

**UPDATED LAGRANGIAN FINITE VOLUME SOLVER FOR LARGE  
DEFORMATION DYNAMIC RESPONSE OF ELASTIC BODY**

UDK 532.5:519.6

**Summary**

The study presented in this paper is a part of the project whose goal is to build a self-contained Finite Volume Method (FVM) Fluid-Structure Interaction (FSI) solver to simulate dynamic interaction between an incompressible Newtonian fluid and an elastic solid with the assumption of large structural deformation. The structural component of the FSI solver is described and validated. It applies the second-order cell-centred finite volume method for spatial discretisation of the linear momentum conservation law in the updated Lagrangian formulation. Temporal discretisation is performed by using a fully implicit second-order accurate three-time-level scheme and equations are solved by using a segregated iterative algorithm. The solver is validated on the vibration of a three-dimensional (3-D) cantilevered elastic beam and used to simulate the dynamic response of an axial turbine blade. It is shown that an accurate and stable numerical solution can be obtained over a long time period of transient simulation.

*Key words: elastic body, dynamic response, large deformation, cell-centred finite volume method, updated Lagrangian formulation*

**1. Introduction**

Most Fluid-Structure Interaction (FSI) simulations today involve a combination of solvers, usually with a finite-volume (FV) solver for the fluid flow, a finite-element (FE) solver for the structural analysis and a third code for coupling, data interpolation and simulation management. This approach imposes limitations on the mode of coupling and creates problems in the model setup. Endeavours to overcome this drawback have led to the development of computational techniques that facilitate the FSI modelling entirely within the context of FE [1, 2] or FV [3, 4, 5] methods.

The solution procedure for dynamic FSI problems requires, among other things, a structural dynamic solution procedure which will provide an accurate and stable result over long transient simulations. The solution to structural dynamic problems using FE methods is well established [6]. However, the number of papers on FV methods applied to such problems is still limited [7, 8], especially when large deformations are considered. This paper aims at contributing to the development of FV techniques in three dimensions for the solution to structural dynamic problems with large deformations, used in the context of the time-and-space-accurate dynamic FVM FSI procedure.

The rest of this paper is organised as follows. Section 2 gives a description of a mathematical model governing the dynamic behaviour of a compressible elastic body with the assumption of high deformations. The finite volume discretisation of the governing equation and the solution procedure is presented in Section 3. In Section 4, the solver is tested on a simple case with known solution, and then a simulation of a vibrating axial turbine blade is presented. Section 5 gives a short conclusion.

## 2. Mathematical formulation

The motion of an isothermal continuum in an arbitrary volume  $V$  bounded by a surface  $S$  is governed by the conservation laws for mass and linear momentum:

$$\frac{d}{dt} \int_V \rho \, dV + \oint_S \mathbf{n} \cdot \rho(\mathbf{v} - \mathbf{v}_s) \, dS = 0, \quad (1)$$

$$\frac{d}{dt} \int_V \rho \mathbf{v} \, dV + \oint_S \mathbf{n} \cdot \rho(\mathbf{v} - \mathbf{v}_s) \mathbf{v} \, dS = \oint_S \mathbf{n} \cdot \boldsymbol{\sigma} \, dS + \int_V \rho \mathbf{f}_b \, dV, \quad (2)$$

where  $\rho$  is the continuum density,  $\mathbf{n}$  is the outward pointing unit normal to the surface  $S$ ,  $\mathbf{v}$  is the velocity of the continuum,  $\mathbf{v}_s$  is the velocity of the surface  $S$ ,  $\boldsymbol{\sigma}$  is the Cauchy stress tensor and  $\mathbf{f}_b$  is the resulting body force. The relationship between the rate of change of the volume  $V$  and the velocity  $\mathbf{v}_s$  is defined by the so called *space conservation law* (SCL) [9]:

$$\frac{d}{dt} \int_V dV - \oint_S \mathbf{n} \cdot \mathbf{v}_s \, dS = 0. \quad (3)$$

In this study, the structure is assumed to be elastic and isothermal. The dynamic behaviour of such a continuum is usually described by considering only the linear momentum conservation law in Lagrangian formulation:

$$\frac{D}{Dt} \int_V \rho \mathbf{v} \, dV = \oint_S \mathbf{n} \cdot \boldsymbol{\sigma} \, dS + \int_V \rho \mathbf{f}_b \, dV. \quad (4)$$

The conservation law (4) can be written with respect to the initial (undeformed) configuration as

$$\int_{V_0} \rho_0 \frac{\partial \mathbf{v}}{\partial t} \, dV_0 = \int_{S_0} \mathbf{n}_0 \cdot (\boldsymbol{\Sigma} \cdot \mathbf{F}^T) \, dS_0 + \int_{V_0} \rho_0 \mathbf{f}_b \, dV_0, \quad (5)$$

where the subscripts 0 represent the quantities related to the initial configuration,  $\mathbf{u}$  is the displacement vector,  $\mathbf{v}$  is the velocity vector,  $\mathbf{F}$  is the deformation gradient tensor defined as

$$\mathbf{F} = \mathbf{I} + (\nabla \mathbf{u})^T, \quad (6)$$

and  $\boldsymbol{\Sigma}$  is the second Piola-Kirchhoff stress tensor, which is related to the Cauchy stress tensor by the following expression:

$$\boldsymbol{\sigma} = \frac{1}{\det \mathbf{F}} \mathbf{F} \cdot \boldsymbol{\Sigma} \cdot \mathbf{F}^T. \quad (7)$$

The displacement vector  $\mathbf{u}$  relates the position of the material particle in the initial configuration  $\mathbf{r}_0$  to the position of the material particle in the current configuration  $\mathbf{r}$ ,

$$\mathbf{u} = \mathbf{r} - \mathbf{r}_0. \quad (8)$$

Equation (5) represents the total Lagrangian description of the linear momentum conservation law. The incremental form of this equation can be written as

$$\int_{V_0} \rho_0 \frac{\partial \delta \mathbf{v}}{\partial t} \, dV_0 = \int_{S_0} \mathbf{n}_0 \cdot (\delta \boldsymbol{\Sigma} \cdot \mathbf{F}^T + \boldsymbol{\Sigma} \cdot \delta \mathbf{F}^T + \delta \boldsymbol{\Sigma} \cdot \delta \mathbf{F}^T) \, dS_0 + \int_{V_0} \rho_0 \delta \mathbf{f}_b \, dV_0, \quad (9)$$

where  $\delta \mathbf{u}$  is the displacement vector increment,  $\delta \mathbf{v}$  is the velocity vector increment,  $\delta \mathbf{F} = (\nabla \delta \mathbf{u})^T$  is the deformation gradient tensor increment,  $\delta \Sigma$  is the second Piola-Kirchhoff stress tensor increment, and  $\Sigma$  and  $\mathbf{F}$  are respectively the second Piola-Kirchhoff stress tensor and the deformation gradient tensor at the beginning of the current time increment.

