

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ**

**Федеральное государственное автономное образовательное учреждение
высшего образования
«Национальный исследовательский
Нижегородский государственный университет им. Н.И. Лобачевского»**

Радиофизический факультет

Направление 03.03.03 «Радиофизика»
Профиль «Фундаментальная радиофизика»

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

ЧИСЛЕННОЕ МОДЕЛИРОВАНИЕ МОРСКОЙ ПОВЕРХНОСТИ

«К защите допущен»:

зав. кафедрой общей физики,
д.ф.-м.н., профессор

_____ Бакунов М.И.

Научный руководитель:
д.ф.-м.н., профессор

_____ Бакунов М.И.

Рецензент:
должность,
ученая степень, звание

_____ ФИО

Консультант по технике безопасности:
ст. преп. кафедры акустики, к.ф.-м.н.

_____ Дерябин М.С.

Студент 4-го курса

_____ Понур К.А.

Нижний Новгород
2020 год

Содержание

1 Введение	2
2 Численное моделирование морского волнения	4
2.1 Общие сведения	4
2.2 Двумерная модель поверхностного волнения	6
2.3 Метод «отбеливания» спектра	13
2.3.1 Метод «отбеливания» спектра для одной переменной . .	13
2.3.2 Метод «отбеливания» спектра для двух переменных . .	20
2.3.3 Аппаратное ускорение моделирования	21
2.4 Заостренная морская поверхность	22
2.4.1 Двумерный случай	23
2.4.2 Трехмерный случай	26
3 Моделирование отраженного импульса	29
3.1 Схема измерения	30
3.2 Влияние морского волнения на форму отраженного импульса .	31
3.3 Аналитическая модель для формы формы отраженного импульса	36
3.4 Восстановление параметров морской поверхности.	42
3.5 Восстановление параметров модельных поверхностей	46
3.6 Восстановление параметров морской поверхности поверхности по данным радиовысотомера	48
4 Заключение	52
А Программный код	55

1. Введение

Моделирование морской поверхности является интересной научной задачей, которая привлекает к себе внимание ученых на протяжении длительного времени. Несмотря на значительный прогресс, остается много вопросов, которые требуют дальнейших исследований. Кратко рассмотрим основные используемые подходы и подробно обсудим способ моделирования, который будет использоваться в данной работе. Для описания поверхностного волнения применяют уравнения гидродинамики и в общем виде задача пока не по силам современной вычислительной технике. Благодаря упрощениям и предположениям задача становится «счетной», но всё равно требует много вычислительных ресурсов, поэтому этот подход используется для решения научно-исследовательских задач.

В настоящее время активно применяются радиофизические методы дистанционного зондирования для решения прикладных задач, например, для измерения скорости, ветра, высоты значительного волнения, температуры воды, диагностики разливов нефти и др. Несмотря на значительные успехи существующая измерительная аппаратура не всегда позволяет получить достаточно полное представление о состоянии приповерхностного слоя океана, поэтому постоянно разрабатываются новые радиолокационные системы.

Разработка новой измерительной аппаратуры дистанционного зондирования для космических носителей является сложной научно-технической задачей решение которой требует проведения полного комплекса исследований, включающего теоретический анализ, численное моделирование и эксперимент. В ходе решения ищется оптимальная схема измерения, анализируются требования к измерительной аппаратуре и оценивается эффективность алгоритмов обработки. Проведение численного эксперимента является необходимым этапом, поскольку позволяет максимально быстро проанализировать

разные варианты и предложить схему измерения и состав измерительного комплекса для этапа экспериментальных работ.

В данной работе, кроме моделирования морской поверхности, будет проведен численный эксперимент с орбитальным радиолокатором в целях не только проверить качество моделируемой поверхности, но и оценить точность алгоритмов восстановления данных, которые обычно используются при обработки данных радиовысотомера.

Мы последовательно рассмотрим этапы такого численного эксперимента, начиная с моделирования морского волнения и заканчивая тестированием алгоритмов восстановления параметров волнения по импульсу, отраженному от моделируемой поверхности.

2. Численное моделирование морского волнения

Для выполнения вычислений в режиме реального времени применяются мощные компьютеры. Это позволяет прогнозировать развитие морского волнения и результаты активно используют океанологи и метеорологи. Однако за это приходится «платить» низким пространственным разрешением, упрощением исходных уравнений и моделированием только длинно-волновой составляющей спектра волнения. Для оценки эффективности работы радиолокационной аппаратуры больше подходит хорошо известный подход, опирающийся на модель спектра волнения. В этом случае морская поверхность представляется в виде набора синусоид (гармоник), амплитуда которых вычисляется по спектру волнения[1], [2]. При таком подходе смоделированная морская поверхность утрачивает ряд свойств, присущих реальной морской поверхности, но становится более удобной для счета и моделирование может быть проведено на домашнем компьютере. Именно этот подход выбран для моделирования морской поверхности в данном исследовании. Надо отметить, что для выбранного подхода качество моделирования зависит от используемого спектра волнения и от численной реализации процедуры моделирования.

Предполагается, что гармоники не взаимодействуют друг с другом, поэтому возвышения поверхности, орбитальные скорости, уклоны и другие характеристики волнения являются суммой независимых гармоник.

2.1. Общие сведения

Определим ряд общих понятий, описывающих возвышение взволнованной морской поверхности в рамках теории случайных пространственно-временных

полей. В этом случае поверхность представляется в виде суммы синусоидальных волн со случайными фазами

$$\xi(\vec{r}, t) = \sum_{n=-\infty}^{\infty} A_n(\vec{\kappa}_n) e^{i(\omega_n t + \vec{\kappa}_n \vec{r} + \psi_n)}, \quad (1)$$

где ψ_n – случайная фаза, равномерно распределенная в интервале от 0 до 2π , $A_n(\vec{\kappa}_n)$ – комплексная амплитуда гармоники с волновым числом $\vec{\kappa}_n$ и временной частотой ω_n , связанной с $\vec{\kappa}_n$ известным дисперсионным соотношением

$$\omega(\kappa) = \sqrt{\kappa g + \alpha \kappa^3},$$

где g – ускорение свободного падения, α – коэффициент, зависящий от свойств жидкости.

