

Research Article

Application of Support Vector Regression in Krylov Solvers

Rehana Thalib, Maharani Abu Bakar* and Nur Fadhilah Ibrahim

Universiti Malaysia Terengganu, Malaysia

p3720@pps.umt.edu.my; maharani@umt.edu.my; fadhilah@umt.edu.my

*Correspondence: maharani@umt.edu.my

Received: 17th October 2020; Accepted: 15th November 2020; Published: 20th March 2021

Abstract: Support vector regression (SVR) is well known as a regression or prediction tool under the Machine Learning (ML) which preserves all the key features through the training data. Different from general prediction, here, we proposed SVR to predict the new approximate solutions after we generated some iterates using an iterative method called Lanczos algorithm, one class of Krylov solvers. As we know that all Krylov solvers, including Lanczos methods, for solving the high dimensions of systems of linear equations (SLEs) problems experiences breakdown which causes the sequence of the iterates is incomplete, or the good approximate solution is never reached. By assuming that some iterates exist after the breakdown, then we could predict what they are. It is realized by learning the previous iterates generated by the Lanczos solvers, which is also called the training data. The SVR is then used to predict the next iterate which is expected the sequence now has similar property as the previous one before breaking down. Furthermore, we implemented the hybrid SVR-Lanczos (or SVR-L) in the restarting frame work, then it is called as hybrid restarting-SVR-L. The idea behind the restarting is that one time running hybrid SVR-L cannot obtain a good approximate solution with small residual norm. By taking one iterate which is resulted by the hybrid SVR-L, putting it as the initial guess, will give us the better solution. To test our idea of prediction of SLEs solutions, we also used the regular regression and compared with the SVR. Numerical results are presented and compared between these two predictors. Lastly, we compared our proposed method with existing interpolation and extrapolation methods to predict the approximate solution of SLEs. The results showed that our restarting SVR-L performed better compared with the regular regression.

Keywords: SLEs; Machine Learning; SV; Lanczos solvers; hybrid; restarting

1. Introduction

Machine learning (ML) is the recent technology that is used broadly in many different discipline subjects. Literally, ML means that a computer learns the behaviour or pattern of some data given [1]. The pattern is then used for the classification or prediction. The given data is called the training data, while the prediction is called the output. To realize the process of the classification and prediction, ML requires some techniques such as support vector machine (SVM), support vector regression (SVR), artificial neural network (ANN), etc [2]. The SVM and SVR differs by the purpose of the tools where the first one aims to classify, while the second aims to regress. Different from general prediction, in this study, we use SVR to predict an approximate solution of the systems of linear equations (SLEs) based on the features of some iterates generated by an iterative method, called Lanczos algorithm, which is Krylov based solvers,

Lanczos-type solvers revised Lanczos method proposed by Cornelius Lanczos [3], by introducing the theory of orthogonal polynomials (FOPs), which enables us creating some recurrence relationships between the orthogonal polynomials, P_k [4]. This leads two other classes of Krylov iterative methods, called Baheux-types, [4], and Farooq-types, [5]. The differences between

the two are that Baheux-types were built by the three-terms recurrence relationships of $P_k, P_{k-1}, P_{k-2}, P_k^{(1)}, P_{k-1}^{(1)}$, and $P_{k-2}^{(1)}$, whereas Farooq-types were built from Baheux-types with addition P_{k-3} and $P_{k-3}^{(1)}$. Definitely, Farooq-types need more computation of the coefficient of P_k , compared with Baheux-types, however, they are more robust to find a good solution [5].

In the recent advances of Lanczos-type methods, they focused on dealing with the breakdown, for instance restarting Lanczos-types [6], and switching Lanczos-types [7]. Restarting enables us to restart the Lanczos solvers whenever they faced breakdown. It is involving the choice of a good quality of iterate used to restart with [8]. On the other hand, switching deals with the use of some Lanczos-types running alternately whenever the breakdown occurs. Furthermore, [9] introduced a model prediction to find an approximate solution of SLEs. The methods involving EIEMLA [10] and MEIEMLA [11]. The later one revised the firsts one by the way to interpolate some iterates generated by the Lanczos-types. This study will look at this prediction model by using SVR and the regular regression. The benefit of using these two prediction tools is that they are more accurate to predict compared with the extrapolation method. In fact, SVR is chosen in this study since it is suitable for our data set at the moment.

2. Lanczos-Type Algorithms

Lanczos-type algorithms are well-known as an effective iterative methods for solving non-symmetric system $A\mathbf{x} = \mathbf{b}$, where $A \in \mathbb{R}^{n \times n}$, and $\mathbf{x}, \mathbf{b} \in \mathbb{R}^n$. It is a Krylov-based that employs the theory of formal orthogonal polynomials by defining the linear function c which satisfies $c(t^i P_k(t) = 0)$, for $i = 0, 1, \dots, k-1$ where $k < n$ and n be the dimension of the SLEs, to compute the coefficients of polynomials $\mathbf{r}_k = P_k \mathbf{r}_0$ [3]. The approximate solution \mathbf{x}_k is computed by using the relation of $\mathbf{x}_k = \mathbf{b} - \mathbf{r}_k$, without computing the inverse matrix of A . This leads to several variants of Lanczos algorithms, such as Orthores (A_4), Orthomin, Orthodir, etc [4], and $A_{12}, A_{13}/B_6$ [3].

