SPE Flow Assurance Technology Congress
Rio de Janeiro, Brazil, November 15 – 18 2022

GPU Accelerated Always Hyperbolic Slug Capturing Method

Farhad Nikfarjam[1], Hamidreza Anbarlooei[1*], Daniel O. A. Cruz [2], Celso Dresjan Junior[3]
[1] Department of Applied Mathematics, Federal University of Rio de Janeiro, Brazil, hamidreza@im.ufrj.br.
[2] Mechanical Engineering Program, Federal University of Rio de Janeiro, RJ, Brazil.
[3] Petrogal Brasil.

Abstract
A new slug capturing method, based on the 5-equation (always hyperbolic) formulation is developed. The governing equations are solved using a simplified HLLC solver, and the excessive smearing of the equations is handled by preconditioning technique. To accelerate the computation, the solver is implemented completely on the GPU where the GPU's shared memory is used extensively to minimize the costly global memory access. The method predicts the experimental data very accurately, even for the intermittent flows with very low frequencies where other numerical methods usually fail. The method also results in statistically developed waves for long time integrations, property which other methods based on the common 4-equation two-fluid model lacking. The GPU implementation is compared with the parallel CPU version. For a typical industrial test case (12 km), the GPU code performs up to 140 times faster than CPU with 10 cores.

Keywords
Slug capturing; HLLC Solver; GPU acceleration;

Introduction
The slug capturing method solves the two-fluid model equations in conjunction with a minimal set of empirical closures models. This method is well suited for simulating transient problems, such as start-up, shutdown and depressurization of the pipelines. However, the 4-equation two-fluid model (conservations of mass and momentum for each phase) is not always mathematically hyperbolic and in several important industrial configurations the underlying equations become ill-posed [1]. As a result, unphysical waves start to produce and propagate in the solution. In such cases, one obtains a chaotic set of waves (slugs) during simulation, which mostly interpreted as the (length/frequency) distribution of the slugs. However, it is easy to show that in a carefully designed test problems, without any source of randomness, even after very long integration times, such randomness does not disappear and the waves does not converge to a statistically developed and identical solution. Numerically regularization techniques try to solve this problem by adding an especial interface pressure term [2]. However, this technique is not always able to solve the ill-posed problems.
On the other hand, mathematically regularization technique can eliminate this problem totally. In this technique a new volume fraction evolution equation will be added to the governing equations. Here, the first equation in system (1)

$$(1)\begin{cases} \partial_t \alpha_g + u_i \partial_x \alpha_g = r_p(p_g - p_l) \\ \partial_t(\alpha_k \rho_k) + \partial_x(\alpha_k \rho_k u_k) = 0 \\ \partial_t(\alpha_k \rho_k u_k) + \partial_x(\alpha_k \rho_k u_k^2 + \alpha_k p_k) - p_k^i \, \partial_x \alpha_k = S_k \end{cases}$$

where $\alpha_k, \rho_k, u_k, p_k, p_k^i$ and S_k stand for the volume fraction, density, velocity, pressure, interfacial pressure and the source terms which contain the wall friction and the gravity effects. In the first equation $r_p$is the relaxation parameter which regulates the interaction between phases.
Although this method has many desirable mathematical and numerical properties, it suffers from two issues. First, 5-equation system (1) experiences an excessive numerical dissipation compared to the 4-equation formulation. In the present work, the preconditioning technique is adapted to solve this issue. This matter has been addressed elsewhere [3] and is not topic of the present work. The second issue, which is inherent to all slug capturing methods, is that these methods are computation-hungry. Slug capturing methods need very fine computational grids, with cell-sizes about the pipe diameter, compared to the "unit cell" and "slug tracking" methods where cells, several orders of magnitude larger, can be used. As a result, using this method to simulate practical applications (in order of dozen kilometers), if impossible, is very limited and needs proper computational resources.
In the present work, we report our work on accelerating the slug capturing method using GPU compute device.

