i+1)-2$, i.e., when $i \ge 1$. So $_{n\text{-star}}(i) <_{n\text{-star}} (i+1)$ for all $1 \le i \le n-2$. Thus, the expression $_{n\text{-star}}(1) = f(0)+f(1)+\cdots+f(1)$ is the cheapest cost for a star. □

3.3 General Trees

Since trees are quite complicated, we start by finding a general cost expression for building trees in any order. Then, we minimize this cost using inequality 1. A repeatedly to find the order for the cheapest cost.



S_5



Lemma 2.1. *For any tree T and any order of its vertices, the cost is*

$$f(0)+\sum_{i=1}^{d} a_i[(i-1)f(0)+f(i)]$$

for some nonnegative integers a_i, where d is the maximum vertex degree of T.

Example 2.1. *Take the graph S_5 shown above as an example. If we build the star in the order $\{a,b,c,d,e,f\}$, the cost will be $f(0)+5f(1)$. In this case, $a_1 = 5$ and $a_i = 0$ for $i = 2,\cdots,5$. If we build the star in the order $\{b,c,e,a,d,f\}$, the cost will be $3f(0)+2f(1)+f(3) = f(0)+2[0\cdot f(0)+f(1)]+2f(0)+f(3)$ where $a_1 = 2$, $a_3 = 1$ and $a_2 = a_4 = a_5 = 0$.*

Proof of Lemma 2.1. Fix an n-vertex tree T and some permutation of its ver-

tices, and let a_i be the number of vertices that cost $f(i)$. Thus, we have $\sum_{i=0}^{d} a_i = n$, and the cost for the tree is $a_0 f(0) + a_1 f(1) + \cdots + a_d f(d)$, where d is the greatest degree of the tree. By Lemma 0.1, we have $a_0 \cdot 0 + \sum_{i=1}^{d} a_i \cdot i = |E|$. Since $|E| = |V| - 1 = n - 1$ in a tree, by the Handshaking Lemma, we have $a_0 \cdot 0 + \sum_{i=1}^{d} a_i \cdot i = \sum_{i=1}^{d} a_i \cdot i = n - 1$, so $n = 1 + \sum_{i=1}^{d} a_i \cdot i$. Since $\sum_{i=0}^{d} a_i = n$, we have

$$\begin{aligned} a_0 &= n - \sum_{i=1}^{d} a_i \\ &= 1 + \sum_{i=1}^{d} a_i \cdot i - \sum_{i=1}^{d} a_i \\ &= 1 + \sum_{i=1}^{d} a_i \cdot (i-1). \end{aligned}$$

Therefore, the cost of the tree is

$$\begin{aligned} a_0 f(0) + a_1 f(1) + \cdots + a_d f(d) &= \left[1 + \sum_{i=1}^{d} a_i \cdot (i-1)\right] f(0) + \sum_{i=1}^{d} a_i f(i) \\ &= f(0) + \sum_{i=1}^{d} a_i \cdot (i-1) f(0) + \sum_{i=1}^{d} a_i f(i) \\ &= f(0) + \sum_{i=1}^{d} a_i \left[(i-1) f(0) + f(i)\right]. \square \end{aligned}$$

Lemma 2.2. *A graph G has an order with cost $f(0) + f(1) + f(1) + \cdots + f(1)$ if and only if G is a tree.*

Proof. First assume that G is a tree. We pick an arbitrary vertex, costing $f(0)$, to be our first vertex. After this, if we continue choosing to build vertices that already have a neighbor built, then the cost of each of these vertex is $f(1)$. So the total cost is $f(0) + f(1) + f(1) + \cdots + f(1)$. Next, assume that the graph G has a permutation with cost $f(0) + f(1) + f(1) + \cdots + f(1)$. To prove G is a tree, we need to prove (i) G is connected, and (ii) G has no cycles. Since there is only one $f(0)$, the graph G has to be connected, as the number of $f(0)$ is at least the number of components. If G has a cycle, then the last vertex of the cycle to be build will cost $f(k)$ with $k \geq 2$. But there is no $f(2)$ in the cost expression, so G is acyclic. Since G is a connected acyclic graph, we have that G is a tree. $\square$

Theorem 3. *The cost $f(0) + f(1) + \cdots + f(1)$ is a cheapest cost for any tree.*

Proof. By Lemma 2.2, for any tree we can achieve the cost expression $f(0) + f(1) + \cdots + f(1)$. From Lemma 2.1, we know that the cost for an arbitrary order of a tree is $f(0) + \sum_{i=1}^{d} a_i[(i-1)f(0) + f(i)]$. Using inequality 1. A, our strategy is to minimize the expression by repeatedly replacing expressions of the form $f(0) + f(a)$ with $f(1) + f(a-1)$. We continue taking away $f(0)$s and implementing this substitution until all $f(0)$s are exhausted. Indeed, for each

term $(i-1)f(0)+f(i)$ where $i \geq 2$, we get

$$\begin{aligned}
\underbrace{f(0)+f(0)+\cdots+f(0)}_{i-1}+f(i) &= \underbrace{f(0)+f(0)+\cdots+f(0)}_{i-2}+f(0)+f(i) \\
&> \underbrace{f(0)+f(0)+\cdots+f(0)}_{i-2}+f(1)+f(i-1) \\
&= \underbrace{f(0)+f(0)+\cdots+f(0)}_{i-3}+f(0)+f(i-1)+f(1) \\
&> \underbrace{f(0)+f(0)+\cdots+f(0)}_{i-3}+f(1)+f(i-2)+f(1) \\
&\vdots \\
&> f(0)+f(1)+f(2)+\underbrace{f(1)+\cdots+f(1)}_{i-3} \\
&> f(1)+f(1)+f(1)+\underbrace{f(1)+\cdots+f(1)}_{i-3} \\
&= f(1)+\cdots+f(1).
\end{aligned}$$

Therefore, for each $a_i[(i-1)f(0)+f(i)]$ where $1 \leq i \leq d$, we have $a_i[(i-1)f(0)+f(i)] > a_i \cdot i \cdot f(1)$. Since $\sum_{i=1}^{d} a_i \cdot i = n-1$, we conclude that the cost $f(0)+f(1)+\cdots+f(1)$ is a cheapest cost for a tree. □

4 Cycles

In Section 2, we computed costs for all 3-vertex and 4-vertex graphs. For the 3-vertex cycle  , every permutation gives the cost expression $f(0)+f(1)+f(2)$, while for the four-cycle  , the two possible cost expressions are

$$f(0)+\begin{cases} f(0)+f(2)+f(2) \\ f(1)+f(1)+f(2). \end{cases}$$

Using inequality 1. A, we were able to compare different build orders to find that the cheapest cost is $f(0)+f(1)+f(1)+f(2)$. We can use this framework to find the cheapest cost expression for cycles of any length.

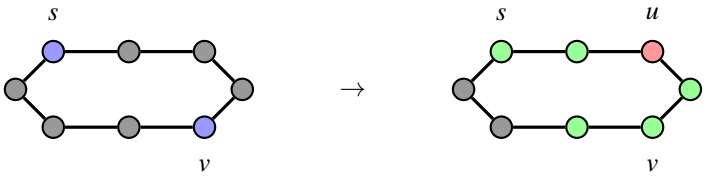
Theorem 4. *The cost $f(0)+f(1)+f(1)+\cdots+f(1)+f(2)$ is the cheapest for a cycle.*

Proof. We can achieve this if we build a cycle clockwise or counterclockwise and each vertex built next is a neighbor of the vertex built before. Now we show that it is minimal.

As in each example above, the cost expression of a cycle starts with the first isolated vertex resulting in $f(0)$. Also, each expression of a cycle has at least one $f(2)$: by definition, every vertex in a cycle has degree 2, and therefore the last vertex will always have two neighbors already built, resulting in the last $f(2)$ in the expression.

After building the first vertex s, we can either build an adjacent vertex costing

$f(1)$ or build an isolated vertex. But whenever we build another isolated vertex, v, costing $f(0)$, this isolated vertex will be in a part that is not connected to the first vertex. Thus, when we build other vertices later, there is a "connecting" vertex, u, connecting the part where the first vertex s is to the isolated part where the vertex v is. So in a later term of the expression, there is an $f(2)$ for the vertex u. An example is below. The vertex u connects the two green parts where s and v separately are.



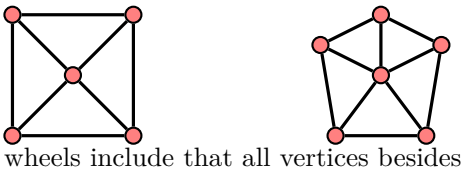
This continues similarly when we build other new isolated vertices: each $f(0)$ corresponds to creating a new component, and each $f(2)$ (other than the last vertex) corresponds to merging two components. Thus, the general cost expression of a cycle with n vertices is

$$m[f(0)+f(2)]+(n-2m)f(1)$$

where $1 \leq m \leq \left\lfloor \frac{n}{2} \right\rfloor$. This general formula allows us to use inequality 1. A in the form $f(0)+f(2) > f(1)+f(1)$ to reduce m by 1 while making the cost smaller. Therefore, the cost $f(0)+f(1)+f(1)+\cdots+f(1)+f(2)$ is the cheapest cost for a cycle. □

5 Wheels

As described in [2], a *wheel* with n vertices is formed from a cycle C_{n-1} (called the *rim*) by adding a vertex (called the *hub*) and, for each vertex in the rim, an edge that is incident with that vertex and the hub. Some examples of wheels are as follows:



Some properties of wheels include that all vertices besides the central vertex have degree 3 and the central vertex has degree $n-1$.

Lemma 4.1. *The general cost expression of a wheel is*

$$af(0)+bf(1)+cf(2)+f(a+b+c)+(a-c)f(3)+k[f(1)+f(3)]+\ell f(2)$$

where a, b, c, k, l are nonnegative integers such that $\ell = n-(2a+b+1+2k)$ *and either* $a=b=c=0$, *or* $a=c$ *and* $a+b+c=n-1$ *and* $k=0$, *or* $c \leq a-1$.

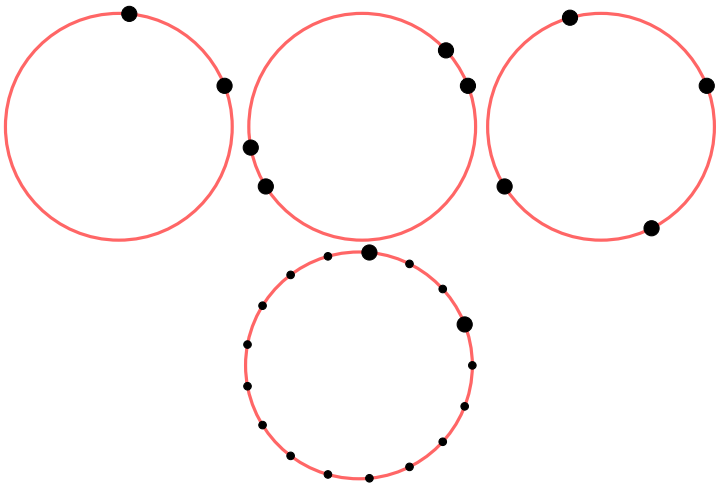
Proof. In the case where we build the hub first, we see by the proof of Theorem 4 that the resulting cost will be $f(0)+k[f(1)+f(3)]+(n-1-2k)f(2)$ where k and $n-1-2k$ are nonnegative integers, and this is exactly the case $a=b=c=0$ of the claim.

Similarly, in the case where we build the hub last, we see from the proof of Theorem 4 that the resulting cost will be $a[f(0)+f(2)]+(n-1-2a)f(1)+f(n-$

1), which is exactly the case $a = c$ and $a+b+c = n-1$ and $k = 0$ of the claim. Now consider the case that we do not build the hub first or last.

Let us divide the target expression into two parts: (i) $af(0)+bf(1)+cf(2)$ which will represent the cost before we build the hub, and (ii) $f(a+b+c)+(a-c)f(3)+k[f(1)+f(3)]+\ell f(2)$ which will represent the cost of the part of the graph that we build after the hub (including the cost of the hub itself).

For the first part, let $i = a+b+c$ be the number of vertices that we build on the rim before we build the hub. The cost of each of these vertices is either $f(0)$, $f(1)$, or $f(2)$. Let a be the number of these vertices that cost $f(0)$, let b be the number that cost $f(1)$, and let c be the number that cost $f(2)$. At this stage, a vertex costing $f(2)$ is only built when it joins two components, each of which was started by a vertex that cost $f(0)$, so $c \le a-1$.



Then build the hub, which has i neighbors on the rim and so costs $f(i)$. As we complete the rim, there must be $m = a-c$ vertices that join rim components that were built before the hub, and these all cost $f(3)$; moreover, each time we create a new rim component (at a cost $f(1)$), we are forced later to merge two rim components (at a cost $f(3)$). Let the number of times we do that be k, accounting for $2k$ vertices. Each of the other $\ell = n-(i+1+m+2k)$ vertices not already accounted for is built on the rim with one rim neighbor and the hub as a neighbor, and so costs $f(2)$. Every vertex falls into one of these cases; substituting $m = a-c$ and $i = a+b+c$ where needed completes the proof. □

Theorem 5. *The cheapest cost for the wheel with n vertices is $f(0)+f(1)+(n-3)f(2)+f(3)$.*

Proof. First we show that this value is achievable. We pick the central vertex first, with cost $f(0)$. Then we build vertices consecutively around the cycle. The cost of the first vertex on the cycle in this order is $f(1)$, and the cost of the last vertex (which closes the wheel) is $f(3)$. Each other vertex is adjacent to the central vertex and the one built immediately before it, so has cost $f(2)$. Therefore the cost expression under this construction is $f(0)+f(1)+(n-3)f(2)+f(3)$. We can also get this cost expression if one vertex on the wheel is built first and

the central vertex is built second. We claim this cost expression is the cheapest cost for the wheel.

Now consider an arbitrary order, with cost given by Lemma 4.1. By the inequality 1. A, we have $f(1)+f(3)>2f(2)$ and $f(0)+f(2)>2f(1)$. By applying the first of these inequalities $k+(a-c-1)$ times and the second one $a-1$ times, we get

$$\begin{aligned} &af(0)+bf(1)+cf(2)+f(a+b+c)+(a-c)f(3)+\\ &\quad+(n-2a-b-1-2k)f(2)+k[f(1)+f(3)]\\ &>f(0)+(a+b+c-1)f(1)+(n-a-b-c-2)f(2)+f(a+b+c)+f(3). \end{aligned} \tag{5. U}$$

If $a+b+c=1$ or 2, this is exactly the inequality we claimed. Suppose instead that $a+b+c\geq 3$, and let us deal first with the sum of the second and the fourth terms $(a+b+c-1)f(1)+f(a+b+c)$. By the inequality 1. A, we get $f(1)+f(i)>f(2)+f(i-1)$ for all $i\geq 3$. We keep pairing one copy of $f(1)$ with the extra term, applying the inequality $f(1)+f(a+b+c-k)>f(2)+f(a+b+c-k-1)$ for $k=0,1,\cdots,a+b+c-3$. Thus, for $a+b+c\geq 3$, we have the inequalities

$$\begin{aligned} &(a+b+c-1)f(1)+f(a+b+c)\\ &\qquad>f(1)+(a+b+c-2)f(2)\\ &\qquad+f(a+b+c-(a+b+c-2))\\ &\qquad=f(1)+(a+b+c-1)f(2). \end{aligned} \tag{5. V}$$

Combining the inequalities (5. V) and (5. U), we have that, for $a+b+c\geq 3$, the general cost expression from Lemma 4.1 is larger than

$$\begin{aligned} &[f(0)+(n-a-b-c-2)f(2)+f(3)]+[f(1)+(a+b+c-1)f(2)]\\ &\qquad=f(0)+f(1)+(n-3)f(2)+f(3). \end{aligned}$$

Therefore, since we also have the same inequality for $a+b+c<3$, we conclude that the cheapest cost for the wheel is $f(0)+f(1)+(n-3)f(2)+f(3)$, as claimed. □

6 Ladder Graphs, Grid Graphs, and Complete Bipartite Graphs

In this section, we consider three families of bipartite graphs: the ladder graphs, the general grid graphs, and the complete bipartite graphs.

6.1 Ladder Graphs

A ladder graph is formed by taking two paths of the same length and connecting corresponding vertices by an edge. Thus, when drawn in the plane, the ladder graph L_n looks like $n-1$ "boxes" stacked on top of each other, and has $2n$ vertices and $3n-2$ edges. Some examples of ladders are shown in Figure 1.

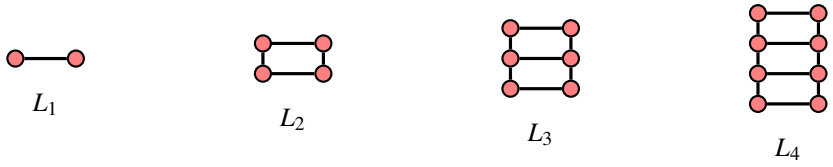


Figure 1: Small ladder graphs

Lemma 5.1. *The general cost expression for the ladder graph L_n is*

$$af(0)+bf(1)+c(2)+df(3), \qquad (6.\ \mathrm{W})$$

where a,b,c,d are nonnegative integers such that $a+b+c+d=2n$, $b+2c+3d=3n-2$, and $1\le a\le n$.

Proof. Obviously, we have $a+b+c+d=2n$ since there are $2n$ vertices. By Lemma 0.1, we get $a\cdot 0+b\cdot 1+c\cdot 2+d\cdot 3=3n-2$, i.e., $b+2c+3d=3n-2$. The cost of the first vertex built is always $f(0)$ so a is at least 1. For any graph and any order of the vertices, the vertices of cost $f(0)$ form an independent set in the graph. Since the largest independent set in L_n has size n, we get that $1\le a\le n$. □

Theorem 6. *The cheapest cost for the ladder graph L_n is $f(0)+nf(1)+(n-1)f(2)$.*

Proof. We start to build either the left or the right vertex of the top row of the ladder L_n, and its cost is $f(0)$. Then keep building a vertex that is one row lower and adjacent to the last vertex until the last row, and then build the vertex that is adjacent to the last vertex, and the cost of each is $f(1)$. We build the remaining vertices consecutively all the way up, and each costs $f(2)$. So the cost expression is $f(0)+nf(1)+(n-1)f(2)$. We claim that this expression is a cheapest cost for the ladder graph L_n. Since we have $a+b+c+d=2n$, we get $b+c+d=2n-a$. We can rewrite $\begin{cases} b+c+d=2n-a \\ b+2c+3d=3n-2 \end{cases}$ as $\begin{cases} c=n-2d+a-2 \\ b=n+d-2a+2 \end{cases}$. Plugging these in the general cost expression (6. W) for L_n, we get that the cost of any order is of the form $af(0)+(n+d-2a+2)f(1)+(n-2d+a-2)f(2)+df(3)$. By the inequality 1. A, we have $f(0)+f(3)>f(1)+f(2)$ and $f(1)+f(3)>2f(2)$.

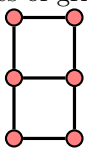
Applying these to the general cost expression, we get inequalities

$$\begin{aligned} & af(0)+(n+d-2a+2)f(1)+(n-2d+a-2)f(2)+df(3) \\ & > f(0)+(n+d-2a+2)f(1)+(n-2d+a-2)f(2)+(d-a+1)f(3) \\ & \quad +(a-1)f(1)+(a-1)f(2) \\ & > f(0)+nf(1)+(n-2d+2a-3)f(2)+2(d-a+1)f(2) \\ & = f(0)+nf(1)+(n-1)f(2). \end{aligned}$$

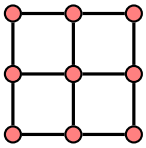
Therefore, $f(0)+nf(1)+(n-1)f(2)$ gives us the cheapest cost. □

6.2 Grid Graphs

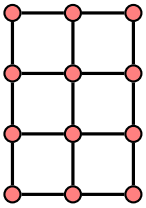
For the grid graph $G_{m\times n}$, m is the number of rows and n is the number of columns, so there are mn vertices and $(m-1)n+m(n-1)=2mn-m-n$ edges. Since the highest degree for a grid graph is 4, the largest possible input for the function f is 4. Thus, the cost expression for the $G_{m\times n}$ is $af(0)+b(1)+cf(2)+df(3)+ef(4)$, where we have $a+b+c+d+e=mn$ and $b+2c+3d+4e=2mn-m-n$. Here are some examples of grid graphs:



$G_{3\times 2}$



$G_{3\times 3}$



$G_{4\times 3}$

Theorem 7. *The cheapest cost for the grid graph $G_{m\times n}$ is $f(0)+(m+n-2)f(1)+(mn-m-n+1)f(2)$.*

Proof. First we show that this cost is achievable. If we start from the leftmost vertex of top row, and then build down consecutively until the bottom row and then build right consecutively until the last column; and we build second column from the bottom to top, and keep doing that consecutively for the remaining columns, we will get the cost expression $f(0)+(m+n-2)f(1)+(mn-m-n+1)f(2)$. It remains to show that this cost is optimal.

Solving the equations $\begin{cases} a+b+c+d+e=mn \\ b+2c+3d+4e=2mn-m-n \end{cases}$ for c and d gives

$$\begin{cases} d=2a+b-2e-m-n \\ c=mn-3a-2b+e+m+n \end{cases}. \qquad (6.\ \text{X})$$

An arbitrary cost expression for a grid graph $G_{m\times n}$ is $af(0)+bf(1)+cf(2)+df(3)+ef(4)$. Plugging 6. X into this expression, it becomes

$$af(0)+bf(1)+(mn-3a-2b+e+m+n)f(2)$$
$$+(2a+b-2e-m-n)f(3)+ef(4). \quad (6.\ \mathrm{Y})$$

We split our analysis into two cases: (i) $a+b-m-n+1\geq 0$, and (ii) $a+b-m-n+1<0$. The inequality 1. A gives $f(2)+f(4)>2f(3)$, $f(0)+f(3)>f(1)+f(2)$, and $f(1)+f(3)>2f(2)$. Thus, for the case (i), replacing $ef(2)+ef(4)$ with the cheaper $2ef(3)$ and $(a-1)f(0)+(a-1)f(3)$ with the cheaper $(a-1)f(1)+(a-1)f(2)$ in the cost expression 6. Y gives the relation

$$\begin{aligned}
&af(0)+bf(1)+(mn-3a-2b+e+m+n)f(2)\\
&\quad+(2a+b-2e-m-n)f(3)+ef(4)\\
>&f(0)+(a+b-1)f(1)+(mn-2a-2b+m+n-1)f(2)\\
&\quad+(a+b-m-n+1)f(3).
\end{aligned}$$

Since $a+b-m-n+1\geq 0$ in this case, replacing $(a+b-m-n+1)f(1)+(a+b-m-n+1)f(3)$ with the cheaper $2(a+b-m-n+1)f(2)$ in this last expression gives

$$\begin{aligned}
&f(0)+(a+b-1)f(1)+(mn-2a-2b+m+n-1)f(2)\\
&\quad+(a+b-m-n+1)f(3)\\
>&\ f(0)+(m+n-2)f(1)+(mn-2a-2b+m+n-1)f(2)\\
&\quad+2(a+b-m-n+1)f(2)\\
=&\ f(0)+(m+n-2)f(1)+(mn-m-n+1)f(2).
\end{aligned}$$

For case (ii), when $a+b-m-n+1<0$, we have $m+n-a-b-1>0$. In this case, replace $ef(2)+ef(4)$ with the cheaper $2ef(3)$ and then replace $(2a+b-m-n)f(0)+(2a+b-m-n)f(3)$ with the cheaper $(2a+b-m-n)f(1)+(2a+b-m-n)f(2)$; after that, replace $(m+n-a-b-1)f(0)+(m+n-a-b-1)f(2)$ with the cheaper $2(m+n-a-b-1)f(1)$. This gives

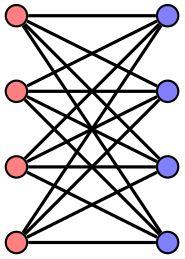
$$\begin{aligned}
&af(0)+bf(1)+(mn-3a-2b+e+m+n)f(2)+(2a+b-2e-m-n)f(3)+ef(4)\\
>&af(0)+bf(1)+(mn-3a-2b+m+n)f(2)+(2a+b-m-n)f(3)\\
>&(m+n-a-b)f(0)+(2a+2b-m-n)f(1)+(mn-a-b)f(2)\\
>&f(0)+(m+n-2)f(1)+(mn-m-n+1)f(2).
\end{aligned}$$

Since the two cases are exhaustive and the inequality holds in both cases, our claimed cost expression is optimal. □

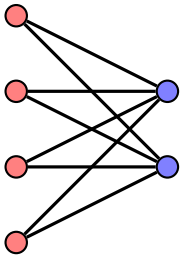
6.3 Complete Bipartite Graphs

A bipartite graph is a graph with two vertex sets, M and N, such that no vertex in M is connected to another vertex in M, and similarly no vertex in N is connected to another vertex in N. The complete bipartite graph $K_{m,n}$ is characterized by the fact that M has size m, N has size n, and every vertex in set M is connected to every vertex in set N. Thus, the degree of each vertex in M is equal to n, and similarly, the degree for each vertex in N is equal to m. Some complete bipartite graphs, denoted $K_{n,n}$, have the same number of vertices in each set. Below are a few examples.

