Operads as a potential foundation for systems of systems

David I. Spivak

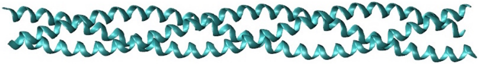
dspivak@math.mit.edu
Mathematics Department
Massachusetts Institute of Technology

Presented on 2016/03/22

at the Mathematical Biosciences Institute
Generalized Network Structures and Dynamics Workshop
The Ohio State University

Outline

1. **Introduction**
   - What is meant by "foundation for systems"
   - Pictures of composition styles
   - The operadic approach
2. Different operads for different systems
3. Compositional mappings
4. Conclusion



What a foundation should provide

A foundation for a subject should include a language and a theory, which

- articulates the subject faithfully and flexibly,
- connects individual sub-languages and sub-theories,
- captures relevant examples, possibly quite disparate in nature,
- delivers useful tools for synthesis and analysis.

My goal is to show that operads serve as a foundation for systems.

What are systems?

- A system includes many parts, assembled into a whole.
- The whole can again be a part in a larger system.
  - And a part may be considered, in a smaller context, as a whole.
  - Example: a node in a network may house a subnetwork.
- We can observe a system in a variety of ways.
  - Its behavior is not reducible to the behaviors of its parts,
  - however, it is reducible to those *plus* all their interactions.

A system is an assemblage of subsystems,
whose behavior emerges from local interactions.

What are systems?

- A system includes many parts, assembled into a whole.
- The whole can again be a part in a larger system.
  - And a part may be considered, in a smaller context, as a whole.
  - Example: a node in a network may house a subnetwork.
- We can observe a system in a variety of ways.
  - Its behavior is not reducible to the behaviors of its parts,
  - however, it is reducible to those *plus* all their interactions.

A system is an assemblage of subsystems,
whose behavior emerges from local interactions.

System composition styles

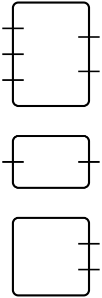
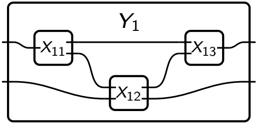
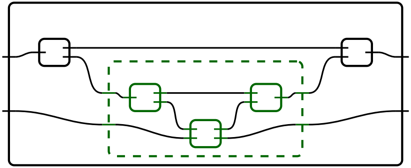
We can classify systems by their composition-style:

- Feedforward composition (no cycles allowed):
  - nodes in a feedforward neural net, passing activation levels
  - cells in a spreadsheet, returning values
- Feedback composition: any output can be fed into any input
  - neurons in a brain region, passing spike trains
  - modules in a computer program, returning values
- Port graph composition: no distinction between inputs and outputs
  - electrical circuits—resistors, capacitors, inductors—transferring current
  - systems of constraints, matching values

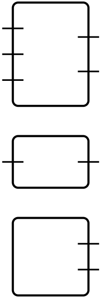
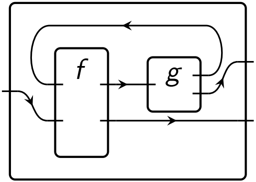
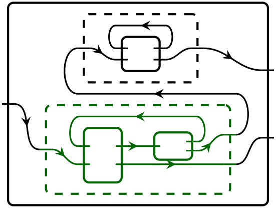
Each type of composition-style makes sense for certain types of systems.

Feedforward composition style

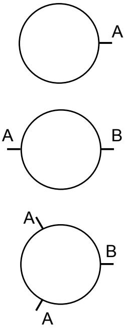
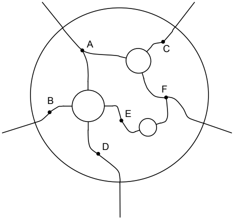
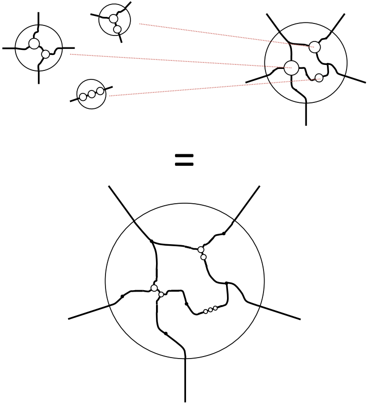
A composition style is a choice of interfaces, composition, and nesting.

