

# COMPARISON OF MONOLITHIC AND PARTITIONED APPROACHES FOR PONDING ANALYSIS ON MEMBRANE STRUCTURES

Navaneeth K Narayanan<sup>1,2</sup>, Roland Wüchner<sup>2</sup> and Joris Degroote<sup>1,3</sup>

<sup>1</sup> Department of Electromechanical, Systems and Metal Engineering, Ghent University  
Sint-Pietersnieuwstraat 41, 9000 Ghent, Belgium  
navaneeth.narayanan@ugent.be

<sup>2</sup> Chair of Structural Analysis, Technical University of Munich  
Arcisstr. 21, 80333 Munich, Germany

<sup>3</sup> Flanders Make, Belgium

**Key words:** Monolithic method, Partitioned method, Hydrostatic load.

**Abstract.** Membrane structures are vulnerable to ponding due to their large deformation characteristic. The ponding on membrane structures is usually caused by rainfall on an already deformed structure due to a seeding event such as snow accumulation. This paper discusses two monolithic methods and a partitioned method for determining the static deformation of the membrane structure due to a given volume of ponding water. The monolithic methods involve simultaneously solving the structural equations and the fluid equations under static conditions, to obtain the structural deformation. The partitioned method on the other hand involves external coupling iterations involving a structural solver and volume conserving solver, where the volume conserving solver is responsible for updating the free surface to maintain a given volume of water. The discussed methods are compared in terms of robustness and computing time. It was found that the monolithic methods were computationally efficient. However, the partitioned method-apart from being modular-was found to be more robust with quasi-Newton convergence accelerators.

## 1 INTRODUCTION

Ponding on membrane structures is a phenomenon of water accumulation on a localized region where the structure deforms under the load of certain volume of water. This is in most cases initiated by a seeding event such as snow accumulation. Following the seeding event, if there is rainfall, water is gradually filled in the local depression leading to further deformation. In the extreme event, due to the absence of sufficient stiffness, it may lead to a vicious cycle of further deformation and accumulation of water resulting in a structural collapse. Moreover, if the rain is accompanied by strong wind, the wind flow around the structure can cause large vertical oscillations, as happened in 2011 during the Pukkelpop festival held in Kiewit (Belgium). This phenomenon is one of the reasons for the extensive damage of the festival tents. The investigation of such events requires fluid-structure interaction (FSI) simulation involving membrane structure, water and wind flow. Imposing an initial condition on the FSI simulation with a pond on a membrane structure requires computation of static deformation of the

membrane structure under the load of fixed volume of ponding water, which is the main motivation of the current work.

In the literature, there are some studies related to the problem discussed in this work. One such study is by Szyszkowski and Glockner [1] which deals with ponding stability and deformation on spherical inflatables by solving axi-symmetric membrane equations with hydrostatic loads. The use of axi-symmetric membrane equations can be also found in [2], where the author has focussed on deformations of initially flat, simply supported circular membrane structures under load of gradually accumulating fluid. A more general approach in the finite element framework is discussed in [3] where they linearize the static behaviour of an incompressible fluid under gravity to obtain the symmetric load stiffness matrix used in the Newton-Raphson (N-R) iterations. A more recent paper by Hoareau and Deü [4, 5] uses a level set approach for the numerical integration on the loaded surface to compute volume, nodal forces and load stiffness matrix, where the element faces were part of a quadratic hexahedral mesh. They computed the deformed shape of tanks partially filled with liquid by performing volume conservation in every structural N-R iteration with the added load stiffness matrix discussed in [3].

The first monolithic method discussed in this paper involves free surface update using volumes-conserving algorithm after every structural N-R iteration and is similar to the method discussed in [4, 5]. The difference lies in the use of a more efficient and robust method for volume-conservation, because our applications comprises of much more flexible structures, sometimes almost flat geometry at the beginning of the solution procedure. The second monolithic method introduced in the paper solves the structural equations under hydrostatic loads with an additional constraint equation that the fluid volume should be equal to the target volume. The combined system of equations is solved using the N-R method by linearizing the equations with this constraint. The proposed method is therefore called monolithic method with volume conservation as constraint (MVCC), while the first monolithic method is called monolithic method with volume conservation inside structural iterations (MVCIS). The MVCC method was developed because it was found that the MVCIS method which imposes strict volume conservation at every N-R iteration failed to converge in some numerical experiments.

The monolithic methods discussed above can only be implemented if the structural solver can be accessed for modification. In some cases, where the solver is a commercial software, this is not possible and therefore the ponding analysis can be only performed using the partitioned approach. One example of this approach is presented in the work of Bown et al. [6], where an in-house structural code inTENS is coupled with a shallow water solver in a partitioned method to analyze ponding on tensioned membrane structures. The partitioned approach discussed in this paper involves external coupling iterations between a structural solver and the volume-conserving solver. In every coupling iteration, the output of one solver is used as an input to the other solver. The coupling iterations are continued till there is no change in the structural displacements. To stabilize and accelerate the convergence of the coupling iterations, convergence accelerators such as Aitken relaxation [7, 8] and IQN-ILS [9] are used.

