Available online at www.prace-ri.eu

Partnership for Advanced Computing in Europe

Scalable Parallel Nonlinear Parameter Optimization Algorithm with Parameter Pools

Ahmet Duran[a,b]* and Mehmet Tuncel[a,c]

[a]*Istanbul Technical University, National Center for High Performance Computing of Turkey (UHeM), Istanbul 34469, Turkey*
[b]*Istanbul Technical University,Department of Mathematics, Istanbul 34469, Turkey*
[c]*Istanbul Technical University, Informatics Institute, Istanbul 34469, Turkey*

17 June 2014, updated: 11 August 2014

Abstract

In this project, we propose a new hybrid algorithm for parameter optimization and implement it using MPI. In particular, we study a scalable parallel nonlinear parameter optimization algorithm with parameter pools for a nonlinear dynamical system called the asset flow differential equations (AFDEs) in $\Re^4$. We generate time series pairs as proxy to market price and net asset value by using random walk simulation where the volatilities of the time series are similar to that of real closed-end funds traded on New York Stock Exchange (NYSE). When we apply the algorithm by using simulations for a set of time series, we observe that the computed optimal parameter values, average number of quasi-Newton iterations, the average nonlinear least squares errors, and the average maximum improvement factors can converge certain values within corresponding small ranges, after oscillations. Moreover, we tested for 64, 128, 256 and 512 cores using the 512 initial parameter vectors. We achieved speed-up for the time series to run up to 512 cores. The algorithm is applicable for parameter optimization of the related nonlinear dynamical system of differential equations with thousands of parameters as well.

1. Introduction

In this work, we study a scalable parallel nonlinear parameter optimization algorithm with parameter pools for a nonlinear dynamical system called the asset flow differential equations (AFDEs) in $\Re^4$. The algorithm in this work is important also for parameter optimization of the related dynamical system of differential equations.

Parallel methods for parameter optimization based on the quasi-Newton method with the Broyden–Fletcher–Goldfarb–Shanno formula (see [1-2]) have been given attention in academic and industrial literature. A parallel approach was described for single and multiple Gaussian data fitting problems where the reference functions for fitting are known explicitly (see Caprioli and Holmes [3] and references contained therein). We focus on a problem where the reference functions for fitting are not known explicitly and come from the numerical solution of a challenging nonlinear dynamical system.

AFDEs have been developed by Caginalp and collaborators since 1989 (see [4,5] and references contained therein). This important mathematical model may explain different nonlinear behaviours such as overreaction, bubbles, momentum, and crashes in experimental asset markets and real financial markets (see [6] and [7]). It incorporates several motivations for buying or selling stock with the finiteness of assets and microeconomic principles. The dynamical microeconomic model provides valuable constraints analogous to conservation laws in physics, rather than the classical time series analysis with a single stage approach (see Duran and Caginalp [8]).

Duran (see [9]) introduced a serial algorithm called the asset flow optimization forecast algorithm. An inverse problem involving parameter optimization for AFDEs has been used in order to forecast near term market

* Corresponding author. *E-mail address*: aduran@itu.edu.tr.

returns by following an out-of-sample procedure (see [8]). A quasi-Newton (QN) weak line search with the Broyden–Fletcher–Goldfarb–Shanno formula and their semi-dynamic initial parameter pool are utilized in conjunction with daily market prices (MPs) and net asset values (NAVs) to determine the parameters for which the AFDEs yield the best fit for the previous *n* days in the optimization procedure. They use nonlinear least-square technique with initial value problem (IVP) approach by focusing on the market price variable x_1 since any real data for the other three variables x_2, x_3 , and x_4 in the dynamical system is not available explicitly. The gradient (∇F (x)) is approximated by using the central difference formula, and step length *s* is determined by the backtracking line search (Nocedal and Wright [10]). They construct a pool of initial parameters K_i chosen via a set of grid points in a hyper-box ([8]). They select an initial parameter vector from the initial parameter pool because the optimization success of quasi-Newton method in the algorithm depends on the initial parameter. Besides the fixed part of various initial parameters, the dynamic part of the pool is updated by adding successful parameters so that they keep a pool of different and most recently used candidate parameters. It is a feasible dynamic multi-start approach without a convexity assumption for their semi-unconstrained optimization problem. These optimal parameters are useful for making a forecast for market prices for the following days.