Interfaces	Composition	Nesting
		

Feedback composition style

Interfaces	Composition	Nesting
		

Port (hyper-)graph composition style

Interfaces	Composition	Nesting
		

The operadic approach

- Operad = composition style, mathematically formalized.
    - Feedback, feedforward, port graph, many others.
- Algebra = type of system, adhering to composition rules.
    - Dynamical systems, spreadsheets, electrical circuits, etc.
- Maps connect different types of systems or composition styles.
    - Compositional mappings formalize cross-disciplinary collaboration.
    - Example: bifurcation diagrams are compositional mappings.

The operadic approach:

> *To model a system, consider what type of system it is, including composition style. If you want to ask questions about it, find compositional mappings.*

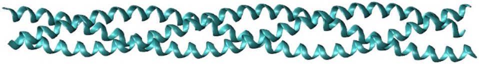
Category Theory

- Operads are a sub-discipline of category theory (CT).
- Since its invention in the 1940s, CT has revolutionized pure math.
  - It is able to connect disparate disciplines into a unified framework.
  - It abstracts common themes from algebra, topology, and logic.
  - Opinion: CT is the gateway to accessing the world of *pure math*.
- Category theory has been *applied* outside of math as well.
  - *Computer science*: functional programming, databases;
  - *Physics*: Feynman diagrams, quantum information theory;
  - *Design*: cyber-physical systems, resource theory;
  - *Materials science*: hierarchical protein materials.

Plan for the talk

- Briefly discuss the "compromising assumptions" for networks.
  - How this approach might deal with multiplicity, change, and noise.
- Give several examples of operads and algebras, relevant to networks.
  - Computer science,
  - bio-materials,
  - dynamical systems,
  - neuroscience.
- Discuss compositional mappings and what they do for us.
  - Formal cross-disciplinary communication,
  - computational techniques.
- Explain that steady states of dynamical systems are compositional.

Outline



Assumptions that compromise validity for simplicity

In the workshop announcement, the organizers listed four assumptions:

- The network has only a single type of interaction among entities.
- The interactions among entities can be completely described as pairwise.
- The network is static over time.
- The impact of noise can be ignored.

Specific responses, in the case of "mode-dependent wiring diagrams":

- Wires can carry different signaling alphabets or languages.
- Wires can simultaneously connect many interfaces (nodes).
- Mode-dependent WDs change topology in response to local dynamics.
- Noise introduces non-determinism in the dynamical systems.

But these responses actually miss a bigger point.

Broader outlook

- Sometimes, it can be *useful* to take on the assumptions; sometimes
  - we can assume the network shape does not change,
  - or we find a network with only pairwise interactions,
  - or we want to study the behavior in a noise-free environment.
  - Sometimes we can assume our dynamical systems are linear.
- Do we need a single model for networks, once and for all?
  - Must we forever dispense with the above four assumptions?
- Alternative approach: a framework, where you can choose your style.
  - We should add and remove assumptions based on convenience.
  - We want to compare results from different sets of assumptions.

Different operads and algebras make different assumptions; the power comes from mapping between them.

Examples from computer science

We can think of operads in terms of *modularity*.

- Modularity is building complex from simple building blocks.
  - This notion is important in almost any scientific discipline.
  - Modularity is highly-regarded in computer science.
  - The modules are interfaces, which are composed and nested.
- Modularity comes in many forms, each with a composition style.
- Examples of composition styles and corresponding system types:
  - Context free grammars and domain specific languages.
  - Port graphs and database queries.
  - We'll discuss these examples (of operads and algebras) below.

Every context-free grammar (CFG) is an operad

The composition style of postal addresses: [1]