Корреляционную функцию $K_\xi(\vec{r}, t)$ поля $\xi(\vec{r}, t)$ определим стандартным образом [3]

$$K_\xi[\vec{r}_1, \vec{r}_2, t_1, t_2] = \langle \xi(\vec{r}_1, t_1) \xi^*(\vec{r}_2, t_2) \rangle \quad (2)$$

Поле высот в нашей задаче считаем стационарным в широком смысле, то есть $K_\xi[\vec{r}_1, \vec{r}_2, t_1, t_2] = K_\xi[\vec{\rho} = \vec{r}_2 - \vec{r}_1, \tau = t_2 - t_1]$. Будем считать, гармоники независимыми друг от друга, а значит перекрестные члены в уравнении (2) равны нулю.

Тогда корреляционная функция поверхности (1) примет следующий вид

$$K_\xi[\vec{\rho}, \tau] = \sum_{n=-\infty}^{\infty} \frac{A_n^2}{2} \exp\{i(\vec{\kappa}_n \vec{\rho} + \omega \tau)\} \quad (3)$$

Для решения задачи моделирования отраженного от морской поверхности импульса достаточно рассматривать мгновенный снимок моделируемой поверхности, в момент отражения а значит можно положить $\tau = \text{const} = 0$ и тогда $K_\xi[\rho, \tau] = K_\xi[\rho]$.

В этом случае справедлива формула Винера-Хинчина [3]

$$\begin{aligned} S_\xi(\vec{k}) & \int_{-\infty}^{\infty} K_\xi[\rho] \exp\{-i\vec{\kappa}\vec{\rho}\} d\rho \\ K_\xi[\rho] & = \frac{1}{2\pi} \int_{-\infty}^{\infty} S_\xi(\vec{k}) \exp\{+i\vec{\kappa}\vec{\rho}\} d\rho. \end{aligned} \quad (4)$$

Предположим, что спектр морского волнения можно представить в виде функции с разделяющимися переменными, где $S_\xi(\kappa)$ определяет зависимость спектральной плотности мощности от волнового числа, а функция $\Phi(\kappa, \varphi)$ – описывает зависимость спектральной плотности мощности от азимутального угла для выбранного волнового числа

$$S_\xi(\vec{\kappa}) = S_\xi(\kappa) \Phi(\kappa, \varphi),$$

где $\kappa = \sqrt{\kappa_x^2 + \kappa_y^2}$, $\varphi = \arctg \frac{\kappa_x}{\kappa_y}$. Для удобства, угловое распределение нормируется так, чтобы

$$\int_{-\infty}^{\infty} \Phi(\kappa, \varphi) d\varphi = 1.$$

2.2. Двумерная модель поверхностного волнения

В соответствии с предыдущим разделом, для моделирования случайной поверхности $\xi(\vec{r}, t)$ будем использовать её представление в виде суперпозиции плоских волн с различными частотами и случайными фазами ψ_{nm} , бегущих под разными азимутальными углами φ_m [2]:

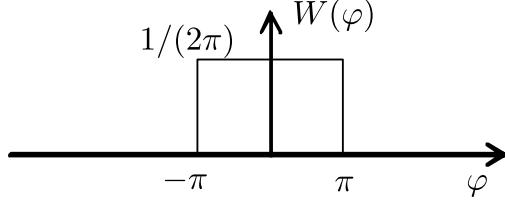


Рис. 1: Плотность вероятности случайной фазы φ .

$$\xi(\vec{r}, t) = \sum_{n=1}^N \sum_{m=1}^M A_n(\kappa_n) \cdot F_m(\kappa_n, \varphi_m) \cos(\omega_n t + \vec{\kappa} \vec{r} + \psi_{nm}), \quad (5)$$

где ψ_{nm} – случайная фаза, равномерно распределенная в интервале от 0 до 2π (см. рис. 1), $F_m(\kappa_n, \varphi_m)$ – азимутальное распределение для гармоники с волновым числом κ_n , $\vec{\kappa}_n = (\kappa_{nx}, \kappa_{ny})$ – волновой вектор.

Амплитуда n -ой гармоники A_n есть мощность на интервале $\Delta\kappa_n$, которая вычисляется по спектру моделируемой поверхности $S_\xi(\kappa)$. Пользуясь определением корреляционной функции (3) и формулой Винера-Хинчина (4) получим точное выражение для нахождения амплитуды n -ой гармоники A_n

$$\begin{aligned} K_\xi[\vec{\rho}, \tau] &= \frac{1}{(2\pi)^2} \int_{-\infty}^{\infty} S_\xi(\vec{\kappa}) e^{i\vec{\kappa}\vec{\rho}} d\vec{\kappa} = \frac{1}{(2\pi)^2} \int_{-\infty}^{\infty} \int_{-\pi}^{\pi} S_\xi(\kappa) \Phi_\xi(\varphi) \kappa e^{i\vec{\kappa}\vec{\rho}} d\kappa d\varphi = \\ &= \frac{1}{(2\pi)^2} \int_{-\infty}^{\infty} \kappa S_\xi(\kappa) e^{i\vec{\kappa}\vec{\rho}} d\kappa = \sum_{n=-\infty}^{\infty} \frac{(A_n(\vec{\kappa}_n))^2}{2} e^{i\vec{\kappa}_n \vec{\rho}} \end{aligned}$$

$$A_n(\kappa_n) = \frac{1}{2\pi} \sqrt{\int_{\Delta\kappa_n} 2\kappa S_\xi(\kappa) d\kappa}$$

Для удобства, введем новое обозначение для спектра $S(\kappa_n) \equiv \kappa_n S_\xi(\kappa_n)$.

Аналогично вычислению амплитуд, можно вычислить азимутальное рас-

пределение F_m следующим образом:

$$F_{nm}(\kappa_n, \varphi_m) = \sqrt{\int_{\Delta\varphi_m} \Phi_\xi(\kappa_n, \varphi) d\varphi},$$

где $\Delta\varphi = \frac{2\pi}{M}$ – шаг по азимутальному углу.

