

## 물 표면 시뮬레이션을 위한 보존적 USCIP법

전세종                      송오영\*

세종대학교 소프트웨어학과

ypland@gmail.com, oysong@sejong.ac.kr

### A Conservative USCIP Simulation Method for Shallow Water

Sejong Jeon                      Oh-young Song\*

Department of Software, Sejong University

#### 요약

이 논문은 물 표면을 효율적, 효과적으로 표현하기 위한 물리적 시뮬레이션 방법을 제안한다. 이 논문에서 표현하고자 하는 물은 깊이에 비해 너비가 매우 크고 상하 유동이 적은 상태로서, 이를 효율적으로 계산하기 위해 Navier-Stokes 방정식을 간략화한 천수방정식(shallow water equation)을 지배방정식으로 사용한다. 천수방정식의 대류항을 수치적으로 계산하기 위한 방법으로 기존의 Constrained Interpolation Profile(CIP)법을 개선하여, 수치적인 정확성을 높이고 물리량을 보존할 수 있는 Conservative Unsplit Semi-lagrangian CIP(CUSCIP)을 소개한다. 이 방법은 Kim 등이 제안한 USCIP[9]기법에서 사용하는 제약 조건에 적분값을 반영한 항을 추가하여 대류항을 계산한다. 실험결과를 통해, CUSCIP방법은 수치 소산(numerical dissipation)으로 인한 물리량 손실에 강건하여, 물결의 세밀함과 더불어 물결의 지속성이 향상됨을 알 수 있다.

#### Abstract

We propose a physical simulation method based on the shallow water equation(SWE) to represent water surface effectively. In this paper, the water which can be represented has a much larger width compared to the depth does not have a large vertical direction flow. In order to calculate the water flow efficiently, we start with the shallow water equation as the governing equation, which is a simplified version of the Navier-Stokes equation. In order to numerically calculate the advection term of the SWE, we introduce a new conservative USCIP(CUSCIP) method which improves the Constrained Interpolation Profile (CIP) method to preserve the physical quantity while increasing the numerical accuracy. The proposed method is based on Kim et. al.'s Unsplit Semi-lagrangian CIP[9], and calculates advection term with additional constraints on term that consider integral values. The experimental results show that the CUSCIP method is robust to the loss of physical quantity due to numerical dissipation, which improves wave detail and persistence.

**키워드:** 유체 시뮬레이션, 천수방정식, CIP법, 물리기반 애니메이션

**Keywords:** fluid simulation, shallow water equation, CIP, physical based animation

\*corresponding author: Oh-young Song/Sejong University(oysong@sejong.ac.kr)

## 1. 서론

지구상에서 관찰할 수 있는 자연 현상들의 사실적 표현은 컴퓨터 그래픽스 분야의 필수적인 목표이다. 그중에서도 물은 자연에서 가장 흔하게 관찰할 수 있는 물질 중 하나로서, 이를 사실적으로 표현하기 위한 기술은 매우 중요한 과제로 연구되어왔다. 현재는 전산유체역학의 이론을 이용한 시물레이션 기법으로 이와 같은 물의 움직임을 보다 사실적으로 재현하고 있다.

유체를 시물레이션하기 위해서는 유체의 움직임을 의미하는 지배방정식을 풀어야 한다. 이는 Navier-Stokes 방정식이라 불리는 비선형 편미분 방정식으로 표현된다. 유체 시물레이션은 실제의 물리량을 이산화하고 그 값을 컴퓨터의 반복 연산으로 풀어내는 방식으로 유체의 움직임을 재현한다. 물이 깊이에 비해 너비가 훨씬 크고 상하 유동이 적은 상태일 경우 3차원의 Navier-Stokes 방정식은 2차원으로 간략화할 수 있으며 이를 천수방정식(Shallow Water Equation; SWE)이라 한다.

이 논문에서는 천수방정식의 대류항을 푸는 과정에서 발생하는 수치적 손실을 줄이기 위해 Constrained Interpolation Profile(CIP)법을 도입할 것이며, 천수에서 특히 중요한 물리량인 높이값을 보존하기 위해 일반적인 CIP를 개선한 방법, CUSCIP(Conservative Unsplit Semi-lagrangian CIP)를 제안한다.

## 2. 관련 연구

Kass와 Miller[3]는 2차원 천수방정식으로 높이 필드를 구하여 수면을 표현하였다. 이후 Layton과 Panne[13]은 안정성을 보장하는 시물레이션 방법을 개발했다. Yuksel, House, Keyser[14]는 접근 방식을 바꾸어 높이값을 변화시키는 파동 자체를 하나의 입자(particle)로 보고 그 입자를 시물레이션하는 방법을 사용하였다. Jeschke와 Wojtan[15]은 입자를 파동묶음(wave packet)으로 확장하여 다양한 주파수의 수면파를 시물레이션 하는 방법을 개발하였다. Lee와 Han[16]은 2차원 상에 체적값을 갖는 입자들을 SPH[17]로 시물레이션하여 천수방정식을 푸는 방법을 제안하였다. 수면파를 이산 푸리에 변환으로 해석하여 움직임을 표현하는 방법들도 제안되었다[18,19]. 한편, 쇄파나 분무 현상을 표현 못하는 천수방정식의 한계를 극복하기 위해 다양한 논문들이 발표되었다. O'Brien과 Hodgins[20], Thürey와 동료들[21]은 이 문제를 입자 시스템(particle system)을 도입하여 해결하였다. 즉, 높이 필드에서 쇄파나 분무 현상이 일어날 곳을 찾아 그 위치에 입자를 생성하는 방식이다. Irving과 동료들[22]은 쇄파나 분무 같은 세밀한 현상이 물 표면에

서만 일어난다는 사실에 착안하여 2차원 격자 기반의 높이 필드와 일반적인 3차원 유체 시물레이션을 결합하는 방법을 제안하였다.

한편, 유체 시물레이션 대류항을 계산하는 가장 대표적인 수치 해법은 세미-라그랑지안 기법(Semi-Lagrangian method)이다. 전산 유체역학 분야에서 Courant, Isaacson, Rees[1]에 의해 제안되었고 그래픽스 분야에서는 Stam[2]이 최초로 소개하였다. 대류항을 수치적으로 풀 때 필요한 보간 과정에서의 정확도를 높이기 위해 Yabe와 동료들[4,5,6,7]은 CIP법을 고안하였다. CIP법은 물리량을 보간할 때 미분값을 함께 고려하는 방법이다. 최초의 CIP는 안정성을 보장하지 못하는 문제를 가지고 있었다. Song과 Shin, Ko는 CIP의 안정성 문제를 해결하기 위해 미분값을 보정하는 Monotonic CIP(MCIP)법을 제안하였다[8]. 이 방법은 Courant-Friedrichs-Lewy(CFL)조건에 무관하게 안정적이다. 이후 Kim 등은 [8]에서 야기되는 차원 분할 과정을 생략하기 위해 Unsplit Semi-lagrangian CIP(USCIP)법을 제안하였다[9]. 이 방법은 MCIP보다 정확도가 향상 되었으며, 계산 비용을 대폭 절감하였다. 물리량을 보존하기 위해 개발된 CIP-Conservative Semi-Lagrangian(CIP-CSL)법[10,11]은 적분값을 함께 고려한다. 2009년 Yabe와 Ogata는 이를 천수방정식에 적용하는 방법을 소개하였다[12].

