A Method of the Lagrange Polynomial Interpolation with Weighting Factors for Reducing Computational Time

*Uthai Prasopchingchana**
Department of Mechanical Engineering, Faculty of Engineering,
Chulalongkorn University, Bangkok, 10330, Thailand.
**Uthai.P@Student.chula.ac.th*

ABSTRACT

This paper proposes a novel extrapolation method known as the Lagrange polynomial interpolation with weighting factors (LPI-WF) to determine initial guess values in each time step for solving equation systems of transient problems using iterative methods. The LPI-WF method is developed from the Lagrange polynomial interpolation to determine the direct extrapolation values and multiply them with the weighting factors. The weighting factors are calculated by considering the number of the previous time steps involved in the extrapolation and the duration between the present time step and the previous time steps. Thus, the LPI-WF method is proper for use with high-order temporal schemes. The key advantages of the LPI-WF method are that the computational time required to achieve the steady-state condition of the transient problems is reduced and the computation codes with the LPI-WF method is more stable than without the LPI-WF method at high time step values. A performance test of the LPI-WF method is carried out by comparing the computational time based on the "lid-driven cavity flow" problem for Reynolds numbers of 1000 and 5000. The test result shows that the computational time of the problem when the LPI-WF method is adopted can be reduced up to 10.46 % compared to the conventional method.

Keywords: *extrapolation, initial guess value, iterative method, Lagrange polynomial interpolation, weighting factors.*

ISSN 1823-5514, eISSN 2550-164X
Received for review: 2020-03-18
Accepted for publication: 2020-10-16
Published: 2021-01-15

Nomenclature

b	Width of the cavity, m.
EWF	Extrapolation weighting factor.
$EWFN$	Individual extrapolation weighting factor.
LIP	Lagrange interpolating polynomial.
LPI	Lagrange polynomial interpolation.
LPI-WF	Lagrange polynomial interpolation with weighting factors.
LTP	Temporal coefficient of the LPI method.
p	Pressure, Pa.
Re	Reynolds number.
SIMPLE	Semi-implicit method for pressure linked equations.
$SUMEWFN$	Summation of the individual extrapolation weighting factors.
$SUMWD$	Summation of the temporal individual weighting differences, s.
t	Time, s.
TT	Temporal difference value, s.
U	Velocity of the cavity lid, m/s.
u	Velocity component in the x direction, m/s.
v	Velocity component in the y direction, m/s.
WD	Temporal weighting difference, s.
WDN	Temporal individual weighting difference, s.
x	Cartesian coordinate in the horizontal direction of the cavity, m.
y	Cartesian coordinate in the vertical direction of the cavity, m.
ϕDEV	Direct extrapolation value.
ϕEV	Extrapolation value.
Greek symbol	
μ	Viscosity of fluid, kg/(s·m).
ρ	Density of fluid, kg/m^3.
ϕ	Solution in the previous time step.
ψ	Streamline.
ω	Vorticity.
Superscript	
**	Dimensionless.
Subscript	
ntmax	Maximum number of time steps.
max	Maximum.
min	Minimum.

Introduction

The iterative method is a popular method used to solve equation systems obtained from discrete methods, such as the finite difference method [1]-[3], the finite volume method [4, 5], the finite element method [6]-[9], the lattice Boltzmann method [1, 10], and the spectral element method [11, 12], due to the ease of code development and low computational time and memory. For solving transient problems, the use of an explicit iterative method has a limitation of time step values. Thus, properly imposing initial guess values in each time step is important to reduce computational time.

Extrapolation is a popular technique used to reduce the computational time of transient problems. It helps to determine the initial guess values of variables in the equations using the solutions of the previous time steps. In recent decades, several extrapolation methods have been proposed. The deformation gradient extrapolation method was proposed by Rashid [8]. This method was used to provide initial predictors for the next time step in the solutions of the large deformation finite element analysis. Three numerical examples showed that the use of the method to calculate the initial predictors could reduce the numbers of iterations required to achieve convergence. Also, the reliability and robustness of the equilibrium search of the solutions were improved. Markovinovic and Jansen [2] used the reduced-order models to determine initial guessed values for accelerating the solution convergence. They achieved 67 % maximum reduction in the computational time. Leemput et al. [10] employed the polynomial backward extrapolation to prescribe the initial guessed values for one-dimensional advection solved by using the lattice Boltzmann method. Liu et al. [13] proposed a methodology to improve the computational time of unsteady flow simulation called the dynamic mode extrapolation initial condition method. The method was applied by expressing the function in terms of time and space. The robustness of the method in reducing the computational time was confirmed by comparison test with the Lagrange extrapolation initial condition and the natural initial condition. The comparison results indicated that the method achieved a high computational time reduction for all cases. Moreover, extrapolation techniques proposed in [3]-[5], [7, 11, 12] were adopted for the purpose of reduction of computational time of many scientific, engineering, and economic problems. Some investigative studies [2, 3, 7, 8, 11] obtained positive results, whereas some [10, 12] could not verify any advantages of the extrapolation methods.

This article aims to present a new extrapolation method called LPI-WF, which helps to determine the initial guess values of each time step of the iterative methods for solving equation systems of transient problems approaching the steady-state condition. Furthermore, a performance test of the LPI-WF method is carried out with the well-known problem, “lid-driven cavity flow,” for *Re* of 1000 and 5000.

