

# **PIC Live:**

## **Real-time interactive programming in scientific simulation**

**Ben Swift**

**Andrew Sorensen**

**Henry Gardner**

Australian National  
University

**Viktor K Decyk**

University of California,  
Los Angeles





88

TaxSlayer  
.com

Great Clips

77

BANDIT  
WARRIOR

CAMRY

# SCF Frontiers: towards exascale...

we're doing it **all wrong!**

but we're here at the frontier, we're optimistic—we have ideas about how to make things better

in that context: this is real-time interactive programming in scientific simulation

# Outline

a **tool**: the Extempore programming language/  
environment

a **demonstration**: live programming a particle-in-cell  
(PIC) plasma physics simulation

a **discussion**: what use (if any) is "liveness" in  
computational and simulation science?



(lisp)

◀ COMPUTING

# Computer programming

Learn how to program drawings, animations, and games using JavaScript & ProcessingJS, or learn how to create webpages with HTML & CSS. You can share whatever you create, explore what others have created and learn from each other!

## Create:

- [New Program \(JS + ProcessingJS\)](#)
- [New Webpage](#)
- [New SQL script](#)

## Browse Creations



## Help Requests



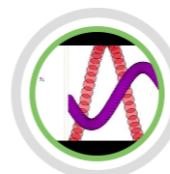
### Intro to JS: Drawing & Animation

In these tutorials, you'll learn how to use the JavaScript language and the ProcessingJS library to create fun drawings and animations. If you've never programmed before, start here to learn how!



### Advanced JS: Games & Visualizations

Now that you know how to program in JavaScript and make basic drawings and animations, how could you use that knowledge to make games and visualizations?



### Advanced JS: Natural Simulations

Learn how to use JavaScript, ProcessingJS, and mathematical concepts to simulate nature in your programs. These tutorials are a derivative of "The Nature of Code" book by Daniel Shiffman ([natureofcode.com](http://natureofcode.com)), used under CC BY-NC.



### HTML/CSS: Making webpages

Learn how to use HTML and CSS to make webpages. HTML is the markup language that you surround content with and it tells the browser about the parts of your page like headings, lists, and tables. CSS is the stylesheet language that you style the page with, which tells the browser to change the color, font, layout, and more.

# Interactive (live) programming

tight feedback loop between programmer **action** and system **reaction**, between **intuition** and **validation**

not so common in HPC (for a variety of socio-technical reasons) although people are starting to talk about real-time/online—streaming, in-situ analysis etc.

# extempore

a programming language & runtime environment

interactive: JIT-compiled on node to native code using LLVM

**subroutines can be hot-swapped, even in tight loops**

eval/update code while the program is running, on one or many nodes

open source (MIT): <https://github.com/digego/extempore>

# But our codes work pretty well...

extempore is ABI compatible with C/Fortran, so you can

- call compiled C/Fortran routines from a shared library
- read and write directly to memory (pointers)
- use e.g. MPI for message passing, OpenMP for threading

*...all with **live** hot-swapping*

# Live PIC: a case study

2½D electromagnetic PIC skeleton code

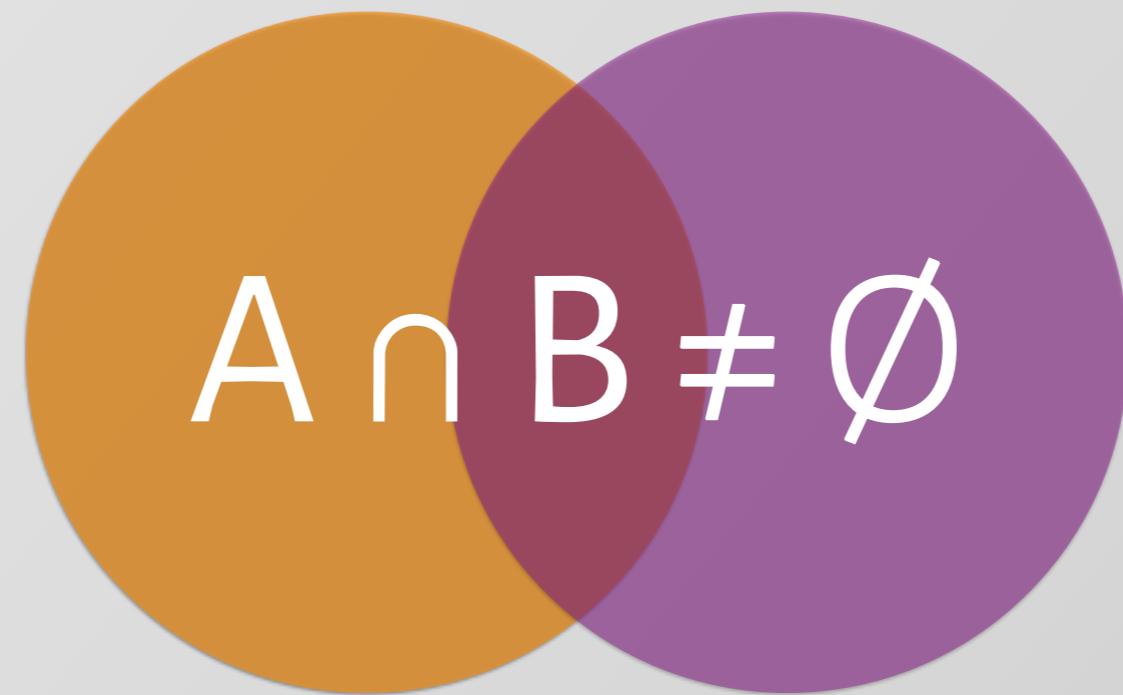
running **live**, distributed memory (MPI)

on my laptop (for practical reasons) but could be on a cluster

visualisation code also live, running on head node

demo

# *When might you want to do this?*



you're not sure *exactly*  
what you're looking for

you can't afford to dump  
*everything* to disk for later  
post-processing

# *Why might you want to do this?*

parameter space exploration (e.g. finding the right beam shape)

in-situ & in-transit analysis

stopping conditions/convergence properties are obvious to a (human) domain expert, but nontrivial to encode in an algorithm

many-worlds: multiple concurrent simulations, make different changes to each, view diffs