Baheux-types have been developed by using the combinations of orthogonal polynomials $P_k, P_{k-1}, P_{k-2}, P_k^{(1)}, P_{k-1}^{(1)}$, and $P_{k-2}^{(1)}$, in the recurrence relationships, [4]. Note here, $P_k^{(1)}$ is called adjacent orthogonal polynomials which related to the linear function $c^{(1)}$ with the similar property as function c . Farooq-types were built from Baheux-types with addition P_{k-3} and $P_{k-3}^{(1)}$, [5].

The only one drawback that typically all Krylov-based solvers has is that they faced the breakdown which causes their convergence is unstable. It is commonly caused by the division by zero on the Lanczos formula. It is not our authority to discuss further regarding this phenomenon, however, our investigation of the model prediction is motivated by the issue of breakdown. Particularly, we are interested in looking at the pattern of the iterates generated by the Lanczos algorithms. Good study regarding this can be found in [10] and [11], where both articles discuss the prediction model by using interpolation and extrapolation methods which the model is based on the pattern persistent in the Lanczos vector sequences. This leads at least two methods called EIEMLA and MEIEMLA.

3. EIEMLA and MEIEMLA in the Solution Prediction of SLEs

One of the strategies to deal with the breakdown in Krylov-based solvers such the Lanczos-type algorithms, is the predicting solution. Let a Lanczos-type solves a non-symmetric SLEs, $A\mathbf{x} = \mathbf{b}$, where $A \in \mathbb{R}^{n \times n}$, and $\mathbf{x}, \mathbf{b} \in \mathbb{R}^n$ for a fixed number of iterations, say k , or before the Lanczos algorithm faced the breakdown. The sequence $S = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m, \dots, \mathbf{x}_k\}$ of all iterates can be visualized by using parallel coordinate system as follows.

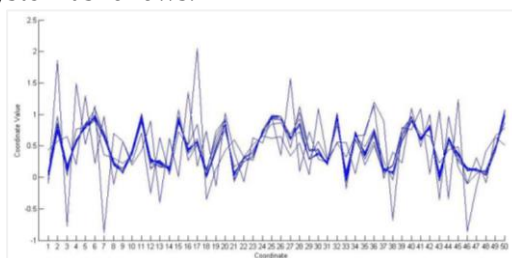


Figure 1. Parallel Coordinate System representation of iterates generated by Orthodir [9]

As can be seen from Figure 1, the bold curves are the pattern formed by some iterates with small residual norms. The idea then comes up with modeling the iterates by using interpolation method, particularly by using PCHIP (piecewise cubic hermite interpolation polynomial). The extrapolation method is then needed to get an approximate solution which preserves the properties of the sequence S . This procedure was implemented in the method called embedding interpolation and extrapolation model in Lanczos-types (EIEMLA) [9].

In EIEMLA, the interpolation is done over all entries of the first iterate, $\{x_1^{(1)}, x_2^{(1)}, \dots, x_n^{(1)}\}$, all entries of the second iterate $\{x_1^{(2)}, x_2^{(2)}, \dots, x_n^{(2)}\}$, etc. MEIEMLA, revised the EIEMLA by interpolating all iterates in the sequence at one time. Both EIEMLA and MEIEMLA predicted the approximate solution of SLEs within the range of interval $[1, s]$, where $s < k$. In this study, we use regression methods which allow us to predict out of the range, or $[1, k_1]$, for $k_1 > k$. The benefit of this kind of prediction is that we obtain a new iterate which is not the Krylov basis, particularly when it is employed in the restarting frame work, the next approximate solutions would be totally different from the Krylov basis. It becomes interesting when similar procedure is implemented in other iterative methods.

4. Support Vector Regression

Support Vector Regression (SVR) is one of main application in Support Vector Machine (SVM) to solve regression problem [12]. Following [13], for our training dataset $\{(x_1, y_1), (x_2, y_2), \dots, (x_i, y_i)\} \subset \chi \times R$, where χ denotes the spaces of the input patterns. The main goal of SVR is to find a function that has the largest deviation of the actual obtained targets for all the training data. In short, SVR would like to finds a function of that can approximate our output to an actual target, with minimum tolerance error of ε . The regression function of $f(x)$ is described as [12]:

$$f(x) = \langle w, x \rangle + b \quad (1)$$

where $w \in \chi$ and $b \in R$. The coefficients of w and b are estimated by minimizing the risk function defined on below equation:

$$\text{Minimize } \frac{1}{2} \|w\|^2 + C \sum_{i=1}^i (\xi_i + \xi_{i^*}) \quad (2)$$

$$\text{subject to } \begin{cases} y_i - \langle w, x_i \rangle - b \leq \varepsilon + \xi_i \\ \langle w, x_i \rangle + b - y_i \leq \varepsilon + \xi_{i^*} \\ \xi_i, \xi_{i^*} \geq 0 \end{cases} \quad (3)$$

where C is the constant variable that greater than 0, and ξ_i, ξ_{i^*} are slack variables to cope with otherwise infeasible constraints of the optimization problem.

