# Variable-Step Multi-Stage Integration Methods for Fast and Accurate Power System Dynamics Simulation

Shrirang Abhyankar, Emil M. Constantinescu, Alexander Flueck

***Abstract*—Fast dynamics simulation of large-scale power systems is a computational challenge because of the need to solve a large set of stiff, nonlinear differential-algebraic equations (DAE) at every time step. Central to the solution of this DAE system is a numerical integration scheme with the second-order alternating explicit integration scheme being a popular choice. In this paper, we present an evaluation of variable step multi-stage time-stepping (numerical integration) schemes for accelerating power system dynamics simulation. Two classes of multi-stage numerical integration methods; Runge-Kutta and Rosenbrock; are introduced and evaluated on different power system networks demonstrating their accuracy and performance. Performance of these variable time-stepping schemes is compared against existing numerical integration schemes used in dynamics simulations on three test systems; 9-bus, 118-bus and 9451 bus networks. Impact of discontinuities or limit handling on variable time-stepping is also examined.**

## I. Introduction

Power systems undergo disturbances of various types such as balanced and unbalanced short circuits, equipment outage of generators, transmission lines and other equipments, breaker tripping, and others [1]. Maintaining stability of the power system on incidence of such disturbances is of prime importance to system planning engineers for ensuring a secure and uninterrupted supply to the consumers. Power system stability is the ability of an electric power system, for a given initial operating condition, to regain a state of operating equilibrium after being subjected to a physical disturbance, with most system variables bounded so that practically the entire system remains intact [2]. Transient stability analysis studies the ability of the power system to keep its rotating machines in synchronism, when subjected to a disturbance (perturbation), during and after the perturbation. A common disturbance simulation involves applying a fault at a given bus at some pre-specified time and removing the fault by opening a circuit element at another pre-specified time. This type of disturbance scenario can help determine the critical clearing time of the circuit breakers and thereby protect the electrical machines from going out of synchronism.

Shrirang Abhyankar (e-mail: abhyshr@anl.gov) is with the Energy Sciences Division, Argonne National Laboratory, Argonne, IL 60439.

Emil Constantinescu (e-mail: econsta@mcs.anl.gov) is with the Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL 60439.

Alexander Flueck (e-mail: flueck@iit.edu) is with the Department of Electrical and Computer Engineering, Illinois Institute of Technology, Chicago, IL 60616.

System planning engineers routinely conduct dynamic security assessment (DSA) studies to analyze the impact of different events, such as a new generator interconnection request or a new equipment installation, e.g. SVCs, or the loss of critical buses, lines, or loads. Such studies require the simulation of a lot of what-if scenario that analyze the system impact to disturbances to the base case and different contingency conditions. Typically, such studies for large utility networks are extremely time-consuming and hence DSA is restricted to offline analysis. With the power system becoming tightly interconnected and the advent of new technologies on the distribution system, conducting even larger dynamics impact studies in shorter time-frame is a crucial need. This need for faster power grid dynamics simulation (or transient stability analysis) has also been reiterated by the power system community in recent years [3], [4]. Providing system operators with the capability to examine the system dynamics in real-time, or faster than real time, creates a new paradigm in power system operation by facilitating real time or predictive control actions. This is particularly important because the power system behavior is becoming increasingly dynamic.

A number of factors contribute to this increased dynamic behavior of power systems. First, today's power systems are being stressed due to increasing load in certain zones, yet transmission line capacity is stagnant. Second, the economic benefits of deregulation have pushed the system operation to its limits, decreasing the stability margins. Third, the distribution system is becoming increasingly "active", leading to possible two-way power flows from distribution to transmission. Fourth, the lowering of system inertia, as more renewable energy sources are installed, has introduced frequency and voltage control problems. Fifth, situational awareness of variability and unpredictability of these renewable energy sources, for example, decreased output of a solar PV installation due to cloud cover, is a need expressed by the operators. Finally, the prediction and mitigation of cascading outages, that cannot be predicted by snapshots of the current operating state obtained every few seconds by state estimation, is an important concern for utilities.

A key requirement to achieve faster dynamics analysis is the acceleration of dynamics simulation tools. However, dynamics simulation of a utility-sized networks comprising thousands of buses is challenging because of the difficulty in solving a set of stiff nonlinear differential-algebraic equations containing tens of thousands of variables. For instance, Huang and Nieplocha [5] reported that a simulation of 30 seconds of

dynamic behavior of the Western Interconnection of the United States requires about 10 minutes of computation time on an optimized single processor.

The central goal of this work is to explore accurate and fast time-stepping or numerical integration methods. In this paper, we introduce several multi-stage variable time-stepping numerical integration schemes and examine their performance, in terms of robustness and speed, on dynamics simulation of different test cases. Section II presents the different-algebraic power system model and discusses the existing numerical integration schemes and solution approaches. Next, we expand this discussion by including variable time-step adaptivity based on local truncation error and handling of discontinuities using root-finding or event detection. This section also briefly touches upon time-step adaptivity based on local truncation error. Two classes of multi-stage integration schemes – Implicit Runge-Kutta and Rosenbrock – are introduced in section III. Accuracy and speed evaluation of the variable time-step multi-stage schemes against the existing approaches is presented in section IV on the IEEE-9 bus, IEEE-118 bus and PEGASE 9451 [6] bus test cases. All the numerical integration schemes discussed in this paper are publicly available through the open-source high-performance library PETSc [7].

## II. Power system dynamics equations

The analysis of bulk power system dynamics is computationally challenging because of the presence of a large set of stiff, nonlinear differential-algebraic equations (DAEs). The electrical power system is expressed as a set of nonlinear DAEs, where the differential equations model dynamics of the rotating machines (e.g., generators and motors) and the algebraic equations represent the transmission system and quasi-static loads.

$$\begin{aligned} \frac{dx}{dt} &= f(x,y), \qquad p(x^-) \le x \le p(x^+) \\ 0 &= g(x,y) \end{aligned} \tag{1}$$

The solution of the dynamic model given in (1) needs the following:

- A numerical integration scheme to convert the differential equations in algebraic form
- A nonlinear solution scheme to solve the resultant nonlinear algebraic equations
- A linear solver to solve the update step at each iteration of the nonlinear solution

In prior work on accelerating transient stability simulations, we focused on accelerating the linear solver [8], [9] : one of the main bottlenecks of dynamics simulation. In this paper, we restrict our discussion to the numerical integration or time-stepping scheme used for the solution of the DAE system. Note in (1), a subset of the differential variables, $x$, are discontinuous in nature due to the presence of nonlinearities such as saturation, over and under excitation limits, valve limiters, and others. Thus, (1) is a discontinuous differential-algebraic model (DDAE) or hybrid model.

