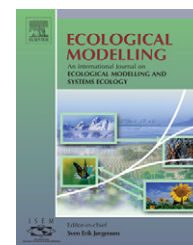


available at www.sciencedirect.comjournal homepage: www.elsevier.com/locate/ecolmodel

Cycling in ecosystems: An individual based approach

C. Kazanci^{a,b,*}, L. Matamba^a, E.W. Tollner^{b,c}

^a Department of Mathematics, University of Georgia, USA

^b Faculty of Engineering, University of Georgia, USA

^c Department of Biological and Agricultural Engineering, University of Georgia, USA

ARTICLE INFO

Keywords:

EcoNet
Ecological modeling
Cycling index
Network analysis
Simulation
Individual based
Flow analysis
Food web
Ecosystem modeling

ABSTRACT

Cycling index is an important ecological indicator used in ecosystem analysis. The higher the cycling in an ecosystem, the higher the utilization of mass and energy within the system before it is lost due to respiration and other factors. For a stock-flow type ecosystem model at steady state, Finn's cycling index (FCI) can be computed using simple matrix algebra. However, it is difficult to measure how well this index represents the actual cycling occurring in the system. In this paper, we study cycling in ecological networks using an individual based approach (particle tracking algorithm). This new simulation method provides access to the pathway data of individual particles that flow in the system, therefore one can quantify cycling using this pathway data quite literally. We used particle tracking simulations (PTS) to compute a cycling index using Finn's idea of flux based cycling. Our simulation based results (using no matrix algebra) agree with Finn's cycling index, verifying the accuracy of both the PTS, and the original linear algebraic formulation of FCI.

Published by Elsevier B.V.

1. Introduction

Cycling of nutrients, biomass or energy within ecosystems is an essential feature of these systems. Cycling is identified by Odum (1969) as a measure of maturity of a system. Odum observes that mature systems, as compared to developing ones, have a greater capacity to retain nutrients through cycling. Cycling of Carbon, however, can be a sign of a more stressed community, and appears to be a counter-indicator of ecosystem health (Wulff and Ulanowicz, 1989; Costanza and Norton, 2002). Nevertheless, amount of cycling within an ecosystem can be an important indicator of ecosystem health, and a useful diagnostic in the design of environmental impact statements (May, 1981).

Finn (1976) developed an index, a quantitative description of cycling in ecosystems. He later modified his index, where the new index was a normalized version of the previous one, ranging between 0 and 1 (Finn, 1980). Ulanowicz (1983)

described a different method, where he decomposed the network into weighted cyclic and non-cyclic (tree) sub-networks. Patten and Higashi (1984) extended Finn's index to incorporate not only the flow of nutrients among compartments, but also the storage of nutrients within compartments. Based on Finn's original work, Han (1997) defined the cycling matrix, measuring interactions among compartments due to cycling. Finn's cycling index (FCI) was further improved by Allesina and Ulanowicz (2004) to include all fluxes generated by cycling. They defined the comprehensive cycling index (CCI), which is cumbersome to compute, but is correlated with FCI.

Most studies related to cycling in ecosystems are based on Finn's original work (Finn, 1976). Finn's approach to quantify cycling is to measure the flux of nutrient in the system (TST) as a whole, and identify the portion of this flux (TST_c) that is due to cycling. Finn's cycling index is then defined as the fraction of flux that is cycled relative to the total system flux (Finn, 1980). FCI is formulated as a linear algebraic expression based

* Corresponding author. Tel.: +1 706 224 2221; fax: +1 706 542 8806.

E-mail address: caner@uga.edu (C. Kazanci).

0304-3800/\$ – see front matter. Published by Elsevier B.V.

doi:10.1016/j.ecolmodel.2008.09.013

on the flow rates among compartments, and environmental inputs and outputs. An interesting question is “How well does this algebraic expression represent the actual cycling in an ecosystem?”

