

# Real Time Large Scale Fluids for Games

Daniel Kallin\*

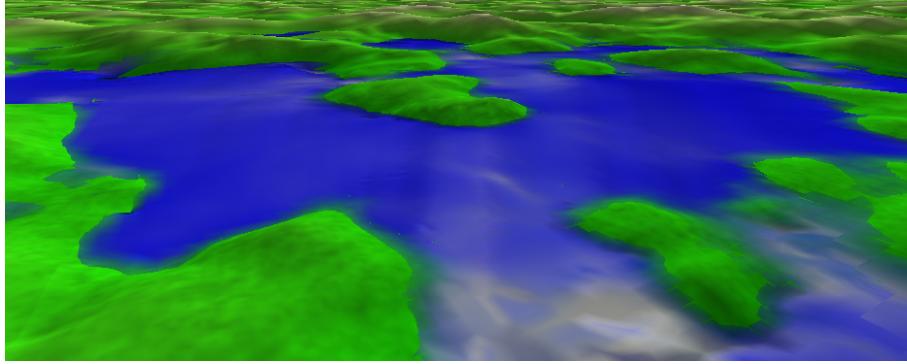


Figure 1: A virtual sea

## Abstract

This paper presents an implementation of a stable height-field fluid solver on non-uniform quadtree grids. Smoothing kernel interpolation allow one to use semi-Lagrangian advection on the non-uniform grid. A modification to the advection make it mass conserving. The non-uniform grid allow the model to run on a very tight cell budget with high frame rates even for arbitrary sized environments. Gravity acceleration is implemented as a modified explicit Euler step with upwind differencing. The differencing is based on a definition of a node neighbour which doesn't require the neighbor to be a quadtree leaf. This solves the problems of hanging nodes inherent with quadtrees. This model is suitable for water propagating over height field terrains in interactive environments like video games.

**Keywords:** fluid dynamics, semi-Lagrangian advection, upwind differencing, adaptive grid, quadtree

## 1 Introduction

This paper describes the research into faster than real time fluid simulations done as a master thesis project at Avalanche Studios, an independent video game developer in Stockholm, Sweden. The premise for the research was the special set of demands that contemporary video games have. It is the desire to improve the interactivity of the game environment that pushes the technological boundaries. At the same time the physical accuracy of the simulations are of lower priority than the interactivity and the plausibility of the en-

vironment as perceived by a casual player. The speed requirements are very strict while the demands on the algorithm are very high with complex boundary conditions consisting of both fixed boundaries at the terrain-fluid interface and dynamic boundaries at the air-fluid interface.

## 2 Previous work

The field of computational fluid dynamics has been extensively researched, primarily for engineering purposes but also extensively by the graphics and animation community for visualization purposes. While it is beyond the scope of this report to present an exhaustive breakdown of the field some key publications which has inspired this research will be briefly presented.

### 2.1 Navier Stokes integration

On a full 3D grid the Navier Stokes equations can be discretized and solved numerically, an approach which can capture all the phenomena of real world fluids of longer wavelength than the grid cell size.

Jos Stam [Stam 2003] show an implementation of the Navier Stokes equations that is unconditionally stable with time steps of arbitrary lengths. It uses implicit diffusion and a combination of particle and grid based implicit advection.

### 2.2 Height field simulation

When not using uniform isotropic grids one can reduce the complexity of the model. Shallow lakes and puddles are examples of situations where a uniform isotropic grid could be replaced by a height field model where there is no grid subdivision along one axis.

Kass and Miller [Kass and Miller 1990] describe an implementation of a height field fluid based on a simplified set of the Shallow Water equations. Their integrator use alternating direction differencing with implicit integration. Their simplified model does not capture the advection of velocity through the fluid domain. It can handle dynamic boundaries of the fluid volume which means that their fluid can move across dry terrain and recede from flooded areas.

---

\*e-mail: daniel.kallin@gmail.com

Maes, Fujimoto and Chiba [Maes et al. 2006] describe a height field model where the columns are subdivided to capture vertical inhomogeneities of the fluid. Their simulations however only show minor quality gains from this increase in resolution.

### 2.3 Particle based simulation

Smoothed Particle Hydrodynamics (SPH) has since its introduction to the graphics community attracted attention and Hegeman, Carr and Miller [Hegeman et al. 2006] show how the simulation can be processed on the GPU. Even if SPH models have cubic complexity when filling volumes the general problem is not the simulation but rather the subsequent visualization of the data [Miller-Fischer 2007].

Müller, Schirm and Duthaler [Müller et al. 2007] show how the visualization of particle fluids can be done in screen space for increased performance.

### 2.4 Hybrid approaches

Irving, Guendelman, Losasso and Fedkiw [Irving et al. 2006] describe a combination of a height field and a 3D grid to very accurately model the free surface of water with arbitrary topology while maintaining fast simulation speeds. Their implementation is as such adaptive in the up direction but not adaptive along any other axes. This makes this approach slow for fluid volumes that cover large areas but the visual quality of their simulation is very good.