In the case of the updated Lagrangian formulation used in this study, the last calculated configuration is always used as the reference configuration. Hence, the incremental form of the linear momentum conservation law in the updated Lagrangian formulation follows from Eqn. (9) by introducing the identity  $\mathbf{F} = \mathbf{I}$ :

$$\int_{V_u} \rho_u \frac{\partial \delta \mathbf{v}}{\partial t} dV_u = \int_{S_u} \mathbf{n}_u \cdot (\delta \Sigma_u + \Sigma_u \cdot \delta \mathbf{F}_u^T + \delta \Sigma_u \cdot \delta \mathbf{F}_u^T) dS_u + \int_{V_u} \rho_u \delta \mathbf{f}_b dV_u, \quad (10)$$

where the subscripts  $u$  represent quantities related to the updated configuration. After updating the configuration, the accumulated second Piola-Kirchhoff stress  $\Sigma_u + \delta \Sigma_u$  also needs to be transformed for the new configuration. This tensorial transformation is identical to the transformation of the second Piola-Kirchhoff stress tensor into the Cauchy stress tensor for the time increment prior to the configuration update.

In this study, the constitutive equation for a St. Venant-Kirchhoff material is used as the relation between the second Piola-Kirchhoff stress tensor and the Green-Lagrangian strain tensor:

$$\Sigma = 2\mu \mathbf{E} + \lambda \text{tr}(\mathbf{E}) \mathbf{I}, \quad (11)$$

where  $\mu$  and  $\lambda$  are the Lamé's coefficients and  $\mathbf{E}$  is the Green-Lagrangian strain tensor defined as

$$\mathbf{E} = \frac{1}{2} [\nabla \mathbf{u} + (\nabla \mathbf{u})^T + \nabla \mathbf{u} \cdot (\nabla \mathbf{u})^T]. \quad (12)$$

The incremental counterpart of the constitutive equation (11) reads:

$$\delta \Sigma = 2\mu \delta \mathbf{E} + \lambda \text{tr}(\delta \mathbf{E}) \mathbf{I}, \quad (13)$$

where  $\delta \mathbf{E}$  is the increment of the Green-Lagrangian strain tensor for the total Lagrangian description:

$$\delta \mathbf{E} = \frac{1}{2} [\nabla \delta \mathbf{u} + (\nabla \delta \mathbf{u})^T + \nabla \delta \mathbf{u} \cdot (\nabla \mathbf{u})^T + \nabla \mathbf{u} \cdot (\nabla \delta \mathbf{u})^T + \nabla \delta \mathbf{u} \cdot (\nabla \delta \mathbf{u})^T]. \quad (14)$$

The increment of the Green-Lagrangian strain tensor for the updated Lagrangian description follows from Eqn. (14) by introducing the identity  $\nabla \mathbf{u} = \mathbf{0}$ :

$$\delta \mathbf{E}_u = \frac{1}{2} [\nabla \delta \mathbf{u} + (\nabla \delta \mathbf{u})^T + \nabla \delta \mathbf{u} \cdot (\nabla \delta \mathbf{u})^T]. \quad (15)$$

When Eqn. (15) is substituted into Eqn. (13), one obtains the final form of the incremental constitutive relation for the updated Lagrangian description:

$$\delta \Sigma_u = \mu [\nabla \delta \mathbf{u} + (\nabla \delta \mathbf{u})^T] + \lambda \text{tr}(\delta \mathbf{u}) \mathbf{I} + \mu \nabla \delta \mathbf{u} \cdot (\nabla \delta \mathbf{u})^T + \frac{1}{2} \lambda (\nabla \delta \mathbf{u} : \nabla \delta \mathbf{u}) \mathbf{I}. \quad (16)$$

Introducing constitutive equation (16) into Eqn. (10), one obtains the linear momentum conservation equation for an elastic solid in the updated Lagrangian description, with the displacement vector increment  $\delta \mathbf{u}$  as the primitive variable:

$$\int_{V_u} \rho_u \frac{\partial \delta \mathbf{v}}{\partial t} dV_u - \oint_{S_u} \mathbf{n}_u \cdot (2\mu + \lambda) \nabla \delta \mathbf{u} dS_u = \oint_{S_u} \mathbf{n}_u \cdot \mathbf{q} dS_u + \int_{V_u} \rho_u \delta \mathbf{f}_b dV_u. \quad (17)$$

where

$$\begin{aligned} \mathbf{q} = & \mu(\nabla\delta\mathbf{u})^T + \lambda \operatorname{tr}(\delta\mathbf{u}) \mathbf{I} - (\mu + \lambda)\nabla\delta\mathbf{u} \\ & + \mu \nabla\delta\mathbf{u} \cdot (\nabla\delta\mathbf{u})^T + \frac{1}{2}\lambda (\nabla\delta\mathbf{u} : \nabla\delta\mathbf{u}) \mathbf{I} + \Sigma_u \cdot \delta\mathbf{F}_u^T + \delta\Sigma_u \cdot \delta\mathbf{F}_u^T. \end{aligned} \quad (18)$$

Tensor  $\mathbf{q}$  consists of nonlinear and coupling terms which are treated explicitly after the discretisation so that the discretised equation can be solved by using a segregated algorithm. According to Jasak and Weller [10], the efficiency of the segregated solution procedure can be improved by using the diffusivity  $(2\mu + \lambda)$  in the Laplacian at the left-hand side of Eqn. (17).

The specification of the problem is completed with the definition of the domain in space and time and the initial and boundary conditions. The initial condition consists of the specified distribution of  $\delta\mathbf{u}$  and  $\delta\mathbf{v}$  at time zero. The boundary conditions, either constant or time varying, can be of the following type: fixed displacement increment, plane of symmetry, fixed pressure increment, fixed traction increment and free surface.

The fixed pressure increment and the fixed traction increment boundary conditions are implemented as a fixed normal derivative Neumann boundary condition on the displacement increment. The normal derivative of the displacement increment is calculated from the specified traction and pressure increment as follows. Let  $\delta\mathbf{t}$  and  $\delta p$  be the traction and pressure increment related to the current configuration, then

$$\mathbf{n}_u \cdot \delta\Sigma_u = \left[ (\delta\mathbf{t} - \delta p \mathbf{n}) \frac{dS}{dS_u} - \mathbf{n}_u \cdot \Sigma_u \cdot \delta\mathbf{F}_u^T \right] \cdot (\mathbf{F}_u^{-1})^T, \quad (19)$$

where  $\mathbf{F}_u$  is the deformation gradient tensor with respect to the updated configuration:

$$\mathbf{F}_u = \mathbf{I} + \delta\mathbf{F}_u = \mathbf{I} + (\nabla\delta\mathbf{u})^T. \quad (20)$$

By introducing Eqn. (16) into Eqn. (19), the normal derivative of the displacement increment at the boundaries with specified traction and pressure increment can be expressed in the following way:

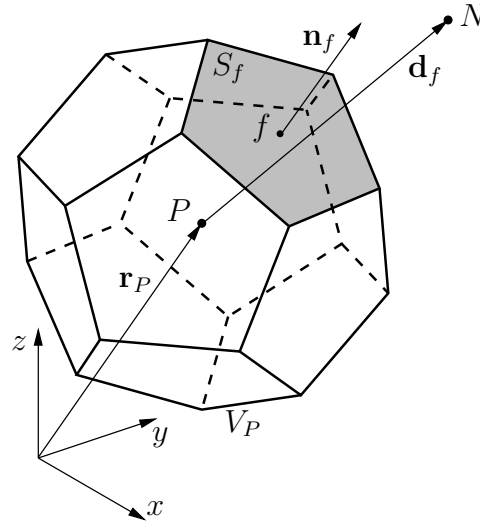
$$\begin{aligned} \mathbf{n}_u \cdot \nabla\delta\mathbf{u} = & \left[ (\delta\mathbf{t} - \delta p \mathbf{n}) \frac{dS}{dS_u} - \mathbf{n}_u \cdot \Sigma_u \cdot \delta\mathbf{F}_u^T \right] \cdot (\mathbf{F}_u^{-1})^T / (2\mu + \lambda) \\ & - \mathbf{n}_u \cdot \left[ \mu(\nabla\delta\mathbf{u})^T + \lambda \operatorname{tr}(\delta\mathbf{u}) \mathbf{I} - (\mu + \lambda)\nabla\delta\mathbf{u} \right] / (2\mu + \lambda) \\ & - \mathbf{n}_u \cdot \left[ \mu \nabla\delta\mathbf{u} \cdot (\nabla\delta\mathbf{u})^T + \frac{1}{2}\lambda (\nabla\delta\mathbf{u} : \nabla\delta\mathbf{u}) \mathbf{I} \right] / (2\mu + \lambda). \end{aligned} \quad (21)$$

### 3. Finite volume discretisation and solution procedure

The finite volume (FV) discretisation is based on the integral form of the conservation equation over a fixed or moving control volume (CV). The discretisation procedure is divided into two parts: discretisation of the computational domain and equation discretisation.

#### 3.1. Discretisation of the computational domain

The discretisation of the computational domain consists of the discretisation of the time interval and the discretisation of space. The time interval is split into a finite number of time-steps  $\delta t$  and the equations are solved in a time-marching manner. The unstructured FVM discretises the computational space by splitting it into a finite number of convex polyhedral cells bounded by



**Fig. 1** Polyhedral control volume (cell)

convex polygons. The cells do not overlap and fill the space completely. Fig. 1 shows a sample cell around the computational point  $P$  located in its centroid, the face  $f$ , the face area  $S_f$ , the face unit normal vector  $\mathbf{n}_f$  and the centroid  $N$  of the neighbouring CV sharing the face  $f$ .

In this study, the dynamic behaviour of the structure is described by the linear momentum conservation law (17) written in the incremental updated Lagrangian formulation. Therefore, the computational mesh must be moved at the beginning of each time step by using the displacement increment vector obtained in the previous time step. In the applied cell-centred finite volume method the unknown displacement increment is calculated at the control volume centre. In order to move the mesh, the displacement increment must be interpolated from the control volume centres to the control volume points [11]. This is done by using the following weighted averaging interpolation:

$$\delta \mathbf{u}_p = \frac{\sum_c w_{pc} [\delta \mathbf{u}_c + (\mathbf{r}_p - \mathbf{r}_c) \cdot (\nabla \delta \mathbf{u})_c + \frac{1}{2} (\mathbf{r}_p - \mathbf{r}_c)^2 \cdot (\nabla \nabla \delta \mathbf{u})_c]}{\sum_c w_{pc}}, \quad (22)$$

where  $\delta \mathbf{u}_p$  is the point displacement increment,  $\delta \mathbf{u}_c$  is the cell centre displacement increment and the summation is executed over all cells sharing the point  $p$ . The weighting factor  $w_{pc}$  is defined as

$$w_{pc} = \frac{1}{|\mathbf{r}_p - \mathbf{r}_c|}, \quad (23)$$

where  $\mathbf{r}_c$  is the cell centre position vector and  $\mathbf{r}_p$  is the point position vector.

### 3.2. Equation discretisation

The second-order FV discretisation of an integral conservation equation transforms the surface integrals into sums of face integrals and approximates them and the volume integrals to the second order accuracy by using the mid-point rule. The spatially discretised form of Eqn. (17)

for the updated control volume  $V_{P_u}$  reads:

$$\begin{aligned} \rho_{P_u} V_{P_u} \frac{\partial \delta \mathbf{v}_P}{\partial t} - \sum_f (2\mu_f + \lambda_f) \mathbf{n}_{f_u} \cdot (\nabla \delta \mathbf{u})_f S_{f_u} \\ = \sum_f \mathbf{n}_{f_u} \cdot \mathbf{q}_f S_{f_u} + \sum_f \rho_u (\delta \mathbf{f}_b)_P V_{P_u}, \end{aligned} \quad (24)$$

where the subscripts  $P$  and  $f$  represent the cell and face values. The face values of all dependent variables are calculated by linear interpolation of the neighbouring cell values.

The face normal derivative of the displacement increment  $\mathbf{n}_{f_u} \cdot (\nabla \delta \mathbf{u})_f$  is discretised as follows (see [12, 13]):

$$\mathbf{n}_{f_u} \cdot (\nabla \delta \mathbf{u})_f = \underbrace{|\Delta_{f_u}| \frac{\delta \mathbf{u}_N - \delta \mathbf{u}_P}{|\mathbf{d}_{f_u}|}}_{\text{Orthogonal contribution}} + \underbrace{(\mathbf{n}_{f_u} - \Delta_{f_u}) \cdot (\nabla \delta \mathbf{u})_f}_{\text{Non-orthogonal correction}}, \quad (25)$$

where  $\Delta_f = \frac{\mathbf{d}_f}{\mathbf{d}_f \cdot \mathbf{n}_f}$ , Fig. 1. The orthogonal contribution in Eqn. (25) is treated implicitly, while the non-orthogonal correction is explicit.

The cell gradient of the displacement increment vector used for the calculation of the tensor  $\mathbf{q}$  and the non-orthogonal correction, is obtained by using the least-squares fit [14]. This method produces a second-order accurate gradient irrespective of the mesh quality [11].