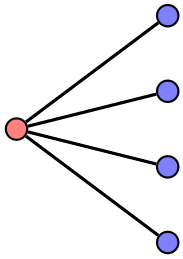
Example 7.1.



$K_{4,4}$

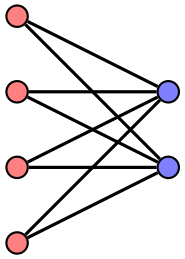


$K_{4,2}$

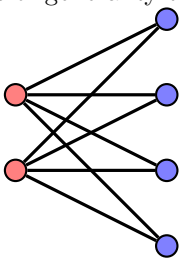


$K_{1,4}$, *or a claw*

It is important to note that $K_{n,m}$ and $K_{m,n}$ are isomorphic, as illustrated in the figure below, so that we can assume without loss of generality that $m \geq n$.



$K_{4,2}$



$K_{2,4}$

Theorem 8. *The cheapest cost of a complete bipartite graph $K_{m,n}$, where $m \geq n$, is $f(0)+2[f(1)+f(2)+f(3)+\cdots+f(n-1)]+(m-n+1)f(n)$, where m is the number of vertices in the larger set and n is the number of vertices in the smaller set.*

Proof. Given a complete bipartite graph $K_{m,n}$, consider the following special kinds of permutations of the vertices: start by building a vertex in one of the vertex sets, then the second vertex built must be from the other set. The third vertex should be from the same set as the first vertex built. Continue building this complete bipartite graph by alternating between vertex sets. Once the smaller set has been completed and all vertices in that set are built, build the remaining vertices in the larger set in any order. We will refer to this as "the zigzag way," and we represent these orders by the permutations $\{M,N,M,N,\cdots,M,N,M,M,\cdots,M\}$ and $\{N,M,N,M,\cdots,N,M,M,\cdots,M\}$, depending on which side we start on.

Our claim is that these two are the cheapest ways to build $K_{m,n}$. We observe that they both have cost $f(0)+2[f(1)+f(2)+\cdots+f(n-1)]+(m-n+1)f(n)$. We will prove that this is optimal by contradiction, starting with the assumption that there is a cheaper non-zigzag way to build $K_{m,n}$. Here are some non-zigzag examples: $\{M,M,M,N,N,S\}$, $\{N,N,M,N,M,N,M,S\}$, and $\{M,N,M,N,\cdots,M,M,M,N,M,N,S\}$, where S represents an arbitrary order of whatever vertices remain. No matter the permutation, every non-zigzag permutation belongs in one of these two cases: (i) there are two or more Ms built consecutively before all Ns have been built, or (ii) there are two or more Ns built consecutively.

We first prove that if two vertices from M are built consecutively before all vertices from N have been built, then the permutation is not optimal. For instance, take the permutation $Y_1=\{M,N,M,N,\cdots,M,N,M,M,N,S\}$, where there are two Ms built together, followed by an N. Here, S is the rest of the permutation after the change in pattern, and is arbitrary. Define a new permutation $Z_1=\{M,N,M,N,\cdots,M,N,M,N,M,S\}$ by switching the second of the two consecutive Ms with the N after it. Note here that S remains the same, thus the rest of the permutation is identical for both Y_1 and Z_1. Then, we have that the cost expression for Y_1 and Z_1 respectively are

$$\begin{aligned}\text{Cost}_{Y_1} &= f(0)+2f(1)+2f(2)+\cdots+2f(b)\\ &\quad +f(b)+f(b+2)+\text{Cost}_S.\end{aligned}$$

and

$$\begin{aligned}\text{Cost}_{Z_1} &= f(0)+2f(1)+2f(2)+\cdots+2f(b)\\ &\quad +2f(b+1)+\text{Cost}_S.\end{aligned}$$

where b is the number of Ns before the two consecutive Ms in Y_1. Since $f(b)+f(b+2)>2f(b+1)$ by the inequality 1. A, we have $\text{Cost}_{Y_1}>\text{Cost}_{Z_1}$.

More generally, suppose we have a permutation Y_k that (after some zigzag alternation) has a group of $k+1$ Ms built consecutively before the last N where $k\geq 1$. If Y_k starts with M, then we can write it as

$$Y_k=\{M,N,M,N,\cdots,M,N,M,\underbrace{M,\cdots,M}_{k},N,S\}.$$

Then define

$$Z_k=\{M,N,M,N,\cdots,M,N,M,\underbrace{M,\cdots,M}_{k-1},N,M,S\}.$$

Their cost expressions are

$$\begin{aligned}\text{Cost}_{Y_k} &= f(0)+2f(1)+2f(2)+\cdots+2f(b)\\ &\quad +[kf(b)+f(b+k+1)]+\text{Cost}_S.\end{aligned}$$

and

$$\begin{aligned}\text{Cost}_{Y_k} &= f(0)+2f(1)+2f(2)+\cdots+2f(b)\\ &\quad +[kf(b)+f(b+k+1)]+\text{Cost}_S.\end{aligned}$$

Since $f(b)+f(b+k+1)>f(b+1)+f(b+k)$ by inequality 1. A, we have that $kf(b)+f(b+k+1)>(k-1)f(b)+f(b+1)+f(b+k)$, so that $\text{Cost}_{Y_k}>\text{Cost}_{Z_k}$, and therefore Y_k is not optimal. The case that Y_k begins with N is very similar,

and we omit the details.
Now we prove that if two or more vertices from N are built consecutively, then the permutation is not optimal. We choose some such permutation

$$\overline{Y}_k = \{M,N,M,N,\cdots,M,N,\underbrace{N,\cdots,N}_{k},M,S\},$$

and its cost expression is

$$\begin{aligned}\text{Cost}_{\overline{Y}_k} &= f(0)+2f(1)+2f(2)+\cdots+2f(b)\\ &\quad +f(b+1)+kf(b+1)+f(b+k+1)+\text{Cost}_S.\end{aligned}$$

Then define the permutation

$$\overline{Z}_k = \{M,N,M,N,\cdots,M,N,\underbrace{N,\cdots,N}_{k-1},M,N,S\},$$

whose cost expression is

$$\begin{aligned}\text{Cost}_{\overline{Z}_k} &= f(0)+2f(1)+2f(2)+\cdots+2f(b)\\ &\quad +f(b+1)+(k-1)f(b+1)+f(b+k)+f(b+2)+\text{Cost}_S.\end{aligned}$$

Since $f(b+1)+f(b+k+1) > f(b+2)+f(b+k)$, we get $\text{Cost}_{\overline{Y}_k} > \text{Cost}_{\overline{Z}_k}$. The case that $\overline{Y}_k$ begins with N is again similar.
Now let Y be an optimal permutation. If Y is not a zigzag way, then we have shown that there is another permutation Z which costs less than Y, a contradiction. Therefore, the zigzag way is the cheapest permutation to build the complete bipartite graph $K_{m,n}$. □

7 Cost Analysis for Graphs on Five Vertices and Six Vertices

Now that we have analyzed various families of graphs to find the cheapest cost, we can extend this paper to graphs that are not necessarily categorized into a family, such as bipartite graphs or trees. Instead, we can look at all possible connected graphs on five and six vertices, as well as the Petersen graph. As discussed in Section 2, we know that there are 21 connected graphs on five vertices and 112 connected graphs on six vertices. To explore these graphs, we used an algorithm to compute the optimal degree sequence with the cheapest cost of each graph on five and six vertices. We chose two positive, decreasing, convex functions to analyze: $f(x) = (1+x)^{-1}$ and $g(x) = 2^{-x}$.
Using the programming language Python and the package NetworkX, we first found all possible permutations of the vertices for each graph on five or six vertices. Then, we recorded the degree of every vertex in each permutation at the time is was built. Finally, we collected this data and were able to compute the costs for each permutation, outputting an array with the cheapest cost and its optimal degree sequence. The code can be found below in Appendix A.
In this experiment, we observed that for all connected graphs on five and six vertices, the cheapest cost was achieved with the same orders across both cost functions. We see this by following the degreeswhenbuilt variable, which records what orders the vertices must be built to attain the cheapest cost. Each graph

had the same orders for degreeswhenbuilt, illustrating that for 5-vertex and 6-vertex graphs, the orders for their respective cheapest costs are the same. The optimal degree sequences for finding the cheapest cost for each cost function on five-vertex graphs are given in Table 1. Finally, we tested the Petersen graph which has ten vertices, to see if it would present us with the same conclusion. We found that for the Petersen graph, it was again true that the orders for the cheapest costs using functions f and g were the same. This code can be used in principle to see if the same conclusion holds for higher order graphs. Our limitations included computer processing time, since it was able to quickly give back data on five and six vertices, but not on higher order graphs.

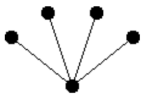
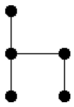
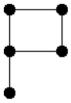
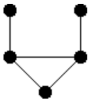
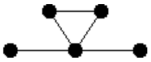
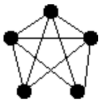
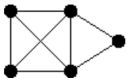
8 Can we go further?/Future expectation

In every case we studied above, the optimal cost is achieved for a permutation in which we never build a vertex that creates a new component. Thus, we conjecture that, for any graph, to get the minimum cost, the optimal permutation must be a connected order, i.e., we never start a new component. There are a lot of families of simple connected graphs other than those considered in this paper, and there are also graphs that are not simple connected. Thus, there are many ways to extend this project and further discover more about cost effectively building graphs. Here are some examples that seem particularly interesting.

Question 8.1. *What are the optimal orders for building hypercube graphs?*

Question 8.2. *How does the question change for graphs that are not simple (i.e., with loops or multiple edges allowed)? What about directed graphs (where the cost to build v is $f(k)$ where k is the number of vertices w such that $w \to v$ is a directed edge in the graph)?*

It might also be interesting to study whether finding optimal costs is easier in special families of graphs, like Eulerian or Hamiltonian graphs.

Graph	Optimal degrees
	0, 1, 1, 1, 1
	0, 1, 1, 1, 1
	0, 1, 1, 1, 1
	0, 1, 1, 1, 2
	0, 1, 1, 1, 2
	0, 1, 1, 1, 2
	0, 1, 1, 1, 2
	0, 1, 2, 3, 4
	0, 1, 2, 3, 3
	0, 1, 2, 2, 3
	0, 1, 2, 2, 3

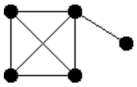
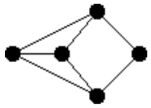
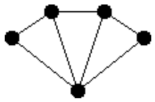
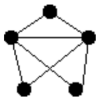
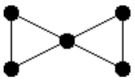
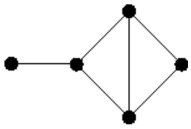
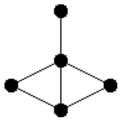
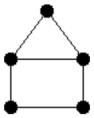
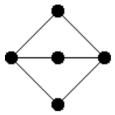
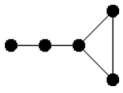
Graph	Optimal degrees
	0, 1, 1, 2, 3
	0, 1, 2, 2, 2
	0, 1, 2, 2, 2
	0, 1, 2, 2, 2
	0, 1, 1, 2, 2
	0, 1, 1, 2, 2
	0, 1, 1, 2, 2
	0, 1, 1, 2, 2
	0, 1, 1, 2, 2
	0, 1, 1, 1, 2

Table 1: This table shows every isomorphism class of connected graphs on 5 vertices and the cheapest order to build them.

9 Acknowledgements

This project was carried out as an undegraduate research project at —George Washington University, under the mentorship of Professor —Joel Lewis . We are grateful to an anonymous referee for detailed comments that improved the paper.

Bibliography

[1] Novikoff, Timothy. *Algorithmic Education Theory.* Ph.D. thesis, Cornell University, 2013. https://ecommons.cornell.edu/handle/1813/33956

[2] Bonin, Joseph. *Notes for Math 3632 Introduction to Graph Theory.* April 12 2021.

[3] Maclagan, Diane, *Connected simple graphs on four vertices*, Lecture notes, University of Warwick, 2005. https://homepages.warwick.ac.uk/~masgar/Teach/2005_428/2005_09_12lecture_paths.pdf

[4] Sloane, N. J. A. *Number of connected graphs with n nodes.* The On-Line Encyclopedia of Integer Sequences. August 15 2023https://oeis.org/A001349

0 APPENDIX A. Python code

Below includes code specifically for star graphs, random trees, and complete bipartite graphs. Each of them need to be commented out individually to see the code executed. All connected graphs on five and six vertices need to be built according to NetworkX commands found here https://networkx.org/documentation/stable/tutorial.html.

```
import networkx as nx
import pandas as pd
from math import *
import numpy as np

# code for stars
#v = 4
#G = nx.star_graph(v)

# code for random trees
#v = 4
#G = nx.random_tree(v)

# code for complete bipartite graphs
#n_1 = 4
#n_2 = 2
#nodes = list(range(0,n_1+n_2))
#G = nx.complete_bipartite_graph(n_1,n_2,create_using=None)
```

```
nodes = list(range(0,v))
nx.draw(G, with_labels=True)


# Finds all possible orders of building a graph with v
vertices.
Source: https://www.geeksforgeeks.org/generate-all-the-
permutation-of-a-list-in-python/

def permutation(lst):
    if len(lst) == 0:
        return []
    if len(lst) == 1:
        return [lst]
    l = []
    for i in range(len(lst)):
        m = lst[i]
        remLst = lst[:i] + lst[i+1:]
        for p in permutation(remLst):
            l.append([m] + p)
    return l

# Takes each permutation and computes the cost using
two different functions
(costreciprocal and costexponential). Records the
degree of each vertex
(degreeswhenbuilt) at the time it is built and puts
both the costs and the
degreeswhenbuilt into their respective arrays.

data = list(G.nodes)
float_list_Rec = []
float_list_Ex = []
array_recip=[]
array_exp=[]
for p in permutation(data):
    costreciprocal = 0
    costexponential = 0
    degreeswhenbuilt = []
    dictRec = {"A":[],"B":[]}
    for x in nodes:
        H = G.subgraph([p[i] for i in range(0, x+1)])
        d = list(H.degree([p[x]]))[0][1]
        degreeswhenbuilt.append(d)
        costreciprocal += 1/(d+1)
        costexponential += 2**(-d)
```

```
        if x==v-1:
            if [costreciprocal, str(degreeswhenbuilt)]

            not in array_recip:
                array_recip.append([costreciprocal,

                str(degreeswhenbuilt)])
            if [costexponential, str(degreeswhenbuilt)]
            not in array_exp:
                array_exp.append([costexponential,

                str(degreeswhenbuilt)])
            float_list_Rec.append(costreciprocal)
            float_list_Ex.append(costexponential)

# Sorts vales in ascending order.

array_recip.sort()
array_exp.sort()

print('The minimum cost using the reciprocal cost function is ',
      min(float_list_Rec), ', and here is the list of degree
      sequences and costs:')
print(array_recip)
print()
print('The minimum cost using the exponential cost function is ',
      min(float_list_Ex), ', and here is the list of degree
      sequences and costs:')
print(array_exp)
```

Ball State Undergraduate Mathematics Exchange, 18(1), Fall 2024, 51-64
https://openjournals.bsu.edu/mathexchange
DOI: 10.33043/965597rq

Pages 51 – 64

A New Triangle Generation of Some Generalized Genocchi Numbers

Feryal Alayont, Stephanie Loewen, Vasily Zadorozhnyy*

Feryal Alayont is a professor of mathematics at Grand Valley State University. She received her Ph.D. in mathematics from the University of Minnesota. Her research interests include combinatorics and the teaching and learning of mathematics.

Stephanie Loewen received her B.S. in mathematics from Grand Valley State University, where this research was conducted, and her M.S. in Data Science from the University of Notre Dame. Currently Stephanie is employed at Ground Vehicle Systems Center as a mathematician, where she primarily performs data analysis.

Vasily Zadorozhnyy worked on this project when he was a mathematics undergraduate student at the Grand Valley State University during the Fall 2016 semester. Later, Vasily received his PhD in mathematics from the University of Kentucky and currently is an Advanced Computer Scientist in Machine and Deep Learning at SRI International in Princeton, NJ.

Abstract

The two-dimensional rook theory can be generalized to three and higher dimensions by assuming that rooks attack along hyperplanes. Using this generalization, Alayont and Krzywonos defined two separate families of boards in three and higher dimensions generalizing the two-dimensional triangular boards whose rook numbers correspond to generalizations of Stirling numbers of the second kind and Genocchi numbers. This combinatorial interpretation of the Genoc-

***Corresponding author**: alayontf@gvsu.edu

chi numbers was shown to provide a new triangle generation of the Genocchi numbers. In this paper, we prove similar triangle generations for the third and fourth generalized Genocchi numbers using rook numbers of boards in four and five dimensions.

1 Introduction

The theory of rook polynomials (in two dimensions) was developed to provide a way of counting permutations with restricted positions. Consider a non-attacking rook placement on a square board with restricted cells where each row and column has one rook. This placement corresponds to a permutation σ with restrictions by letting a rook placed in row i column j to mean $\sigma(i) = j$. In other words, the permutation is obtained by reading the column numbers of the rooks from top row to bottom. For example, the configuration of non-attacking rooks shown in Fig. 1 corresponds to the permutation (132) in cycle notation, or 312 in word notation. On this board, the restrictions for the permutations, represented with the shaded cells, are $\sigma(1) \neq 1,2$ and $\sigma(2) \neq 2$.

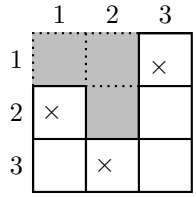


Figure 1: The rook placement corresponding to the permutation (132).

It is known that the numbers of ways to place non-attacking rooks on certain families of boards correspond to well-known number sequences. One such example is the family of triangular boards. The two-dimensional size 5 triangular board is shown in Fig. 2. It is well known that the number of ways to place k non-attacking rooks on a triangular board of size n corresponds to the Stirling numbers of the second kind, $S(n+1,n+1-k)$. Recall that the Stirling number $S(n,k)$ counts the number of ways to partition a set of n elements into k non-empty subsets. We review the proof of this result in Section 2.

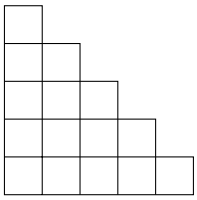


Figure 2: The size 5 triangular board in 2D.

The two-dimensional rook theory can be generalized to higher dimensions by letting rooks attack along hyperplanes [9]. Using this generalization, the two-

dimensional triangular boards were generalized to three and higher dimensions in two different ways, yielding families of triangular boards and Genocchi boards [1]. The numbers of ways to place k non-attacking rooks on the three-dimensional triangular boards result in the central factorial numbers [1]. In four and higher dimensions, k non-attacking rooks on the triangular boards generate the generalized central factorial numbers [3]. In this paper, we focus on the rook numbers of the Genocchi boards. Placing the maximum number of rooks on the three-dimensional Genocchi boards corresponds to the $(n+1)$st unsigned Genocchi numbers (of even index) $1,1,3,17,155,2073,38227,929569,\ldots$, and hence the name Genocchi boards [1]. In higher dimensions, the resulting sequence is the generalized Genocchi numbers [3].

Using the representation of the Genocchi numbers as the rook numbers of the three-dimensional Genocchi boards and the recursive nature of these boards, one obtains a combinatorial proof of a new recurrence relation for Genocchi numbers [2]. This recurrence relation is different than the Seidel generation or other triangles mentioned in [4, 6, 7]. This paper expands this recurrence relation of rook numbers of Genocchi boards in three dimensions to four and five dimensions, thus providing triangle generations of the third and fourth generalized Genocchi numbers. Along with these recurrence relations, we will describe how to visualize Genocchi boards in four and five dimensions.

In Section 2, we review the background material including the proof of the triangle generation of the Genocchi numbers in [2] obtained by using rook placements on Genocchi boards in three dimensions. We also include the combinatorial interpretation of Genocchi numbers and the recurrence relation using the rook representation. Then in Sections 3 and 5, we progress to the proof of the recurrence relations of generalized Genocchi numbers using Genocchi boards in four and five dimensions to obtain a triangle generation of the generalization of Genocchi numbers.

2 Two and Three Dimensions

We follow the notation and terminology of rook theory as described in [1]. In two dimensions, a *board* can be visualized as a collection of unit square cells chosen from a large size chess board. Each cell can be represented as (i, j) where i represents the row number (from top to bottom) and j represents the column number (from left to right). Rooks placed on these boards can attack along rows and columns. Therefore, a *non-attacking placement of rooks* on a board means that no two rooks lie in the same row or column.

The *two-dimensional triangular board of size n* consists of ordered pairs (i, j) with $j \leq i$ and $1 \leq i \leq n$, as illustrated in Fig. 2. It is known that the number of placements of k non-attacking rooks on the size n triangular board is equal to $S(n+1, n+1-k)$ where $S(n,k)$ are the Stirling numbers of the second kind. The proof of the relationship uses induction and the fact that the Stirling numbers satisfy the recurrence relation

$$S(n,k) = S(n-1,k-1) + kS(n-1,k)$$

with initial values $S(n,1) = 1$ and $S(n,n) = 1$. There are two cases of rook

placements on the triangular board: one where there is no rook on the bottom row and the other where there is a rook on the bottom row. If there is no rook on the bottom row, the k rooks are placed on the top $n-1$ rows and, by induction, there are $S(n,n-k)$ such placements. If there is a rook on the bottom row, then $k-1$ rooks must be placed on the top $n-1$ rows. Thus, there are $S(n,n-k+1)$ ways to place them. Once these rooks have been placed, the last rook will have $n-k+1$ available cells. Therefore, there are $(n-k+1)S(n,n-k+1)$ total choices in this case. Adding the two cases yields $S(n+1,n+1-k)$ using the Stirling number recurrence relation, and, by induction, the claim holds for all n. Using ideas similar to those in [9], the classical rook theory was generalized to three and higher dimensions in [1] by letting rooks attack along hyperplanes consisting of cells with one fixed coordinate. In three dimensions, the cells are represented as triples (i,j,k) and a rook on this cell will attack cells of the form $(i,*,*),(*,j,*)$, and $(*,*,k)$. We use *wall, slab* and *layer* to refer to the plane of cells with the first, second and third coordinate fixed, respectively. Hence, in three dimensions, rooks attack along walls, slabs and layers. Layers are numbered from top to bottom, and in each fixed layer, the numbering of rows and columns follows the same convention as in the two-dimensional case. In [1], the two-dimensional triangular board was also generalized to three dimensions to consist of the cells of the form (i,j,k) with $1 \leq i,j \leq k$ and $1 \leq k \leq n$. The size 4 triangular board in three dimensions is pictured in Fig. 3.

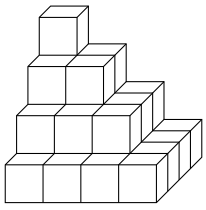


Figure 3: A triangular board in 3D.

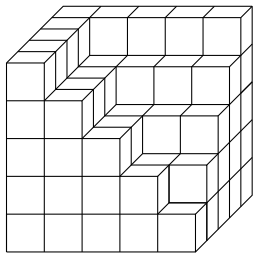


Figure 4: A Genocchi board in 3D.

Notice that Figures 3 and 4 together form a complete size 5 three-dimensional board. Using this relationship, it was shown in [1] that the number of non-attacking placements of n rooks on a size n board as shown in Fig. 4 corresponds to the $(n+1)$st unsigned Genocchi numbers of even index. Consequently, these boards were named the Genocchi boards, denoted $\Gamma_n^{(3)}$ for the dimension being 3. In terms of cell coordinates, $\Gamma_n^{(3)}$ consists of triples of the form (i,j,k), where $1 \leq i,j,k \leq n$ and $\min\{i,j\} \leq k$.

In Fig. 4 showing a size 5 Genocchi board, we can see a smaller size 4 Genocchi board if we remove the back wall and the left slab. Using this inclusion, we can find a recurrence relation between the numbers of rook placements on size n and $n-1$ Genocchi boards as described in the following theorem, proven in [2].

Theorem 1. *Let $r_k(\Gamma_n^{(3)})$ denote the number of ways to place k non-attacking rooks on $\Gamma_n^{(3)}$, the size n Genocchi board in three dimensions. Then $r_k(\Gamma_n^{(3)})$*

satisfies the recurrence relation

$$\begin{aligned} r_k(\Gamma_n^{(3)}) &= r_k(\Gamma_{n-1}^{(3)}) \\ &\quad + r_{k-1}(\Gamma_{n-1}^{(3)})(2(n-k)+1)(n-k+1) \\ &\quad + r_{k-2}(\Gamma_{n-1}^{(3)})(n-k+2)(n-k+1)^3, \end{aligned}$$

where $r_k(\Gamma_n^{(3)}) = 0$ *for* $k < 0$.
Below is an outline of the proof given in [2].