There are two approaches to solve (1) - alternating explicit and simultaneous implicit [1]. In the alternating approach, the differential variables are first updated, $x^{n+1} \leftarrow \mathcal{H}(x^n)$, where $\mathcal{H}$ is an explicit integration scheme. Once the dynamic variables are updated, the $y$ variables are solved for the $n+1$ time-instant by solving the algebraic equations $g(y^{n+1}) = 0$. A typical choice of the explicit integration scheme is a second-order Runge-Kutta method or Trapezoidal method. The advantage of an alternating explicit approach is the ease of implementation due to separation of the differential and algebraic parts. However, they may need to use shorter time-steps to ensure stability of the numerical integration scheme. Commercial transient stability tools, for e.g., PSSE [10], PowerWorld [11], use an alternating explicit approach for the simulation of power system transient stability equations.

In the simultaneous implicit approach, the differential equations are first converted to algebraic form using an implicit numerical integration formula and the two-sets of nonlinear equations solved together using Newton's method as given in (2).

$$\begin{aligned} \mathcal{H}(f(x^{n+1},y^{n+1})) &= 0, \qquad x^- \le x^{n+1} \le x^+ \\ g(x^{n+1},y^{n+1}) &= 0 \end{aligned} \tag{2}$$

A simultaneous implicit approach requires the solution of a bigger nonlinear algebraic system given in (2), thus more time-consuming than the alternating explicit approach. But, it has better stability properties even for larger time-steps. EUROSTAG [12] and Powertech's TSAT[13] are commercial softwares that employ implicit integration schemes.

### A. Variable time-stepping

A fixed time-step numerical integration scheme is popular in the research community and employed in all commercial transient stability tools. However, adapting the step-size according to the dynamic behavior can be advantageous particularly if the dynamics are smooth allowing the use of a larger time-step. Most variable step methods alter the integration step-size based on a calculation of the local truncation error (LTE). If the error is large, a smaller step is taken, and vice-versa. A typical calculation of LTE entails a comparison with an estimated solution obtained cheaply through a lower order method.

$$\Delta t_{new}(t) = \Delta t_{old} \frac{||z_n - \hat{z_n}||}{\epsilon_{abs} + \max(z_n, \hat{z}_n)\epsilon_{rel}} \tag{3}$$

Here, $z_n := (x_n, y_n)$ is the solution at $n^{\text{th}}$ timestep, $\hat{z}_n$ is the estimated solution obtained using lower order method, $\Delta t_{new}(t)$ is the new time-step taken, and $\epsilon_{abs}$, $\epsilon_{rel}$ are the absolute and relative tolerances.

In terms of the dynamic behavior, a small time-step is taken when the dynamics are varying rapidly, and vice-versa. The LTE calculation is done using a comparison with a lower-order method. The largest and the smallest step taken by the integration scheme can be restricted to reasonable values to ensure satisfactory convergence of the scheme and avoid missing discontinuities, such as voltage regulator limits and others.

### B. Handling discontinuities

One of the reasons variable time-stepping schemes are seldom explored for dynamics simulation is due to the discontinuities in the power system dynamics equations. A characteristic of power system transient stability problem is its discontinuous nature due to the presence of various time and state-based nonlinearities such as faults, limiters, hysteresis, and others. Many of the generator controllers have some form of limiters that restrict the excursion of a particular state beyond a certain threshold. For instance, the IEEE Type-1 DC exciter model has a wind-up limiter on the voltage regular state, as shown in Fig. 1, that restricts the voltage regulator ouput $V_R$ within its lower and upper limits $V_{Rmin}$ and $V_{Rmax}$, respectively.

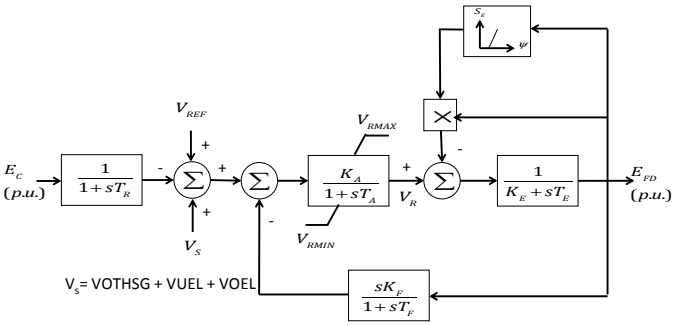


Fig. 1. IEEE type-1 DC exciter model.

Such nonlinearities give rise to the following conditionals[1] introduced in the DAE equations:

$$\begin{cases} x - x^+ = 0, & \text{if } x \geq x^+ \\ x - x^- = 0, & \text{if } x \leq x^- \\ \dot{x} = f(x,y), & \text{otherwise} \end{cases} \tag{4}$$

Detecting and locating such discontinuities is done using an *event* handler or root-finding method. A switching function $h(t,x,y) = 0$ is propagated along with the DAE equations. The time-stepping solver checks for the *zero-crossing* of the event function at every time-step. The zero-crossing of an event is detected by the sign change of the event function, i.e. $h(t_n) \neq h(t_{n+1})$. If this condition is true, the event is said to be detected and the solution rolled back to $t_n$. The mechanism of event detection and location is illustrated in Fig. 2. Using linear interpolation and successively shrinking the time-boundaries, the zero-crossing of the event function is detected when its value is within a specified tolerance. At this time instant, $t_n^*$ in Fig. 2, the discontinuity is applied and an additional step is taken to synchronize with $t_{n+1}$. Further improvements in speeding the detection of event zero crossing can be done using the Illinois algorithm [14] and Anderson–Björck method. [15]. In the case of multiple events detected during the same time-step, the event detection mechanism uses the smallest interpolated time-step from the list of events.

## III. Multi-stage methods

In this section, we present the fundamentals of two classes of multi-stage methods – Runge-Kutta and Rosenbrock. These

[1]Eq. 4 shows one form for illustrative purposes. In general, the conditionals can include functions of the state variables instead of simple box constraints.

