50th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition
09 - 12 January 2012, Nashville, Tennessee

AIAA 2012-0761

Parallel, Adaptive Grid Computing of Multiphase Flows in Spacecraft Fuel Tanks

Chih-Kuang Kuan[*] and Jaeheon Sim[†]
University of Michigan, Ann Arbor, MI, 48109

and

Wei Shyy[‡]
University of Michigan, Ann Arbor, MI, 48109
Hong Kong University of Science and Technology, Kowloon, Hong Kong

A parallel adaptive Eulerian-Lagrangian method is developed for effective large scale multiphase flow computation. The intricate complexity of parallelism for the Eulerian-Lagrangian method is discussed and spatial domain decomposition strategies are proposed. An efficient communication approach is presented with MPI standard communication and the performance of the parallel Eulerian-Lagrangian method is evaluated. The speedup on problems having million of cells shows super-linear behavior due to the effect of cache hit rate, and the parallel efficiency peaks at the trade-off point of the cache hit rate and the computation-to-communication ratio. A first-principle-based phase change model is introduced by computing the rate of phase change in each phase as well as across liquid/vapor interface. The self-pressurization in a liquid hydrogen fuel tank is simulated and it is shown that the phase change in a liquid phase has a dominant contribution comparing phase change on the liquid/vapor surface, and the rate decreases as saturation temperature increases. The effects of heat flux amount and liquid fuel fill-level are investigated, and the pressurization accelerates showing longer transition period with higher pressure rise rate as heat flux and fill level increases.

I. Introduction

The reliable and economic management of cryogenic liquids in a spacecraft including storing and transferring becomes more critical for the long duration future space missions such as interplanetary exploration. Cryogenic liquid is required for human life support systems and spacecraft propulsion systems, and its management determines human survival in space and the success of the mission. Even a small heat leak from incident solar radiation, propulsion system, aerodynamics heating, or the spacecraft structure can cause vaporization of the liquid due to extremely low boiling temperatures and long duration missions. The vaporization results in the increase of pressure in storage tanks which causes safety issues and design difficulties in liquid storages. The vent system releasing high pressured vapor can result in loss of cryogenic liquids. Improving the cryogenic storage and lowering the pressurization leads to significant cost savings through the reduction of launch mass[1]. Various pressure control techniques have been proposed; passive thermal insulation, cryocooler mixing, (passive) thermodynamic vent systems, and active thermodynamic vent systems with heat exchanger and fluid mixing techniques[1, 2]. The concept of zero boiloff (ZBO) storage of cryogenics has been developed by NASA for future space mission with duration on the order of years[1].

In order to understand and estimate the self-pressurization in cryogenic liquid storages, many theoretical, numerical, and experimental studies have been conducted. The homogeneous model[3-6] (or thermodynamic model) and surface evaporation model have been widely used due to their mathematical simplicity. The

[*] Graduate Student, Department of Aerospace Engineering, University of Michigan; AIAA Student member.

[†] Post-doctoral Research Fellow, Department of Aerospace Engineering, University of Michigan; AIAA member.

[‡] Provost & Chair Professor, Department of Mechanical Engineering, Hong Kong University of Science and Technology; Adjunct Professor, Department of Aerospace Engineering, University of Michigan; AIAA Fellow.

homogeneous model assumes that both liquid and vapor phases are at the thermodynamic equilibrium states with the saturation temperature of the cryogenic fluid at the total tank pressure. This model shows lower pressurization than experiment, and is referenced usually as the lower limit of pressure rise. The surface evaporation model assumes that all the heat influx is used to evaporate liquid, maintaining the initial liquid temperature. The vapor temperature is thus at the saturation temperature corresponding to the final system pressure. This surface evaporation model is referred to as the upper limit of pressure rise. In the experimental studies, several different variables were studied, including the effect of gravity, heat flux amount and distribution, tank size, tank geometry, fill level, and mixing. One of the noticeable studies was conducted by Aydelott[3, 4], who showed the rate of pressure rise is lower in reduced-gravity than normal gravity because of the increased liquid-wetted wall area and increasing natural convection. He also mentioned the faster increasing vapor temperatures is the difference between the homogeneous model and experiment. Poth and Van Hook showed that a mixing jet could be used to minimize thermal stratification and reduce pressure rise in a tank[7, 8]. Many other studies showed that the difference between the homogeneous model and experiment are much large in a larger storage and/or higher heat influx[5]. However, the final rate of pressure rise after initial transient is very close to that of homogeneous model. Nevertheless, the final pressure rise cannot be predicted correctly without appropriate initial transient studies. The numerical studies followed for deeper understanding and overcome the experimental limitation. However, due to its complexity including flow motions, heat transfer, and phase change, the numerical and theoretical studies are conducted in a very limited condition. Liquid Conduction-only model[9] are implemented in the beginning, and later, the lumped vapor model were the most popular approaches by solving thermal effect in cryogenic liquid region with assumption of homogenous condition in vapor phases[6, 10]. However, the lack of agreement between these simple mathematical/numerical approaches and experiment, and the difficulties in the experimental studies encourage finer numerical approaches by solving full Navier-Stokes and energy equation for both phases.

Considering microgravity conditions in a typical spacecraft environment, the capillary effect is critical due to small Bond number, the ratio of body (or gravitational) forces to surface tension forces. Thus, a study on the multiphase flow including interfacial dynamics is required in order to understand the liquid fuel dynamics in a spacecraft. However, experimental studies are limited because the microgravity conditions are hard to realize on the ground. Thus, a high fidelity numerical simulation of such a multiphase flow in a space environment is crucial to compensate the limitation of experiment. Though numerical simulation is a useful approach, it is still challenging because of the multiple time/length scales, large variations in fluid properties, moving boundaries, and phase changes in multiphase flow. To fully capture the physics of interfacial dynamic, detail geometrical features of irregular interface shape has to be resolved by fine resolution. Such a multiple length scale problem demands expensive computing resource in modeling. The varying source terms in governing equations, such as mass transfer and surface tension, may restrict the time step for numerical stability. Moreover, an additional difficulty arise when the jump of density across fluid interface is large, the encounter of numerical stiffness is inevitable.

To provide a compatible computational platform with simulating large-scale multiphase flows, distributed computation via parallel computing is preferable. Parallelized multiphase flow simulations using the volume of fluid method is studied broadly and showed good parallel performance for varying large problems[11]. Sussman demonstrated efficient parallel algorithm based on coupled level set/volume-of-fluid method.[12]. In case of Eulerian-Lagrangian methods, its' applicability to parallelization is complex since interactions between Eulerian and Lagrangian data is intricate. Unlike the volume of fluid and level-set method which are both categorized as Eulerian computations, the major challenge of Eulerian-Lagrangian methods is the decomposition of computation on Lagrangian domain and the mapping of data in between Eulerian and Lagrangian coordinates. Parallelism for both Eulerian and Lagrangian coordinates have to be considered.

In the present study, we have further developed the previously reported 3-D adaptive Eulerian-Lagrangian method[13-15] by parallel implementation for both continuous and sharp interface methods and energy equation with a phase change effect. This numerical method utilizes the stationary (Eulerian) frame to resolve the flow field, and the marker-based triangulated moving (Lagrangian) surface meshes to treat the phase boundary interfaces. The large-deformable fluid boundaries are modeled using a continuous interface method, and the surface tension between fluid interfaces is smeared within finite distance. The solid boundaries are treated by a sharp interface method along with the ghost cell method by reconstructing the solution on the ghost cell based on the known solid boundary condition. The contact line where the fluid interface meets the solid boundary is modeled using a contact line force model, which enforces the given contact angle dynamically. For moving contact line treatment, local slip condition is applied around the contact line. The first-principle-based phase change model is introduced by computing the amount of a phase change in the pure material region as well as across liquid/vapor interface. For more effective computation in large scale problem, the adaptive Cartesian grid method and the multi-domain parallel computing technique is implemented with the help of the MPI standard communication and METIS library. The contribution of phase change in each phase and on the interface surface are investigated, and the effects of various heat flux amount and liquid fuel fill-levels are followed on a liquid hydrogen fuel tank under normal gravity.

II. Eulerian-Lagrangian Methods

A. Governing equations

The marker-based 3-D adaptive Eulerian-Lagrangian method is a multiphase flow algorithm which is able to resolve interfacial dynamic with surface tension computation and phase change modeling. The bulk flow variables are solved on the stationary (Eulerian) background grid, whereas interface variables are handled by moving (Lagrangian) surface meshes. Figure 1 shows a brief framework of the present Eulerian-Lagrangian method, with the detailed numerical method found in the previous works[13-15]. Figure 1 also shows the schematic drawing of the present approach for multiphase flow computation.

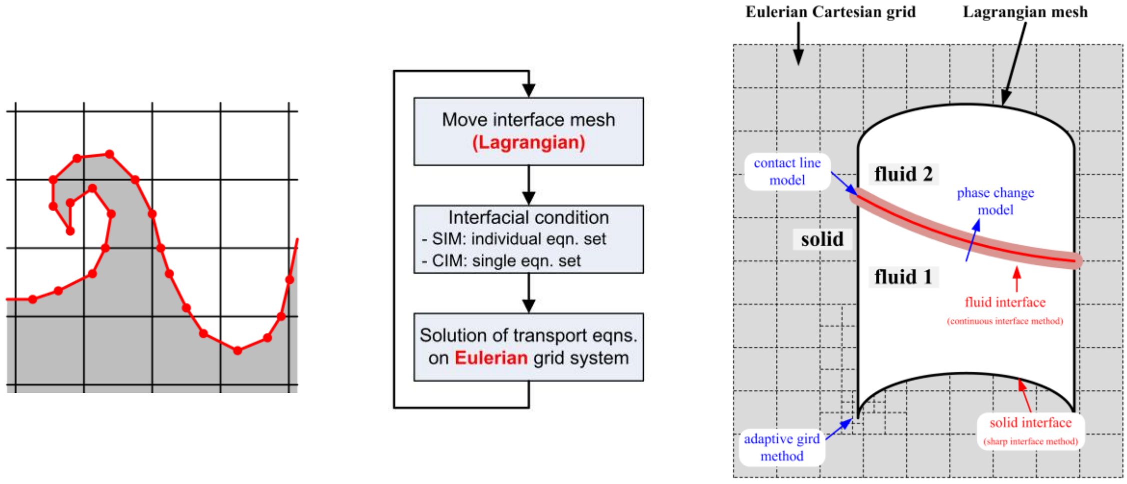
Non-dimensionalized continuity, momentum and energy equations for incompressible Newtonian fluid can be written in Eqs. (1)–(3), which account for the interfacial dynamics as source terms in the governing equations; surface tension effects of fluid interfaces as a momentum forcing term ($\boldsymbol{F}_f$), and the latent heat effects across fluid interfaces as a energy source term (Q_f). In Eqs. (1) and (2), $\boldsymbol{u}$ is the velocity vector, and ρ, μ, and p is the density, viscosity, and pressure, respectively.

