Основы

Задача– получить новые значения высот точек на основе текущих. Для этого мы решаем дифференциальные уравнения, описывающие поведение водной поверхности.

Основных подзадач 2, у каждой есть несколько вариантов решения:

- Продифференцировать высоты по времени (составить систему уравнений).
  - Волновое уравнение
  - Уравнения мелкой жидкости
- Проинтегрировать высоты по времени (решить систему уравнений).
  - Метод Эйлера
  - Метод Рунге-Кутты
  - Метод Верле

Дифференцирование

Для каждой точки водной поверхности помимо координат x и y необходимо хранить другие значения. Это, само собой, высота точки h, а также ее скорости. Эти величины удобно хранить и обрабатывать вместе как вектор. Данный вектор будет обозначаться как P.

Задача дифференцирования в рамках работы - составление функции

$$f(P) = \frac{\partial P}{\partial t}$$

Метод конечных разностей

В работе значения некоторых необходимых производных мы не можем получить. В качестве решения этой проблемы используется замена этих производных на разностные схемы. Их суть в следующем:

Если мы хотим получить производную A по B, в точке n, то для ее приближения с помощью разностной схемы достаточно знать разницы значений A и B между нашей точкой и ее соседом:

$$\frac{\partial A}{\partial B} = \frac{A_n - A_{n-1}}{B_n - B_{n-1}}$$

Более высокая точность будет, если использовать 2 соседей, но брать значения не из них, а из точек между текущей и соседями (метод центральных конечных разностей):

$$\frac{\partial A}{\partial B} = \frac{A_{n+0.5} - A_{n-0.5}}{B_{n+0.5} - B_{n-0.5}}$$

Такие точки существуют не всегда (зато их вполне можно интерполировать, просто взяв средние значения).

Важно: в формулах ниже используются просто индексы соседей ($n+1$ и $n-1$). Это немного снижает точность. При необходимости их можно заменить описанными выше индексами.

Волновое уравнение

$$\ddot{h} = \upsilon_p^2 \Delta h = \upsilon_p^2 \left(\frac{\partial^2 h}{\partial x^2} + \frac{\partial^2 h}{\partial y^2}\right),$$

где $\ddot{h}$ - вторая производная высоты, Δ - лапласиан, υ_p – фазовая скорость, определяющая скорость движения волн.

Данное уравнение в таком виде нам не подходит из-за производной второго порядка. Однако эту проблему легко решить с помощью замены:

$$\begin{cases} \dot{h} = \upsilon \\ \dot{\upsilon} = \upsilon_p^2 (\frac{\partial^2 h}{\partial x^2} + \frac{\partial^2 h}{\partial y^2}) \end{cases},$$

где υ - скорость изменения высоты точки, она же вертикальная скорость.

Так как эта скорость является параметром точки, то

$$P = \begin{pmatrix} h \\ \upsilon \end{pmatrix}, \text{ а } f(P) = f\begin{pmatrix} h \\ \upsilon \end{pmatrix} = \begin{pmatrix} \upsilon \\ \upsilon_p^2 (\frac{\partial^2 h}{\partial x^2} + \frac{\partial^2 h}{\partial y^2}) \end{pmatrix}$$

После применения метода конечных разностей, получим:

$$f(P_{i,j}) = \begin{pmatrix} \upsilon_{i,j} \\ \upsilon_p^2 (\frac{h_{i+1,j} + h_{i-1,j} - 2h_{i,j}}{\Delta x^2} + \frac{h_{i,j+1} + h_{i,j-1} - 2h_{i,j}}{\Delta y^2}) \end{pmatrix},$$

где Δx и Δy - шаги сетки (расстояния между соседними точками по разным направлениям).