## 3. 천수방정식의 수치해법

### 3.1 천수방정식

천수방정식을 유도하기 위해서는 다음과 같은 조건들이 필요하다.

- 1) 유체는 얇은(shallow) 상태이다. 이는 수평 방향의 크기가 수직 방향의 크기보다 월등히 크다는 것을 의미한다.
- 2) 유체는 유동이 적은, 안정적인 상태이다. 유체는 특정한 방향으로 흐르고 있지 않으며, 기본적으로 작용하는 외력은 중력뿐이다. 따라서 압력은 더 이상 미지의 값이 아니며, 깊이에 따라 선형으로 비례한다.
- 3) 유체는 높이값을 갖는 자유표면으로 표현된다(height field). 따라서 쇄파(breaking wave)나 분무(spray)현상은 표현할 수 없다.

이와 같은 조건으로부터 유체는 수직 방향으로의 속도 변화를 무시할 수 있으며, 연직 방향에 대하여 정역학적(hydrostatic)상태라는 가정이 가능하다. 따라서 유체의 압력은 중력의 영향만을 고려하여 유체의 깊이에 선형으로 비례한다. 여기서 주의할 점은 수직 방향의 속도가 반드시 0은 아니라는 점이다. 속도의 수직 성분은 유체가 존재하는

지면의 높이와 중력에 의해 발생하며, 수평 방향의 속도가 깊이에 변화에 영향을 받지 않을 뿐이다. 이와 같은 가정들을 고려하여, 일반적인 유체의 지배방정식인 Navier-Stokes 방정식으로부터 천수방정식을 유도할 수 있다.

점성을 고려하지 않을 경우, Navier-Stokes 방정식은 다음과 같이 표현된다:

$$\begin{aligned} \frac{\partial \vec{u}}{\partial t} &= -\vec{u} \cdot \nabla \vec{u} + \frac{\vec{f}}{\rho} - \frac{\nabla p}{\rho}, \\ \nabla \cdot \vec{u} &= 0. \end{aligned} \quad (1)$$

위 식에서 첫 번째 식은 유체의 속도에 대한 식이며, 두 번째 식은 비압축성에 대한 식이다.  $\vec{u}$ 는 유체의 속도 필드,  $t$ 는 시간,  $\rho$ 는 밀도,  $\vec{f}$ 는 외력, 그리고  $p$ 는 압력을 의미한다. 속도는 방향을 가지고 있으므로 3차원 벡터  $(u, v, w)$ 로 표현되며, 여기서 연직 방향의 성분  $v$ 에 대한 식만 살펴보면 다음과 같다:

$$\frac{\partial v}{\partial t} + u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} + w \frac{\partial v}{\partial z} + \frac{1}{\rho} \frac{\partial p}{\partial y} = -g, \quad (2)$$

$g$ 는 중력가속도이며 연직방향의 성분에만 영향을 미친다. 위의 가정에 의해, 성분  $v$ 의 대류항들을 무시할 수 있으며, 이에 따라 중력가속도  $g$ 에 의한 압력의 관계식으로 간략화 된다:

$$\frac{1}{\rho} \frac{\partial p}{\partial y} = -g. \quad (3)$$

$y = h$  ( $h$ 는 물 표면 지점의 높이)인 지점에서  $p = 0$ 이므로 이를  $p$ 에 대한 식으로 다시 쓰면, 아래와 같다:

$$p(x, y, z, t) = \rho g (h(x, y, t) - y). \quad (4)$$

수평 성분의 방정식에서 가정에 의해  $\frac{\partial u}{\partial y} = \frac{\partial w}{\partial y} = 0$ 이며, 식(4)에 의해 압력 구배에 대한 항은 높이의 구배로 치환할 수 있다:

$$\begin{aligned} \frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + w \frac{\partial u}{\partial z} + g \frac{\partial h}{\partial z} &= 0, \\ \frac{\partial w}{\partial t} + u \frac{\partial w}{\partial x} + w \frac{\partial w}{\partial z} + g \frac{\partial h}{\partial z} &= 0. \end{aligned} \quad (5)$$

이처럼 연직 방향의 속도 변화를 무시하고 수평 성분의 속도로 관계식을 새로 정립함으로써 속도에 대한 3차원 미분방정식을 2차원으로 간략화 하였다.

일반적으로 유체의 표면은 레벨셋 값(level set value)[23]  $\phi$ 가 0인 지점으로 생각할 수 있으며, 천수의 경우  $y = h$ 인 지점에 해당한다. 따라서  $\phi(x, y, z) = y - h(x, z)$ 로 치환이 가능하며, 이 값을 대류 방정식을 만족한다:

$$\begin{aligned} \frac{\partial \phi}{\partial t} + u \frac{\partial \phi}{\partial x} + v \frac{\partial \phi}{\partial y} + w \frac{\partial \phi}{\partial z} &= 0, \\ -\frac{\partial h}{\partial t} + u \left(-\frac{\partial h}{\partial x}\right) + v(1) + w \left(-\frac{\partial h}{\partial z}\right) &= 0. \end{aligned} \quad (6)$$

여기서, 수직 성분  $v$ 는 비압축성 조건을 이용하여 구할 수 있다. 비압축성 조건에 따라  $y$ 에 따른  $v$ 의 변화율은 아래와 같다:

$$\frac{\partial v}{\partial y} = -\frac{\partial u}{\partial x} - \frac{\partial w}{\partial z}. \quad (7)$$

$v$ 를 구하기 위해 필요한 유체와 지표면 사이의 경계조건은  $\vec{u} \cdot \hat{n} = 0$ 이며, 지표면  $b(x, z)$ 에서의 법선벡터는  $v(-\frac{\partial b}{\partial x}, 1, -\frac{\partial b}{\partial z})$ 에 비례하므로, 경계 조건에 적용하면,

$$v(x, y, z, t) = u \frac{\partial b}{\partial x} + w \frac{\partial b}{\partial z} - \left(\frac{\partial u}{\partial x} + \frac{\partial w}{\partial z}\right)(y - b), \quad (8)$$

로 정리된다. 이제 식(8)을 식(6)에 대입한 후 정리하면 천수의 높이값에 대한 편미분방정식이 아래와 같이 유도된다:

$$\frac{\partial (h - b)}{\partial t} + u \frac{\partial (h - b)}{\partial x} + w \frac{\partial (h - b)}{\partial z} = -(h - b) \left(\frac{\partial u}{\partial x} + \frac{\partial w}{\partial z}\right). \quad (9)$$