This is a difficult question to answer, unless there is a way to label each nutrient molecule, and track their locations as they flow in the system. Cycling information can be derived from the individual path history of all nutrients that have been a part of the system. This is not only an impossible experiment to perform in real life, but also an impracticable simulation to run on a computer. However, if one is allowed to use identities larger than individual molecules, there is a feasible numerical method that accomplishes exactly what we have described. “Particle tracking algorithm” (Tollner and Kazanci, 2007; Kazanci and Tollner, in preparation) is a numerical simulation method where each energy (or mass) packet is labeled and tracked in time as it flows through the network. The flow history of each and individual particle is recorded, therefore this method is computationally intensive. However, it enables us to investigate many interesting ecological network properties, such as cycling, residence time, and indirect effects (Higashi and Patten, 1986).

In this paper, we compute FCI without using matrix algebra, a first since its original development.¹ We use Finn’s original idea of defining the cycling index as a fraction of fluxes that contain cycling (Finn, 1976). The only difference is that we have access to pathway data of all particles that have been in the system. We can then skip all approximations, and compute precisely the fraction of flux due to cycling, a luxury made possible by recent mathematical developments and modern computers. We describe the details of this computation, and compare our results to the algebraic formulation FCI in Section 4.2, followed by a discussion of feasibility and accuracy of this new approach. Finally, we discuss new opportunities made possible by the particle tracking algorithm.

2. Models and methods

2.1. Ecosystem model

In this section, we introduce the ecosystem model type that Finn’s work is applicable to, and then develop the mathematical equations necessary to compute FCI. The ecosystem models we study have a network structure, with many interactions among multiple identities. Identities (compartments, stocks, and nodes) can range from accumulated organic matter to hundreds of species, depending on the focus and complexity of the particular model. Interactions represent flow of energy, biomass, or a specific element such as C, N or P. For sake of simplicity and uniformity, we will use the term “nutrient” to describe the entity that flows among compartments.

In Fig. 1, we show an example model, the Calcium flows in the Hubbard Brook temperate forest ecosystem. Depending on the topology of the system, nutrients can flow between

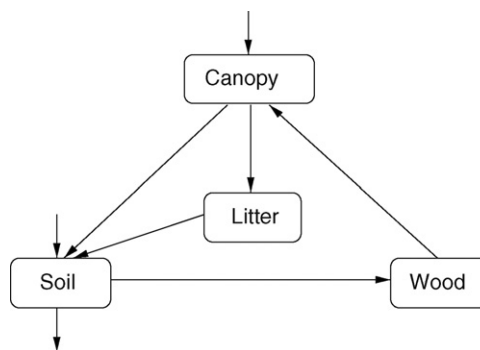


Fig. 1 – Ecosystem model representing Calcium flows in the Hubbard Brook temperate forest ecosystem, Hubbard Brook, New Hampshire, USA (Finn, 1980).

any two compartments in the system. For example, in Fig. 1, Calcium can not flow directly from *canopy* into *wood*. However, a portion of the Calcium flowing out of *canopy* might get into *wood* via *litter* and *soil*. To form a mathematical description of such an ecosystem model, we define the following parameters:

f_{ij} : flow rate of nutrients from compartment j to compartment i

z_i : rate of environmental input to compartment i

y_i : rate of environmental output from compartment i

x_i : storage value of compartment

Based on these parameters, we define the corresponding flow matrix F , input vector z , output vector y , and the storage vector x as follows:

$$F = \begin{bmatrix} f_{11} & \cdots & f_{1n} \\ \vdots & \ddots & \vdots \\ f_{n1} & \cdots & f_{nn} \end{bmatrix} \quad z = \begin{bmatrix} z_1 \\ \vdots \\ z_n \end{bmatrix} \quad y = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} \quad x = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}.$$

For an evolving ecosystem, all parameters defined above may be varying in time, hence each will be a function of time. We can construct a differential equation for the change of storage value of compartment k ($\dot{x}_k = dx_k/dt$) as follows:

$$\dot{x}_k = \underbrace{\sum_{j=1}^n f_{kj}}_{\text{input to } x_k} + z_k - \underbrace{\sum_{j=1}^n f_{jk}}_{\text{output from } x_k} - y_k. \quad (1)$$

This equation is valid for all compartments ($k = 1, 2, \dots, n$), therefore the storage value functions $x_k(t)$ are defined by the system of differential equations in (1).

