

Московский государственный университет имени М.В. Ломоносова
Факультет вычислительной математики и кибернетики



ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

**Реализация программного модуля суммирования
сейсмической миграции с использованием графических
ускорителей**

Выполнил:

Хомидов Носирхуджса Хомидович

Научный руководитель:

д.ф.-м.н. Баев Андрей Владимирович

Москва, 2022

Содержание

1 Введение	2
2 Постановка задачи	4
2.1 Асимптотическое решение задач волнового продолжения и сейсмической миграции до суммирования	4
2.2 Получение функций Грина методом трассировок лучевых трубок	7
2.3 Математическая формулировка задачи построения изображения	10
3 Программная реализация на Cuda	12
4 Результаты вычислительных экспериментов и их анализ	15
5 Выводы	16

1 Введение

Актуальность темы

Под сейсморазведкой понимают метод геофизического исследования земной коры и его геологической среды, который основан на анализе естественных и искусственно возбужденных упругих волн и их поведение при встрече с горными породами. Таким образом, основной задачей можно назвать исследование сейсмической миграции, которая представляет из себя реконструкцию 3-мерного сейсмического изображения глубинных неоднородностей изучаемой земной среды. В итоге, мы получим 3-х мерное сейсмическое изображение, которое можно использовать совместно со скважинными данными в процессе их комплексной геологической интерпретации, результатом которой является цифровая геологическая модель месторождения. В настоящее время наиболее востребованной является 3-мерная глубинная миграция до суммирования, учитывающая преломление сейсмических волн в фоновой модели неоднородной земной среды. Однако объем исходных сейсмических, используемых для построения изображения может быть очень большим, вплоть до 10 ТБ цифровой информации, а наиболее экономичное приближенное решение задачи методом Кирхгофа требует порядка 10^{17} арифметических операций. Таким образом, задача сейсмической миграции является одной из самых сложных и востребованных научно-технических задач, для решения которой требуется использование суперкомпьютеров.

Обзор источников

В настоящее время известны два основных подхода к решению задачи сейсмической миграции. Первый основан на постановке и решения линеаризованной обратной задачи рассеяния волн, в то время как второй, представляющий основное практическое значение - на обращенном волновом продолжении. Важный вклад в разработку метода обращенного волнового продолжения в начале 70-х годов прошедшего века внесли американские исследователи Дж. Клаербоут и Дж. Газдак и отечественные ученые Г. И. Петрашень и С.А. Нахимкин. В результате своей деятельности они получили различные методы и алгоритмы. Так например, основанные на обращенном волновом продолжении методы и алгоритмы сейсмической миграции, связанные с

решением множества прямых 3-мерных задач на распространения волн от точечных источников, число которых может достигать порядка 10^5 . Если говорить про экономичность, то этому критерию отвечает асимптотический метод решения в предположении однолучевого распространения. Последнее означает, что из источника в произвольную глубинную точку среды приходит единственный луч по траектории, соответствующей кратчайшему времени пробега. Основанные на этом приближенные методы сейсмической миграции сводятся к явным интегральным формулам и называются миграцией методом Киргхгофа. В неоднородных сложнопостроенных средах предположение об однолучевом распространении часто нарушается. Кроме того, присутствие складок фронтов волн и областей их фокусировки, называемых каустиками, требует корректного учета, который недостижим в рамках однолучевой модели. Асимптотический метод разработанный отечественным ученым М.М. Поповым, известный под названием "суммирование гауссовых пучков до сих пор не вошел в число общеупотребительных средств сейсмической миграции.

Следуя общей схеме рассуждений, представленной в [4], рассмотрим задачу для дивергентного вида уравнения Гельмгольца, описывающего распространение монохроматических колебаний в неоднородной среде:

$$\operatorname{div}(k \operatorname{grad} u) + \omega^2 \rho u = 0 \quad (1)$$

Решение задачи волнового продолжения в неоднородное полупространство:

$$u(r, \omega) = -\frac{1}{4\pi} \iint_S u_0(r_0, \omega) k(r_0) \frac{\partial G(r_0|r; \omega)}{\partial n} dr_0 \quad (2)$$

где $u|_s = u_0$ - функция, задающая граничное условие Дирихле, S - граница неоднородного полупространства $z > 0$, ω - частота; $G(r_0|r; \omega)$ - функция Грина; $\partial/\partial n$ - производная по направлению внешней нормали к S ; $k = k(r)$ и $\rho = \rho(r)$ заданные параметры неоднородной среды, определяющие локальную скорость распространения колебаний $\nu(r) = \sqrt{k(r)/\rho(r)}$; dr_0 - элемент поверхности интегрирования S .

Введем обращенное время $\tilde{t} = T - t$ запишем в системе координат обращенного времени функцию $\tilde{u} = \tilde{u}(r_s, r_g, \tilde{t})$, где $u(r_s, r_g, t)$ описывает результаты регистрации волн произвольно расположенным в полупространстве $z > 0$ приемниками в точках $r_g = (x_g, y_g, z_g)$ от источника в произвольных точках $r_s = (x_s, y_s, z_s)$. Зафиксируем некоторый произвольный источник s . Используя граничное условие

$u_0 = u_0(s, g, \omega) = u(r_s, r_g, t)|_{z_s=z_g=0}$ можно применить интегральную формулу волнового продолжения (2) и получить результат обращенного волнового продолжения в нижнее полупространство $z_g > 0$:

$$u(s, r_g; \omega) = -\frac{1}{4\pi} \iint_{S_g} u_0(s, g; \omega) k(g) \frac{\partial G(g|r_g; \omega)}{\partial \mathbf{n}_s} dg \quad (3)$$

где dg - элемент площади интегрирования, n_g - внешняя нормаль к поверхности S_g . Полученное интегральное выражение представляет собой результат восстановления в произвольной точке r_g полупространства $z_g > 0$ волнового поля по его измеренным значениям на поверхности наблюдений S_g для произвольного фиксированного источника s .

