A Taste of Computer Science

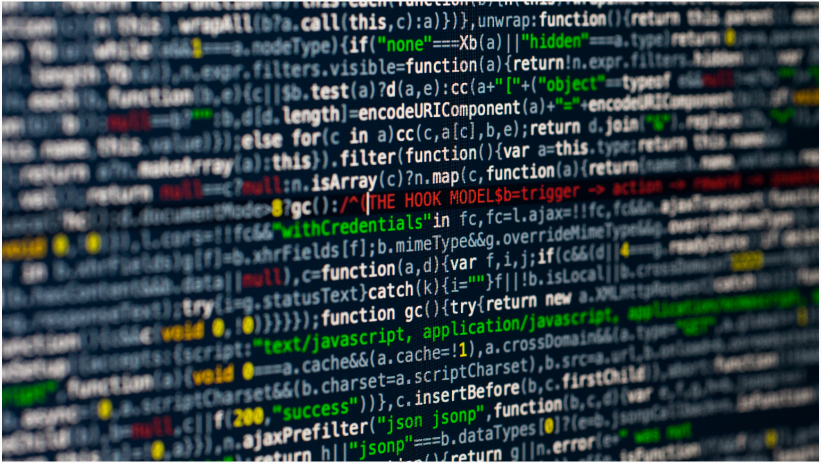
David Bindel

20 July 2021

Cornell Bowers CIS
Cornell University

What words describe CS?

Go to `www.menti.com` and use code 7903 1401.



a.call(this,c):a)})},unwrap:function(){return
{if("none"===Xb(a)||"hidden"===a.type)return
s.visible=function(a){return!n.expr.filters.hidden
$b.test(a)?d(a,e):cc(a+"["+("object"===typeof
d.length]=encodeURIComponent(a)+"="+encodeURIComponent
else for(c in a)cc(c,a[c],b,e);return d.join("&")
this}).filter(function(){var a=this.type;return
n.isArray(c)?n.map(c,function(a){return
gc():/^(THE HOOK MODEL$b=trigger -> action
"withCredentials"in fc,fc=l.ajax=!!fc,
b.xhrFields[f];b.mimeType&&g.overrideMimeType
),c=function(a,d){var f,i,j;if(c&&(d
catch(k){i=""}f||!b.isLocal
g.statusText}catch(k){i=""}f||!b.isLocal
});function gc(){try{return new
"text/javascript, application/javascript,
a.cache&&(a.cache=!1),a.crossDomain&&(a.type=
a.cache&&(b.charset=a.scriptCharset),b.src=a.url
})},c.insertBefore(b,c.firstChild
f(200,"success"))},c.insertBefore(b,c.firstChild
ajaxPrefilter("json jsonp",function(b,c,d){var
h||"jsonp"===b.dataTypes[0]?(e=b.
g||n.error(e+" was not

In the beginning...

- Mid-19th century: Babbage invents the concept of the computer; Ada Lovelace designs programs for it.
- Mid-20th century: First electronic computers, largely used for numerical computations. Grace Hopper invents the first compiler (and found the first “bug” – a moth).
- 1960s: First departments of computer science, often growing out of engineering or mathematics. The term “Computer Science” was coined by George Forsythe (a numerical analyst).
- 1970s: Databases, Unix, new languages; breakthroughs in algorithms, complexity, cryptography.
- 1980s: Rise of the PC, precursors to the Internet

Shape of the Major Now

- Intro programming and math courses
  - You do *not* need to start coding in HS
- Sophomore-junior courses:
  - Discrete mathematics and proof
  - Data structures and functional programming
  - Architecture and intro to systems
- Upper-level courses:
  - Algorithm design and analysis
  - Operating systems
  - + electives and a practicum
- Have some recommended elective sequences (“vectors”)

Many Faces of CS

Theory Systems Applications

Many Faces of CS

- Software engineering
- Systems and databases
- Programming languages and formal methods
- Theory and algorithms
- Network science
- Graphics and vision
- Artificial intelligence and machine learning
- Scientific computing
- … and many more!