Details of the LPI-WF Method

The LPI-WF method is developed for solving equation systems of transient problems using iterative methods. The LPI-WF method accelerates solution convergence by applying initial guess values in each time step calculated from the solutions of the previous time steps. The LPI-WF method uses the LPI to extrapolate values directly from solutions of the previous time steps. The *EWF*s are calculated from time step values and numbers of time steps. The ϕEV of the LPI-WF method is the summation of the products of the *EWF* and the ϕDEV. Thus, ϕEV can be expressed as follows:

$$\phi EV = \sum_{n=1}^{ntmax-1} (EWF_n \phi DEV_n), \tag{1}$$

where *ntmax* is the maximum number of time steps required to solve the problems. A schematic diagram of the LPI-WF method is shown in Figure 1 when *ntmax* = 4. The *EWF* for each time step *n* can be obtained from

$$EWF_n = \frac{EWFN_n}{SUMEWFN}. \tag{2}$$

The *EWFN* for each time step *n* and the *SUMEWFN* are defined as follows:

$$SUMEWFN = \sum_{n=1}^{ntmax-1} (EWFN_n), \tag{3}$$

$$EWFN_n = \frac{WDN_n}{SUMWD}. \tag{4}$$

The *WDN* for each time step *n* and the *SUMWD* are defined as follows:

$$SUMWD = \sum_{n=1}^{ntmax-1} \left(\sum_{ntd=1}^{n} (WD_{n-ntd+1}) \right), \tag{5}$$

$$WDN_n = \sum_{ntd=1}^{n} (SUMWD - WD_{n-ntd+1}). \tag{6}$$

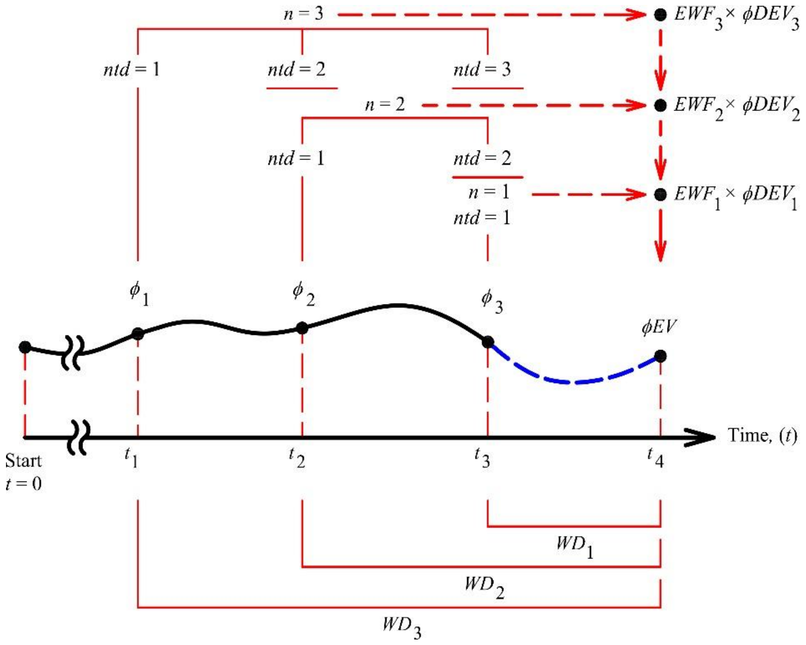


Figure 1: Schematic diagram of the LPI-WF method when *ntmax* = 4.

The *WD* is expressed as follows:

$$WD_{n-ntd+1} = t_{ntmax} - t_{ntmax-(n-ntd+1)}, \tag{7}$$

for $n = 1, 2, \ldots, ntmax - 1$ and $ntd = 1, 2, \ldots, n$

where *t* is the program time of each time step in transient problems. The *ϕDEV* can be obtained using the LPI method. Thus,

$$\phi DEV_n = \sum_{ntd=1}^{n} LTP_{n,ntd}\, \phi_{(ntmax-1)-n+ntd}. \tag{8}$$

The *LTP* is given as:

$$LTP_{1,1} = 1, \text{ for } n = 1,$$

$$LTP_{n,ntd} = \prod_{\substack{ntt=1 \\ ntt \neq ntd}}^{n} \frac{\left(-TT_{n,ntt}\right)}{\left(TT_{n,ntd} - TT_{n,ntt}\right)}, \quad \text{for } n = 2, 3, \dots, ntmax - 1. \qquad (9)$$

The *TT*s are computed from:

$$TT_{n,ntt} = t_{(ntmax-1)-n+ntt} - t_{ntmax}, \qquad (10)$$

$$TT_{n,ntd} = t_{(ntmax-1)-n+ntd} - t_{ntmax}, \qquad (11)$$

for $n = 1, 2, \dots, ntmax - 1$, $ntd = 1, 2, \dots, n$,
$ntt = 1, 2, \dots, n$, $\quad and \;\; ntt \neq ntd$.

From the preceding texts, the procedures for determining the initial guess values from the solutions of the previous time steps using the LPI-WF method are as follows:

1. Calculate the *WD* from (7).
2. Determine the *WDN* from (6).
3. Find the *SUMWD* using (5).
4. Compute the *EWFN* from (4).
5. Calculate the *SUMEWFN* from (3).
6. Compute the *EWF* using (2).
7. Determine the *TT*s from (10) and (11).
8. Calculate the *LTP* of the LPI method from (9).
9. Find the *ϕDEV*s from Equation (8).
10. Compute the *ϕEV*s from (1). The calculated values of *ϕEV*s are adopted as the initial guess values for iterative methods.

