Monolithic and partitioned approaches to determine static deformation of membrane structures due to ponding

N.K. Narayanan[a,b,*], R. Wüchner[a], J. Degroote[b,c]

[a] *Chair of Structural Analysis, Technical University of Munich*
[b] *Department of Electromechanical, Systems and Metal Engineering, Ghent University*
Sint-Pietersnieuwstraat 41, 9000 Ghent, Belgium
[c] *Flanders Make, Belgium*

Abstract

This paper proposes monolithic and partitioned methods to calculate the static deformation of a membrane structure due to a given volume of ponding water. The partitioned methods involve coupling of a structural solver for membranes and a volume-conserving solver, modeling static incompressible fluid. Two methods of this type are proposed, either using coupling iterations with convergence accelerator between structural solver and volume-conserving solver or adding the linearized fluid behavior in the structural solver in addition to the external coupling iterations. The monolithic methods solve the system of structural equations under hydrostatic load with the volume conservation behavior of the fluid included in the Newton-Raphson (N-R) iterations of the structural solver. One such method was already discussed in the literature and updates the free surface plane to conserve volume exactly after every N-R iteration. In the second, new monolithic method, the vol-

*Corresponding author
Email address: navaneeth.narayanan@ugent.be (N.K. Narayanan)

ume conservation constraint is added as an additional equation and solved together with the structural equations. It was found that the partitioned method used with a quasi-Newton convergence accelerator was very robust but slower than the monolithic methods. On the other hand, the new monolithic method proposed in this paper was found to be both computationally efficient and robust.
Keywords: Ponding, Partitioned method, Monolithic method, Hydrostatic load, Volume-conserving solver

1. Introduction

Membrane structures have a unique characteristic of carrying loads by undergoing significant deflection. This makes them efficient in terms of material usage compared to the load capacity but also makes them vulnerable to ponding. Most light weight structures are designed with sufficient gradient to avoid this scenario. However, there are cases where a seeding event such as snowfall can create a local depression to trigger ponding. Following the seeding event, based on the initial prestress, the type of cable supports and the elastic property of the membrane material, rain can lead to a stable or unstable water pond. The latter scenario will be fatal for the structure as this will result in indefinite increase of accumulating water till the structural collapse or failure. Therefore, it is important to evaluate membrane structures for stability under ponding. This requires a fluid-structure (FSI) simulation between the membrane and the ponding water.

Even in the stable pond scenario, if the rain is accompanied by strong winds, the wind flow around the structure may induce large oscillations. In

2011, during the Pukkelpop festival held in Kiewit (Belgium) [1], a strong wind interacting with ponding water led to huge swaying of the large festival tents, eventually resulting in the collapse of these structures. Studying such cases will involve fluid-structure interaction simulation between the membrane structure, the water and the wind flow. Imposing an initial condition with a pond on a membrane requires computation of the static deformation of the membrane structure under the load of a given volume of ponding water, which is the main motivation of the current work.

Some of the other applications of this analysis include floating caps of oil storage tanks [2], and optical reflector forming using ponding loads [3]. Compared to the many other aspects of analysis on membrane structures such as large deformation analysis, form finding [4], wrinkling [5] and membrane wind interaction [6], the analysis involving ponding water on a membrane structure is relatively rare. What makes this type of analysis in membrane structures challenging is the that the shape of the ponding fluid on the structure is unknown. Therefore, the region of fluid loading is not known before; in most cases the structure will be initially flat before the ponding analysis and so cannot contain any fluid. The deformation of the structure is dependent on structural stiffness and pressure exerted by the fluid on the structure, which is in turn a function of structural deformation. Therefore, the problem of finding the deformed shape of a structure under the hydrostatic load exerted by a given volume of ponding fluid is very non-linear.

In the literature, stability behavior under ponding has been extensively discussed by Szyszkowski and Glockner [7] where they studied ponding stability and deformation on spherical inflatables by solving axi-symmetric mem-

brane equations with the hydrostatic loads. Tuan [3] in his work focused on large deformations and strains of initially flat, simply supported circular membranes under gradually accumulated fluid. He used fourth-order Runge-Kutta numerical integration with an iterative finite element analysis using shell elements to calculate the deformation due to ponding. However, these studies only involved axi-symmetric geometries. A more general approach to calculate deformation due to hydrostatic follower forces on structures in the finite element framework is discussed in [8] where they linearize the static behaviour of incompressible fluid under gravity to obtain the symmetric load stiffness matrix used in the Newton-Raphson (N-R) iterations. The symmetry of the load stiffness matrices is also discussed in [9, 10] with the name elasto-gravity operator. Similar work can be also found in a more recent paper by Hoareau and Deü [11, 12], where a level set approach is used for 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 [8]. Since their primary interest was to study deformation of tanks under hydrostatic loads, a good initial geometry was available that can contain fluid, and thus relatively simpler than the ponding analysis on large membrane tents. An example closely related to ponding on membrane structures can be found in [13] where they studied stability of a hydrostatic load on a flat circular membrane. They used a generalized path-following scheme [14] with free surface height as a controlling parameter to plot the equilibrium path of the structure. In their

analysis, they found several limit points on the equilibrium path when the free surface height was used as a controlling parameter and suggested to use the volume of the fluid instead.

All the studies discussed in the previous paragraph fall under the category of monolithic methods to compute structural deformation under hydrostatic loads. The current paper discusses two monolithic methods to calculate static deformation due to a fixed volume of ponding fluid. The first method which imposes conservation of volume after every structural N-R iteration, similar to one discussed in [11] but a faster and robust iteration scheme is used for volume conservation. Therefore, in this paper it is called monolithic method with volume conservation inside structural iterations (MVCIS). This is because unlike a flexible water tank which has some stiffness due to the geometry, the ponding analysis involving a relatively flat and flexible membrane structure will undergo large deformation during initial N-R iterations of the structural solver, thus requiring an efficient and robust algorithm for volume conservation. The main problem with this monolithic method is that it enforces the volume conservation constraint exactly in non-equilibrium shapes found during structural N-R iterations, which is unnecessary and in some cases it led to divergence. The second method, which is a novelty, solves the structural equations under hydrostatic loads with the constraint that the fluid volume should be equal to the target volume. The structural equations with the constraint are solved using N-R iterations by linearizing the system of equations with the constraint. This way the structural equilibrium equations and volume constraint are satisfied only at the end of N-R iterations. The proposed method therefore is called monolithic method with

volume conservation as constraint (MVCC). This method was found to be more robust than MVCIS, which will be shown later in Section 8.2.

Clearly, implementation of the above methods is only possible if the structural solver is accessible, but in some cases where the solver is proprietary software, ponding analysis can be only performed with partitioned methods. Therefore, in this paper we also a present partitioned methods to perform ponding analysis. One example of this approach is presented in the work of Bown et al. [15], 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 methods for ponding analysis presented in this paper use a volume-conserving solver instead of a transient shallow water solver as used by Bown *et al.* The volume-conserving solver models the quasi-static behavior of fluid by updating the free surface, which is a plane perpendicular to gravity, to conserve a given volume of the ponding fluid. In this method the structural solver and the volume-conserving solver are executed sequentially inside a loop with the output of the other solver as its input. The volume-conserving solver takes the displacement field of the structure as input and updates the free surface plane to conserve the volume, while the structural solver uses the updated hydrostatic pressure, which depends on the new vertical height of the free surface plane to calculate a new displacement field, resulting in a fixed point iteration. Convergence accelerators are used to speed up the convergence and stabilize the fixed point iteration [16, 17]. The iterations are continued till the norm of the fixed point residual, defined later in Section 6.1, is below certain tolerance.

Additionally, a second partitioned method is presented in this paper in

which a linearized fluid behavior is added inside the structural solver to accelerate the fixed-point iterations. Technically, this is not a pure partitioned method, since the method involves modifying a structural solver. However, the method is classified as partitioned method because it still involves outer fixed point iterations to solve the problem. While this method loses the advantage of code modularity, it has better convergence characteristics than the pure partitioned method due to the inclusion of linearized behavior of the fluid solver in the structural solver. However, it has one problem at the first coupling iteration when the fluid volume increment is large, which will be discussed in Section 8.2.

The outline of the paper is as following. In Section 2, the mathematical formulation of the ponding problem is presented, which involves non-linear equilibrium equations of the membrane structure and equilibrium equations of the fluid. This is followed by constitutive equations for the isotropic plane-stress linear elastic and hyper-elastic material model. Subsequently, the two solver components used in the analysis are presented in Section 3: the structural solver with membrane elements and the volume-conserving solver, which models the quasi-static behavior of the fluid. The linearization of the fluid loading used in N-R iterations of three of the discussed methods is explained in Section 4. Section 5 and Section 6 discuss the monolithic and partitioned methods for ponding analysis, respectively. The procedure for integration on a discretized surface required by the various methods is presented in Section 7. Finally, in Section 8 the proposed methods are analyzed and compared using numerical examples.

2. Mathematical formulation

Consider a membrane structure, denoted by $\partial\Omega_s$, containing a certain volume V_t of incompressible fluid of specific weight γ_f. The fluid region is denoted by Ω_f, which is enclosed by free surface of the fluid ($\partial\Omega_f$) and wetted surface of the membrane ($\partial\Omega_{fs}$). This system has two components: fluid and membrane structure. To find the static deformation due to ponding, the equilibrium equations of both fluid and membrane have to be solved along with the constraint that the volume of fluid is equal to V_t.

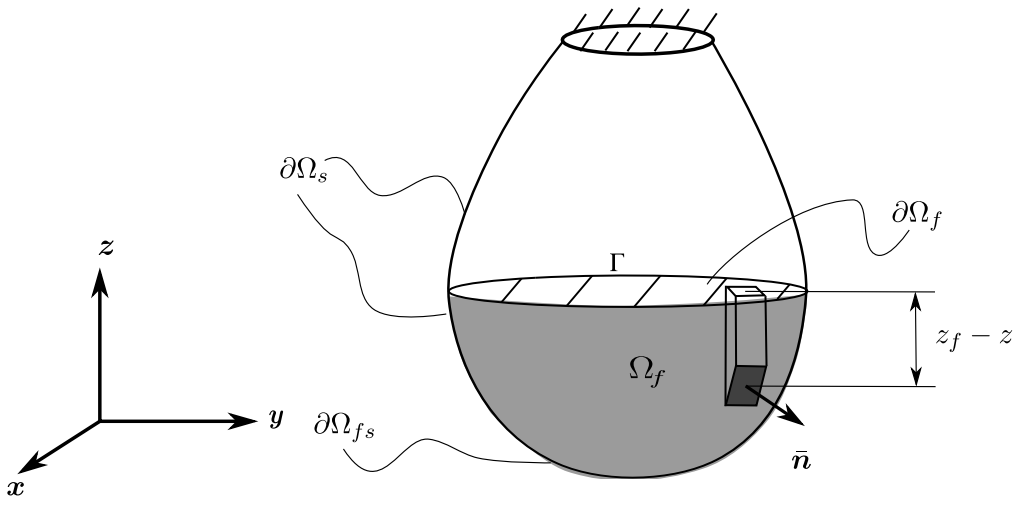


Figure 1: Ponding on a membrane structure.

2.1. Fluid equations

Under static conditions, the free surface of the fluid is always flat and perpendicular to gravity. For the sake of brevity, we assume that gravity is along negative z-direction and therefore $\boldsymbol{e}_z$ is the unit normal at any point on the free surface. Additionally, the pressure p at any point on the free surface

is zero (relative to atmosphere). This boundary condition with the fluid equilibrium equation at static conditions, given in Eq.(1) and the constraint that the volume of fluid region Ω_f should be equal to V_t, forms the system of equations for the fluid at rest,

$$\nabla p = -\gamma_f \boldsymbol{e}_z \qquad \forall \boldsymbol{x} \in \Omega_f \tag{1}$$

$$p = 0 \qquad \forall \boldsymbol{x} \in \partial\Omega_f \tag{2}$$

$$\int_{\Omega_f} dV = V_t. \tag{3}$$

However, the system of equations Eqs. (1), (2) and (3) requires volume discretization of Ω_f. This is avoided by expressing it in terms of surface quantities. To that end, integrating Eq.(1) with Eq.(2) as boundary condition results in the familiar hydrostatic loading on the wetted surface, $p = -\gamma_f(z - z_f)$ with $z = \boldsymbol{x} \cdot \boldsymbol{e}_z$, $\forall \boldsymbol{x} \in \partial\Omega_{fs}$ and z_f as the z-coordinate of the free surface. Furthermore, with the absence of shear stress under hydrostatic condition the traction at the wetted surface can be written in terms of pressure and the unit normal $\bar{\boldsymbol{n}}$ as $\boldsymbol{t} = p\bar{\boldsymbol{n}}$. For the membrane surface which is not in contact with the fluid ($\partial\Omega_s \setminus \partial\Omega_{fs}$), the pressure relative to the atmosphere is zero. Consequently, in the absence of any other external load the traction is equal to the zero vector in $\partial\Omega_s \setminus \partial\Omega_{fs}$. The volume conservation constraint Eq. (3) can also be expressed in terms of a surface integral of infinitesimal vertical volume elements $dV = (z - z_f)\boldsymbol{e}_z \cdot \bar{\boldsymbol{n}}\, dS$, which results in the following set of equations for the fluid in terms of surface quantities,

$$\boldsymbol{t} = -\gamma_f (z - z_f)\bar{\boldsymbol{n}} \qquad \forall \boldsymbol{x} \in \partial\Omega_{fs}, \tag{4}$$

$$\boldsymbol{t} = \boldsymbol{0} \qquad \forall \boldsymbol{x} \in \partial\Omega_s \setminus \partial\Omega_{fs},$$

$$\int_{\partial\Omega_{fs}} (z - z_f)\, \boldsymbol{e}_z \cdot \bar{\boldsymbol{n}}\, dS = V_t, \tag{5}$$

2.2. Structural equations

The membrane structure shown in Fig. 1 is in static equilibrium with the ponding fluid. Therefore, by applying the principle of virtual work for the structure in its current configuration we can write,

$$\underbrace{\int_{\partial\Omega_s} t\, \boldsymbol{\sigma} : \delta\boldsymbol{\epsilon}\, dS}_{\delta W_{int}} - \underbrace{\int_{\partial\Omega_s} \boldsymbol{t} \cdot \delta\boldsymbol{u}\, dS}_{\delta W_{ext}} = 0, \qquad \forall \delta\boldsymbol{u} \in \mathcal{C}_u \tag{6}$$

where $\boldsymbol{\sigma}$ is the Cauchy stress tensor and $\boldsymbol{\epsilon} = \frac{1}{2}\left(\nabla_{\boldsymbol{x}}\delta\boldsymbol{u} + \nabla_{\boldsymbol{x}}^T\delta\boldsymbol{u}\right)$ is the virtual Eulerian strain tensor, with $\nabla_{\boldsymbol{x}}\bullet = \frac{\partial\bullet}{\partial\boldsymbol{x}}$, $\delta\boldsymbol{u}$ as virtual displacement field and $\mathcal{C}_u$ 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 (δW_{int}), and the external virtual work (δW_{ext}). In the total Lagrangian formulation, the internal virtual work is written in the reference configuration as,

$$\delta W_{int} = \int_{\partial\Omega_s^0} t\, \boldsymbol{S} : \delta\boldsymbol{E}\, dS^0, \tag{7}$$

where $\boldsymbol{S}$ is the second Piola-Kirchhoff stress tensor and $\delta\boldsymbol{E} = \frac{1}{2}\left(\delta\boldsymbol{F}^T\boldsymbol{F} + \boldsymbol{F}^T\delta\boldsymbol{F}\right)$ is the virtual Green-Lagrange strain tensor, with $\delta\boldsymbol{F} = \nabla_{\boldsymbol{X}}\delta\boldsymbol{u}$ and $\boldsymbol{F} =$

$\nabla_{\boldsymbol{X}}\boldsymbol{u}$. The operator $\nabla_{\boldsymbol{X}}\bullet = \frac{\partial \bullet}{\partial \boldsymbol{X}}$ is defined as the gradient of a field with respect to the material coordinate $\boldsymbol{X}$. In a typical displacement based approach [18], the internal and external work are expressed in terms of the displacement field as unknown. To that end, the stress tensor at any point is expressed as a function of the strain tensor which is in turn written as a function of the displacement field using the strain definition above.