Если $\Delta x = \Delta y$, то формулу можно сократить:

$$f(P_{i,j}) = \begin{pmatrix} \upsilon_{i,j} \\ \frac{\upsilon_p^2}{\Delta x^2} (h_{i+1,j} + h_{i-1,j} + h_{i,j+1} + h_{i,j-1} - 4h_{i,j}) \end{pmatrix}$$

Уравнения мелкой жидкости

$$\begin{cases} \dfrac{\partial h}{\partial t} = -\dfrac{\partial(h\mu)}{\partial x} - \dfrac{\partial(h\upsilon)}{\partial y} \\ \dfrac{\partial(h\mu)}{\partial t} = -\dfrac{\partial}{\partial x}\left(h\mu^2 + \dfrac{gh^2}{2}\right) - \dfrac{\partial(h\mu\upsilon)}{\partial y}, \\ \dfrac{\partial(h\upsilon)}{\partial t} = -\dfrac{\partial}{\partial y}\left(h\upsilon^2 + \dfrac{gh^2}{2}\right) - \dfrac{\partial(h\mu\upsilon)}{\partial x} \end{cases}$$

где μ и υ - горизонтальные скорости по x и y соответственно, g – ускорение свободного падения (определяет скорость движения волн).

Данная система состоит из уравнений первого порядка, однако, производные берутся от произведений, что требует замены:

$U = h\mu$

$V = h\upsilon$

$$\begin{cases} \dfrac{\partial h}{\partial t} = -\dfrac{\partial U}{\partial x} - \dfrac{\partial V}{\partial y} \\ \dfrac{\partial U}{\partial t} = -\dfrac{\partial}{\partial x}\left(\dfrac{U^2}{h} + \dfrac{gh^2}{2}\right) - \dfrac{\partial(\frac{UV}{h})}{\partial y} \\ \dfrac{\partial V}{\partial t} = -\dfrac{\partial}{\partial y}\left(\dfrac{V^2}{h} + \dfrac{gh^2}{2}\right) - \dfrac{\partial(\frac{UV}{h})}{\partial x} \end{cases}$$

Можно заметить вероятное деление на 0, при средней высоте равной 0. Но основное условие данных уравнений - горизонтальный масштаб должен быть много больше вертикального, что означает, что высоты должны мало колебаться относительно среднего значения:

$h = H \pm \varepsilon;\ \varepsilon \ll H$, где H - средняя высота, ε - амплитуда волн/колебаний. Это условие противоречит возможности высоты стать нулевой. Для гарантии можно в системе заменить h на H.

Система уравнений мелкой жидкости содержит 3 параметра точки:

$$P = \begin{pmatrix} h \\ U \\ V \end{pmatrix}, \text{ а } f(P) = f\begin{pmatrix} h \\ U \\ V \end{pmatrix} = \begin{pmatrix} -\dfrac{\partial U}{\partial x} - \dfrac{\partial V}{\partial y} \\ -\dfrac{\partial}{\partial x}\left(\dfrac{U^2}{h} + \dfrac{gh^2}{2}\right) - \dfrac{\partial(\frac{UV}{h})}{\partial y} \\ -\dfrac{\partial}{\partial y}\left(\dfrac{V^2}{h} + \dfrac{gh^2}{2}\right) - \dfrac{\partial(\frac{UV}{h})}{\partial x} \end{pmatrix}$$

Применим метод конечных разностей:

$$f(P_{i,j}) = \begin{pmatrix} -\frac{U_{i+1,j}-U_{i-1,j}}{2\Delta x} - \frac{V_{i,j+1}-V_{i,j-1}}{2\Delta y} \\ -\frac{\left(\frac{U_{i+1,j}^2}{h_{i+1,j}} + \frac{gh_{i+1,j}^2}{2}\right) - \left(\frac{U_{i-1,j}^2}{h_{i-1,j}} + \frac{gh_{i-1,j}^2}{2}\right)}{2\Delta x} - \frac{\left(\frac{U_{i,j+1}V_{i,j+1}}{h_{i,j+1}}\right) - \left(\frac{U_{i,j-1}V_{i,j-1}}{h_{i,j-1}}\right)}{2\Delta y} \\ -\frac{\left(\frac{V_{i,j+1}^2}{h_{i,j+1}} + \frac{gh_{i,j+1}^2}{2}\right) - \left(\frac{V_{i,j-1}^2}{h_{i,j-1}} + \frac{gh_{i,j-1}^2}{2}\right)}{2\Delta y} - \frac{\left(\frac{U_{i+1,j}V_{i+1,j}}{h_{i+1,j}}\right) - \left(\frac{U_{i-1,j}V_{i-1,j}}{h_{i-1,j}}\right)}{2\Delta x} \end{pmatrix}$$

