

Original Paper

On the efficient non-linear solver for hydraulic fracturing and well cementing simulations based on Anderson acceleration

D.Yu. Derbyshev, S.A. Boronin^{*}, G.V. Ovchinnikov, A.A. Osiptsov

Skolkovo Institute of Science and Technology (Skoltech), Bolshoy Boulevard 30, bld. 1, Moscow, 121205, Russia

ARTICLE INFO

Article history:

Received 26 September 2023

Received in revised form

19 June 2024

Accepted 21 June 2024

Available online 28 June 2024

Edited by Yan-Hua Sun

Keywords:

Anderson acceleration

Non-linear solver

Hydraulic fracturing

Well cementing

Yield-stress fluid

ABSTRACT

The aim of this study is to create a fast and stable iterative technique for numerical solution of a quasi-linear elliptic pressure equation. We developed a modified version of the Anderson acceleration (AA) algorithm to fixed-point (FP) iteration method. It computes the approximation to the solutions at each iteration based on the history of vectors in extended space, which includes the vector of unknowns, the discrete form of the operator, and the equation's right-hand side. Several constraints are applied to AA algorithm, including a limitation of the time step variation during the iteration process, which allows switching to the base FP iterations to maintain convergence. Compared to the base FP algorithm, the improved version of the AA algorithm enables a reliable and rapid convergence of the iterative solution for the quasi-linear elliptic pressure equation describing the flow of particle-laden yield-stress fluids in a narrow channel during hydraulic fracturing, a key technology for stimulating hydrocarbon-bearing reservoirs. In particular, the proposed AA algorithm allows for faster computations and resolution of unyielding zones in hydraulic fractures that cannot be calculated using the FP algorithm. The quasi-linear elliptic pressure equation under consideration describes various physical processes, such as the displacement of fluids with viscoplastic rheology in a narrow cylindrical annulus during well cementing, the displacement of cross-linked gel in a proppant pack filling hydraulic fractures during the early stage of well production (fracture flowback), and multiphase filtration in a rock formation. We estimate computational complexity of the developed algorithm as compared to Jacobian-based algorithms and show that the performance of the former one is higher in modelling of flows of viscoplastic fluids. We believe that the developed algorithm is a useful numerical tool that can be implemented in commercial simulators to obtain fast and converged solutions to the non-linear problems described above.

© 2024 The Authors. Publishing services by Elsevier B.V. on behalf of KeAi Communications Co. Ltd. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

1. Introduction

1.1. Motivation

Design of oilfield service technologies including hydraulic fracturing is typically developed using commercial simulators based on mathematical models of continuum mechanics (Osiptsov, 2017). One of the strict requirements to commercial simulators is their ability to carry out a very fast numerical solution of governing equations, which allows performing massive computations in the framework of parametric studies and optimization tasks. After governing equations of a fluid flow process are formulated, their

numerical solution is usually obtained by a certain discretization with a consequent solution of non-linear algebraic equations. Iterative non-linear solvers are usually composed of two parts, namely, linearization of the problem (e.g., by fixed-point or Jacobian-based methods) and a solution of the linear system (e.g., by families of the Krylov space or multigrid methods).

Our previous study (Muravleva et al., 2021) discussed the efficient numerical implementation of finding the solution to a large system of linear equations using the black-box multigrid solver with matrix-dependent prolongation operators. This method was applied to a five-diagonal block matrix obtained by discretization of a quasi-linear Poisson equation (2D problem) of the form:

^{*} Corresponding author.

E-mail address: s.boronin@skoltech.ru (S.A. Boronin).

$$\nabla \cdot (\mathcal{A}(p) \nabla p) = f. \quad (1)$$

Here, $p(\mathbf{r}) \in \mathcal{C}^2(\Omega)$ is a smooth function defined over a domain in Euclidean space $\Omega \in \mathbf{R}^n$, where n depends on the dimension of the corresponding mathematical model ($n = 1, 2, 3$), and coefficient $\mathcal{A}$ is a function of the solution p , so the equation is essentially non-linear.

The current study is aimed at developing an easy to implement yet effective and stable numerical algorithm based on the Anderson acceleration (AA) of the fixed-point (FP) method for iterative solution of the nonlinear problem described by Eq. (1), which is the first part of the numerical solution workflow discussed above.

Eq. (1) describes various physical processes, including multiphase flow of a particle-laden non-Newtonian suspension in a hydraulic fracture, multiphase flow in a rock formation (used in modeling enhanced oil recovery technologies and reservoir production simulations), hydraulic fracture cleanup, and displacement of fluids with complex rheology in a cylindrical annulus during well drilling and cementing (Bittleston and Hassager, 1992; Pearson, 1994; Hammond, 1995; Pelipenko and Frigaard, 2004b; Adachi et al., 2007; Boronin and Osipov, 2010; Boronin et al., 2015a; Osipov et al., 2020).

Hydraulic fracturing (HF) is a key well stimulation technology required to develop low-permeability oil and gas fields (see reviews by Osipov (2017), Barboza et al. (2021)). Various aspects of HF modelling maintain significant interest of petroleum scientific and engineering community (e.g., see recent papers by Zhang et al. (2017), Yuwei et al. (2020); Li et al. (2023a, 2023b) Isaev et al. (2023)). Below we describe exiting methods of numerical solution of governing equations describing hydraulic fracturing (HF) of underground hydrocarbon-bearing rock formation and the need for fast and stable numerical solution of Eq. (1).

A mathematical model of the HF process describes two essentially coupled phenomena, namely, elastic rock response to fluid pressure change in the fracture and flow of fluids transporting solid particles (proppant) in the fracture channel (Adachi et al., 2007). Governing equations describing these processes are usually solved numerically by applying one of the following coupling strategies: (i) semi-implicit coupling, in which the mass conservation for suspension and rock elasticity equations are solved simultaneously to obtain distributions of fluid pressure and fracture aperture; then proppant concentration advection equations are solved explicitly by using the obtained distributions of particle velocity and fracture aperture (Adachi et al., 2007; Dontsov and Peirce, 2015; Shiozawa and McClure, 2016); (ii) explicit coupling, when elastic rock equations and a simplified variant of suspension mass conservation equation (with fixed distribution of fluid viscosity and/or with reduced dimension in case of 1D rock elasticity models) are solved to obtain fracture aperture; then the calculated aperture is used to solve general 2D suspension flow mass balance equation described by Eq. (1) (with coefficient $\mathcal{A}$ being dependent on local pressure gradient due to complex fluid rheology, details are provided in Section 2.1 below); the obtained fluid and particle velocity fields are then used to update concentrations of fluids and proppant particles according to advection equations (Shiozawa and McClure, 2016; Isaev et al., 2023).

The semi-implicit coupling strategy allows us to obtain more accurate solution to the full system of equations as compared to the explicit one, while it requires the solution of the two coupled and essentially non-linear (in terms of the fluid pressure and fracture aperture) equations. The corresponding matrix determining the linearized system is dense and large (twice the number of mesh nodes, as the vector of unknowns contains the approximated pressure and aperture, see details of the numerical solution of

governing equations in the framework of semi-implicit coupling in the study by Adachi et al. (2007)). The system of these equations is additionally complicated due to a non-Newtonian (power-law or viscoplastic) fluid rheology. The numerical solution of these equations consumes the major part of overall CPU time required for the solution of the total system of equations. The explicit coupling strategy described above is more flexible and potentially allows us to obtain faster numerical solution. Elastic rock response and suspension flow models of different dimensions can be coupled using this approach to speed-up computations significantly, while preserving the ability to calculate proppant transport accurately (e.g., 1D pseudo 3D geomechanics model (Dontsov and Peirce, 2015) can be used to obtain fracture aperture distribution and 2D proppant transport model can be implemented to describe proppant transport and settling as described in (Isaev et al., 2023)); in the framework of more accurate calculations using 2D geomechanics (Planar3D, see Adachi et al. (2007)) and proppant transport models, the explicit coupling can still be beneficial. It allows simplifying the non-linear system of rock-fluid equations described above by neglecting several physical effects, while preserving the total mass balance of suspension in the hydraulic fracture (e.g., gravitational convection, proppant settling, pressure gradient-dependent fluid viscosity for power-law and viscoplastic fluids). In the framework of the numerical solution of governing equations using the explicit coupling strategy, quasi-linear elliptic Eq. (1) have to be solved at consequent time instants assuming the that the fracture aperture is a known function of spacial coordinates and time.

The study aims to enhance standard proppant transport models used in commercial simulators. The current models use the power-law model to describe the rheology of fracturing fluids and are effective in describing the rheology of slick-water and linear gels at arbitrary flow shear rates. However, cross-linked polymer gels exhibit yield-stress behavior in rheology experiments (Acharya and Deysarkar, 1987; Hu et al., 2015; Barbati et al., 2016). We obtained that during the formation of unyielded zones, the number of FP iterations per time step required to solve numerically Eq. (1) was extremely high. These zones are characterized by a local pressure gradient that is not sufficient to induce a viscous stress above a yield stress limit, causing the viscoplastic fluid to behave as a solid body and not move. Additionally, the iterative procedure diverged in the majority of time steps. As demonstrated below, the FP iteration method cannot resolve unyielded zones, significantly reducing calculation accuracy and resulting in unrealistic behavior of viscoplastic fluids.

1.2. Brief overview of non-linear solvers

Below we present a brief overview of existing nonlinear solvers and studies related to the development and application of the AA method.

One of the easiest to implement methods for solving nonlinear problems described by Eq. (1) is the FP iteration method (also known as Picard iteration), which is usually applied to equations of the form

$$\mathbf{A}(\mathbf{p}) = \mathbf{p}. \quad (2)$$

Here, $\mathbf{A}$ is a mapping operator $\mathbf{R}^n \rightarrow \mathbf{R}^n$ and $\mathbf{p} \in \mathbf{R}^n$ is a vector of unknowns (e.g., pressure field approximation in mesh nodes). The method is based on iterative computation of approximations to the solution $\mathbf{p}_{s+1}$ using the following equation:

$$\mathbf{p}_{s+1} = \mathbf{A}_{s+1}(\mathbf{p}_s). \quad (3)$$

Here, s is the number of the current iteration. This is a variant of the

general Krasnoselskii–Mann (KM) method (Mann, 1953; Hillel, 1975) based on weighing the results of applying the transformation operator to the solution guess at the previous iteration and the guess:

$$\mathbf{p}_{s+1} = (1 - \alpha)\mathbf{p}_s + \alpha\mathbf{A}_{s+1}(\mathbf{p}_s), \quad 0 < \alpha \leq 1. \quad (4)$$

The convergence of the KM method is proved for non-expansive mapping operators for which the following inequality holds:

$$\|\mathbf{A}(\mathbf{p}_1) - \mathbf{A}(\mathbf{p}_2)\|_2 \leq \|\mathbf{p}_1 - \mathbf{p}_2\|_2. \quad (5)$$

Here, $\mathbf{p}_1, \mathbf{p}_2 \in \mathbf{R}^n$ are arbitrary vectors and $\|\cdot\|$ is the standard norm defined in the Euclidean space $\mathbf{R}^n$. While the KM algorithm is relatively easy to implement, it usually converges slowly and requires a large number of iterations. To improve the convergence rate, a balance between the number of iterations and the computational cost of each iteration is required.

Consider a function $\mathbf{A}$ defined as

$$\mathbf{A}(\mathbf{p}) = \mathbf{p} - \alpha \nabla F(\mathbf{p}). \quad (6)$$

This is a problem of finding a minimum of a differentiable function F of a vector argument $\mathbf{p}$ using the gradient descent method. The problem can be solved using a family of Newton–Raphson methods or accelerated gradient descent methods (Luenberger et al., 1984). There are also hybrid methods based on coupled KM and Newton algorithms that aim to maintain a fast convergence rate and reduce the computational cost of each iteration (Scieur et al., 2020). Note that the elements of the Jacobi matrix corresponding to Eq. (1) describing a flow of non-Newtonian (power-law or Herschel–Bulkley) fluids can be calculated either numerically or analytically. Due to the complicated expression of the fluid mobility (coefficient λ in Eq. (1)) in terms of the pressure gradient, both these variants require time-consuming CPU operations to raise a number to a non-integer power, potentially reducing the performance of the computations.

The vast majority of existing numerical methods developed to solve nonlinear problems have been shown to converge only under certain assumptions imposed on the operator $\mathbf{A}$, for example that $\mathbf{A}$ is differentiable in the neighborhood of the solution $\mathbf{p}$ (Schlenkerich and Walther, 2009; Chen et al., 2014) or that the Jacobi matrix is symmetric (Li and Fukushima, 1999; Zhou and Li, 2007). In most of the algorithms mentioned above, a line search is used to guarantee the global convergence, which usually leads to a significant increase in computation time if the computation of the operator $\mathbf{A}(\mathbf{p})$ requires a large number of mathematical operations.

AA algorithm is proposed to achieve the global convergence of the numerical algorithm with the minimum number of mathematical operations per iteration (Anderson, 1965; Fang and Saad, 2009). According to various test calculations, the variant of the AA algorithm proposed by Fang and Saad (2009) converges faster compared to the original version (Anderson, 1965), while the former is less stable in terms of convergence (Fang and Saad, 2009; O'donoghue et al., 2016). Existing studies of the AA applied to nonlinear problems (with the exception of the variant of Fang and Saad (2009)) show that its convergence requires differentiable operator $\mathbf{A}$ and is either proved locally (Gay and Schnabel, 1978; Rohwedder and Schneider, 2011; Toth and Kelley, 2015) or requires non-expanding mapping (Scieur et al., 2017, 2020). AA method is widely and successfully used to solve nonlinear problems for which the convergence cannot be strictly proved (Pavlov et al., 2018; Matveev et al., 2018; Walker and Ni, 2011; Potra and Engler, 2013); the algorithm is implemented in several computational libraries including SCS (O'donoghue et al., 2016) and CVXPY (Agrawal et al., 2018).

The base variant of AA algorithm applied to the FP iteration is shown to be a useful tool in solving non-linear numerical problems of fluid mechanics and geomechanics, including those arising in oilfield technological processes: saddle-point problem in fluid mechanics (Ho et al., 2017); modeling of multiphase flows in porous media (Salinas et al., 2017) and high-rate hydraulic fractures driven by wellbore energy source (Chertov and Chaplygin, 2019); numerical coupling of wellbore and hydraulic fracture models to study proppant placement and flowback problems (Aksenov et al., 2021; Sinkov et al., 2021).

1.3. Contribution

Current study is aimed at developing a variant of the AA algorithm to solve essentially non-linear problem formulated during a numerical solution of Eq. (1) describing the flow of particle-laden yield stress fluids in a hydraulic fracture. Based on test numerical simulations of viscoplastic suspension flow in a narrow channel approximating a hydraulic fracture, we show that the developed algorithm allows us not only to speed up the basic FP iterative algorithm, but also to obtain a converged solution in the presence of unyielding zones formed in a hydraulic fracture.

A parametric study of the convergence rate is performed to identify the optimal values of the tuning parameters of AA algorithm (namely, memory and delay) that provide the best performance. The set of test cases is representative in terms of ranges of values of input parameters (flow rates, rheology of fluids, fracture dimensions), so that the obtained results can be used to perform CPU-time efficient and convergent computations of problems described by the quasi-linear Eq. (1), including those describing hydraulic fracturing and well cementing processes.

To the best of our knowledge, the proposed modification of the AA algorithm is novel. Our estimations show that its performance is significantly better as compared to that of the family of Jacobian-based algorithms (typically used to solve the equations describing hydraulic fracturing (see Adachi et al. (2007), Shiozawa and McClure (2016), Dontsov and Peirce (2015)) as applied to the problem of viscoplastic fluid flows in narrow channels.

2. Mathematical model

Hydraulic fracturing is a technology widely used in the oilfield services industry to stimulate hydrocarbon-bearing formations (eg., Economides and Nolte (2000)). It usually comprised of the following steps: (i) injection of a fluid into the target rock formation through the perforated wellbore at bottom-hole pressures exceeding the effective minimum rock stress to create a fracture; (ii) placement of solid proppant particles into the open fracture channel by injection of a particle-laden suspension; (iii) appropriate well clean-up procedures to prevent proppant flowback leading to disconnection of the created fracture from the wellbore and to clean the proppant pack from the fracturing fluid to maintain long-term production of hydrocarbons. The result is a highly conductive, proppant-filled channel that significantly increases hydrocarbon production compared to the untreated formation. A single hydraulic fracture wing attached to a vertical well is shown schematically in Fig. 1.

2.1. Governing equations

In the following, we describe the mathematical model for the flow of a particle-laden suspension in a single wing of the hydraulic fracture approximated by a narrow planar channel. The governing equations are formulated in the framework of the lubrication approximation to the 3D Navier-Stokes equations describing the

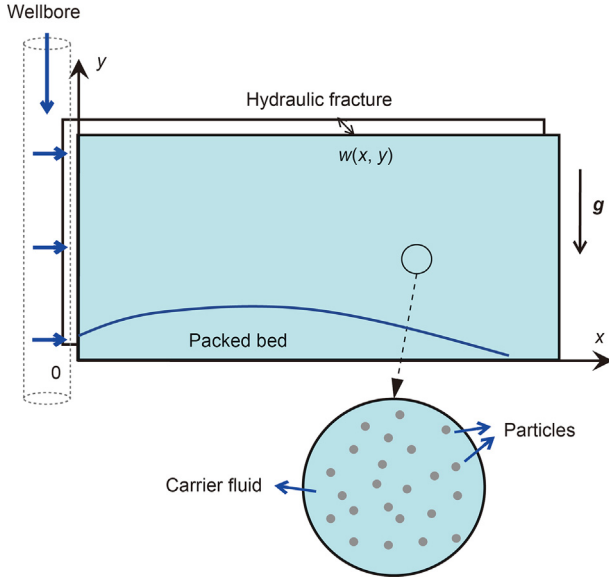


Fig. 1. Schematics of suspension flow in a single wing of the hydraulic fracture.

flow of the carrier fluid and the particulate phase using the two-fluid approach (e.g., see Brennen (2005)).