Cords [Cords 2007] describe an application of mode-splitting where the volume filling flow is simulated using smoothed particle hydrodynamics and the high frequency surface waves are simulated as a 2D wave equation superimposed on the surface generated from the SPH simulation. While this approach yields extremely attractive water surfaces the wave equation must still be solved for a high resolution grid.

## 3 Theory

### 3.1 Navier-Stokes equations

The Navier-Stokes equations governing the dynamics of incompressible fluids can be stated as [Hoffman and Johnson 2006]:

$$\frac{\partial \mathbf{u}}{\partial t} = -(\mathbf{u} \cdot \nabla) \mathbf{u} + \nu \nabla^2 \mathbf{u} + \mathbf{f}$$

$$\nabla \cdot \mathbf{u} = 0$$

These equations describe the behavior of the velocity field  $\mathbf{u}$  with the acceleration  $\frac{\partial \mathbf{u}}{\partial t}$  as a sum of three terms. The first term  $-(\mathbf{u} \cdot \nabla) \mathbf{u}$  describes advection: the inertial transport of velocity with the fluid flow. The second term  $\nu \nabla^2 \mathbf{u}$  models the viscosity of the fluid as diffusion of the velocity field.  $\mathbf{f}$  is the sum of all applied forces such as gravity or forced flows. The second equation  $\nabla \cdot \mathbf{u} = 0$  represents the requirement of the fluid to be incompressible.

### 3.2 Shallow Water Equations

The Shallow Water Equations are derived from the Navier-Stokes equations with certain assumptions and approximations made: they apply to thin, non turbulent layers of fluid which acts as a height field [Wong 2003]. These equations have been used extensively in graphics applications [Kass and Miller 1990; Thurey et al. 2007] due to their simplicity and speed - the assumption of 2D reduces the worst-case complexity of the simulation to  $\mathcal{O}(n^2)$  instead of  $\mathcal{O}(n^3)$  which applies to the full Navier Stokes 3D-grid simulations.

As stated in Bridson and Müllers SIGGRAPH course on fluid simulation [Bridson and Müller-Fischer 2006] the numerical dissipation is similar to but dominant over the real world viscous dissipation for most fluids which means that the viscosity term often can be ignored. However solvers that attempt to operate at interactive frame rates must take large time steps  $\Delta t$  which negatively affects the stability of the model. Imposing explicit non-physical dissipation in these cases would increases stability.

Below the modified shallow water equations with diffusion and viscosity implemented is presented:

$$\frac{\partial h}{\partial t} = -\nabla \mathbf{u} d + \kappa \nabla^2 h$$

$$\frac{\partial \mathbf{u}}{\partial t} = -g \nabla h - (\mathbf{u} \cdot \nabla) \mathbf{u} + \nu \nabla^2 \mathbf{u}$$

These equations model the change in height  $\frac{\partial h}{\partial t}$  as a function of the velocity  $\mathbf{u}$ , the depth  $d$  (defined as the difference between the water height  $h$  and the height of the sea floor) and a diffusion term  $\kappa \nabla^2 h$ . The acceleration of the velocity field is modelled as a sum of the acceleration due to gravity  $-g \nabla h$ , an advection term and a viscosity term  $\nu \nabla^2 \mathbf{u}$ .

### 3.3 Eulerian and Lagrangian models

Eulerian models use discretized grids that remain fixed in the frame of reference. The individual cells communicate with each other and phenomena are propagated by transmissions across cell boundaries. Discrete versions of the Navier-Stokes or the Shallow Water equations are integrated using a solver of one's choice. These models are beneficial since they put the designer in control over the resolution. Their regularity also allow for easy visualization and quick isosurface extraction when simulating the fluid as a 2.5D height field [Kass and Miller 1990]. Eulerian models are rather poor at modelling advection and energy and mass conservation. Special care must be taken to guarantee constant volume in the fluid either in the solver itself or as a post processing step. Even worse is that an inappropriate choice of integrator for the advection will lead to severe instabilities when the time step is lengthened, something which today is required when attempting to attain real time speeds.

Lagrangian particle models model interactions in a moving frame of reference. The fluid is represented by a multitude of particles which interact with each other and their surroundings. This model is by its nature mass conserving since every quantum of fluid is represented by a particle and is thus accounted for by the model. As a drawback the model can invest a large amount of memory and computing power in volumes that are filled but might be uninteresting to an observer. The visualization of the fluid is also problematic since the generation of an isosurface is more expensive when dealing with the randomly distributed particles [Green 2008]. Hugo-son and Nilsson describes an approach of surface normal blending which eschews the generation of a geometric isosurface altogether [Hugoson and Nilsson 2007].

### 3.4 Explicit integrations

As noted by Kass and Miller [Kass and Miller 1990] explicit integrators such as Euler's method are a poor choice when solving the Shallow Water equations. When the wave velocity approaches one grid cell per time step the simulation tend to diverge. The time step  $\Delta t$  and gravity  $g$  must be strictly limited to avoid oscillations and divergent behavior [Bridson et al. ].

To maintain stable simulations implicit solvers are desired. The Implicit Euler method attempt to find a state which when run backwards in time using the explicit method yield the initial state. As noted by Stam [Stam 2003] this approach is problematic since the fluid is advected by the velocity field which in turn is self-adverted. A possible approach is to use different solvers for different terms in the equations. Stam describes how one can integrate the advection using a Lagrangian method and the diffusion using an implicit Eulerian method with Gauss Seidel relaxation.