After the parametric sensitivity analysis (see [11]), Duran studied the stability analysis of the AFDEs, in three versions, analytically and numerically (see [12]). It is crucial to analyze the sources of ill-posedness in mathematical modeling. Duran showed that the existence of multiple roots and that of non-isolated roots are sources of the ill-posedness for the first two versions of AFDEs (see[12]). He illustrated how to reformulate the problem in order to eliminate any hypersensitivity in the mathematical model.

There are several challenges while studying numerical parameter optimization of the nonlinear dynamical systems. For example, some initial parameters may lead to singularities in the AFDE during parameter optimization process. Our implementation handles this kind of problems. Moreover, for optimization methods using derivatives in a nonlinear model it is important to start the iteration close enough to the potential global minimum to get rid of being caught in a local minimum. There is no strategy that will guarantee the number of necessary iterations to discover the neighbourhood of the global optimum (see [13, Chapter 23]). There is also a wide range of variability in obtaining optimal parameters for the nonlinear problem. That is, the residual values may change between 10^{-1} and 10^{-14}. Therefore, we need sufficiently large number of initial parameters systematically via high performance computing.

The remainder of this work is organised as follows: Section 2 includes the 3rd version of AFDEs and the problem constraints. In Section 3, the parallel nonlinear optimization algorithm is presented. In Section 4, the scalability test results and the convergence results of the numerical parameter optimizations are discussed. Section 4 concludes this work.

2. AFDEs version 3

We rewrite the dynamical system of asset flow differential equations in [5] in the following equivalent form:

$$x_1' = x_1 \delta \log((k(1-x_2)) / ((1-k)x_2))$$

$$x_2' = x_2(1-x_2)\delta \log((k(1-x_2)) / ((1-k)x_2)) + k(1-x_2) + (k-1)x_2$$

$$x_3' = c_1(q_1 \delta \log((k(1-x_2)) / ((1-k)x_2)) - x_3)$$

$$x_4' = c_2(q_2((P_a - x_1)/P_a - D(x_1(t-1), P_a(t-1), x_1(t-2), P_a(t-2), ..., x_1(t-n), P_a(t-n))) - x_4)$$

where

$$D(x_1(t-1), P_a(t-1), x_1(t-2), P_a(t-2), ..., x_1(t-n), P_a(t-n))$$

is the chronic discount over the past few finite $n \geq 1$ days. The constraints are

$x_1 > 0$ (positivity of prices)

$0 < x_2 < 1$

$-1 < x_3 + x_4 < 1$

$P_a > 0$ (positivity of prices)

$K = (\delta, c_1, q_1, c_2, q_2) \in \Re_+^5$ (positivity of parameter vectors)

where

$x_1(t)$: The market price (MP) of the single asset at time *t*,

$x_1^{'}(t) / x_1(t)$: The relative price change,

$P_a(t)$: The fundamental value,

$V(t)$: The net asset value (NAV) price at time *t*,

$x_2(t)$: The fraction of total funds in the risky asset,

$x_3(t)$: The trend-based component of the investor preference,

$x_4(t)$: The value-based component of the investor preference,

$k(t)$: The transition rate and $k = 0.5 + 0.5\tanh(x_3 + x_4)$.

k should take values within (0,1) and *tanh(x)* can be approximated by *x* around (-1,1) as in [5]. The constants δ, 1/ c_1 and 1/ c_2 are the time scales, respectively, for the price equation, the momentum and valuation strategies. We take δ as 1. The parameters q_1 and q_2 are the coefficients of the trend-based and value-based sentiment, respectively.