Цель работы

Главной целью данной работы является создание программного модуля получения сейсмического изображения использованием графических ускорителей.

Задачи исследования

Для достижения цели решаются следующие задачи:

- Моделирование волн от точечных источников в неоднородной среде.
- Разработка алгоритма и написание программного модуля миграционного преобразования методом обращённого волнового продолжения.

2 Постановка задачи

2.1 Асимптотическое решение задач волнового продолжения и сейсмической миграции до суммирования

Согласно с [2] можно написать формальное асимптотическое приближение функции Грина из (2) в виде:

$$G(r_0|r_*; \omega) \approx \sum_m |A_m(r_0|r_*)| e^{i\omega\tau_m(r_0|r_*)} e^{iInd_m \operatorname{sgn}\omega\pi/2},$$

где $|A_m(r_0|r_*)| \exp(i\omega\tau_m(r_0|r_*)e^{iInd_m \operatorname{sgn}\omega\pi/2})$ - одно из слагаемых лучевой функции Грина, отвечающих непрерывному участку фронта волны – «лепестку» с номером m , связанному с пучком лучей, исходящих из глубинной точки r_* , с фиксированной

величиной накопленного индекса $Ind_m(r_0|r_*)$. $Ind_m(r_0|r_*)$ отвечает полному числу обращений в ноль якобиана $J(\mathbf{r}(\tau, \varphi, \theta) | \mathbf{r}_*) = (\partial\mathbf{r}/\partial\tau, \partial\mathbf{r}/\partial\varphi, \partial\mathbf{r}/\partial\theta)$ вдоль соответствующей лучевой трубки. $A_m(r_0|r_*)$ представляет собой лучевую амплитуду m -лепестка в точке r_0 выхода соответствующей лучевой трубки на поверхность наблюдений, имеющую вид:

$$A(\mathbf{r} | \mathbf{r}_*) = \frac{1}{k_*} \left(\frac{\nu_* \rho_* |\sin \theta|}{\nu \rho \left[\frac{J(\mathbf{r}|\mathbf{r}_*)}{\nu} \right]} \right)^{1/2}. \quad (4)$$

Отсюда можно получить:

$$\begin{aligned} \frac{\partial G(\mathbf{r}_0 | \mathbf{r}_*; \omega)}{\partial \mathbf{n}} &\approx \left(\sum_m i\omega |A_m(\mathbf{r}_0 | \mathbf{r}_*)| e^{i\omega\tau(\mathbf{r}_0|\mathbf{r}_*)} e^{i\text{Ind}_m \text{sgn } \omega\pi/2} \nabla_{\mathbf{r}_0} \tau_m(\mathbf{r}_0 | \mathbf{r}_*), \mathbf{n} \right) = \\ &= i\omega \sum_m |A_m(\mathbf{r}_0 | \mathbf{r}_*)| \frac{\cos \alpha_m}{v(\mathbf{r}_0)} e^{i\omega\tau_m(\mathbf{r}_0|\mathbf{r}_*)} e^{i\text{Ind}_m \text{sgn } \omega\pi/2} \end{aligned} \quad (5)$$

α_m - угол между направлением выхода луча, принадлежащего m -лепестку, в точке r_0 поверхности S и нормалью \mathbf{n} . Выполнив подстановку (5) в (2) в итоге можно получить:

$$\begin{aligned} u(\mathbf{r}_*; \omega) &\approx -\frac{1}{4\pi} \iint_S u_0 k_0 \sum_m i\omega |A_m(\mathbf{r}_0 | \mathbf{r}_*)| \frac{\cos \alpha_m}{\nu(\mathbf{r}_0)} e^{i\omega\tau_m(\mathbf{r}_0|\mathbf{r}_*)} e^{i\text{Ind}_m \text{sgn } \omega\pi/2} d\mathbf{r}_0 \approx \\ &\approx -\frac{1}{4\pi} \sum_m \iint_{S_m} u_0 k_0 i\omega |A_m(\mathbf{r}_0 | \mathbf{r}_*)| \frac{\cos \alpha_m}{\nu(\mathbf{r}_0)} e^{i\omega\tau_m(\mathbf{r}_0|\mathbf{r}_*)} e^{i\text{Ind}_m \text{sgn } \omega\pi/2} d\mathbf{r}_0 \end{aligned} \quad (6)$$

где $d\mathbf{r}_0$ - элемент площади поверхности S ; S_m - участок поверхности, связанный с m -лепестком многозначного эйконала $\tau_m(\mathbf{r}_0 | \mathbf{r}_*)$. Можно заметить, что входящая в (6) амплитуда $|A_m(\mathbf{r}_0 | \mathbf{r}_*)|$ в соответствии с (4) является неограниченной функцией в окрестности каустик.