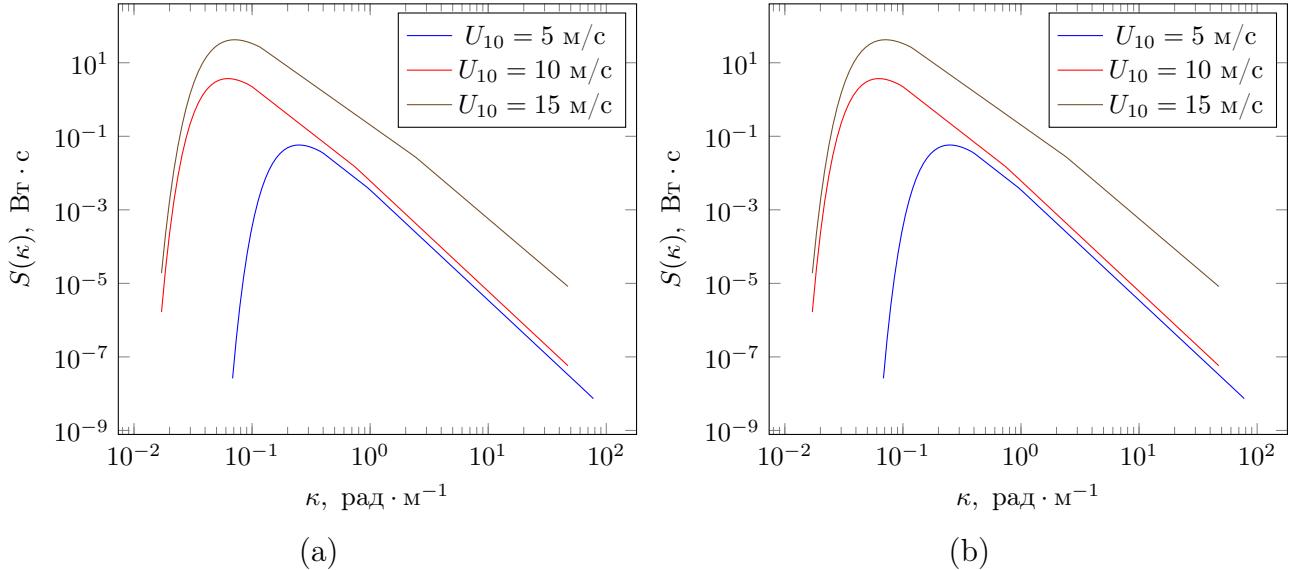


Рис. 2: Спектр высот $S(\kappa)$ для разных скоростей ветра: синяя кривая - 5 м/с, красная кривая - 10 м/с, коричневая кривая - 15 м/с, (а) Ку-диапазон, (б) С-диапазон.

Графики $S(\kappa)$ и $\Phi_\xi(\kappa)$ для приведены на рис. 2 и рис. 3 соответственно [?]. Вычисления на рис. 2 выполнены для скоростей ветра $U_{10} = 5$ м/с (синяя кривая), 10 м/с (красная кривая) и 15 м/с (коричневая кривая), также на рис. 2 учитывается граничное волновое число для моделирования поверхности для двух диапазонов излучения: *Ku* и *C*. В рамках двух масштабной модели рассеивающей поверхности [4]. Волновое число κ_m соответствует максимуму спектра волнения $S(\kappa)$. Стоит заметить, что с ростом скорости ветра число используемых гармоник, необходимых для получения одинакового качества моделирования, возрастает.

Это обусловлено тем, что растет интервал волновых чисел κ , на котором определен спектр волнения.

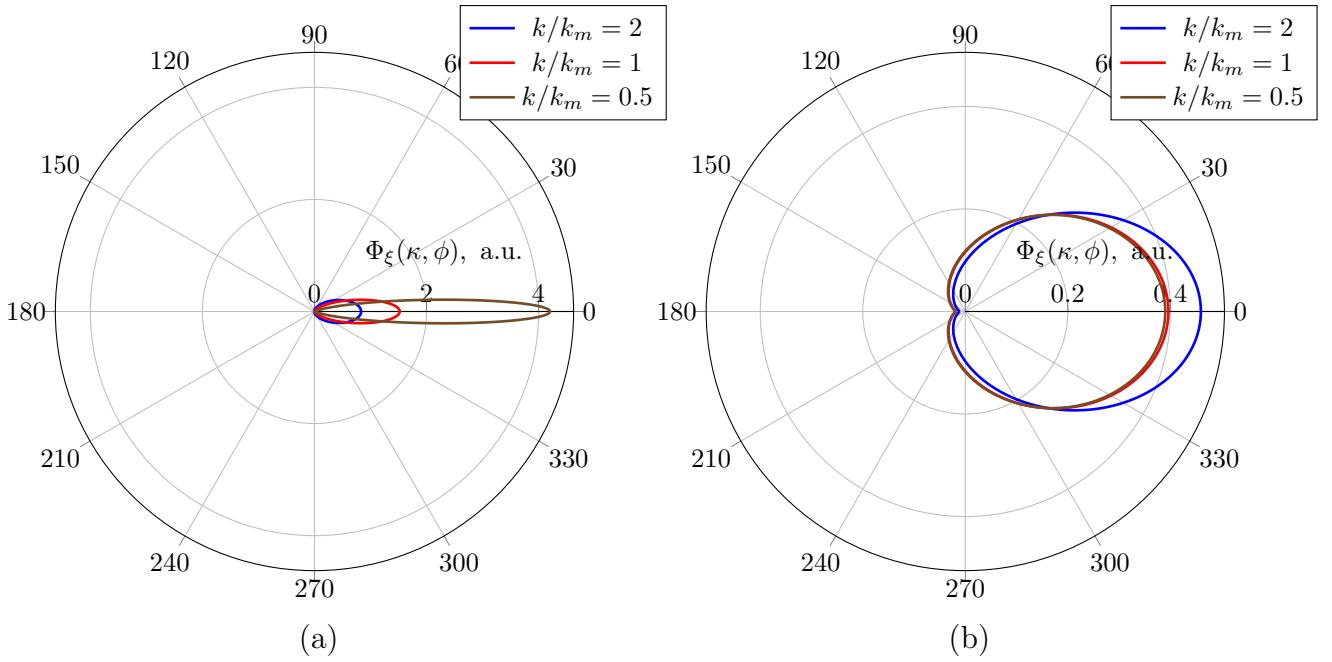


Рис. 3: Спектр $\Phi(\kappa, \varphi)$ для разных соотношений κ/κ_m

На рис. 4 изображены поверхности, построенные по формуле (5).

Для рис. 4а доминантная длина волны равна 23 м, высота значительного волнения 0.83 м, для рис. 4б доминантная длина волны равна 206 м, а высота значительного волнения 6.58 м.