3. Parallel nonlinear parameter optimization algorithm with parameter pools

Duran (see [9]) introduced a serial algorithm called the asset flow optimization forecast algorithm. Given an *n*-day period of market prices (MP) and net asset values (NAV) from day *i* to day *i+n-1* as *i*th event where $n = \tau_1 + 1$ and $i > \tau_2$, we compute optimal parameter vector K_i for the period *i*. Then, we obtain *m-i+1* optimal parameters for the overlapping periods such as *[i, i+n-1], [i+1, i+n], ..., [m, m+n-1]* for the MP sequence of size *m+n-1*. There is a tradeoff for selection of *n*. We choose *n* sufficiently small in order to use the daily market price and net asset value. Moreover, local price patterns which are related to 3 to 15 trading days on average (see [7] and [14]) can be used by small values of *n* during optimization and prediction processes. On the other hand, *n* should be large enough so that the parameter optimization process can capture the price trend reasonably. For example, we tested for *n=5* trading days which may reflect weekly economic and financial indicators. See [8] for details and the cases for *n=5* and *n=10*.

ALGORITHM 3.1 The parallel nonlinear parameter optimization algorithm

Stage 1. Obtain classified initial parameter pool having partitions that can generate different curves having various behaviors.
Stage 2. Apply pool partitioning for parallelism. Each core should find the local optimal parameter(s) by using its local initial parameters.
Stage 3. Find the global parameter that can minimize the nonlinear least squares error.

In this study, we propose a parallel nonlinear parameter optimization algorithm. One of the novel components of the former algorithm was the presence of the dynamic initial parameter pool that contains most recently used successful parameters, besides the various fixed parameters from a set of grid points in a hyper-box. Therefore, it has dependencies on the most recently used successful parameters.

We use fixed initial parameter pool with a larger number of parameter vectors in Algorithm 3.1 so that we can get rid of the dependencies. Unlike the serial algorithm, the new algorithm has a classified initial parameter pool with partitions that can generate different curves having behaviors such as almost steady, uptrend, downtrend, strong uptrend and strong downtrend. In Stage 2, as a part of the computational parallelization strategy, a fixed number of initial parameters is assigned to each core. For example, when there are *s* available cores, each core is responsible for the parameter optimization with 512/s initial parameters for the corresponding curve segment. Each core performs curve fitting by using its own initial parameters and the steps in the serial algorithm (see [8] for details) are followed to find the local optimal parameters. In Stage 3, the nonlinear least squares errors coming from the different cores are compared and the parameter vector that can minimize the error is picked as the global parameter for the corresponding curve segment. These three stages are repeated for each consecutive curve segment.

4. Test results

We generate time series pairs as proxy to market price and net asset value by using random walk simulation where the volatilities of the time series are similar to that of real closed-end funds traded on NYSE (see [11]). See Anderson and Born [15] and Bodie, Kane, and Marcus [16] for more information about the closed-end funds.

Table 1 displays the design and threshold values for the computational optimization process. Table 2 describes the simulated market price and net asset value time series of length 1000, their volatility behavior and ranges.

Table 1. The computational optimization by finding parameters in the AFDE for a large sample data set. Quasi-Newton method with weak line search is applied.

Event period	5
Runge–Kutta (RK4) method step size	0.05
Number of parameter vectors in the pool	512
Threshold for the gradient	10^{-5}
Threshold for the nonlinear least squares error	0.16

Table 2. Description of the time series .