The temporal discretisation of Eqn. (24) is performed by using two implicit schemes: the first-order accurate Euler scheme and the second-order accurate three time level scheme [15] referred to as the backward scheme. In both schemes, all terms of Eqn. (24) are evaluated at the new time instance  $t^n = t^o + \delta t$ , and the velocity increment  $\delta \mathbf{v}$  and its temporal derivative are discretised by using one old-time level in the Euler scheme:

$$\delta \mathbf{v}_P^n = \frac{\delta \mathbf{u}_P^n - \delta \mathbf{u}_P^o}{\delta t}, \quad (26)$$

$$\left( \frac{\partial \delta \mathbf{v}_P}{\partial t} \right)^n = \frac{\delta \mathbf{v}_P^n - \delta \mathbf{v}_P^o}{\delta t}, \quad (27)$$

and two old-time levels in the backward scheme:

$$\delta \mathbf{v}_P^n = \frac{3\delta \mathbf{u}_P^n - 4\delta \mathbf{u}_P^o + \delta \mathbf{u}_P^{oo}}{2\delta t}, \quad (28)$$

$$\left( \frac{\partial \delta \mathbf{v}_P}{\partial t} \right)^n = \frac{3\delta \mathbf{v}_P^n - 4\delta \mathbf{v}_P^o + \delta \mathbf{v}_P^{oo}}{2\delta t}, \quad (29)$$

where  $\delta \mathbf{v}_P^n = \delta \mathbf{v}_P(t^o + \delta t)$ ,  $\delta \mathbf{v}_P^o = \delta \mathbf{v}_P(t^o)$  and  $\delta \mathbf{v}_P^{oo} = \delta \mathbf{v}_P(t^o - \delta t)$ .

When Eqs. (25, 28 and 29) are substituted into Eqn. (24), the fully discretised form of the linear momentum conservation law (17) can be written in the form of linear algebraic equation, which for cell  $P$  reads:

$$a_P \delta \mathbf{u}_P^n + \sum_N a_N \delta \mathbf{u}_N^n = \mathbf{r}_P, \quad (30)$$

where the diagonal coefficient  $a_P$ , the neighbour coefficient  $a_N$  and the source term  $\mathbf{r}_P$  are defined for the backward scheme by the following expressions:

$$a_P = \frac{9\rho_{P_u} V_{P_u}}{4\delta t} + \sum_f (2\mu_f^n + \lambda_f^n) \frac{|\Delta_{f_u}|}{|\mathbf{d}_{f_u}|} S_{f_u}, \quad (31)$$

$$a_N = - \sum_f (2\mu_f^n + \lambda_f^n) \frac{|\Delta_{fu}|}{|\mathbf{d}_{fu}|} S_{fu}, \quad (32)$$

$$\begin{aligned} \mathbf{r}_N = & \rho_{Pu} V_{Pu} \left( \frac{3\delta\mathbf{u}_P^o}{\delta t^2} - \frac{3\delta\mathbf{u}_P^{oo}}{4\delta t^2} + \frac{2\delta\mathbf{v}_P^o}{\delta t} + \frac{\delta\mathbf{v}_P^{oo}}{2\delta t} \right) \\ & + \sum_f (2\mu_f + \lambda_f) (\mathbf{n}_{fu} - \Delta_{fu}) \cdot (\nabla\delta\mathbf{u})_f^n S_{fu} \\ & + \sum_f \mathbf{n}_{fu} \cdot \mathbf{q}_f^n S_{fu} + \sum_f \rho_{Pu} (\delta\mathbf{f}_b)_P^n V_{Pu}. \end{aligned} \quad (33)$$

### 3.3. Solution procedure

Equation (30) is assembled for all CVs in the computational mesh, resulting in the following system of linear algebraic equations:

$$[A] \cdot \{\delta\mathbf{u}\} = \{\mathbf{r}\}, \quad (34)$$

where  $[A]$  is the sparse matrix, with coefficients  $a_P$  on the diagonal and  $a_N$  off the diagonal,  $\{\delta\mathbf{u}\}$  is the vector consisting of displacement increments  $\delta\mathbf{u}_P$  for all CVs and  $\{\mathbf{r}\}$  is the right-hand side vector consisting of source terms  $\mathbf{r}_P$  for all CVs.

The system of equations (34) is solved by using a segregated algorithm, where the three components of the displacement vector increment are solved separately from each other. Since nonlinear and coupling terms depending on the unknown displacement vector increment are placed in the right-hand side vector, the system is solved in the iterative manner where, each outer iteration the right hand-side vector  $\{\mathbf{r}\}$  is updated at the beginning of each outer iteration by using the displacement vector increment from the previous iteration. When the solution changes less than some pre-defined tolerance, the system is considered to be solved. This is done for every time-step of the transient simulation.

The matrix  $[A]$  from Eqn. (34) is symmetric and diagonally dominant and the corresponding system of equations is solved by using the Incomplete Cholesky Conjugate Gradient iterative solver (ICCG) [16]. There is no need to solve the system to a fine tolerance since the right-hand side vector is only an approximation based on the displacement vector increment from the previous iteration.

At the end of each time step, the total displacement vector and the total second Piola-Kirchhoff stress tensor are updated as follows:

$$\mathbf{u}^n = \mathbf{u}^o + \delta\mathbf{u}^n, \quad (35)$$

$$\Sigma_u^n = \Sigma_u^o + \delta\Sigma_u^n. \quad (36)$$

where the second Piola-Kirchhoff stress tensor increment  $\delta\Sigma_u^n$  is calculated according to Eqn. (16) by using the newly calculated displacement vector increment  $\delta\mathbf{u}^n$ . When the mesh is moved at the beginning of the new time step by using the displacement increment from the previous time step, the second Piola-Kirchhoff stress tensor related to the old updated configuration must be transformed to the new updated configuration:

$$\Sigma_u^o = \frac{1}{\det \mathbf{F}_u^o} \mathbf{F}_u^o \cdot \Sigma_u^o \cdot (\mathbf{F}_u^o)^T. \quad (37)$$

### 3.4. Implementation of the solver

The above described computational structural dynamic (CSD) solver is implemented in OpenFOAM [17, 18]. OpenFOAM is an Open Source object-oriented C++ library for numerical simulations in continuum mechanics, which implements the solvers by mimicking the governing equations in software. Figure 2 shows a part of the source code of the CSD solver implementation.

```

for (runTime++; !runTime.end(); runTime++)
{
    Info<< "Time = " << runTime.timeName() << nl << endl;

#    include "moveMesh.H"

    sigma = (F & sigma & F.T())/det(F);

    int iCorr = 0;

    do
    {
        fvVectorMatrix DUEqn
        (
            rho*fvm::d2dt2(DU)
            == fvm::laplacian(2*mu + lambda, DU)
            - fvc::laplacian(mu + lambda, DU)
            + fvc::div
            (
                mu*gradDU.T()
                + lambda*(I*tr(gradDU))
                + mu*(gradDU & gradDU.T())
                + 0.5*lambda*(gradDU && gradDU)*I
                + (sigma + Dsigma) & DF.T()
            )
        );

        solverPerf = DUEqn.solve();

        gradDU = fvc::grad(DU);
        DF = gradDU.T();
        Depsilon = symm(gradDU) + 0.5*(gradDU & gradDU.T());
        Dsigma = 2*mu*Depsilon + I*(lambda*tr(Depsilon));
    }
    while
    (
        solverPerf.initialResidual() > convergenceTolerance
        && ++iCorr < nCorr
    );

    F = I + DF;
    U += DU;
    sigma += Dsigma;
}