2.2. Throughflow analysis

Throughflow value T_k (or T_k^{out}) is the rate at which nutrients move through compartment k . It is defined as the sum of flows from compartment k to other compartments and to the envi-

¹ Ulanowicz (1983) has also studied cycling in ecological networks without resorting to matrix algebra.

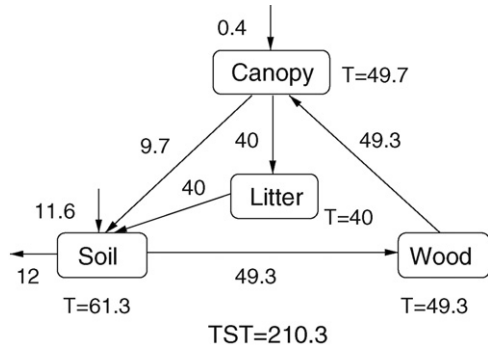


Fig. 2 – A weighted directed graph representing the amount in kg/ha/yr of Calcium that flows in the Hubbard brook temperate forest ecosystem (Finn, 1980). The numbers besides the arrows represent flow (f_{ij}), input (z_k) and output (y_k) rates at steady state. Throughflow of each compartment, and the total system throughflow (TST) is indicated.

ronment:

$$T_k = \sum_{j=1}^n f_{jk} + y_k. \tag{2}$$

In other words, the throughflow T_k is the total flow that leaves compartment k at a given time. T_k accounts for each flow that enters compartment k and does not contribute to storage (\dot{x}_k). As such, T_k is a measure of the activity at compartment k . Similarly, total system throughflow (TST), is the sum of all compartmental throughflows in the system:

$$TST = \sum_{k=1}^n T_k. \tag{3}$$

Therefore TST accounts for the total activity in the system at a given time. In Fig. 2, we show the flow, input, output, throughflow, and TST values for the Hubbard brook ecosystem model (Finn, 1980).

Combining Eqs. (1) and (2), we get

$$\dot{x}_k = \sum_{j=1}^n f_{jk} + z_k - T_k. \tag{4}$$

If each internal flow f_{ij} is expressed as a fraction (g_{ij}) of throughflow T_j of the donor compartment, then we define g_{ij} as follows:

$$g_{ij} = \frac{f_{ij}}{T_j} \text{ or } f_{ij} = g_{ij}T_j. \tag{5}$$

Note that by definition, we have $0 \leq g_{ij} \leq 1$ for all $i, j = 1, \dots, n$. Therefore the matrix $G = [g_{ij}]_{n \times n}$ can be viewed as a normalized version of the flow matrix $F = [f_{ij}]_{n \times n}$. Combining Eqs. (4)

and (5), and rearranging terms, we get

$$z_k - \dot{x}_k = T_k - \sum_{j=1}^n g_{kj}T_j.$$

We can rewrite this equation in matrix form as follows:

$$z - \dot{x} = (I - G)T$$

where I represents the identity matrix of size $n \times n$, and $T = [T_i]_{n \times 1}$ is a column vector of throughflows. Assuming that the matrix $I - G$ is invertible, we get the following equation:

$$N(z - \dot{x}) = T, \quad N = (I - G)^{-1}.$$

If we further assume that the ecosystem operates near steady state ($\dot{x} \approx 0$), the equation above becomes simpler:

$$Nz = T.$$

Note that at steady state, N , z and T are all constants. Therefore the matrix N is actually a linear mapping from the input vector z to the throughflow vector T :

$$N : z \mapsto T.$$

In other words, we can calculate the throughflow T generated by any input vector v by computing Nv . For example:

$$N \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \begin{bmatrix} N_{11} \\ N_{21} \\ \vdots \\ N_{n1} \end{bmatrix} = \bar{T}.$$

Here, the vector \bar{T} represents the throughflow generated at each compartment by a hypothetical unit input to the first compartment. As an example, $\bar{T}_3 = N_{31}$ represents the throughflow generated at compartment 3 by a unit input to compartment 1. Note that the compartments 1 and 3 need not be directly connected to each other for the value of N_{31} to be positive. In other words, $F_{31} = 0 < N_{31}$ is possible if there is an indirect flow from compartment 1 to compartment 3, such as

$$1 \rightarrow 2 \rightarrow 3.$$

In general, N_{ij} represents the throughflow generated at compartment i by a unit input to compartment j over all possible pathways. Assuming that the system is near steady state, we expect that $N_{ii} \geq 1$, as a unit input to compartment i will also be leaving that compartment at the same rate, and hence will generate a unit throughflow. But then how can N_{ii} be larger than 1, not just be equal to 1? Consider the following case where some fraction of a unit input to compartment i cycles and comes back to itself. In this case, compartment i will be experiencing a throughflow value of larger than 1 for a unit input to itself. Finn (1976) uses this exact idea to construct his cycling index, that $N_{ii} - 1$ represents the amount of throughflow generated at compartment i through cycling in the ecosystem.