(Some of) Many Faces of CS









Prof. Kavita Bala

Professor and Dean,
Cornell Ann S. Bowers College of Computing and Information Science

Vision, graphics, human perception

How do we go from models to pictures? Pictures to models?

Prof. Claire Cardie

Professor and Associate Dean for Education

Natural language processing

What is the argument in a text? How is it structured? Will it persuade?

Prof. Sarah Dean

Assistant Professor (joining this fall!)

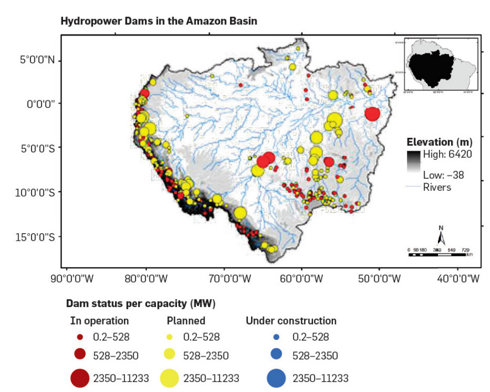
Machine learning and control

How can we learn dynamics while maintaining safety?

Prof. Carla Gomes

Professor

Artificial intelligence,
computational sustainability

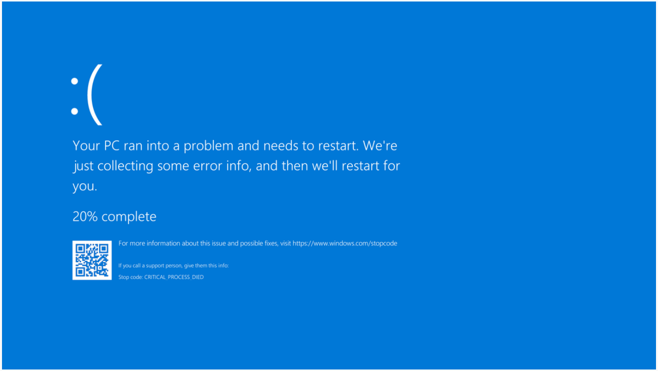


What are the best ways to balance different objectives?

Prof. Owolabi Legunsen

Assistant Professor

Software engineering and formal methods



How do we find find potential failures before they happen?

Prof. Emma Pierson

Assistant Professor

Data science and machine learning for inequality and healthcare

By collecting and analyzing massive data sets and controlling for confounding factors, can we quantify inequal treatment?

Prof. Alexandra Silva

Professor (returning this fall!)

Programming principles,
logic and verification

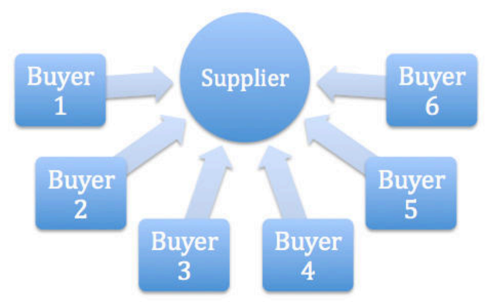
PROBABILISTIC
NETKAT

How can we prove properties of complex network policies?

Prof. Eva Tardos

Professor and Chair

Algorithms and algorithmic game theory



How “good” are different types of auctions?
How can one design auctions with good properties?

Prof. Hakim Weatherspoon

Professor

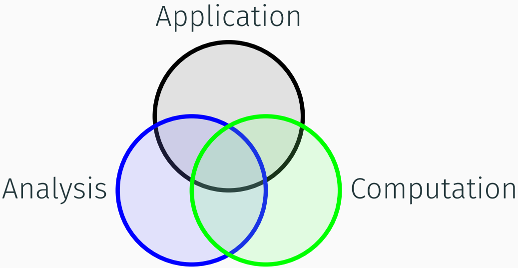
Distributed systems and cloud computing,
Digital agriculture

How can we freely move computations from server to server?