$$\frac{\partial \rho}{\partial t} + \boldsymbol{\nabla} \cdot (\rho \boldsymbol{u}) = 0 \tag{1}$$

$$\frac{\partial \rho \boldsymbol{u}}{\partial t} + \boldsymbol{\nabla} \cdot (\rho \boldsymbol{u}\boldsymbol{u}) = -\boldsymbol{\nabla} p + \frac{1}{Re} \boldsymbol{\nabla} \cdot (\mu \boldsymbol{\nabla} \boldsymbol{u} + \mu \boldsymbol{\nabla}^T \boldsymbol{u}) + \frac{1}{Fr} \rho \boldsymbol{g} + \frac{1}{We} \boldsymbol{F}_f \tag{2}$$

$$\frac{\partial \rho C T}{\partial t} + \boldsymbol{\nabla} \cdot (\rho C T \boldsymbol{u}) = \boldsymbol{\nabla} \cdot (K \boldsymbol{\nabla} T) + Q_f \tag{3}$$

Here, variables are non-dimensionalized by a characteristic velocity (U) and length scale (L), standard gravity



(a) (b) (c)

Figure 1. Eulerian-Lagrangian methods. (a) Interface representation and tracking using moving meshes (red) on the stationary Cartesian grid. (b) Solution procedures. (c) Illustration of the present numerical approach.

($\mathrm{g_0}$), and liquid material properties (density ρ_l, viscosity μ_l, and surface tension σ). The Reynolds, Froude, and Weber numbers in Eq. (2) are defined as, $\mathrm{Re}=\rho_l \mathrm{UL}/\mu_l$, $\mathrm{Fr}=\mathrm{U}^2/(\mathrm{g_0 L})$, and $\mathrm{We}=\rho_l \mathrm{U}^2\mathrm{L}/\sigma$. In addition, the capillary number and bond number can be defined as $\mathrm{Ca}=\mathrm{We}/\mathrm{Re}=\mu_l \mathrm{U}/\sigma$ and $\mathrm{Bo}=\mathrm{We}/\mathrm{Fr}=\rho_l \mathrm{g_0 L}^2/\sigma$, respectively.

Using the continuous interface method, the single set of equations for all fluid phases in the whole domain is achieved with the help of an approximate Dirac delta function (δ_h) and an indicator function (I). The approximate Dirac delta function, originally proposed by Peskin,[16] is calculated over finite thickness of four cell widths. It provides a weighting projection for variables communication between the Eulerian and Lagrangian domains. Besides this, an indicator function varying from zero to one smoothly across the interface has a value of 0.5 at the interface location. The smoothed fluid properties such as density and viscosity are computed by indicator function in Eqs. (4)-(7).

$$\rho=\rho_2+\ (\rho_1-\rho_2)I \tag{4}$$

$$\frac{\rho}{\mu}=\left(\frac{\rho}{\mu}\right)_2+\left[\left(\frac{\rho}{\mu}\right)_1-\left(\frac{\rho}{\mu}\right)_2\right]I \tag{5}$$

$$\rho C=(\rho C)_2+\ [(\rho C)_1-(\rho C)_2]\,I \tag{6}$$

$$\frac{\rho C}{k}=\left(\frac{\rho C}{k}\right)_2+\left[\left(\frac{\rho C}{k}\right)_1-\left(\frac{\rho C}{k}\right)_2\right]I \tag{7}$$

Separate indicator functions for fluid and solid interfaces are used to separate the designation of the solid interface, modeled by a sharp interface method, from the fluid interface modeled by a continuous interface method. The indicator function is computed using a discrete form of the Heaviside step function in Eq. (8) by integrating the 1-D form of discrete Dirac delta function. This approach is known being more general than the Poisson equation solver method since it requires only distance information from interface, and thus gives accurate values even at the boundaries.[14]

$$I(\mathbf{x})=\mathrm{H}\big(\mathrm{r}=\mathbf{n}\cdot(\mathbf{x}-\mathbf{X})\big)=\int_{-\infty}^{\mathrm{r}}\delta(\mathrm{h})\mathrm{dh} \tag{8}$$

The governing equations (1)-(3) are solved using a projection method on staggered grid finite volume formulation. The pressure and fluid properties are stored at the cell center and face-normal velocities are stored on the Cartesian cell faces. The convection term ($\boldsymbol{N}$) is discretized using a 3rd order ENO scheme in space and a 2nd order Runge-Kutta integration in time. The central difference scheme and Crank-Nicholson method is implemented for the viscous term ($\boldsymbol{L}$). The discretized solution procedures are summarized in Eqs. (9)-(12). In Eq. (13), $\nabla\cdot\boldsymbol{u}^{n+1}$ term is zero if the phase change doesn't occur and mass is not transferred across the interface.

$$\frac{\boldsymbol{u}^{*}-\boldsymbol{u}^{n}}{\Delta t}=-\boldsymbol{N}\big(\boldsymbol{u}^{n+1/2}\big)+\frac{1}{\rho^{n+1}}\big[-\nabla p^{n}+\boldsymbol{L}\big(\boldsymbol{u}^{n+1/2}\big)+\boldsymbol{F}^{n+1}\big] \tag{9}$$

$$\boldsymbol{u}^{**}=\boldsymbol{u}^{*}+\frac{\Delta t}{\rho^{n+1}}\nabla p^{n} \tag{10}$$

$$\boldsymbol{u}^{n+1}=\boldsymbol{u}^{**}-\frac{\Delta t}{\rho^{n+1}}\nabla p^{n+1} \tag{11}$$

$$\frac{T^{n+1}-T^{n}}{\Delta t}=-\nabla\cdot(\boldsymbol{u}^{n+1}T)^{n+1/2}+\frac{1}{(\rho C)^{n+1}}\big[\nabla\cdot(K\nabla T)^{n+1}+Q_f^{n+1}\big] \tag{12}$$

The pressure Poisson equation of Eq. (13) is derived by taking divergence of Eq. (11).

$$\nabla^2(p^{n+1}) = -\frac{\Delta t}{\rho^{n+1}}(\nabla\cdot\boldsymbol{u}^{**} - \nabla\cdot\boldsymbol{u}^{n+1}\,) \tag{13}$$

B. Interface representation and tracking

In this marker-based Eulerian-Lagrangian method, the interface is represented by markers in coordination with each other for maintaining the interface connectivity information. The geometrical structure is established via line-segments in two-dimensional and triangles in three-dimensional domains, as represented in Figure 1(a). The marker locations, denoted by $\mathbf{X}$ in Lagrangian frame, are updated from the velocities at its location, $\mathbf{V}(\mathbf{X})$, in Eq. (14).

$$\frac{\partial \mathbf{X}}{\partial \mathrm{t}} = \mathbf{V}(\mathbf{X}) \tag{14}$$

Fluid interfaces use the computed flow solution field to obtain the marker velocities as shown in Eq. (15). In this equation, the discrete Dirac delta function, $\delta(\boldsymbol{x}-\mathbf{X})$, is employed for converting the Eulerian velocity field, $\mathbf{u}(\boldsymbol{x})$, to Lagrangian form, $\mathbf{U}(\mathbf{X})$. The interface velocity is exactly same to fluid velocity if there is no mass transfer in Eq. (15). However, with phase change, the velocity component from mass transfer should be considered. On the other hand, solid interfaces use the prescribed velocity field to advance the marker points using Eq. (14).

$$\boldsymbol{U}(\boldsymbol{X}) = \int_V \boldsymbol{u}(\boldsymbol{x})\delta(\boldsymbol{x}-\boldsymbol{X})dV - \frac{\dot{m}}{\rho_f}\cdot\boldsymbol{n} \tag{15}$$

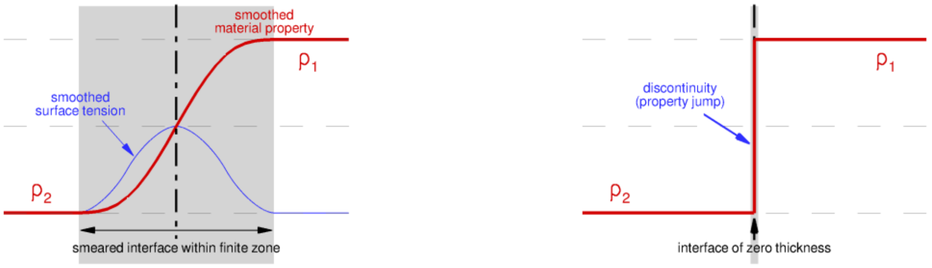
In order to maintain consistent computational accuracy, the spacing between marker points is rearranged by adding or deleting markers whenever two markers come too close or too far from each other. The level-contour-based interface reconstruction technique with connectivity information is also implemented to handle topological changes such as merger or break-ups. The detailed numerical method can be found in the previous works[15].

C. Interfacial dynamics modeling

The surface force computation of Eq. (16), where σ is the surface tension and κ is the curvature of interface, is applied as a source term in the momentum equation, Eq. (2).

$$\boldsymbol{F}_f = \int_{\Gamma(t)} \sigma\kappa\delta(\boldsymbol{x}-\boldsymbol{X})d\Gamma \tag{16}$$

Solid interfaces are modeled using the sharp interface method by reconstructing solution fields around an interface to incorporate the given boundary condition on the solid interface in an Eulerian Cartesian grid. The velocity reconstruction on the solid side is implemented via ghost cells, which are defined as solid cells having at least one neighboring fluid cell. This approach works well even with contact lines where fluid interface meets a solid interface. The basic idea of these two different approaches is shown in Figure 2, and the details can be found in our previous work[15].



(a) **(b)**

Figure 2. Interfacial dynamics modeling. (a) Fluid interface: continuous interface method with smeared interface within finite zone. (b) Solid interface: sharp interface method with the zero-thickness interface.