```

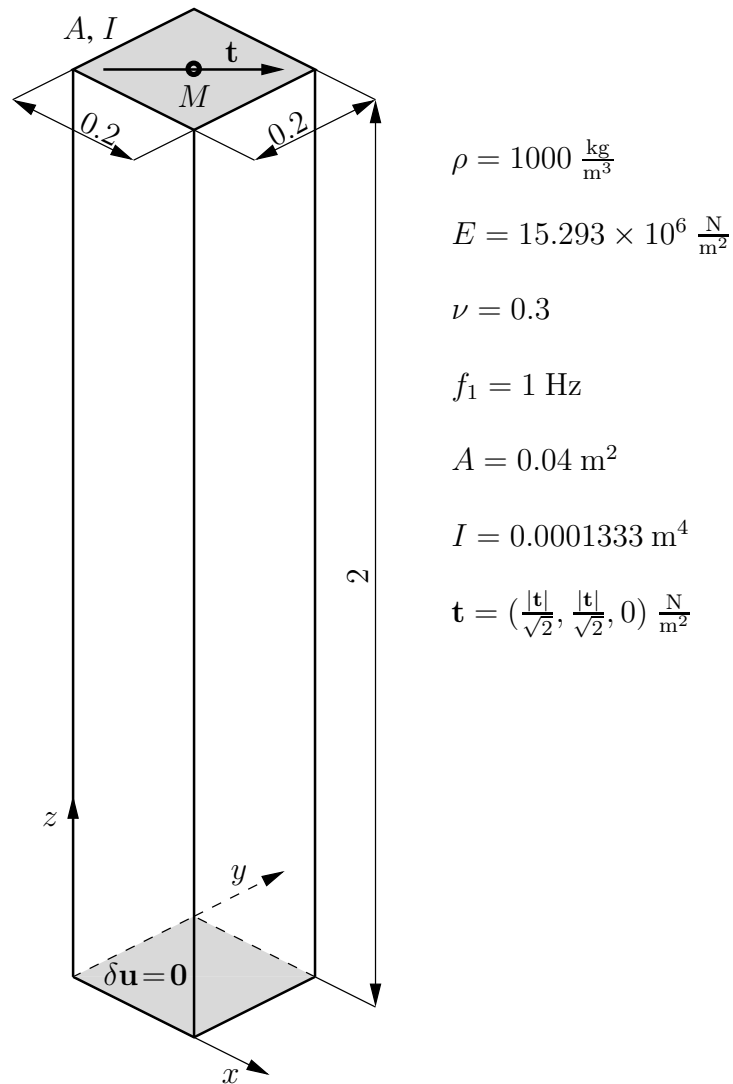
**Fig. 2** Implementation of the CSD solver in OpenFOAM.

#### 4. Test cases

The methodology is first tested on the case of the large amplitude vibration of a 3-D cantilevered beam for which an exact solution for beam deflection exists. The dynamic response of an axial turbine blade discretised by using an unstructured mesh is simulated in the second case.

##### 4.1. Large amplitude vibration of a 3-D cantilevered beam

In this test case, the large amplitude flexural vibration of a 3-D cantilevered beam (Fig. 3) is analysed. The density  $\rho$  and Young's modulus of elasticity  $E$  are selected to obtain the beam first natural frequency equal to  $f_1 = 1$  Hz. The beam is constrained at one end ( $\delta \mathbf{u} = \mathbf{0}$ ) and subjected to a suddenly applied traction force  $\mathbf{t}$  at the other end. The rest of the beam boundary is traction free. The volume of the beam is discretised by the structured computational mesh consisting of hexahedral control volumes (CV).



**Fig. 3** Geometry, physical properties and loading of the 3-D beam, where  $A$  is the beam cross-section area,  $I$  is the area moment of inertia about the neutral axis, and  $\nu$  is the Poisson's ratio.

First, the static equilibrium shape of the beam is calculated for several traction force magnitudes. The calculation is performed by using a transient solution procedure with the

Euler temporal discretisation. In order to efficiently obtain the stationary solution by using the transient solution procedure, the damping term of the form

$$\int_{V_u} K \frac{\partial \delta \mathbf{u}}{\partial t} dV_u \quad (38)$$

is added to the left-hand side of momentum equation (17), where  $K$  is the damping coefficient. When a steady state is reached, the influence of the damping term vanishes.

Fig. 4 shows the calculated static equilibrium shape of the beam for different loading magnitudes represented by the dimensionless traction force

$$\mathcal{F} = \frac{|\mathbf{t}|AL^2}{EI}. \quad (39)$$

The surface of the beam is coloured by the equivalent Cauchy stress. Table 1 shows a comparison of the calculated deflection of the point M (see Fig. 3),

$$d_M = \sqrt{\mathbf{u}_x + \mathbf{u}_y}, \quad (40)$$

with the exact deflection  $d_{Ma}$  calculated by Mattiasson [19] using the numerical evaluation of elliptic integrals. In order to determine the spatial accuracy of the method, the calculation is done for two different resolutions of the computational mesh:  $6 \times 6 \times 60$  and  $10 \times 10 \times 100$  CVs. Only for an extreme beam deformation the accuracy is less than the second order.

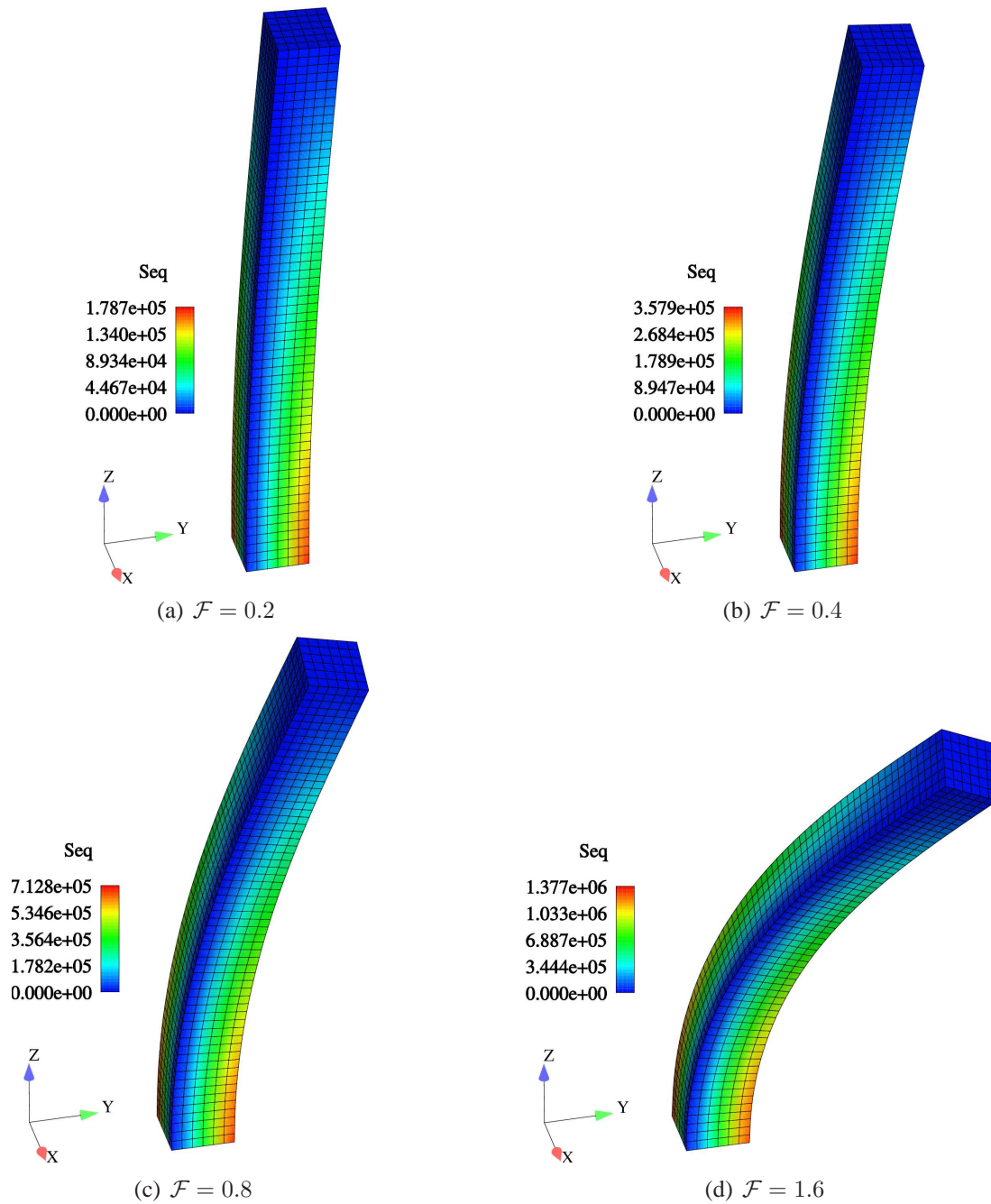
**Table 1** Steady state numerical solution and its accuracy, where  $E$  is the relative error