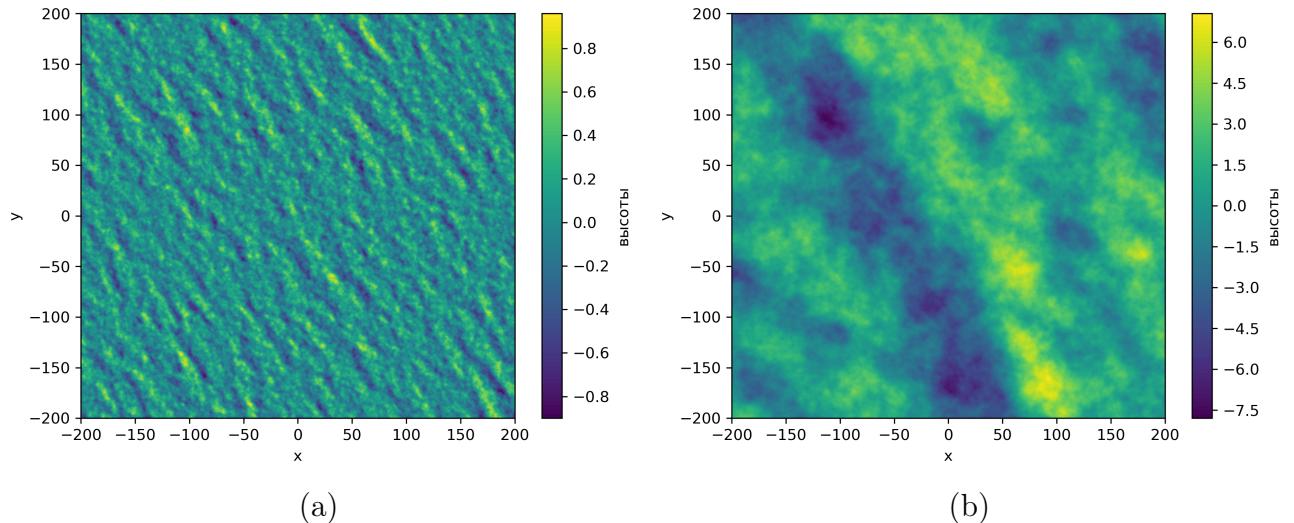


Рис. 4: Полутоновое изображение смоделированного поля высот для направления ветра 30° и разных скоростей ветра (а) $U_{10} = 5\text{м/c}$; (б) $U_{10} = 15\text{м/c}$;

Такой подход к моделированию морской поверхности является одним из самых простых и достаточно эффективным, но у него есть существенные



Рис. 5: Фотография водной поверхности

недостатки.

Прежде всего, моделируемая поверхность получается симметричной, хотя реальная поверхность асимметрична: передний склон волны более крутой и короткий по сравнению с задним склоном.

Кроме того, площадь гребней меньше площади впадин для морского волнения, что также не находит отражения в свойствах моделируемой поверхности. Эти отличия модельной поверхности от морской поверхности не позволяют смоделировать так называемые поправки на состояние морской поверхности [5], [6]. Правильно смоделировать именно поправки на состояние крайне важно для получения достоверных результатов при моделировании формы импульса отраженного радиолокационного сигнала. Как решить эту проблему обсудим в дальнейшем.

Для моделирования морской поверхности необходимо определиться с числом гармоник. Надо отметить, что с ростом скорости ветра число используемых гармоник, необходимых для получения одинакового качества моделирования, будет возрастать. Это обусловлено тем, что увеличивается интервал волновых чисел κ , на котором определен спектр волнения (см. рис. 2).

Следующая задача, которую надо решить, связана с тем, как расположить гармоники по оси волновых чисел. Максимуму спектра волнения соответствует волновое числа κ_m , левую границу спектра определим как $\frac{\kappa}{4}$, а правую границу обозначим κ_{cut} . Это значение будет различаться для КИ- и С-диапазонов. Формула для определения была получена в [2] и приведена в приложении А 1.

Самый простой вариант расположения гармоник это равномерный шаг, который можно определить следующим образом:

$$\Delta\kappa = \frac{k_{cut}}{(N - 1)}, \text{ где}$$

κ_{cut} – граничное волновое число, N – число гармоник.

Критерием качества моделирования, а также оптимального выбора числа гармоник была выбрана близость следующих корреляционных функций высот:

$$K[\rho] = \int_{-\infty}^{\infty} S(\kappa) \cos(\kappa\rho) d\kappa$$

$$\tilde{K}(\rho) = \sum_{n=1}^N \frac{A_n^2}{2} \cos(\kappa_n \rho)$$
(6)

Сравнение корреляционной функции $\tilde{K}[\rho]$ полученной по модели, с теоретической корреляционной функцией $K[\rho]$ позволит оценить качество модели.

Если посмотреть на форму спектра, то задача усложняется тем, что спектр высот является узким и в основном сосредоточен вблизи пика (длинноволновой составляющей спектра волнения). Кроме того, равномерный шаг приводит к появлению «артефактов», что хорошо видно на рис. 6. Вычисления выполнены для полностью развитого ветрового волнения и трех скоростей ветра: 5 м/с, 10 м/с и 15 м/с. Число гармоник для всех скоростей ветра было выбрано равным 256. Для удобства сравнения разных скоростей ветра, при

построении, корреляционные функции были нормированы на дисперсию высот. Частично от артефактов можно избавиться, выбрав неравномерный шаг.