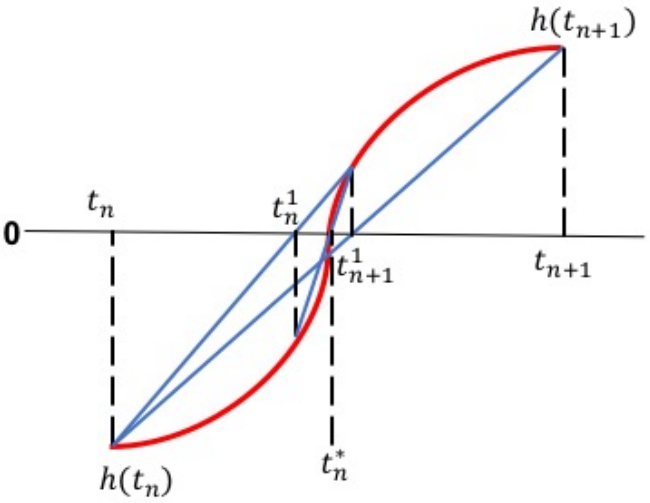


Fig. 2. Detection and location of nonlinearities

are linear time stepping methods that span a wide range of schemes that cover explicit, linearly-implicit, and implicit with various stability properties.

### A. Multi-stage Runge-Kutta methods

A multi-stage Runge-Kutta (RK) method is a linear time stepping scheme that relies on several function evaluations called stages within one time step. An $s$-stage Runge-Kutta method described by the coefficient tableau given in (5).

$$\begin{array}{c|ccccc} c_1 & a_{11} & a_{1,2} & \cdots & a_{1,s-1} & a_{1,s} \\ c_2 & a_{21} & a_{22} & \cdots & a_{2,s-1} & a_{2,s} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ c_{s-1} & a_{s-1,1} & a_{s-1,2} & \cdots & a_{s-1,s-1} & a_{s-1,s} \\ c_s & a_{s1} & a_{s,2} & \cdots & a_{s,s-1} & a_{s,s} \\ \hline & b_1 & b_2 & \cdots & b_{s-1} & b_s \end{array} . \tag{5}$$

Given the solution at time step $n$, the RK process computes the solution at step $n+1$ by

$$\begin{aligned} x_{n+1} &= x_n + \Delta t \sum_{j=1}^{s} b_j f(t_n + c_j \Delta t, X^{(j)}, Y^{(j)}) \\ 0 &= g(t_n + c_i \Delta t, x_{n+1}, y_{n+1}) , \end{aligned} \tag{6}$$

where

$$\begin{aligned} X^{(i)} &= x_n + \Delta t \sum_{j=1}^{s} a_{ij} f(t_n + c_j \Delta t, X^{(j)}, Y^{(j)}) , \\ 0 &= g(t_n + c_i \Delta t, X^{(i)}, Y^{(i)}) \end{aligned} \tag{7}$$

for $i = 1, \ldots, s$.

Lower-order approximations needed in (3) in order to compute $\hat{z}_n$ are obtained by evaluating (6) by using a different set of $b$ coefficients.

It follows that the method is explicit if $a_{i,j} = 0, j \geq i$, diagonal implicit (DIRK) if $a_{i,j} = 0, j > i$ and fully implicit (IRK) if there are terms $a_{i,j} \neq 0, j > i$. The stability as well as the accuracy increase for DIRK and IRK methods; however, so does the computational cost. Implicit methods require solving nonlinear systems of equations at each stage in the case of DIRK and a coupled system of nonlinear equations in the case of IRK. Explicit methods are computationally cheap but not stabile enough and IRK schemes are relatively computationally expensive methods in this study. DIRK methods provide a good tradeoff among accuracy, stability, and computational cost, and therefore, we will only consider DIRK methods in this study. Table I details the DIRK methods considered in this study. In addition, the implicit trapezoidal method is given by

$$\begin{array}{c|cc} 0 & 0 & 0 \\ 1 & \frac{1}{2} & \frac{1}{2} \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array} .$$

TABLE I
MULTI-STAGE RUNGE-KUTTA SCHEMES TESTED IN THIS WORK

| Order | Stages | Stability properties | Variable-step | Reference |
|---|---|---|---|---|
| 2 | 2 | L-Stable | Yes | [16] |
| 3 | 3 | L-Stable | Yes | [17] |
| 4 | 6 | L-Stable | Yes | [17] |

### *B. Multi-stage Rosenbrock*

Rosenbrock methods are linearly-implicit version of the Runge-Kutta methods. They use explicit function evaluations and implicit linear solves, and therefore, they tend to be faster than the implicit Runge-Kutta methods because at each stage only a linear system needs to be solved as opposed to the implicit Runge-Kutta methods that need solving a nonlinear system at each stage. An $s$-stage Rosenbrock method is defined by a coefficient matrices $A = a_{i,j}\,,\ j < i$ and $\Gamma = \gamma_{i,j}\,,\ j \leq i$ and vector $b_i, i = 1, \ldots, s$. The Rosenbrock scheme computes the solution at step $n+1$ by

$$\begin{pmatrix} x_{n+1} \\ y_{n+1} \end{pmatrix} = \begin{pmatrix} x_n \\ y_n \end{pmatrix} + \sum_{j=1}^{s} b_j \begin{pmatrix} \dot{X}^{(j)} \\ \dot{Y}^{(j)} \end{pmatrix} , \tag{8}$$

where

$$\begin{pmatrix} X^{(i)} \\ Y^{(i)} \end{pmatrix} = \begin{pmatrix} x_n \\ y_n \end{pmatrix} + \sum_{j=1}^{i-1} a_{ij} \begin{pmatrix} \dot{X}^{(j)} \\ \dot{Y}^{(j)} \end{pmatrix} , \tag{9}$$

$$\begin{pmatrix} \dot{X}^{(i)} \\ 0 \end{pmatrix} = \Delta t \begin{pmatrix} f(X^{(j)}, Y^{(j)}) \\ g(X^{(j)}, Y^{(j)}) \end{pmatrix} + J \sum_{j=1}^{i-1} \gamma_{ij} \begin{pmatrix} \dot{X}^{(j)} \\ \dot{Y}^{(j)} \end{pmatrix}, \tag{10}$$

$i = 1, \ldots, s$ and

$$J = \Delta t \begin{bmatrix} \frac{\partial f}{\partial x} & \frac{\partial f}{\partial y} \\ \frac{\partial g}{\partial x} & \frac{\partial g}{\partial y} \end{bmatrix} \text{ evaluated at } (x_n, y_n)\,. \tag{11}$$

Here, $J$ is the Jacobian matrix and lower-order approximations are computed in the same way as for RK methods by evaluating (9) with different $b$ coefficients.