For further information on the throughflow matrix N , and Network Environ Analysis (NEA) in general, we refer the reader to Patten et al. (1976), Patten (1978), Fath and Patten (1999), and Patten (1992).

3. Finn's cycling index

Finn defines his cycling index as the fraction of the total system throughflow (TST) that is generated through cycling:

$$FCI = \frac{TST_c}{TST} \quad (6)$$

He measures TST and TST_c using the linear algebraic development explained in Section 2.2. Algebraically, FCI is defined as a weighted sum of cycling efficiencies of all compartments in the ecosystem, where

$$\text{Cycling efficiency of compartment } i : C_i = \frac{N_{ii} - 1}{N_{ii}} \quad (7)$$

In other words, C_i is the ratio of the throughflow generated at node i through cycling only. Then the cycled portion of total system throughflow is

$$TST_c = \sum_{k=1}^n C_k T_k. \quad (8)$$

For an ecosystem model at steady state, Finn's cycling index is algebraically defined as follows:

$$FCI = \sum_{i=1}^n \frac{T_i}{TST} \cdot \frac{N_{ii} - 1}{N_{ii}}. \quad (9)$$

Note that the term (T_i/TST) represents the fraction of the total activity of the system driven by compartment i . Therefore the cycling efficiency of a more active compartment has a higher influence on the cycling index.

4. Simulation based cycling index

Finn's idea of quantifying cycling in an ecosystem is to measure the fraction of the total system throughflow (TST) that is due to cycling (6). Most studies related to cycling in ecosystems are based on Finn's original work (Finn, 1976), and use matrix algebra to study and quantify cycling, as described in preceding sections. Following the idea of Finn, we propose a drastically different way to compute the cycling index. Our method consists of a simple counting algorithm, and is based on the output of a "particle tracking" simulation (Kazanci and Tollner, in preparation; Tollner and Kazanci, 2007). Unlike most previous methods (Allesina and Ulanowicz, 2004; Finn, 1980; Patten and Higashi, 1984; Han, 1997), no matrix algebra is used in the process. Flexibility of the particle tracking algorithm enables us to implement Finn's idea of computing cycling index accurately. Here are the questions that we investigate in this paper:

Suppose that there is a "super microscope" that enables us to see and track individual molecules (or quanta of energy) that flow in the ecosystem. If we use Finn's idea of cycling (just the idea, not the algebraic definition) and the data obtained by this "super microscope", would the results agree? In other words, how well does the algebraic definition (9) reflect the

actual cycling in the ecosystem? Are there "better" ways to quantify cycling in an ecosystem, given the incredible resolution this "super microscope" provides? How would this new method compare to FCI?

4.1. Particle tracking algorithm

First, we would like to describe what "particle tracking algorithm" is. In short, it is an individual based simulation method (also known as *agent based models*) that deduces its rules (on how an individual particle will move) automatically from the differential equation representation (1) of the ecosystem model. Therefore the time courses of compartment storage values and flow rates agree with the differential equation simulation. This is mathematically a challenging task, so we refer the reader to Tollner and Kazanci (2007) and Kazanci and Tollner (in preparation) for more information.

Perhaps the easiest way to demonstrate how particle tracking works is to show the output of a sample simulation. In Fig. 3, we show one of the outputs of a particle tracking simulation for the Hubbard brook ecosystem model (Fig. 2). A particle represents a unit amount of the flow currency specific to the model, which may be N, C, P, energy, biomass, etc. Particle Tracking algorithm assigns a unique label to each particle, and tracks their movement, and records this data in text files.

