

# Adaptive $p$ -finite element method for wind engineering

R. Panneer Selvam<sup>†</sup> and Zu-Qing Qu<sup>‡</sup>

*Computational Mechanics Laboratory, Department of Civil Engineering  
BELL 4190, University of Arkansas, Fayetteville, AR 72701, USA*

**Abstract.** An important goal of computational wind engineering is to impact the design process with simulations of flow around buildings and bridges. One challenging aspect of this goal is to solve the Navier-Stokes (NS) equations accurately. For the unsteady computations, an adaptive finite element technique may reduce the computer time and storage. The preliminary application of a  $p$ -version as well as an  $h$ -version adaptive technique to computational wind engineering has been reported in previous paper. The details on the implementation of  $p$ -adaptive technique will be discussed in this paper. In this technique, two posteriori error estimations, which are based on the velocity and vorticity, are first presented. Then, the polynomial order of the interpolation function is increased continuously element by element until the estimated error is less than the accepted. The second through sixth orders of hierarchical functions are used as the interpolation polynomials. Unequal order interpolations are used for velocity and pressure. Using the flow around a circular cylinder with Reynolds number of 1000 the two error estimators are compared. The result show that the estimated error based on the velocity is lower than that based on the vorticity.

**Key words:** computational fluid dynamics; adaptive finite element method; computational wind engineering.

---

## 1. Introduction

An important goal of computational wind engineering is to impact the design process with simulations of flow around buildings and bridges. One challenging aspect of this goal is to solve the Navier-Stokes (NS) equations accurately. The wind engineering flows are very complex. The flow features changes in time and space. For accurate computing the grid has to be refined in the whole flow region or refine constantly wherever refinement is needed. The first approach is computationally expensive and impractical for most of the practical problems, especially in computing the flow around the bridges (Selvam 1999) where one needs to perform more than 20,000 time steps. This consumes a large amount of storage space and time. In the second approach, procedures like overlapping grid and adaptive grid techniques are used. The adaptive technique is the most viable and efficient approach. Using the adaptive techniques the solution time can be reduced more than 10 times for 2D and 100 times for 3D. Also computation can be performed to a desired accuracy by using posteriori error estimation techniques. Hence this technique becomes a powerful tool to solve many of the practical problems.

The adaptive FEM technique was proposed in early 1980s. In this technique, the errors are first calculated to assess the accuracy of the solution. If the errors are larger than the prescribed, the

---

<sup>†</sup> Professor

<sup>‡</sup> Research Associate

finite element model is then refined through redistributing the nodes (called R-version adaptive FEM), or reducing the size of elements (called H-version adaptive FEM), or increasing the order of the interpolation functions (called P-version adaptive FEM), or using a combination of the above. The new model is then re-analyzed and the errors in the new model are recalculated. The procedure is continued until the calculated errors fall below the specified permissible values.

Although intensive research has been done on the theory and application of adaptive FEM (Li 1997), only a few researchers investigated its application in computational wind engineering (CWE). Choi and Yu (1998, 1999) investigated the  $h$ -refinement for flows over a square cylinder. They used the penalty-function formulation to solve the NS equations. Selvam (2000) applied the mesh enrichment technique ( $h$ -refinement) and  $p$ -refinement techniques to flows over a circular cylinder. He used the primitive variable form to solve the NS equations.

Some details of the  $p$ -refinement and preliminary results were reported in Selvam (2000). Further details on the implementation of  $p$ -adaptive technique and different posteriori error estimates in use will be discussed in this paper. Computed results for flows over a circular cylinder for Reynolds number of 1000 will be considered. The efficiency of the computer time and storage using the adaptive technique versus the regular procedure will be illustrated. The computed drag and lift coefficients will be compared with experimental and other computational results.

## 2. Computer modeling of the flow

### 2.1. Governing equations

In the following discussion Reynolds number  $Re$ , drag coefficient  $C_d$ , lift coefficient  $C_l$ , and Strouhal number  $S_t$  are defined as

$$\begin{cases} R_e = VD/\nu \\ C_d = F_x/(0.5\rho V^2 D) \\ C_l = F_y/(0.5\rho V^2 D) \\ S_t = D/(TV) \end{cases} \quad (1)$$

where  $D$  is the diameter of the cylinder,  $V$  is the reference velocity,  $\nu$  is the kinematic viscosity,  $F_x$  and  $F_y$  are the drag and lift force,  $T$  is the period of oscillation of the lift forces and  $\rho$  is the density.

The flow around a cylinder is represented by the Navier-Stokes (NS) equations. The two and three-dimensional equations for an incompressible fluid in general tensor notation are as follows:

$$\text{Continuity Equation:} \quad U_{i,i} = 0 \quad (2)$$

$$\text{Momentum Equation:} \quad U_{i,t} + U_j U_{i,j} = -(p/\rho)_{,i} + \left[ \nu(U_{i,j} + U_{j,i}) \right]_{,j} \quad (3)$$

where  $U_i$  and  $p$  are the velocity and pressure respectively.  $\rho$  is the mass density of wind. A comma represents one differentiation;  $t$  represent time.  $i = 1, 2$  and  $3$  mean variables in the  $x, y$  and  $z$  directions. To implement higher order approximation of the convection term the following expression is used instead of Eq. (3)

$$U_{i,t} + U_j U_{i,j} - \theta (U_j U_k U_{i,j})_{,k} / 2 = - (p/\rho)_{,i} + [v(U_{i,j} + U_{j,i})] \quad (4)$$

Depending upon the values of  $\theta$ , different procedures can be implemented. For balance tensor diffusivity (BTD) scheme  $\theta = \delta t$  is used; where  $\delta t$  is the time step size used in the integration. For the streamline upwind procedure suggested by Brooks and Hughes (1982),  $\theta$  is considered as

$$\theta = \frac{1}{\max\left(\frac{|U_1|}{\delta x}, \frac{|U_2|}{\delta y}, \frac{|U_3|}{\delta z}\right)} \quad (5)$$

Here  $\delta x$ ,  $\delta y$ , and  $\delta z$  are the control volume length in the  $x$ ,  $y$ , and  $z$  directions;  $U_1$ ,  $U_2$ , and  $U_3$  are the velocities in the three directions. In the present research  $\theta = \delta t$  is used. This has less numerical diffusion for the benchmark problems in Selvam (1998).

