Indian Journal of Science and Technology, *Vol 10(24), DOI: 10.17485/ijst/2017/v10i24/102620, June 2017*
ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

Simulating Electrohydrodynamic Ion-Drag Pumping on Distributed Parallel Computing Systems

Shakeel Ahmed Kamboh[1], Zubair Ahmed Kalhoro[2*], Kashif Ali Abro[3] and Jane Labadin[4]

[1]Department of Mathematics and Statistics, Quaid-e-Awam University of Engineering, Science and Technology, Nawabshah – 67450, Pakistan; shakeel.maths@yahoo.com
[2]Institute of Mathematics and Computer Science, University of Sindh, Pakistan; zubairabassi@gmail.com
[3]Department of Basic Science and Related Studies, Mehran University of Engineering and Technology, Jamshoro, Pakistan; kashif.abro@faculty.muet.edu.pk
[4]Faculty of Computer Science and Information Technology, Universiti Malaysia Sarawak, 94300 Kota Samarahan, Sarawak, Malaysia; ljane@unimas.my

Abstract

Objectives: This paper aims to simulate EHD ion-drag pumping model using Finite Difference Method (FDM) and to apply the idea of parallelism to reduce the computational time. **Methods:** The numerical simulation of EHD ion-drag pumping plays an important part not only to understand the different working principles but also enables to model the designs with better performance. Since the performance of EHD pumps depends on the shapes and geometries of the actuator electrodes, therefore the variation in the geometric dimensions of the electrodes require dense and fine meshes for numerical solution. Consequently, the numerical simulations take unacceptably more execution time on sequential computers. For that reason, a Data Parallel Algorithm for EHD model (DPA-EHD) is designed. To implement the parallel algorithm a distributed parallel computing system using MATLAB Distributed Computing Server (MDCS) is configured. The computational time and speedup with respect to the different number of processors is evaluated. **Findings:** This results show that the parallel algorithm for EHD simulations may provide 4.14 times more speedup over sequential algorithm for large grid sizes. **Improvements:** This study shows the feasibility of using the parallelism to reduce the computational time in the EHD model enabling to simulate the micropumps with very small dimensions of electrodes.

Keywords: Data Parallelism, Electrohydrodynamic, Ion-Drag Pumping, Parallel Algorithms, Parallel Distributed Computing Systems

1. Introduction

The Electrohydrodynamics (EHD) is concerned with the interactions of electrically charged fluids and their motion. This phenomenon is used in a wide range of attractive engineering applications and microfluidic devices. EHD micropumps have gained much attention due to their reliability and distinctive advantages of no moving parts[1]. Experimental research on the ion-drag micropumps has been widely carried out in the current years to improve their performance for different applications, particularly in cryogenic cooling of micro electro mechanical systems and most recently being used in jet printing[2,3]. But the development of new designs of such devices is challenging because of instrumentation cost and microfabrication complexity[4]. This issue provides an extensive research prospect for numerical modeling and simulation of micropumps. Since the performance of the micropumps depends on the shapes and geometries of the actuator electrodes. The variation in the geometric parameters requires the fine meshes resulted from the Discretization of the computational domain. Consequently, the computational time in obtaining the numerical simulation increases indefinitely. That is why; this work is aimed at the reduction of computational time in EHD ion-drag model by utilizing the art of parallelism. Few related studies have already been done to reduce the computational time of numerical simulation of specific EHD governing equation. For instance[5,6] have attempted to simulate only electric potential in 2D and3D prototype

** Author for correspondence*

of ion-drag micropumps on parallel systems respectively. This paper extends the parallelism methodology for the complete set of EHD governing equations including electrostatic and hydrodynamic partial differential equations.

2. Methodology

Ion-drag pumping mechanism is associated with the effect of high electric field strength is applied between the charged electrodes to move the neutral molecules of dielectric fluid as shown in the Figure 1.

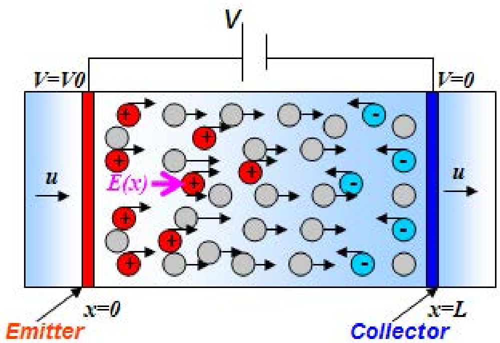

Figure 1. EHD Ion-drag mechanism[7].