여기서, 지면으로부터 수면까지의 거리  $(h - b)$ 를  $\eta$ 로 치환하고 수평 방향의 속도,  $\vec{v} = (u, w)$ 에 대한 식(5)과 함께 정리하면 최종적으로 천수방정식은 아래와 같다:

$$\begin{aligned} \frac{\partial \vec{v}}{\partial t} + (\vec{v} \cdot \nabla) \vec{v} &= -g \nabla h, \\ \frac{\partial \eta}{\partial t} + (\vec{v} \cdot \nabla) \eta &= -\eta \nabla \cdot \vec{v}. \end{aligned} \quad (10)$$

이 수식이 의미하는 바를 살펴보면, 각각의 물리량  $\eta, \vec{v}$ 는 수평 속도  $\vec{v}$ 에 따라 대류하고, 속도는 높이값  $h$ 의 구배에 의해 발생하며,  $\eta$ 는 속도 필드의 발산(divergence)에 의해 변화하는 것을 알 수 있다. 이로써 풀어야 하는 공간 격자는 2차원으로 줄어들었다. 뿐만 아니라 압력을 미리 정해진 값으로 가정하고 있으므로 압력 분포를 알아내기 위한 선형 시스템을 풀 필요가 없는 것도 큰 이점이다.

### 3.2 천수방정식의 수치해법

천수방정식은 대류항(advection term)과 비대류항(non-advection term)으로 나뉘어지는데, 대류항은 Semi-lagrangian 기법으로 풀 수 있다.

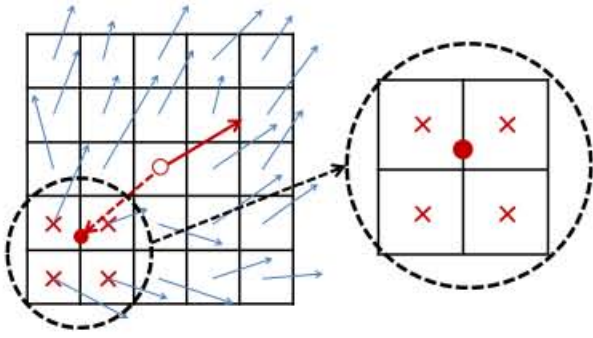


Figure 1: Back-tracking and interpolation in the Semi-lagrangian method

식(10)의 좌변이 대류항인데 이를 다시 쓰면,

$$\begin{aligned} \frac{\partial \vec{v}}{\partial t} + (\vec{v} \cdot \nabla) \vec{v} &= 0, \\ \frac{\partial \eta}{\partial t} + (\vec{v} \cdot \nabla) \eta &= 0. \end{aligned} \quad (11)$$

위 식은 속도와 물의 높이가 격자점 중심의 속도장을 따라 움직이는 것을 나타낸다. Semi-lagrangian 기법은 먼저, 속도장 위에 입자가 존재한다고 가정한다(Figure 1에서 격자 중앙에 빨간색 실선으로 표시된 원). 현재 격자의 중심에 입자들이 있고 속도장이 정의되어 있다면 그곳에 정의된 속도 벡터 값에 따라 그 입자들의 이전 타임 스텝에서의 위치를 역으로 추적하여 알아낼 수 있다(Figure 1에서 빨간 점선으로 표시된 벡터를 따라 이동한 지점의 원). 물론 역으로 계산된 입자의 위치는 격자의 중앙이 아닌 경우가 대부분이므로 인접한 네 개의 격자(X자 형태로 표시된 부분)를 기준으로 보간(interpolation)하는 과정이 필요하다. 이렇게 계산된 물리량을 현재의 값에 반영하여 대류항을 풀 수 있다. 대류 항을 푸는 알고리즘을 의사코드로 요약하면 다음과 같다.

```

advect( $\vec{s}, \vec{v}$ )
  for( $j=1 \rightarrow n-1$ )
    for( $i=1 \rightarrow n-1$ )
       $\vec{x} = (i \cdot \Delta x, j \cdot \Delta x)$ 
       $\vec{x}' = \vec{x} - \Delta t \cdot \vec{v}(\vec{x})$ 
       $s'(i, j) = \text{interpolate}(s, \vec{x}')$ 
    endfor
  endfor
  return( $s'$ )

```

식(10)의 우변인 비대류항은 유한차분법으로 풀 수 있다. 위에서 대류항을 푼 이후의 물리량을 각각  $\vec{v}^*$ ,  $\eta^*$ 라고 한다면, 최종적으로 구해지는 다음 타임스텝에서의 값  $\vec{v}^{n+1}$ ,  $\eta^{n+1}$ 은

$$\begin{aligned} \vec{v}^{n+1} &= \vec{v}^* - (g \nabla h) \Delta t, \\ \eta^{n+1} &= \eta^* - \eta^* (\nabla \cdot \vec{v}) \Delta t, \end{aligned} \quad (12)$$

이며, 유한차분법으로 변환하면 격자  $i, j$ 에서의 값은

$$\begin{aligned} u_{i,j}^{n+1} &= u^* - g \left( \frac{h(i-1, j) - h(i, j)}{\Delta x} \right) \Delta t, \\ v_{i,j}^{n+1} &= v^* - g \left( \frac{h(i, j-1) - h(i, j)}{\Delta x} \right) \Delta t, \\ \eta_{i,j}^{n+1} &= \eta^* - \eta^* \left( \frac{u(i+1, j) - u(i, j)}{\Delta x} + \frac{v(i, j+1) - v(i, j)}{\Delta x} \right) \Delta t, \end{aligned} \quad (13)$$

이 된다. 따라서, 속도와 높이에 대한 비대류항 식을 푸는 과정은 다음과 같이 요약된다.

```

updatevelocities( $h, u, v, g$ )
  for( $j=1 \rightarrow n-1$ )
    for( $i=2 \rightarrow n-1$ )
       $u(i, j) += g \left( \frac{h(i-1, j) - h(i, j)}{\Delta x} \right) \Delta t$ 
    endfor
  endfor
  for( $j=2 \rightarrow n-1$ )
    for( $i=1 \rightarrow n-1$ )
       $v(i, j) += g \left( \frac{h(i, j-1) - h(i, j)}{\Delta x} \right) \Delta t$ 
    endfor
  endfor

```

```

updateheight( $\vec{\eta}, \vec{v}$ )
  for( $j=1 \rightarrow n-1$ )
    for( $i=1 \rightarrow n-1$ )
       $\eta(i, j) -= \eta(i, j) \cdot \left( \frac{u(i+1, j) - u(i, j)}{\Delta x} + \frac{v(i, j+1) - v(i, j)}{\Delta x} \right) \Delta t$ 
    endfor
  endfor
  return( $\vec{\eta}$ )