Proof. Note that $\Gamma_{n-1}^{(3)}$ can be realized inside $\Gamma_n^{(3)}$ by removing the outside wall (cells with $i = 1$) and slab (cells with $j = 1$). In terms of coordinates, we can represent $\Gamma_{n-1}^{(3)}$ as

$$\{(i,j,k) : 2 \le i,j,k \le n \text{ and } \min\{i,j\} \le k\}.$$

This correspondence works by adding 1 to each coordinate of the triples (i,j,k) in the usual representation of $\Gamma_{n-1}^{(3)}$, which is

$$\Gamma_{n-1}^{(3)} = \{(i,j,k) : 1 \le i,j,k \le n-1 \text{ and } \min\{i,j\} \le k\}.$$

The k rook placements on $\Gamma_n^{(3)}$ can then be split into three possible cases: 1) all k rooks are placed on $\Gamma_{n-1}^{(3)}$ inside $\Gamma_n^{(3)}$, or 2) $k-1$ rooks are placed on $\Gamma_{n-1}^{(3)}$ and one on the outside wall or slab, or 3) $k-2$ rooks are placed on $\Gamma_{n-1}^{(3)}$ and two on the outside wall and slab, except on their intersection. The terms in the recurrence relation correspond to each of these cases in order. The r_i terms, rook numbers, count the number of ways of placing i rooks on $\Gamma_{n-1}^{(3)}$. The factors multiplying the rook numbers are the number of choices for placing the remaining one or two rooks on the outside wall and/or slab. Once the majority of the rooks are placed on the inside $\Gamma_{n-1}^{(3)}$, corresponding rows and columns are deleted from the outside wall and slab, leaving us with a subset of cells that the remaining one or two rooks can be placed. The number of such cells are counted by the factors in front of r_i terms in the recurrence relation. □

From the proof, we notice that the recurrence relation holds due to the nature of the difference $\Gamma_n^{(3)} \backslash \Gamma_{n-1}^{(3)}$. Because this difference is composed of a full wall and a slab, where the rooks are placed on $\Gamma_{n-1}^{(3)}$ does not affect the choices for the remaining rooks. Also note that while this geometric proof works well in the three-dimensional case and provides us the intuition to notice the existence of a recurrence relation for the rook numbers of $\Gamma_n^{(3)}$ and how to prove this relation, it becomes challenging in higher dimensions. For this reason, we provide another proof of Theorem 1 via a symbolic approach.

Proof. (Symbolic approach) Again, we consider three cases: 1) all k rooks are placed on $\Gamma_{n-1}^{(3)}$ inside $\Gamma_n^{(3)}$, or 2) $k-1$ rooks are placed on $\Gamma_{n-1}^{(3)}$ and one on the outside wall or slab, or 3) $k-2$ rooks are placed on $\Gamma_{n-1}^{(3)}$, two on the outside wall and slab.
In the first case, there are $r_k(\Gamma_{n-1}^{(3)})$ ways to place the rooks.
In the second case, we first place the $k-1$ rooks on $\Gamma_{n-1}^{(3)}$ in $r_{k-1}(\Gamma_{n-1}^{(3)})$ ways. Then

the last rook can be placed on the outside wall, cells of the form $(1,*,*)$, or on the outside slab, $(*,1,*)$. However, the previous $k-1$ rooks make $k-1$ coordinates ineligible in each of the free positions, leaving us with $2(n-k+1)^2-(n-k+1)$ cells. We subtract $n-k+1$ cells of the form $(1,1,*)$ since they were double counted. This number is exactly the coefficient of $r_{k-1}(\Gamma^{(3)}_{n-1})$ in the recurrence relation we want to show.

In the third case when $k-2$ rooks are on $\Gamma^{(3)}_{n-1}$, the two remaining rooks are placed on the outside wall and slab. Since we cannot place both on the wall or both on the slab, we must place one on the wall and one on the slab. For the rook on the wall $(1,*,*)$, we have $(n-k+1)(n-k+2)$ choices. This is because each of the previous $k-2$ rooks eliminates one coordinate in each position. We also eliminate 1 from the second position since that would place the rook on the intersection of the wall and the slab. For the rook on the slab $(*,1,*)$, we have $(n-k+1)$ choices for each coordinate due to eliminations from the previous $k-1$ rooks. Therefore, we have a total of $(n-k+2)(n-k+1)^3$ choices for the last two rooks, and this is the coefficient of $r_{k-2}(\Gamma^{(3)}_{n-1})$ in the recurrence relation. The sum of all cases yields the recurrence relation. □

In the special case of $k=n$, this algebraic recurrence relation has a combinatorial interpretation. In [3], $r_n(\Gamma^{(3)}_n)$ are interpreted as pairs of permutations (π_1,π_2) of n where $\pi_1(i)\le i$ or $\pi_2(i)\le i$. This correspondence is obtained by generalizing the permutation correspondence described in the Introduction. In three dimensions, rooks are placed in cells (i,j,k). Suppose there are n rooks on $\Gamma^{(3)}_n$. Because of the way rooks attack, this means there is exactly one rook per layer. If we order the rook positions so that their last coordinate is increasing, then we obtain two permutations by reading the first coordinates and the second coordinates. In other words, $\pi_1(k)=i,\pi_2(k)=j$ if the rook is in cell (i,j,k).

Since the permutation interpretation requires a rook in each layer, we will use a modification of permutations to interpret $r_{n-2}(\Gamma^{(3)}_{n-1})$ combinatorially. We use partial permutations, specifically those with one hole, to account for the fact that one layer will not have a rook [5]. A *partial permutation of r with one hole* is a bijection between two size $r-1$ subsets of $\{1,2,\ldots,r\}$. It can be represented as a string of $r-1$ distinct numbers chosen from 1 to r and $\Diamond$, representing the hole. Using the same idea as when we placed n rooks on $\Gamma^{(3)}_n$, we can interpret $r_{n-2}(\Gamma^{(3)}_{n-1})$ as pairs (π_1,π_2) of partial permutations of $n-1$ with one hole in the same place so that $\pi_1(i)\le i$ or $\pi_2(i)\le i$ when i is not the hole. Then the algebraic recurrence relation translates into a recurrence relation for the number of pairs of permutations of n expressed in terms of the numbers of pairs of permutations of $n-1$ and pairs of partial permutations of $n-1$ with one hole in the same place, all pairs with the condition that $\pi_1(i)\le i$ or $\pi_2(i)\le i$.

Using similar ideas as in the symbolic proof of Theorem 1, we will obtain recurrence relations and their combinatorial interpretations for four and five dimensions in the next two sections.

3 Four Dimensions

The three-dimensional Genocchi boards were generalized to four and higher dimensions in [3]. In four and higher dimensions, rooks attack along hyperplanes, which consist of cells with one fixed coordinate. In four dimensions, we use *walls, slabs, layers* and the *time* to indicate the hyperplanes of cells with fixed 1^{st}, 2^{nd}, 3^{rd} and 4^{th} coordinates, respectively. The cells of $\Gamma_n^{(4)}$, the size n four-dimensional Genocchi board, are tuples of the form (i,j,k,ℓ), where $1 \le i,j,k,\ell \le n$ and $\min\{i,j,k\} \le \ell$. One way to visualize four dimensional Genocchi boards is to look at the board at a fixed time, i.e. with fixed last coordinate values. The size 3 Genocchi board in four dimensions is shown below, with each piece representing a different time value.

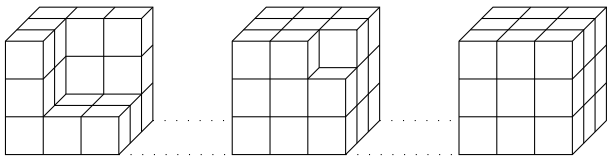


Figure 5: Size 3 Genocchi board in four dimensions.

Note that for ease of visualization, in four dimensions, we number layers (k coordinate) from bottom to top, and time (ℓ) from left to right. Thus the leftmost piece of the board above corresponds to cells (i,j,k,ℓ) with $\ell = 1$ and i or j or $k = 1$.

Similar to the recursive nature of the boards in two and three dimensions, $\Gamma_n^{(4)}$ includes the board $\Gamma_{n-1}^{(4)}$, identified as

$$\Gamma_{n-1}^{(4)} = \{(i,j,k,\ell) : 2 \le i,j,k,\ell \le n \text{ and } \min\{i,j,k\} \le \ell\}.$$

We remove the cells with coordinates $i = 1$, $j = 1$, or $k = 1$ from the larger Genocchi board to obtain the smaller board. In Fig. 5 showing $\Gamma_3^{(4)}$, this corresponds to removing all the outside walls in the back (corresponding to $i = 1$), outside slabs on the left (corresponding to $j = 1$) and outside layers at the bottom (corresponding to $k = 1$). Thus, we can visualize $\Gamma_2^{(4)}$ as the size 2 cube at the front top right corner of the rightmost size 3 cube and the size 2 cube with one cell missing at the same place of the middle size 3 cube.

More generally, the difference $\Gamma_n^{(4)} \backslash \Gamma_{n-1}^{(4)}$ consists of one outside wall corresponding to cells with $i = 1$, one outside slab for cells with $j = 1$ and one outside layer corresponding to cells with $k = 1$. The cells with coordinates $(1,1,1,*)$ belong to the intersection of the outside wall, slab and layer, which we call the ***triple intersection***. Any pair of wall, slab, or layer share n cells of the form $(1,1,*,*)$, or $(1,*,1,*)$, or $(*,1,1,*)$, which we call ***pair intersections***. When placing more than one rook on the difference $\Gamma_n^{(4)} \backslash \Gamma_{n-1}^{(4)}$, we must be careful about the intersections.

The relationship between $\Gamma_n^{(4)}$ and $\Gamma_{n-1}^{(4)}$ allows us to derive a recurrence relation for the rook numbers, as demonstrated by the following theorem.

Theorem 2. *Let $r_k(\Gamma_n^{(4)})$ denote the number of ways to place k non-attacking rooks on the size n four dimensional Genocchi board. Then $r_k(\Gamma_n^{(4)})$ satisfies the*

recurrence relation

$$\begin{aligned} r_k(\Gamma_n^{(4)}) &= r_k(\Gamma_{n-1}^{(4)}) \\ &\quad + r_{k-1}(\Gamma_{n-1}^{(4)})(n-k+1)(3(n-k+1)^2-3(n-k+1)+1) \\ &\quad + r_{k-2}(\Gamma_{n-1}^{(4)})(n-k+2)3(n-k+1)^5 \\ &\quad + r_{k-3}(\Gamma_{n-1}^{(4)})(n-k+3)(n-k+2)^4(n-k+1)^4, \end{aligned}$$

where $r_k(\Gamma_n^{(4)}) = 0$ *for* $k < 0$ *and* $r_0(\Gamma_n^{(4)}) = 1$.

Proof. The number of ways to place k non-attacking rooks on $\Gamma_n^{(4)}$ can be partitioned into 4 cases: 1) all k rooks are placed on $\Gamma_{n-1}^{(4)}$ realized inside $\Gamma_n^{(4)}$, 2) $k-1$ rooks are placed on $\Gamma_{n-1}^{(4)}$, 3) $k-2$ rooks are placed on $\Gamma_{n-1}^{(4)}$, and 4) $k-3$ rooks are placed on $\Gamma_{n-1}^{(4)}$.

In the first case, there are $r_k(\Gamma_{n-1}^{(4)})$ ways to place the rooks.

In the second case, there are $r_{k-1}(\Gamma_{n-1}^{(4)})$ possible placements for $k-1$ rooks inside $\Gamma_{n-1}^{(4)}$. Once $k-1$ rooks are placed on $\Gamma_{n-1}^{(4)}$, there is one rook to place on the outside wall, slab, or layer. Each has $(n-k+1)^3$ cells eligible due to the previous placement of $k-1$ rooks on $\Gamma_{n-1}^{(4)}$ disqualifying $k-1$ of the n available coordinates in two free positions, leading to a total of $3(n-k+1)^3$ options. However, this approach does not count the intersections properly. Therefore, we use the *Inclusion-Exclusion Principle* to obtain the result. Altogether, we have $r_{k-1}(\Gamma_{n-1}^{(4)})(n-k+1)(3(n-k+1)^2-3(n-k+1)+1)$ ways to place k rooks on $\Gamma_n^{(4)}$ where $k-1$ rooks lie on $\Gamma_{n-1}^{(4)}$.

In the third case, there are $r_{k-2}(\Gamma_{n-1}^{(4)})$ ways to place $k-2$ rooks on $\Gamma_{n-1}^{(4)}$. Note that in this case we cannot place either of the final two rooks on the triple intersection since a rook on the triple intersection disqualifies all the remaining cells on the outside wall, slab, and layer for the second rook. However, the pair intersections can be used and we consider this case separately. We have two subcases: 3.1) both rooks on outside wall, slab, or layer excluding the pair intersections, and 3.2) one on a pair intersection and the other on the non-attacked wall, slab, or layer.

For 3.1), we first choose which two of the wall, slab, and layer to use for the final two rooks, which can be done in 3 ways. Let us suppose we chose wall and slab to use. On the wall, $(1,*,*,*)$, we have $(n-k+2)$ choices for the last coordinate due to the previous $k-2$ rook. We then have $(n-k+1)$ choices for each of the middle two coordinates since we cannot use 1 and the options eliminated by the previous $k-2$ rooks. After placing this rook, for the last rook, we can use the slab cells $(*,1,*,*)$. For these, we have $(n-k+1)^2(n-k)$ choices. For this subcase, we obtain a total of $3(n-k+2)(n-k+1)^4(n-k)$ choices.

For 3.2), we first choose which pair intersection to use for one of the final two rooks, and there are 3 options. Once we decide on the pair intersection, only one of the wall, slab, or layer will be available to use for the second rook. Say we chose the pair intersection of the wall and slab, $(1,1,*,*)$. We have $(n-k+2)(n-k+1)$ cells available due to the previous $k-2$ rooks places and 1 not

being available in the third position. Then on the remaining layer $(*,*,1,*)$, we have $(n-k+1)^3$ options. Putting these together gives us $3(n-k+2)(n-k+1)^4$ options.

Since 3.1) and 3.2) are mutually exclusive cases we add them to obtain $r_{k-2}(\Gamma_{n-1}^{(4)})3(n-k+2)(n-k+1)^4\big((n-k)+1\big)$ ways to place k rooks on $\Gamma_n^{(4)}$ where $k-2$ rooks lie on $\Gamma_{n-1}^{(4)}$, which simplifies to $r_{k-2}(\Gamma_{n-1}^{(4)})(n-k+2)3(n-k+1)^5$.

In the fourth case, there are $r_{k-3}(\Gamma_{n-1}^{(4)})$ ways to place $k-3$ rooks on $\Gamma_{n-1}^{(4)}$. In this case we cannot place any of our final rooks on the triple intersection or on the pair intersections. Therefore, one rook is on the wall, one on the slab, and one on the layer. For the cells $(1,*,*,*)$, we have $(n-k+3)(n-k+2)^2$ options due to the previous $k-3$ rooks and 1 not being available in the second or third coordinates. For the cells $(*,1,*,*)$, we have $(n-k+2)^2(n-k+1)$ options due to the previous $k-2$ rooks and 1 not being available in the third coordinate. In this case, we do not reduce our options for the first coordinate since the rook on the wall used the 1 option. Finally, on the layer, we have $(n-k+1)^3$ options. Putting all of this together, we obtain $r_{k-3}(\Gamma_{n-1}^{(4)})(n-k+3)(n-k+2)^4(n-k+1)^4$ ways to place k rooks on $\Gamma_n^{(4)}$ where $k-3$ rooks lie on $\Gamma_{n-1}^{(4)}$.

Combining all these cases together, we obtain the recurrence relation in general. □

Using the above recurrence relation, we obtain Table 1 providing the number of ways to place k non-attacking rooks on $\Gamma_n^{(4)}$.

$n\backslash k$	0	1	2	3	4	5
1	1	**1**				
2	1	15	**7**			
3	1	72	505	**145**		
4	1	220	7525	33135	**6631**	
5	1	525	55445	1207260	3778201	**566641**

Table 1: The coefficients $r_k(\Gamma_n^{(4)})$.

Notice that as was shown before, the highlighted entries follow the sequence A064624 in [8]: *Generalization of the Genocchi numbers given by the Gandhi polynomials*

$$A(n+1,r) = r^3 \cdot A(n,r+1) - (r-1)^3 \cdot A(n,r); A(1,r) = r^3 - (r-1)^3.$$

Using the combinatorial interpretation of non-attacking rooks on a Genocchi board in the special case of $k=n$, the algebraic recurrence relation translates into a recurrence relation for the number of triples of permutations of n expressed in terms of the numbers of triples of permutations of $n-1$, triples of partial permutations of $n-1$ with one hole in the same place, and triples of partial permutations of $n-1$ with two holes in the same places. For all permutations and partial permutations considered, we require the condition that $\pi_1(i) \leq i$ or $\pi_2(i) \leq i$ or $\pi_3(i) \leq i$.

4 Five Dimensions

The three and four dimensional Genocchi boards can be generalized to five dimensions. In five dimensions, we use *walls, slabs, layers, time* and *hyper-time* to indicate the hyperplanes of cells with fixed 1^{st}, 2^{nd}, 3^{rd}, 4^{th} and 5^{th} coordinates respectively. The cells of $\Gamma_n^{(5)}$, the size n five-dimensional Genocchi board, are tuples of the form (i,j,k,l,h), where $1 \le i,j,k,l,h \le n$ and $\min\{i,j,k,l\} \le h$.
One way to visualize a 5-dimensional Genocchi board is to look at the board with the last two coordinates fixed. The size 3 Genocchi board in five dimensions is shown in Fig. 6.

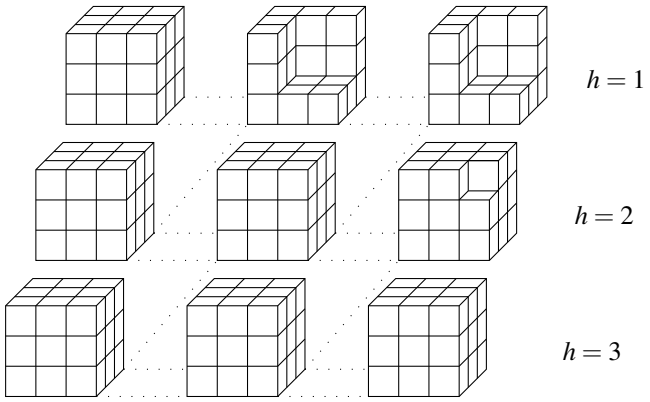


Figure 6: Size 3 Genocchi board in five dimensions.

We follow the same convention as in four dimensions within each row: rows are numbered from back to front, columns left to right, layers bottom to top, and time from left to right. The hyper-time increases from top to bottom. In the bottom row when $h = 3$, due to the definition, the coordinates i,j,k,ℓ can take any of the values 1, 2, 3. In the middle row, when $h = 2$, the coordinates i,j,k are allowed to be 1, 2, 3 when $\ell = 1,2$, but when $\ell = 3$, we need $\min\{i,j,k\} = 2$. Therefore, when $h = 2, \ell = 3$, the cell $(3,3,3,3,2)$ is missing. When $h = 1$, the coordinates i,j,k are allowed to be 1, 2, 3 when $\ell = 1$, and otherwise, at least one of i,j,k is 1.
Similar to the four-dimensional case, $\Gamma_n^{(5)}$ includes the smaller board $\Gamma_{n-1}^{(5)}$, which can be formally defined as

$$\Gamma_{n-1}^{(5)} = \{(i,j,k,l,h) : 2 \le i,j,k,l,h \le n \text{ and } \min\{i,j,k,l\} \le h\}.$$

The difference $\Gamma_n^{(5)} \backslash \Gamma_{n-1}^{(5)}$ consists of one outside slab with cells $(1,*,*,*,*)$ on $\Gamma_n^{(5)}$, one outside wall with cells $(*,1,*,*,*)$, one outside layer with cells $(*,*,1,*,*)$, and one outside time with cells $(*,*,*,1,*)$. The outside wall, slab, layer and time share the cells $(1,1,1,1,*)$, the ***quadruple intersection***. Intersections of three are called the ***triple intersections*** and the intersections of any pair of wall, slab, or layer are the ***pair intersections***.

The relationship between $\Gamma_n^{(5)}$ and $\Gamma_{n-1}^{(5)}$ allows us to derive a recurrence relation

for the rook numbers, as demonstrated by the following theorem.

Theorem 3. *Let $r_k(\Gamma_n^{(5)})$ denote the number of ways to place k non-attacking rooks on the size n five dimensional Genocchi board. Then $r_k(\Gamma_n^{(5)})$ satisfies the recurrence relation*

$$\begin{aligned} r_k(\Gamma_n^{(5)}) &= r_k(\Gamma_{n-1}^{(5)}) \\ &\quad + r_{k-1}(\Gamma_{n-1}^{(5)})(n-k+1)\sum_{i=1}^{4}(-1)^{i+1}\binom{4}{i}(n-k+1)^{4-i} \\ &\quad + r_{k-2}(\Gamma_{n-1}^{(5)})(n-k+2)(n-k+1)^5(6(n-k+1)^2+1) \\ &\quad + r_{k-3}(\Gamma_{n-1}^{(5)})(n-k+3)(n-k+2)^5(n-k+1)^5(4(n-k+1)+2) \\ &\quad + r_{k-4}(\Gamma_{n-1}^{(5)})(n-k+4)\prod_{i=1}^{3}(n-k+i)^5, \end{aligned}$$

where $r_k(\Gamma_n^{(5)}) = 0$ for $k < 0$ and $r_0(\Gamma_n^{(5)}) = 1$.

Proof. The number of ways to place k non-attacking rooks on the five dimensional Genocchi board size n can be separated into 5 possible cases: 1) all k rooks are placed on $\Gamma_{n-1}^{(5)}$ realized inside $\Gamma_n^{(5)}$, 2) $k-1$ rooks are placed on $\Gamma_{n-1}^{(5)}$, 3) $k-2$ rooks are placed on $\Gamma_{n-1}^{(5)}$, 4) $k-3$ rooks are placed on $\Gamma_{n-1}^{(5)}$, or 5) $k-4$ rooks are placed on $\Gamma_{n-1}^{(5)}$.

In the first case, there are $r_k(\Gamma_{n-1}^{(5)})$ ways to place the rooks.

In the second case, there are $r_{k-1}(\Gamma_{n-1}^{(5)})$ ways to place $k-1$ rooks on $\Gamma_{n-1}^{(5)}$. Once $k-1$ rooks are placed on $\Gamma_{n-1}^{(5)}$, there is one rook to place on the outside wall, slab, layer, or time. Similar to the 4D proof, we use the *Inclusion-Exclusion Principle* to count available cells to be $4(n-k+1)^4-6(n-k+1)^3+4(n-k+1)^2-(n-k+1)$. Thus, there are $\Gamma_n^{(5)}(n-k+1)(4(n-k+1)^3-6(n-k+1)^2+4(n-k+1)-1)$ ways to place k rooks on $\Gamma_n^{(5)}$ where $k-1$ rooks lie on $\Gamma_{n-1}^{(5)}$. We can also count this case by noting that the number of cells is found by excluding the cells where none of the first four coordinates are 1. There are $(n-k+1)^4-(n-k)^4$ options for the first four coordinates, and $(n-k+1)$ options for the last. This gives us a total of $r_{k-1}(\Gamma_{n-1}^{(5)}(n-k+1)((n-k+1)^4-(n-k)^4))$, which equals the expression given in the recurrence relation above.

In the third case, we place $k-2$ rooks on $\Gamma_{n-1}^{(5)}$ and two on the outside. We split how to place the last two rooks into four subcases: 3a) neither rook on an intersection, with only allowed cells being $(1,*,*,*,*),(*,1,*,*,*),(*,*,1,*,*)$, (*, *, *, 1, *) where no $*$ is 1; 3b) one rook on a pair intersection, one on no intersection; 3c) both rooks on pair intersections; and 3d) one rook on a triple intersection and one not. There are 6 possibilities for the subcase 3a) based on which two coordinates have 1's. After choosing which possibility we have, we have $n-k+2$ choices for the hyper-time coordinate of the second to last rook and every other coordinate besides the coordinate with 1 has $n-k+1$ choices. For the final rook, we have $n-k+1$ hyper-time choices. For the coordinate that corresponds to where 1 is in the second to last rook, we again have $n-k+1$ choices. For the two other remaining free coordinates, we have $n-k$ choices

each. Therefore, each possibility has $(n-k+2)(n-k+1)^5(n-k)^2$ options. There are 12 possibilities for the subcase 3b), each with $(n-k+2)(n-k+1)^5(n-k)$ options using a similar coordinate option counting. There are 3 possibilities for the subcase 3c), each with $(n-k+2)(n-k+1)^5$ options. Finally, 3d) has 4 possibilities and each has $(n-k+2)(n-k+1)^5$. Adding all the options gives $(n-k+2)(n-k+1)^5(6(n-k)^2+12(n-k)+7)$, which is the coefficient of $r_{k-2}(\Gamma^{(5)}_{n-1})$ given in the recurrence relation above.