The mechanism can be translated mathematically by using the laws of electrodynamics and fluid dynamics. The complete set of EHD governing equations is derived from the Maxwell's electromagnetism equations and fluid flow conservative equations[8,9]. The simplified system of 3D partial differential equations that govern the ion-drag pumping is listed as follows:

Electric body force (Coulomb force),

$$F_e = -q_e(x, y, z)\left(\frac{\partial V}{\partial x} + \frac{\partial V}{\partial y} + \frac{\partial V}{\partial z}\right) \tag{1}$$

Poisson's equation,

$$\varepsilon\left(\frac{\partial^2 V}{\partial x^2} + \frac{\partial^2 V}{\partial y^2} + \frac{\partial^2 V}{\partial z^2}\right) = -q_e(x,y,z) \tag{2}$$

Electric filed,

$$\vec{E} = -\left(\frac{\partial V}{\partial x} + \frac{\partial V}{\partial y} + \frac{\partial V}{\partial z}\right) \tag{3}$$

Conservation of charges,

$$\left(\frac{\partial J}{\partial x} + \frac{\partial J}{\partial y} + \frac{\partial J}{\partial z}\right) = 0 \tag{4}$$

Current density

$$J = \mu_e\, q_e(x,y,z)\left(\frac{\partial V}{\partial x} + \frac{\partial V}{\partial y} + \frac{\partial V}{\partial z}\right) \tag{5}$$

Space charge density

$$E\left(\frac{\partial q_e}{\partial x} + \frac{\partial q_e}{\partial y} + \frac{\partial q_e}{\partial z}\right) = -\frac{q_e^2}{\varepsilon} \tag{6}$$

Flow continuity equation

$$\left(\frac{\partial u}{\partial x} + \frac{\partial\ v}{\partial y} + \frac{\partial\ w}{\partial z}\right) = 0 \tag{7}$$

Navier-Stokes (Momentum) equations,

$$\rho\left(u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} + w\frac{\partial u}{\partial z}\right) = -\frac{\partial p}{\partial x} + \eta\left(\frac{\partial^2 u}{\partial^2 x} + \frac{\partial^2 u}{\partial^2 y} + \frac{\partial^2 u}{\partial^2 z}\right) - q_e\left(\frac{\partial V}{\partial x}\right)\Big\} \tag{8}$$

$$\rho\left(u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} + w\frac{\partial v}{\partial z}\right) = -\frac{\partial p}{\partial y} + \eta\left(\frac{\partial^2 v}{\partial^2 x} + \frac{\partial^2 v}{\partial^2 y} + \frac{\partial^2 v}{\partial^2 z}\right) - q_e\left(\frac{\partial V}{\partial y}\right)\Big\} \tag{9}$$

$$\rho\left(u\frac{\partial w}{\partial x} + v\frac{\partial w}{\partial y} + w\frac{\partial w}{\partial z}\right) = -\frac{\partial p}{\partial z} + \eta\left(\frac{\partial^2 w}{\partial^2 x} + \frac{\partial^2 w}{\partial^2 y} + \frac{\partial^2 w}{\partial^2 z}\right) - q_e\left(\frac{\partial V}{\partial z}\right)\Big\} \tag{10}$$

Equations (1-10) have to be solved iteratively using finite difference method for potential field *V* electric field *E*, space charge density *qe*, fluid velocity components (*u, v, w*) and the pressure field *p*. Once the potential distribution *V*, electric field *E* and the charge density *qe*, are determined then the magnitude of the Coulomb force F_e is obtained and coupled with the momentum equations i.e., Eqs. (8-10). The current density *J* can be computed from Eq. 5 and the power consumption is obtained by the relation $P = IV$. The numerically feasible solution of the governing equations requires the complete description of appropriate initial and boundary conditions for the computational domain of micropump. In this study, the basic design, boundary conditions and the sequential solution algorithm is adopted from[10] and is extended to 3D case.

