



## Parallel Solving of The Wind System Control at Variable Wind Speed

### KEYWORDS

parallel computing, parallel structure, serial computing, batch job, wind energy, movement equation of the wind system, mechanical angular speed

### Florica Balog

Polytechnic University of Timisoara/ Department of Computer Science/ Timisoara, Romania

### Horia Ciocarlie

Polytechnic University of Timisoara/ Department of Computer Science/ Timisoara, Romania

**ABSTRACT** *The main idea of this paper is reducing of the elapsed time required for the serial solving of a complex wind-energy problem, by using the parallel computing. This paper shows both the serial and parallel solving of a wind-energy problem. The serial solving of this problem involves the development of a Matlab program and running it on the local Matlab client machine. On the other hand, the parallel solving refers to running the Matlab program in parallel on the local cluster of Matlab workers, by using a proposed parallel structure.*

### I. INTRODUCTION

As a renewable energy source, the wind is the "raw material" for all type of the wind systems.

Due to the fact that the wind speed's behavior is random [19], the wind power will take different values (the power being a function of wind).

The wind-energy systems are characterized by irregularity and unpredictability [19], due to the time-variation of the wind speed, such as the real case of Romania.

Due to the time-variation of the wind speed, the optimum mechanical angular speed takes different values, thus involving a corresponding control of the wind system in order to achieve the maximum wind energy.

This has been the main idea of the articles [8] and [9], which treat the wind-energy problem of this paper.

Thus, the wind-energy problem refers to achieving the time evolution of mechanical angular speed that is given by the motion equation of both the wind turbine and the permanent magnet synchronous generator in (2) [8].

Equation (2) was initially solved by using the Scientific WorkPlace software in [9]. Here, the authors have had a problem related to the long time to solving this complex equation (by the order of minutes).

Therefore, in order to reduce the elapsed time required to solve this complex equation, the authors have been implemented this time-consuming problem by using the Matlab language (the version R2015a).

In this sense, this paper presents the serial solution of the wind problem, and, on the other hand, the parallel solution has been implemented by using two different methods.

In order to highlight the time gained from solving the ordinary differential equation (2), by using the two parallel methods, the authors have been measured both serial and parallel execution time.

In order to obtain maximum wind energy from the variable wind, the wind system must be optimally controlled [6], [7].

In this sense, it is necessary to control the mechanical angular speed by imposing the reference speed on the interval  $\Delta t$ , and therefore the available maximum wind energy is extracted at the wind speed  $V(t)$  [1-5].

Due to the time variation of the wind speed, the optimum mechanical angular speed takes different values, thus involving a corresponding adjustment of the generator load [6].

The time evolution of mechanical angular speed is given by the motion equation of both wind turbine and permanent magnet synchronous generator [8], [9], in the form (1) or (2). Equations (1) and (2) [9] are obtained based on the results from [8] and [9].

$$40 \cdot \frac{d\omega}{dt} = M_{T\omega} - M_G \quad (1)$$

$$40 \cdot \omega \cdot \frac{d\omega}{dt} = 23.584 \cdot \left( (16 - 6 \cdot \sin(0.17943t)) \cdot e^{-t/3600} \right) / \omega - 2.1231 \cdot 10^{-2} \cdot e^{-98.04(16 - 6 \cdot \sin(0.17943t)) \cdot e^{-t/3600}} \omega^{-0.0525} \cdot \left( (16 - 6 \cdot \sin(0.17943t)) \cdot e^{-t/3600} \right)^3 - 5445 \cdot \omega^2 \cdot (5 \cdot 464.55 + 8) \cdot \frac{4 \cdot \omega^2 + 625 \cdot 464.55^2 + 2000 \cdot 464.55 + 1600}{(1250 \cdot 464.55^2 + 4000 \cdot 464.55 + 3200 + 7 \cdot \omega^2)^2}$$

$$\omega(0) = 552.95 \quad (2)$$

In (1),  $M_{T\omega}$  represents the moment of the wind turbine,  $M_G$  is the moment of the permanent magnet synchronous generator, and  $\omega$  is the mechanical angular speed. The load resistance of the generator is  $R(0)=464.55\Omega$ .

In fact, the motion equation (2) is an ordinary differential equation that has the initial values:  $R(0)=464.55\Omega$  and  $\omega(0)=552.95[\text{rad/s}]$ . For this reason, the authors called it an initial value problem.

### II. SERIAL SOLUTION OF THE ORDINARY DIFFERENTIAL EQUATION

This paper shows the problem solution by using the classic four-stage Runge-Kutta method and by using the MATLAB ode23 code, as well.