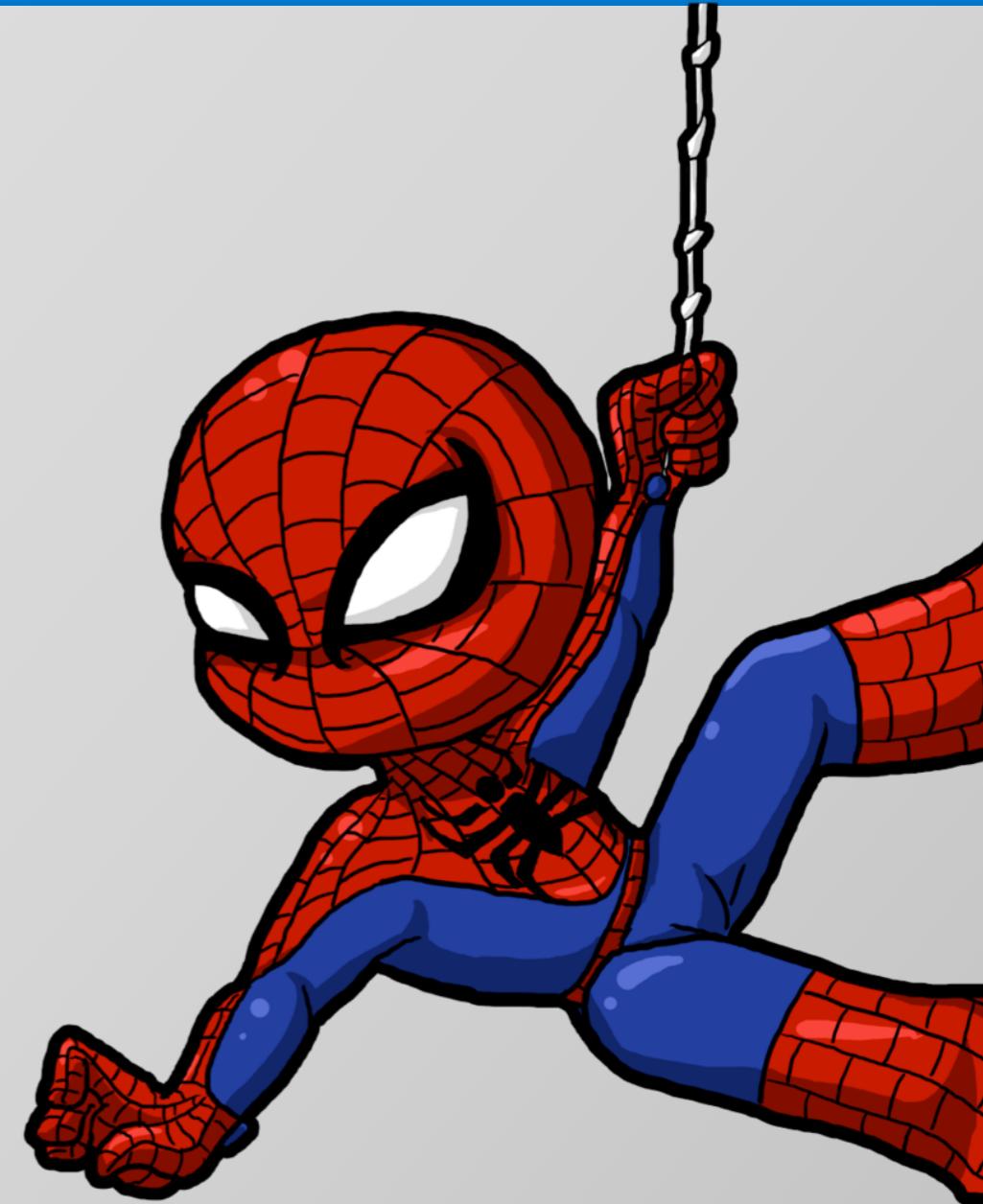
smaller development cluster

# Challenges

safety (the Peter Parker principle)

reproducibility (and provenance)

“stage fright”



allowing scientists to get  
**intimate** with their codes



*Morpho eugenia*  
Guyana



*Morpho melacheilus*  
Peru



*Morpho rhetenor*  
French Guiana



*Morpho godarti*  
Bolivia



*Morpho achilles*  
Brasil



*Morpho catenaria*  
Brasil



*Morpho cypris*  
Colombia



*Morpho helenor*  
Peru



*Morpho ana*  
Brasil



*Morpho adonis*  
Peru



*Morpho menelaus*  
Brasil



*Morpho aega*  
Brasil



*Morpho themisto*  
Brasil



*Morpho telemachus*  
French Guiana



*Morpho cacica*  
Peru



*Morpho sulkowskyi*  
Colombia





ben.swift@anu.edu.au

<https://github.com/digego/extempore>

<http://benswift.me/extempore-docs>

#extempore on freenode