Several incompressible and immiscible carrier fluids and several proppant types (non-Brownian spherical particles) are assumed as interpenetrating continua. The rheology of the fluids is described using the Herschel-Bulkley model. Note that the power-law model is typically used to describe the rheology of fracturing fluids in industrial simulators, which fits the rheological behavior of slick-water and linear gels very well, while cross-linked gels show yield-stress behavior in rheological experiments (Acharya and Deysarkar, 1987; Hu et al., 2015). Governing equations describing the flow of Newtonian or power-law particle-laden fluids in a narrow hydraulic fracture are provided, for example, in the studies of Adachi et al. (2007), Boronin and Osipov (2010), Isaev et al. (2023), while the model for particle-free flow of yield-stress fluids in a hydraulic fracture and narrow cylindrical annulus during well cementing is formulated in Boronin et al. (2015a), Bittleston and Hassager (1992), Pelipenko and Frigaard (2004a), Carrasco-Teja and Frigaard (2010).

While a hydraulic fracture propagation is essentially coupled process of rock deformation/fracturing and fluid flow in the open channel, in this study we focus only on the latter problem. We assume that the fracture aperture dynamics is calculated using the appropriate geomechanics model (Adachi et al., 2007; Dontsov and Peirce, 2015) in the framework of explicit numerical coupling as discussed in Section 1.1, so that the suspension flow equations formulated below are solved at consecutive time instants.

Consider a Cartesian coordinate system Oxy with the horizontal x -axis and the vertical y -axis, the origin O is located at the fracture midplane (see Fig. 1). The following scales are used: fluid velocity at the fracture inlet (injection zone) U , which is calculated based on the volumetric injection rate of suspension; fracture length L and aperture d_0 (so that $d_0/L \ll 1$ is a small parameter), density ρ_1 , viscosity μ_1 and $\tau_{y,1}$ scales are calculated using the parameters of the fluid 1 (assuming that $\tau_{y,1} > 0$); note that the viscosity scale is calculated at the shear rate $\dot{\gamma}_0 = 6U/d_0$ using the power-law model $\mu_1 = K_1 \dot{\gamma}_0^{n_1}$. The coordinates x and y are scaled by the fracture length L so that the flow domain is $(x, y) \in [0, 1] \times [0, h]$, where $h = H/L$ is the fracture height to length ratio ($H \sim L$).

Mass conservation of fluids and particles at constant substance

densities are formulated as follows:

$$\frac{\partial w(1 - C_p)C_i}{\partial t} + \nabla \cdot (w(1 - C_p)C_i \mathbf{v}_f) = -2v_{l,i}C_i, \quad i = 1, \dots, l, \quad (7)$$

$$\frac{\partial wC_p}{\partial t} + \nabla \cdot (wC_p \mathbf{v}_p) = 0, \quad (8)$$

$$v_{l,i} = \frac{C_{l,i}}{\sqrt{t - t_0}}.$$

Here, $i = 1, \dots, l$ is the fluid number (with $l \geq 2$ being the total number of fluids); C_i are the fluid volume concentrations; C_p is the proppant volume concentration (for simplicity we assume a single proppant type); $w(x, t)$ is the fracture aperture, which is assumed to be a predefined function of coordinates and time as described above; $v_{l,i}$ is the leak-off velocity of a fluid ' i ' through the fracture walls, which is determined according to the Carter law with "fluid exposure time" $t_0(x, y)$ (the time instant, at which the fracture tip reached the location (x, y)) and (constant) coefficients $C_{l,i}$; $\mathbf{v}_f$ and $\mathbf{v}_p$ are the carrier fluid and particle (mass-averaged) velocities, respectively.

Gap-averaged momentum conservation equations formulated for carrier fluids and particles in the lubrication limit take the following form:

$$\mathbf{v}_f = -\frac{w^2}{12\mu_m} Y(\psi) \mathbf{F}, \quad \mathbf{v}_p = \mathbf{v}_f + \mathbf{v}_s, \quad \mathbf{v}_s = -v_s \mathbf{e}_2, \quad (9)$$

$$v_s = v_{st} f(C_p), \quad v_{st} = \frac{(\rho_p - \rho_f)gD^2}{18\mu_f \mu_1 U}, \quad f(C_p) = (1 - C_p)^{5.1}, \quad (10)$$

$$\mu_m = \mu_s(C_p)\mu_f, \quad \mu_f = \frac{Fw}{12k} \left(\frac{2K}{Fw} \right)^{1/n}, \quad \mathbf{F} = \nabla p + \text{Bu} \rho_m \mathbf{e}_y, \quad F = |\mathbf{F}|, \quad (11)$$

$$k = \frac{n^2}{(n+1)(2n+1)}, \quad Y(\psi) = \frac{1}{2}(1 - 2\psi)^{1+1/n} \left(1 + \frac{1}{n} + 2\psi \right), \quad (12)$$

$$\psi = \frac{\text{Bn} \tau_y}{wF}, \quad \mu_s(C_p) = \left(1 - \frac{C_p}{C_{\max}} \right)^{-2.5n}, \quad (13)$$

$$K = \sum_{i=1}^N C_i M_i, \quad n = \sum_{i=1}^N C_i n_i, \quad \tau_y = \sum_{i=1}^N C_i T_i, \quad (14)$$

$$\rho_m = (1 - C_p) \left(C_1 + \sum_{i=2}^N C_i \zeta_i \right) + C_p \zeta_p, \quad v_l = \sum_{i=1}^N C_i v_{l,i}, \quad v_{l,i} = \frac{C_{l,i}}{\sqrt{t - t_0}}, \quad (15)$$

$$\text{Bu} = \frac{\rho_1 g d^2}{\mu_1 U}, \quad \text{Bn} = \frac{\tau_{y,1} d}{\mu_1 U}, \quad \zeta_i = \frac{\rho_i}{\rho_1} \quad (i = 2, \dots, N), \quad (16)$$

$$\zeta_p = \frac{\rho_p}{\rho_1}, \quad M_i = \frac{K_i \dot{\gamma}_0^{n_i-1}}{K_1 \dot{\gamma}_0^{n_1-1}}, \quad T_i = \frac{\tau_{y,i}}{\tau_{y,1}}. \quad (17)$$

Here, p is the fluid pressure; F is the fluid pressure gradient

accounting for gravity; K , n , and τ_y are the dimensionless consistency index, power-law index and yield stress determining the rheology of a mixture of viscoplastic fluids according to the Herschel–Bulkley model; v_l is the total leak-off velocity of the fluids; ρ_m and μ_m are the density and shear viscosity of the suspension, respectively; ρ_p and D are particle substance density and diameter, respectively.

Eqs. (10)–(14) determine the closure relations for the apparent suspension viscosity μ_m and the particle settling velocity v_s in terms of the flow parameters; Eqs. (9), (11), and (12) describe the flow of yield-stress fluids only if the maximum local viscous stress along the fracture aperture is greater than the suspension yield stress τ_m , which is described in a non-dimensional form as $\psi < 0.5$ ($Y(\psi) > 0$) and is equivalent to $F > F_c = 2Bn\tau_y/w$ according to Eq. (12). If the local pressure gradient is below the threshold F_c , the flow stops (the fluid is unyielded), which is described as follows:

$$\mathbf{v}_f = 0 \text{ or } Y(\psi) = 0 \text{ if } \psi \geq 0.5. \quad (18)$$

The dimensionless groups in Eqs. (16) and (17) are calculated based on the parameters of the fluid 1: Bu is the buoyancy number determining the effect of gravitational convection; Bn is the Bingham number (the ratio of yield stress scale to the viscous stress scale), M_i and ζ_i are the apparent viscosity and density ratios of fluid ‘ i ’ to fluid ‘1’, respectively, T_i is the yield stress ratio of fluid ‘ i ’ to fluid ‘1’, and ζ_p is the density ratio of proppant to fluid ‘1’.

Mass conservation equations (see Eqs. (7) and (8)) and expressions for the carrier fluid and particle velocities (Eq. (9)) allow us to derive the quasi-linear elliptic equation determining pressure evolution in the flow domain:

$$\nabla \cdot \left(\frac{w^3}{12\mu_m} Y(\psi) \mathbf{F} \right) = \frac{\partial w}{\partial t} + 2v_l + \nabla \cdot (wC_p \mathbf{v}_s). \quad (19)$$

It is obtained by (i) summing Eqs. (7) and (8); (ii) applying the volumetric balance condition $C_1 + \dots + C_l = 1$; (iii) expressing the fluid velocity $\mathbf{v}_f$ in terms of the pressure gradient F via Eq. (9) and (iv) expressing the particle velocity $\mathbf{v}_p$ in terms of the pressure gradient F and the particle volume concentration C_p via Eqs. (9) and (10).

Note that parameters μ_m and v_s in Eq. (19) depend on the local volume concentrations of particles C_p and fluids C_i (see Eqs. (13) and (14)), so the system described by Eqs. (7)–(9) and (19) is essentially coupled. The viscosity of the suspension μ_m and the mobility coefficient $Y(\psi)$, which determine the yield-stress behavior of Herschel–Bulkley carrier fluids, depend on the local pressure gradient F (Eqs. (11) and (12)), so that the second-order partial differential Eq. (19) is essentially nonlinear with respect to pressure.

As mentioned above, a similar set of governing equations is used to describe a wide range of oilfield problems, including well cementing and multiphase flow in a proppant-filled hydraulic fracture during clean-up and production (Bittleston and Hassager, 1992; Pearson, 1994; Hammond, 1995; Pelipenko and Frigaard, 2004b; Osiptsov et al., 2020). Multiphase flow in hydrocarbon reservoirs using the black oil model is typically used for routine 3D reservoir production modeling and is also described by a very similar set of governing equations. The only difference is that adequate reservoir modeling requires consideration of fluid compressibility, so that the Eq. (1) (or (19)) becomes parabolic. Nevertheless, both quasi-elliptic and quasi-parabolic nonlinear pressure equations can be solved using a similar finite-difference (or finite-volume) spatial approximation scheme.

2.2. Initial and boundary conditions

For the hyperbolic transport Eqs. (7) and (8), the boundary conditions at the inlet $x = 0$ (perforated segment of the well) and the initial distribution of fluid and proppant concentrations are specified:

$$C_i(0, y \in I_{\text{perf}}, t) = C_i^{\text{in}}, \quad i = 1, \dots, l; \quad C_p(0, y \in I_{\text{perf}}, t) = C_p^{\text{in}}; \quad (20)$$

$$C_i(x, y, 0) = C_i^0(x, y), \quad i = 1, \dots, l; \quad C_p(x, y, 0) = C_p^0(x, y). \quad (21)$$

Boundary conditions described by Eq. (20) are imposed at the subsection of the left vertical boundary of the fracture corresponding to the perforations $y \in I_{\text{perf}}$, which is the inlet zone of the computational domain.

The quasi-elliptic equation in terms of fluid pressure (see Eq. (19)) requires either Neumann or Dirichlet boundary conditions. We impose the no-flow boundary conditions for the fluids at the upper and lower horizontal boundaries. At the perforated region of the left boundary, there is a specified volumetric injection rate, which is expressed in terms of the pressure gradient according to Eq. (9).

We consider two flow configurations, namely a Hele–Shaw cell and a propagating hydraulic fracture. For a Hele–Shaw cell (a narrow channel with an outlet at the right vertical boundary), we impose a variant of the “soft” boundary condition at the outlet $x = 1$ so that the pressure distribution is hydrostatic (equivalent to zero vertical fluid velocity) at each time instant. For a propagating hydraulic fracture, all boundaries except the inlet zone are impermeable (no-flux condition), while the fluid injection rate, the total fluid leakoff rate, and the rate of change of the fracture volume are subject to a volumetric equilibrium condition at each time instant. Note that since the fluids are incompressible and Neumann boundary conditions are imposed at all boundaries of the computational domain, the solution to Eq. (19) is determined up to a free additive constant. Therefore, a unique solution to Eq. (19) formulated for a propagating hydraulic fracture is found by imposing a certain pressure value at any location within the hydraulic fracture (e.g., bottomhole pressure corresponding to the perforated region).

3. Numerical solution using the FP algorithm

The quasi-elliptic pressure equation (see Eq. (19)) can be formulated in the following operator form:

$$A_{\Delta}(p)p = f, \quad A_{\Delta} = \sum_{i=1,2} \frac{\partial}{\partial x_i} \left[\mathcal{A}(p) \frac{\partial}{\partial x_i} \right]. \quad (22)$$

Here, x_i are coordinates, the differential operator A_{Δ} is the Laplacian and $\mathcal{A} = w^3 Y(\psi) / 12\mu_m$ is the fluid mobility multiplied by the fracture aperture w .

An alternative form of Eq. (22) can be formulated using the inverse Laplace operator $A_{\Delta}^{-1}(p)$ acting on the right-hand side f :

$$p = A_{\Delta}^{-1}(p)f = a(p). \quad (23)$$

In fact, $A_{\Delta}^{-1}(p)$ is a solution of the quasi-linear Laplace Eq. (22) subject to the boundary conditions formulated in Section 2.

We consider the discretization of Eq. (22) at the mesh $\Omega(N_x, N_y)$, where N_x and N_y are the number of mesh cells in horizontal and vertical directions, respectively, so that $\mathbf{p} = \{p_{ij}, 0 \leq i \leq N_x, 0 \leq j \leq N_y\}$ is an approximation of the pressure field. The discrete analogue of the Laplacian A_{Δ} is represented by the matrix operator

$\mathbf{A}$, so that the differential problem defined by Eq. (22) becomes the matrix problem:

$$\mathbf{A}(\mathbf{p})\mathbf{p} = \mathbf{f}, \quad \mathbf{A} = \mathbf{A}_{\Delta\Omega}. \quad (24)$$

Similar to Eq. (23), Eq. (24) can be represented as follows (provided that the matrix $\mathbf{A}$ is non-degenerate):

$$\mathbf{p} = \mathbf{A}^{-1}(\mathbf{p}) \cdot \mathbf{f}. \quad (25)$$

As mentioned above, one of the most robust strategies for the numerical solution of Eqs. (7)–(9) and (19) is based on the splitting of the governing equations by physical processes, which leads to a FP problem described by Eq. (24).

3.1. Description of the numerical algorithm for solving the pressure equation

We consider the standard finite-difference discretization using the 5-point “cross” stencil, a rectangular staggered uniform grid and the approximation of the derivatives by central differences (see detailed description of the approximation in Appendix A). The following numerical algorithm is used to solve the pressure equation at each time step:

1. The pressure equation (see Eq. (19)) is discretized assuming a fixed distribution of fluid tracers C_i , particle concentrations C_p and particle slip velocity $\mathbf{v}_s$, so that the coefficients are fixed. The corresponding linear system can be solved by a certain effective linear solver (e.g. the multigrid method proposed in Muravleva et al. (2021));
2. Due to the Herschel-Bulkley rheology of the fluids, the suspension viscosity μ_m depends on the local pressure gradient F according to Eqs. (11) and (12), so the pressure equation is non-linear and the solution performed in step 1) typically leads to a large residual; according to the FP iteration method, after step 1) we recalculate the coefficients in Eq. (19) and perform the calculations according to step 1).

The iterative process 1–2 ends when certain convergence criteria are met. For example, the following criteria can be used ($s \geq 1$ is the number of current iteration of the solution of the pressure equation at a given time step):

$$\frac{\|\mathbf{p}_s - \mathbf{p}_{s-1}\|}{\|\mathbf{p}_s\|} < \epsilon_1, \quad \frac{\|\mathbf{v}_{f,s} - \mathbf{v}_{f,s-1}\|}{\|\mathbf{v}_{f,s}\|} < \epsilon_2. \quad (26)$$

Here, $\epsilon_{1,2}$ are certain small numbers that determine the accuracy of the calculations, and $\|\cdot\|$ is any suitable type of norm, for example:

$$\|\mathbf{a}\|_{L_2} = \sqrt{\sum_i \mathbf{a}_i^2}, \quad \|\mathbf{a}\|_{L_\infty} = \max_i |\mathbf{a}_i|. \quad (27)$$

Multiple convergence criteria can be used simultaneously to ensure proper control of the accuracy of the calculations.

Note that the pressure equation (see Eq. (19)) degenerates when there are unyielding zones in the computational domain. The reason is that the fluid mobility $\lambda(p) = w^3 Y(\psi) / (12\mu_m)$ in these zones is zero according to the condition described by Eq. (18). To solve this issue, we apply a regularization to the mobility of a yield-stress fluid described in the study of Boronin et al. (2015a), so that the coefficient $Y(\psi)$ takes a small non-zero value in the unyielding zones.

After solving the pressure equation at the current time step, the

advection equations (see Eqs. (7) and (8)) are solved explicitly using a particular finite-difference scheme (e.g., the second-order TVD scheme with a flux limiter, see Leveque (1996)).

The entire algorithm of solving the governing equations describing the suspension flow in a hydraulic fracture is similar to the “Implicit Pressure–Explicit Saturation” (IMPES) method for the numerical solution of multiphase filtration problems (Aziz and Settari, 1979).

