

Efficient Smoke Simulation on Curvilinear Grids (Supplementary Material)

Vinicius C. Azevedo[†] and Manuel M. Oliveira[‡]

Instituto de Informática – UFRGS

1 Semi-Lagrangian advection on curvilinear coordinates

To show that the velocity in the semi-Lagrangian algorithm works by substituting the transformed velocity in the back-trajectory step, we define the **covariant** basis vectors as

$$\mathbf{g}_1 = \frac{\mathbf{g}^2 \times \mathbf{g}^3}{\vartheta} \quad \mathbf{g}_2 = \frac{\mathbf{g}^3 \times \mathbf{g}^1}{\vartheta} \quad \mathbf{g}_3 = \frac{\mathbf{g}^1 \times \mathbf{g}^2}{\vartheta}, \quad (18)$$

where \mathbf{g}^i are the vectors of the contravariant basis defined in Eq. 7, and $\vartheta = \mathbf{g}^1 \cdot (\mathbf{g}^2 \times \mathbf{g}^3)$. Assuming that the velocity on the physical domain is given by $\vec{v} = ds/dt$, we can use the chain rule to express it on the computational domain in a contravariant basis as:

$$\vec{v} = \frac{\partial \xi}{\partial x} \frac{dx}{dt} + \frac{\partial \eta}{\partial y} \frac{dy}{dt} + \frac{\partial \tau}{\partial z} \frac{dz}{dt} = \mathbf{g}_1 \frac{dx}{dt} + \mathbf{g}_2 \frac{dy}{dt} + \mathbf{g}_3 \frac{dz}{dt}. \quad (19)$$

The unit covariant basis can be expressed as

$$\mathbf{g}_{(i)} = \frac{1}{h_i} \mathbf{g}_i, \quad (20)$$

and the velocity \vec{v} in this unit covariant basis as

$$\vec{v} = u^i \mathbf{g}_i = h_i u^i \mathbf{g}_{(i)}, \quad (21)$$

where h_i are defined in Eq. 9, and u^i are the vectors of the covariant basis. Thus, combining Eq. (21) and Eq. (19)

$$\mathbf{h}_1 u^1 \mathbf{g}_{(1)} + \mathbf{h}_2 u^2 \mathbf{g}_{(2)} + \mathbf{h}_3 u^3 \mathbf{g}_{(3)} = \mathbf{g}_1 \frac{dx}{dt} + \mathbf{g}_2 \frac{dy}{dt} + \mathbf{g}_3 \frac{dz}{dt}. \quad (22)$$

Using Eq. (20) to cancel the $\mathbf{g}_{(i)}$ terms gives us

$$\mathbf{h}_1 u^1 + \mathbf{h}_2 u^2 + \mathbf{h}_3 u^3 = \mathbf{h}_1 \frac{dx}{dt} + \mathbf{h}_2 \frac{dy}{dt} + \mathbf{h}_3 \frac{dz}{dt}. \quad (23)$$

We use the notion of physical vectors

$$\mathbf{u}_{(i)} = h_i u^i = \frac{1}{h_i} u_i, \quad (24)$$

to further transform Eq. (23) in

$$\vec{v} = \frac{dx}{dt} \mathbf{i} + \frac{dy}{dt} \mathbf{j} + \frac{dz}{dt} \mathbf{k} = \frac{\mathbf{u}_{(1)}}{\mathbf{h}_1} + \frac{\mathbf{u}_{(2)}}{\mathbf{h}_2} + \frac{\mathbf{u}_{(3)}}{\mathbf{h}_3}. \quad (25)$$

This equation is substituted back in Eq. 4, yielding the semi-Lagrangian method in curvilinear coordinates.

2 Multigrid on curvilinear overlapping grids

The *multigrid method* is a fast ($O(N)$) algorithm for solving linear systems [BHM00], widely used in computer graphics.

For regular grids, the matrix coefficients of the linear system are copied from the finest to the coarsest levels. On curvilinear grids, however, the coarsening process degenerates the shape of the discretized object. Thus, simply copying the values of the matrix coefficients affects the method's convergence. To improve it, we use the scheme described in [HSGN96]. In such a scheme, the areas and volumes of the coarse-grid cells are defined as the sum of the corresponding cells in the finest grid, and the coarse matrix coefficients are obtained as the area-mean of the coefficients in the finest grid. Thus, the coarse grid areas A_c and volumes V_c are defined as $A_c = \sum_{i \in \text{finest}} A_f^i$ and $V_c = \sum_{i \in \text{finest}} V_f^i$, where the A_f^i and V_f^i are the corresponding fine grid areas and volumes, respectively. The coarse grid coefficients C_c are the area means of associated fine cell coefficients C_f^i : $C_c = \frac{1}{A_c} \sum_{i \in \text{finest}} C_f^i A_f^i$, since the Poisson matrix is defined as function of the cell areas (Eq. 15).