```

위의 절차들을 종합하여, 천수방정식을 푸는 알고리즘은 다음과 같이 정리된다.

```

solveshallowwater( $\vec{\eta}, \vec{v}, b$ )
   $\vec{\eta} = \text{advect}(\vec{\eta}, \vec{v})$ 
   $u = \text{advect}(u, \vec{v})$ 
   $v = \text{advect}(v, \vec{v})$ 
  updateheight( $\vec{\eta}, \vec{v}$ )
   $h = \vec{\eta} + b$ 
  updatevelocities( $h, u, v$ )

```

#### 4. 천수방정식에서의 Constrained Interpolation Profile(CIP)법



#### 4.1 Constrained Interpolation Profile(CIP)법

Constrained Interpolation Profile(CIP)법으로 대류항을 푸는 방법을 설명하기 위해, 아래의 1차원의 대류 방정식을 보면,

$$\frac{\partial \phi}{\partial t} + u \frac{\partial \phi}{\partial x} = 0, \quad (14)$$

수치적인 해법으로 이 문제를 풀 것이므로, 격자점  $x_i, x_{i+1}$ 과 이에 대응하는 물리량  $\phi_i, \phi_{i+1}$ 만을 알 수 있고, 그 사이 지점에 대응하는 물리량은 보간하여 구한다.

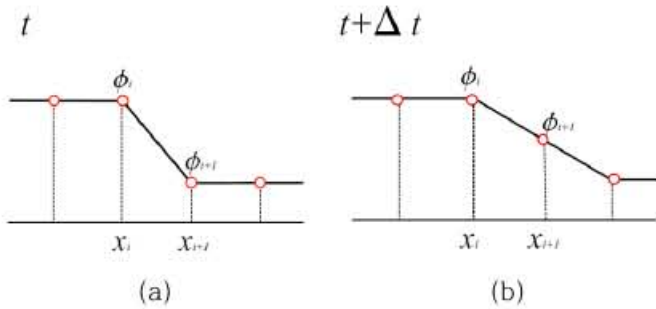


Figure 2: The changes of physical quantities( $\phi_i, \phi_{i+1}$ ) by linear interpolation method

가장 간단한 보간법은 선형 보간법이다. 선형 보간식은 식(15)와 같은 형태의 1차식이며, 선형 보간된 물리량의 변화는 Figure 2와 같다.

$$\phi(x) = (1-x)\phi_i + x\phi_{i+1}. \quad (15)$$

선형 보간법은 간단하고 안정적이지만 1차의 정확도만을 가지므로, Figure 2에서 보듯이 수치 소산이 빠르게 일

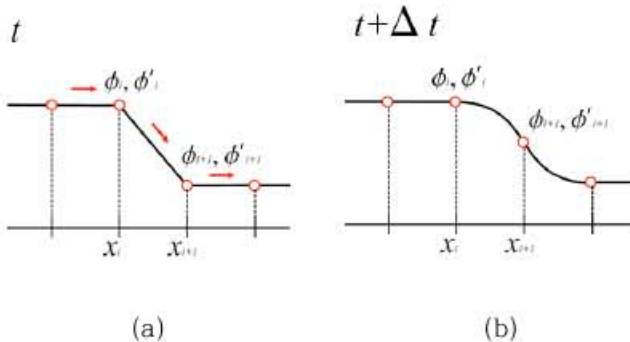


Figure 3: The changes of physical quantities( $\phi_i, \phi_{i+1}$ ) by Constrained Interpolation Profile(CIP) method

어난다. CIP법은 Figure 3에서 보는 바와 같이, 값을 보간하는 과정에서 주어진 값뿐만 아니라 그 미분값까지 함께 감안하는 것이 기본 아이디어이다. 대류항의 미분식은 식

(14)를  $x$ 로 미분하여,

$$\frac{\partial \phi_x}{\partial t} + u \frac{\partial \phi_x}{\partial x} = -u_x \frac{\partial \phi}{\partial x} \quad (16)$$

로 표현할 수 있다. 즉, 각 격자점에 정의된 값  $\phi$ 와 그 미분값  $\phi_x$ 를 모두 알아내면, 이를 기반으로  $\phi$ 에 대한 3차 보간식을 세울 수 있다. 편의상 격자점 사이의 거리를 1로 정규화하였을 때, 보간식은 다음과 같다:

$$\phi(x) = [(C_0 x + C_1)x + \phi_x]x + \phi_i. \quad (17)$$

여기서,  $C_0$ 와  $C_1$ 은 격자점 상의  $\phi$ 와  $\phi_x$ 로 이루어진 계수로,

$$\begin{aligned} C_0 &= \phi_{x_i} + \phi_{x_{i+1}} - 2(\phi_{i+1} - \phi_i), \\ C_1 &= 3(\phi_{i+1} + \phi_i) - 2(\phi_{x_i} - \phi_{x_{i+1}}), \end{aligned} \quad (18)$$

이 된다. 천수방정식에 적용하기 위해서 이를 2차원으로 확장할 경우, CIP식은 두 개의 공간 변수  $x, y$ 에 대한 3차 다항식으로 표현된다:

$$\phi(x, y) = \sum_{0 \leq i+j \leq 3} C_{ij} x^i y^j. \quad (19)$$

CIP를 처음 소개한 Yabe와 동료들은 자신의 논문에서 CIP의 2차원 및 3차원 다항식에 대한 계수를 밝힌 바 있다 [7]. CIP는 3차의 정확도를 가지고 있으며 이로 인해 얻을 수 있는 시각적인 세밀함은 선형 보간법보다 월등하다는 것이 여러 논문을 통해 입증되었다[8,9]. 하지만 아래와 같은 두 가지의 문제가 있다.

- 1) 수치적 안정성이 떨어진다. 3차 보간식의 특성상 격자점 사이의 값을 벗어나는 범위로 보간이 되면 시뮬레이션 시스템이 점차로 불안정해지는 원인이 될 수 있다.
- 2) 물리량이 보존되지 못한다. 수치소산으로 인한 물리량의 감소는 앞서 기술한 천수방정식에서 높이값  $\eta$ 의 손실을 가져오며, 결과적으로 물의 총량이 서서히 줄어들게 된다.

#### 4.2 Unsplit Semi-lagrangian Constrained Interpolation Profile(USCIP)법

Unsplit Semi-lagrangian CIP(USCIP)법은 2008년 Kim등[9]에 의해 제안된 방법으로, 2차원 이상의 격자에서 기존의 CIP 다항식은 주어진 미분값들의 정보를 완전히 활용하지 않고 있다는 점에 착안하여 여기에 새로운 항을 추가함으로써 나머지 미분값들을 반영하는 방법이다.