The serial solution has been achieved in order to compare

the obtained results with those from parallel solving, and to highlight the performance gain, as well.

Therefore, the serial solution is a typical implementation of a Matlab program, which runs the Matlab program called the\_main\_code.m on the local Matlab client machine.

In order to compute the serial solution of the initial value problem, the authors have been developed the Matlab script the\_main\_code.m that implements both the ode23 and the ode45 solvers (which solve the ordinary differential equations (2)). On the other hand, the authors have been developed a handle function Ode\_function.m that implements the ordinary differential equation (2).

The program the\_main\_code.m applies the Runge-Kutta method of both the order 2 and 4 to the differential equation (2) defined in the M-file Ode\_function.m. Finally, the program the\_main\_code.m calls the function handle Ode\_function.m in order to solve this initial value problem. Further, it will be describe the implementation of the Runge-Kutta method of the order 2.

In this case, ode23 is the integration method [10] for the ordinary differential equation (2) using second and third order Runge-Kutta-Fehlberg formulas with automatic and variable step-size.

The function ode23 computes the numerical approximation [10] to the solution of the ordinary differential equation (2), and then plot the numerical solution, as shown in Figure 1.

The first argument of the ode23 solver of the Matlab program the\_main\_code.m is the function handle Ode\_function.m that implements the differential equation (2). This function handle accepts two arguments, t, and omega. The result of evaluating the function Ode\_function.m is a column vector, DomegaDt, containing the values of the derivatives, domega/dt.

The second input argument of the ode23 solver is a vector that has two components: t0=0[s], and tfinal=35[s]. Thus, the integration is carried out over the interval  $0 \leq t \leq 35$ .

The initial condition ( $\omega_0=552.95[\Omega]$ ) of the ordinary differential equations (2) is the third-argument of ode23.

In the case of this initial value problem, the solutions achieved by using ode23 solver (red color), is almost the same as that achieved by using ode45 solver (blue color), as shown in Figure 1.

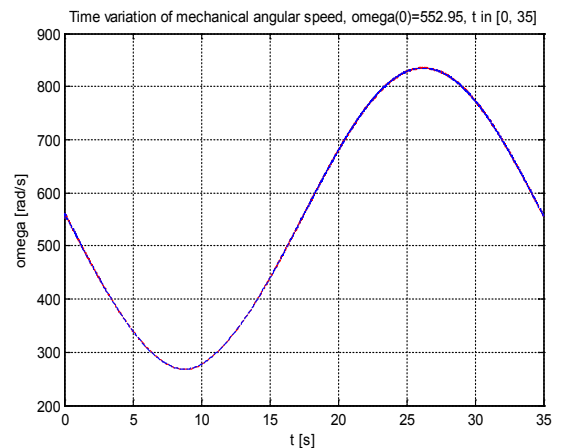
The total time spent in the computation (the elapsed time), in order to achieve the serial solution of the wind-energy problem is 1.388989 seconds, as shown in the listing below:

```
>> the_main_code
```

```
Serial solution:
```

```
Elapsed time is 1.388989 seconds.
```

This elapsed time of the serial solution has been achieved by running the program the\_main\_code.m on the Matlab client, without using a parallel pool of Matlab workers. The obtained solution is shown graphically in Figure 1.



**Figure 1: time variation of mechanical angular speed (serial solution)**

### III. PARALLEL SOLUTION OF THE ORDINARY DIFFERENTIAL EQUATION

Parallel computing can save time and money. Also, it can solve larger problems, provide concurrency, use the non-local resources and limit the serial computation [13].

Users develop parallel computing algorithms either reduce computation time or undertake the analysis of larger data sets [14].

If the time spent in computation is less than the time spent in communication between workers or in network traffic in order to transfer data, than the parallel computing is useless.

Due to the possible wide parameter sweeps required for analysis, the analysis with smaller data sets can also be impractical with respect to time [14].

The above arguments are just several reasons why parallel computing should be used in order to solve large applications.

The parallel solution of the wind-energy problem was achieved by using the Parallel Computing Toolbox of Matlab 2015a.

Parallel Computing Toolbox allows solving computationally and data-intensive problems by using multicore processors, GPUs and computer clusters [12]. In order to parallelize (without CUDA or MPI programming) this wind-energy problem, the authors have been used the multicore processors capability of the Parallel Computing Toolbox.

Furthermore, Parallel Computing Toolbox provides a local cluster of workers for the client machine [12].

In order to parallelize the wind application, the authors have been used a supercomputer having a quad core i7 processor. For this reason, the program the\_main\_code.m, which calls the function Ode\_function.m that implements the ordinary differential equation (2), has been run on a parallel pool (consisted of four Matlab workers) at the same time.