Time series	Standard deviation	Max	Min
Price_1k_v1	3.94	65.68	48.19
Nav_1k_v1	2.25	58.03	48.32
Price_1k_v2	5.01	67.50	48.77
Nav_1k_v2	2.38	63.94	53.68
Price_1k_v3	2.66	61.68	49.38
Nav_1k_v3	1.81	58.78	49.99
Price_1k_v4	2.04	57.45	46.69
Nav_1k_v4	1.51	57.89	50.58
Price_1k_v5	7.16	78.61	53.76
Nav_1k_v5	4.75	71.40	51.42
Price_1k_v6	3.63	56.92	42.00
Nav_1k_v6	3.15	60.18	47.92
Price_1k_v7	3.63	67.31	51.82
Nav_1k_v7	2.91	61.55	48.45

4.1 Scalability results

We tested for 64, 128, 256 and 512 cores using the 512 initial parameter vectors. Since we applied the partitioning of the initial parameter vectors to available cores as the computational parallelization strategy, each core performed computational parameter optimization with 16, 8, 4, 2 and 1 initial parameter vectors respectively. This approach can be followed by different initial parameter pools having larger number of initial parameters and larger number of cores.

We obtained speed-up for the time series 1k_v1, 1k_v2 and 1k_v3 to run up to 512 cores in Table 3. We obtained almost similar results for the time series. Figures 1 and 2 illustrate the speed-up behaviour for the time series 1k_v1 and 1k_v3, respectively.

Table 3. Wall clock time and normalized speed-up for 1k_v1, 1k_v2 and 1k_v3 on the Ege Server (HP ProLiant BL2x220c G5 Blade) (see [17]).

# of cores	1k_v1		1k_v2		1k_v3	
	Wall clock time (s)	Speed-up	Wall clock time (s)	Speed-up	Wall clock time (s)	Speed-up
64	40556.60	1	40791.01	1	37012.84	1
128	28071.28	1.45	27677.68	1.47	24972.18	1.48
256	16257.88	2.50	16242.60	2.51	14508.45	2.55
512	11310.66	3.59	11217.48	3.64	9984.60	3.71