Выполнив обратное преобразование Фурье по времени (6), можно получить решение задачи волнового продолжения во временной области:

$$u(\mathbf{r}_*, t) \approx -\frac{1}{4\pi} \sum_m \iint_{S_m} k_0 \left| \tilde{A}_m(\mathbf{r}_0 | \mathbf{r}_*) \right| \frac{\cos \alpha_m}{\nu(\mathbf{r}_0)} H^{\text{Ind}_m}(u_0)'_t(x_0, y_0; t + \tau_m(\mathbf{r}_0 | \mathbf{r}_*)) dx_0 dy_0 \quad (7)$$

Асимптотического решения задачи 3D-миграции до суммирования данных многократных перекрытий:

$$f(\mathbf{r}_*) \approx \left(\frac{1}{4\pi} \right)^2 \sum_{l, l'} \int_{S_l} \int_{S_{l'}} k_s k_g \frac{\cos(\alpha_s)_l \cos(\alpha_g)_{l'}}{\nu_s \nu_g} [A_s]_l [A_g]_{l'} H^{\text{Ind}_l + \text{Ind}_{l'}}(u_0)'_t(\mathbf{s}, \mathbf{g}; (\tau_s)_l + (\tau_g)_{l'}) ds dg \quad (8)$$

где S_l и $S_{l'}$ - участки площади поверхности, связанные с лепестками l, l' и отвечающими им участками фронта волны с постоянным индексом для точечного источника, расположенного в \mathbf{r}_* . Суммирование выполняется по всем возможным сочетаниям лепестков для источников и приемников. $f(\mathbf{r}_*)$ - искомое сейсмическое изображение в глубинной точке \mathbf{r}_* ; $u_0(\mathbf{s}, \mathbf{g}, t)$ - исходные данные многократных перекрытий для

источников и приемников с координатами (\mathbf{s}, \mathbf{g}) ; $\tau_s + \tau_g$ – сумма времен пробега из \mathbf{r}_* до источников и приемников; ν и k – заданные параметры среды, входящие в волновое уравнение (1); H – преобразование Гильберта; $[A_s]$ и $[A_g]$ величины лучевых амплитуд для источников и приемников.

Во внутреннем 4-кратном интеграле (8) по координатам источников и приемников можно так структурировать исходные данные, чтобы обеспечить миграционное преобразование в сейсмическое изображение однократных сейсмических кубов квазиравных удалений/азимутов направлений источники - приемники. С этой целью авторы ввели «биноевые» координаты средних точек и половинных удалений: $\mathbf{m} = (\mathbf{s} + \mathbf{g})/2$, $\mathbf{h} = (\mathbf{s} - \mathbf{g})/2$. Средние точки $\mathbf{m} = \mathbf{m}(m_x, m_y)$ образуют плотную и регулярную биновую сетку. Можно назвать «покрытием» совокупность исходных сейсмических трасс, однократно заполняющих узлы этой биновой сетки для некоторого фиксированного вектора \mathbf{h} (h_x, h_y). С учетом сказанного можно переписать аппроксимацию (8):

$$\begin{aligned}
f(\mathbf{r}_*) &\approx \left(\frac{1}{4\pi}\right)^2 \sum_{l,l'} \int_{S_l} \int_{S_{l'}} k_s k_g \frac{\cos(\alpha_s)_l \cos(\alpha_g)_{l'}}{\nu_s \nu_g} [A_s]_l [A_g]_{l'} H^{\text{Ind}_l + \text{Ind}_{l'}} (u_0)'_t(\mathbf{s}, \mathbf{g}; (\tau_s)_l + \\
&\quad + (\tau_g)_{l'}) d\mathbf{s} d\mathbf{g} = \\
&= \left(\frac{1}{2\pi}\right)^2 \sum_{l,l'} \int_{S_l} \int_{S_{l'}} k_s k_g \frac{\cos(\alpha_s)_l \cos(\alpha_g)_{l'}}{\nu_s \nu_g} [A_s]_l [A_g]_{l'} H^{\text{Ind}_l + \text{Ind}_{l'}} (u_0)'_t(\mathbf{m} + \\
&\quad + \mathbf{h}, \mathbf{m} - \mathbf{h}; (\tau_s)_l + (\tau_g)_{l'}) d\mathbf{m} d\mathbf{h} = \\
&= \left(\frac{1}{2\pi}\right)^2 \int d\mathbf{h} \left(\sum_{l,l'} \int_{S_l} k_s k_g \frac{\cos(\alpha_s)_l \cos(\alpha_g)_{l'}}{\nu_s \nu_g} [A_s]_l [A_g]_{l'} H^{\text{Ind}_l + \text{Ind}_{l'}} (u_0)'_t(\mathbf{m} + \right. \\
&\quad \left. + \mathbf{h}, \mathbf{m} - \mathbf{h}; (\tau_s)_l + (\tau_g)_{l'}) d\mathbf{m} \right) \tag{9}
\end{aligned}$$

Итоговое глубинное изображение $f(\vec{r}_*)$ представляется набором изображений $f(\vec{r}_*, \vec{h})$ результатов миграции фиксированного покрытия $\mathbf{h} = \text{const}$. Выражение, стоящее в круглых скобках, представляет собой результат миграции фиксированного покрытия $\mathbf{h} = \text{const}$, а внешний интеграл по \mathbf{h} превращается в обычную сумму по покрытиям.