Parallelizing the problem by using the quad-parallel

structure

In this case, in order to solve the program the\_main\_code.m (which implements the wind-energy problem) in parallel, the authors have been proposed a parallel structure that consist of four workers grouped into a parallel pool, and which the authors have been called it quad-parallel structure.

Thus, this pool is a set of four MATLAB workers in the „local“ cluster on which the pool runs, as shown in Figure 2. This parallel pool can be accessible (as a single entity) at a time from the MATLAB client session.

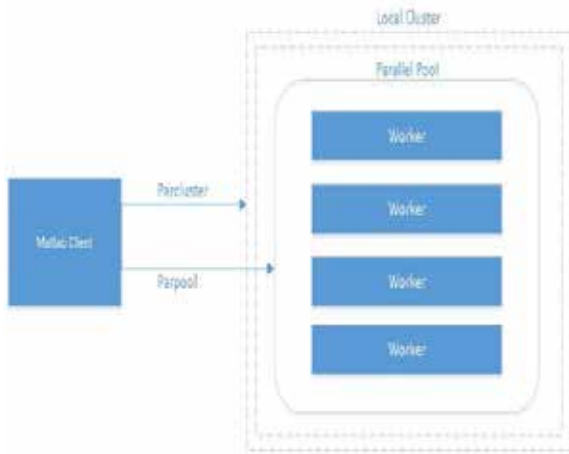


Figure 2: quad-parallel structure

Figure 2 shows the quad-structure of the parallel pool that has been used in order to solve the wind application in parallel, so as to reduce the solving time.

The parallel pool that has been used in order to solve the program the\_main\_code.m, which implements the wind-energy problem, consist of four workers, and it has been started by using the local profile. Thus, a parallel session

has been opened, in which the program the\_main\_code.m has been solved by using the four workers of the pool.

The result achieved from the parallel solving, by using the quad-parallel structure, shows that the elapsed time of the parallel solution is 0.651275 seconds, as shown in the listing below:

Starting parallel pool (parpool) using the 'local' profile ... connected to 4 workers.

```
>> the_main_code
```

Parallel solution:

Elapsed time is 0.651275 seconds.

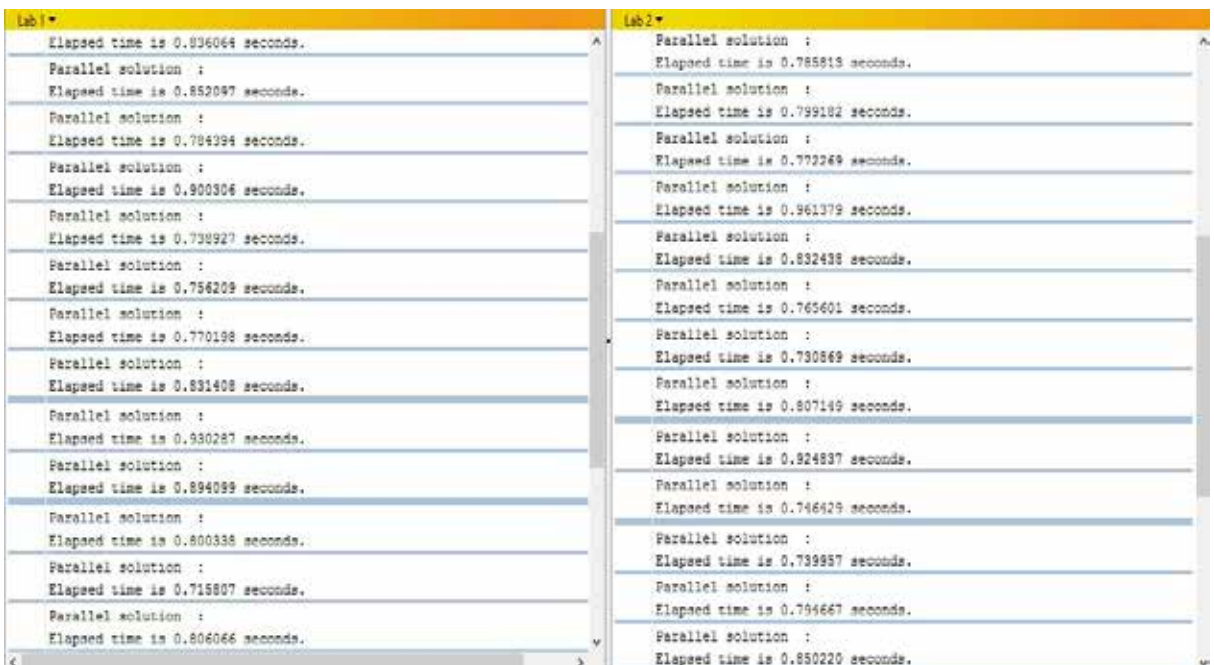
By comparing the elapsed time of the parallel solution (0.651275 seconds), using the quad-parallel structure, with the elapsed time achieved from serial solution (1.388989 seconds), it can be observed the achieved performance gain that is 0,737714 seconds (which means more than 50% less time to solve the wind-energy problem).

