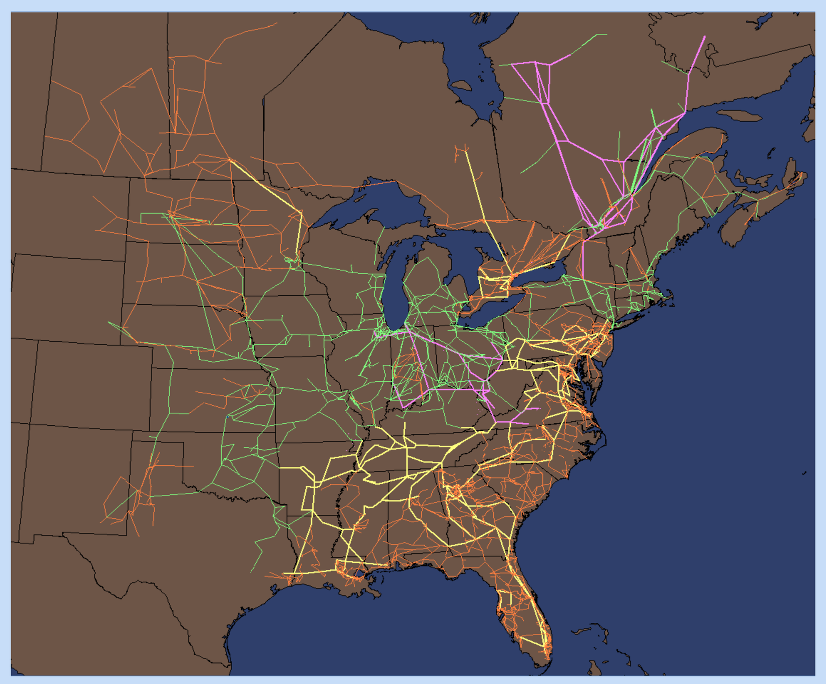
Dynamic Power System Analysis

Ashley Cliff Central College, Iowa
Mentors: Srdjan Simunovic, Aleksandar Dimitrovski, ORNL; Kwai Wong, UTK

Abstract

The purpose of this project is to create accurate simulations of power outages that can be used to shorten the duration and number of occurrences of power failures. For the simulations to be useful, they must be able to run faster than real time, to determine what will happen when there's an outage before the outcome occurs.



Steady State System

- Basis for Dynamic Systems
- Determine voltage and voltage angles
- Match real power and imaginary power generation to consumption

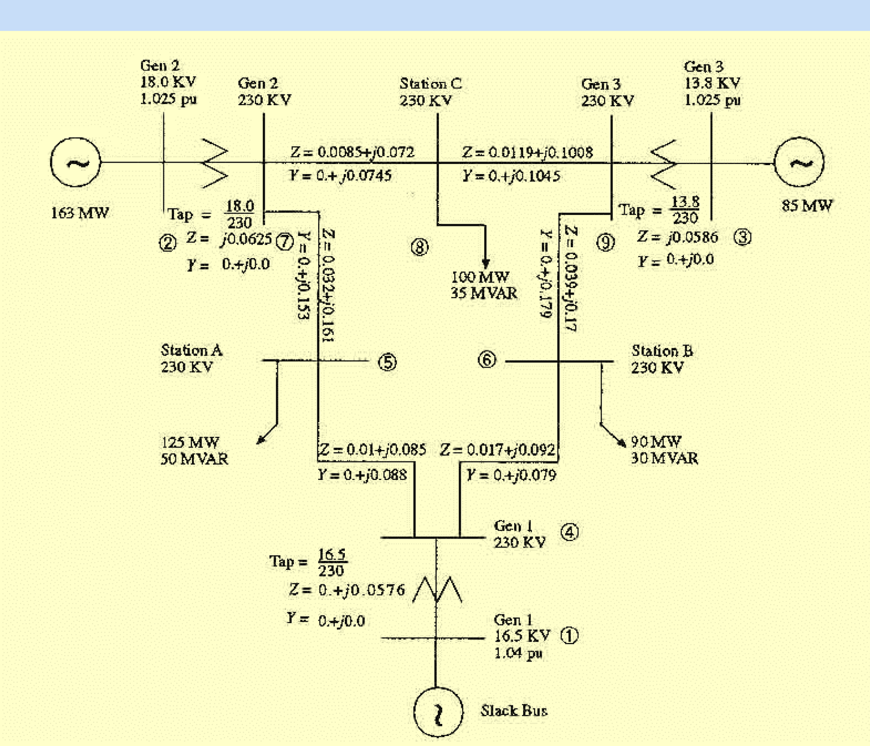
$$P_i^{sp} = P_i(\theta, V) = V_i \sum_{k=1}^{n} V_k (G_{ik} \cos\theta_{ik} + B_{ik} \sin\theta_{ik})$$

$$Q_i^{sp} = Q_i(\theta, V) = V_i \sum_{k=1}^{n} V_k (G_{ik} \sin\theta_{ik} - B_{ik} \cos\theta_{ik})$$