$\mathcal{F}$	$d_{Ma}$	$d_{M,6x60}$	$E_{6x60}$	$d_{M,10x100}$	$E_{10x100}$	Accuracy
0.2	0.13272	0.1255	5.44 %	0.1309	1.37 %	2.68
0.4	0.26196	0.2481	5.29 %	0.2581	1.47 %	2.50
0.8	0.49890	0.4797	3.85 %	0.4909	1.60 %	1.71
1.6	0.85882	0.8290	3.47 %	0.8507	0.95 %	1.50

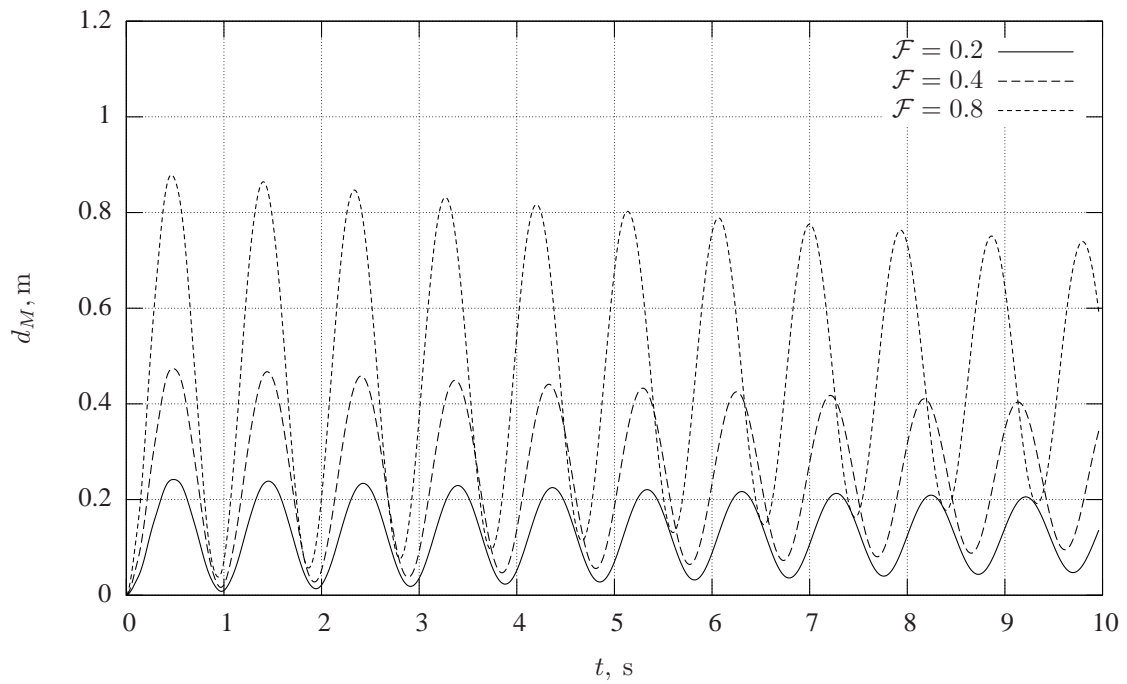
In the second phase, the dynamic response of the beam to a suddenly applied traction force is calculated. As no damping effect exists in the considered elastic system, one would expect an undamped vibration of the beam about its static equilibrium. The damping effect may occur only as a consequence of the numerical dissipation introduced by the applied numerical methodology.

The transient simulation is performed on the computational mesh with the resolution of  $6 \times 6 \times 60$  and with the time step size 0.002 s corresponding to a maximum wave speed Courant number  $Co$  of 6, where  $Co = c(\Delta t/\Delta s)$ ,  $\Delta s$  is the minimum cell dimension, and  $c = \sqrt{E/\rho}$  is the unconfined solid wave speed.