These methods can be as stable and accurate as DIRK methods, and therefore, good candidates for efficient methods. The Rosenbrock methods evaluated in this work are given in Table II. The second order method is given by

$$a_{ij} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} \quad \gamma_{ij} = \begin{bmatrix} 1 - \frac{1}{\sqrt{2}} & 0 \\ \frac{\sqrt{2}}{2} - 2 & 1 - \frac{1}{\sqrt{2}} \end{bmatrix} , \tag{12a}$$

$$b = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \end{bmatrix} . \tag{12b}$$

TABLE II
MULTI-STAGE ROSENBROCK SCHEMES TESTED IN THIS WORK

| Order | Stages | Stability properties | Variable-step | Reference |
|---|---|---|---|---|
| 2 | 2 | L-Stable | Yes | (12) |
| 3 | 4 | L-Stable | Yes | [18] Tab. 4.3 |
| 4 | 4 | A-Stable and L-Stable | Yes | [19] Sec. 4 Tab. 7.2 |

## IV. NUMERICAL RESULTS

In this section, we compare the accuracy and performance of the multi-stage numerical integration introduced in section III against the currently existing schemes, i.e. a) fixed-step alternating-explicit modified Euler and, (b) implicit trapezoidal integration scheme. The impact of variable time-stepping and handling of discontinuities is also demonstrated. We also include the performance of a variable-step implicit trapezoidal scheme that uses first-order explicit Euler method to calculate the local truncation error.

The dynamics simulation application code is written in C language using the scientific computing library Portable Extensible Toolkit for Scientific Computation (PETSc) [20], [7], [21]. A brief introduction to the PETSc library is given in Appendix A. All the tests are conducted on 2.8 GHz Intel Core i7 processor with 16 GB 1600 MHz DDR3 RAM. GNU compiler with compiler optimization *-O3* was used.

The comparisons are done on three test networks: (a) IEEE 9-bus system, (b) IEEE-118 bus system, and (c) PEGASE 9241 bus system. The inventory for the test cases is given in Table III.

TABLE III
INVENTORY OF TEST CASES

| Case Name | Buses | Generators | Branches |
|---|---|---|---|
| IEEE 9 | 9 | 3 | 9 |
| IEEE 118 | 118 | 54 | 176 |
| 9241pegase | 9241 | 1445 | 16049 |

A solid three-phase temporary bus fault is applied as the disturbance scenario at a bus starting at 0.1 seconds for six cycles (0.1 seconds). The simulation time-length is set to five (5) seconds with an initial time-step of half-cycle (0.008333 seconds). All generators are modeled with a *dq*

axis generator model with an exciter (IEEET1 model), turbine-governor (TGOV1 model), and a stablizer (STAB1 model). All loads are modeled as constant impedance loads.

The alternating explicit trapezoidal and the fixed-step implicit method uses a fixed step of 0.008333 (half-cycle at 60Hz frequency) seconds. The variable time-stepping schemes adapt their time-step based on a local truncation error calculation. The min. and max. time-steps allowed for the variable time-stepping methods is 0.008333 and 0.04 seconds. The step-size is restricted to a max. value to avoid (a) prohibitively large steps causing divergence of the solver, and (b) missing discontinuities, such as exciter limits. A relative and absolute tolerance of 1e-2 is used for controlling the error calculation given in Eq. 3. Discontinuities in the DAE equations are detected and located using an event handling mechanism as described in Section II-B.

### A. 9-bus system

Figures 3 and 4 show the frequency for the generator having the maximum frequency deviation, and the voltage magnitude for the bus having the maximum voltage deviation. Following the fault at bus 9 at 0.1 seconds, the generator at bus 3, close to the fault location, experiences an imbalance in its mechanical energy input and electrical output as a result of which it starts oscillating. These oscillations are damped after the fault is cleared indicating that the system is stable. The voltage magnitude at bus 9 experiences a temporary collapse as a result of the short-circuit at 0.1 seconds, but quickly recovers to its initial voltage level after the fault is cleared. As seen from figures 3 and 4, all the time-stepping methods have the same accuracy as the explicit alternating trapezoidal method.

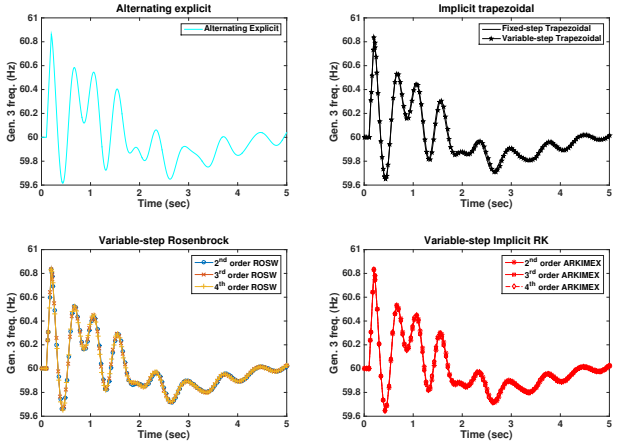


Fig. 3. Comparison of frequency for the generator with largest frequency deviation for the 9-bus system

Figure 5 displays the time-steps taken by each of the numerical integration schemes tested. The alternating explicit trapezoidal and the fixed-step implicit methods always use a fixed step of 0.008333 seconds. The variable time-stepping schemes adapt their time-step based on a local truncation error

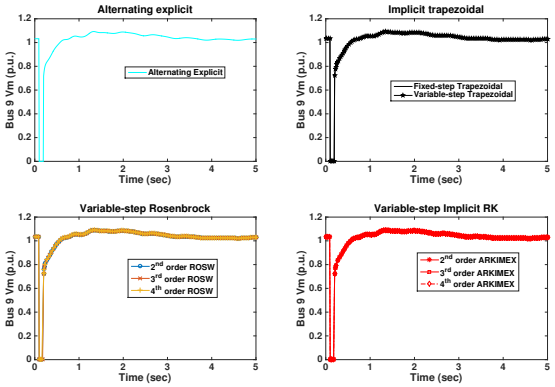


Fig. 4. Comparison of voltage magnitude for the bus with largest voltage deviation for the 9-bus system

calculation. During the pre-fault stage, all the variable time-stepping schemes use a time-step of 0.04 seconds, the max. step-size allowed. During the fault-on and the initial post-fault period, a state of fast varying dynamics, smaller time-steps are taken due a higher local truncation error. Beyond this period, the dynamics become smooth again leading to larger steps being taken by the variable time-stepping schemes. As seen in Fig. 5, the variable-step trapezoidal method takes much more smaller time-steps, as compared to RK and Rosenbrock methods, before finally taking the max. step size of 0.04 seconds.

