AppliedMath

MDPI

Article

Some Comments about Zero and Non-Zero Eigenvalues from Connected Undirected Planar Graph Adjacency Matrices

Daniel A. Griffith

School of Economic, Political, and Policy Science, University of Texas at Dallas, Richardson, TX 75080, USA; dagriffith@utdallas.edu

Abstract: Two linear algebra problems implore a solution to them, creating the themes pursued in this paper. The first problem interfaces with graph theory via binary 0-1 adjacency matrices and their Laplacian counterparts. More contemporary spatial statistics/econometrics applications motivate the second problem, which embodies approximating the eigenvalues of massively large versions of these two aforementioned matrices. The proposed solutions outlined in this paper essentially are a reformulated multiple linear regression analysis for the first problem and a matrix inertia refinement adapted to existing work for the second problem.

Keywords: adjacency matrix; eigenvalue; linearly dependent subsets; planar graph; spatial statistics/econometrics

check for updates

Citation: Griffith, D.A. Some Comments about Zero and Non-Zero Eigenvalues from Connected Undirected Planar Graph Adjacency Matrices. *AppliedMath* **2023**, *3*, 771–798. https://doi.org/10.3390/appliedmath3040042

Academic Editor: Tommi Sottinen

Received: 30 September 2023
Revised: 20 October 2023
Accepted: 25 October 2023
Published: 1 November 2023

Copyright: © 2023 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

This paper addresses two related linear algebra-based graph theory problems involving a binary 0-1 symmetric adjacency matrix, say **C**, representing connected undirected planar graphs with no self-loops, relevant to spatial statistics/econometrics that is generalizable to other especially applied matrix and linear algebra situations. Both pertain to the inertia of a matrix (i.e., its number of negative, zero, and positive—respectively, n_-, n_0, and n_+—eigenvalues, say $\lambda_i(\mathbf{C})$, i = 1, 2, . . . , n; e.g., see [1,2]; for a "matrix inertia theory and its applications" exposition, see Chapter 13 in Lancaster and Tismenetsky [3]). Its novelty is twofold within the context of planar graph theory: it contributes knowledge to help fill an existing gap in graph adjacency matrix nullity— the multiplicity of the zero eigenvalue in a graph adjacency matrix spectrum—theory (e.g., [4–8]); and, it adds to the scant literature published to date about matrix inertia (e.g., [9]). The first problem, which goes beyond the well-known solutions for simply calculating zero eigenvalues, concerns enumeration of linearly dependent adjacency matrix column/row subsets affiliated with these zero eigenvalues (e.g., see [10]), a single challenge here as a result of symmetry. The second problem concerns approximation of the real eigenvalues (e.g., see [11]) of these adjacency matrices, particularly their row-standardized version, say **W**, a linearly transformed version of a Laplacian matrix, that enjoys extremely popular usage in spatial statistics/econometrics applications and conceptualization nomenclature.

With regard to the first dilemma, discussion posted on the internet (see, for example: https://www.mathworks.com/matlabcentral/answers/574543-algorithm-to-extract-linearly-dependent-columns-in-a-matrix (accessed on 24 October 2023); https://stackoverflow.com/questions/28816627/how-to-find-linearly-independent-rows-from-a-matrix (accessed on 24 October 2023); and, https://stackoverflow.com/questions/28928893/linear-dependent-rows-huge-sparse-matrix (accessed on 24 October 2023)) suggests the existence of inadequate understanding about identifying a complete suite of (most likely non-unique) linearly dependent p-tuple, $p \geq 2$, subsets of adjacency (or other square) matrix (a finite collection of matrix columns/rows in the same vector space is linearly dependent if some non-zero scalar multiples of them sum to the zero vector; for the simplest 2-tuple case and a graphic theoretic

adjacency matrix, identical pairs are easily identified by a brute force non-zero element-by-element comparison algorithm) columns/rows, even if the number of zero eigenvalues is known or calculated. The proposed solution here exploits statistical linear regression estimation theory. Meanwhile, the second difficulty spotlights the desire for a refinement of an approximation offered by Griffith and Luhanga [12] paralleling the regular square lattice (e.g., see [13]) solution by Griffith [14]. Partly stimulating this inclination is an increasing availability of readily accessible high quality fine geographic resolution data for national landscapes partitioned into irregular tessellations, such as census tract or block group polygons for the United States (US), and dissemination area and block polygons for Canada (see https://www.census.gov/geographies/reference-files/time-series/geo/tallies.html (accessed on 24 October 2023) and https://www12.statcan.gc.ca/census-recensement/2021/geo/sip-pis/boundary-limites/index2021-eng.cfm?Year=21 (accessed on 15 June 2022), to name some of the germane countries, whose numbers are in the tens-of-thousands to millions; e.g., 84,414 census tracts and 8,132,968 block groups for the US in 2020, and 57,932 dissemination areas and 498,547 blocks for Canada in 2021. In other words, a substantial applied mathematics demand already exists for solving these two problems.

2. Background to the Pair of Problems

Let $G_n = (V, E)$ be a simple undirected connected graph with n vertices $V = \{v_1, \ldots, v_n\}$, and $m \leq n(n-1)/2$ edges $E = \{e_{ij}$ linking vertices v_i and v_j: $i = 1, 2, \ldots, n$ and $i = 1, 2, \ldots, n$; $i \neq j$ due to the absence of self-loops$\}$. Often in spatial statistics/econometrics, G_n is both complete and planar, and hence $m \leq 3(n-2)$; sometimes G_n is near-planar, with $m \leq 8n$. Several well-known properties of the sparse adjacency matrices for these graphs pertaining to both their zero and non-zero eigenvalues include:

(1) the number of zero eigenvalues count linearly dependent column/row subsets (e.g., [15]);

(2) the interval $\left[\underset{1\leq i\leq n}{\text{MAX}}\left\{\sqrt{\sum_{i=1}^{n} n_i^2/n}, \sum_{i=1}^{n} n_i/n\right\}, \underset{1\leq i\leq n}{\text{MAX}}\left\{\left(1+\sqrt{4n_{j|i}-4n_i+1}\right)/2\right\}\right]$ (e.g., [16,17]) contains the principal eigenvalue (i.e., spectral radius) of the graph G_n adjacency matrix $\mathbf{C}$, where n_i is the ith row sum, and $n_{j|i}$ is the sum of vertex i's linked row sums n_j;

(3) adding a link to graph G_n strictly increases its adjacency matrix $\mathbf{C}$ principal eigenvalue(e.g., [18]), whereas removing a link decreases the magnitude of this eigenvalue (e.g., [19]);

(4) the principal eigenvalue of matrix $\mathbf{W}$ is one (via the Perron-Frobenius theorem (e.g., [20]));

(5) the extreme maximum eigenvalue of matrix $\mathbf{C}$, and the extreme negative eigenvalue of matrix $\mathbf{W}$, for graph G_n compute very quickly [21];

(6) the variance of the eigenvalues of matrix $\mathbf{C}$ is $\mathbf{1}^T\mathbf{C}\mathbf{1}/n$, and of matrix $\mathbf{W}$ is $\mathbf{1}^T\mathbf{D}^{-1}\mathbf{C}\mathbf{D}^{-1}\mathbf{1}/n$, where $\mathbf{1}$ is an n-by-1 vector of ones, superscript T denotes the matrix transpose operator, and $\mathbf{D}$ is a diagonal matrix whose (i, i) cell entry is n_i [21];

(7) the sum of the positive eigenvalues equals minus the sum of the negative eigenvalues [all diagonal entries of matrix $\mathbf{C}$ are zero in the absence of self-loops, implying that $\sum_{i=1}^{n} \lambda_i(\mathbf{C}) = 0$];

(8) the sum of the k largest eigenvalues of matrix $\mathbf{C}$ is at most $(\sqrt{k}+1)n/2$ [22];

(9) the line graph, $L(G_n)$, furnishes a lower bound, whereas the maximally connected graph [23], $L(G_2) + L(G_{n-2})$, furnishes an upper bound configuration for numerical planar graph eigenfunction analysis;

(10) for irregular graphs G_n (i.e., n_i has a relatively small modal frequency coupled with a relatively large range), the theoretical maximum number of negative eigenvalues is $3n/4$ [24], whereas empirically this maximum number almost always is less than $2n/3$, and usually between $n/2$ and $3n/5$ [25], even in the presence of numerous completely connected K_4 (in graph theory parlance) subgraphs (i.e., the maximum fully connected subgraph Kuratowski's theorem allows to exist in a planar graph);

(11) a bipartite graph is always 2-colorable (i.e., one only needs to assign at most two different colors to all graph vertices such that no two adjacent vertices have the same color), and vice-versa (see [26]), implying knowledge about the number and positioning of K_4 subgraphs potentially is informative; and,

(12) if the maximum degree (i.e., n_i; see property #2) $\Delta > 3$, η denotes nullity, and G_n is not complete bipartite, then $\eta < (\Delta - 2)n/(\Delta - 1)$ [5]—this is not a very useful spatial statistics/econometrics upper bound since most irregular surface partitionings have at least one polygon areal unit with Δ sufficiently large (e.g., in the range 10–28) that $(\Delta - 2)/(\Delta - 1)$ is close to one.

Although this list is not exhaustive, and hence could contain many more eigenvalue properties, these twelve provide a foundation for the findings summarized in this paper. They also highlight that missing from this list is an effective method for determining the set of zero eigenvalues for an adjacency matrix, one of this matrix's three inertia counts.

Mohar's [22] result (i.e., property #8) needs modification in order to translate it from matrix **C** to matrix **W**. Although the line graph, $L(G_n)$ (i.e., property #9) gives a lower bound configuration for practical spatial statistical problems, an undirected star graph (i.e., K_1 in graph theory parlance) gives its absolute lower bound for the sum of positive eigenvalues, which is one for matrix **W**; this particular graph also highlights the importance of accounting for zero adjacency matrix eigenvalues. In other words, given that the largest eigenvalue, $\lambda_1(\mathbf{W})$, equals 1, the upper bound for the k positive eigenvalues is k, denoted here by matrix inertia notation n_+; realization of this value is infeasible for a connected $L(G_n)$ since all non-principal positive eigenvalues other than $\lambda_1(\mathbf{W})$ are guaranteed to be < 1. Thus, for other symmetric eigenvalue distributions (e.g., those characterizing a regular square tessellation lattice forming a complete rectangular region), this loose upper bound also is $n/2$ at most, and actually is closer to $n_+/2$; such a tessellation forming a complete $\sqrt{n}$-by-$\sqrt{n}$ square region, for a perfect square number n, with links in its dual planar graph defined by pixels sharing a non-zero length boundary, also has $\sqrt{n}$ zero eigenvalues (e.g., see Comellas et al., 2008), slightly reducing this upper bound for it by a factor of $(1 - 1/\sqrt{n})$. Meanwhile, the sum of positive $L(G_n)$ eigenvalues is approximately/exactly (depending upon whether n is even or odd) equal to $\left\{1 + \mathrm{SIN}\left[\frac{2k-1}{2(n-1)}\pi\right]/\mathrm{SIN}\left[\frac{\pi}{2(n-1)}\right]\right\}/2$, implying a sharper upper bound on k for the sum of its first k positive eigenvalues, and $n/3$—which can be <k—for the sum of all of its positive eigenvalues. This outcome suggests the following more general proposition:

Conjecture 1. *Let* $G_n = (V, E)$ *be a simple connected undirected planar graph with n vertices* $V = \{v_1, \ldots, v_n\}$, *and* $m \leq n(n-1)/2$ *edges* $E = \{e_{ij}$ *linking vertices* v_i *and* v_j: $i = 1, 2, \ldots, n$ *and* $j = 1, 2, \ldots, n; i \neq j\}$. *For* $n \geq 100$ *nodes and no self-loops,* k *is an upper bound for the sum of the first k positive eigenvalues of its row-standardized adjacency matrix* **W**.

Rationale. *By the Perron-Frobenius theorem,* $\lambda_1(\mathbf{W}) = 1$, *and all other positive eigenvalues in this case are such that* $0 < \lambda_j(\mathbf{W}) < 1$, $j = 2, 3, \ldots, k$. Taliceo and Griffith [25] *synthesize simulation experiment calculations that suggest a minimum value of around 100 for* n, *the part of this conjecture requiring proof.* Tait and Tobin [23] *underline the potential for* G_n *displaying anomalous qualities when* n *is small, noting that the maximum connectivity case requires n to exceed 8 to be true. Furthermore, in their extensive investigation of planar graphs housing embedded* K_4 *subgraphs,* Taliceo and Griffith [25] *reveal that* $3n/4$ *negative eigenvalues occur, at least asymptotically with increasing n, for certain supra-structure wheel and line graphs organizing a substructure of only* K_4 *subgraphs, which is a peculiarity vis-à-vis empirical surface partitionings. Their empirical inventory, together with the observational roster appearing in Table 1 (also see Appendix A), implies that most irregular surface partitions have a relatively small percentage of their edges arranged in the precise formation of these subgraphs.*

Table 1. Selected chosen specimen geographic landscape surface partitioning descriptors (also see Appendix A).

Geographic Landscape	n	K_4 Count	Matrix C					Matrix W	
			UB	$\lambda_1(C)$	LB	$\lambda_n(C)$	1^TC1	$\lambda_2(W)$	$\lambda_n(W)$
England †	839	9	7.59	6.07	4.97	−4.19	3564	0.9976	−0.93
Chicago ‡	2067	35	8.52	6.22	5.30	−3.73	10,562	0.9987	−0.86
Edmonton §	2098	121	10.95	6.95	5.62	−4.32	10,640	0.9985	−0.77
North Carolina ‡	2195	52	8.52	6.28	5.77	−3.76	12,206	0.9994	−0.70
US HSA ♯	3408	204	12.13	7.82	5.94	−4.46	18,308	0.9996	−0.96
Texas ‡	5265	185	12.43	7.43	5.66	−4.52	28,176	0.9995	−0.84
Syracuse †	7249	322	12.30	7.38	5.40	−5.02	35,100	0.9994	−0.94

NOTE: n denotes the number of graph G_n nodes; K_4 denotes the well-known completely connected G_4; λ denotes eigenvalue; UB/LB denote upper/lower bound; § Statistics Canada 2011 census tracts; ♯ US HSA denotes 2017 US hospital service areas.; † Appears in Griffith and Luhanga (2011). ‡ US Census Bureau 2000, 2010, or 2020 census tracts.

The rare event nature of this preceding K_4 situation reflects upon a need to adequately differentiate theoretical and practical consequences.