Следуя известным представлениям о сейсмической миграции как об операторе, переводящем однократный сейсмический куб исходных данных в глубинный куб сейсмического изображения, была реализована возможность в качестве исходного куба данных подавать на вход многолучевой миграции однократные кубы квазиравных

удалений/азимутов направлений источники – приемники. Число таких кубов, на которые заранее рассортируются исходные сейсмические данные, отвечает номинальной кратности съемки. После выполнения миграционного преобразования каждого исходного куба в глубинный куб сейсмического изображения выполняется сборка сейсмограмм общей точки изображения, далее – их осреднение с подбором мьютинга и получение суммарного куба сейсмического изображения.

2.2 Получение функций Грина методом трассировок лучевых трубок

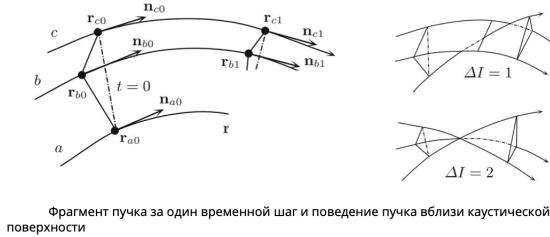
Согласно [11] модифицированная система Гамильтона в неоднородной изотропной среде с распределением скоростей $v = v(r)$ параметризуется временем пробега τ . Его дают:

$$\begin{aligned}\dot{\mathbf{r}} &= v^2 \mathbf{p} \\ \dot{\mathbf{p}} &= \frac{v^2}{2} \nabla \left(\frac{1}{v^2} \right)\end{aligned}$$

Где r - положение луча, а p направлен ортогонально фронту волны $\tau = \text{const.}$ в изотропной среде это направление касается траектории луча. Абсолютное значение вектора $|p|$ обратна скорости в точке. Система ОДУ решается методом Рунге-Кутты 4-го порядка. Абсолютное значение p сохраняется правильным путем перенормировки вектора к $1/v$ на каждом шаге метода. Трассировка лучей выполняется в объеме, ограниченном сверху конусом, вершина которого находится в источнике лучей. В боковом направлении он ограничен прямоугольником, размер которого определяется максимальным расстоянием между источником и приемником плюс апертурой (которая обычно составляет $4 \div 5$ км). Глубина трассировки лучей в целом не превышает 10 км. Переворот ребер позволяет аппроксимировать сложные сильно растянутые части волнового фронта с помощью относительно небольшого числа треугольников. Это происходит за счет заметных сложностей в алгоритмах поддержания сетки. Давайте вспомним, что для каждого луча мы локально имеем нормаль к фронту волны. Если наше приближение хорошее, то оно также близко к нормали треугольника. Следовательно, мы предполагаем, что переворот ребер улучшает триангуляцию, если измененные треугольники становятся близкими к интерполированным нормалям

в их середине.

Траектория треугольника с лучами в каждой вершине образует луч. Для каждого временного шага трассировки лучей волновой фронт состоит из фрагментов луча, нарезанных интервалом временного шага. Любой фрагмент образован двумя треугольными срезами луча постоянными по времени поверхностями и линиями, соединяющими соответствующие вершины (рис.).



Фрагмент пучка за один временной шаг и поведение пучка вблизи каустической поверхности

Для вычислении функции грина необходимо найти все узлы декартовой сетки, попадающий в фрагмент. Для каждого узла нужно получить время, когда фронт касается точки, индекс КМАН (число изменений фаз волны при прохождении волнового фронта через каустику), усредненную амплитуду луча, направление луча, приходящего в точку, и ширину усреднения по времени сейсмических данных. Для краткости процедуру получения данных можно назвать "захват" данных.

Во время β -трассировки отслеживаются все лучи, сохраненные в истории α -трассировки. Также ведется учет всех лучей независимо от временного шага, с которым они были созданы в α -трассировке. Якобианы и индексы КМАН вычисляются для всех лучей, но процедура захвата вызывается только для тех, которые присутствовали на этом временном шаге в α -трассировке.

Для луча используются обозначения, в которых индексы a , b и c соответствуют лучам (рис.). Индекс 0 зарезервирован для значений до временного шага, а индекс 1 - для значений после временного шага. Таким образом, для временного шага соответствующий фрагмент луча имеет начальные координаты луча r_{a0} , r_{b0} и r_{c0} . Конечными координатами лучей для шага являются r_{a1} , r_{b1} , r_{c1} . Начальным и конечным направлениями распространения лучей являются n_{a0} , n_{b0} , n_{c0} и n_{a1} , n_{b1} , n_{c1} соответственно. Если ввести параметр $t \in [0, 1]$, где $t = 0$ связано с началом шага, конец $t = 1$ связан с его концом. Тогда можно ввести линейную аппроксимацию

$$\mathbf{r}(t) = \mathbf{r}_0 + (\mathbf{r}_1 - \mathbf{r}_0)t, \quad \mathbf{n}(t) = \mathbf{n}_0 + (\mathbf{n}_1 - \mathbf{n}_0)t.$$

Якобиан из n, t к t преобразуется следующим образом:

$$\mathcal{J}(t) = \frac{J}{\sin \theta} \approx \frac{1}{\Delta \Omega_0} ((\mathbf{r}_a - \mathbf{r}_b), (\mathbf{r}_b - \mathbf{r}_c), \langle \mathbf{n} \rangle) \equiv P_0 + P_1 t + P_2 t^2 + P_3 t^3,$$

где $\Delta \Omega_0$ начальная площадь поперечного сечения луча, $\langle \mathbf{n} \rangle = (\mathbf{n}_a + \mathbf{n}_b + \mathbf{n}_c)/3$.