The external virtual work is generally written in terms of the quantities in the current configuration which depends on the traction field $\boldsymbol{t}$ from the ponding fluid resulting in the final expression of the principle of virtual work that needs to be satisfied at equilibrium,

$$\int_{\partial\Omega_s^0} t\ \boldsymbol{S} : \delta\boldsymbol{E}\ dS^0 - \int_{\partial\Omega_s} \boldsymbol{t}\cdot\delta\boldsymbol{u}\ dS = 0. \tag{8}$$

2.3. Constitutive models

The relation between stress and strain tensor is described using the constitutive model or material law. In the numerical example presented in the paper, two types of hyper-elastic materials are used: the Saint-Venant Kirchhoff material law, given in Eq. (9) and the incompressible Mooney-Rivlin material law, given in Eq. (10). The former material law is applicable for large displacements and small strains cases, while the latter is applicable for large displacements and finite strains [19].

$$\boldsymbol{S}_{SV} = 2\mu\boldsymbol{E} + \lambda tr(\boldsymbol{E})\boldsymbol{I}, \tag{9}$$

$$\boldsymbol{S}_{MR} = \beta\boldsymbol{C}^{-1} + 2\frac{\partial\Psi_{MR}(\boldsymbol{C})}{\partial\boldsymbol{C}}, \tag{10}$$

where the subscripts SV and MR stand for Saint-Venant Kirchhoff and Mooney Rivlin, respectively. As clear from Eq.(9), the relation between the

2^{nd} Piola Kirchhoff and the Green-Lagrange strain tensor $\boldsymbol{E} = \frac{1}{2}\left(\boldsymbol{F}^T\boldsymbol{F} - \boldsymbol{I}\right)$ is linear. The two constants appearing in Eq. (9) are called Lamé constants which are related to the material properties, Young's modulus E and Poisson's ratio ν as

$$\lambda = \frac{\nu E}{(1+\nu)(1-2\nu)}, \mu = \frac{E}{2(1+\nu)}. \tag{11}$$

The relation between stress and strain tensor for the incompressible Mooney-Rivlin material on the other hand is non-linear. The expression of the 2^{nd} Piola-Kirchhoff stress tensor is given in terms of the Cauchy-Green strain tensor ($\boldsymbol{C} = \boldsymbol{F}^T\boldsymbol{F}$), strain energy function Ψ_{MR} and Lagrange multiplier β. The most commonly used expression of the strain energy function is written in terms of the first invariant (I_1) and second invariant (I_2) of the Cauchy-Green strain tensor,

$$\Psi_{MR} = c_1(I_1 - 3) + c_2(I_2 - 3), \tag{12}$$

where $I_1 = tr(\mathbf{C})$ and $I_2 = \frac{1}{2}\left(I_1^2 - tr(\mathbf{C}^T\mathbf{C})\right)$ with material constants c_1 and c_2 [20].

For the plane stress case the 2^{nd} Piola-Kirchhoff stress tensor can be further simplified because the components along the thickness direction vanish, leading to a form:

$$\boldsymbol{S} = \begin{bmatrix} S_{11} & S_{12} & 0 \\ S_{21} & S_{22} & 0 \\ 0 & 0 & 0 \end{bmatrix}, \tag{13}$$

where direction 3 is normal to the membrane surface (thickness direction).

Moreover, by using the small thickness assumption of the membrane as compared to the other spatial dimensions, the off-diagonal components along the thickness direction can be neglected. As a result, the Cauchy-Green tensor and Green-Lagrange strain tensor have the following simplified forms:

$$\boldsymbol{C} = \begin{bmatrix} C_{11} & C_{12} & 0 \\ C_{21} & C_{22} & 0 \\ 0 & 0 & C_{33} \end{bmatrix}, \boldsymbol{E} = \begin{bmatrix} E_{11} & E_{12} & 0 \\ E_{21} & E_{22} & 0 \\ 0 & 0 & E_{33} \end{bmatrix}. \tag{14}$$

Using the condition $S_{33} = 0$, we get the linear stress-strain relation for Saint-Venant Kirchhoff material expressed in Voigt notation,

$$\begin{bmatrix} S_{11} \\ S_{22} \\ S_{12} \end{bmatrix} = \frac{E}{1-\nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix} \begin{bmatrix} E_{11} \\ E_{22} \\ 2E_{12} \end{bmatrix}. \tag{15}$$

Similarly, for the case of the Mooney-Rivlin material law the value of β is determined by using the plane stress condition $S_{33} = 0$ and using the Cauchy-Green tensor of the form given in Eq. (14). When the obtained value of β is substituted in Eq. (10), we get the following relation between stress and strain:

$$\begin{bmatrix} S_{11} \\ S_{22} \\ S_{12} \end{bmatrix} = 2c_1 \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} - \frac{2c_1}{(C_{11}C_{22} - C_{12}^2)^2} \begin{bmatrix} C_{22} \\ C_{11} \\ -C_{12} \end{bmatrix} + \frac{2c_2}{C_{11}C_{22} - C_{12}^2} \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}$$
$$+ 2c_2 \left(1 - \left(C_{11}C_{22} - C_{12}^2\right)^2 (C_{11} + C_{22})\right) \begin{bmatrix} C_{22} \\ C_{11} \\ -C_{12} \end{bmatrix}. \tag{16}$$

3. Solver components

The deformed shape of the membrane is found when the virtual work equation in Eq. (8) is satisfied together with the fluid equations given in Eq. (4) and (5). The virtual work equation is solved using a structural solver with the load boundary condition from Eq. (4), while the volume conservation constraint is implemented using a volume-conserving solver. Three of the four methods discussed in this paper find the solution by coupling these solvers to determine the deformed shape. The only exception is the second monolithic method (MVCC), where the structural equations are modified to include the volume conservation constraint without using the volume conservation solver, which will be discussed in Section 5.2.

3.1. Structural solver

In this section we will briefly discuss how the structural solver solves the virtual work equation Eq. (8), given a traction field $\boldsymbol{t}$ from the ponding fluid. Using the stress-strain relation given in Eqs. (15) and (16) and the strain definition discussed before, we can express the internal virtual work in terms

of only the displacement field as unknown. The resultant expression will be non-linear for large displacements, regardless of the use of linear or non-linear material law, as $\boldsymbol{C}$ and $\boldsymbol{E}$ are non-linear functions of the displacement field. Moreover, if the traction or the resultant external forces on the structure depends on the deformed state, the external virtual work will be also a non-linear function of $\boldsymbol{u}$. In fact, the hydrostatic forces on the structure due to ponding is one such example of so-called follower forces.

The finite element discretization of the internal and external virtual work gives a non-linear residual equation, Eq. (17) as a function of the nodal displacement vector $\hat{\boldsymbol{u}}$, where the displacement and virtual displacement field are approximated using the shape function matrix $\boldsymbol{N}$ as $\boldsymbol{u} \approx \boldsymbol{u}^h = \boldsymbol{N}\hat{\boldsymbol{u}}$ and $\delta\boldsymbol{u} \approx \delta\boldsymbol{u}^h = \boldsymbol{N}\delta\hat{\boldsymbol{u}}$, respectively. The superscript $\bullet^h$ represents the approximation of a given field with finite element discretization and the accent $\hat{\bullet}$ represents the associated nodal vector for the approximation. The resultant residual equation from the discretized virtual work expression can be written as,

$$\hat{\boldsymbol{f}}_{ext}(\hat{\boldsymbol{u}}) - \hat{\boldsymbol{f}}_{int}(\hat{\boldsymbol{u}}) = 0, \tag{17}$$

where $\hat{\boldsymbol{f}}_{int}$ is the internal nodal forces and $\hat{\boldsymbol{f}}_{ext}$ is the external nodal forces. The structural solver used in the current work is implemented in an open-source FEM code called KRATOS [21], which uses the N-R algorithm to solve the vector equation given in Eq. (17), where at every iteration we solve a linear system

$$\boldsymbol{K}_{tan}\Delta\hat{\boldsymbol{u}} = \hat{\boldsymbol{r}} \tag{18}$$

to obtain the update in the nodal displacements $\Delta\hat{\boldsymbol{u}}$, where $\hat{\boldsymbol{r}} = \hat{\boldsymbol{f}}_{ext} - \hat{\boldsymbol{f}}_{int}$ is the out of balance force vector or residual vector, and $\boldsymbol{K}_{tan}$ is the tangent stiffness matrix, where

$$\boldsymbol{K}_{tan} = \boldsymbol{K}_{mem} - \boldsymbol{K}_{l}. \tag{19}$$

The matrix $\boldsymbol{K}_{mem}$ is the familiar global tangent stiffness matrix from membrane elements, interested readers can refer to [22] for more details. The second matrix in Eq. (19) is called the load stiffness matrix, which depends on the type of follower load [8, 23]. In the current paper, the follower tangent stiffness matrix $\boldsymbol{K}_l$ depends on the type of approach, whether it is partitioned or monolithic.

3.2. Volume-conserving solver

The ponding fluid on membrane structures (generally water) is always incompressible and therefore the volume of the ponding fluid is always conserved. Moreover, under static conditions the free surface of the fluid is flat and perpendicular to gravity i.e. normal to the z-direction in Fig. 1. These properties of the ponding fluid under static conditions are used to update the free surface position using an algorithm called the volume-conserving solver. The volume-conserving solver consists of two components: a volume calculation algorithm and an iterative algorithm to conserve a given volume by moving the flat and horizontal free surface vertically. The volume of the fluid enclosed by $\partial\Omega_f \cup \partial\Omega_{fs}$ in Fig. 1 can be calculated using the relation,

$$V_f = \int_{\partial\Omega_{fs}} (z - z_f)\, \boldsymbol{e}_z \cdot \bar{\boldsymbol{n}}\, dS \tag{20}$$

as already discussed in Section 2.1, and its derivative with respect to z_f,

$$\frac{\partial V_f}{\partial z_f} = \int_{\partial\Omega_{fs}} -\boldsymbol{e}_z \cdot \bar{\boldsymbol{n}}\, dS = A_f, \tag{21}$$

where A_f is the area of the free surface. In the current work, the leap-frogging Newton's method is used for conserving a given volume. This method is discussed in detail in [24]. It consists of a Newton step followed by a pseudo secant step, as shown in Fig. 2. The main advantage of this method is that it has cubic convergence at a simple root with computational efficiency comparable to that of Newton's method. Newton's method and leap-frogging Newton method were tested for volume conservation with some axi-symmetric geometries and it was found that the leap-frogging Newton was much more robust and had faster convergence rate than Newton's method. Hence, it was chosen over the other. The equations used for iteration to conserve volume are given in Eqs. (22) and (23), 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 $\check{\bullet}$ is used to specify quantities at the intermediate position. It should be noted that the denominator in Eq. (23) can cause rounding-off problems, as it could become very small quickly. To avoid this problem, using the machine precision n_{pre} we add another stopping criteria for the iterations, $|f(z_f^m) - f(\check{z}_f^m)| = |V_f^m - \check{V}_f^m| < 10^{n_{pre}-1}$.

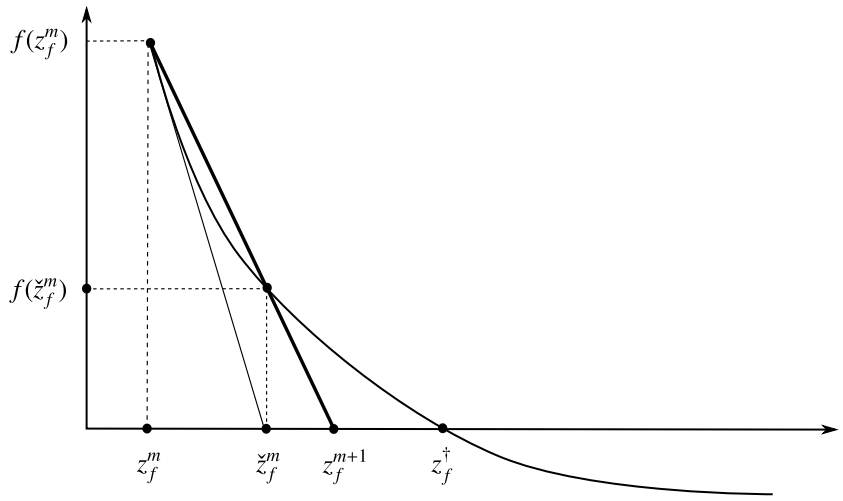


Figure 2: Leap froging Newton's algorithm adopted from [24], with $z_f^\dagger$ as the root of the function $f(z_f)$.

$$\check{z}_f^m = z_f^m - \frac{f(z_f^m)}{f'(z_f^m)} \tag{22}$$

$$z_f^{m+1} = z_f^m - \frac{f(z_f^m)^2}{f'(z_f^m)\left(f(z_f^m) - f(\check{z}_f^m)\right)} \tag{23}$$

Using Eqs. (20)-(23) we can write an algorithm for the volume-conserving solver. As already discussed, the leap-frogging Newton algorithm is robust and has good convergence characteristics. However, for certain cases it would update the free surface position below the wetted surface. As a result, the algorithm would fail to update the free surface in the next step since the calculated volume and free surface area would be zero. One example of such a case is shown in Fig. 3, where the intermediate position of the free surface plane after the n^{th} iteration goes below the membrane surface i.e. $\check{z}_f^m < z_f^*$ or $\Delta\check{z}_f^m < z_f^* - z_f$ with $V_f(z_f^*) = 0$. As evident from Fig. 3a the limit value

of the slope is given by $f'_{min} < (V_f^m - 0)/(z_f^m - z_f^*)$. Therefore, the standard leap-frogging Newton algorithm is modified to limit the slope to avoid such cases and the final proposed algorithm can be found in Algorithm 1.

Algorithm 1 Modified Leap-frogging Newton's method for volume conservation.

$m = 0$
while $\left|\frac{V_f^m - V_t}{V_t}\right| > \varepsilon$ and $|V_f^m - \check{V}_f^m| < 10^{n_{pre}-1}$ and $m < m_{max}$ **do**
    Calculate $f(z_f^m) = V_f^m - V_t$
    Calculate $f'(z_f^m) = A_f^m$
    if $\Delta\check{z}_f^{m+1} < (z_f^* - z_f^m)$ **then**
        $A_f = V_f^m/(z_f^m - z_f^*)$
    end if
    Calculate $\check{z}_f^m$ using Eq. (22). Move the plane to $\check{z}_f^m$.
    Calculate $f(\check{z}_f^m) = \check{V}_f^m - V_t$.
    Calculate z_f^{m+1} using Eq. (23). Move the plane to z_f^{m+1}.
    $m = m + 1$
end while

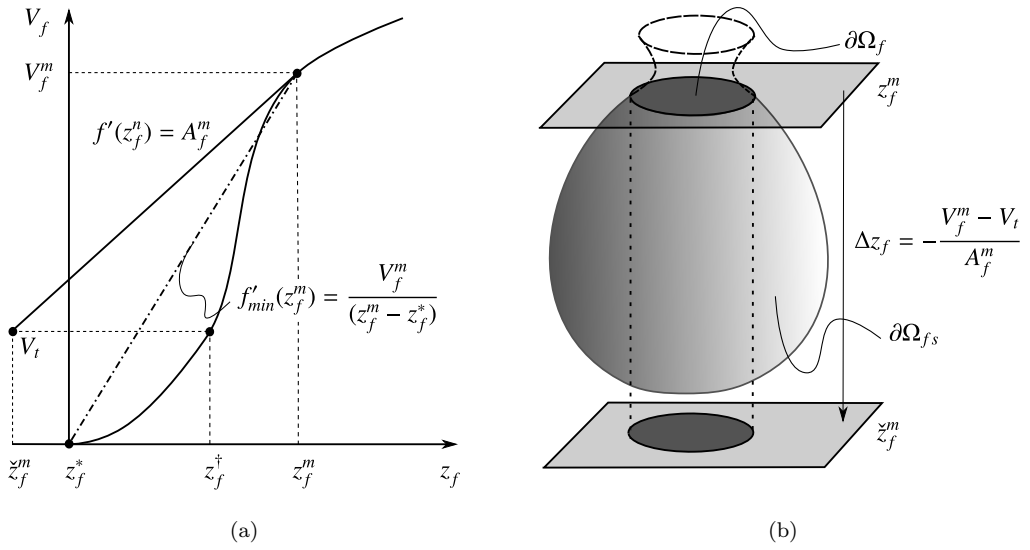


Figure 3: An example where the standard leap-frogging Newton volume conservation algorithm will fail, when implemented without any condition on the slope of the volume residual: a) volume of the ponding fluid V_f vs position of the free surface z_f showing the minimum slope to avoid failure of the algorithm , b) corresponding membrane structure and the free surface update that will move the free surface below the membrane surface.

4. Linearization of the fluid loading

The ponding fluid interacts with the structure by applying traction on the wetted region, which manifests as an external nodal force vector $\hat{\boldsymbol{f}}_{ext}$ on the structure. It will be explained in this section that the external nodal force vector is a non-linear function of the displacement field. Therefore, by including the linearized behavior of the fluid loading in the N-R iteration convergence speed can be greatly improved. In terms of implementation this means including a load stiffness matrix, mentioned in Section 3.1 in the N-R iterations of the structural solver. In this section, we present the full linearization of the fluid loading and discuss the load stiffness matrices associated with the different contributions to the load behavior of the fluid. The derived load stiffness matrices will be used fully or partially depending on the method.