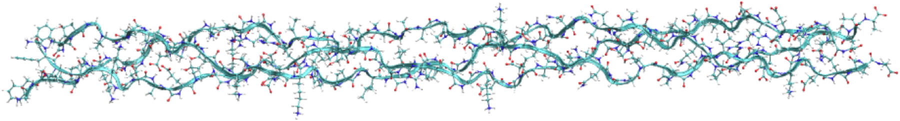
⟨postal-address⟩	::=	⟨name-part⟩ ⟨street-address⟩ ⟨zip-part⟩
⟨name-part⟩	::=	⟨personal-part⟩ ⟨last-name⟩ ⟨opt-suffix-part⟩ ⟨EOL⟩
	|	⟨personal-part⟩ ⟨name-part⟩
⟨personal-part⟩	::=	⟨first-name⟩|⟨initial⟩ "."
⟨street-address⟩	::=	⟨house-num⟩ ⟨street-name⟩ ⟨opt-apt-num⟩ ⟨EOL⟩
⟨zip-part⟩	::=	⟨town-name⟩ "," ⟨state-code⟩ ⟨ZIP-code⟩ ⟨EOL⟩
⟨opt-suffix-part⟩	::=	"Sr." | "Jr." | ⟨roman-numeral⟩ | ""
⟨opt-apt-num⟩	::=	⟨apt-num⟩ | ""

- Everything in ⟨brackets⟩ is an interface.
- Each line is a composition: many put together to form one.
- Context free grammars are used to create domain-specific languages.
- In particular, each DSL is an operad and an algebra on it.

[1] Copied verbatim from Wikipedia page on Backus-Naur Form.

A DSL for hierarchical protein materials

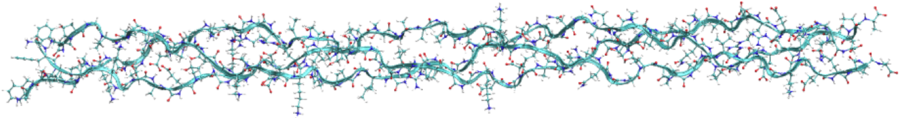
- We compose complex proteins from simpler ones.
- Collagen, the most common protein in mammals, is hierarchical.
- That is, it's formed as a nested composition.
- A DSL called *Matriarch* easily creates structures like this.



```
a1             =  chain(seq1)
a2             =  chain(seq2)
hel1           =  helix(a1, rad=1.5, pitch=9.5, handed=L)
hel2           =  helix(a2, rad=1.5, pitch=9.5, handed=L)
helhel1        =  helix(hel1, rad=4, pitch=85, handed=R)
helhel2        =  helix(hel2, rad=4, pitch=85, handed=R)
helhel1rot     =  rigidMotion(helhel1, rotate=120, shift=2.8)
helhel2rot     =  rigidMotion(helhel2, rotate=240, shift=-5.6)
tropocollagen  =  overlay(helhel1, helhel1rot, helhel2rot)
collagen       =  makeArray(tropocollagen, 1000, 1000, distance=8.1)
```

A DSL for hierarchical protein materials

- A fibril of collagen is an array of tropocollagen molecules.
- Each molecule of tropocollagen is a right-handed triple helix.
- Each of its three strands is a left-handed helix.
- Each of these individual helices is a chain of many amino acids.

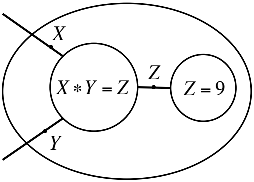


a1	=	chain(seq1)
a2	=	chain(seq2)
hel1	=	helix(a1, rad=1.5, pitch=9.5, handed=L)
hel2	=	helix(a2, rad=1.5, pitch=9.5, handed=L)
helhel1	=	helix(hel1, rad=4, pitch=85, handed=R)
helhel2	=	helix(hel2, rad=4, pitch=85, handed=R)
helhel1rot	=	rigidMotion(helhel1, rotate=120, shift=2.8)
helhel2rot	=	rigidMotion(helhel2, rotate=240, shift=-5.6)
tropocollagen	=	overlay(helhel1, helhel1rot, helhel2rot)
collagen	=	makeArray(tropocollagen, 1000, 1000, distance=8.1)

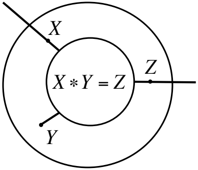
Port graphs and constraint systems

Recall the "circle diagrams" or port graph operad.

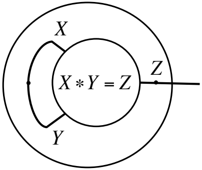
- This composition style allows multiple connections between ports.
- The interfaces (circles) can be inhabited by constraints (relations).
  - Each subsystem says "I am happy whenever…."
  - And the whole system is happy when its interconnected pieces are.