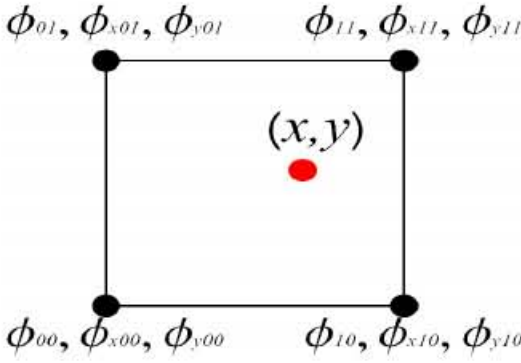


Figure 4: physical quantities  $\phi$  and their x-derivatives and y-derivatives in a 2D grid

Figure 4에서 보듯이, 4개의 2차원 격자점에서 우리는 물리량  $\phi$ 와 그  $x, y$  방향으로의 편미분값을 사전에 알고 있다고 가정하면, 이 값들을 근거로 연립방정식을 세워 식 (19)에서 제시된 2차원 CIP 다항식의 계수를 아래와 같이 구할 수 있다:

$$\begin{aligned}
 \phi_{00} &= C_{00} \\
 \phi_{10} &= C_{00} + C_{10} + C_{20} + C_{30} \\
 \phi_{01} &= C_{00} + C_{01} + C_{02} + C_{03} \\
 \phi_{11} &= C_{00} + C_{01} + C_{02} + C_{03} + C_{10} + C_{11} + C_{12} + C_{20} + C_{21} + C_{30} \\
 \phi_{x00} &= C_{10} \\
 \phi_{x10} &= C_{10} + 2C_{20} + 3C_{30} \\
 \phi_{x01} &= C_{10} + C_{11} + C_{12} \\
 \phi_{y00} &= C_{01} \\
 \phi_{y10} &= C_{01} + C_{11} + C_{21} \\
 \phi_{y01} &= C_{01} + 2C_{02} + 3C_{03}
 \end{aligned} \tag{20}$$

CIP 다항식의 개수는 10개이므로 계수 또한 10개를 구해야 하지만, 우리는 12개의 값을 알고 있으므로 이를 반영하기 위해 기존의 다항식에 2개의 항을 더 추가할 수 있다.

$$\phi(x, y) = \sum_{0 \leq i+j \leq 3} C_{ij} x^i y^j + C_{31} x^3 y + C_{13} x y^3. \tag{21}$$

추가적인 항의 차수를 위와 같이 선택한 이유와 계수의 값은 해당 논문에 구체적으로 서술되어 있다[9].

### 4.3 적분값을 이용한 Constrained Interpolation Profile(CIP)법

적분값을 이용한 CIP법은 Yabe 등이 CIP-CSL(CIP-Conservative Semi Lagrangian)법[12]을 최초로 제안하였다. CIP-CSL법은 격자 구간 사이의 체적을 보간 정보로 활용함으로써 물리량을 보존하는 방법이다. 1차원의 경우, 적분값이 추가되므로 보간식은 4차로 구성되며, 다섯 개의 항이 필요하다:

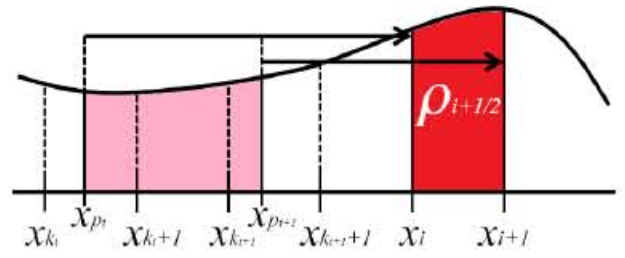


Figure 5: 1D advection of an integration profile

$$\phi(x) = C_4 x^4 + C_3 x^3 + C_2 x^2 + \phi_{x0} x + \phi_0. \tag{22}$$

격자 간격이 1로 정규화 되어있을 경우, 주어진 다섯 개의 프로파일을 이용하여 계수를 구하면 다음과 같다:

$$\begin{aligned}
 C_4 &= [-5(6(\phi_1 + \phi_0) - (f_{x1} - f_{x0}) - 12\rho_{i\alpha\beta})]/2, \\
 C_3 &= [4(7\phi_1 + 8\phi_0) - (\phi_{x1} - (3/2)\phi_{x0}) - 15\rho_{i\alpha\beta}], \\
 C_2 &= [-3(4(2\phi_1 + 3\phi_0) - (\phi_{x1} - 3\phi_{x0}) - 20\rho_{i\alpha\beta})]/2.
 \end{aligned} \tag{23}$$

한편, 적분값은 해당 격자를 통과하는 유량(flux)에 따라 갱신된다.

$$\rho_{i+1/2}^{n+1} = \rho_{i+1/2}^n + \Delta\rho_i - \Delta\rho_{i+1}. \tag{24}$$

적분값 갱신에 필요한 유량은 Semi-Lagrangian으로 구한다. Figure 5와 같이 먼저 격자점의 위치를 되추적한 다음, 해당 범위 내의 적분값을 가져오면 된다. Figure 5에서, 다음 타임 스텝의 유량  $\rho_{i+1/2}^{n+1}$ 은

$$\begin{aligned}
 \rho_{i+1/2}^{n+1} &= \int_{x_i}^{x_{k+1/2}} \phi dx + \dots + \int_{x_{k+1/2}}^{x_{i+1}} \phi dx \\
 &= \Delta\rho_{x_{k+1}} + \dots + [\rho_{k_{i+1}+1/2} - \Delta\rho_{k_{i+1}+1}],
 \end{aligned} \tag{25}$$

이 된다. 편의상 CFL수치, 즉  $u\Delta t/\Delta x < 1$ 이라고 가정한다면 구해지는 적분값은

$$\rho_{i+1/2}^{n+1} = \Delta\rho_{x_{k_{i+1}}} + \rho_{k_{i+1}+1/2} - \Delta\rho_{k_{i+1}+1} \tag{26}$$

으로 표현할 수 있다.

2차원의 경우, Figure 6:(a)와 같이 각 격자점을 추적했을 때, 적분되는 영역의 형태가 비정규적으로 나타난다. 이와 같은 형태로 나타나는 적분 영역은 바로 구할 수 없기 때문에 차원분할(dimensional splitting)과정이 필요하다. 즉, Figure 6:(b)와 같이 격자점의 경계면에 1차원 적분값을 저장하고, 각각의 방향 별로 들어오는 유량을 계산하는 것이다.