The principal eigenvalue $\lambda_1(\mathbf{C})$ upper bound result proved and reported by Wu and Liu [17] can extend this conjecture to binary 0-1 graph adjacency matrices with an improved bound of $k \times \underset{1 \le i \le n}{MAX}\left\{\left(1+\sqrt{4n_{j|i}-4n_i+1}\right)/2\right\} < (\sqrt{k}+1)n/2$. This value outperforms even the extreme negative eigenvalue case of $k = n/4$ for all positive eigenvalues: for example, using the true n_+ value, $(7.59)(839)/2$ (≈3184.05) is closer than $(\sqrt{393}+1)(839)/2$ (≈ 8738.76) to 656.86, the ensuing empirical England illustration numbers.

Finally, while a $\lambda_n(\mathbf{W})$ value of -1 implies the foregoing loose upper bound of $n/2$, as $\lambda_n(\mathbf{W})$ goes toward its smallest absolute value of roughly $-1/2$ (e.g., [27]), property #10 implies that this loose upper bound goes to $n/4$. One implication here is that $\lambda_n(\mathbf{W})$ can either directly or indirectly contribute to establishing sharper upper bounds. Taliceo and Griffith [25] reveal that values greater than the extreme case of -1 move the loose upper bound toward $2n/5$. This hypothesis merits future research attention. Therefore, as the discussion in this section reflects, a prominent theme in the existing literature concerns various aspects of positive eigenvalues, relegating negative eigenvalues to a secondary topic, while nearly ignoring zero eigenvalues altogether.

3. Specimen Empirical Surface Partitionings and Their Dual Graphs

Although the first problem this paper addresses entailing subsets of linearly dependent columns/rows appears to be of interest for any size matrix (e.g., Appendix A furnishes an easily replicated and control example, suitable for launching exploratory work, with much known about its zero eigenvalues for any size adjacency matrix), the second entailing eigenvalue approximations is of pragmatic interest—beyond verification exercises—especially in spatial statistics/econometrics, and only for extremely large matrices involving n in the 100,000s or greater (i.e., matrices for which numerical computation of eigenvalues is prohibitive in terms of time or computer memory requirements that surpass ultramodern resource capabilities). This section treats a relatively large n that is much smaller than this restrictive magnitude, but sufficiently large to engage computational challenges while allowing comparisons with accurate numerical solutions, namely n between 839 and 7249. The seven selected cases are actual real-world surface partitionings, with their selection decisions partly attributable to their relative and absolute sizes. Table 1 summarizes some of their relevant features. Its upper bound for $\lambda_1(\mathbf{C})$ exhibits a marked decrease—by an average of over 50%—from the maximum n_i values delivered by the Perron-Frobenius theorem.

Figure 1 portrays the selected empirical eigenvalue distributions. The complete sets of eigenvalues tend to resemble, and somewhat align with, truncated gamma probability density function distributions. The positive eigenvalue subsets more closely resemble skewed Weibull distributions (with most of the analyzed cases here displaying conspicuous tail deviations). These subset graphics help visualize the aforementioned conjecture.

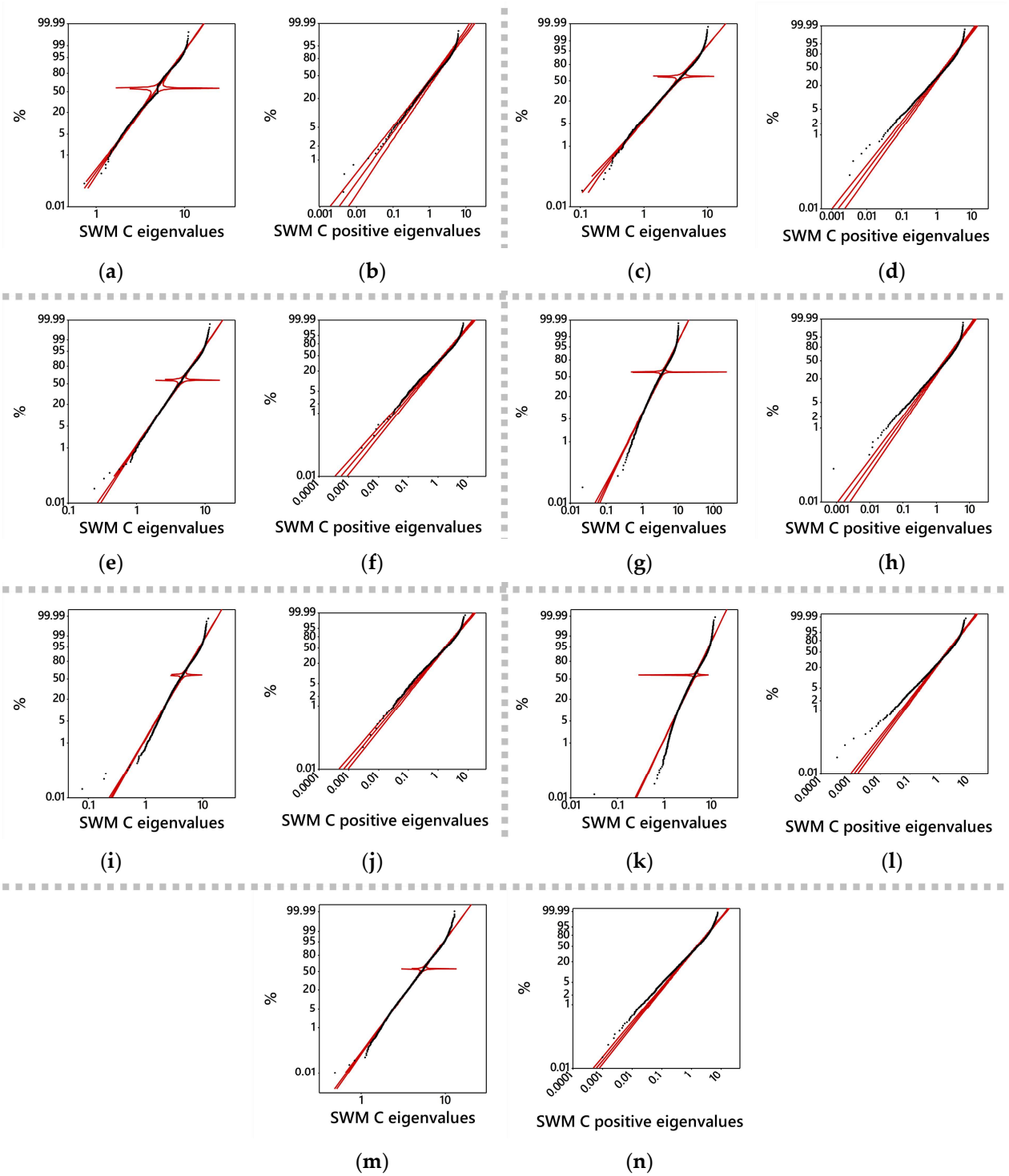


Figure 1. Matrix **C** twosomes: eigenvalue gamma and positive eigenvalue Weibull quantile plots (black) with superimposed 95% confidence intervals and trend lines (red). Top left: (**a**,**b**) England. Top right (**c**,**d**): Chicago. Top middle left (**e**,**f**): Edmonton. Top middle right (**g**,**h**): North Carolina. Bottom middle left (**i**,**j**): US HSA. Bottom middle right (**k**,**l**): Texas. Bottom (**m**,**n**): Syracuse.

Table 2 presents a second tabulation with entries for the five selected empirical adjacency matrices containing zero eigenvalues. These recordings insinuate that 2-tuples may be the dominant source of zero eigenvalues in empirical graph adjacency matrices (also see), an anticipation consistent with intuition. The affiliated zero eigenvalues are a straightforward consequence:

Table 2. Selected descriptors of the chosen specimen geographic landscape surface partitionings containing zero λs.

Feature	Removed λ = 0	England	Edmonton	US HSA	Texas	Syracuse
$\lambda_1(\mathbf{C})$	none	6.071787	6.950707	7.820136	7.431868	7.375069
	2-tuples	6.027092	6.905243	7.820136	7.413686	7.352621
	p-tuples	6.026312	6.886547	7.820136	7.406109	7.352365
$\lambda_n(\mathbf{C})$	none	−4.185268	−4.324066	−4.458946	−4.523357	−5.024567
	2-tuples	−3.729453	−4.236484	−4.353676	−4.512665	−5.018984
	p-tuples	−3.729440	−4.236477	−4.353676	−4.506244	−5.017506
$\lambda_n(\mathbf{W})$	none	−0.925413	−0.773818	−0.956681	−0.843875	−0.935861
	2-tuples	−0.900969	−0.731063	−0.956681	−0.843875	−0.935861
	p-tuples	−0.900969	−0.728697	−0.956681	−0.843875	−0.935861
$\mathbf{1}^{\mathrm{T}}\mathbf{C1}$	none	3564	10,640	18,308	28,176	35,100
	2-tuples	3400	10,584	18,294	28,142	34,932
	p-tuples	3372	10,536	18,290	28,126	34,878
$\mathbf{1}^{\mathrm{T}}\mathbf{D}^{-1}\mathbf{C}\,\mathbf{D}^{-1}\mathbf{1}$	none	177.14	397.38	620.01	977.65	1440.31
	2-tuples	165.02	394.03	619.06	976.71	1434.92
	p-tuples	161.61	391.09	618.58	976.35	1431.56
p-tuple frequency	2	79	23	6	7	48
	3	5	0	1	1	7 †
	4	3	4	1	2	9
	5	4	4	0	0	3
	6	1	1	0	0	1
	7	0	1	0	0	0
	8	0	0	0	0	2
	10	2	0	0	0	0
	11	1	0	0	0	0

NOTE: λ denotes eigenvalue; † one 2-tuple overlaps with, and hence may mask, two 3-tuples.

Lemma 1. *If each of two nodes in a graph has a single link that connects them only with a common other node, then this graph's binary 0-1 adjacency matrix* **C** *has a zero eigenvalue.*

Proof. Without loss of generality, arrange an adjacency matrix such that the two nodes in question occupy the first and second column/row positions of the adjacency matrix under study. Utilizing elementary matrix determinant column/row operations (e.g., [28,29]),

subtracting the second row from the first row of $\det|(\mathbf{C}-\lambda\mathbf{I})|$, where det denotes matrix determinant, and then adding the first column to the new second column, yields.

$$\det\left|\begin{pmatrix} -\lambda & 0 & \mathbf{0} & 1 & \mathbf{0} \\ 0 & -\lambda & \mathbf{0} & 1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & & & \\ 1 & 1 & & \mathbf{C}_{n-2,n-2} & \\ \mathbf{0} & \mathbf{0} & & & \end{pmatrix}\right| = \det\left|\begin{pmatrix} -\lambda & 0 & \mathbf{0} & 0 & \mathbf{0} \\ 0 & -\lambda & \mathbf{0} & 1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & & & \\ 1 & 2 & & \mathbf{C}_{n-2,n-2} & \\ \mathbf{0} & \mathbf{0} & & & \end{pmatrix}\right| \quad (1)$$

Next, implementing the Laplacian expansion by minors for cell entry (1, 1) renders $\lambda = 0$. This result is corroborated by the determinant property that the eigenvalues of a block diagonal matrix are the union of the individual block matrix eigenvalue sets. □

Corollary 1. *Lemma 1 applies* to $K > 1$ *separate distinct pairs of nodes in a graph that have a single link connecting each of a duo only with another common node.*

Proof. Without loss of generality, arrange an adjacency matrix such that the K pairs of nodes in question contiguously occupy its first 2K column/row positions, reducing the core adjacency matrix to $\mathbf{C}_{n-2K,n-2K}$. Recursively applying the Lemma 1 proof technique to the K twosomes yields K zero eigenvalues. □

Lemma 2. *If each of K nodes in a graph has a single link that connects them only with a common other node (i.e., a star graph, a special case of a complete bipartite graph), then this graph's binary 0-1 adjacency matrix* **C** *has at least* $K-1$ *zero eigenvalues.*

Proof. Without loss of generality, arrange an adjacency matrix such that the K nodes in question contiguously occupy its last K column/row positions, reducing the core adjacency matrix to $\mathbf{C}_{n-K,n-K}$. Then,

$$\det\left|\begin{pmatrix} \mathbf{C}_{n-K,n-K}-\lambda\mathbf{I}_{n-K} & \mathbf{0}_{n-K-1,K} \\ \mathbf{0}_{K,n-K-1} \quad \mathbf{1}_{K,1} & \mathbf{1}_{K,1}^{T} \\ & -\lambda\mathbf{I}_{K} \end{pmatrix}\right| =$$
$$\det\left|\begin{pmatrix} \mathbf{C}_{n-K,n-K}-\lambda\mathbf{I}_{n-K} & \mathbf{0}_{n-K-1,K} \\ & 1 \quad \mathbf{0}_{K-1,1}^{T} \\ \mathbf{0}_{K,n-K-1} \quad 1 & (K-1)-\lambda \quad \mathbf{0}_{K-1,1}^{T} \\ \mathbf{0}_{K-1,1} & \mathbf{0}_{K-1,1} \quad -\lambda\mathbf{I}_{K-1} \end{pmatrix}\right| \quad (2)$$

Next, implementing the Laplacian expansion by minors for the ending $(K-1)$-by-$(K-1)$ cell entries, $-\lambda_{K-1}$, is equivalent to applying it to a star graph for which one of its nodes is contained in the core adjacency matrix, $\mathbf{C}_{n-K,n-K}$, rendering at least $K-1$ zero eigenvalues (see [30], p. 266). □

Corollary 2. *If each of K nodes in a square mesh lattice (i.e., regular square tessellation) graph has a single link that connects them only with a common other node (i.e., a star graph), then this graph's binary 0-1 adjacency matrix* **C** *has at least* $P+K-2$ *zero eigenvalues.*

Proof. Extending the proof for Lemma 2, the core adjacency matrix, $C_{n-K,n-K} = C_{P^2,P^2}$ for a P-by-P regular square tessellation, in isolation has P zero eigenvalues for its rook adjacency definition—its eigenvalues are given by $2\{\mathrm{COS}[\pi j/(P+1)] + \mathrm{COS}[\pi k/(P+1)]\}$, $j = 1, 2, \ldots,$ P and $k = 1, 2, \ldots, P$ [31], with the P cases of $k = [(P+1) - j]$ yielding zero values. One of these matches occurs for each integer j, yielding a total of P zero eigenvalues. But the single

arbitrary node to which the star graph attaches has its column/row modified by its P + 1 column and row additional entries that render the matrix determinant

$$\det \left| \begin{pmatrix} \mathbf{C}_{P^2,P^2} - \lambda \mathbf{I}_{P^2} & \mathbf{0}_{P^2-1,1} \\ \mathbf{0}_{1,P^2-1} & 1 \quad (K-1) - \lambda \end{pmatrix} \right|, \tag{3}$$

and hence removes one of the P zero eigenvalues. Hence, the number is $P - 1 + K - 1 = P + K - 2$. □

Remark. *This corollary produces an exact expectation for algorithm and conjecture verification exercises like the simulation specimens outlined in Appendix A.*

Corollary 3. *If each of H sets of K_h nodes forms a separate, distinct star subgraph, then this graph's binary 0-1 adjacency matrix* **C** *has at least* $\sum_{h=1}^{H} K_h - H$ *zero eigenvalues.*

Proof. Recursively applying the Lemma 2 proof technique to the H collections of K_h nodes yields $\sum_{h=1}^{H} K_h - H$ zero eigenvalues. □

Remark. *These star subgraphs need to attach to nodes that contribute to linear combinations of the core adjacency matrix* $C_{n-\sum_{h=1}^{H} n-K_h, \sum_{h=1}^{H} K_h}$ *in order to eliminate one of its zero eigenvalues (see Corollary 2 proof).*

Table 2 entries also raise questions about the claim by Feng et al. [18] that adding links strictly increases the matrix **C** principal eigenvalue's magnitude (e.g., the US HAS specimen graph), which is the reason this table's entries show so many decimal places; rather, this spectral radius amount looks to be simply monotonically non-decreasing. This assertion may well apply to the extreme negative eigenvalues, too: they also appear to be monotonically increasing in absolute value.