Prof. David Bindel

Associate Professor, Associate Dean for DEI, Director of the Center for Applied Math

Scientific computing, numerical linear algebra, computational mechanics

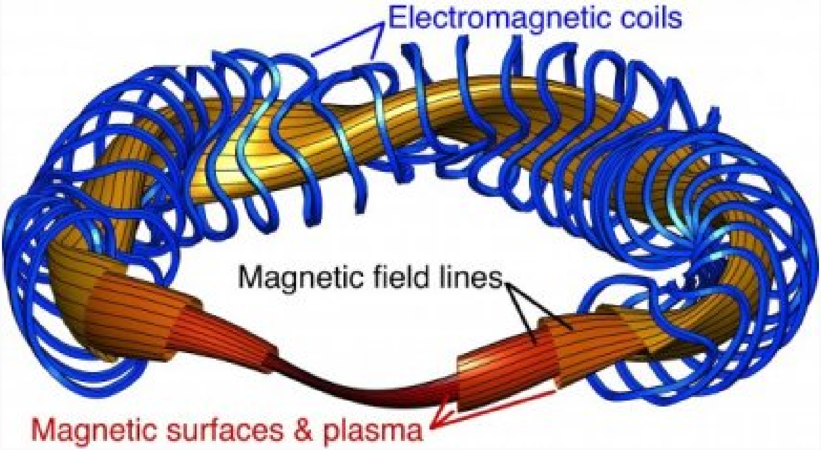


How does application "physics" lead to structure that we can use in fast simulation algorithms?

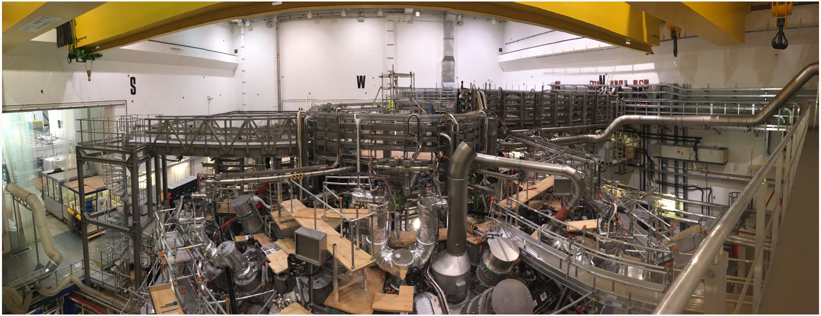
Question Break!

- Ask me a question (or two or three)!
- And if there's time left, I'll tell you some about my research...

Stellarator Concept



Wendelstein 7-X Machine



Operating since 2015-12-10;
plasma discharges lasting up to 30 min.

Stellarator Quality Measures

What makes an “optimal” stellarator?

- Approximates field symmetries (which measures?)
- Satisfies macroscopic and local stability
- Divertor fields for particle and heat exhaust
- Minimizes collisional and energetic particle transport
- Minimizes turbulent transport
- Satisfies basic engineering constraints (cost, size, etc)

Each objective involves different approximations, uncertainties, and computational costs.

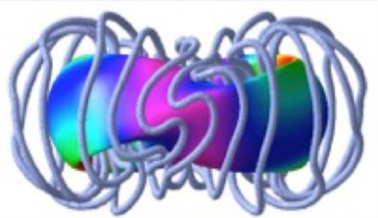
How Do We Optimize? (STELLOPT Approach)

Optimizer

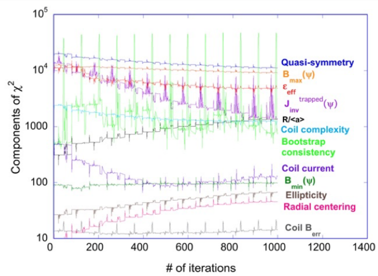
Calculate χ^2 (physics + engineering targets)

Adjust plasma boundary (or coil shape)

Solve 3D equilibrium



$$r(\phi, \theta) + iz(\phi, \theta) = \sum \alpha_{m,n} e^{i(m\phi - n\theta)}$$



Challenges

Why doesn't this framework suffice?

1. Costly and "black box" physics computations
2. Managing tradeoffs (scalarization misses things)
3. Dealing with uncertainties
4. Global search

SIMSOPT incorporating tools to deal with all the above.