## 2.2. Solution of the equations

Using an implicit method, which is similar to Selvam (1997), the NS equations are solved. This procedure eliminates the restriction on time step. The three-step advancement scheme for Eqs. (2) and (4) is as follows:

- Step 1: Solve for  $U_i$  from Eq. (3). The diffusion and the higher order convection terms are considered implicitly to be in the current time and the first order convection terms are considered explicitly from the previous time step. The pressure is considered in the right hand side of the equation. This set of equations leads to a symmetric matrix and the preconditioned conjugate gradient (PCG) procedure is used to solve it.
- Step 2: Solve for pressure correction from

$$(\delta p_{,i})_{,i} = U_{i,i} / \delta t .$$

- Step 3: Correct the velocity for incompressibility:

$$U_i = U_i - \delta t (\delta p_{,i})$$

where  $U_i$  is not specified and update the pressure  $p = p + \delta p$ .

The velocity and pressure are approximated using unequal order interpolation. When equal order interpolation is used, the solution started to diverge due to the violation of Babuska and Brezzi condition (Selvam 2000). To solve the velocities an underrelaxation factor of 0.7 is used. The iteration is done until the absolute sum of the residual of the equation reduces to  $1 \times 10^{-7}$  times the number of nodes for each time step. Usually the pressure and momentum equations take about 50 and 10 iterations for PCG solution.

## 3. *P*-adaptive finite element technique

The error estimate is one of the most important steps in the adaptive technique. It gives the error distribution in the present finite element grid, which can be used as indicator to refine the grid. The discretization errors of a finite element solution can be estimated by implicit or explicit method (Babuska 1994). In structural engineering, stress recovery techniques are usually used to estimate the errors because they are much easier to implement in the programming. As the exact solution is

generally not known, most of the approaches are concerned with posteriori error estimates.

In the present research, the error estimator based on the velocity as well as the vorticity is considered. The error is computed by considering the higher order solution to be exact and the lower order to be approximate. The difference between the two is defined as the error in the present grid, that is

$$\eta_v^i = \frac{\sqrt{\int_{\Omega_i} (v^p - v^{p-1})^T (v^p - v^{p-1}) d\Omega_i}}{\sqrt{\int_{\Omega_i} (v^p)^T v^{p-1} d\Omega_i}} \quad (6)$$

where  $\eta_v^i$  denotes the error of the  $i$ th element based on the velocity;  $v^p$  and  $v^{p-1}$  are the velocities within the element;  $p$  and  $p-1$  indicate the velocity computed from finite element model in which the highest polynomial orders are  $p$  and  $p-1$  respectively.  $\Omega_i$  is the area of the  $i$ th element. Similarly, the error estimation based on the vorticity can be obtained as

$$\eta_w^i = \frac{\sqrt{\int_{\Omega_i} (w^p - w^{p-1})^T (w^p - w^{p-1}) d\Omega_i}}{\sqrt{\int_{\Omega_i} (w^p)^T w^{p-1} d\Omega_i}} \quad (7)$$

where  $w$  denotes vorticity and is defined as

$$w = u_{,y} - v_{,x}$$

The order of polynomial to start with is quadratic for velocity. Hence it is easy to compute from quadratic to linear, third order to quadratic, and etc.

After the errors have been estimated and are higher than the accepted level, the next step is to refine the finite element model so as to reduce the errors. The accuracy of a finite element solution depends upon the shape and size of the elements and the order of the interpolation functions. Consequently, there are three methods to refine the finite element model.

*P*-refinement, which is considered in this paper, increases the order of the interpolation functions while keeping the mesh unchanged. Higher order elements generally provide a better description of the domain geometrically. They are particularly useful in regions where use of lower order elements would result in a mesh with poor aspect ratios in those elements (Li 1997). From the point of view of solution accuracy, higher order elements are usually more accurate than the lower order elements.

The hierarchical functions are applied here to increase the order of the interpolation function because this method allows implementing compatibility with neighboring elements easily. The compatibility is accomplished by keeping the node that is introduced on the side where in the order of the neighboring element is one less to be zero (Peano *et al.* 1979, Zienkiewicz, Taylor 1989, Szabo and Babuska 1990). This is possible because the coefficients of the hierarchical functions are not actual value of the unknowns but they are extra terms added to the basic unknowns that are in this case coefficient of linear shape functions.

For one-dimensional elements, the linear shape functions are usually defined as

$$P_0(\xi) = \frac{1-\xi}{2}, \quad P_1(\xi) = \frac{1+\xi}{2} \quad (8)$$

where  $\xi$  ( $-1 \leq \xi \leq 1$ ) is a non-dimensional coordinate. There are many ways to construct the higher order shape function (Zienkiewicz 1989). The following shape functions are used here :

$$P_s(\xi) = \begin{cases} \xi^s - 1 & s \text{ even} \\ \xi^s - \xi & s \text{ odd} \end{cases} \quad (9)$$

where  $s$  ( $\geq 2$ ) is the degree of the introduced polynomial. The corresponding displacement function can be obtained as

$$u(\xi) = \sum_{i=0}^n a_i P_i(\xi) \quad (10)$$

Using these one-dimensional formulae, it is easy to derive two-dimensional shape functions that are listed in the appendix.

#### 4. Numerical simulation

The computational region of flow around a circular cylinder is shown in Fig. 1. The diameter of the cylinder is considered to be a non-dimensional length of one. The inlet velocities in the  $x$  and  $y$  directions are considered to be one and zero respectively. On the top and bottom sides, the vertical velocities and the normal derivative of the velocities are considered to be zero. On the surface of the cylinder the velocities are also considered to be zero (no slip). The flow around the circular cylinder for Reynolds number of 1000 is simulated. The flow is run for 60 seconds.

