Computational Challenges in Power Grid Simulations

Opportunities for Massively Parallel Processing

Zeb Tate
zeb.tate@utoronto.ca

Department of Electrical and Computer Engineering
Energy Systems Group
University of Toronto

Research supported by NSERC DG

July 14, 2010 / GPU Workshop

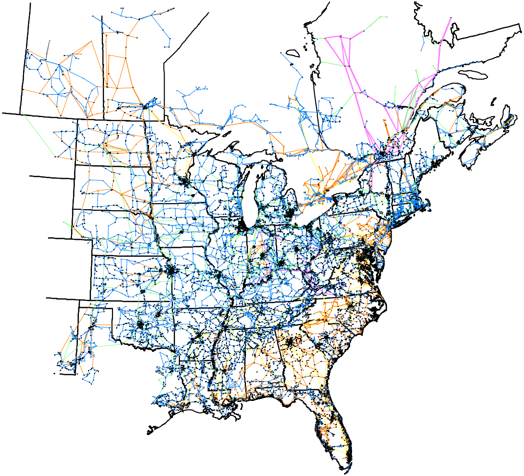
Outline

- Power System Model
  - Steady-State Models
  - Dynamic Models
  - Cyberinfrastructure
- Applications
  - Transient Stability
  - Contingency Analysis
  - Market Operations
- Current Research
  - Accelerated Solution of Linear Systems Arising from Power System Computations

Overview

- The power grid is (arguably) the most complex man-made system ever constructed
    - Many of the other complex systems we simulate (e.g., the Internet) are components of the power grid
- Sparse, unstructured, nonlinear, differential-algebraic
- Example: the eastern interconnection (EI), which includes Ontario, is the largest power grid in North America
    - 45,000 high voltage (13.8 kV and above) nodes
    - 63,000 branches (transmission lines and transformers)
    - 28,000 loads (650 GW)
    - 7,000 generators (670 GW)

Diagram of the EI High Voltage Network



Outline

Power System Model
- Steady-State Models
- Dynamic Models
- Cyberinfrastructure

Applications
- Transient Stability
- Contingency Analysis
- Market Operations

Current Research
- Accelerated Solution of Linear Systems Arising from Power System Computations

Ac Power Flow Model

- Used to analyze the steady-state behavior of the power system
- Input data
  - Branch models (e.g., the resistances, inductances, and capacitances of the transmission lines)
  - Description of the devices connected to each node (e.g., the power demands of each load)
- Output data
  - Voltages (magnitude and phase) at each node
  - Currents through each branch
  - Generation levels (if not specified as an input)

Ac Power Flow Model

- Constant frequency is assumed, so the unknowns are the **magnitude** and **angle** of the bus voltages ($\sqrt{2}\mathbf{V} \times \cos(2\pi f \times t + \theta)$)
- 2 equations are written at each node of the system based on conservation of (complex) power, resulting in a total of $2N$ equations (for EI, approximately 90,000 equations and unknowns)
- Nonlinear because the withdrawal/injection at each bus is given in terms of power, not current
- Usually solved with Newton-Raphson and LU decomposition
- Account for the sparsity of the system (typical branching factors are 2-3) throughout

Dc Power Flow Model

- Simplication of the ac power flow based on several assumptions
  - Voltage magnitudes are near their nominal values
  - Phase angle differences across branches are small
- Results in a linear system of equations
- Simplest model of the power grid that accounts for differences in branch impedances throughout the network
- Works poorly when the system is heavily stressed, but in many cases this is the model used due to computational limitations

Outline

Power System Model
- Steady-State Models
- Dynamic Models
- Cyberinfrastructure

Applications
- Transient Stability
- Contingency Analysis
- Market Operations

Current Research
- Accelerated Solution of Linear Systems Arising from Power System Computations

Electromechanical Model

- Generators are electromechanical systems with complex dynamics and controls
- To simulate the system trajectory (e.g., during and after a substation explodes) when the system is not in steady state, differential equations for the generators and loads must be included
- The simplest model treats the shaft as a rigid body, neglects all electrical transients within the machine, and assumes the mechanical input power is fixed
- More complex models include:
    - Mechanical dynamics within the shaft
    - Electrical transients within the machine
    - Mechanical input power control
- Reduced order dynamic models and network equivalents are often used to trade accuracy for reduced computation times