The *p*-tuples acknowledged in Table 2 are the linearly dependent subsets of columns/rows producing zero eigenvalues in their respective adjacency matrices. The next section discusses a statistical regression estimation operator mechanism for uncovering these linear combinations. Their important attributes alluded to in this table include overlaps (i.e., a column/row can be a member of more than one linearly dependent subset), size variation (i.e., the number of columns/rows in a linear combination equation), and frequency that seems to decrease with increasing size, perhaps mirroring a Poisson distribution. Overlaps emphasize that the discovered p-tuples acknowledged in Table 2 are not necessarily unique; those removed for the construction of this table are ones that maintain a connected graph [documented by $\lambda_2(\mathbf{W}) \neq 1$] as well as contain the smallest possible n_is (most often just one).

4. Identifying Linearly Dependent Column/Row Subsets in Adjacency Matrices

Hawkins [32] gives an insightful clue to identifying linearly dependent column/row subsets in an adjacency matrix by noting that eigenvectors of (near-)zero principal components extracted from a correlation matrix constructed by amalgamating a dependent variable and its accompanying set of covariates suggest possible alternative sub-regressions. One weakness of his conceptualization occurs with multiple zero eigenvalues: their eigenspace is not unique [33,34], and hence is an arbitrary choice among all of the possible correct eigenvectors for a repeated (non-degenerate) eigenvalue from a real symmetric matrix. Consequently, if only a few zero eigenvalues exist, then multivariate principal components analysis (PCA) tends to recover p-tuples like the simple structure ones referenced in Table 2 acquired with the SAS PROC TRANSREG [35] procedure. Otherwise, it can confuse even the stand-alone 2-tuple sources. Accordingly, still pursuing this regression approach, standard linear regression procedures designed to detect and convey a user message equivalent to "model is not of full rank" (e.g., the SAS PROC REG [35] message) furnishes a process that yields the desired p-tuples.

Overlapping linear combinations constitute a complication whose astute management and treatment remain both elusive and a fruitful subject of future research. The Syracuse entry in Table 2 as well as Appendix A underscore some of the data analytic complexities that can arise in the presence of this phenomenon.

4.1. Competing Algorithms

SAS champions the sweep algorithm (aka SWP [35,36], [37] (pp. 95–99), [38]; Tsatsomeros [39] furnishes an excellent literature review and historical overview for it), whereas much contemporary mathematical work advocates for the classical QR with a column pivoting solution (e.g., [40–42]). As an aside, Lange ([37], pp. 93–94) notes that "[t]he sweep operator . . . is the workhorse of computational statistics," and that "[a]lthough there are faster and numerically more stable algorithms for . . . solving a least-squares problem, no algorithm matches the conceptual simplicity and utility of sweeping" due to its superior abilities to efficiently and effectively exploit linear regression matrix symmetries; he then notes that its historical competitors were/are the Cholesky decomposition, the Gram-Schmidt orthogonalization, and Woodbury's formula. Meanwhile, generally speaking, this former QR algorithm, the most serious contemporary competitor of the sweep operator, performs elementary row operations on a system of linear equations, which for least squares regression is the set of normal equations. It enables the execution of this linear regression by sweeping particular rows of the sums-of-squares-and-cross-products matrix $\mathbf{X}^T\mathbf{X}$ in or out, where sweeping is similar to executing a Gauss-Jordan elimination. This sweep procedure involves the following sequence of matrix row adjustments (the basic row operations, pivoting exclusively on diagonal matrix elements, are the multiplication of a row by a constant, and the addition of a multiple of one row to another) based upon diagonal element h in matrix $\mathbf{X}^T\mathbf{X} = \langle a_{jk} \rangle$ whose nonzero value is c_{hh}:

1. j = k = h: matrix entry $\mathbf{X}^T\mathbf{X}[h, h] = a_{hh}$,
2. j = h & k $\neq$ h: matrix entry $\mathbf{X}^T\mathbf{X}[h, k] = a_{hk}/a_{hh}$, and
3. j $\neq$ h & k $\neq$ h: matrix entry $\mathbf{X}^T\mathbf{X}[j, k] = a_{jk} - a_{jh}a_{kh}/a_{hh}$.

The row operations generating these quantities simultaneously update estimates for the regression coefficients, error sum of squares, and generalized inverse of the given system of equations. Matrix $\mathbf{X}^T\mathbf{X}$ symmetry requires calculations for only its upper triangle, reaping execution time savings. It also achieves in-place mapping with minimal storage. Meanwhile, sweeping in and out refers to the reversibility of this operation. These preceding added effects constitute "sweeping in." They can be undone by sweep a second time, constituting "sweeping out." This sweep operator is quite simple, making it an extremely useful tool for computational statisticians.

In contrast, a QR algorithm, which tends to be slow but accurate for ill-conditioned systems, applies the QR matrix decomposition (aka QR or QU matrix factorization**)** of $\mathbf{X}^T\mathbf{X} = \mathbf{QR}$, where **Q** is an orthonormal matrix, and **R** is an upper triangular matrix. The appeal of such an orthogonal matrix is that its transpose and inverse are equal. Thus, the linear regression system of equations $\mathbf{X}^T\mathbf{Xb} = \mathbf{(X^TY)}$ reduces to a triangular system $Rb = Q^T(X^TY)$, which is much easier to solve. Implementation of this QR algorithm can be carried out with column pivoting, which bolsters its ability to solve rank-deficient systems of equations while tending to provide better numerical accuracy. However, by doing so, it solves the different system of equations $\mathbf{X}^T\mathbf{XP} = \mathbf{QR}$, or $\mathbf{X}^T\mathbf{X} = \mathbf{QRP}^T$, where P is a permutation matrix linked to the largest remaining column (i.e., column pivoting) at the beginning of each new step. The selection of matrix **P** usually is such that the diagonal elements of matrix **R** are non-increasing. This system of equations switch potentially adds further complexity to handling this algorithm. Nevertheless, column pivoting is useful when eigenvalues of a matrix actually, or are expected to approach, zero, the topic of this paper. Consequently, the relative simplicity of the sweep algorithm frequently makes it preferable to a QR algorithm.

A systematic comparative review or numerical examination between the sweep operator and its handful of prominent competitors mentioned here is beyond the scope of this paper. The conceptual discussion in this section identifies certain strengths and weaknesses

of this sweep operator, which is a reasonable starting point for embarking upon such a future assessment.

4.2. SAS Procedures: REG and TRANSREG

PROC REG [35] sequentially applies a sweep algorithm to the mathematical statistics normal equations formed from the standard linear regression cross-products covariate matrix $\mathbf{X}^{\mathrm{T}}\mathbf{X}$, according to the order of the covariates appearing in its SAS input statement. It always begins by sweeping the first column in matrix $\mathbf{X}$, followed by the next column in this matrix if the pivot is less than a near-zero value whose default SAS threshold magnitude is 1.0×10^{-9} for most machines (i.e., if that column is not a linear function of the preceding column), then continuing sequentially to each of the next columns if their respective pivots are not less than this threshold amount, until it passes through all of the columns of matrix $\mathbf{X}$. This method is accurate for most undirected connected planar graph adjacency matrixes since they are reasonably scaled and not too collinear. Given this setting, this SAS procedure can uncover linearly dependent subsets of adjacency matrix columns by specifying its first column, C_1, as a dependent variable, and its remaining $(n - 1)$ columns, C_2-C_n, as covariates. SAS output from this artificial specification includes an enumeration of the existing linearly dependent subsets of columns. A second regression that is stepwise in its nature and execution can check whether or not C_1 itself is part of a linear combination subset. Simplicity dissipates here when n becomes too large, with prevailing numerical precision resulting in some linear combinations embracing numerous superfluous columns with computed near-zero regression coefficients (e.g., 1.0×10^{-11}). This rounding error corruption emerges in PROC REG [35] between n = 2100 (e.g., the Chicago empirical example) and n = 3408 (e.g., the US HSA empirical example).

In contrast, PROC TRANSREG [35] executes nonsequential sweeps, finding the best column to sweep first, followed by the second best, then the third best, and so on. This deviation from a sequential SAS input ordering tends to be more numerically accurate. Again, simplicity dissipates here when n becomes too large, although a much larger n than that for PROC REG [35], emerging here somewhere between n = 5265 (e.g., the Texas empirical example) and n = 7249 (e.g., the Syracuse empirical example). In such messy situations, which usually are for the larger -tuple values of p, ignoring the near-zero regression coefficients exposes the correct latent subset linear combinations. Therefore, PROC TRANSREG [35] is the SAS procedure most often utilized in this paper, with a few of its smaller n results checked or carried out with PROC REG [35].

4.3. Disclosing Subset Linear Combinations for the Specimen Empirical Surface Partitionings

We denote the columns of a given n-by-n matrix by COLj, j = 1, 2, . . . , n (i.e., number them from left to right in a matrix with consecutive positive integers). Among other listings, Table 2 details the numbers and sizes of subset linear combinations of the specimen graph adjacency matrix columns, once more understanding that symmetry means row outcomes are exact parallels. Neither the Chicago nor the North Carolina empirical examples contain zero eigenvalues, and their PROC TRANSREG [35] results yield no "less than full rank model" (the PROC TRANSREG message) warning.

Identifying 2-tuples is relatively easy and efficient for an undirected connected planar graph G_n. Since its sparse binary 0-1 adjacency matrix may be stored as a list of no more than $6(n - 2)$ pairs of edges/links, rather than either a full n^2 or no-self-loops $n(n - 1)$ set of pairings, a fast comparison of the position of ones in two columns is possible. These revelations, which primarily are the ones Griffith and Luhanga [12] report, can serve as a check for generated regression disclosures. Sylvester's [43] matrix algebra inertia theorem transfers these linear combinations from matrix $\mathbf{C}$ to matrix $\mathbf{W}$.

Figure 2 portrays the SAS output for the Texas empirical example. The order of column appearances in an equation is irrelevant, although switching them between the right- and left-hand sides of the equations can alter the itemized linear combination coefficients, if only in terms of their signs. The seven 2-tuples here corroborate the fast brute force pairwise

comparison matchings outcome based upon a sparse adjacency matrix file. In addition, PROC TRANSREG [35] identifies the columns constituting the exclusive 3-tuple (i.e., {C_{1594}, C_{1668}, C_{2451}}) and the two 4-tuple (i.e., {C_{3263}, C_{3264}, C_{3265}, C_{3270}} and {C_{4842}, C_{4847}, C_{4848}, C_{4849}}) subsets.

The coefficient for Identity(COL645) was set to zero due to a linear dependency among the transformed variables.	
Identity(COL645) =	Identity(COL643)

The coefficient for Identity(COL1156) was set to zero due to a linear dependency among the transformed variables.	
Identity(COL1156) =	Identity(COL420)

The coefficient for Identity(COL1668) was set to zero due to a linear dependency among the transformed variables.	
Identity(COL1668) =	1.0000 * Identity(COL1594) + 1.0000 * Identity(COL2451)

The coefficient for Identity(COL3265) was set to zero due to a linear dependency among the transformed variables.	
Identity(COL3265) =	- 1.0000 * Identity(COL3263) + 1.0000 * Identity(COL3264) + Identity(COL3270)

The coefficient for Identity(COL3722) was set to zero due to a linear dependency among the transformed variables.	
Identity(COL3722) =	Identity(COL2260)

The coefficient for Identity(COL4842) was set to zero due to a linear dependency among the transformed variables.	
Identity(COL4842) =	Identity(COL4847) - Identity(COL4848) + Identity(COL4849)

The coefficient for Identity(COL4918) was set to zero due to a linear dependency among the transformed variables.	
Identity(COL4918) =	Identity(COL4915)

The coefficient for Identity(COL4944) was set to zero due to a linear dependency among the transformed variables.	
Identity(COL4944) =	Identity(COL4943)

The coefficient for Identity(COL5182) was set to zero due to a linear dependency among the transformed variables.	
Identity(COL5182) =	Identity(COL5178)

The coefficient for Identity(COL5233) was set to zero due to a linear dependency among the transformed variables.	
Identity(COL5233) =	Identity(COL1594)

Figure 2. Partial SAS PROC TRANSREG output for the Table 2 Texas empirical example; matrix column (designated by COL#, where # denotes column position) linear combinations for each of the matrix's zero eigenvalues. The left-hand table column contains particular redundant matrix columns, and the right-hand expression is each redundant column's linear combination.

Figure 3 portrays part of the SAS output for the Syracuse empirical example. The near-zero coefficients (e.g., all ± 0.000197 or ± 0.000031) are unessential (i.e., attributable to numerical imprecision), whereas the few ± 1.0000 coefficients indicate the latent subset: C_{55}, C_{97}, and C_{141}. The term "Identity" appears in this computer hardcopy since PROC TRANSREG [35] enables and implements variable transformations, with this particular formulation option instructing this procedure to retain the initial untransformed variables themselves.

The coefficient for Identity(COL97) was set to zero due to a linear dependency among the transformed variables.	
Identity(COL97) =	1.0000 * Identity(COL55) + 1.0000 * Identity(COL141) + 0.000031 * Identity(COL177) - 0.000031 * Identity(COL179) + 0.000031 * Identity(COL196) + 0.000031 * Identity(COL270) - 0.000031 * Identity(COL271) - 0.000031 * Identity (COL314) + 0.000031 * Identity(COL318) + 0.000197 * Identity(COL327) + 0.000197 * Identity(COL376) - 0.000197 * Identity(COL379) - 0.000197 * Identity(COL409) + 0.000197 * Identity(COL424) - 0.000197 * Identity(COL439) + 0.000197 * Identity(COL440) - 0.000197 * Identity(COL465) + 0.000197 * Identity(COL466) - 0.000031 * Identity (COL470) + 0.000197 * Identity(COL490) - 0.000197 * Identity(COL492) - 0.000031 * Identity(COL508) + 0.000031 * Identity(COL578) + 0.000031 * Identity(COL751) - 0.000031 * Identity(COL950)

Figure 3. Partial SAS PROC TRANSREG output for the Table 2 Syracuse empirical example; matrix column (designated by COL#, where # denotes column position) linear combinations for each of the matrix's zero eigenvalues The left-hand table column contains particular redundant matrix columns, and the right-hand expression is each redundant column's linear combination.