The expression of the nodal force vector can be obtained by considering the discretized virtual external work δW^h_{ext} associated with δW_{ext} in Eq. (6) and using the traction $\boldsymbol{t}$ from Eq. (4),

$$\begin{aligned}
\delta W^h_{ext} &= \int_{\partial\Omega_{fs}} \delta\boldsymbol{u}^h \cdot -\gamma_f (z^h - z_f)\bar{\boldsymbol{n}}^h \, dS + \int_{\partial\Omega_s \setminus \partial\Omega_{fs}} \delta\boldsymbol{u}^h \cdot \boldsymbol{0} \, dS \\
&= -\gamma_f \int_\eta \int_\xi \delta\boldsymbol{u}^h \cdot (z^h - z_f)\boldsymbol{n}^h \, d\xi d\eta \\
&= \delta\hat{\boldsymbol{u}}^T \underbrace{\left(-\gamma_f \int_\eta \int_\xi (z^h - z_f)\boldsymbol{N}^T \boldsymbol{n}^h \, d\xi d\eta \right)}_{\hat{\boldsymbol{f}}_{ext}} = \delta\hat{\boldsymbol{u}}^T \hat{\boldsymbol{f}}_{ext}.
\end{aligned} \tag{24}$$

In the equations above it can be seen that the domain of integration is changed to the parametric space $\xi - \eta$ of the discretized wetted surface with

base vectors $\boldsymbol{g}_{\xi}^{h} = \boldsymbol{x}_{,\xi}^{h}$ and $\boldsymbol{g}_{\eta}^{h} = \boldsymbol{x}_{,\eta}^{h}$, where $\bullet_{,r} = \frac{\partial \bullet}{\partial r}$ for any parameter r. This transformation of integration domain uses the definitions of a normal vector at any point on the discretized wetted surface $\boldsymbol{n}^{h} = \boldsymbol{g}_{\xi}^{h} \times \boldsymbol{g}_{\eta}^{h} = \left\| \boldsymbol{g}_{\xi}^{h} \times \boldsymbol{g}_{\eta}^{h} \right\| \ \bar{\boldsymbol{n}}^{h}$ and an infinitesimal surface area $dS = \left\| \boldsymbol{g}_{\xi}^{h} \times \boldsymbol{g}_{\eta}^{h} \right\| \ d\xi d\eta$. It can observed that the discretized virtual external work and consequently nodal external force vector is a non-linear function of nodal displacement vector $\hat{\boldsymbol{u}}$ as

$$
\begin{aligned}
\boldsymbol{n}^{h} &= \boldsymbol{g}_{\xi}^{h} \times \boldsymbol{g}_{\eta}^{h} = \boldsymbol{x}_{,\xi}^{h}\left(\hat{\boldsymbol{u}}\right) \times \boldsymbol{x}_{,\eta}^{h}\left(\hat{\boldsymbol{u}}\right), \\
z &= \boldsymbol{x}^{h}\left(\hat{\boldsymbol{u}}\right) \cdot \boldsymbol{e}_{z},
\end{aligned}
$$

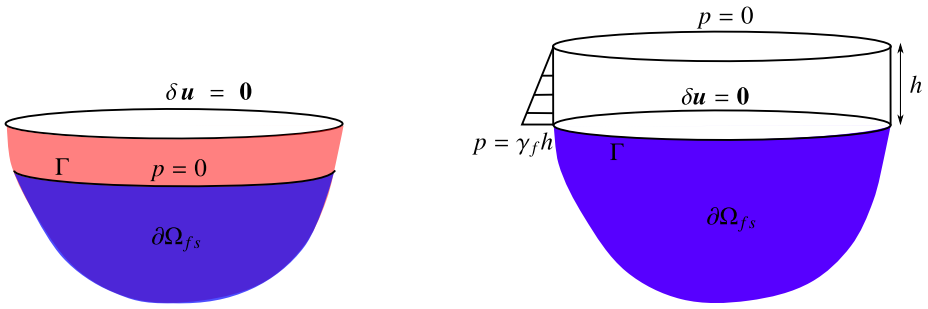
and $z_f = z_f\left(\hat{\boldsymbol{u}}\right)$ from the volume conservation constraint. Linearizing the discretized virtual work we get,

$$
\begin{aligned}
\delta W_{ext}^{h} + \Delta\delta W_{ext}[\Delta \boldsymbol{u}^{h}] = & - \int_{\eta}\int_{\xi} \delta \boldsymbol{u}^{h} \cdot \gamma_f (z^{h} - z_f) \boldsymbol{n}^{h} \ d\xi d\eta \\
& \underbrace{- \gamma_f \int_{\eta}\int_{\xi} (z^{h} - z_f) \ \delta \boldsymbol{u}^{h} \cdot \ \Delta \boldsymbol{n}^{h}[\Delta \boldsymbol{u}^{h}] \ d\xi d\eta}_{\Delta\delta W_{ext}^{\Delta \boldsymbol{n}}[\Delta \boldsymbol{u}^{h}]} \\
& \underbrace{-\gamma_f \int_{\eta}\int_{\xi} \delta \boldsymbol{u}^{h} \cdot \Delta z^{h}[\Delta \boldsymbol{u}^{h}] \ \boldsymbol{n}^{h} \ d\xi d\eta}_{\Delta\delta W_{ext}^{\Delta z}[\Delta \boldsymbol{u}^{h}]} \\
& \underbrace{+ \gamma_f \int_{\eta}\int_{\xi} \delta \boldsymbol{u}^{h} \cdot \Delta z_f[\Delta \boldsymbol{u}^{h}] \ \boldsymbol{n}^{h} \ d\xi \eta}_{\Delta\delta W_{ext}^{\Delta z_f}[\Delta \boldsymbol{u}^{h}]} .
\end{aligned} \tag{25}
$$

The linear part of change in external work due to $\Delta \boldsymbol{u}^{h}$ can be split into three components: $\Delta\delta W_{ext}^{\Delta \boldsymbol{n}}[\Delta \boldsymbol{u}^{h}]$, $\Delta\delta W_{ext}^{\Delta z}[\Delta \boldsymbol{u}^{h}]$ and $\Delta\delta W_{ext}^{\Delta z_f}[\Delta \boldsymbol{u}^{h}]$. The second term in rhs of Eq. (25) $\Delta\delta W_{ext}^{\Delta \boldsymbol{n}}[\Delta \boldsymbol{u}^{h}]$ accounts for the change in

normal vector due to the wetted surface movement with constant hydrostatic pressure. The effect of change in hydrostatic pressure from the movement of the wetted surface alone is represented by $\Delta\delta W_{ext}^{\Delta z}[\Delta \boldsymbol{u}^h]$. Finally, the contribution from the movement of the free surface to conserve volume is captured by $\Delta\delta W_{ext}^{\Delta z_f}[\Delta \boldsymbol{u}^h]$. The three components of the change in external virtual work can be written in the form of $\delta\hat{\boldsymbol{u}}^T \boldsymbol{K}_l^{\bullet} \Delta\hat{\boldsymbol{u}}$ such that the linear part of change in the nodal force vector due to the nodal displacement increment $\Delta\hat{\boldsymbol{u}}$ can be written as $\Delta\hat{\boldsymbol{f}}^{\bullet}[\Delta\hat{\boldsymbol{u}}] = \boldsymbol{K}_l^{\bullet}\Delta\hat{\boldsymbol{u}}$, where superscript $\bullet$ represents the three contributions that we discussed before. Thus, we have three load stiffness matrices: $\boldsymbol{K}_l^{\Delta \boldsymbol{n}}$, $\boldsymbol{K}_l^{\Delta z}$ and $\boldsymbol{K}_l^{\Delta z_f}$.

Additionally, it is well known that a constant pressure and hydrostatic pressure loading on large displacement cases are conservative [25]. Therefore, the associated load stiffness matrices are symmetric. The proof of symmetry for the constant pressure can be found in [23] and for hydrostatic pressure with constant fluid volume can be found in [8, 9]. The symmetric part of the load stiffness matrices is obtained by performing integration by parts and some algebraic manipulations. The interested readers are encouraged to refer to [8] for detailed derivation. In the derivation, they list five conditions at the boundary of the wetted surface Γ to have symmetric load stiffness matrices. If atleast one of the conditions is satisfied it would lead to symmetric load stiffness matrices. Among these, either (i) $p = 0$ or (ii) $\delta\boldsymbol{u} = 0$ on Γ is always satisfied in ponding scenario, see Fig. 4. With the derivation of the symmetric load stiffness matrices already given in some of the previous work [8, 9] and more recently [11], we directly state the linear part of the change in discretized external virtual work containing only symmetric terms:

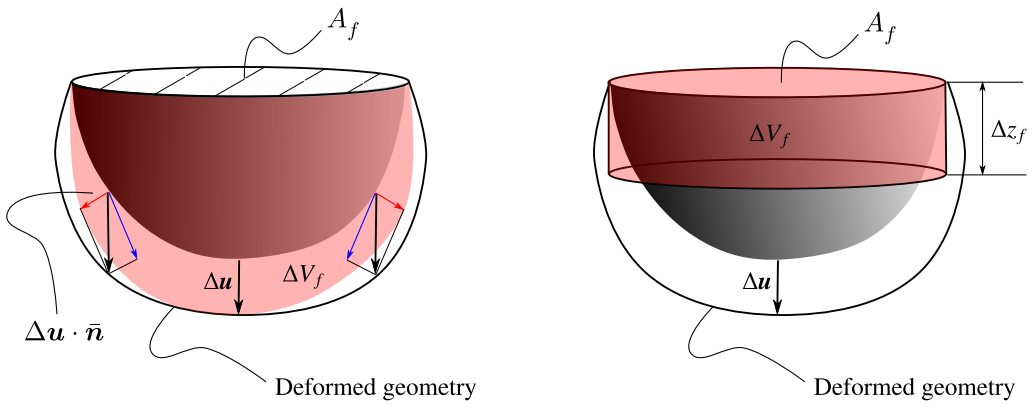


(a) (b)

Figure 4: Two scenarios of ponding on a membrane structure where the boundary conditions at the wetted surface Γ lead to symmetric load stiffness matrices: a) the membrane structure is partially filled ($p = 0$ at Γ) and b) the membrane structure is fully filled and the free surface moves above the fixed boundary ($\delta \boldsymbol{u} = \mathbf{0}$ at Γ).

$$
\begin{aligned}
\Delta\delta W^h_{ext}[\Delta \boldsymbol{u}^h] = &-\frac{1}{2}\gamma_f \int_\eta \int_\xi (z^h - z_f)\ \left(\delta \boldsymbol{u}^h_{,\xi} \times \boldsymbol{g}^h_\eta - \delta \boldsymbol{u}^h_{,\eta} \times \boldsymbol{g}^h_\xi\right) \cdot \Delta \boldsymbol{u}^h\ d\xi d\eta \\
&+\frac{1}{2}\gamma_f \int_\eta \int_\xi (z^h - z_f)\ \delta \boldsymbol{u}^h \cdot \left(\boldsymbol{g}^h_\eta \times \Delta \boldsymbol{u}^h_{,\xi} - \boldsymbol{g}^h_\xi \times \Delta \boldsymbol{u}^h_{,\eta}\right)\ d\xi d\eta \\
&-\frac{1}{2}\gamma_f \int_\eta \int_\xi \delta \boldsymbol{u}^h \cdot \left(\boldsymbol{n}^h \otimes \boldsymbol{e}_z + \boldsymbol{e}_z \otimes \boldsymbol{n}^h\right) \cdot \Delta \boldsymbol{u}^h\ d\xi d\eta \\
&-\frac{\gamma_f}{A_f} \int_\eta \int_\xi \delta \boldsymbol{u}^h \cdot \boldsymbol{n}^h\ d\xi d\eta \int_\eta \int_\xi \boldsymbol{n}^h \cdot \Delta \boldsymbol{u}^h\ d\xi d\eta, \qquad (26)
\end{aligned}
$$

where the terms in the first three lines are the sum of the contributions from change in normal and change in hydrostatic pressure from the movement of the wetted surface alone i.e $\Delta\delta W^{\Delta \boldsymbol{n}}_{ext} + \Delta\delta W^{\Delta z}_{ext}$. The last term is the contribution from the free surface movement to conserve volume ($\Delta\delta W^{\Delta z_f}_{ext}$), which is obtained by substituting the expression of the linear part of the free surface movement $\Delta z_f[\Delta \boldsymbol{u}^h]$ given in Eq. (27) in the expression of $\Delta\delta W^{\Delta z_f}_{ext}$



(a) (b)

Figure 5: Volume conservation using the linear part of the change in volume from membrane deformation: a) the linear part of change in volume ΔV_f due to deformation of the wetted surface, indicated in red color, b) free surface update Δz_f by considering a cylinder of volume ΔV_f with base area equal to the free surface area and height equal to Δz_f.

in Eq. (25). The linear part of the free surface movement can be obtained by an observation that only the normal component of the wetted surface displacement contributes to volume change and dividing the obtained volume change (ΔV_f) by the free surface area gives the linear part of the free surface movement, as illustrated in Fig. 5.

$$\Delta z_f[\Delta \boldsymbol{u}^h] = -\frac{\Delta V_f}{A_f} = -\frac{\int_\eta \int_\xi \left(\Delta \boldsymbol{u}^h \cdot \boldsymbol{n}^h\right)\, d\xi d\eta}{A_f} \tag{27}$$

To obtain the load stiffness matrices the external virtual work expression in Eq. (26) can be written in terms of the associated nodal vectors and shape function matrix, which leads to the following expression:

$$
\begin{aligned}
\Delta\delta W_{ext}[\Delta \boldsymbol{u}^h] = &- \delta\hat{\boldsymbol{u}}^T \frac{1}{2}\gamma_f \int_\eta \int_\xi (z^h - z_f)\ \left(\boldsymbol{N}_{,\xi}^T \boldsymbol{\Omega}_\eta^h \boldsymbol{N} - \boldsymbol{N}_{,\eta}^T \boldsymbol{\Omega}_\xi^h \boldsymbol{N}\right)\ d\xi d\eta\ \Delta\hat{\boldsymbol{u}} \\
&+ \delta\hat{\boldsymbol{u}}^T \frac{1}{2}\gamma_f \int_\eta \int_\xi (z^h - z_f)\ \left(\boldsymbol{N}^T \boldsymbol{\Omega}_\eta^h \boldsymbol{N}_{,\xi} - \boldsymbol{N}^T \boldsymbol{\Omega}_\xi^h \boldsymbol{N}_{,\eta}\right)\ d\xi d\eta\ \Delta\hat{\boldsymbol{u}} \\
&- \delta\hat{\boldsymbol{u}}^T \frac{1}{2}\gamma_f \int_\eta \int_\xi \boldsymbol{N}^T \left(\boldsymbol{n}^h \otimes \boldsymbol{e}_z + \boldsymbol{e}_z \otimes \boldsymbol{n}^h\right) \boldsymbol{N}\ d\xi d\eta\ \Delta\hat{\boldsymbol{u}} \\
&- \delta\hat{\boldsymbol{u}}^T \frac{\gamma_f}{A_f} \left(\int_\eta \int_\xi \boldsymbol{N}^T \boldsymbol{n}^h\ d\xi d\eta\right) \left(\int_\eta \int_\xi \boldsymbol{N}^T \boldsymbol{n}^h\ d\xi d\eta\right)^T \Delta\hat{\boldsymbol{u}}.
\end{aligned}
\tag{28}
$$

where $\boldsymbol{\Omega}_\xi^h$ and $\boldsymbol{\Omega}_\eta^h$ are the skew matrices associated with the cross product of the base vectors $\boldsymbol{g}_\xi^h$ and $\boldsymbol{g}_\eta^h$, respectively. Finally, the symmetric load stiffness matrix associated with each part can be extracted by comparing with the expression $\delta\hat{\boldsymbol{u}}^T \boldsymbol{K}_l^\bullet \Delta\hat{\boldsymbol{u}}$,

$$
\begin{aligned}
\boldsymbol{K}_l^{\Delta \boldsymbol{n}} = &- \frac{1}{2}\gamma_f \int_\eta \int_\xi (z^h - z_f)\ \left(\boldsymbol{N}_{,\xi}^T \boldsymbol{\Omega}_\eta^h \boldsymbol{N} - \boldsymbol{N}_{,\eta}^T \boldsymbol{\Omega}_\xi^h \boldsymbol{N}\right)\ d\xi d\eta \\
&+ \frac{1}{2}\gamma_f \int_\eta \int_\xi (z^h - z_f)\ \left(\boldsymbol{N}^T \boldsymbol{\Omega}_\eta^h \boldsymbol{N}_{,\xi} - \boldsymbol{N}^T \boldsymbol{\Omega}_\xi^h \boldsymbol{N}_{,\eta}\right)\ d\xi d\eta \\
\boldsymbol{K}_l^{\Delta z} = &- \frac{1}{2}\gamma_f \int_\eta \int_\xi \boldsymbol{N}^T \left(\boldsymbol{n}^h \otimes \boldsymbol{e}_z + \boldsymbol{e}_z \otimes \boldsymbol{n}^h\right) \boldsymbol{N}\ d\xi d\eta \qquad (29) \\
\boldsymbol{K}_l^{\Delta z_f} = &- \frac{\gamma_f}{A_f} \left(\int_\eta \int_\xi \boldsymbol{N}^T \boldsymbol{n}^h\ d\xi d\eta\right) \left(\int_\eta \int_\xi \boldsymbol{N}^T \boldsymbol{n}^h\ d\xi d\eta\right)^T \Delta\hat{\boldsymbol{u}}. \qquad (30)
\end{aligned}
$$

5. Monolithic methods for ponding analysis

5.1. Monolithic method with volume conservation inside structural iterations (MVCIS)

Having discussed the linearized equations for the fluid loading in the previous section, we can now start using it in the different methods for ponding

analysis. The first method which is classified as 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. In terms of implementation this means we use the load stiffness matrices discussed in Section 4 along with the nested iterations of the volume-conserving solver to update the free surface after every N-R iteration. The complete algorithm is written in Algorithm. 2

Algorithm 2 Monolithic method with volume conservation inside structural iterations (MVCIS)