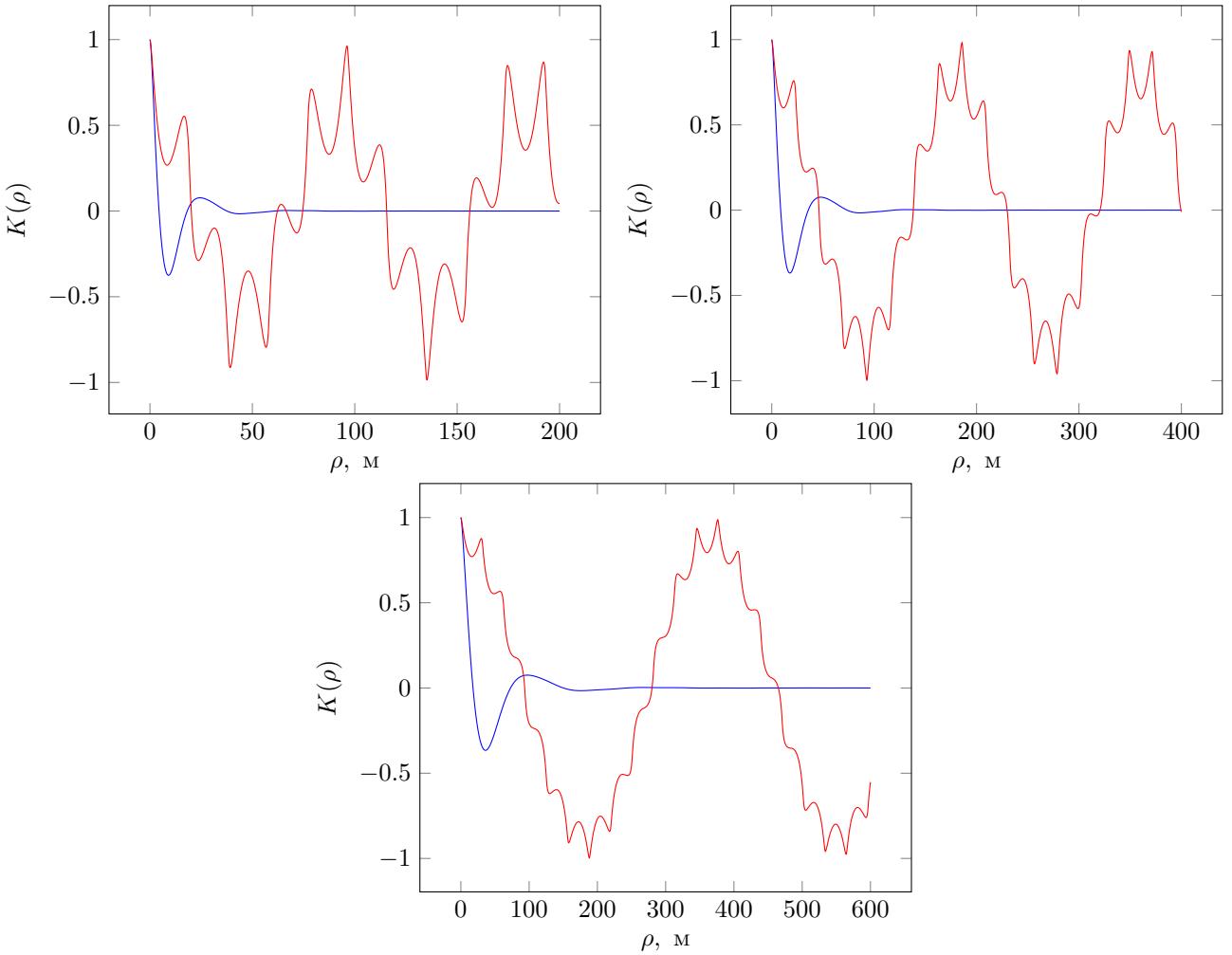


Рис. 6: Нормированная корреляционная функция высот для равномерного распределения и 3-х скоростей ветра: (а) 5 м/с, (б) 10 м/с, (в) 15 м/с и числе гармоник $N = 256$

Нужно задать распределение таким образом, чтобы вблизи малых значений волнового числа (вблизи пика спектра) плотность расположения гармоник была существенно выше, чем при больших κ .

Можно предложить несколько вариантов неравномерного распределения и ниже протестируем два варианта.

В качестве первого распределения выберем следующую формулу

$$k_i = k_{min} + \frac{k_{cut} - k_{min}}{(N - 1)^2} (i - 1)^2$$

Для второго распределения выберем «логарифмический» шаг и положения гармоник определим следующим образом

$$k_i = k_{i-1} \cdot \Delta\kappa$$

На рис. 7 и 8 видно, что лучше всех «работает» логарифмическое разбиение интервала волновых чисел.

Как было отмечено выше, с увеличением скорости ветра число гармоник, необходимых для получения одинакового качества моделирования, возрастает. На их рис.8 видно, что с увеличением скорости ветра отклонение «модельной» корреляционной функции $\tilde{K}[\rho]$ от «истинной» $K[\rho]$ увеличивается и, чтобы уменьшить ошибку моделирования, необходимо увеличить число гармоник (синусоид), а это будет замедлять скорость моделирования морской поверхности.

Как показало тестовое моделирование, для получения «качественной» численной реализации рассеивающей поверхности требуется большей число гармоник, что делает процесс вычислений длительным. Для уменьшения числа гармоник был рассмотрен следующий подход.

2.3. Метод «отбеливания» спектра

2.3.1 Метод «отбеливания» спектра для одной переменной

Для оптимизации времени построения поверхности и уменьшения количества гармоник без уменьшения качества моделирования, предлагается ис-