5. Hybrid Restarting SVR-Lanczos

We adopt the procedure of MEIEMLA to predict the approximate solution of SLE after collecting a data set of iterates generated by Lanczos-type algorithms. In this case, we employed Orthodir algorithm. In general, this study has three stages to achieve our objective, which are collecting data set by running Orthodir algorithm, predicting the next point by using the SVR, and restarting the hybrid SVR-L based on the predicted data.

In collecting data set, we fixed a-100 iteration in each time running the Orthodir algorithm, it is assumed that the solution is found within 100 iterations (or before the breakdown occurred). Next, we collect all the 100 iterates, $\{x_1, x_2, \dots, x_k\}$, and their corresponding residual norms, $\{\|r_1\|, \|r_2\|, \dots, \|r_k\|\}$, hence used them as the training data on the SVR process. On the next stage, we used the iterate with the minimum residual norm, $\{x_m\}$, as our response variable. The idea behind this is that our prediction solution would be similar as $\{x_m\}$, but it is not the Krylov basis so that it doesn't bring the breakdown inherent. Up to this step, all the procedures is constricted under the hybrid SVR-Lanczos (SVR-L) algorithm.

One time of running the hybrid SVR-L for solving the system $Ax = b$, the new approximate solution as its product, doesn't meet the small residual norm. Therefore, putting the hybrid SVR-L in a restarting framework is necessarily done. In practice, for the last stage of the procedures of

hybrid restarting-SVR-L, the SVR-L algorithm is repeated for several times until the approximate solution met a certain tolerant. There are two benefits of employing our proposed method, first, the approximate solution resulted is better than other approximate solutions generated by the Orthodir individually, and second, the potential breakdown can be avoided. All of the process of this method is described in the Figure 2 below, while the algorithms of hybrid restarting-SVR-L is presented in Algorithms 1 and Algorithm 2.

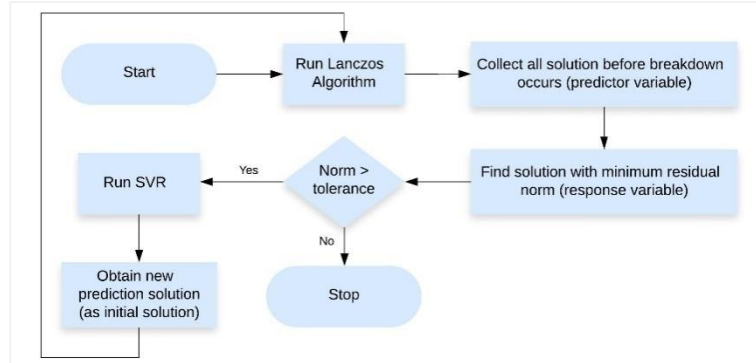


Figure 2. Process flow of hybrid restarting SVR-L

To justify our prediction of solution is valid, we evaluate the mean absolute error (MAE), mean squared error (MSE), root mean squared error (RMSE) and coefficient of determination (R-squared). The MAE, MSE, RMSE, and R-squared metrics are mainly used to evaluate the prediction error rates and model performance in regression analysis [14-16]. The above metrics can be expressed as:

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \tag{4}$$

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \tag{5}$$

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \tag{6}$$

$$R^2 = 1 - \frac{\sum (y_i - \hat{y}_i)^2}{\sum (y_i - \bar{y})^2} \tag{7}$$

where the observed value is represented as y_i , the predicted value is shown as \hat{y}_i , and the mean of the observed value is represented as \bar{y}_i .