## 2 MATHEMATICAL FORMULATION

Let us consider a membrane structure whose surface is denoted by  $\partial\Omega_s$ . We seek the equilibrium shape of the membrane structure under the load of a fixed target volume  $V_t$  of an incompressible fluid with specific gravity  $\gamma_f$ . This requires that the equilibrium equations of the structure under the hydrostatic load from the fluid are satisfied along with the constraint that the fluid volume should be equal to the

target fluid volume.

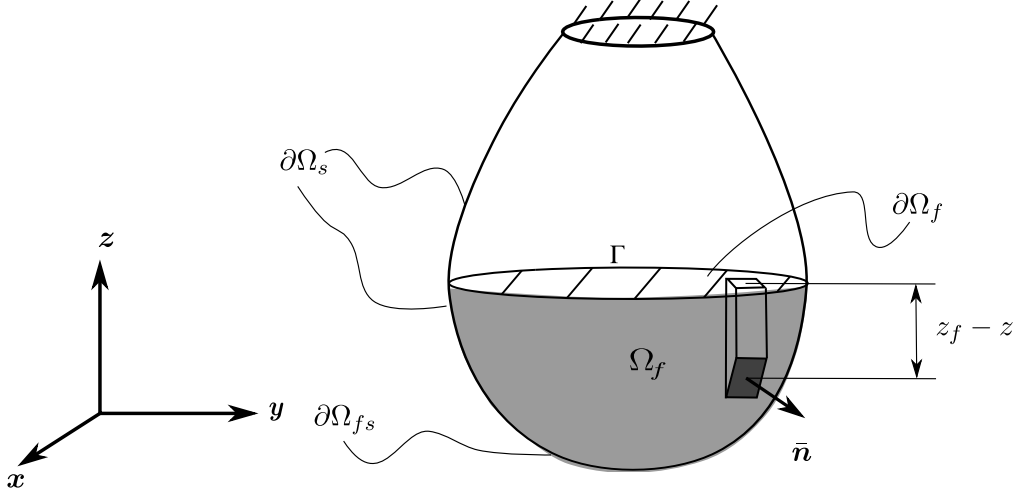


Figure 1: Ponding on a membrane structure.

The equilibrium equations of the structure in the current configuration can be stated using the principle of virtual work

$$\underbrace{\int_{\partial\Omega_s} t \boldsymbol{\sigma} : \delta\boldsymbol{\varepsilon} dS}_{\delta W_{int}} - \underbrace{\int_{\partial\Omega_s} \mathbf{t} \cdot \delta\mathbf{u} dS}_{\delta W_{ext}} = 0, \quad \forall \delta\mathbf{u} \in C_u \quad (1)$$

where  $\boldsymbol{\sigma}$  is the Cauchy stress tensor and  $\boldsymbol{\varepsilon} = \frac{1}{2} (\nabla_{\mathbf{x}} \delta\mathbf{u} + \nabla_{\mathbf{x}}^T \delta\mathbf{u})$  is the virtual Eulerian strain tensor, with  $\nabla_{\mathbf{x}} \bullet = \frac{\partial \bullet}{\partial \mathbf{x}}$ ,  $\delta\mathbf{u}$  as virtual displacement field and  $C_u$  is the kinematically admissible space of smooth enough functions. The thickness of the membrane is denoted by  $t$ , which need not be constant. The equation has two terms: the internal virtual work ( $\delta W_{int}$ ), and the external virtual work ( $\delta W_{ext}$ ) from traction  $\mathbf{t}$ . In order to solve Eq. 1, we express the stresses in terms of strains and subsequently the displacements as unknown. The relation between the stresses and strains is modeled by the constitutive relation. Different constitutive relations are used based the application. In this paper we use the simplest linear material law: plane-stress Saint-Venant Kirchhoff [10]. The relation between the stress and strain tensor can be expressed in Voigt notation as,

$$\begin{bmatrix} \boldsymbol{\sigma}_{11} \\ \boldsymbol{\sigma}_{22} \\ \boldsymbol{\sigma}_{12} \end{bmatrix} = \frac{E}{1-\nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix} \begin{bmatrix} \boldsymbol{\varepsilon}_{11} \\ \boldsymbol{\varepsilon}_{22} \\ 2\boldsymbol{\varepsilon}_{12} \end{bmatrix}, \quad (2)$$

where  $E$  and  $\nu$  are the Young's Modulus and Poisson's ratio, respectively.

The traction  $\mathbf{t}$  in the expression of  $\delta W_{ext}$  is obtained from the hydrostatic loading of the ponding fluid. If we denote the membrane surface below the free surface as  $\partial\Omega_{fs}$ , then we can write,

$$\mathbf{t} = -\gamma_f(z - z_f)\bar{\mathbf{n}} \quad \forall \mathbf{x} \in \partial\Omega_{fs}, \quad (3)$$

$$\mathbf{t} = \mathbf{0} \quad \forall \mathbf{x} \in \partial\Omega_s \setminus \partial\Omega_{fs}, \quad (4)$$

Additionally, we have the volume constraint,

$$V_f = V_t, \quad (5)$$

that needs to be satisfied, where the fluid volume  $V_f$  can be expressed in terms of a surface integral of infinitesimal vertical volume elements  $dV = (z - z_f)\mathbf{e}_z \cdot \bar{\mathbf{n}} dS$  as,

$$V_f = \int_{\partial\Omega_{fs}} (z - z_f) \mathbf{e}_z \cdot \bar{\mathbf{n}} dS, \quad (6)$$

where  $z$  and  $z_f$  are the z-coordinate of a point on  $\partial\Omega_{fs}$  and z-coordinate of the horizontal free surface  $\partial\Omega_f$ , respectively and  $\bar{\mathbf{n}}$  is the outward-pointing unit normal vector of a point on  $\partial\Omega_{fs}$ .