Parallelizing the problem by using interactive parallel mode

In this case, our problem was solved by using the interactive parallel mode available in the MATLAB client, thus allowing the interactive parallel execution of the MATLAB program the\_main\_code.m by using the same quad-parallel structure.

Thus, parallelizing the wind-energy problem has been achieved by defining and submitting a communicating job, and opening the Parallel Command Window (shown in Figure 3) connected to those four workers (which belong to the quad-parallel structure) running the submitted job.

Therefore, a local scheduler has been used and those four workers have been run on the local



Lab 1	Lab 2
Parallel solution : Elapsed time is 0.732432 seconds.	Parallel solution : Elapsed time is 0.730115 seconds.
Parallel solution : Elapsed time is 0.703087 seconds.	Parallel solution : Elapsed time is 0.734903 seconds.
Parallel solution : Elapsed time is 0.828920 seconds.	Parallel solution : Elapsed time is 0.723941 seconds.
Parallel solution : Elapsed time is 0.725151 seconds.	Parallel solution : Elapsed time is 0.730889 seconds.
Parallel solution : Elapsed time is 0.784011 seconds.	Parallel solution : Elapsed time is 0.794201 seconds.
Parallel solution : Elapsed time is 0.717480 seconds.	Parallel solution : Elapsed time is 0.763240 seconds.
Parallel solution : Elapsed time is 0.720923 seconds.	Parallel solution : Elapsed time is 0.803720 seconds.
Parallel solution : Elapsed time is 0.724379 seconds.	Parallel solution : Elapsed time is 0.758400 seconds.
Parallel solution : Elapsed time is 0.689175 seconds.	Parallel solution : Elapsed time is 0.670731 seconds.
Parallel solution : Elapsed time is 0.919268 seconds.	Parallel solution : Elapsed time is 0.729989 seconds.
Parallel solution : Elapsed time is 0.738208 seconds.	Parallel solution : Elapsed time is 0.701128 seconds.
Parallel solution : Elapsed time is 0.730560 seconds.	Parallel solution : Elapsed time is 0.790056 seconds.
Parallel solution : Elapsed time is 0.735129 seconds.	Parallel solution : Elapsed time is 0.675348 seconds.

Figure 3: parallel command window

MATLAB client machine. The parallel mode has been started by using the 'local' profile connected to the four workers.

Basically, four local workers have been started, and a communicating job has been created in order to be run simultaneously on the workers. Each worker has run the script the\_main\_code.m on its own workspace and on its own variables. Each of the four workers becomes idle when it completes the work, waiting until all the workers that work on the job have completed the same work.

The parallel command window (Figure 3) shows the elapsed times of the parallel solutions, on each worker (lab), after each interactive parallel execution of the program the\_main\_code.m.

In order to demonstrate the effectiveness of the quad-parallel structure for this interactive parallel mode of the wind-energy problem solving, the authors have been achieved several elapsed times (which have been obtained from several parallel solving), thus, all labs contain several elapsed times (one for each parallel solving).

The authors have been found that each worker has different elapsed time for the same parallel solving (they have been used all four workers simultaneously at each parallel solving), and all the elapsed times (from each lab) are less than the elapsed time of the serial solution of our application, as well.

#### IV. CONCLUSIONS

The authors found that, even in the case of the serial solving, it was achieved a serial time less than in the case of the problem solving by using the Scientific WorkPlace software. Therefore, the problem solving has been successfully implemented by using the Matlab language.

For this application, the authors have been proved that the proposed quad-structure was optimally for both the parallel pool mode and the interactive parallel mode, as well.

The elapsed time required to solve the wind-energy problem, by using the quad-parallel structure, in a parallel pool has been reduced considerably (0.651275 seconds), by comparing this time with the elapsed time of serial solving (1.388989 seconds).

On the other hand, the elapsed times obtained from several interactive parallel running that have been achieved by using the quad-parallel structure, in the interactive parallel mode, have values close to 0.651275 seconds, as shown in Figure 3, and this thing is a performance gain, as in the previous case.