$n = 0$
Find z_f^0 using Algorithm 1 with V_t as input argument
while $\left\| \hat{\boldsymbol{f}}_{ext} - \hat{\boldsymbol{f}}_{int} \right\| > \varepsilon$ and $n < n_{max}$ **do**
    Update: $\boldsymbol{K}_{mem}$, $\boldsymbol{K}_l^{\Delta \boldsymbol{n}}$, $\boldsymbol{K}_l^{\Delta z}$, $\boldsymbol{K}_l^{\Delta z_f}$, $\hat{\boldsymbol{f}}_{ext}$ and $\hat{\boldsymbol{f}}_{int}$ using $\hat{\boldsymbol{u}}^n$ and z_f^n
    Solve: $\left(\boldsymbol{K}_{mem} - \boldsymbol{K}_l^{\Delta \boldsymbol{n}} - \boldsymbol{K}_l^{\Delta z} - \boldsymbol{K}_l^{\Delta z_f} \right) \Delta \hat{\boldsymbol{u}}^{n+1} = \hat{\boldsymbol{f}}_{ext} - \hat{\boldsymbol{f}}_{int}$
    Update displacement: $\hat{\boldsymbol{u}}^{n+1} = \hat{\boldsymbol{u}}^n + \Delta \hat{\boldsymbol{u}}^{n+1}$
    Update structure: $\hat{\boldsymbol{x}}^{n+1} = \hat{\boldsymbol{X}} + \hat{\boldsymbol{u}}^{n+1}$
    Update free surface using Algorithm 1 with V_t as input argument
    $n = n + 1$
end while

5.2. Monolithic method with volume conservation as a constraint (MVCC)

The problem of determining the static deformation of a structure under the load of fixed volume of fluid in a monolithic approach with volume conservation as constraint $g(\hat{\boldsymbol{u}}, z_f)$ can be stated as follows:

$$\hat{\boldsymbol{f}}_{int}(\hat{\boldsymbol{u}}) - \hat{\boldsymbol{f}}_{ext}(\hat{\boldsymbol{u}}, z_f) = 0, \tag{31}$$

$$g(\hat{\boldsymbol{u}}, z_f) = V_f(\hat{\boldsymbol{u}}, z_f) - V_t = 0. \tag{32}$$

Here, we introduce an additional independent variable z_f, the z-coordinate of the free surface, which allows the volume conservation constraint to be incorporated in the system of equations. The system of equations given in Eqs. (31) and (32) can be solved using N-R algorithm, where the linearized form at iteration $n+1$ can be written as:

$$\left(\frac{\partial \hat{\boldsymbol{f}}_{int}\left(\hat{\boldsymbol{u}}^n\right)}{\partial \hat{\boldsymbol{u}}} - \frac{\partial \hat{\boldsymbol{f}}_{ext}\left(\hat{\boldsymbol{u}}^n, z_f^n\right)}{\partial \hat{\boldsymbol{u}}}\right)\Delta\hat{\boldsymbol{u}}^{n+1} - \frac{\partial \hat{\boldsymbol{f}}_{ext}\left(\hat{\boldsymbol{u}}^n, z_f^n\right)}{\partial z_f}\Delta z_f^{n+1} = \hat{\boldsymbol{f}}_{ext}\left(\hat{\boldsymbol{u}}^n, z_f^n\right) - \hat{\boldsymbol{f}}_{int}\left(\hat{\boldsymbol{u}}^n\right) \tag{33}$$

$$\frac{\partial g\left(\hat{\boldsymbol{u}}^n, z_f^n\right)}{\partial \hat{\boldsymbol{u}}}\Delta\hat{\boldsymbol{u}}^{n+1} + \frac{\partial g\left(\hat{\boldsymbol{u}}^n, z_f^n\right)}{\partial z_f}\Delta z_f^{n+1} = V_t - V\left(\hat{\boldsymbol{u}}^n, z_f^n\right). \tag{34}$$

In Eq. (33), $\frac{\partial \hat{\boldsymbol{f}}_{int}}{\partial \hat{\boldsymbol{u}}}$ is the familiar global membrane tangent stiffness matrix $\boldsymbol{K}_{mem}$ [22]. The second term, $\frac{\partial \hat{\boldsymbol{f}}_{ext}}{\partial \hat{\boldsymbol{u}}}$ in the equation is the sum $\boldsymbol{K}_l^{\Delta \boldsymbol{n}} + \boldsymbol{K}_l^{\Delta z}$, discussed in Section 4. The derivative $\frac{\partial \hat{\boldsymbol{f}}_{ext}}{\partial z_f}$ can be obtained by differentiating $\hat{\boldsymbol{f}}_{ext}$ given in Eq. (24) with respect to free surface height. We have not discussed this before as z_f was not an independent variable. The obtained expression is given in Eq. (35). To obtain the terms in Eq. (34), we only need to calculate the derivative of the fluid volume with respect to the variables, since V_t is constant. The first term in the left-hand side of Eq. (34) represents the change in fluid volume with respect to the nodal displacement vector. This can be obtained by substituting $\Delta \boldsymbol{u}^h = \boldsymbol{N}\Delta\hat{\boldsymbol{u}}$ in the expression of ΔV_f

in Eq. 27, writing in the form of $\bullet\ \Delta\hat{\boldsymbol{u}}$ and removing $\Delta\hat{\boldsymbol{u}}$. The second term $\frac{\partial g}{\partial z_f}$ is already discussed in Section 3.2, given in Eq. (21).

$$\frac{\partial \hat{\boldsymbol{f}}_{ext}}{\partial z_f} = \gamma_f \int_\eta \int_\xi \boldsymbol{N}^T \boldsymbol{n}^h \, d\xi d\eta \tag{35}$$

$$\frac{\partial g}{\partial \hat{\boldsymbol{u}}} = \left(\int_\eta \int_\xi \boldsymbol{N}^T \boldsymbol{n}^h \, d\xi d\eta \right)^T . \tag{36}$$

Substituting all the terms and eliminating Δz_f^{n+1} from Eqs. (33) and (34) we get,

$$\left(\boldsymbol{K}_{mem} - \boldsymbol{K}_l^{\Delta \boldsymbol{n}} - \boldsymbol{K}_l^{\Delta z} - \boldsymbol{K}_l^{\Delta z_f} \right) \Delta\hat{\boldsymbol{u}}^{n+1} = \hat{\boldsymbol{f}}_{ext} - \hat{\boldsymbol{f}}_{int} + \frac{\gamma_f \left(V_t - V_f^n \right)}{A_f} \int_\eta \int_\xi \boldsymbol{N}^T \boldsymbol{n}^h \, d\xi d\eta \tag{37}$$

$$\Delta z_f^{n+1} = \underbrace{\frac{\left(V_t - V_f^n \right)}{A_f}}_{\Delta z_{f,\,\Delta V}^{n+1}} - \underbrace{\frac{\left(\int_\eta \int_\xi \boldsymbol{N}^T \boldsymbol{n}^h \, d\xi d\eta \right)^T \Delta\hat{\boldsymbol{u}}^{n+1}}{A_f}}_{\Delta z_{f,\,\Delta \boldsymbol{u}}^{n+1}} . \tag{38}$$

Finally, this monolithic method has been written in Algorithm 3. Note that the main difference between this monolithic method and MVCIS is that there is no nested volume conservation iterations inside the structural solver but instead 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 (volume residual), while the second part, $\Delta z_{f,\,\Delta \boldsymbol{u}}$ takes into account the change in volume due the deformation of the wetted surface. If we carefully observe, the volume

residual also appears as an extra pressure,

$$p_{\Delta V} = \gamma_f \left(V_t - V_f\right) / A_f, \tag{39}$$

with the corresponding nodal force vector,

$$\hat{\boldsymbol{f}}_{\Delta V} = p_{\Delta V} \int_{\eta} \int_{\xi} \boldsymbol{N}^T \boldsymbol{n}^h \, d\xi d\eta, \tag{40}$$

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. Additionally, in order to prevent the free surface to move below the lowest point of wetted surface, the area of free surface is modified for the calculation of $\Delta z_{f,\,\Delta V}$ when this happens, while $\Delta z_{f,\,\Delta \boldsymbol{u}}$ is applied unchanged with the assumption that $\Delta z_{f,\,\Delta \boldsymbol{u}} > z_f^* - z_f$ always holds. The modification in $\Delta z_{f,\,\Delta V}$ is similar to the one discussed in Section 3.2, except here we also account for the movement of the free surface due to the deformation of wetted surface, $\Delta z_{f,\,\Delta \boldsymbol{u}}$. Moreover, we consider the updated structure for the volume calculation as we want the free surface to remain above the structure in the updated configuration. However, we cannot prevent the membrane surface to go above the free surface during N-R iterations and therefore if this happens the algorithm will fail. The same is true for MVCIS where this is more probable, as will be explained later in Section 8.2.

It is worth observing that in step 5 of Algorithm 2 and Algorithm 3, $\boldsymbol{K}_l^{\Delta z_f}$ is a rank one update in the linear system of equations, which results in a dense matrix. In case of multiple ponds at different locations, this matrix will result in rank r update where r is the number of ponds. When solved directly, this drastically increases the computing time of linear solvers.

Therefore, we use the Woodbury formula [26] to solve the equation in Step 4, which computes the inverse of a rank-r correction of a matrix by performing a rank-r correction to the inverse of the original matrix. For the special case where $r = 1$, the formula reduces to Sherman-Morrison formula [26].

Algorithm 3 Monolithic method with volume conservation as a constraint (MVCC)

1: $n = 0$
2: Find z_f^0 using Algorithm 1 with V_t as input argument
3: **while** $\left\| \hat{\boldsymbol{f}}_{ext} - \hat{\boldsymbol{f}}_{int} + \hat{\boldsymbol{f}}_{\Delta V} \right\| > \varepsilon$ and $n < n_{max}$ **do**
4: Update: $\boldsymbol{K}_{mem}$, $\boldsymbol{K}_l^{\Delta \boldsymbol{n}}$, $\boldsymbol{K}_l^{\Delta z}$, $\boldsymbol{K}_l^{\Delta z_f}$, $\hat{\boldsymbol{f}}_{ext}$, $\hat{\boldsymbol{f}}_{\Delta V}$ and $\hat{\boldsymbol{f}}_{int}$ using $\hat{\boldsymbol{u}}^n$ and z_f^n
5: Solve: $\left(\boldsymbol{K}_{mem} - \boldsymbol{K}_l^{\Delta \boldsymbol{n}} - \boldsymbol{K}_l^{\Delta z} - \boldsymbol{K}_l^{\Delta z_f} \right) \Delta \hat{\boldsymbol{u}}^{n+1} = \hat{\boldsymbol{f}}_{ext} - \hat{\boldsymbol{f}}_{int} + \hat{\boldsymbol{f}}_{\Delta V}$
6: Update displacement: $\hat{\boldsymbol{u}}^{n+1} = \hat{\boldsymbol{u}}^n + \Delta \hat{\boldsymbol{u}}^{n+1}$
7: Update structure: $\hat{\boldsymbol{x}}^{n+1} = \hat{\boldsymbol{X}} + \hat{\boldsymbol{u}}^{n+1}$
8: Calculate Δz_f^{n+1} using Eq. (38)
9: **if** $\Delta z_f^{n+1} < z_f^* - z_f$ **then**
10: $A_f = V_f^{n+1} / (z_f^n - z_f^* + \Delta z_{f,\, \Delta \boldsymbol{u}}^{n+1})$
11: $\Delta z_{f,\, \Delta V}^{n+1} = (V_t - V_f^{n+1}) / A_f$
12: $\Delta z_f^{n+1} = \Delta z_{f,\, \Delta V}^{n+1} + \Delta z_{f,\, \Delta \boldsymbol{u}}^{n+1}$
13: **end if**
14: $z_f^{n+1} = z_f^n + \Delta z_f^{n+1}$
15: $n = n + 1$
16: **end while**

6. Partitioned methods for ponding analysis

This section discusses two partitioned methods to calculate the static deformation of a membrane structure due to ponding. In this section, the accent $\hat{\bullet}$ which was used in the previous sections to denote the nodal vectors, will be omitted to avoid multiple accents on a symbol. As a result, in the discussion that follows all the nodal vectors are denoted by small letter bold characters.

6.1. Implicit partitioned coupling method (IPC)

In the partitioned approach the problem of finding the static deformation under ponding load is formulated as a fixed-point problem, where the structural solver and the volume-conserving solver are called sequentially. The structural solver takes the nodal vertical distance vector $\boldsymbol{d}$ $(= z - z_f)$ from the free surface as input to solve for the nodal displacement vector $\boldsymbol{u}$ from the resultant hydrostatic pressure on the membrane surface. The volume-conserving solver on the other hand determines the free surface position based on the new structural deformation. This process is continued till the norm of the fixed point residual, discussed below, is lower than a certain pre-defined tolerance. Sometimes, the fixed point iteration implemented in this manner may diverge or the convergence rate can be very slow. Therefore, convergence accelerators such as Aitken relaxation [16, 27] and IQN-ILS [17] are used to achieve faster convergence. Mathematically, if we denote volume-conserving solver as an operator $\mathcal{F}$ and the structural solver for membrane surface as $\mathcal{S}$, then we can write,

$$\boldsymbol{d} = \mathcal{F}(\boldsymbol{u})$$
$$\boldsymbol{u} = \mathcal{S}(\boldsymbol{d})$$

The problem of finding an equilibrium shape of the structure under the hydrostatic load of a fixed volume of fluid can be written as a fixed point problem

$$\boldsymbol{u} = \mathcal{S} \circ \mathcal{F}(\boldsymbol{u}). \tag{41}$$

If k represents the iteration number for the coupling iterations, then the residual of Eq. (41) (fixed point residual) at the k^{th} iteration is given by Eq. (42), where $\boldsymbol{u}^k$ is the displacement at the k^{th} iteration and $\tilde{\mathbf{u}}^{k+1} = \mathcal{S} \circ \mathcal{F}(\boldsymbol{u}^k)$.

$$\mathbf{r}^k = \tilde{\boldsymbol{u}}^{k+1} - \boldsymbol{u}^k \tag{42}$$

With all these definitions, we can write the implicit partitioned coupling method for the problem in Algorithm 4. Note that in each coupling iteration, the structural solver receives the fluid loading as pressure fields and therefore only $\boldsymbol{K}_l^{\Delta \boldsymbol{n}}$ is used in the non-linear iterations of the structural solver, which is generally implemented. Compared to the monolithic methods discussed before, the main advantage of this method apart from being modular is that it is more robust as we are doing volume conservation on constant pressure equilibrium shapes of the structure. Therefore, it is less likely that the structural displacement between the coupling iterations will be such that the wetted surface moves above the free surface causing the algorithm to fail.

Algorithm 4 Implicit partitioned coupling (IPC) method

1: $k = 0$
2: $\tilde{\boldsymbol{u}}^1 = \mathcal{S} \circ \mathcal{F}(\boldsymbol{u}^0)$
3: $\mathbf{r}^0 = \tilde{\boldsymbol{u}}^1 - \boldsymbol{u}^0$
4: **while** $\left\|\mathbf{r}^k\right\| > \varepsilon$ and $k < k_{max}$ **do**
5: Calculate $\boldsymbol{u}^{k+1}$ using convergence accelerator [17, 16, 27]
6: $k = k + 1$
7: $\tilde{\boldsymbol{u}}^{k+1} = \mathcal{S} \circ \mathcal{F}(\boldsymbol{u}^k)$
8: $\mathbf{r}^k = \tilde{\boldsymbol{u}}^{k+1} - \boldsymbol{u}^k$
9: **end while**

6.2. Implicit partitioned coupling method with fluid load linearization (IPCFL)

During the numerical experiments, it was observed that the IPC method can require a large number of coupling iterations for convergence depending on the structural and fluid properties. Therefore a modification of the above algorithm was made where the modularity was sacrificed for increased convergence speed. The main idea behind the modification is that if we include linearized behavior of the fluid in the structural solver, the structure can anticipate the free surface movement and the resulting pressure fields. Consequently, convergence speed of the coupling iterations will be improved. To that end, the linearization of the fluid loading discussed in Section 4 is added in every structural N-R iteration as load stiffness matrices ($\boldsymbol{K}_l^{\Delta \boldsymbol{n}} + \boldsymbol{K}_l^{\Delta z} + \boldsymbol{K}_l^{\Delta z_f}$) along with the linear update of the free surface, Eq. (27). As a result, during the structural iterations the free surface is updated to conserve the volume between the structural iterations but because it is a linear update, the volume is not maintained as V_t. Nevertheless,

this is corrected afterwards by the volume-conserving solver (Algorithm 1) in the next coupling iteration. This is repeated till the norm of the fixed point residual, discussed in Section 6.1, is below a certain tolerance. It was found that in all the numerical experiments, the number of coupling iterations required for convergence was much lower and hence there was no need for the convergence accelerator. Finally, we can write the IPCFL method for ponding analysis in Algorithm 5, where the modified structural solver with the linearized fluid loading is denoted as $\mathcal{S} \oplus \mathcal{L}(\mathcal{F})$, which is given by Algorithm 6.

Algorithm 5 Partitioned FSI iterations to calculate structural deformation under ponding load.