Figure 4 demonstrates the failure of PCA to always produce simple linear combinations (i.e., simple structure) due to the non-uniqueness of the eigenspaces involved. Multiple linear regression allocates the repeated adjacency matrix column C_{1460} in this example to

only one subset linear combination, whereas PCA introduces some confusion by allocating it to the various subsets to which it also could belong. In other words, the linear regression approach employing a sweep operator furnishes a more parsimonious solution here. One important aspect of this exemplification is that PCA muddling still materializes in the presence of a relatively large n (i.e., 3408) coupled with a quite small number of redundant columns (i.e., eight).

4.4. Discussion: Selected Sweep Operator Properties

Computational complexity and robustness are two salient properties beyond advantages established via competitor algorithm comparisons that demand commentary here supplementing that already appearing in Section 4.1, if for no other reason than to better contextualize the sweep operator within the existing literature. In statistics, although Goodnight [36] popularized this operator, Dempster [44] essentially introduced it to this discipline, with Andrews [45] evaluating it in terms of robustness; noteworthy is that non-statisticians call it by other names, including the principal pivot transform [46], gyration [47], and exchange [48]. Two of its close relatives are the (previously mentioned) Gauss-Jordan elimination ([42], p. 166) and the forward Doolittle ([36], pp. 151–152), an execution step in the LU decomposition, procedures. Thus, on the one hand, its computational complexity—the solution time and memory/storage space resources required by it to solve a specific zero eigenvalue problem—are not extreme, relatively speaking. For example, SAS reports total processing CPU execution time of 5:22.59 for n = 7249 (the largest graph analyzed in this paper) on a desktop computer with four CPUs. It quickly regresses all variables, such as the second through nth columns, against one specified variable, such as the first column, of a graph adjacency matrix, obtaining in a single execution linear regression computations and all perfect linear combinations of columns that are present. Therefore, it is relatively efficient and scalable to the same degree as any linear regression algorithm; linear regression scalability is not computationally heavy.

On the other hand, computer algorithm robustness here—a desirable characteristic ensuring the sweep algorithm maintains reliability and resilience across differing circumstances—refers to its ability to consistently and accurately uncover zero eigenvalue linear combinations of covariates in the presence of various types of disturbances, errors, and/or unexpected inputs (e.g., the random planar graphs of Appendix A). Accordingly, the criteria it must satisfy should ensure that it is able to identify perfect linear combinations that are: (1) for multiple replications of a single adjacency matrix column/row (see Corollary 2); (2) smaller ones embedded in larger ones (e.g., see Table 2); (3) different but sharing columns/rows without double-counting them (e.g., see Figure A1); (4) not confused by numerical precision/rounding error (e.g., see Figure 3); and, (5) for excessively complicated outlier graph structures (e.g., Corollary 2) involving very many columns/rows (see Figure A3). In other words, the sweep operator is resistant since it continues to Identify all latent zero eigenvalue linear combinations even if a small fraction of a graph adjacency matrix undergoes detrimental alterations, or the recovered linear combination is tremendously sophisticated (see Appendix A). With regard to the preceding fourth point, PROC TRANSREG [35] often appears to be more resilient than PROC REG [35], at least in their SAS enactment. However, the reported trivial regression coefficients are so small that they become obvious precision errors (see Section 4.2) to the naked eye.

The coefficient for Identity(COL89) was set to zero due to a linear dependency among the transformed variables.	
Identity(COL89) =	Identity(COL74)
The coefficient for Identity(COL1327) was set to zero due to a linear dependency among the transformed variables.	
Identity(COL1327) =	-Identity(COL1307) + Identity(COL1310) + Identity(COL1336)
The coefficient for Identity(COL1446) was set to zero due to a linear dependency among the transformed variables.	
Identity(COL1446) =	1.0000 * Identity(COL1448) + 1.0000 * Identity(COL1460)
The coefficient for Identity(COL1505) was set to zero due to a linear dependency among the transformed variables.	
Identity(COL1505) =	Identity(COL1460)
The coefficient for Identity(COL1581) was set to zero due to a linear dependency among the transformed variables.	
Identity(COL1581) =	Identity(COL1562)
The coefficient for Identity(COL2155) was set to zero due to a linear dependency among the transformed variables.	
Identity(COL2155) =	Identity(COL2089)
The coefficient for Identity(COL2746) was set to zero due to a linear dependency among the transformed variables.	
Identity(COL2746) =	Identity(COL1599)
The coefficient for Identity(COL3152) was set to zero due to a linear dependency among the transformed variables.	
Identity(COL3152) =	Identity(COL1345)

preceding linear combination	3rd	6th	5th	8th	7th	1st		2nd
C74	0	0	0	0	0	0.707	0	0
C89	0	0	0	0	0	−0.707	0	0
C1307	0	0	0	0	0	0	0	−0.316
C1310	0	0	0	0	0	0	0	0.447
C1327	0	0	0	0	0	0	0	−0.632
C1336	0	0	0	0	0	0	0	0.548
C1345	0	0	0	0.707	0	0	0	0
C1446	**−0.267**	0	0	0	0	0	**−0.707**	0
C1448	0.189	0	0	0	0	0	0.500	0
C1460	0.756	0	0	0	0	0	0	0
C1505	**−0.567**	0	0	0	0	0	**0.500**	0
C1562	0	0	0.707	0	0	0	0	0
C1581	0	0	−0.707	0	0	0	0	0
C1599	0	0	0	0	0.707	0	0	0
C2089	0	0.707	0	0	0	0	0	0
C2155	0	−0.707	0	0	0	0	0	0
C2746	0	0	0	0	−0.707	0	0	0
C3152	0	0	0	−0.707	0	0	0	0

Figure 4. Partial SAS PROC TRANSREG (top) and PROC PRINCOMP (bottom) output for the Table 2 US HAS empirical example; matrix column (designated by COL#, where # denotes column position) linear combinations for each of the matrix's zero eigenvalues. The left-hand table column contains particular redundant matrix columns, and the right-hand expression is each redundant column's linear combination, based upon the last eight extracted components (a la [32]). Bold font denotes compound loadings; underlining denotes incorrect loadings.

5. Approximating the Eigenvalues of Matrix W

The spatial statistical problem of interest here is estimation of the auto-normal model normalizing constant (e.g., [49]), the Jacobian of the transformation from a spatially auto-

correlated to a spatially unautocorrelated mathematical space using calculus terminology. This quantity almost always is a function of the undirected connected planar graph dual row-standardized adjacency matrix **W** eigenvalues for a configuration of polygons forming a two-dimensional surface partitioning. Such eigenvalue calculations are possible for quite large n, In the lower 10,000s, but become Impossible beyond the frontiers of computing power, a constantly expanding resource size. This normalizing constant is the existing work adaption target for the matrix inertia refinement this paper disseminates.

An undirected star graph, $K_{1,n}$, perceptively illustrates the problem here in its extreme: its eigenvalues are 1, n − 2 0s, and −1 for matrix **W**, whereas they are $\pm\sqrt{1^T\mathbf{C}1}$ and n − 2 0s for matrix **C**. In this kind of situation, Bindel and Dong [15], for example, show how eigenvalue frequency distributions can become dominated by a spike at zero (also see Figure 5b,d). A null eigenvalue fails to add any amount to the sum of the n eigenvalues, regardless of the value of n for star graph G_n. For all n > 1, the variance of these eigenvalues is $\mathbf{1}^T\mathbf{D}^{-1}\mathbf{C}\,\mathbf{D}^{-1}\mathbf{1} = 2$. More realistic graphs show that although $\lambda_1(\mathbf{W}) = 1$ remains unchanged, adding redundant links that introduce zero eigenvalues can alter $\lambda_n(\mathbf{C})$ and $\lambda_n(\mathbf{W})$ as well as some or all of the intermediate eigenvalues (see Table 2). Furthermore, the matrix **W** eigenvalues have a more limited range (Table 2) that causes their density to increase faster than it does for their binary 0-1 parent matrix **C** eigenvalues.

Estimating a complete set of n eigenvalues with a method of moments type technique means spreading the variance over all n values while constraining their sum to equal zero. Griffith and Luhanga [12] address the need to include matrix inertia information in this process: zero eigenvalues should not have non-zero allocations during an approximation routine, and the number of positive and negative eigenvalues needs differentiation in asymmetric eigenvalue distributions since the same cumulative sum almost certainly is spread over fewer individual λs in one of the two instances (almost always the positive domain). Taliceo and Griffith [25] explore this latter trait. Incorporating these two facets for remotely sensed data (e.g., see [13]) analyses renders remarkably accurate eigenvalue approximations for that spatial statistical landscape [14], in part due to its symmetric eigenvalue distribution for the often employed rook adjacency definition. In contrast, ignoring these two influential facets tends to cause typical administrative districts (e.g., counties, census tracts) forming irregular surface partitioning to approximate eigenvalues displaying a discontinuity separation gap at zero (e.g., Figure 5a,c; a finding reported by Griffith in earlier publications). Improved adoption of these two eigenvalue properties should enable a better approximation fabrication for them that exhibits a smoother transition between positive and negative λs.

5.1. Preliminary Eigenvalue Approximation Steps

Eigenvalue approximation can follow a three-step sequence of directives. The first step approximates the extreme eigenvalues of matrices **C** and **W** [16,50]. A formulated algorithm employing solely the sparse version of an adjacency matrix (i.e., a sequential listing of only the row-column cells containing a one) quickly produces these values, with extreme accuracy for both $\lambda_1(\mathbf{C})$ and $\lambda_n(\mathbf{W})$ (Table 3); the most relevant of these two values for spatial statistics/econometrics is $\lambda_n(\mathbf{W})$. This algorithm builds its $\lambda_n(\mathbf{C})$ approximation with the normalized principal eigenvector, $\mathbf{E}_1$, estimated during the iterative calculation of $\lambda_1(\mathbf{C})$. Next, this algorithm builds its $\lambda_n(\mathbf{W})$ estimate with the normalized eigenvector $\mathbf{E}_n$ approximated during the iterative calculation of $\lambda_n(\mathbf{C})$. Given that $\lambda_1(\mathbf{W}) \equiv 1$ is known theoretically, the appealing outcome is that $\lambda_n(\mathbf{W})$ is knowable for massively large matrices.

The second step of the proposed eigenvalue approximation procedure determines the number of zero eigenvalues. The sparse version of an adjacency matrix also allows a quick identification of all 2-tupes generating zero eigenvalues (Table 4), which in general, and as a rational expectation, seem to account for a preponderance of zero eigenvalues (e.g., [12], Tables 1 and 2). However, even the triplet computations—executing n(n − 1)(n − 2) trivariate regressions—for the larger of the seven specimen landscapes studied in this paper are too computer intensive for practical purposes. Moreover, presently the best estimate of

n_0 is the number of 2-tuples for massively large square matrices. In addition, this approach fails to be effective when extended to more general real matrices since their non-sparseness requires a replacement of each $n(n-1)$ simple comparison with a bivariate regression; columns/rows can be proportional rather than just equal. This daunting computational demand argues for keeping the original PROC TRANSREG [35] operationalization with its $n(n-1)$ sweep operations.

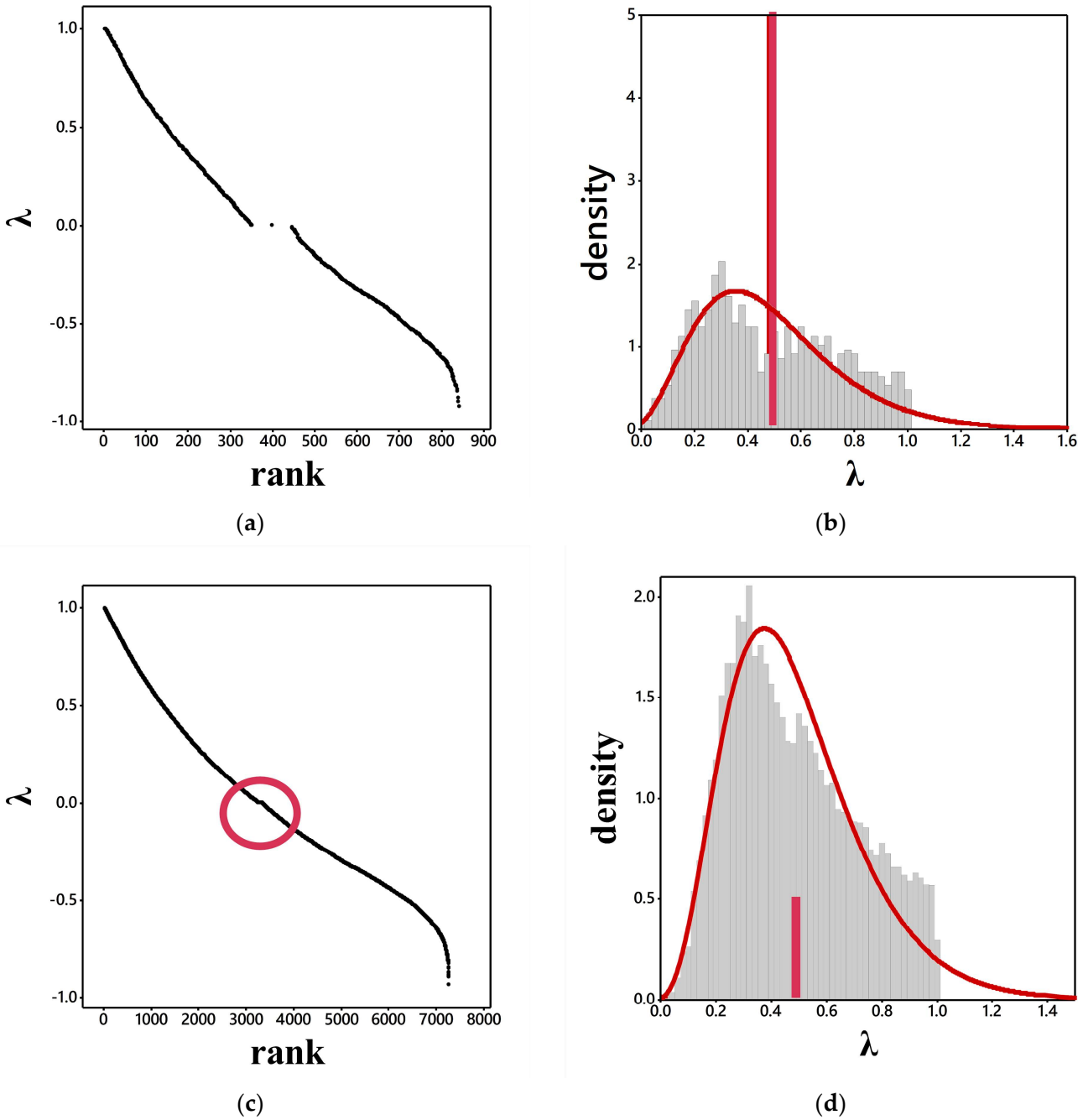


Figure 5. Specimen empirical adjacency matrix-based **W** eigenvalues with a more conspicuous zero value presence. Top left (**a**): England descending rank ordering. Top right (**b**): England histogram with a superimposed gamma distribution (red curve line; see Figure 1) and a highlighted zero value frequency (red bar). Bottom left (**c**): Syracuse descending rank ordering. Bottom right (**d**): Syracuse histogram with a superimposed gamma distribution (red curve line; see Figure 1) and a highlighted zero value frequency (red bar).