Numerical Method

The governing equations (1) can be written as

(2) $\partial_t U + \partial_x F(U) + H(U)\partial_x \alpha_g = R(U) + S(U)$

where $U = \left[\alpha_g, \alpha_k\rho_k, \alpha_k\rho_k u_k\right]^T$ is the vector of unknowns, $F(U) = [0, \alpha_k\rho_k u_k, \alpha_k\rho_k u_k^2 + \alpha_k p_k]^T$ is the flux vector, $H(U) = \left[u_i, 0, (-1)^k p_k^i\right]^T$ is the non-conservative part of the flux, $R(U) = \left[r_p\left(p_g - p_l\right), 0, 0\right]^T$ is the relaxation term and $S(U) = [0, 0, S_k]^T$ is the source terms. The usual Godunov method (finite volume) is used to discretize Eq. (2). For this purpose, the system (2) is divided into the homogenous part (hyperbolic step) and two other steps (relaxation and source terms) as [4]

(3) $\partial_t U = R(U),\ \partial_t U = S(U)$.

The hyperbolic part has been solved using an approximate Riemann solver. The HLLC solver of Lochon et al. [5] is adapted for this purpose. One can find the solution at the end of the hyperbolic part as

(4) $U_i^H = U_i^n - \frac{\Delta t}{\Delta x}\left(F_{i+\frac{1}{2}}^n - F_{i-\frac{1}{2}}^n\right) - \frac{\Delta t}{\Delta x}\left(H_{i+\frac{1}{2}}^n - H_{i-\frac{1}{2}}^n\right)$,

where the flux can be obtained as $F(U) = [0, G_K(U)]^T$. Here the first element of the flux vector is related to the first equation in (1). Considering that $u_k \ll c_k$ (subsonic flow) and $c_l > c_g$, where c_k stands for the sound speed of the phase k, one can simplify the original solver considerably. It is easy to show that the flux of the liquid part becomes

(5) $G_{liq} = \begin{cases} G_L^{liq} + S_L^{liq}\left(U_L^{*liq} - U_L^{liq}\right) & S_M^{liq} \geq 0 \\ G_R^{liq} + S_R^{liq}\left(U_R^{*liq} - U_R^{liq}\right) & S_M^{liq} < 0 \end{cases}$

where S_L^k, S_R^k and S_M^k are the speeds of the left going, right going and contact discontinuity waves of each phase. For the gas phase and when $S_M^{liq} < S_M^{gas}$ we have

(6) $G_{gas} = \begin{cases} G_L^{gas} + S_L^{gas}\left(U_L^{*gas} - U_L\right) & S_M^{liq} > 0 \\ G_R^{*gas} + S_K^{gas}\left(U_M^{*gas} - U_R^{*gas}\right) & S_M^{liq} < 0 < S_M^{gas} \\ G_R^{gas} + S_R^{gas}\left(U_R^{*gas} - U_R^{gas}\right) & S_M^{gas} < 0 \end{cases}$

The other case ($S_M^{liq} > S_M^{gas}$) can be obtained similarly. The third entry of the H vector (for the momentum equation) is only important and is calculated as

(7) $H_k^{mom} = \frac{1 \mp sign\left(S_M^{liq}\right)}{2}(-1)^k\left[\alpha_R^{liq} p_R^{*liq} - \alpha_L^{liq} p_L^{*liq}\right]$

Here $k = 1$ for the gas phase and $k = 2$ for the liquid phase. More details can be found in [5].

After solving one step of the hyperbolic part, one needs to apply the effect of the source terms. For this, we have

(8) $U_i^S = U_i^H + \Delta t\, S(U^n)$

Finally, the result of the source term step is used as the initial condition for the relaxation step. Here, we assume $r_p \to \infty$ which is equivalent to assuming $p_g = p_l = p$. Considering that $\alpha_k\rho_k$ remains constant during this step, one has

(9) $1 = \alpha_g + \alpha_l = \frac{\alpha_g\rho_g}{\rho_g(p)} + \frac{\alpha_l\rho_l}{\rho_l(p)}$.