A four-node quadrilateral element is originally used to describe the geometry of the element. The polynomial orders considered for velocity are from 2 to 6, which means in each direction the interpolation functions of orders 2 through 6 are considered. For pressure, 1 (linear) is considered when order 2 or 3 is used for velocity, and 2, 3, and 4 are, respectively, considered when order 4, 5, and 6 for velocity. The size of the element stiffness matrix for velocity varied from  $9 \times 9$  for order 2 to  $47 \times 47$  for order 6. The elements of the system matrices are numerically integrated. The integration points considered at this time are  $3 \times 3$  for order 2 to  $7 \times 7$  for order 6.

Two finite element grids as shown in Fig. 2 will be considered in this paper. The numbers of elements and nodes are listed in Table 1. When making a grid, it is recommended that the size of

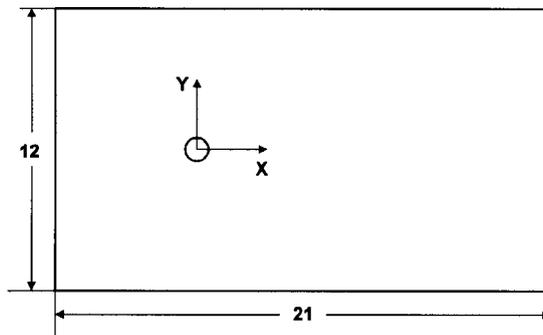
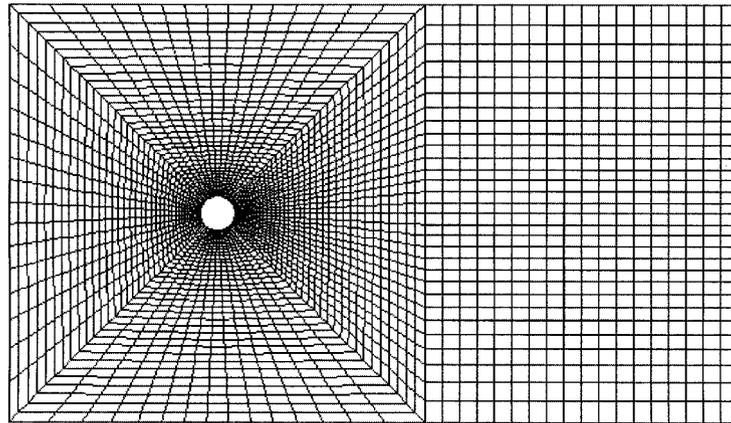
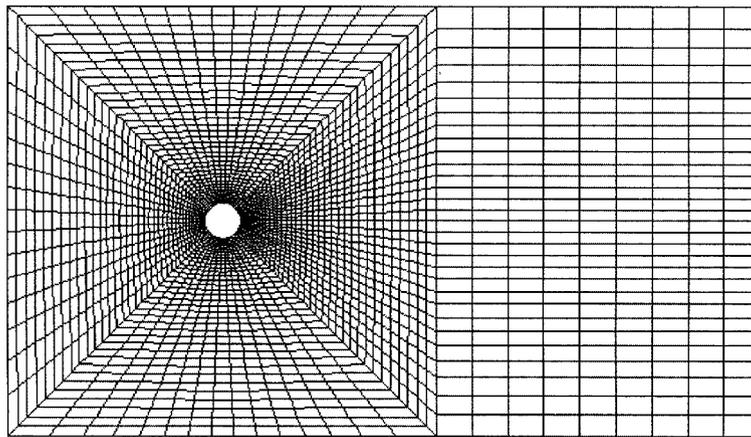


Fig. 1 Computational region of the flow around the cylinder



(a)



(b)

Fig. 2 Finite element grids (a) grid I (b) grid II

Table 1 Properties of the two finite element grids

Grid Number	Number of Elements	Number of Nodes
Grid-I	$3465 (75 \times 39 + 30 \times 18)$	3558
Grid-II	$3195 (75 \times 39 + 30 \times 9)$	3279

the element changes slowly. For comparison purpose, only 270 elements are used in the vortex shade region of the Grid-II while it is 540 in Grid-I.

The solution is sought using the initial grid. After three seconds of the solution, error is estimated in each element. Wherever the error is higher than the accepted level, the order of the polynomial is increased to the higher level. Then the error is estimated every one second to update the order of the polynomial. Since the hierarchical concept of finite element shape function is introduced, it is very easy to make compatibility from one order of element to another order as discussed by Peano *et al.* (1979). The new nodes introduced on the common boundary are constrained.

## 5. Results and discussions

### 5.1. Error estimation based on velocity

The error estimation based on the velocity is first considered and the permissible error is set 5%. In order to survey the effect of the order of polynomial on the accuracy, the highest order of the polynomial  $P$  is set 2 through 6 and run separately. The element distributions of the polynomial order and error at the end of 60-second for the case of  $P = 6$  are plotted in Fig. 3. It can be seen from this figure that the polynomial orders of only a few elements are increased to 3, 4, 5, and 6 respectively to meet the error requirement. There is only one element with sixth order of polynomial function because the grid is already well refined. The maximum error is 4.3% at the end of 60-second and is lower than the prescribed.