Code Validation

To prove the efficiency of the LPI-WF method in computational time reduction for calculating the initial guess values in each time step of the transient problems using iterative methods, a new code is developed for the performance test of the LPI-WF method. A well-known problem called "lid-driven cavity flow," for *Re* of 1000 and 5000 is selected for the performance test. This is because these numbers are the representatives of low and high laminar flows. Moreover, the numbers are used in the recognized solutions for code validation presented by Boltella and Peyret [14] and Bruneau and Saad [15]. The finite volume method with the LIP scheme [16] and the SIMPLE algorithm are employed to discretize the partial differential equations of the problem and to couple the continuity equation and the momentum equations, respectively. Code validation is carried out to ensure that the code delivers correct solutions.

The comparisons of solutions computed from the code with the benchmark and published numerical solutions of the problem reported in [14, 15] are used for the code validation. The code is based on a transient condition. However, the solutions reported in [14, 15] are the steady-state condition solutions. Therefore, a convergent criterion of the code for the final solutions is that the relative residuals of the present time step solutions and the previous time step solutions must be less than or equal to 10^{-5}, and must achieve the criterion consecutively at least ten times. Also, the criterion ensures that the solutions converged to the steady-state condition. A dimensionless time step value Δ*t*** = 0.001 (using *t*** = Δ*t* |*U*|/*b*,) and non-uniform mesh sizes of 150 × 150 and 200 × 200 are employed for computing the *Re* of 1000 and 5000, respectively. The performance test of the LPI-WF method is carried out on four categories as detailed in Table 1. Also, the code validation is performed on four categories simultaneously. The conventional method shown in Table 1 is a means whereby the initial guess values are equally imposed on the solutions in the latest previous time step.

"Lid-driven cavity flow" is a classical problem in Computational Fluid Dynamics, which is employed by numerous researchers [17]-[24] to verify or validate their methods or codes. Details of the problem are shown in Figure 2. Fluid flow in a square cavity is defined by the continuity equation and the momentum equations. Furthermore, the gravitational acceleration is neglected.

Table 1: Details of each category for the performance test of the LPI-WF method.

Category	*ntmax*	Extrapolation method
A	2	Conventional method
B	4	Conventional method
C	4	LPI method
D	4	LPI-WF method

For two-dimensional simulation, the equations can be written as:

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0, \tag{12}$$

$$\rho\left(\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y}\right) = -\frac{\partial p}{\partial x} + \mu\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right), \tag{13}$$

$$\rho\left(\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y}\right) = -\frac{\partial p}{\partial y} + \mu\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right), \tag{14}$$

where ρ, and μ are the density, and viscosity of the incompressible fluid in the square cavity, respectively.

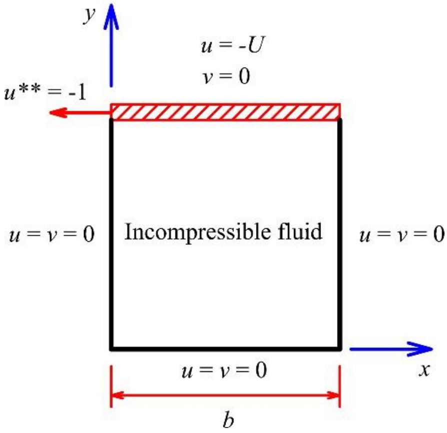
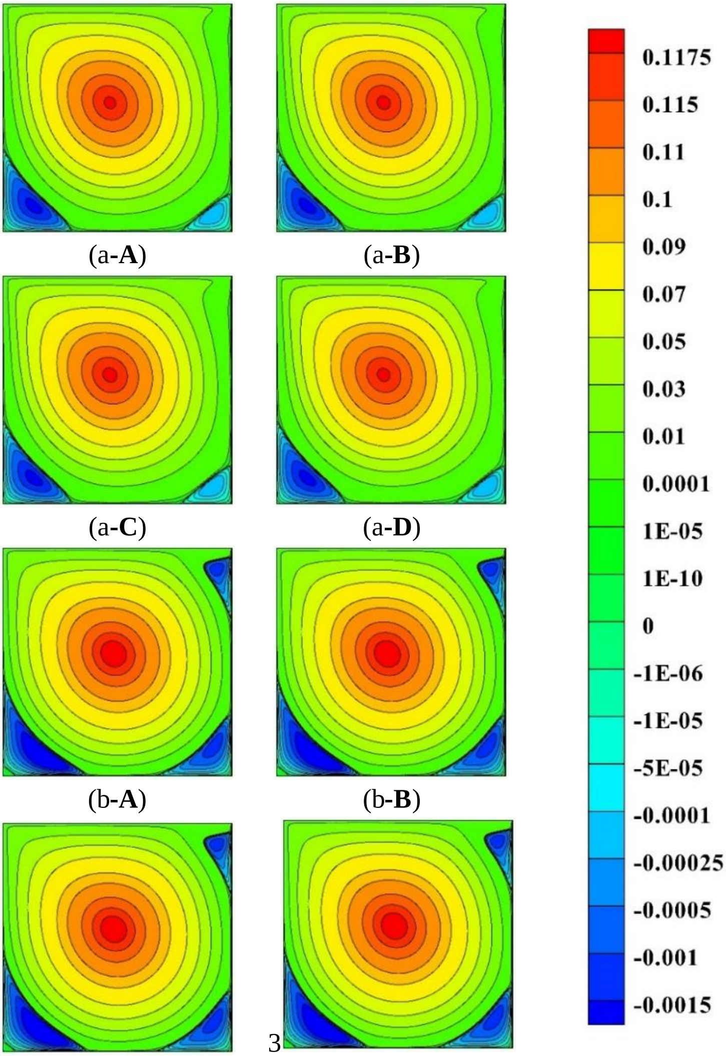


Figure 2: Details of the “lid-driven cavity flow” problem.