When a fluid-fluid interfaces intersects a solid surface, the treatment of the tri-junction locations, called contact lines, is required to account for the presence and interactions of all three phases, fluid-fluid-solid. One of the most discussed issues for modeling these contact lines with Navier-Stokes equations is that the imposed no-slip condition for velocity leads to a non-integrable singularity in stress. In the present research, the contact line force is imposed with local slip condition to overcome this singularity issue.

D. Mass transfer due to phase change across liquid/vapor interface

Mass transfer computation is one of the key issues in the phase change process since it is directly related to the movement of phase boundaries and the amount of latent heat in the energy transfer process. In the energy equation of Eq. (3), the latent heat is computed by Eq. (17), where $\dot{m}_f$ is mass transfer per unit area across interface due to phase change and L is the latent heat.

$$Q_f = \int_\Gamma \dot{m}_f L \delta(\boldsymbol{x} - \boldsymbol{X}) d\Gamma \tag{17}$$

Similar to the transformation of surface tension, the latent heat of the interface due to phase change is also transformed from a Lagrangian quantity ($\boldsymbol{X}$) to an Eulerian quantity ($\boldsymbol{x}$) via the approximate discrete Dirac delta function, $\delta_h(\boldsymbol{x} - \boldsymbol{X})$.

In the present study, the mass transfer is computed in Eq. (18) based on the Stefan condition using the temperature gradient with discontinuous material properties for simplicity and accuracy. L is the latent heat, k is thermal conductivity and $\boldsymbol{q_1}$ and $\boldsymbol{q_2}$ are heat flux vectors transporting across the liquid-vapor interface. The interface temperature is assumed equal to the saturation temperature which is an adequate assumption in macroscopic problems.[17]

$$\dot{m}_f L = (\boldsymbol{q_1} - \boldsymbol{q_2}) \cdot \hat{n} = -k_1 \left.\frac{\partial T}{\partial n}\right|_1 + k_2 \left.\frac{\partial T}{\partial n}\right|_2 \tag{18}$$

The energy equation of Eq. (3) is solved using continuous treatment with smoothed material properties in Eqs. (4)-(7), and the projection method in Eqs. (9)-(13) is applied to solve momentum equation. The mass conservation equation in Eq. (1) is enforced by assuming $\nabla \cdot \boldsymbol{u}^{n+1} = 0$ when the pressure Poisson equation is solved in Eq. (13). However, the divergence of the velocity is not zero around the interface if the phase change occurs and mass is transferred across the interface. Shin and Juric developed the conservation of mass in Eq. (19),[18] and in the present work, the modified version of Eq. (20) is implemented for the non-conservative form of the equation.

$$\nabla \cdot (\rho \boldsymbol{u}) = \boldsymbol{u} \cdot \nabla \rho + \frac{(\rho_2 - \rho_1)}{\rho_f} \int_{\Gamma(t)} \dot{m}_f \delta(\boldsymbol{x} - \boldsymbol{X}) d\Gamma \tag{19}$$

$$\nabla \cdot \boldsymbol{u} = \frac{1}{\rho}\left[(\rho_2 - \rho_1)\frac{\dot{m}_f}{\rho_f}\nabla I\right] = \frac{1}{\rho}\left[\frac{(\rho_2 - \rho_1)}{\rho_f}\int_{\Gamma(t)} \dot{m}_f \delta(\boldsymbol{x} - \boldsymbol{X}) d\Gamma\right] \tag{20}$$

In typical phase change models, the mass transfer occurs only across the interface between liquid fuel and vapor. However, this assumption is not appropriate because the liquid can be heated up to higher temperature than the saturation condition as heat is absorbed. In fact, phase change should occur if the liquid temperature is higher than the saturation temperature. Moreover, condensation happens if vapor temperature is lower than saturation temperature. We further define our phase change model more strictly by considering the mass flux per unit volume due to phase change in a liquid and a vapor region, denoted as $\dddot{m}_l$ and $\dddot{m}_v$, and as well as an interface region as followings.

- Interface region (0<I<1): phase change occurs based on the Stefan condition

  $\dot{m}_f L = (q_2 - q_1) \cdot \hat{n} = -k_2 \left.\frac{\partial T}{\partial n}\right|_2 + k_1 \left.\frac{\partial T}{\partial n}\right|_1$

- Liquid region (I=1): phase change occurs if T > T_{sat}

  $\ddot{m}_l L = \frac{1}{\Delta t}\rho_l V_l C_l \max(T - T_{sat}, 0)$ and $T = \min(T_{sat}, T)$
- Vapor region (I=0): phase change occurs if T < T_{sat}

  $\ddot{m}_v L = -\frac{1}{\Delta t}\rho_v V_v C_v \min(T - T_{sat}, 0)$ and $T = \max(T_{sat}, T)$

E. Thermodynamic model for spacecraft fuel tank computation

In the present study, the incompressible flow solver can be applicable to the spacecraft self-pressurization problems by implementing Boussinesq approximation for buoyancy effect and modifying thermodynamic properties of vapor, and saturation pressure and temperature according to the given thermodynamic conditions as phase change occurs.

The total phase change rate from liquid to vapor can be defined as Eq. (21) in a partially filled cryogenic storage tank.

$$\frac{dM_l}{dt} = \int_\Gamma \dot{m}_f d\Gamma + \int_{liquid} \ddot{m}_l dV - \int_{vapor} \ddot{m}_v dV \tag{21}$$

Where, M and V are mass and volume, and subscript l and v means liquid and vapor, respectively.

Then, the vapor volume change rate, and vapor volume at a given time, t, can be computed in Eq. (22) and Eq. (23). The density of liquid, ρ_l, is assumed constant.

$$\frac{dV_v}{dt} = \frac{1}{\rho_l}\cdot\frac{dM_l}{dt} \tag{22}$$

$$V_v(t) = V_v(0) + \frac{1}{\rho_l}\cdot\int_0^t dM_l \tag{23}$$

From the total mass and volume conservation in Eq. (24) and (25), the vapor density of Eq. (26) can be obtained.

$$\rho_v(t)V_v(t) + \rho_l V_l(t) = \rho_v(0)V_v(0) + \rho_l V_l(0) \tag{24}$$

$$V_v(t) + V_l(t) = V_v(0) + V_l(0) = V_{tank} \tag{25}$$

$$\rho_v(t) = \rho_l - \left(\rho_l - \rho_v(0)\right)\frac{V_v(0)}{V_v(t)} \tag{26}$$

The saturation pressure and temperature can be computed from the ideal gas law with compressibility factor, z, and Clausius-Clapeyron relation in Eq. (27) and (28) .

$$P_s(t) = \rho(t)z(T_s)RT_s(t) \tag{27}$$

$$\frac{dP_s}{dT_s} = \frac{L}{T_s\cdot\Delta V} \tag{28}$$

In the spacecraft fuel tank self-pressurization problem, the temperature in a tank is very low - around 20 K. In this low temperature environment, the material properties are not constant, and the ideal gas law is not appropriate as depicted in Figure 3. Therefore, the properties should be modified according to the temperature. In the present study, the real gas data related to the material properties and compressibility factor are obtained from NIST chemistry webbook.

F. Adaptive Grid

Multiphase flow problems involve multiple length scales. In order to effectively resolve the flow features in such cases, adaptive grids based on isotropic refinement is implemented. The cells are split into four and eight equal sibling cells in two- and three-dimensions, respectively, to better handle regions which require higher resolution. The grid is represented using unstructured data that connects cells through cell faces. The details of the algorithm can be found in Singh and Shyy.[13] In present study, only the static grid adaption is implemented for the parallel Eulerian-Lagrangian method. The static grid adaption method refines the Cartesian cells around the solid and fluid interfaces at the begging of simulation to provide finer resolution. The dynamic grid adaption involving repartition of Eulerian domain at runtime is not included in the current parallel code.

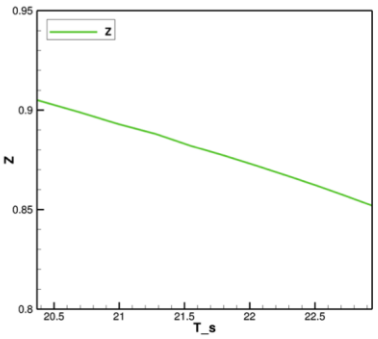


Figure 3. Low temperature effects on the compressibility factor of hydrogen with saturation temperature.

III. Parallelization of Eulerian-Lagrangian Method

We now present the parallelization of the adaptive Eulerian-Lagrangian method. Our goal of current study is to develop a parallel Eulerian-Lagrangian method for large problems that provides substantial computational power with scalability, compact data structures and efficient communication methodologies. For the purpose of portability, this parallel program is designed for distributed memory multiprocessors architecture with MPI standard for communication. Third party libraries, PETSc[19] and HYPRE are implemented for solving linear system of equations.

As addressed in previous sections, the Eulerian-Lagrangian method resolves the flow filed on a stationary Eulerian grid with a moving Lagrangian mesh superimposed on it. Eulerian and Lagrangian variables are stored on different spatial domains but interact with each other by projecting quantities through the Dirac delta function. Therefore, two issues need to be considered: decomposition strategies for both the Eulerian and Lagrangian domain and additional communication due to Eulerian-Lagrangian interaction.

Spatial domain decomposition is the most widely adopted for computational fluid dynamic methods because of its repeated calculation on a data set originated from discretized differential equation. Using spatial domain decomposition, the concurrent operation of each processor working on one piece of entire data can maintain load balance easily. In case of Eulerian-Lagrangian method, which has data stored on two different coordinates, the data parallelism framework favorable for Eulerian domain, but a suitable approach for decomposition of Lagrangian interface should be evaluated. The parallelism adopted for Lagrangian domain should provide load balance and minimize communication in frequency and data size as well.

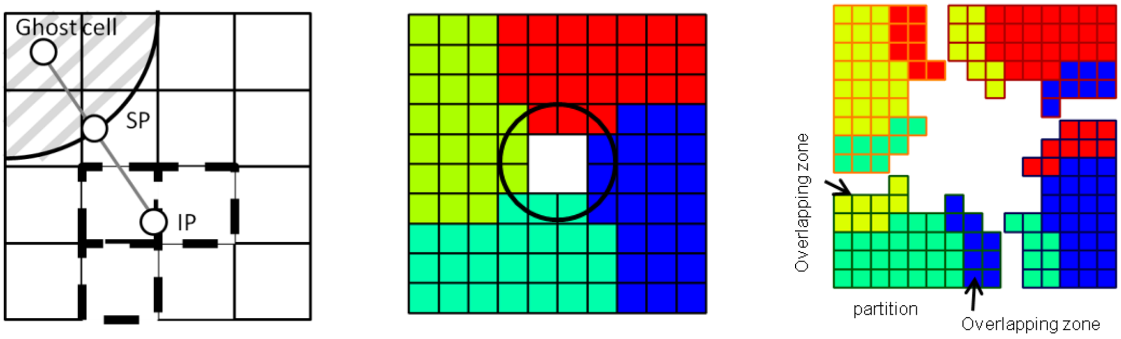
The detail approaches of parallel implementation are addressed in the following sections.