In the fourth case, we split how to place the last three rooks into two subcases: 4a) none of the rooks is on an intersection; and 4b) one rook on a pair intersection including time intersections, and the other two on no intersections. There are 4 possibilities for the subcase 4a based on which coordinate is never 1. After choosing which possibility we have and ordering the rooks arbitrarily based on where the 1's are, we have $n-k+3$ choices for the hyper-time coordinate of the third to last rook and every other coordinate besides the coordinate with 1 has $n-k+2$ choices. For the middle rook, we have $n-k+2$ choices for hyper-time, $n-k+2$ choices for the coordinate that overlaps with the coordinate 1 of the third to last rook, and $n-k+1$ choices for each of the other two coordinates. For the final rook, we have $n-k+1$ hyper-time choices. For the two coordinates that correspond to where 1 was in the previous two rooks, we again have $n-k+1$ choices. For the last coordinate, we have $n-k$ choices. Therefore, each possibility has $(n-k+3)(n-k+2)^5(n-k+1)^5(n-k)$ options. There are 6 possibilities for the subcase 4b each with $(n-k+3)(n-k+2)^5(n-k+1)^5$ options using a similar coordinate option counting. Adding all the options gives $(n-k+3)(n-k+2)^5(n-k+1)^5(4(n-k)+6)$.

In the fifth case, there are $r_{k-4}(\Gamma^{(5)}_{n-1})$ ways to place $k-4$ rooks on $\Gamma^{(5)}_{n-1}$. Therefore, there are four rooks left that to place on the outside wall, slab, layer or time. Note in this case we cannot place any of these final rooks on an intersection. Therefore, all must be placed on cells of the form $(1,*,*,*,*),(*,1,*,*,*)$, (*, *, 1 , *, *), (*, *, *, 1, *) where no $*$ is 1 and there must be one rook on each of the outside wall, slab, layer, and the time. Counting options for each coordinate as in the previous cases, we find that there are $r_{k-4}(\Gamma^{(5)}_{n-1})(n-k+4)(n-k+3)^3(n-k+3)^2(n-k+2)^2(n-k+2)^2(n-k+1)(n-k+2)(n-k+1)^2(n-k+1)^2$ ways to place k rooks on $\Gamma^{(5)}_n$ where $k-4$ rooks lie on $\Gamma^{(5)}_{n-1}$. This can be simplified to $r_{k-4}(\Gamma^{(5)}_{n-1})(n-k+4)(n-k+3)^5(n-k+2)^5(n-k+1)^5$.

Putting all these cases together, we obtain the recurrence relation in general. □

Using the above recurrence relation, we obtain Table 2 that gives the number of ways to place k non-attacking rooks on the size n five dimension Genocchi board.

As expected, the highlighted entries follow the sequence A064625 in [8]: *Generalization of the Genocchi numbers given by the Gandhi polynomials*

$$A(n+1,r) = r^4 \cdot A(n,r+1) - (r-1)^4 \cdot A(n,r); A(1,r) = r^4 - (r-1)^4.$$

Similar to the four-dimensional case, when $k=n$, the algebraic recurrence relation above translates into a recurrence relation for the number of quadruples of permutations of n expressed in terms of the numbers of the quadruples of

$n\backslash k$	0	1	2	3	4	5
1	1	**1**				
2	1	31	**15**			
3	1	226	3345	**1025**		
4	1	926	100875	954815	**209135**	
5	1	2771	1245715	87547640	598789745	**100482849**

Table 2: The coefficients $r_k(\Gamma_n^{(5)})$.

permutations of $n-1$, quadruples of partial permutations of $n-1$ with one hole in the same place, quadruples of partial permutations of $n-1$ with two holes in the same place and quadruples of partial permutations of $n-1$ with three holes. In all cases, we require that $\pi_1(i) \leq i$ or $\pi_2(i) \leq i$ or $\pi_3(i) \leq i$ or $\pi_4(i) \leq i$.

5 Acknowledgements

Stephanie Loewen and Vasily Zadorozhnyy completed part of this work as their senior thesis. We would like thank the Department of Mathematics at Grand Valley State University for giving us the opportunity to pursue this project. We are also grateful to the anonymous reviewer whose feedback helped us improve the paper immensely.

Bibliography

[1] Feryal Alayont and Nicholas Krzywonos. Rook polynomials in three and higher dimensions. *Involve: A mathematics journal*, 6(1):35–52, 2013. doi:10.2140/involve.2013.6.35.

[2] Feryal Alayont and Nicholas Krzywonos. Another recurrence relation for Genocchi numbers. Unpublished manuscript, available at https://faculty.gvsu.edu/alayontf/notes/genocchi_triangle.pdf, 2016.

[3] Feryal Alayont, Rachel Moger-Reischer, and Ruth Swift. Combinatorial interpretations of generalized central factorial and Genocchi numbers. Unpublished manuscript, available at https://faculty.gvsu.edu/alayontf/notes/generalized_central_genocchi_numbers_preprint.pdf, 2012.

[4] Peter Bala. A triangle for calculating the unsigned Genocchi numbers (of the first kind) A110501. Unpublished manuscript, 2017. URL: https://oeis.org/A110501/a110501_1.pdf.

[5] Anders Claesson, Vít Jelínek, Eva Jelínková, and Sergey Kitaev. Pattern avoidance in partial permutations. *Electronic Journal of Combinatorics*, 18(1), 2011. URL: http://www.combinatorics.org/ojs/index.php/eljc/article/view/v18i1p25.

[6] Dominique Dumont and Arthur Randrianarivony. Dérangements et nombres de Genocchi. *Discrete Mathematics*, 132(1):37 – 49, 1994. URL: http://

www.sciencedirect.com/science/article/pii/0012365X94902305, doi:https://doi.org/10.1016/0012-365X(94)90230-5.

[7] Richard Ehrenborg and Einar Steingrímsson. Yet another triangle for the Genocchi numbers. *European Journal of Combinatorics*, 21(5):593 – 600, 2000. URL: http://www.sciencedirect.com/science/article/pii/S0195669899903700, doi:https://doi.org/10.1006/eujc.1999.0370.

[8] OEIS Foundation Inc. The On-Line Encyclopedia of Integer Sequences, 2024. Published electronically at http://oeis.org.

[9] Benjamin Zindle. Rook polynomials for chessboards of two and three dimensions. 2007. Master's thesis, available at https://scholarworks.rit.edu/cgi/viewcontent.cgi?article=1727&context=theses.

Ball State Undergraduate Mathematics Exchange, 18(1), Fall 2024, 65-76
https://openjournals.bsu.edu/mathexchange
DOI: 10.33043/ydyzb82y

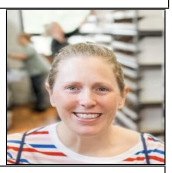
Pages 65 –76

Triangles and variance of the distance to the boundary

Alastair Fletcher , Katherine Fletcher, Joseph Wasiqi*

Alastair Fletcher is a Professor at Northern Illinois University and currently Director of Undergraduate Studies. He finds assisting students at all levels and of all backgrounds fulfilling, and puts up with the department making him wear Halloween costumes.

Katherine Fletcher holds a Master's in Mathematics, teaches at Elgin Community College and is Director of the MathLab. She enjoys helping students reach their potential.



Joseph Wasiqi is currently a Master's degree student at Northern Illinois University. He hopes to carry his mathematical knowledge into medicine and/or bioinformatic.

Abstract

We consider the variance of the distance to the boundary for planar triangles. Our main result is that if γ is a line segment joining a vertex of a triangle to a point on the opposite side, then the variance restricted to γ is a convex function.

1 Introduction

Let U be a compact convex set in $\mathbb{R}^2$. For $z=(x,y)$ contained in U and $\theta \in [0,2\pi)$, we define $d_z^U(\theta)$ to be the distance from z to ∂U in the direction θ. Then for $k \in \mathbb{N}$ we define

$$I_k^U(z) = \frac{1}{2\pi}\int_0^{2\pi} [d_z^U(\theta)]^k \, d\theta.$$

***Corresponding author**: afletcher@niu.edu

Finally, the variance from z to ∂U is defined by

$$\mathrm{var}_U(z) = I_2^U(z) - [I_1^U(z)]^2. \tag{1. A}$$

If the context is clear, we may drop the U from the above notation and just write $d_z(\theta), I_k(z)$ and so on.

The notion of variance of the distance to the boundary was used in Strawbridge et al [4] as the basis of a robust algorithm to determine whether individual cells in an embryo are interior or exterior cells based purely on knowing the location of the nuclei of the cells. In particular, it was shown in [4] that in the special case of the unit ball in $\mathbb{R}^3$, the variance of the distance to the boundary has a unique minimum at the center of the ball. In this paper, we initiate a mathematical study of the variance by focusing on the case of planar triangles. In Figures 1 and 2, we see a plot of the variance function inside a certain isosceles triangle.

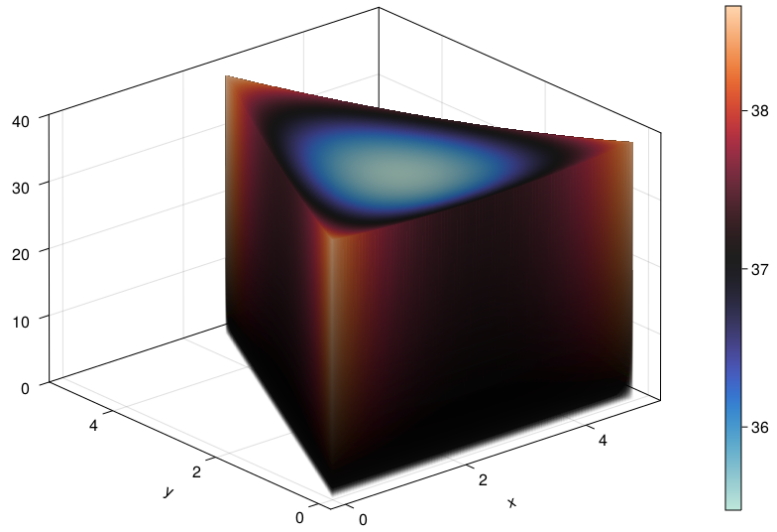


Figure 1: Surface plot generated with 80199 variance values with step size for x and y coordinates equal to 0.0125.

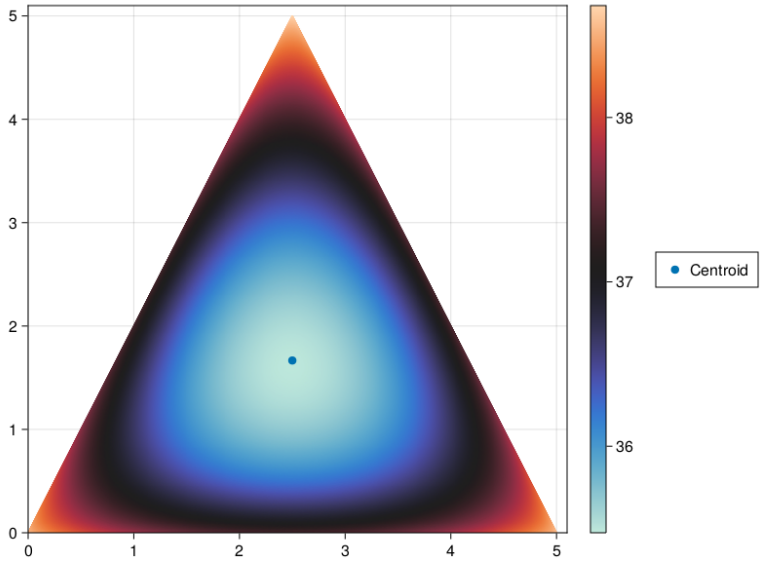


Figure 2: Flattened version of Figure 1 with the centroid of the triangle marked.

We start by giving some general properties of variance.

Theorem 1. *Let U be a compact convex set in $\mathbb{R}^2$.*

(i) If $f : \mathbb{R}^2 \to \mathbb{R}^2$ is an isometry, then

$$var_{f(U)}(f(z)) =_U (z).$$

(ii) If $f : \mathbb{R}^2 \to \mathbb{R}^2$ is a similarity with scaling factor $C > 0$, then

$$var_{f(U)}(f(z)) = C_U^2(z).$$

This result will allow us to pass from general triangles to triangles with vertices at $(0,0),(1,0)$ and (P,Q) with $Q > 0$. Additionally, this result also implies that if U has a certain symmetry, then the variance function will also have the analogous symmetry. For example, in an isosceles triangle, the variance is unchanged under a reflection about the line of symmetry. Next, we make the important observation that in two dimensions, $I_2(z)$ is in fact constant in z.

Theorem 2. *For any bounded convex domain $U \subset \mathbb{R}^2$ and any $z \in U$ we have*

$$I_2(z) = \frac{1}{\pi}\,\mathrm{Area}(U).$$

Our main result is as follows.

Theorem 3. *Let T be a triangle and let $\gamma : [0,1] \to T$ parameterize a straight line segment from one vertex of T to a point on the side opposite this vertex. Then $|_\gamma$ is a convex function.*

In particular, it will follow from Theorem 3 that on each line segment emanating from a vertex of T, there is a unique minimum of the variance. Unfortunately, this does not appear to be quite enough to guarantee that the variance is a convex function on the whole of T and thus that the variance has a unique minimum on T. If we could show that the variance is convex on any line segment

joining two points on the boundary of T, then this would be enough. However, a typical such line segment will divide T into a triangle and a quadrilateral. Computations analogous to those in this paper for quadrilaterals get messy very quickly. We leave these questions open as a topic for future work, and just note that Figures 2 and 4 below give evidence of a unique minimum.
The paper is organized as follows. In Section 2 we prove the preliminary results Theorem 1 and Theorem 2, while in Section 3 we prove our main result, Theorem 3. We make some concluding remarks and acknowledgements in Section 4.

2 Preliminaries

In this section, we first show that the variance is well behaved under isometries and similarities.

Proof of Theorem 1. Suppose first that f is a translation, rotation or reflection. Then it is clear that I_1 and I_2 are unchanged by applying f and so the same is true of the variance. If f is a scaling by factor $d > 0$, then we have

$$d_{f(z)}^{f(U)}(\theta) = C \cdot d_z^U(\theta),$$

which proves (b) in this case.
For the general case, an isometry is a composition of translations, rotations and reflections, which proves (a). For (b), a similarity is a composition of isometries and scalings as above, which completes the proof. □

Next we prove that I_2 is constant for bounded convex planar domains.

Proof of Theorem 2. For this proof, we will work in complex coordinates. Suppose first that ∂U is smooth and that we parameterize ∂U by fixing $z \in \overline{U}$ and setting $w = \gamma(\theta) = d_z(\theta)e^{i\theta}$ for $0 \le \theta \le 2\pi$. Then γ is a smooth function and

$$\gamma'(\theta) = (d_z'(\theta) + id_z(\theta))e^{i\theta}.$$

As $d_z(\theta)$ is real-valued, we have

$$\begin{aligned}\int_\gamma \overline{w}\, dw &= \int_0^{2\pi} \overline{(d_z(\theta)e^{i\theta})}(d_z'(\theta) + id_z(\theta))e^{i\theta})\, d\theta \\ &= \int_0^{2\pi} (d_z(\theta)d_z'(\theta) + i(d_z(\theta))^2)\, d\theta.\end{aligned}$$

Integrating by parts, we have

$$\begin{aligned}\int_0^{2\pi} d_z(\theta)d_z'(\theta)\, d\theta &= \left[d_z(\theta)^2\right]_{\theta=0}^{2\pi} - \int_0^{2\pi} d_z(\theta)d_z'(\theta)\, d\theta \\ &= 0 - \int_0^{2\pi} d_z(\theta)d_z'(\theta)\, d\theta.\end{aligned}$$

Thus

$$\int_0^{2\pi} d_z(\theta)d_z'(\theta)\, d\theta = 0$$

and we have

$$\int_\gamma \overline{w}\, dw = i\int_0^{2\pi} d_z(\theta)^2\, d\theta = 2\pi i I_2(z).$$

By Green's Theorem, $\int_\gamma \overline{w}\,dw$ is nothing other than $2i$ multiplied by the area enclosed by γ, from which Theorem 2 follows in the smooth case.

For the general case, fix $z \in U$ and suppose that $\varphi : \mathbb{D} \to U$ is a conformal bijection from the unit disk onto U given by the Riemann Mapping Theorem. For $0 < r < 1$, set S_r to be the circle centred at the origin of radius r, D_r to be the open disk centred at the origin of radius r, and set $U_r = \varphi(D_r)$. We may assume that r_0 is chosen close enough to 1 that $z \in U_r$ for $r > r_0$. By a theorem of Study [5], see also for example [3], it follows that U_r is a convex domain for all $0 < r < 1$. Moreover, as $\varphi(S_r)$ is a smooth Jordan curve it follows by the argument above that for $r_0 < r < 1$,

$$I_2^{U_r}(z) = \frac{1}{\pi}\,\mathrm{Area}(U_r).$$

As ∂U is a Jordan curve, by Carathéodory's Theorem it follows that φ extends to a homeomorphism from $\overline{\mathbb{D}}$ onto $\overline{U}$. Thus the extended function is uniformly continuous, as it is continuous on a compact subset of $\mathbb{C}$. It follows that $d_z^{U_r}(\theta) \to d_z^U(\theta)$ as $r \to 1$ uniformly in θ. A standard integral estimate then shows that $I_2^{U_r}(z) \to I_2^U(z)$ as $r \to 1$. Putting this all together, we have

$$I_2^U(z) = \lim_{r\to 1} I_2^{U_r}(z) = \lim_{r\to 1} \frac{1}{\pi}\,\mathrm{Area}(U_r) = \frac{1}{\pi}\,\mathrm{Area}(U),$$

which completes the proof. □

3 Convexity of variance along rays

By Theorem 2, I_2 is a constant. As T is a non-degenerate triangle, $I_1 : T \to \mathbb{R}$ is a strictly positive function. It follows from (1. A) that to show the variance restricted to γ is a convex function, it is enough to show that I_1 restricted to γ is a concave function.

Our approach to proving Theorem 3 relies on the following special case where γ is an edge of the triangle.

Theorem 4. *Let T be a triangle with vertices $(0,0),(1,0)$ and (P,Q) where $P \in \mathbb{R}$ and $Q > 0$. Define $h : [0,1] \to \mathbb{R}$ by $h(x) = I_1(x,0)$. Then h is a concave function.*

With this in hand, the proof of Theorem 3 is as follows.

Proof of Theorem 3. Let T be a triangle with vertices u, v and w. Without loss of generality, suppose that $\gamma : [0,1] \to T$ parameterizes a line segment from the vertex u to a point u' on the side of T joining v to w. Then γ divides T into two triangles T_1 and T_2 with vertices u, u', v and u, u', w respectively.

Let f be a similarity map with scaling factor C which maps the line segment $[u,u']$ onto the line segment L joining $(0,0)$ to $(1,0)$. Then by the proof of Theorem 1,

$$\begin{aligned} I_1^T(z) &= \frac{1}{2\pi}\int_0^{2\pi} d_z^T(\theta)\,d\theta \\ &= \frac{1}{2\pi C}\int_0^{\pi} d_{f(z)}^{f(T)}(\theta)\,d\theta + \frac{1}{2\pi C}\int_\pi^{2\pi} d_{f(z)}^{f(T)}(\theta)\,d\theta \\ &= \frac{1}{C} I_1^{T_1}(f(z)) + \frac{1}{C} I_1^{T_2}(f(z)). \end{aligned}$$

By Theorem 4, $I_1^{T_1}|_L$ is concave and, by applying a reflection and again using Theorem 4, $I_1^{T_2}|_L$ is also concave. As the sum of two concave functions is concave, it follows that $I_1^T|_\gamma$ is a concave function. By the observation at the start of this section, it follows that the variance restricted to γ is a convex function, which completes the proof. □

The rest of this section is devoted to the proof of Theorem 4.

3.1 Computing I_1 In the following lemma, we explicitly compute the integral in the formula for I_1 from a vertex of a triangle.

Lemma 5. *Let T be a triangle with vertices u,v,w and side lengths opposite each vertex a,b,c respectively. Let p be the perpendicular distance from u to the line passing through v and w. Then*

$$I_1(u) = \frac{p}{2\pi}\left[\ln(a+b+c) - \ln(-a+b+c)\right]. \tag{3. B}$$

Proof. Let ϕ_u, ϕ_v and ϕ_w be the angles subtended at the vertices u,v and w respectively. We set $\theta = 0$ to correspond the the direction from u towards v and $\theta = \pi - \phi_v - \phi_w$ to be the direction from u towards w. Then it follows from elementary trigonometry that

$$\cos(\pi/2 - \phi_v - \theta) = \frac{p}{d_u(\theta)}.$$

Using the identity $\cos(\pi/2 - t) = \sin t$, we therefore have

$$d_u(\theta) = \frac{p}{\sin(\phi_v + \theta)}.$$

Now, by definition we have

$$I_1(u) = \frac{1}{2\pi}\int_0^{\pi-\phi_v-\phi_w} d_u(\theta)\,d\theta = \frac{1}{2\pi}\int_0^{\pi-\phi_v-\phi_w} p\csc(\phi_v+\theta)\,d\theta.$$

Making the change of variables $\alpha = \phi_v + \theta$, we obtain

$$\begin{aligned}
I_1(u) &= \frac{1}{2\pi}\int_{\phi_v}^{\pi-\phi_w} p\csc(\alpha)\,d\alpha \\
&= \frac{1}{2\pi}\left[\ln\left|\frac{1-\cos\alpha}{\sin\alpha}\right|\right]_{\alpha=\phi_v}^{\pi-\phi_w} \\
&= \frac{1}{2\pi}\left[\ln\left|\tan\left(\frac{\alpha}{2}\right)\right|\right]_{\alpha=\phi_v}^{\pi-\phi_w} \\
&= \frac{p}{2\pi}\left(\ln\left|\tan\left(\frac{\pi-\phi_w}{2}\right)\right| - \ln\left|\tan\left(\frac{\phi_v}{2}\right)\right|\right] \\
&= \frac{p}{2\pi}\ln\left|\cot(\phi_v/2)\cot(\phi_w/2)\right|.
\end{aligned}$$

Here we have used the half-angle formula for tan and the identity $\tan(\pi/2-t) = \cot t$. Let $s = (a+b+c)/2$ be the semi-perimeter of T. Then by the law of cotangents we have

$$\frac{\cot(\phi_v/2)}{s-b} = \frac{\cot(\phi_w/2)}{s-c} = \frac{1}{r},$$

where

$$r = \left(\frac{(s-a)(s-b)(s-c)}{s}\right)^{1/2}.$$

Combining this with the expression for $I_1(u)$ above, we find that

$$\begin{aligned} I_1(u) &= \frac{p}{2\pi} \ln \left| \left(\frac{s-b}{r} \right) \left(\frac{s-c}{r} \right) \right| \\ &= \frac{p}{2\pi} (\ln s - \ln(s-a)) \end{aligned}$$

which gives (3. B) once the $\ln(1/2)$ terms are cancelled. □

3.2 Splitting T into two sub-triangles For $x \in (0,1)$, we decompose T into two triangles A, B where A has vertices $(0,0)$, $(x,0)$ and (P,Q) and B has vertices $(x,0)$, $(1,0)$ and (P,Q). Let $J_1 \in [0,2\pi)$ be the interval corresponding to angles giving triangle A viewed from x, and let J_2 be the corresponding interval for B. For $\theta \in [0,2\pi) \setminus (J_1 \cup J_2)$ we have $d_x(\theta) = 0$ and so

$$h(x) = \frac{1}{2\pi} \left(\int_{J_1} d_x(\theta)\, d\theta + \int_{J_2} d_x(\theta)\, d\theta \right).$$

For $i \in \{1,2\}$, let

$$f_i(x) = \int_{J_i} d_x(\theta)\, d\theta.$$

Here we are using a slight abuse of notation to write d_x instead of $d_{(x,0)}$.