### 3.5 Semi-Lagrangian advection

A hybrid approach which has been embraced by the interactive computational fluid dynamics community is the semi-Lagrangian model [Stam 1999]. It represents the fluid using a grid and models the dynamics using particles. All attributes such as velocity, density and pressure are bound to the grid but their advection by the velocity field is not handled by cell to cell communications. Instead, mass-less particles follows the velocity field and transports the attributes from their departure points to their destinations after a single time step  $\Delta t$ . This tracing can be performed forward and backwards in time, yielding the forward and backward semi-Lagrangian model.

In the backward semi-Lagrangian model the particles' destinations are set as the centers of the grid cells and their departure points are calculated from the velocity field. The particles carry with them the attributes from their departure points which does not necessarily line up with the grid. At this point one must therefore extract an interpolated value from the grid [Bridson et al. ].

## 4 Techniques

The model used in the prototype software is a height field fluid simulation on a 2.5D terrain. Both the fluid and the terrain data reside in the same quad tree data structure where every node in the tree contains a complete set of data. Non-leaf nodes have values that are averages of its four child nodes' values. The fluid is simulated using a modified set of the Shallow Water Equations where the advection is handled using semi-Lagrangian advection and the influence of gravity, friction and diffusion are modelled using slightly modified explicit Euler integrators.

### 4.1 Gravity acceleration

The Shallow Water Equations (see Section 3.2) model the acceleration due to gravity as  $\frac{\partial \mathbf{u}}{\partial t} = -g \nabla h$ . The prototype implementation the gravity acceleration is explicitly modelled. This led to severe artifacts such as introduced oscillations caused by the inherent properties of explicit Euler integrators. While a solution to this problem would be to use implicit Euler integration for the acceleration this approach was rejected as it would result in an iterating implementation with all of the associated complexity. An alternative would be to keep the explicit Euler method but instead reduce the time step exclusively for the gravity acceleration for thin sheet of fluid where the overshoot was dominant over the expected behaviour of the fluid. The acceleration due to gravity is thus modelled as:

$$\frac{\partial \mathbf{u}}{\partial t} = -\frac{g \nabla h}{\max(1, d)}$$

In the prototype it is observed that without this modification the model exhibit spontaneous oscillations when the cell size is smaller than the depth.

### 4.2 Upwind differencing

An important part of the model is the upwind differencing which is used in the derivative approximation in the discretized grid [Fitzpatrick ] [Rogers and Kwak 1991]. Early prototypes clearly showed that central differences resulted in high frequency noise and undamped oscillations.

The upwind differencing scheme defines the gradient of the height field  $h$  as a finite difference whose direction is dependant on the velocity vector  $\mathbf{u} = (u, v)$ .

$$\nabla h = \left( \frac{\partial h}{\partial x}, \frac{\partial h}{\partial y} \right)$$

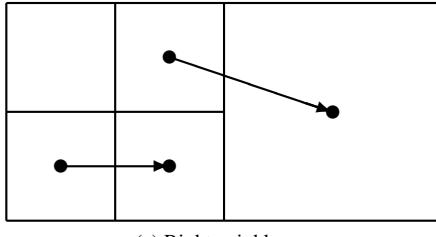
$$\nabla h \approx \left( \frac{\Delta^x h_{ij}}{\Delta x}, \frac{\Delta^y h_{ij}}{\Delta y} \right)$$

$$\begin{aligned} \Delta^x h_{ij} &\equiv \begin{cases} h_{i,j} - h_{i-1,j} & u \geq 1 \\ h_{i+1,j} - h_{i,j} & u < 1 \end{cases} \\ \Delta^y h_{ij} &\equiv \begin{cases} h_{i,j} - h_{i,j-1} & v \geq 1 \\ h_{i,j+1} - h_{i,j} & v < 1 \end{cases} \end{aligned}$$