Assuming proper equation of state for each phase as $p_k(\rho_k)$, one can solve Eq. (9) for the equilibrium pressure. Then, obtaining α_k^R form p is straightforward.

Finally, one has $U_i^{n+1} = \left[\alpha_g^r, (\alpha_k\rho_k)^H, (\alpha_k\rho_k u_k)^S\right]^T$. As mentioned before, the 5-equation system experiences excessive numerical diffusion compared to the 4-equation counterparts. In the present work, this issue has been solved using preconditioning technique which regulates the characteristic velocities in the liquid phase. The detailed explanation of this technique is discussed in [3].

GPU Acceleration

Graphics processing units (GPUs), with their many-core architectures and higher memory bandwidth, has become one of the most important parts of high-performance computing nowadays. Considering that the method described in the previous section has a very small (local) discrete stencil and it's time marching nature, it is possible to utilize the power of the GPUs very efficiently and accelerate simulations tremendously.

In the present work, the numerical method explained before is fully implemented on the GPU using CUDA. In this implementation, all data are stored on the GPU's memory (to minimize the data transfer between CPU and GPU) and the subsequent calculations have been done there too. When necessary to transfer data from GPU to CPU (save the results), the copy process is overlapped with kernel launches. It has been tried to use the shared memory as much as possible and also maximize coalesced access to the global memory where possible. The 1D nature of the simulations let to use efficiently achieve these.

Results and Discussion

The CPU and GPU versions of the aforementioned method have been compared extensively with the benchmarks test cases, to verify the implementations. Also, several different experiments have been simulated and the results are compared with. Few have been reported before [5]. Here we just report the results showing the strength of our numerical method and GPU speed-ups. Figure 1 shows an example of the intermittent flow in a straight pipe [6] ($36m$ long pipe, inlet with $u_{sl} = 0.41\ m/s$, $u_{sg} = 2.36\ m/s$ and outlet to the ambient pressure), where usual 4-equation solvers have problem in predicting slug flow [7]. Our method captures the intermittent nature of the flow correctly and predicts the frequency of the slug flow accurately. As evident, the results (slugs) finally reach to a statistically develop state. It is very important to notice that there is no source of randomness in this problem (fixed inlet and outlet), and such a behavior is naturally expected. However, the solvers based on the 4-equation formulations will produce several different waves (slugs) which are due to the ill-posedness of their governing equations. Because these waves are unphysical, connecting them to statistical phenomena such as slug length/frequency distribution is baseless and wrong. Another

important feature of our method is its independency to the grid resolution. While many 4-equation solvers produce good results just for a specific range of grid resolution (constant cell size over diameter), our method shows gird independency as expected from a well-posed method.

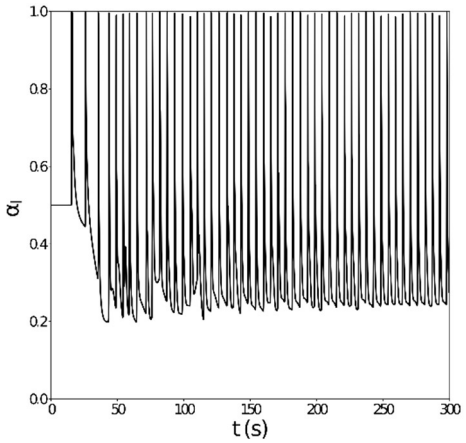

Figure 1. Time history of liquid volume fraction in exit of the straight pipe, with fixed inlet and outlets.

Figure 2 shows almost same problem as Figure 1, but for the case where fluids are oil and air. This figure compares the predicted slug frequency with the experimental data for different gas velocities.