3.2. Improved Anderson acceleration

We started by implementing the original version of AA method (Algorithm 1 in Appendix B) to solve the nonlinear problem described by Eq. (19). Test calculations showed that for the viscoplastic fluid rheology, the number of time steps at which the iterative algorithm described above diverges is unacceptably large for reasonably small values of the tolerance parameters $\epsilon_{1,2}$ described above.

To address this issue, we made the following modifications to the original AA algorithm:

1. Each time step begins with several iterations using the standard FP iteration procedure. These iterations are not stored in the AA memory array and are not used during the optimization procedure. This allows the algorithm to reach the convergence region. We refer to this parameter as the delay of AA algorithm, as it is commonly termed in the literature (Walker and Ni, 2011).
2. At each iteration, a solution guess is computed using solutions obtained from several previous iterations. The number of previous iterations used is determined by the memory parameter, typically introduced to conserve memory.
3. The optimization process is applied to the expanded vector composed of the linearized pressure approximation operator (matrix) $\mathbf{A}_s$, the right-hand side $\mathbf{f}_s$, and the approximation to the solution at each iteration $\mathbf{p}_s$, not just to the vector of the approximate solution at each FP iteration $\mathbf{p}_s$.
4. At each iteration s , the fluid mobility λ (see Eq. (22)) is updated based on fluid pressure values $\mathbf{p}_{s-1}$ obtained from a discretized problem determined by Eq. (19). A linearized pressure approximation operator is then calculated, along with the right-hand side, to form a mapping over the expanded vector space.
5. Transport equations (Eqs. (7) and (8)) are solved using an explicit 2nd-order TVD approximation scheme, and the time step is determined according to the Courant–Friedrichs–Lewy (CFL) stability condition. A modification of pressure leading to a change in fluid velocities according to Eq. (9) results in a change in the time step calculated by the CFL condition. If the value changes significantly compared to the previous iteration (e.g., the relative difference exceeds 1%), subsequent iterations will use the standard FP iteration with AA algorithm switched off to prevent divergence caused by significant changes in the pressure approximation matrix during the iterative procedure.

Modifications 1 and 2 are commonly used in published studies and are considered part of the basic AA algorithm. Modifications 3–5 are novel, and Appendix B provides a detailed line-by-line comparison of the basic (Algorithm 1) and proposed (Algorithm 2) AA algorithms to highlight these differences. It is important to note that all unknown quantities, namely pressure $\mathbf{p}$, the matrix representing the linearized operator $\mathbf{A}$, and the right-hand side $\mathbf{f}$, enter the algorithm in a vector form.

4. Numerical simulations

This section presents the results of the numerical solution to

Eqs. (7)–(9) and (19) for several flow configurations typical of real hydraulic fracturing jobs.

4.1. Input parameters of test cases

Our goal is to compare the performance and convergence of the modified AA algorithm (Algorithm 2 or MAA) with the standard FP iterations. Comparison is made in terms of the computational time (seconds of CPU time), convergence rate (average number of iterations required for convergence of the non-linear pressure solver per time step) and convergence quality (percentage of time steps, at which the non-linear pressure solver could not converge in 50 iterations). Note that CPU time is provided solely for the purpose of comparing results obtained using different non-linear solvers on a specific computer. Therefore, absolute values of CPU time are meaningless.

In all tests, the computational domain representing a single wing of the hydraulic fracture is set to 100 m × 100 m box. The injection zone is located in the middle of the left vertical boundary and occupies a single mesh cell (its size depends on the mesh resolution and affects the linear fluid velocity scale in the near-wellbore zone). This represents a transverse hydraulic fracture attached to a horizontal well, which is part of the completion of multistage hydraulically fractured wells. These wells are widely used to stimulate production in low-permeability oil and gas reservoirs, including shales. The maximum fracture opening is set to 0.01 m, which is typical for real hydraulic fractures. The channel aperture is described by the function $w(x, y, t)$ as explained below.

For Test 1, which considers the flow in an open channel (Hele–Shaw cell), the aperture remains constant:

$$w(x, y, t) = w_0 = 0.01 \text{ m}. \quad (28)$$

Tests 2–4 examine a suspension flow in the elliptic hydraulic fracture with the tip r_{tip} propagating at a constant velocity v_{tip} :

$$w(x, y, t) = \begin{cases} w_0 \sqrt{1 - r^2 / r_{\text{tip}}^2(t)}, & \text{if } r \leq r_{\text{tip}}(t), \\ w_{\min} = w_0 \epsilon_w, & \text{otherwise} \end{cases}, \quad (29)$$

$$r^2 = (x - x_0)^2 / 4 + (y - y_0)^2, \quad r_{\text{tip}}(t) = r_0 + v_{\text{tip}} t.$$

Here, $x_0 = 0$, $y_0 = 50$ m; the parameter $r_{\text{tip}}(t)$ determines the location of the fracture tip propagating with a constant fracture tip velocity v_{tip} ; in Tests 2, 3, and 4, v_{tip} is set to 0.015 m/s and 0.06 m/s, respectively; $r_0 = 5$ m and $w_0 = 0.01$ m; $w_{\min}$ is the minimum aperture of the computational domain outside of the fracture; ϵ_w is a small regularization parameter that is set to 10^{-3} in the test calculations. Note that according to Eq. (29), the fracture shape at each time instant is a semi-ellipse with an axis ratio of 2:1, elongated horizontally.

In cases of propagating hydraulic fractures, the value of injection rate Q_{in} at each time instant is consistent with the rate of change in the fracture volume Q_{frac} and total fluid leakoff rate Q_l to preserve the volumetric balance. It is formulated as follows:

$$Q_{\text{in}} = Q_{\text{frac}} - Q_l. \quad (30)$$

Eq. (30) is used to calculate the injection rate at each time instant, while the rate of fracture volume change rate is determined by the aperture model described by Eq. (29), and the total leakoff rate is set to 20% of the injection rate. It is important to note that the leakoff is only applied to the area of the computational domain where the fracture is open (i.e., where condition $w > w_{\min}$ is met), and the leakoff distribution along the open fracture is uniform for

simplicity.

The input parameters for Tests 1–4, including fluid and proppant properties and injection schedules, are described in Table 1.

In this section, we have set the values of tuning parameters of MAA algorithm to $m = 20$ (memory) and $d = 0$ (delay) (see description of Algorithm 2 in Appendix B). These values were chosen based on the performance of the MAA algorithm in a parametric study of proppant transport using the developed numerical algorithm. We discuss the sensitivity of the MAA algorithm performance with respect to these parameters in Section 5. Calculations are performed on mesh with resolutions of 32×32 , 64×64 , and 128×128 cells. These resolutions are commonly used by hydraulic fracturing engineers to perform routine calculations in commercial simulators.

Before jumping to description of results, we discuss the choice of input parameters of simulations and their relevance to real hydraulic fracturing jobs.

Input parameters of the test simulations are set specifically to cover the range of variations of parameters (dimensional and non-dimensional ones) typical of real hydraulic fracturing jobs (e.g., see these ranges in the study by Boronin et al. (2015b)). In the test cases, key physical parameters are as follows: fracture aperture scale is 0.01 m, length and height are 100 m, linear velocity at perforations is from 0.1 m/s (early stage of fracture growth in Tests 2 and 3 and 32×32 mesh) up to 15 m/s (Test 4 at 128×128 mesh), viscosity (at perforations) is from 0.001 up to 0.3 Pa·s, fluid density is close to that of water (1000 kg/m^3), proppant density is 2600 kg/m^3 (sand or ceramic proppant); these parameters are consistent with the ranges presented in the study by Boronin et al. (2015b) and other papers describing physical parameters of real fracturing jobs.

The flow is determined by the set of non-dimensional parameters entering the system of Eqs. (7)–(9) and (19). In particular, the effect of yield-stress rheology on the flow (and associated computational difficulty determining the convergence of the iterative algorithm applied to Eq. (19)) is the Bingham number (Bn) representing the ratio of the yield stress to the viscous stress scales. It is estimated to be in the range in between 0 (insignificant yield-stress rheology effect) and 1.5 for slow flows of viscoplastic fluids in hydraulic fractures, where yield stress effect is significant (see the range of non-dimensional parameters typical of hydraulic fracturing in Boronin et al. (2015b)). In our calculations, depending on the mesh resolution determining the size of perforations zone and the linear velocity scale, the Bingham number varies in the range between 0 (Test 2) and 20 (Test 3 at 32×32 mesh), which covers the range described above. The same conclusion applies to the rest of the non-dimensional parameters as we have chosen the flow conditions typical of hydraulic fracturing jobs.

Therefore, the results of test cases are representative and the obtained values of tuning parameters determining the performance of MAA algorithm (i.e., the performance of the developed algorithm) are expected to be suitable for simulations of real fracturing jobs.

4.2. Test 1: displacement of the viscoplastic fluid in the open channel

This test simulates fluid–fluid displacement in an open channel with immobile walls in the absence of proppant particles. As discussed above, a hydrostatic pressure (soft boundary condition) is applied to the right vertical boundary, which is the outlet region of the cell. At the initial time instant, power-law fluid 3 fills the channel and is subsequently displaced by the Herschel–Bulkley fluid 2 and the Newtonian fluid 1 according to the injection schedule outlined in Table 1. Injection occurs in three stages, each lasting 10 min, with a volumetric injection rate of $0.05 \text{ m}^3/\text{s}$.

Table 1
Input data for Tests 1–4.

Group	Parameter	Test 1	Test 2	Test 3	Test 4
Fracture	Tip velocity v_{tip} , m/s	—	0.015	0.015	0.06
Fluid					
1	Flow behaviour index n	0.8	0.6	0.6	0.6
	Consistency index K , $\text{Pa}\cdot\text{s}^n$	0.03	0.7	0.3	0.3
	Yield stress τ_y , Pa	2	0	2	2
	Density ρ , kg/m^3	1.025×10^3	10^3	10^3	10^3
2	Flow behaviour index n	0.5	0.1	0.5	0.5
	Consistency index K , $\text{Pa}\cdot\text{s}^n$	0.3	0.5	0.03	0.03
	Yield stress τ_y , Pa	0	0	0	0
	Density ρ , kg/m^3	10^3	10^3	10^3	10^3
3	Flow behaviour index n	1	0.5	—	—
	Consistency index K , $\text{Pa}\cdot\text{s}^n$	3×10^{-3}	0.1	—	—
	Yield stress τ_y , Pa	0	0	—	—
	Density ρ , kg/m^3	10^3	10^3	—	—
Proppant					
1	Density ρ_p , kg/m^3	—	2.6×10^3	2.6×10^3	2.6×10^3
	Diameter D , m	—	10^{-3}	10^{-3}	10^{-3}
2	Density ρ_p , kg/m^3	—	2.6×10^3	2.6×10^3	2.6×10^3
	Diameter D , m	—	5×10^{-4}	5×10^{-4}	5×10^{-4}
Stage					
1	Fluid/proppant number	1/—	1/1	1/1	1/1
	Volume concentration C_p	—	0.1	0.1	0.1
	Duration, s	600	500	1500	375
2	Fluid/proppant number	2/—	2/2	2/2	2/2
	Vol. concentration C_p	—	0.1	0.1	0.1
	Duration, s	600	500	1500	375
3	Fluid/proppant number	3/—	3/1	—	—
	Volume concentration C_p	—	0.1	—	—
	Duration, s	600	500	—	—
4	Fluid/proppant number	—	1/2	—	—
	Volume concentration C_p	—	0.1	—	—
	Duration, s	—	1500	—	—

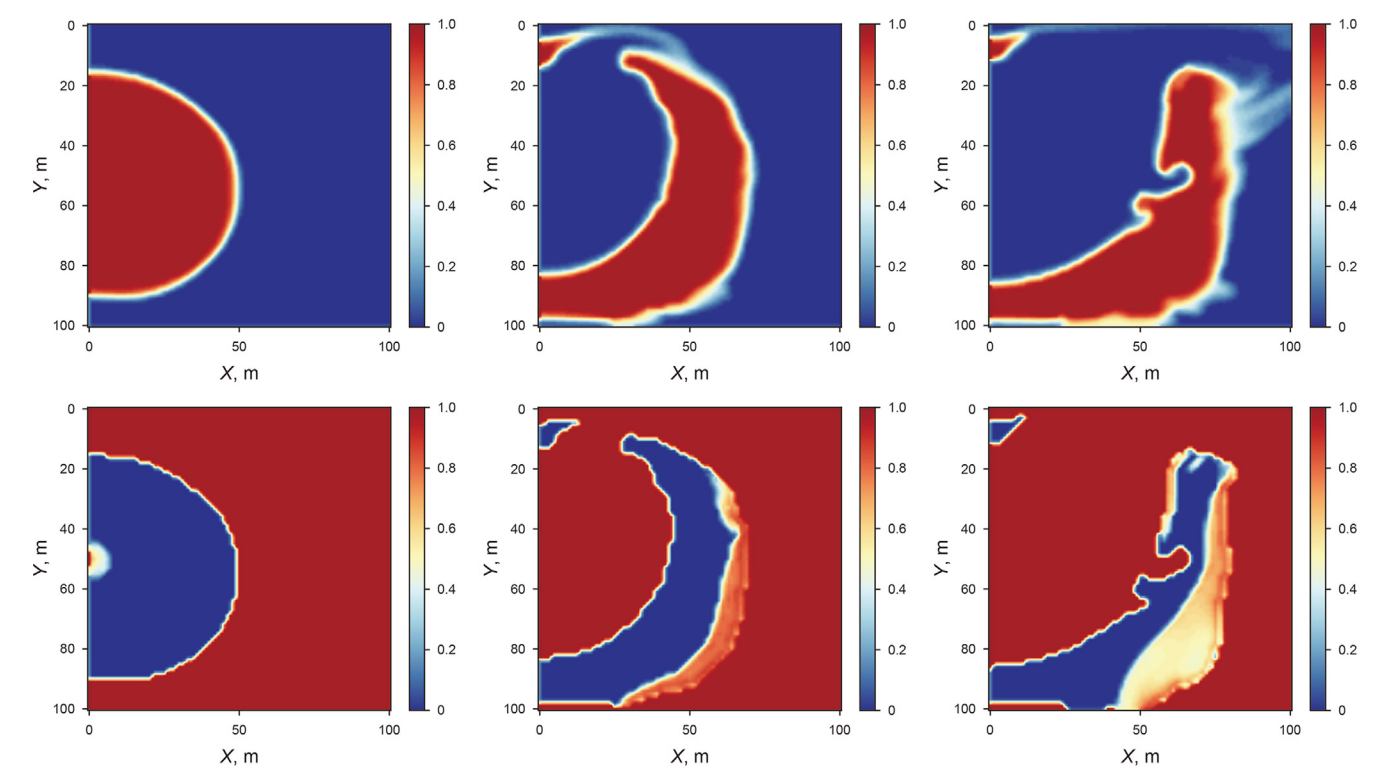


Fig. 2. Distributions of the volume concentration (top row) and the mobility coefficient $Y(\psi)$ (bottom row) (see Eq. (12)) of the viscoplastic fluid 1 in Test 1 obtained using the base FP algorithm at time instants 600 s (left column), 1200 s (middle column) and 1800 s (right column) at the grid with 64×64 cells.

Fluids are injected through a narrow region in the middle of the left vertical boundary, creating a flow configuration that is close to radial near the inlet. This can be observed in the left plots of Figs. 2 and 3. The densities of the fluids do not differ significantly, and the power-law fluid 2 forms a narrow channel in the layer of viscoplastic fluid 1, so the local velocity is high and the viscosity is low. As soon as the channel of fluid 2 is formed in the layer of fluid 1, the latter becomes unyielded. This is shown in the bottom row of plots in Figs. 2 and 3, where unyielded regions correspond to small values of Y .

Table 2 compares the statistics of calculations performed by FP iteration and MAA algorithms. It was found that the FP iteration method diverges in the majority of time steps (more than 98%) and therefore cannot provide the required accuracy for calculations.

Due to the poor convergence of the FP iteration algorithm, the distribution of function $Y(\psi)$ describing the unyielded zones appears unrealistic. This can be observed by comparing the bottom plots in Figs. 2 and 3: (i) during stage 1, when the viscoplastic fluid 1 is injected, the distribution of function $Y(\psi)$ have to be continuous in the area filled with fluid 1 to preserve volumetric balance: as fluids are incompressible and the flow is close to radial-symmetric in the vicinity of the injection zone, the velocity depends on the distance from the perforations only; this behavior is correctly reproduced in simulations carried out using the MAA algorithm, while in simulations carried out using the base FP algorithm, the function $Y(\psi)$ takes minimal value in the majority of area filled with fluid 1 (see bottom left plots in Figs. 2 and 3); (ii) during stage 2, the power-law fluid 2 is injected and triggers the Saffman–Taylor instability at the interface with fluid 1 due to a contrast of viscosities; it breaks through the layer of fluid 1 via a narrow channel, so that the layer of fluid 1 is still moving, and the function $Y(\psi)$ is continuous; while this behavior is reproduced in the simulations carried out using the MAA algorithm, the base FP algorithm

Table 2

The statistics of calculations performed using the base FP and MAA algorithms in Test 1.