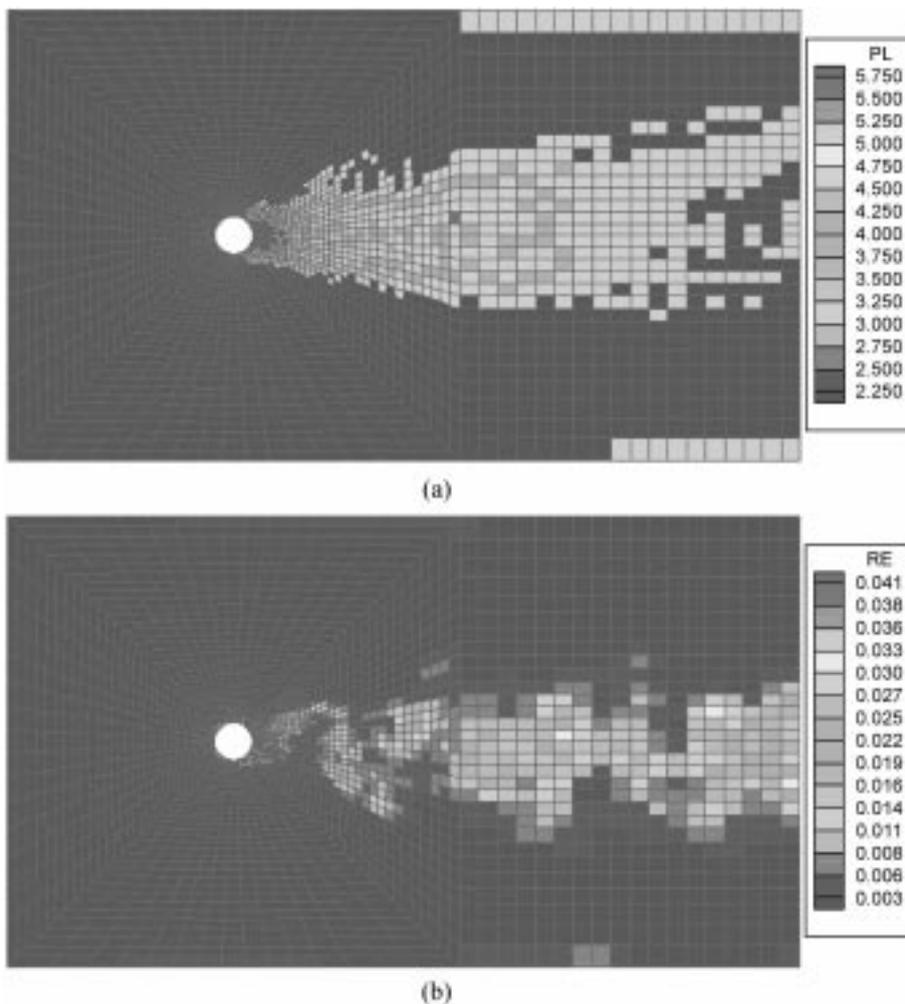


Fig. 3 Distributions of polynomial orders and errors for the case of  $p = 6$ ,  $\eta_w = 0.05$ , and Grid I (a) polynomial order; (b) error

Table 2 Numbers of elements and maximum errors with orders 2 through 6 for  $\eta_v = 0.05$  and Grid I

Highest Polynomial Order ( $P$ )	Order 2	Order 3	Order 4	Order 5	Order 6	Maximum Error
2	3465	0	0	0	0	0.160988
3	2644	821	0	0	0	0.080954
4	2620	655	190	0	0	0.085197
5	2606	498	178	183	0	0.042759
6	2606	498	177	183	1	0.042755

The numbers of elements with the second through sixth orders of polynomials are listed in Table 2 for the cases of  $P = 2$  through 6 (as the maximum order) respectively. These results are defined at the end of 60-second. For example, if the highest order of the polynomial  $P$  is 5, the numbers of the elements with the second through fifth orders of polynomial are, respectively, 2606, 498, 178, and 183. The corresponding maximum errors are also listed in the table for comparison purpose. Only a few elements are increased to higher order of polynomials when the accepted error is defined as 5%. The number of the elements with orders 3, 4, 5, and 6 for the case of  $P = 6$ , for example, is 24.8% of the total. This means that it is unnecessary to increase the order of interpolation polynomial to the 5th for all the elements. If the whole region is refined by 5th order of polynomials, the number of unknowns for velocity would be increased to 5.17 times, whereas, it is 21,197 and approximately 1.51 times of the original when the adaptive technique is applied. Hence, the storage and CPU time can be saved. This is one of the advantages of adaptive technique.

The error distribution plotted in Fig. 3 is at 60 seconds. Actually, the error distribution depends on the time. The maximum errors at ten specific times are listed in Table 3. Generally, the largest errors reduce with the increase of the highest polynomial order even though there is some fluctuation. The error at time 59.3-second, for example, reduces from 0.263134 to 0.052475 when the highest order increases from 2 to 6. This means that this scheme is convergent when the velocity is considered as the error indicator. The results in the columns  $P = 5$  and  $P = 6$  are very close because only one element increases to the 6th order which is indicated in Table 2 and Fig. 3.

The Grid-I is run again when the error tolerance is set 10%. The results corresponding to those

Table 3 Maximum errors at different times for  $\eta_v = 0.05$  and Grid I

Time (s)	Highest Polynomial Order ( $P$ )				
	2	3	4	5	6
50.3	0.195970	0.122675	0.105871	0.063609	0.063616
51.3	0.261792	0.176934	0.124404	0.062349	0.062376
52.3	0.261159	0.210993	0.106362	0.061637	0.061669
53.3	0.179489	0.127903	0.193338	0.087109	0.084705
54.3	0.258756	0.126718	0.145842	0.055646	0.055694
55.3	0.199193	0.121276	0.183497	0.090539	0.090453
56.3	0.193137	0.147167	0.117373	0.055231	0.055231
57.3	0.217649	0.159189	0.112478	0.088124	0.078881
58.3	0.267054	0.154314	0.151953	0.055649	0.053845
59.3	0.263134	0.151167	0.121222	0.052613	0.052475

Table 4 Numbers of elements and maximum errors with orders 2 through 6 for  $\eta_v = 0.1$  and Grid I

Highest Polynomial Order ( $P$ )	Order 2	Order 3	Order 4	Order 5	Order 6	Maximum Error
2	3465	0	0	0	0	0.160988
3	3177	288	0	0	0	0.083766
4	3150	282	33	0	0	0.087803
5	3132	271	48	14	0	0.084884
6	3132	271	48	14	0	0.084884