A. Domain Decomposition of Eulerian domain

The partitioning methodology of Eulerian domain needs to consider the following issues: load balance, communication cost; data dependency arising from ghost cell method. Among many approaches for spatial decomposition of a graph, the most popular two methods are space filling curves method and heuristic method such as partitioning libraries METIS[20]. A space-filling curve traverses N-dimensional graph and map it bijectively to a one-dimensional array. The ordered array is split into P parts which is the total number of processors. Consequently, the load balance is satisfied but total edges cut, that is, the size of partition boundary requiring information exchange is not an objective of space-filling curve method. On the other hand, graph partitioning packages, such as METIS using k-way multilevel heuristic approach as partition algorithm usually provide satisfactory load balance and fewer edges cut compared with space-filling curves method[21]. Moreover, it is easy to implement and provide flexible weighting function to manipulate preferred partition scenario. In this study METIS is implemented as the partitioning tool.

In our Eulerian-Lagrangian method, the boundary condition of solid interface is enforced by ghost cell method. First of all, solid boundaries are represented by immersed sharp interface that identifies the fluid and solid side. As shown in Figure 4(a), a cell in solid side adjacent to the immersed boundary would be recognized as ghost cell to help the boundary condition enforcement. Image points IP of a ghost cell are determined by extending the local normal of solid boundary into fluid side in a fixed distance. The variables at an image point is obtained by interpolation method using information from surrounding fluid cells, which are denoted with dash line. Then ghost cell velocity is calculated by linear extrapolation from the known values at image point and velocity at solid point SP, which is zero for no-slip boundary condition. Pressure and temperature at ghost cells are obtained as the same manner. As a result, a ghost cell has geometrical dependency to its corresponding fluid cells. Such one-to-multiple dependency of ghost cells on solid boundary results in ordering difficulties if using space-filling curve, but no hurdle whenever METIS is applied. In addition, a partition methodology that supports additional constrain for cell workload can help even distribution of ghost cells on requested processors, which decreases possibilities of processor idling. Vertex weighting of METIS provides this functionality for users to adjust flexible workload.

The procedure of decomposition Eulerian domain starts with the creating of background grid and identification of ghost and fluid cells by computing indicator functions. By providing connectivity information of a graph to METIS with appropriate weighting information, an index array is returned with an integer indicating in which processor a cell is placed. Figure 4(b) shows a problem with a circle representing solid interface situated in the center of Cartesian grid is decomposed into 4 partitions. Each partition acquires 3 layers of cells from neighbor partitions to be the overlapping zone. As shown in Figure 4(c), a sub-domain occupied by a processor, is then defined by the combination of a processor’s partition and overlapping zone. The overlapping zone serves as a buffer region for synchronized of information.



(a) **(b)** **(c)**

Figure 4. The Eulerian sub-domain with sharp interface method; (a) Ghost cell method using interpolation of image point to impose boundary condition (b) A circular solid interface located inside two-dimensional domain partitioned by 4 processors (c) Definition of overlapping zone for a sub-domain

B. Domain decomposition of Lagrangian interface

The parallelisms for Lagrangian interface considered are task parallelism, atomic decomposition, and spatial decomposition. An example in Figure 5 illustrates how a problem having Lagrangian markers denoted in circle superposing on the Eulerian Cartesian grid could be decomposed with 4 processors.

The first feasible parallelism is task parallelism which is reserving specific processors to dealing with Lagrangian data exclusively and using other processors for calculation of Eulerian data. This approach takes advantage of the concurrency of Eulerian and Lagrangian tasks to accelerate computation. One of the difficulties is the optimal arrangement of fair work load among Eulerian and Lagrangian tasks for general problems. The ratio of workload between Eulerian and Lagrangian domain is very problem-dependent and varies with the evolution of the interface. Complex management is required to avoid processes idling due to racing among tasks. Moreover, because of dependency between Eulerian and Lagrangian variables, data exchange under task parallelism intensely relies on all to one and one to all collective communication that may lead to communication overhead.

On the other hand, data parallelism distributes the entire data over processors to be processed concurrently. Under the data parallelism, one of the choice to decompose Lagrangian interface is atomic decomposition[22] as illustrated in Figure 5(c). Atomic decomposition evenly distributes Lagrangian markers to all processors regardless their locality. It achieves load balance automatically. However, frequent communication for updating interface information between processors due to low locality is inevitable. Thus deterioration of parallel efficiency is expected arising from communication overhead.

Unlike atomic decomposition, spatial decomposition spreads entire Lagrangian interface into sub-domains on the basis of spatial inhabitation of markers. In Figure 5(d), Lagrangian markers encompassed by a sub-domain owned by a particular processor should be assigned to this processor. This approach takes the advantage of data locality that interface information is available in each processor, and both communication frequency and data exchanged can be minimized. The load balance of Lagrangian computation may not be satisfied but reasonable scalability is expected in general cases since the cost of pure calculation of Lagrangian data is much smaller than cost of flow solver. Besides, the methodology of spatial decomposition of the Lagrangian domain is similar to Eulerian domain that makes the implementation straightforward. As a result, the spatial decomposition is considered as the partitioning approach for the Lagrangian interface in this study.

The procedure of Lagrangian domain decomposition is summarized as following. A marker locality array is used to account marker inhabitation by providing a primary processor that situates the marker. As shown in Figure 6, localities of the solid and dashed markers inside a particular Eulerian sub-domain are assigned with this primary processor ID. The collection of partition-own markers, represented by solid circle, is the sub-set of local Lagrangian sub-domains that operations looping on, and dashed markers in overlapping zone are served as buffers whose information is updated by neighboring processors at the end of each time step. Note that the global Lagrangian interface connectivity will not be discarded but preserved for repartitioning. Repartitioning is necessary when a marker moves out of its Eulerian sub-domain. In addition, any geometrical operation such as coarsening, refining, and smoothing on Lagrangian interface affecting global connectivity requires repartitioning to keep consistent connectivity. In such case the data locality has to be updated and Lagrangian markers repartitioned

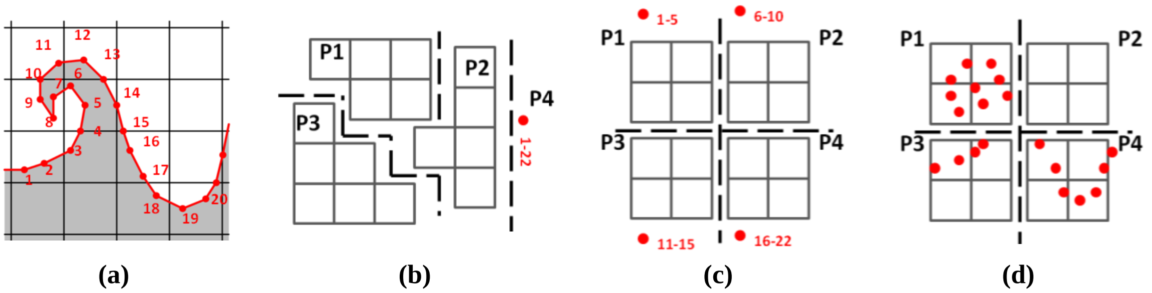


Figure 5. Strategies of decomposition of Lagrangian interface (a) 2D Lagrangian mesh on Eulerian domain (b) task parallelism (c) atomic decomposition (d) spatial decomposition

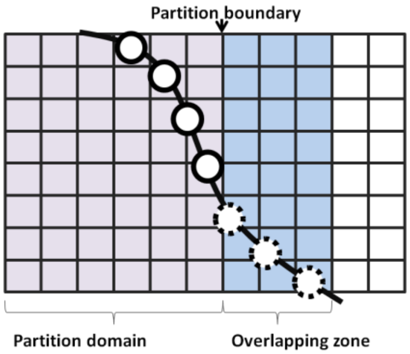


Figure 6. Lagrangian markers inhabiting in a sub-domain

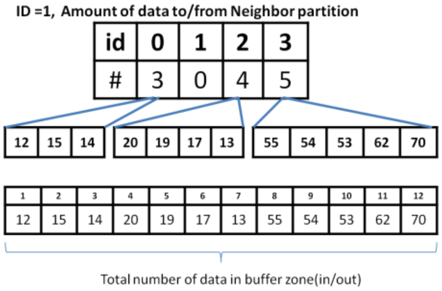
C. Communication methodology

Data in an overlapping zone of a sub-domain needs to be synchronized with neighbor sub-domains for data consistency. For point-to-point communication, i.e., a sub-domain synchronizing data in overlapping zone with neighbor partitions, the non-blocking send and receive are used to avoid blocking delay. The sent-out data of a sub-domain is collected and concatenated into an array on the sequence add-up manner. Figure 7(a) illustrates how the data in overlapping zone is grouped in a chunk of continuous memory. This problem is executed with 4 processors and the partitioned domain of processor 1 is adjacent to processor 0, 2 and 3 with 3, 4, and 5 entries in overlapping zone respectively. A temporary array is then created by concatenation of prepared data. The pointer of this array is passed to the subroutine, *MPI_ISEND* to initialize communication process. The procedure of receiving data from neighbor partitions has similar approach by allocating one-dimensional space beforehand for arriving data. An

MPI_WAIT is placed in the end of initialization of send and receive subroutines to check if procedures are completed correctly. The pseudo code of the communication procedure is shown in Figure 7(b). This design allows a processor finishing synchronization earlier to work on following operations earlier. Such design provides a sort of overlapping between communication and computation that improves parallel efficiency.

The performance of our methodology is evaluated with a one million cell problem. The problem is executed on the NYX machine in the Center of Advance Computing at University of Michigan. The nodes we used on NYX are comprised with Intel Nehalem Xeon E5540 CPU with InfiniBand networking whose latency is about 10^{-6} second. Each Intel Xeon E5540 CPU has 4 cores with 8M Bytes L3 shared cache and total memory available per CPU is 12G Bytes.

