

MACHINE-READABLE PHYSICS FOR SPACE-TIME-DEPENDENT SIMULATIONS IN MBE

Jerome Szarazi, Koneksys

Conrad Bock, NIST

Overview

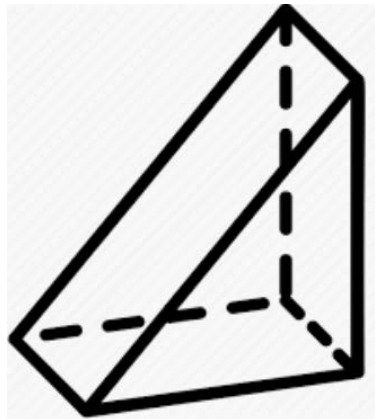
- Simulation in engineering
- Tools and approaches
- PDEs as model definition
- Machine readable physics
 - Motivation and approach
 - From equations to computation
 - From physics to solvers
- Benefits and usage
- Summary

Overview

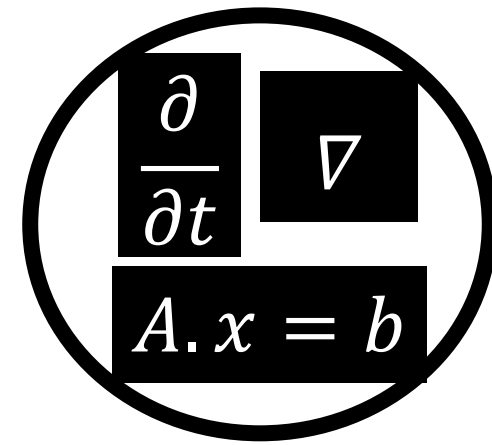
- Simulation in engineering
- Tools and approaches
- PDEs as model definition
- Machine readable physics
 - Motivation and approach
 - From equations to computation
 - From physics to solvers
- Benefits and usage
- Summary

What do we simulate?

- A virtual product is a mathematical representation of product geometry created or represented with software (CAD)
- The behavior of a virtual product is specified by partial differential equations (PDEs) of physics that connects configuration variables to loads.



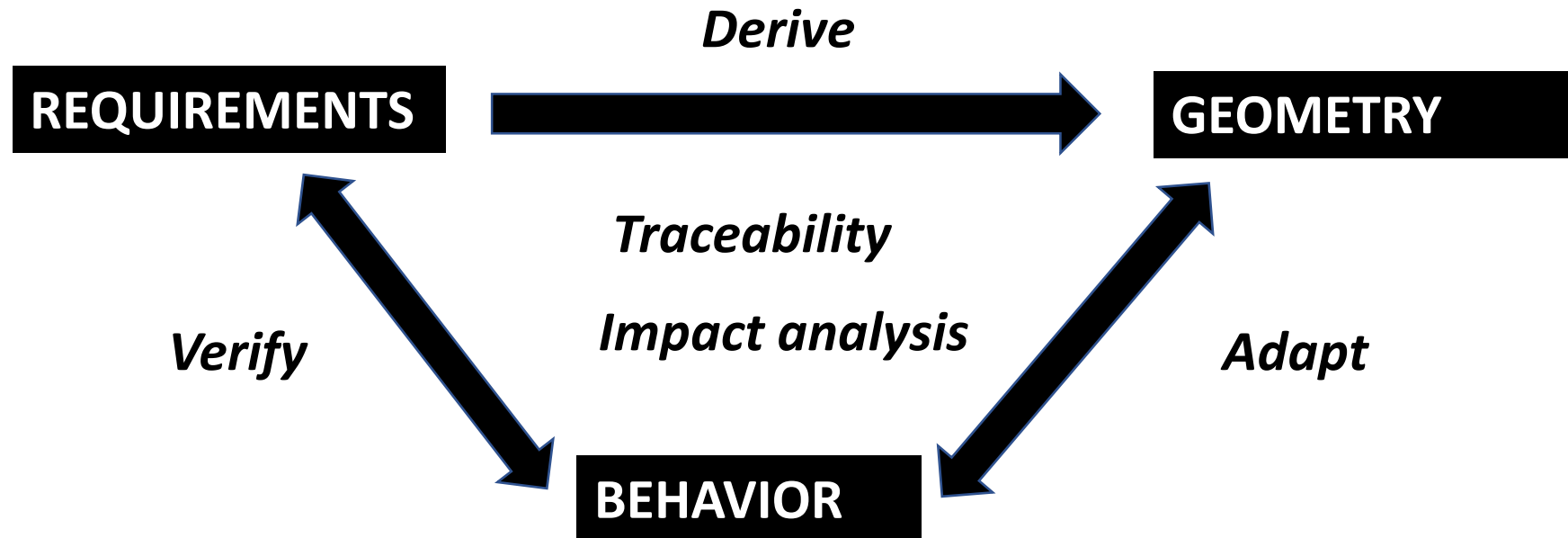
GEOMETRY



BEHAVIOR

Why do we simulate?

- A geometry is derived from requirements and verified by simulation behavior ...
- ...because it is cheaper than testing physical prototypes

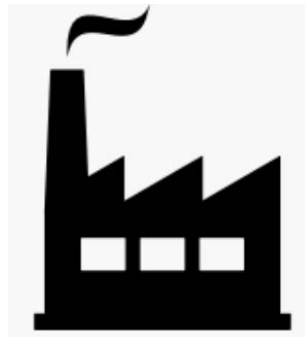


When do we simulate?

- Ideally, we simulate over the whole product lifecycle
- We simulate designs to verify requirements, manufacturing processes to evaluate impact of manufacturing on design, and compare sensor data for life prediction in service



DESIGN



MANUFACTURING



SERVICE

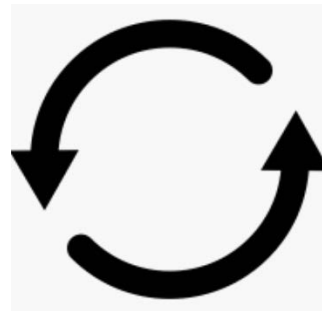
What and how do we simulate in design?

- What?: Behavior of product geometry
- How?: Three options:
 - Create geometry then test behavior
 - Iterative behavior testing of geometry (or simulation driven design)
 - Generate geometry from behavior

WATERFALL



ITERATION

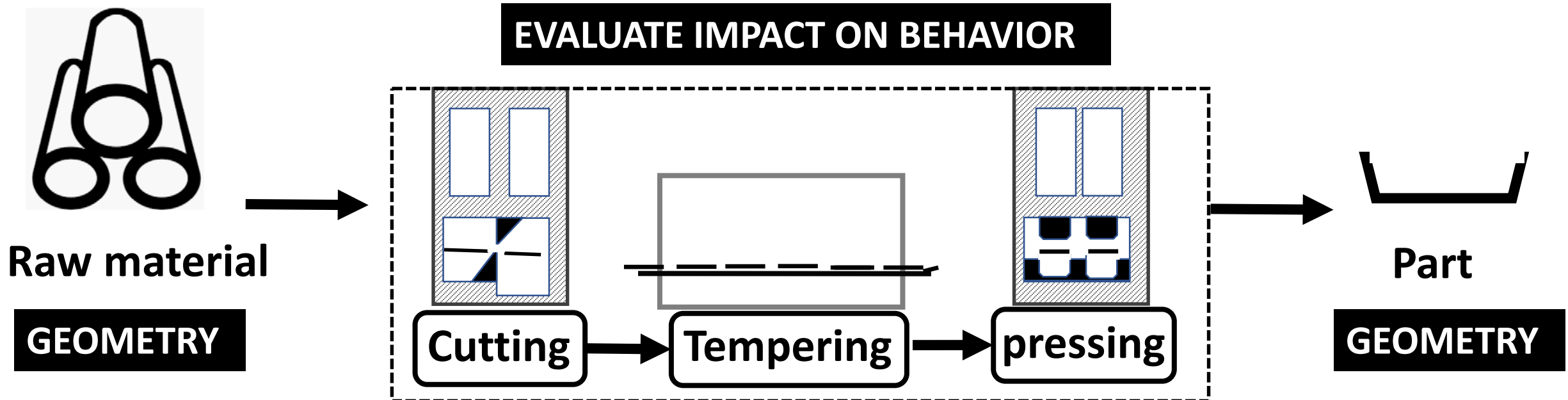


GENERATIVE



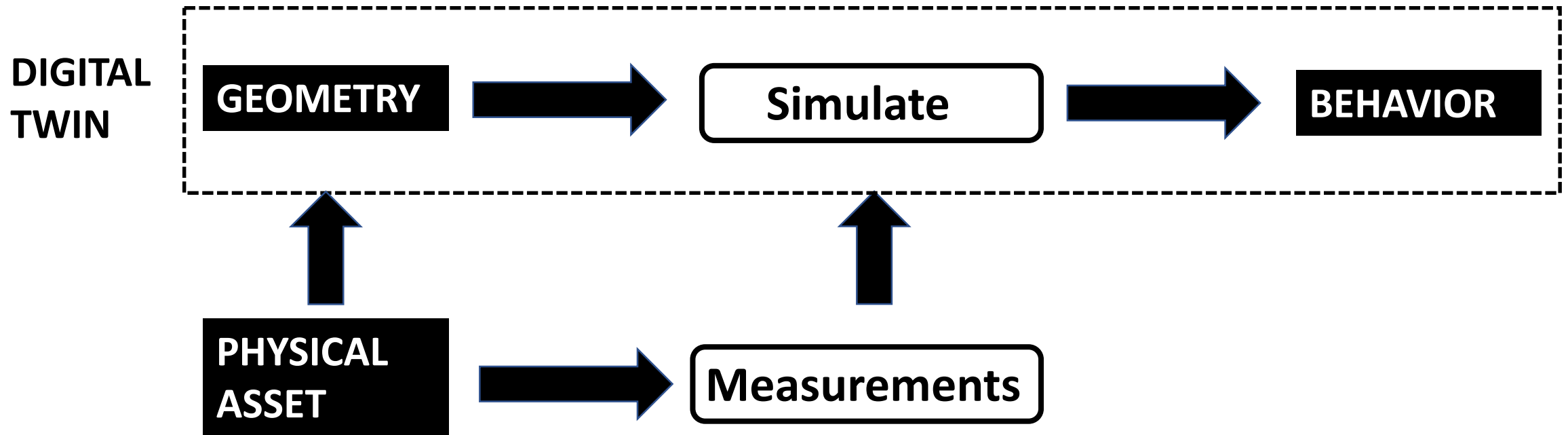
What and how do we simulate in manufacturing?

- What?: Change geometry of raw material to match product geometry
- How?: By simulating a chain of manufacturing processes that transforms the raw material geometry into the product geometry



What and how do we simulate products in service?