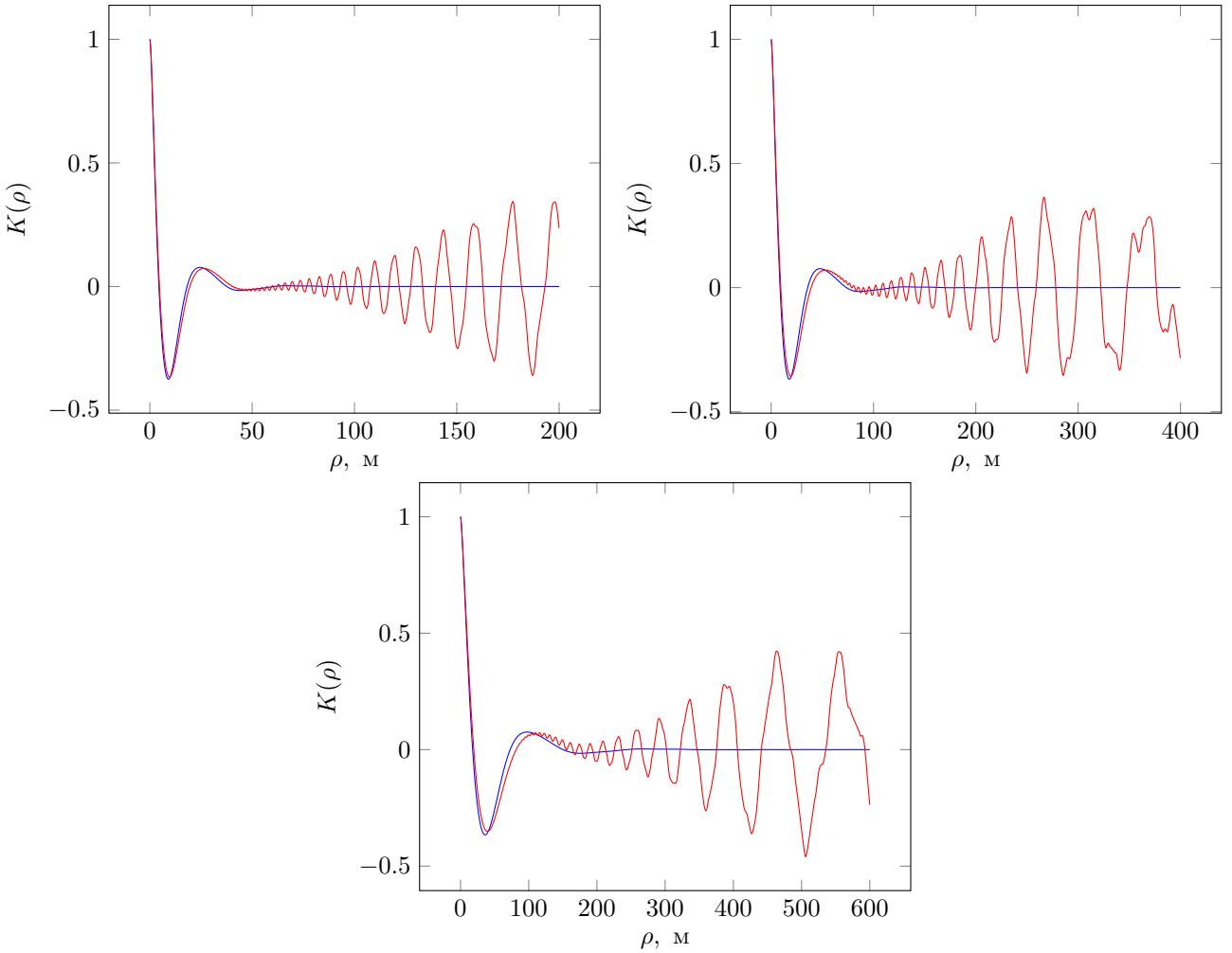


Рис. 7: Нормированная корреляционная функция высот для логарифмического распределения и 4-х скоростей ветра: (а) 5 м/с, (б) 10 м/с, (в) 15 м/с и числе

пользовать следующий метод.

Предположим, что при больших ρ гармонические составляющие корреляционной функции не зависят друг от друга и мы можем пренебречь их взаимной корреляцией. Тогда мощность «шума» функции $\tilde{K}(\rho)$ определяется выражением $\sigma_{\text{шум}}^2 = \sum_{n=1}^N \frac{1}{2} \left(\frac{A_i^2}{2} \right)^2 \equiv \sum_{n=1}^N \frac{b_i^2}{2}$.

В областях малых ρ , напротив, гармоники должны сильно взаимодействовать и соответствующая мощность равна $\tilde{K}^2(0) = \left(\sum_{n=1}^N b_i \right)^2$ (см. (6)).

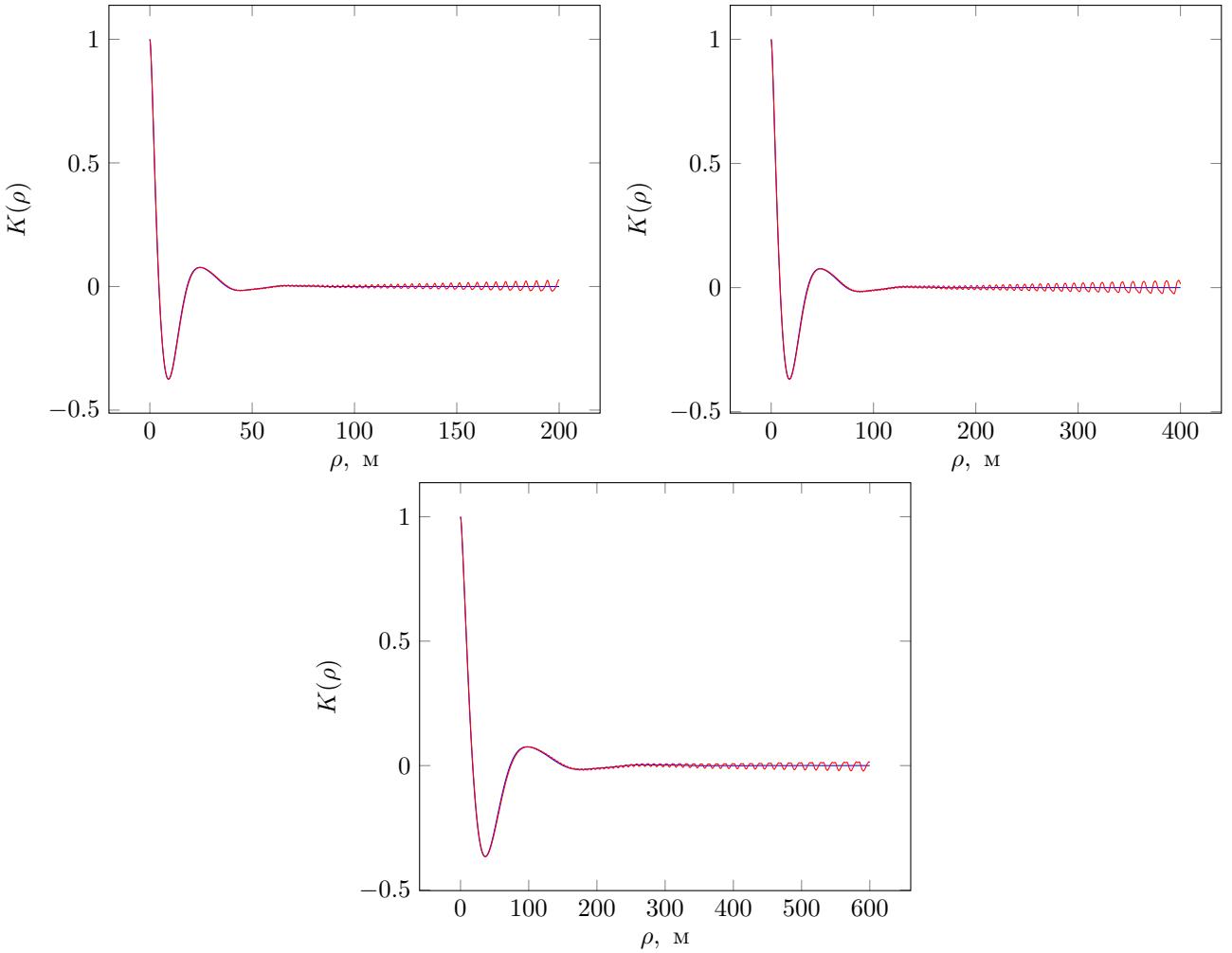


Рис. 8: Нормированная корреляционная функция высот для неравномерного распределения и 3-х скоростей ветра: (а) 5 м/с, (б) 10 м/с, (в) 15 м/с и числе

Образуем величину

$$Q = \frac{\sigma_{\text{шум}}^2}{\tilde{K}^2(0)},$$

которая характеризует относительную мощность шумов. Минимум этой величины находится путём решения системы уравнений

$$\begin{aligned} \frac{\partial Q}{\partial b_i} &= 0, \text{ для } i = 1, 2, \dots, N. \\ \frac{b_i \left(\sum_{n=1}^N b_n \right)^2 - 2 \sum_{n=1}^N b_i \sum_{n=1}^N \frac{b_n^2}{2}}{\left(\sum_{n=1}^N b_n \right)^4} &= 0 \end{aligned}$$

Частным результатом её решения является $b_1 = b_2 = \dots = b_N$.