Имея приближение Якоби по времени можно интерполировать его значения по фронту волны. Для этого интерполируются коэффициенты полинома в сетку лучей. После этого полученный якобиан на лучах можно линейно интерполировать в любую точку луча.

В процедуре захвата необходимо найти точки декартовой сетки, лежащие внутри луча, и рассчитать трек для каждой найденной точки. Считается, что точка принадлежит внутренности луча, если она попадает в какое-либо одно треугольное сечение при некотором значении $t \in [0, 1]$.

Амплитуда луча вычисляется по формуле

$$A = \frac{1}{v_0} \sqrt{\frac{v_r \sin \theta}{|J|}}, \quad J = \left(\frac{\partial \mathbf{r}}{\partial \varphi}, \frac{\partial \mathbf{r}}{\partial \theta}, \frac{\partial \mathbf{r}}{\partial \tau} \right)$$

Данная формула может содержать интегрируемую сингулярность, которая интегрируется по бину. Для получения точного результата якобиан аппроксимируется как

$$\mathcal{J} \approx \mathcal{J}_0 + \mathcal{J}_1 \xi, \quad \mathcal{J}_0 \geq 0, \quad \mathcal{J}_1 \geq 0,$$

где ξ — координата поверхности, направленная по градиенту якобиана, вычисляемому вдоль поверхности, значение \mathcal{J}_0 — значение якобиана в точке глубины для луча, соединяющего ее с источником.

Следующая формула используется для получения интеграла амплитуды луча по бину:

$$\langle A \rangle_i = \Delta \frac{\cos \theta_s}{\sqrt{v_s}} \int_{-\frac{\Delta}{2}}^{\frac{\Delta}{2}} \frac{\mathcal{H} \left([\mathcal{J}_0 + \mathcal{J}_1 \xi]^{-\frac{1}{2}} \right)}{[\mathcal{J}_0 + \mathcal{J}_1 \xi]^{-\frac{1}{2}}} d\xi = \frac{\Delta \cos \theta_s}{\mathcal{J}_1 \sqrt{v_s} \frac{1}{2}} \left. (\mathcal{J}_0 + \mathcal{J}_1 \xi)^{\frac{1}{2}} \right|_{\xi_{\min}}^{\frac{\Delta}{2}},$$

где \mathcal{H} ступенчатая ступенчатая функция Хевисайда, $\Delta = \sqrt{S_i}$ линейный размер бина и ξ_{\min} положение поверхности каустики,

$$\xi_{\min} = \max \left(-\frac{\Delta}{2}, -\frac{\mathcal{J}_0}{\mathcal{J}_1} \right) = -\min \left(\frac{\Delta}{2}, \frac{\mathcal{J}_0}{\mathcal{J}_1} \right)$$

2.3 Математическая формулировка задачи построения изображения

Функция Грина от приемника g или источника s задана в узлах равномерной прямоугольной сетки $\{r_i^G\}$, с шагами $\Delta_x^G, \Delta_y^G, \Delta_z^G$. Изображение вычисляется в узлах равномерной прямоугольной сетки $\{r_i^I\}$ с шагами $\Delta_x^I, \Delta_y^I, \Delta_z^I$, совпадающей с сеткой задания функции Грина по латерали, и измельченной (по сравнению с сеткой задания функции Грина) по вертикали в n_Z раз, $n_Z \geq 1$, таким образом размер сетки изображения по вертикали N_z^I

$$N_z^I = (N_z^G - 1)n_Z + 1, \quad \Delta_{x,y}^I = \Delta_{x,y}^G, \quad \Delta_z^I = \Delta_z^G / n_Z,$$

где N_z^G — размер сетки функции Грина по вертикали. Для вычисления изображения в произвольном узле с номером по глубине z^I требуются функции Грина от приемника G^g и источника G^s в соседних по вертикали узлах сверху $z_u^G = z^I/n_Z$ и снизу $z_d^G = z^I/n_Z + 1$, второй узел нужен только если $z^I n_Z \neq z_u^G$.

В каждом узле r^G может существовать несколько треков (возможностей прохода сигнала из источника/приемника), характеризующихся амплитудой A , временем прохождения τ , индексом I и параметрами окна фильтрации по времени

$$\Delta^\tau = (\Delta^{\tau x}, \Delta^{\tau y}) = \left(\frac{\Delta_{\text{bin}x} n_{0x}}{v_0}, \frac{\Delta_{\text{bin}y} n_{0y}}{v_0} \right),$$

где $\Delta_{\text{bin}x,y}$ — размеры площадки источника, $n_{0x,y}$ — компоненты направления выхода луча из источника, v_0 — скорость в источнике в направлении луча.