$k = 0$
$\tilde{\boldsymbol{u}}^1 = (\mathcal{S} \oplus \mathcal{L}(\mathcal{F})) \circ \mathcal{F}(\boldsymbol{u}^0)$
$\mathbf{r}^0 = \tilde{\boldsymbol{u}}^1 - \boldsymbol{u}^0$
while $\left\|\mathbf{r}^k\right\| > \varepsilon$ and $k < k_{max}$ **do**
    $\boldsymbol{u}^{k+1} = \tilde{\boldsymbol{u}}^{k+1}$
    $k = k + 1$
    $\tilde{\boldsymbol{u}}^{k+1} = (\mathcal{S} \oplus \mathcal{L}(\mathcal{F})) \circ \mathcal{F}(\boldsymbol{u}^k)$
    $\mathbf{r}^k = \tilde{\boldsymbol{u}}^{k+1} - \boldsymbol{u}^k$
end while

Algorithm 6 The modified structural solver $\mathcal{S} \oplus \mathcal{L}(\mathcal{F})$ with linearized fluid load

$n = 0$
while $\left\| \hat{\boldsymbol{f}}_{ext} - \hat{\boldsymbol{f}}_{int} \right\| > \varepsilon$ and $n < n_{max}$ **do**
    Update: $\boldsymbol{K}_{mem}$, $\boldsymbol{K}_l^{\Delta \boldsymbol{n}}$, $\boldsymbol{K}_l^{\Delta z}$, $\boldsymbol{K}_l^{\Delta z_f}$, $\hat{\boldsymbol{f}}_{ext}$ and $\hat{\boldsymbol{f}}_{int}$ using $\hat{\boldsymbol{u}}^n$ and z_f^n
    Solve $\left(\boldsymbol{K}_{mem} - \boldsymbol{K}_l^{\Delta \boldsymbol{n}} - \boldsymbol{K}_l^{\Delta z} - \boldsymbol{K}_l^{\Delta z_f} \right) \Delta \hat{\boldsymbol{u}}^{n+1} = \hat{\boldsymbol{f}}_{ext} - \hat{\boldsymbol{f}}_{int}$
    Update displacement: $\boldsymbol{u}^{n+1} = \boldsymbol{u}^n + \Delta \boldsymbol{u}^{n+1}$
    Update structure: $\boldsymbol{x}^{n+1} = \boldsymbol{X} + \boldsymbol{u}^{n+1}$
    Calculate Δz_f^{n+1} using Eq. (27)
    $z_f^{n+1} = z_f^n + \Delta z_f^{n+1}$
    $n = n + 1$
end while

7. Integration on the wetted surface

If we look back at the discussion on various methods for ponding analysis, one common aspect in all the methods is the integration of quantities on the wetted surface $\partial \Omega_{fs}$, be it the calculation of volume, the nodal force vectors or the load stiffness matrices. In the finite element framework, this is commonly performed by numerical integration based on Gauss quadrature [28]. For a special case when the integration of $f(\xi, \eta)$ is sought on a 2D-surface $\partial \Omega$, where the surface is parameterized by the parameters $\xi \in [-1, 1]$ and $\eta \in [-1, 1]$, the integration is written as the weighted sum of the function values at $n \times n$ unevenly distributed points in the parametric space $\xi - \eta$, as given below:

$$\int_{\partial\Omega} f(\xi,\eta)\, dS = \int_{\xi}\int_{\eta} f(\xi,\eta)\|\boldsymbol{g}_\xi \times \boldsymbol{g}_\eta\|\, d\xi d\eta$$
$$= \sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij} f(\xi_i,\eta_j)\|\boldsymbol{g}_\xi \times \boldsymbol{g}_\eta\| \tag{43}$$

where, the $\boldsymbol{g}_\xi$ and $\boldsymbol{g}_\eta$ are the base vectors, discussed in Section 4, and w_{ij} is the weight corresponding to the parametric coordinates (ξ_i, η_j). For more discussion on these special points in the parametric domain and their corresponding weights, the interested readers can refer to [28]. Typically, the integration surface is discretized using a suitable elements based on the application. The numerical integration is straightforward for a surface which is discretized using elements conforming to the surface boundary, where the integration is performed by Gauss quadrature in every element and the contributions from all elements are added to get the required global quantity. However, in our case often the integration domain $\partial\Omega_{fs}$ will be non-conforming to the structural elements during the solution process as the free surface plane can move independent of the membrane discretization. One solution is to remesh or displace the mesh every time the free surface moves to make it conforming. However, this is not practical and would have a detrimental effect on the computation time as it has to be performed in every volume-conserving iteration. An alternative is to perform integration on the wetted surface by subdividing the elements that are cut by the free surface before performing Gauss quadrature. Note that we are not adding new elements or nodes during this process; the subdivision is only performed to carry out integration accurately. The process is clearly shown in Fig. 6, where we consider only linear triangle elements, which can be of course extended to other elements, like the one dis-

cussed in [11, 29]. As our focus was more on different algorithms for ponding analysis than developing subdividing procedures for performing integration on different elements, we restricted ourselves to the linear triangular case. In Fig. 6, we clearly see that the there are four different possibilities. The first case is when the elements lie above the free surface i.e the $elem \notin \partial\Omega^h_{fs}$, where the integration is 0. The second and third case shows different possibilities of triangle elements cut by the free surface and their corresponding subdivisions. In these scenarios, integration is performed only in the parts below the free surface i.e. the shaded region. In terms of implementation, the Gauss points in the subdivided shaded triangles are used in Eq. (43). Finally, the last case is when the elements are below the free-surface, where the usual Gauss quadrature procedure is followed.

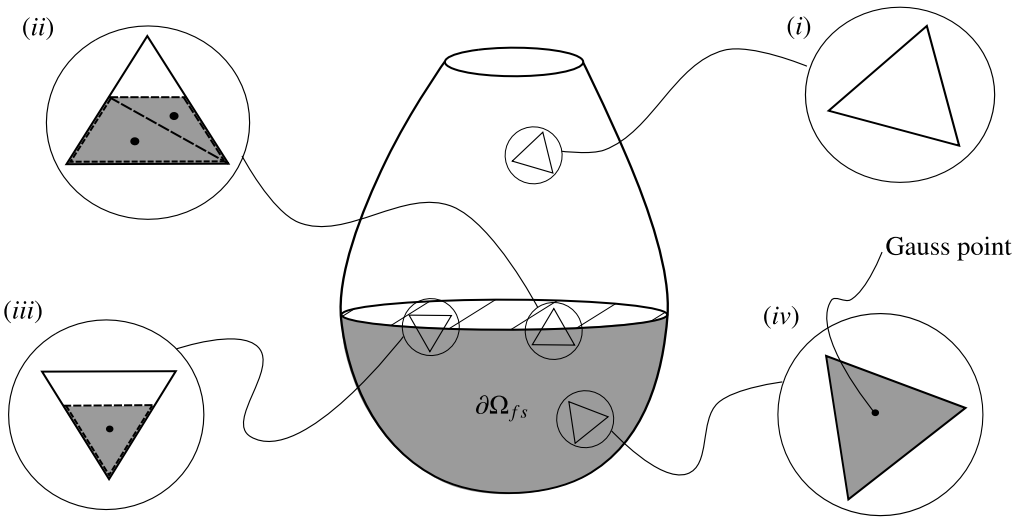


Figure 6: Integration on the wetted surface when discretized with linear triangle elements, where the subdivided triangles for integration are shown in dashed lines.

8. Numerical Examples

The discussed methods for ponding analysis were developed in an open-source finite element code called KRATOS. In this section we present three numerical examples: (i) ponding on a circular membrane [13],(ii) Ponding on an inflated membrane structure and (iii) Ponding on a square membrane. The first example is an academic case which is used for validating the implemented methods discussed in this paper. Subsequently, the different methods are compared for computation speed and robustness. The second example is used as an application case where the ponding analysis is performed on a real membrane structure. The objective of this example is to demonstrate how the ponding analysis can be used to calculate deformation of membrane structures due to ponding and also to show the strengths and weaknesses of the discussed methods. Finally, the third example is used to demonstrate the application of the discussed algorithms for non-axisymmetric cases.

Before going to the numerical examples of ponding, we present the validation of volume calculation algorithm and volume-conserving solver which are the main components of all the methods. To that end, we take a hollow sphere of radius $R = 1.0\ m$ cut at a height of $0.75\ m$ above its center. The volume enclosed by the cut sphere and a flat free surface located at the top-most point is given by $V_{cutSphere} = 245/192\pi = 4.0088\ m^3$, which was also found using the volume calculation algorithm, thus verifying the algorithm. To check the volume conserving solver at some intermediate position. The volume enclosed by a horizontal plane, $0.5\ m$ below the center of the sphere and the sphere surface is given by, $V = 5\pi/24 = 0.6545\ m^3$. The volume-conserving solver is run with a target volume $V_t = 0.6545 m^3$ and initial free

surface position coinciding with the center of the sphere. The solver was run with 1632, 6700 and 27016 linear triangle elements, respectively. The results are summarized in Table 1. It clearly shows that the volume-conserving solver is able to determine the plane's position accurately, limited by the discretization error in the surface.

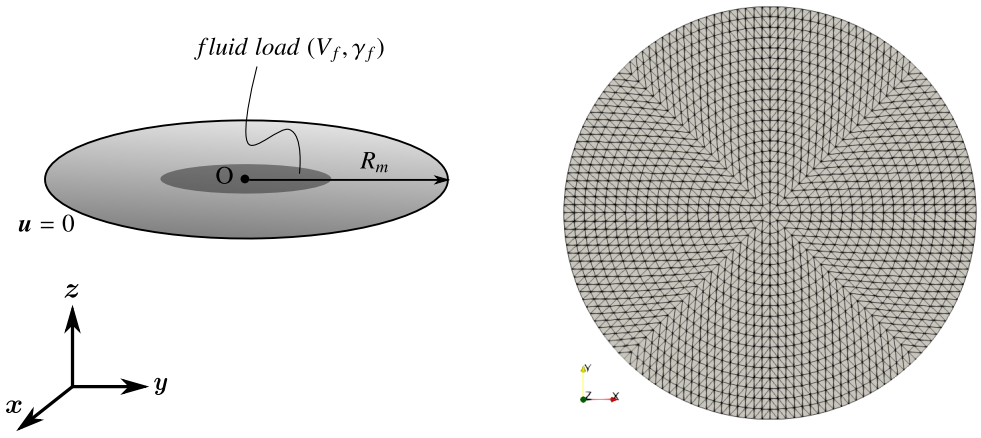
Table 1: Free surface position calculated by the volume-conserving solver with $V_t = 0.6545\ m^3$.

# Elements	$z_{comp}\ (m)$
1632	-0.4984
6700	-0.4996
27016	-0.4999

8.1. Ponding on a circular membrane

In the first numerical example we consider a horizontal circular membrane of radius $R_m = 10\ mm$ and uniform thickness $t_m = 0.01\ mm$ at $z_m = 0$. The membrane is modeled as incompressible Mooney-Rivlin plane-stress material with material constants $c_1 = 1.92 \times 10^5\ Pa$ and $c_2 = 1.92 \times 10^4\ Pa$, corresponding to the shear modulus $\mu = 0.4225\ MPa$ and $k = c_2/c_1 = 0.1$, discussed in [13]. During the simulation all the boundary nodes are fixed and the membrane is filled with fluid of density $\rho = 10^{-5}\ kg/mm^3$ (10 times that of water) in fluid volume increments of $\Delta V_f = 200mm^3$. The acceleration due to gravity is assumed to be $g = 10\ m/s^2$, along the negative z-direction. The problem set up of the case is clearly shown in Fig. 7a. For all simulations, we consider a mesh of $n_{el} = 3200$ linear triangle membrane elements (Fig. 7b),

which is more than the number of elements considered in [13]. It was observed during the simulation that when the simulations were initialized with a flat circular sheet, none of the methods was able to converge even with some prestress to provide some stiffness at the first N-R iteration. Therefore, in the first step of the simulation a pressure of magnitude $p_{in} = 500Pa$ was applied to get a good initial geometry, and then the methods were run on the deformed geometry with the pressure value set to zero. This is in agreement with what would happen in the real scenario where the ponding process is preceded by a seeding event for flat geometries.



(a) (b)

Figure 7: Ponding on a circular membrane: a) problem set up, b) meshed geometry (viewed from top).

In order to validate all the methods, we plot volume of the fluid V_f vs central deflection $|u_{z,O}|$ and compare our results with the reference. Figs. 8a-8d show the relationship between the volume of fluid and the central deflection when computed using partitioned and monolithic methods. The relation

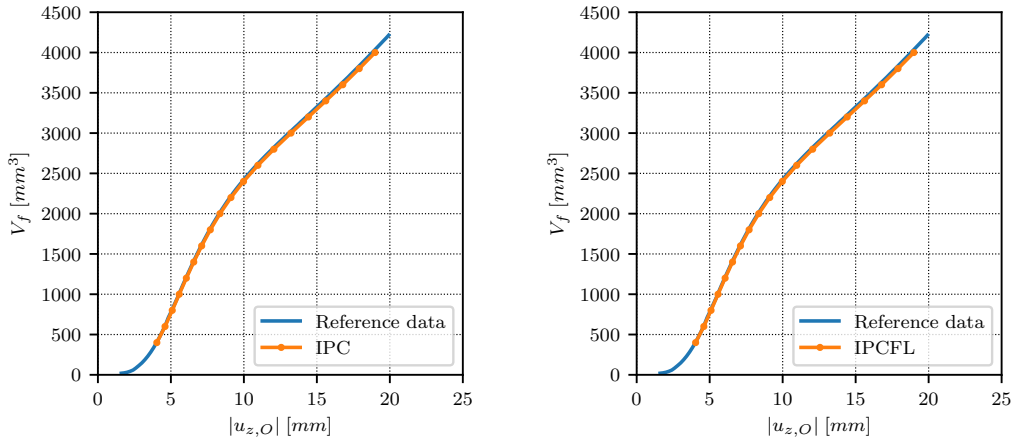
is clearly non-linear. The corresponding membrane deformations for three different fluid volumes are shown in Fig. 9. It should be noted that when $z_f > 0$, which is the case when the free surface plane goes above the membrane boundary (Fig. 9b), it is assumed that a vertical cylinder of radius $R = R_m$ is placed above the boundary. Clearly, the results are in good agreement with the data from the reference. The pressure distribution was also checked if it was linearly varying with the vertical distance below the free surface. The pressure distribution and the maximum principal stresses on the membrane surface are plotted in Figs. 10 and 11 corresponding to the fluid volumes in Fig. 9.

Next, we compare the partitioned approaches in terms of number of iterations and computing time. The IPC method discussed in Section 6.1 can have multiple variants based on the convergence accelerator that is being used. In the current work we use three different convergence accelerators: Gauss-Seidel with constant relaxation, Aitken relaxation [16, 27] and IQN-ILS [17]. Among partitioned methods Gauss-Seidel with a constant relaxation performed worst; it also diverged at $V_f = 3000\ mm^3$ and therefore the results after that step are absent in Fig. 12 and 13. The initial relaxation factor, $\alpha_{in} = 0.6$ for Aitken and IQN-ILS convergence accelerators was chosen based on numerical experiments with different values of α_{in} and the maximum value that achieved convergence for all the fluid volumes was chosen for the simulation. The same value was used as the constant relaxation factor α for Gauss-Seidel iterations. As evident from Fig. 12 and 13 the IPCFL has the fastest convergence among all the different partitioned methods, which is closely followed by IPC with IQN-ILS as convergence accelerator. The

performance in terms of computing time of IPCFL shown in Fig. 13 can be further improved by storing the factorization for the rank one update, where the inverse of the system matrix is required twice in every structural N-R iteration. However, this was not possible in the framework where the code was implemented. Moreover, one can also choose iterative linear solvers for computation where the factorization is not applicable.

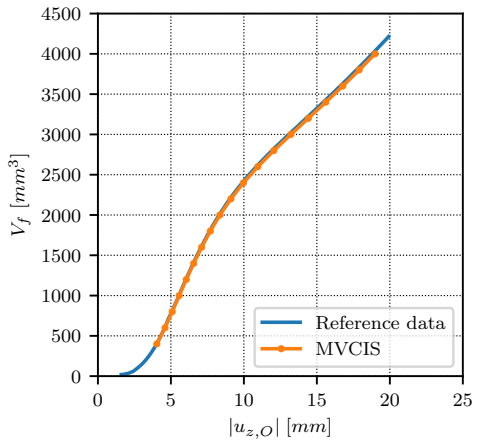
Now we look at the monolithic methods where we first plot the convergence characteristics of the two methods at different load steps (n_{step}), see Fig. 14. As seen in this figure, the two methods have quadratic convergence near the root. The effect of follower load stiffness matrices can be also observed in Fig. 15, where there is a clear improvement in the convergence characteristics when the load stiffness matrices $\boldsymbol{K}_l^{\Delta z}$ and $\boldsymbol{K}_l^{\Delta z_f}$ are added in the tangent stiffness matrix used in the N-R iteration of the structural solver. Note that $\boldsymbol{K}_l^{\Delta \boldsymbol{n}}$ is always included in the tangent stiffness matrix irrespective of the methods, as discussed before; therefore, its effect is not shown in the figure. Finally, comparing the monolithic methods and partitioned methods, we see that the two monolithic methods have similar performance and they are superior compared to partitioned methods in terms of computation time, as shown in Fig. 16.

[1]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.