### 3 SOLVER COMPONENTS

#### 3.1 Structural solver

As explained in the previous section, in order to obtain equilibrium shapes of the membrane structure under the hydrostatic load from the ponding fluid we express the virtual work equation, Eq. (1) with only displacements as unknown. For large displacement cases, the internal virtual work  $\delta W_{int}(\mathbf{u})$  is a non-linear function of  $\mathbf{u}$ . Moreover, if the traction depends on the deformed shape of the structure, the external virtual work also depends non-linearly on the displacement field. Few examples of such load cases can be found in [3, 11]. The hydrostatic load from the ponding fluid also falls in this category as the pressure and nodal force vector depends on the membrane deformation below the free surface [3]. The resulting non-linear equation in the continuous displacement field cannot be generally solved analytically. Therefore, techniques such as finite element method are used to transform the non-linear virtual work equation into a set of a finite number of non-linear equations which depend on a discrete nodal displacement vector  $\hat{\mathbf{u}}$ , which can be written as,

$$\hat{\mathbf{f}}_{ext}(\hat{\mathbf{u}}) - \hat{\mathbf{f}}_{int}(\hat{\mathbf{u}}) = 0, \quad (7)$$

where  $\hat{\mathbf{f}}_{int}$  is the internal nodal forces and  $\hat{\mathbf{f}}_{ext}$  is the external nodal forces with the accent  $\hat{\bullet}$  representing the associated nodal quantities. Eq. (7) is typically solved using the N-R method, where the linearized equation at iteration  $n + 1$  can be written as,

$$\begin{aligned} \left( \frac{\partial \hat{\mathbf{f}}_{int}}{\partial \hat{\mathbf{u}}} - \frac{\partial \hat{\mathbf{f}}_{ext}}{\partial \hat{\mathbf{u}}} \right) \Delta \hat{\mathbf{u}}^{n+1} &= \hat{\mathbf{f}}_{ext}(\hat{\mathbf{u}}^n) - \hat{\mathbf{f}}_{int}(\hat{\mathbf{u}}^n) \\ (\mathbf{K}_{mem}(\hat{\mathbf{u}}^n) - \mathbf{K}_l(\hat{\mathbf{u}}^n)) \Delta \hat{\mathbf{u}}^{n+1} &= \hat{\mathbf{f}}_{ext}(\hat{\mathbf{u}}^n) - \hat{\mathbf{f}}_{int}(\hat{\mathbf{u}}^n) \end{aligned} \quad (8)$$

The matrix  $\mathbf{K}_{mem}$  is the familiar global tangent stiffness matrix from membrane elements; interested readers can refer to [12] for more details. The second matrix in Eq. (8) is called the load stiffness matrix,

which depends on the type of follower load [3, 11]. In this paper, following the work of [3], we can split the load stiffness matrix into three parts,

$$\mathbf{K}_l = \mathbf{K}_l^{\Delta n} + \mathbf{K}_l^{\Delta z} + \mathbf{K}_l^{\Delta z_f}, \quad (9)$$

where  $\mathbf{K}_l^{\Delta n}$  represents the change in nodal force from the change in surface normal due to the membrane deformation,  $\mathbf{K}_l^{\Delta z}$  captures the effect of change in hydrostatic pressure from membrane displacement alone and the effect of change in hydrostatic pressure due to free surface movement owing to the incompressible behaviour of the fluid is taken into account by  $\mathbf{K}_l^{\Delta z_f}$ . The notation used here is different from that in [3] for clearly showing the role of each component. Nevertheless, the expression of each component can be easily inferred from the paper. Depending on the method, if it is monolithic or partitioned, we use either full or partial load stiffness matrices, as explained in Sections 4 and 5.

The procedure for building and solving the system of equations in Eq. (8) is implemented in an open-source finite element framework called KRATOS [13]. The required load stiffness matrices are added or removed based on the type of the method being used for the calculation.

### 3.2 Volume-conserving solver

The incompressible behaviour of the ponding fluid combined with the requirement of a horizontal flat free surface under static conditions means that when the membrane surface below the free surface deforms, the free surface has to move vertically to conserve the fluid volume. This behavior of the fluid is simulated by a solver called volume-conserving solver. The solver takes the target fluid volume as an input and moves the free surface vertically to achieve the target fluid volume. It consists of two parts: volume calculation and an iterative algorithm to update the free surface. The equation to calculate the volume enclosed by  $\partial\Omega_f \cup \partial\Omega_{fs}$  is already given in Eq. (6) and its derivative with respect to  $z_f$  can be written as,

$$\frac{\partial V_f}{\partial z_f} = \int_{\partial\Omega_{fs}} -\mathbf{e}_z \cdot \bar{\mathbf{n}} \, dS = A_f, \quad (10)$$