Algorithm 1 Hybrid SVR-L

1. Fix the number of iterations to say, k , and the tolerance, ε , to 1E-13
2. Initialization. Choose \mathbf{x}_0 and \mathbf{y}_0 . Set $\mathbf{x} = \mathbf{x}_0$, $\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0$, $\mathbf{y}_0 = \mathbf{y}$, and $\mathbf{z}_0 = \mathbf{r}_0$
3. **for** $k = 0, 1, 2, \dots$ **do**
4. $\mathbf{y}_k = \mathbf{A}^T \mathbf{y}_k - \mathbf{1}$
5. $\mathbf{A}_{k+1} = \frac{\langle -\mathbf{y}_k | \mathbf{r}_k \rangle}{\langle \mathbf{y}_k | \mathbf{A} \mathbf{z}_k \rangle}$
6. $\mathbf{x}_{k+1} = \mathbf{x}_k - \mathbf{A}_{k+1} \mathbf{z}_k$
7. $\mathbf{r}_{k+1} = \mathbf{r}_k + \mathbf{A}_{k+1} \mathbf{A} \mathbf{z}_k$
8. **end for**
9. $S = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m, \dots, \mathbf{x}_k\}$
10. **for** i in each row of S **do**
11. Collect training data, $\mathbf{X} = (\mathbf{1}, \mathbf{2}, \mathbf{3}, \dots, \mathbf{k})$ and $\mathbf{Y} = (x_{1,i}, x_{2,i}, x_{3,i}, \dots, x_{k,i})$
12. Choose $kernel = 'rbf'$ and $gamma = 0.1$
13. $\hat{y}_i = predict(\mathbf{X}, \mathbf{Y})$ by using SVR
14. **end for**
15. $sol_{pred} = \hat{\mathbf{y}}$
16. $norm_{pred} = \|\mathbf{r}_{pred}\|$
17. Stop

To compare with other regression tool, we also employed the regular regression under the polynomial regression. Similar procedures of this can be found in Algorithms 3 and 4.

Algorithm 2 Hybrid Restarting-SVR-L

1. Fix the number of iterations to, say, k and the tolerance, ϵ , to 1E-13
2. Run hybrid SVR-L for k iterations
3. **while** $norm_{pred} \geq \epsilon$ **do**
4. Initialize the algorithm with

$$\mathbf{x} = sol_{pred},$$

$$\mathbf{y} = \mathbf{b} - \mathbf{Ax}.$$
5. **end while**
6. Take predicted solution $\hat{\mathbf{y}}$ as the approximate solution
7. Stop

Algorithm 3 Hybrid Regression-L

1. Fix the number of iterations to say, k , and the tolerance, ϵ , to 1E-13
2. Initialization. Choose \mathbf{x}_0 and \mathbf{y}_0 . Set $\mathbf{x} = \mathbf{x}_0$, $\mathbf{r}_0 = \mathbf{b} - \mathbf{Ax}_0$, $\mathbf{y}_0 = \mathbf{y}$, and $\mathbf{z}_0 = \mathbf{r}_0$
3. **for** $\mathbf{k} = 0, 1, 2, \dots, k$ **do**
4. $\mathbf{y}_k = \mathbf{A}^T \mathbf{y}_k - \mathbf{1}$
5. $\mathbf{A}_{k+1} = \frac{(-\mathbf{y}_k | \mathbf{r}_k)}{(\mathbf{y}_k | \mathbf{Az}_k)}$
6. $\mathbf{x}_{k+1} = \mathbf{x}_k - \mathbf{A}_{k+1} \mathbf{z}_k$
7. $\mathbf{r}_{k+1} = \mathbf{r}_k + \mathbf{A}_{k+1} \mathbf{Az}_k$
8. **end for**
9. $\mathbf{S} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m, \dots, \mathbf{x}_k\}$
10. **for** \mathbf{i} in each row of \mathbf{S} **do**
11. Set $\mathbf{X} = (1, 2, 3, \dots, k)$ as predictor variables and $\mathbf{Y} = (x_{1,i}, x_{2,i}, x_{3,i}, \dots, x_{k,i})$ as response variable
12. Create linear regression model and fit it using polynomial features with $degree = 4$
13. $\hat{\mathbf{y}}_i = \text{predict}(\mathbf{X}, \mathbf{Y})$ by using polynomial regression
14. **end for**
15. $sol_{pred} = \hat{\mathbf{y}}$
16. $norm_{pred} = \|\mathbf{r}_{pred}\|$
17. Stop

Algorithm 4 Hybrid Restarting-Regression-L

1. Fix the number of iterations to say, k , and the tolerance, ϵ , to 1E-13.
2. Run hybrid Regression-L for k iterations
3. **while** $norm_{min} \geq \epsilon$ **do**
4. Initialize the algorithm with

$$\mathbf{x} = sol_{min},$$

$$\mathbf{y} = \mathbf{b} - \mathbf{Ax}.$$
5. **end while**
6. Take predicted solution $\hat{\mathbf{y}}$ as the approximate solution
7. Stop.

6. Numerical Results

We solved different size problems of SLEs $\mathbf{Ax} = \mathbf{b}$, ranging from dimension 1,000 to 50,000 with the maximum iteration for each cycle is 100. The matrix A of the system is obtained as a result in discretizing the PDE equations $-\frac{\partial^2}{\partial x^2} - \frac{\partial^2}{\partial y^2} = f$ in the closed region, whereas the vector \mathbf{b} is formed so that $\mathbf{b} = \mathbf{Ax}$, where $\mathbf{x} = [1 \ 2 \ 3 \ \dots \ n]^T$. The implementation of all algorithms of this study was using Python under Windows.

