DLA_GUI Documentation

Michael J. Adams *
Jonathan Vollmer

September 22, 2008

1 Introduction

The program DLA_GUI has been written to carry out computations associated with demographic loop analysis as described in [1]. We refer the user to the paper just cited for the mathematical justifications of the computations carried out by DLA_GUI and described in this documentation. For additional information on loop analysis see [4, 5, 6, 7]. For an introduction to the use of matrix population models in ecology we recommend [3].

This software was written using MATLAB Release R2008a with the Optimization Toolbox. Please see the associated copyright file.

Start the program DLA_GUI.m. A graphical user interface will appear with a 5-by-5 input field for a population projection matrix initialized to 0. The user is required to provide as input an n-by-n population projection matrix A. The size can be reset by entering a new value in the size field then clicking on the [Set Size of Matrix:] button. Use the tab or arrow keys to move through the input field and be certain to hit [Enter] on the keyboard before clicking the [GO] button to run the analysis.

It is assumed that the matrix is nonnegative (that is, each component $a_{i,j} \geq 0$), $2 \leq n \leq 100$, and that A is an irreducible (though not necessarily primitive) matrix. This last condition is equivalent to the requirement that the associated life cycle graph D be strongly connected (see, for example, [2]). If these conditions are not met, an error message will be generated.

The program performs a series of computations on the matrix A to provide the user with the following basic demographic information:

- The dominant eigenvalue of A, λ, representing the finite rate of population change. Because A is irreducible we are guaranteed that $\lambda > 0$ and is the unique real eigenvalue satisfying $\lambda \geq |\lambda_i|$ for all other eigenvalues λ_i. This value will be displayed in the interface.

*Department of Mathematics and Computer Science, Truman State University, Kirksville, MO 63501 Email: mjadams@truman.edu. This work was partially supported by NSF Grant No. 0436348 through Truman State University's Mathematical Biology Initiative.

- The sensitivity matrix S where $s_{i,j} = \dfrac{\partial\lambda}{\partial a_{i,j}}$. This matrix can be viewed by clicking the [Sensitivity Matrix] button.

- The elasticity matrix E where $e_{i,j} = \dfrac{a_{i,j}}{\lambda}\dfrac{\partial\lambda}{\partial a_{i,j}}$. The values have been scaled to sum to 100, so each entry represents the percentage contribution of $a_{i,j}$ to the total elasticity of λ. This matrix can be viewed by clicking the [Elasticity Matrix] button.

- The vector of reproductive values. This is a left-eigenvector of A for the eigenvalue λ normalized so the first component is equal to one. This vector will be displayed in the interface.

- The stable stage distribution. This is a right-eigenvector of A for the eigenvalue λ normalized so the components sum to one. This vector will be displayed in the interface.

The matrix A and all of the output described above will also be written to a program-generated text file named DLA_GUI_Output.txt.

2 Computing Loop Decompositions

In order to carry out loop analysis as described in [1], it is necessary to identify all loops (simple directed cycles) of the life cycle graph D. Each loop represents a possible life history path traced through the life cycle graph by an individual in the population. Under the assumption of irreducibility of A, D is a strongly connected digraph with n nodes and m edges, including self-loops which correspond to nonzero diagonal elements of A. Identifying a single loop in such a graph is relatively simple; identifying all such loops is more difficult in general because of the potential for a combinatorial explosion in the number of such loops as n and m increase. For example, a population projection matrix all of whose entries are positive corresponds to a complete digraph, that is, a digraph in which every node has a self-loop and between any two distinct nodes i and j, there is an edge directed from i to j and an edge directed from j to i. A complete digraph on three nodes will have 8 loops, on four nodes it will have 24 loops, on five nodes it will have 89 loops, and on six nodes it will have 415 loops.

A loop is represented by the sequence of nodes visited, starting and ending with the node of smallest index. For example, the loop of length 3 represented by the sequence [2 3 5 2] starts at node 2 and follows directed edges in D to node 3, then to node 5, then back to node 2 to complete the cycle. The same loop could be described by [3 5 2 3] or [5 2 3 5], but to avoid counting the same loop more than once it will always be described by the sequence which starts and ends with the node of smallest index appearing in the loop. The loops and their lengths are listed in increasing lexicographic order of the sequences which represent them. This information is displayed on the interface and the list of loops (*sans* lengths) is written to DLA_GUI_Output.txt.

It is important to understand the distinction between the characteristic elasticity of a loop and loop elasticity. In loop analysis, the total elasticity of λ is partitioned over the set of loops subject to the requirement that every edge in a given loop is assigned the same elasticity: this value is the characteristic elasticity of the loop. The loop elasticity is the sum of the elasticities assigned to its edges, i.e., it is the characteristic elasticity multiplied by the length of the loop. It is the loop elasticity that is of interest in ecology.