Table 5 Maximum errors at different times for  $\eta_v = 0.1$  and Grid I

Time (s)	Highest Polynomial Order ( $P$ )				
	2	3	4	5	6
50.3	0.195970	0.205947	0.188437	0.173341	0.173341
51.3	0.261792	0.180981	0.292294	0.139677	0.139677
52.3	0.261159	0.161265	0.146487	0.104099	0.104099
53.3	0.179489	0.124180	0.168261	0.150782	0.150782
54.3	0.258756	0.154281	0.194155	0.115234	0.115234
55.3	0.199193	0.170389	0.174156	0.188298	0.188298
56.3	0.193137	0.150464	0.101355	0.120437	0.120437
57.3	0.217649	0.147745	0.092485	0.111743	0.111743
58.3	0.267054	0.196698	0.173950	0.141964	0.141964
59.3	0.263134	0.168014	0.169602	0.109196	0.109196

listed in Tables 2 and 3 are listed in Tables 4 and 5. For the case of  $P = 6$ , only 9.6% of the total elements are increased by higher orders of polynomial which is much smaller than the above mentioned 24.8%. The results for the cases of  $P = 5$  and  $P = 6$  are same as shown in Tables 4 and 5. This means that the highest polynomial order  $P = 5$  is enough to satisfy the permissible error. The distributions of the polynomial order and error in the element level for the case of  $P = 5$  are plotted in Fig. 4.

The largest errors at different times for Grid II are listed in Table 6. The error tolerance for velocity is 10 percent. These results show that the errors reduce generally when the highest order of the polynomial increases. Hence, the error estimation based on the vorticity seems feasible. However, there is some kind of fluctuation especially for the results in Table 6 where the errors for the case of  $P = 3$  are lower than or close to those for  $P = 6$ . The reason might be that the order of polynomial for pressure does not change when the order for velocity increases from 2 to 3.

### 5.2. Error estimation based on vorticity

Vorticity is also a very important parameter in CWE. Hence, the possibility of implementing it as an error estimator is investigated.

The numbers of elements with the 2nd through 5th orders of polynomials for grid I at the end of 60-second are listed in Table 7. Here, the error tolerance  $\eta_w$  is 10% and the cases of  $P = 2, 3, 4,$  and  $5$  are considered. The distributions of the polynomial order and error in the element level are plotted

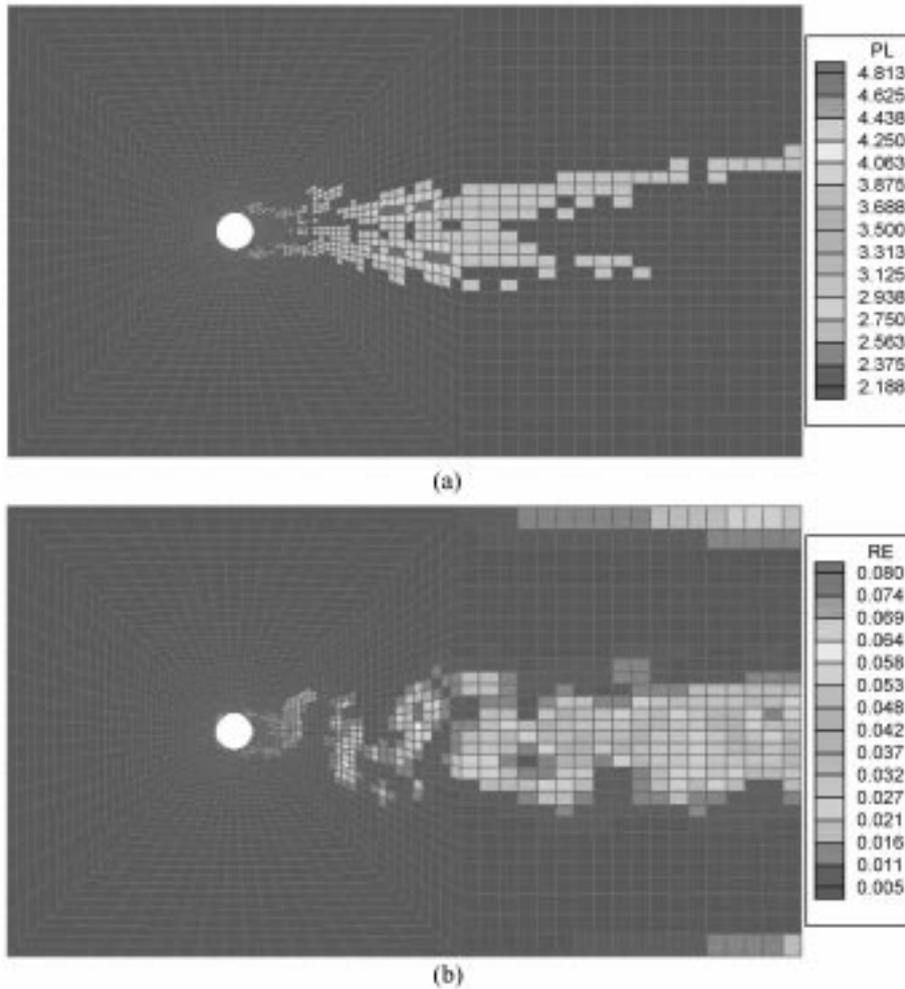


Fig. 4 Distributions of polynomial orders and errors for the case of  $p = 5$ ,  $\eta_v = 0.1$ , and Grid I: (a) polynomial order; (b) error

Table 6 Maximum errors at different times for  $\eta_v = 0.1$  and Grid II

Time (s)	Highest Polynomial Order ( $P$ )				
	2	3	4	5	6
50.3	0.250028	0.168422	0.150883	0.187755	0.185344
51.3	0.390422	0.085862	0.178424	0.144909	0.145114
52.3	0.326127	0.168135	0.231421	0.163934	0.104425
53.3	0.289508	0.176795	0.126859	0.126996	0.127785
54.3	0.222695	0.166687	0.194164	0.188824	0.205791
55.3	0.194293	0.164140	0.107705	0.168732	0.170727
56.3	0.281389	0.138609	0.182484	0.110528	0.164940
57.3	0.294436	0.176626	0.212669	0.111168	0.109624
58.3	0.305424	0.136638	0.106783	0.142887	0.140860
59.3	0.293086	0.145484	0.214223	0.169105	0.176693

Table 7 Number of elements and maximum errors with orders 2 through 5 for  $\eta_w = 0.1$  and Grid I