A differential equation simulation tracks the storage values of compartments changing over time. Particle tracking algorithm tracks the same information. Furthermore, it also identifies which particles form the storage value of each compartment. For example, the differential equation simulation will inform us that the storage value of compartment Litter at time $t = 14.3$ is 23.7 kg/ha. If a particle is defined initially as 100 g/ha, particle tracking simulation (PTS) will give the specific tags of all 237 particles that are contained in compartment Litter at time $t = 14.3$.

If a system is at steady state, there will be no changes in storage values, therefore the differential equation simulation will give constant values for storage values over time. Therefore an ODE simulation creates the illusion that the system has "stopped". In fact the system is still active and working, but rate of input and output of each compartment stays equivalent in time. Particle tracking simulations enable us to see this activity hidden by ODE solutions. For a steady state ecosystem model, PTS will demonstrate that although the storage values of compartments stay constant, the names of particles that are stored in compartments continuously change.

Ecosystem models are open systems, therefore new particles will be introduced in the system continuously. Each of these new particles will have new tags, therefore the parti-

```

:
Particle 12157 flows from Canopy to Litter at time 29.872
Particle 12148 flows from Wood to Canopy at time 29.878
Particle 12149 dissipates from Soil at time 29.916
Particle 12158 is input to Soil at time 29.921
Particle 12157 flows from Litter to Soil at time 29.94
:

```

Fig. 3 – Partial output of a particle tracking simulation for the Hubbard brook ecosystem model shown in Fig. 2.

cle tag numbers will increase during a simulation. However particles will also dissipate from the ecosystem to the environment. PTS cleverly recycles the memory freed by dissipated particles, records their data in text files (stored in computer's hard drive), and reuses this available memory for new particles. Through the extensive use of such computational tricks, a PTS will take only a few seconds to run.

Particle tracking algorithm is a sophisticated, accurate, and yet, a feasible numerical simulation method. For further information, we refer the interested reader to Tollner and Kazanci (2007) and Kazanci and Tollner (in preparation). Next, we explain how one can compute cycling index using PTS.

4.2. Cycling index via particle tracking simulation

The output of PTS shown in Fig. 3 is not easy to utilize for cycling index computation. In Fig. 4, we show another output of PTS which is more suitable for our purpose. This output contains the pathways of all particles that are either in the system, or have dissipated from the system. We define the pathway of a particle as an ordered list of compartments visited by that particle. We use the pathway data of many particles to compute an approximate value for the cycling index. The more the number of particle pathways, the more accurate this approximate cycling index is.

Fig. 4 shows the pathway data of seven particles. Next, we demonstrate how we can compute Finn's cycling index based on the pathway data of these seven particles. Note that only seven particle pathways is not sufficient to get an accurate estimate for cycling index, so this computation is for demonstration purposes only. Finn defines his cycling index as follows:

$$FCI = \frac{TST_c}{TST} \tag{10}$$

The total system throughflow (TST) accounts for each time a particle moves from one compartment to another, or to the environment. Therefore TST can be computed by counting the number of compartments visited by each particle. The cycled portion (TST_c) of TST accounts for the total system throughflow that occurs due to cycling. We can compute TST_c by counting each time that a particle moves out of a compartment which was visited by that particle previously. Table 1 shows the calculation of TST, TST_c, and FCI using the

```

      :
Path of Particle 12144: 0 2 3
Path of Particle 12152: 3 1 0 3 1 0 3 1 0 2 3 1 0 2 3 1 0 3
Path of Particle 12141: 3 1 0 2 3 1 0 2 3 1 0 3
Path of Particle 12153: 3 1 0 3 1 0 2
Path of Particle 12156: 3 1 0 2 3 1 0 2
Path of Particle 12154: 3 1 0 2 3 1 0 2 3 1 0 2 3
Path of Particle 12155: 3 1 0 3
      :
    
```

Fig. 4 – Partial output of particle tracking simulation for the Hubbard brook ecosystem model shown in Fig. 2. Here, the numbers 0, 1, 2 and 3 correspond to compartments Canopy, Wood, Litter, and Soil respectively. For example, Particle 12,144 moved from compartment Canopy to Soil, and then to Wood.