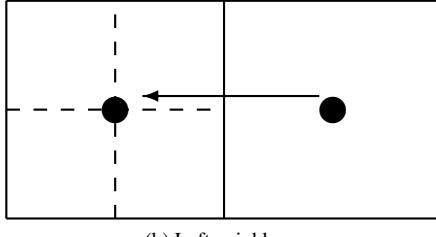
### 4.3 Adaptive grids

In contemporary video games dynamic level of detail adjustment is employed to optimize the complexity of game elements such as geometry to ensure high frame rates [Luebke et al. 2003]. As opposed to individual models which can be up or downsampled individually a flooded terrain is a very large and potentially continuous piece of geometry which would require different levels of detail in different areas [Duchaineau et al. 1997]. Computational fluid dynamics is demanding and for this reason it is desired to use different resolutions in different areas of the grid [Mller-Fischer 2007]. The required level of detail is dependent on the static terrain through which the fluid flows, the actual distribution of fluid and its velocity and the position of the observer. Apart from the terrain all of these are dynamic factors which would require us to dynamically change the resolution of the grid. It is beneficial to base the grid on a quadtree, since they allow for arbitrary subdivision of cells without affecting the surrounding cells thus simplifying dynamic resolution changes. The grid cells are also all square and symmetric further simplifying the discretization of the underlying equations. A quadtree can potentially possess hanging nodes, something which more complex subdivision techniques such as ROAM [Duchaineau et al. 1997] lack. This means that situations can arise where a single edge of a cell can touch several smaller neighboring cells. When discretizing the equations ambiguities arise in how one should formulate the differences approximating derivatives for a large cell that have more than one neighbor on each side. In the explicit gravity acceleration integrator this is solved by forcing cells to never deal with neighbors that are smaller than themselves. Instead they consider the quadtree node that shares the same level as itself to be its neighbor. This non-leaf cell's mass and velocity are calculated as the recursive average of its children's attributes. Smaller cells do not have this ambiguity since they have a single - albeit larger - neighbor.

As a precomputing step each node in the quadtree (on all levels) is assigned four neighbors at their own or a higher level (see figure



(a) Right neighbors



(b) Left neighbor

Figure 2: The neighbors of the quadtree nodes are defined as being at least as large as the node itself. (a): Two cells with arrows pointing to their right-hand neighbors. (b): One cell with arrow pointing to its left-hand neighbor which is a parent of four smaller nodes.

2). These neighbor pointers are used when computing differences for the discretized fluid equations. The actual simulation is only conducted in the leaf nodes but after all leafs have been traversed and simulated, averages are computed from the bottom up to fill every node in the tree with updated values. These averaged values are then returned when a neighbor which isn't a leaf node is polled for its properties (like velocity or fluid height).

Each cell also contains flags that decide if that particular node and its children require updates. All nodes are in the dry initial state set to not require updates and when fluid is added the flags of the filled cells and their neighbors are recursively from the bottom up set to require updates.

In the early prototypes it was observed that explicit methods with parameters that give stable simulations on an uniform grid can produce instabilities in heterogeneous grids in the boundaries between high and low resolution. This hints at a demand for implicit methods that are stable.

#### 4.4 Smoothing kernel interpolation

When implementing semi-Lagrangian advection neighboring cells do not affect each other by virtue of them being neighbors. Instead advection is handled by the transport of mass and velocity properties from the departure point to the arrival point of a massless tracer particle. At the departure point an interpolated value must be extracted from the grid. For uniform grids this can be implemented as a simple bilinear interpolation with cell values defined at a single point somewhere on the cell (usually in the center) [Stam 2003]. On a non-uniform grid however it is not obvious how one extracts an interpolated value from an arbitrary point in an adaptive grid when one is near the interface between high and low resolution (see Fig 3).

The chosen approach was to define the cell properties to be constant over the entire area covered by the cell. This yields a completely uninterpolated grid with possible discontinuities at every cell boundary, which is unacceptable. But on this piecewise constant field values can be interpolated using a smoothing kernel that returns a

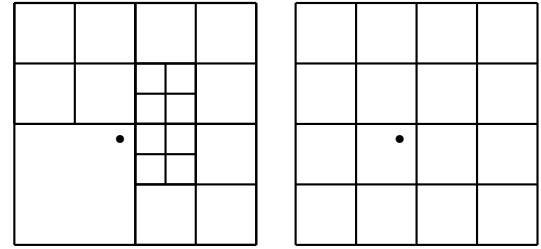


Figure 3: Finding the interpolated value at the dot is non trivial in a quad tree as opposed to a regular grid where bilinear interpolation is sufficient.

weighted integral  $\hat{f}$  of the surrounding cells' values. This approach allows to both read and write values to an adaptive grid at arbitrary positions with correct averaging of values as defined by the kernel function  $K$ .

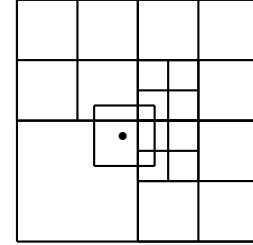


Figure 4: The value at the dot is extracted by using the cells relative coverage of the window as weights in a weighted interpolation of their values.

A uniform square kernel with constant weight within its coverage defined by the bandwidth  $h$  can be defined as:

$$\hat{f}(x, y) = \frac{1}{h^2} \int_{\mathbb{R}^2} K\left(\frac{x - x'}{h}, \frac{y - y'}{h}\right) f(x', y') dx' dy'$$

$$K(x, y) = \begin{cases} \frac{1}{4} & |x|, |y| < 1 \\ 0 & \text{otherwise} \end{cases}$$

A central issue when using smoothing kernels is to find an optimal value for the bandwidth  $h$ . In quad tree based adaptive grids with arbitrary large difference between cell sizes any constant choice of bandwidth could result in over or under smoothing at some points on the grid. The solution — and a crucial part of the implementation of the semi-Lagrangian method — is to let the bandwidth  $h$  of the kernel function depend on the size of the cell which the tracer particle arrives at. In the semi Lagrangian approach each tracer particle belongs to an individual cell with the size  $s$ . The kernel function used by the particle at its departure point is defined with a bandwidth  $h = \frac{s}{2}$ . With uniform kernel functions this corresponds to a square window with sides of length  $s$ . The extent to which this window covers the underlying cells is used as a weight when interpolating the properties of all the underlying cells (see Fig. 4). For uniform grids this approach is equivalent to bilinear interpolation, which is a requirement since the algorithm must conform with existing implementations when fed with a uniform grid.