Grid resolution	Algorithm	N_{time}	N_p	N_{nc}	N_{eff}	$T_{\text{CPU, s}}$
32×32	MAA	1080	5487	0	5.08	7.0
32×32	FP	1082	80976	1065	74.83	68.7
64×64	MAA	3840	23223	0	6.04	77.9
64×64	FP	3872	290485	3821	75.02	611.4
128×128	MAA	15176	95328	0	6.28	1500
128×128	FP	15106	1136607	14953	75.24	10166.6

N_{time} is the total number of time steps, N_p is the total number of iterations required to solve the pressure equation, N_{nc} is the total number of time steps, at which the non-linear solvers diverged, $N_{\text{eff}} = N_p/N_{\text{time}}$ is the convergence efficiency of the non-linear algorithm (average number of pressure iterations per time step), T_{CPU} is the CPU time required for the calculations.

produces the unyielded layer of fluid 1 with artificial diffusive layer at its leading boundary; (iii) at the end of stage 3, both fluids 2 and 3 break through the layer of fluid 1 and form wide channels, so that fluid 1 is unyielded; while the MAA algorithm provides mostly uniform distribution of function $Y(\psi)$, this is not the case in the simulations made with the FP method (see wide diffusive zone at the leading boundary of the fluid 1 layer).

The base FP algorithm cannot provide a realistic distribution of function Y , and therefore the fluid mobility, when dealing with viscoplastic fluids that are either in motion or unyielded. The FP iteration resulted in poor convergence, leading to inaccurate calculations and unrealistic behavior of the viscoplastic fluid. Additionally, this method required significantly more CPU time for calculations compared to the MAA method (6.8x to 9.8x slower depending on the grid size).

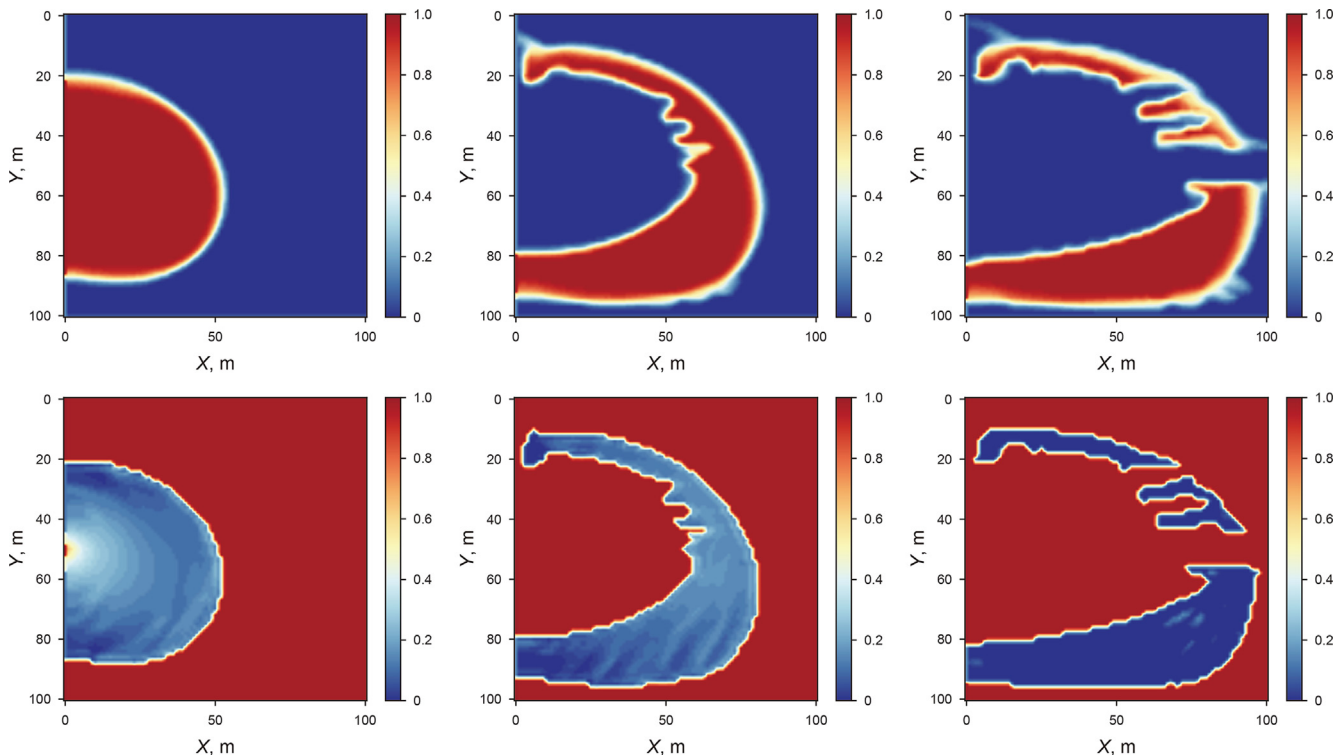


Fig. 3. Distributions of the volume concentration (top row) and the mobility coefficient $Y(\psi)$ (bottom row) (see Eq. (12)) of the viscoplastic fluid 1 in simulations of Test 1 obtained using the MAA algorithm at time instants 600 s (left column), 1200 s (middle column) and 1800 s (right column) on the grid with 64×64 cells.

4.3. Test 2: Injection of power-law fluids with proppant into the propagating elliptic fracture

This test simulates the sequential injection of power-law fluids transporting proppant particles into the propagating elliptic hydraulic fracture. The fluid properties are similar to those of actual hydraulic fracturing fluids (refer to parameters of Test 2 in Table 1). Note that fluid 2 has a relatively small flow behavior index value of 0.1, which significantly complicates the numerical solution of the nonlinear pressure equation (Eq. (19)).

Distributions of proppant concentrations are shown in Fig. 4, while statistics of calculations is summarized in Table 3. We discovered that the MAA algorithm has no impact on the convergence of the base FP iterations. This is due to the small number of iterations needed for convergence of the FP algorithm at each time step, averaging 3–4 iterations per time step. The total number of non-linear iterations and CPU time are nearly identical for both methods. Note that during the calculations carried out at the grid with 128×128 cells, there are several divergent time steps for both the base FP and MAA algorithms. Therefore, we can conclude that in the case of a power-law fluid rheology, the base FP method usually allows obtaining a converged solution. Applying MAA algorithm does not lead to a performance degradation, especially for low resolution grids.

4.4. Test 3: Injection of viscoplastic and power-law fluids with proppant into the propagating elliptic fracture

We consider the sequential injection of viscoplastic and power-law fluids carrying proppant into the propagating elliptic hydraulic fracture. The parameters for Test 3 can be found in Table 1. Expressions for proppant settling velocity (Eq. (10)) and correction for suspension viscosity due to particle volume concentration (Eq. (13)) are valid for power-law fluids, but are applied to the flow of viscoplastic fluid to compare the performance of nonlinear solvers applied to the pressure equation (see Eq. (19)). For more information on accurate modeling of flows of particle-laden viscoplastic fluids, please refer to other sources.

Distributions of proppant concentrations at the end of the simulation are shown in Fig. 5, and the corresponding statistics are summarized in Table 4. As in Test 1, the base FP iteration method fails to produce a converged solution for a large number of time steps (approximately 44% at 32×32 grid, 39% at 64×64 grid and

Table 3

Statistics of calculations made by the base FP and MAA algorithms in Test 2.

Grid resolution	Algorithm	N_{time}	N_p	N_{nc}	N_{eff}	$T_{\text{CPU}}, \text{ s}$
32×32	MAA	834	3051	0	3.65	4.8
32×32	FP	834	3139	0	3.76	4.7
64×64	MAA	2982	8617	0	2.88	67.4
64×64	FP	2982	9436	0	3.16	66.5
128×128	MAA	11942	32654	15	2.73	813.2
128×128	FP	11951	33247	16	2.781	777.6

17% at 128×128 grid). The MAA algorithm achieved convergence at each time step for the 32×32 and 64×64 mesh. On the finer 128×128 mesh, the number of diverged time steps was minimal, accounting for less than 1% of the total time steps taken. This enables the desired accuracy of calculations to be maintained while achieving a significant speed-up of simulations compared to the FP iteration algorithm. Specifically, at the grids 32×32 , 64×64 and 128×128 , the speed-up is 6x, 7.3x, and 2.2x, respectively.

4.5. Test 4: Injection of viscoplastic and power-law fluids with proppant into the elliptical hydraulic fracture at a higher tip velocity

This test replicates the flow configuration, fluids, and proppants used in Test 3. The only difference is that the fracture tip velocity is four times larger, with a velocity of 0.06 m/s compared to 0.015 m/s. The duration of the stages was reduced in proportion to the increased tip velocity to maintain the final dimensions of the fracture (refer to parameters of Test 4 in Table 1). It is important to note that an increase in the fracture tip velocity, as considered in the current test compared to Tests 2 and 3, results in an increase in the injection and leakoff rates according to balance condition described by Eq. (30). This, in turn, leads to a decrease in the apparent fluid viscosity and an increase in the proppant settling velocity.

The final distributions of the proppant concentrations are shown in Fig. 6. There is a noticeable increase in the area filled with settled proppant 2, which can be attributed to the decrease in viscosity of the power fluid at higher velocities. Statistics of calculations is summarized in Table 5. In this test, the MAA resulted in 2.5x, 8.7x and 3.1x acceleration of computations made at 32×32 , 64×64 and 128×128 grids as compared to those carried out using the base FP algorithm, respectively. There are fewer diverged time

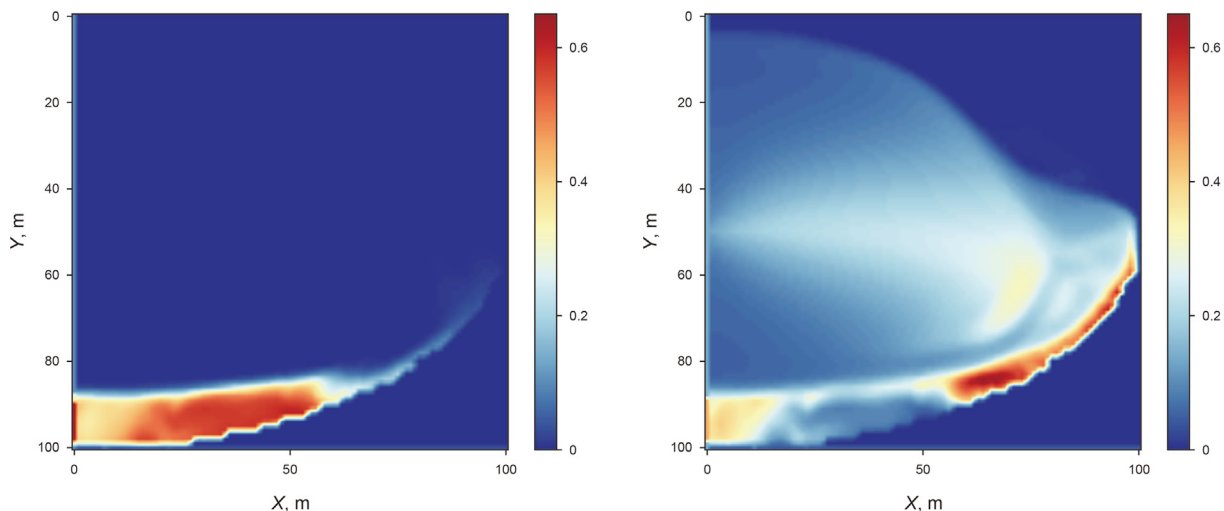


Fig. 4. Volume concentration of proppant 1 (left) and 2 (right) obtained in simulations of Test 2 at $t = 3000 \text{ s}$ using the MAA algorithm. The grid resolution is 64×64 .

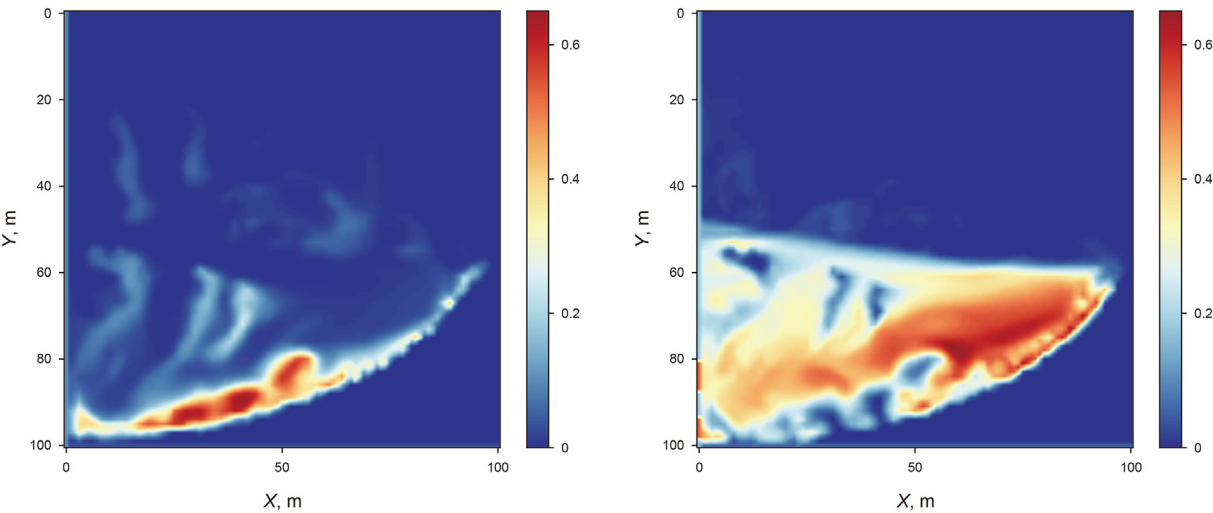


Fig. 5. Volume concentration of proppant 1 (left) and 2 (right) obtained in simulations of Test 3 at $t = 3000$ s using the MAA algorithm. The grid resolution is 64×64 .

Table 4
Statistics of calculations made by the base FP and MAA algorithms in Test 3.

Grid resolution	Algorithm	N_{time}	N_p	N_{nc}	N_{eff}	$T_{\text{CPU}}, \text{ s}$
32×32	MAA	1113	5056	0	4.54	8.1
32×32	FP	1072	39191	475	36.55	48.9
64×64	MAA	3516	9707	0	2.76	45.8
64×64	FP	3286	103116	1281	31.38	336.0
128×128	MAA	12449	41664	108	3.34	1428.6
128×128	FP	13238	195605	2210	14.77	3102.2

steps of the base FP iteration compared to the previous test, which we attribute to a smaller viscosity contrast in the flow domain due to a larger propagation velocity of the fracture.

5. Sensitivity of simulations with respect to values of tuning parameters of the MAA algorithm

The AA algorithm includes several tuning parameters, namely, memory size m and delay parameter d (see formal description of the base and modified versions of algorithms in Appendix B). This

Table 5
Statistics of calculations made by the base FP and MAA algorithms in Test 4.

Grid resolution	Algorithm	N_{time}	N_p	N_{nc}	N_{eff}	$T_{\text{CPU}}, \text{ s}$
32×32	MAA	750	3037	0	4.04	5.2
32×32	FP	750	10749	108	14.33	13.1
64×64	MAA	2840	9936	2	3.49	80.1
64×64	FP	2846	125643	1595	44.14	694.8
128×128	MAA	11197	38200	63	3.41	1226.3
128×128	FP	11165	223767	2653	20.04	3789.0

section analyzes the sensitivity of the performance of AA algorithms applied to Tests 1, 3 and 4 with respect to the values of m and d . Calculations are carried out at the mesh resolution of 64×64 . Test 2 (injection of particle-laden power-law fluids) is not considered in this case since the AA algorithm performs similar to the base FP iteration. This is due to the very fast convergence of the FP algorithm, as discussed in Section 4.2. The performance of the AA algorithm is also studied in different configurations: (i) AA is applied to the linearized operator $\mathbf{A}$, the solution guess $\mathbf{x}$ and the right-hand side $\mathbf{b}$, which is the MAA algorithm (see Algorithm 2 in Appendix B); (ii) AA is applied to the matrix $\mathbf{A}$ and the right-hand

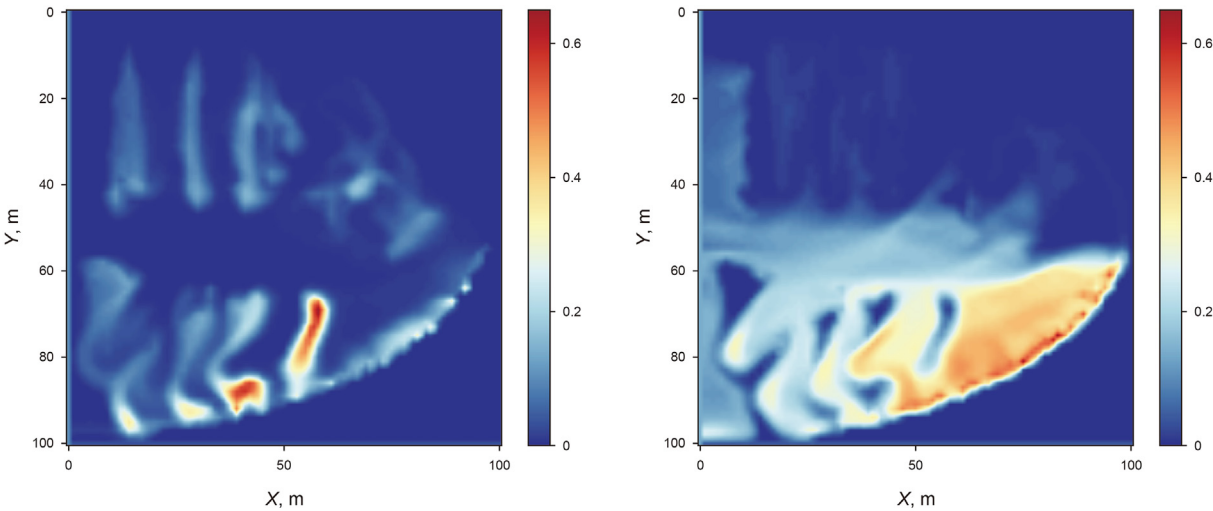


Fig. 6. Volume concentration of proppant 1 (left) and 2 (right) obtained in simulations of Test 4 at $t = 750$ s using the MAA algorithm at the grid with 64×64 resolution.

side **b**; (iii) AA is applied to the solution guess x only, which is the base algorithm (see Algorithm 1 in Appendix B). In all figures shown in this section, the dashed line represents the value obtained using the FP algorithm.