"all pairs of integers (*X,Y*) whose product is 9"



"all pairs of integers (*X,Z)* in which *Z* is divisible by *X*."

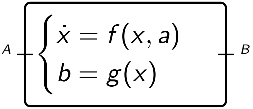


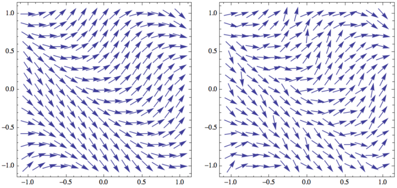
"all perfect squares *Z*"

Another example: dynamical systems

Dynamical systems are "machines" that evolve according to rules.

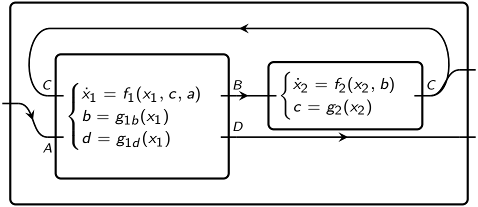
- A dynamical system is *open* if it may receive and send signals.
- The received signals can influence the trajectory of the system.
- The system outputs signals, as a function of the current state.



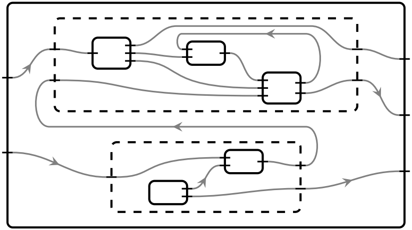


- Each inhabits a box—an interface—with input and output ports.

Open dynamical systems (DS's) can be composed by interconnection:



Nesting is coherent: the result is the same no matter how you group.

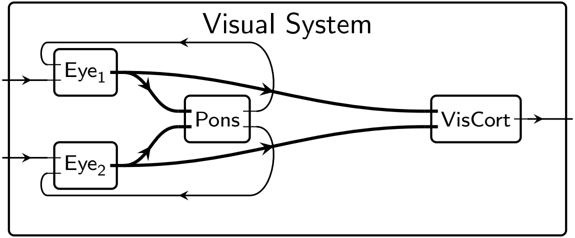


You can group physically (brain region) or logically (cognitive function).

Mode-dependent networks of dynamical systems

Real-world networks need not be static; they can be “adaptive”.

- Change in topology is often dictated by dynamics on the nodes.
  - Nodes send each other signals, accordingly updating their state.
  - The state of each node determines how it “wants” to communicate.
  - A person extends a hand, an eye closes, a cell opens a channel.
  - Each node has a set of “communicative modes”.



Operad for mode dependent systems

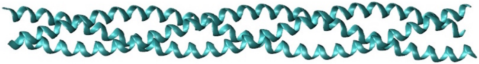
There is an operad whose composition style is mode-dependent.

- An interface is a box with a set of modes determining its ports.
- What is a composition of nodes $N_1, \ldots, N_k$ forming a higher node P?
  - It is a contract: given a communicative mode for each N_i,...
  - these modes together determine a wiring pattern for the network
  - as well as a communicative mode for P.

This operad supports adaptive networks of dynamical systems.

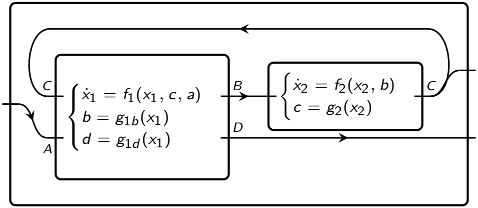
- Each box houses a DS; its mode is determined by the DS's state.
- The current wiring shape causes certain signals to arrive.
- These lead to a state evolution, which may cause boxes to reorganize.

Outline



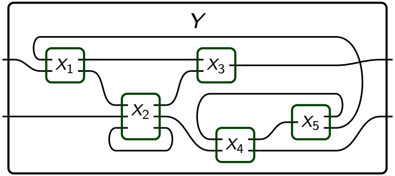
Recall: feedback composition of dynamical systems

Above we discussed, open dynamical systems can be composed by interconnection:

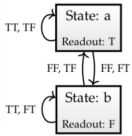


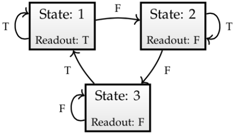
Discrete DS's also have this composition style

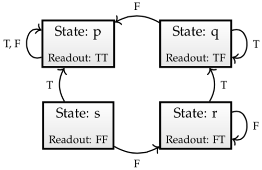
There is more than one type of system with this composition style.