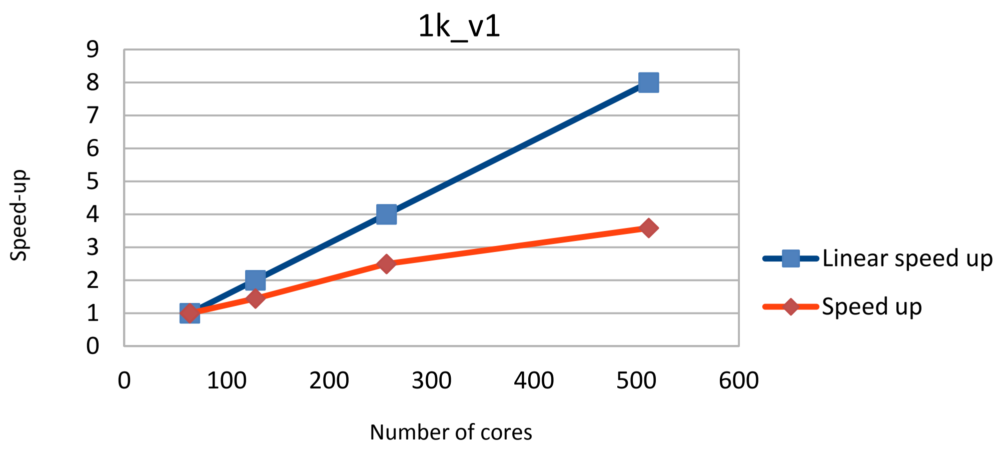

Fig. 1. Speed-up for Price_1k_v1

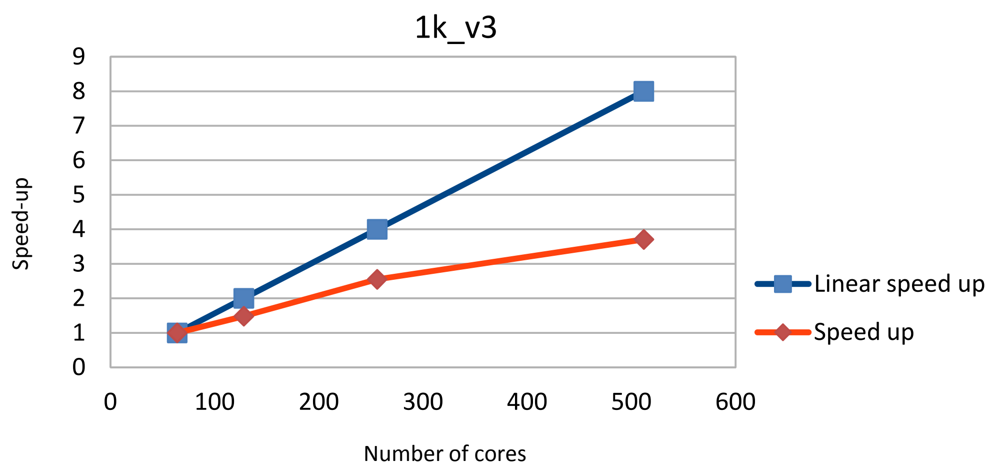

Fig. 2. Speed-up for Price_1k_v3

4.2 Convergence results of the parameter optimization

Table 4 shows the wall time for testing the parallel nonlinear parameter optimization algorithm for 128 cores on the Ege Server (HP ProLiant BL2x220c G5 Blade) (see [17]). Table 5 illustrates the Monte Carlo simulation results for the parameters, the average number of QN iteration, the average nonlinear least squares error and the average maximum improvement factor (MIF) where MIF is used to measure the performance of the optimization process and it is defined as the ratio of the final nonlinear least squares error to the initial nonlinear least squares error. Generally, the smaller MIF corresponds to a better performance, which depends on the closeness of the initial parameter to the optimal one as well.

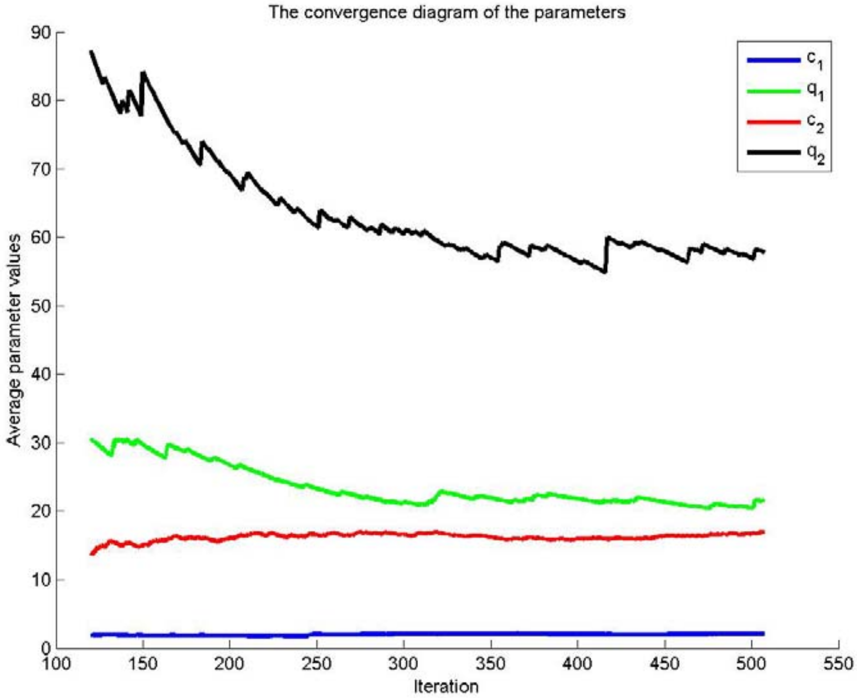

Fig. 3. Monte Carlo simulation of the parameters for curve fitting via Price_1k_v2

Table 4. Wall clock time for 128 cores on the Ege Server (see [17]) available at UHeM.

Time series	Wall clock time (s)
Price_1k_v1	28071.28
Price_1k_v2	27677.68
Price_1k_v3	24972.18
Price_1k_v4	28195.40
Price_1k_v5	27389.56
Price_1k_v6	28273.22
Price_1k_v7	25723.34

Table 5. Monte Carlo simulations results.