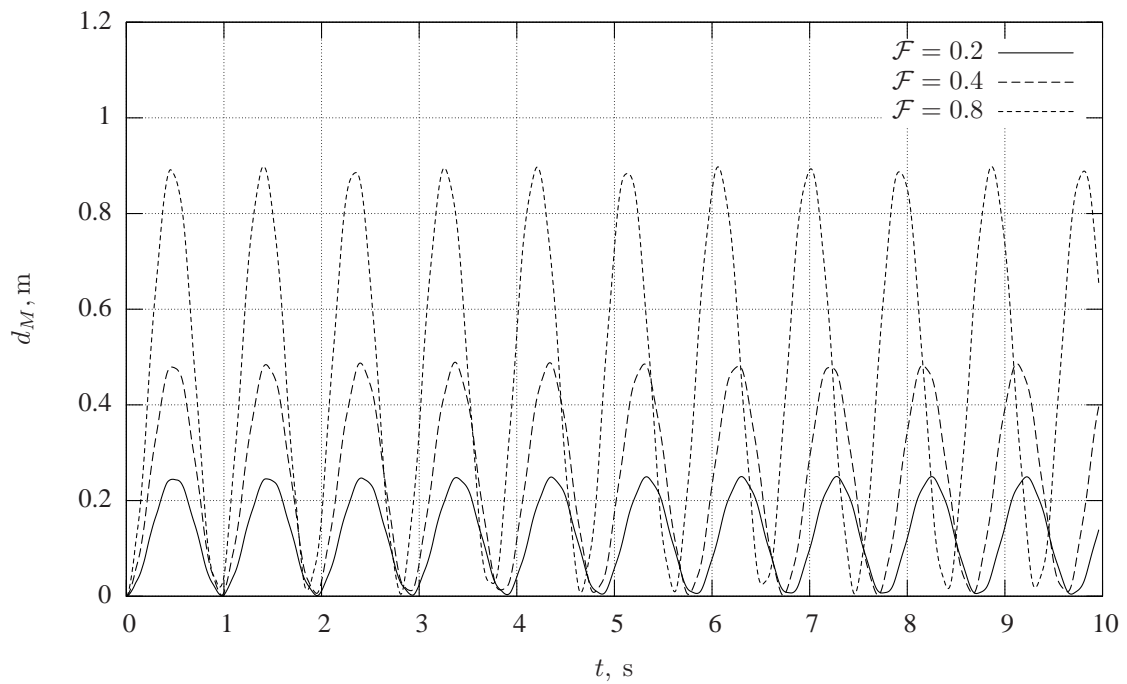
Figures 5 and 6 show the calculated deflection  $d_M$  of the point  $M$  as a function of time obtained by using the Euler and backward temporal discretisation scheme for various loadings. One can note that both schemes predict vibration frequency close to the first natural frequency of the beam. Also as expected, the frequency increases with the amplitude of the vibrations. The predicted mean deflection of the point M is in accordance with the static deflection calculated in the first phase. The vibration amplitude decays significantly with time when the Euler temporal discretisation scheme is used. When the temporal discretisation is performed with the second order accurate backward scheme, the amplitude of vibration remains constant with time. The stability of the calculation is preserved over a long time period; the simulation is performed without a break over a time period of 100 s, which is prerequisite for an FSI simulation.



**Fig. 4** Steady state shape of the beam for different loading magnitudes.



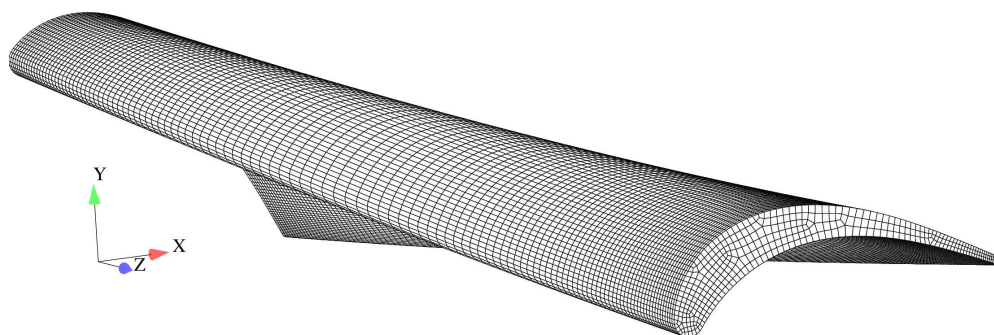
**Fig. 5** Deflection history of the beam tip for Euler temporal discretisation.



**Fig. 6** Deflection history of the beam tip for backward temporal discretisation.

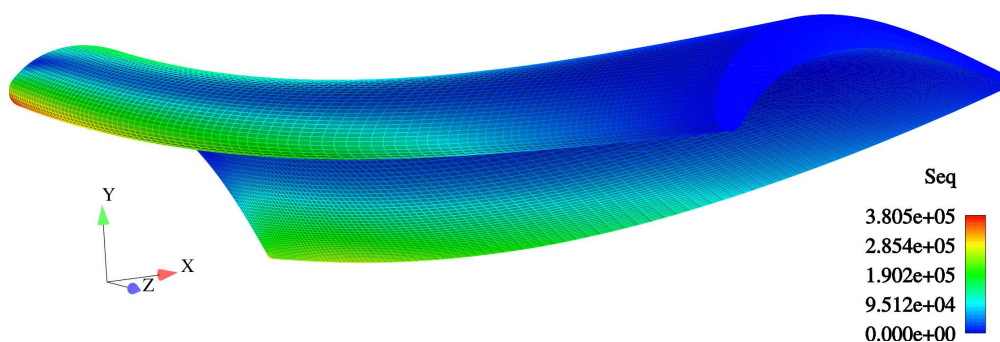
#### 4.2. Dynamic response of an axial turbine blade

In order to show the performance of the CSD solver on a real engineering problem, the dynamic response of a twisted axial turbine blade is simulated. The blade of 0.8 m in height is made by the extrusion of an aerodynamically optimised profile [20] of 0.2 m cord, Fig. 7. The twist angle of the blade is  $40^\circ$  over the blade height. Mechanical properties of the blade material are the same as of the 3-D beam. The blade is constrained at one end ( $\delta \mathbf{u} = \mathbf{0}$  at  $z = 0$ ) and subjected to a suddenly applied traction force  $\mathbf{t} = (1500, 1500, 0)$  Pa at the other end ( $z = 0.8$ ). The rest of the blade boundary is traction free. The unstructured mesh consists of 50000 hexahedral cells.



**Fig. 7** Geometry and computational mesh for the axial turbine blade.

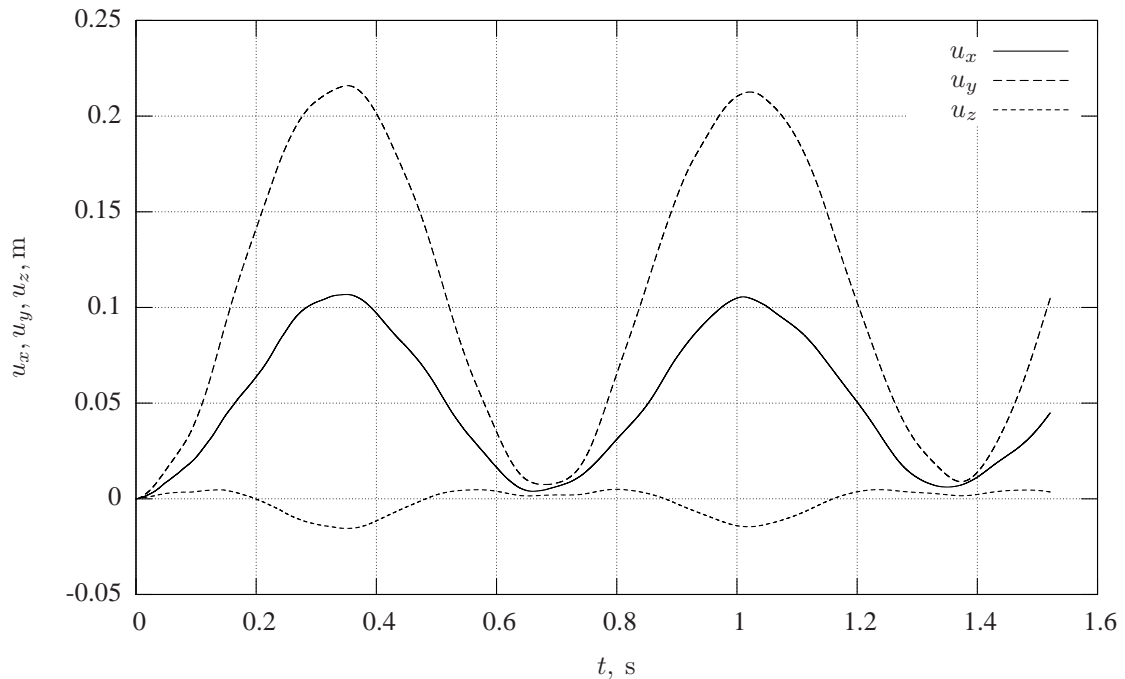
As in the previous test case, we first compute the static equilibrium shape of the blade, which is shown in Fig. 8, where the blade surface is coloured by the equivalent Cauchy stress. Results of the dynamic response of the blade are shown in Fig. 9 as a temporal variation of the displacement vector components of the point at the top of the blade leading edge. The backward temporal discretisation scheme with the time step of 0.00001 s is used for the transient calculation. The complete simulation consists of 152700 time steps and has been performed in 200 h on an Intel Pentium 4 CPU 2.8 GHz machine.