- Above we were really describing *continuous* dynamical systems.
- A *discrete* dynamical system instead has a discrete state space.
    - The input and output signals are also discrete.
    - Boolean networks are a special case, where all signals are T or F.

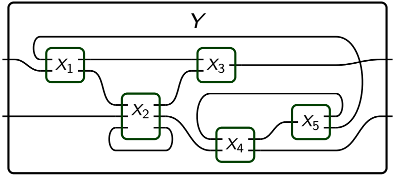






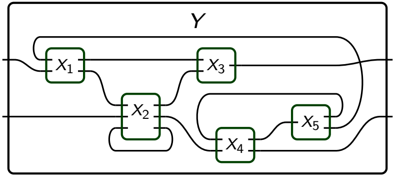
Is Euler's method compositional?

We've mentioned two types of system that have this composition style.



- Continuous dynamical systems are governed by ODEs.
- Discrete dynamical systems are governed by discrete state transitions.
- Euler's method converts a continuous system to a discrete one:
  - Choose $\epsilon > 0$ and discretely transition according to derivative.
  - Is Euler's method compositional for systems of systems?

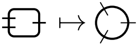
Compositional mappings

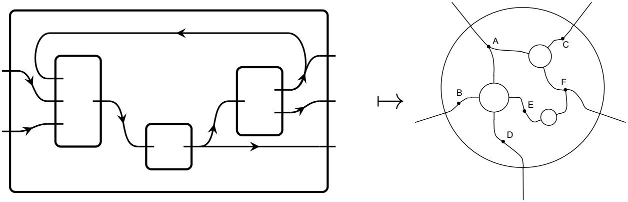


- Euler got lucky this time: yes, it's compositional.
  - This means we can put together discretized pieces without loss.
- Explaining "compositional mapping" by analogy to *averaging*.
  - Suppose wiring doesn't matter. Each box above is a population.
  - Each population has an average height.
  - Can you put together the averages to get the avg. of the whole?
- Certain invariants are compositional, others are not.

Changing the operad

- Feedback composition also supports constraint systems.
- To make sense of this, we first apply a change of operad.
  - This way, we can recast constraint systems in the richer language.

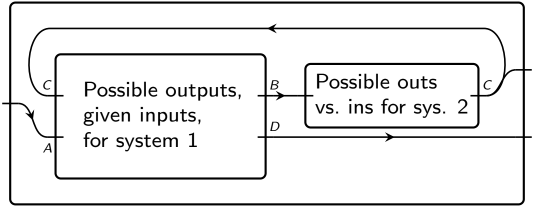




- Now, dynamical systems can be mapped to constraint systems.
  - Because both are system types with the same composition style.

Properties of dynamical systems

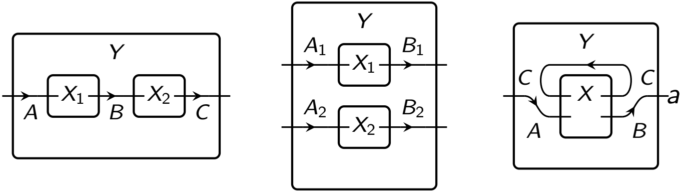
- A compositional mapping from dynamical systems to constraints:
  - Each system has a set of all possible input/output streams.
  - This is a relation on input streams and output streams.
  - Knowledge about subsystems produces knowledge about whole.



- This mapping is compositional:
  - One derives properties of the whole from properties of the parts.

Feedback composition supports matrix arithmetic

Perhaps surprisingly, the same operad supports matrix arithmetic.

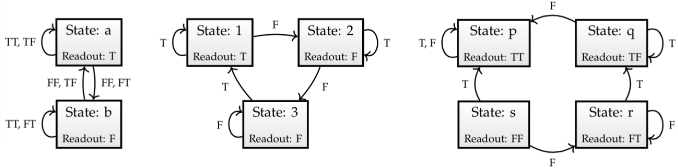


- Given matrices on the inner boxes, one is produced on the outer box.
  - Serial composition is *matrix multiplication*, $M_1 M_2$.
  - Parallel composition is *Kronecker product*, $M_1 \otimes M_2$.
  - Feedback composition is *partial trace*, $\mathrm{Tr}_{A,B}^{C} M$.
- Nesting works correctly: any wiring diagram makes sense.