Then the underlying sequential algorithm is parallelized using data parallel algorithm. For the sack of brevity, the DPA-EHD algorithm is illustrated by a flow chart as shown by Figure 2. Where, the symbols used in Figure 2 are described as follows:

D is the computational domain[10]; *l, m, n* are the number of finite difference cells along x, y, and z axis respectively *P* is the maximum number of workers (processors) used at a time; D_1, *D2, …,* D_p are the sub-domains obtained by partitioning the whole domain

D; W_1, $W2$, …, W_P are the parallel workers; ε is the error tolerance set as 0.0001; γ is the iteration number; Φ is the message passing mapping that sends the data to neighboring workers; ψ is the message passing mapping that receives data from the neighboring workers; $F_e^1 = -q_e \frac{\partial V}{\partial x}$, $F_e^2 = -q_e \frac{\partial V}{\partial y}$, $F_e^3 = -q_e \frac{\partial V}{\partial z}$; ε_l is the local error obtained by local solutions at individual worker; and ε_g is the global error that is obtained by a global operation on all workers.

3. Results and Discussion

Parallel numerical solution is computed by implementing the DPA-EHD algorithm using MATLAB by configuring parallel/distributed computing environment with the same resources as used by[6]. The EHD equations with different grid sizes are solved on a cluster of 2, 4, 8, 12 and 16 MATLAB in distributed memory workers. The most