Time series	Parameters				Average number of QN iteration	Average NLS error	Average MIF
	c_1	q_1	c_2	q_2			
Price_1k_v1	1.9536	18.7947	18.7911	45.4401	169.99	0.0116	0.1979
Price_1k_v2	2.0913	21.5848	16.9350	57.7354	166.88	0.0164	0.2083
Price_1k_v3	2.0609	16.4288	17.8410	40.5121	160.19	0.0115	0.1973
Price_1k_v4	1.4278	17.9718	13.5950	34.9702	144.25	0.0154	0.2434
Price_1k_v5	1.9814	20.7925	16.4631	55.8295	163.55	0.0171	0.2180
Price_1k_v6	2.1081	18.1281	17.9992	45.9608	147.66	0.0134	0.2321
Price_1k_v7	1.5803	22.8690	16.6832	52.3300	144.70	0.0161	0.2427

When we compare the serial algorithm with dynamic initial parameter pool having up to 80 initial parameter vectors and the new parallel algorithm having 512 initial parameter vectors in the classified pool, the average numbers of QN iteration are close to each other. On the other hand, we obtained smaller nonlinear least squares errors and better maximum improvement factors via the new parallel algorithm, in quality of the solution. For example, the average numbers of QN iteration is 162.15, the average nonlinear least squares error is 0.0231 and the average maximum improvement factor is 0.3399 for Price_1k_v1 using the serial algorithm. For the new parallel algorithm those values are 169.99, 0.0116 and 0.1979, respectively, in Table 5.

In Table 5, for the new algorithm by using simulations for a set of time series, we observe that the computed optimal parameter values, average number of quasi-Newton iterations, the average nonlinear least squares errors, and the average maximum improvement factors can converge certain values within corresponding small ranges, after fluctuations. For example, Figures 3-6 show the convergence diagrams for the curve fitting of the time series Price_1k_v2 using Runge–Kutta (RK4) method in order to solve the dynamical system numerically.

Fig. 4. Monte Carlo simulation of the number of quasi-Newton iteration for curve fitting of Price_1k_v2