Данная система может приводить к артефактам (визуально – «зубастость» воды, не растущая со временем). Причина этого в том, что для вычисления значений в точке используются U и V других точек, но производные берутся для собственных U и V. Исправить это можно, храня часть значений в других точках. Один из вариантов исправлений:

$$f(P_{i,j}) = \begin{pmatrix} -\frac{U_{i+1,j}-U_{i,j}}{\Delta x} - \frac{V_{i,j+1}-V_{i,j}}{\Delta y} \\ -\frac{\left(\frac{U_{i,j}^2}{h_{i,j}} + \frac{gh_{i,j}^2}{2}\right) - \left(\frac{U_{i-1,j}^2}{h_{i-1,j}} + \frac{gh_{i-1,j}^2}{2}\right)}{\Delta x} - \frac{\left(\frac{U_{i,j}V_{i,j}}{h_{i,j}}\right) - \left(\frac{U_{i,j-1}V_{i,j-1}}{h_{i,j-1}}\right)}{\Delta y} \\ -\frac{\left(\frac{V_{i,j}^2}{h_{i,j}} + \frac{gh_{i,j}^2}{2}\right) - \left(\frac{V_{i,j-1}^2}{h_{i,j-1}} + \frac{gh_{i,j-1}^2}{2}\right)}{\Delta y} - \frac{\left(\frac{U_{i,j}V_{i,j}}{h_{i,j}}\right) - \left(\frac{U_{i-1,j}V_{i-1,j}}{h_{i-1,j}}\right)}{\Delta x} \end{pmatrix}$$

При равенстве $\Delta x = \Delta y$, формулу можно сократить, вынеся их.

Интегрирование

Нам дано значение P в точке в конкретный момент времени и функция вычисления производной f. Необходимо получить значение в этой точке в следующий момент времени. Иными словами, нужен явный метод для решения задачи Коши.

Метод Эйлера

$y_i = y_{i-1} + (x_i - x_{i-1}) f(x_{i-1}, y_{i-1})$

Данный метод интегрирования достаточно прост, однако крайне неточен. Улучшить результат можно, используя новое значение в точке как предсказание и пересчитав его:

$y_i = y_{i-1} + (x_i - x_{i-1}) \frac{f(x_{i-1}, y_{i-1}) + f(x_i, \widetilde{y_i})}{2}$,

где $\widetilde{y_i}$ - предсказанное значение.

Для нашей задачи формулы примут вид:

$\tilde{P}_{t+\tau} = P_t + \tau f(P_t)$;

$P_{t+\tau} = P_t + \tau \frac{f(P_t) + f(\tilde{P}_{t+\tau})}{2}$,

где τ - шаг по времени.

Метод Рунге-Кутты

Метод Рунге – Кутты четвёртого порядка при вычислениях с постоянным шагом интегрирования столь широко распространён, что его часто называют просто методом Рунге – Кутты:

$y_{n+1} = y_n + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4);$
$k_1 = f(x_n +, y_n);$
$k_2 = f\left(x_n + \frac{h}{2}, y_n + \frac{h}{2}k_1\right);$
$k_3 = f\left(x_n + \frac{h}{2}, y_n + \frac{h}{2}k_2\right);$
$k_4 = f(x_n + h, y_n + hk_3),$
где h - шаг сетки по x.