In the current work, the leap-frogging Newton's method is used for conserving a given volume. This method is discussed in detail in [14]. The equations used for the iterations to conserve the volume are given in Eqs. (11) and (12), with the function  $f(z_f^m)$  being the volume residual ( $V_f^m - V_t$ ), and  $f'(z_f^m)$  its derivative with respect to  $z_f$ , where the superscript  $\bullet^m$  denotes the iteration number and the accent  $\checkmark$  is used to specify quantities at the intermediate position.

$$z_f^m = z_f^m - \frac{f(z_f^m)}{f'(z_f^m)} \quad (11)$$

$$z_f^{m+1} = z_f^m - \frac{f(z_f^m)^2}{f'(z_f^m) \left( f(z_f^m) - f(z_f^m) \right)} \quad (12)$$

## 4 MONOLITHIC METHODS

### 4.1 Monolithic method with volume conservation inside structural solver (MVCIS)

The first method which is classified as a monolithic method involves volume conservation inside the structural N-R iterations. In this method we solve the structural equations by incorporating the full linearized equations of the fluid loading inside the N-R iterations of the structural solver, which means the load stiffness matrices  $\mathbf{K}_l^{\Delta n}$ ,  $\mathbf{K}_l^{\Delta z}$ ,  $\mathbf{K}_l^{\Delta z_f}$  are included in the N-R iterations. Along with this, the volume conserving-algorithm discussed in Section 3.2 is used to update the free surface after every N-R iteration. The complete algorithm is written in Algorithm. 1

---

#### Algorithm 1 Monolithic method with volume conservation inside structural iterations (MVCIS)

---

- 1:  $n = 0$
  - 2: Find  $z_f^0$  using volume-conserving solver with  $V_t$  as input argument
  - 3: **while**  $\|\hat{\mathbf{f}}_{ext} - \hat{\mathbf{f}}_{int}\| > \varepsilon$  and  $n < n_{max}$  **do**
  - 4:     Update:  $\mathbf{K}_{mem}$ ,  $\mathbf{K}_l^{\Delta n}$ ,  $\mathbf{K}_l^{\Delta z}$ ,  $\mathbf{K}_l^{\Delta z_f}$ ,  $\hat{\mathbf{f}}_{ext}$  and  $\hat{\mathbf{f}}_{int}$  using  $\hat{\mathbf{u}}^n$  and  $z_f^n$
  - 5:     Solve:  $(\mathbf{K}_{mem} - \mathbf{K}_l^{\Delta n} - \mathbf{K}_l^{\Delta z} - \mathbf{K}_l^{\Delta z_f}) \Delta \hat{\mathbf{u}}^{n+1} = \hat{\mathbf{f}}_{ext} - \hat{\mathbf{f}}_{int}$
  - 6:     Update displacement:  $\hat{\mathbf{u}}^{n+1} = \hat{\mathbf{u}}^n + \Delta \hat{\mathbf{u}}^{n+1}$
  - 7:     Update structure:  $\hat{\mathbf{x}}^{n+1} = \hat{\mathbf{X}} + \hat{\mathbf{u}}^{n+1}$
  - 8:     Update free surface using volume-conserving solver with  $V_t$  as input argument
  - 9:      $n = n + 1$
  - 10: **end while**
- 

### 4.2 Monolithic method with volume conservation as constraint (MVCC)

In the monolithic method discussed in Section 4.1 (MVCIS), we update the free surface after every N-R iteration to achieve the target volume. This will mostly work if the initial shape of the structure is close to the solution but if the initial shape is far away from the solution then the N-R iterations may diverge. It was noticed during the numerical experiments that during the initial N-R iterations sometimes the structure moved above the free surface and the volume conserving solver failed as the fluid volume and free surface area required for the algorithm becomes zero. The reason is believed to be the enforcement of the volume constraint exactly in every N-R iteration and the lack of linearized behavior of the other solver in both solvers, which will be explained later in this section.

The main idea behind this monolithic method is that there is no need to satisfy the volume constraint in every N-R iteration. Instead, the volume constraint is only required to be satisfied when the convergence is achieved. The formulation of this method is as follows: we seek a solution to the discrete equilibrium equation given in Eq. (7) with the volume constraint  $g(\hat{\mathbf{u}}, z_f)$ ,

$$\hat{\mathbf{f}}_{int}(\hat{\mathbf{u}}) - \hat{\mathbf{f}}_{ext}(\hat{\mathbf{u}}, z_f) = 0, \quad (13)$$

$$g(\hat{\mathbf{u}}, z_f) = V_f(\hat{\mathbf{u}}, z_f) - V_t = 0. \quad (14)$$

Here, we consider  $z_f$  as an additional independent variable, which allows the volume conservation constraint to be incorporated in the system of equations. The system of equations given in Eqs. (13) and (14) can be solved using the N-R algorithm, where the linearized form at iteration  $n + 1$  can be written as:

$$\left( \frac{\partial \hat{\mathbf{f}}_{int}(\hat{\mathbf{u}}^n)}{\partial \hat{\mathbf{u}}} - \frac{\partial \hat{\mathbf{f}}_{ext}(\hat{\mathbf{u}}^n, z_f^n)}{\partial \hat{\mathbf{u}}} \right) \Delta \hat{\mathbf{u}}^{n+1} - \frac{\partial \hat{\mathbf{f}}_{ext}(\hat{\mathbf{u}}^n, z_f^n)}{\partial z_f} \Delta z_f^{n+1} = \hat{\mathbf{f}}_{ext}(\hat{\mathbf{u}}^n, z_f^n) - \hat{\mathbf{f}}_{int}(\hat{\mathbf{u}}^n) \quad (15)$$