A summary of computational procedures is as follows:

1. Input the fluid properties into the code.
2. Specify *Re* and the velocity of the cavity lid.
3. Calculate the width of the cavity.
4. Input the initial values of the variables in the problem,
5. Compute the velocities and pressures of the fluid from (12), (13), and (14) by using the finite volume method with the LIP scheme and the SIMPLE algorithm to generate discrete equation systems. Then, solve the discrete equation systems using an iterative method.
6. Check the convergent criterion for each time step. If the solutions do not meet the convergent criterion, return to step 5. until the convergent criterion is achieved.
7. Determine the initial guessed values of the variables in the problem for computing in the next time step by using the extrapolation method (i.e., the conventional method, the LPI method, the LPI-WF method).
8. Repeat step 5. to 6.
9. Check the convergent criterion for termination of the code. If the solutions do not reach the convergent criterion, return to step 7. until the convergent criterion is achieved.

Figure 3 and Figure 4 display the contours of the ψ and ω computed from the code based on the “lid-driven cavity flow” problem for *Re* of 1000 and 5000. The patterns of the ψ and ω shown in Figure 3 and Figure 4 are similar to the patterns of the ψ and ω shown in [14, 15].



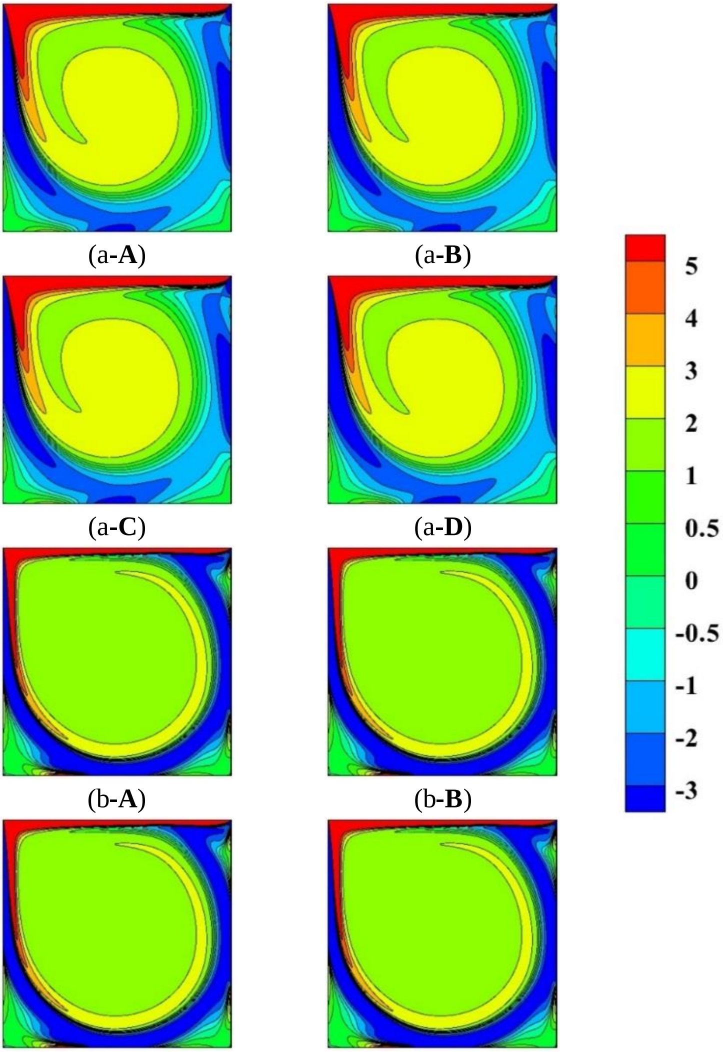
(a-**A**) (a-**B**)

(a-**C**) (a-**D**)

(b-**A**) (b-**B**)

(b-**C**) (b-**D**)

Figure 3: Contours of the ψ computed from the code based on the "lid-driven cavity flow" problem: (a-**A**) category **A** for *Re* = 1000, (a-**B**) category **B** for *Re* = 1000, (a-**C**) category **C** for *Re* = 1000, (a-**D**) category **D** for *Re* = 1000, (b-**A**) category **A** for *Re* = 5000, (b-**B**) category **B** for *Re* = 5000, (b-**C**) category **C** for *Re* = 5000 and (b-**D**) category **D** for *Re* = 5000.



(a-**A**) (a-**B**)

(a-**C**) (a-**D**)

(b-**A**) (b-**B**)

(b-**C**) (b-**D**)

Figure 4: Contours of the ω computed from the code based on the "lid-driven cavity flow" problem: (a-**A**) category **A** for *Re* = 1000, (a-**B**) category **B** for *Re* = 1000, (a-**C**) category **C** for *Re* = 1000, (a-**D**) category **D** for *Re* = 1000, (b-**A**) category **A** for *Re* = 5000, (b-**B**) category **B** for *Re* = 5000, (b-**C**) category **C** for *Re* = 5000 and (b-**D**) category **D** for *Re* = 5000.

The dimensionless horizontal and vertical velocities (i.e., $u^{**} = u/|U|$ and $v^{**} = v/|U|$) on the dimensionless vertical and horizontal center lines (i.e., $y^{**} = y/b$ and $x^{**} = x/b$) of the cavity, respectively, are selected for comparison based on the "lid-driven cavity flow" problem for $Re = 1000$. The comparison between the dimensionless velocities computed from the code of this work and the dimensionless velocities reported in [14, 15] is shown in Table 2 and Table 3.