The results indicate that the performance of the MAA algorithm is significantly lower than that of the FP algorithm in simulations of Test 1 with non-zero delay ($d > 0$) and large memory $m > 3$ (see Fig. 7, where the computational time of the FP algorithm is shown by the dashed line). We address it to a rapid rate of divergence of the FP algorithm in case of the strong non-linearity of the pressure equation due to a viscoplastic rheology of fracturing fluids: at each time step, even few iterations made by the FP algorithm lead to large residuals, significantly increasing the number of consequent iterations required to obtain the converged solution.

During the calculations, time steps are determined by the CFL stability condition due to an explicit approximation of hyperbolic transport equations, so that they are calculated based on the velocity field and a fixed value of the Courant number. Therefore, the total number of time steps should be identical in the calculations carried out using all algorithms under consideration provided the tolerance is maintained small (i.e., when the solution is converged). This is confirmed by the results of calculations (see Fig. 8).

Each iteration made during the solution of a non-linear pressure equation requires the solution of a corresponding linear problem as discussed in the introduction, which is made using the blackbox multigrid algorithm (Muravleva et al., 2021). The MAA algorithm has the potential to reduce the total number of iterations in a global (non-linear) algorithm and the number of iterations made by an internal linear solver. The former can be reduced due to a modification in the linearized matrix and the right-hand side, while the latter can be reduced by providing a better initial guess to the solution at each iteration of the outer “non-linear” algorithm. The results show that the MAA algorithm significantly reduces the number of iterations in the outer “non-linear” algorithm. Typically, only one iteration of the linear solver is required per iteration of the “non-linear” solver, indicating high performance of the blackbox multigrid algorithm (see Figs. 9 and 10). The only exception is Test 1, where for small values of memory m or large delay d , the number of iterations performed by the linear solver per iteration of the MAA algorithm is larger than that of the FP algorithm.

Next we analyze the convergence rate of the developed MAA algorithm. As previously mentioned, the performance of the numerical algorithm for solving the non-linear pressure equation that describes the flow of power-law fluids is comparable between the

standard FP and the MAA algorithms. Therefore, both algorithms converge in the vast majority of time steps in Test 2. However, this is not the case for calculations performed during Tests 1, 3, and 4. Test 1 shows the worst convergence efficiency with the dependency of the number of divergent time steps on the values of the tuning parameters m and d of the MAA algorithm (see Fig. 11). The worst convergence efficiency is observed in Test 1. The best convergence of the developed MAA algorithm is achieved at $d = 0$ (zero delay, so that AA is applied at the start of each time step) and $m \geq 20$ (sufficiently large memory). If we consider the additional calculations required due to an increase in the memory parameter and the associated increase in CPU time (refer to Fig. 7), $m = 20$ is the optimal value.

Above we discussed the MAA algorithm, which modifies not only the initial guess, but also the linearized matrix and the right-hand side at each iteration, in comparison to the base FP algorithm. The performance of the AA algorithm was analyzed for different configurations, including the conventional AA algorithm (which modifies only the initial guess at each iteration) and an alternative algorithm that modifies the linearized matrix and the right-hand side of the pressure equation (excluding the initial guess from Algorithm 2). The overall performance of these algorithms is shown in Figs. 12–15.

First, we analyze the performance of the variant of the AA algorithm that deals with the initial guess only (base AA algorithm described by Algorithm 1 in Appendix B). The number of non-linear iterations in the considered version of the AA algorithm is similar to those obtained using the base FP algorithm (see Fig. 16), as the numerical procedures producing the matrix of the linearized pressure equation and its right-hand side are similar in both algorithms. Therefore, the base AA algorithm cannot produce a converged solution for the pressure equation that describes a flow of viscoplastic fluids. Note that the AA algorithm results in additional overhead compared to the base FP method, which increases with an increase in the memory parameter m . The number of iterations made by the linear solver cannot be reduced, as discussed above. Therefore, applying this variant of the AA algorithm only slows down the computations. Nevertheless, we obtained a certain decrease in the average number of iterations performed by the linear solver per iteration of the Algorithm 1 as compared to that obtained using the Algorithm 2 (compare Figs. 10 and 17). This effect is most noticeable in Test 1 for a non-zero delay and sufficiently large memory.

Next we analyze the performance of the AA algorithm based on

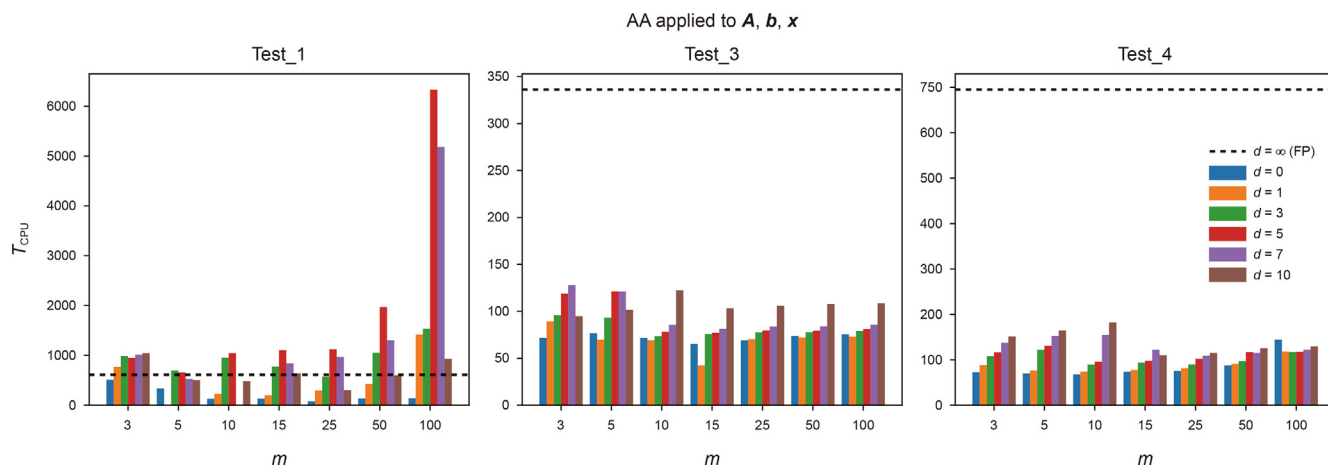


Fig. 7. CPU time of calculations T_{CPU} (in s) in Tests 1, 3 and 4 depending on values of tuning parameters m and d ; calculations are made using the MAA algorithm described in Appendix B (Algorithm 2).

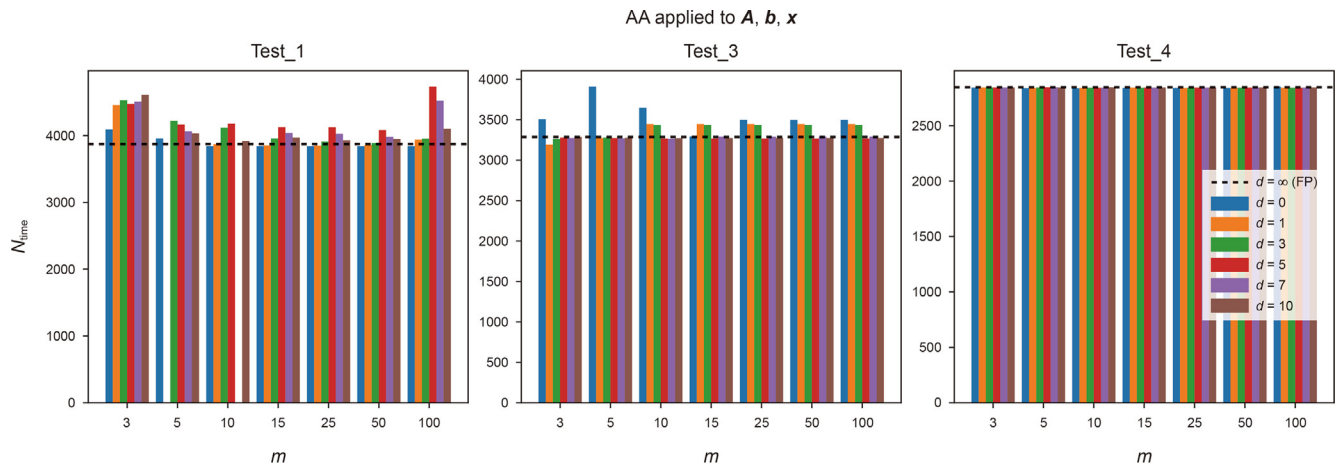


Fig. 8. Number of time steps N_{time} in Tests 1, 3 and 4 depending on values of tuning parameters m and d ; calculations are made using the MAA algorithm described in [Appendix B \(Algorithm 2\)](#).

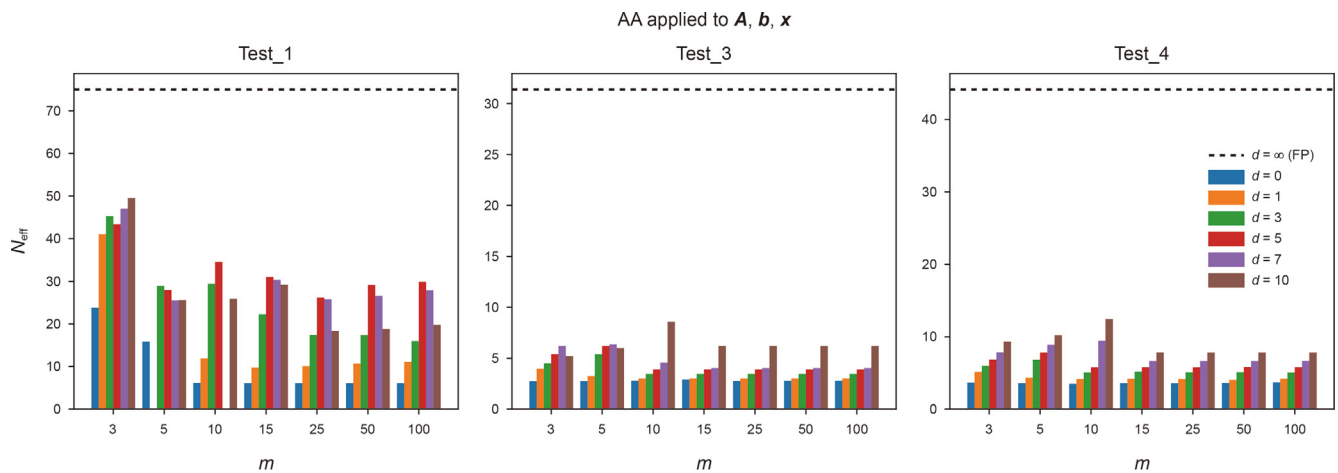


Fig. 9. Average number of non-linear solver iterations per time step N_{eff} in Tests 1, 3 and 4 depending on parameters m and d , MAA algorithm is described in [Appendix B \(Algorithm 2\)](#).

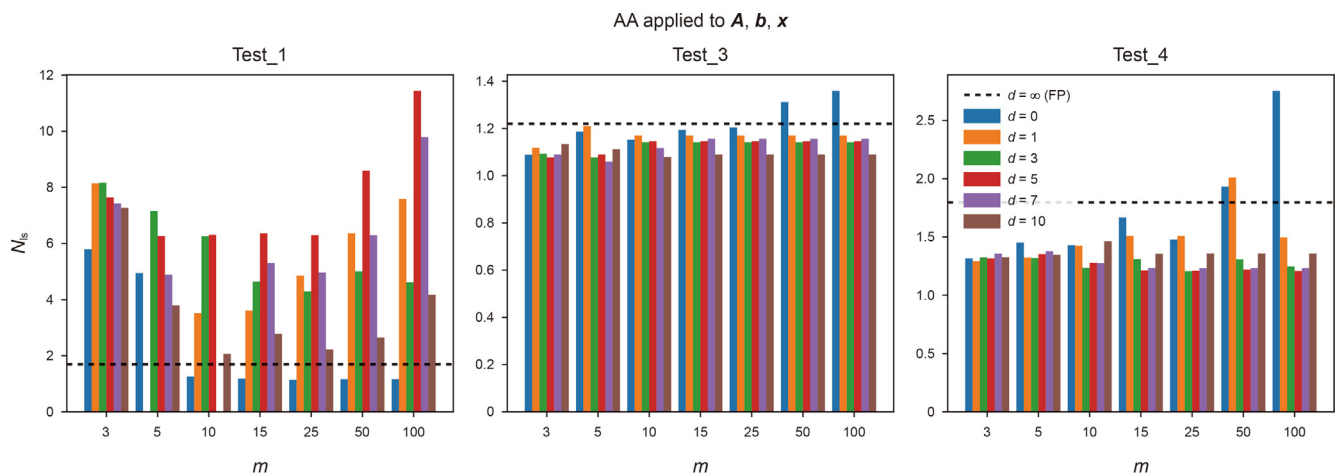


Fig. 10. Average number of iterations made by the linear solver (blackbox multigrid) per iteration of the non-linear solver N_{ls} in Tests 1, 3 and 4 depending on parameters m and d , MAA algorithm is described in [Appendix B \(Algorithm 2\)](#).

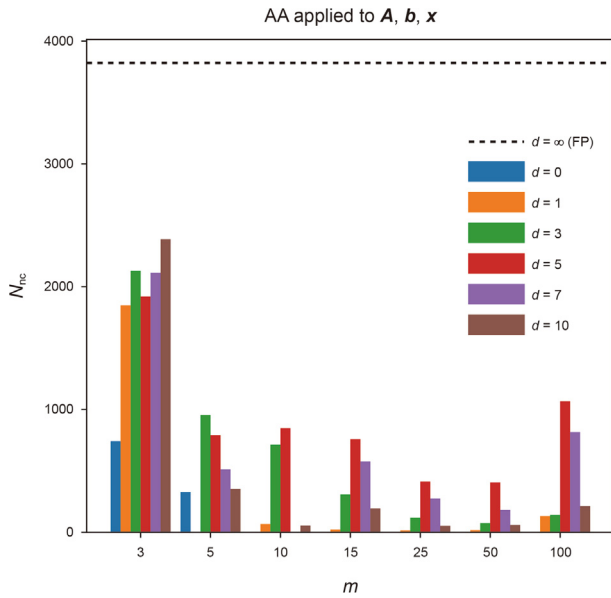


Fig. 11. Number of time steps, at which the developed AA algorithm diverged N_{nc} (i.e., convergence criterion is not met in 50 iterations) in Test 1.

the optimization of a vector composed of the matrix describing linearized pressure equation and the right-hand side at each iteration, while the initial guess is ignored in the optimization procedure. Comparing Figs. 7 and 12, 9 and 13, as well as 10 and 14, we conclude that this version of the algorithm shows slightly better performance as compared to Algorithm 2. We attribute this improvement to the efficient linear solver, which typically converges in a single iteration regardless of the initial guess. To avoid over-fitting, we recommend using a more general version of the AA algorithm (Algorithm 2) to solve non-linear problems described by quasi-linear elliptic equations.

6. Estimation of algorithm complexity and comparison with Jacobian-based methods

The time and memory complexity of the entire algorithm is dependent on the convergence rate. Nevertheless, the time and memory complexity of the additional computations for a single iteration of the AA algorithm (i.e., the overhead introduced over the base FP iteration) can be estimated. The algorithm's memory complexity is $O(m[N_x N_y]^2)$ when the optimization vector includes the linearized pressure equation $\mathbf{A}$, and $O(mN_x N_y)$ otherwise, where $N_x N_y$ is the total number of mesh cells and m is the memory

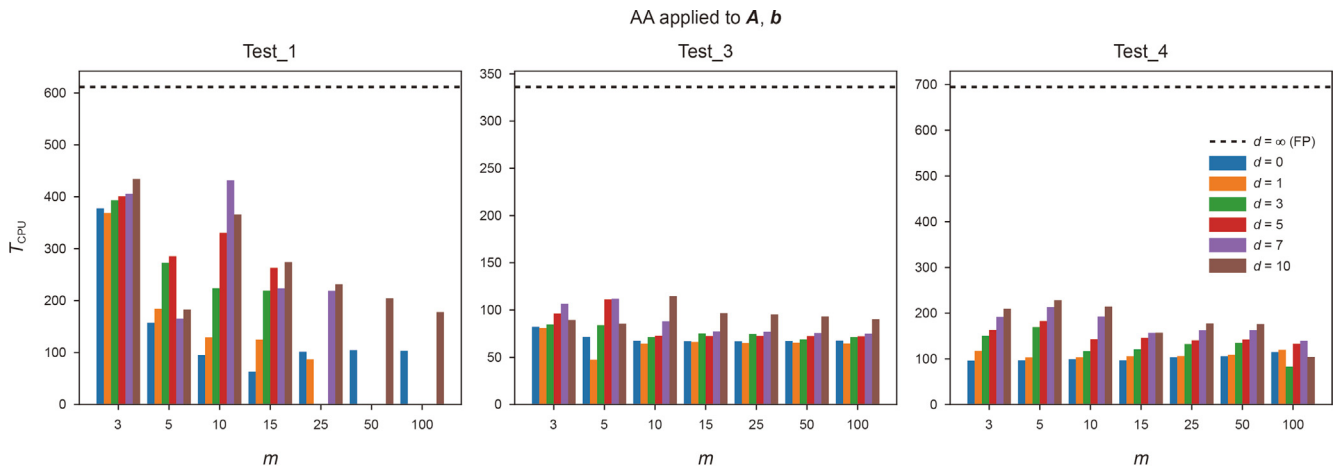


Fig. 12. CPU time of calculations T_{CPU} (in s) for Tests 1, 3 and 4 depending on parameters m and d , AA is applied to linearized pressure equation $\mathbf{A}$ and its right-hand side $\mathbf{b}$.