#### 4.5 Mass conserving advection

When modelling a three dimensional incompressible fluid as a height field the model is equivalent to a compressible two dimen-

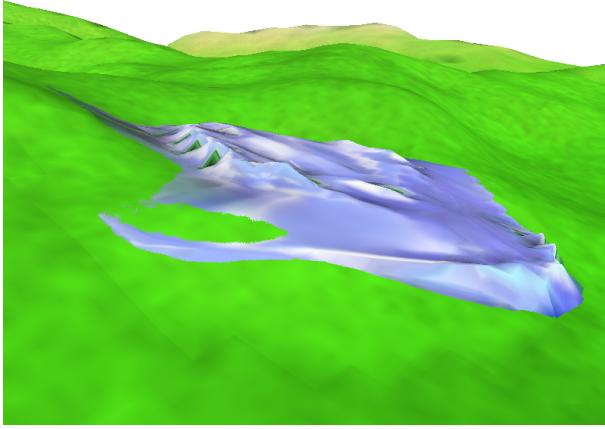


Figure 5: A mass of water flowing down a slope.

sional fluid. Compressibility is required for simulating waves in the model but it causes the velocity fields to no longer be divergence free. Therefore Stam’s method to decompose the velocity field and forcing it to be divergence free can not be applied.

The solution was to modify the semi-Lagrangian advection model. Originally it only passively reads values at the tracer particle’s departure point and replaces the values at the arrival point. The idea is that the values of the cells surrounding the departure point will attain new values from their tracer particles departure points. This leads to non-physical behavior in a compressible fluid. Instead the model actually subtracts fluid volume from the departure point and add this volume to the arrival point. This subtraction relies on kernel approach for quad tree interpolation as the amount of fluid to be subtracted from each of the surrounding cells is weighted by the kernel function.

#### 4.6 Diffusion

Introducing diffusion into a model is a simple and effective way to increase the stability of the simulation. It is employed in the implementation to dampen high frequency oscillations which arise with long time steps. Issues with volume conservation is encountered when applying diffusion to the fluid height property. Because of this only the velocities are diffused and not the fluid heights. This is motivated by the implicit advection and the upwind differencing scheme which introduce significant damping to the model. A central feature of the model is that the velocities are not only diffused inside the fluid domain but across the entire grid. No diffusion is added to the fluid mass which is only indirectly affected by the velocity diffusion and therefore keeps within its bounds. The velocity however is allowed to diffuse beyond the bounds of the fluid mass and is treated as a virtual velocity present in dry terrain. This virtual velocity is used in the advection step and is essentially what allows the implicit advection routine to propagate fluid out onto dry terrain. Below the definition of the viscous contribution to the velocity derivative and the discretized explicit interpretation is presented:

$$\frac{\partial \mathbf{u}}{\partial t}_{diff} = \nu \nabla^2 \mathbf{u}$$

$$\mathbf{u}_{n+1} = \Delta t \nu \nabla^2 \mathbf{u}_n$$

#### 4.7 Suppression of thin flows

The flow model presented so far contains no mechanism for simulating the friction between fluid and ground. For large flows where

most of the water glides on the bottom-most layer the friction is negligible but for thin and fast flows the friction becomes dominant since fluid friction scales quadratically with the flow velocity.

$$\mathbf{F} \propto \rho \mathbf{u}^2$$

The modelling of this phenomenon is further complicated by the effects of absorption when small volumes of water would rather penetrate the ground than flow on top of it. Several different approximations of friction with the initial approach incorporating the absorption effect was tested.

One approach was to specify a minimum depth  $d_{min}$  and for cells with shallower depths than this threshold forcing an artificial decay of the fluid velocity. Small amounts of fluid would then be locked to terrain instead of flowing over it freely approximating the absorption. In the subsequent visualization the drawing of fluid layers shallower than  $d_{min}$  was omitted. This approach gave visually satisfactory results but produced artifacts in the behavior of wavefront propagating over dry terrain.

Instead a proper fluid velocity decay was derived from the equations of fluid friction. The frictional force per unit area is  $F = -ku^2$  in the direction of flow  $\hat{\mathbf{u}}$ . In the height field model all properties including forces are treated as being constant along the entire column heights and thus the mass this force acts on is proportional to the depth. The resulting frictional acceleration is  $\mathbf{a} = -\frac{ku^2}{d}\hat{\mathbf{u}}$ . With forward Euler integration one obtains  $\mathbf{u}_{n+1} = \mathbf{u}_n + \Delta t \mathbf{a}$ . An observation is that long time steps can result in frictional forces actually reversing the flow, a behavior which is clearly unphysical and due to the shortcomings of the forward Euler method. Similarly to the treatment of the gravitational acceleration the frictional force is clipped at the value where it passes zero. With this modification and by transforming the frictional addition to a multiplicative factor one yield:

$$\mathbf{u}_{n+1} = \mathbf{u}_n \max \left( 0, 1 - \Delta t \frac{kv}{d} \right)$$