Highest Polynomial Order ( $P$ )	Order 2	Order 3	Order 4	Order 5	Maximum Error
2	3465	0	0	0	0.420721
3	2619	846	0	0	0.147642
4	2563	638	264	0	0.183051
5	2473	368	191	433	0.094439

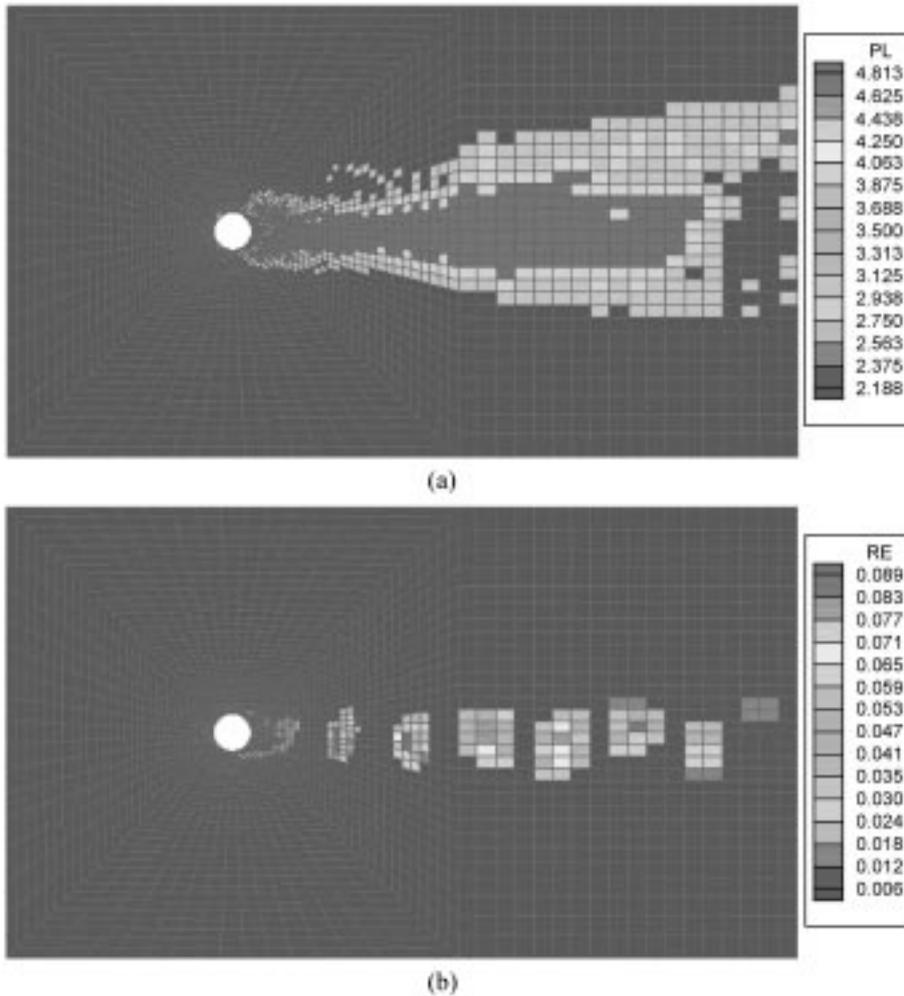


Fig. 5 Distributions of polynomial orders and errors for the case of  $p = 6$ ,  $\eta_w = 0.1$ , and Grid I: (a) polynomial order; (b) error

in Fig. 5. Totally, 28.6% of all the elements are defined by the 3rd, 4th, and 5th order of polynomial for the case of  $P = 5$ , while it is 9.6% in Table 4. It seems that the value of the error based on the vorticity is much higher than that of velocity.

The corresponding maximum errors for Grid I at ten specific times are listed in Table 8. Even though the maximum error is 9.4% at the end of 60-second as shown in Fig. 5 and less than the

Table 8 Maximum errors at different times for  $\eta_w = 0.1$  and Grid I

Time (s)	Highest Polynomial Order ( $P$ )			
	2	3	4	5
50.3	0.524203	0.223650	0.400943	0.347235
51.3	0.612420	0.236439	0.358181	0.222678
52.3	0.600521	0.228422	0.351574	0.228674
53.3	0.535412	0.200794	0.478089	0.324771
54.3	0.608478	0.203296	0.487932	0.463235
55.3	0.530800	0.205499	0.355667	0.225110
56.3	0.611393	0.238096	0.653971	0.317377
57.3	0.523419	0.248520	0.363541	0.234527
58.3	0.615641	0.196858	0.514696	0.194942
59.3	0.601942	0.217063	0.462589	0.145696

Table 9 Maximum errors at different times for  $\eta_w = 0.1$  and Grid II

Time (s)	Highest Polynomial Order ( $P$ )			
	2	3	4	5
50.3	0.632139	0.341736	0.747839	0.676253
51.3	0.531782	0.384012	0.820789	0.346858
52.3	0.672570	0.373122	1.272311	0.642495
53.3	0.663743	0.343420	0.589324	0.685651
54.3	0.608990	0.422948	0.794185	0.861923
55.3	0.521911	0.332177	1.308439	0.661017
56.3	0.609646	0.209758	0.798121	0.526909
57.3	0.665467	0.417948	0.507647	0.637538
58.3	0.627186	0.309212	0.527995	0.423094
59.3	0.647309	0.281683	1.210576	0.438141

accepted error 10%, the maximum error at other specific times are much higher than 10% as shown in Table 8. The maximum errors for the case of  $P = 3$  are lower than or close to those of case  $P = 5$  which is similar to the results in Table 6. The errors fluctuate a little more violently than those from velocity.

The largest errors at different times for Grid II are listed in Table 9. These results also shown that the error estimation based on the vorticity is high.

### 5.3. Aerodynamic forces

The drag and lift force coefficients computed using Grid-II are plotted in Figs. 6 and 7. The prescribed error limitation of velocity is 10%. After the simulation runs 40 seconds, both the drag and the lift force coefficients become stable. It can be seen from Figs. 6 and 7 that there is no much difference among the amplitudes of the drag and lift coefficients for different  $P$ s. For accuracy purpose, only the last four periods of the drag plots and three periods of the lift plots are considered for comparison in the following.