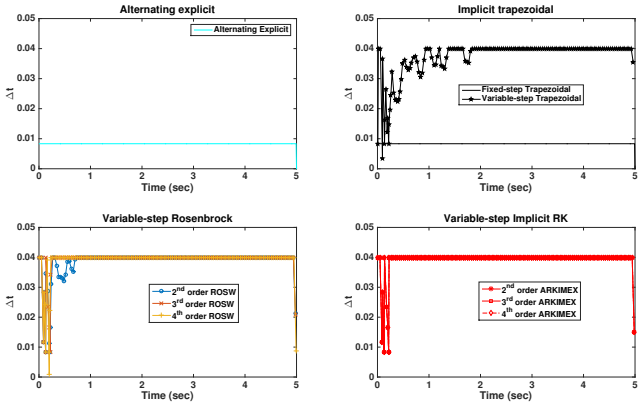


Fig. 5. Time-steps taken by different numerical integration schemes for the 9-bus system

Table IV shows the performance of the various time-stepping methods tested comparing their execution time and the number of steps. The $3^rd$ order Rosenbrock scheme has the fastest execution time with a speed-up of approximately 5X over the alternating explicit trapezoidal scheme and 8X over the implicit trapezoidal scheme. The variable-step implicit Runge-Kutta methods take the least number of steps. Yet, their execution time is slower as they involve more stages with nonlinear solves (using Newton's method). While only a linear solve needs to be performed with the Rosenbrock methods as they are linearly implicit.

TABLE IV
EXECUTION TIME AND STEPS COMPARISON FOR 9-BUS TEST CASE

| Integration scheme | Execution time (sec) | Number of steps |
|---|---|---|
| Alternating explicit | 3.00 | 601 |
| Fixed-step Implicit trapezoidal | 4.83 | 601 |
| Variable-step Implicit trapezoidal | 0.67 | 141 |
| Variable-step Rosenbrock (2nd order) | 0.65 | 134 |
| Variable-step Rosenbrock (3rd order) | 0.64 | 132 |
| Variable-step Rosenbrock (4th order) | 0.66 | 133 |
| Variable-step Implicit RK (2nd order) | 0.68 | 129 |
| Variable-step Implicit RK (3rd order) | 2.73 | 129 |
| Variable-step Implicit RK (4th order) | 2.89 | 129 |

### *B. 9-bus system with automatic voltage regulator limits enforced*

In this experiment, we restricted the voltage regulator output of all generators to $0 \leq V_R \leq 3.0$. The dynamics of voltage regulator output $V_R$ is shown in Fig. 6. Following the fault, the voltage regulator for the exciter on generator 3 hits its limit before recovering and then again hitting its limit after the fault clearing. Since this voltage regulator output cannot be increased, it stays constant at 3.0 throughout the simulation time horizon. The exciter at generator 2 follows suit resulting in its $V_R$ being limited. As a result of this $V_R$ limit enforcement, the generator terminal voltage magnitude is unable to be regulated resulting in an unstable condition as shown in Figs. 7 and 8.

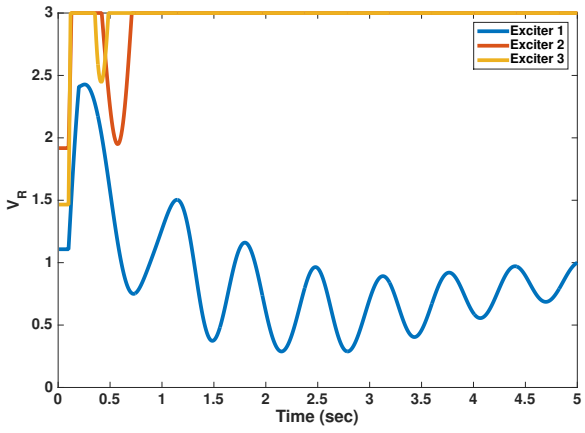


Fig. 6. Voltage regulator output with limits enforced

As seen in Fig. 9, the 2nd order ROSW scheme uses smaller time-steps as the instability increases. The higher order Rosenbrock and the Runge-Kutta schemes all take larger time-steps. The best time obtained for this unstable case is for the 3rd order Rosenbrock scheme with approximately 4.6X and 4.7X speedup over alternating explicit and implicit trapezoidal, respectively.

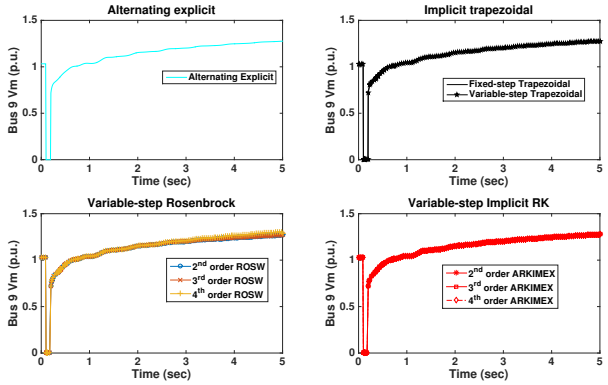


Fig. 7. Comparison of voltage magnitude for the bus with largest voltage deviation for unstable 9-bus case

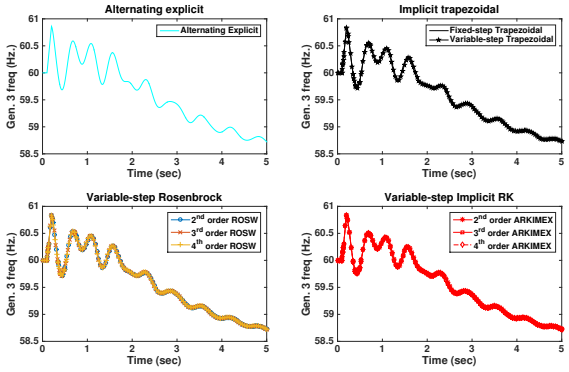


Fig. 8. Comparison of frequency for the generator with largest frequency deviation for unstable 9-bus case