All of results of SVR-L are presented in Table 1 and Table 2, also the performance of this algorithm is visualized in Fig. 3. We can see from Table 1 that there is improvement of residual

norm when the Lanczos is combined with SVR. Moreover, when the hybrid SVR-L was put in the restarting framework, the performance of the algorithm improved significantly (see Table 2).

Table 1. Performance of Hybrid SVR-L to improve the individual Lanczos/Orthodir

Dim (n)	Lanczos $\ r_k\ $	Hybrid SVR-L $\ r_{prediction}\ $
1000	1.72E+04	8.28E+01
2000	5.48E+03	2.69E+02
3000	7.64E+03	2.89E+03
4000	7.11E+03	1.68E+03
5000	3.48E+04	1.79E+03
6000	2.76E+03	2.36E+03
7000	8.50E+04	1.33E+03
8000	2.18E+03	6.23E+03
9000	4.04E+05	2.58E+03
10000	3.06E+05	5.48E+03
20000	4.75E+05	3.97E+03
30000	1.91E+04	3.76E+03
40000	3.53E+10	8.27E+03
50000	2.35E+04	7.27E+03

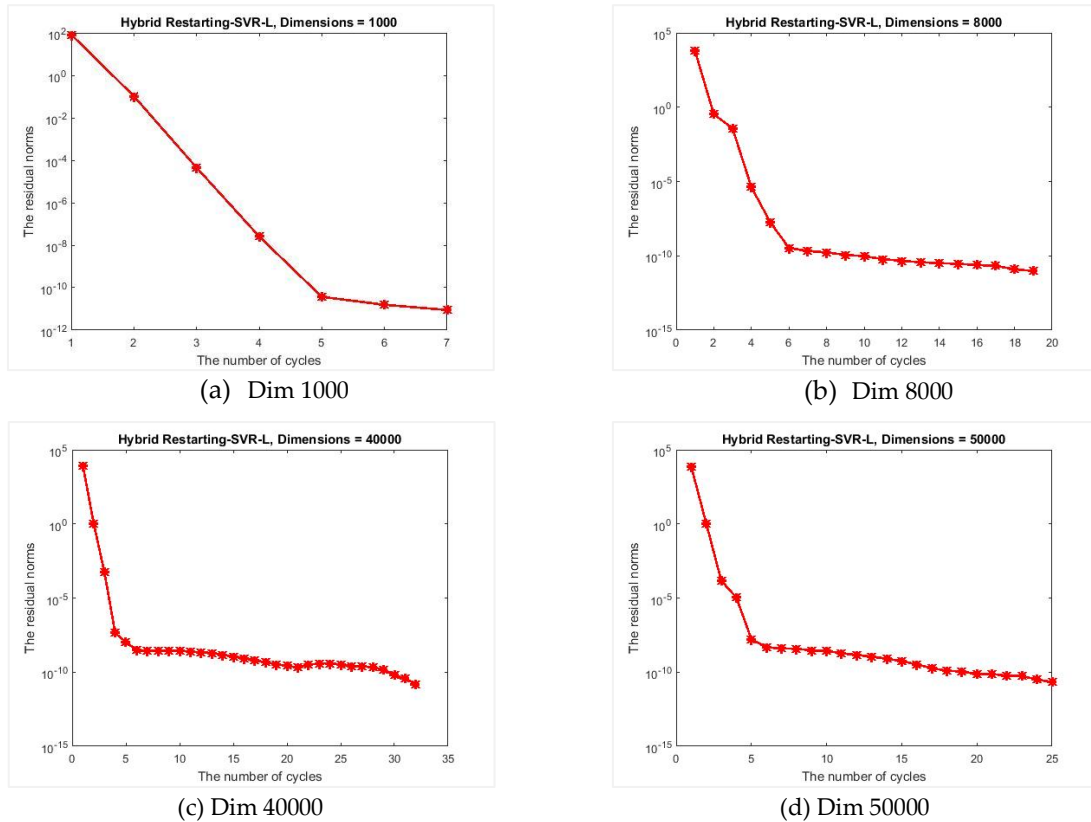


Figure 3. Performance of Hybrid Restarting-SVR-L of several dimensions

Table 2. The validation of prediction of the solution generated by Hybrid Restarting- SVR-L

Dim (n)	$\ r_{prediction}\ $	Time (s)	Cycles	MAE	MSE	RMSE
1000	8.92E-12	4.844s	7	1.68E-13	8.98E-26	3.00E-13
2000	9.81E-12	24.624s	21	5.67E-14	2.98E-26	1.73E-13
3000	9.86E-12	55.870s	33	1.75E-14	1.58E-26	1.26E-13
4000	8.48E-12	56.148s	25	4.62E-15	4.29E-27	6.55E-14
5000	3.00E-11	138.711s	50	5.55E-14	5.50E-26	2.34E-13
6000	0.00E+00	88.288s	26	0.00E+00	0.00E+00	0.00E+00
7000	0.00E+00	91.864s	23	0.00E+00	0.00E+00	0.00E+00
8000	9.49E-12	89.009s	19	2.22E-15	2.07E-27	4.55E-14
9000	5.30E-12	107.993s	21	2.02E-16	1.84E-28	1.36E-14