Спектр модельного поля при этом имеет близкий к белому вид, а выравнивание амплитуд спектральных компонент поля $S(\kappa)$ сводится к разбиению области определения спектра $[\kappa_{min}, \kappa_{max}]$ на участки $\Delta\kappa_i$, интегралы по которым от функции $S(\kappa)$ имеют одно и тоже значение $b_i = b_0 = \frac{\sigma^2}{N}$.

Заметим теперь, что рассуждая о способах разбиения интервала частот $[\kappa_{min}, \kappa_{max}]$ на участки $\Delta\kappa_i$ мы оставляли нерешенным вопрос о выборе расположения гармоник κ_i внутри этих участков. Обычно κ_i ставится у правой границы ячейки $\Delta\kappa_i$. При этом, однако, оказывается, что модельная корреляционная функция плохо совпадает с экспериментальной корреляционной функцией в области малых ρ . Для достижения лучшего согласия следует потребовать сопряжения всех производных (от первого до N -го порядка) функций $\tilde{K}[\rho]$ и $K[\rho]$ при $\rho = 0$. Поскольку $K'[\rho] = \frac{\partial^2 K[\rho]}{\partial \rho^2}$, это условие эквивалентно требованию сопряжения моментов спектра модельного и реального полей, которое записывается в виде

$$\sum_{p=1}^N b_i \kappa_i^{2p} = \int_0^\infty \kappa^{2p} S(\kappa) d\kappa,$$

Полученная система N уравнений для N неизвестных κ_i не имеет общего решения и потому может анализироваться лишь численно. Чтобы упростить решение нашей задачи, потребуем облегченного, по сравнению с предыдущим, условия сопряжения вторых моментов модельного и реального спектров высот

$$b_i \kappa_i^2 = \int_{\Delta\kappa_i} \kappa^2 S(\kappa) d\kappa,$$

где $b_i = A_i^2/2$

Из него непосредственно следует правило нахождения узлов κ_i

$$\kappa_i = \sqrt{\frac{1}{b_0}} \int_{\Delta \kappa_i} \kappa^2 S(\kappa) d\kappa. \quad (7)$$

Формула (7) выведена для спектра высот поверхности волнения. Когда возникает необходимость моделирования уклонов, то необходима сделать замену переменной $S(\kappa) \rightarrow k^2 S(\kappa)$, чтобы получить формулу для нахождения правила расположения гармоник для уклонов

$$\kappa_i = \sqrt{\frac{1}{b_0}} \int_{\Delta \kappa_i} \kappa^4 S(\kappa) d\kappa. \quad (8)$$

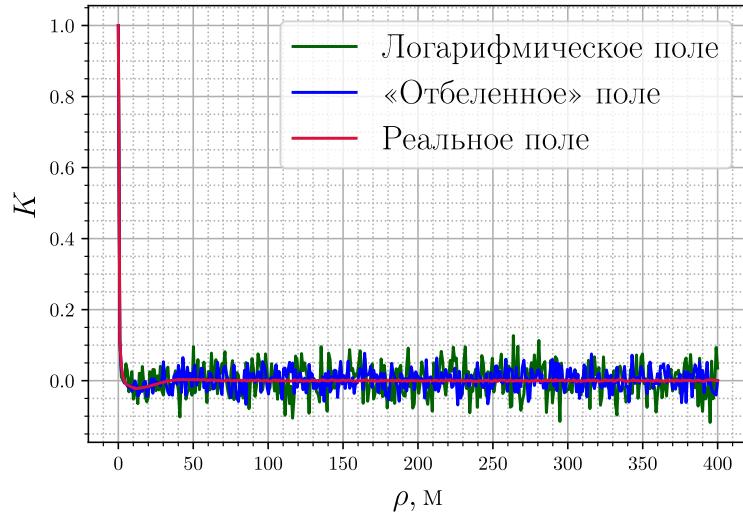


Рис. 9: Нормированная корреляционная функция уклонов для логарифмического расположения гармоник (зеленая кривая) и расположения по методу отбеливания спектра (синяя кривая) для скорости ветра 10 м/с

Такой способ выбора расположения гармоник, как нетрудно убедиться, обеспечивает сопряжение корреляционных функций реального и модельного полей по второй производной в нуле, или, иначе говоря, равенство дисперсий кривизн этих полей.

В результате метод «отбеливания» дает лучший результат из всех рас-

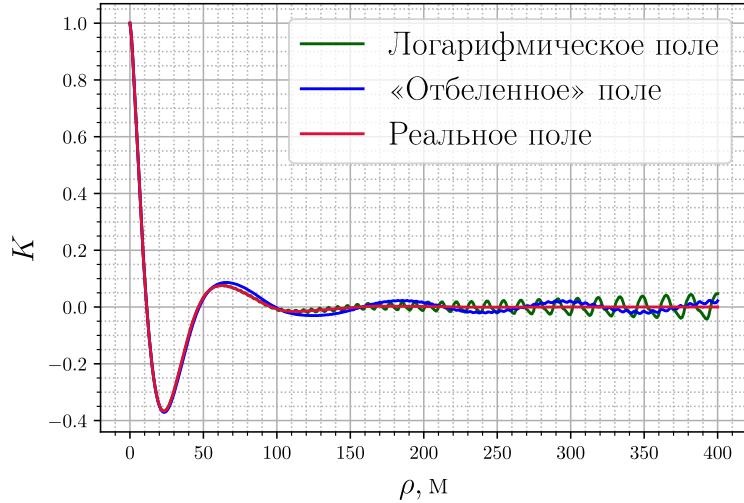


Рис. 10: Нормированная корреляционная функция высот для логарифмического расположения гармоник (зеленая кривая) и расположения по методу отбеливания спектра (синяя кривая) для скорости ветра 10 м/с

смогутных подходов.

