SimEcon: A Project to Improve Computation in Economics

Kenneth L. Judd

March 25, 2021

Current state of computational economics

- Reliance on Matlab
- Reliance on laptops and small clusters
- Nice for courses, very limiting for research

Software Resources – Matlab

- Algorithms
  - Linear algebra is very good
  - fmincon and fsolve are not great
  - No automatic differention
  - Few economists are familiar with the options
- Possible improvements
  - Automatic differentation
    - AdiMat
    - CASADI
  - Interfaces with superior solvers
    - Knitro, Ipopt, Snopt, Path,.. via TomLab (not free)
    - Can be slow due to call backs
  - Use parallelization (probably limited capacity)

Software Resources – Modeling languages

- They allow for simple expressions of problems – objective functions, constraints, equations
- They automatically compute derivatives and sparseness structure
- The provide an interface to multiple solvers
- Examples
  - Mature and popular examples: AMPL, GAMS, AIMMS
  - Python: Pyomo
  - AMPL API for Matlab

Hardware Resources

- Modest clusters on campuses
  - UW: >10,000 workstations
  - Purdue: >10,000 workstations
  - PSU: ??
- CoLab
  - Python tools
  - Cost: a cheap pizza per month
- NEOS - a free service
  - University of Wisconsin project funded by NSF
  - submit AMPL or GAMS problem descriptions and get access to EVERYTHING
  - Carlos Rangel constructed a parallelization script

NEOS Server: State-of-the-Art Solvers for Numerical Optimization

The **NEOS Server** is a free internet-based service for solving numerical optimization problems. Hosted by the Wisconsin Institute for Discovery at the University of Wisconsin in Madison, the NEOS Server provides access to more than 60 state-of-the-art solvers in more than a dozen optimization categories. Solvers hosted by the University of Wisconsin in Madison run on distributed high-performance machines enabled by the HTCondor software; remote solvers run on machines at Arizona State University, the University of Klagenfurt in Austria, and the University of Minho in Portugal.

The **NEOS Guide** website complements the NEOS Server, showcasing optimization case studies, presenting optimization information and resources, and providing background information on the NEOS Server.

NEOS Server

- Submit a job to NEOS
- View Job Queue and Job Results
- User's Guide to the NEOS Server
- NEOS Server FAQ
- NEOS Support

NEOS Guide

- NEOS Case Studies
- NEOS Optimization Guide
- NEOS Server Information
- Optimization Resources, LP FAQ and NLP FAQ

Advanced Tools

- Statistics: solvers, websites
- Job Archives 🔒 (password required)
- Downloads: Client Tools (GitHub) and Kestrel

Latest NEOS News

NEOS
@NeosOpt

GAMS (34.3.0) and CPLEX (20.1) are both up-to-date on NEOS now. Thank you @GamsSoftware and @IBM for your support!

Mar 16, 2021

NEOS
@NeosOpt

We've updated all of our AMPL solvers on NEOS. Thanks @AMPLopt for continuing to support us!

Mar 9, 2021

NEOS
@NeosOpt

It is amazing. NEOS works Perfectly. No bugs! :-)

Linear Programming

- bpmpd [AMPL] [LP] [MPS] [QPS]
- Clp [MPS]
- CPLEX [AMPL] [GAMS] [LP] [MPS] [NL]
- FICO-Xpress [AMPL] [GAMS] [MOSEL] [MPS] [NL]
- Gurobi [AMPL] [GAMS] [LP] [MPS] [NL]
- MOSEK [AMPL] [GAMS] [LP] [MPS] [NL]
- OOQP [AMPL]
- SoPlex80bit [LP] [MPS]

Nonlinearly Constrained Optimization

- ANTIGONE [GAMS]
- CONOPT [AMPL] [GAMS]
- FICO-Xpress [MOSEL]
- filter [AMPL]
- Ipopt [AMPL] [GAMS] [NL]
- Knitro [AMPL] [GAMS]
- LANCELOT [AMPL]
- LOQO [AMPL]
- MINOS [AMPL] [GAMS]
- PATHNLP [GAMS]
- SNOPT [AMPL] [GAMS] [NL]

Mathematical Programs with Equilibrium Constraints

- filterMPEC [AMPL]
- Knitro [AMPL] [GAMS]
- NLPEC [GAMS]

Supercomputers

- DOE
  - Priority given to nuclear bomb work
  - Does not like economic policy work
- XSEDE
  - Access to NSF supercomputers
  - Easy to get initial allocation
  - Cloud computing
  - AWS, etc.
  - Open Science Grid

HTCondor

- Developed by Miron Livny at the University of Wisconsin
- "High Throughput" computing
- Distributes parallel tasks across unused CPUs
  - Communication time is greater than supercomputers
  - Designed to deal with other users
  - Enormous capacity
- Doorway to OSG
- Portable to cloud computing systems

One Success: DSICE

Dynamic Stochastic Integration of Climate and Economy (DSICE)

- Includes uncertainty in both the future climate and economy
- Nine dimensions
  - Nordhaus' climate system, LRR productivity shocks, Tipping points
  - 600-year horizon
  - Time period: one year, results were the same down to two weeks.