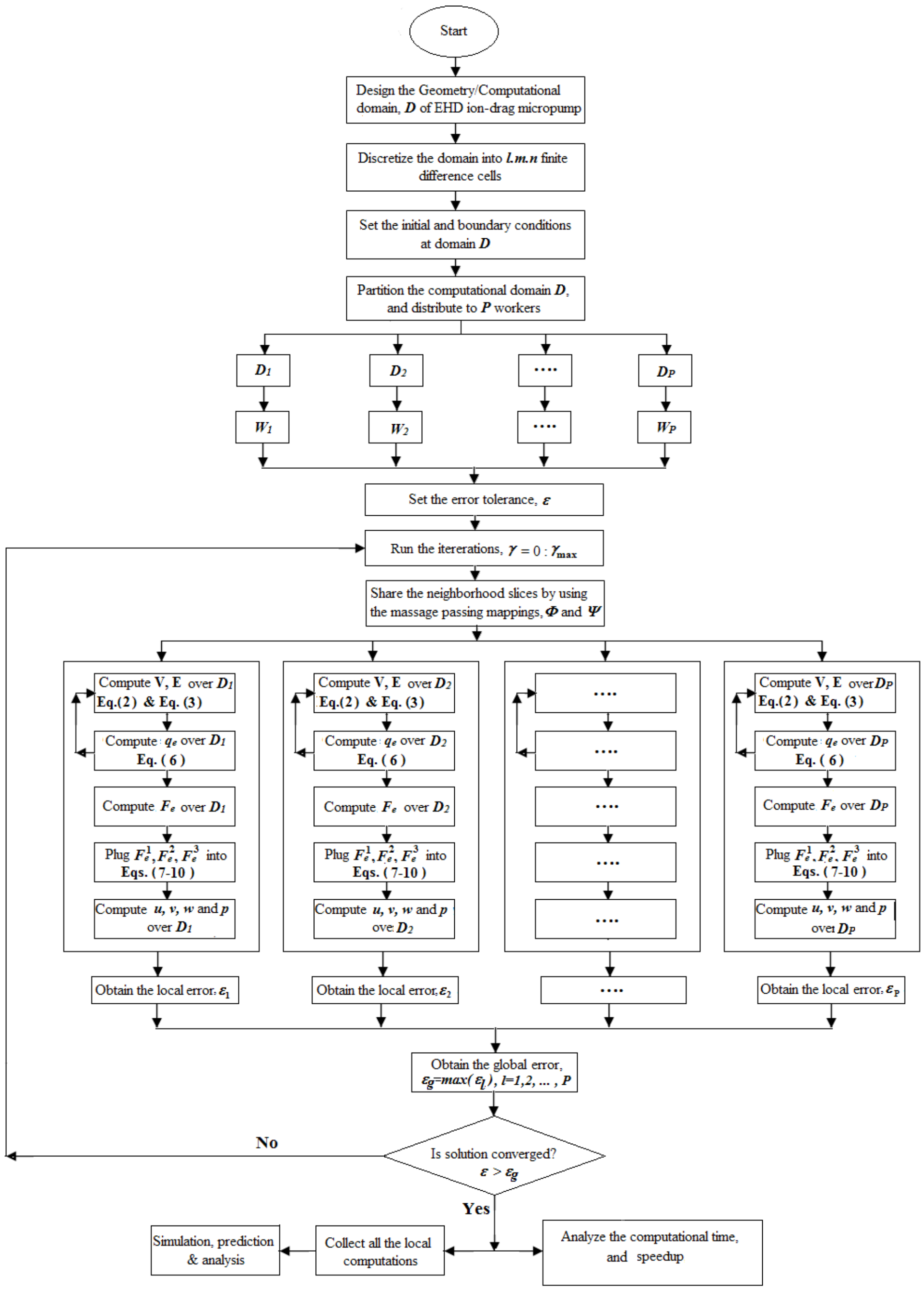

Figure 2. DPA-EHD for parallel numerical solution of EHD equations.

important parameters to analyze are the computational time (sec) and the measured speedup. Figure 3 the computational time obtained on different grid sizes at the specific number of workers is presented. It is observed that the sequential computational time at P=1 increases with increase in the grid size. While the parallel computational time at *P=2, 4, 8, 12, 16* is reduced by some fraction of the sequential time. However, there is no significant reduction in time after particular number of workers. The DPA-EHD can be more efficient for *P=2, 4, 8* and at most *P=12* but after this the parallel computational time gets increasing. The computational time is shown by using another orientation in Figure 4, where it is represented as a function of *P* number of workers at a specific grid size. This gives a mixed impression of increase and decrease in the computational time with respect to number of workers. It also shows that for a particular grid size the computational time decreases by increasing the number of workers but after some degree the computing time increases indefinitely. Which lead to the conclusion that increasing the number of workers does not necessarily reduce the computing time.

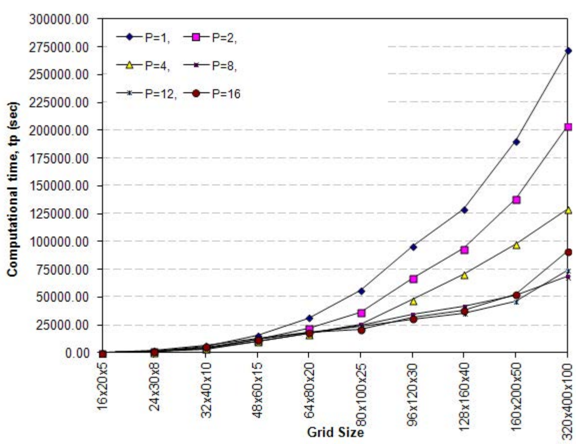

Figure 3. Computational time versus grid size with respect to different workers.

In fact, the computational time is a function of several influential parameters but the most common and important are the input data size and the number of workers used. In the current case, the computing time is analyzed with respect to grid size and the number of workers independently. It is obvious that the computational time will increase by increasing the data size but its behavior depends on the nature of computational problem and the parallel hardware.

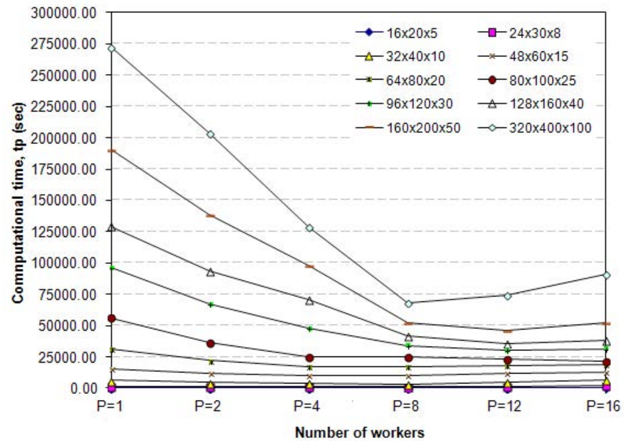

Figure 4. Computational time versus workers with respect to different grid sizes.