In DLA_GUI, a loop decomposition is given by a nonnegative vector x satisfying the matrix equation

$$\Lambda x = e \tag{1}$$

where Λ (called Loop_Mat in the code) is an m-by-l 0-1 matrix whose rows are indexed by the edges of D and whose columns are the characteristic vectors of the loops and e is an m-by-1 vector whose components are the elasticities of the edges in D, i.e., elasticities of the transitions in the life cycle graph. **For any such vector x, the ith component x_i represents the characteristic elasticity of the ith loop.** Multiplying each component x_i by l_i, the length of the ith loop, for $i = 1, 2, \ldots, l$ yields the loop elasticities.

Let S denote the set of all nonnegative solutions of equation (1). There are two cases to consider: either S consists of a single vector, or S is a convex polytope containing a continuum of vectors. Which of the two cases obtain, and how DLA_GUI handles these cases, depends on the the quantity $s = l - (m - n + 1)$. Briefly, $m - n + 1$ is the dimension of the cycle space of D and this space is spanned by the loops of D, so s is the dimension of the nullspace of Λ. If $s = 0$ then the columns of Λ are linearly independent and 1 has a unique solution. (The existence of a solution is guaranteed by the fact that the vector e of elasticities is itself an element of the cycle space of D.) If $s > 0$, then S is an s-dimensional convex polytope, the feasible set of a linear programming problem with two sets of constraints: $\Lambda x = e$ and $x \geq 0$.

The dimension of the cycle space and the dimension of S are displayed on the interface and written to DLA_GUI_Output.txt.

In the event that $s = 0$, S consists of a single vector and the loop decomposition is unique. The program then solves the system $\Lambda x = e$ and returns both the characteristic elasticities and loop elasticities for each of the loops. This information is displayed on the interface and written to DLA_GUI_Output.txt. Execution is then halted until a new matrix A is entered and processed.

If $s > 0$ the program returns the minimum and the maximum possible loop elasticities for each loop. This is accomplished by using the Simplex algorithm for linear programming to find the minimum and maximum values of the linear function $F_i = l_i x_i$ (where i is the index for the ith loop) as x varies over S. **This computation is carried out separately for each loop.** This information is displayed on the interface and written to DLA_GUI_Output.txt.

Except for the highly constrained situation in which $s = 0$, it is not possible to find a loop decomposition for which every loop has the maximum possible elasticity. In general, increasing the elasticity of one loop comes at the expense of reducing the elasticity in other loops. A set of vectors describing these tradeoffs is given by any basis for the nullspace of Λ.

When $s > 0$, there are infinitely many loop decompositions. DLA_GUI allows the user to identify loop decompositions which satisfy a simple linear constraint: for any user-selected subset L of the loops, the program will identify a loop decomposition for which the sum of the loop elasticities for the loops in L is the minimum possible out of all loop decompositions, and it will also identify a loop decomposition for which the sum of the loop elasticities for the loops in L is the maximum possible out of all loop decompositions. Selecting the set of loops in L is accomplished by clicking on the loops of interest in the list of all loops. The selected loops will be indicated in the Interactive Optimization of Loop Elasticities field. Once the set has been selected, click the [Optimize] button and DLA_GUI will then compute and return two linear programming solutions of equation 1 which yield the minimum (respectively, maximum) possible value of $\sum l_i x_i$ where the sum is taken over the set of indices i such that $L_i \in L$. These solutions are basic feasible solutions of the linear programming program and need not be unique. The sums of the loop elasticities for the loops in L are also displayed. This information is written to DLA_GUI_Output.txt. The user may clear the selection field, enter a new set of loops, and repeat the process.

References

[1] Adams, Michael J. 2008. Graph decompositions for demographic loop analysis. Journal of Mathematical Biology **57**: 209-221.

[2] Brualdi, R., Ryser, H. 1991. Combinatorial Matrix Theory. Encyclopedia of Mathematics and its Applications. Cambridge University Press, Cambridge UK.

[3] Caswell, H. 2001. Matrix Population Models: construction, analysis, and interpretation, second edition. Sinauer Associates, Sunderland, Massachusetts, USA.

[4] Güneralp, B. 2007. An Improved Formal Approach to Demographic Loop Analysis. Ecology **88**: 2124-2131.

[5] Sun, L., Wang, M. 2007. An algorithm for a decomposition of weighted digraphs: with applications to life cycle analysis in ecology. Journal of Mathematical Biology **54**: 199-226.

[6] van Groenendael, J., de Kroon, H., Kalisz, S., Tuljapurkar, S. 1994. Loop Analysis: Evaluating Life History Pathways in Population Projection Matrices. Ecology **75**: 2410-2415.

[7] Wardle, G. M. 1998. A Graph Theory Approach to Demographic Loop Analysis. Ecology **79**: 2539-2549.