$$\frac{\partial g(\hat{\mathbf{u}}^n, z_f^n)}{\partial \hat{\mathbf{u}}} \Delta \hat{\mathbf{u}}^{n+1} + \frac{\partial g(\hat{\mathbf{u}}^n, z_f^n)}{\partial z_f} \Delta z_f^{n+1} = V_t - V_f(\hat{\mathbf{u}}^n, z_f^n). \quad (16)$$

with,

$$\begin{aligned} \frac{\partial \hat{\mathbf{f}}_{int}}{\partial \hat{\mathbf{u}}} &= \mathbf{K}_{mem}; & \frac{\partial \hat{\mathbf{f}}_{ext}}{\partial \hat{\mathbf{u}}} &= \mathbf{K}_l^{\Delta n} + \mathbf{K}_l^{\Delta z}; & \frac{\partial \hat{\mathbf{f}}_{ext}}{\partial z_f} &= \gamma_f \int_{\eta} \int_{\xi} \mathbf{N}^T \mathbf{n} d\xi d\eta \\ \frac{\partial g}{\partial \hat{\mathbf{u}}} &= \left( \int_{\eta} \int_{\xi} \mathbf{N}^T \mathbf{n} d\xi d\eta \right)^T; & \frac{\partial g}{\partial z_f} &= A_f, \end{aligned}$$

where  $\mathbf{N}$  is the shape function matrix used in finite elements. Substituting all the terms and eliminating  $\Delta z_f^{n+1}$  from Eqs. (15) and (16) we get,

$$\left( \mathbf{K}_{mem} - \mathbf{K}_l^{\Delta n} - \mathbf{K}_l^{\Delta z} - \mathbf{K}_l^{\Delta z_f} \right) \Delta \hat{\mathbf{u}}^{n+1} = \hat{\mathbf{f}}_{ext} - \hat{\mathbf{f}}_{int} + \underbrace{\frac{\gamma_f (V_t - V_f^n)}{A_f} \int_{\eta} \int_{\xi} \mathbf{N}^T \mathbf{n}^h d\xi d\eta}_{\mathbf{f}_{\Delta V}} \quad (17)$$

$$\Delta z_f^{n+1} = \underbrace{\frac{(V_t - V_f^n)}{A_f}}_{\Delta z_{f, \Delta V}^{n+1}} - \underbrace{\frac{\left( \int_{\eta} \int_{\xi} \mathbf{N}^T \mathbf{n}^h d\xi d\eta \right)^T \Delta \hat{\mathbf{u}}^{n+1}}{A_f}}_{\Delta z_{f, \Delta u}^{n+1}}. \quad (18)$$

where  $\mathbf{K}_l^{\Delta z_f} = \gamma_f \left( \int_{\eta} \int_{\xi} \mathbf{N}^T \mathbf{n} d\xi d\eta \right) \left( \int_{\eta} \int_{\xi} \mathbf{N}^T \mathbf{n} d\xi d\eta \right)^T$ . The method is finally written in Algorithm 2. Note that the main difference between this monolithic method and MVCIS is that instead of a nested volume conservation, there is an explicit equation for the free surface update which can be split into two parts. The first part  $\Delta z_{f, \Delta V}$  compensates for the difference between the current fluid and the target fluid volume, while the second part,  $\Delta z_{f, \Delta u}$  moves the free surface to mitigate the change in volume due to the deformation of the wetted surface. The volume residual also appears as an extra pressure,  $p_{\Delta V} = \gamma_f (V_t - V_f) / A_f$  in the structural equations, which should converge to zero when the solver converges. If the volume difference is positive then it applies a positive extra pressure on the structure resulting in increase in fluid volume and vice versa. Clearly, the linearized behavior of the free surface update is incorporated in the structural solver and the linearized behavior of the structural solver in the free surface update, making it more robust. By contrast, in MVCIS this is only the case with the structural solver.

---

**Algorithm 2** Monolithic method with volume conservation as a constraint (MVCC)
 

---

```

1:  $n = 0$ 
2: Find  $z_f^0$  using volume-conserving solver with  $V_t$  as input argument
3: while  $\|\hat{\mathbf{f}}_{ext} - \hat{\mathbf{f}}_{int} + \hat{\mathbf{f}}_{\Delta V}\| > \varepsilon$  and  $n < n_{max}$  do
4:   Update:  $\mathbf{K}_{mem}, \mathbf{K}_l^{\Delta n}, \mathbf{K}_l^{\Delta z}, \mathbf{K}_l^{\Delta z_f}, \hat{\mathbf{f}}_{ext}, \hat{\mathbf{f}}_{\Delta V}$  and  $\hat{\mathbf{f}}_{int}$  using  $\hat{\mathbf{u}}^n$  and  $z_f^n$ 
5:   Solve:  $(\mathbf{K}_{mem} - \mathbf{K}_l^{\Delta n} - \mathbf{K}_l^{\Delta z} - \mathbf{K}_l^{\Delta z_f}) \Delta \hat{\mathbf{u}}^{n+1} = \hat{\mathbf{f}}_{ext} - \hat{\mathbf{f}}_{int} + \hat{\mathbf{f}}_{\Delta V}$ 
6:   Update displacement:  $\hat{\mathbf{u}}^{n+1} = \hat{\mathbf{u}}^n + \Delta \hat{\mathbf{u}}^{n+1}$ 
7:   Update structure:  $\hat{\mathbf{x}}^{n+1} = \hat{\mathbf{X}} + \hat{\mathbf{u}}^{n+1}$ 
8:   Calculate  $\Delta z_f^{n+1}$  using Eq. (18)
9:    $z_f^{n+1} = z_f^n + \Delta z_f^{n+1}$ 
10:   $n = n + 1$ 
11: end while
    
```