10000	2.39E-11	151.214s	27	1.28E-14	1.19E-26	1.09E-13
20000	7.50E-12	399.096s	35	9.10E-17	1.65E-28	1.29E-14
30000	0.00E+00	446.153s	26	7.40E-21	1.64E-36	1.28E-18
40000	0.00E+00	753.027s	33	0.00E+00	0.00E+00	0.00E+00
50000	0.00E+00	742.043s	26	2.22E-21	2.47E-37	4.97E-19

Similar story as the performance of hybrid regression-L (Table 3) and hybrid restarting-regression-L (Table 4). We can see here that the hybrid SVR-L and its hybrid restarting-SVR-L version performed better than the hybrid regression-Lanczos.

Table 3. Performance of Hybrid Regression-Lanczos to improve the individual Lanczos/Orthodir

Dim (<i>n</i>)	Lanczos $\ r_k\ $	Regression-Orthodir $\ r_{prediction}\ $
1000	1.72E+04	7.62E+01
2000	5.48E+03	2.19E+02
3000	7.64E+03	7.09E+03
4000	7.11E+03	2.90E+03
5000	3.48E+04	6.88E+03
6000	2.76E+03	1.31E+04
7000	8.50E+04	1.79E+03
8000	2.18E+03	1.02E+04
9000	4.04E+05	8.81E+03
10000	3.06E+05	1.94E+04
20000	4.75E+05	2.09E+05
30000	1.91E+04	1.07E+04
40000	3.53E+10	6.37E+05
50000	2.35E+04	2.61E+03

Table 4. The validation of prediction of the solution generated by Hybrid Restarting- Regression-Lanczos

Dim (<i>n</i>)	$\ r_{prediction}\ $	Time (s)	Cycles	MAE	MSE	RMSE
1000	1.74E-12	17.713s	17	1.14E-15	1.29E-28	1.14E-14
2000	2.05E-12	102.717s	55	6.25E-16	8.40E-29	9.17E-15
3000	1.87E-12	102.543s	36	3.79E-16	6.89E-29	8.30E-15
4000	1.56E-10	103.564s	27	5.49E-13	1.34E-24	1.16E-12
5000	2.33E-10	102.643s	28	6.18E-13	9.18E-25	9.58E-13
6000	2.90E-10	119.308s	24	6.30E-13	1.19E-24	1.09E-12
7000	2.71E-10	158.179s	26	6.52E-13	1.60E-24	1.27E-12
8000	2.60E-10	370.497s	57	5.65E-13	1.70E-24	1.30E-12
9000	2.73E-10	279.691s	34	4.14E-13	1.22E-24	1.10E-12
10000	2.78E-10	283.855s	33	2.68E-13	7.34E-25	8.57E-13
20000	2.46E-10	632.926s	44	5.97E-14	3.81E-25	6.18E-13
30000	2.56E-10	870.312s	42	2.48E-13	1.94E-25	4.40E-13
40000	2.03E-10	1096.711s	41	1.56E-14	6.71E-26	2.59E-13
50000	1.98E-10	1714.899	55	8.00E-15	5.59E-026	2.37E-13

In this section, we also present the comparison of the hybrid restarting SVR-L and hybrid restarting regression-Lanczos

We also compared both of our algorithms with previous study that we adopted, which is MEIEMLA. The comparison of these three algorithms can be seen in Fig. 5. We can see here that the hybrid restarting-SVR-L performed the best in the term of residual norm, where restarting MEIEMLA needs more cycles to obtain a good prediction, compared with the two hybrids.