The averages, amplitudes, and periods of the drag and lift force coefficients plotted in Figs. 6 and 7

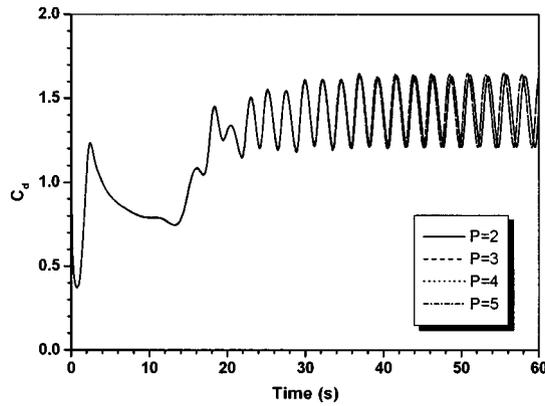


Fig. 6 Drag coefficients for Grid-II

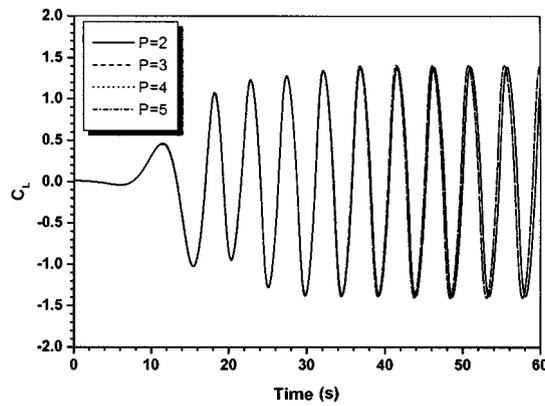


Fig. 7 Lift coefficients for Grid-II

Table 10 Drag and lift coefficients for Grid-I

Prescribed Error	$P$	Drag Coefficient			Lift Coefficient		
		Average	Amplitude	Period (s)	Average	Amplitude	Period (s)
10%	2	1.423	0.213	2.355	0.000	1.389	4.711
10%	3	1.426	0.219	2.324	0.000	1.407	4.652
10%	4	1.426	0.219	2.324	0.000	1.407	4.652
10%	5	1.426	0.219	2.324	0.000	1.407	4.652
5%	3	1.428	0.223	2.320	0.000	1.414	4.626
5%	4	1.428	0.223	2.320	0.000	1.413	4.639
5%	5	1.428	0.223	2.318	0.000	1.413	4.637

are listed in Table 10. When the polynomial orders of all the elements are two, the average, amplitude, and period of the drag coefficient are 1.423, 0.213, and 2.355 respectively. When the highest polynomial order  $P$  is set as 3, the average and amplitude increase to 1.426 and 0.219 while the period of the drag coefficient decrease a little. After that, even though the highest polynomial  $P$

has been increased to 4 and 5, the average, amplitude, and period change very slightly. As for the lift force coefficient, it changes a little with the increase of the highest polynomial order  $P$ . When the prescribed error is 5%, these results change a lot from  $P = 2$  to  $P = 3$ . Then, there is no much difference for the  $P = 3, 4,$  and  $5$ . The computed  $C_d$  are 1.426 and 1.428 for  $\eta_v = 10\%$  and  $\eta_v = 5\%$  respectively. They are a little higher than the result, 1.37, reported by Tamura, Ohta, and Kuwahara (1990).

## 6. Conclusions

Detailed information on the implementation of  $p$ -version finite element technique in CWE is discussed in this paper. Two error estimators, which help us know the level and the location of the error and give confidence in the solution, have been presented. They are based on the velocity and vorticity respectively. Since the exact solution is generally unknown, they are posteriori errors. The 2nd through 6th orders of hierarchical functions are used as the interpolation polynomials. In the numerical example, the flow around a circular cylinder with Reynolds number of 1000 is considered. Unequal order interpolation is used for velocity and pressure. Two different finite element grids are used to compare the two error estimators. The results show that the error based on the velocity is lower than that based on the vorticity.

Instead of increasing the unknowns to 5.17 times of the original to achieve less than 5% error, only 1.51 times of the initial unknowns is needed by an adaptive technique. Hence, the adaptive technique is an efficient procedure with respect to the CPU time and storage space. Only the refinement procedure is implemented in the present work. If the unrefinement procedure is included, the computation will be much more efficient. The computed drag force coefficient is a little higher than other reported results. At this time, the adaptive technique is only applied to the benchmark problem. The application of this technique to practical problems, such as bridge flutter under wind flow, is under way.

## Acknowledgements

This study has been partially funded by Department of Defence, AFOSR under DEPCOR program and Federal Highway Administration through Lendis Corporation, McLean, Virginia, University of Arkansas.

## References

- Babuska, I., Strouboulis, T., Upadhyay, C.S., Gangaraj, S.K. and Copps, K. (1994), "Validation of a posteriori error estimators by numerical approach", *Int. J. Numer. Meth. Eng.*, **37**, 1073-1123.
- Brooks, A.N. and Hughes, T. J. R. (1982), "Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations", *Comput. Meth. Appl. Mech. Eng.*, **32**, 199-259.
- Choi, C.K. and Yu, W.J. (1998), "Adaptive finite element wind analysis with mesh refinement and recovery", *Wind and Structures*, **1**, 111-125.
- Choi, C.K. and Yu, W.J. (1999), "Adaptive refinement/recovery for analysis of wind around structure", *J. Aerospace Eng.*, **12**, 168-175.
- Li, L.-Y. (1997), "Adaptive finite element methods: a review", *Applied Mechanics Reviews*, **50**, 581-591.
- Peano, A., Pasini, A., Riccioni, R. and Sardella, L. (1979), "Adaptive approximations in finite element structural