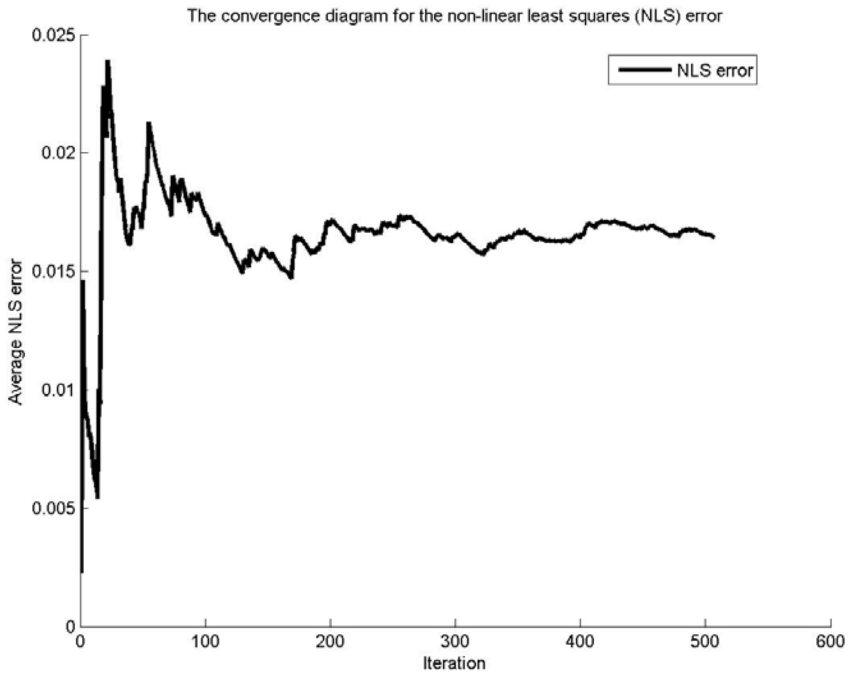

Fig. 5. Monte Carlo simulation of the NLS error for curve fitting of Price_1k_v2

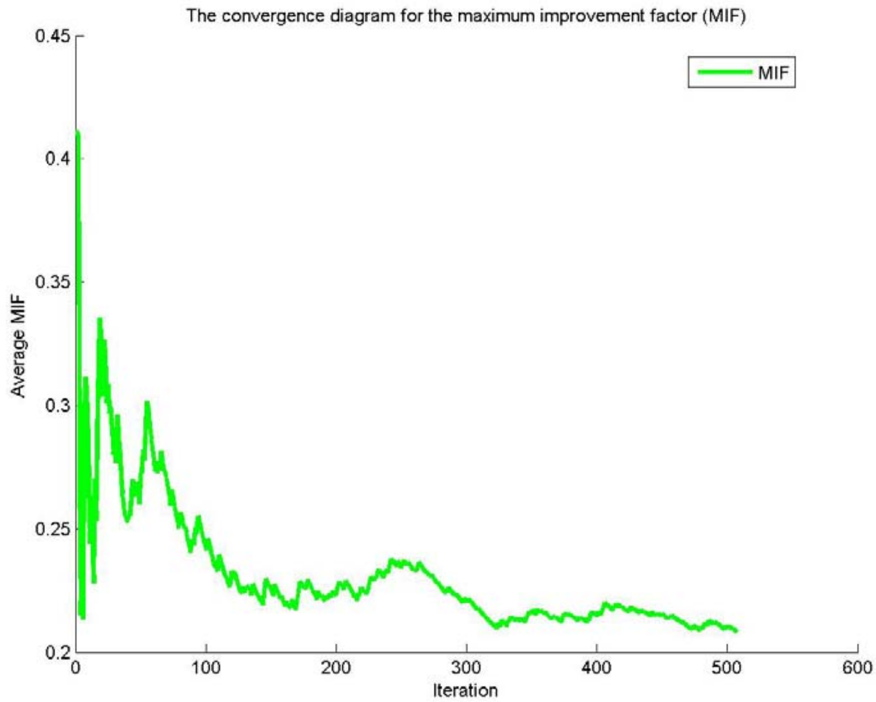

Fig. 6. Monte Carlo simulation of the MIF for curve fitting of Price_1k_v2

5. Conclusions

In this study, we propose a scalable parallel nonlinear parameter optimization algorithm with parameter pools for the asset flow differential equations (AFDEs) in $\Re 4$. The algorithm in this work is applicable for parameter optimization of the related nonlinear dynamical system of differential equations with thousands of parameters.

We find that the new parallel algorithm having a classified initial parameter pool with partitions that can generate different curves outperforms the sequential parameter optimization algorithm using dynamic initial parameter pool in quality of the solution. We obtained smaller nonlinear least squares errors, better maximum improvement factor (MIF), and curve fitting for more curve segments, by the advantage of using sufficiently large number of initial parameters methodically. For example, the nonlinear least squares error was reduced to half and the MIF quality was enhanced approximately 14.2 % (from 0.3399 to 0.1979) via the new parallel algorithm, where generally the smaller maximum improvement factor corresponds to a better quality as described before, which may depend on the proximity of the initial parameter to the optimal one as well.

As a part of the computational parallelization strategy, a fixed number of initial parameters is assigned to each core. When there are s available cores, each core is responsible for the parameter optimization with 512/s initial parameters for the corresponding curve segment. We tested for 64, 128, 256 and 512 cores using the 512 initial parameter vectors. We achieved almost similar speed-up behaviour for the time series to run up to 512 cores. This approach can be followed by different initial parameter pools having larger number of initial parameters and larger number of cores.