Outline

Power System Model

- Steady-State Models
- Dynamic Models
- Cyberinfrastructure

Applications

- Transient Stability
- Contingency Analysis
- Market Operations

Current Research

- Accelerated Solution of Linear Systems Arising from Power System Computations

Modeling the Power Grid’s Cyberinfrastructure

- “Smart grid” applications rely on better data collection, communication, & computation
- Currently dominated by point-to-point communications (e.g., modem over leased line)
- Moving towards routable protocols within the substation & between substations (IEC 61850)
- Non-deterministic latency impacts control capability

Outline

Power System Model

- Steady-State Models
- Dynamic Models
- Cyberinfrastructure

Applications

- Transient Stability
- Contingency Analysis
- Market Operations

Current Research

- Accelerated Solution of Linear Systems Arising from Power System Computations

Transient Stability

- Does the system remain stable after something happens (e.g., a generator suddenly goes offline)?
- Because there is no closed form solution, numerical integration is used
- Often want to know margins (i.e., “room left” before a problem occurs), which requires repeated simulations
- The search for critical points (from which the margins are calculated) should benefit from parallel computation

Outline

Power System Model
  - Steady-State Models
  - Dynamic Models
  - Cyberinfrastructure

Applications
  - Transient Stability
  - Contingency Analysis
  - Market Operations

Current Research
  - Accelerated Solution of Linear Systems Arising from Power System Computations

Contingency Analysis

Present Requirements

- No single contingency (e.g., a line touching a tree and being switched out of service) may result in a loss of load
- Predict contingency effects using a state estimate
- Linearized ac (more accurate) or dc (faster) power flow models are used to simulate contingencies
    - Although each contingency can be solved in parallel, some data sharing may be of use depending on the contingency under consideration

Contingency Analysis

Future Requirements

- Check multiple contingencies
    - Quickly becomes intractable (e.g., TVA (Tennessee Valley Authority) uses a state estimator that models over 7,000 buses and 10,000 branches)
- Probabilistic criteria (e.g., $P(\text{Loss of load}) \leq x\%$)
- Online transient stability assessment
- Base evaluation on forecasts of renewable and distributed generation
- Use higher resolution data

Outline

Power System Model
- Steady-State Models
- Dynamic Models
- Cyberinfrastructure

Applications
- Transient Stability
- Contingency Analysis
- Market Operations

Current Research
- Accelerated Solution of Linear Systems Arising from Power System Computations

Unit Commitment

- Unit commitment is the selection of which generators will be on/off
- Carried out 6-12 hours before the day begins based on bids submitted to the market
- Optimal solution is difficult to obtain
    - Generator statuses are binary variables
    - Uncertain load, generator availability, generator output
    - Inter-time dependencies due to ramp rates, startup times, minimum up/down times
- Simple network models are used to obtain mixed-integer linear programs
- Stochastic nature of generation/load ignored, or extreme values used
    - Monte Carlo simulation could be used instead to provide a better estimate of possible outcomes
- Similar issues in real-time market dispatch

Outline

- Power System Model
  - Steady-State Models
  - Dynamic Models
  - Cyberinfrastructure
- Applications
  - Transient Stability
  - Contingency Analysis
  - Market Operations
- Current Research
  - Accelerated Solution of Linear Systems Arising from Power System Computations

Overview

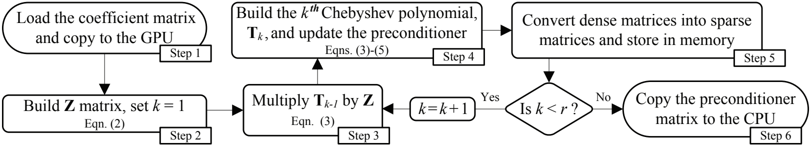
Primary developer - Amirhassan Asgari (M.A.Sc. student)

- Linear system solution is the main computational kernel in power system simulation
- High-dimensional, sparse, direct solvers (e.g., based on LU decomposition) are difficult to adapt to the GPU architecture
  - Diminishing opportunities for parallel computation as more rows are eliminated
  - Difficult to make memory accesses coherent
- Indirect solvers (e.g., conjugate gradient) are well-suited to the GPU architecture, but challenges must be overcome
  - Ill-conditioned nature of power system matrices, particularly if the system is close to maximum loadability
  - Sparsity must be maintained to ensure storage is not exhausted
- Develop GPU-based preconditioner and solver for use in power system simulations

Chebyshev Preconditioner

Overview

- Polynomial preconditioners rely on matrix-matrix multiplies
- Prior research indicates that Chebyshev polynomial preconditioner are a viable option
- Implementation flowchart:



Load the coefficient matrix and copy to the GPU
Step 1

Build **Z** matrix, set $k = 1$
Eqn. (2)
Step 2

Multiply $\mathbf{T}_{k-1}$ by **Z**
Eqn. (3)
Step 3

Build the k^{th} Chebyshev polynomial, $\mathbf{T}_k$, and update the preconditioner
Eqns. (3)-(5)
Step 4

Convert dense matrices into sparse matrices and store in memory
Step 5

Is $k < r$?