Для нашей задачи формулы примут вид:
$P_{t+\tau} = P_t + \frac{\tau}{6}(k_1 + 2k_2 + 2k_3 + k_4);$
$k_1 = f(P_t);$
$k_2 = f\left(P_t + \frac{\tau}{2}k_1\right);$
$k_3 = f\left(P_t + \frac{\tau}{2}k_2\right);$
$k_4 = f(P_t + \tau k_3)$

Как можно заметить, для значений k (кроме первого) мы считаем производную в точке, которая на самом деле не существует. Так как ее вычисление подразумевает использование значений в соседних точках, то мы должны использовать для расчета значения несуществующих соседей.

Поэтому для нашей задачи нам необходимо для каждого параметра k создавать массив абстрактных точек, а после считать производную в каждой точке, в качестве соседей используя точки из того же массива.

Порядок расчета будет выглядеть подобным образом:

1. На основе массива P посчитать массив $k_1 = f(P)$
2. Создать массив абстрактных точек $P_2 = P + \frac{\tau}{2}k_1$
3. $k_2 = f(P_2)$
4. $P_3 = P + \frac{\tau}{2}k_2$
5. $k_3 = f(P_3)$
6. $P_4 = P + \tau k_3$
7. $k_4 = f(P_4)$
8. $P_{t+\tau} = P_t + \frac{\tau}{6}(k_1 + 2k_2 + 2k_3 + k_4)$

Настройка и тестирование

Даже правильно реализованный алгоритм не обязательно будет выдавать нужный результат. Описанные выше методы можно оценить по следующим качествам:

Точность. Играет минимальную роль, так как мы не знаем «правильного» поведения воды.

Сохранение энергии. Если она не сохраняется – вода будет затухать или расти в объеме, но подбор параметров позволит минимизировать негативное воздействие на симуляцию. Из названных методов только интегрирование Верле сохраняет энергию.

Устойчивость. Ключевое качество. Частный случай предыдущего, неустойчивая схема приведет к экспоненциальному росту энергии и, как следствие, «взрыву». Параметры должны быть подобраны очень тонко, так как такой быстрый рост проявляет себя за короткий промежуток времени. Метод Рунге-Кутты устойчив, но в определенных интервалах параметров.

Параметры

Упомянутые выше методы используют 5 констант, от которых зависит качество и скорость симуляции:

- Шаги сетки Δx и Δy. Соответствуют расстоянию между точками. Должны быть близки. В идеале – равны.
- Фазовая скорость v_p. Определяет скорость распространения волн при использовании волнового уравнения. Большие значения делают схему неустойчивой.
- Средняя высота H. В целом не оказывает сильного воздействия на симуляцию, если не выставить слишком малые или большие значения.
- Ускорение свободного падения g. Определяет скорость распространения волн при использовании уравнений мелкой жидкости. Большие значения делают схему неустойчивой.
- Шаг времени τ. Определяет скорость симуляции. Очень сильно влияет на устойчивость.

Зависимости величин:

- $v_p^2 \sim gH$. Позволяет сравнивать методы дифференцирования.
- $\tau \approx \frac{\Delta x}{v_p}$. Простой способ выбрать первое значение для шага времени.

Граничные условия

Как можно было заметить, использование метода конечных разностей приводит к тому, что для расчетов каждой точки необходимы значения величин в ее соседях. В результате точки на границах массива становятся особым случаем, так как у них нет всех соседей. Некоторые способы решения проблемы:

- Не проводить расчеты на границах, принимая значения величин и их производные равными 0. Волны станут отражаться от границ.
- Аналогично предыдущему, но точкам, стоящим рядом с граничными, необходимо устанавливать значения равными 0. Волны будут исчезать на границах (Не было проверено).
- Для всех расчетов изменить индексирование точек. Записи вида $i + 1$ заменить на $(i + 1)\%N$, а записи вида $i - 1$ на $(i - 1 + N)\%N$, где N – число точек по соответствующей оси, а % - операция нахождения остатка от деления. Подразумевается отчет от 0. Волны будут выходить с противоположного края массива.