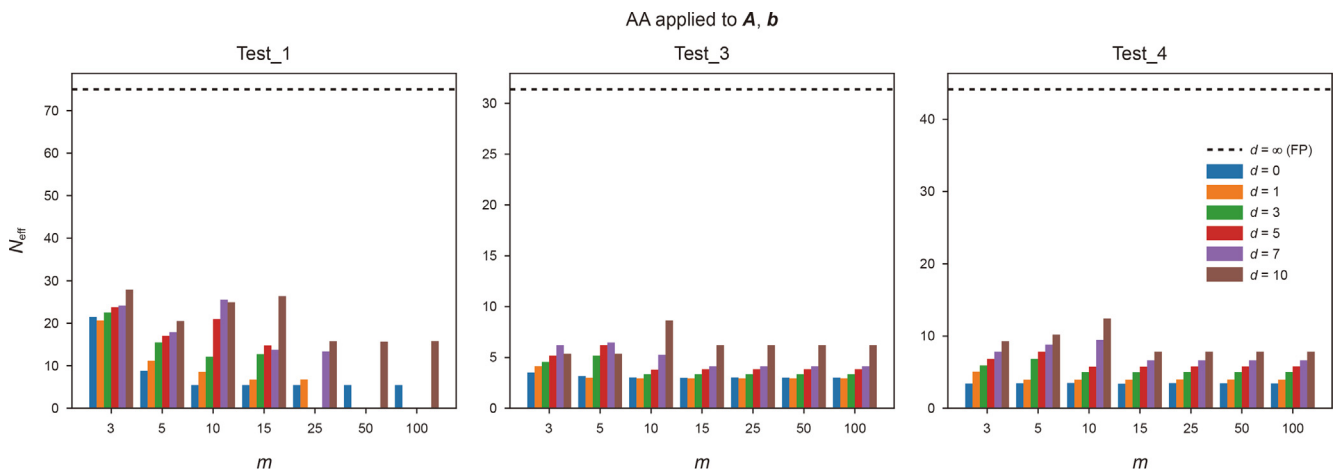


Fig. 13. Average number of non-linear solver iterations per time step N_{eff} for Tests 1, 3 and 4 depending on parameters m and d , AA is applied to linearized pressure equation $\mathbf{A}$ and its right-hand side $\mathbf{b}$.

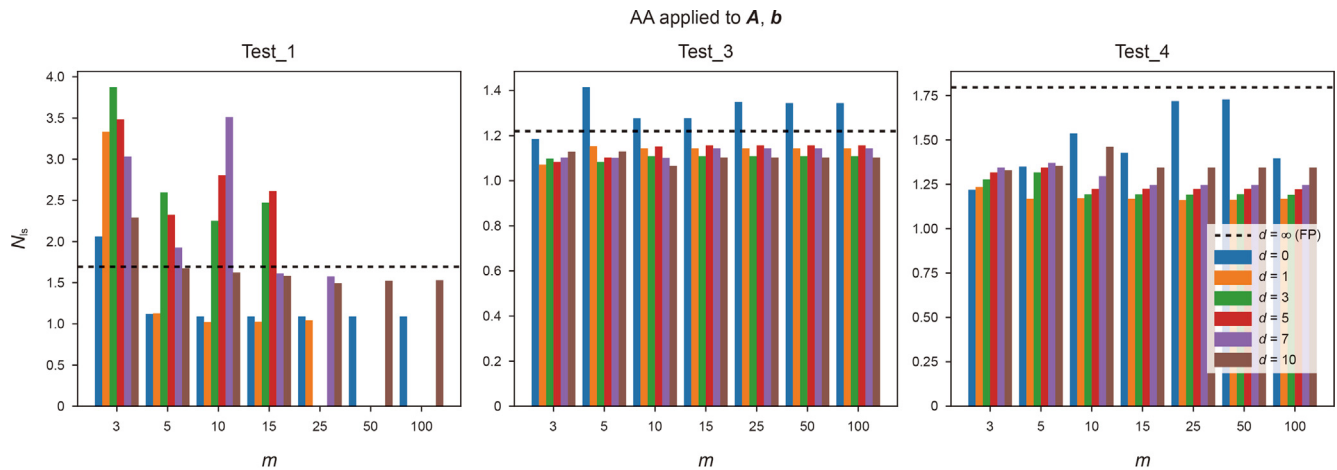


Fig. 14. Average number of linear solver (blackbox multigrid) iterations per iteration of non-linear solver N_{ls} for Tests 1, 3 and 4 depending on parameters m and d , AA is applied to linearized pressure equation $\mathbf{A}$ and its right-hand side $\mathbf{b}$.

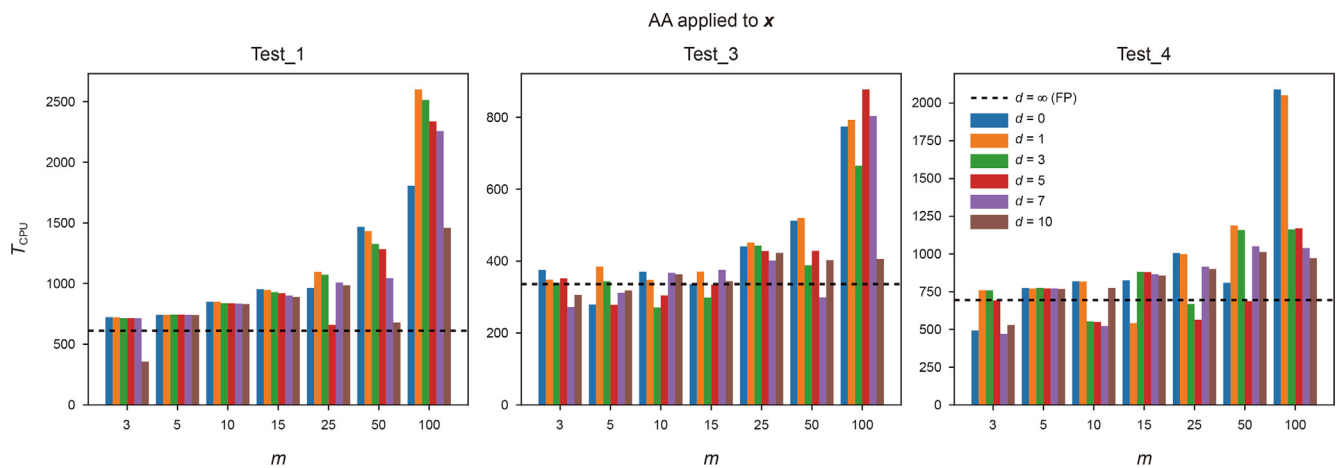


Fig. 15. CPU time of calculations T_{CPU} (in s) for Tests 1, 3 and 4 depending on parameters m and d , AA is applied to solution guess $\mathbf{x}$ (base AA algorithm as described by Algorithm 1 in Appendix B).

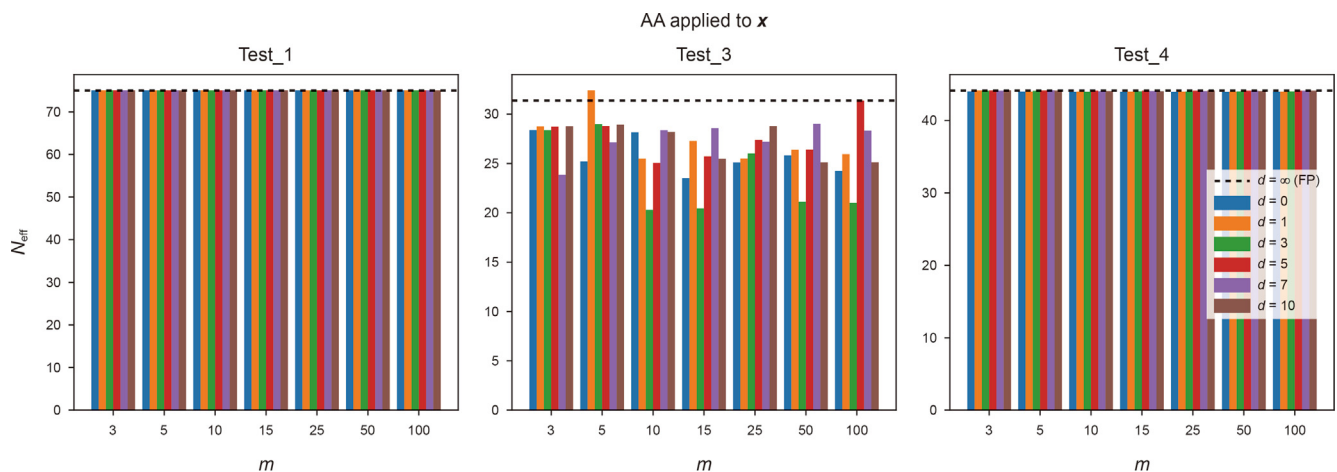


Fig. 16. Average number of non-linear solver iterations per time step N_{eff} for Tests 1, 3 and 4 depending on parameters m and d , AA is applied to solution guess $\mathbf{x}$ (base AA algorithm as described by Algorithm 1 in Appendix B).

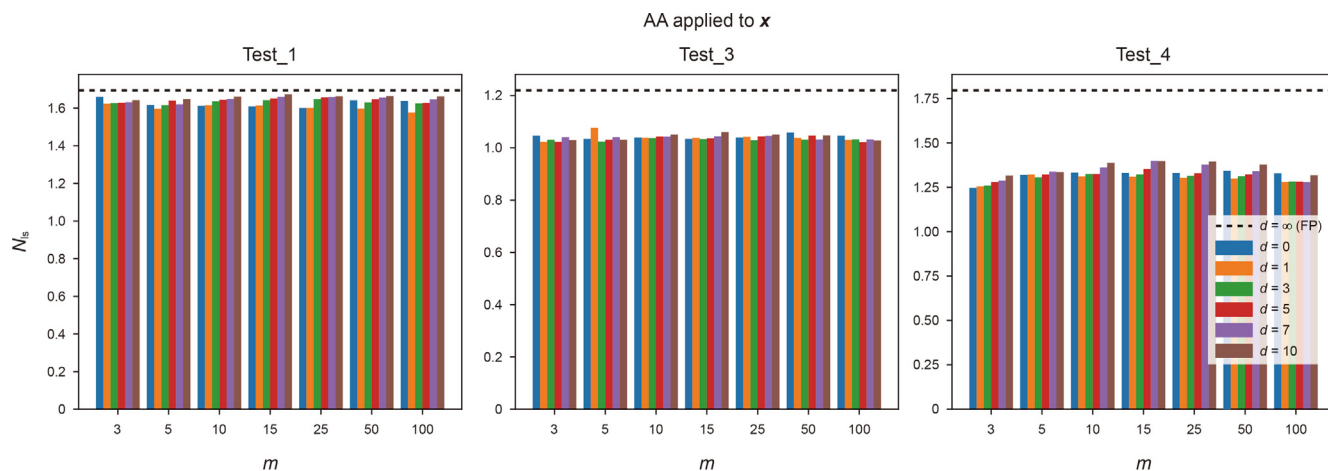


Fig. 17. Average number of linear solver (blackbox multigrid) iterations per iteration of non-linear solver N_{ls} for Tests 1, 3 and 4 depending on parameters m and d , AA is applied to solution guess $\mathbf{x}$ (base AA algorithm as described by Algorithm 1 in Appendix B).

parameter. The time complexity is determined by solving a minimization problem and is either $O(m^2[N_x N_y]^2)$ or $O(m^2 N_x N_y)$, depending on the algorithm version. Therefore, even for the most computationally intensive version of the algorithm, time and memory overhead is estimated to be less than that of a single FP iteration applied to the quasi-linear pressure equation.

We now estimate the complexity of Jacobian-based algorithms (e.g., Newton–Raphson, NR) applied to the problem under consideration as their typical convergence rate is faster as compared to that of the family of FP algorithms. We consider the standard 5-point “cross” stencil and central differences to discretize the pressure equation (Eq. (19)). Our test calculations showed that the numerical resolution of unyielded zones requires accurate approximation of the function $Y(\psi)$ (see Eq. (12)), which is a function of the local pressure gradient F determining the formation of unyielded zones and entering the fluid mobility coefficients in the elliptic equation. This requires approximation of the pressure gradient at larger (as compared to standard “cross”) stencil, namely, 5-point stencil to approximate the pressure derivatives in each direction (see Appendix A). As a result, the Jacobi matrix required to implement NR method has 25 non-zero diagonals, so that the corresponding solution of linear system requires larger number of mathematical operations as compared to that of a single FP iteration at least by a factor of 5 (see details in Appendix C).

Note that this estimate is actually a lower bound of the NR algorithm overhead as compared to the calculations made in the framework of the developed AA method. The reason is that the blackbox multigrid linear solver we use in course of our study is developed specifically for 5-diagonal matrices and is shown to be substantially (typically, by a factor of several dozens) faster as compared to other existing linear solvers including the family of Algebraic Multigrids (see Muravleva et al. (2021)). Therefore, the overhead of calculations in course of NR algorithm is expected to be significantly larger due less efficient linear solvers suitable for 25-diagonal matrices. As discussed in Section 4, the developed MAA algorithm requires from 3 to 6 iterations to converge per time step (see values of parameter N_{eff} in Tables 2, 4 and 5), and it does not significantly increase the CPU time of computations performed during a single iteration as compared to that of base FP algorithm. Therefore, the Jacobian-based algorithms must converge in a single iteration per time step to compete in performance with the developed AA algorithms, which is quite unrealistic due to essential non-linearity of the problem under consideration.

7. Summary and conclusions

On the basis of AA algorithm we developed a new efficient solver to the non-linear pressure equation describing a flow of particle-laden viscoplastic fluids in a hydraulic fracture. In contrast to the standard AA method, the developed MAA algorithm optimizes the vector of unknowns at each iteration based not only on the history of approximate solutions calculated at several previous iterations, but also on the combined vector formed by the vector of unknowns, the matrix approximating the non-linear operator, and the right-hand side. MAA is implemented in the framework of the coupling strategy of solving the system of governing equations describing a particle-laden flow in a hydraulic fracture similar to the well-known IMPES method. Reliable convergence of the overall algorithm was found to require a certain limitation of the time step variation during the iterative procedure.

The convergence and performance of the MAA algorithm is compared against: (i) the base FP algorithm; (ii) standard AA algorithm applied to the solution guess at each iteration; (iii) the alternative AA algorithm applied to the linearized pressure equation and the right-hand side. The set of test calculations considered in the current study covers injection scenarios typical of real hydraulic fracturing operations. It includes the injection of particle-free and particle-laden power-law and viscoplastic fluids into a plane channel with steady walls and propagating hydraulic fractures, so that the obtained results are reliable and can be used in the engineering practice.

Simulation results show that both the FP and proposed AA algorithms reliably converge in simulations of the injection of power-law fluids with zero yield stress. The difference in their performance is insignificant due to the small number of pressure iterations made per time step. However, in simulations of viscous fluid flows, the FP and standard AA algorithms showed significant instability in calculations of injection of viscoplastic fluids. They diverged at the majority of time steps due to the presence of unyielded zones, such as those formed by the development of the Saffman-Taylor instability (viscous fingers). As a result, these algorithms cannot be used to solve the non-linear elliptic equation that describes the injection of viscoplastic fluids into a hydraulic fracture, as the solution is shown to be physically unrealistic. The MAA method proposed in this study allows for obtaining a converged solution. This results in a physically realistic distribution of unyielded zones in a hydraulic fracture during unstable fluid-

fluid displacement.

A sensitivity study of the performance of the considered AA algorithms was carried out, and it was shown that the best performance and convergence in the simulation of suspension flow in a narrow channel is obtained when the memory parameter is large enough (above 20), while the delay parameter is set to zero (so that the MAA algorithm is applied starting from the first iteration at each time step).

We have analyzed the computational complexity of Jacobian-based methods in course of discretization of quasi-linear elliptic equation considered in the current study. Due to significant overhead in mathematical operations per time step of the Jacobian-based algorithms as compared to the family of FP algorithms, the former ones must converge in a single iteration per time step to compete in performance with the MAA algorithm, which is quite unrealistic due to essential non-linearity of the problem under consideration due to complex fluid rheology.

Input parameters of test cases considered in the current study cover the range of variation of key physical parameters (injection rates, fracture shape, fluid rheology properties) determining the suspension flow in hydraulic fractures. While we consider only fluid mechanics problem, the MAA algorithm is expected to be applicable to solution of the full fracture propagation problem as soon as the proper explicit coupling of the solution of equations is implemented. Therefore, the MAA algorithm and optimal values of its tuning parameters are expected to perform well in simulations of hydraulic fracturing process.

The developed MAA algorithm can be applied to a numerical solution of non-linear problems described by general quasi-linear elliptic (or parabolic) equation (see Eq. (1)), which describes a wide range of oilfield processes including hydraulic fracturing, well cementing, fracture flowback and reservoir production; we expect that it is specifically effective in solutions of essentially non-linear problems, for example, those corresponding to flows of viscoplastic fluids.