Lemma 6. *We have*

$$f_1(x) = \frac{Qx}{\sqrt{P^2+Q^2}} \ln \left(\frac{\sqrt{P^2+Q^2} + x + \sqrt{(P-x)^2+Q^2}}{-\sqrt{P^2+Q^2} + x + \sqrt{(P-x)^2+Q^2}} \right)$$

and

$$f_2(x) = \frac{Q(1-x)}{\sqrt{(P-1)^2+Q^2}} \ln \left(\frac{\sqrt{(P-1)^2+Q^2} + (1-x) + \sqrt{(P-x)^2+Q^2}}{-\sqrt{(P-1)^2+Q^2} + (1-x) + \sqrt{(P-x)^2+Q^2}} \right).$$

Proof. An elementary computation shows that the perpendicular distance from x to the line passing through $(0,0)$ and (P,Q) is Qx and the perpendicular distance from x to the line passing through $(1,0)$ and (P,Q) is $Q(1-x)$. The lemma then follows by applying Lemma 5 twice. □

Lemma 7. *We have*

$$f_1'(x) = \frac{Q}{\sqrt{P^2+Q^2}} \ln \left(\frac{\sqrt{P^2+Q^2} + x + \sqrt{(P-x)^2+Q^2}}{-\sqrt{P^2+Q^2} + x + \sqrt{(P-x)^2+Q^2}} \right) - \frac{Q}{((P-x)^2+Q^2)^{1/2}} \tag{3. C}$$

and

$$f_1''(x) = Q((P-x)^2+Q^2)^{-3/2} \left(P - \frac{P^2+Q^2}{x} \right). \tag{3. D}$$

Proof. The derivative of $\ln(\sqrt{P^2+Q^2} + x + \sqrt{(P-x)^2+Q^2})$ is

$$\frac{1 - (P-x)/\sqrt{(P-x)^2+Q^2}}{\sqrt{P^2+Q^2} + x + \sqrt{(P-x)^2+Q^2}}$$

and the derivative of $\ln(-\sqrt{P^2+Q^2} + x + \sqrt{(P-x)^2+Q^2})$ is

$$\frac{1 - (P-x)/\sqrt{(P-x)^2+Q^2}}{-\sqrt{P^2+Q^2} + x + \sqrt{(P-x)^2+Q^2}}.$$

Thus

$$f_1'(x) = \frac{Q}{\sqrt{P^2+Q^2}} \ln\left(\frac{\sqrt{P^2+Q^2}+x+\sqrt{(P-x)^2+Q^2}}{-\sqrt{P^2+Q^2}+x+\sqrt{(P-x)^2+Q^2}}\right) + \frac{Qx}{\sqrt{P^2+Q^2}}\left(1-\frac{(P-x)}{\sqrt{(P-x)^2+Q^2}}\right)\left[\frac{1}{\sqrt{P^2+Q^2}+x+\sqrt{(P-x)^2+Q^2}} - \frac{1}{-\sqrt{P^2+Q^2}+x+\sqrt{(P-x)^2+Q^2}}\right].$$

By putting the terms over a common denominator, the term in the square brackets is

$$\frac{-2\sqrt{P^2+Q^2}}{-(P^2+Q^2)+(x+\sqrt{(P-x)^2+Q^2})^2} = \frac{-2\sqrt{P^2+Q^2}}{2x^2-2xP+2x\sqrt{(P-x)^2+Q^2}} = \frac{-\sqrt{P^2+Q^2}}{x(x-P+\sqrt{(P-x)^2+Q^2})}.$$

We also note that

$$1-\frac{P-x}{\sqrt{(P-x)^2+Q^2}} = \frac{\sqrt{(P-x)^2+Q^2}+x-P}{\sqrt{(P-x)^2+Q^2}}$$

from which it follows that

$$f_1'(x) = \frac{Q}{\sqrt{P^2+Q^2}} \ln\left(\frac{\sqrt{P^2+Q^2}+x+\sqrt{(P-x)^2+Q^2}}{-\sqrt{P^2+Q^2}+x+\sqrt{(P-x)^2+Q^2}}\right) + \frac{Qx}{\sqrt{P^2+Q^2}}\left(\frac{\sqrt{(P-x)^2+Q^2}+x-P}{\sqrt{(P-x)^2+Q^2}}\right)\left[\frac{-\sqrt{P^2+Q^2}}{x(x-P+\sqrt{(P-x)^2+Q^2})}\right].$$

By cancelling terms here, we obtain (3. C). For the second derivative, we can use the work above to differentiate the first term in (3. C) and the quotient rule for the second term to obtain

$$\begin{aligned} f_1''(x) &= -\frac{Q}{x\sqrt{(P-x)^2+Q^2}} - \frac{Q(P-x)}{((P-x)^2+Q^2)^{3/2}} \\ &= Q\left(\frac{-(P-x)^2-Q^2-Px+x^2}{x((P-x)^2+Q^2)^{3/2}}\right) \\ &= -\frac{Q(P^2+Q^2-Px)}{x((P-x)^2+Q^2)^{3/2}} \end{aligned}$$

which is (3. D). □

Next we give the analogous result for f_2.

Lemma 8. *We have*

$$f_2'(x) = -\frac{Q}{\sqrt{(P-1)^2+Q^2}} \ln\left(\frac{\sqrt{(P-1)^2+Q^2}+(1-x)+\sqrt{(P-x)^2+Q^2}}{-\sqrt{(P-1)^2+Q^2}+(1-x)+\sqrt{(P-x)^2+Q^2}}\right) + \frac{Q}{((P-x)^2+Q^2)^{1/2}} \qquad (3.\ \text{E})$$

and

$$f_2''(x) = Q((P-x)^2+Q^2)^{-3/2}\left(1-P-\frac{(P-1)^2+Q^2}{1-x}\right). \qquad (3.\ \mathrm{F})$$

Proof. The theme of the computation is similar to that of the proof of Lemma 7 and so we omit the details of the derivation of (3. E). Differentiating (3. E) yields

$$\begin{aligned} f_2''(x) &= -\frac{Q}{(1-x)\sqrt{(P-x)^2+Q^2}} + \frac{Q(P-x)}{((P-x)^2+Q^2)^{3/2}} \\ &= Q((P-x)^2+Q^2)^{-3/2}\left(-\left(\frac{(P-x)^2+Q^2}{1-x}\right)+P-x\right) \\ &= Q((P-x)^2+Q^2)^{-3/2}\left(-\left(\frac{(P-1+1-x)^2+Q^2}{1-x}\right)+P-1+1-x\right) \\ &= Q((P-x)^2+Q^2)^{-3/2}\left(1-P-\frac{(P-1)^2+Q^2}{1-x}\right) \end{aligned}$$

which is (3. F). □

By combining (3. D) and (3. F), we see that

$$\begin{aligned} h''(x) &= \frac{Q((P-x)^2+Q^2)^{-3/2}}{2\pi}\left(P-\frac{P^2+Q^2}{x}+1-P-\frac{(P-1)^2+Q^2}{1-x}\right) \\ &= \frac{Q((P-x)^2+Q^2)^{-3/2}}{2\pi}\left(1-\frac{P^2+Q^2}{x}-\frac{(P-1)^2+Q^2}{1-x}\right). \qquad (3.\ \mathrm{G}) \end{aligned}$$

For $x \in (0,1)$, set

$$j(x) = 1-\frac{P^2+Q^2}{x}-\frac{(P-1)^2+Q^2}{1-x}. \qquad (3.\ \mathrm{H})$$

Then from (3. G), it is evident that h is a concave function if and only if $j(x) < 0$ for $x \in (0,1)$.

3.3 Showing that j is always negative

Lemma 9. *We have* $\lim_{x\to 0^+} j(x) = -\infty$, $\lim_{x\to 1^-} j(x) = -\infty$, *and* j *has a unique critical point on* $(0,1)$ *at*

$$x_0 = \frac{\sqrt{P^2+Q^2}}{\sqrt{P^2+Q^2}+\sqrt{(P-1)^2+Q^2}} \qquad (3.\ \mathrm{I})$$

with critical value

$$j(x_0) = 1-\left(P^2+Q^2+2\sqrt{P^2+Q^2}\sqrt{(P-1)^2+Q^2}+(P-1)^2+Q^2\right). \qquad (3.\ \mathrm{J})$$

Moreover, for any $P \in \mathbb{R}$ *and* $Q > 0$, *we have* $j(x_0) < 0$.

Proof. The claims about the limits follow immediately from (3. H). Next,

$$j'(x) = \frac{P^2+Q^2}{x^2}-\frac{(P-1)^2+Q^2}{(1-x)^2}.$$

Solving $j'(x) = 0$ yields

$$(1-x)^2(P^2+Q^2) = x^2((P-1)^2+Q^2).$$

As $x \in (0,1)$, we may take positive square roots of both sides to obtain

$$(1-x)\sqrt{P^2+Q^2} = x\sqrt{(P-1)^2+Q^2}$$

and solving for x yields (3. I). Plugging this value into the formula for j and some elementary algebra then yields (3. J).
For the final claim, set $J(P,Q)$ to be the right hand side of (3. J) as a function in terms of P and Q. Then

$$\begin{aligned} J(P,0) &= 1 - \left(P^2 + 2\sqrt{P^2}\sqrt{(P-1)^2} + (P-1)^2\right) \\ &= -2P(P-1) - 2|P(P-1)|. \end{aligned}$$

If $P(P-1) \leq 0$, then $J(P,0) = 0$, otherwise $J(P,0) < 0$. In either case, we have $J(P,0) \leq 0$. Now, for $Q > 0$, the partial derivative of J with respect to Q is

$$J_Q = -4Q - \frac{2Q\sqrt{P^2+Q^2}}{\sqrt{(P-1)^2+Q^2}} - \frac{2Q\sqrt{(P-1)^2+Q^2}}{\sqrt{P^2+Q^2}} < 0.$$

We conclude that for any fixed $P \in \mathbb{R}$, $\lim_{Q\to 0^+} J(P,Q) \leq 0$ and $J_Q < 0$, from which it follows that $J(P,Q) < 0$ for all $P \in \mathbb{R}$ and $Q > 0$. This gives the final claim. □

From Lemma 9, we conclude that the unique critical point of j on $(0,1)$ is a maximum, and that maximum value is negative. Thus, by (3. G), h is a concave function on $[0,1]$ which completes the proof of Theorem 4.

4 Concluding remarks

The topic of this paper was a Research Rookies project undertaken by the third named author at NIU during the 2022-2023 academic year with the first named author as advisor. The authors would like to thank the organizers of the Research Rookies program at NIU for stimulating this work and thank the anonymous referee for comments which improved the exposition in the paper.
The images were created using Julia and Makie, see [1] and [2] respectively. We finish with another visual example of the variance. Note the different scales on the axes in Figures 3 and 4. This triangle is long and thin, but for ease of visual representation, the scales were chosen this way. Comparing Figures 1 and 2 with 3 and 4, we see that the variance achieves much higher values for long, thin triangles.

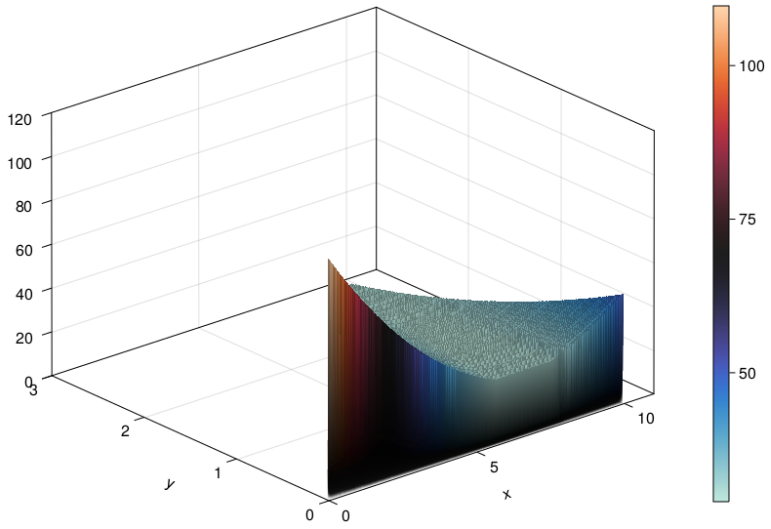


Figure 3: Surface plot generated with 99620 variance values with step size for x and y coordinates equal to 0.01.

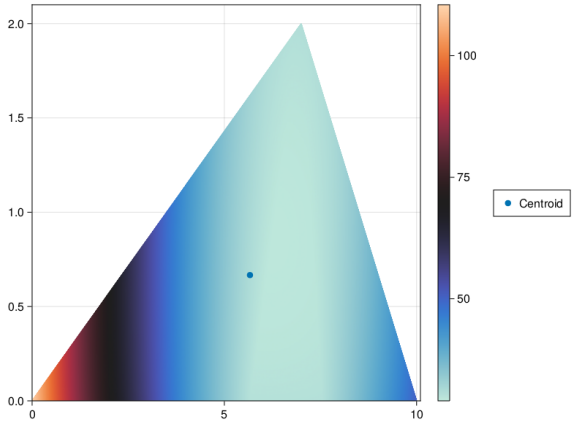


Figure 4: Flattened version of Figure 3 with the centroid marked.

Bibliography

[1] J. Bezanson, A. Edelman, S. Karpinski, V. B. Shah, Julia: A Fresh Approach to Numerical Computing, *SIAM Review*, **59**, no.1 (2017), 65-98.

[2] S. Danisch, J. Krumbiegel, Makie.jl: Flexible high-performance data visualization for Julia, *Journal of Open Source Software*, 6(65), 3349 (2021).

[3] P. Duren, *Univalent functions*, Grundlehren der mathematischen Wissenschaften **259**, Springer-Verlag, New York, 1983,

[4] S. E. Strawbridge, A. Kurowski, E. Corujo-Simon, A. N. Fletcher, J. Nichols and A. G. Fletcher, insideOutside: an accessible algorithm for classifying interior and exterior points, with applications in embryology, *Biol. Open*, **12**, no. 9 (2023): bio060055.

[5] E. Study, *Vorlesungen über ausgew"ahlte Gegenst"ande der Geometrie (German edition)*, Cor- nell University Library, 1911.

Ball State Undergraduate Mathematics Exchange, 18(1), Fall 2024, 77-87
https://openjournals.bsu.edu/mathexchange
DOI: 10.33043/r6a225bc

Pages 77 – 87

Continued Fractions, a-Fibonacci numbers, and the middle b-noise

*Aakash Gurung, Cheng-Han Pan**

Aakash Gurung is a junior at the University of Alabama, pursuing a dual major in Mathematics and Physics. This project was undertaken during his time at Juniata College. He is currently exploring diverse areas of mathematics, driven by a long-term aspiration to pursue a career in mathematical research.

Cheng-Han Pan received his Ph.D. in mathematics from West Virginia University. He served as a visiting assistant professor and faculty advisor for Juniata Problem Solving Group at Juniata College before joining Western New England University. His research interests focus on the foundations of real analysis, especially exploring paradoxical examples of functions and sets.

Abstract

Problem 1186 in The College Mathematics Journal asked for a closed form expression of the continued fraction $[1,1,\ldots,1,3,1,1,\ldots,1]$, and reappeared as Problem 1385 in the PME journal. In this paper, we present a generalization to $[a,a,\ldots,a,b,a,a,\ldots,a]$ with a-Fibonacci numbers and discuss how much the middle b-noise would impact the continued fractions with all a's.

1 Introduction

This paper is based on our participation in the Juniata Problem Solving Group in Fall 2022 and our further discoveries in one problem that we solved which can be traced to Problem 1186, proposed by Gregory Dresden and ZhenShu Luan, in The College Mathematics Journal [5]. The problem states: *Find a closed-form expression for the continued fraction* $[1,1,\ldots,1,3,1,1,\ldots,1]$*, which*

***Corresponding author**: cp621920@wne.edu

has n ones before and after, the middle three.

A simple finite continued fraction is an expression of the form

$$a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{\ddots + \cfrac{1}{a_{n-1} + \cfrac{1}{a_n}}}}},$$

where $a_0 \in \mathbb{Z}$ and $a_i \in \mathbb{Z}^+$ for $i \geq 1$. It is often denoted by $[a_0, a_1, a_2, \ldots, a_{n-1}, a_n]$. For example,

$$[1,3,1] = 1 + \frac{1}{3+\frac{1}{1}} = \frac{5}{4} \qquad \text{and} \qquad [1,1,3,1,1] = 1 + \frac{1}{1+\frac{1}{3+\frac{1}{1+\frac{1}{1}}}} = \frac{16}{9}.$$

Although a solution by Walther Janous was quickly published in [6], a closed-form expression can be derived differently. Since the Juniata Problem Solving Group did not exist until Fall 2022, the problem that later caught our attention was reproposed by Hongwei Chen as PME Problem 1385 in [3].

In PME Problem 1385, after a nice introduction to continued fractions and Fibonacci numbers, the proposers stated that they found a neat closed-form expression for $[1,1,\ldots,1,3,1,1,\ldots,1]$ and challenged the readers to prove it. In particular, readers are invited to show that

$$[\underbrace{1,1,\ldots,1}_{n\text{-times}},3,\underbrace{1,1,\ldots,1}_{n\text{-times}}] = \frac{F_{n+4}F_{n+1}}{F_{n+2}^2},$$

where F_n represents the nth Fibonacci number listed as follows. (fn. 1)

·····	F_{-1}	F_0	F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F_9	·····
·····	1	0	1	1	2	3	5	8	13	21	34	·····

The Juniata Problem Solving Group submitted a solution to PME Problem 1385, (fn. 2) which was published in [4].

In the next section, we present a generalization from $[\underbrace{1,\ldots,1}_{n\text{-times}},3,\underbrace{1,\ldots,1}_{n\text{-times}}]$ to $[\underbrace{a,\ldots,a}_{n\text{-times}},b,\underbrace{a,\ldots,a}_{n\text{-times}}]$ with a-Fibonacci numbers. In Section 3, we show how much the middle b changes the values of such a continued fraction when $b \neq a$ (the surprising answer is: not very much at all). Lastly in Section 4, we will come back and briefly address Juniata Problem Solving Group's approach in [4] and Walther Janous' approach in [3].

1. Although Fibonacci numbers usually start with F_0 and F_1, note that one can extend the indexes of Fibonacci numbers to negative integers by using $F_n = F_{n+2} - F_{n+1}$. We will particularly need $F_{-1} = F_1 - F_0 = 1$ in some future formulas. On the other hand, Fibonacci numbers can be defined for negative indexes by $F_{-n} \coloneqq (-1)^{n+1}F_n$ as well.

2. Members included Emmanuel Adutwum, Aakash Gurung, Saea Eun Lee, and Johnathan Park at the time.

2 Generalization to $[a,\ldots,a,b,a,\ldots,a]$

Why just 1 and 3? An immediate generalization would be $[\underbrace{a,\ldots,a}_{n\text{-times}},b,\underbrace{a,\ldots,a}_{n\text{-times}}]$ for $a,b\in\mathbb{Z}^+$. In this section, we will need to work with a-Fibonacci numbers. To increase the readability of the calculations in proofs throughout the paper, we will always use the shorthand notation $\mathbf{F}_n$ to represent the nth a-Fibonacci number.

For $a\in\mathbb{Z}^+$, the nth a-Fibonacci number, denoted by $\mathbf{F}_n$, is the nth term in the a-Fibonacci sequence defined recursively as

$$\mathbf{F}_n := \begin{cases} \mathbf{F}_{n+2}-a\mathbf{F}_{n+1} & \text{for } n\le -1\\ 0 & \text{for } n=0\\ 1 & \text{for } n=1\\ a\mathbf{F}_{n-1}+\mathbf{F}_{n-2} & \text{for } n\ge 2 \end{cases}.$$

From the definition above, we see that a-Fibonacci numbers clearly generalize the original Fibonacci numbers. (fn. 3) Throughout the paper, we will repeatedly use the particular recursive relation $\mathbf{F}_n = a\mathbf{F}_{n-1}+\mathbf{F}_{n-2}$ to simplify our work. To acquaint readers with the recursive relation, we establish the validity of Cassini's identity, Honsberger's identity, and d'Ocagne's identity for a-Fibonacci numbers. These identities serve as essential tools in the proofs of Theorem 5, Corollary 7, and Remark 8.

Proposition 1. *(i) Cassini's identity: If $n\in\mathbb{N}$ and $a\in\mathbb{Z}^+$, then*

$$\boldsymbol{F}_{n+1}\boldsymbol{F}_{n-1}=(\boldsymbol{F}_n)^2+(-1)^n.$$

(ii) Honsberger's identity: If $m,n\in\mathbb{N}$ and $a\in\mathbb{Z}^+$, then

$$\boldsymbol{F}_{m+n}=\boldsymbol{F}_m\boldsymbol{F}_{n+1}+\boldsymbol{F}_{m-1}\boldsymbol{F}_n.$$

(iii) d'Ocagne's identity: If $m,n\in\mathbb{N}$ and $a\in\mathbb{Z}^+$, then

$$(-1)^n\boldsymbol{F}_{m-n}=\boldsymbol{F}_m\boldsymbol{F}_{n+1}-\boldsymbol{F}_{m+1}\boldsymbol{F}_n.$$

Proof. To show (i), we run induction on $n\in\mathbb{N}$. It is easy to verify the basic step. $n=0$ implies

$$\mathbf{F}_1\mathbf{F}_{-1}=1\cdot 1=1=0^2+(-1)^0=(\mathbf{F}_0)^2+(-1)^0.$$

Suppose $\mathbf{F}_{k+1}\mathbf{F}_{k-1}=(\mathbf{F}_k)^2+(-1)^k$ for some $k\ge 0$. We want to show that

$$\mathbf{F}_{k+2}\mathbf{F}_k=(\mathbf{F}_{k+1})^2+(-1)^{k+1}.$$

Working on the left hand side, we break $\mathbf{F}_{k+2}$ into $a\mathbf{F}_{k+1}+\mathbf{F}_k$ and get

$$\mathbf{F}_{k+2}\mathbf{F}_k=(a\mathbf{F}_{k+1}+\mathbf{F}_k)\mathbf{F}_k=a\mathbf{F}_{k+1}\mathbf{F}_k+(\mathbf{F}_k)^2.$$

By the inductive hypothesis we replace $(\mathbf{F}_k)^2$ with $\mathbf{F}_{k+1}\mathbf{F}_{k-1}-(-1)^k$ and have

$$\begin{aligned} a\mathbf{F}_{k+1}\mathbf{F}_k+(\mathbf{F}_k)^2 &= a\mathbf{F}_{k+1}\mathbf{F}_k+\mathbf{F}_{k+1}\mathbf{F}_{k-1}-(-1)^k\\ &= \mathbf{F}_{k+1}(a\mathbf{F}_k+\mathbf{F}_{k-1})-(-1)^k\\ &= \mathbf{F}_{k+1}(\mathbf{F}_{k+1})-(-1)^k=(\mathbf{F}_{k+1})^2+(-1)^{k+1} \end{aligned}$$

3. Let $F_n(x)$ denote the nth Fibonacci Polynomial, which is another generalization of Fibonacci numbers. We want to mention that the nth a-Fibonacci number $\mathbf{F}_n$ can also be obtained by evaluating $F_n(a)$. This well-known fact can be easily proved by a strong induction.

as needed.
To see (ii), one would again run induction on $n \in \mathbb{N}$ while m is arbitrarily fixed. First, one shows the identity holds for $\mathbf{F}_{m+0}$ and $\mathbf{F}_{m+1}$. Assuming the identity holds for $\mathbf{F}_{m+k-1}$ and $\mathbf{F}_{m+k}$, the goal is to show that the identity also holds for $\mathbf{F}_{m+k+1} = a\mathbf{F}_{m+k} + \mathbf{F}_{m+k-1}$. We omit the straightforward but tedious details.
(iii) can be proved in the same way as described above. □

Lemma 2. *If $n \in \mathbb{N}$ and $a \in \mathbb{Z}^+$, then*

(i) $[\underbrace{a,a,\ldots,a}_{n\text{-times}},x] = \dfrac{x\boldsymbol{F}_{n+1}+\boldsymbol{F}_n}{x\boldsymbol{F}_n+\boldsymbol{F}_{n-1}}.$

(ii) $[x,\underbrace{a,a,\ldots,a}_{n\text{-times}}] = x + \dfrac{\boldsymbol{F}_n}{\boldsymbol{F}_{n+1}}.$

Proof. For $n = 0$, $[x] = x = \frac{x\cdot 1+0}{x\cdot 0+1} = \frac{x\mathbf{F}_1+\mathbf{F}_0}{x\mathbf{F}_0+\mathbf{F}_{-1}}$ is clear. Suppose $[\underbrace{a,\ldots,a}_{k\text{-times}},x] = \frac{x\mathbf{F}_{k+1}+\mathbf{F}_k}{x\mathbf{F}_k+\mathbf{F}_{k-1}}$ for some $k \geq 0$. We want to show that $[\underbrace{a,a,\ldots,a}_{(k+1)\text{-times}},x] = \frac{x\mathbf{F}_{k+2}+\mathbf{F}_{k+1}}{x\mathbf{F}_{k+1}+\mathbf{F}_k}$.
Indeed,

$$\begin{aligned}[\underbrace{a,a,\ldots,a}_{(k+1)\text{-times}},x] &= a + \frac{1}{[\underbrace{a,\ldots,a}_{k\text{-times}},x]} = a + \frac{x\mathbf{F}_k+\mathbf{F}_{k-1}}{x\mathbf{F}_{k+1}+\mathbf{F}_k} \\ &= \frac{ax\mathbf{F}_{k+1}+a\mathbf{F}_k+x\mathbf{F}_k+\mathbf{F}_{k-1}}{x\mathbf{F}_{k+1}+\mathbf{F}_k} \\ &= \frac{x(a\mathbf{F}_{k+1}+\mathbf{F}_k)+a\mathbf{F}_k+\mathbf{F}_{k-1}}{x\mathbf{F}_{k+1}+\mathbf{F}_k} = \frac{x\mathbf{F}_{k+2}+\mathbf{F}_{k+1}}{x\mathbf{F}_{k+1}+\mathbf{F}_k}\end{aligned}$$

completing the proof of (i). To see (ii), we simply apply (i) with $x = a$ and obtain

$$[x,\underbrace{a,a,\ldots,a}_{n\text{-times}}] = x + \frac{1}{[\underbrace{a,a,\ldots,a}_{n\text{-times}}]} = x + \frac{a\mathbf{F}_{n-1}+\mathbf{F}_{n-2}}{a\mathbf{F}_n+\mathbf{F}_{n-1}} = x + \frac{\mathbf{F}_n}{\mathbf{F}_{n+1}}.$$

□

Moreover, the identity $[\underbrace{a,\ldots,a}_{n\text{-times}}] = \frac{\mathbf{F}_{n+1}}{\mathbf{F}_n}$ is well-known and immediate from Lemma 2 with $x = a$. We are now able to derive a closed-form expression for $[\underbrace{a,\ldots,a}_{n\text{-times}},b,\underbrace{a,\ldots,a}_{n\text{-times}}]$. Note that the expression in Lemma 3 may not be in its most elegant form, but it serves as the common ground for Theorem 5 and Remark 9.