**Table 1 – The partial particle pathway data from Fig. 4 is used to compute an approximate cycling index for the model in Fig. 2. Each compartment visit contributes to total system throughflow, therefore the values in TST column are computed by simply counting the number of compartments visited by each particle. Compartments visited by a particle more than once are underlined. The values in the last column represent throughflows that contribute to cycling, and are computed by simply counting the number of underlined compartments visited by each particle. The simulation based cycling index is then computed as follows:
FCI = TST_c/TST ≈ 45/67 ≈ 0.67.**

Particle	Pathway	TST	TST _c
12144	0 2 3	3	0
12152	3 1 0 <u>3</u> 1 0 <u>3</u> 1 0 <u>2</u> <u>3</u> 1 0 <u>2</u> <u>3</u> 1 0 <u>3</u>	18	14
12141	3 1 0 <u>2</u> <u>3</u> 1 0 <u>2</u> <u>3</u> 1 0 <u>3</u>	12	8
12153	3 1 0 <u>3</u> 1 0 2 3	8	4
12156	3 1 0 <u>2</u> <u>3</u> 1 0 2 3	9	5
12154	3 1 0 <u>2</u> <u>3</u> 1 0 <u>2</u> <u>3</u> 1 0 <u>2</u> <u>3</u>	13	9
12155	3 1 0 <u>3</u>	4	1
Total:		67	45

PTS output from Fig. 4. Using these partial pathway data, we approximate the Finn's cycling index as follows:

$$FCI = \frac{TST_c}{TST} \approx \frac{45}{67} \approx 0.67.$$

Note that we did not use Finn's algebraic definition of cycling index 9 which was based on the matrix N. Actually, we did not use any linear algebra at all. All we used was a simple counting algorithm and the output of a particle tracking simulation. We still call it "Finn's cycling index", because we did use Finn's original idea of formulating his cycling index, that FCI = (TST_c/TST). Then we investigate whether Finn's matrix based definition (9) agree with this computational version.

Note that PTS is a stochastic method, therefore each run generates different pathway data for the same model. Computing an approximate value for FCI based on a stochastic method raises some issues about the stability and accuracy of the computation itself:

1. How does the cycling index vary with respect to the amount of pathway data used?
2. What is the minimum number of pathways needed to obtain an accurate cycling index?
3. How does the computed cycling index change when we run multiple instances of the simulation?
4. How does this approximate cycling index value compare to the original FCI?

We address all the above questions through Fig. 5. Particle tracking algorithm is a stochastic numerical method, therefore each simulation looks slightly different. Furthermore, it provides only an approximation for FCI, it is not an exact computation. In Fig. 5, we see that the PTS based FCI ranges between 0.76 and 0.82 among four PTS runs, when only 100 particle pathway data are used. The oscillations in the simulation based FCI die out as the number of pathways used in the computation increases. Moreover the discrepancy between the values for multiple runs vanishes for higher number of

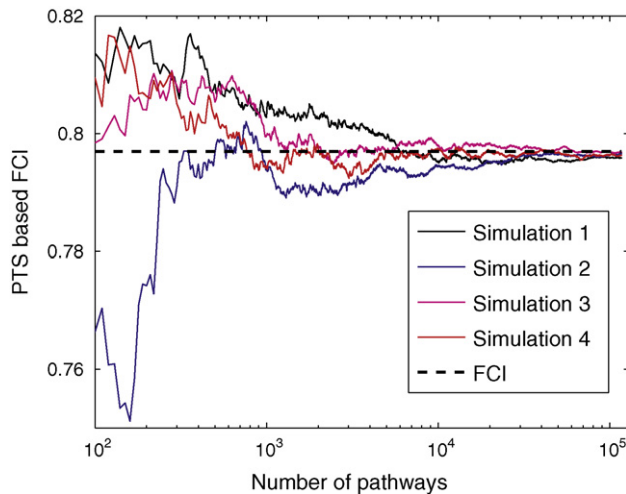


Fig. 5 – Finn cycling index is computed using the pathway data generated by four different Particle Tracking Simulations (PTS) for the same model, the Hubbard Brook temperate forest ecosystem model. PTS is a stochastic simulation method, therefore each run generates different pathway data. The x– axis represents the amount of pathway data used to approximate the FCI.