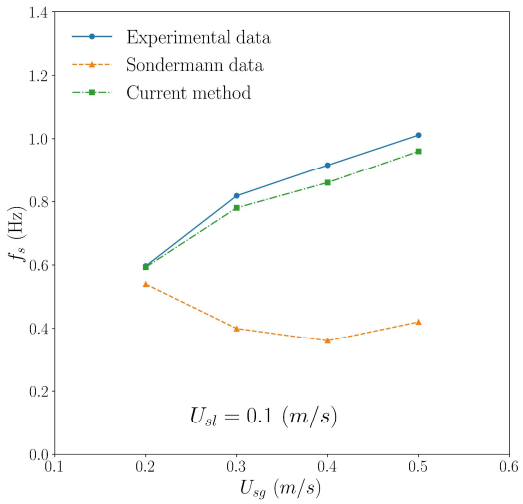

Figure 2. Predicted slug frequencies compared to the experimental data and numerical simulations form [7], in oil/air case.

As evident, our method predicts the frequencies very accurately, while the method of [7] can not even capture the trend of the experimental data.
To finalize this section, the results of the GPU code are compared with the CPU version. In this section the results of the GPU code are obtained on a NIVIDIA V100 and the results related to the CPU version of the code are obtained on system with two Intel I7 3.6GHz CPUs (10 cores are only used from 12 available cores). Table 1 compares the time needed by each version of the code to complete simulation of a pipe with specified length and number of computational cells for 1s of physical time. This table covers pipes starting from 192 m to about 12 km (which is in order of a real industrial problem). As evident, the GPU solver outperforms the CPU code by more than 130 times for many situations.

Table 1. Comparing GPU and CPU versions of the code.

Length	Cells	CPU	GPU	Speed-up
192 m	25.6k	4.0m	2.8s	83 X
384 m	51.2k	8.1m	4.7s	103 X
768 m	102.4k	16.7m	8.5s	117 X
1.54 km	204.8k	33.6m	15.5s	130 X
4.07 km	409.6k	68.3m	28.6s	143 X
6.14 km	819.2k	133m	54.7s	146 X
12.3 km	1638.4k	265m	117s	136 X

Considering that one usually needs to simulate such system for more than 500s (physical time) to see the effects, the last case on GPU needs about 16 hours to complete. Same problem on CPU takes around 92 days! This table clearly indicates that GPUs can be used to accelerate slug capturing methods tremendously. This will open the door to use these methods for real industrial applications, which was impossible before by using CPUs.

Conclusions

A new slug capturing method, based on the 5-equation always hyperbolic formulation, is developed here. The method shows very interesting features such as grid independency and converging to a statistically developed solution in problems with no source of randomness. GPUs can be used to tackle the time-consuming issue of the method. Our numerical experiments show that using GPUs, it is possible to simulate industrial size problems in matter of hours. Considering just the current hardware advancement trends (Moore's law for CPUs and Huang's law for GPUs), slug capturing methods will become one of the main industrial tools in less than 5 years. There are other areas needed to be studied further, which could benefit more the slug capturing methods. One of them is the machine learning techniques and their integration with the simulation techniques. This is the topic of our ongoing research.

Acknowledgments

The authors acknowledge the support awarded by GALP/Petrogal Brazil and ANP – agencia Nacional de Petróleo, Gás Natural e Biocombustível.

Responsibility Notice

The authors are the only responsible for the paper content.

References

[1] Bonizzi, M.; *PhD thesis. Imperial College London (University of London)*, 2003.

[2] Dinh, T.N.; Nourgaliev, R.R.; Theofanous, T.G.; *10th International Topical Meeting on Nuclear Reactors Thermal Hydraulics*. Korea, 2003.

[3] Nikfarjam, F.; Anbarlooei, H.R.; Cruz, D.O.A.; Preconditioning hyperbolic 5-equation two-phase model for low Mach number simulations, to be published.
[4] Ferrari, M.; Bonzanini, A.; Poesio, P., *Int. J. Numer. Methods Fluids*,85, 2017.
[5] Nikfarjam, F.; Anbarlooei, H.R.; Cruz, D.O.A.; in *Multiphase Flow Dynamics: A Perspective from the Brazilian Academy and Industry*,2022.
[6] Ujang, P.M.; Lawrence, C.J.; Hale, C.P.; Hewitt G.F.; *Int. J. Multiph. Flow*, 35, 2006.
[7] Sondermann, C.N.; *PhD thesis. Federal University of Rio de Janeiro*, 2021.