Table 2: Comparison between the dimensionless horizontal velocities computed from the code and the dimensionless horizontal velocities reported in [14, 15] based on the "lid-driven cavity flow" problem for $Re = 1000$.

y^{**}	u^{**}					
	Present work				[14]	[15]
	A	**B**	**C**	**D**		
1.0000	-1.0000000	-1.0000000	-1.0000000	-1.0000000	-1.0000000	-1.00000
0.9766	-0.6617013	-0.6620655	-0.6623109	-0.6619848	-0.6644227	
0.9688	-0.5776700	-0.5781232	-0.5784306	-0.5780242	-0.5808359	-0.58031
0.9609	-0.5133807	-0.5139125	-0.5142733	-0.5137946	-0.5169277	
0.9531	-0.4684216	-0.4690181	-0.4694226	-0.4688833	-0.4723329	-0.47239
0.8516	-0.3332189	-0.3336413	-0.3339621	-0.3335436	-0.3372212	
0.7344	-0.1867607	-0.1867906	-0.1868397	-0.1867802	-0.1886747	-0.18861
0.6172	-0.0563917	-0.0564056	-0.0564167	-0.0564001	-0.0570178	
0.5000	0.0619424	0.0617708	0.0616691	0.0618117	0.0620561	0.06205
0.4531	0.1076536	0.1074753	0.1073726	0.1075166	0.1081999	
0.2813	0.2779799	0.2781329	0.2782668	0.2780928	0.2803696	0.28040
0.1719	0.3849464	0.3854361	0.3857839	0.3853204	0.3885691	
0.1016	0.2964872	0.2972227	0.2977117	0.2970553	0.3004561	0.30029
0.0703	0.2198370	0.2204753	0.2208919	0.2203312	0.2228955	
0.0625	0.1995407	0.2001395	0.2005284	0.2000049	0.2023300	0.20227
0.0547	0.1787697	0.1793253	0.1796841	0.1792022	0.1812881	
0.0000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.00000

The dimensionless horizontal and vertical velocities computed from the code with category **A** yield the highest maximum differences compared to the solutions reported in [14, 15]. The highest maximum differences of the dimensionless horizontal velocity are 1.39 % and 1.35 % compared to the solutions reported in [14, 15] on $y^{**} = 0.0547$ and $y^{**} = 0.0625$, respectively. Thus, the highest maximum differences of the dimensionless vertical velocity are 1.49 % and 1.45 % compared to the solutions reported in [14, 15] on $x^{**} = 0.9375$ and $x^{**} = 0.9297$, respectively. Besides, the maximum differences between the solutions, dimensionless horizontal velocity, computed from the code with category **D**, and the solutions reported in [14, 15] are 1.15 % and

1.12 % on $y^{**} = 0.0547$ and $y^{**} = 0.0625$, respectively. Also, the maximum differences between the solutions, dimensionless vertical velocity, computed from the code with category **D**, and the solutions reported in [14, 15] are 1.28 % and 1.24 % on $x^{**} = 0.9375$ and $x^{**} = 0.9297$, respectively.

Table 3: Comparison between the dimensionless vertical velocities computed from the code and the dimensionless vertical velocities reported in [14, 15] based on the “lid-driven cavity flow” problem for $Re = 1000$.

x^{**}	v^{**}					
	Present work				[14]	[15]
	A	**B**	**C**	**D**		
0.0000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.00000
0.0312	-0.2251103	-0.2254315	-0.2256545	-0.2253564	-0.2279225	
0.0391	-0.2903928	-0.2907894	-0.2910641	-0.2906971	-0.2936869	-0.29330
0.0469	-0.3517003	-0.3521544	-0.3524685	-0.3520490	-0.3553213	
0.0547	-0.4065557	-0.4070475	-0.4073865	-0.4069337	-0.4103754	-0.41018
0.0937	-0.5229326	-0.5233599	-0.5236483	-0.5232631	-0.5264392	
0.1406	-0.4236065	-0.4239931	-0.4242531	-0.4239061	-0.4264545	-0.42634
0.1953	-0.3168067	-0.3172187	-0.3175008	-0.3171248	-0.3202137	
0.5000	0.0261299	0.0259989	0.0258801	0.0260336	0.0257995	0.02580
0.7656	0.3228789	0.3230505	0.3232078	0.3230060	0.3253592	
0.7734	0.3314039	0.3316044	0.3317819	0.3315510	0.3339924	0.33398
0.8437	0.3726714	0.3732945	0.3737411	0.3731495	0.3769189	
0.9062	0.3283593	0.3292032	0.3297821	0.3290087	0.3330442	0.33290
0.9219	0.3053948	0.3062299	0.3067999	0.3060400	0.3099097	
0.9297	0.2919080	0.2927229	0.2932790	0.2925360	0.2962703	0.29622
0.9375	0.2765008	0.2772919	0.2778314	0.2771112	0.2807056	
1.0000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.00000

Table 4 shows the comparison between the maximum ψ, ω, and location on the primary vortex, and the minimum ψ, ω, and location on the lower-left secondary vortex computed from the code of this work and the results reported in [15] based on the “lid-driven cavity flow” problem for $Re = 5000$. The maximum ψ computed from the code with category **A** delivers the maximum difference of 2.45 % compared to the maximum ψ reported in [15]. Besides, the minimum ψ computed from the code with category **C** grants the maximum difference of 0.05 % compared to the minimum ψ reported in [15]. Moreover, the differences of the maximum and minimum ψ computed from the code with category **D** compared to the solutions reported in [15] are 2.42 % and 0.04 %, respectively.

Table 4: Comparison between the maximum ψ, ω, and location on the primary vortex, and the minimum ψ, ω, and location on the lower-left secondary vortex computed from the code and the results reported in [15] based on the "lid-driven cavity flow" problem for *Re* = 5000.