- What?: Behavior of 3D geometry of a physical asset in real-time
- How?: Input sensor data to simulate behavior for life-time prediction



How do we simulate?

- We use a tessellated (linear) representation of the geometry (a mesh); choose a predefined physics model, assign boundary conditions, input simulation parameters to the simulation template, then simulate.

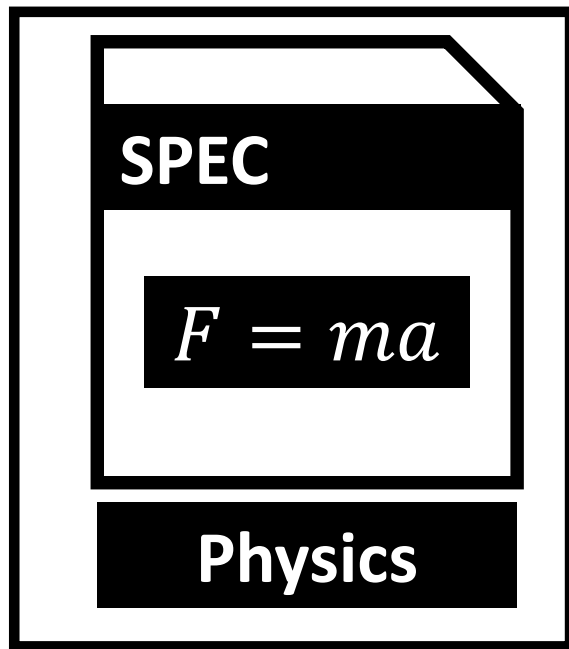


Overview

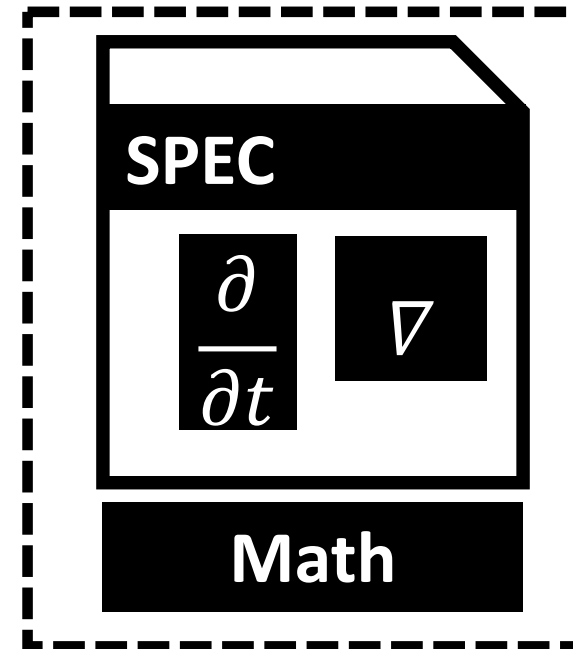
- Simulation in engineering
- **Tools and approaches**
- PDEs as model definition
- Machine readable physics
 - Motivation and approach
 - From equations to computation
 - From physics to solvers
- Benefits and usage
- Summary

What is the tool ecosystem to simulate?

- Physics-dependent commercial off the shelf solutions or physics-independent solutions



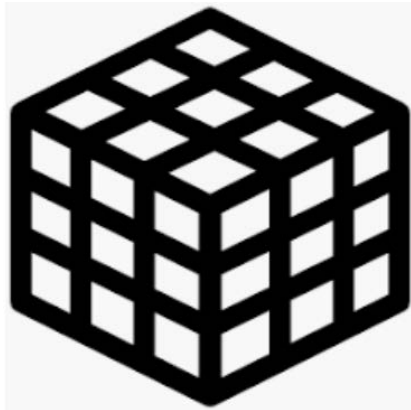
**PHYSICS-DEPENDENT
COMMERICAL OTS**



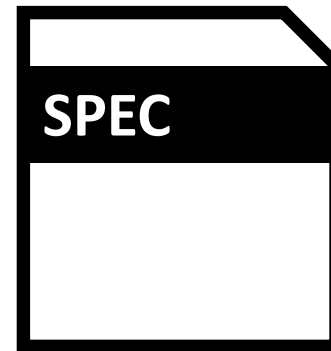
**PHYSICS-INDEPENDENT
OPEN SOFTWARE**

What are the simulation digital assets?

- Product model geometry (mesh)
- behavior specification which consists of simulation specification (model + boundary conditions + parameter settings) and simulation results



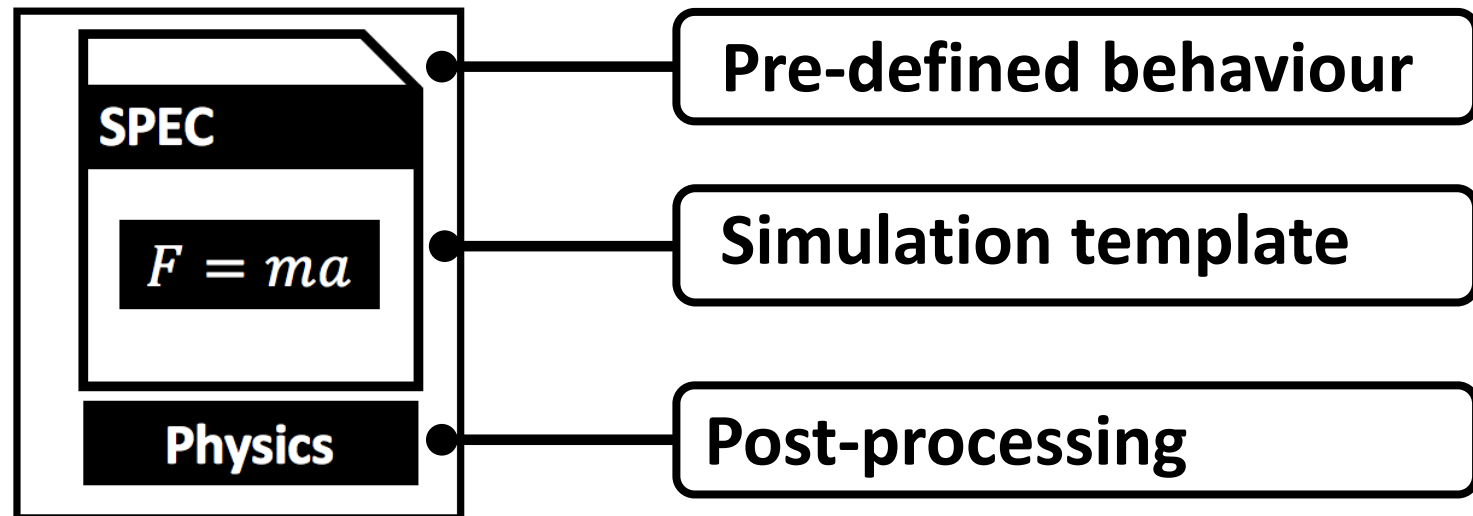
GEOMETRY



BEHAVIOR

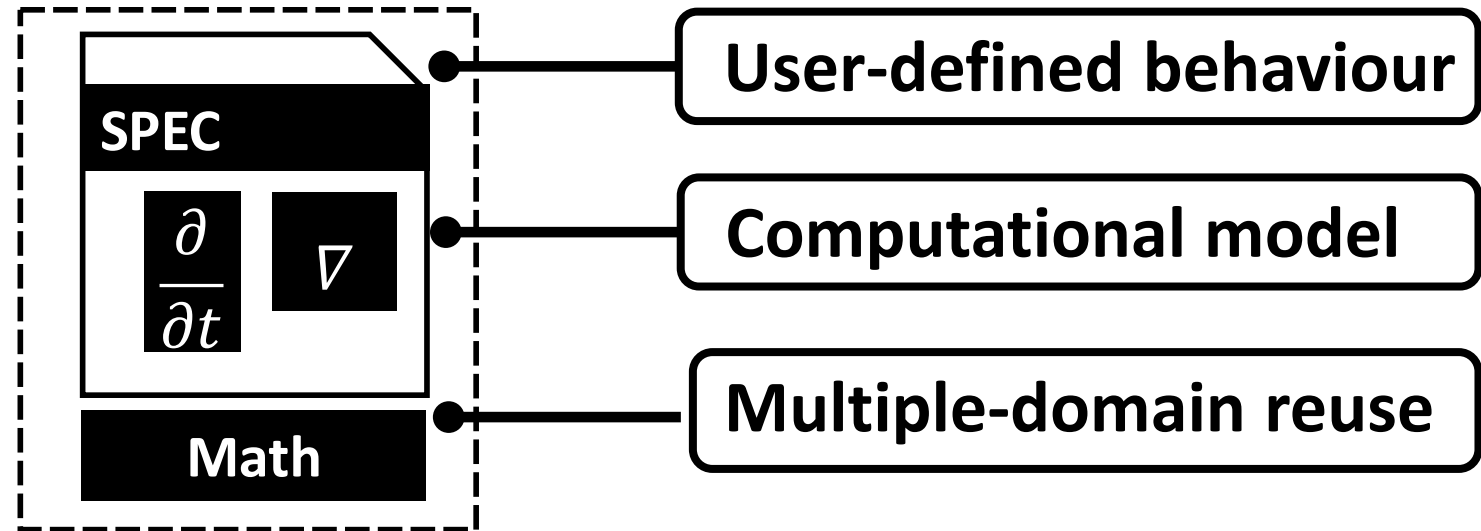
Benefits of physics-dependent software

- Declarative procedure that selects pre-defined behavior models
- Convenient usage with simulation input templates
- Good integration with product model geometry
- Post-processing support “on click”



Benefits of physics-independent software

- Flexibility (user-defined behavior)
- Transparency (computational model)
- Reusability of behavior models for multiple physical domains



Problems of physics-dependent software

- Pre-defined behavior is documented with more or less details (consistency and transparency problem)
- Proprietary reference codes are used to identify pre-defined behavior
- Simulation capabilities limited to pre-defined behaviors available in software library
- Difficult to find the correct behavior due to vast number of behavior variants
- Difficult to compare behaviors between different software



Requires more transparency and flexibility

Any standard for physics-dependent software?

- AP 209: Standard for multidisciplinary analysis and design
- Currently only support structural mechanical engineering and fluid dynamics concepts
- Integrates pre-defined behavior models using proprietary reference code
- Who is in charge of behavior model reconciliation?
- Difficult because COTS are parametrized pre-defined models with varying documentation, rather than computational models



Vendor lock-in

Problems of physics-independent software

- Difficult to use for engineers (e.g. PDEs or PDEs weak forms)
- Not simple to connect math to physics information
- Limited integration with product model geometry
- Requires writing of custom code to post-process results (e.g. secondary variable such as stress)
- No standard for computational model



Requires better usability



What should be the behavior input?