- analysis”, *Computers and Structures*, **10**, 333-342.
- Oden, J.T., Demkowicz, L., Rachowicz, W. and Hardy, O. (1989), “Toward a universal h-p adaptive finite element strategy, Part 1. Constrained approximation and data structure”, *Comput. Meth. Appl. Mech. Eng.*, **77**, 79-112.
- Selvam, R.P. (1997), “Finite element modeling of flow around a circular cylinder using LES”, *J. Wind Eng. Ind. Aerod.*, **67-68**, 129-139.
- Selvam, R.P. (1998), “Computational procedures in grid based computational bridge”, in A. Larsen & S. Eisdahl (eds), *Bridge Aerodynamics*, A.A. Balkema, Rotterdam, 327-336.
- Selvam, R.P. and Bosch, H. (1999), “Finite element modelling of flow around bridges”, in A. Larsen, G.L. Larose & F.M. Livesey (eds), *Wind Engineering into the 21st Century*, A.A. Balkema, Rotterdam, **2**, 1321-1327.
- Selvam, R.P. (2000), “Computation of flow over circular cylinder using p-adaptive FEM”, *Int. Symp. on Wind and Structures for the 21st Century*, Chejudo, Korea, January 26-28.
- Szabo, B. and Babuska, I. (1990), *Finite Element Analysis*, Wiley, New York.
- Tamura, T., Ohta, I. and Kuwahara, K. (1990), “On the reliability of two-dimensional simulation for unsteady flows around a cylinder-type structure”, *J. Wind Eng. Ind. Aerod.*, **35**, 275-298.
- Zienkiewicz, O.C., Gago, J.P.S. and Kelly, D.W. (1983), “The hierarchical concept in finite element analysis”, *Computers & Structures*, **16**, 53-65.
- Zienkiewicz, O.C. and Taylor, R. L. (1989), *The Finite Element Method*, **1**, McGraw-Hill Book Company, England.

## Appendix

Shape functions used in Eq. (10) are listed in the following:

### First order:

(for side nodes)

$$\begin{aligned} N(1) &= (1-\xi)(1-\eta)/4 \\ N(2) &= (1+\xi)(1-\eta)/4 \\ N(3) &= (1+\xi)(1+\eta)/4 \\ N(4) &= (1-\xi)(1+\eta)/4 \end{aligned}$$

### Second Order:

(for side nodes)

$$\begin{aligned} N(5) &= (\xi^2-1)(1-\eta)/2 \\ N(6) &= (1+\xi)(\eta^2-1)/2 \\ N(7) &= (\xi^2-1)(1+\eta)/2 \\ N(8) &= (1-\xi)(\eta^2-1)/2 \\ N(9) &= (\xi^2-1)(\eta^2-1) \end{aligned}$$

### Third Order:

(for side nodes)

$$\begin{aligned} N(10) &= (\xi^3-\xi)(1-\eta)/2 \\ N(11) &= (1+\xi)(\eta^3-\eta)/2 \\ N(12) &= (\xi^3-\xi)(1+\eta)/2 \\ N(13) &= (1-\xi)(\eta^3-\eta)/2 \end{aligned}$$

(for interior nodes)

$$\begin{aligned} N(14) &= (\xi^2-1)(\eta^3-\eta) \\ N(15) &= (\xi^3-\xi)(\eta^2-1) \\ N(16) &= (\xi^3-\xi)(\eta^3-\eta) \end{aligned}$$

### Fourth Order:

(for side nodes)

$$\begin{aligned} N(17) &= (\xi^4-1)(1-\eta)/2 \\ N(18) &= (1+\xi)(\eta^4-1)/2 \\ N(19) &= (\xi^4-1)(1+\eta)/2 \\ N(20) &= (1-\xi)(\eta^4-1)/2 \end{aligned}$$

(for interior nodes)

$$\begin{aligned} N(21) &= (\xi^4-1)(\eta^2-1) \\ N(22) &= (\xi^2-1)(\eta^4-1) \\ N(23) &= (\xi^4-1)(\eta^3-\eta) \\ N(24) &= (\xi^3-\xi)(\eta^4-1) \\ N(25) &= (\xi^4-1)(\eta^4-1) \end{aligned}$$

**Fifth Order:**

(for side nodes)

$$N(26) = (\xi^5 - \xi)(1 - \eta)/2$$

$$N(27) = (1 + \xi)(\eta^5 - \eta)/2$$

$$N(28) = (\xi^5 - \xi)(1 + \eta)/2$$

$$N(29) = (1 - \xi)(\eta^5 - \eta)/2$$

(for interior nodes)

$$N(30) = (\xi^5 - \xi)(\eta^2 - 1)$$

$$N(31) = (\xi^2 - 1)(\eta^5 - \eta)$$

$$N(32) = (\xi^5 - \xi)(\eta^3 - \eta)$$

$$N(33) = (\xi^3 - \xi)(\eta^5 - \eta)$$

$$N(34) = (\xi^5 - \xi)(\eta^4 - 1)$$

$$N(35) = (\xi^4 - 1)(\eta^5 - \eta)$$

$$N(36) = (\xi^5 - \xi)(\eta^5 - \eta)$$

**Sixth Order:**

(for side nodes)

$$N(37) = (\xi^6 - 1)(1 - \eta)/2$$

$$N(38) = (1 + \xi)(\eta^6 - 1)/2$$

$$N(39) = (\xi^6 - 1)(1 + \eta)/2$$

$$N(40) = (1 - \xi)(\eta^6 - 1)/2$$

(for interior nodes)

$$N(41) = (\xi^6 - 1)(\eta^2 - 1)$$

$$N(42) = (\xi^2 - 1)(\eta^6 - 1)$$

$$N(43) = (\xi^6 - 1)(\eta^3 - \eta)$$

$$N(44) = (\xi^3 - \xi)(\eta^6 - 1)$$

$$N(45) = (\xi^6 - 1)(\eta^4 - 1)$$

$$N(46) = (\xi^4 - 1)(\eta^6 - 1)$$

$$N(47) = (\xi^6 - 1)(\eta^5 - \eta)$$

$$N(48) = (\xi^5 - \xi)(\eta^6 - 1)$$

$$N(49) = (\xi^6 - 1)(\eta^6 - 1)$$