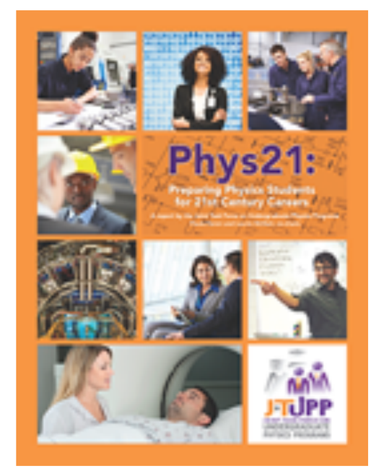
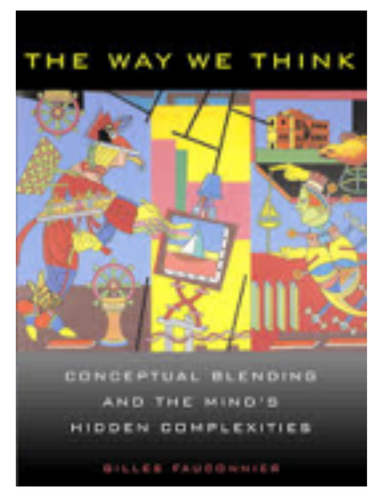
Conceptual Blending in Computational Modeling

Brandon Lunk



Computational modeling is an increasingly important skill to include in the physics classroom[1].



Conceptual Blending[2] is the process of combining multiple ideas in order to form new insights that can be understood within the human experience.

When writing a computational model, one must blend elements of physical principles, mathematics[3], and programming logic (and expectations of the output). Novel structures emerge from this blend.

```
p = p + Fnet * deltat
```

save

reassign

Failure modes[4] can occur when students read/write code without recognizing that:

1. An aspect of the code supports different interpretations in different input spaces
2. The code supports different disciplinary conventions

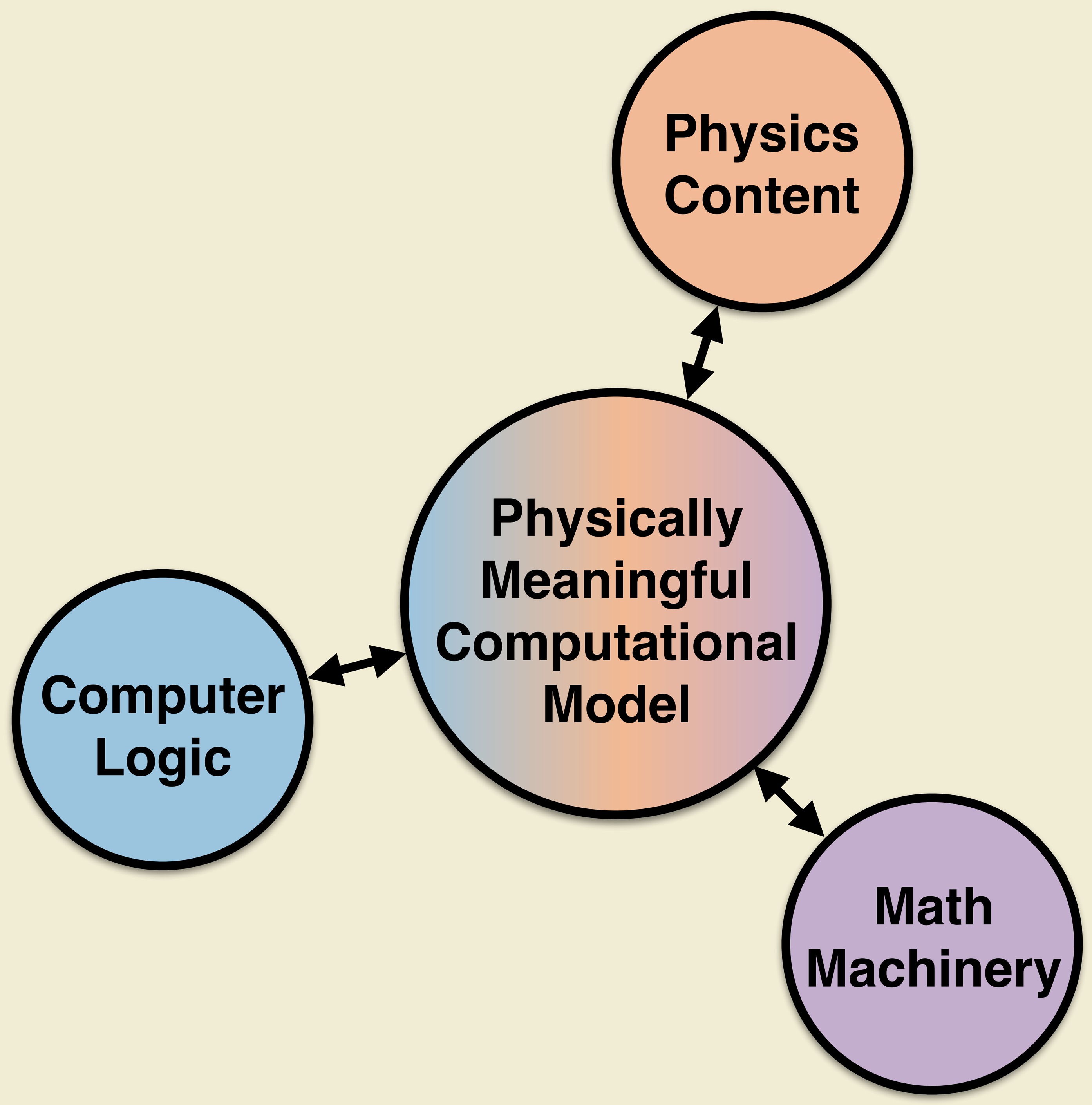
Instructional interventions should build upon familiar experiences, e.g.:

1. Saving documents
2. Spreadsheet iteration

References:

1. JTUPP (2016) Phys21: Preparing physics students for 21st century careers
2. G. Fauconnier & M. Turner (2002) *The Way We Think: Conceptual Blending and the Mind's Hidden Complexities*, Perseus book group, NY
3. T. Bing & E.F. Redish (2006) "The cognitive blending of mathematics and physics knowledge," *PERC proceedings*, Syracuse, NY.
4. L. Gire & E. Price (2014) "Arrows as anchors: An analysis of the material features of electric field vector arrows," *Phys. Rev. PER* **10**, 020112
5. M. Caballero, M. Kohlmyer, & M. Schatz (2012) "Implementing and Assessing Computational Modeling in Introductory Mechanics," *Phys. Rev. ST: PER* **8**, 2
6. B. Lunk (2012) *A Framework for understanding students' computational modeling practices*, NCSU Diss.

Conceptual Blending can help us model student understanding of computational physics and diagnose failure modes.



Documented difficulties writing variable updates[4,5]:

```
deltap = Fnet*deltat
pf - pi = Fnet*deltat
pfinal = p + Fnet*deltat
```

Difficulties interpreting variable updates:

```
G = 6.7e-11
mEarth = 6e24
mcraft = 15e3
deltat = 60

Earth = sphere(pos=vector(0,0,0),
               radius=6.4e6, color = color.cyan)

craft = sphere(pos=vector(-10*Earth.radius,0,0),
               radius=1e6, color = color.yellow)

vcraft = vector(0,2e3,0)
t = 0

while t < 10*365*24*60*60:
    rate(100)

    r = Earth.pos - craft.pos
    rhat = r/mag(r)
    Fnet = G*mEarth*mcraft*rhat/mag(r)**2

    vcraft = vcraft.pos + (Fnet/mcraft)*deltat
```

"**This is an equation it's using?** I think, there's one unknown in this equation; it's gonna be F net. **So you probably use it to find out what F net is?** V craft I know is a vector and that's a vector we already know so we can cross out V craft and the second V craft as known variables."

```
    craft.pos = craft.pos + (vcraft)*deltat
```

"the position of the craft equals the position of the craft plus V craft times delta T. V craft is a known vector. The position of the craft is a known vector as well. And delta T is a known number. So I'm not sure why this equation exists… so that would mean that V craft delta T would equal zero"

```
    t = t + deltat
```

"Time equals time plus delta time. Delta time is zero? Hmmm. What was T? T equals zero. Zero equals zero plus delta time. Delta time is zero? No, delta time is sixty. How is this making sense?"