Summary and benefits of tools/ approaches

Physics-dependent tools (declarative input)

BENEFITS:

Usability (Pre-defined behavior, input using simulation templates post-processing)

LIMITATIONS:

Flexibility, transparency (documentation, behavior equivalence, vendor lock-in)

Physics-independent tools (computational input)

BENEFITS:

Flexibility (User-defined behavior)
transparency (numerical choices)

LIMITATIONS:

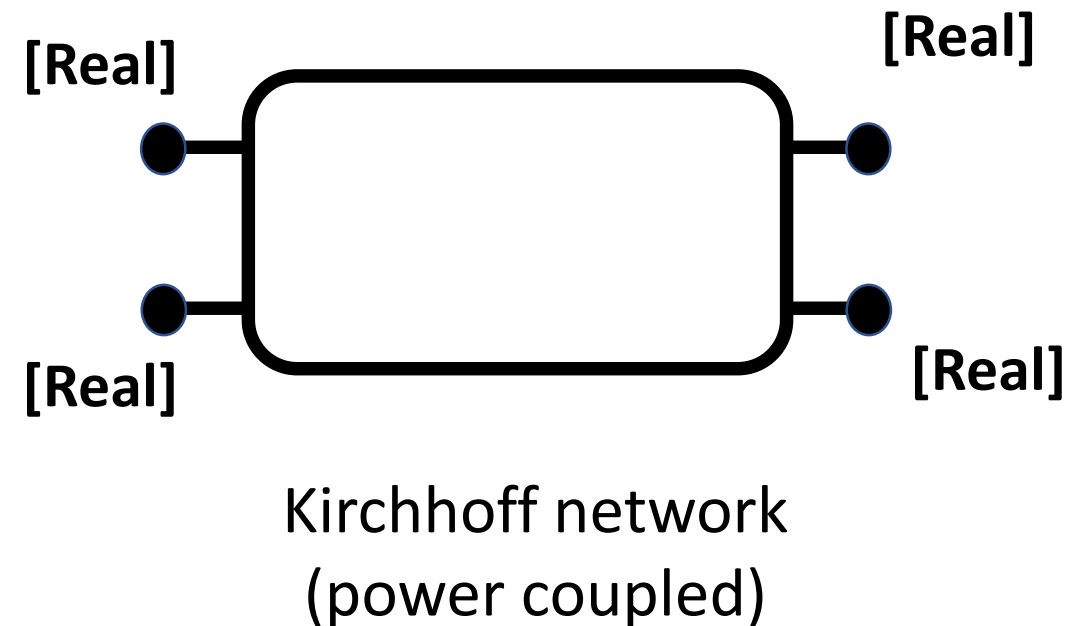
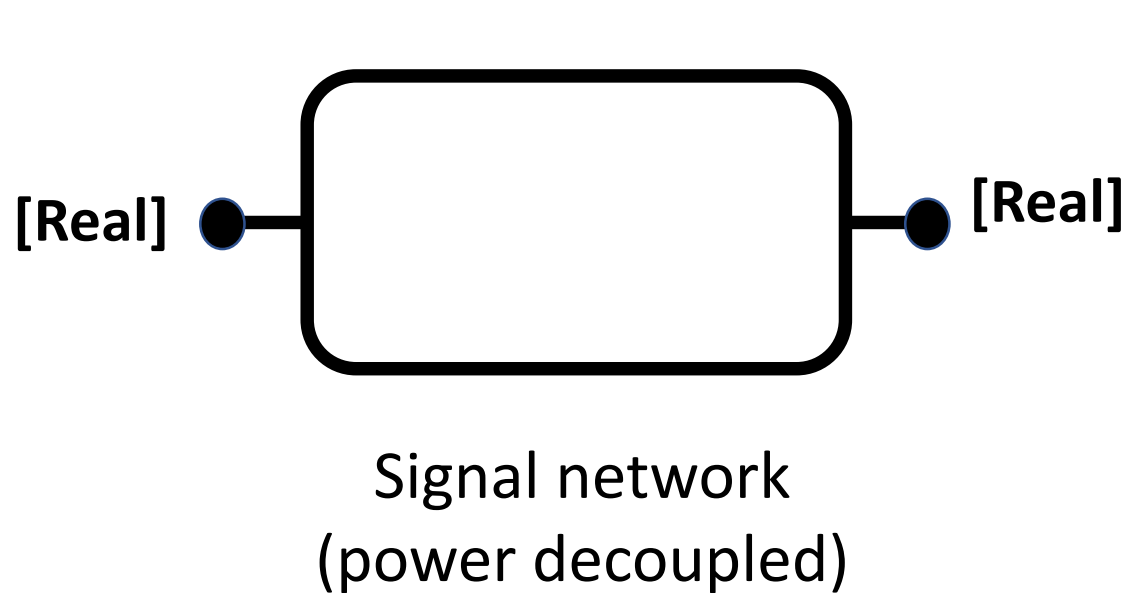
Usability (behavior input is complex) **association to physics** (abstract behavior)

Overview

- Simulation in engineering
- Tools and approaches
- **PDEs as model definition**
- Machine readable physics
 - Motivation and approach
 - From equations to computation
 - From physics to solvers
- Benefits and usage
- Summary

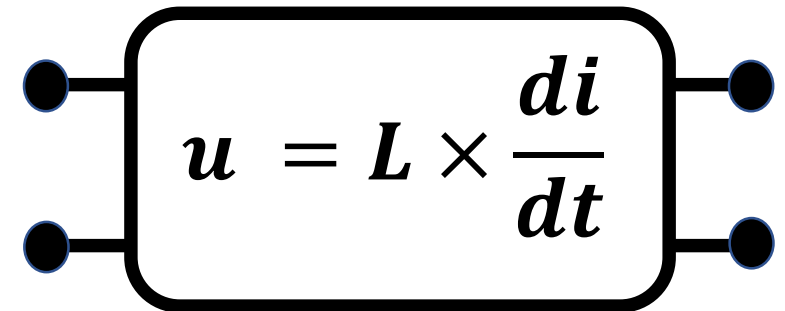
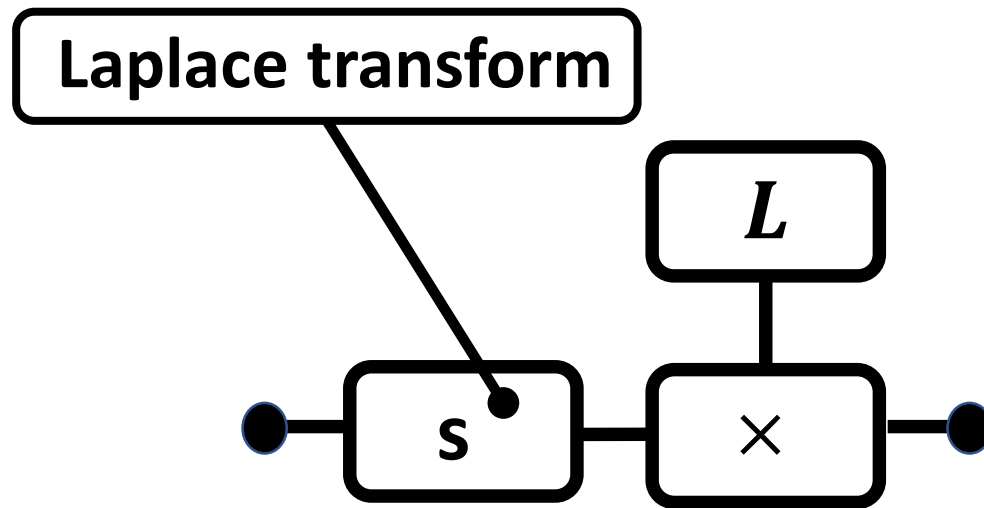
What can we learn from time-only simulation?

- Successful time-only simulation tools and standards are available (e.g. Mathworks Simulink/Simscape, Modelica) that process real numbers.



What do they have in common?

- They support graphical and textual models for networks of **mathematical equations**



What is the benefit of using mathematical equation as input?

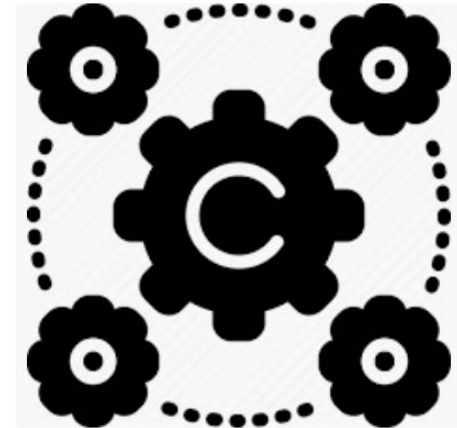
- This translates into simulation credibility, reproducibility and even replicability as well as interoperability (e.g. FMI)



Credibility



**Reproducibility
and replicability**



Interoperability

What about using PDEs for model description?

- PDEs can be classified (e.g. 2nd order PDEs in elliptic, parabolic, hyperbolic)
- PDEs can be associated to physics problems (e.g. heat conduction is a parabolic PDE)

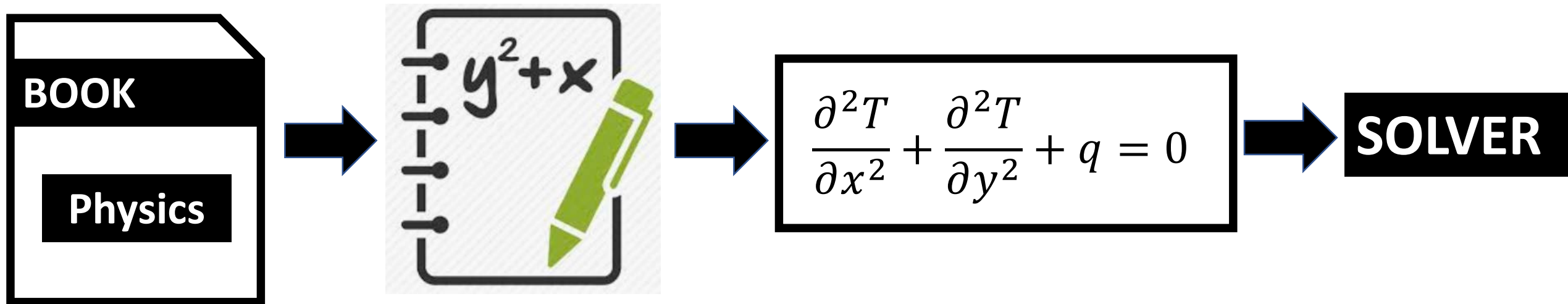
What are the problems with PDE descriptions?

- Boundary conditions refer to physical variables that are not in the PDE
- Secondary physical variables are not part of the equation (e.g. heat flux density for thermal conduction)
- Classifying by PDEs makes sense from the solver side but not from the physics side (actually thermal conduction can be hyperbolic, elliptic or parabolic)
- How would we associate systems of PDEs?