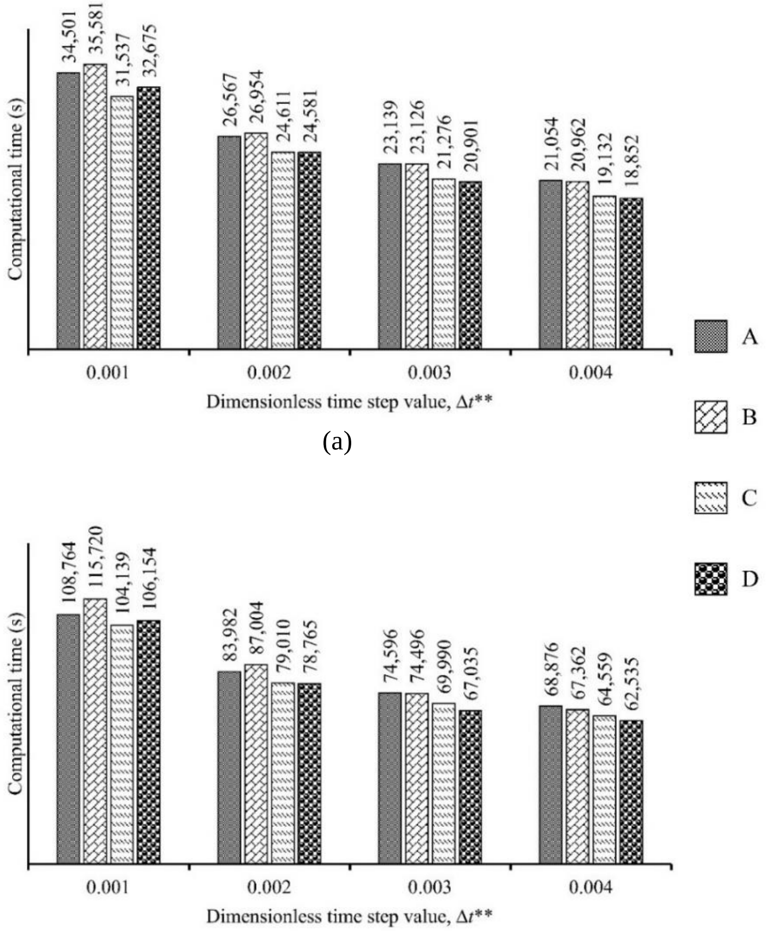
Vortex	Value	Present work				[15]
		A	**B**	**C**	**D**	
Primary vortex	ψ_{max}	0.11898	0.11901	0.11901	0.11901	0.12197
	ω	1.81501	1.81664	1.81683	1.81669	1.9327
	x^{**}	0.48818	0.48818	0.48818	0.48818	0.48535
	y^{**}	0.53493	0.53493	0.53493	0.53493	0.53516
Lower left secondary vortex	ψ_{min}	-0.0030704	-0.0030718	-0.0030721	-0.0030718	-0.0030706
	ω	-2.71265	-2.71275	-2.71269	-2.71273	-2.7244
	x^{**}	0.19452	0.19452	0.19452	0.19452	0.19434
	y^{**}	0.07477	0.07477	0.07477	0.07477	0.073242

Performance Test of the LPI-WF Method

The performance test of the proposed LPI-WF method is carried out by comparing the computational time for solving the "lid-driven cavity flow" problem in the four categories. The "lid-driven cavity flow" problem is simulated for *Re* of 1000 and 5000. The non-uniform mesh sizes used are 100 × 100 and 150 × 150 for *Re* of 1000, and 150 × 150 and 200 × 200 for *Re* of 5000. The computer CPU time is used as the computational time. The computational time reported in the following contents is rounded off to integer values. The processor of the computer employed for computing is Intel ® Core ™ i5-4460 CPU @ 3.20 GHz.

Figure 5, Figure 6 and Table 5 show the computational time comparison of the "lid-driven cavity flow" problem for *Re* of 1000 and 5000 in the different categories and on the dimensionless time step values from 0.001 to 0.004. All results show that the computational time of category **D** is less than the computational time of categories **A** and **B**. The maximum computational time reductions of category **D** are 10.46 % and 11.36 % of the computational time of category **A** for *Re* of 1000 at 100 × 100 mesh size and 0.004 dimensionless time step value, and category **B** for *Re* of 5000 at 150 × 150 mesh size and 0.004 dimensionless time step value, respectively. Also, the computational time of category **D** is less than the computational time of category **C** for *Re* of 1000 at 100 × 100 and 150 × 150 mesh sizes and 0.002, 0.003, and 0.004 dimensionless time step values, and for *Re* of 5000 at 200 × 200 mesh size and 0.003 and 0.004 dimensionless time step values. Thus, at high resolutions and large time step values, category **D** has higher performance than category **C** for reducing computational time. The maximum computational time reduction of category **D** is 3.3 % of the computational time of category **C** for *Re* of 5000 at