The averaged data size of velocity exchanged in a process of communication is from 108K to 41 Bytes on 4 to 96 processors. The start-up time represents the preparation and calling of *MPI_ISEND* and *MPI_IRECV* procedures, and the time to finish synchronization stands for time staying on *MPI_WAITALL*. The cost of one velocity synchronization is on the order of $2*10^{-5}$ second in spite of number of processers used. Our communication strategy has been tested and trivial latency is observed. This communication design is applied for both Eulerian domain and Lagrangian domain communication in present work.


ID =1, Amount of data to/from Neighbor partition
id
0
1
2
3
#
3
0
4
5
12
15
14
20
19
17
13
55
54
53
62
70
1
2
3
4
5
6
7
8
9
10
11
12
12
15
14
20
19
17
13
55
54
53
62
70
Total number of data in buffer zone(in/out)

```
Do : index0 = 1, total number of processors
    IF NOT neighbor partition, CYCLE
    recv → recv_data(index0); source = index0
    tag1 = processor_id
    CALL MPI_IRECV(recv, # of receive, type, source , tag1, comm., reqs)
    send → send_data(index0)
    tag2 = processor_id;  dest. = sent entry index
    CALL MPI_ISEND(send, # of send    , type, dest.   , tag2, comm, reqs)
CALL MPI_WAITALL(reqs, stat, comm)
```

(a) **(b)**

Figure 7. (a) An example of concatenated array of data in overlapping zone of processor ID 0 on 4 processors computation (b) Pseudo code of non-block receiving/sending communication

Table 1. Communication cost for velocity filed in 1 millions cells problem

# of processors	**4**	**8**	**16**	**64**	**96**
Start-up time(s)	3.0*10^{-5}	2.4~10^{-6}	2.0*10^{-5}	2.1*10^{-5}	2.1*10^{-5}
Time to finish synchronization(s)	9.5*10^{-6}	9.7*10^{-6}	9.5*10^{-6}	1.0*10^{-5}	1.1*10^{-5}
Averaged data size (K Bytes)	108	105	84	50	40.5

D. Solving Procedure

We summarize the procedures of parallel Eulerian-Lagrangian method in Figure 8. Note that those operations in italic involving both Eulerian and Lagrangian variables. First of all, the initialization of simulation creates a Cartesian grid to encompass solid interface and refine it adaptively to designated resolution with the identification of ghost cells. The Eulerian domain is then partitioned by METIS, and the Lagrangian domain is spatially decomposed based on marker's locality. In the time integration loop, a marker is updated to new position by a velocity field interpolated from the Eulerian domain and interface shape will be refined if needed. The updated interface renders a new distribution of fluid properties that inter-processors communication is needed to retain properties such as consistency at overlapping zone of sub-domains. Two source terms, surface tension and mass transfer across fluid interface distributes their impact to cells at a fixed distance. The velocity field is updated with Runge-Kutta Crank-Nicolson method by projection method. PETSc and HYPRE are both implemented as the linear solver for the discretized Poisson equation. Dynamic mesh refinement, which is a complex but computationally efficient approach is not addressed in the present study but will be implemented in the near future. Once the divergence-free velocity filed is obtained on Eulerian domain, it is projected to Lagrangian markers by Dirac delta function.

Initialization: Eulerian domain decomposition(METIS)
Lagrangian domain decomposition

Time integration loop :

Marker movement
Interface Restruction: Smooth, refine, coarsen
Determine material and cell properties
Source term: *Surface tension*
Mass transfer/ Heat flux at fluid interface
Intermediate velocity U* by RK-CN integration
Pressure Poisson equation using *HYPRE/PETSc*
Corrected velocity U^{n+1}
Energy equation by RK-CN integration

Dynamic adaptive mesh refinement

- Eulerian domain decomposition(ParaMetis)
- Lagrangian mesh decomposition
- data reallocation

Velocity interpolation to Lagrangian markers
Reconstruction of Lagrangian interface

Figure 8. The solving procedure of parallel Eulerian-Lagrangian method

IV. Validation and Performance of Parallel Implementation

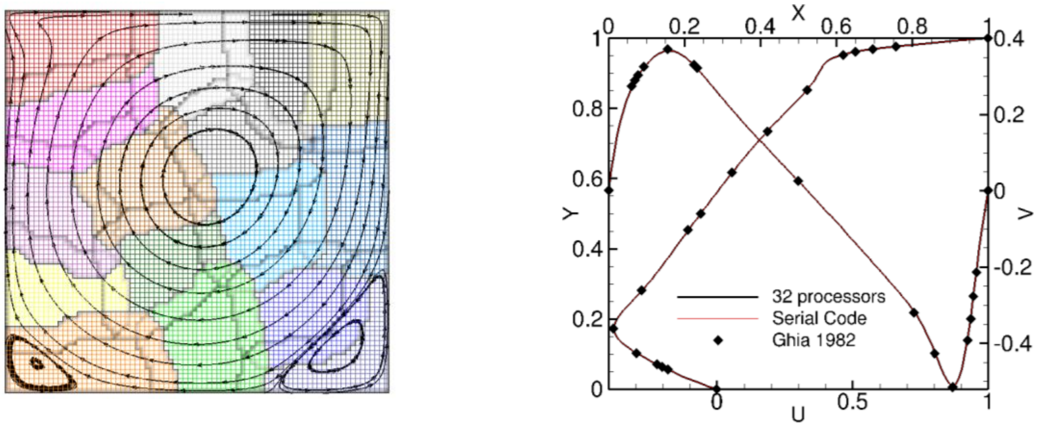
A. Validation

Here we validate the parallel Eulerian-Lagrangian method by comparing solutions from serial and parallel computation: the lid-driven cavity flow, the uniform flow past a cylinder with sharp interface method, and a natural convection problem to verify the energy equation. The three tests are computed with ENO scheme around solid boundary and second order central difference scheme in the rest region.

The first case is lid-driven cavity flow at Reynolds number 1000 with grid size 80 by 80. The streamline of at steady state together with colored sub-domain mesh from 16 processors computation is given in Figure 9(a). The overlapping zone between sub-domains having three layers of cell width is shaded to highlight overlapping. Computational results confirm the solution produced by parallel code is identical with solution from serial code in an error range of 10^{-7}, which is the convergence criterion of the pressure Poisson equation. The velocity profile shown in Figure 9(b) and agrees with Ghia's benchmark solution.

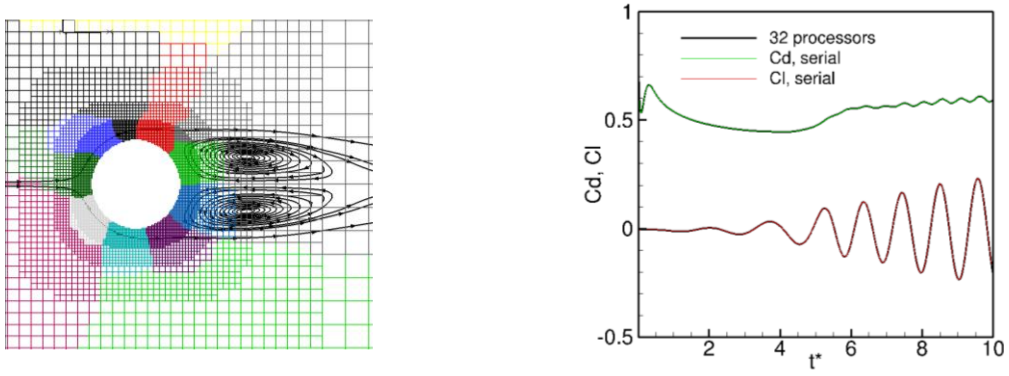
The second test is the uniform flow past a cylinder at Reynolds number 40 and 200 with 5 layer of static mesh adaption around solid interface on 32 processors. At Reynolds number 40, the computed drag coefficient is 1.52 and length of circulation zone is 2.23 which are same as result from serial code. The time-evaluation of drag and lift coefficient at Re = 200 are plotted for parallel and serial computation in Figure 10(b). Drag and lift coefficient computed by the parallel solver is marked with thick black line and the solution form the serial solver in green and red lines. It is clear that parallel code produce numerical identical solutions to the serial code. The solution of parallel code agrees with solution of serial code up to 9 digits, which is the convergence criterion of the Poisson equation.

The natural convection in a square cavity at Rayleigh number 10^5 is selected as the validation test for examining the parallel implementation of energy equation. Dirichlet temperature boundary conditions are applied at left and right wall and insulated wall boundary conditions for top and bottom wall of the cavity. The grid size is 80 by 80 with 16 processors used in this simulation. The sharp interface method is used as the enforcement of the boundary condition of solid interface. The temperature contour and plots of temperature profile along horizontal center and vertical center are presented in Figure 11. The solution of the parallel computation agrees with the serial computation exactly. These three tests demonstrate of correct implementation of the parallel Eulerian-Lagrangian method.



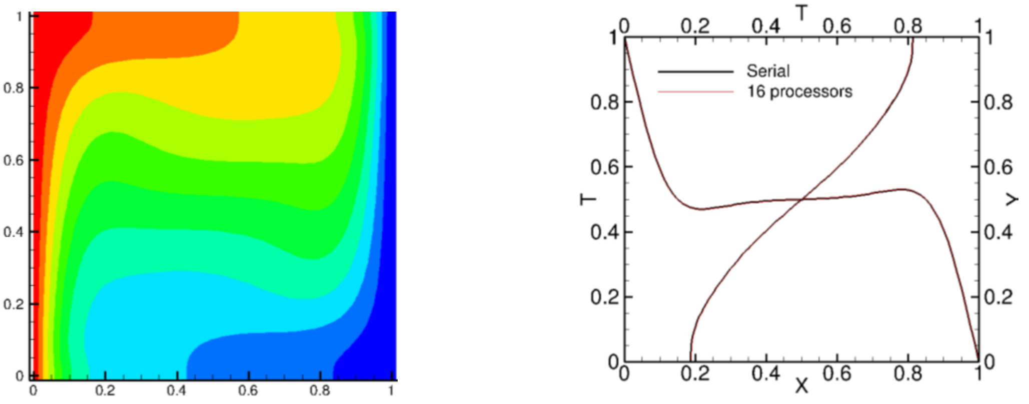
(a) **(b)**

Figure 9 Lid-driven cavity flow at Re = 1000; (a) Stream line on 16 processors; The shadow of edges represents overlapping zone between sub-domains. (b) Velocity profile of U and V along the central vertical line and central horizontal line of cavity; grid size 128*128 on 32 processors at.