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + q = 0$$

How do we derive PDEs?

- Find and select physical laws
- Define model assumptions and perform variable substitution to derive equations (e.g. connecting configuration variable with loads)



Overview

- Simulation in engineering
- Tools and approaches
- PDEs as model definition
- **Machine readable physics**
 - Motivation and approach
 - From equations to computation
 - From physics to solvers
- Benefits and usage
- Summary

Connectivity is efficient knowledge

In a world of time-table

In a world of maps

In a world of computers



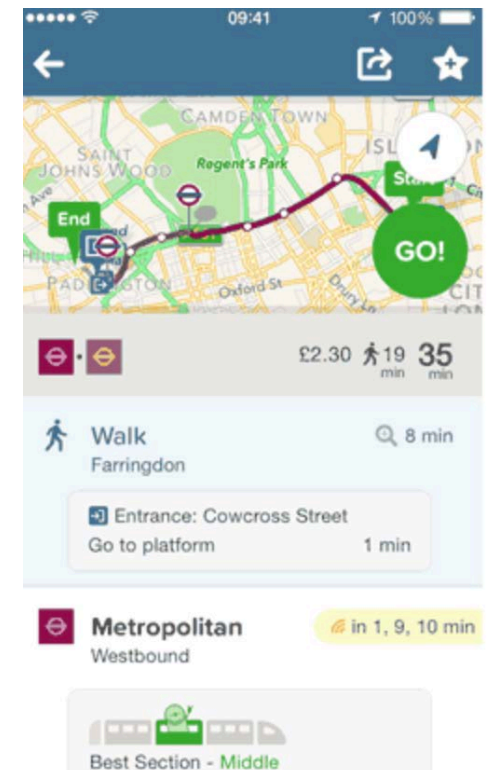
Northern
line



Central
line



Cross-referencing



What about physics?

In a world of physics books



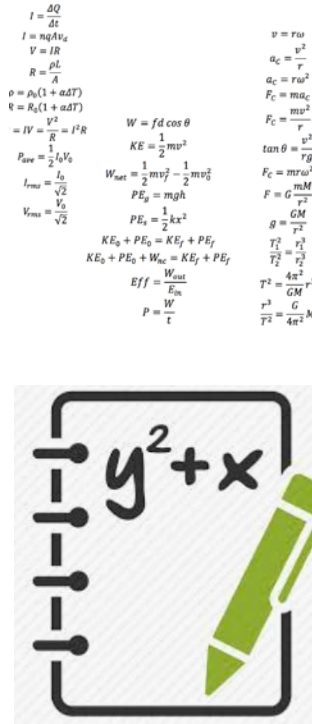
Mechanical



Electrical



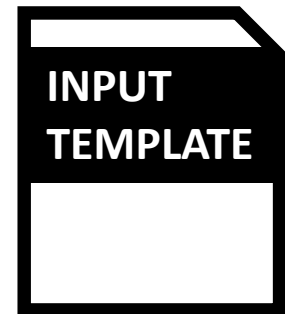
Cross-referencing



?



In a world of
computer simulations

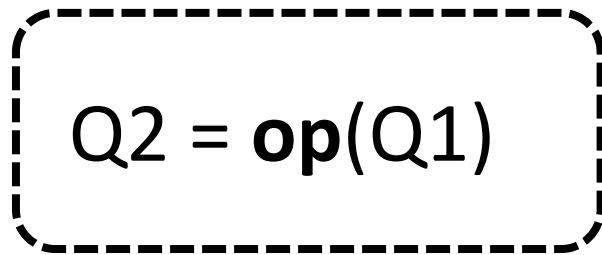


SOLVER

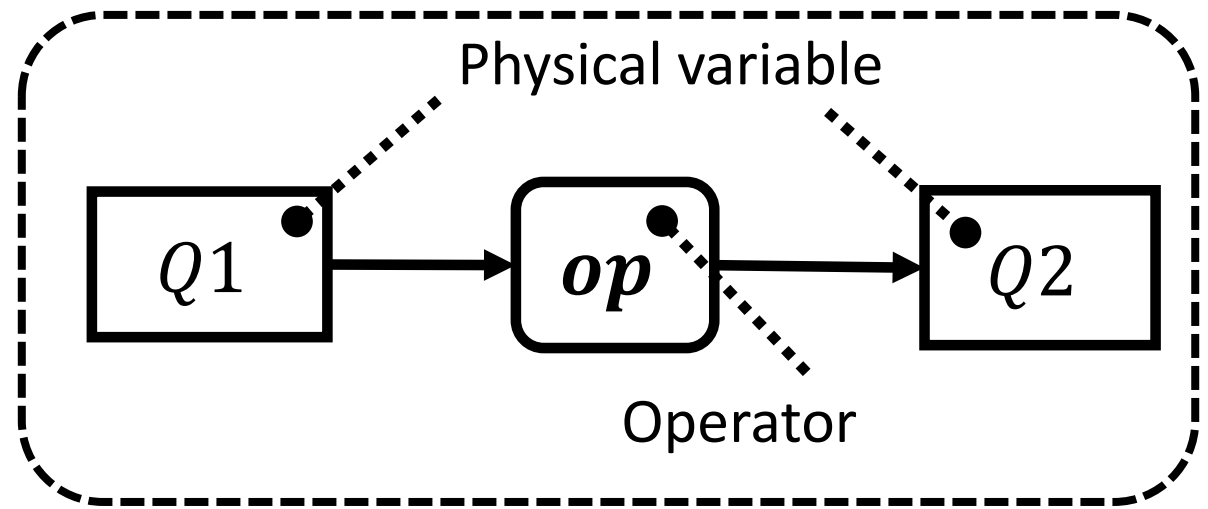
Pillar for machine-readable physics: computation instead of equations

- Our proposition is to store physical laws as computations (*physics graphs*) instead of equations

Equation

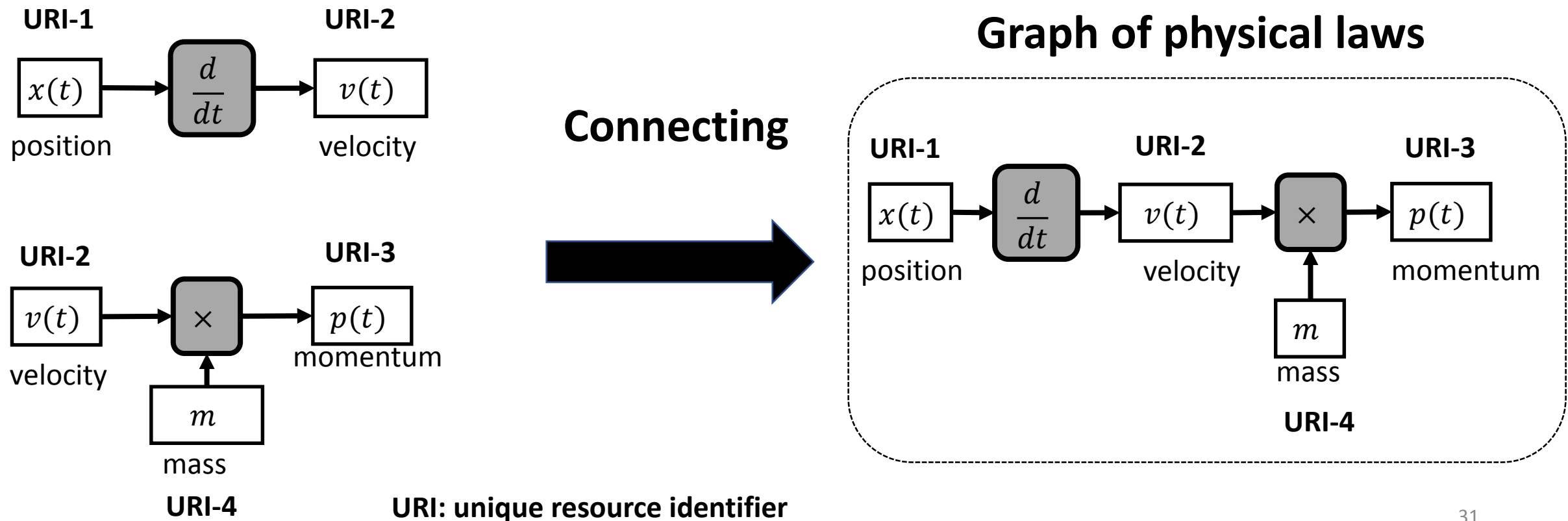


Computation (Physics graphs)



What about connectivity?

- With computations, physical laws can be connected together and relations between them interpreted explicitly (e.g. RDF)



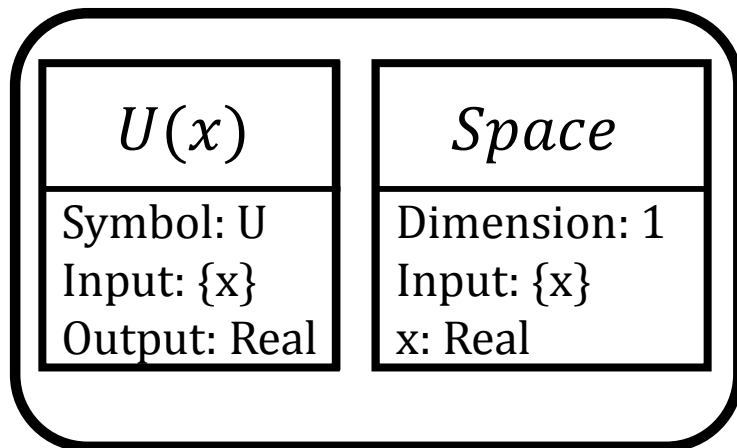
Overview

- Simulation in engineering
- Tools and approaches
- PDEs as model definition
- Machine readable physics
 - Motivation and approach
 - **From equations to computation**
 - From physics to solvers
- Benefits and usage
- Summary

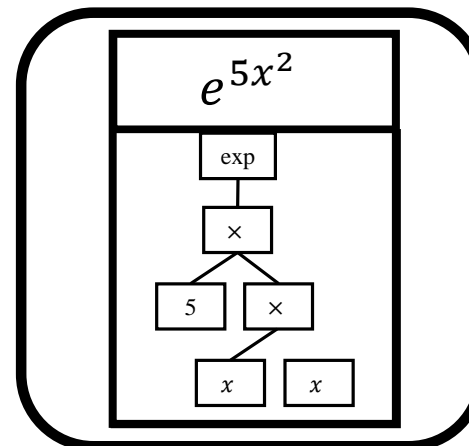
What are we computing?

- We are computing *math objects*, rather than numbers.
- A math object is defined by three different levels of abstraction.
- Starting from the most abstract, we have *function type* then *mathematical expression* and then *data or function plot*.