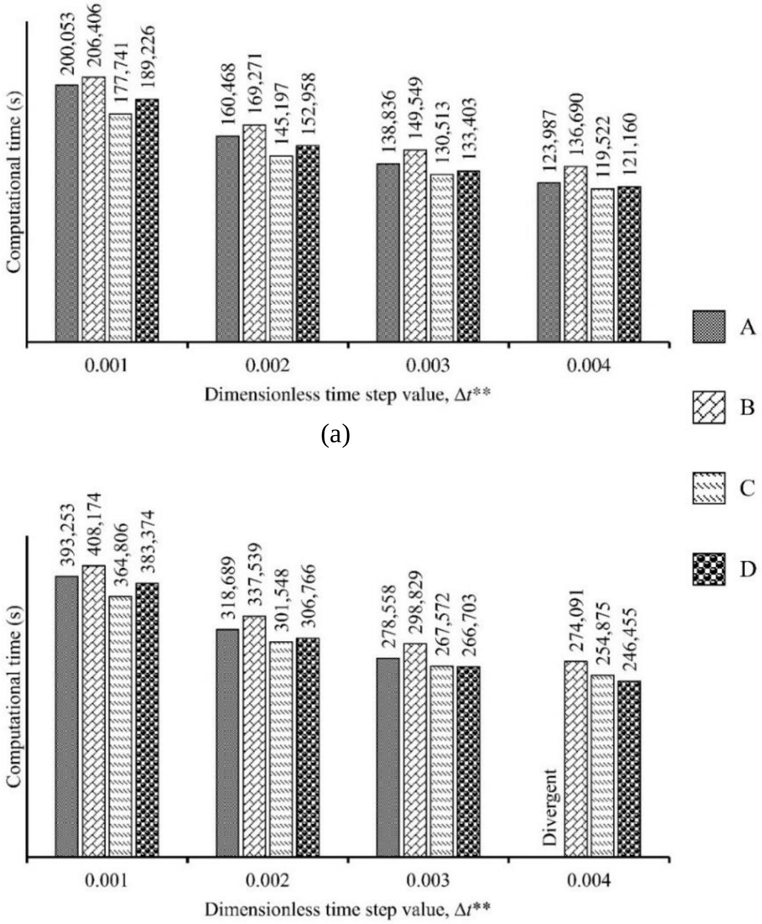
200 × 200 mesh size and 0.004 dimensionless time step value. As shown in Figure 6 (b), the result of category **A** at 0.004 dimensionless time step value, is divergent (i.e., the solution fails to converge). Thus, the computation of category **D** is more stable than the computation of category **A**.



(a)

(b)

Figure 5: Comparisons of computational time based on the "lid-driven cavity flow" problem for *Re* = 1000 in the different categories: (a) 100 × 100 mesh size and (b) 150 × 150 mesh size.



(a)

(b)

Figure 6: Comparisons of computational time based on the "lid-driven cavity flow" problem for *Re* = 5000 in the different categories: (a) 150 × 150 mesh size and (b) 200 × 200 mesh size.

Table 5: Number of iterations and computational time for the first ten time steps based on the "lid-driven cavity flow" problem for *Re* = 5000, 200×200 mesh size, and 0.003 dimensionless time step value.

Time step	Category	Number of iterations per time step	Improvement in the number of iterations (%) compared to category **A**	Computational time (s) per time step	Reduction of computational time (%) compared to category **A**
1	**A**	10	-	1778	-
	B	10	NA	1777	0.06
	C	10	NA	1780	NA
	D	10	NA	1778	NA
2	**A**	9	-	1032	-
	B	7	22.22	918	11.05
	C	7	22.22	901	12.69
	D	7	22.22	901	12.69
3	**A**	8	-	916	-
	B	7	12.50	753	17.79
	C	6	25.00	745	18.67
	D	6	25.00	745	18.67
4	**A**	9	-	795	-
	B	7	22.22	643	19.12
	C	6	33.33	635	20.13
	D	6	33.33	635	20.13
5	**A**	9	-	693	-
	B	7	22.22	544	21.50
	C	7	22.22	536	22.66
	D	7	22.22	535	22.80
6	**A**	9	-	626	-
	B	7	22.22	492	21.41
	C	7	22.22	484	22.68
	D	7	22.22	483	22.84
7	**A**	9	-	568	-
	B	8	11.11	448	21.13
	C	7	22.22	452	20.42
	D	7	22.22	441	22.36
8	**A**	8	-	520	-
	B	7	12.50	412	20.77
	C	7	12.50	442	15.00
	D	7	12.50	434	16.54
9	**A**	9	-	477	-
	B	7	22.22	380	20.34
	C	7	22.22	380	20.34
	D	7	22.22	380	20.34
10	**A**	8	-	439	-
	B	7	12.50	357	18.68
	C	7	12.50	357	18.68
	D	7	12.50	356	18.91

NA = Not available

Conclusions

The details of the LPI-WF method developed for determining the initial guess values of iterative methods to solve transient problems are demonstrated. The efficiency of the LPI-WF method are proven by the performance test carried out with the "lid-driven cavity flow" problem for *Re* of 1000 and 5000 at the different mesh sizes and various dimensionless time step values. The conclusion of this work is outlined in the following items:

1. The LPI-WF method is suitable for use with high-order temporal schemes such as the LIP scheme for extrapolation of initial guess values from solutions of previous time steps.
2 The procedures of the LPI-WF method are not complicated and are easy to add to the code of problem solvers for convergence acceleration to the steady-state condition of solutions.
3. From the results of the performance test, the computational time of the code with the LPI-WF method can be reduced up to 10.46 % of the computational time of the code with the conventional method for obtaining the steady-state condition of the solutions. Also, the computation of the code with the LPI-WF method at high dimensionless time step values is more stable than that of the conventional method.
4. Moreover, the performance of the LPI-WF method for reducing the computational time of the code is better than the performance of the LPI method at the high resolution of the mesh sizes and the large amounts of the dimensionless time step values.

Also, the LPI-WF method is implemented preliminarily as a performance test to verify the robust potential of the method for reducing the computational time of the flow solvers. For future work, to verify clearly that the LPI-WF method accelerates the convergence of solutions, the performance test of the method will be carried out extensively on both the "lid-driven cavity flow" problem at the various Reynolds numbers and other well-known problems.

