Implicit finite element schemes are developed for the stationary compressible Euler equations and extended to a two-fluid model that describes the flow of an inviscid gas carrying a suspension of solid particles.

Due to the lack of diffusive terms, the standard Galerkin discretization is potentially unstable and oscillatory. Therefore, the discretization of inviscid fluxes is performed using a high-resolution finite element scheme based on algebraic flux correction. The artificial diffusion operator is constructed on the algebraic level and fitted to the local solution behavior using a multidimensional flux limiter of TVD type.

The numerical treatment of the boundary conditions for the Euler equations relies on a characteristic decomposition into incoming and outgoing waves. To avoid stability restrictions and severe convergence problems, characteristic boundary conditions of weak Neumann-type are imposed. The fluxes that appear in the boundary integrals of the weak formulation are computed using an approximate or exact Riemann solver, which avoids unphysical boundary states particularly at the early stage of a simulation. Since the fully coupled two-fluid model lacks hyperbolicity, Neumann-type boundary conditions are prescribed separately for each phase. Special care is taken on solid wall boundaries, where standard techniques developed for the gas phase do not prevent the particles from penetrating the wall. To enforce the free-slip condition, a penalty term is added to the momentum equations for both phases. The weak form of boundary conditions is shown to be very accurate and to exhibit far superior robustness and convergence compared to its strong counterpart.

Special emphasis is laid on the efficient computation of steady-state solutions in a wide range of Mach numbers. The stationary system is solved either by a (pseudo) time stepping method or using a Newton-like approach. To avoid computationally expensive nonlinear iterations in every pseudo time step a semi-implicit formulation of the backward Euler method is derived by a time-lagged linearization of the residual. The original Jacobian is replaced by a low-order approximation. For infinite CFL numbers, a Newton-like method is obtained. The high performance of the nonlinear solver and its ability to handle large CFL numbers are demonstrated. Strong numerical evidence of unconditional stability is presented. In particular, it is shown that the nonlinear convergence rates do not deteriorate when the CFL number exceeds a certain upper bound. Steady-state solutions are always achieved, although limiters of the employed family are commonly believed to inhibit convergence. In spite of additional nonlinearities due to the algebraic source terms, the two-fluid model solver is shown to possess most of the properties of its single-phase counterpart.

The two-fluid model of compressible multiphase flows contains algebraic source terms which introduce a two-way coupling. In most codes, these terms are included using fractional-step algorithms based on operator splitting. This approach has an adverse effect on convergence to steady-state and gives rise to a restrictive CFL upper bound. Enhanced robustness and convergence are obtained with a fully coupled iterative solver where the algebraic source terms are incorporated into the preconditioner and operator splitting is avoided. The numerical results to be presented illustrate the advantages of the proposed solution strategy.

The practically unconditionally stable finite element solver for the Euler and two-fluid equations and the achievement of steady-state solutions to the two-fluid model are the main novel contributions. Another important proposal is the use of Neumann-type boundary conditions combined with a ghost state Riemann solver, which enables the convergence rates to increase with increasing CFL numbers. To achieve unconditional stability and convergence for arbitrary CFL numbers, the inclusion of interfacial transfer terms in an implicit way without using operator splitting is another important new contribution.