Thus, the quad-parallel structure is useful for developing more complicated parallel code, in the sense that the computing time can be reduced. Therefore, this parallel structure can also be used for any other complex applications requiring parallel solving in order to reduce the solving time.

## REFERENCE

- [1] Babescu M., Borlea I., Jigoria Oprea D., "Fundamental aspects concerning Wind Power System Operation Part.1, Case Study", IEEE MELECON 2012, Medina Tunisia, 25-28 March, 2012, 978-1-4673-0783-3. | [2] Babescu M., Borlea I., Jigoria Oprea D., "Fundamental aspects concerning Wind Power System Operation Part.2, Mathematical Models", IEEE MELECON 2012, Medina Tunisia, 25-28 March, 2012, 978-1-4673-0783-3. | [3] M. Babescu, O.Gana, L.Clotea", "Fundamental Problems related to the Control of Wind Energy Conversion Systems-Maximum Power Extraction and Smoothing the Power Fluctuations delivered to the Grid", OPTIM-13th International Conference on Optimization of Electrical and Electronic Equipment, Optimum 2012, Brasov, 24-26 May, 2012. | [4] Babescu M., Borlea I., Gana O., Lacatusu F., "Comportarea sistemului electroenergetic eolian la variatii rapide ale vitezei vântului", Producerea, transportul si utilizarea energiei, Editura RISOPRINT Cluj-Napoca, 2010, ISSN 2066-4125, pp 11-24. | [5] Babescu M., Boraci.R, Chioeanu C, Koch C, Gana O., "On Functioning of the Electric Wind System at its Maximum Power", ICC-CONTI 2010, Timisoara, Romania, May 27-29, 2010. | [6] Florica Balog, Horia Ciocarlie, Marius Babescu, Doru-Ionut Petrescu, "Maximizing the Captured Energy of the Wind System", 9th IEEE International Symposium on Applied Computational Intelligence and Informatics, Timisoara, Romania, May 15-17, 2014, 978-1-4799-4694-5/14, p289-293. | [7] Horia Ciocarlie, Florica Balog, Gheza-Mihai Erdodi, Doru-Ionut Petrescu, "Peak Energy Determination by a Sample at Idle Mode Operation", 9th IEEE International Symposium on Applied Computational Intelligence and Informatics, Timisoara, Romania, May 15-17, 2014, 978-1-4799-4694-5/14, p289-293. | [8] Florica Balog, Horia Ciocarlie, Marius Babescu, Gheza-Mihai Erdodi, "Equivalent speed and equivalent power of the wind systems working at variable wind speed", 6th International Workshop on Soft Computing Applications, Timisoara, Romania, 24-26 July 2014. | [9] Horia Ciocarlie, Florica Balog, Gheza-Mihai Erdodi, Doru-Ionut Petrescu, "Behavior of the wind systems optimally controlled at variable wind speeds", 6th International Workshop on Soft Computing Applications, Timisoara, Romania, 24-26 July 2014. | [10] César Pérez López, "MATLAB Differential Equations", Springer, ISBN-13 (electronic): 978-1-4842-0310-1, 2014. | [11] Gilbert Strang, "Differential Equations and Linear Algebra", Wellesley-Cambridge Press, ISBN 978-0-9802327-9-0, 2014. | [12] The MathWorks Inc. 2015, "Parallel Computing Toolbox User's Guide". | [13] B. Barney, "Introduction to parallel computing", [https://computing.llnl.gov/tutorials/parallel\\_comp/#Whatis](https://computing.llnl.gov/tutorials/parallel_comp/#Whatis), 2012. | [14] S. Samsi A. Krishnamurthy, V. Gadepally, "Parallel MATLAB Techniques", 2009. | [15] Gaurav Sharma, Jos Martin, "MATLAB: A Language for Parallel Computing", Int J Parallel Prog (2009) 37:3-36, DOI 10.1007/s10766-008-0082-5, Springer, 2008. | [16] Yung-Sheng Chen, "Image Processing", INTECH, Croatia, ISBN 978-953-307-026-1, pp. 572, 2009. | [17] Robert Hammerling, Othmar Koch, Christa Simon, Ewa B. Weimüller, "Numerical solution of singular ODE eigenvalue problems in electronic structure computations", Elsevier, 1557-1561, 2010. | [18] Pierluigi Amodio, Luigi Brugnano, "Parallel solution in time of ODEs: some achievements and perspectives", ScienceDirect, Applied Numerical Mathematics 59 (2009) 424-435. | [19] Bhupesh Kumar Poosam, Hemant Amhia, "Control System Based on DSP for wind Turbine", Indian Journal of Applied Research, Vol. 3, Issue 1, ISSN - 2249-555X, 2013. |