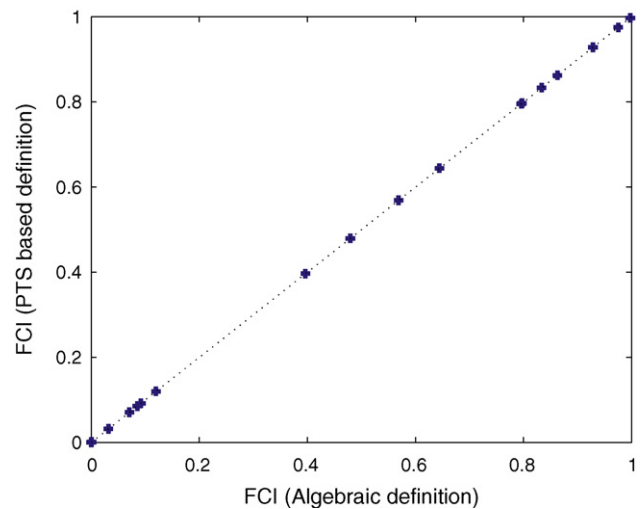


Fig. 6 – Comparison of particle tracking simulation (PTS) based cycling index and Finn's cycling index (FCI) over a collection of 22 ecosystem models of sizes ranging from 4 to 12 compartments.

pathways. Therefore particle tracking algorithm proves to be a feasible method for accurate computation of cycling index.

During a PTS run, particle pathway data keep accumulating over time. Therefore longer PTS runs are required to obtain more particle pathway data, which is needed for accuracy. This raises a question of feasibility: How much computing time is needed to compute the PTS based FCI accurately? The answer will depend on the model. In general, 20,000 particle pathways are enough to compute FCI within 0.1% error. For most models, it takes less than a second for PTS to generate these data on a modern single CPU computer.

In general, individual based models are built specific to an ecosystem. PTS however, is ready to use with any ecosystem model with a differential equation representation. Actually, PTS uses the exact same model input format as EcoNet software (Kazanci, 2007)(<http://eco.engr.uga.edu>). In Fig. 6, we compare the simulation based FCI to the original FCI over a collection of 22 ecosystem models of sizes ranging from 4 to 12 compartments. The collection includes both aquatic and terrestrial networks, both published and unpublished.

5. Discussion

The work we present in this paper is the first application of a new method, the Particle Tracking Algorithm (Tollner and Kazanci, 2007; Kazanci and Tollner, in preparation). It is a very capable tool, as it provides access to pathway history of individual particles (Fig. 3). This comprehensive data output provides a framework suitable for study of many interesting properties regarding ecosystems, such as residence time, dominance of indirect effects, thermodynamics etc.

However, rather than trying to study a new aspect of ecosystems, we chose to re-investigate a well-known prop-

erty, Finn's cycling index. To define cycling index, Finn uses a deterministic continuous process (a differential equation) and linear algebra. In contrast, we use a stochastic process (particle tracking algorithm) and a basic counting algorithm on pathway history of particles. The fact that both methods agree does not only demonstrate that PTS is an accurate, feasible and capable individual based method, but also confirms that Finn's linear algebraic definition of his cycling index is an accurate description of his initial cycling idea (TST_c/TST).

Furthermore, we observe that one can only acquire a snapshot of the ecosystem, and the exchange of currency over that period of time occurs along pathways of finite length. Therefore, using matrix algebra might overestimate the amount of currency exchange during that period of time. Since particle tracking algorithm simulate the real time flow of currency in the ecosystem, it offers a way to evaluate (accurately) the activity of the ecosystem over given period of time.

The study of FCI, a well-known property, rather than a new aspect of ecosystems, enabled us to focus on the feasibility, accuracy and convergence properties of this new sophisticated simulation method. The encouraging results show that we can safely rely on particle tracking algorithm to study system level, organizational and dynamic behavior of ecosystems.

An important aspect of particle tracking algorithm is that the ecosystem model need not be at steady state. PTS is capable to run ecosystem models that evolve in time. Based on such PTS output, we can compute an average cycling index for ecosystems that evolve because of an ongoing climate change, or are subject to periodic (seasonal, day to day) changes. Furthermore, it seems feasible to compute cycling index over a time window, and advance this window in time to obtain a dynamic cycling index.