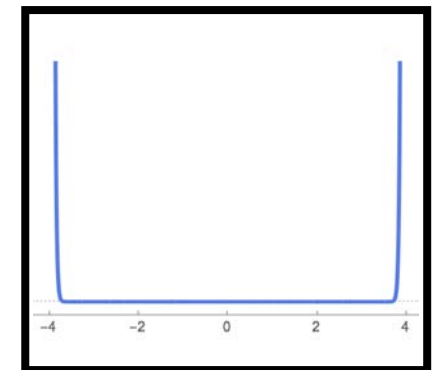
Type



expression

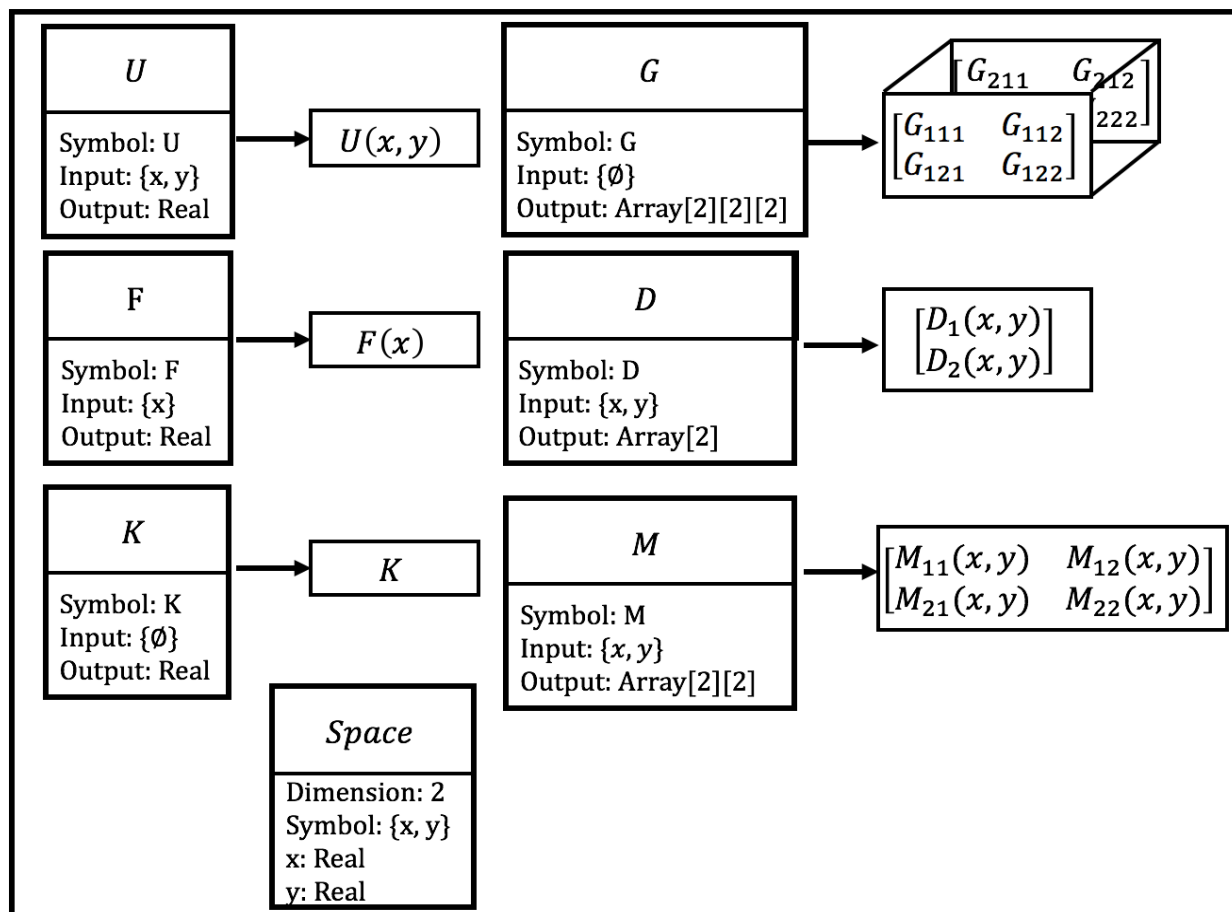


data



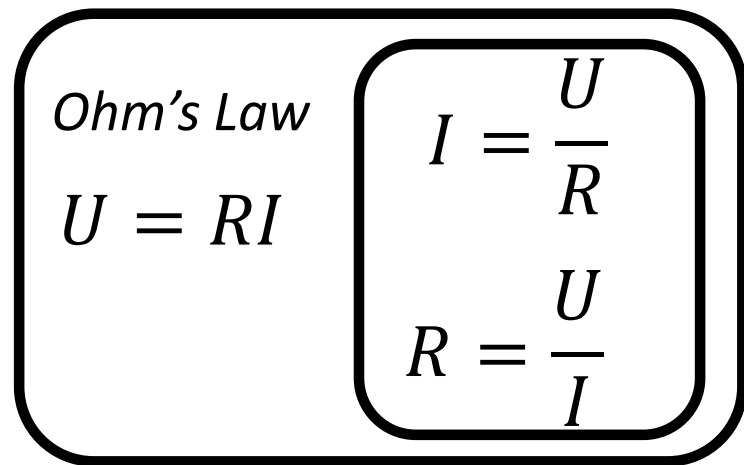
More complex math objects for physics

- Math objects are tensor fields in physics

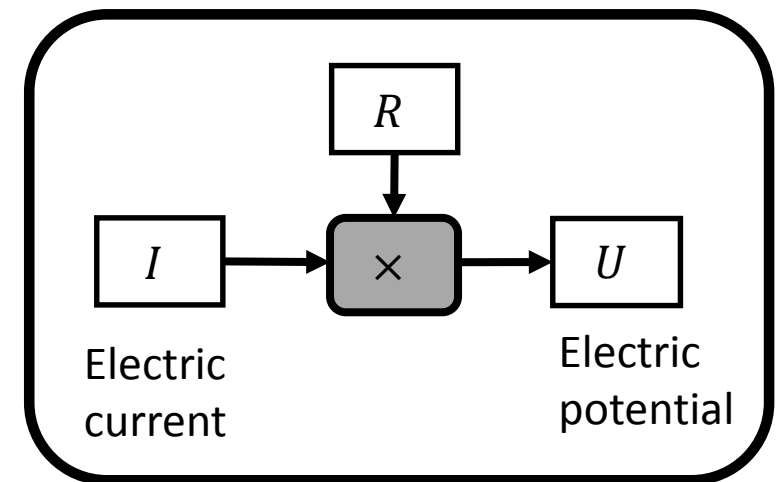


Equations and computations compared

- With equations, many possible rearrangements are possible and implicit
- With computations, the relation is directed and explicit (e.g. metro map)



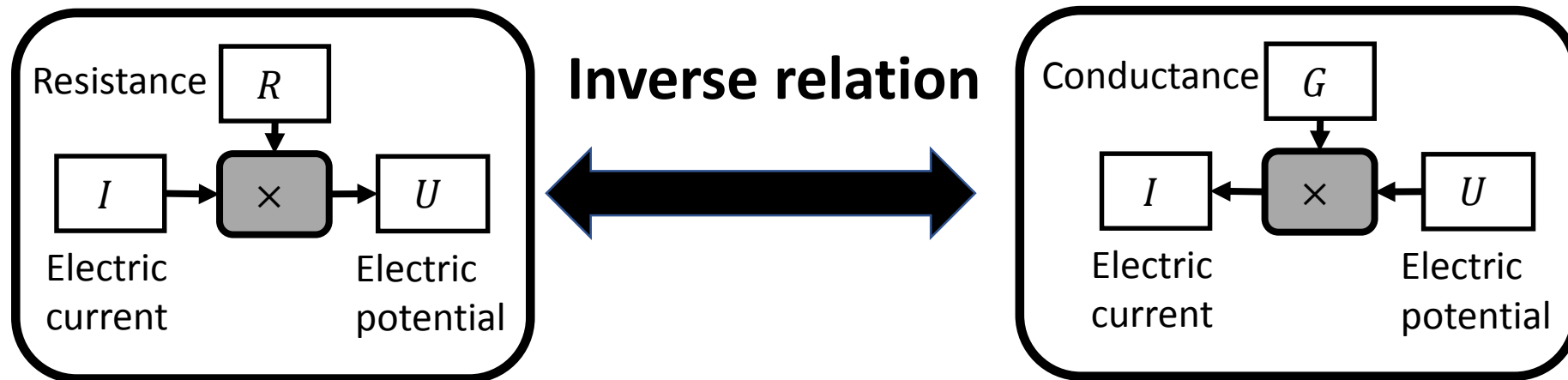
Equation



Computation

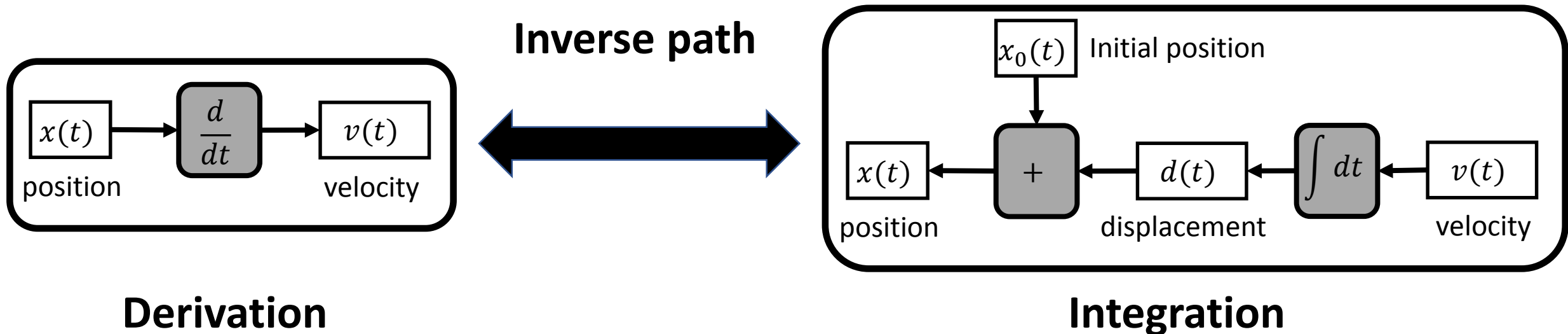
What about inverses for computation?

- Linear physical laws have inverse relations (algebraic relations)



What about inverses for computation?