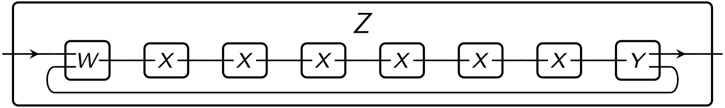
Steady state matrices

There is a compositional mapping from dynamical systems to matrices.

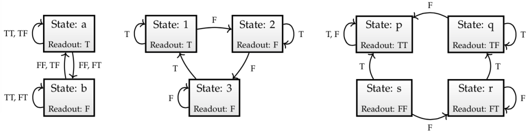
- Given a dynamical system, what is the dimension of the matrix?
  - There is an entry for each input-output pair.
  - For example, input=TT and output=F, or input=TF and output=T
- The value in an entry is the number of steady states.
  - With input=TT, how many steady states output F?
- "Taking steady state matrices" is a mapping, and it's compositional.
- For continuous DS, the steady state matrix is the bifurcation diagram.



Example



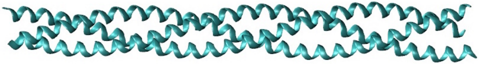
In box W, X, and Y respectively, put the following systems.



The steady state matrices, including for the total (5238-state) system are:

$$w = \begin{pmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix} \qquad x = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \qquad y = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \end{pmatrix} \qquad z = \mathrm{Tr}(wx^6y) = \begin{pmatrix} 9 & 7 \\ 1 & 1 \end{pmatrix}$$

Outline



Acknowledgments

- Collaborators:
  - Discrete dynamical systems: Dylan Rupel
  - Continuous dynamical systems: Dmitry Vagner and Eugene Lerman
  - Mode-dependent networks: Joshua Z. Tan
  - Matriarch (materials architecture): Tristan Giesa, Ravi Jagadeesan, and Markus J. Buehler
- Useful conversations: Patrick Schultz, Christina Vasilakopoulou, and Ryan Wisnesky.
- Grant support at various stages: ONR, AFOSR, NASA, NSF.

Is this a convincing foundation?

- The emphasis has been on compositionality and systems of systems.
  - There are many different operads = composition styles.
  - Each supports many different algebras = system types.
  - One can apply a change of operad or map between algebras.

Is this a convincing foundation?

- The emphasis has been on compositionality and systems of systems.
  - There are many different operads = composition styles.
  - Each supports many different algebras = system types.
  - One can apply a change of operad or map between algebras.
- *Something* has been articulated; is it useful for networks?
  - No talk of central nodes, small-world networks, etc.
  - Q: Are these as relevant when the four assumptions are dropped?
  - We've articulated a framework to formally add/drop assumptions.

Is this a convincing foundation?

- The emphasis has been on compositionality and systems of systems.
  - There are many different operads = composition styles.
  - Each supports many different algebras = system types.
  - One can apply a change of operad or map between algebras.
- *Something* has been articulated; is it useful for networks?
  - No talk of central nodes, small-world networks, etc.
  - Q: Are these relevant when the four assumptions are dropped?
  - We've articulated a framework to formally add or drop assumptions.
- The examples have been quite disparate.
  - Protein materials, databases, dynamical systems on networks.
  - Perhaps we've cast too wide a net; certainly wider than networks.

Is this a convincing foundation?

- The emphasis has been on compositionality and systems of systems.
  - There are many different operads = composition styles.
  - Each supports many different algebras = system types.
  - One can apply a change of operad or map between algebras.
- *Something* has been articulated; is it useful for networks?
  - No talk of central nodes, small-world networks, etc.
  - Q: Are these relevant when the four assumptions are dropped?
  - We've articulated a framework to formally add or drop assumptions.
- The examples have been quite disparate.
  - Protein materials, databases, dynamical systems on networks, and more.
  - Perhaps we've cast too wide a net; certainly wider than networks.
- Tools for analysis: one new technique provided.
  - Steady states for nonlinear systems via matrix arithmetic.
  - This whole approach is at an early stage of development.
  - I'm happy to talk more. THANK YOU!