Yes

$k = k + 1$

No

Copy the preconditioner matrix to the CPU
Step 6

Chebyshev Preconditioner

Implementation

- A custom kernel was developed to perform sparse matrix-matrix multiplication
  - For each element in the product, vector-vector dot product is performed with explicit consideration of sparsity in both operands
  - Intermediate vectors from the product matrix are stored in a dense format, then converted to compressed row storage format via a prefix sum calculation
- Increasing the number of polynomial terms results in diminishing returns, while at the same time decreasing the sparsity of the preconditioned matrix. Using three terms has worked well for the matrices studied so far.

Chebyshev Preconditioner

Results

Matrix Name	Size	Cond# ($r = 0$)	Cond# ($r = 3$)	CG# ($r = 0$)	CG# ($r = 3$)
30-Bus	30	913.23	71.80	37	14
57-Bus	57	1.6×10^3	147.14	57^{nc}	22
118-Bus	118	3.9×10^3	288.47	112	30
300-Bus	300	1.1×10^5	5732.5	300^{nc}	72
494-Bus	494	3.9×10^6	32855.1	494^{nc}	216
685-Bus	685	5.3×10^5	3542.51	685^{nc}	112
EU case	1243	3.2×10^5	14244.3	1243^{nc}	117
bcsstk14	1806	1.3×10^{10}	1.2×10^5	1806^{nc}	52
bcsstk15	3948	7.9×10^9	8.9×10^5	3948^{nc}	247

Chebyshev Preconditioner

Results

Test Matrix		ILU			Chebyshev			Speed-up	
Name	nnz	drop tol.	nnz	Time CPU(ms)	nnz	Time CPU(ms)	Time GPU(ms)	vs. CPU ILU	vs. CPU Cheb.
30-Bus	108	6×10^{-2}	170	0.299	258	0.27	0.47	0.63	0.57
57-Bus	205	4×10^{-2}	322	0.365	461	0.68	1.25	0.29	0.54
118-Bus	464	4×10^{-2}	812	0.488	1198	1.55	2.30	0.21	0.67
300-Bus	1121	5×10^{-4}	7088	2.110	2890	7.69	6.47	0.32	1.18
494-Bus	1080	4×10^{-4}	6534	8.096	4062	19.7	8.06	0.99	1.83
685-Bus	1967	4×10^{-3}	18386	23.33	9337	44.1	13.72	1.70	3.21
EU case	4872	1×10^{-3}	23976	43.42	13873	124.2	21.03	2.06	5.91
bcsstk14	32630	6×10^{-6}	272878	90.12	195654	262.2	33.46	2.69	7.85
bcsstk15	60882	4×10^{-6}	1029908	547.1	527666	591.1	66.18	8.26	8.93

BiCG-STAB Solver

- Initial implementation of the stabilized bi-conjugate gradient solver on the GPU
- Preliminary results:

Matrix Name	Size	NNZ	Time Chebyshev CPU(ms)	Time Chebyshev GPU(ms)	Time BiCG CPU (ms)	Time BiCG GPU (ms)	Speed - up GPU versus CPU
30-Bus	30	108	0.27	0.47	1.6	1.9	0.79
57-Bus	57	205	0.68	1.25	7.1	5.3	1.18
118-Bus	118	464	1.55	2.30	12.3	11.6	0.99
300-Bus	300	1121	7.69	6.47	26.8	25.4	1.08
494-Bus	494	1080	19.7	8.06	44.8	35.1	1.49
685-Bus	685	1967	44.1	13.72	44.13	33.8	1.85
Bcsstk14	1806	32630	262.2	33.46	354.3	193.1	2.68
Bcsstk15	3948	60882	591.1	66.18	2553.1	263.7	9.53

Next steps

- Profiling and refinement of existing code using nsight tools
- Optimization of preconditioner/solver tradeoff
- Evaluation of alternative preconditioners/solvers
- Comparison with commercial simulators

Summary

- Ensuring reliable operation of the power grid is already pushing the limits of serial processing, even when grossly simplified models are used
- New reliability requirements, greater variability in generation, and more sophisticated controls will require parallel computation techniques
- Develop techniques that utilize all available resources (e.g., CPUs, GPUs, FPGAs)