Coupled non-linear algebraic equations

$$Y = \begin{bmatrix} (y_{12}+y_{13}) & -y_{12} & -y_{13} & 0 \\ -y_{12} & (y_{12}+y_{23}) & -y_{23} & 0 \\ -y_{13} & -y_{23} & (y_{13}+y_{23}+y_{34}) & -y_{34} \\ 0 & 0 & -y_{34} & y_{34} \end{bmatrix}$$

Admittance Matrix



3 Generator, 9 Bus System

MatPower takes an admittance matrix created from a bus diagram and solves the power equations

Dynamic System

Rotor Electrical Equations:

$$\frac{dE'_d}{dt} = \frac{1}{T'_{qo}}\left[-E'_d - (X_q - X'_q)I_q\right]$$

$$\frac{dE'_q}{dt} = \frac{1}{T'_{do}}\left[-E'_q + (X_d - X'_d)I_d + E_{fd}\right]$$

Rotor Mechanical Equations:

$$\frac{dS_m}{dt} = \frac{1}{2H}\left[-DS_m + T_m - T_e\right] \qquad \frac{d\delta}{dt} = w_B S_m$$

Excitation Equations:

$$\frac{dV_2}{dt} = \frac{1}{T_F}\left[-V_2 + \frac{K_F}{T_F}E_{fd}\right] \qquad \frac{dV_1}{dt} = \frac{1}{T_R}\left[-V_1 + V_t\right]$$

$$\text{if } T_R = 0 \text{ then } V_1 = V_t$$

$$\frac{dE_{fd}}{dt} = \frac{1}{T_E}\left[-\left(K_E + A_E\left(e^{(B_E E_{fd})}\right)\right)E_{fd} + V_R\right]$$

Governor Equation:

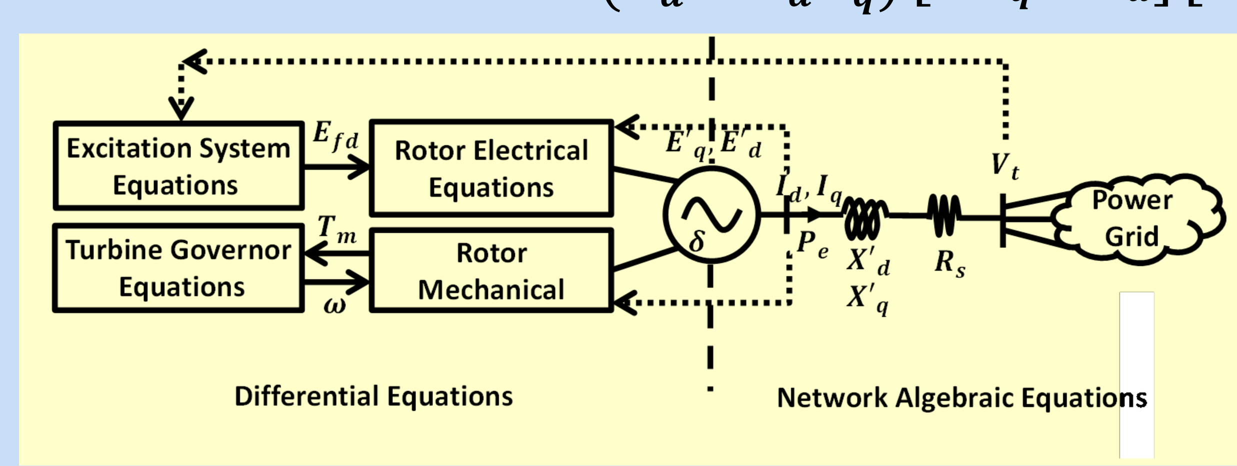
$$\frac{dP_{SV}}{dt} = \frac{1}{T_{SV}}\left[-P_{SV} + P_C - \frac{1}{R_D}S_m\right]$$

Turbine Equation:

$$\frac{dT_m}{dt} = \frac{1}{T_{CH}}\left[-T_m + P_{SV}\right]$$

Network Algebraic Equations:

$$\begin{bmatrix} i_q \\ i_d \end{bmatrix} = \frac{1}{(R_a^2 + \acute{X}_d \acute{X}_q)}\begin{bmatrix} R_a & \acute{X}_d \\ -\acute{X}_q & R_a \end{bmatrix}\begin{bmatrix} \acute{E}_q - V_q \\ \acute{E}_d - V_q \end{bmatrix}$$



