Appendix

C-like pseudocode of the algorithm.

```
float2 cur_corner_points[M+1, N+1];
float2 next_corner_points[M+1, N+1];
float cur_density_field[M, N];
float next_density_field[M, N];
float* phi_params[M, N];

advect_corner_points()
{
      for(j=0; j < N+1; j++) {
          for(i=0; i < M+1; i++) {
               next_corner_points[i, j] = advect_rk4(cur_corner_points[i, j]);
          }
      }
}

compute_cv_mass()
{
     for(j=0; j < N; j++) {
         for(i=0; i < M; i++) {

              float acc_mass = 0;

              Polygon v(next_corner[i, j],next_corner[i+1, j],
                        next_corner[i+1, j+1],next_corner[i, j+1]);

              BoundingBox b = v.bounding_box();

              for(y = b.lower_y; y < b.upper_y; y++) {
                    for(x = b.lower_x; x < b.upper_x; x++) {

                          Polygon e(cur_corner[i, j], cur_corner[i+1, j],
                                    cur_corner[i+1, j+1], cur_corner[i, j+1]);

                          Polygon i = v.intersect(e);
                          acc_mass += riemann_sum(phi_params[x, y], i);

                    }
              }
              next_density_field[i, j] = acc_mass;
          }
      }
```

```
}

main()
{
        init_cur_corner_points();

        while(!done)
        {
           advect_corner_points();
           compute_phi_params(cur_density_field);
           compute_cv_mass();
           display_result(next_density_field);
           swap(cur_density_field, next_density_field);
        }
}
```