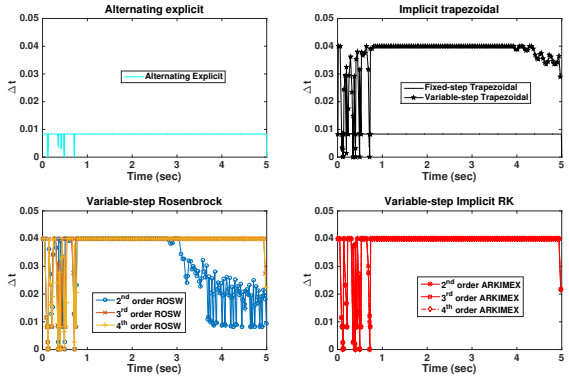


Fig. 9. Time-steps taken by different numerical integration schemes for the unstable 9-bus case

### *C. 118-bus system*

Figures 10 - 12 show the frequency, voltage magnitude, and time-step comparison for the different multi-stage methods tested. The dynamics for this scenario is much more smoother with frequency excursion of less than 0.15 Hz. The voltage magnitude recovery is also much faster. This results in the variable time-stepping multi-stage methods, and even the variable-step implicit trapezoidal method, are able to take the max. step size allowed of 0.04 seconds.

TABLE V
EXECUTION TIME AND STEPS COMPARISON FOR 9-BUS TEST CASE WITH LIMITS ENFORCED

| Integration scheme | Execution time (sec) | Number of steps |
|---|---|---|
| Alternating explicit | 5.75 | 620 |
| Fixed-step Implicit trapezoidal | 5.91 | 625 |
| Variable-step Implicit trapezoidal | 1.43 | 173 |
| Variable-step Rosenbrock (2nd order) | 2.11 | 226 |
| Variable-step Rosenbrock (3rd order) | 1.26 | 159 |
| Variable-step Rosenbrock (4th order) | 1.48 | 168 |
| Variable-step Implicit RK (2nd order) | 1.79 | 148 |
| Variable-step Implicit RK (3rd order) | 1.93 | 147 |
| Variable-step Implicit RK (4th order) | 1.84 | 148 |

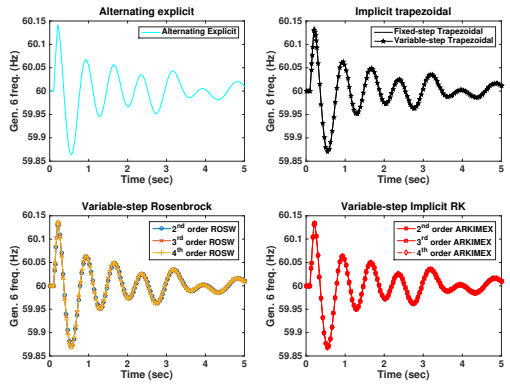


Fig. 10. Comparison of frequency for the generator with largest frequency deviation for the 118-bus case

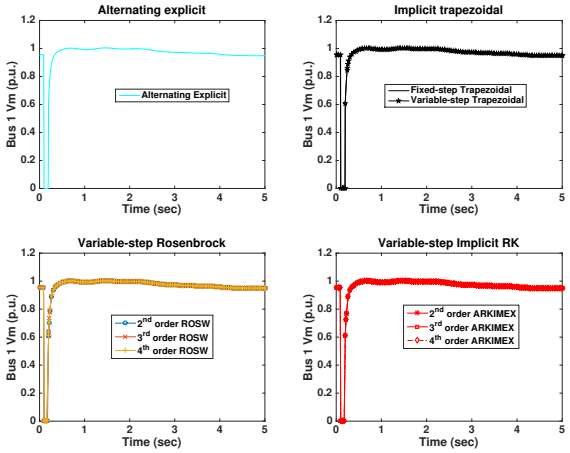


Fig. 11. Comparison of voltage magnitude for the bus with largest voltage deviation for the 118-bus case

Figure 12 displays the time-steps taken by each of the numerical integration schemes tested. The alternating explicit trapezoidal and the fixed-step implicit methods always use a fixed step of 0.008333 seconds. The variable time-stepping schemes adapt their time-step based on a local truncation error calculation. During the pre-fault stage, all the variable time-stepping schemes use a time-step of 0.04 seconds, the max. step-size allowed. The step-size is restricted to a max. value to avoid (a) prohibitively large steps causing divergence of the solver, and (b) missing discontinuities, such as exciter limits. During the fault-on and the initial post-fault period, a state of fast varying dynamics, smaller time-steps are taken due a higher local truncation error. Beyond this period, the dynamics become smooth again leading to larger steps being taken by the variable time-stepping schemes.

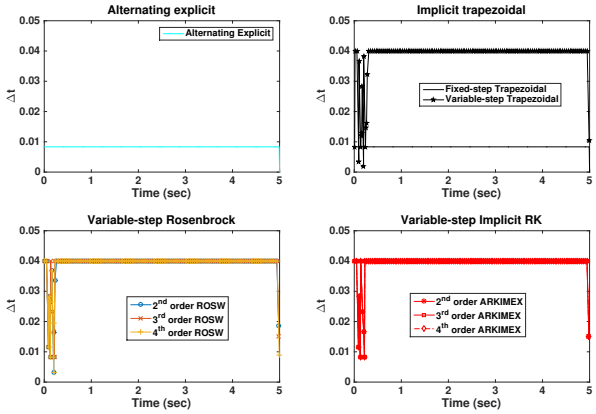


Fig. 12. Time-steps taken by different numerical integration schemes for the 118-bus case

Table VI shows the performance of the various time-stepping methods tested comparing their execution time and the number of steps. The $2^{nd}$ order Rosenbrock scheme has the fastest execution time with a speed-up of approximately 4.5X over the alternating explicit trapezoidal scheme. The variable-step implicit Runge-Kutta methods take the least number of steps. Yet, their execution time is slower as during each stage a nonlinear solve (using Newton's method) is performed. While only a linear solve needs to be performed with the Rosenbrock methods as they are linearly implicit.

### *D. 9241PEGASE test case*

For the 9241 PEGASE test case, a similar set up as the 118-bus test system is created. The dynamics of the frequency and voltage can be seen in Figures 13 – 14. Again for this case, the Rosenbrock methods perform the best with good accuracy and speedup of approximately 6X compared to alternating explicit scheme.