Table 3. Matrix **C** and **W** extreme eigenvalue approximations for the chosen specimen geographic landscape surface partitionings.

Extreme λs	England	Chicago	Edmonton	North Carolina	US HSA	Texas	Syracuse
$\lambda_1(\mathbf{C})$	6.07179	6.21901	6.95071	6.27921	7.82014	7.43187	7.37507
$\lambda_n(\mathbf{C})$	−4.1**7990** (−4.18527)	−3.**73161** (−3.34868)	−**3.88935** (−4.32407)	−3.75**641** (−3.75393)	−4.45**320** (−4.45895)	−4.52**253** (−4.52336)	−**3.76275** (−5.02457)
$\lambda_n(\mathbf{W})$	−0.92541	−0.85971	−0.77382	−0.70273	−0.95668	−0.84387	−0.93586

NOTE: λ denotes eigenvalue; actual $\lambda_n(\mathbf{C})$ values in parentheses, with deviating decimal places in bold font; all $\lambda_1(\mathbf{C})$ and $\lambda_n(\mathbf{W})$ reported decimal place values agree.

Table 4. Relevant eigenvalue approximation measures for the selected specimen graphs.

Quantity	Category	England	Chicago	Edmon-Ton	North Carolina	US HSA	Texas	Syracuse
tuple small p	2	79	0	23	0	6	7	48
	3	6 †	0	9 †	0	1	2 †	7
inertia	n_+	349	891	895	909	1442	2205	3242
	n_0	93	0	41	0	8	10	70
	n_-	397	1176	1162	1286	1958	3050	3937
exponent	γ	1.4178	1.5656	1.2890	1.2279	2.2820	1.6077	1.8094
	θ	1.3732	1.4636	1.4546	1.4297	1.6744	1.4771	1.6084
λ > 0 bivariate regression	intercept	0.0504	0.0361	0.0348	0.0283	0.0643	0.0419	0.0518
	slope	0.9429	0.9700	0.9643	0.9745	0.9632	0.9684	0.9520
	R^2	0.9973	0.9993	0.9992	0.9995	0.9966	0.9988	0.9979

NOTE: λ denotes eigenvalues; 3- and 4-tuples computer execution speed is too slow for massively large graphs; the bivariate regression intercept and slope are close to their respective ideal values of 0 and 1. † includes identified overlap matchings with a 2-tuple.

The third step approximates the non-zero eigenvalues with a method of moments type of calibration methodology, given that their mean (i.e., zero) and variance (i.e., $\mathbf{1}^T\mathbf{D}^{-1}\mathbf{C}\,\mathbf{D}^{-1}\mathbf{1}/n$) are either known or easily computable, and hence available. A favored specification for this task exploits eigenvalue ordering by calibrating expressions such as

$$\left.\begin{array}{l} \lambda(\mathbf{W}) > 0 : [(n_+ + 1 - r_i)/n_+]^{\gamma} \\ \lambda(\mathbf{W}) < 0 : -[(|\lambda_n|\, r_i)/n_-]^{\theta} \end{array}\right\}, \tag{4}$$

where r_i denotes the descending ordering rank of the i^{th} eigenvalue after setting aside the set of zero eigenvalues [which subsequently are merged with the approximations generated by Equation (4)], and exponents γ and θ that match the approximated eigenvalue mean and square root of its sum of squares as closely as possible (guided by the method of moments estimation strategy, and using a mean squared error criterion); all manufactured values of expression (4) belong to the interval [0, 1], as do the absolute values of their corresponding actual eigenvalues, with $\gamma > 1$ and $\theta > 1$ indicative of variance inflation needing removal, and $\gamma < 1$ and $\theta < 1$ indicative of deflated variance needing to be restored. The differences between γ and θ appearing in Table 4 confirm an asymmetric distribution of eigenvalues, whereas the positive eigenvalues bivariate regressions corroborate a well-aligned linear trend with their adjusted rankings. This linear trend tendency is capable of facilitating the drafting of an improved upper bound for the sum of k positive eigenvalues.

5.2. An Instructive Eigenvalue Approximation Assessment

This section summarizes evaluations of two specimen landscapes, namely North Carolina (no zero and 41.4% positive eigenvalues) and Syracuse (48 of 70 easily detectable zero and 45.2% positive eigenvalues). The working assumptions are: (1) the number of 2-tuple zero eigenvalues accurately counts n_0 (the calculable count for massively large G_n), the nearest integer 45% of the non-zero eigenvalues accurately counts n_+ (roughly the mid-point of large-landscape simulation findings reported in [25]), and equation (4) approximates a total set of n eigenvalues. Figure 6 portrays the Jacobian plots across the auto-normal model spatial autocorrelation parameter value contained in its feasible parameter space interval, namely, $1/\lambda_n(\mathbf{W}) < \rho < 1$. These graphical findings imply that the enhanced (i.e., refined) but relatively simple eigenvalue approximations proposed in this paper are sufficiently accurate to support sound spatial autoregressive analyses. More comprehensive appraisals of this proposition merit future research attention.

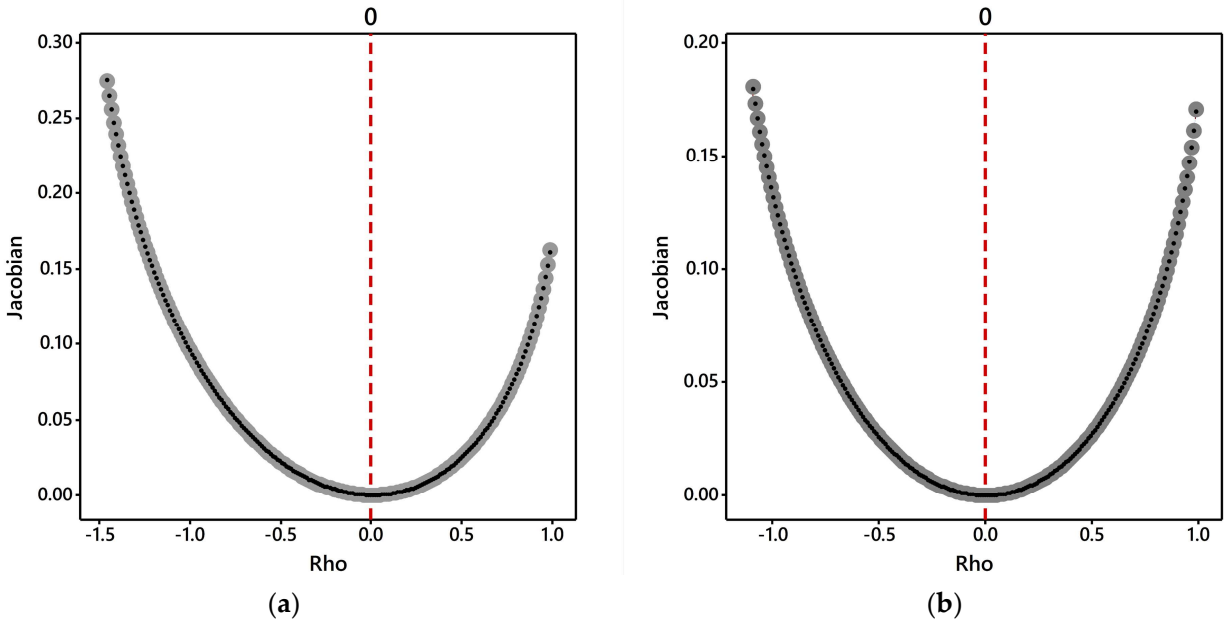


Figure 6. The approximated eigenvalues Jacobian plot (denoted by smaller solid black circles) superimposed upon the true Jacobian plot (denoted by larger solid gray circles) for an auto-normal model specification. Left (**a**): the North Carolina case (γ = 1.1526 and θ = 1.6473). Right (**b**): the Syracuse case (γ = 1.8090 and θ = 1.6227).

6. Discussion

The following two topics warrant further discussion: identifying linearly dependent subsets of columns/rows in more general matrices, and application of the eigenvalue approximation technique to a very sizeable adjacency matrix.

Table 5 presents a posed problem, one that contains column c_6 that is a trivial zero eigenvalue source, and hence ignored in terms of constructing linear combinations with it. The non-square matrix requires analysis of both its original columns and original rows in their matrix transformed styles. PROC TRANSREG [35] identifies the linearly dependent subsets of columns and rows appearing in Figure 7, with a count of five. Column c_1 and transposed column (i.e., row) r_1 serve as the dependent variables in their respective linear regression analyses. Accordingly, since PROC TRANSREG [35] only uncovers linearly dependent subsets in a collection of covariates, both c_1 and r_1 also have to be analyzed in a stepwise PROC REG [35], which exposes the $\{r_1, r_2, r_5\}$ linearly dependent subset (e.g., the root mean squared error, _RMSE_, is zero, and the linear regression multiple correlation, _RSQ_, is 1).

Table 5. The matrix appearing in both https://stackoverflow.com/questions/11966515/find-all-lineary-dependent-subsets-of-vectors (accessed on 24 October 2023) and https://math.stackexchange.com/questions/182753/find-all-linearly-dependent-subsets-of-this-set-of-vectors (accessed on 24 October 2023), with its rows labeled to indicate its transpose.

	$\mathbf{c_1}$	$\mathbf{c_2}$	$\mathbf{c_3}$	$\mathbf{c_4}$	$\mathbf{c_5}$	$\mathbf{c_6}$
$\mathbf{r_1}$	1	1	1	0	1	0
$\mathbf{r_2}$	0	0	1	0	0	0
$\mathbf{r_3}$	1	0	0	0	0	0
$\mathbf{r_4}$	0	0	0	1	0	0
$\mathbf{r_5}$	1	1	0	0	1	0
$\mathbf{r_6}$	0	0	1	1	0	0
$\mathbf{r_7}$	1	0	1	1	0	0

NOTE: c_j denotes column j, and r_j denotes row j; transposing the rows and columns produces the binary rows matrix.

The coefficient for Identity(c5) was set to zero due to a linear dependency among the transformed variables.

Identity(c5) =	Identity(c2)

The TRANSREG Procedure

The coefficient for Identity(r6) was set to zero due to a linear dependency among the transformed variables.

Identity(r6) =	Identity(r2) + Identity(r4)

The coefficient for Identity(r7) was set to zero due to a linear dependency among the transformed variables.

Identity(r7) =	Identity(r2) + Identity(r3) + Identity(r4)

The coefficient for Identity(r7) was set to zero due to a linear dependency among the transformed variables.

Identity(r7) =	Identity(r3) + Identity(r6)

Obs	_MODEL_	_TYPE_	_DEPVAR_	_RMSE_	Intercept	r2	r5	r6	r7	r1	_IN_	_P_	_EDF_	_RSQ_
1	MODEL1	PARMS	r1	0	0	1	1	.	.	-1	2	3	3	1

Figure 7. Linearly dependent subsets in a non-square matrix from SAS PROC TRANSREG.

The 2010 US census tract polygons covering the entire coterminous continental part of that country furnishes a sizeable empirical example of an undirected connected planar G_n, with n = 72,538, $\mathbf{1}^T\mathbf{C1}$ = 402,830, MAX{n_i} = 48, $\mathbf{1}^T\mathbf{D}^{-1}\mathbf{C}\,\mathbf{D}^{-1}\mathbf{1}$ = 12985.54, $\lambda_1(\mathbf{C})$ = 8.63582, and $\lambda_n(\mathbf{W})$ = −0.89732—desktop computer—Intel(R) Xeon(R) CPU, E5640 @ 2.67GHz and 2.66GHz (2 processors), 24.0GB RAM, 64-bit OS, x64-based processor, Windows 10 Enterprise—execution time for calculating these last two quantities was less than 40 s. Crafting this geographic polygons configuration graph constructed for the coterminous US census tracts irregular surface partitioning involved modifying an ESRI ArcMAP .shp file with such manual adjustments as linking islands to the mainland through ferry routes. The simple 2-tuple matching analysis retrieves 32 zero eigenvalue sources. Consequently, the basic input for approximating the matrix **W** eigenvalues, and hence the needed Jacobian term for spatial statistical/econometric autoregressive analysis, is not only feasible, but also available here. This exemplification once again illustrates how the matrix inertia refinement promulgated in this paper is adaptable to existing work.

7. Conclusions and Implications

In conclusion, this paper set out to establish solutions to two seemingly unsolved or poorly solved linear algebra problems contained in topics of more general interest, one frequently mentioned on the internet, and the other becoming of increasing interest and concern as geospatial analysts push their empirical spatial statistical/economics analyses to larger and larger georeferenced datasets (sometimes in the realm of machine learning and data mining) and contemporary social network analysts continue to adapt spatial analysis techniques to their often massively larger—the KONECT archive project (http://konect.cc/ (accessed on 24 October 2023) makes 1326 social networks available to the public, ranging in node count from 16 to 105.2 million, of which just 543 have fewer than 10,000 nodes, and whose size frequencies closely conform to a log-normal distribution—denser, and almost certainly non-planar graph generated adjacency matrices. The first is the enumeration of all non-trivial linearly dependent column/row subsets in a matrix. One on-line commentator mentions that this problem most likely requires a supercomputer for even modestly large matrices. The regression-based solution promoted in this paper successfully, efficiently, and effectively handles matrices with dimensions in the 1000s, employing a laptop personal computer using the following processor: Intel(R) Core(TM) i5-2520M CPU @ 2.50 GHz; this machine has 8.00 GB of installed RAM, and runs with a 64-bit operating system, clearly not state-of-the-art computer or supercomputer technology.