Восстановление функции Грина на сетке изображения производится по формуле

$$\begin{aligned} G = w_u \sum_i^{N_u} A_{u,i} \widehat{T}(|\Delta_{u,i}^\tau|) \widehat{H}(I_{u,i}) u_0 (\tau_{u,i} + \Delta_u n_{zu}) + \\ + w_d \sum_i^{N_d} A_{d,i} \widehat{T}(|\Delta_{d,i}^\tau|) \widehat{H}(I_{d,i}) u_0 (\tau_{d,i} - \Delta_d n_{zd}) \end{aligned} \quad (10)$$

где $w_u = (z_d - z) / \Delta_z^G$, $w_d = 1 - w_u$ — веса линейной интерполяции по вертикали, $N_{u,d}$ — число треков в верхнем и нижнем узлах, $\widehat{T}(\Delta^\tau)$ — оператор фильтрации сигнала в окне ширины Δ^τ , $\widehat{H}(I)$ — оператор гильберт–преобразования с индексом I , $n_{zu,d}$ —

z -компоненты направления распространения лучей в соотв. точках глубинной сетки (косинусы углов **прихода** относительно вертикали), $\Delta_{u,d}$ — задержки сигнала,

$$\Delta_u = \frac{2w_d \Delta_z^G}{v_u + v_c}, \quad \Delta_d = \frac{2w_u \Delta_z^G}{v_d + v_c}, \quad v_c = w_d v_u + w_u v_d,$$

$v_{u,d}$ — скорости в верхнем и нижнем узлах. При суммировании используется выражение

$$\begin{aligned} S &= w_u \sum_i^{N_u^s} \sum_j^{N_u^g} A_{u,i}^s A_{u,j}^g \widehat{T}(|\Delta_{u,i}^{\tau_s} + \Delta_{u,j}^{\tau_g}|) \widehat{H}(I_{u,i}^s + I_{u,j}^g) \\ &u_0 (\tau_{u,i}^s + \tau_{u,j}^g + (n_{zu,i} + n_{zu,j}) \Delta_u) + w_d \sum_i^{N_d^s} \sum_j^{N_d^g} A_{d,i}^s A_{d,j}^g \widehat{T}(2|\Delta_{d,i}^{\tau_s} + \Delta_{d,j}^{\tau_g}|) \widehat{H}(I_{d,i}^s + I_{d,j}^g) \\ &u_0 (\tau_{d,i}^s + \tau_{d,j}^g - (n_{zd,i} + n_{zd,j}) \Delta_d). \end{aligned} \quad (11)$$

Рассмотрим построение гильберт-преобразования и фильтрации в скользящем окне. Пусть исходная сейсмическая трасса $u_0(t)$ представлена в виде кусочно-постоянной функции $\{u_{0i}\}$ с шагом дискретизации Δ_t :

$$u_0(t) = u_{0i} \quad \text{при } (i-1)\Delta_t \leq t < i\Delta_t.$$

Гильберт-преобразование $\{g_i\}$ для индекса единица можно записать как свертку с дискретным антисимметричным ядром H_i полуширины $N_H = 20$:

$$g_i = \sum_{k=-N_H}^{N_H} u_{0i+k} H_k, \quad H_{-k} = -H_k, \quad H_0 = 0,$$

$H_1 = 0.430$	$H_2 = 0.162$	$H_3 = 0.098$	$H_4 = 0.068$	$H_5 = 0.051$
$H_6 = 0.039$	$H_7 = 0.031$	$H_8 = 0.025$	$H_9 = 0.021$	$H_{10} = 0.017$
$H_{11} = 0.014$	$H_{12} = 0.011$	$H_{13} = 0.009$	$H_{14} = 0.007$	$H_{15} = 0.006$
$H_{16} = 0.004$	$H_{17} = 0.003$	$H_{18} = 0.002$	$H_{19} = 0.001$	$H_{20} = 0.000$

При расчете Гильберт-преобразования для произвольного индекса I используется выражение:

$$\widehat{H}(I)u_{0i} = \begin{cases} u_{0i} & \text{при } \mod(I, 4) = 0, \\ g_i & \text{при } \mod(I, 4) = 1, \\ -u_{0i} & \text{при } \mod(I, 4) = 2, \\ -g_i & \text{при } \mod(I, 4) = 3, \end{cases}$$

где \mod — остаток от деления.

Для ускорения оператора фильтрации в скользящем окне $\widehat{T}(\Delta^\tau)$ перейдем от

кусочно–постоянных u_0 и g к их линейно–интерполированным первообразным F, G :

$$F(t) = (1 - q)F_{i-1} + qF_i, \quad q = t/\Delta_t - (i - 1), \quad \text{при } (i - 1)\Delta_t \leq t < i\Delta_t,$$

$$G(t) = (1 - q)G_{i-1} + qG_i, \quad q = t/\Delta_t - (i - 1), \quad \text{при } (i - 1)\Delta_t \leq t < i\Delta_t,$$

$$F_i = \sum_{k=1}^i f_k, \quad G_i = \sum_{k=1}^i g_k.$$

Тогда

$$\widehat{T}(\Delta^\tau) F(t) = \begin{cases} \frac{F(t+\Delta^\tau) - F(t-\Delta^\tau)}{2\Delta^\tau} & \text{при } \text{floor}\left(\frac{t+\Delta^\tau}{\Delta_t}\right) \neq \text{floor}\left(\frac{t-\Delta^\tau}{\Delta_t}\right) \\ \frac{F_i - F_{i-1}}{\Delta_t} & \text{при } i = \text{floor}\left(\frac{t+\Delta^\tau}{\Delta_t}\right) = \text{floor}\left(\frac{t-\Delta^\tau}{\Delta_t}\right) \end{cases}$$