## V. CONCLUSIONS

In this paper, we presented a thorough evaluation of variable-step multi-stage methods for accelerating power system dynamics simulation. Two different classes of numerical integration methods; Variable-step Rosenbrock, and Variable-step Runge-Kutta method; were introduced and their accuracy and performance examined on the three test systems ranging from 9 to 9,000 buses. Impacts of variable time-step and limit

TABLE VI
EXECUTION TIME AND STEPS COMPARISON FOR 118-BUS TEST CASE

| Integration scheme | Execution time (sec) | Number of steps |
|---|---|---|
| Alternating explicit | 6.22 | 601 |
| Fixed-step Implicit trapezoidal | 10.54 | 601 |
| Variable-step Implicit trapezoidal | 2.52 | 136 |
| Variable-step Rosenbrock (2nd order) | 1.63 | 133 |
| Variable-step Rosenbrock (3rd order) | 1.79 | 132 |
| Variable-step Rosenbrock (4th order) | 1.71 | 133 |
| Variable-step Implicit RK (2nd order) | 3.04 | 129 |
| Variable-step Implicit RK (3rd order) | 3.61 | 129 |
| Variable-step Implicit RK (4th order) | 5.23 | 129 |

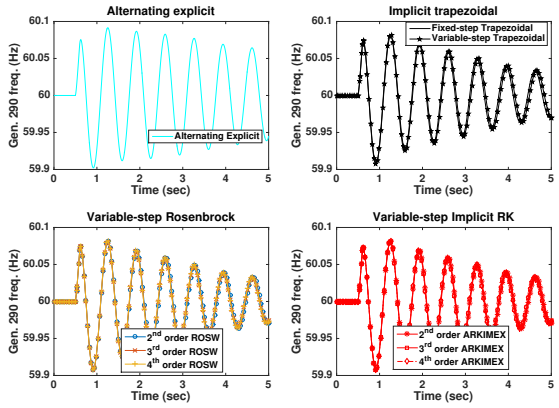


Fig. 13. Comparison of frequency for the generator with largest frequency deviation for the PEGASE 9241 bus case

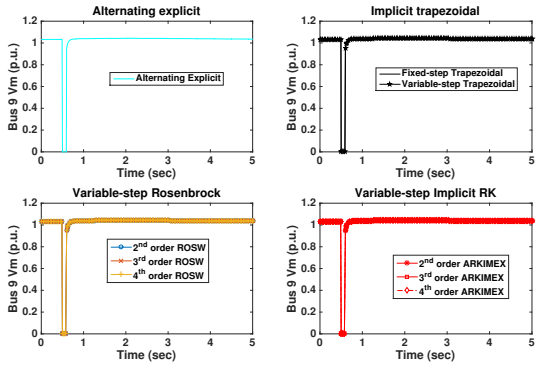


Fig. 14. Comparison of bus voltage magnitude with largest voltage deviation for the PEGASE 9241 bus case

handling were presented. Out of all the numerical integration schemes tested, Rosenbrock methods were found to be most optimal for dynamics simulation providing good accuracy and performance.

TABLE VII
EXECUTION TIME AND TIME-STEPS COMPARISON FOR 9241 PEGASE TEST CASE

| Integration scheme | Execution time (sec) | Number of steps |
|---|---|---|
| Alternating explicit | 41.37 | 601 |
| Fixed-step Implicit trapezoidal | 47.31 | 601 |
| Variable-step Implicit trapezoidal | 11.01 | 133 |
| Variable-step Rosenbrock (2nd order) | 7.19 | 132 |
| Variable-step Rosenbrock (3rd order) | 9.20 | 132 |
| Variable-step Rosenbrock (4th order) | 9.48 | 132 |
| Variable-step Implicit RK (2nd order) | 19.81 | 129 |
| Variable-step Implicit RK (3rd order) | 29.10 | 129 |
| Variable-step Implicit RK (4th order) | 46.96 | 129 |

## ACKNOWLEDGMENTS

This work was supported by U.S. Dept. of Energy, Office of Science, Advanced Scientific Computing Research, under Contract DE-AC02-06CH11357.

## APPENDIX A
## PORTABLE EXTENSIBLE TOOLKIT FOR SCIENTIFIC COMPUTING (PETSC)

PETSc is an open source package (BSD-style license) for the numerical solution of large-scale applications and provides the building blocks for the implementation of large-scale application codes on parallel (and serial) computers. PETSc utilizes the Message Passing Interface (MPI) standard for all message passing communication. It is portable to different architectures (loosely/tightly coupled clusters, many-core/multi-core machines, GPGPUs), operating systems (Unix, Linux, Macintosh, Windows), precision (single, double), and scalar types (real, complex), and the application code can be written in C, C++, Fortran, or Python. The PETSc package consists of a set of libraries for creating parallel vectors, matrices, and distributed arrays and scalable linear, nonlinear, and time-stepping solvers. The organization of PETSc is shown in Figure 15.

The wide range of sequential and parallel linear solvers, preconditioners, reordering strategies, flexible run-time options, ease of code implementation, debugging options, and a comprehensive source code profiler makes PETSc an attractive experimentation platform for developing parallel power system dynamics analysis applications.

We have implemented time-stepping solvers in PETSc providing support for both implicit and explicit time-stepping schemes of various orders (2-5), including support for variable time-stepping and handling discontinuities. A comprehensive list of the time-stepping solvers available with PETSc can be found in [7].

## REFERENCES

[1] P. Sauer and M. A. Pai, *Power System Dynamics and Stability*. Prentice Hall Inc., 1998.

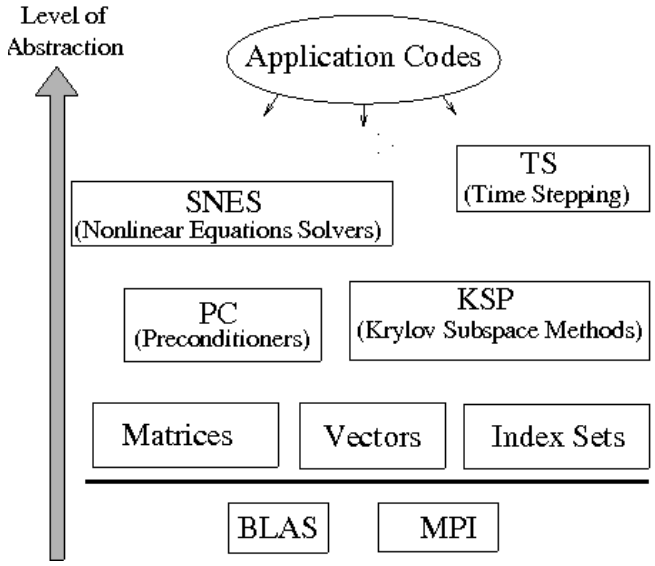


Fig. 15. Organization of the PETSc library [7]


[2] P. Kundur, J. Paserba, V. Ajjarapu, G. Andersson, A. Bose, C. Canizares, N. Hatziargyriou, D. Hill, A. Stankovic, C. Taylor, T. V. Cutsem, and V. Vittal, "Definition and classification of power system stability ieee/cigre joint task force on stability terms and definitions," *IEEE Transactions on Power Systems*, vol. 19, no. 3, pp. 1387–1401, Aug 2004.