The second problem of approximating eigenvalues for massively large matrices builds upon an evolving foundation, with demand for its solution being prompted by the existence of a parallel solution for remotely sensed satellite data whose G_n adjacency matrix size often is in the 10,000,000s [14]. This magnitude echoes that for the aforementioned KONECT project social network entries in excess of one million nodes (reflecting a practical upper limit size), which number 18 and have a median of 30 million nodes, and whose Jacobian term requires the irregular tessellation category of eigenvalue approximations outlined in this paper. Another imperative application arena is the present-day expansion of data mining and machine learning analyses to georeferenced, social media, and other substantially large datasets. The solution advocated for in this paper streamlines earlier results by exploiting extremely large adjacency matrix inertia.

The fundamental implications launched by this paper may be summarized as a combination of the preceding and following conjectures:

Conjecture 2. *Let* $G_n = (V, E)$ *be a simple connected undirected planar graph with n vertices* $V = \{v_1, \ldots, v_n\}$, *and* $m \leq n(n-1)/2$ *edges* $E = \{e_{ij}$ *linking vertices* v_i *and* v_j: $i = 1, 2, \ldots, n$ *and* $j = 1, 2, \ldots, n; i \neq j\}$. *If* G_n *has no* K_4 *subgraphs, then the maximum value of* n_- *is* $2n/3$.

Conjecture 3. *Let* $G_n = (V, E)$ *be a simple connected undirected planar graph with n vertices* $V = \{v_1, \ldots, v_n\}$, *and* $m \leq n(n-1)/2$ *edges* $E = \{e_{ij}$ *linking vertices* v_i *and* v_j: $i = 1, 2, \ldots, n$ *and* $j = 1, 2, \ldots, n; i \neq j\}$. *The sum of the* k *largest matrix* **W** *positive eigenvalues least upper bound for a self-loopless* G_n*'s adjacency matrix is* $\sum_{i=1}^{k} [(n+1-ri)/n+]\gamma + \varepsilon$ *for some suitably small* ε.

A more general implication is that geographical analysis matrices continue to be a fertile research subject. Concerted quantitative geography efforts in this area began in the 1960s, and have become progressively more sophisticated with the passing of time.

Finally, future research needs to upgrade the estimation of $\lambda_n(\mathbf{C})$, enhance the accurate determination of n_+ (and hence n_-), expand the quick calculation of p-tuple linearly dependent subsets of adjacency matrix columns/rows greater than $p = 2$, accurately quantify the negative eigenvalue part of Equation (4) with a nonlinear regression equation specification, and refine and then convert the three stated conjectures into theorems with proofs. Future research also should explore furthering the effectiveness of the proposed sweep-algorithm-based solution utilized here, especially for much larger n, by reformulating it to be optimal for the nullity problem addressed here, rather than for its current generic multiple linear regression implementation, allowing a meaningful reduction in its customized computational complexity. One way to achieve this particular goal might be by more accurately addressing

the underlying structure of empirical planar graphs, perhaps utilizing special properties of the Laplacian version of their adjacency matrices to attain faster execution times.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: For the pseudorandom number generators used to create test data see https://documentation.sas.com/doc/en/pgmsascdc/9.4_3.2/lefunctionsref/p0fpeei0opypg8n1b06qe4r040lv.htm (accessed on 19 May 2022). Empirical adjacency matrices available from the author upon request.

Conflicts of Interest: The University of Texas at Dallas subscribes to an annual license for the latest version of Mathematica. The University of Texas System subscribes to an annual license for SAS. Otherwise, the author declares he has no personal circumstances or interest that may be perceived as inappropriately influencing the representation or interpretation of reported research results. Since the research summarized here had no funding support, no funders had a role in: the design of the study; collection, analyses, or interpretation of data; writing of the manuscript; or, decision to publish the results.

Appendix A. Selected Supplemental Specimen Geographic Landscapes

A database comprising roughly seven dozen empirical surface partitioning dual graphs studied elsewhere (e.g., [51]) yields a much smaller number of graphs with zero eigenvalues. The assortment in Table 1 spans a wade range of n; Table A1 reports selected summary statistics for the remaining ten. Figure A1 also enumerates the source linear combinations uncovered by a linear regression sweep algorithm.

Table A1. Selected chosen specimen geographic landscape surface partitioning descriptors.

Geographic Landscape	n	Nullity	K_4 Count	Matrix C			Matrix W	
				$\lambda_1(C)$	$\lambda_n(C)$	1^TC1	$\lambda_2(W)$	$\lambda_n(W)$
Columbus, OH 1980 ‡	49	1	0	5.908	−3.097	232	0.963	−0.651
Winnipeg 1971 §	101	1	1	5.738	−3.279	498	0.968	−0.727
Stockholm, Sweden	119	1	1	6.016	−3.420	582	0.987	−0.893
Henan, China	130	1	5	6.111	−3.029	668	0.980	−0.894
Brazilian Amazon †	323	1	12	6.276	−3.430	1608	0.996	−0.911
Minnesota tree stands	513	5	29	6.118	−4.127	2462	0.994	−0.763
Germany kreise 1930 *	743	28	7	6.271	−3.412	3706	1.000 ‡	−0.793
Sheffield, UK	930	1	33	6.322	−3.940	4708	0.999	−0.833
Chicago 1990 ‡	1754	1	23	6.310	−3.605	8820	0.998	−0.861
US counties ‡	3111	3	18	6.264	−3.426	17,508	0.999	−0.814

NOTE: n denotes the number of graph G_n nodes; K_4 denotes the well-known completely connected G_4; λ denotes eigenvalue; nullity denotes the number of zero eigenvalues; § Statistics Canada 1971 census tracts; ‡ this surface partitioning comprises two separated sub-graphs of sizes 35 and 708; † Amazon basin municipalities; ‡ US Census Bureau 1980 or 1990 census tracts. * O′ Loughlin, J., C. Flint and L. Anselin. 19943. The Geography of the Nazi Vote: Context, Confession and Class in the Reichstag Election of 1930, *Annals*, American Association of Geographers, 84 (3): 351–380; map on p. 361.

The TRANSREG Procedure

The coefficient for Identity(c49) was set to zero due to a linear dependency among the transformed variables.

Identity(c49) =	Identity(c45)

planar graph	*zero eigenvalue linear combinations of columns equivalencies*		
Columbus, OH 1980	C45 = C49		
Winnipeg 1971	C67 = C69		
Stockholm, Sweden	C92 = C93 + C104		
Henan, China	C80 = C79 + C87		
Brazilian Amazon	C277 = C287		
Minnesota tree stands	C127 = C132 C260 = C268 C406 = C433	C41 = C23 + C36 C257 = C126 − C127 − C161 + C219 + C296	
Germany kreise 1930	C113 = C114 C182 = C193 C251 = C282 C251 = C288 C261 = C276 C263 = C721 C263 = C722 C263 = C280 C263 = C283	C270 = C271 C304 = C312 C369 = C370 C369 = C371 C508 = C618 C508 = C626 C611 = C637 C607 = C645 C607 = C656 C624 = C636	C633 = C638 C630 = C652 C640 = C651 C649 = C667 C665 = C668 C700 = C701 C700 = C702 C700 = C703 C679 = C683 + C684
Sheffield, UK	C222 = C224 + (C219 + C226 + C231 + C233 + C237 − C223 − C228 − C236)/2		
Chicago 1990	C225 = c256		
US counties	C1511 = c1524	C1611 = c1616	C1613 = c1624

Figure A1. Partial SAS PROC TRANSREG output for Table A1 examples; matrix column (designated by C#, where # denotes column position) linear combinations for each of the matrix's zero eigenvalues. The top panel contains illustrative SAS output. In the bottom panel, the left-hand table column contains particular redundant matrix columns, and the right-hand expression is each redundant column's linear combination.

Appendix B. A Conrolled Experiment: A P-By-P Regular Square Tessellation Forming a Complete Rectangular Region, Rook Adjacency Definition, with a (P, P) Node Corner Landscape Central Node K-Star Subgraph

This particular graph contains a near-block diagonal P-by-P regular square tessellation plus a near-K-star supplemental subgraph (i.e., both subgraphs share a node, causing them to overlap). The expected number of zero eigenvalues for the two subgraphs in isolation is (P + K − 2). Corollary 2 furnishes this counting result, explicitly indicating that P − 1 linear combinations come from the core adjacency matrix, and K − 1 come from simple pairwise node redundancies.

Figure A2 displays the outcome for the P = 9 and K = 19 case. Its anticipated number of zero eigenvalues is 8 + 18 = 26. Given that the corner tessellation node adds two additional links to the start subgraph, 18 of these zero eigenvalues relate to the same redundancy. The remainder relate to more complex uncorrupted linear combinations of columns/rows arising from the set of eigenvectors that, in isolation, are functions of $\mathrm{SIN}[\pi j/(P+1)] \times \mathrm{SIN}[\pi k/(P+1)]$. An interested reader can easily replicate this finding.

COL73 =	−COL9 + COL17 − COL25 + COL33 − COL41 + COL49 − COL57 + COL65
COL74 =	−COL8 + COL16 + COL18 − COL24 − COL26 + COL32 + COL34 − COL40 − COL42 + COL48 + COL50 − COL56 − COL58 + COL64 + COL66
COL75 =	−COL7 + COL15 + COL17 − COL23 − COL25 − COL27 + COL31 + COL33 + COL35 − COL39 − COL41 − COL43 + COL47 + COL49 + COL51 − COL55 − COL57 − COL59 + COL65 + COL67
COL76 =	−COL6 + COL14 + COL16 − COL22 − COL24 − COL26 + COL30 + COL32 + COL34 + COL36 − COL38 − COL40 − COL42 − COL44 + COL46 + COL48 + COL50 + COL52 − COL56 − COL58 − COL60 + COL66 + COL68
COL77 =	−COL5 + COL13 + COL15 − COL21 − COL23 − COL25 + COL29 + COL31 + COL33 + COL35 − COL37 − COL39 − COL41 − COL43 − COL45 + COL47 + COL49 + COL51 + COL53 − COL57 − COL59 − COL61 + COL67 + COL69
COL78 =	−COL4 + COL12 + COL14 − COL20 − COL22 − COL24 + COL28 + COL30 + COL32 + COL34 − COL38 − COL40 − COL42 − COL44 + COL48 + COL50 + COL52 + COL54 − COL58 − COL60 − COL62 + COL68 + COL70
COL79 =	−COL3 + COL11 + COL13 − COL19 − COL21 − COL23 + COL29 + COL31 + COL33 − COL39 − COL41 − COL43 + COL49 + COL51 + COL53 − COL59 − COL61 − COL63 + COL69 + COL71
COL80 =	−COL2 + COL10 + COL12 − COL20 − COL22 + COL30 + COL32 − COL40 − COL42 + COL50 + COL52 − COL60 − COL62 + COL70 + COL72
COL83 =	COL82
COL84 =	COL82
COL85 =	COL82
COL86 =	COL82
COL87 =	COL82
COL88 =	COL82
COL89 =	COL82
COL90 =	COL82
COL91 =	COL82
COL92 =	COL82
COL93 =	COL82
COL94 =	COL82
COL95 =	COL82
COL96 =	COL82
COL97 =	COL82
COL98 =	COL82
COL99 =	COL82
COL100 =	COL82

Figure A2. Partial SAS PROC REG output for the artificial example; matrix column (designated by COL#, where # denotes column position) linear combinations for each of the matrix's zero eigenvalues. The left-hand table column contains particular redundant matrix columns, and the right-hand expression is each redundant column's linear combination.

Table A2 furnishes randomly simulated specimen planar graph adjacency matrix results as an exploratory confirmatory exercise. The sweep algorithm successfully identifies the redundant columns/rows in every case, a count Corollary 2 proclaims.

Table A2. Simulated results for cases specified by Corollary 2.

Random Selection		Number of Nodes	P + K − 2	Number of Separate Linear Regression Sweep Algorithm Collinearities
P ∈ [6, 50]	K ∈ [2, 500]			
16	390	646	404	404
15	408	633	421	421
34	495	1651	527	527
12	312	656	322	322
39	289	1810	326	326
23	375	904	396	396
46	84	2200	128	128
37	409	1778	444	444
39	165	1686	202	202
19	11	372	28	28

NOTE: P denotes the dimensions of a P-by-P regular square tessellation; K denotes the number of nodes forming a star attached to the southeast corner of the square mesh.

Appendix C. Expanding the Graph Validation Dataset

The empirical graphs dataset employed for illustrative purposes is limited since its compilation is for surface partitionings referenced in the literature. Many of these adjacency matrices are free of zero eigenvalues, which is not uncommon in spatial analysis practice. Griffith [51] offers one way to circumvent this restriction. This appendix summarizes synthetic random planar graphs for n = 100, 500, 1000, and 2000 generated specifically for analyzing zero eigenvalues. Of 25 such generated graphs for n = 100, six have at least one zero eigenvalue: two have eight, and four have exactly one. Figure A3 displays this outcome. Of note is that its first entry has nine linear combinations for eight zero eigenvalues; the linear combination for COL91 also is part of the linear combination for COL100, and hence is for the same, not for a separate, zero eigenvalue.

None of these synthetic specimens has the star subgraph typically encountered in administrative district surface partitionings. Rather, each of the single zero eigenvalues here relates to a sizeable linear combination of other adjacency matrix columns/rows. Nevertheless, this example furnishes additional evidence about the usefulness and robustness of the regression sweep algorithm for solving this linear algebra problem. Most of the linear combinations reported in this appendix are far more complicated than those appearing in either this paper's main text or Appendices A and B.

Figure A4 furnishes several additional random planar network outcomes. The random sampling produced adjacency matrices with zero eigenvalues with the following frequencies: two out of 10 for n = 500; two out of 35 for n = 1000; and, one out of 25 for n = 2000.