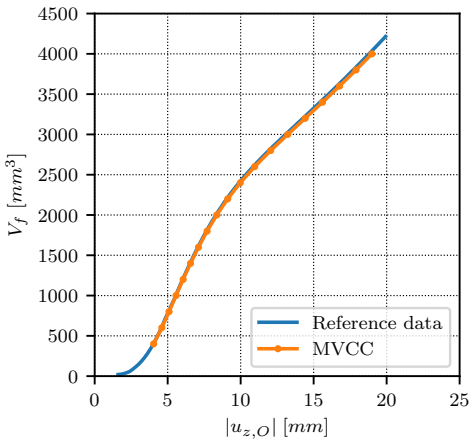


(a)

(b)

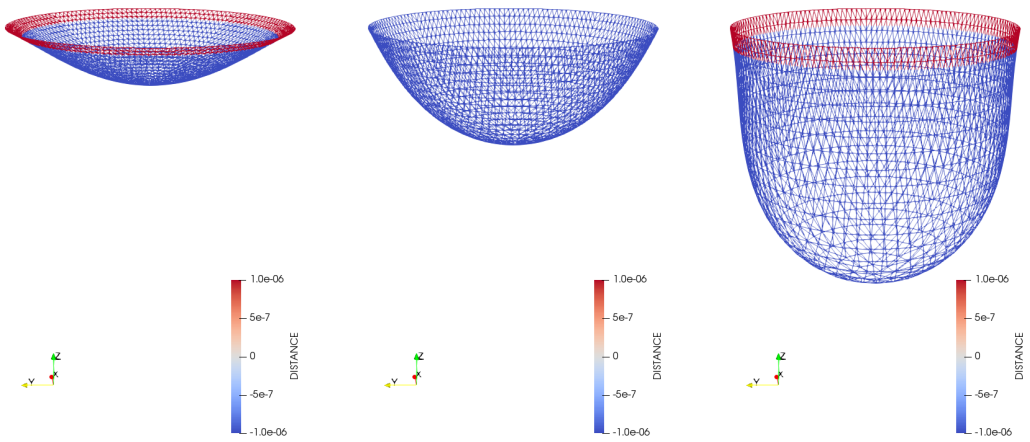


(c)



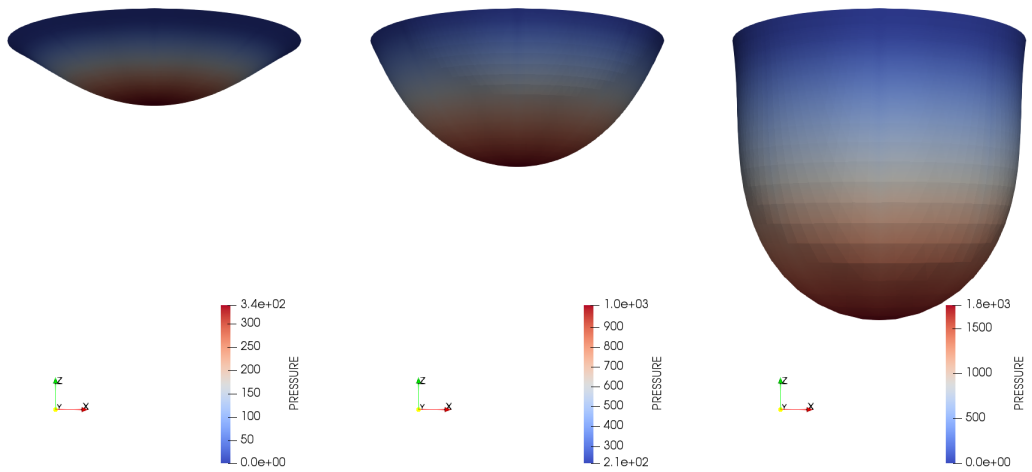
(d)

Figure 8: Comparison of fluid volume, V_f vs central deflection, $|u_{z,O}|$ for all the four methods, where the reference data is from [13]: a) IPC, b) IPCFL c) MVCIS, d) MVCC.



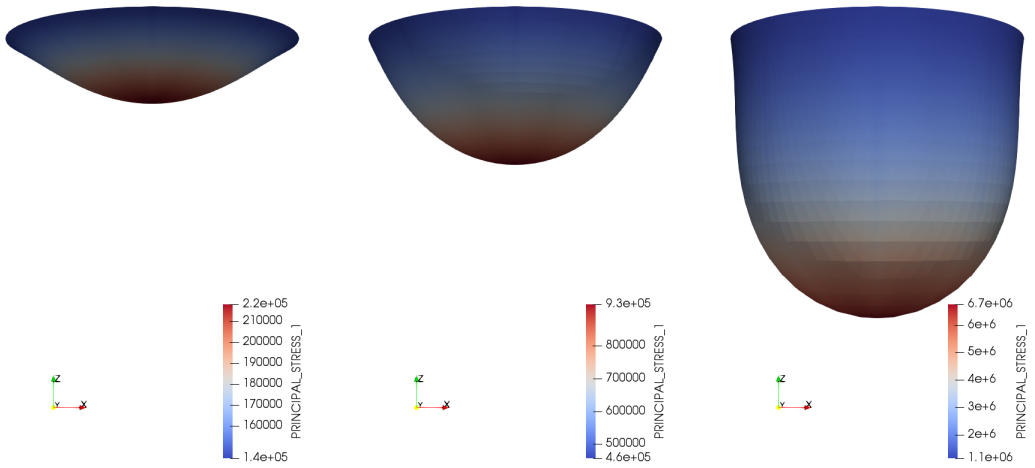
(a) (b) (c)

Figure 9: Deformation of circular membrane due to hydrostatic loading from different volumes of fluid with $\rho = 10^{-5}\ kg/mm^3$, where the blue color on the surface indicates negative vertical distance from the free surface, and therefore represents the wetted region.[1]: a) $V_f = 400\ mm^3$, b) $V_f = 2000\ mm^3$ (when the free surface surface is above the membrane boundary), c) $V_f = 4000\ mm^3$.



(a) (b) (c)

Figure 10: Hydrostatic pressure distribution on the circular membrane due to hydrostatic loading from different volumes of fluid with $\rho = 10^{-5}\ kg/mm^3$: a) $V_f = 400\ mm^3$, b) $V_f = 2000\ mm^3$ (when the free surface surface is above the membrane boundary), c) $V_f = 4000\ mm^3$.



(a) (b) (c)

Figure 11: Maximum principal stress distribution on the circular membrane due to hydrostatic loading from different volumes of fluid with $\rho = 10^{-5}\ kg/mm^3$: a) $V_f = 400\ mm^3$, b) $V_f = 2000\ mm^3$ (when the free surface surface is above the membrane boundary), c) $V_f = 4000\ mm^3$.

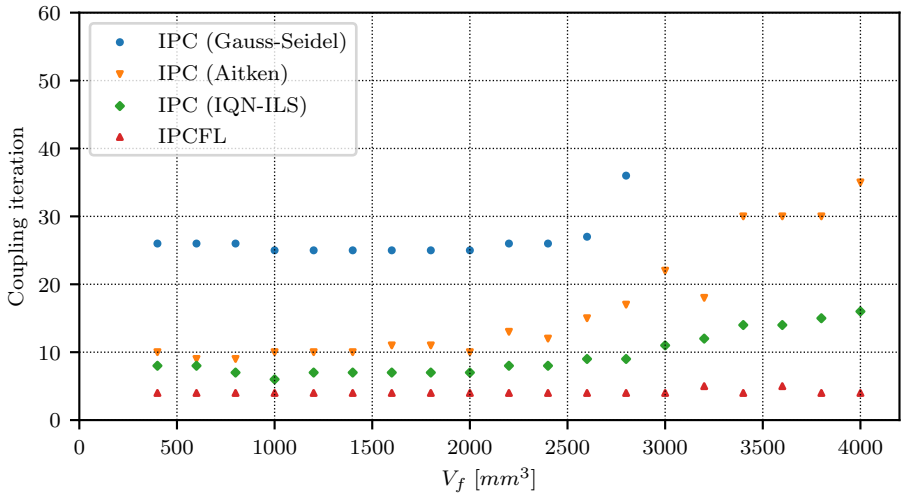


Figure 12: Comparison of number of coupling iterations for different fluid volumes (V_f) for partitioned methods i.e. IPC (Gauss-Seidel) with constant relaxation $\alpha = 0.6$, IPC (Aitken) with initial relaxation $\alpha_{in} = 0.6$, IPC (IQN-ILS) with initial relaxation $\alpha_{in} = 0.6$ and IPCFL.

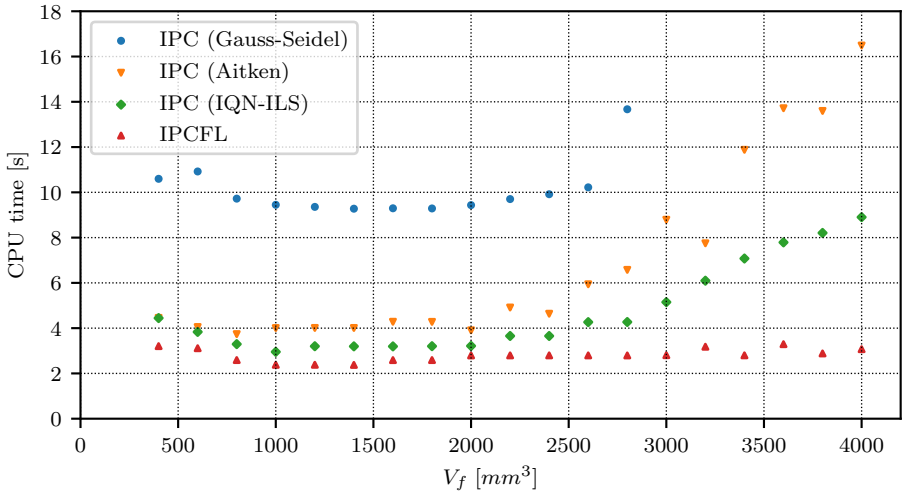


Figure 13: Comparison of CPU time for different fluid volumes (V_f) for partitioned methods i.e. IPC (Gauss-Seidel) with constant relaxation $\alpha = 0.6$, IPC (Aitken) with initial relaxation $\alpha_{in} = 0.6$, IPC (IQN-ILS) with initial relaxation $\alpha_{in} = 0.6$ and IPCFL.

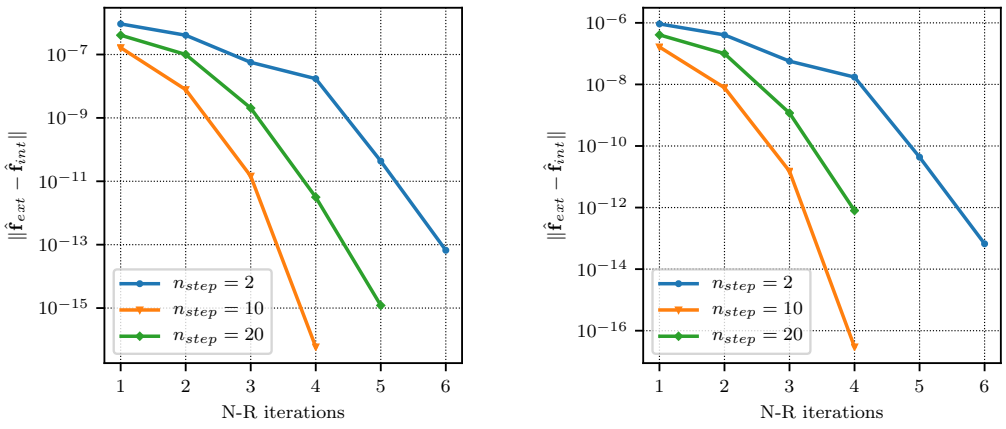


Figure 14: Convergence plot for monolithic methods at $n_{step} = 2$ ($V_f = 400\ mm^3$), $n_{step} = 10$ ($V_f = 2000\ mm^3$) and $n_{step} = 20$ ($V_f = 4000\ mm^3$): a) MVCIS, b) MVCC.

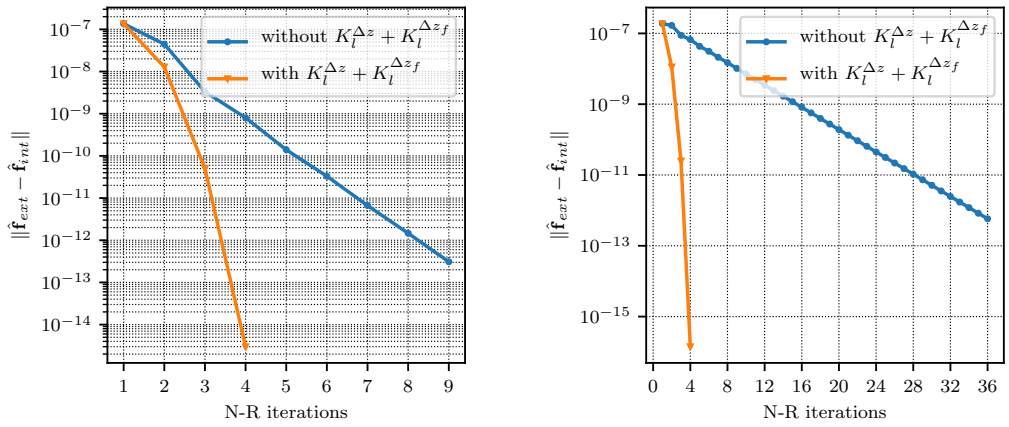


Figure 15: Effect of addition and removal of $\boldsymbol{K}_l^{\Delta z}$ and $\boldsymbol{K}_l^{\Delta z_f}$ on convergence speed: a) $n_{step} = 6$ ($V_f = 1200\ mm^3$), b) $n_{step} = 12$ ($V_f = 2400\ mm^3$).

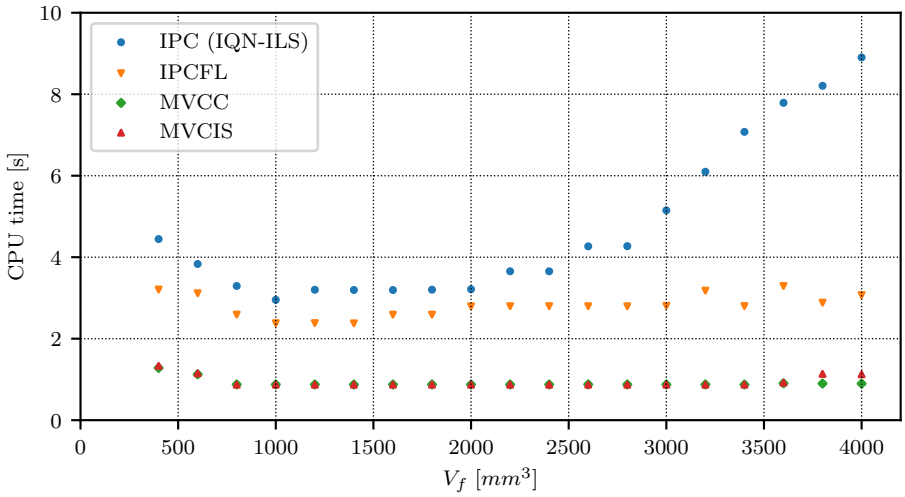


Figure 16: Comparison of CPU time for different fluid volumes (V_f) for IPC (IQN-ILS) with initial relaxation $\alpha_{in} = 0.6$, IPCFL, MVCIS, MVCC.

8.2. Ponding on an inflated membrane structure

In the next numerical example, we consider an application case where the ponding analysis is applied on an inflated thin-walled membrane hemisphere. The material is modeled as Saint-Venant Kirchhoff plane stress material with material properties: Young's modulus $E = 7 \times 10^6\ N/m^2$, Poisson ratio $\nu = 0.45$ and thickness $t = 0.002\ m$. A difference in pressure of $p = 0.5\ kPa$ with respect to the atmospheric pressure is applied at the internal surface. To get the perfect hemisphere of diameter $D = 20\ m$ after the application of this pressure, an isotropic normal pre-stress calculated from the formula $\sigma_{mem} = pD/4t = 1.25MPa$ is applied on the membrane with zero shear stress. The hemisphere is clamped at the bottom boundary, and for reducing the computational time only a quarter section of the hemisphere is simulated considering the symmetry of the problem. The acceleration due to gravity is assumed to be $g = 9.8\ m/s^2$, along the negative z-direction. To start the ponding process, we first apply a dead load of $w = 1\ kPa$ on the top surface of the membrane enclosed by a circle of radius $R_{dead} = 1.736\ m$ (corresponding to a 10° sector) as a seed event, which is applied throughout the simulation. Due to the dead load, there will be a local depression in the hemisphere. In the created depression, water is added in volume increment steps and the resulting deformation is obtained by the four methods. In all the simulations considered in the example, the volume-conserving solver is initialized from the topmost point of the undeformed hemisphere ($z_f = 0$) and similar to the previous example if the free surface does not intersect the membrane surface then a cylinder of radius equal to R_{dead} is assumed above the surface. The discussed boundary and load conditions are clearly shown in Fig. 17a and

the discretized quarter model used for the simulation is shown in Fig. 17b, where the symmetry boundary conditions are applied on the nodes at x-z and y-z plane. For comparing different methods, only IPC with IQN-ILS is presented among different variants of IPC because the comparison between the different convergence accelerators is already discussed in Section 8.1.