[3] J. H. Eto and R. J. Thomas, "Computational needs for the next generation electric grid," U.S. Department of Energy, Tech. Rep., 2011, http://energy.gov/sites/prod/files/FINAL_CompNeeds_Proceedings2011.pdf.

[4] E. P. R. Institute, "Grid transformation workshop results," Electric Power Research Institute, Tech. Rep., 2012, http://my.epri.com/portal/server.pt?space=CommunityPage&cached=true&parentname=ObjMgr&parentid=2&control=SetCommunity&CommunityID=404&RaiseDocID=000000000001025087&RaiseDocType=Abstract_id.

[5] Z. Huang and J. Nieplocha, "Transforming power grid operations via high performance computing," in *Proceedings of the IEEE Power and Energy Society General Meeting 2008*, 2008.

[6] C. Josz, S. Fliscounakis, J. Maeght, and P. Panciatici, "AC power flow data in MATPOWER and QCQP format: iTesla, RTE snapshots, and PEGASE," 2016.

[7] S. Balay, S. Abhyankar, M. F. Adams, J. Brown, P. Brune, K. Buschelman, L. Dalcin, V. Eijkhout, W. D. Gropp, D. Kaushik, M. G. Knepley, L. C. McInnes, K. Rupp, B. F. Smith, S. Zampini, H. Zhang, and H. Zhang, "PETSc users manual," Argonne National Laboratory, Tech. Rep. ANL-95/11 - Revision 3.7, 2016. [Online]. Available: http://www.mcs.anl.gov/petsc

[8] S. Abhyankar and A. Flueck, "Real-time power system dynamics simulation using a parallel block-jacobi preconditioned Newton-GMRES scheme," in *Proceedings of the $2^{nd}$ International Workshop on High Performance Computing, Networking and Analytics for the Power Grid*. IEEE, 2012.

[9] S. Abhyankar, E. Constantinescu, B. Smith, A. Flueck, and D. Maldonado, "Parallel dynamics simulation using a krylov-schwarz linear solution scheme," *IEEE Transactions on Smart Grid*, vol. 8, pp. 1378–1386, 2017.

[10] *PSSE 34.1 - Program Operational Manual*, SIEMENS.

[11] *PowerWorld Users Guide*, PowerWorld Corporation.

[12] J. J. Sanchez-Gasca, R. D'Aquila, W. W. Price, and J. J. Paserba, "Variable time-step, implicit integration for extended-term power system dynamic simulation," in *IEEE Conference Proceedings of Power Industry Computer Application Conference*. IEEE, 1995.

[13] Powertech, "Power TSAT Web page," http://www.powertechlabs.com/areas-of-focus/software-technologies-/dsatools-software/transient-security-assessment-tool/, 2017. [Online]. Available: http://www.powertechlabs.com/areas-of-focus/software-technologies-/dsatools-software/transient-security-assessment-tool/

[14] M. Dowell and P. Jarratt, "A modified regular falsi method for computing the root of an equation," *BIT*, vol. 11, pp. 168–174, 1971.

[15] S. Galdino, "A family of regular falsi root-finding methods," in *Proceedings of the 2011 World Congress on Engineering and Technology*, 2011.

[16] F. Giraldo, J. Kelly, and E. Constantinescu, "Implicit-explicit formulations of a three-dimensional nonhydrostatic unified model of the atmosphere (NUMA)," *SIAM Journal on Scientific Computing*, vol. 35, no. 5, pp. B1162–B1194, 2013.

[17] C. Kennedy and M. Carpenter, "Additive Runge-Kutta schemes for convection-diffusion-reaction equations," *Appl. Numer. Math.*, vol. 44, no. 1-2, pp. 139–181, 2003.

[18] J. Rang and L. Angermann, "New Rosenbrock W-methods of order 3 for partial differential algebraic equations of index 1," *BIT Numerical Mathematics*, vol. 45, no. 4, pp. 761–787, 2005.

[19] E. Hairer and G. Wanner, *Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems*. Springer, 2002.

[20] S. Balay, S. Abhyankar, M. F. Adams, J. Brown, P. Brune, K. Buschelman, L. Dalcin, V. Eijkhout, W. D. Gropp, D. Kaushik, M. G. Knepley, L. C. McInnes, K. Rupp, B. F. Smith, S. Zampini, and H. Zhang, "PETSc Web page," http://www.mcs.anl.gov/petsc, 2015. [Online]. Available: http://www.mcs.anl.gov/petsc

[21] S. Abhyankar, B. Smith, H. Zhang, and A. Flueck, "Using PETSc to develop scalable applications for next-generation power grid," in *Proceedings of the 1st International Workshop on High Performance Computing, Networking and Analytics for the Power Grid*. ACM, 2011. [Online]. Available: http://www.mcs.anl.gov/uploads/cels/papers/P1957-0911.pdf


**Shrirang Abhyankar** received his M.S degree (2006) and his Ph.D. degree (2011) in electrical engineering from Illinois Institute of Technology, Chicago. He is currently a computational engineer in the Energy Sciences Division at Argonne National Laboratory. His research interests include scalable algorithms for large-scale transient stability analysis, combined electromechanical and electromagnetic simulation, and co-simulation of transmission-distribution dynamics. He is a co-developer of the PETSc library.

**Emil M. Constantinescu** received his Ph.D. degree in computer science from Virginia Tech, Blacksburg, in 2008. He is currently a computational mathematician in the Mathematics and Computer Science Division at Argonne National Laboratory. His research interests include numerical analysis of time-stepping algorithms and their applications to energy systems.

**Alexander J. Flueck** received his B.S. degree (1991), M.Eng. degree (1992) and Ph.D. degree (1996) in electrical engineering from Cornell University. He is currently an associate professor at the Illinois Institute of Technology in Chicago. His research interests include three-phase transient stability, electromagnetic transient simulation, autonomous agent applications to power systems, transfer capability of large-scale electric power systems, contingency screening with respect to voltage collapse, and parallel simulation of power systems via message-passing techniques on distributed-memory computer clusters.