---

## 5 PARTITIONED METHOD

In the partitioned approach the problem of finding the static deformation under ponding load is formulated as a fixed-point problem with structural displacement field as the variable. The structural solver uses the vertical distance field  $\mathbf{d}$  ( $= z - z_f$ ) from the free surface to calculate the hydrostatic pressure and solves for the displacement  $\mathbf{u}$ . Based on the updated structural deformation the volume-conserving solver updates the free surface position. This process is continued till the norm of the fixed point residual, discussed below, is lower than a certain pre-defined tolerance. The fixed point iterations discussed above can be used with the convergence accelerators such as Aitken relaxation [7, 8] and IQN-ILS [9] to achieve faster convergence. Mathematically, if we denote the volume-conserving solver as an operator  $\mathcal{F}$  and the structural solver for the membrane surface as  $\mathcal{S}$ , then we can write,

$$\begin{aligned} \mathbf{d} &= \mathcal{F}(\mathbf{u}) \\ \mathbf{u} &= \mathcal{S}(\mathbf{d}) \end{aligned}$$

and the resulting fixed point problem can be written as,

$$\mathbf{u} = \mathcal{S} \circ \mathcal{F}(\mathbf{u}). \quad (19)$$

The residual of Eq. (19) (fixed point residual) at the  $k^{th}$  iteration is written as  $\mathbf{r}^k = \tilde{\mathbf{u}}^{k+1} - \mathbf{u}^k$  where  $\mathbf{u}^k$  is the displacement at the  $k^{th}$  iteration and  $\tilde{\mathbf{u}}^{k+1} = \mathcal{S} \circ \mathcal{F}(\mathbf{u}^k)$ .

The implicit partitioned coupling method for the problem is given in Algorithm 3. Note that in each coupling iteration, the structural solver uses  $\mathbf{K}_l^{\Delta n}$  in the non-linear iterations as the input is a pressure field (calculated from  $\mathbf{d}$ ) on the wetted surface. Compared to the monolithic methods discussed before, the main advantage of this method is its modularity. Moreover, it is also more robust since we are doing volume conservation on the equilibrium shapes obtained with stationary pressure loading. Therefore, it is less likely that there will be a drastic difference between the structural displacements of consecutive coupling iterations that might otherwise result in the wetted surface to move above the free surface.



---

**Algorithm 3** Implicit partitioned coupling (IPC) method

---

```

1:  $k = 0$ 
2:  $\tilde{\mathbf{u}}^1 = \mathcal{S} \circ \mathcal{F}(\mathbf{u}^0)$ 
3:  $\mathbf{r}^0 = \tilde{\mathbf{u}}^1 - \mathbf{u}^0$ 
4: while  $\|\mathbf{r}^k\| > \varepsilon$  and  $k < k_{max}$  do
5:   Calculate  $\mathbf{u}^{k+1}$  using convergence accelerator [9, 7, 8]
6:    $k = k + 1$ 
7:    $\tilde{\mathbf{u}}^{k+1} = \mathcal{S} \circ \mathcal{F}(\mathbf{u}^k)$ 
8:    $\mathbf{r}^k = \tilde{\mathbf{u}}^{k+1} - \mathbf{u}^k$ 
9: end while

```

---

## 6 NUMERICAL EXAMPLE

The discussed methods for ponding analysis were developed in an open-source finite element code called KRATOS. In this section we present a numerical example: Ponding on a square membrane. The numerical example is used to compare the computing time and robustness of the algorithms. It is to be noted that the volume calculation algorithm and the volume conserving algorithm was checked by applying them on a hollow hemisphere where the analytically formula for the volume is available. Due to the space constraint this is not discussed here.

### 6.1 Ponding on a square membrane

We perform ponding analysis on a flat square membrane of side  $a = 10\text{ m}$  clamped along the edges using the monolithic and partitioned methods. The material is modeled as Saint-Venant Kirchhoff plane stress material with material properties: Young's modulus  $E = 10^8\text{ N/m}^2$ , Poisson ratio  $\nu = 0.3$  and thickness  $t = 0.001\text{ m}$ . The membrane is located in the X-Y plane with the centre O at the origin as shown in Fig. 2a. The geometry is discretized with 7748 linear triangle elements shown in Fig. 2b. As seeding event we apply an initial deformation,  $\mathbf{u}_{init} = [0, 0, -\cos(x\pi/a)\cos(y\pi/a)]$ . The ponding fluid, water ( $\gamma_f = 10^4\text{ N/m}^3$ ) in this case is added gradually to simulate ponding. Membrane deformations due to three different volumes of water are shown in Fig. 3. In the figure, the purpose of choosing the color bar from negative to positive value very close to zero is to distinguish the wetted surface or region from the remaining membrane surface.