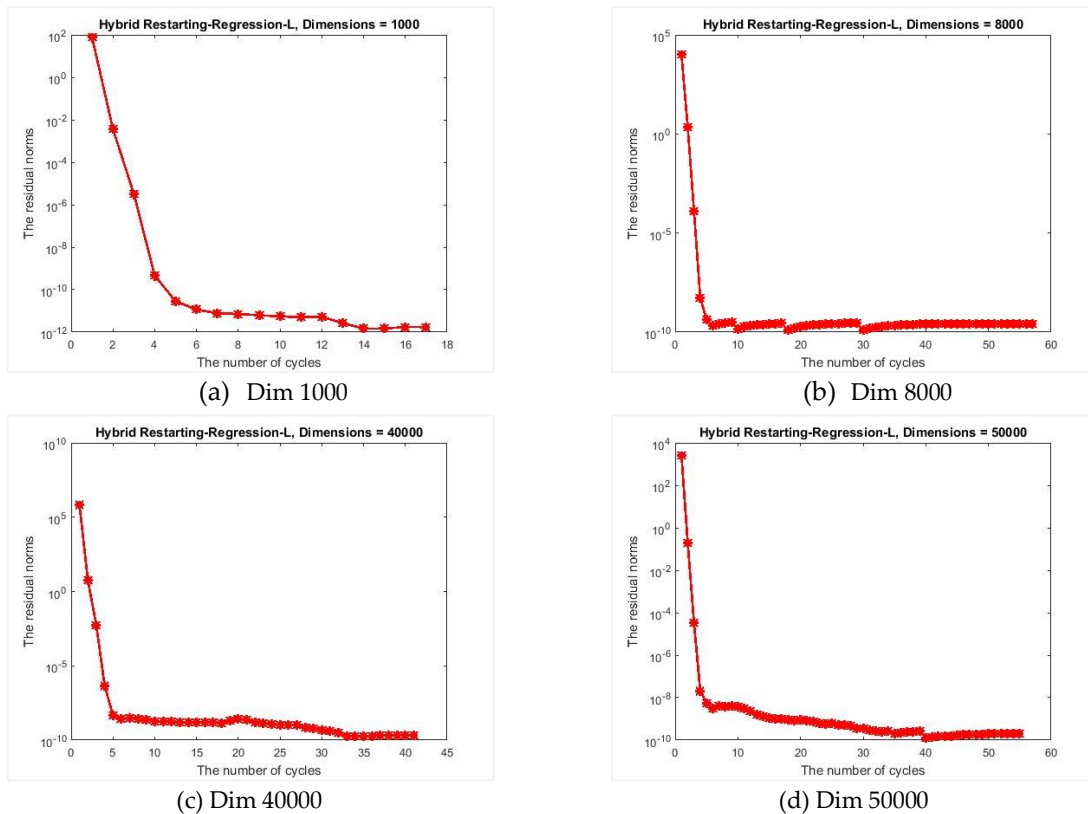


Figure 4. Performance of Hybrid Restarting-Regression-L of several dimensions

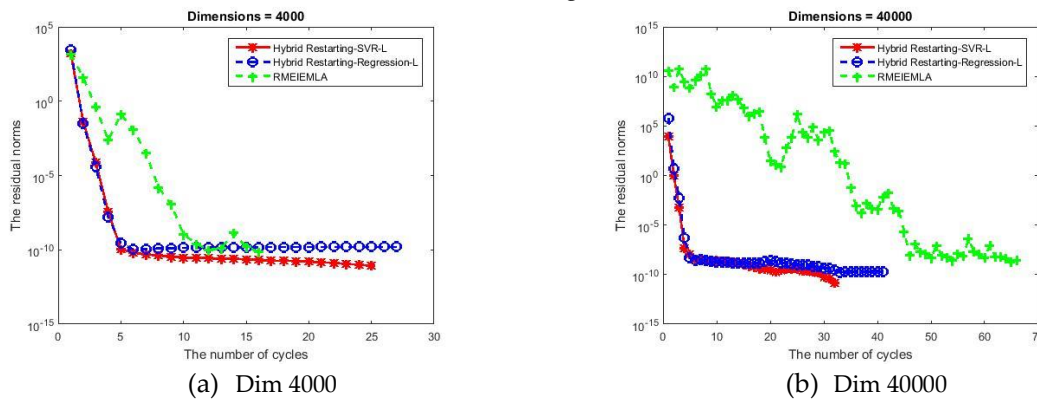


Figure 5. Comparison Results

7. Discussion

According to Tables 1 and 2, overall, the proposed methods of hybrid SVR-L and hybrid Regression-L were able to reduce the residual norms of the approximate solutions generated by the original Lanczos method. The significant results appeared, for instance, when solving 1000 dimensions of SLEs, where the hybrid SVR-L reduced from $1.72E+04$ to $8.28E+01$. Other SLE problems such as dimensions 20000, 30000, and 40000 were also reduced significantly. This situation, however, did not occur when using hybrid Regression-L.

The use of restarting framework to speed up the convergence of the hybrids, worked properly. It can be seen in Tables 3 and 4 for restarting hybrid SVR-L and restarting hybrid Regression-L, respectively. These results were compared with the restarting MEIEMLA. Overall, the prediction of approximate solutions generated by both hybrid restarting-SVR-L and hybrid restarting-regression-L algorithms, were more accurate than the one produced by restarting MEIEMLA. This is shown by the residual norms which were smaller than the residual norms generated by restarting MEIEMLA. This comparison is clearly visible in Figure 5.

However, from our observations, some drawbacks found for the high dimension problems, namely that the larger the dimensions that we calculate, the longer computational time to reach our

desired tolerance error, since we undergo slow decline of error. These are appeared in Figure 3 (d) and Figure 4 (c) and (d). One way to improve this issue is, perhaps, to reduce the dimension of our SLEs which can help shorten the computational time.