CRedit authorship contribution statement

D.Yu. Derbyshev: Writing – review & editing, Writing – original draft, Validation, Software, Investigation, Formal analysis, Data curation. **S.A. Boronin:** Writing – review & editing, Writing – original draft, Validation, Supervision, Data curation, Conceptualization. **G.V. Ovchinnikov:** Methodology. **A.A. Osipov:** Writing – original draft, Supervision, Resources, Project administration, Funding acquisition, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work received partial financial support from Gazpromneft Science and Technology Center.

Appendix A. Approximation of particle transport equations

We consider a finite-difference approximation of Eqs. (7)–(9) and (19) using a uniform rectangular staggered grid. The parabolic advection equations for fluid tracer and particle volume concentrations are approximated using a second-order TVD flux-limiting scheme, which is not shown here for brevity of notation.

As we are concerned with the numerical solution of parabolic

Eq. (19), below we present details of its approximation, which is considered in the framework of current study. We reformulate Eq. (19) and its boundary conditions below for the convenience of the reader:

$$\frac{\partial}{\partial x} \left(wK \frac{\partial p}{\partial x} \right) + \frac{\partial}{\partial y} \left(wK \left[\frac{\partial p}{\partial y} + \text{Bu} \rho_m \right] \right) = \frac{\partial w}{\partial t} + 2v_l - \frac{\partial w C_p v_s}{\partial y}. \quad (31)$$

$$x = 0 : y \in I_{\text{in}} : u_f = u_{\text{in}}, \quad y \notin I_{\text{in}} : u_f = 0; \quad (32)$$

$$x \in [0, 1], \quad y = 0, \quad h : w_f = 0; \quad (33)$$

$$x = 1, \quad y \in [0, h] : p = p_{\text{out}}(y) \quad \text{or} \quad u_f = 0. \quad (34)$$

$$u_{\text{in}} = \frac{Q_{\text{in}}}{w_{\text{in}} H_{\text{in}}}, \quad p_{\text{out}}(y) = -\text{Bu} \int_y^h \rho_m(y') dy'.$$

Here, $K = w^2 Y(\psi) / (12 \mu_{\text{eff}})$ is the fluid mobility; u_f and v_f are the horizontal and vertical components of the fluid velocity, respectively; u_{in} is the fluid velocity at the perforations calculated using the prescribed volumetric injection rate Q_{in} ; H_{in} is the height of the perforation interval; w_{in} is the fracture aperture in the vicinity of the perforated interval; I_{in} is the interval of the perforations (inlet region); h is the non-dimensional height of the computational domain. Boundary conditions described above (Eqs. (32)–(34)) correspond to two flow configurations, namely, an open channel with a “soft” pressure boundary condition at the outlet p_{out} (in-situ hydrostatic pressure at the right vertical boundary) and a hydraulic fracture immersed into a rectangular computational domain with zero flux through the right vertical boundary. Note that in the latter flow configuration, flux condition is imposed at all boundaries of the computational domain (Neumann problem), so that a pressure needs to be set in one of the grid nodes to obtain a unique solution. Since multiphase flow is considered, K is a function of spatial coordinates (x, y) and varies rapidly in the vicinity of interfaces of fluids with viscosity contrast, and we have moved to the right-hand side of Eq. (19) all terms that do not explicitly depend on the fluid pressure p .

The computational domain $(x, y) \in [0, 1] \times [0, H/L]$ is approximated by the staggered rectangular mesh Ω with the mesh nodes introduced as follows:

$$\Omega = \{ (x_i, y_j), \quad x_i = ih_x, \quad y_j = jh_y, \quad i \in [0, N_x], \quad j \in [0, N_y] \}, \quad (35)$$

where $h_x = 1/N_x$ and $h_y = H/(LN_y)$ are dimensions of mesh cells in horizontal and vertical directions, respectively. Note that according to Eq. (35), there are mesh nodes located at the boundaries of the computational domain. The fluid pressure p , the volume concentrations C_i , C_p , the density ρ_m and the viscosity μ_{eff} of the mixture are approximated at the mesh nodes (x_i, y_j) , while the velocities are approximated at the cell faces (semi-integer nodes): the horizontal velocities are approximated at $(x_{i \pm 0.5}, y_j)$ and vertical ones at $(x_i, y_{j \pm 0.5})$, where $x_{i \pm 0.5} = ih_x \pm 0.5h_x$, $y_{j \pm 0.5} = jh_y \pm 0.5h_y$.

To approximate Eq. (31), a standard 5-point “cross” mesh is used. In the internal nodes $(i, j) \in [1, N_x - 1] \times [1, N_y - 1]$, the approximation scheme at a given time step is as follows:

$$\begin{aligned}
& (wK)_{i+0.5j}^n \frac{p_{i+1j}^{n+1} - p_{ij}^{n+1}}{h_x^2} - (wK)_{i-0.5j}^n \frac{p_{ij}^{n+1} - p_{i-1j}^{n+1}}{h_x^2} \\
& + (wK)_{ij+0.5}^n \frac{p_{ij+1}^{n+1} - p_{ij}^{n+1}}{h_y^2} - (wK)_{ij-0.5}^n \frac{p_{ij}^{n+1} - p_{ij-1}^{n+1}}{h_y^2} \\
& = -Bu \frac{(K\rho_m)_{ij+0.5}^n - (K\rho_m)_{ij-0.5}^n}{h_y} + \frac{w_{ij}^{n+1} - w_{ij}^n}{\Delta t} \\
& + 2v_{l,ij} - \frac{(wC_p\nu_s)_{ij+0.5}^n - (wC_p\nu_s)_{ij-0.5}^n}{h_y},
\end{aligned} \quad (36(a))$$

$$(wK)_{i\pm 0.5j}^n = \frac{1}{4} (w_{ij}^n + w_{i\pm 1j}^n) (K_{ij}^n + K_{i\pm 1j}^n), \quad (36(b))$$

$$(wK)_{ij\pm 0.5}^n = \frac{1}{4} (w_{ij}^n + w_{ij\pm 1}^n) (K_{ij}^n + K_{ij\pm 1}^n). \quad (36(c))$$

As described above, the fluid mobility K contains the function $Y(\psi)$ (see Eq. (12)), which is approximated at the mesh nodes using a 5-point stencil as shown:

$$\psi_{ij} = \frac{Bn \tau_{m,ij}}{w_{ij} F_{ij}}, \quad F_{ij} = \sqrt{\left(\frac{\partial p}{\partial x}\right)_{ij}^2 + \left(\frac{\partial p}{\partial y}\right)_{ij}^2}, \quad \left(\frac{\partial p}{\partial x}\right)_{ij} = \frac{1}{h_x} \mathbf{l}_{ij}^{\text{ap}} \cdot \mathbf{p}_{ij}^{\text{st}}. \quad (37)$$

Here, approximation stencil $\mathbf{p}_{ij}^{\text{st}}$ and approximation coefficients $\mathbf{l}_{ij}^{\text{ap}}$ depend on the mesh node (i, j) , at which the approximation to the pressure derivative is considered and described as follows:

$$\mathbf{l}_{ij}^{\text{ap}} = \begin{cases} \left\{ \frac{1}{12}, -\frac{2}{3}, 0, \frac{2}{3}, -\frac{1}{12} \right\}, & 2 \leq i \leq N_x - 2; \\ \left\{ -\frac{1}{4}, -\frac{5}{6}, \frac{3}{2}, -\frac{1}{2}, \frac{1}{12} \right\}, & i = 1; \\ \left\{ -\frac{1}{12}, \frac{1}{2}, -\frac{3}{2}, \frac{5}{6}, \frac{1}{4} \right\}, & i = N_x - 1; \\ \left\{ -\frac{25}{12}, 4, -3, \frac{4}{3}, -\frac{1}{4} \right\}, & i = 0; \\ \left\{ \frac{1}{4}, -\frac{4}{3}, 3, -4, \frac{25}{12} \right\}, & i = N_x. \end{cases} \quad (38)$$

$$\mathbf{p}_{ij}^{\text{st}} = \begin{cases} \{p_{i-2j}, p_{i-1j}, p_{ij}, p_{i+1j}, p_{i+2j}\}, & 2 \leq i \leq N_x - 2; \\ \{p_{i-1j}, p_{ij}, p_{i+1j}, p_{i+2j}, p_{i+3j}\}, & i = 1; \\ \{p_{i-3j}, p_{i-2j}, p_{i-1j}, p_{ij}, p_{i+1j}\}, & i = N_x - 1; \\ \{p_{ij}, p_{i+1j}, p_{i+2j}, p_{i+3j}, p_{i+4j}\}, & i = 0; \\ \{p_{i-4j}, p_{i-3j}, p_{i-2j}, p_{i-1j}, p_{ij}\}, & i = N_x. \end{cases} \quad (39)$$

Eqs. (37)–(39) describe an approximation to the pressure derivative with respect to the x coordinate. Similarly, the pressure derivative with respect to the y coordinate which enters the pressure gradient F , is approximated using a similar scheme.

The approximation scheme defined by Eq. (36) can be formulated in a simplified form as follows:

$$S_{ij}p_{ij-1} + W_{ij}p_{i-1j} + C_{ij}p_{ij} + E_{ij}p_{i+1j} + N_{ij}p_{ij+1} = R_{ij}$$

$$S_{ij} = \frac{(wK)_{ij-0.5}}{h_y^2}, \quad W_{ij} = \frac{(wK)_{i-0.5j}}{h_x^2},$$

$$E_{ij} = \frac{(wK)_{i+0.5j}}{h_x^2}, \quad N_{ij} = \frac{(wK)_{ij+0.5}}{h_y^2},$$

$$C_{ij} = -(S_{ij} + W_{ij} + E_{ij} + N_{ij}),$$

$$\begin{aligned}
R_{ij} = & -Bu \frac{(K\rho_m)_{ij+0.5} - (K\rho_m)_{ij-0.5}}{h_y} + \frac{w_{ij}^{n+1} - w_{ij}^n}{\Delta t} \\
& + 2v_{l,ij} - \frac{(wC_p\nu_s)_{ij+0.5} - (wC_p\nu_s)_{ij-0.5}}{h_y}.
\end{aligned}$$

Here, S , W , C , E , N are matrices of coefficients corresponding to “south”, “west”, “center”, “east”, “north” nodes of the approximation stencil; R is the matrix representing right-hand side.

Boundary conditions determined by Eqs. (32)–(34) are formulated in terms of the fluid velocity and are approximated using ghost cells (denoted below by indexes out of the range described by Eq. (35)) as follows:

$$\begin{aligned}
w_{in}u_{inj} &= 0.5(w_{-0.5j}u_{-0.5j} + w_{0.5j}u_{0.5j}), \\
w_{-0.5j}u_{-0.5j} &= 2w_{in}u_{inj} - w_{0.5j}u_{0.5j}, \\
u_{inj} &= u_{in}, \text{ if } j \in J_{in} \text{ and } u_{inj} = 0, \text{ otherwise;}
\end{aligned} \quad (40)$$

$$w_{ij-0.5}v_{ij-0.5} = -w_{ij+0.5}v_{ij+0.5}, \quad 0 \leq i \leq N_x, \quad j = 0 \text{ \& } N_y; \quad (41)$$

$$w_{N_x+0.5j}u_{N_x+0.5j} = -w_{N_x-0.5j}u_{N_x-0.5j}, \quad 0 \leq j \leq N_y. \quad (42)$$

Here, J_{in} is the interval of mesh nodes corresponding to the perforated region I_{in} . It is important to note that volumetric fluxes were approximated instead of fluid velocities to preserve mass conservation in discrete form.

At the left boundary ($i = 0$), the approximation is made taking into account the boundary conditions defined by Eq. (40):

$$S_{0j} = \begin{cases} 0, & j = 0 \\ \frac{(wK)_{0j-0.5}}{h_y^2}, & 0 < j < N_y, \\ \frac{2(wK)_{0j-0.5}}{h_y^2}, & j = N_y \end{cases},$$

$$N_{0j} = \begin{cases} \frac{2(wK)_{0j+0.5}}{h_y^2}, & j = 0 \\ \frac{(wK)_{0j+0.5}}{h_y^2}, & 0 < j < N_y, \\ 0, & j = N_y \end{cases},$$

$$W_{0j} = 0, \quad E_{0j} = \frac{2(wK)_{0.5j}}{h_x^2}, \quad C_{0j} = -(S_{0j} + E_{0j} + N_{0j}),$$

$$R_{ij} = \begin{cases} -2\text{Bu} \frac{(K\rho_m)_{i,0.5}}{h_y} - \frac{2u_{in,j}w_{in}}{h_x}, & j = 0 \\ -\text{Bu} \frac{(K\rho_m)_{0,j+0.5} - (K\rho_m)_{0,j-0.5}}{h_y} - \frac{u_{in,j}w_{in}}{h_x}, & 0 < j < M \\ 2\text{Bu} \frac{(K\rho_m)_{i,M-0.5}}{h_y} - \frac{2u_{in,j}w_{in}}{h_x}, & j = M \end{cases}$$

No-flux conditions (see Eqs. (41) and (42)) are used at the top and bottom boundaries ($j = 0, N_y$; $0 < i < N_x$):

$$S_{ij} = \begin{cases} \frac{2(wK)_{0,j-0.5}}{h_y^2}, & j = N_y \\ 0, & j = 0 \end{cases}, \quad N_{ij} = \begin{cases} \frac{2(wK)_{0,j+0.5}}{h_y^2}, & j = 0 \\ 0, & j = N_y \end{cases}$$

$$W_{ij} = \frac{(wK)_{i-0.5,j}}{h_x^2}, \quad E_{ij} = \frac{(wK)_{i+0.5,j}}{h_x^2}, \quad C_{ij} = -(W_{ij} + E_{ij} + N_{ij}), \quad j = 0, N_y$$

$$R_{ij} = \begin{cases} -2\text{Bu} \frac{(K\rho_m)_{i,j+0.5}}{h_y}, & j = 0, \\ 2\text{Bu} \frac{(K\rho_m)_{i,j-0.5}}{h_y}, & j = N_y, \end{cases}$$

The no-flux condition at the right vertical boundary of the computational domain with an immersed propagating hydraulic fracture ($i = N_x$, see Eq. (42)) is approximated as follows:

$$S_{Nj} = \begin{cases} 0, & j = 0 \\ \frac{(wK)_{N_x,j-0.5}}{h_y^2}, & 0 < j < N_y \\ \frac{2(wK)_{N_x,j-0.5}}{h_y^2}, & j = N_y \end{cases}, \quad N_{Nj} = \begin{cases} \frac{2(wK)_{N_x,j+0.5}}{h_y^2}, & j = 0 \\ \frac{(wK)_{N_x,j+0.5}}{h_y^2}, & 0 < j < N_y \\ 0, & j = N_y \end{cases}$$

$$W_{Nj} = \frac{2(wK)_{N-0.5,j}}{h_x^2}, \quad E_{0j} = 0, \quad C_{Nj} = -(S_{Nj} + W_{Nj} + N_{Nj}),$$

$$R_{Nj} = \begin{cases} -2\text{Bu} \frac{(K\rho_m)_{N_x,0.5}}{h_y}, & j = 0 \\ -\text{Bu} \frac{(K\rho_m)_{N_x,j+0.5} - (K\rho_m)_{N_x,j-0.5}}{h_y}, & 0 < j < N_y \\ 2\text{Bu} \frac{(K\rho_m)_{N_x,N_y-0.5}}{h_y}, & j = N_y \end{cases}$$

For an open channel (or Hele–Shaw cell), the pressure boundary condition at the right vertical boundary is approximated as follows:

$$p_{N_x,N_y} = 0, \quad p_{N_x,j-1} = p_{N_x,j} + \text{Bu} \rho_{m,N_x,j-0.5} h_y, \quad 0 \leq j \leq N_y$$

Note that the pressure inside incompressible fluids is determined up to a constant. Therefore, it is assumed to be zero at the right vertical boundary ($i = N_x, j = N_y$) of the domain.