Our multigrid implementation uses the *incomplete multigrid method* (ICMG) [HF91] with Dirichlet boundary conditions to transfer information between overlapping grids. ICMG constrains the communication between different domains to the finest level; it contrasts with the *complete multigrid method* (CCMG) [Hen05], which exchanges information among all grid levels. ICMG may have convergence problems when two curvilinear grids overlap each other. In this case, the CCMG approach is more robust, handling multiple overlapping configurations without affecting the multigrid's convergence speed. However, CCMG is harder to implement, due to the nature of the coarsening operator that may generate complex interpolation configurations, which require special treatment [Hen05]. For the purpose of demonstrating the results of our technique, we restricted

our implementation to the ICMG approach, even though our method should work equally well with CCMG.

Exchanging interpolated pressure information between overlapping domains may introduce discontinuities in the resulting pressure fields, most notably when the interpolation direction is going from the foreground to the background grid. Thus, at each iteration of the pressure solver, we use Henshaw's approach [Hen05], which consists of applying additional Gauss-Seidel iterations on neighborhoods of the background boundary cells. This process smooths out the high-frequency error modes that may appear due the interpolation process.

Since different grids on the simulation domain may converge at different rates, we dynamically adjust the number of smoothing iterations (v) for each grid based on its residual error. We compare the residual ratios between the foreground and background grids res_{bg}/res_{fg} to $\sigma^{1/v}$, since it allows larger values of v to change more easily. The residual ratio σ is defined within lower (σ_-) and upper (σ_+) boundaries. This strategy accelerates the simulation up to $5\times$. Numerical experimentation of [Hen05] defined $(\sigma_-, \sigma_+) = (\frac{1}{2}, 2)$. Algorithm 1 summarizes the steps of our multigrid solution.

Algorithm 1 Multigrid on overlapping domains

```

while  $res_{bg} + res_{fg} < res_{max}$  do
  perform a full V-Cycle on the background grid
   $res_{bg} \leftarrow \text{calculateResidual}(\text{backgroundGrid})$ 
  // Interpolate pressures from the background to the foreground
  overlapped grid
  for all cells on the boundary of the foreground grid do
     $p_{fg} \leftarrow \text{interpolatePressureField}(p_{bg}, x_{fg})$ 
  end for
  perform  $v$  full V-Cycles on the foreground grid
   $res_{fg} \leftarrow \text{calculateResidual}(\text{foregroundGrid})$ 
  // Interpolate pressures from the foreground overlapped to the
  background grid
  for all boundary cells on the background grid do
     $p_{bg} \leftarrow \text{interpolatePressureField}(p_{fg}, x_{bg})$ 
  end for
  // Remove high frequencies that may appear due interpolation
  perform smoothCells(backgroundGrid,  $v$ )
  // Adjust the number of sub-smooths on the foreground grid
  if  $res_{bg}/res_{fg} < \sigma_-^{1/v}$  do
     $v \leftarrow \min(1, v - 1)$  // decrease number of sub-smooths
  else if  $res_{bg}/res_{fg} > \sigma_+^{1/v}$  do
     $v \leftarrow \max(v + 1, num_{max})$  // increase number of sub-
    smooths
  end if
end while

```

References

[BHM00] BRIGGS W. L., HENSON V. E., MCCORMICK S. F.: *A multigrid tutorial: second edition*. Society for Industrial and

Applied Mathematics, Philadelphia, PA, USA, 2000. 1

[Hen05] HENSHAW W. D.: On multigrid for overlapping grids. *SIAM J. Sci. Comput.* 26, 5 (May 2005), 1547–1572. 1, 2

[HF91] HINATSU M., FERZIGER J. H.: Numerical computation of unsteady incompressible flow in complex geometry using a composite multigrid technique. *International Journal for Numerical Methods in Fluids* 13, 8 (1991), 971–997. 1

[HSGN96] HE P., SALCUDEAN M., GARTSHORE I., NOWAK P.: Multigrid calculation of fluid flows in complex 3d geometries using curvilinear grids. *Computers Fluids* 25, 4 (1996), 395 – 419. 1