## 5 Results

Since the method uses a non-uniform grid that is only traversed where it contains fluid it is meaningless to talk about performance as related to grid size. Benchmarking shows that there is no practical performance hit of having inactive (dry) areas in the grid making it feasible to let the grid cover the entire virtual environment in anticipation of added fluid. After the fluid has passed over an area it would be possible to deactivate the cells and even dynamically reshape the quadtree when the observer moves across the terrain. Because of this the performance is presented as the time cost per active cell. The approximate time per iteration per active cell for a 2 GHz Athlon64 test computer were 1.7 microseconds. It should be noted that the implementation could be further optimized and that the model as such allow for computational grids which contain very few grid cells.

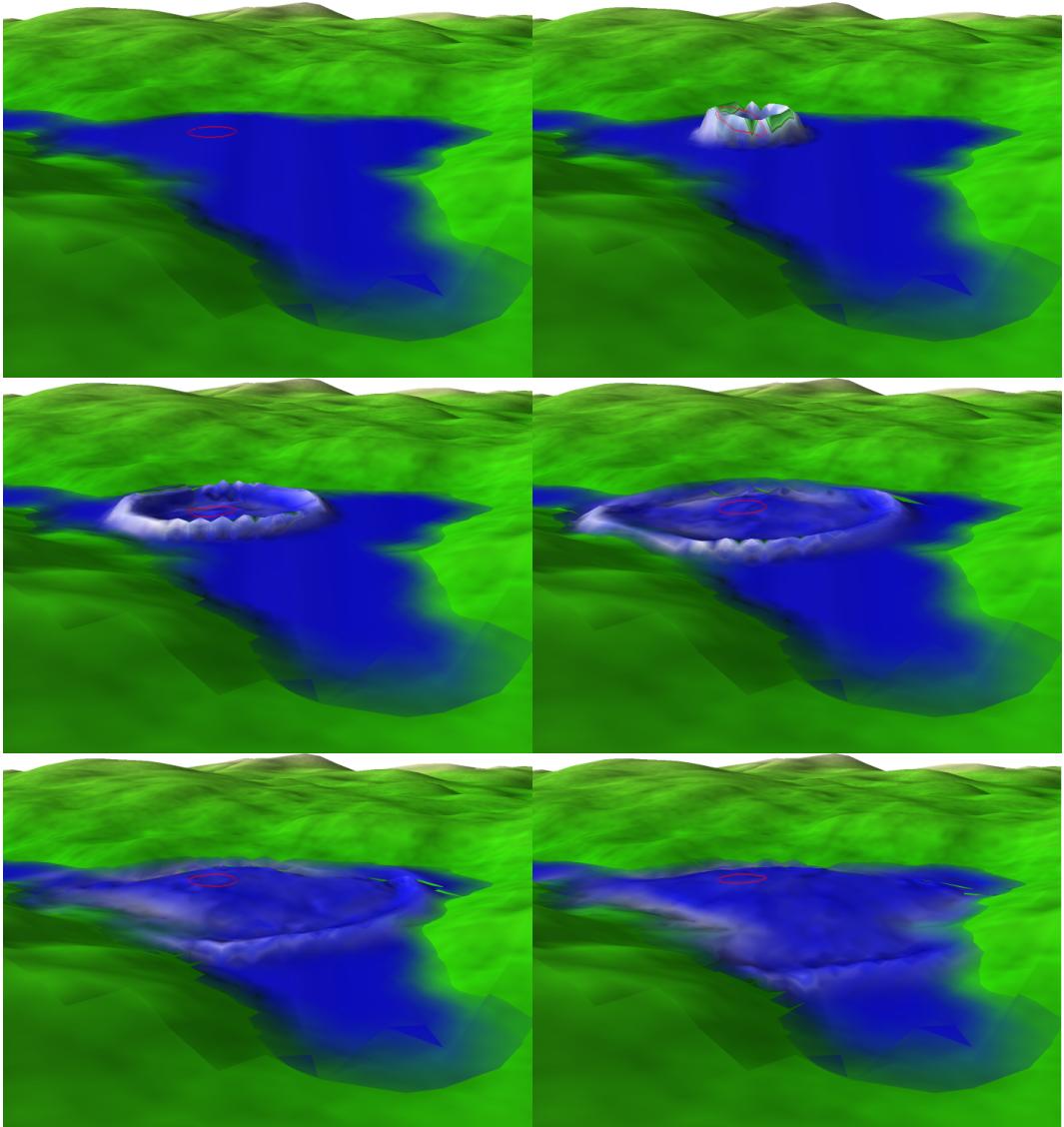


Figure 6: A wave spreading in the prototype software. Wave is spawned at the position marked by the red circle.

## 6 Discussion

### 6.1 Design choices in the model

While the modified shallow water equations can be integrated in a unified explicit step the approach described in this report have divided the fluid model into tree separate routines: dissipation, gravity acceleration and advection. These routines are performed in a simulation pass that calculates a new state from the old one. This is followed by a separate update pass that sets the current state as the new state.