Alternatively, we may try dynamic initial parameter vector assignments to cores. For example, first, each core can start with one parameter vector in Stage 2 in Algorithm 3.1 and attempt to take new one when it finishes the task. As a future work we will test different work scheduling and load balancing strategies and perform overhead analysis. However, the number of QN iteration for the parameter optimization varies depending on the corresponding initial parameter vector from the parameter pool and heterogeneously distributed. Therefore, it does not affect the wall clock time due to the bottleneck effect of the dominating number of QN iteration. Moreover, we will adapt the parallel QN approach (see [18]) in our parallel nonlinear parameter optimization algorithm with parameter pools for a better scalability and discuss the tradeoffs.

References

[1] C.G. Broyden, The convergence of a class of double rank minimization algorithms, part 1, J. Inst. Math. Appl. 6 (1970) 76–90.

[2] C.G. Broyden, The convergence of a class of double rank minimization algorithms, part 2, J. Inst. Maths. Appl. 6 (1970) 222–231.

[3] P. Caprioli and M.H. Holmes, A parallel quasi-Newton method for Gaussian data fitting, Parallel Computing 24 (1998) 1635-1651.

[4] G. Caginalp and B. Ermentrout, Numerical studies of differential equations related to theoretical financial markets, Appl. Math. Lett. 4 (1991) 35–38.

[5] G. Caginalp and D. Balenovich, Asset flow and momentum: deterministic and stochastic equations, Phil. Trans. R. Soc. 357 (1999) 2119–2133.

[6] G. Caginalp, D. Porter, and V.L. Smith, Overreactions, momentum, liquidity, and price bubbles in laboratory and field asset markets, J. Psychol. Finan. Mark. 1 (2000) 24–48.

[7] A. Duran and G. Caginalp, Overreaction diamonds: precursors and aftershocks for significant price changes, Quant. Finance 7 (2007) 321–342.

[8] A. Duran and G. Caginalp, Parameter optimization for differential equations in asset price forecasting, Optim. Methods Softw. 23 (2008) 551–574.

[9] A. Duran, Overreaction Behavior and Optimization Techniques in Mathematical Finance, PhD thesis, University of Pittsburgh, Pittsburgh, PA, Aug. 1st 2006.

[10] J. Nocedal and S.J. Wright, Numerical Optimization, 2nd ed., Springer Series in Operations Research, Springer-Verlag, New York, 2006.

[11] A. Duran, Sensitivity analysis of asset flow differential equations and volatility comparison of two related variables, Numer. Funct. Anal. Optim. 30 (2009) 82–97.

[12] A. Duran, Stability analysis of asset flow differential equations, Applied Mathematics Letters 24 (2011) 471-477.

[13] M. Bartholomew-Biggs, Nonlinear Optimization with Financial Applications, KluwerAcademic Publishers, Boston, 2005.

[14] G. Caginalp and D. Balenovich, A theoretical foundation for technical analysis, J. Tech. Anal. 59 (2003) 5-22.

[15] S.C. Anderson and J.A. Born, Closed-End Fund Pricing: Theories and Evidence, Kluwer Academic Publishers, Boston, MA, 2002.

[16] Z. Bodie, A. Kane, and A.J. Marcus, Investments, 6th ed., Irwin McGraw-Hill, Boston, MA, 2005.

[17] http://www.uybhm.itu.edu.tr/page.php?id=106

[18] I. Pardines and F. Rivera, Parallel Quasi-Newton Optimization on Distributed Memory Multiprocessors, Proceedings of the International Conference Parallel Computing (ParCo2001), Naples, Italy, 4-7 September 2001, Imperial College Press, 338 - 345.
[19] http://www.uybhm.itu.edu.tr

Acknowledgements

This work was financially supported by the PRACE project funded in part by the EUs 7th Framework Programme (FP7/2011-2014) under grant agreement no. RI-283493. The work was achieved using the computing resources of the National Center for High Performance Computing of Turkey (UHeM) (see [19]).