Lemma 3. *If $n \in \mathbb{N}$ and $a,b \in \mathbb{Z}^+$, then*

$$[\underbrace{a,\ldots,a}_{n\text{-times}},b,\underbrace{a,\ldots,a}_{n\text{-times}}] = \frac{b(\boldsymbol{F}_{n+1})^2+2\boldsymbol{F}_{n+1}\boldsymbol{F}_n}{b\boldsymbol{F}_{n+1}\boldsymbol{F}_n+(\boldsymbol{F}_n)^2+\boldsymbol{F}_{n+1}\boldsymbol{F}_{n-1}}.$$

Proof. We first write $[\underbrace{a,\ldots,a}_{n\text{-times}},b,\underbrace{a,\ldots,a}_{n\text{-times}}]$ as $[\underbrace{a,\ldots,a}_{n\text{-times}},[b,\underbrace{a,\ldots,a}_{n\text{-times}}]]$ so that Lemma

2 becomes applicable and gives

$$[\underbrace{a,\ldots,a}_{n\text{-times}},[b,\underbrace{a,\ldots,a}_{n\text{-times}}]] = [\underbrace{a,\ldots,a}_{n\text{-times}},b+\frac{\mathbf{F}_n}{\mathbf{F}_{n+1}}] = \frac{\left(b+\frac{\mathbf{F}_n}{\mathbf{F}_{n+1}}\right)\mathbf{F}_{n+1}+\mathbf{F}_n}{\left(b+\frac{\mathbf{F}_n}{\mathbf{F}_{n+1}}\right)\mathbf{F}_n+\mathbf{F}_{n-1}}.$$

After simplifying the complex fraction, we obtain

$$\begin{aligned}[\underbrace{a,\ldots,a}_{n\text{-times}},b,\underbrace{a,\ldots,a}_{n\text{-times}}] &= \frac{(b\mathbf{F}_{n+1}+\mathbf{F}_n)\mathbf{F}_{n+1}+\mathbf{F}_{n+1}\mathbf{F}_n}{(b\mathbf{F}_{n+1}+\mathbf{F}_n)\mathbf{F}_n+\mathbf{F}_{n+1}\mathbf{F}_{n-1}}\\ &= \frac{b(\mathbf{F}_{n+1})^2+2\mathbf{F}_{n+1}\mathbf{F}_n}{b\mathbf{F}_{n+1}\mathbf{F}_n+(\mathbf{F}_n)^2+\mathbf{F}_{n+1}\mathbf{F}_{n-1}}\end{aligned}$$

as needed. □

3 The middle b-noise in convergents toward the ath metallic ratio

What do we mean by the middle b-noise? Comparing

$$[\underbrace{a,\ldots,a}_{n\text{-times}},b,\underbrace{a,\ldots,a}_{n\text{-times}}] \text{ and } [\underbrace{a,\ldots,a}_{n\text{-times}},a,\underbrace{a,\ldots,a}_{n\text{-times}}],$$

we wonder how much difference $b \neq a$ would cause. In this section, we will formulate the middle b-noise in different ways.

Remark 4. *It is well-known that the the infinite continued fraction* $[1,1,1,\ldots]$ *converges to the golden ratio. Moreover,* $[2,2,2,\ldots]$ *and* $[3,3,3,\ldots]$ *also converge, and those values are called the silver and the bronze ratios respectively. In fact, the infinite continued fraction* $[a,a,a,\ldots]$ *converges for each* $a \in \mathbb{Z}^+$*. In particular,*

$$[a,a,a,\cdots] = \frac{a+\sqrt{a^2+4}}{2},$$

and its value is called the a*th metallic ratio.*

Note that $[\underbrace{a,\ldots,a}_{n\text{-times}}] = \frac{\mathbf{F}_{n+1}}{\mathbf{F}_n}$ is called the nth convergent of $[a,a,a,\cdots]$. In Theorem 5 and Corollary 6, we first view

$$[\underbrace{a,\ldots,a}_{n\text{-times}},b,\underbrace{a,\ldots,a}_{n\text{-times}}] = [\underbrace{a,\ldots,a}_{n\text{-times}},[b,\underbrace{a,\ldots,a}_{n\text{-times}}]]$$

as $[\underbrace{a,\ldots,a}_{n\text{-times}}]$ attached with a $[b,\underbrace{a,\ldots,a}_{n\text{-times}}]$ tail part, and formulate how much the tail part alters the value of $[\underbrace{a,\ldots,a}_{n\text{-times}}]$.

Theorem 5. *If* $n \in \mathbb{N}$ *and* $a,b \in \mathbb{Z}^+$*, then*

$$[\underbrace{a,\ldots,a}_{n\text{-times}},b,\underbrace{a,\ldots,a}_{n\text{-times}}] = \frac{\boldsymbol{F}_{n+1}}{\boldsymbol{F}_n+\frac{(-1)^n}{b\boldsymbol{F}_{n+1}+2\boldsymbol{F}_n}}$$

Proof. To further rearrange the formula in Lemma 3, we simply apply Cassini's

identity to replace $\mathbf{F}_{n+1}\mathbf{F}_{n-1}$ with $(\mathbf{F}_n)^2+(-1)^n$ and obtain

$$[\underbrace{a,\ldots,a}_{n\text{-times}},b,\underbrace{a,\ldots,a}_{n\text{-times}}] = \frac{b(\mathbf{F}_{n+1})^2+2\mathbf{F}_{n+1}\mathbf{F}_n}{b\mathbf{F}_{n+1}\mathbf{F}_n+(\mathbf{F}_n)^2+(\mathbf{F}_n)^2+(-1)^n}$$
$$= \frac{\mathbf{F}_{n+1}(b\mathbf{F}_{n+1}+2\mathbf{F}_n)}{\mathbf{F}_n(b\mathbf{F}_{n+1}+2\mathbf{F}_n)+(-1)^n} = \frac{\mathbf{F}_{n+1}}{\mathbf{F}_n+\frac{(-1)^n}{b\mathbf{F}_{n+1}+2\mathbf{F}_n}}$$

as needed. □

From Theorem 5, It is easy to see that $[\underbrace{a,\ldots,a}_{n\text{-times}},b,\underbrace{a,\ldots,a}_{n\text{-times}}]$ still converges to the ath metallic ratio as n goes to infinity. Indeed, the $[b,\underbrace{a,\ldots,a}_{n\text{-times}}]$ tail part does not change the convergents in the long run since $\frac{(-1)^n}{b\mathbf{F}_{n+1}+2\mathbf{F}_n}$ tends to vanish as n goes to infinity.

How about the difference in value between $[\underbrace{a,\ldots,a}_{n\text{-times}},[b,\underbrace{a,\ldots,a}_{n\text{-times}}]]$ and $[\underbrace{a,\ldots,a}_{n\text{-times}}]$ by subtraction? Instead of running into the unpleasant algebra of arranging a common denominator for

$$[\underbrace{a,\ldots,a}_{n\text{-times}},b,\underbrace{a,\ldots,a}_{n\text{-times}}]-[\underbrace{a,\ldots,a}_{n\text{-times}}] = \frac{\mathbf{F}_{n+1}}{\mathbf{F}_n+\frac{(-1)^n}{b\mathbf{F}_{n+1}+2\mathbf{F}_n}}-\frac{\mathbf{F}_{n+1}}{\mathbf{F}_n},$$

we take the difference of their reciprocals since they already have the same denominator.

Corollary 6. *If $n\in\mathbb{N}$ and $a,b\in\mathbb{Z}^+$, then*

$$[0,\underbrace{a,\ldots,a}_{n\text{-times}},b,\underbrace{a,\ldots,a}_{n\text{-times}}]-[0,\underbrace{a,\ldots,a}_{n\text{-times}}] = \frac{(-1)^n}{\boldsymbol{F}_{n+1}(b\boldsymbol{F}_{n+1}+2\boldsymbol{F}_n)}.$$

Proof.

$$[0,\underbrace{a,\ldots,a}_{n\text{-times}},b,\underbrace{a,\ldots,a}_{n\text{-times}}]-[0,\underbrace{a,\ldots,a}_{n\text{-times}}] = \frac{1}{[\underbrace{a,\ldots,a}_{n\text{-times}},b,\underbrace{a,\ldots,a}_{n\text{-times}}]}-\frac{1}{[\underbrace{a,\ldots,a}_{n\text{-times}}]}$$
$$= \frac{\mathbf{F}_n+\frac{(-1)^n}{b\mathbf{F}_{n+1}+2\mathbf{F}_n}}{\mathbf{F}_{n+1}}-\frac{\mathbf{F}_n}{\mathbf{F}_{n+1}} = \frac{(-1)^n}{\mathbf{F}_{n+1}(b\mathbf{F}_{n+1}+2\mathbf{F}_n)}.$$

□

Secondly, we view $[\underbrace{a,\ldots,a}_{n\text{-times}},b,\underbrace{a,\ldots,a}_{n\text{-times}}]$ as $[\underbrace{a,\ldots,a}_{n\text{-times}},a,\underbrace{a,\ldots,a}_{n\text{-times}}]$ disturbed by the middle b-noise. Instead of comparing their actual difference, we again compare the difference of their reciprocals.

Corollary 7. *If $n\in\mathbb{N}$ and $a,b\in\mathbb{Z}^+$, then*

$$[0,\underbrace{a,\ldots,a}_{n\text{-times}},b,\underbrace{a,\ldots,a}_{n\text{-times}}]-[0,\underbrace{a,\ldots\ldots,a}_{(2n+1)\text{-times}}] = \frac{(-1)^n(a-b)}{(a\boldsymbol{F}_{n+1}+2\boldsymbol{F}_n)(b\boldsymbol{F}_{n+1}+2\boldsymbol{F}_n)}.$$

Proof. We prepare algebra rearrangements of $\frac{1}{[\underbrace{a,\ldots,a}_{n\text{-times}},b,\underbrace{a,\ldots,a}_{n\text{-times}}]}$ and $\frac{1}{[\underbrace{a,\ldots\ldots,a}_{(2n+1)\text{-times}}]}$

separately before taking the subtraction. First, from Theorem 5 we have

$$\frac{1}{[\underbrace{a,\ldots,a}_{n\text{-times}},b,\underbrace{a,\ldots,a}_{n\text{-times}}]}=\frac{\mathbf{F}_n+\frac{(-1)^n}{b\mathbf{F}_{n+1}+2\mathbf{F}_n}}{\mathbf{F}_{n+1}}=\frac{(\mathbf{F}_n+\frac{(-1)^n}{b\mathbf{F}_{n+1}+2\mathbf{F}_n})(\mathbf{F}_{n+2}+\mathbf{F}_n)}{\mathbf{F}_{n+1}(\mathbf{F}_{n+2}+\mathbf{F}_n)}$$

$$=\frac{[\mathbf{F}_{n+2}\mathbf{F}_n+(\mathbf{F}_n)^2]+\frac{(-1)^n(\mathbf{F}_{n+2}+\mathbf{F}_n)}{b\mathbf{F}_{n+1}+2\mathbf{F}_n}}{\mathbf{F}_{n+1}(\mathbf{F}_{n+2}+\mathbf{F}_n)}.$$

Secondly, we want to rewrite $\frac{1}{[\underbrace{a,\ldots\ldots,a}_{(2n+1)\text{-times}}]}=\frac{\mathbf{F}_{2n+1}}{\mathbf{F}_{2n+2}}$. Applying Honsberger's identity on both $\mathbf{F}_{2n+1}$ and $\mathbf{F}_{2n+2}$ and using Cassini's identity to replace $(\mathbf{F}_{n+1})^2$ with $\mathbf{F}_{n+2}\mathbf{F}_n-(-1)^{n+1}$ give

$$\frac{\mathbf{F}_{2n+1}}{\mathbf{F}_{2n+2}}=\frac{\mathbf{F}_{(n+1)+n}}{\mathbf{F}_{(n+1)+(n+1)}}=\frac{(\mathbf{F}_{n+1})^2+(\mathbf{F}_n)^2}{\mathbf{F}_{n+1}\mathbf{F}_{n+2}+\mathbf{F}_n\mathbf{F}_{n+1}}=\frac{[\mathbf{F}_{n+2}\mathbf{F}_n+(\mathbf{F}_n)^2]-(-1)^{n+1}}{\mathbf{F}_{n+1}(\mathbf{F}_{n+2}+\mathbf{F}_n)}.$$

Notice that $[\mathbf{F}_{n+2}\mathbf{F}_n+(\mathbf{F}_n)^2]$ in their numerators directly cancel each other in the subtraction. We continue simplify their difference and obtain

$$\frac{\frac{(-1)^n(\mathbf{F}_{n+2}+\mathbf{F}_n)}{b\mathbf{F}_{n+1}+2\mathbf{F}_n}+(-1)^{n+1}}{\mathbf{F}_{n+1}(\mathbf{F}_{n+2}+\mathbf{F}_n)}=\frac{(-1)^n\left(\frac{a\mathbf{F}_{n+1}+2\mathbf{F}_n}{b\mathbf{F}_{n+1}+2\mathbf{F}_n}-1\right)}{\mathbf{F}_{n+1}(a\mathbf{F}_{n+1}+2\mathbf{F}_n)}=\frac{(-1)^n\frac{(a-b)\mathbf{F}_{n+1}}{b\mathbf{F}_{n+1}+2\mathbf{F}_n}}{\mathbf{F}_{n+1}(a\mathbf{F}_{n+1}+2\mathbf{F}_n)}$$

$$=\frac{(-1)^n(a-b)}{(a\mathbf{F}_{n+1}+2\mathbf{F}_n)(b\mathbf{F}_{n+1}+2\mathbf{F}_n)}$$

as needed. □

Putting Corollary 6 and Corollary 7 together, not only did the similarity between the formulas interest us but also the role of the $a-b$ term in Corollary 7. If we consider $b\to\infty$, then we are no longer restricted to their reciprocals. We include our observations in the next remark.

Remark 8. *The difference between* $[\underbrace{a,\ldots,a}_{n\text{-times}},b,\underbrace{a,\ldots,a}_{n\text{-times}}]$ *and* $[\underbrace{a,\ldots,a}_{n\text{-times}}]$ *is independent of the distance between* a *and* b*. In fact, as* b *goes to infinity, the difference goes to* 0*. In particular, we have*

$$\lim_{b\to\infty}[\underbrace{a,\ldots,a}_{n\text{-times}},b,\underbrace{a,\ldots,a}_{n\text{-times}}]-[\underbrace{a,\ldots,a}_{n\text{-times}}]=\lim_{b\to\infty}[\underbrace{a,\ldots,a}_{n\text{-times}}+\frac{1}{b+X}]-[\underbrace{a,\ldots,a}_{n\text{-times}}]$$

$$=[\underbrace{a,\ldots,a}_{n\text{-times}}]-[\underbrace{a,\ldots,a}_{n\text{-times}}]=0,$$

where $X=[0,\underbrace{a,\ldots,a}_{n\text{-times}}]$*. Moreover, the limit is the same regardless the value of* X*.*

On the contrary, the difference between $[\underbrace{a,\ldots,a}_{n\text{-times}},b,\underbrace{a,\ldots,a}_{n\text{-times}}]$ *and* $[\underbrace{a,\ldots\ldots,a}_{(2n+1)\text{-times}}]$ *is obviously dependent on the distance between* a *and* b*. Moreover, we have*

(i) $\lim_{b\to\infty}[\underbrace{a,\ldots,a}_{n\text{-times}},b,\underbrace{a,\ldots,a}_{n\text{-times}}]-[\underbrace{a,\ldots\ldots,a}_{(2n+1)\text{-times}}]=\frac{(-1)^n\boldsymbol{F}_{n+1}}{\boldsymbol{F}_{2n+1}\boldsymbol{F}_n}.$

(ii) $\lim_{b\to\infty}[0,\underbrace{a,\ldots,a}_{n\text{-}times},b,\underbrace{a,\ldots,a}_{n\text{-}times}]-[0,\underbrace{a,\ldots\ldots,a}_{(2n+1)\text{-}times}]=\dfrac{(-1)^{n+1}}{\boldsymbol{F}_{2n+2}}.$

Proof. Apply the limits. We are indeed computing $\frac{\mathbf{F}_{n+1}}{\mathbf{F}_n}-\frac{\mathbf{F}_{2n+2}}{\mathbf{F}_{2n+1}}$ and $\frac{\mathbf{F}_n}{\mathbf{F}_{n+1}}-\frac{\mathbf{F}_{2n+1}}{\mathbf{F}_{2n+2}}$ for (i) and (ii) respectively. From d'Ocagne's identity with $m=2n+1$, $\mathbf{F}_{2n+1}\mathbf{F}_{n+1}-\mathbf{F}_{2n+2}\mathbf{F}_n=(-1)^n\mathbf{F}_{n+1}$. We immediately have

$$\frac{\mathbf{F}_{n+1}}{\mathbf{F}_n}-\frac{\mathbf{F}_{2n+2}}{\mathbf{F}_{2n+1}}=\frac{\mathbf{F}_{2n+1}\mathbf{F}_{n+1}-\mathbf{F}_{2n+2}\mathbf{F}_n}{\mathbf{F}_{2n+1}\mathbf{F}_n}=\frac{(-1)^n\mathbf{F}_{n+1}}{\mathbf{F}_{2n+1}\mathbf{F}_n}$$

and

$$\frac{\mathbf{F}_n}{\mathbf{F}_{n+1}}-\frac{\mathbf{F}_{2n+1}}{\mathbf{F}_{2n+2}}=\frac{\mathbf{F}_{2n+2}\mathbf{F}_n-\mathbf{F}_{2n+1}\mathbf{F}_{n+1}}{\mathbf{F}_{2n+2}\mathbf{F}_{n+1}}=\frac{(-1)^{n+1}\mathbf{F}_{n+1}}{\mathbf{F}_{2n+2}\mathbf{F}_{n+1}}=\frac{(-1)^{n+1}}{\mathbf{F}_{2n+2}}$$

as needed. □

Alternately, both (i) and (ii) can be obtained without d'Ocagne's identity. Using Honsberger's identity, it is easy to see that $\mathbf{F}_{2n+2}=(a\mathbf{F}_{n+1}+2\mathbf{F}_n)\mathbf{F}_{n+1}$. (fn. 4) From Corollary 7, we have

$$\begin{aligned}\lim_{b\to\infty}[0,\underbrace{a,\ldots,a}_{n\text{-times}},b,\underbrace{a,\ldots,a}_{n\text{-times}}]-[0,\underbrace{a,\ldots\ldots,a}_{(2n+1)\text{-times}}]&=\lim_{b\to\infty}\frac{(-1)^n(a-b)}{(a\mathbf{F}_{n+1}+2\mathbf{F}_n)(b\mathbf{F}_{n+1}+2\mathbf{F}_n)}\\&=\frac{(-1)^n(-1)}{(a\mathbf{F}_{n+1}+2\mathbf{F}_n)\mathbf{F}_{n+1}}=\frac{(-1)^{n+1}}{\mathbf{F}_{2n+2}},\end{aligned}$$

which is (ii). With (ii) and a simple move in algebra, we can obtain (i). (fn. 5) In the end of this section, we introduce some terminology connected to the first three metallic ratios. Recall that $[\underbrace{a,\ldots,a}_{n\text{-times}}]=\frac{\mathbf{F}_{n+1}}{\mathbf{F}_n}$, where $\mathbf{F}_n$ denotes the nth a-Fibonacci number, converges to the golden, the silver, and the bronze ratio when $a=1,2,3$ respectively. As 1-Fibonacci numbers are the original ones, 2-Fibonacci and 3-Fibonacci numbers are also called the silver and the bronze Fibonacci numbers respectively. More interestingly for $n\in\mathbb{Z}^+$, the nth 2-Fibonacci number coincides with the nth Pell number. Pell numbers are named after John Pell, since they appear in the study of Pells equations $x^2-2y^2=\mp1$. (fn. 6) In particular, the odd terms of the sequence of ordered pairs defined recursively as

$$(x_n,y_n):=\begin{cases}(1,1) & \text{for } n=1\\(3,2) & \text{for } n=2\\2(x_{n-1},y_{n-1})+(x_{n-2},y_{n-2}) & \text{for } n\geq 2\end{cases}$$

4. In particular, $\mathbf{F}_{2n+2}=\mathbf{F}_{n+1}\mathbf{F}_{n+2}+\mathbf{F}_n\mathbf{F}_{n+1}=(\mathbf{F}_{n+2}+\mathbf{F}_n)\mathbf{F}_{n+1}=(a\mathbf{F}_{n+1}+2\mathbf{F}_n)\mathbf{F}_{n+1}$.

5. $\frac{\mathbf{F}_{n+1}}{\mathbf{F}_n}-\frac{\mathbf{F}_{2n+2}}{\mathbf{F}_{2n+1}}=-\frac{\mathbf{F}_{n+1}}{\mathbf{F}_n}\frac{\mathbf{F}_{2n+2}}{\mathbf{F}_{2n+1}}(\frac{\mathbf{F}_n}{\mathbf{F}_{n+1}}-\frac{\mathbf{F}_{2n+1}}{\mathbf{F}_{2n+2}})=-\frac{\mathbf{F}_{n+1}}{\mathbf{F}_n}\frac{\mathbf{F}_{2n+2}}{\mathbf{F}_{2n+1}}\frac{(-1)^{n+1}}{\mathbf{F}_{2n+2}}=\frac{(-1)^n\mathbf{F}_{n+1}}{\mathbf{F}_n\mathbf{F}_{2n+1}}$.

6. Unfortunately, Euler erroneously attributed another English mathematician Lord William V. Brouncker's work to John Pell, who had negligible contribution to the study.

are solutions to $x^2-2y^2=-1$, and those even terms are solutions to $x^2-2y^2=1$. Note that y_n defined above is called the nth Pell number, and it indeed matches the nth 2-Fibonacci number. (fn. 7) With the original Fibonacci numbers, Pell numbers, and the bronze Fibonacci numbers, we present some nice identities in the next remark. (fn. 8)

Remark 9. *Let F_n, P_n, B_n denote the nth Fibonacci number, Pell number, bronze Fibonacci number respectively. Then we have*

(i) $[\underbrace{1,\ldots,1}_{n\text{-times}},3,\underbrace{1,\ldots,1}_{n\text{-times}}] = \dfrac{F_{n+4}F_{n+1}}{(F_{n+2})^2}.$

(ii) $[\underbrace{2,\ldots,2}_{n\text{-times}},4,\underbrace{2,\ldots,2}_{n\text{-times}}] = \dfrac{2P_{n+2}P_{n+1}}{(P_{n+1}+P_n)^2}.$

(iii) $[\underbrace{3,\ldots,3}_{n\text{-times}},5,\underbrace{3,\ldots,3}_{n\text{-times}}] = 1+\dfrac{4(B_{n+1})^2-(B_n)^2}{(B_{n+1}+B_n)^2}.$

Proof. (i) is already proved in [4, 6]. Note that in the computation of (ii) and (iii), we will be using a simple identity that $a\mathbf{F}_{n+1}\mathbf{F}_n+\mathbf{F}_{n+1}\mathbf{F}_{n-1}=(\mathbf{F}_{n+1})^2$ for $a=2$ and $a=3$ respectively.

To see (ii), we take the formula from Lemma 3 with $a=2$ and $b=4$. In the case of $a=2$, we will be using $\mathbf{F}_{n+2}=2\mathbf{F}_{n+1}+\mathbf{F}_n$ recursively. Recall that 2-Fibonacci numbers and Pell numbers coincide, that is, $\mathbf{F}_n=P_n$ when $a=2$. Consequently, we have

$$\begin{aligned}[\underbrace{2,\ldots,2}_{n\text{-times}},4,\underbrace{2,\ldots,2}_{n\text{-times}}] &= \frac{4(P_{n+1})^2+2P_{n+1}P_n}{4P_{n+1}P_n+(P_n)^2+P_{n+1}P_{n-1}}\\ &= \frac{2P_{n+1}(2P_{n+1}+P_n)}{2P_{n+1}P_n+(P_n)^2+2P_{n+1}P_n+P_{n+1}P_{n-1}}\\ &= \frac{2P_{n+1}P_{n+2}}{2P_{n+1}P_n+(P_n)^2+(P_{n+1})^2} = \frac{2P_{n+1}P_{n+2}}{(P_{n+1}+P_n)^2}\end{aligned}$$

as needed.