**Fig. 8** Steady state shape of the axial turbine blade.

## 5. Conclusion

This paper presents the numerical methodology for the calculation of a large deformation dynamic response of an elastic body. The linear momentum conservation law in the updated Lagrangian formulation is discretised in space by using the second order accurate finite volume method and in time by using the first order accurate Euler scheme and the second order



**Fig. 9** Displacement vector components of the point at the top of the blade leading edge as a function of time.

accurate backward scheme. The system of linearised and decoupled algebraic equations for displacement increment is solved by using the segregated solution procedure.

The methodology is first validated on the calculation of the static shape of a 3-D cantilevered beam loaded at the free end. The predicted deflection of the beam is in good agreement with the exact solution. Numerical results of the beam vibration simulation show a good performance of the solver in respect to the frequency and amplitude prediction. Simulation of the dynamic response of an axial turbine twisted blade shows that the solver can be used for a structural dynamic simulation of a complex elastic body discretised with an unstructured computational mesh. Hence, the CSD solver described in this paper can be used as part of the self-contained FVM FSI solver for the FSI simulation with large structural deformations.

## REFERENCE

- [1] Wall, W. A., Forster, C., Neumann, M., Ramm, E., Advances in fluid-structure interaction, K. Gurlbeck, C. Konke (Editors) Proceedings of 17th international conference on the application of computer science and mathematics in architecture and civil engineering, Weimar, Germany, 2006.
- [2] Taylor, C. A., Hughes, T. J. R., Zarins, C. K., Finite element modeling of blood flow in arteries, Computer methods in applied mechanics and engineering 158 (1998), pp. 155–196.
- [3] Slone, A. K., Pericleous, K., Bailey, C., Cross, M., Bennett, C., A finite volume unstructured mesh approach to dynamic fluid-structure interaction: an assessment of the challenge of predicting the onset of flutter, Applied mathematical modelling 28 (2004), pp. 211–239.

- [4] Lv, X., Yhao, Y., Huang, X. Y., Xia, G. H., Su, X. H., A matrix-free implicit unstructured multigrid finite volume method for simulating structural dynamics and fluid-structure interaction, *Journal of computational physics* x (x) (2007), pp. x–x.
- [5] Karač, A., Drop impact of fluid-filled polyethylene containers, Ph.D. thesis, Imperial College, University of London, 2003.
- [6] Bath, K. J., Hahn, W. F., On transient analysis of fluid structure systems, *Computers and structures* 10 (1997), pp. 383–391.
- [7] Maneeratana, K., Development of the finite volume method for non-linear structural applications, Ph.D. thesis, Imperial College, University of London, 2000.
- [8] Slone, A. K., Bailey, C., Cross, M., Dynamic solid mechanics using finite volume methods, *Applied mathematical modelling* 27 (2003), pp. 69–87.
- [9] Demirdžić, I., Perić, M., Space conservation law in finite volume calculations of fluid flow, *International journal for numerical methods in fluids* 8 (1988), pp. 1037–1050.
- [10] Jasak, H., Weller, H. G., Application of the finite volume method and unstructured meshes to linear elasticity, *International journal for numerical methods in engineering* 48 (2000), pp. 267–287.
- [11] Jasak, H., Tuković, Ž., Automatic mesh motion for the unstructured finite volume method, *Transaction of FAMENA* 30 (2) (2006), pp. 1–20.
- [12] Jasak, H., Error analysis and estimation for finite volume method with applications to fluid flows, Ph.D. thesis, Imperial College, University of London, 1996.
- [13] Tuković, Ž., Finite volume method on domains of varying shape (in Croatian), Ph.D. thesis, Faculty of Mechanical Engineering and Naval Architecture, University of Zagreb, 2005.
- [14] Demirdžić, I., Muzaferija, S., Numerical method for coupled fluid flow, heat transfer and stress analysis using unstructured moving meshes with cells of arbitrary topology, *Computer methods in applied mechanics and engineering* 125 (1995), pp. 235–255.
- [15] Ferziger, J. H., Perić, M., *Computational methods for fluid dynamics*, Springer Verlag, Berlin-New York, 1995.
- [16] Jacobs, D. A. H., Preconditioned Conjugate Gradient methods for solving systems of algebraic equations, Technical Report RD/L/N193/80, Central Electricity Research Laboratories, 1980.
- [17] Jasak, H., Weller, H., Nordin, N., In-cylinder CFD simulation using a C++ object-oriented toolkit, SAE 2004 World Congress & Exhibition, SAE Technical Paper 2004-01-0110, 2004.
- [18] Weller, H. G., Tabor, G., Jasak, H., Fureby, C., A tensorial approach to computational continuum mechanics using object orientated techniques, *Computers in physics* 12 (6) (1998), pp. 620–631.

- [19] Mattiasson, K., Numerical results from large deflection beam and frame problems analysed by means of elliptic integrals, *International journal for numerical methods in engineering* 17 (1) (1981), pp. 145–153.
- [20] Tuković, Ž., The flow characteristics of aerodynamically optimal cascades of axial turbines (in Croatian), Master's thesis, Faculty of Mechanical Engineering and Naval Architecture, University of Zagreb, 2001.

Submitted: 16.4.2007.

Accepted: 18.6.2007.

Dr. sc. Željko Tuković  
zeljko.tukovic@fsb.hr  
Faculty of Mechanical Engineering  
and Naval Architecture,  
University of Zagreb,  
Ivana Lučića 5,  
10 000 Zagreb, Croatia

Doc. dr. sc. Hrvoje Jasak  
h.jasak@wikki.co.uk  
Wikki Ltd.  
33 Palmerston House,  
60 Kensington Place,  
London W8 7PU, England