Computational features

- Massive parallelism: 80,000 cores for six hours were used to solve one example
- 10% efficiency (which is good) and scalable up to 80,000 cores
- Verification – first time done in economics
- Uncertainty quantification (serious parameter sensitivity)

Solves difference equations in Banach spaces

- Describes most models is economics.
- Uses best available numerical methods for approximation, quadrature and optimization
- Easy to incorporate new methods

Recent paper of mine

- Cai, Judd-not-listed-as-author, and Lontzek (JPE, December, 2019)
- All software written in 2013

Our computer

Other Software Projects

- DPSOL, a general solver for DP problems
- Implement alternative assumptions on preferences
  - Risk-sensitive preferences
  - Robust decision making
  - Ambiguity
- Solve dynamic games
  - Supergame algorithm by Yeltekin-Cai-Judd
    - Find all Nash equilibria in games with states
    - Scalable up to 160,000 cores
    - Obvious candidate for asynchronous parallelization, and for exaflop machines
  - Compute all MPE in dynamic games

- DSGE models
  - Judd-Maliar-Maliar GSSA and EDS
    - Has been used on simple models with ~200 dimensions on a desktop
    - Clearly parallelizable
    - Obviously applicable to asymmetric information models, hidden state problems
  - Perturbation methods
  - NLCEQ method
    - Dominates log-linearization since it is a global method
    - Parallelizable
- Solving polynomials
  - Homotopy methods for polynomials
  - Groebner bases for polynomial equations
- Applications
  - Dynamic tax policy (with Yeltekin, Mueller)
    - Barro's random walk theory is wrong
    - Sargent's 2002 JPE paper is a computational atrocity
  - Optimal life cycle tax policy with borrowing costs and constraints (with Rangel, Mueller)
  - Computing likelihood level sets (with Reich)
  - Solving for time consistent solutions of dynamic games

Radical Idea: Do what Astrophysicists Do: FLASH

- Community code for astrophysics and others who want to model explosions
- Economics and Astrophysics are similar
  - Experiments are very difficult
    - Must use observations to build models
    - Must use computational tools to validate the models
- Developed by Bob Rosner (UChicago, Argonne, DOE Advanced Scientific Computing,...)
- FLASH

My Proposal: SimEcon

- Create a community library of code of common economic models
- Codes should be in multiple languages
- Codes should have access to multiple solvers
- Library should contain examples of deploying the codes to alternative systems
  - Local clusters
  - XSEDE supercomputers
  - HTCondor, Open Science Grid

SimEcon applications

- Use the algorithms we have and computer power we could have to solve for equilibria of economic models
  - In macro, specify a combination of multiple sectors, agent types, heterogenous beliefs, bounded rationality, etc.
  - In empirical IO, specify payoff functions, structure of games, etc.
- Specify parameters and use DPSOL, GSSA, EDS, NLCEQ, FEniCS, PETSc, COMSOL, LS-DYNA, Trilinos, APPSPACK, POUNDERS, BORG, etc to solve the model
- Applications
  - Generate synthetic data
  - Apply your simple models and empirical methods to the synthetic data
  - Determine if your simplifications (models, approximations, empirical methods) can produce correct insights about the true, complex model.

- Purpose – "Dimension Reduction"
  - It would be infeasible for each researcher to solve a large model and incorporate it into their theoretical or empirical analysis.
  - However, researchers should limit themselves to methods that we know can reliably analyze models that are far more complex and realistic.
  - Parsimony may be a virtue for standard economic research papers but only reliable parsimonious models.
  - Which parsimonious models should be used can be determined by testing them against data generated by a range of complex models that are far more realistic
  - This is done in climate work, including in Nordhaus' first papers.

SimEcon: Outline of Strategy

- Expand training of software tools
  - Some students often know more than the profesors
  - Incentivize those to teach other students
  - The old students can teach the next generation
- Improve access to hardware tools
  - Provide material to help students learn how to gain access to those tools
  - Deploy major examples on those systems to help students learn how to use them

SimEcon: Action Items

- Create a community
  - My collaborators
  - Students from my 2020 course
  - PSU students
- Collect information
  - What are the resources at U of X?
  - Computer manuals are useless; often written for CS people
  - We need manuals written for economists
- Organize computational projects
  - Replication, but on various systems
  - Introduce unused computational tools: DFO, for example
- Judd's role
  - Coordination
  - Quality control
  - Organize meetings - some even in California!
  - Provide some funding, hoping it can be leveraged
  - Publicize the results

Conclusions

- Economists currently use only a fraction of the computational tools available
- Existing academic infrastructure is inadequate
- Young people have a lot of computational expertise
- Let's work together to expand the role of modern computational science in economics.