We should further emphasize the most essential property of PTS, that it is fully compatible with the differential equation representation of the ecosystem model. Note that it is possible to construct a similar numerical method that labels and

moves particles among compartments, like PTS. However, the dynamic behavior of this method will probably be different than the differential equation representation, unless careful attention is given to its compatibility with the *master equation* (Gillespie, 1992, 2000). Therefore any ecosystem property computed using such a simulation method will probably result in different value than the differential equation based formulation.

We believe that differential equation representation of ecosystems preserve causality. An individual based method that behaves differently than the differential equation representation requires further attention. On the other hand, PTS provides compatible simulation results with the differential equation representation, and gives detailed information about the ecosystem behavior, dynamics and organization.

Acknowledgments

We would like to thank the Systems and Engineering Ecology group (<http://eco.engr.uga.edu/S&EE>) at the University of Georgia for their valuable support. L. Matamba wishes to thank the Department of Mathematics and the Faculty of Engineering at the University of Georgia, who co-founded this work in the form of an internship during the Summer 2007. We would like to thank B. C. Patten and S. J. Whipple for their insightful comments.

REFERENCES

- Allesina, S., Ulanowicz, R.E., 2004. Cycling in ecological networks: Finn's index revisited. *Computational Biology and Chemistry* 28 (3), 227–233.
- Costanza, R., Norton, B.G., 1992. *Ecosystem Health: New Goals for Environmental Management*. Island Press.
- Fath, B.D., Patten, B.C., 1999. Review of the foundations of network environ analysis. *Ecosystems* 2, 167.
- Finn, J.T., 1976. Measures of ecosystem structure and function derived from analysis of flows. *Journal of Theoretical Biology* 56 (2), 363–380.
- Finn, J.T., 1980. Flow analysis of models of the Hubbard Brook ecosystem. *Ecology* 61 (3), 562–571.
- Gillespie, D.T., 2000. The chemical Langevin equation. *The Journal of Chemical Physics* 113 (1), 297.
- Gillespie, D.T., 1992. A rigorous derivation of the chemical master equation. *Physica A* 188, 404–425.
- Han, B.P., 1997. On several measures concerning flow variables in ecosystems. *Ecological Modelling* 104, 289–302.
- Higashi, M., Patten, B.C., 1986. Further aspects of the analysis of indirect effects in ecosystems. *Ecological Modelling* 31 (1), 69–77.
- Kazanci, C., 2007. Econet: a new software for ecological modeling, simulation and network analysis. *Ecological Modelling* 208 (1), 3–8.
- Kazanci, C., Tollner, E.W. Particle tracking: an individual based approach for ecological network analysis, in preparation.
- May, Robert M., 1981. A cycling index for ecosystems. *Nature* 292 (July (5819)), 105–105 (10.1038/292105a0).
- Odum, E.P., 1969. The strategy of ecosystem development. *Science* 164 (3877), 262–270.
- Patten, B.C., 1978. Systems approach to the concept of environment. *Ohio Journal of Science* 78 (4), 206–222.
- Patten, B.C., 1992. Energy, emergy and environs. *Ecological Modelling* 62 (1), 29–69.
- Patten, B.C., Higashi, M., 1984. Modified cycling index for ecological applications. *Ecological Modelling* 25, 69–83.
- Patten, B.C., Bosserman, R.W., Finn, J.T., Cale, W.G., 1976. Propagation of cause in ecosystems. In: Patten, B.C. (Ed.), *Systems Analysis and Simulation in Ecology*, vol. 4, Academic Press, New York, pp. 457–579.
- Tollner, E.W., Kazanci, C., 2007. Defining an ecological thermodynamics using discrete simulation approaches. *Ecological Modelling* 208 (1), 68–79.
- Ulanowicz, R.E., 1983. Identifying the structure of cycling in ecosystems. *Mathematical Biosciences* 65 (2), 219–237.
- Wulff, F., Ulanowicz, R.E., 1989. A comparative anatomy of the Baltic sea and Chesapeake bay ecosystems. *Network Analysis in Marine Ecosystems*. Springer-Verlag, London, pp. 232–258.