To see (iii), we take the formula from Lemma 3 with $a=3$ and $b=5$. In the case of $a=3$, we will be using $\mathbf{F}_{n+2}=3\mathbf{F}_{n+1}+\mathbf{F}_n$ recursively. Recall that a 3-Fibonacci number is also called the bronze Fibonacci number, that is, $\mathbf{F}_n=B_n$

7. See [2, Chapter 2] for more details.

8. It is a remark because we cannot claim they are all our discoveries. In particular, we contributed a proof to (i) while the closed-form expression was already given in [3]. Similarly, we provide a proof for (ii), but the closed-form expression is suggested by our referees.

when $a = 3$. Consequently, we have

$$\begin{aligned}
[\underbrace{3,\ldots,3}_{n\text{-times}},5,\underbrace{3,\ldots,3}_{n\text{-times}}] &= \frac{5(B_{n+1})^2 + 2B_{n+1}B_n}{5B_{n+1}B_n + (B_n)^2 + B_{n+1}B_{n-1}} \\
&= \frac{4(B_{n+1})^2 + (B_{n+1})^2 + 2B_{n+1}B_n}{2B_{n+1}B_n + (B_n)^2 + 3B_{n+1}B_n + B_{n+1}B_{n-1}} \\
&= \frac{4(B_{n+1})^2 + (B_{n+1}+B_n)^2 - (B_n)^2}{2B_{n+1}B_n + (B_n)^2 + (B_{n+1})^2} \\
&= \frac{4(B_{n+1})^2 + (B_{n+1}+B_n)^2 - (B_n)^2}{(B_{n+1}+B_n)^2} \\
&= 1 + \frac{4(B_{n+1})^2 - (B_n)^2}{(B_{n+1}+B_n)^2}
\end{aligned}$$

as needed. □

4 Two approaches to the same closed-form expression

While $[\underbrace{1,\ldots,1}_{n\text{-times}},3,\underbrace{1,\ldots,1}_{n\text{-times}}] = \frac{F_{n+4}F_{n+1}}{(F_{n+2})^2}$ is the original inspiration for this paper, it is worthwhile to briefly address the key idea in two other approaches that we know of.

In [4], Juniata Problem Solving Group observed how a continued fraction is evaluated and noticed the computation went "flipping the bottom value" and "adding 1" repeatedly. We therefore defined a function $f(x) := 1 + \frac{1}{x}$ that represented the evaluating process. For example, $[1,1,1,x] = f(f(f(x))) = f^{(3)}(x)$. We showed $f^{(n)}(x) = \frac{xF_{n+1}+F_n}{xF_n+F_{n-1}}$ and used it to compute

$$[\underbrace{1,1,\ldots,1}_{n\text{-times}},3,\underbrace{1,1,\ldots,1}_{n\text{-times}}] = f^{(n)}(2 + f^{(n)}(1)).$$

On the other hand, Walther Janous in [6] used

$$[\underbrace{1,1,\ldots,1}_{n\text{-times}}] = \frac{F_{n+1}}{F_n} \text{ and } \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}^n = \begin{bmatrix} F_{n+1} & F_n \\ F_n & F_{n-1} \end{bmatrix}$$

to compute the entries in the first column of $\begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}^n \begin{bmatrix} 3 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}^n$. It worked elegantly because of the connection between

$$[a_1, a_2, \ldots, a_n] = \frac{P_n}{Q_n} \text{ and } \prod_{i=1}^{n} \begin{bmatrix} a_i & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} P_n & P_{n-1} \\ Q_n & Q_{n-1} \end{bmatrix}.$$

More information regarding the matrix representation of continued fractions and their convergents can be found in [1].

Acknowledgements

We wish to express our heartfelt gratitude to the referees and editors for their meticulous reviews and constructive suggestions, which have greatly enhanced the quality of our paper. We are also deeply thankful to Professor LaBuz for introducing us to Mathematics Exchange and encouraging us to submit our work. Last but not least, we extend our sincere appreciation to Professor Escuadro, whose support and delightful weekly snacks have been essential to the existence of the Juniata Problem Solving Group. His insightful feedback during the early stages of this project proved invaluable.

Bibliography

[1] J. Sutherland Frame, *Continued Fractions and Matrices*, The American Mathematical Monthly, Taylor & Francis, Vol. 56, No. 2, 1949, pp. 98–103. DOI: 10.2307/2306169.

[2] Thomas Koshy, *Pell and PellLucas Numbers with Applications*, 1st ed., Springer New York, NY, 2014, pp. XXIII+431. ISBN: 978-1-4614-8488-2. DOI: 10.1007/978-1-4614-8489-9.

[3] Steven J. Miller, *Problem Department*, The Pi Mu Epsilon Journal, Vol. 15, No. 6, 2022, pp. 374–375.

[4] Steven J. Miller, *Problem Department*, The Pi Mu Epsilon Journal, Vol. 15, No. 7, 2022, pp. 445–446.

[5] Greg Oman and Charles N. Curtis, *Problems and Solutions*, The College Mathematics Journal, Taylor & Francis, Vol. 51, No. 5, 2020, pp. 386–392. DOI: 10.1080/07468342.2020.1826771.

[6] Greg Oman and Charles N. Curtis, *Problems and Solutions*, The College Mathematics Journal, Taylor & Francis, Vol. 52, No. 5, 2021, pp. 388–396. DOI: 10.1080/07468342.2021.1969181.

Ball State Undergraduate Mathematics Exchange, 18(1), Fall 2024, 88-106
https://openjournals.bsu.edu/mathexchange
DOI: 10.33043/r2y588ab

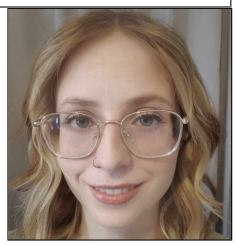
Pages 88 – 106

Catalan Number Sequences and Generalized Action Graphs

*Drew Caldwell, Ali Cochran, Nathan Glisson, Bryce Jennings, Katy McDicken, Luke James Proctor, Sarah Klanderman, Amelia Tebbe**

Drew Caldwell is an undergraduate student studying Mathematics and Computer Science at Indiana University-Kokomo who is expected to graduate in May of 2025. Drew graduated from Western High School in Russiaville, Indiana. Drew chose to study mathematics after teaching himself how to learn math in preparation for the SAT. His favorite theorem is the "Shoes and Socks theorem", and favorite math class is Calculus II.

Ali Cochran is an undergraduate student studying mathematics at Indiana University Kokomo. She plans to pursue graduate studies to deepen her expertise in the field. Outside of academia, she runs a crochet business and enjoys recording music, blending creativity with her analytical pursuits.



Nathan Glisson hails from Cincinnati, Ohio. Currently a senior double majoring in physics and mechanical engineering, he will be receiving his degree in physics from Marian University and his degree in mechanical engineering from Purdue. He is a four-year member of the Marian Knights Football team and volunteers with a local youth ministry group called Str8Up Life. In his spare time, he enjoys hanging out with friends, sitting out in nature, and building with Legos.

***Corresponding author:** antebbe@iu.edu

Bryce Jennings is an undergraduate student at Indiana University Kokomo and is pursuing a double major in Mathematics and Computer Science, with a projected gradation date of May 2024. Growing up in Fort Wayne, Indiana, Bryce found his appreciation for math at Fort Wayne Snider High School when taking AP Calculus. Bryce is also a cross country and track athlete for IU Kokomo.

Katy McDicken worked on this paper as a junior at Marian University. She is currently a 5th year in a dual degree program with Purdue University Indianapolis studying Mathematics and Mechanical Engineering. She is also on the cycling team at Marian and hopes to continue pursuing cycling and work in the engineering field after graduation.

Luke James Proctor Proctor graduated from Marian University with a Bachelors in Mathematics and Philosophy. He is interested in Abstract Algebra, Topology, Economic Theories, Philosophy of Science, German Philosophy, and Neo-Aristotelianism. Luke currently works at Dayton Freight as a W&R inspector and plans to work in software engineering and to eventually attend graduate school to get a PhD in Mathematics.

Sarah Klanderman is an Assistant Professor of Mathematics at Marian University. Her research interests include computations related to topological Hochschild homology, studying students' transition to proof-writing courses, connections between mathematics and other disciplines, and working with undergraduate research students. Her grant work is focused on supporting underrepresented students in STEM.

Amelia Tebbe is an Associate Professor of Mathematics at Indiana University Kokomo. Her research interests include algebraic topology, particularly functor calculus and applications of combinatorics to homotopy theory, and working with undergraduates. Outside of work, she enjoys hiking with her dog, gardening, and knitting.

1 Background

Action graphs emerged from work of Bergner and Hackney on category actions in the context of Reedy categories [4]. Alvarez, Bergner, and Lopez showed that action graphs could be inductively generated without reference to category actions, and they proved that the number of vertices added to A_n is the n-th Catalan number [2].

Definition 1.1 ([2])**.** The sequence of **action graphs** $\{A_n\}$ is defined inductively. Action graph A_0 is one vertex labeled 0 with no edges. Construct A_{k+1} from A_k by considering each vertex v in A_k. For each path from v to a vertex labeled k in A_k, add a new edge with source v to a new target vertex labeled $k+1$. Note that trivial paths are included in the paths considered from vertex v.
The first few graphs can be seen in Figure 1.

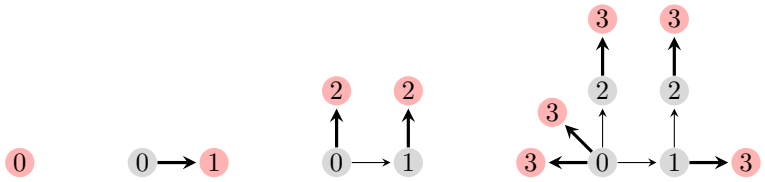


Figure 1: Action Graphs A_0 through A_3, with new edges and vertices highlighted

Definition 1.2. The **Catalan numbers** are a sequence of natural numbers given by

$$C_0 = 1, \quad C_n = \sum_{i=0}^{n-1} C_i C_{n-1-i} = \binom{2n}{n} \frac{1}{n+1}.$$

The first few numbers of the sequence are $1, 1, 2, 5$, and 14.
The action graphs in [2] were further generalized in [3] by Cressman, Lin, Nguyen, and Wiljanen, who showed that the Fuss-Catalan numbers have a similar relation to another set of inductively defined directed graphs. The Fuss-Catalan numbers are a generalization of the Catalan numbers.

Definition 1.3 ([1], A14)**.** The **Fuss-Catalan numbers** are defined by

$$C_{n,k} = \sum_{n_1+n_2+\cdots+n_{k+1}=n-1} \prod_{i=1}^{k+1} C_{n_i,k} = \frac{\binom{n(k+1)}{n}}{kn+1}.$$

The sequence of the Fuss-Catalan numbers when $k = 2$, that is $C_{n,2}$, is $1, 1, 3, 12, 55, \ldots$.
Observe that the Fuss-Catalan numbers agree with the Catalan numbers when $k = 1$, that is $C_{n,1} = C_n$.
Cressman et al. expanded on the work of Alvarez, Bergner, and Lopez, developing new action graphs for the Fuss-Catalan numbers, which they called *generalized action graphs* [3].

Definition 1.4. [3] The **generalized action graphs** $T_{n,k}$ for the Fuss-Catalan numbers $C_{n,k}$ are defined inductively. For all k, $T_{0,k}$ is one vertex labeled 0 with no edges. Construct $T_{n,k}$ from $T_{n-1,k}$ by considering each vertex v in $T_{n-1,k}$. For each path of length ℓ from v to a vertex labeled $n-1$ in $T_{n-1,k}$, add $\binom{\ell+k-1}{\ell}$ new vertices labeled n and new edges from v to each of those new target vertices.

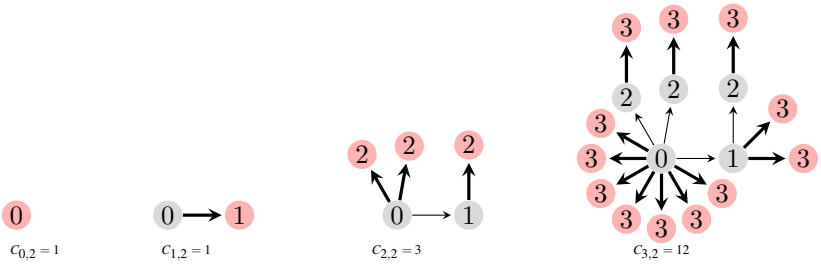


Figure 2: Generalized Action Graphs $T_{0,2}$ through $T_{3,2}$

Cressman et al. prove that the number of new vertices added to the generalized action graph $T_{n,k}$ is the Fuss-Catalan number $C_{n,k}$.
In this paper, we first discuss our notion of generalized action graphs. Then, in Sections 2, 3, 4, and 5, we consider several other sequences related to the Catalan numbers, namely $(a,b)-$Catalan numbers, Catalan's triangle, internal triangles, and super Catalan numbers respectively. We show action graphs cannot be generalized to Catalan's triangle, $(a,b)-$Catalan numbers, nor internal triangles. We also conjecture a method for constructing action graphs for the super Catalan numbers.

1.1 Conventions and notation.

We use the following conventions for descendent and subtree in this paper.

Definition 1.5. A **descendent** of a vertex v in a directed graph is any vertex w such that there is a directed path from v to w. For a rooted tree T and a vertex v of T, the **subtree of T with root** v is the induced subgraph of vertex v and its descendants.
Because generalized action graphs can easily become unwieldy in size, we introduce the following condensed notation to ease bookkeeping.
Since generalized action graphs have many identical subgraphs with the same labels, we collapse them. For an edge from a vertex labeled k to a vertex labeled n, the multiplier, $\times m$, indicates the number of such edges from a vertex labeled k to vertices labeled n in the original graph. For a vertex in the condensed form, we can find the number of vertices it represents in the standard form by multiplying the labels along the path from the root to that vertex. For example, the upper right vertex labeled 2 in the condensed graph in Figure 3 represents the $2 \times 2 = 4$ vertices labeled 2 that come off of the two vertices labeled 1 in the original graph.

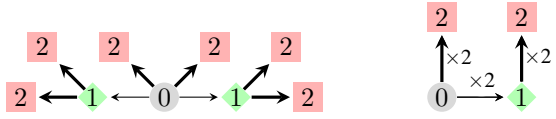


Figure 3: A directed rooted tree in standard and condensed form

As in Figure 3, we will sometimes change the color and shape of vertices in examples. This is intended only to highlight existing structure in the graphs, not to indicate any additional graph coloring or labeling.
This paper will reference the sequences given in Table 1.

Name	**Notation & Definition**	**Number Sequence**
Catalan numbers	$C_n = \binom{2n}{n}\frac{1}{n+1}$	$1,1,2,5,14,\ldots$
Fuss-Catalan numbers	$C_{n,k} = \frac{1}{kn+1}\binom{n(k+1)}{n}$	$k=1$: $1,1,2,5,14,\ldots$ $k=2$: $1,1,3,12,55,\ldots$
Catalan's triangle	$C(n,k) = \frac{n-k+1}{n+1}\binom{n+k}{k}$	Table 2
$(a,b)-$Catalan numbers	$Cat(a,b) = \frac{1}{a+b}\cdot\binom{a+b}{a}$	Table 3
Weak $(a,b)-$Catalan numbers	$cat(a,b) = \frac{1}{a+b}\cdot\binom{a+b}{a}$	Definition 2.2
Internal triangles	$t(n) = (n+2)C_{n-1} - 2C_n$	$2,14,72,\ldots$
Super Catalan numbers	$S(m,n) = \frac{(2m)!(2n)!}{m!n!(m+n)!}$	$m=0$: $1,2,6,20,70,\ldots$ $m=1$: $2,2,4,10,28,\ldots$

Table 1: A table of sequences referenced in this paper.

1.2 Generalized action graph properties

We expand on the work of [3] by investigating whether similar graphs can be constructed for other sequences. Action graphs corresponding to the Catalan numbers and the Fuss-Catalan numbers have been defined, but there does not exist a general solution for any sequence of numbers in the literature. Based on the previous work on this topic, we define three axioms that describe generalized action graphs and explore some of the properties of such graphs.

Definition 1.6. The sequence $\{G_n\}$ of **generalized action graphs** for a particular sequence $\{s_n\}$ of positive integers is a sequence of directed, labeled graphs such that:

Axiom 1 We define G_0 as the graph with s_0 vertices labeled 0 and no edges. We construct G_n from G_{n-1} by adding s_n new vertices, which are each labeled n.

Axiom 2 For vertex v in G_n, the subtree of G_n with root v is isomorphic to some G_k such that $k \leq n$.

Axiom 3 All leaves in the graph G_n have label n.

Based on these axioms, some necessary conditions for generalized action graphs follow.

Lemma 1. *In order for a sequence $\{s_n\}_{n\geq 0}$ to form a valid sequence of generalized action graphs, it must have the property that $s_0 = 1$.*

Proof. Assume that $\{s_n\}$ has the property that $s_0 \neq 1$. Then, G_0 would have multiple nodes labeled 0. If we consider the subtree rooted at any one of those 0 nodes in G_0, that subtree will contain just a single node, and therefore is not

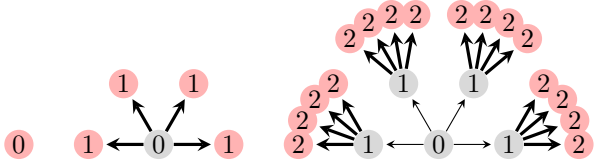
isomorphic to a generalized action graph in the sequence, which violates Axiom 2. Therefore, any sequence that has $s_0 \neq 1$ cannot have valid generalized action graphs. □

Lemma 2. *In order for a sequence $\{s_n\}_{n\geq 0}$ to form a valid sequence of generalized action graphs, it must satisfy $s_2 \geq s_1^2$.*

Proof. By Lemma 1, $s_0 = 1$, and thus G_0 is a single node labeled 0. Suppose for contradiction that $s_1 = a$ and $s_2 = b$ such that $a^2 > b$. To create G_1, we add s_1 new leaves labeled 1. So G_1 has a total of a new nodes labeled 1 with edges from the single node labeled 0. Axiom 2 implies that in all future generalized action graphs, each node with path length zero to the newest node will result in a new nodes.
To create G_2, we will add s_2 nodes labeled 2. However, as stated above, each node labeled 1 will result in a new nodes labeled 2. This forces a^2 nodes labeled 2 to be added to the graph, which is greater than our allowed $s_2 = b$ nodes labeled 2. Therefore, in order for a generalized action graph to exist, it must be that $s_2 \geq s_1^2$. □

The following example gives a more visual representation of the property described by Lemma 2.

Example 3. *Consider a sequence $\{a_n\}$ given by $1,4,14,48,\ldots$.*
Since $a_0 = 1$, this creates a single node on the first step labeled as 0. Then, since $a_1 = 4$, four nodes labeled with 1 have edges from the node labeled 0, which sets the precedent (by Axiom 2) that new nodes get four new leaves in the next step. Then, in the third step, since the most recently added nodes are known to receive four new nodes as descendants, we conclude that there should be at least 16 nodes labeled 2. However, $a_2 = 14$. If we add 16 nodes, then the graph will not satisfy Axiom 1.



2 $(a,b)-$Catalan numbers

A sequence of numbers that is closely related the Catalan numbers is the sequence of $(a,b)-$Catalan numbers.

Definition 2.1. [1, A16] For relatively prime integers $a \geq 0$ and $b \geq 0$, the $(a,b)-$**Catalan number**, denoted $Cat(a,b)$, is given by:

$$Cat(a,b) = \frac{1}{a+b} \cdot \binom{a+b}{a} = \frac{(a+b-1)!}{a!b!}.$$

Notice that the definition of an $(a,b)-$Catalan number requires that a and b be relatively prime integers. In order to assist in some of our exploration

and calculations, it is useful to drop the relatively prime requirement. For non-negative integers a and b, we define a generalization of the $(a,b)-$Catalan numbers that we call *weak* $(a,b)-$*Catalan numbers*. As far as we are aware, this sequence has not previously been studied.

Definition 2.2. For any non-negative integers a and b that are not both zero, the **weak** $(a,b)-$**Catalan number** is given by

$$cat(a,b) = \frac{1}{a+b} \cdot \binom{a+b}{a}.$$

Remark 4. *An important difference between standard* $(a,b)-$*Catalan numbers,* $Cat(a,b)$*, and weak* $(a,b)-$*Catalan numbers,* $cat(a,b)$*, is that a standard* $(a,b)-$*Catalan number will always be an integer, while a weak* $(a,b)-$*Catalan number may not be an integer.*

Example 5. *Notice that* $a = 3$ *and* $b = 6$ *are not relatively prime numbers. We compute the weak* $(a,b)-$*Catalan number*

$$cat(3,6) = \frac{(3+6-1)!}{3!6!} = \frac{8!}{3!6!} = \frac{28}{3}.$$

For the remainder of this paper, when we refer to an $(a,b)-$Catalan number, we will mean the more general weak $(a,b)-$Catalan number.

2.1 Properties of $(a,b)-$Catalan numbers

We will discuss some important properties of $(a,b)-$Catalan numbers, including how they are connected to the Fuss-Catalan and Catalan numbers.
Observe that for any a and b, $cat(a,b) = cat(b,a)$ since

$$cat(a,b) = \frac{(a+b-1)!}{a!b!} = \frac{(b+a-1)!}{b!a!} = cat(b,a). \qquad (2.\ \text{A})$$

Next, we will see the connection between the Fuss-Catalan numbers and the $(a,b)-$Catalan Numbers. Note that $cat(n,kn+1) = Cat(n,kn+1)$ here because n and $kn+1$ are relatively prime.

Theorem 6. *[1, A16a] For all* $n \geq 0$ *and* $k \in \mathbb{N}$*,*

$$cat(n,kn+1) = C_{n,k}.$$

Proof. Recall that $C_{n,k} = \frac{1}{(kn+1)} \cdot \binom{n(k+1)}{n}$. Then, consider that

$$C_{n,k} = \frac{1}{(kn+1)} \binom{n(k+1)}{n} = \frac{1}{kn+1} \cdot \frac{(n(k+1))!}{n! \cdot (n(k+1)-n)!} = \frac{(nk+n)!}{n! \cdot (nk+1)!}.$$

Now, notice that

$$cat(n,kn+1) = \frac{(n+(kn+1)-1)!}{n!(kn+1)!} = \frac{(nk+n)!}{n!(nk+1)!}.$$

Thus, $cat(n,kn+1) = C_{n,k}$ for all $n \geq 0$ and $k \in \mathbb{N}$. □

Corollary 7 ties the $(a,b)-$Catalan numbers in with the Catalan numbers. This result can be observed using the previous theorem and the connection between the Fuss-Catalan numbers and Catalan numbers, $C_{n,1} = C_n$.

Corollary 7. *[1, A16a] For all $n \geq 0$,*

$$cat(n, n+1) = C_n.$$

To assist in proving Proposition 9, we state the following lemma.

Lemma 8. *For $a \in \mathbb{N}$, $cat(a,1) = cat(1,a) = 1$.*

Proof. Let $a \in \mathbb{N}$. By definition of (a,b)-Catalan number:

$$cat(a,1) = \frac{(a+1-1)!}{a! \cdot 1!} = \frac{a!}{a!} = 1.$$

Applying Equation (2. A), $cat(a,1) = cat(1,a) = 1$. □

Our goal is to explore whether we can make generalized action graphs for the $(a,b)-$Catalan numbers. The following proposition will assist us in that exploration.

Proposition 9. *For all $n \geq 4$,*

$$n^2 \cdot cat(n+1,1)^2 > n \cdot cat(n+2,2).$$

Proof. We will prove this proposition using induction.
For the base case, consider $n = 4$. We have $4^2 cat(5,1)^2 > 4cat(6,2)$ since $4^2 cat(5,1)^2 = 16$ and $4cat(6,2) = 14$.
Suppose there is some $k \geq 4$ such that $k^2 \cdot cat(k+1,1)^2 > k \cdot cat(k+2,2)$. By Lemma 8, we know that $cat(a,1) = 1$ for any $a \in \mathbb{N}$. So, we have

$$k^2 > k \cdot cat(k+2,2).$$

Since $k > 0$, the above equation gives $k > cat(k+2,2)$, which implies that $k \cdot \frac{k+4}{k+3} > cat(k+2,2) \cdot \frac{k+4}{k+3}$. Note that by definition:

$$cat(k+2,2) \cdot \frac{k+4}{k+3} = \frac{(k+3)!}{(k+2)!2!} \cdot \frac{k+4}{k+3} = \frac{(k+4)!}{(k+3)!2!} = cat(k+3,2).$$

So, we have $k \cdot \frac{k+4}{k+3} > cat(k+3,2)$.
Note that $m^2 + 4m + 3 > m^2 + 4m$ for any $m \in \mathbb{N}$. Since

$$(m+1)(m+3) = m^2 + 4m + 3 > m^2 + 4m = m(m+4),$$

we can conclude that $m+1 > m \cdot \frac{m+4}{m+3}$. Thus the same is true for k, that is

$$k+1 > k \cdot \frac{k+4}{k+3} > cat(k+3,2).$$

Since $cat(k+2,1)^2 = 1^2 = 1$ by Lemma 8, we know

$$k+1 = (k+1) \cdot cat(k+2,1)^2,$$

and therefore $(k+1) \cdot cat(k+2,1)^2 > cat(k+3,2)$, which implies

$$(k+1)^2 \cdot cat(k+2,1)^2 > (k+1) \cdot cat(k+3,2).$$

This finishes the inductive step.