The algorithms are first compared in terms of computing time, shown in Fig. 4. In these simulations, a volume of increment of  $20\text{ m}^3$  is added in every load step to simulate ponding. From Fig. 4, we can clearly observe that the monolithic methods are approximately 3 times faster than the partitioned method. Moreover, the two monolithic methods discussed in the paper have similar performance, if we look at the computing time. However, when the two algorithms were compared in terms of robustness MVCC seems to be more robust. This was observed by checking the convergence behavior of the algorithms with increasing volume increments. The MVCC could handle the volume increment of  $200\text{ m}^3$  but MVCIS failed with the volume increment  $150\text{ m}^3$ . The algorithm MVCIS failed when the wetted surface moved above the free surface position set at the previous non-linear iterations. This is not observed in MVCC because both solvers have the knowledge of the other solver's behaviour built into it and also because the volume conservation was not strictly imposed on non-equilibrium shapes of the structure, as discussed in Section 4.2. It was also found that the IPC algorithm had the same robustness characteristics when

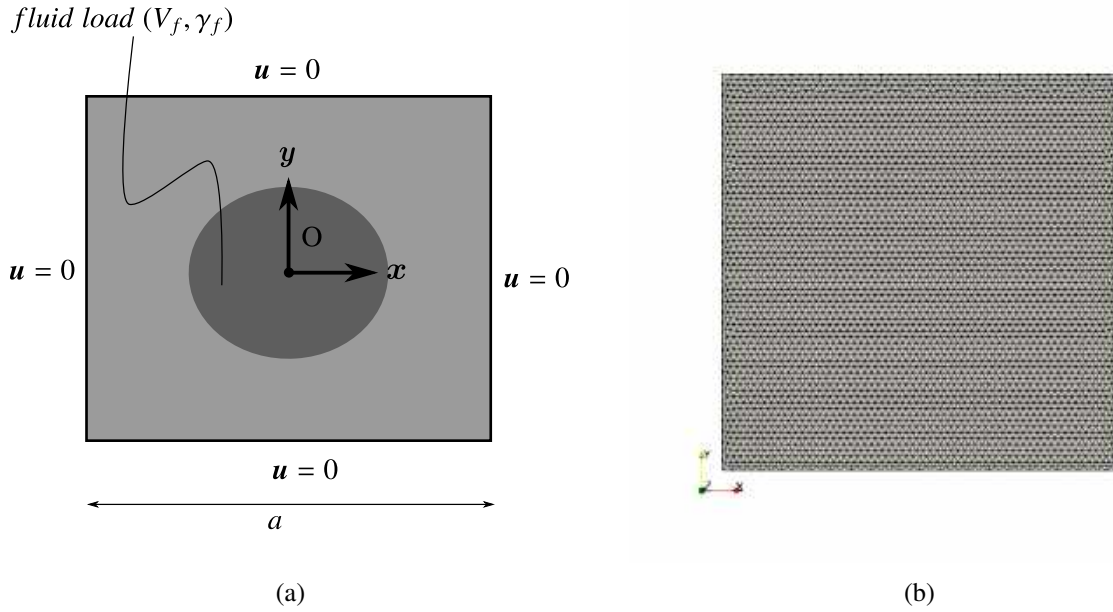


Figure 2: Ponding on a square membrane: a) problem set up, b) meshed geometry (viewed from top).

used with a quasi-Newton convergence accelerator and was able to find the equilibrium shapes up to  $200 \text{ m}^3$  volume increment. This is because the volume conservation is applied on the equilibrium shapes obtained with stationary pressure loading, as explained in Section 5.

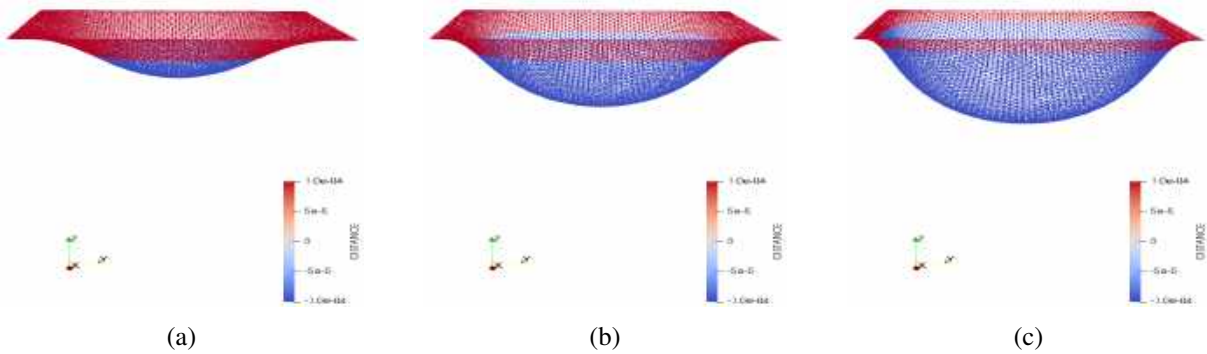


Figure 3: Deformation of square membrane due to hydrostatic loading from different volumes of water, where the blue color on the surface indicates negative vertical distance from the free surface, and therefore represents the wetted region: a)  $V_f = 20 \text{ m}^3$ , b)  $V_f = 100 \text{ m}^3$ , c)  $V_f = 200 \text{ m}^3$ .