#### 6.1.1 Explicit dissipation

The dissipation is explicitly modelled. While this leads to instabilities with long time steps and heavy dissipation the explicit dissipation can be considered a sensible choice since the target fluid - water - has a low natural viscosity. Therefore the parameters are still small enough to stay within the bounds of stability and if one wishes to model a more viscous fluid there are other approaches that are more suitable models than the one presented here.

Problems with mass conservation arose along fluid boundaries when diffusing the fluid mass itself and therefore the model only applies diffusion to the velocity field. Greater stability might be attained by diffusing the height field as well with the possibility of using a smaller diffusion parameter. The reader is invited to examine this problem of efficient volume conserving diffusion.

#### 6.1.2 Explicit gravity acceleration

The gravity acceleration step is the weakest point of the model. The acceleration is clipped on large water depths (see section 4). At larger time steps it promptly introduces instabilities in small cells, as expected from an explicit routine. The reader is advised to substitute this approach for a more stable implicit routine if larger time steps is a requirement.

#### 6.1.3 Advection

The model's advection routine is a mass conserving semi Lagrangian method with smoothing kernel interpolation on non-uniform grids. The fact that each advection operation in itself is mass conserving with subtraction at one point and addition at another makes it possible to skip the expensive conservation passes employed by Stam [Stam 2003] and others. The smoothing kernel approach is essential for implementing semi Lagrangian advection on adaptive grids. The downside is that there is a lot of overhead associated with the advection routine. Especially the interpolation step is expensive in the prototype. Twice per cell and iteration the quadtree is traversed from the root to the leaves where every cell that is covered by the kernel function is modified. It should be feasible to use the presented approach without starting the search from the root. It is theoretically possible but in practice rare that the departure point of the tracer particle lies far away from the arrival point. It might be worthwhile to research optimized advection routines that still uses kernel interpolation.

#### 6.1.4 Other applications

The core idea of this paper is the use of a semi-Lagrangian approach on an adaptive grids using a kernel function to read and write values to and from the grid. This approach should be applicable to other problems than just the Shallow Water equations. On a three dimensional adaptive grid we can use the same approach to solve

the Navier Stokes equations. Other problems than fluid dynamics might also be suitable for simulation using a semi-Lagrangian model with non uniform grid.

## 7 Conclusion

The method presented in this report have certain shortcomings but it presents valuable solutions which should be readily applicable to other methods. It is the author's conclusion that real time fluid dynamics in computer games is possible. But it is not unconditionally possible. It is associated with a significant computational cost and — more importantly — a significant investment in research.

The research is not necessarily about physics or computation. An important question is what we humans associate with realistic fluid behaviour. With such research we might be able to develop routines to simulate these perceived fluid behaviours.

## References

- ASPLUND, P. 2007. *Dynamic Spreading of Wildfire*. Master's thesis, Kungliga Tekniska Hgskolan.
- AXELSSON, T. 2007. *Realtidssimulering av vtskor i datorspel mha partikelsystem*. Master's thesis, Hgskolan Skvde.
- BEAUDOIN, P., PAQUET, S., AND POULIN, P. 2001. Realistic and controllable fire simulation. In *Graphics Interface 2001*, 159–166.
- BRIDSON, R., AND MLLER-FISCHER, M., 2006. Fluid simulation course notes, siggraph.
- BRIDSON, R., AND MLLER-FISCHER, M. 2007. Combating dissipation. SIGGRAPH.
- BRIDSON, R., AND MLLER-FISCHER, M. 2007. Fluid simulation. SIGGRAPH.
- BRIDSON, R., AND MLLER-FISCHER, M. 2007. More accurate pressure solves. SIGGRAPH.
- BRIDSON, R., FEDKIW, R., AND MLLER-FISCHER, M. Fluid simulation course notes, siggraph 2007.
- CORDS, H. 2007. Mode-splitting for highly detailed, interactive liquid simulation. In *GRAPHITE '07: Proceedings of the 5th international conference on Computer graphics and interactive techniques in Australia and Southeast Asia*, ACM, New York, NY, USA, 265–272.
- DUCHAINEAU, M. A., WOLINSKY, M., SIGETI, D. E., MILLER, M. C., ALDRICH, C., AND MINEEV-WEINSTEIN, M. B. 1997. ROAMing terrain: real-time optimally adapting meshes. In *IEEE Visualization*, 81–88.
- DUNN, A., AND MILNE, G. 2004. Modelling wildfire dynamics via interacting automata. In *Cellular Automata*, Springer Berlin, vol. 3305, 395–404.
- ELCOTT, S. 2005. *Discrete, Circulation-Preserving, and Stable Simplicial Fluids*. Master's thesis, California Institute of Technology.
- FITZPATRICK, R. Upwind differencing. [Online; accessed 9-June-2008].
- FOSTER, N., AND METAXAS, D. 1996. Realistic animation of liquids. *Graphical models and image processing: GMIP* 58, 5, 471–483.
- GREEN, S., 2008. Particle-based fluid simulation, nvidia.