- Physical laws involving derivatives with a single independent variable have no inverse relation but an inverse path
- Integration introduces a new physical variable and a boundary condition



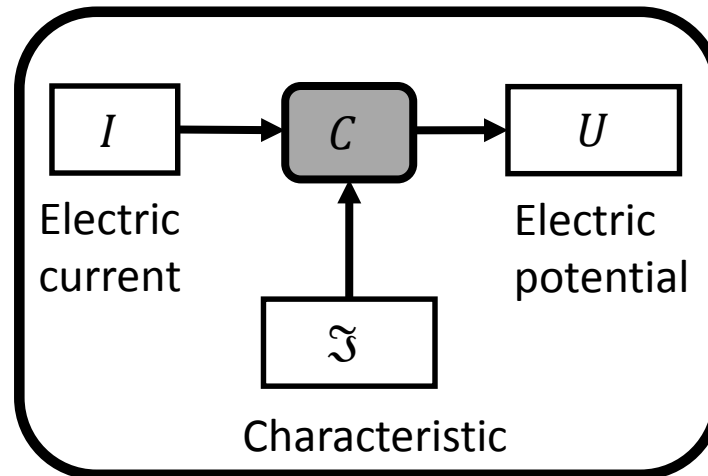
What about inverses for computation?

- Some physical laws have no inverse relation or path:
 - Non-linear compositions or differential derivatives
 - Higher order derivatives, introduce boundary conditions required for solvers.

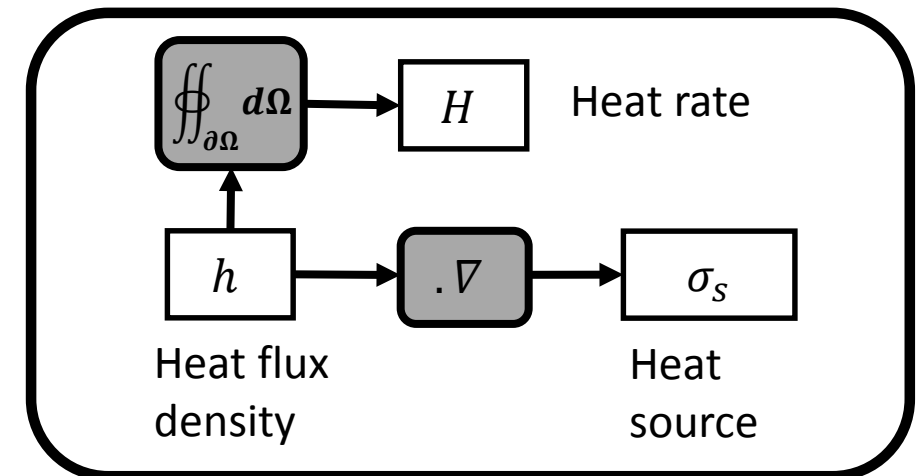
No inverse path

No inverse relation

Nonlinear law



Differential law

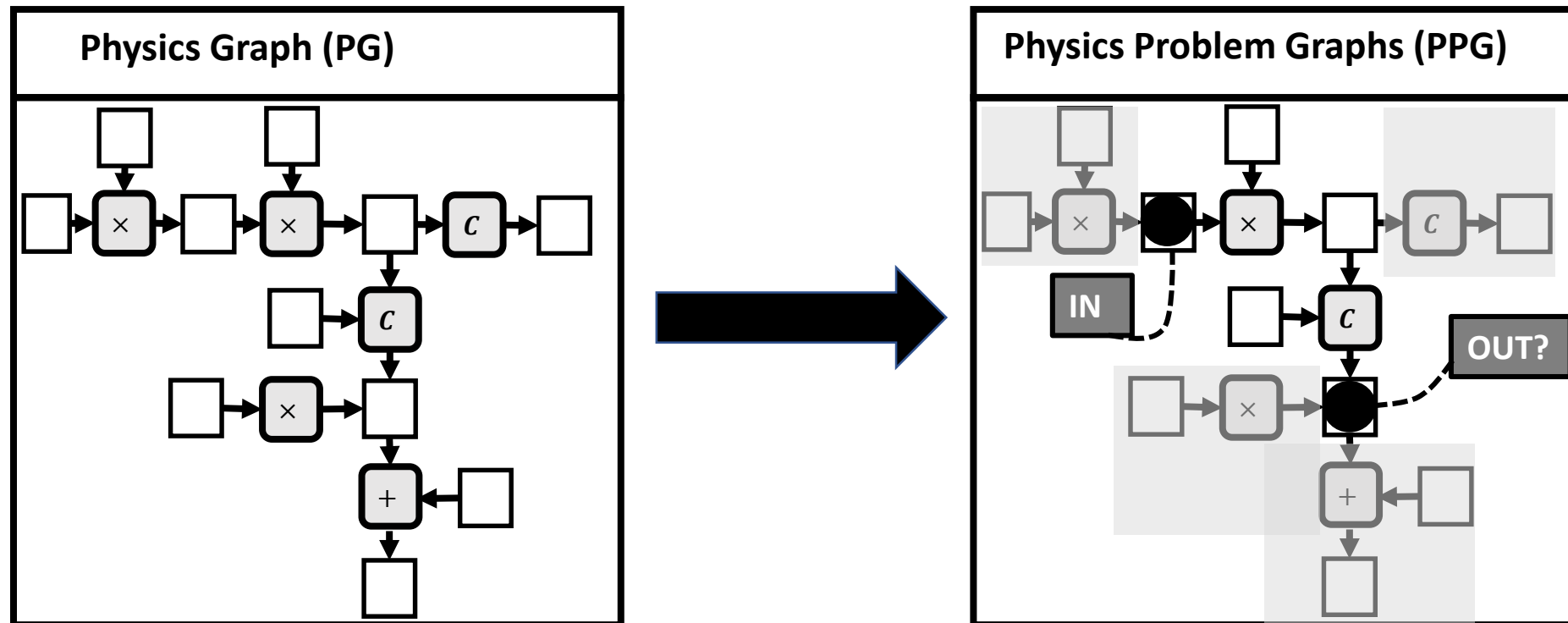


Overview

- Simulation in engineering
- Tools and approaches
- PDEs as model definition
- Machine readable physics
 - Motivation and approach
 - From equations to computation
 - From physics to solvers
- Benefits and usage
- Summary

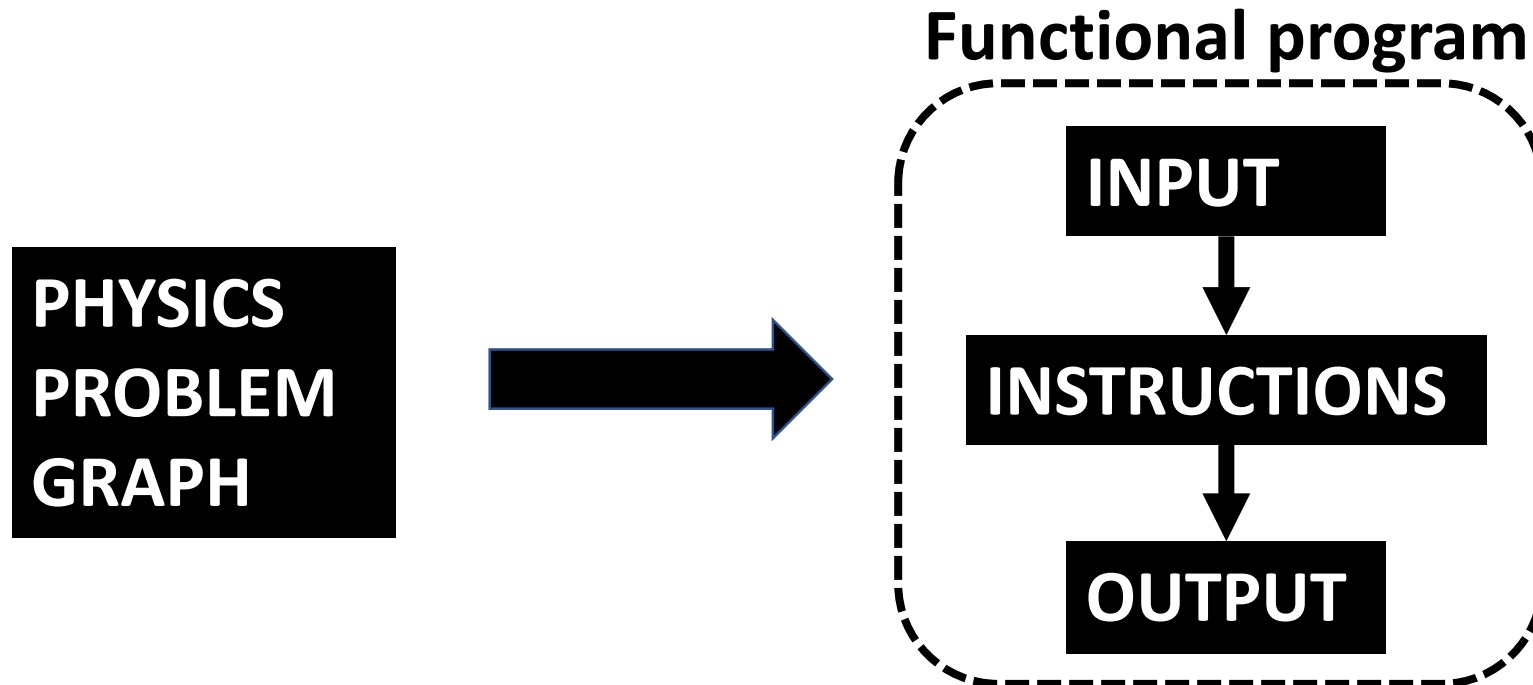
How do we define physics problems?

- They are defined by known(s) and unknown(s) pairs of physical variables.
- Paths through physics graphs between known and unknown variables can be extracted into *physics problem graphs*



How do we compute outputs from inputs?