8. Conclusion

We have implemented the hybrid restarting-SVR-L in predicting the new approximate solution of SLEs. This innovation showed a good performance of reducing the residual norms when the individual Lanczos/Orthodir was used to solve the SLE problems. We have also implemented the hybrid restarting-regression-L with the same purposes. Based on the numerical results, overall, the hybrid restarting-SVR-L was more accurate in obtaining the approximate solutions, compared with hybrid restarting-regression-L and MEIEMLA. It also showed the best performance in term of efficiency; it consistently took the shortest time on all problems.

References

- [1] Shwartz S.S and David S.B. (2014). Understanding Machine Learning: From Theory to Algorithms. Cambridge University Press, USA.
- [2] Khun M and Johnson K. (2013). Applied Predictive Modeling. Springer, USA.
- [3] Brezinski C. and Sadok H. (1993). Lanczos-type Algorithms for Solving Systems of Linear Equations. In Journal of Applied Numerical Mathematics, Vol. 11, pg. 443-473. Available: [https://doi.org/10.1016/0168-9274\(93\)90087-8](https://doi.org/10.1016/0168-9274(93)90087-8).
- [4] Baheux C. (1995). New Implementation of Lanczos Method. In Journal of Computational and Applied Mathematics, Vol. 57, pg. 3-15. Available: [https://doi.org/10.1016/0377-0427\(93\)E0230-I](https://doi.org/10.1016/0377-0427(93)E0230-I).
- [5] Farooq M. and Salhi A. (2012). New Recurrence Relationships between Orthogonal Polynomials which Lead to New Lanczos-type Algorithms. In Journal of Prime Research in Mathematics, Vol. 8, pg. 61-75. Available: <http://arXiv.1403.0323>.
- [6] Farooq M. and Salhi A. (2013). A Preemptive Restarting Approach to Beating Inherence Instability. In Iranian Journal of Science and Technology Transaction a Science, Vol. 12, pg. 349-358. Available: <https://www.sid.ir/en/journal/ViewPaper.aspx?ID=355408>
- [7] Farooq M. and Salhi A. (2014). A Switching Approach to Avoid Breakdown in Lanczos-type Algorithms. In Applied Mathematics and Information Sciences, Vol. 8, pg. 2161-2169. Available: <http://10.12785/amis/080509>.
- [8] Maharani and Salhi A. (2015). Restarting from Specific Points to Cure Breakdown in Lanczos-type Algorithms. In Journal of Mathematical and & Fundamental Science, Vol. 47, pg. 167-184. Available: <http://10.5614/j.math.fund.sci.2015.47.2.5>.
- [9] Maharani., Salhi A and Suharto R.A. (2018). Introduction of Interpolation and Extrapolation Model in Lanczos-type Algorithms A13/B6 and A13/B3 to Enhance their Stability. In Journal of Mathematical and & Fundamental Science, Vol. 50, pg. 148-165. Available: <http://10.5614/j.math.fund.sci.2018.50.2.4>.
- [10] Maharani and Salhi A. (2019). Lanczos-type Algorithms with Embedded Interpolation and Extrapolation Models for Solving Large Scale Systems of Linear Equations. In International Journal of Computing Science and Mathematics, Vol. 10, pg.429-442. Available: <http://10.1504/IJCSM.2019.103675> .
- [11] Maharani and Salhi A. (2019). RMEIEMLA: The Recent Advance in Improving the Robustness of Lanczos-type Algorithms. In Proceeding AIP Conference Proceedings, Vol. 2138. Available: <http://doi.org/10.1063/1.5121046> .
- [12] Wang L. (2005). Support Vector Machines: Theory and Applications. Springer: Netherland.
- [13] Smola A and Scolkopf B. (2003). A Tutorial on Support Vector Regression. In Journal of Statistics and Computing, Vol. 14, pg. 199-222. Available: <https://doi.org/10.1023/B:STCO.0000035301.49549.88> .
- [14] Chai T and Draxler R.R. (2014). Root Mean Square Error (RMSE) or Mean Absolute Error (MAE)? –Arguments Against Avoiding RMSE in The Literature. In Journal of Geosci. Model Dev, Vol. 7, pg. 1247-1250. Available: <https://10.5194/gmd-7-1247-2014> .
- [15] Agustina S.D., Mustakim, Okfalisa, Bella C. and Ramadhan M.A. (2018). Support Vector Regression Algorithm Modeling to Predict the Availability of Foodstuff in Indonesia to Face the Demographic Bonus. In Journal of Physics: Conf. Series 1028. Available: <https://iopscience.iop.org/article/10.1088/1742-6596/1028/1/012240>.
- [16] Willmott C.J and Kenji M. (2005) Advantages of The Mean Absolute Error (MAE) Over the Root Mean Square Error (RMSE) in Assessing Average Model Performance. Article in Climate Research. Pg. 30-79. Available: <http://10.3354/cr030079>.