References

[1] D. Arumuga Perumal, I. M. Gowhar, S. A. Ananthapuri and V. Jayakrishnan, "Computation of temperature distributions on uniform and non-uniform lattice sizes using mesoscopic lattice Boltzmann method," *Journal of Mechanical Engineering*, vol. 11, no. 2, pp 53-65, 2014.

[2] R. Markovinović and J. D. Jansen, "Accelerating iterative solution methods using reduced-order models as solution predictors," *International Journal for Numerical Methods in Engineering*, vol. 68, no. 5, pp 525-541, 2006.

[3] H. Hu, C. Chen and K. Pan, "Time-extrapolation algorithm (TEA) for linear parabolic problems," *Journal of Computational Mathematics*, vol.

32, no. 2, pp 183-194, 2014.

[4] P. Birken, T. Gleim, D. Kuhl and A. Meister, "Fast solvers for unsteady thermal fluid structure interaction," *International Journal for Numerical Methods in Fluids*, vol. 79, no. 1, pp 16-29, 2015.

[5] S. Sachs, M. Streitenberger, D. C. Sternel and M. Schäfer, "Extrapolation methods for accelerating unsteady partitioned fluid-structure interaction simulations," *International Journal of Multiphysics*, vol. 5, no. 4, pp 287-297, 2011.

[6] J. Huang, H. Wang and H. Yang, "Int-Deep: A deep learning initialized iterative method for nonlinear problems," *Journal of Computational Physics*, vol. 419, 2020.

[7] A. V. Malevsky and D. A. Yuen, "Large-scale numerical simulations of turbulent non-Newtonian thermal convection using method of characteristics," *Computer Physics Communications*, vol. 73, no. 1-3, pp 61-71, 1992.

[8] M. M. Rashid, "Deformation extrapolation and initial predictors in large-deformation finite element analysis," *Computational Mechanics*, vol. 16, no. 5, pp 281-289, 1995.

[9] Q. Zhang, S. Gui, H. Li and B. Lu, "Model reduction-based initialization methods for solving the Poisson-Nernst-Plank equations in three-dimensional ion channel simulations," *Journal of Computational Physics*, vol. 419, 2020.

[10] P. Van Leemput, M. Rheinländer and M. Junk, "Smooth initialization of lattice Boltzmann schemes," *Computers and Mathematics with Applications*, vol. 58, no. 5, pp 867-882, 2009.

[11] L. Grinberg and G. Em Karniadakis, "Extrapolation-based acceleration of iterative solvers: Application to simulation of 3D flows," *Communications in Computational Physics*, vol. 9, no. 3, pp 607-626, 2011.

[12] B. E. Merrill, Y. T. Peet, P. F. Fischer and J. W. Lottes, "A spectrally accurate method for overlapping grid solution of incompressible Navier-Stokes equations," *Journal of Computational Physics*, vol. 307, pp 60-93, 2016.

[13] Y. Liu, G. Wang and Z. Ye, "Dynamic mode extrapolation to improve the efficiency of dual time stepping method," *Journal of Computational Physics*, vol. 352, pp 190-212, 2018.

[14] O. Botella and R. Peyret, "Benchmark spectral results on the lid-driven cavity flow," *Computers and Fluids*, vol. 27, no. 4, pp 421-433, 1998.

[15] C. H. Bruneau and M. Saad, "The 2D lid-driven cavity problem revisited," *Computers and Fluids*, vol. 35, no. 3, pp 326-348, 2006.

[16] U. Prasopchingchana and T. Manewattana, "A new scheme for the finite volume method verified with two dimensional laminar natural convection in a square cavity," *Engineering Journal*, vol. 19, no. 4, pp 133-152, 2015.

[17] B. An, F. Mellibovsky, J. M. Bergadà and W. M. Sang, "Towards a better understanding of wall-driven square cavity flows using the lattice

Boltzmann method," *Applied Mathematical Modelling*, vol. 82, pp 469-486, 2020.

[18] N. B. Barik and T. V. S. Sekhar, "Mesh-free multilevel iterative algorithm for Navier–Stokes equations," *Numerical Heat Transfer, Part B: Fundamentals*, 2020.

[19] B. J. Gross, N. Trask, P. Kuberry and P. J. Atzberger, "Meshfree methods on manifolds for hydrodynamic flows on curved surfaces: A Generalized Moving Least-Squares (GMLS) approach, " *Journal of Computational Physics*, vol. 409, 2020.

[20] W. He, G. Qin, J. Lin and C. Jia, "A segregated spectral finite element method for the 2D transient incompressible Navier–Stokes equations," *Computers and Mathematics with Applications*, vol. 79, no. 2, pp 521-537, 2020.

[21] M. A. D. S. Lourenço and E. L. M. Padilla, "An octree structured finite volume based solver," *Applied Mathematics and Computation*, vol. 365, 2020.

[22] Q. Luo, "Discretized pressure Poisson algorithm for steady incompressible flow on two-dimensional triangular unstructured grids," *European Journal of Mechanics, B/Fluids*, vol. 80, pp 187-194, 2020.

[23] M. Ramos Ortega, A. Beaudoin and S. Huberson, "Optimized incompressible smoothed particle hydrodynamics methods and validations," *International Journal for Numerical Methods in Fluids*, vol. 92, no. 11, pp 1528-1550, 2020.

[24] Y. Vasyliv and A. Alexeev, "Development of General Finite Differences for complex geometries using a sharp interface formulation," *Computers and Fluids*, vol. 193, 2019.