Parareal Implementation

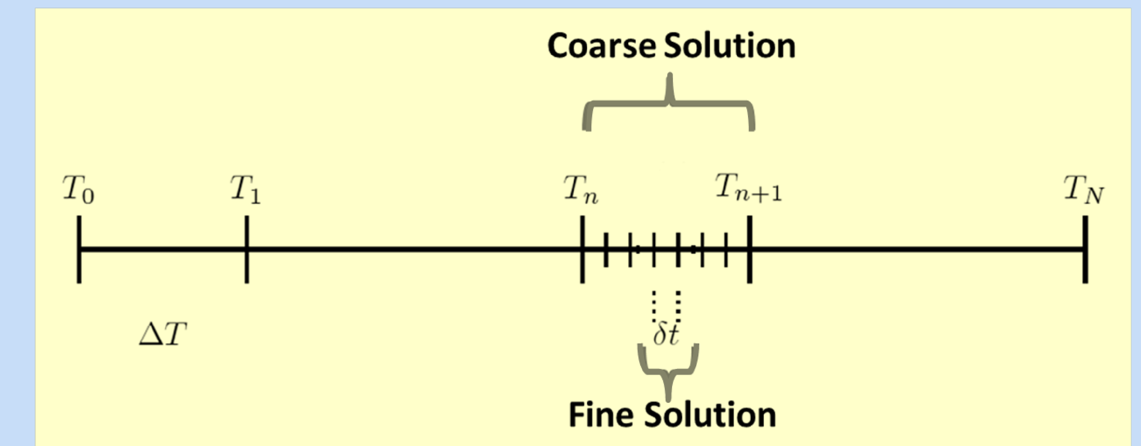
Once a fault has tripped, the final solution for the steady state system is used as the initial values for the dynamic system problem. The goal is to accurately simulate how the fault changes the system as time goes on.

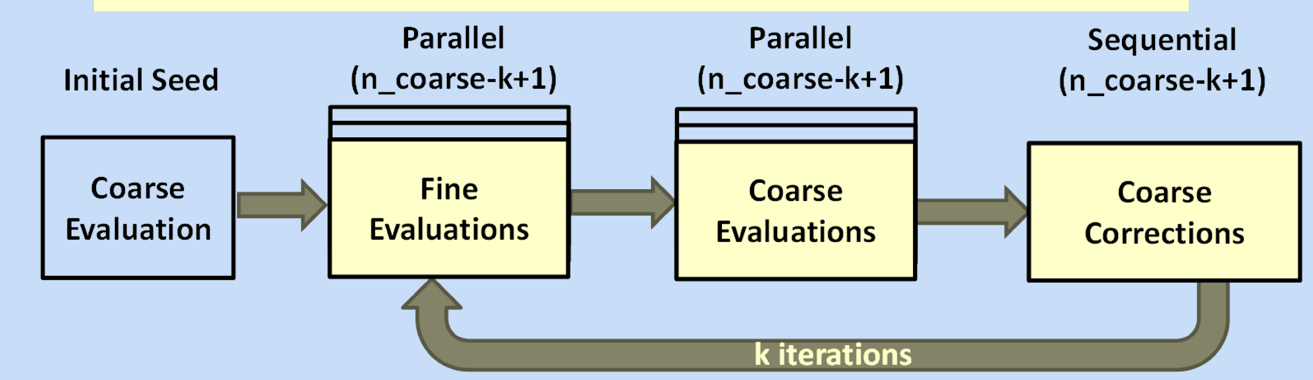
The RK4 method is then used to determine the state values for the next iteration. This process is repeated until the error is within a designated margin or the max number of iterations is reached.

Parareal: Time sections can run at the same time, with a coarse approximation used to generate initial values for each iteration

Parareal Concept

The Parareal in Time Algorithm divides the time domain into intervals, and integrates concurrently over each interval.





MATLB Pseudocode

Trapezoid Function Call – Initial coarse evaluation
While iterations less than max number of iterations:
    For each coarse section:
        Runge-Kutta 4 Function Call - fine evaluation
    Correct coarse evaluation
    Add one to iteration count

Current Work

- Developing code so that 'for loop' runs in parallel
- Test parallel capabilities of MATLAB
  - Matrix-Matrix Multiplication 512 iterations
    - Serial Time: 236 seconds
    - Parallel Time: 262 seconds
  - Matrix-Matrix Multiplication 1024 iterations
    - Serial Time: 401 seconds
    - Parallel Time: 314 seconds
- Only after a large number of iterations, parallel executes faster

Acknowledgements

Thanks to NSF, University of Tennessee Knoxville and Oak Ridge National Lab.