3 Программная реализация на Cuda

Программа была реализована на языке Cuda версии 11.2.152. Для сохранения, загрузки и дальнейшей работы с данными была написана структура `aiv_arr` в которой данные хранятся в массиве:

```
template<class Tffloat, int dim>
struct aiv_arr {
    array<int, dim> size;
    Tffloat *data;
public:
    aiv_arr() {}
    aiv_arr(array<int, dim> N): size(N) {
        reset();
    }
    long int size_arr;
    void reset() {
        long int sz=size[0];
        for(int d=1; d<dim; d++) sz *= size[d];
        cudaMallocManaged(&data, sz*sizeof(Tffloat));
        size_arr = sz;
    }
}
```

```

~aiv_arr(){
    cudaFree(data);
}

__device__ __host__ Tffloat& operator[](array<int, dim> I) {
    long int sh=I[dim-1];
    for(int d=dim-2; d>=0; d--) sh = I[d]+size[d]*sh;
    return data[sh];
}

```

Для выделения памяти используется функция **cudaMallocManaged**, для освобождения - **cudaFree**. Так как данные мы получаем в виде файлов, необходимо написать функцию, которая читает эти файлы и корректно их сохраняется в программе:

```

void load(const char* fn) {
    int rs=0;
    try {
        FILE* pFile=fopen(fn, "r");
        if(pFile == NULL) throw("При открытии файла возникли проблемы.
        Возможно, его нет или он закрыт на чтение.");
        int head_len, D, szT, N[dim];
        rs=fread(&head_len, 4, 1, pFile); if(rs != 1) throw "head_len";
        if(head_len<0 || head_len>4096) throw "head_len too small or big";
        if(head_len>0) {
            char head[head_len];
            rs=fread(head, 1, head_len, pFile); if(rs != head_len) throw "head";
        }
        rs=fread(&D, 4, 1, pFile); if(rs != 1 || D!=dim) throw "D";
        rs=fread(&szT, 4, 1, pFile); if(rs != 1 || szT!=sizeof(Tffloat)) throw "szT";
        rs=fread(N, 4, D, pFile); if(rs != D) throw "N";
        for(int d=0; d<dim; d++) size[d] = N[d];
        reset();
        rs=fread(&data[0], szT, size_arr, pFile);
    }
}

```

```

    if(rs != size_arr) throw "data";
    fclose(pFile);
} catch(const char* err) {
    printf("Error on load <%s>: %s, rs=%d\n", fn, err, rs);
    throw(-1);
} catch(...) {
    printf("Some error on drop <%s>, rs=%d\n", fn, rs);
    throw(-1);
}
}

```

Итоговое изображение вычисляется на ядре Cuda и представляется в виде структуры aiv_arr. Для отрисовки изображения была выбрана программа VirtualGL 3.0. Для сохранения изображения и дальнейшей отрисовки была написана функция, которая сохраняет данные из структуры aiv_arr в файл:

```

void drop(const char* fn) {
    int rs=0;
    try {
        FILE* pFile=fopen(fn, "w");
        if(pFile == NULL)
            throw("При открытии файла возникли проблемы");
        int head_len=0, D=dim, szT=sizeof(Tfloat), N[dim];
        for(int d=0; d<dim; d++) N[d] = size[d];

        int rs=fwrite(&head_len, 4, 1, pFile); if(rs != 1) throw "head_len";
        rs=fwrite(&D, 4, 1, pFile); if(rs != 1) throw "dim";
        rs=fwrite(&szT, 4, 1, pFile); if(rs != 1) throw "szT";
        rs=fwrite(N, 4, D, pFile); if(rs != D) throw "N";
        rs=fwrite(&data[0], szT, size_arr, pFile); if(rs != size_arr) throw "data";
        fclose(pFile);
    } catch(const char* err) {

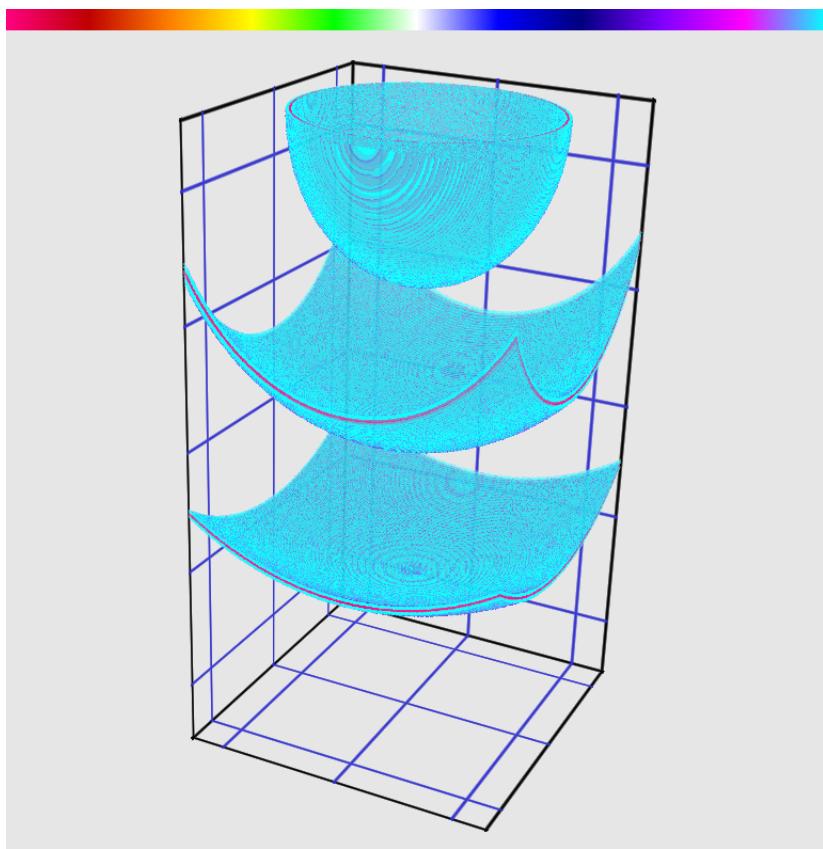
```

```

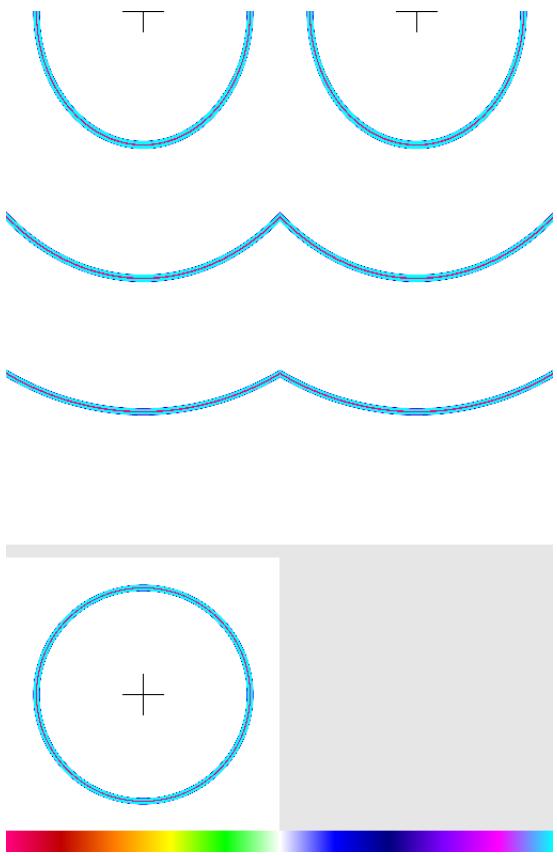
printf("Error on drop <%s>: %s, rs=%d\n", fn, err, rs);
throw(-1);
} catch(...) {
    printf("Some error on drop <%s>, rs=%d\n", fn, rs);
    throw(-1);
}
}