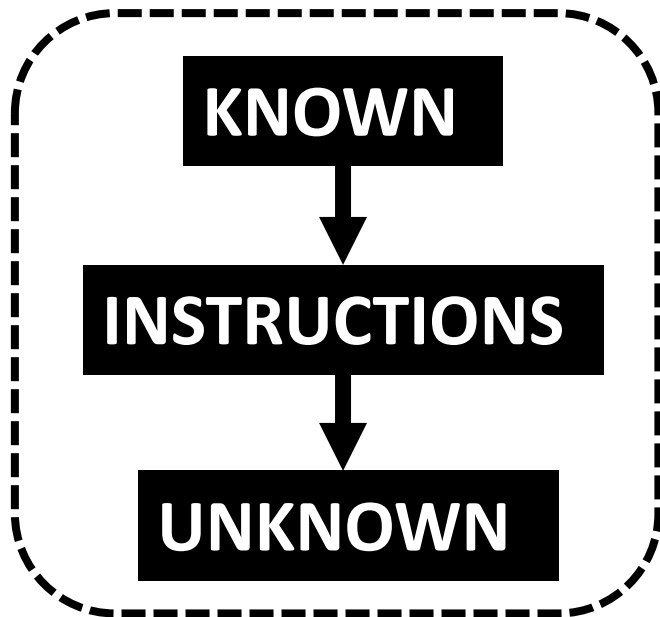
- Generate *functional programs* by collecting the operations along a path through a physics graph, providing the mathematical expressions for calculating outputs from inputs



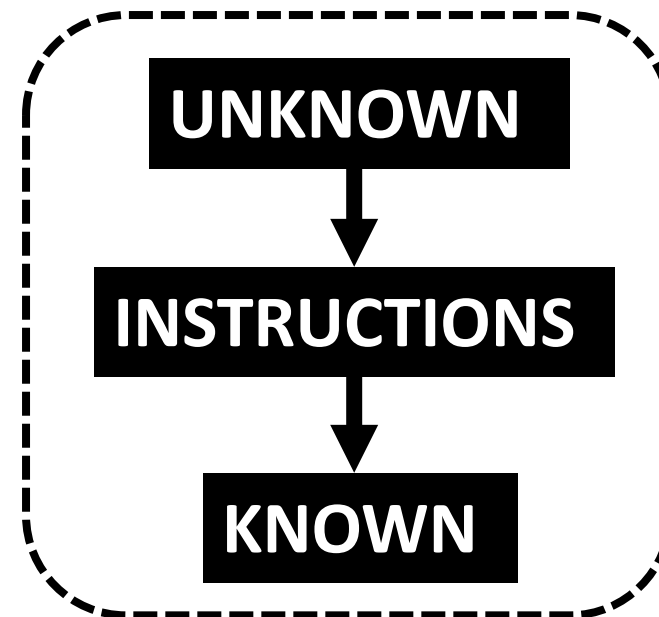
Which kind of functional programs do we have?

- When the input is known, we have an *evaluation program*
- When the input is unknown, we have a *solver program*.

Evaluation program



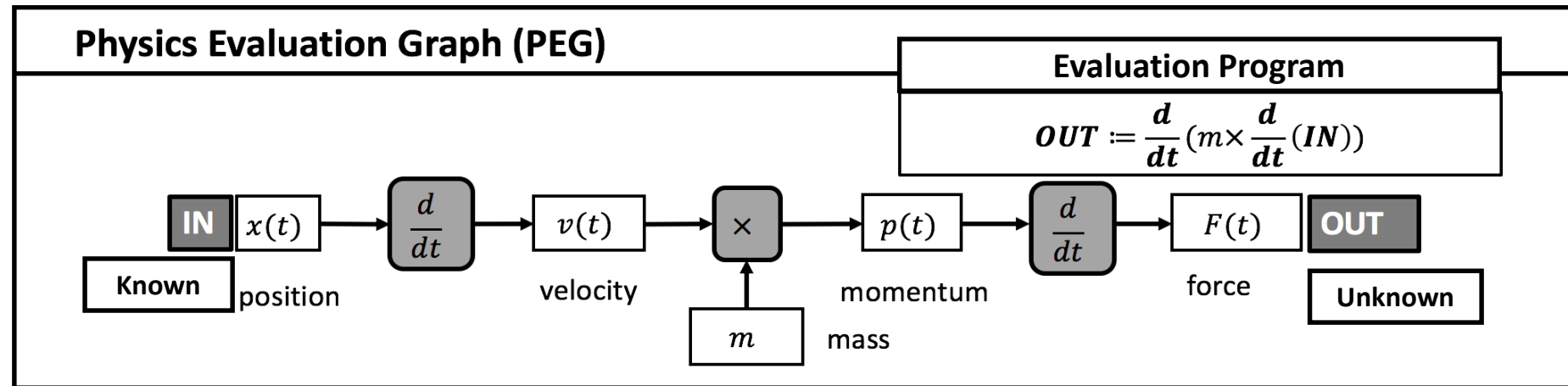
Solver program



Examples of physics evaluation problems

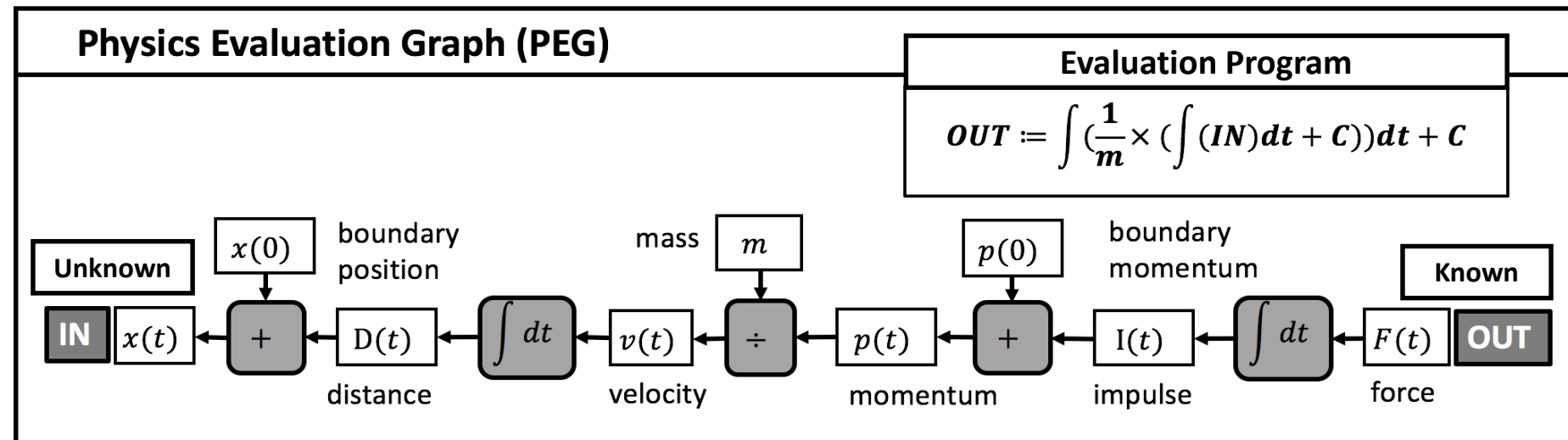
Example 1:

Known is position
Unknown is force

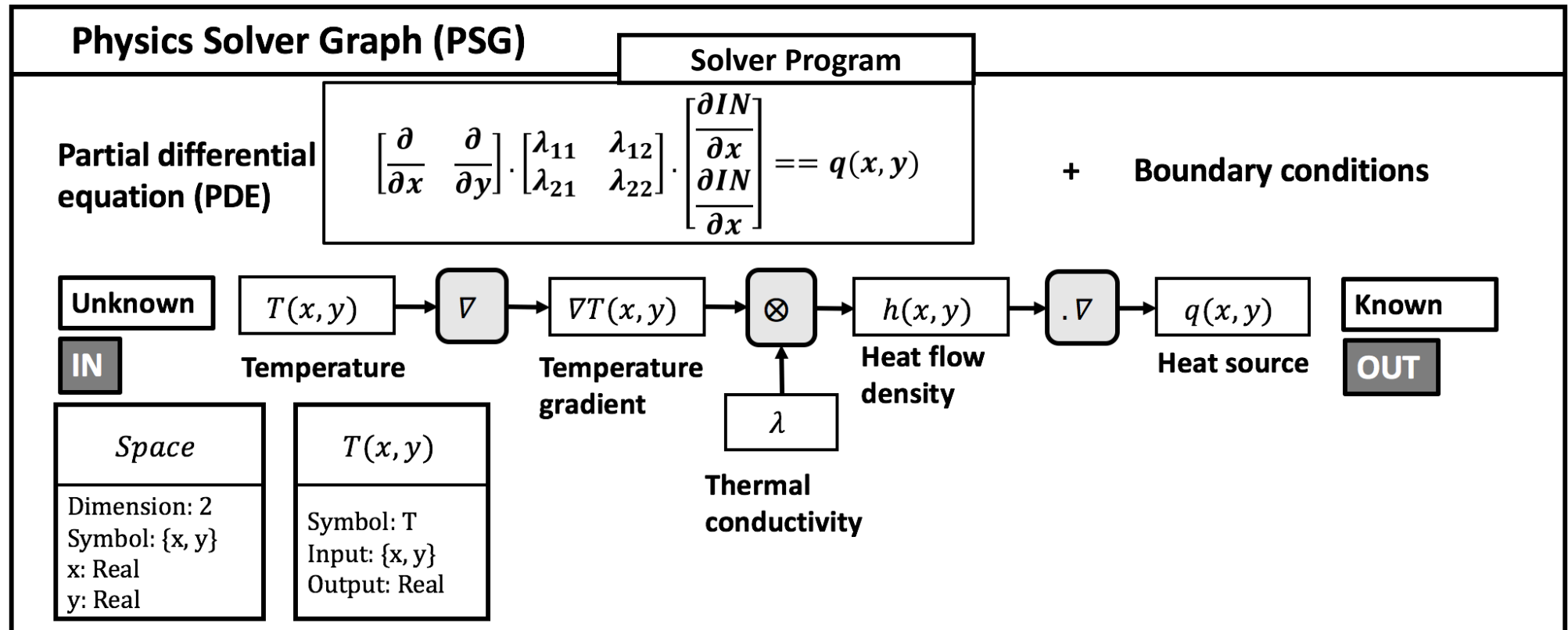


Example 2:

Known is force
Unknown is position



Example of physics solver graph



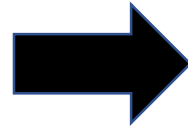
Example 3:

Unknown is temperature

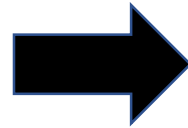
Known is heat source

Numerical graphs augment physical graphs for finite elements

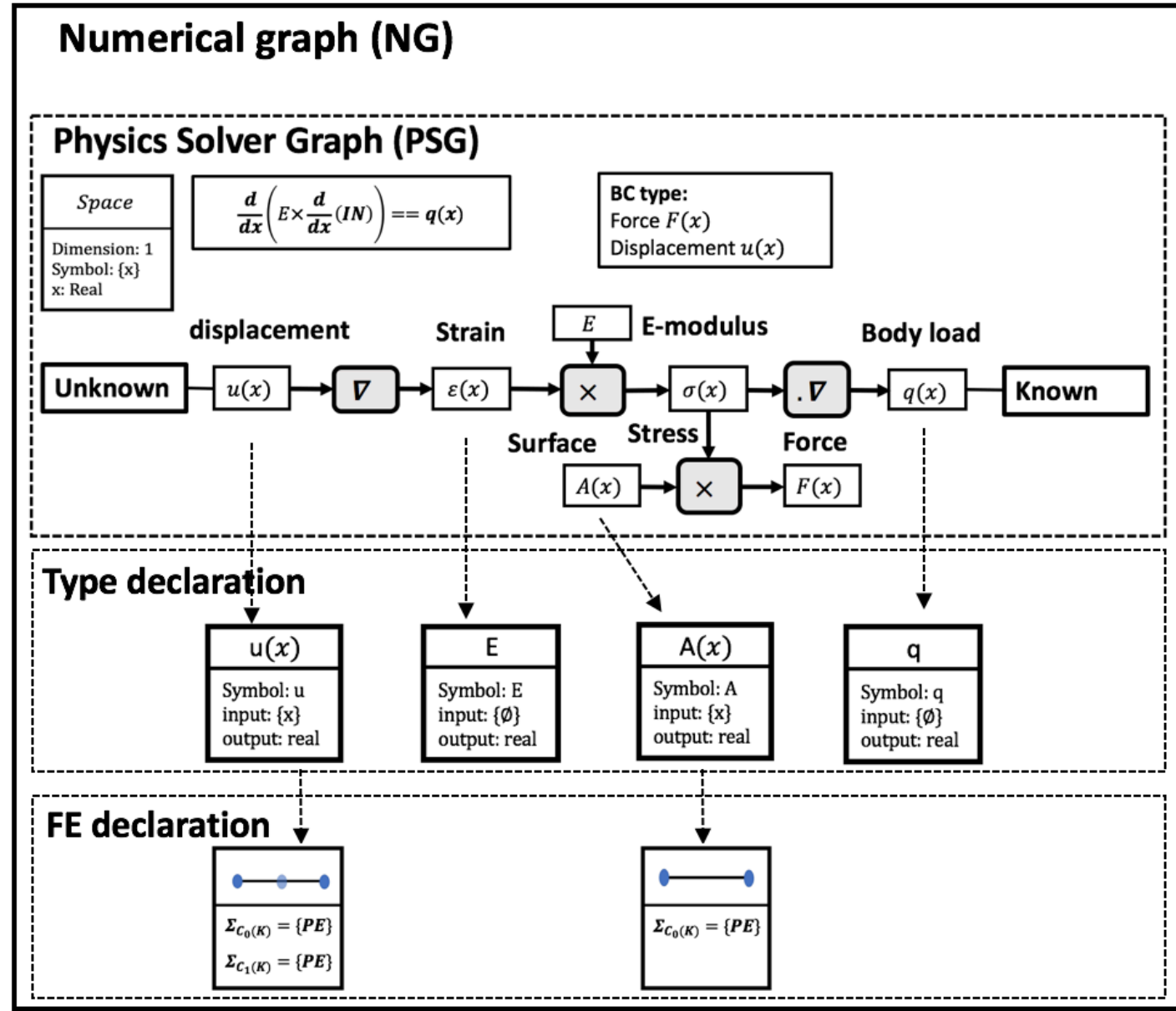
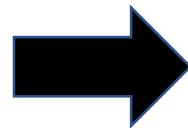
Physics problem definition



Function type declaration

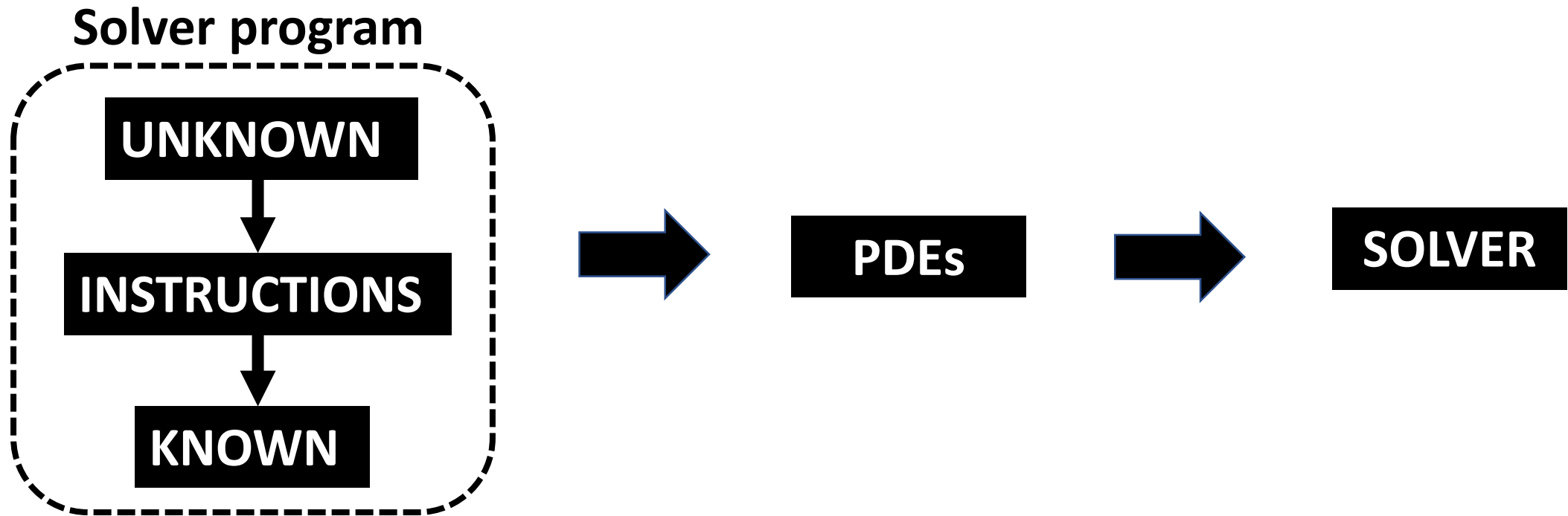


Finite element declaration



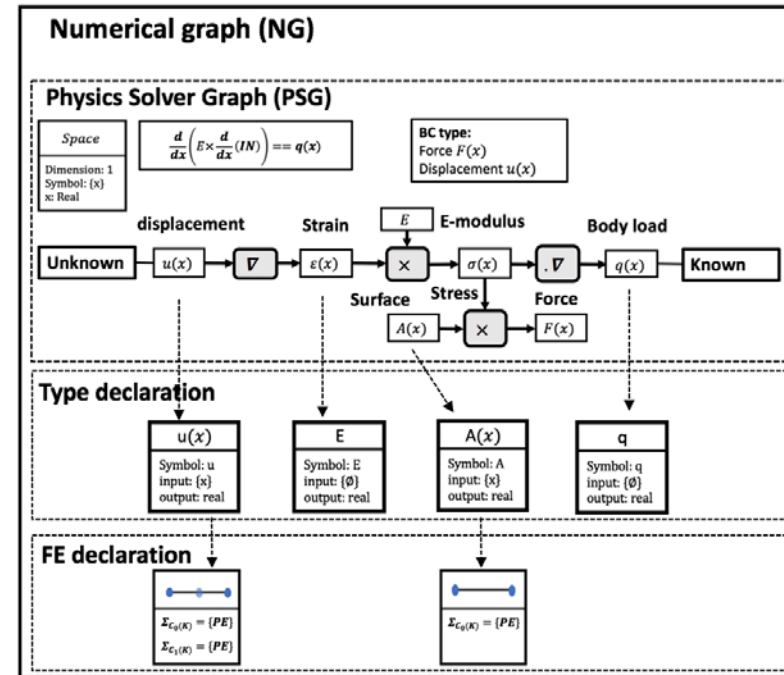
Associate physics solver graphs to solvers via functional programs

- Solver programs define PDEs that can be input to solvers.

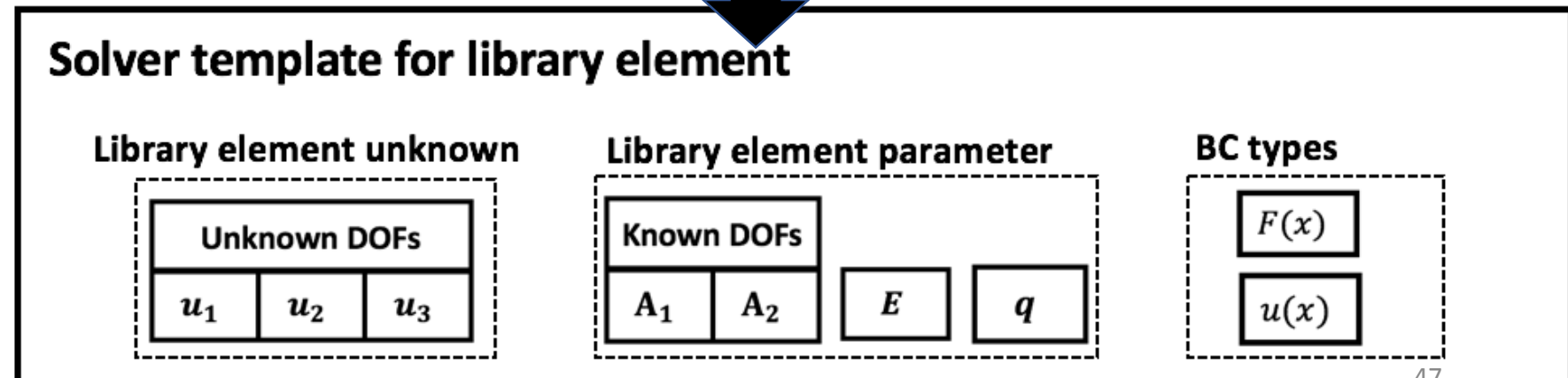


Generating solver input from numerical graphs

Numerical graph



Solver input template



Overview

- Simulation in engineering
- Tools and approaches
- PDEs as model definition
- Machine readable physics
 - Motivation and approach
 - From equations to computation
 - From physics to solvers
- **Benefits and usage**
- Summary

What problems does machine-readable physics solve?

- It reduces the limitations of physics-dependent and physics-independent software by combining the benefits of both.

New approach (computational model of physics)

BENEFITS:

Usability (simulation template generation, post-processing), **flexibility** (user-defined behavior), **transparency** (computational model, documentation generation)

LIMITATIONS:

limited integration with product model **geometry**

Who is going to use it?

- Numerical code developers
 - Use numerical graphs (NGs) as software specifications to create new solvers
- Vendors or open source communities use
 - Use NGs to associate domain independent models to their existing solvers
- FEA engineers use
 - Use NGs to design pre-defined behaviors
- Standardization bodies use
 - Use NGs to define libraries
- Design engineers
 - Use simulation templates generated from NGs as input, and physics solver graphs (PSG) as model documentation
- System engineers
 - Connect requirements to physical variables in PSGs (e.g. maximum stress)

Overview

- Simulation in engineering
- Tools and approaches
- PDEs as model definition
- Machine readable physics
 - Motivation and approach
 - From equations to computation
 - From physics to solvers
- Benefits and usage
- **Summary**

Summary

- Capture physical laws as computations (physics graphs)
 - Function specification of physical variables
 - Can be stored in a database
- Define physics problems by
 - Defining pairs of known/unknown physical variables
 - Extracting physics problems as subgraphs
- Generate functional programs from physics problems (evaluation & solver)
- Transparent definition of numerical choices (e.g., finite elements)
- Provides all information needed to implement solvers
- Enables multiple solvers to be used for same functional program