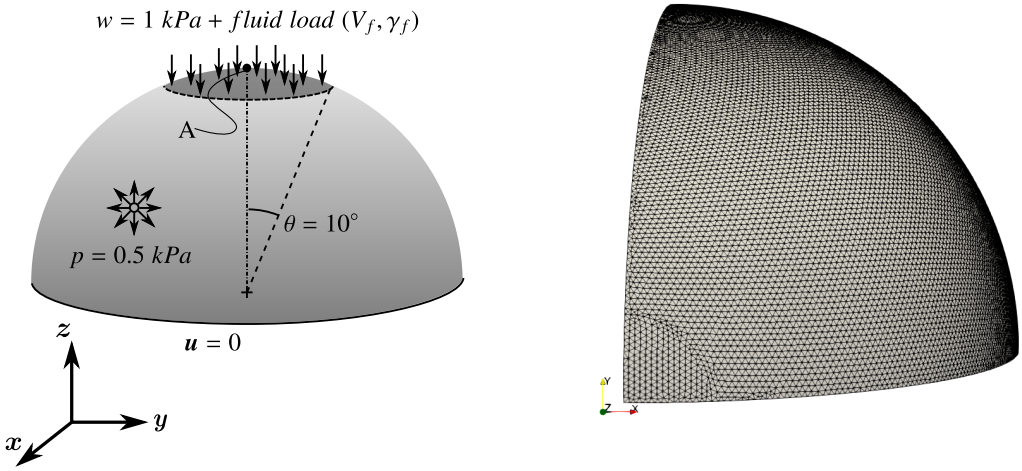


Figure 17: Ponding on an inflated hemisphere: a) problem set up, b) quarter model of the hemisphere discretized with 19830 linear triangle elements (viewed from top)

In this numerical example, apart from looking at the deformation results and computing time, we will be also evaluating the robustness of the methods by running with increasing value of ΔV_f. First, we consider the results with volume increment steps of $\Delta V_f = 0.2m^3$ where all the methods converged to the solution. Clearly, the results in Fig. 18a are consistent with the previous numerical example: the monolithic methods are superior in terms of computing time compared to the partitioned methods, and among the partitioned

methods overall IPCFL is better than IPC (IQN-ILS). However, only at the first volume increment step we see a deviation in computing time for IPCFL. This deviation is more obvious in Fig. 18b, where fluid volume $V_f = 0.8\ m^3$ is applied in the first step[2]. The IPCFL method in this step takes far more time to converge than all the other methods. However, if we look at the number of iterations, it takes 50% fewer iterations than IPC (IQNILS). Therefore, the problem must be in the $\mathcal{S} \oplus \mathcal{L}(\mathcal{F})$ solver. Fig. 19a shows the evolution of the fluid volume with the $\mathcal{S} \oplus \mathcal{L}(\mathcal{F})$ N-R steps, where we can clearly see that the fluid volume jumps to a value higher than the target fluid volume when $\Delta V_f = 0.2m^3$ in the first coupling iteration. This is because the solver conserves volume based on the linearized change in volume from the membrane movement in the N-R iteration. When the structural movement is small, the difference between the actual change in volume and the linearized change in volume is also small, which happens after some N-R iterations and therefore the fluid volume can be seen constant through the later N-R iterations. As explained in Section 6.2, the resulting error in the volume is corrected in the next coupling iterations, which can be observed in Figs. 19a and 19b. This behavior of the solver at the first fluid load step has more pronounced negative effect when $\Delta V_f = 0.4\ m^3$, as seen in Fig. 19b; for larger ΔV_f it even diverges, and therefore Fig. 18c has no data from IPCFL.

Among monolithic methods, if MVCC and MVCIS converge they have

[2]In all the simulations, at the first volume increment step (second load step of the simulation), $V_f = 2\ \Delta V_f$ because the fluid volume load is modeled as $V_f = n_{step}\ \Delta V_f$, where n_{step} is the load step and the value of n_{step} is set to zero during the application of the seeding load.

almost same computing time. However, MVCC is observed to be more robust than MVCIS because for $\Delta V_f = 0.4\ m^3$ and $0.8\ m^3$ the MVCIS failed. Therfore, its data is absent in Figs. 18b and 18c. The reason why MVCIS failed in these cases is because during one of the N-R iterations the structure moved above the free surface and as a result the volume-conserving algorithm failed. This scenario although possible in MVCC is however less likely because the effect of free surface movement due to volume difference and the volume-conserving behavior are included as an additional pressure ($p_{\Delta V}$) and load stiffness matrix $\boldsymbol{K}_l^{\Delta z_f}$, respectively inside the structural solver. At the same time, the effect of structural movement on the free surface position is also added as $\Delta z_{f,\Delta \boldsymbol{u}}$. This intricate coupling between the two parts of the solver, which is absent in MVCIS makes this scenario less probable in MVCC. IPC (IQN-ILS) on the other hand was found to be very robust but was comparatively slower than monolithic methods because of the coupling iterations. We think the robustness of the methods is due to two reasons: use of IQN-ILS algorithm, which approximates the inverse Jacobian of the fixed point residual and application of the volume-conserving algorithm on the constant pressure solution of the structure which is less likely to change abruptly between consecutive coupling iterations. In contrast, MVCIS uses the volume-conserving algorithm in N-R iterations which can assume any arbitrary non-equilibrium shape between consecutive iterations.

The deformation of the inflated hemispherical membrane under ponding loads can be viewed in Fig. 20, where the deformed shapes under two different volumes of water are shown. The pressure distribution and the maximum principal stresses on the membrane surface are plotted in Figs. 21 and 22 cor-

responding to the fluid volumes in Fig. 20. Finally, the relation between the magnitude of the vertical deflection of the top most point of the membrane, A(0, 0, 10) and the free surface height from the ground is plotted in Fig. 23a and its variation with the fluid volume is shown in Fig. 23b. The relationship unlike previous example appears to be linear in this fluid volume range.

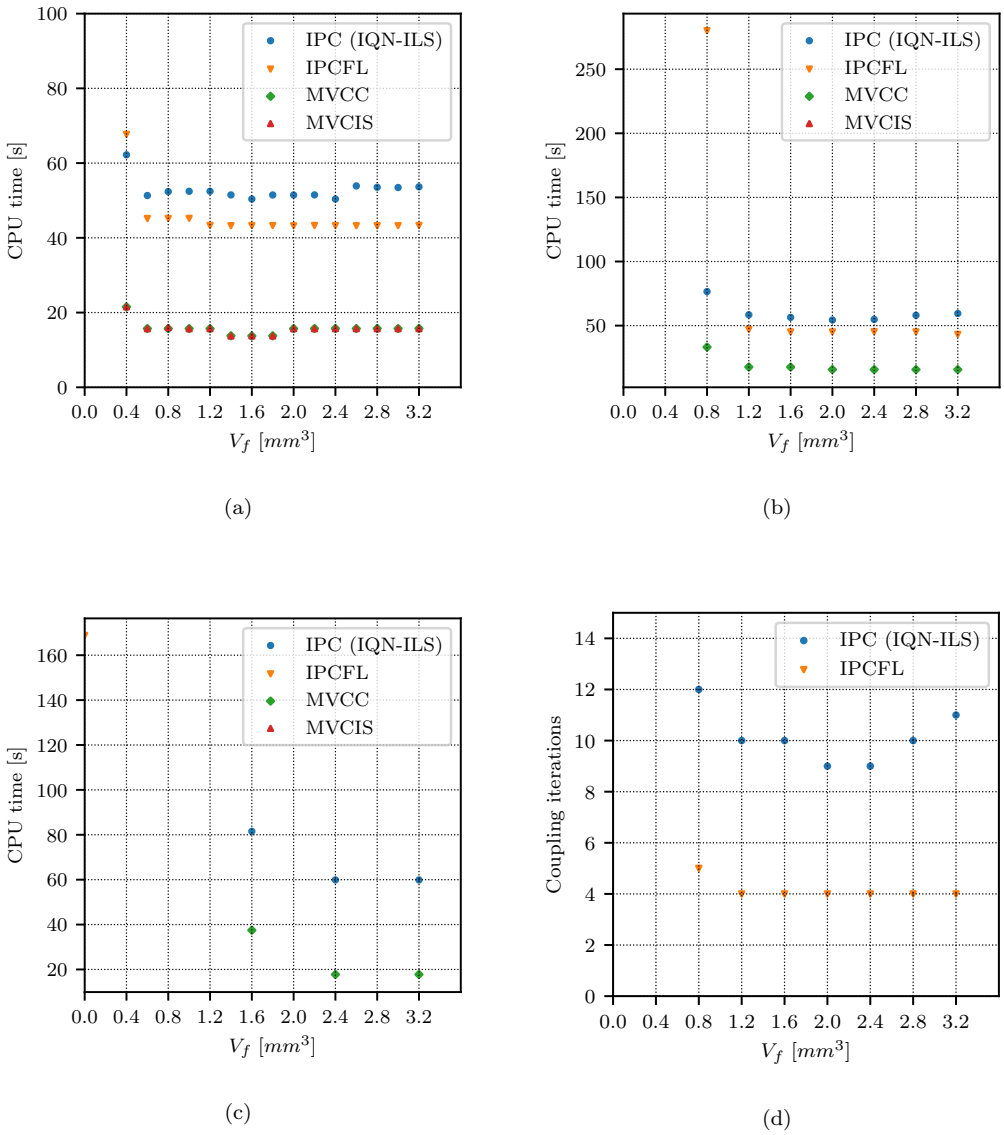
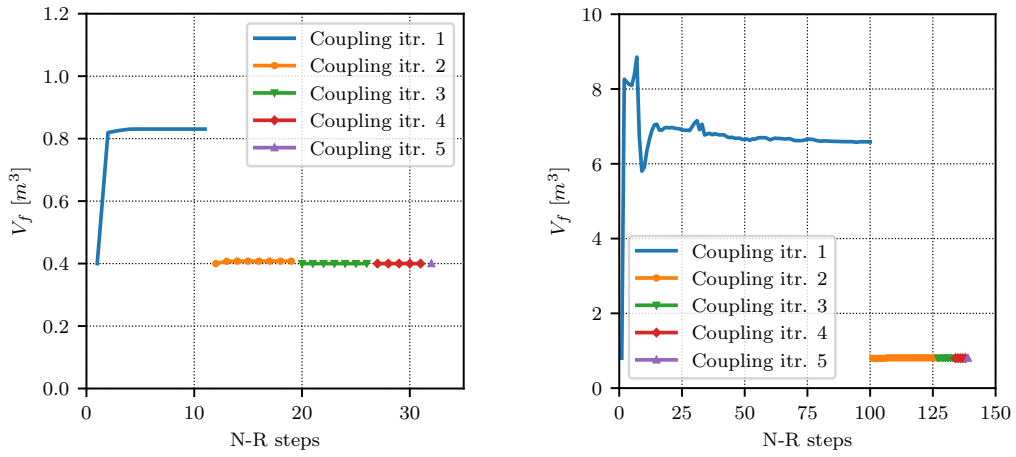


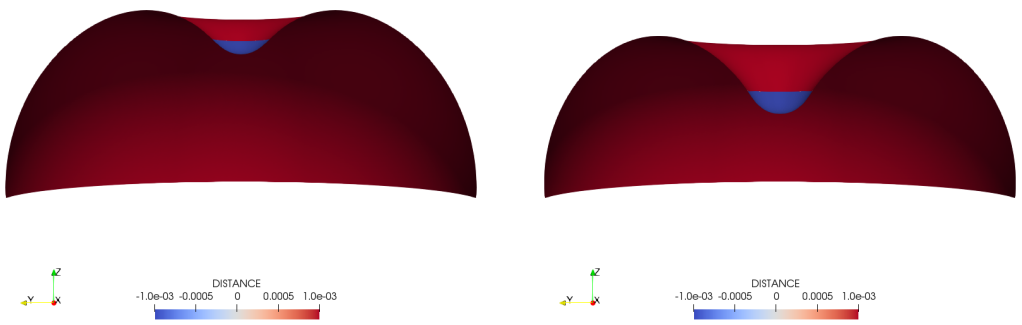
Figure 18: Comparison of monolithic and partitioned methods for various volume increments: a) CPU time vs fluid volume V_f with volume increment, $\Delta V_f = 0.2\ m^3$, b) CPU time vs fluid volume V_f with volume increment, $\Delta V_f = 0.4\ m^3$, c) CPU time vs fluid volume V_f with volume increment, $\Delta V_f = 0.8\ m^3$, d) number of coupling iterations for partitioned methods vs fluid volume V_f with volume increment $\Delta V_f = 0.4\ m^3$.



(a) (b)

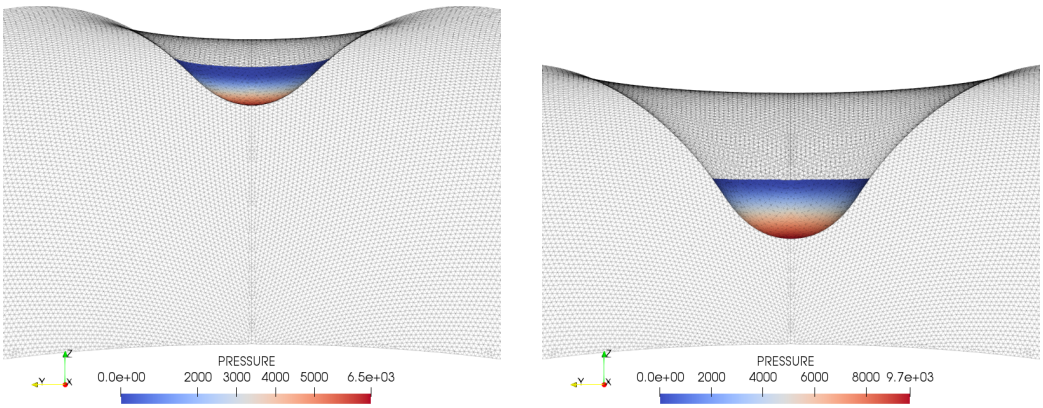
Figure 19: Evolution of fluid volume with N-R steps of $\mathcal{S} \oplus \mathcal{L}(\mathcal{F})$ at the first volume increment step for IPCFL method: a) volume increment $\Delta V_f = 0.2\ m^3$, which corresponds to $V_f = 0.4\ m^3$ at the first volume increment step, b) volume increment $\Delta V_f = 0.4\ m^3$, which corresponds to $V_f = 0.8\ m^3$ at the first volume increment step.

[3]See footnote 1.



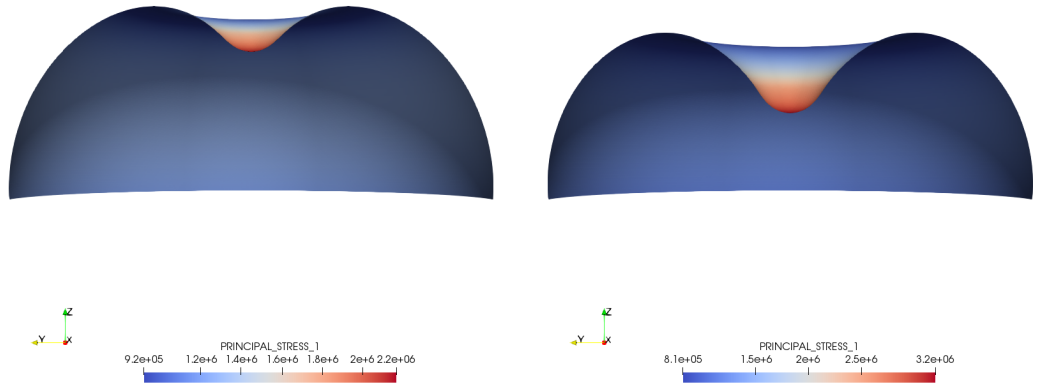
(a) (b)

Figure 20: Deformed shape of the inflated hemisphere under the initial dead load and hydrostatic load from water, where the blue color on the surface indicates negative vertical distance from free surface, and therefore represents wetted surface.[3]: a) $V_f = 1.6\ m^3$, b) $V_f = 3.2\ m^3$.



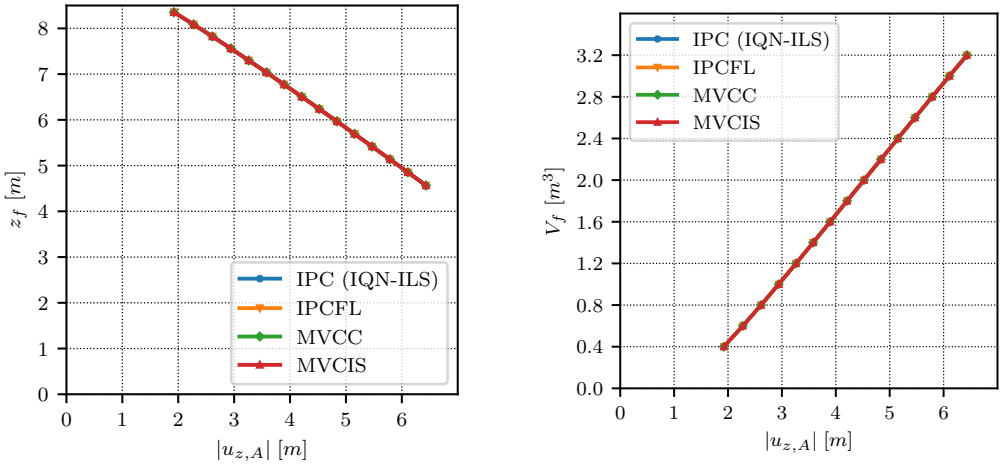
(a) (b)

Figure 21: Hydrostatic pressure due to water on the surface of the inflated hemispherical membrane where the seeding load is applied: a) $V_f = 1.6\ m^3$, b) $V_f = 3.2\ m^3$.



(a) (b)

Figure 22: Maximum principal stress due to water on the inflated hemispherical membrane: a) $V_f = 1.6\ m^3$, b) $V_f = 3.2\ m^3$.