□

The following two lemmas provide other properties of $(a,b)-$Catalan numbers.

Lemma 10. *For any non-negative integers a and b,*

$$cat(a,b) = cat(a,b-1) + \frac{a-1}{a} cat(a-1,b).$$

Proof. We can use the definition of the $(a,b)-$Catalan numbers and simplify:

$$\begin{aligned}
cat(a,b-1)+\frac{a-1}{a}cat(a-1,b) &= \frac{(a+b-2)!}{a!(b-1)!}+\frac{a-1}{a}\frac{(a+b-2)!}{(a-1)!b!}\\
&= \frac{b(a+b-2)!}{a!b!}+\frac{(a-1)(a+b-2)!}{a!b!}\\
&= \frac{(a+b-2)!(b+(a-1))}{a!b!}\\
&= \frac{(a+b-1)!}{a!b!}\\
&= \frac{1}{a+b}\frac{(a+b)!}{a!b!}\\
&= cat(a,b).
\end{aligned}$$

□

Lemma 11. *For any integer $n \geq 2$,*

$$cat(3,n) = cat(3,n-1)+\frac{n+1}{3}.$$

Proof. We will proceed by induction. As a base case, consider $n=2$. We have

$$cat(3,2) = 2 = 1+\frac{2+1}{3} = cat(3,2-1)+\frac{2+1}{3}.$$

Assume for some $k \geq 2$ that $cat(3,k) = cat(3,k-1)+\frac{k+1}{3}$. Then consider,

$$\begin{aligned}
cat(3,k+1) &= \frac{1}{3+k+1}\frac{(3+k+1)!}{3!(k+1)!} && \text{(def of } (a,b)-\text{Catalan)}\\
&= \frac{k^2+5k+6}{3!}\\
&= \frac{(k+1)k}{3!}+\frac{4k+6}{3!}\\
&= \frac{(k+1)(k)}{3!}+\frac{2k+3}{3}\\
&= \frac{1}{k+2}\frac{(k+2)!}{3!(k-1)!}+\frac{2k+3}{3}\\
&= \frac{1}{3+k-1}\frac{(3+k-1)!}{3!(k-1)!}+\frac{2k+3}{3}\\
&= cat(3,k-1)+\frac{k+1}{3}+\frac{k+2}{3} && \text{(def of } (a,b)-\text{Catalan)}\\
&= cat(3,k)+\frac{k+2}{3} && \text{(inductive hypothesis)}\\
&= cat(3,(k+1)-1)+\frac{(k+1)+1}{3}.
\end{aligned}$$

Thus the inductive step holds, and our desired result follows. □

Observe that for arbitrary integers a and b, the weak (a,b)-Catalan number $cat(a,b)$ is generally a fraction. However, action graphs are constructed based

on whole number sequences. Therefore, we will introduce Catalan's triangle as a sequence that is closely related to the (a,b)-Catalan numbers, as described in Lemma 12, but whose entries are whole numbers.

3 Catalan's triangle

Another construction of numbers related to the Catalan numbers is Catalan's triangle.

Definition 3.1 ([6])**. Catalan's triangle** has entries denoted $C(n,k)$ that are defined as follows for integers n and k with $n \geq k \geq 0$:

$$C(n,k) = \frac{n-k+1}{n+1}\binom{n+k}{k}.$$

We compute the entries of Catalan's triangle up to $n = 8$ in Table 2.

n \ k	0	1	2	3	4	5	6	7	8
0	1								
1	1	1							
2	1	2	2						
3	1	3	5	5					
4	1	4	9	14	14				
5	1	5	14	28	42	42			
6	1	6	20	48	90	132	132		
7	1	7	27	75	165	297	429	429	
8	1	8	35	110	275	572	1001	1430	1430

Table 2: A table of Catalan's triangle

We make the following observations about the entries of Catalan's triangle:

(i) $C(n,0) = 1$, for $n \geq 0$.
(ii) $C(n,1) = n$, for $n \geq 1$.
(iii) $C(n+1,k) = C(n+1,k-1) + C(n,k)$, for $1 < k < n+1$.
(iv) $C(n+1,n+1) = C(n+1,n)$, for $n \geq 1$.

Because Catalan's triangle is an array rather than a sequence, there are many ways to create sequences from the entries. We consider building generalized action graphs for the sequences that arise as columns, rows, and diagonals of the triangle. We will describe these results in Section 3.2, after we have shown the relationship between Catalan's triangle and the $(a,b)-$Catalan numbers.

3.1 Relation to (a,b)-Catalan numbers

In this section, we will show that there is an interesting connection between the $(a,b)-$Catalan numbers and Catalan's triangle.

Lemma 12. *If $a \geq b \geq 0$ and $a \geq 1$, then $C(a-1,b) = (a-b)cat(a,b)$.*

Proof. Consider that for a Catalan's triangle entry $C(a-1,b)$,

$$\begin{aligned} C(a-1,b) &= \frac{a-b}{a}\binom{a+b-1}{b} \\ &= \frac{a-b}{a} \cdot \frac{(a+b-1)!}{b!(a-1)!} \\ &= \frac{(a-b)(a+b-1)!}{b!a!} \\ &= (a-b)cat(a,b). \end{aligned}$$

□

This leads to some methods for concluding whether generalized action graphs can be formed for these sequences. First, we look at a consequence of Lemma 12.

Lemma 13. *Any $(a,b)-$Catalan number with $a \neq b$ can be written as a fraction with denominator $|a-b|$. In particular, when $a > b$,*

$$cat(a,b) = \frac{C(a-1,b)}{a-b}.$$

Proof. Recall from Equation (2. A) that $cat(a,b) = cat(b,a)$. Without loss of generality, let $a > b$. By Lemma 12, $(a-b) \cdot cat(a,b) = C(a-1,b)$. Then $cat(a,b) = \frac{C(a-1,b)}{a-b}$, where $C(a-1,b)$ and $a-b$ are whole numbers. □

To demonstrate this re-writing of $(a,b)-$Catalan numbers, we have Table 3. Note that the diagonals represent where the differences in a and b are the same, and so they are written with the same denominators.

As previously noted, the weak $(a,b)-$Catalan numbers are frequently fractions rather than whole numbers. It is unclear to us what a fractional number of vertices should mean, so we will not try to directly construct generalized action graphs for this sequence. Instead, we will focus on the numerators in Table 3, which, as a result of the preceding lemmas, are Catalan's triangle numbers. Observe that the numerators of Table 3 are exactly the entries in Table 2.

3.2 Generalized action graphs for $(a,b)-$Catalan numbers and Catalan's triangle

In this section we will find that generalized action graphs for the most part cannot be made for sequences built from Catalan's triangle, and equivalently the numerators of the $(a,b)-$Catalan numbers. We will explore the columns, rows, and diagonals of Catalan's triangle.
First, we will consider the columns.

a \ b	0	1	2	3	4	5	6	7	8
1	1/1								
2	1/2	1/1							
3	1/3	2/2	2/1						
4	1/4	3/3	5/2	5/1					
5	1/5	4/4	9/3	14/2	14/1				
6	1/6	5/5	14/4	28/3	42/2	42/1			
7	1/7	6/6	20/5	48/4	90/3	132/2	132/1		
8	1/8	7/7	27/6	75/5	165/4	297/3	429/2	429/1	
9	1/9	8/8	35/7	110/6	275/5	572/4	1001/3	1430/2	1430/1

Table 3: $(a,b)-$Catalan Numbers

Proposition 14. *The columns of Catalan's triangle do not correspond to generalized action graphs, except in the case where $k = 0$.*

Proof. The entries of the columns correspond to $C(n,k)$, where k is fixed and n is iterated. Except for the columns of $k = 0$ and $k = 1$, the columns have first entry greater than 1, since the $n = k$ diagonal is the Catalan numbers. (For illustration, see Table 2.) By Lemma 1, these columns cannot be used to make a generalized action graph. For the column where $k = 1$, notice that it contradicts Lemma 2, and thus we cannot make a generalized action graph with this column. Therefore, the sequences made from the columns of Catalan's triangle cannot be modeled using (nontrivial) action graphs.
For the column where $k = 0$, we can make a somewhat trivial generalized action graph, where only one node is added each step. □

Now, we will explore the rows.

Proposition 15. *The rows of Catalan's triangle do not correspond to generalized action graphs.*

Proof. The entries of the rows correspond to $C(n,k)$, where n is fixed and k is iterated. First, we claim that for nonnegative n, $C(n,2) < C(n,1)^2$. To see this result, observe

$$C(n,2) = \frac{n-2+1}{n+1}\binom{n+2}{2} = \frac{n-1}{n+1}\cdot\frac{(n+2)!}{2!((n+2)-2)!} = \frac{n^2+n-2}{2}.$$

Then by properties of Catalan's triangle, $C(n,1) = n$, thus $C(n,1)^2 = n^2$. Now, consider that

$$\frac{n^2+n-2}{2} - n^2 = \frac{-n^2+n-2}{2}$$

and $\frac{-n^2+n-2}{2} < 0$ for all n. This implies that

$$\frac{n^2+n-2}{2} - n^2 = C(n,2) - C(n,1)^2 < 0,$$

and thus $C(n,2) < C(n,1)^2$.
So this means that using the rows to try and formulate generalized action graphs will lead to a contradiction of Lemma 2, since the second value, $C(n,1)$, when squared is always more than the third value in the sequence, $C(n,2)$. □

Next, we will explore the diagonals. We will consider the i-th diagonal to be the sequence of entries $C(i+k,k)$ where k is iterated.

Example 16. *The third diagonal in Catalan's triangle is the sequence* $C(3,0), C(4,1), C(5,2), C(6,3), \ldots$.

Proposition 17. *Action graphs cannot be formulated for the i-th diagonal of Catalan's triangle when $i \geq 3$.*

Proof. Consider Proposition 9, which says that

$$m^2 \cdot cat(m+1,1)^2 > m \cdot cat(m+2,2)$$

for all $m \geq 4$. Combining this with Lemma 12, which implies that

$$cat(m+j,j) = \frac{1}{m} C(m+j-1,j),$$

we see that

$$m^2 \frac{1}{m^2} C(m+1-1,1)^2 > m \frac{1}{m} C(m+2-1,2),$$

and hence

$$C((m-1)+1,1)^2 > C((m-1)+2,2),$$

for $m-1 \geq 3$. This results in a contradiction with Lemma 2, which implies that we cannot construct generalized action graphs for the i-th diagonal of Catalan's triangle for any $i \geq 3$.

□

There still remain two Catalan's triangle diagonals to consider. The first diagonal consists of the Catalan numbers, and thus we know they have corresponding action graphs. Constructing generalized action graphs, or proving they do not exist, for the second diagonal of Catalan's triangle remains as future work.

4 Internal Triangles.

One application of the Catalan numbers is counting the number of triangulations of polygons, using diagonals that do not intersect in the interior of the polygon. More specifically, for a convex $(n+2)$-gon $\mathscr{P}_{n+1}$, the number of triangulations is given by C_n [1]. There is a sequence related to these triangulations referred to as the *internal triangles*, which count the number of ways to draw triangles within a polygon without using the sides of the polygon.

Lemma 18 ([1], A5)**.** *The total number of internal triangles of a polygon with $n+2$ sides is given by*

$$t(n) = (n+2)C_{n-1} - 2C_n = 2\binom{2n-3}{n-4}.$$

Via either of these formulas, we can compute $t(4) = 2, t(5) = 14, t(6) = 72$, and so on.

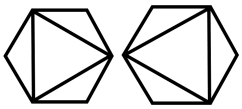


Figure 4: $t(4) = 2$, the number of internal triangles for hexagons.

We will use the term *path length rule* to refer to the number of new vertices added to a vertex based on the length of paths from the vertex to a leaf labeled n. For $n \geq 0$, ℓ_n denotes the number of new vertices added for each path of length n.

Theorem 19. *The internal triangles cannot be modeled using generalized action graphs.*

Proof. We will use path length rules to show that we cannot build consistent generalized action graphs beyond $t(6)$. Assume for a contradiction that we can model the internal triangle numbers using generalized action graphs. To satisfy Lemma 1, we begin with one vertex labeled 0. We must construct the subsequent graphs to satisfy the three generalized action graph axioms.
Since $t(4) = 2$, there is a single trivial path from the zero vertex to itself and we must have two new vertices labeled 1 coming off of our 0 vertex. So, $\ell_0 = 2$. In the next generalized action graph, our ℓ_0 rule implies that we add two new vertices for each trivial path at a vertex labeled 1. Since $t(5) = 14$, we then need to add ten additional vertices. Since there are two paths of length 1, we have $\ell_1 = 5$. These generalized action graphs are shown below in Figure 5.

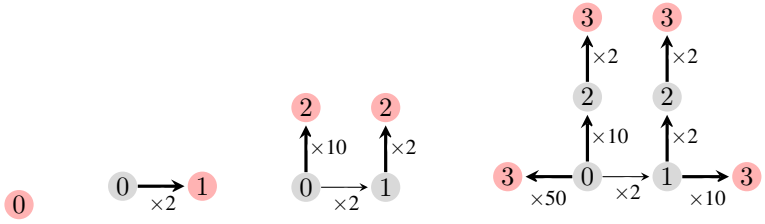


Figure 5: Attempted generalized action graphs for the internal triangle numbers

To create the next generalized action graph, note that we have fourteen vertices labeled 2 with trivial edges. By the path length rule ℓ_0, we add two vertices labeled 3 for each vertex labeled 2, giving us 28 new vertices total. We have four paths of length one starting at vertices labeled 1, and ten paths of length

one starting at vertex 0. Since $\ell_1 = 5$, we add $4 \times 5 = 20$ new vertices with edges from the vertices labeled 1 and $10 \times 5 = 50$ new vertices with edges from vertex 0. Recall that $t(6) = 72$, but we already added ninety-eight $(28 + 20 + 50)$ new vertices to this generalized action graph. Because of this contradiction, the internal triangle numbers cannot be modeled using generalized action graphs. □

5 The super Catalan numbers

The super Catalan numbers are a generalized form of the Catalan numbers with two arguments, m and n.

Definition 5.1 ([1], A17)**.** The **super Catalan numbers** are defined by

$$S(m,n) = \frac{(2m)!(2n)!}{m!n!(m+n)!}.$$

When $m = 0$, the first few numbers in the sequence are $1, 2, 6, 20$, and 70.
We wish to construct generalized action graphs for the sequence of super Catalan numbers where $m = 0$. In the rest of this section, we will state a method for constructing a sequence of directed graphs, conjecture that these graphs are indeed generalized action graphs for the sequence $S(0,n)$, and discuss our progress towards proving this conjecture.

Definition 5.2. We construct the sequence **generalized action graphs**, denoted $\{G_n\}$, for the super Catalan numbers as sequence of directed graphs defined inductively in the following way. The graph G_0 is a single vertex labeled 0. To construct G_{n+1} from G_n, consider each vertex v in G_n. For each $0 \le \ell \le n$, add $p(v,\ell) \cdot \frac{2}{2^\ell}$ new vertices labeled $n+1$ with edges from v, where $p(v,\ell)$ is the number of paths of length ℓ from v to vertices labeled n in G_n.
For an example of the construction of G_0, G_1, G_2, and G_3 see Figure 6. Note that we use the condensed notation introduced in Notation 5.

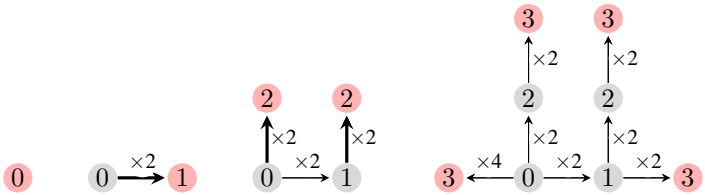


Figure 6: Generalized action graphs G_0 through G_3 for super Catalan numbers

Definition 5.2 satisfies Axiom 3, since each leaf in G_n will be labeled n and will have a path of length 0 to itself, and thus will have at least two new descendants, which will be labeled $n+1$, in G_{n+1}.

Conjecture 5.2.1. Each graph G_n in the sequence of graphs constructed in Definition 5.2 satisfies Axiom 1 and Axiom 2.
In support of this conjecture, consider the following example. In Figure 7,

we use colors to show that each subtree of G_3 is isomorphic to a previous generalized action graph. Because the graphs are drawn in condensed notation, the multiplier on the edge from the 0 vertex indicates the number of subtrees that are isomorphic to a previous generalized action graphs. In particular, considering the subtrees rooted at vertices adjacent to the zero vertex: G_3 has two subtrees isomorphic to G_2, two subtrees isomorphic to G_1, and four subtrees isomorphic to G_0, each with the labels of their vertices shifted up by an appropriate amount.

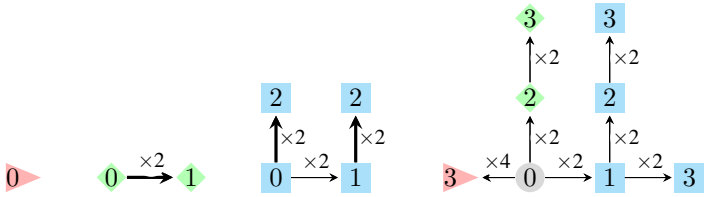


Figure 7: Highlighting the isomorphic subtrees of G_3

We will discuss our progress towards proving Conjecture 5.2.1. The data found in $\{G_n\}$ can be encoded into tables, which we call *n-tables*.
Let $K_{\ell,v,n}$ be the number of paths of length ℓ in G_n that start at a vertex labeled v and end at a vertex labeled n. For a given n, the table of $K_{\ell,v,n}$ for all values of ℓ and v is called the ***n*-table**.
Figure 8 shows G_3, the conjectured generalized action graph for $S(0,3)$ in condensed notation, and Table 4 is the corresponding n-table. For example, to find $K_{1,2,3}$, count the paths of length 1 from vertices labeled 2 to vertices labeled 3 by taking the $2\times\ 2=4$ paths from the leftmost 2 to vertices labeled 3, and adding the $2\times 2\times\ 2=8$ paths from the rightmost 2 to vertices labeled 3. Together that is 12, as seen in the corresponding entry of Table 4.

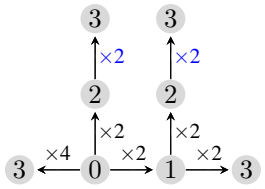


Figure 8: Conjectured G_3

$n = 3$	$v = 0$	$v = 1$	$v = 2$	$v = 3$
$\ell = 0$	0	0	0	20
$\ell = 1$	4	4	12	0
$\ell = 2$	8	8	0	0
$\ell = 3$	8	0	0	0

Table 4: $K_{\ell,v,3}$ values for G_3

Each entry in the n-table is completely determined by the graph G_n that it represents. Assuming Conjecture 5.2.2 is true, we found a way to compute the entries without having to count each path from each physically drawn generalized action graph:

- When $\ell = 0$, $K_{0,n,v} = 0$ for all v values except when $v = n$; in that case $K_{0,n,n} = S(0,n)$.
- For $\ell > 0$, entry $K_{l,v,n}$ in the n-table can be found using Conjecture 5.2.2, which uses the entries from the corresponding column of the $(n-1)$-table for the graph G_{n-1}.

Conjecture 5.2.2. The number of paths of length ℓ in G_{n+1} that start at a vertex labeled v and end at a vertex labeled $n+1$ is given by

$$K_{\ell,v,n+1} = \sum_{i=0}^{n+1-\ell-v} \frac{2}{2^i} K_{\ell-1+i,v,n}.$$

We have checked that Conjecture 5.2.2 holds up through the 7-tables, but for larger n the generalized action graphs and the associated tables become cumbersome to both create and count. As an example of how to use Conjecture 5.2.2, consider the n-tables for the graphs of $S(0,3)$ and $S(0,4)$ in Tables 5 and 6 respectively.

ℓ \ v	0	1	2	3
$\ell = 0$	0	0	0	20
$\ell = 1$	4	4	12	0
$\ell = 2$	8	8	0	0
$\ell = 3$	8	0	0	0

Table 5: $K_{\ell,v,3}$ values for G_3

ℓ \ v	0	1	2	3	4
$\ell = 0$	0	0	0	0	70
$\ell = 1$	10	8	12	40	0
$\ell = 2$	20	16	24	0	0
$\ell = 3$	24	16	0	0	0
$\ell = 4$	16	0	0	0	0

Table 6: $K_{\ell,v,4}$ values for G_4

We will show how to compute $K_{1,0,4} = 10$ from Table 6 using Conjecture 5.2.2:

$$\begin{aligned} K_{1,0,4} &= \sum_{i=0}^{3} \frac{2}{2^i} K_{i,0,3} \\ &= \frac{2}{2^0}(0) + \frac{2}{2^1}(4) + \frac{2}{2^2}(8) + \frac{2}{2^3}(8) \\ &= 10. \end{aligned}$$

If Conjecture 5.2.2 holds, we believe that Conjecture 5 can be used to compute the next super Catalan number, $S(0,n+1)$, using the $K_{\ell,v,n}$ values from the n-table for the graph of $S(0,n)$, which would show the graph G_n satisfies Axiom 1.

The subsequent super Catalan number can be computed from the n-table of its previous action graph via

$$S(0,n+1) = \sum_{\ell=0}^{n} \left(\frac{2}{2^\ell} \sum_{v=0}^{n} K_{\ell,v,n} \right).$$

This conjecture uses the sums of each column from Table 5. We will check that Conjecture 5 is true for $n = 3$:

$$\begin{aligned} S(0,4) &= \sum_{\ell=0}^{3} \left(\frac{2}{2^\ell} \sum_{v=0}^{3} K_{\ell,v,3} \right) \\ &= \frac{2}{2^0} \sum_{v=0}^{3} K_{0,v,3} + \frac{2}{2^1} \sum_{v=0}^{3} K_{1,v,3} + \frac{2}{2^2} \sum_{v=0}^{3} K_{2,v,3} + \frac{2}{2^3} \sum_{v=0}^{3} K_{3,v,3} \\ &= 2(0+0+0+20) + 1(4+4+12+0) + \frac{1}{2}(8+8+0+0) \\ &+ \frac{1}{4}(8+0+0+0) \\ &= 70, \end{aligned}$$

which is known to be the super Catalan number $S(0,4)$.

6 Acknowledgements

This project was supported by a grant from the Center for Undergraduate Research in Mathematics, which is funded by the National Science Foundation DMS awards 0636648, 1148695, and 1722563. Thanks also to Finn Richert and Maddie Dobrocky for their work in the early stages of these projects

Bibliography

[1] Richard Stanley. "Catalan Numbers" *Cambridge University Press, New York* (2015)

[2] Alvarez, Bergner, Lopez. "Action Graphs and Catalan Numbers." *Journal of Integer Sequences* Vol. 18 (2015) https://arxiv.org/abs/1503.00044v1

[3] Danielle Cressman, Jonathan Lin, An Nguyen, and Luke Wiljanen. "Generalized action graphs." (In preparation)
[4] Julia E. Bergner and Philip Hackney. "Reedy categories which encode the notion of category actions." *Fundamenta Mathematicae* 228.3 (2015) p. 193-222. http://eudml.org/doc/282637
[5] H.W.Gould and Jocelyn Quaintance. "Combinatorial Identities: Table II: Advanced Techniques for Summing Finite Series." https://-math.wvu.edu/ hgould/Vol.5.PDF
[6] Online Encyclopedia of Integer Sequences, https://oeis.org/A009766

Ball State University

Explore a Master's in Mathematical Sciences

Ball State's graduate programs are nationally ranked. They feature flexibility and immersive learning – transformative opportunities with student---driven projects, lasting outcomes, and community partnerships. All of our graduate programs are taught by outstanding faculty and in top---notch facilities. Courses in Mathematics Education are available online.

Master of Arts in Actuarial Science
Master of Arts in Mathematics Education
Master of Arts or Science in Mathematics
Master of Arts or Science in Statistics

Ball State's graduate programs will prepare you for professional work or doctoral study. Recent mathematics graduates have secured positions in the educational, pharmaceutical, insurance, and financial fields as well as continued their education to earn their doctorates.

If you want to gain important teaching, administrative, or research experience, apply for one of our **graduate assistantships.** They are awarded to students on a competitive basis.

Contact Us

e---mail: mathsci@bsu.edu
website: bsu.edu/math
Department of Mathematical Sciences
Ball State University
Muncie, IN 47306---0490
Tel: 765---285---8640

BALL STATE
UNIVERSITY

BALL STATE
UNIVERSITY