After analyzing the parallel computational time of DPA-EHD it is important to measure the speedup achieved by the algorithm. Figure 5 exhibits the speedup as functions of number of workers. The speedup for large grid sizes improves up to some extent and then decreases as more number of workers is added. For 160 x 200 x 50) the DPA-EHD can reach a maximum speed up of 4.14 with *P=12* numbers of workers. Whilst, for the grid size (320 x 400 x 100) the maximum speedup of *4* with *P=8* can be achieved. This demonstrates that there is optimum number of workers for each grid size and for this reason the scalability of the DPA-EHD is restricted to limit number of workers. It is because of the reason that in the small grid sizes the fine granularity occurs.

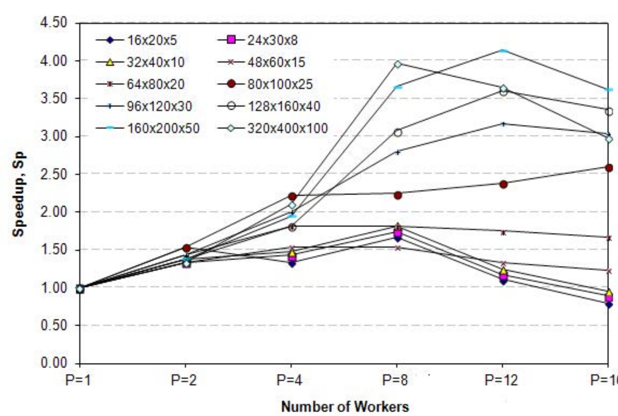

Figure 5. Speedup versus workers with respect to different grid sizes.

4. Conclusions

The work presented in this paper was motivated by the need of parallelism to reduce the computing time in simulation of a computationally intensive problem. The complete EHD ion-drag pumping model was parallelized using data parallel technique and implemented on parallel/ distributed computing environment using MATLAB. The parallel results demonstrate the feasibility of using the parallelism in EHD model with reduced computational time. It is revealed that to process the solution on 1600000 finite difference cells as a maximum speed up of about 4.14 can be achieved on 12 workers. Hence the parallel algorithm for EHD model reduces computational effort on distributed memory systems.

5. Acknowledgement

The authors would like to acknowledge the Quaid-e-Awam University of Engineering Science and Technology, Nawabshah, Pakistan and the University Malaysia Sarawak for providing the facilities to conduct this research.

6. References

1. Darabi J, Wang H. Development of an electrohydrodynamic injection micropump and its potential application in pumping fluids in cryogenic cooling systems. Journal of Microelectromechanical Systems. 2005; 14(4):747–55. Crossref
2. Fylladitakis ED, Theodoridis MP, Moronis AX. Review on the history, research, and applications of electrohydrodynamics. IEEE Transactions on Plasma Science. 2014; 42(2):358–75.
3. Park JU, Hardy M, Kang SJ, Barton K, AdairK, Mukhopadhyay KD. High-resolution Electrohydrodynamic Jet Printing. Nature Materials. 2007; 6(10):782–9. Crossref
4. Shoushtari AH. Experimental and computational analysis of an electrohydrodynamic mesopump for spot cooling applications. DRUM Digital Repository of the University of Maryland; 2004.
5. Kamboh SA, Labadin J, Rigit ARH, Chaw LT. Parallel computation of electric potential in the EHD ion-drag micropump and the performance analysis of the parallel system. Proceedings of 8th CITA, Malaysia; 2013. p. 195–9. Crossref
6. Kamboh SA, Labadin J, Rigit ARH, Ling TC, Amur KB, Chaudhary MT. Computational time analysis of the numerical solution of 3D electrostatic Poisson's equation. Proceedings of Conference Artificial Intelligence Modelling and Simulation. Malaysia; 2015. p. 1–7. Crossref
7. Kamboh SA, Labadin J, Rigit ARH. 3D modeling and simulation of electro hydrodynamic ion-drag micropump with different configurations of collector electrode. Proceedings of Intelligent System Modelling and Simulation, Malaysia; 2012. p. 417–22.
8. Castellanos A. Electrohydrodynamics. Springer, New York; 1998. Crossref
9. Melcher JR. Continuum Electromechanics. Cambridge: MIT Press; 1981.
10. Kamboh SA, Labadin J, Rigit ARH. Computational modeling and simulation of EHD ion-drag pumping using finite difference method. Proceedings of IEEE Artificial Intelligence Modelling and Simulation. Malaysia; 2013. p. 207–11. Crossref