(a) **(b)**

Figure 10 Uniform flow past circular cylinder 32 processors (a) Stream line plot at Re = 40; The drag coefficient is 1.52 and the length of circulation zone is 2.24 (b) Lift and drag coefficient of uniform flow past circular cylinder at Re = 200



(a) **(b)**

Figure 11 Natural convection in square cavity at Rayleigh number 10^5 on 16 processors; (a) Temperature contour of natural convection (b) Temperature along vertical center and horizontal center of the square cavity

B. Parallel performance

We evaluate parallel performance by simulating the lid-driven cavity flow with three problem size: problem 1 has 1.6x10^5 cells; problem 2 has 1 million cells, and problem 3 has 4 million cells. This experiment is executed on the NYX machine. Its specification has been addressed in the preceding section.

The linear solver library used in this test is PETSc' conjugate gradient method with Jacobi preconditioner. Since the graph of a linear system of discretized Poisson equation is changed with the number of partitions, the iteration number reaching a constant rounding-error varies with the number of processors used. In order to control the workload spent on solving a linear system on varying number of processors used in a computation, executing time is measured under the basis of the fixed iteration number of the linear solver.

It is observed that up to 90% of cost of a time step is spent on solving the linear system of discretized Poisson equation. As a result, overall parallel performance is dominated by the efficiency of linear solver, which is controlled by two effects: the Computation-to-communication ratio and shared cache size per CPU. For a fixed-size problem, increasing the number of processor decreases computation-to-communication ratio that results in the slowdown of speedup. In case of shared cache effect, its impact to parallel performance is related to the size of working data in solving a linear system. A typical parallel program assigns a fraction of the entire data to each processor. It occurs that the entire data set does not fit into the cache on a single processor execution but a parallel computations executing on the same problem with multiple processors may have fraction of entire data assigned to each processor fits in its local shared cache. In case of a computation having decomposed working data fully cached, it leads to higher hit rate and produces super-linear speedup.

The executing time and speedup is shown in Figure 12. In problem 1, the speedup increases gradually with increasing number of processor but slowdown on 16 processors due to decreasing computation-to-communication ratio. In problem 2, super-linear speedup due to higher cache hit rate is observed in between 64 to 108 processors and the slowdown of speedup due to decreasing computation-to-communication ratio on 140 processors. Problem 3 has 4 millions cells such that the working data is way too large with respect to the 8M-Bytes L3 cache on serial execution, and results in significant low cache hit rate that overestimates the speedup of multiple-CPU computation. The parallel efficiency gradually ascends to 1.2 on 240 processors (60 CPUs). Deterioration of the efficiency arising from the decreasing computation-to-communication ratio is not observed in the scope of 240 processors.

Parallel efficiency on the basis of number of cells per processor is given in Figure 13. In general, the efficiency reaches peak roughly at 1.5x10^4 cells per processor and deterioration due to decreasing computation-to-communication ratio is around 10^4 cells per processor. It suggests that parallel efficiency doesn't benefit from further decomposition of entire data. On the other hand, relative low efficiency at larger number of cells per processor is observed because of lower cache hit rate. Not until the fraction of working data can be cached is the highest efficiency reached. As a result, it is said the 8M-Bytes shared cache affects the behaviors parallel efficiency at larger problems. Moreover, for working data that can be fully cached on serial execution, the cache effect on parallel efficiency will not be seen as the case of 0.16 million cells problem. The overall performance of the Eulerian-Lagrangian method demonstrates outstanding efficiency for problems in the scope of several millions of cells on Intel Nehalem Xeon architecture.

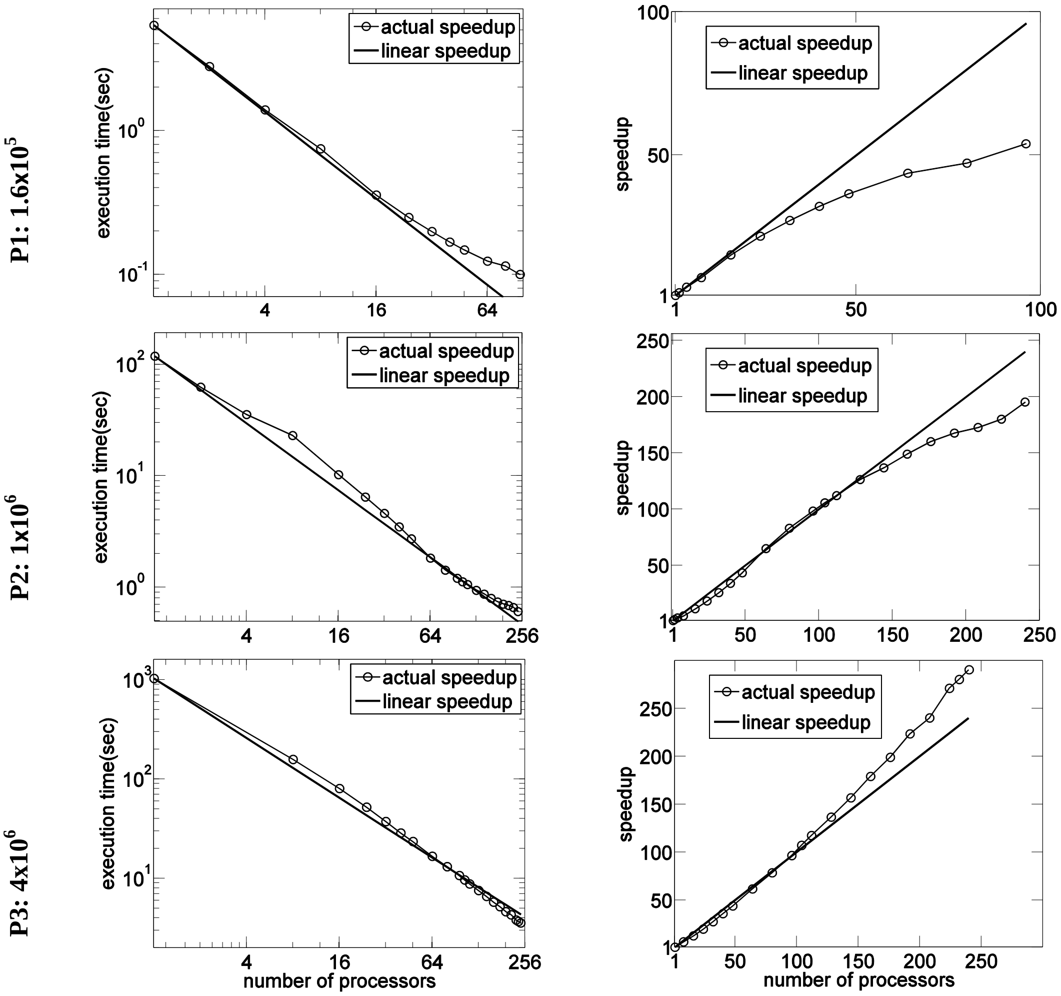


Figure 12. Executing time and speedup as a function of number of processors for various problem size

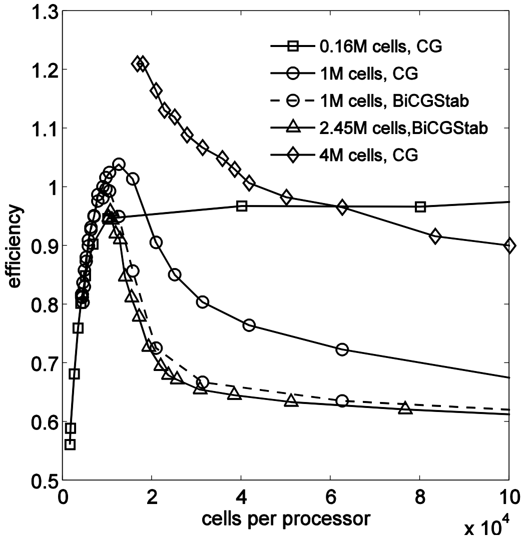


Figure 13. Parallel efficiency with respect to number of cells per processor

V. Self-pressurization in a Spacecraft Cryogenic Fuel Tank

The self-pressurization in a cylindrical liquid hydrogen fuel tank is considered. Figure 14(a) shows the geometric configuration of the test case. The ground test at normal gravity is assumed since the interface can be more distorted and the liquid position and vapor bubble distributions are usually unknown under microgravity condition. The axisymmetric numerical computation in Figure 14(b) is conducted due to its effective computation in order to establish a guideline on the amount of self-pressurization at a given heat leak.

First, q_H = 1.0 W/m^2 case is simulated with uniform heating distribution at 50% fill level in order to understand the self-pressurization mechanism in a cryogenic fuel tank. For studying the influence of heat amount, four different uniform heating cases (q_H=0.1, 0.5, 1.0, and 2.0 W/m^2) are simulated with 50% fill level. Investigation of the liquid fill-level effect is followed for uniform heating case with three different fill-levels of 25%, 50% and 75%. The total heat power input is Q_T=4.71mW for q_H=0.1 W/m^2 case, and intial temperature is same to saturation temperature for all cases. The used material properties at T=20.369 K and non-dimensional parameters are summarized in the Table 2 and Table 3.

Table 2. Material properties at T=20.369 K.

	Vapor hydrogen	Liquid hydrogen
Density, ρ [kg/m^3]	1.3322	70.850
Thermal conductivity, k [J/m·s·K]	0.017085	0.10382
Dynamic viscosity, μ [kg/m·s]	1.0786×10^{-6}	1.3320×10^{-5}
Specific heat, C_p [J/kg·K]	1.2036×10^4	9.772×10^3
Thermal diffusivity, α [m^2/s]	2.56×10^{-6}	1.82×10^{-7}
Thermal expansion coefficient, β [K^{-1}]	0.04909	0.0168
Surface tension, σ [N/m]	1.9452×10^{-3}	
Latent heat, L [J/kg]	4.456×10^5	
Normal boiling temperature, T_B [K]	20.369	
Normal saturation pressure, P_s [Pa]	1.01400×10^5	

Table 3. Reference scales and non-dimensional parameters.