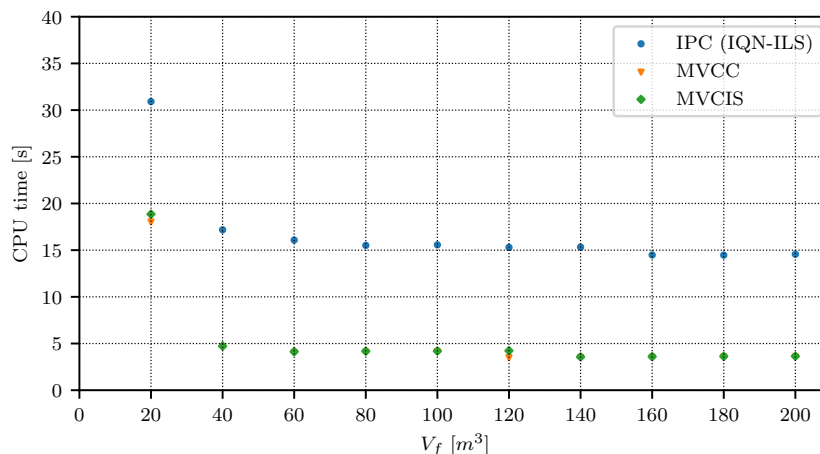


Figure 4: Comparison of CPU time for different fluid volumes ( $V_f$ ) for IPC, MVCIS, MVCC.

## 7 CONCLUSIONS

In this paper, the performance of two monolithic methods and a partitioned method was compared. It was found that the monolithic methods were superior to the partitioned method, if we just look at the computing time. However, when it comes to robustness the partitioned method was found to be more robust than MVCIS. The problem with the MVCIS was that the volume conservation was applied on non-equilibrium shapes of the structure and the linearized behavior of the other solver was not included in both solvers. These problems were addressed in MVCC algorithm by formulating volume conservation as an additional non-linear equation and its subsequent linearization in the N-R iterations, making it a faster and more robust algorithm for ponding analysis.

## 8 ACKNOWLEDGMENT

The authors gratefully acknowledge the funding of the Research Foundation–Flanders (FWO) for this work (project number G086517N)

## REFERENCES

- [1] W. Szyszkowski and P. Glockner, “Finite deformation and stability behaviour of spherical inflatables subjected to axi-symmetric hydrostatic loading,” *International Journal of Solids and Structures*, vol. 20, no. 11-12, pp. 1021–1036, 1984.
- [2] C. Y. Tuan, “Ponding on circular membranes,” *International Journal of Solids and Structures*, vol. 35, no. 3-4, pp. 269–283, 1998.
- [3] T. Rumpel and K. Schweizerhof, “Hydrostatic fluid loading in non-linear finite element analysis,” *International Journal for Numerical Methods in Engineering*, vol. 59, no. 6, pp. 849–870, 2004.
- [4] C. Hoareau and J.-F. Deü, “Nonlinear equilibrium of partially liquid-filled tanks: A finite element/level-set method to handle hydrostatic follower forces,” *International Journal of Non-Linear Mechanics*, vol. 113, pp. 112–127, jul 2019.

- [5] C. Hoareau and J.-F. Deü, “Non-linear finite element analysis of an elastic structure loaded by hydrostatic follower forces,” *Procedia Engineering*, vol. 199, pp. 1302–1307, 2017.
- [6] A. Bown, T. Makin, and D. Wakefield, “Beyond static analysis: Investigation of membrane structure performance using time stepping, transient and progressive analyses,” *Procedia Engineering*, vol. 155, pp. 313–322, 2016.
- [7] U. Küttler and W. A. Wall, “Fixed-point fluid–structure interaction solvers with dynamic relaxation,” *Computational Mechanics*, vol. 43, no. 1, pp. 61–72, 2008.
- [8] B. M. Irons and R. C. Tuck, “A version of the aitken accelerator for computer iteration,” *International Journal for Numerical Methods in Engineering*, vol. 1, pp. 275–277, jul 1969.
- [9] J. Degroote, “Partitioned simulation of fluid-structure interaction,” *Archives of Computational Methods in Engineering*, vol. 20, no. 3, pp. 185–238, 2013.
- [10] Holzapfel, *Nonlinear Solid Mechanics*. John Wiley & Sons, 2000.
- [11] H. D. Hibbit, “Some follower forces and load stiffness,” *International Journal for Numerical Methods in Engineering*, vol. 14, no. 6, pp. 937–941, 1979.
- [12] R. L. Taylor, E. Oñate, and P.-A. Ubach, “Finite element analysis of membrane structures,” *Computational Methods in Applied Sciences*, pp. 47–68, Springer-Verlag, 2005.
- [13] P. Dadvand, R. Rossi, and E. Oñate, “An object-oriented environment for developing finite element codes for multi-disciplinary applications,” *Archives of Computational Methods in Engineering*, vol. 17, no. 3, pp. 253–297, 2010.
- [14] A. B. Kasturiarachi, “Leap-frogging newton’s method,” *International Journal of Mathematical Education in Science and Technology*, vol. 33, pp. 521–527, 2002.