- HEGEMAN, K., CARR, N. A., AND MILLER, G. S. 2006. Particle-based fluid simulation on the gpu. In *Computational Science – ICCS 2006*, Springer, V. N. Alexandrov, G. D. van Albada, P. M. Sloot, and J. Dongarra, Eds., vol. 3994 of *LNCS*, 228–235.
- HINSINGER, D., NEYRET, F., AND CANI, M.-P. 2002. Interactive animation of ocean waves. In *ACM-SIGGRAPH/EG Symposium on Computer Animation (SCA)*.
- HOFFMAN, J., AND JOHNSON, C. 2006. *Computational Turbulent Incompressible Flow*. Springer.
- HUGOSON, P., AND NILSSON, A. 2007. *Hybrid Fluid Simulation*. Master's thesis, Hgskolan Kalmar.
- IRVING, G., GUENDELMAN, E., LOSASSO, F., AND FEDKIW, R. 2006. Efficient simulation of large bodies of water by coupling two and three dimensional techniques. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Papers*, ACM, New York, NY, USA, 805–811.
- KARAFYLLODIS, I., AND THANAILAKIS, A. 1997. A model for predicting forest fire spreading using cellular automata. *Ecological Modelling* 99, 1, 87–97.
- KASS, M., AND MILLER, G. 1990. Rapid, stable fluid dynamics for computer graphics. In *SIGGRAPH '90: Proceedings of the 17th annual conference on Computer graphics and interactive techniques*, ACM, New York, NY, USA, 49–57.
- KPPEN, T., AND NIVFORS, A. 2005. *Realtidsrendering av stora vattenytter*. Master's thesis, Hgskolan Kalmar.
- LAYTON, A. T., AND VAN DE PANNE, M. 2002. A numerically efficient and stable algorithm for animating water waves. In *The Visual Computer*, Springer-Verlag, 41–53.
- LEE, H., KIM, L., MEYER, M., AND DESBRUN, M. 2001. Meshes on fire. In *Proceedings of the Eurographic workshop on Computer animation and simulation*, Springer-Verlag New York, Inc., New York, NY, USA, 75–84.
- LIU, P.-S., AND CHOU, Y.-H. 1997. A grid automation of wildfire growth simulation.
- LOMAX, H., THOMAS H. PULLIAM, AND ZINGG, D. W. 2001. *Fundamentals of Computational Fluid Dynamics*. Springer-Verlag.
- LUEBKE, D., REDDY, M., COHEN, J. D., VARSHNEY, A., WATSON, B., AND HUEBNER, R. 2003. *Level of Detail for 3D Graphics*. Morgan Kaufmann.
- MAES, M. M., FUJIMOTO, T., AND CHIBA, N. 2006. Efficient animation of water flow on irregular terrains. In *GRAPHITE '06: Proceedings of the 4th international conference on Computer graphics and interactive techniques in Australasia and Southeast Asia*, ACM, New York, NY, USA, 107–115.
- MITTRING, M. 2007. Finding next gen: Cryengine 2. In *SIGGRAPH '07: ACM SIGGRAPH 2007 courses*, ACM, New York, NY, USA, 97–121.
- MILLER-FISCHER, M. 2007. Real time fluids in games. SIGGRAPH.
- MÜLLER, M., SCHIRM, S., AND DUTHALER, S. 2007. Screen space meshes. In *SCA '07: Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation*, Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 9–15.
- NOE, K. O., 2004. Implementing rapid, stable fluid dynamics on the gpu.
- O'BRIEN, J. F., AND HODGINS, J. K. 1995. Dynamic simulation of splashing fluids. In *Computer Animation '95*, 198–205.
- ROGERS, S. E., AND KWAK, D. 1991. An upwind differencing scheme for the incompressible navier-stokes equations. *Appl. Numer. Math.* 8, 1, 43–64.
- STAM, J. 1999. Stable fluids. In *Siggraph 1999, Computer Graphics Proceedings*, Addison Wesley Longman, Los Angeles, A. Rockwood, 121–128.
- STAM, J. 2001. A simple fluid solver based on the fft. *J. Graph. Tools* 6, 2, 43–52.
- STAM, J., 2003. Real-time fluid dynamics for games.
- TESSENDORF, J. 1999. Simulating ocean water. In *SIGGRAPH Course Notes*, SIGGRAPH.
- THUREY, N., MÜLLER-FISCHER, M., SCHIRM, S., AND GROSS, M. 2007. Real-time breakingwaves for shallow water simulations. In *PG '07: Proceedings of the 15th Pacific Conference on Computer Graphics and Applications*, IEEE Computer Society, Washington, DC, USA, 39–46.
- WEISSTEIN, E. W. Distribution function. accessed 12-March-2008.
- WEISSTEIN, E. W. Probability function. accessed 12-March-2008.
- WIKLUND, O. 2007. *A Computational Study of a Finite Element Method for Two Phase Flow*. Master's thesis, Kungliga Tekniska Hgskolan.
- WONG, A. C. 2003. *The Moving Contact Line in a Shallow Water Simulation*. Master's thesis, University of British Columbia.
- W.W., H., R.H., G., M.G., T., W.H., R., AND D.G., D. 2000. Simulating fire patterns in heterogeneous landscapes. *Ecological Modelling* 135, 2-3, 243–263.