	Reference case ($g = g_0$, $q_H = 1\,W/m^2$)
Reference length scale, L [m]	0.05
Laplace number, $La = \frac{\sigma\rho_l L}{\mu_l^2}$	3.88×10^7
Bond number, $Bo = \frac{\rho_l g L^2}{\sigma}$	893
Prandtl number, $Pr = \frac{\nu_l}{\alpha_l}$	1.033
Rayleigh number, $Ra = \frac{g\beta\Delta T L^3}{\alpha_l \nu_l} = \frac{g\beta q_H L^4}{k_l \alpha_l \nu_l}$	2.90×10^8

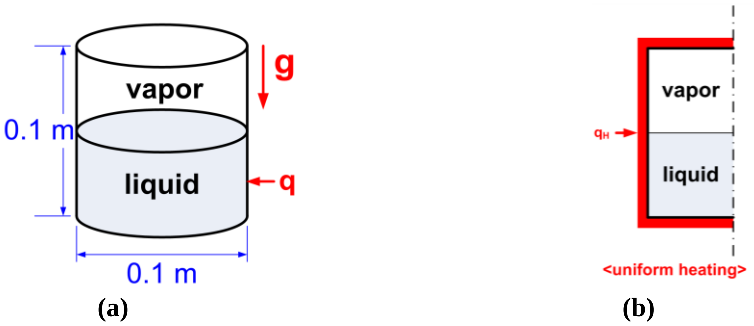
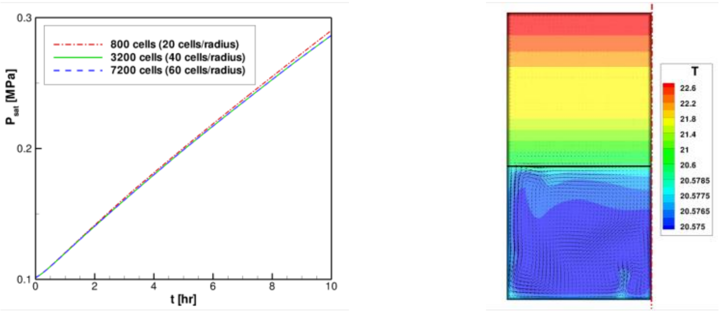


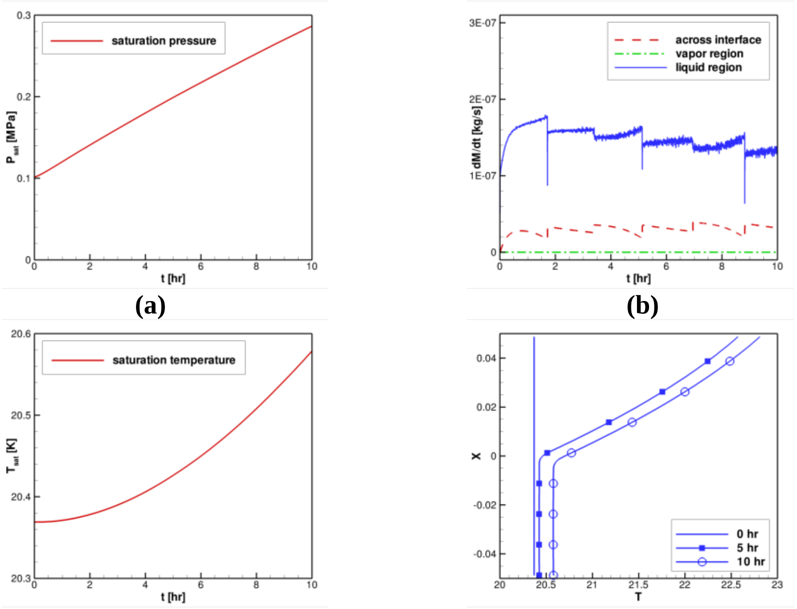
Figure 14. The numerical configuration. (a) Fuel tank geometry. (b) Axisymmetric computational domain with uniform tank-wall heat distributions.

Uniform heating case with heat flux q_H = 1.0 W/m^2 at 50% fill level is first tested in order to understand the self-pressurization mechanism in a cryogenic fuel tank. The grid convergence test in Figure 15 shows almost identical results for saturation pressure computation when the grid is finer than 3200cells with adaptive meshes. From the temperature contour and velocity vector plot, we may need finer grid in order to resolve the flow fields up to desirable resolution, but in the present study, 3200-7200 cells are used for computation of self-pressurization with the help of adaptive mesh refinement. Figure 16(a) shows the saturation pressure change, which increases with time, but the rise rate decreases and almost constant after transient time. The implemented first principle-based phase change model is supported in Figure 16(b), where the rate of phase change for each phases and on the interface surface are depicted. It is shown that the phase change in liquid phase has dominant contribution comparing phase change on the interface surface. As expected, the phase change in vapor region is negligible since the vapor temperature is higher than saturation temperature. The rate of phase change in liquid phase is dominant, but decreases in time as the saturation temperature rises and the heat is used to warm up the liquid to the saturation temperature before vaporization and to heat up the vapor on the top surface. The increasing saturation temperature and following thermal stratification in the vapor region is depicted in Figure 16.



(a) (b)

Figure 15. A sensitivity study on the spatial resolution. (a) grid convergence test. (b) Temperature contour with velocity vector.



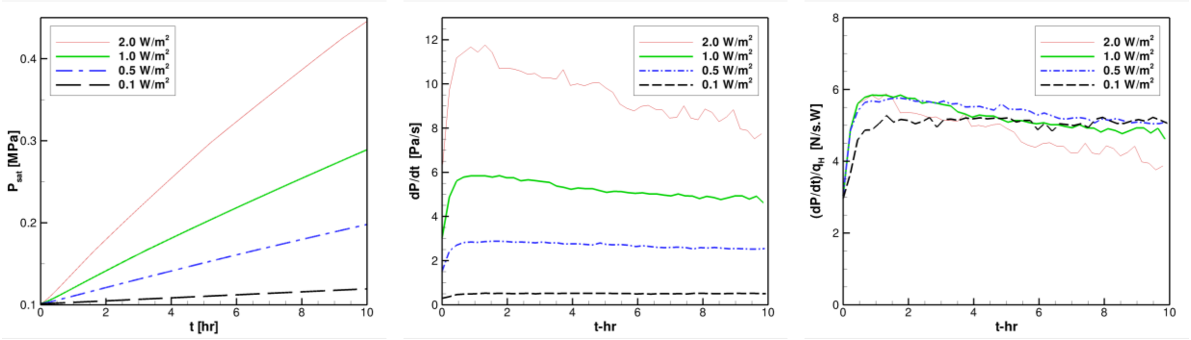
(a) (b)

(c) (d)

Figure 16. Self-pressurization results with uniform q_H=1.0W/m^2 at 50% fill level. (a) The saturation pressure change. (b) The rate of phase change for each phases and interface surface. (c) Saturation temperature change. (d) Temperature profile along center line of the cylindrical tank.

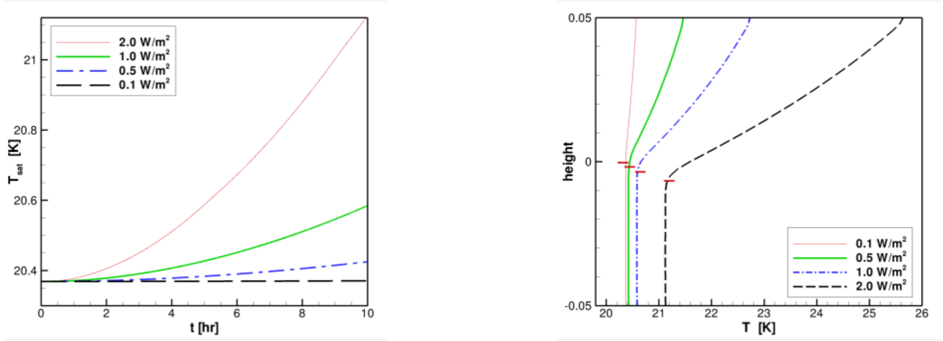
A. Influence of heat flux amount

The effects of heat flux amount are studied with the same 50% fill level and four different heat flux amount; 0.1, 0.5, 1.0, 2.0 W/m^2, where the total heat power is 4.71 mW for 0.1 W/m^2 heat flux case. In Figure 17(a), the saturation pressure increases in time as vaporization occurs due to heat influx from the tank walls. The saturation pressure increases more rapidly with increasing heat flux amount as expected and previous researches addressed. The pressure rise rate is plotted in Figure 17(b), where the smallest heat flux case with 0.1 W/m^2 shows almost constant pressure rise rate after short transient period due to weak natural convection and almost constant saturation temperature. The larger heat amount shows longer transient period with steeper initial pressure rise, but the rate decreases as time goes. In Figure 17(c), the pressure rise rates are normalized by the heat flux amount, and they are not perfectly proportional to the heat amount. Except the smallest heat flux case, the pressure rise rate is high in the beginning, and then decreases. It can be noticed that the higher heat flux amount produces less pressure increase per heat flux amount after transient period. This can be explained in Figure 18. With higher heat influx, the saturation temperature increases faster, and the absorbed heat is utilized to warm up the liquid before vaporization since liquid vaporization occur when the liquid temperature reaches saturation temperature. It is also shown that the thermal stratification in vapor region increases as heat flux amount increases, and thus, the heat absorbed is used for increasing vapor temperature instead of vaporizing liquid.



(a) (b) (c)

Figure 17. The influence of heat flux amount on the saturation pressure. (a) Saturation pressure (b) pressure rise rate (c) pressure rise rate normalized by heat flux amount.



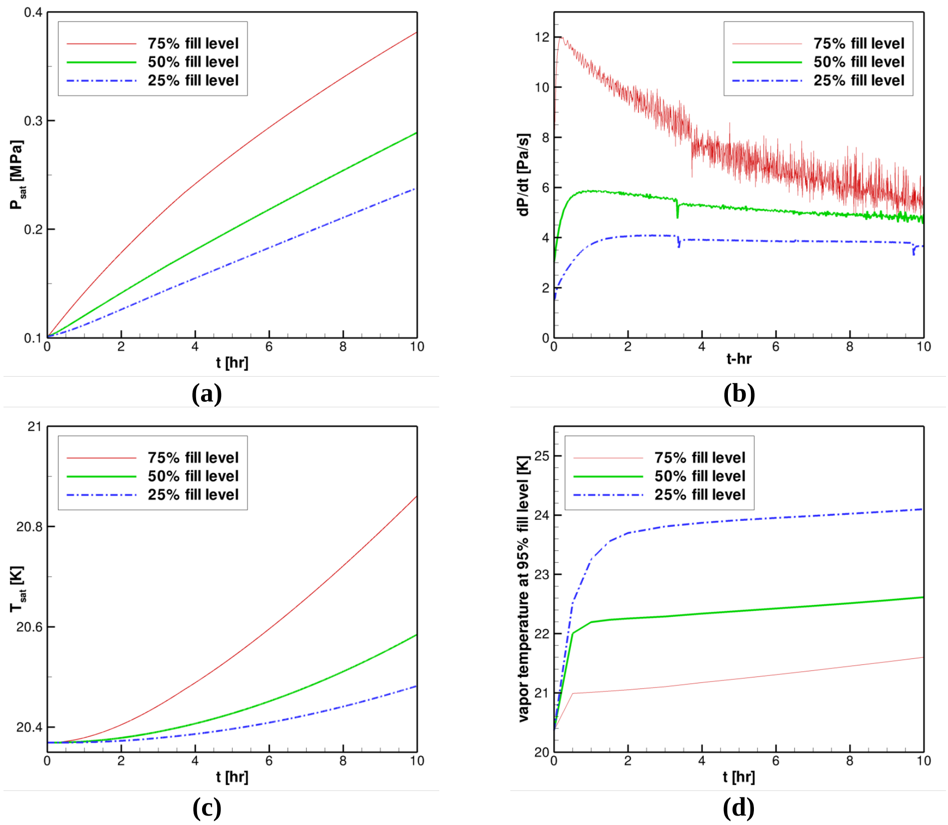
(a) (b)

Figure 18. The influence of heat flux amount on the saturation temperature. (a) Saturation temperature (b) temperature profile along storage center line.