COL91 =	COL10 − COL19 + COL28 − COL37 + COL46 − COL55 + COL64 − COL73 + COL82
COL92 =	COL9 − COL18 − COL20 + COL27 + COL29 − COL36 − COL38 + COL45 + COL47 − COL54 − COL56 + COL63 + COL65 − COL72 − COL74 + COL81 + COL83
COL94 =	COL7 + COL8 − COL16 − COL17 − COL18 − COL19 + COL25 + COL26 + COL27 + COL28 + COL29 + COL30 − COL34 − COL35 − COL36 − COL37 − COL38 − COL39 − COL40 + COL43 + COL44 + COL45 + COL46 + COL47 + COL48 + COL49 − COL52 − COL53 − COL54 − COL55 − COL56 − COL57 − COL58 + COL61 + COL62 + COL63 + COL64 + COL65 + COL66 + COL67 − COL71 − COL72 − COL73 − COL74 − COL75 − COL76 + COL82 + COL83 + COL84 + COL85 − COL93
COL95 =	COL6 − COL8 − COL15 + COL19 + COL24 − COL30 − COL33 + COL42 + COL50 − COL51 − COL59 + COL68 + COL71 − COL77 − COL82 + COL86 + COL93
COL96 =	COL5 + 2 * COL8 − COL14 − COL16 – 2 * COL17 – 2 * COL19 + COL23 + COL25 + 2 * COL26 + COL27 + 2 * COL28 + 2 * COL30 − COL32 − COL34 − 2 * COL35 − COL36 − 2 * COL37 − COL38 − 2 * COL39 + COL41 + COL43 + 2 * COL44 + COL45 + 2 * COL46 + COL47 + 2 * COL48 + COL49 − COL52 − 2 * COL53 − COL54 − 2 * COL55 − COL56 − 2 * COL57 − COL58 − COL60 + 2 * COL62 + COL63 + 2 * COL64 + COL65 + 2 * COL66 + COL67 + COL69 − 2 * COL71 − 2 * COL73 − COL74 − 2 * COL75 − COL76 − COL78 + 2 * COL82 + 2 * COL84 + COL85 + COL87 − 2 * COL93
COL97 =	COL4 − COL13 − COL15 + COL22 + COL24 + COL26 − COL31 − COL33 − COL35 − COL37 + COL42 + COL44 + COL46 + COL48 − COL53 − COL55 − COL57 − COL59 + COL64 + COL66 + COL68 + COL70 − COL75 − COL77 − COL79 + COL86 + COL88
COL98 =	COL3 + 2 * COL8 − COL12 − COL14 − 2 * COL17 − 2 * COL19 + COL21 + COL23 + COL25 + 2 * COL26 + 2 * COL28 + 2 * COL30 − COL32 − COL34 − 2 * COL35 − COL36 − 2 * COL37 − 2 * COL39 + COL43 + 2 * COL44 + COL45 + 2 * COL46 + COL47 + 2 * COL48 − 2 * COL53 − COL54 − 2 * COL55 − COL56 − 2 * COL57 − COL58 + 2 * COL62 + 2 * COL64 + COL65 + 2 * COL66 + COL67 + COL69 − 2 * COL71 − 2 * COL73 − 2 * COL75 − COL76 − COL78 − COL80 + 2 * COL82 + 2 * COL84 + COL87 + COL89 − 2 * COL93
COL99 =	COL2 − COL11 − COL13 + COL22 + COL24 − COL33 − COL35 + COL44 + COL46 − COL55 − COL57 + COL66 + COL68 − COL77 − COL79 + COL88 + COL90
COL100 =	Intercept − COL2 − COL3 − COL4 − COL5 − COL6 − COL7 − 7 * COL8 − COL9 − **COL10** + COL13 + COL14 + COL15 + COL16 + 7 * COL17 + COL18 + 7 * **COL19** − COL21 − COL22 − COL23 − 2 * COL24 − 2 * COL25 − 8 * COL26 − 2 * COL27 − 8 * **COL28** − COL29 − 7 * COL30 + COL32 + COL33 + COL34 + 8 * COL35 + 2 * COL36 + 8 * **COL37** + COL38 + 7 * COL39 − COL41 − COL42 − 2 * COL43 − 8 * COL44 − 2 * COL45 − 9 * **COL46** − 2 * COL47 − 8 * COL48 − COL49 − COL50 + COL52 + 7 * COL53 + 2 * COL54 + 8 * **COL55** + COL56 + 8 * COL57 + COL58 + COL59 − COL61 − 7 * COL62 − 2 * COL63 − 8 * **COL64** − 2 * COL65 − 8 * COL66 − COL67 − 2 * COL68 − COL69 − COL70 + 6 * COL71 + COL72 + 7 * **COL73** + COL74 + 7 * COL75 + COL76 + COL77 + COL79 − COL81 − 7 * **COL82** − COL83 − 7 * COL84 − COL85 − COL86 − COL87 − COL88 − COL90 + 6 * COL93

COL91 =	COL10 − COL19 + COL28 − COL37 + COL46 − COL55 + COL64 − COL73 + COL82
COL92 =	COL9 − COL18 − COL20 + COL27 + COL29 − COL36 − COL38 + COL45 + COL47 − COL54 − COL56 + COL63 + COL65 − COL72 − COL74 + COL81 + COL83
COL93 =	COL8 − COL17 − COL19 + COL26 + COL28 + COL30 − COL35 − COL37 − COL39 + COL44 + COL46 + COL48 − COL53 − COL55 − COL57 + COL62 + COL64 + COL66 − COL71 − COL73 − COL75 + COL82 + COL84
COL94 =	COL7 − COL16 − COL18 + COL25 + COL27 + COL29 − COL34 − COL36 − COL38 − COL40 + COL43 + COL45 + COL47 + COL49 − COL52 − COL54 − COL56 − COL58 + COL61 + COL63 + COL65 + COL67 − COL72 − COL74 − COL76 + COL83 + COL85

Figure A3. *Cont*.

COL95 =	COL6 − COL15 − COL17 + COL24 + COL26 + COL28 − COL33 − COL35 − COL37 − COL39 + COL42 + COL44 + COL46 + COL48 + COL50 − COL51 − COL53 − COL55 − COL57 − COL59 + COL62 + COL64 + COL66 + COL68 − COL73 − COL75 − COL77 + COL84 + COL86
COL97 =	COL4 − COL13 − COL15 + COL22 + COL24 + COL26 − COL31 − COL33 − COL35 − COL37 + COL42 + COL44 + COL46 + COL48 − COL53 − COL55 − COL57 − COL59 + COL64 + COL66 + COL68 + COL70 − COL75 − COL77 − COL79 + COL86 + COL88
COL98 =	−0.66667 * COL2 + COL3 − 1.33333 * COL5 − COL12 + 0.66667 * COL13 + 0.33333 * COL14 + 1.33333 * COL16 + COL21 − 0.33333 * COL23 − 0.66667 * COL24 − 0.33333 * COL25 − 1.33333 * COL27 + 0.33333 * COL32 + 0.33333 * COL34 + 0.66667 * COL35 + 0.33333 * COL36 + 1.33333 * COL38 − 1.33333 * COL41 − 0.33333 * COL43 − 0.33333 * COL45 − 0.66667 * COL46 − 0.33333 * COL47 − 1.33333 * COL49 + 1.33333 * COL52 + 0.33333 * COL54 + 0.33333 * COL56 + 0.66667 * COL57 + 0.33333 * COL58 + 1.33333 * COL60 − 1.33333 * COL63 − 0.33333 * COL65 − 0.33333 * COL67 − 0.66667 * COL68 − 0.33333 * COL69 + 1.33333 * COL74 + 0.33333 * COL76 + 0.33333 * COL78 + 0.66667 * COL79 − COL80 − 1.33333 * COL85 − 0.33333 * COL87 + COL89 − 0.66667 * COL90 + 1.33333 * COL96
COL99 =	COL2 − COL11 − COL13 + COL22 + COL24 − COL33 − COL35 + COL44 + COL46 − COL55 − COL57 + COL66 + COL68 − COL77 − COL79 + COL88 + COL90
COL97 =	−COL2 + COL4 − COL12 + COL14 + COL16 − COL20 + COL25 − COL29 + COL35 − COL39 − COL44 − COL49 + COL51 + COL56 + COL58 − COL69 − COL73 + COL77 − COL79 + COL83 + COL89 − COL91 + COL95
COL97 =	COL68 − COL69 − COL73 + COL74 − COL75 + COL76 + COL78 − COL79 + COL80 − COL81 + COL82 − COL83 + COL84 − COL85 + COL86 − COL87 − COL89 + COL90 − COL91 + COL92 + COL96
COL100 =	−COL2 + COL3 − COL5 + 2 * COL7 + COL8 + 2 * COL9 − COL10 − 3 * COL11 − COL13 + 2 * COL15 + COL16 + 2 * COL17 − COL18 − 2 * COL19 + COL20 + 2 * COL21 − 2 * COL22 − 4 * COL23 − 3 * COL24 + 2 * COL25 + COL26 + 3 * COL27 + COL28 − COL29 − COL31 + COL33 − COL34 + COL35 − COL37 + COL38 − COL39 − COL40 − 4 * COL41 + COL43 + COL44 + 4 * COL45 − COL47 − COL48 − 4 * COL49 + COL53 + 2 * COL54 + 2 * COL55 + 3 * COL56 + 2 * COL57 + 2 * COL58 − 2 * COL59 − 3 * COL60 − 2 * COL61 − 2 * COL62 − COL63 + COL64 + COL65 + COL66 + COL67 + COL70 − COL71 + 3 * COL75 + COL77 − 2 * COL79 − COL80 − 2 * COL81 + COL82 + 3 * COL83 + COL85 + 2 * COL86 − 4 * COL87 − COL88 − 2 * COL90 − 2 * COL91 + 3 * COL93 + 2 * COL94 + COL95 − 2 * COL96 − 2 * COL97
COL9 =	−COL2 + COL4 + COL7 + 147 × 10^{-13} * COL16 − 147 × 10^{-13} * COL17 + 118 × 10^{-13} * COL19 − 135 × 10^{-13} * COL26 + 147 × 10^{-13} * COL29 + 147 × 10^{-13} * COL30 + 126 × 10^{-13} * COL31 − 18 × 10^{-12} * COL34 − 15 × 10^{-12} * COL35 + 151 × 10^{-13} * COL39 − 135 × 10^{-13} * COL48 + 126 × 10^{-13} * COL50 + 147 × 10^{-13} * COL51 + 147 × 10^{-13} * COL52 − 15 × 10^{-12} * COL54 − 18 × 10^{-12} * COL55 + 151 × 10^{-13} * COL58 − 134 × 10^{-13} * COL64 − 156 × 10^{-13} * COL72 − 14 × 10^{-12} * COL74 + 231 × 10^{-13} * COL77 − 217 × 10^{-13} * COL82 + 118 × 10^{-13} * COL86 − 174 × 10^{-13} * COL93 − 13 × 10^{-12} * COL94 + 138 × 10^{-13} * COL96 + 148 × 10^{-13} * COL97 − 13 × 10^{-12} * COL98

Figure A3. Partial SAS PROC REG output for the simulated examples; matrix column (designated by COL#, where # denotes column position) linear combinations for each of the matrix's zero eigenvalues. The left-hand table column contains particular redundant matrix columns, and the right-hand expression is each redundant column's linear combination.

n = 500	93 collinear columns	C264 =	C29 + C30 + C33 + C34 + C45 + C46 − C73 − C74 − C75 − C76 − C77 − C78 − C79 − C88 − C89 − C90 + C117 + C118 + C119 + C120 + C121 + C122 + C123 + C124 + C125 + C132 + C133 + C134 − C164 − C165 − C166 − C167 − C168 − C169 − C170 − C171 − C172 − C177 − C178 − C179 + C211 + C212 + C213 + C214 + C215 + C216 + C217 + C218 + C219 + C222 + C223 + C224 − C257 − C258 − C259 − C260 − C261 − C262 − C263 − C265 − C266 − C267 + C303 + C304 + C305 + C306 + C307 + C308 + C309 + C310 + C311 + C312 − C350 − C351 − C352 − C353 − C354 − C355 − C356 − C357 + C397 + C398 + C399 + C400 + C401 + C402 − C444 − C445 − C446 − C447 + C490 + C491
n = 500	117 collinear columns	C360 =	C17 + C18 + C22 + C23 − C38 − C39 − C40 − C44 − C45 − C46 + C60 + C61 + C62 + C66 + C67 + C68 − C83 − C84 − C85 − C89 − C90 − C91 + C106 + C107 + C108 + C112 + C113 + C114 − C129 − C130 − C131 − C135 − C136 − C137 + C152 + C153 + C154 + C158 + C159 + C160 − C175 − C176 − C177 − C181 − C182 − C183 + C198 + C199 + C200 + C204 + C205 + C206 − C221 − C222 − C223 − C227 − C228 − C229 + C244 + C245 + C246 + C250 + C251 + C252 − C267 − C268 − C269 − C273 − C274 − C275 + C290 + C291 + C292 + C296 + C297 + C298 − C313 − C314 − C315 − C319 − C320 − C321 + C335 + C336 + C337 + C341 + C342 + C343 − C358 − C359 − C364 − C365 − C366 + C381 + C382 + C383 + C387 + C388 + C389 − C405 − C406 − C410 − C411 − C412 + C433 + C434 + C435 + C453 + C454 − C456 − C457 − C458 − C477 − C478 + C480 + C481
n = 1000	62 collinear columns	C406	C15 − C17 − C46 + C50 + C76 − C82 − C106 + C114 + C135 − C144 − C163 + C174 + C193 − C207 − C224 + C240 + C255 − C273 − C285 + C305 + C315 − C336 − C345 + C368 + C375 − C401 + C432 + C435 − C465 + C497 − C498 + C531 − C560 − C563 + C590 + C595 − C621 − C628 + C650 + C659 − C681 − C692 + C712 + C725 − C743 − C758 + C774 + C790 − C804 − C823 + C835 + C856 − C865 − C888 + C896 + C920 − C926 − C953 + C957 + C984 − C986
n = 1000	238 collinear columns	C608 =	− C2 + C4 − C8 + C10 − C14 + C16 − C20 + C21 − C25 + C27 − C31 + C33 − C37 + C39 − C61 + C65 − C67 + C71 − C73 + C77 − C79 + C82 − C84 + C88 − C90 + C94 − C96 + C100 − C122 + C124 − C128 + C130 − C134 + C136 − C140 + C141 − C145 + C147 − C151 + C153 − C157 + C159 − C181 + C185 − C187 + C191 − C193 + C197 − C199 + C202 − C204 + C208 − C210 + C214 − C216 + C220 − C242 + C244 − C248 + C250 − C254 + C256 − C260 + C261 − C265 + C267 − C271 + C273 − C277 + C279 − C301 + C305 − C307 + C311 − C313 + C317 − C319 + C322 − C324 + C328 − C330 + C334 − C336 + C340 − C362 + C364 − C368 + C370 − C374 + C376 − C380 + C381 − C385 + C387 − C391 + C393 − C397 + C399 − C421 + C425 − C427 + C431 − C433 + C437 − C439 + C442 − C444 + C448 − C450 + C454 − C456 + C460 − C482 + C484 − C488 + C490 − C494 + C496 − C500 + C501 − C505 + C507 − C511 + C513 − C517 + C519 − C541 + C545 − C547 + C551 − C553 + C557 − C559 + C562 − C564 + C568 − C570 + C574 − C576 + C580 − C602 + C604 + C610 − C614 + C616 − C620 + C621 − C625 + C627 − C631 + C633 − C637 + C639 − C661 + C665 − C667 + C671 − C673 + C677 − C679 + C682 − C684 + C688 − C690 + C694 − C696 + C700 − C722 + C724 − C728 + C730 − C734 + C736 − C740 + C741 − C745 + C747 − C751 + C753 − C757 + C759 − C781 + C785 − C787 + C791 − C793 + C797 − C799 + C802 − C804 + C808 − C810 + C814 − C816 + C820 − C842 + C844 − C848 + C850 − C854 + C856 − C860 + C861 − C865 + C867 − C871 + C873 − C877 + C879 − C901 + C905 − C907 + C911 − C913 + C917 − C919 + C922 − C924 + C928 − C930 + C934 − C936 + C940 − C962 + C964 − C968 + C970 − C974 + C976 − C980 + C981 − C985 + C987 − C991 + C993 − C997 + C999
n = 2000	45 collinear columns	C1957 =	− C45 + C88 − C132 + C176 − C220 + C263 − C306 + C350 − C394 + C436 − C478 + C521 − C564 + C608 − C652 + C695 − C739 + C783 − C826 + C870 − C913 + C957 − C1001 + C1045 − C1088 + C1132 − C1176 + C1220 − C1263 + C1307 − C1349 + C1392 − C1434 + C1478 − C1522 + C1566 − C1610 + C1653 − C1697 + C1740 − C1784 + C1828 − C1871 + C1914
	88 collinear columns	C1973 =	− C28 + C30 + C71 − C75 − C115 + C121 + C159 − C167 − C203 + C213 + C246 − C258 − C289 + C303 + C333 − C349 − C377 + C395 + C419 − C439 − C461 + C483 + C505 − C528 − C547 + C573 + C591 − C619 − C635 + C665 + C679 − C710 − C722 + C766 + C799 − C809 − C843 + C853 + C886 − C896 − C930 + C940 + C974 − C984 − C1018 + C1028 + C1062 − C1072 − C1105 + C1115 + C1149 − C1159 − C1193 + C1203 + C1237 − C1280 − C1292 + C1324 + C1336 − C1365 − C1381 + C1408 + C1425 − C1451 − C1471 + C1495 + C1517 − C1539 − C1563 + C1583 + C1609 − C1626 − C1654 + C1670 + C1700 − C1714 − C1745 + C1757 + C1791 − C1801 − C1836 + C1844 + C1882 − C1888 − C1927 + C1931 + C1971