차원 분할은 각각의 방향 별로 격자점의 위치를 하나씩 되추적 한 다음 이에 대해 적분 연산을 수행하므로 높은



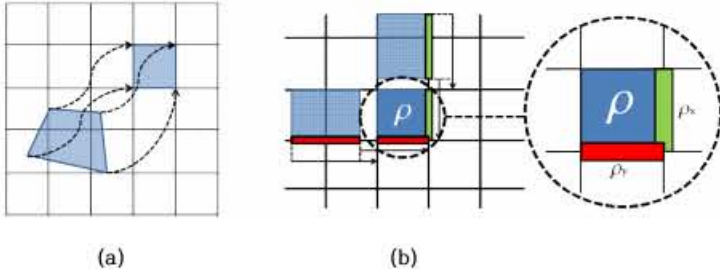


Figure 6: 2D advection of an integration profile

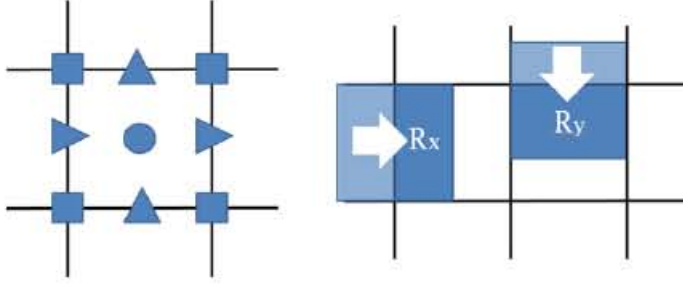


Figure 7: The proposed conservative USCIP method

계산 비용이 필요하다. 이를 절감하기 위해 본 논문에서는 Conservative USCIP(CUSCIP)라고 명명한 새로운 CIP기법을 제안한다. 앞서 살펴본 바와 같이, Kim은 Yabe가 최초 제안한 CIP다항식에 차수가 대칭(Symmetric)이면서도 최소의 차수를 갖는 2개의 항을 추가하였다. 본 논문에서 제안하는 방법은 여기에 대칭을 유지하는 또 다른 항  $C_{22}x^2y^2$ 를 아래와 같이 추가한다.

$$\phi(x,y) = \sum_{0 \leq i+j \leq 3} C_{ij}x^i y^j + C_{31}x^3y + C_{13}xy^3 + C_{22}x^2y^2. \quad (27)$$

이로써 다항식은 13개의 항을 가지게 되며, 4개의 격자 점과 그 중앙에 저장된 13개의 보간 정보로 계수를 구할 수 있다.

$$\begin{aligned} C_{31} &= -2\phi_{00} + 2\phi_{01} + 2\phi_{10} - 2\phi_{11} - \phi_{x00} + \phi_{x01} - \phi_{x10} + \phi_{x11}, \\ C_{22} &= -(3/2)(6\phi_{00} + 6\phi_{01} + 6\phi_{10} + 6\phi_{11} + \phi_{x00} + \phi_{x01} - \phi_{x10} \\ &\quad - \phi_{x11} + \phi_{y00} - \phi_{y01} + \phi_{y10} - \phi_{y11} - 24\rho), \\ C_{13} &= -2\phi_{00} + 2\phi_{01} + 2\phi_{10} - 2\phi_{11} - \phi_{y00} - \phi_{y01} + \phi_{y10} + \phi_{y11}, \\ C_{30} &= 2\phi_{00} - 2\phi_{10} + \phi_{x00} + \phi_{x10}, \\ C_{21} &= 1/2(24\phi_{00} + 12\phi_{01} + 12\phi_{10} + 24\phi_{11} + 7\phi_{x00} - \phi_{x01} \\ &\quad - \phi_{x10} - 5\phi_{x11} + 3\phi_{y00} - 3\phi_{y01} + 3\phi_{y10} - 3\phi_{y11} - 72\rho), \\ C_{12} &= 1/2(24\phi_{00} + 12\phi_{01} + 12\phi_{10} + 24\phi_{11} + 3\phi_{x00} + 3\phi_{x01} \\ &\quad - 3\phi_{x10} - 3\phi_{x11} + 7\phi_{y00} - \phi_{y01} - \phi_{y10} - 5\phi_{y11} - 72\rho), \\ C_{03} &= 2\phi_{00} - 2\phi_{01} + \phi_{y00} + \phi_{y01}. \end{aligned}$$

$$\begin{aligned} C_{20} &= -3\phi_{00} + 3\phi_{10} - 2\phi_{x00} - \phi_{x10}, \\ C_{11} &= 1/2(-20\phi_{00} - 16\phi_{01} - 16\phi_{10} - 20\phi_{11} - 5\phi_{x00} - \phi_{x01} \\ &\quad + 3\phi_{x10} + 3\phi_{x11} - 5\phi_{y00} + 3\phi_{y01} - \phi_{y10} + 3\phi_{y11} + 72\rho), \\ C_{02} &= -3\phi_{00} + 3\phi_{01} - 2\phi_{y00} - \phi_{y01}, \\ C_{10} &= \phi_{x00}, \\ C_{01} &= \phi_{y00}, \\ C_{00} &= \phi_{00} \end{aligned} \quad (28)$$

$\phi$ 와  $\phi_x, \phi_y$ 는 Semi-lagrangian기법과 동일한 방법으로 갱신하며, 적분값  $\rho$ 는 현재 타임스텝의 값  $\rho^n$ 으로부터 명시적(explicit)으로 이동한다.

Figure 7과 같이 정사각형 모양의 적분 영역이 이동했다면, 적분값  $\rho$ 가 이웃 격자 영역으로 유입되는 각각의 유량(flux)은  $R_x, R_y$  영역의 물리량을 각각 이중 적분함으로써 구할 수 있다.

$$\begin{aligned} \Delta\rho_x &= \iint_{R_x} \phi(x,y) dx dy, \\ \Delta\rho_y &= \iint_{R_y} \phi(x,y) dx dy. \end{aligned} \quad (29)$$

이와 같이 각 영역별로 나누어진 적분값을 해당 격자의  $\Delta\rho$ 로 사용하면, 모든 격자점의  $\rho$ 값을 갱신할 수 있다. 이 방법을 사용하면 모든 영역에 걸쳐 적분 값을 감안하므로 물리량을 보존할 수 있으며, 또한 차원 분할 과정이 생략되므로 계산 시간이 절감된다. 또한, 기존의 USCIP에 적분값이라는 보간 정보를 하나 더 추가하기 때문에 보다 정확한 결과를 얻을 수 있다.

## 5. 실험 결과

본 논문은 인텔 i7 4.2GHz CPU 32GB RAM, NVIDIA GeForce GTX 1080 하드웨어 환경에서 구현하였다. 시뮬레이션은 CPU에서 연산되며, 2차원 물결 렌더링은 GPU에서 연산되도록 하였다.

본 논문에서 사용한 방법의 정확도를 검증하기 위해, 고정된 속도장을 따라 움직이는 1차원 대류식을 푸는 과정을 시각화하였다(Figure 8).