B. Effect of liquid fuel fill-level

The influence of the liquid fill-levels is investigated with three different levels; 25%, 50%, and 75%. The higher fill level shows higher saturation pressure in Figure 19(a). The pressure rise rate of Figure 19(b) shows that 75% fill-level has dramatic rise rate in the beginning due to faster heat transport from the top surfaces and larger wet surface area where heat is absorbed directly into liquid. Furthermore, the small volume ratio of a vapor region results in faster pressurization since even the same amount of phase change makes higher density and pressure rise of vapor in a smaller vapor volume. In lower fill-level, the absorbed heat is used to increase vapor temperature instead of vaporization of liquid. From the vapor temperature history of Figure 19(d), we can find that more heat remains in the vapor region for lower fill level. Therefore, the pressure will increase with increasing fill level as it is in the present study. However, such a huge difference in the pressure rise rate diminishes after transient time.

In the present study, higher fill level shows higher pressure rise rate. However, many other previous studies showed quite different results in the influence of the liquid fill level. The experimental studies in an oblate spheroidal storage, the pressure rise rate at 29% is higher than 83%, and 49% shows the lowest pressure rise[3, 4]. On the other hand, the lower fill level shows higher pressure rise rate in the homogeneous model[6]. The lumped vapor model shows the same trends with homogeneous model, but the trend are the opposite in the beginning[6]. The homogeneous model assumes homogeneous condition on the whole storage, and the lumped vapor model assumes homogeneous condition for vapor region, and these assumptions may cause these difference. Moreover, in the present test, the cylindrical storage tank is tested unlike the oblate spheroidal tank in other studies. The different fill level in the oblate spheroidal tank has large influence on the liquid surface area and vapor volume ratio. Thus, in order to understand the effect of fill level correctly and explain these disagreements, more parameter studies are needed including tank shapes, tank sizes, the liquid surface area, wet surface area and the vapor volume ratio.



(a)

(b)

(c)

(d)

Figure 19. The effects of liquid hydrogen fill-level for fuel tank pressurization. (a) The saturation pressure in a fuel tank. (b) The saturation pressure rise rates. (c) Saturation temperature. (d)The vapor temperature history at 95% fill level.

VI. Summary and Conclusions

In this paper, a 3-D adaptive Eulerian-Lagrangian method is further developed with multi-domain parallel computation technique for the large scale multiphase flow computation. Spatial domain decomposition parallelism is adopted for both Eulerian and Lagrangian domain. The multi-domain parallel computation presents the capability for various large scale problems including arbitrary solid boundary and thermal effects, and the parallel performance of the current Eulerian-Lagrangian method shows remarkable speedup for problems in the scope of several millions cells on Intel Nehalem Xeon architecture. The highlights of parallel computing of the Eulerian-Lagrangian method are addressed as following.

(1) The interactions between Eulerian and Lagrangian domain results in difficulties of problem decomposition, especially communications complexities between both domains. With the spatial domain decomposition for Lagrangian markers based on the their inhabitation to an Eulerian sub-domain, communication cost over processors could be minimized,

(2) The communication methodology reported in the current study shows that latency of a synchronization process is weakly dependent on the local data size of a processor,

(3) Experiments show monotonically ascending speedup until the problem is over-partitioned, Parallel efficiency reaches the peak around 1.5x10^4 cells per processor, which is a trade-off between cache hit rate and communication latency.

The first principle-based phase change model is introduced by computing the rate of phase change in each phase as well as across the liquid/vapor interface. Specifically, the following observations can be made in the fuel tank self-pressurization case:

(1) The rate of phase change in a liquid phase is dominant, and should be considered with the phase change on the liquid surface in the fuel tank self-pressurization. The rate of phase change decreases with increasing saturation temperature,

(2) With increasing heat flux amount, the pressure rise rate and thermal stratification in vapor region also increase and larger heat amount shows longer transition with steep initial pressure rise,

(1) Higher liquid fill level shows higher pressure rise rate due to smaller vapor region and larger wet surface area, but the rate becomes similar after transient time with steep pressure rise. The liquid fill level determines various important parameters such as the liquid volume, wet area, liquid surface area and their ratios according to its geometric tank shape, and more parameter study is required to understand the influence of liquid fill level correctly.

Acknowledgments

The work reported in this paper has been partially supported by NASA Constellation University Institutes Program (CUIP), Claudia Meyer and Jeff Rybak program managers. We have benefited from the discussion with John Peugeot and Jeff West of NASA Marshall Space Flight Center.

References

[1] Hastings, L.J., Plachta, D.W., Salerno, L., and Kittel, P., "An overview of NASA efforts on zero boiloff storage of cryogenic propellants," *Cryogenics*, Vol. 41, No. 11-12, 2001, pp. 833 - 839.

[2] Lin, C., Dresar, N.V., and Hasan, M., "Pressure Control Analysis of Cryogenic Storage Systems," *Journal of Propulsion and Power*, Vol. 20, No. 3, 2004, pp. 480-485.

[3] Aydelott, J.C., "Effect of gravity on self-pressurization of spherical liquid-hydrogen tankage," NASA-TN-D-4286, 1967.

[4] Aydelott, J.C. and Spuckler, C.M., "Effect of size on normal-gravity self- pressurization of spherical liquid hydrogen tankage," NASA-TN-D-5196, 1969.

[5] Hasan, M.M., Lin, C.S., and Van Dresar, N.T., "Self-pressurization of a flightweight liquid hydrogen storage tank subjected to low heat flux," NASA-TM-103804, 1991.

[6] Barsi, S. and Kassemi, M., "Numerical and experimental comparisons of the self-pressurization behavior of an LH2 tank in normal gravity," *Cryogenics*, Vol. 48, No. 3-4, 2008, pp. 122 - 129.

[7] Poth, L.J. and van Hook, J.R., "Control of the Thermodynamic State of Space-Stored Cryogens by Jet Mixing," *Journal of Spacecraft and Rockets*, Vol. 9, No. 5, 1972, pp. 332-336.

[8] Aydelott, J.C., "Modeling of space vehicle propellant mixing," NASA-TP-2107, 1983.

[9] Hochstein, J.I.J.H.-C.A.J.C., "Prediction of self-pressurization rate of cryogenic propellant tankage," *Journal of Propulsion and Power*, Vol. 6, 1990, pp. 0748-4658.

[10] Panzarella, C.H. and Kassemi, M., "On the validity of purely thermodynamic descriptions of two-phase cryogenic fluid storage," *Journal of Fluid Mechanics*, Vol. 484, 2003, pp. 41-68.

[11] Agbaglah, G., Delaux, S.a., Fuster, D., Hoepffner, J.e., Josserand, C., Popinet, S.h., Ray, P., Scardovelli, R., and Zaleski, S.h., "Parallel simulation of multiphase flows using octree adaptivity and the volume-of-fluid method," *Comptes Rendus M?anique*, Vol. 339, No. 2-3, pp. 194-207.

[12] Sussman, M., "A parallelized, adaptive algorithm for multiphase flows in general geometries," *Computers & Structures*, Vol. 83, No. 6-7, 2005, pp. 435 - 444.

[13] Singh, R. and Shyy, W., "Three-dimensional adaptive Cartesian grid method with conservative interface restructuring and reconstruction," *Journal of Computational Physics*, Vol. 224, No. 1, 2007, pp. 150-167.

[14] Uzgoren, E., Sim, J., and Shyy, W., "Marker-based, 3-D adaptive Cartesian grid method for multiphase flow around irregular geometries," *Communications in Computational Physics*, Vol. 5, No. 1, 2009, pp. 1-41.

[15] Sim, J. and Shyy, W., "Interfacial flow computations using adaptive Eulerian-Lagrangian method for spacecraft applications," *International Journal for Numerical Methods in Fluids*, 2010.

[16] Peskin, C.S., "The immersed boundary method," *Acta Numerica*, Vol. 11, 2003, pp. 479-517.

[17] Juric, D. and Tryggvason, G., "Computations of boiling flows," *International Journal of Multiphase Flow*, Vol. 24, No. 3, 1998, pp. 387-410.

[18] Shin, S. and Juric, D., "Modeling three-dimensional multiphase flow using a level contour reconstruction method for front tracking without connectivity," *Journal of Computational Physics*, Vol. 180, No. 2, 2002, pp. 427-470.

[19] Balay, S., Brown, J., Buschelman, K., Gropp, W.D., Kaushik, D., Knepley, M.G., McInnes, L.C., Smith, B.F., and Zhang, H., *PETSc Web page*.

[20] Karypis, G. and Kumar, V., *MeTis: Unstructured Graph Partitioning and Sparse Matrix Ordering System, Version 4.0*. 2009, \url{http://www.cs.umn.edu/~metis}.

[21] Schamberger, S. and Wierum, J.-M., *Graph Partitioning in Scientific Simulations: Multilevel Schemes versus Space-Filling Curves*, in *Parallel Computing Technologies*, V. Malyshkin, Editor. Springer Berlin / Heidelberg. 2003, p. 165-179.

[22] Plimpton, S., "Fast parallel algorithms for short-range molecular dynamics," *J. Comput. Phys.*, Vol. 117, 1995, pp. 1--19.