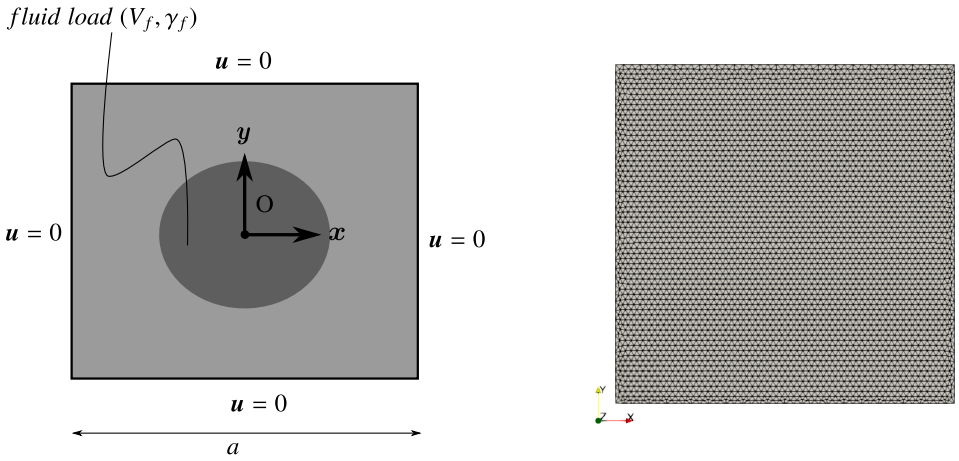
(a) (b)

Figure 23: Variation of magnitude of vertical deflection of top most point of hemisphere, point A (0, 0, 10) with fluid volume (V_f) and free surface height (z_f) calculated using the partitioned and monolithic methods : a) free surface height vs magnitude of vertical deflection of point A, b) volume of fluid vs magnitude of vertical deflection of point A.

8.3. Ponding on a square membrane

The previous two examples show the axi-symmetric cases which can be also analysed using axisymmetric formulations discussed in [7, 3]. Therefore, in this numerical example, to show the general applicability of the discussed algorithms, we present an additional numerical example where we perform ponding analysis on a flat square membrane of side $a = 10\ m$ clamped along edges. The material is modeled as Saint-Venant Kirchhoff plane stress material with material properties: Young's modulus $E = 10^8\ N/m^2$, Poisson ratio $\nu = 0.3$ and thickness $t = 0.001\ m$. The membrane is located in X-Y plane with the centre O at the origin as shown in Fig. 24a . The geometry is discretized with 7748 linear triangle elements shown in Fig. 24b. As a seeding event we apply an initial deformation, $\boldsymbol{u}_{init} = [0, 0, -cos(x\pi/a)\ cos(y\pi/a)]$. The ponding fluid, water ($\gamma_f = 10^4$) in this case is added gradually to simulate ponding.

The performance of the the different algorithms discussed in the paper were found to be in agreement with the findings of the previous numerical examples. Therefore, we only discuss one of the practical applications of the ponding analysis: stability of the pond on the given membrane structure. Basically, we want to check if the given structure can limit the amount of ponding fluid in the event of rainfall or any other similar event. The analysis involves adding fluid volume in steps while observing the free surface position. If the free surface position goes above the clamped edges, the pond is stable with some maximum fluid volume $V_{f,max}$. In Fig. 25, we can clearly see that pond is stable for the current problem. The dashed black line in the figure represents the free-surface at the clamped edge. The fluid volume at which



(a) (b)

Figure 24: Ponding on a square membrane: a) problem set up, b) meshed geometry (viewed from top).

the free surface goes above this line gives the value of $V_{f,max}$. The simulation results after this point (shown in dashed lines) are obtained by assuming a vertical wall along the clamped edge. In the real scenario, the water will overflow after this point. As seen in the figure, one can decrease this volume by pre-stressing the membrane. The deformation of the square membrane under ponding loads can be viewed in Fig. 26, where the effect of membrane pre-stress on the deformed shapes under two different volumes of water are shown.

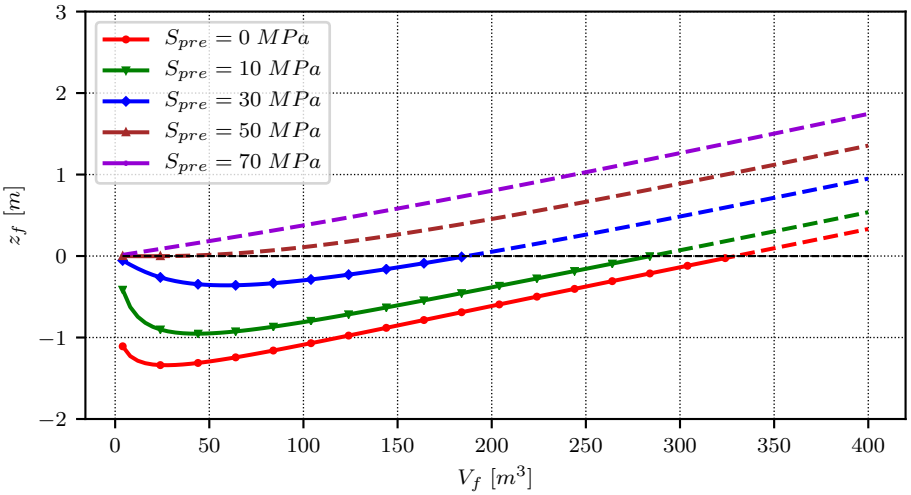


Figure 25: Ponding stability analysis on the square membrane with different pre-stresses S_{pre}, where the black dashed lines represents the maximum position of the free surface in the real scenario.

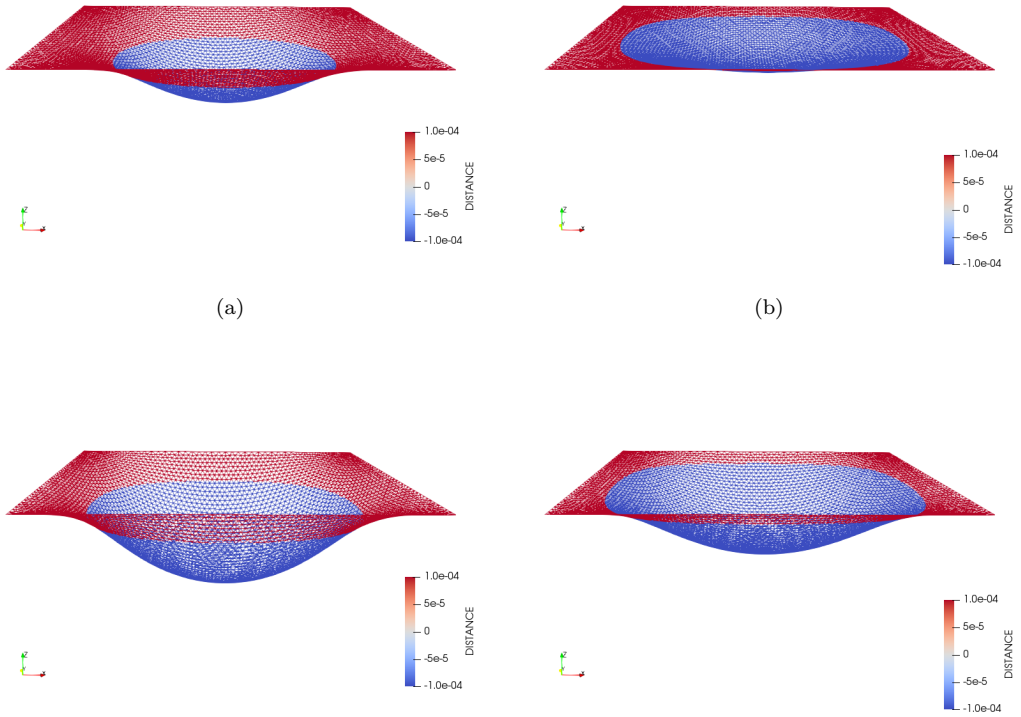


Figure 26: Deformation of the square membrane due to the hydrostatic loading from water with different membrane pre-stress S_{pre}, where the blue color on the surface indicates negative vertical distance from the free surface, and therefore represents the wetted region: a) $V_f = 20\ m^3$ and $S_{pre} = 10\ MPa$, b) $V_f = 20\ m^3$ and $S_{pre} = 30\ MPa$, c) $V_f = 60\ m^3$ and $S_{pre} = 10\ MPa$, d) $V_f = 60\ m^3$ and $S_{pre} = 30\ MPa$.

9. Conclusions

In this paper, we presented two monolithic and two partitioned methods to compute static deformation of membrane structures under ponding loads. In the first partitioned method (IPC) the problem of finding the static deformation under ponding loads was formulated as a fixed point problem where the structural solver and volume-conserving solver, were coupled externally using fixed point iterations or coupling iterations. When the fixed point iterations were used without any modification, the coupling iterations took longer to converge to a solution or sometimes did not converge based on the fluid and structural properties. Therefore, in order to accelerate and stabilize the convergence, convergence accelerators such as Aitken relaxation and IQN-ILS were used. It was found that IQN-ILS had better convergence characteristics than Aitken relaxation because unlike Aitken relaxation, IQN-ILS computes a low rank approximation of the inverse Jacobian of the fixed point residual. This observation is consistent with the literature [30, 17]. A second partitioned method was proposed where the structural solver was modified ($\mathcal{S} \oplus \mathcal{L}(\mathcal{F})$) to include the linearized behavior of the fluid, which was called IPCFL. As expected IPCFL had better convergence characteristics than IPC for small fluid volume increments. However, for larger increments the method had problems due to the linearization error in the initial $\mathcal{S} \oplus \mathcal{L}(\mathcal{F})$ N-R iterations.

In the monolithic methods the structural solver was modified to include the volume conservation property of fluid and the solution were obtained at the end of the N-R iterations. Therefore, there was no need for any external coupling iterations. The first monolithic method (MVCIS) used the volume-

conserving solver to update the free surface after every N-R iteration while in the second method, which was called MVCC, the volume constraint was included in the structural equations, where the constraint is only satisfied at the end of the N-R iterations. The performance of both methods was on par in terms of computational time but the MVCC method introduced in this paper was found to be more robust. Therefore, it is recommended to use IPC with IQN-ILS or with any other quasi-Newton convergence accelerator if code modularity and use of a pre-existing solver is a priority but if the computational cost is most important and if the structural solver can be modified, MVCC seems to be a better option.

10. Future Work

The algorithms discussed in the paper will be used to find the initial conditions for the FSI simulation where the effect of ponding on the membrane structure during windy weather conditions will be investigated. The ponding fluid and the wind in the FSI simulation will be simulated using the volume of fluid method.

11. Acknowledgment

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

References

[1] P. D. Meutter, L. Gerard, G. Smet, K. Hamid, R. Hamdi, D. Degrauwe, P. Termonia, Predicting small-scale, short-lived downbursts:

Case study with the NWP limited-area ALARO model for the pukkelpop thunderstorm, Monthly Weather Review 143 (3) (2015) 742–756. doi:10.1175/mwr-d-14-00290.1.

[2] H. I. Epstein, T. J. Strnad, Liquid-filled, liquid-supported, circular structural membranes, Computers & structures 21 (3) (1985) 443–451. doi:10.1016/0045-7949(86)90234-8.

[3] C. Y. Tuan, Ponding on circular membranes, International journal of solids and structures 35 (3-4) (1998) 269–283. doi:10.1016/S0020-7683(97)00076-0.

[4] K.-U. Bletzinger, R. Wüchner, F. Daoud, N. Camprubí, Computational methods for form finding and optimization of shells and membranes, Computer methods in applied mechanics and engineering 194 (30-33) (2005) 3438–3452. doi:10.1016/j.cma.2004.12.026.

[5] A. Jarasjarungkiat, R. Wüchner, K.-U. Bletzinger, A wrinkling model based on material modification for isotropic and orthotropic membranes, Computer Methods in Applied Mechanics and Engineering 197 (6-8) (2008) 773–788. doi:10.1016/j.cma.2007.09.005.

[6] G. De Nayer, A. Apostolatos, J. N. Wood, K.-U. Bletzinger, R. Wüchner, M. Breuer, Numerical studies on the instantaneous fluid–structure interaction of an air-inflated flexible membrane in turbulent flow, Journal of Fluids and Structures 82 (2018) 577–609. doi:10.1016/j.jfluidstructs.2018.08.005.

[7] W. Szyszkowski, P. Glockner, Finite deformation and stability behaviour of spherical inflatables subjected to axi-symmetric hydrostatic loading, International Journal of Solids and Structures 20 (11-12) (1984) 1021–1036. doi:10.1016/0020-7683(84)90088-x.

[8] T. Rumpel, K. Schweizerhof, Hydrostatic fluid loading in non-linear finite element analysis, International Journal for Numerical Methods in Engineering 59 (6) (2004) 849–870. doi:10.1002/nme.892.

[9] O. Morand, Fluid Structure Interaction, John Wiley & Sons, 1995.

[10] J.-S. Schotté, R. Ohayon, Incompressible hydroelastic vibrations: finite element modelling of the elastogravity operator, Computers & Structures 83 (2-3) (2005) 209–219. doi:10.1016/j.compstruc.2004.03.084.

[11] C. Hoareau, 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 113 (2019) 112–127. doi:10.1016/j.ijnonlinmec.2019.03.014.

[12] C. Hoareau, J.-F. Deü, Non-linear finite element analysis of an elastic structure loaded by hydrostatic follower forces, Procedia engineering 199 (2017) 1302–1307. doi:10.1016/j.proeng.2017.09.320.

[13] Y. Zhou, A. Nordmark, A. Eriksson, Instability of thin circular membranes subjected to hydro-static loads, International Journal of Non-Linear Mechanics 76 (2015) 144–153. doi:10.1016/j.ijnonlinmec.2015.06.010.

[14] A. Eriksson, Structural instability analyses based on generalised path-following, Computer Methods in Applied Mechanics and Engineering 156 (1-4) (1998) 45–74. doi:10.1016/s0045-7825(97)00200-4.

[15] A. Bown, T. Makin, D. Wakefield, Beyond static analysis: Investigation of membrane structure performance using time stepping, transient and progressive analyses, Procedia Engineering 155 (2016) 313–322. doi:10.1016/j.proeng.2016.08.034.

[16] U. Küttler, W. A. Wall, Fixed-point fluid–structure interaction solvers with dynamic relaxation, Computational mechanics 43 (1) (2008) 61–72. doi:10.1007/s00466-008-0255-5.

[17] J. Degroote, Partitioned simulation of fluid-structure interaction, Archives of computational methods in engineering 20 (3) (2013) 185–238. doi:10.1007/s11831-013-9085-5.

[18] M. Bischoff, K.-U. Bletzinger, W. A. Wall, E. Ramm, Models and Finite Elements for Thin-Walled Structures, American Cancer Society, 2004, Ch. 3. arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/0470091355.ecm026, doi:10.1002/0470091355.ecm026.

[19] J. Bonet, R. D. Wood, Nonlinear Continuum Mechanics for Finite Element Analysis, Cambridge University Press, 2008. doi:10.1017/cbo9780511755446.

[20] Holzapfel, Nonlinear Solid Mechanics, John Wiley & Sons, 2000.

[21] P. Dadvand, R. Rossi, E. Oñate, An object-oriented environment for developing finite element codes for multi-disciplinary applications,

907 Archives of computational methods in engineering 17 (3) (2010) 253–
908 297. doi:10.1007/s11831-010-9045-2.

909 [22] R. L. Taylor, E. Oñate, P.-A. Ubach, Finite element analysis of mem-
910 brane structures, in: Computational Methods in Applied Sciences,
911 Springer-Verlag, 2005, pp. 47–68. doi:10.1007/1-4020-3317-6_4.

912 [23] H. D. Hibbit, Some follower forces and load stiffness, International
913 Journal for Numerical Methods in Engineering 14 (6) (1979) 937–941.
914 doi:10.1002/nme.1620140613.

915 [24] A. B. Kasturiarachi, Leap-frogging newton's method, International
916 Journal of Mathematical Education in Science and Technology 33 (4)
917 (2002) 521–527. doi:10.1080/00207390210131786.

918 [25] G. Romano, Potential operators and conservative systems, Meccanica
919 7 (3) (1972) 141–146. doi:10.1007/bf02128759.

920 [26] W. W. Hager, Updating the inverse of a matrix, SIAM Review 31 (2)
921 (1989) 221–239. doi:10.1137/1031049.

922 [27] B. M. Irons, R. C. Tuck, A version of the aitken accelerator for computer
923 iteration, International Journal for Numerical Methods in Engineering
924 1 (3) (1969) 275–277. doi:10.1002/nme.1620010306.

925 [28] Bathe, Finite element procedures, Prentice Hall, Englewood Cliffs, N.J,
926 1996.

927 [29] K.-H. Lee, P.-S. Lee, Nonlinear hydrostatic analysis of flexible floating

928 structures, Applied Ocean Research 59 (2016) 165–182. doi:10.1016/
929 j.apor.2016.05.016.

930 [30] J. Degroote, R. Haelterman, S. Annerel, P. Bruggeman, J. Vieren-
931 deels, Performance of partitioned procedures in fluid–structure inter-
932 action, Computers & structures 88 (7-8) (2010) 446–457. doi:10.1016/
933 j.compstruc.2009.12.006.