```

4 Результаты вычислительных экспериментов и их анализ



На фото изображено возможное положение отражателей в глубине среды. Итоговое изображение среды как раз получается суммированием множества таких потенциальных положений.



Сечение изображения возможного положения отражателей.

5 Выводы

В процессе написания выпускной квалификационной работы мной было изучены основы теории многолучевой сейсмической миграции, процедуры трассировок, процедура получении сейсмического изображения. Так же были разработаны структуры данных и алгоритмы необходимые для реализации процедуры суммирования на GPU. В конечном итоге удалось создать программный модуль суммирования сейсмической миграции, который был протестирован на представленном наборе сейсмических данных.

Список литературы

- [1] Гогоненков Г.Н., Мороз Б.П., Плешкевич А.Л., Турчанинов В.И. Теоретические основы и практическое использование отечественной программы 3D-глубинной сейсмической миграции до суммирования. Геофизика. 2007. № 4. С. 15–24.
- [2] Плешкевич А.Л., Иванов А.В., Левченко В.Д., Хилков С. А. Многолучевая 3D-глубинная сейсмическая миграция до суммирования с сохранением амплитуд Геофизика. 2017. № S (спец. выпуск). С. 76–84.
- [3] Плешкевич А.Л. Методы реконструкции изображения глубинных неоднородностей земной среды по сейсмическим данным («сейсмическая миграция»): дис. . . . доктор. ф.-м. наук: 1.6.9 – Геофизика / Плешкевич А.Л. – НСК., 2021. – 216 с.
- [4] Морс Ф. М., Фешбах Г. Методы теоретической физики, т.1 – М.: ИЛ, 1958. – 931 с.
- [5] Боганик Г.Н., Гурвич И.И. Сейсмическая разведка – Тверь, АИС 2006 г., – 744 с.
- [6] Клаербоут Д. Ф. Сейсмическое изображение земных недр – М.: Недра 1989 г. – 407 с.
- [7] Воскресенский Ю.Н. Построение сейсмических изображений: Учебное пособие. – М.: РГУ нефти и газа им. И.М. Губкина, 2006. - 116 с.
- [8] Вайнберг, Б.Р. Асимптотические методы в уравнениях математической физики – М.: Изд-во Моск. ун-та, 1982. – 296 с.
- [9] Бронштейн И.Н., Семенджиев К.А. Справочник по математике для инженеров и учащихся втузов – М.: Наука, 1986. – 544 с.
- [10] Козлов Е. А. Миграционные преобразования в сейсморазведке – М.: Недра, 1986. – 247 с.
- [11] Pleshkevich, A., Ivanov, A., Levchenko, V., Khilkov, S. (2020). Kirchhoff-Type Implementation of Multi-Arrival 3-D Seismic Depth Migration with Amplitudes Preserved. In: Voevodin, V., Sobolev, S. (eds) Supercomputing. RuSCDays 2020. Communications in Computer and Information Science, vol 1331. Springer, Cham.