Appendix B. Comparison of base and modified AA algorithms

Algorithm 1. Anderson acceleration with delay d and memory m

Input initial guess p_0 , non-expanding mapping defined by a linear operator A , maximum iterations limit $s_{\max}$

Output pressure p

- 1: $a_0 = Ap_0$
- 2: $p_1 = a_0$
- 3: $f_0 = a_0 - p_0$
- 4: **for** $s \leftarrow 1, s_{\max}$ **do**
- 5: $a_s = Ap_s$
- 6: $f_s = a_s - p_s$
- 7: $k = \max(0, \min(m, s - d))$
- 8: Find $\alpha \in \mathbb{R}^{k+1}$ which minimizes $\|\sum_{j=0}^k \alpha_j f_{s-j}\|_2$ subject to $\sum_{j=0}^k \alpha_j = 1$
- 9: $p_{s+1} = \sum_{j=0}^k \alpha_j a_{s-j}$
- 10: **if** convergence criteria **then**
- 11: **break**
- 12: **end if**
- 13: **end for**
- 14: **return** p_{s+1}

Algorithm 2. Modified Anderson acceleration

Input initial guess p_0 ; algorithm $J(p)$ producing linearized pressure equation in a form $Ap = b$; solver $S(A, p, b)$ which solves linearized equation; maximum iterations limit $s_{\max}$, metric norm N defined on matrix and two vectors

Output $u = \{p, A, b\}$

- 1: $\{A_0, b_0\} = J(p_0)$
- 2: $\{u_{A_0}, u_{p_0}, u_{b_0}\} = \{A_0, p_0, b_0\}$
- 3: $p_1 = S(A_0, p_0, b_0)$
- 4: $\{A_1, b_1\} = J(p_1)$
- 5: $\{a_{A_1}, a_{p_1}, a_{b_1}\} = \{A_1, p_1, b_1\}$
- 6: $\{u_{A_1}, u_{p_1}, u_{b_1}\} = \{A_1, p_1, b_1\}$
- 7: $f_{A_1} = a_{A_1} - u_{A_1}$
- 8: $f_{p_1} = a_{p_1} - u_{p_1}$
- 9: $f_{b_1} = a_{b_1} - u_{b_1}$
- 10: **for** $s \leftarrow 1, s_{\max}$ **do**
- 11: $p_s = S(u_{A_s}, u_{p_s}, u_{b_s})$
- 12: $\{A_s, b_s\} = J(p_s)$
- 13: $\{a_{A_s}, a_{p_s}, a_{b_s}\} = \{A_s, p_s, b_s\}$
- 14: $f_{A_s} = a_{A_s} - u_{A_s}$
- 15: $f_{p_s} = a_{p_s} - u_{p_s}$
- 16: $f_{b_s} = a_{b_s} - u_{b_s}$
- 17: $k = \max(0, \min(m, s - d))$
- 18: Find $\alpha \in \mathbb{R}^{k+1}$ which minimizes $\|\sum_{j=0}^k \alpha_j (f_{A_{s-j}}, f_{p_{s-j}}, f_{b_{s-j}})\|_N$ subject to $\sum_{j=0}^k \alpha_j = 1$
- 19: $u_{A_{s+1}} = \sum_{j=0}^k \alpha_j a_{A_{s-j}}$
- 20: $u_{p_{s+1}} = \sum_{j=0}^k \alpha_j a_{p_{s-j}}$
- 21: $u_{b_{s+1}} = \sum_{j=0}^k \alpha_j a_{b_{s-j}}$
- 22: **if** residual increased drastically from last iteration and $s > d$ **then**
- 23: Disable AA for current loop by setting $d = s_{\max}$
- 24: **end if**
- 25: **if** convergence criteria **then**
- 26: **break**
- 27: **end if**
- 28: **end for**
- 29: **return** u_{s+1}

Appendix C. Structure of the Jacobi matrix

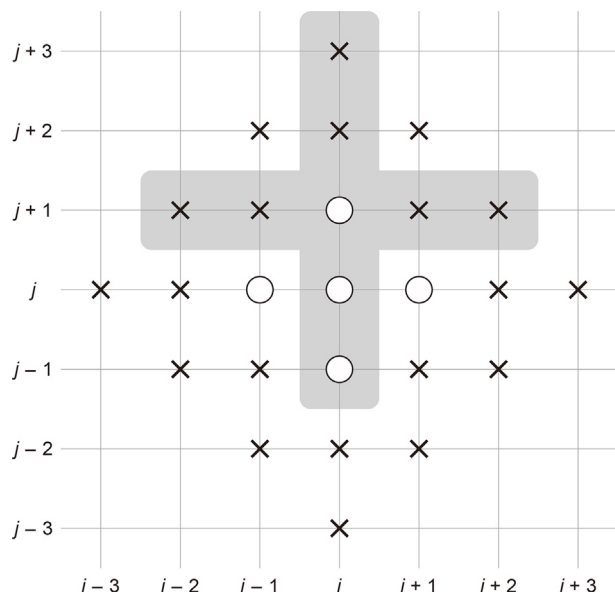


Fig. 18. The template for approximation of elliptic pressure equation (see Eq. (19)) in the mesh node (i, j) considered in the current study (away from the boundaries); standard 5-point “cross” approximation stencil applied to approximate the equation is marked with circles; shaded in grey is the stencil applied to approximate mobility of yield-stress fluids (function Y , see Eq. (12)) in the mesh node $(i, j + 1)$; corresponding extended approximation stencil used to form the Jacobi matrix is marked with crosses.

We consider the calculations required to apply Jacobian-based methods to numerical solution of quasi-linear elliptic pressure equation defined by Eq. (19) (e.g., Newton–Raphson or NR method).

As explained in Appendix A, we use a standard 5-point “cross” approximation stencil to approximate the elliptic equation using central differences (see circles in Fig. 18). The approximation of mobility coefficients in Eq. (19) requires larger (5-point in each direction) approximation stencil to resolve unyielded zones (the nodes involved into the approximation of fluid mobility in the node $(i, j + 1)$ are marked with the grey area in Fig. 18). As a result, the corresponding Jacobi matrix has 25 non-zero elements in rows representing internal nodes (e.g., it is a 25-diagonal one) as shown by crosses in Fig. 18.

References

- Acharya, A., Deysarkar, A., 1987. Rheology of fracturing fluids at low-shear conditions. In: SPE Annual Technical Conference and Exhibition. <https://doi.org/10.2118/16917-MS>.
- Adachi, J., Siebrits, E., Peirce, A., Desroches, J., 2007. Computer simulation of hydraulic fractures. *Int. J. Rock Mech. Min. Sci.* 44, 739–757. <https://doi.org/10.1016/j.ijrmms.2012.05.013>.
- Agrawal, A., Verschueren, R., Diamond, S., Boyd, S., 2018. A rewriting system for convex optimization problems. *J. Control Decis.* 5, 42–60. <https://doi.org/10.1080/23307706.2017.1397554>.
- Aksenov, V., Chertov, M., Sinkov, K., 2021. Application of accelerated fixed-point algorithms to hydrodynamic well-fracture coupling. *Comput. Geotech.* 129, 103783. <https://doi.org/10.1016/j.compgeo.2020.103783>.
- Anderson, D.G., 1965. Iterative procedures for nonlinear integral equations. *J. ACM* 12, 547–560. <https://doi.org/10.1145/321296.321305>.
- Aziz, K., Settari, A., 1979. *Petroleum Reservoir Simulation*. Chapman & Hall.
- Barbati, A.C., Desroches, J., Robisson, A., McKinley, G.H., 2016. Complex fluids and hydraulic fracturing. *Annu. Rev. Chem. Biomol. Eng.* 7, 415–453. <https://doi.org/10.1146/annurev-chembioeng-080615-033630>.
- Barboza, B.R., Chen, B., Li, C., 2021. A review on proppant transport modelling. *J. Petrol. Sci. Eng.* 204, 108753. <https://doi.org/10.1016/j.petrol.2021.108753>.
- Bittleston, S.H., Hassager, O., 1992. Flow of viscoplastic fluids in a rotating concentric annulus. *J. Non-Newtonian Fluid Mech.* 42, 19–36. [https://doi.org/10.1016/0377-0257\(92\)80002-F](https://doi.org/10.1016/0377-0257(92)80002-F).

- Boronin, S., Osipov, A., Desroches, J., 2015a. Displacement of yield-stress fluids in a fracture. *Int. J. Multiphas. Flow* 76, 47–63. <https://doi.org/10.1016/j.ijmultiphaseflow.2015.07.001>.
- Boronin, S., Osipov, A., Desroches, J., 2015b. Displacement of yield-stress fluids in a fracture. *Int. J. Multiphas. Flow* 76, 47–63. <https://doi.org/10.1016/j.ijmultiphaseflow.2015.07.001>.
- Boronin, S.A., Osipov, A.A., 2010. Two continua model of suspension flow in a hydraulic fracture. *Dokl. Phys.* 55, 192–202. <https://doi.org/10.1134/S1028335810040117>.
- Brennen, C.E., 2005. *Fundamentals of Multiphase Flow*. Cambridge University Press.
- Carrasco-Teja, M., Frigaard, I., 2010. Non-Newtonian fluid displacements in horizontal narrow eccentric annuli: effects of slow motion of the inner cylinder. *J. Fluid Mech.* 653, 137–173. <https://doi.org/10.1017/S0022112010000212>.
- Chen, Z., Cheng, W., Li, X., 2014. A global convergent quasi-Newton method for systems of monotone equations. *J. Appl. Math. Comput.* 44, 455–465. <https://doi.org/10.1007/s12190-013-0702-0>.
- Chertov, M., Chaplygin, A., 2019. Evaluating characteristics of high-rate hydraulic fractures driven by wellbore energy source. *Eng. Fract. Mech.* 222, 106702. <https://doi.org/10.1016/j.engfracmech.2019.106702>.
- Dontsov, E., Peirce, A., 2015. An enhanced pseudo-3D model for hydraulic fracturing accounting for viscous height growth, non-local elasticity, and lateral toughness. *Eng. Fract. Mech.* 142, 116–139. <https://doi.org/10.1016/j.engfracmech.2015.05.043>.
- Economides, M., Nolte, K., 2000. *Reservoir Stimulation*. John Wiley & Sons, Location New York, NY.
- Fang, H., Saad, Y., 2009. Two classes of multisecant methods for nonlinear acceleration. *Numer. Lin. Algebra Appl.* 16, 197–221. <https://doi.org/10.1002/nla.617>.
- Gay, D.M., Schnabel, R.B., 1978. Solving systems of nonlinear equations by broyden's method with projected updates. In: *Nonlinear Programming 3*. Elsevier, pp. 245–281. <https://doi.org/10.1016/B978-0-12-468660-1.50014-1>.
- Hammond, P.S., 1995. Settling and slumping in a Newtonian slurry, and implications for proppant placement during hydraulic fracturing of gas wells. *Chem. Eng. Sci.* 50, 3247–3260. [https://doi.org/10.1016/0009-2509\(95\)00152-U](https://doi.org/10.1016/0009-2509(95)00152-U).
- Hillam, B.P., 1975. A generalization of krasnoselski's theorem on the real line. *Math. Mag.* 48, 167–168. <https://doi.org/10.1080/0025570X.1975.11976471>.
- Ho, N., Olson, S.D., Walker, H.F., 2017. Accelerating the uzawa algorithm. *SIAM J. Sci. Comput.* 39, S461–S476. <https://doi.org/10.1137/16M1076770>.
- Hu, Y.T., Kishore, T., Maxey, J., Loveless, D., 2015. Effects of crosslinking chemistry on proppant suspension in guar networks. In: *SPE International Symposium on Oilfield Chemistry*. <https://doi.org/10.2118/173726-MS>.
- Isaev, V.I., Idimeshev, S.V., Semin, L.G., Tikhonov, A.A., 2023. A Lagrangian method for slurry flow modeling in hydraulic fractures. *Geoenergy Sci. Eng.* 231, 212272. <https://doi.org/10.1016/j.geoen.2023.212272>.
- Leveque, R.J., 1996. High-resolution conservative algorithms for advection in incompressible flow. *SIAM J. Numer. Anal.* 33, 627–665. <https://doi.org/10.1137/0733033>.
- Li, D., Fukushima, M., 1999. A globally and superlinearly convergent gauss–Newton-based bfgs method for symmetric nonlinear equations. *SIAM J. Numer. Anal.* 37, 152–172. <https://doi.org/10.1137/S0036142998335704>.
- Li, Y., Peng, G., Tang, J., Zhang, J., Zhao, W., Liu, B., Pan, Y., 2023a. Thermo-hydro-mechanical coupling simulation for fracture propagation in CO₂ fracturing based on phase-field model. *Energy* 284, 128629. <https://doi.org/10.1016/j.energy.2023.128629>.
- Li, Y., Wu, J., Zhang, J., Yang, H., Zeng, B., Xiao, Y., Liu, J., et al., 2023b. Optimization method of oriented perforation parameters improving uneven fractures initiation for horizontal well fracturing. *Fuel* 349, 128754. <https://doi.org/10.1016/j.fuel.2023.128754>.
- Luenberger, D.G., Ye, Y., et al., 1984. *Linear and Nonlinear Programming*, Vol. 2. Springer. <https://doi.org/10.1007/978-3-030-85450-8>.
- Mann, W.R., 1953. Mean value methods in iteration. *Proc. Am. Math. Soc.* 4, 506–510.
- Matveev, S., Stadnichuk, V., Tyrtysnikov, E., Smirnov, A., Ampilogova, N., Brilliantov, N., 2018. Anderson acceleration method of finding steady-state particle size distribution for a wide class of aggregation–fragmentation models. *Comput. Phys. Commun.* 224, 154–163. <https://doi.org/10.1016/j.cpc.2017.11.002>.
- Muravleva, E.A., Derbyshev, D.Y., Boronin, S.A., Osipov, A.A., 2021. Multigrid pressure solver for 2D displacement problems in drilling, cementing, fracturing and eor. *J. Petrol. Sci. Eng.* 196, 107918. <https://doi.org/10.1016/j.petrol.2020.107918>.
- O'donoghue, B., Chu, E., Parikh, N., Boyd, S., 2016. Conic optimization via operator splitting and homogeneous self-dual embedding. *J. Optim. Theor. Appl.* 169, 1042–1068. <https://doi.org/10.1007/s10957-016-0892-3>.
- Osipov, A., Garagash, I., Boronin, S., Tolmacheva, K., Lezhnev, K., Paderin, G., 2020. Impact of flowback dynamics on fracture conductivity. *J. Petrol. Sci. Eng.* 188, 106822. <https://doi.org/10.1016/j.petrol.2019.106822>.
- Osipov, A.A., 2017. Fluid mechanics of hydraulic fracturing: a review. *J. Petrol. Sci. Eng.* 156, 513–535. <https://doi.org/10.1016/j.petrol.2017.05.019>.
- Pavlov, A.L., Ovchinnikov, G.W., Derbyshev, D.Y., Tsetserukou, D., Oseledets, I.V., 2018. Aa-icp: iterative closest point with anderson acceleration. In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3407–3412. <https://doi.org/10.1109/ICRA.2018.8461063>.
- Pearson, J., 1994. On suspension transport in a fracture: framework for a global model. *J. Non-Newtonian Fluid Mech.* 54, 503–513. [https://doi.org/10.1016/0377-0257\(92\)80002-F](https://doi.org/10.1016/0377-0257(92)80002-F).

- 0377-0257(94)80039-1.
- Pelipenko, S., Frigaard, I., 2004a. Mud removal and cement placement during primary cementing of an oil well—part 2; steady-state displacements. *J. Eng. Math.* 48, 1–26. <https://doi.org/10.1023/B:ENGL.00000009499.63859.f0>.
- Pelipenko, S., Frigaard, I., 2004b. Two-dimensional computational simulation of eccentric annular cementing displacements. *IMA J. Appl. Math.* 69, 557–583. <https://doi.org/10.1093/imamat/69.6.557>.
- Potra, F.A., Engler, H., 2013. A characterization of the behavior of the anderson acceleration on linear problems. *Lin. Algebra Appl.* 438, 1002–1011. <https://doi.org/10.1016/j.laa.2012.09.008>.
- Rohwedder, T., Schneider, R., 2011. An analysis for the diis acceleration method used in quantum chemistry calculations. *J. Math. Chem.* 49, 1889. <https://doi.org/10.1007/s10910-011-9863-y>.
- Salinas, P., Pavlidis, D., Xie, Z., Adam, A., Pain, C., Jackson, M., 2017. Improving the convergence behaviour of a fixed-point-iteration solver for multiphase flow in porous media. *Int. J. Numer. Methods Fluid.* 84, 466–476. <https://doi.org/10.1002/flid.4357>.
- Schlenkrich, S., Walther, A., 2009. Global convergence of quasi-Newton methods based on adjoint broyden updates. *Appl. Numer. Math.* 59, 1120–1136. <https://doi.org/10.1016/j.apnum.2008.05.007>.
- Scieur, D., Bach, F.R., d'Aspremont, A., 2017. Nonlinear Acceleration of Stochastic Algorithms. NIPS.
- Scieur, D., d'Aspremont, A., Bach, F., 2020. Regularized nonlinear acceleration. *Math. Program.* 179, 47–83.
- Shiozawa, S., McClure, M., 2016. Simulation of proppant transport with gravitational settling and fracture closure in a three-dimensional hydraulic fracturing simulator. *J. Petrol. Sci. Eng.* 138, 298–314. <https://doi.org/10.1016/j.petrol.2016.01.002>.
- Sinkov, K., Weng, X., Kresse, O., 2021. Modeling of proppant distribution during fracturing of multiple perforation clusters in horizontal wells. In: SPE Hydraulic Fracturing Technology Conference and Exhibition. <https://doi.org/10.2118/204207-MS>.
- Toth, A., Kelley, C., 2015. Convergence analysis for anderson acceleration. *SIAM J. Numer. Anal.* 53, 805–819. <https://doi.org/10.1137/130919398>.
- Walker, H.F., Ni, P., 2011. Anderson acceleration for fixed-point iterations. *SIAM J. Numer. Anal.* 49, 1715–1735. <https://doi.org/10.1137/10078356X>.
- Yuwei, L., Min, L., Jizhou, T., Mian, C., Xiaofei, F., 2020. A hydraulic fracture height mathematical model considering the influence of plastic region at fracture tip. *Petrol. Explor. Dev.* 47, 184–195. [https://doi.org/10.1016/S1876-3804\(20\)60017-9](https://doi.org/10.1016/S1876-3804(20)60017-9).
- Zhang, F., Zhu, H., Zhou, H., Guo, J., Huang, B., 2017. Discrete-element-method/computational-fluid-dynamics coupling simulation of proppant embedment and fracture conductivity after hydraulic fracturing. *SPE J.* 22, 632–644. <https://doi.org/10.2118/185172-PA>.
- Zhou, W., Li, D., 2007. Limited memory bfgs method for nonlinear monotone equations. *J. Comput. Math.* 89–96.