Figure A4. *Cont.*

	88 collinear columns	C1976 =	− C25 + C27 + C68 − C72 − C112 + C118 + C156 − C164 − C200 + C210 + C244 − C255 − C286 + C300 + C330 − C346 − C374 + C392 + C416 − C436 − C458 + C480 + C502 − C525 − C544 + C570 + C588 − C616 − C632 + C662 + C676 − C707 − C719 + C753 + C763 − C798 − C806 + C844 + C850 − C893 − C933 + C937 + C977 − C981 − C1021 + C1025 + C1065 − C1069 − C1108 + C1152 + C1158 − C1196 − C1204 + C1240 + C1250 − C1283 − C1295 + C1327 + C1339 − C1368 − C1384 + C1411 + C1428 − C1454 − C1474 + C1498 + C1520 − C1542 − C1566 + C1586 + C1612 − C1629 − C1657 + C1673 + C1703 − C1716 − C1748 + C1760 + C1794 − C1804 − C1839 + C1847 + C1885 − C1891 − C1930 + C1934 + C1974
	88 collinear columns	C1996 =	− C5 + C7 + C49 − C53 − C92 + C98 + C136 − C144 − C180 + C190 − C236 + C269 + C280 − C314 − C326 + C360 + C372 − C405 − C416 + C449 + C460 − C494 − C506 + C538 + C550 − C584 − C596 + C630 + C642 − C676 − C687 + C721 + C733 − C767 − C779 + C812 + C824 − C858 − C870 + C903 + C915 − C949 − C961 + C995 + C1007 − C1041 − C1053 + C1086 + C1098 − C1132 − C1144 + C1178 + C1190 − C1224 − C1236 + C1269 + C1281 − C1315 − C1327 + C1359 + C1370 − C1403 − C1414 + C1448 + C1460 − C1494 − C1506 + C1540 + C1552 − C1586 − C1598 + C1631 + C1643 − C1677 − C1689 + C1722 + C1734 − C1768 + C1814 − C1824 − C1859 + C1867 + C1904 − C1910 − C1950 + C1953 + C1994

Figure A4. Partial SAS PROC TRANSREG output for five simulated examples; matrix column (designated by C#, where # denotes column position) linear combinations for each of the matrix's zero eigenvalues. From left to right, the column contents are: n (i.e., number of vertices), size of linear combination yielding a zero eigenvalue, a particular redundant matrix column causing collinearity, and the expression for the redundant column's linear combination.

References

1. Fan, Y.; Wang, L. Bounds for the positive and negative inertia index of a graph. *Linear Algebra Its Appl.* **2017**, *522*, 15–27. [CrossRef]
2. Li, S.; Sun, W. On the relation between the positive inertia index and negative inertia index of weighted graphs. *Linear Algebra Its Appl.* **2019**, *563*, 411–425. [CrossRef]
3. Lancaster, P.; Tismenetsky, M. *The Theory of Matrices*; Academic Press: New York, NY, USA, 1985.
4. Wang, L. Nullity of a graph in terms of path cover number. *Linear Multilinear Algebra* **2021**, *69*, 1902–1908. [CrossRef]
5. Cheng, B.; Liu, M.; Tam, B. On the nullity of a connected graph in terms of order and maximum degree. *Linear Algebra Its Appl.* **2022**, *632*, 193–232. [CrossRef]
6. Hicks, I.; Brimkov, B.; Deaett, L.; Haas, R.; Mikesell, D.; Roberson, D.; Smith, L. Computational and theoretical challenges for computing the minimum rank of a graph. *INFORMS J. Comput.* **2022**, *34*, 2868–2872. [CrossRef]
7. Alaeiyan, M.; Obayes, K.; Alaeiyan, M. Prediction nullity of graph using data mining. *Results Nonlinear Anal.* **2023**, *6*, 1–8.
8. Arumugam, S.; Bhat, K.A.; Gutman, L.; Karantha, M.; Poojary, R. Nullity of graphs—A survey and some new results. In *Applied Linear Algebra, Probability and Statistics: A Volume in Honour of CR Rao and Arbind K. Lal*; Bapat, R., Karantha, M., Kirkland, S., Neogy, S., Pati, S., Puntanen, S., Eds.; Springer Nature: Singapore, 2023; pp. 155–175.
9. Druinsky, A.; Carlebach, E.; Toledo, S. Wilkinson's inertia-revealing factorization and its application to sparse matrices. *Numer. Linear Algebra Appl.* **2018**, *25*, e2130. [CrossRef]
10. Fan, Y.; Wang, Y.; Bao, Y.; Wan, J.; Li, M.; Zhu, Z. Eigenvectors of Laplacian or signless Laplacian of hypergraphs associated with zero eigenvalue. *Linear Algebra Its Appl.* **2019**, *579*, 244–261. [CrossRef]
11. Nakatsukasa, Y.; Noferini, V. Inertia laws and localization of real eigenvalues for generalized indefinite eigenvalue problems. *Linear Algebra Its Appl.* **2019**, *578*, 272–296. [CrossRef]
12. Griffith, D.; Luhanga, U. Approximating the inertia of the adjacency matrix of a connected planar graph that is the dual of a geographic surface partitioning. *Geogr. Anal.* **2011**, *43*, 383–402. [CrossRef]
13. Comellas, F.; Dalfó, C.; Fiol, M.; Mitjana, M. The spectra of Manhattan street networks. *Linear Algebra Its Appl.* **2008**, *429*, 1823–1839. [CrossRef]
14. Griffith, D. Approximation of Gaussian spatial autoregressive models for massive regular square tessellation data. *Int. J. Geogr. Inf. Sci.* **2015**, *29*, 2143–2173. [CrossRef]
15. Bindel, D.; Dong, K. Modified kernel polynomial method for estimating graph spectra. In Proceedings of the SIAM Workshop on Network Science, Snowbird, UT, USA, 15–16 May 2015. Available online: https://www.cs.cornell.edu/~bindel/papers/2015-siam-ns.pdf (accessed on 24 October 2023).
16. Griffith, D. Extreme eigenfunctions of adjacency matrices for planar graphs employed in spatial analyses. *Linear Algebra Its Appl.* **2004**, *388*, 201–219. [CrossRef]
17. Wu, X.-Z.; Liu, J.-P. Sharp upper bounds for the adjacency and the signless Laplacian spectral radius of graphs. *Appl. Math.—A J. Chin. Univ.* **2019**, *34*, 100–112. [CrossRef]
18. Feng, L.; Yu, G.; Zhang, X.-D. Spectral radius of graphs with given matching number. *Linear Algebra Its Appl.* **2007**, *422*, 133–138. [CrossRef]

19. Milanese, A.; Sun, J.; Nishikawa, T. Approximating spectral impact of structural perturbations in large networks. *Phys. Rev. E* **2010**, *81*, 046112. [CrossRef]
20. Meyer, C. Chapter 8: Perron–Frobenius theory of nonnegative matrices. In *Matrix Analysis and Applied Linear Algebra*; SIAM: Philadelphia, PA, USA, 2000; pp. 661–704.
21. Griffith, D. Eigenfunction properties and approximations of selected incidence matrices employed in spatial analyses. *Linear Algebra Its Appl.* **2000**, *321*, 95–112. [CrossRef]
22. Mohar, B. On the sum of k largest eigenvalues of graphs and symmetric matrices. *J. Comb. Theory Ser. B* **2009**, *99*, 306–313. [CrossRef]
23. Tait, M.; Tobin, J. Three conjectures in extremal spectral graph theory. *J. Comb. Theory Ser. B* **2017**, *126*, 137–163. [CrossRef]
24. Elphick, C.; Wocjan, P. An inertial lower bound for the chromatic number of a graph. *Electron. J. Comb.* **2016**, *24*, 1–58.
25. Taliceo, N.; Griffith, D. The K_4 graph and the inertia of the adjacency matrix for a connected planar graph. *STUDIA KPZK PAN (Publ. Pol. Acad. Sci.)* **2018**, *183*, 185–209.
26. Keller, M.; Trotter, W. Chapter 5: Graph theory. In *Applied Combinatorics*; LibreTexts: Davis, CA, USA, 2023. Available online: https://math.libretexts.org/Bookshelves/Combinatorics_and_Discrete_Mathematics/Applied_Combinatorics_(Keller_and_Trotter) (accessed on 24 October 2023).
27. Griffith, D.; Sone, A. Trade-offs associated with normalizing constant computational simplifications for estimating spatial statistical models. *J. Stat. Comput. Simul.* **1995**, *51*, 165–183. [CrossRef]
28. Venkateshan, S.; Swaminathan, P. Chapter 2: Solution of linear equations. In *Computational Methods in Engineering*; Elsevier: Amsterdam, The Netherlands, 2013; pp. 19–103.
29. Venkateshan, S.; Swaminathan, P. Chapter 4: Solution of algebraic equations. In *Computational Methods in Engineering*; Elsevier: Amsterdam, The Netherlands, 2013; pp. 155–201.
30. Bollobás, B. *Modern Graph Theory*; Graduate Texts in Mathematics; Springer: Berlin/Heidelberg, Germany, 1998; Volume 184.
31. Ord, J. Estimating methods for models of spatial interaction. *J. Am. Stat. Assoc.* **1975**, *70*, 120–126. [CrossRef]
32. Hawkins, D. On the investigation of alternative regressions by principal component analysis. *J. R. Stat. Soc. Ser. C (Appl. Stat.)* **1973**, *22*, 275–286. [CrossRef]
33. Gupta, R. Chapter 6: Eigenvalues and eigenvectors. In *Numerical Methods: Fundamentals and Applications*; Cambridge University Press: Cambridge, UK, 2019; pp. 268–298.
34. Taboga, M. Algebraic and Geometric Multiplicity of Eigenvalues. Lectures on Matrix Algebra. 2021. Available online: https://www.statlect.com/matrix-algebra/algebraic-and-geometric-multiplicity-of-eigenvalues (accessed on 18 June 2022).
35. SAS Institute Inc. *SAS/STAT®15.1 User's Guide*; SAS Institute Inc.: Cary, NC, USA, 2018. Available online: https://documentation.sas.com/doc/en/pgmsascdc/9.4_3.4/statug/statug_reg_details36.htm (accessed on 24 October 2023).
36. Goodnight, J. A tutorial on the sweep operator. *Am. Stat.* **1979**, *33*, 149–158.
37. Lange, K. *Numerical Analysis for Statisticians*, 2nd ed.; Springer: New York, NY, USA, 2010.
38. Ahamed, M.; Biswa, A.; Phukon, M. A study on multicollinearity diagnostics and a few linear estimators. *Adv. Appl. Stat.* **2023**, *89*, 29–54. [CrossRef]
39. Tsatsomeros, M. Principal pivot transforms: Properties and applications. *Linear Algebra Its Appl.* **2000**, *307*, 151–165. [CrossRef]
40. Duersch, J.; Gu, M. Randomized QR with column pivoting. *SIAM J. Sci. Comput.* **2017**, *39*, C263–C291. [CrossRef]
41. Martinsson, P.; OrtI, G.Q.; Heavner, N.; Van De Geijn, R. Householder QR factorization with randomization for column pivoting (HQRRP). *SIAM J. Sci. Comput.* **2017**, *39*, C96–C115. [CrossRef]
42. Li, H. *Numerical Methods Using Java: For Data Science, Analysis, and Engineering*; APress: New York, NY, USA, 2022.
43. Sylvester, J. A demonstration of the theorem that every homogeneous quadratic polynomial is reducible by real orthogonal substitutions to the form of a sum of positive and negative squares. *Philos. Mag. 4th Ser.* **1852**, *4*, 138–142. [CrossRef]
44. Dempster, A. *Elements of Continuous Multivariate Analysis*; Addison-Wesley: Reading, MA, USA, 1969.
45. Andrews, D.F. A robust method for multiple linear regression. *Technometrics* **1974**, *16*, 523–531. [CrossRef]
46. Tucker, A. Principal pivot transforms of square matrices. *SIAM Rev.* **1963**, *5*, 305.
47. Duffin, R.; Hazony, D.; Morrison, N. Network synthesis through hybrid matrices. *SIAM J. Appl. Math.* **1966**, *14*, 390–413. [CrossRef]
48. Stewart, M.; Stewart, G. On hyperbolic triangularization: Stability and pivoting. *SIAM J. Matrix Anal. Appl.* **1998**, *19*, 847–860. [CrossRef]
49. Bivand, R.; Hauke, J.; Kossowski, T. Computing the Jacobian in Gaussian spatial autoregressive models: An illustrated comparison of available methods. *Geogr. Anal.* **2013**, *45*, 150–179. [CrossRef]
50. Griffith, D.; Bivand, R.; Chun, Y. Implementing approximations to extreme eigenvalues and eigenvalues of irregular surface partitionings for use in SAR and CAR models. *Procedia Environ. Sci.* **2015**, *26*, 119–122. [CrossRef]
51. Griffith, D. Generating random connected planar graphs. *GeoInformatica* **2018**, *22*, 767–782. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.