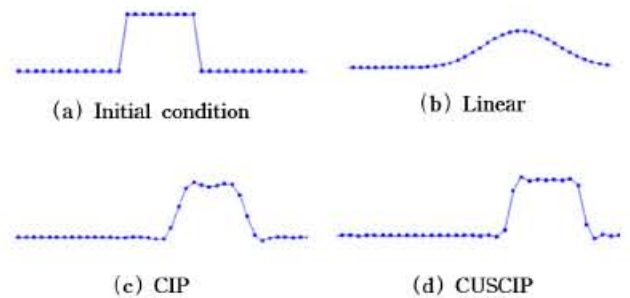


Figure 8: 1D advection examples of linear, CIP and CUSCIP

선형 보간 모델에 비해 CIP와 CUSCIP는 초기조건의 수치적 소산이 없음을 확인할 수 있었다. 게다가 CUSCIP는 제약조건으로 적분값을 추가하였으므로 셀 단위의 물리량이 보존됨을 확인할 수 있었다.

다음 실험으로 1차원 천수방정식으로부터 물의 파동을 시뮬레이션 하였다. Figure 9에서 CUSCIP의 경우, 선형 모델보다 월등히 수치소산이 적음을 확인하였다. CUSCIP와 CIP를 직접적으로 비교한 결과(Figure 10), CUSCIP는 CIP보다 물결의 더 오래 지속되고 높이값의 총량도 유지됨을 확인할 수 있었다. 2차원 천수방정식의 시뮬레이션 결과는 GLSL을 이용하여 렌더링 하였으며, 모두 실시간으로 구동된다(Figure 11과 Figure 12 참조). 시뮬레이션 해상도가 높지 않은 환경에서도 CUSCIP 방법은 상세한 물결의 움직임, 즉, 높은 주파수의 물결까지 표현할 수 있음을 확인하였다.

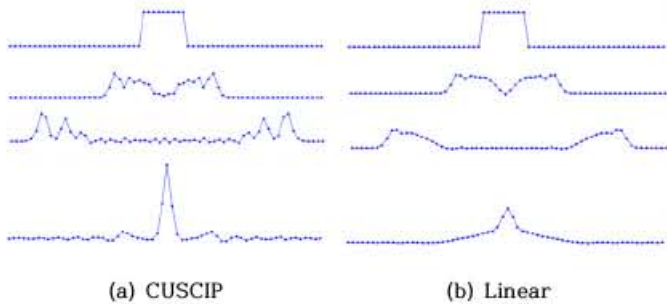


Figure 9: 1D shallow water simulation by CUSCIP and linear

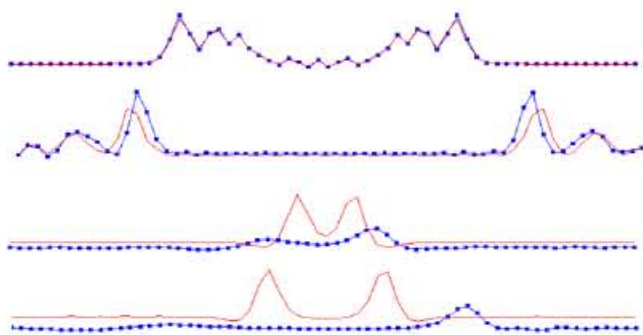


Figure 10: 1D shallow water simulation by CUSCIP (red color) and CIP (blue color)

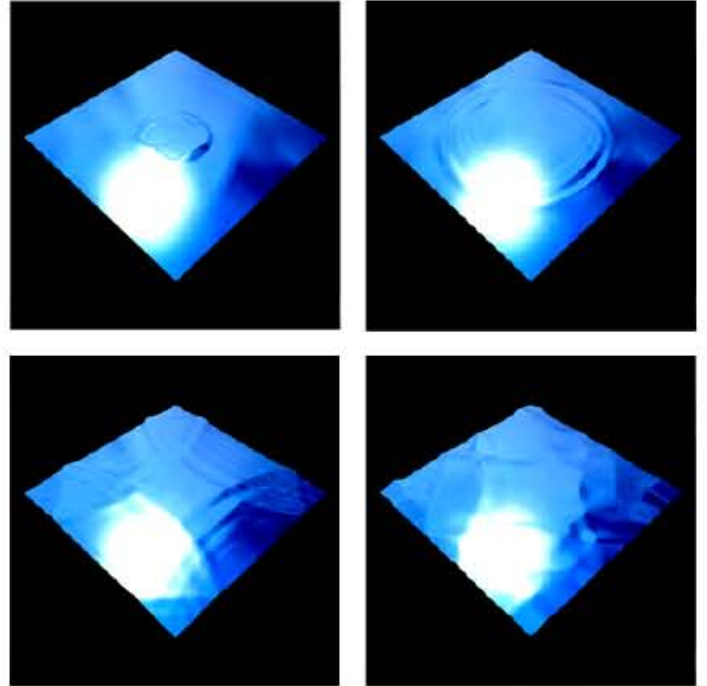


Figure 11: 2D shallow water simulation by CUSCIP (resolution: 64x64)

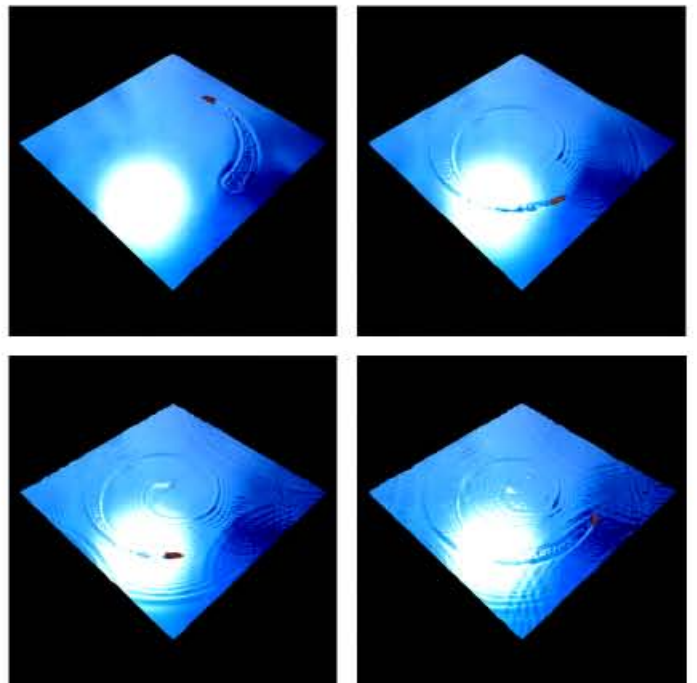


Figure 12: 2D shallow water simulation with boat by CUSCIP (simulation resolution 128x128)

## 6. 결론 및 향후 연구

이 논문은 천수에서 물결의 세부묘사를 표현하면서 물리량은 보존할 수 있도록 하는 CUSCIP 방법을 개발하였



다. 본 논문에서 제시한 방법이 천수방정식의 높이값 보존을 위해 시작된 것이기는 하지만, 활용적인 측면에서 천수방정식으로 한정할 필요는 없다. 물리량의 보존은 모든 유체 시뮬레이션에 필요한 특성이므로 해당 논문의 방법을 일반화하여 모든 유체로 확장하는 것은 중요한 연구 방향이 될 것이다.

본 논문에서 제시한 CUSCIP 방법의 한계는 상당히 큰 시뮬레이션 시간 간격이 제시될 경우, 시뮬레이션 안정성이 보장되지 못한다는 점이다. 이는 격자점의 값을 넘어서 적분값을 가져오게 되며, 각 격자에서의 물리량의 보존을 강제하기 위해 불필요한 진동이 생길 수 있다. 물론 적절한 시간 간격에서의 시뮬레이션을 수행할 경우는 수치적인 안정성은 보장된다.

## 감사의 글

본 연구는 과학기술정보통신부 및 정보통신기획평가원의 대학ICT연구센터지원사업의 연구결과로 수행되었음 (IITP-2019-2016-0-00312)

## 참고 문헌

- [1] R. Courant, E. Isaacson, and M. Rees, "On the solution of nonlinear hyperbolic differential equations by finite differences," *Comm. Pure Appl. Math.*, 5:243, 1952.
- [2] J. Stam, "Stable fluids," *Computer Graphics(Proc. ACM SIGGRAPH '99)*, 33:121-128, 1999.
- [3] M. Kass and G. Miller, "Rapid, stable fluid dynamics for computer graphics," *Computer Graphics(Proc. ACM SIGGRAPH '90)*, 24(4):49-57, 1990.
- [4] H. Takewaki, A. Nishiguchi, and T. Yabe, "Cubic interpolated pseudoparticle method(cip) for solving hyperbolic type equations," *J. Comp. Phys.*, 61:261, 1985.
- [5] H. Takewaki and T. Yabe, "The cubic interpolated pseudoparticle method(cip): Application to nonlinear and multi-dimensional hyperbolic equations," *J. Comp. Phys.*, 70:355, 1987.
- [6] T. Yabe and T. Aoki, "A universal solver for hyperbolic equations by cubic polynomial interpolation i. one-dimensional solver," *J. Comp. Phys. Comm.*, 66:219-232, 1991.
- [7] T. Yabe and T. Aoki, "A universal solver for hyperbolic equations by cubic polynomial interpolation ii. two- and three dimensional solvers," *J. Comp. Phys. Comm.*, 66:233-242, 1991.
- [8] O.-y. Song, H. Shin, and H.-S. Ko, "Stable but Nondissipative Water," *ACM Transactions on Graphics* Vol. 24, No.1, pp 81-97, 2005.
- [9] D. Kim, O.-y. Song, and H.-S. Ko, "A Semi-Lagrangian CIP Fluid Solver without Dimensional Splitting," *Computer Graphics Forum(Proc. EUROGRAPHICS 2008)* Vol. 27, No. 2, pp 467-476
- [10] T. Yabe, R. Tanaka, T. Nakamura, and F. Xiao, "An exactly conservative semi-lagrangian scheme(cip-csl) in one dimension," *Mon. Weather Rev.*, 129:332, 2001.
- [11] T. Nakamura, R. Tanaka, T. Yabe, and K. Takizawa, "Exactly conservative semi-lagrangian scheme for multi-dimensional hyperbolic equations with directional splitting technique," *J. Comp. Phys.*, 174:171-207, 2001.
- [12] T. Yabe and Y. Ogata, "Conservative semi-lagrangian cip technique for the shallow water equations," *Comput. Mech.* Vol. 46, No 1, pp. 125-134, 2010.
- [13] A. T. Layton and M.v.d. Panne, "A numerically efficient and stable algorithm for animating water waves," *the Visual Computer*, Vol. 18, No. 1, pp. 41-53, 2002.
- [14] C. Yuksel, D. H. House, and J. Keyser, "Wave particles," *ACM Transactions on Graphics (SIGGRAPH 2007)* 26, 2, 2007.
- [15] S. Jeschke and C. Wojtan, "Water Wave Packets," *ACM Transactions on Graphics (SIGGRAPH 2017)* 36, 4, 2017
- [16] H. Lee and S. Han, "Solving the shallow water equations using 2D SPH particles for interactive applications," *The Visual Computer*, Vol. 26, No. 6-8, 865-872, 2010.
- [17] J. J. Monaghan, "An introduction to SPH," *Com. Phy. Comm* Vol. 48, pp. 88-96, 1988.
- [18] J. Tassendorf, "eWave: Using an Exponential Solver on the iWave Problem," *Technical Note*, 2014.
- [19] C. J. Horvath, "Empirical Directional Wave Spectra for Computer Graphics," *In Proceedings of the 2015 Symposium on Digital Production (DigiPro '15)*. ACM.
- [20] J. F. O'Brien and J. K. Hodgins, "Dynamic simulation of splashing fluids," *In CA '95: Proc. Computer Animation*, p.198. Washington, DC: IEEE Computer Society, 1995.
- [21] N. Thuerey, F. Sadlo, S. Schirm, M. Mueller-Fischer, M. Gross, "Real-time Breaking Waves for Shallow Water Simulations," *Proceedings of the Pacific Conference on Computer Graphics and Applications (Maui, Hawaii, USA, October 29-30, 2007)*, pp. 39-46, 2007
- [22] G. Irving, E. Guendelman, F. Lossaso, and R. Fedkiw, "Efficient simulation of large bodies of water by coupling two and three dimensional techniques," *ACM Transactions*

on Graphics (SIGGRAPH 2006) 25, 3, 2006

[23] S. Osher and R. Fedkiw, "Level set methods and dynamic implicit surfaces," *New York: Springer-Verlag*, 2002.

## < 저자 소개 >



- 전 세 중
- 2009년 세종대학교 디지털콘텐츠학과 (학사)
- 2011년 세종대학교 디지털콘텐츠학과 공학석사
- 2011년-2014년 네오플 던파개발실 프로그래머
- 2014년-2015년 라인플레이 개발팀 프로그래머
- 2015년-2017년 오리진게임즈 개발팀 프로그래머
- 2017년-2019년 메이커스게임즈 개발팀 프로그래머
- 2019년-현재 직방 개발팀 프로그래머
- 관심분야: 컴퓨터그래픽스, 물리기반 애니메이션, 게임 프로그래밍, 웹 프로그래밍
- <https://orcid.org/0000-0003-1156-8198>



- 송 오 영
- 1998년 서울대학교 전기공학부 (학사)
- 2000년 서울대학교 전기컴퓨터공학부 공학석사
- 2004년 서울대학교 전기컴퓨터공학부 공학박사
- 2004년-2006년 서울대학교 자동화시스템공동연구소, 박사후 연구원
- 2006년-현재 세종대학교 디지털콘텐츠/소프트웨어학과 교수
- 관심분야: 컴퓨터그래픽스, 물리기반 애니메이션, 가상현실/증강현실, 인공지능
- <https://orcid.org/0000-0002-7142-5976>