Из рис. 11 и 12 видно, что определение положения гармоник по методу отбеливания является эффективным только для той переменной, которая использовалась в процедуре отбеливания. Для другой переменной результат получается не слишком хорошим, что свидетельствует о необходимости использования другого подхода при необходимости одновременного моделирования поля высот и поля уклонов.

Для решения задачи рассеяния электромагнитного излучения морской поверхностью, необходимо моделировать поле высот (определяет форму импульса) и поле уклонов, которое определяет условие обратного рассеяния падающего излучения.

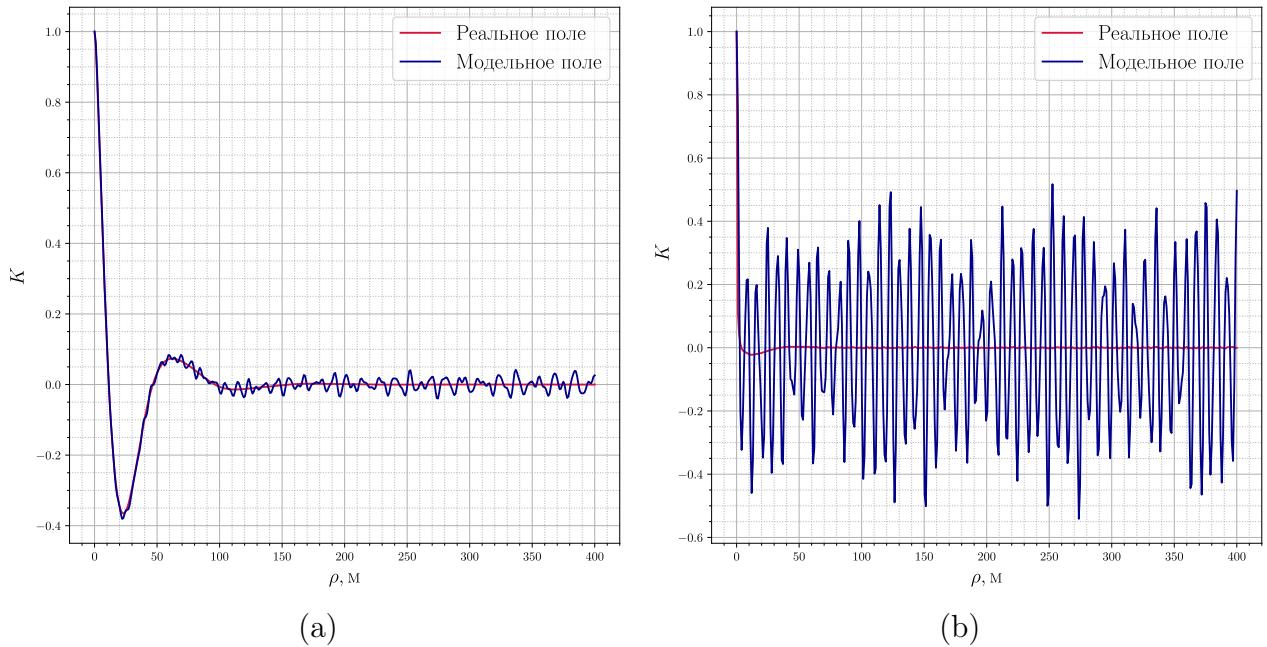


Рис. 11: Нормированная корреляционные функции высот (а) и уклонов (б) при расположении гармоник по методу «отбеливания» спектра по формуле (7)

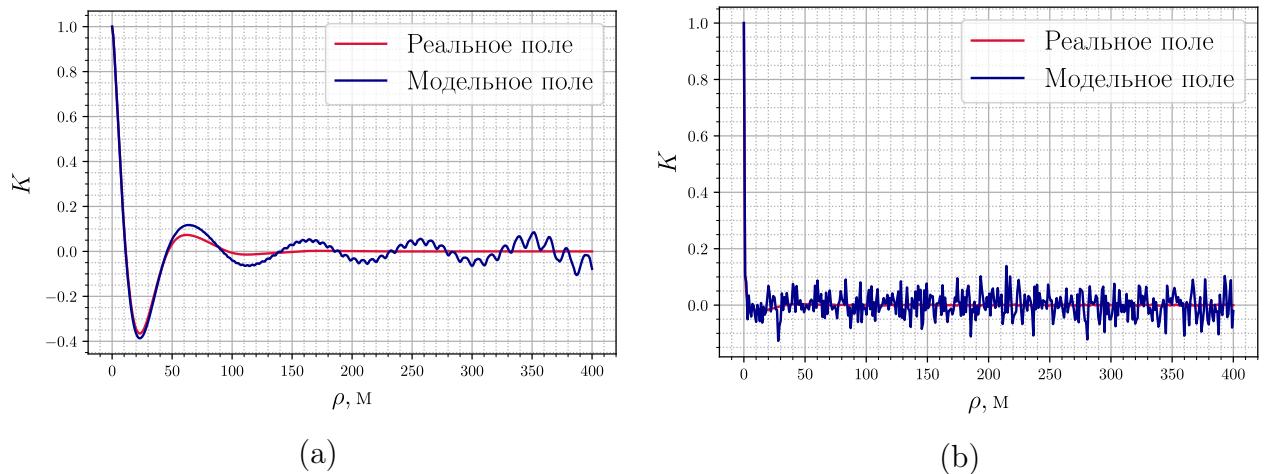


Рис. 12: Нормированная корреляционные функции высот (а) и уклонов (б) при расположении гармоник по методу «отбеливания» спектра по формуле (8)

2.3.2 Метод «отбеливания» спектра для двух переменных

Для такой задачи необходима рассмотреть другую функцию относительных шумов Q , например

$$Q = \frac{(\sigma_{\text{шум}}^{\text{h}})^2}{(\tilde{K}^{\text{h}}(0))^2} + \frac{(\sigma_{\text{шум}}^{\text{v}})^2}{(\tilde{K}^{\text{v}}(0))^2}, \quad (9)$$

где индексы « h » и « v » соответствуют наклонам и высотам. Учитывая то, что оба слагаемых в уравнении (9) вещественны и положительны, то экстремум функции Q можно найти, зная экстремум каждого слагаемого по отдельности.

Тогда, гармоники, определяющие минимум первого слагаемого описываются формулой (7), а минимум второго – формулой (8).

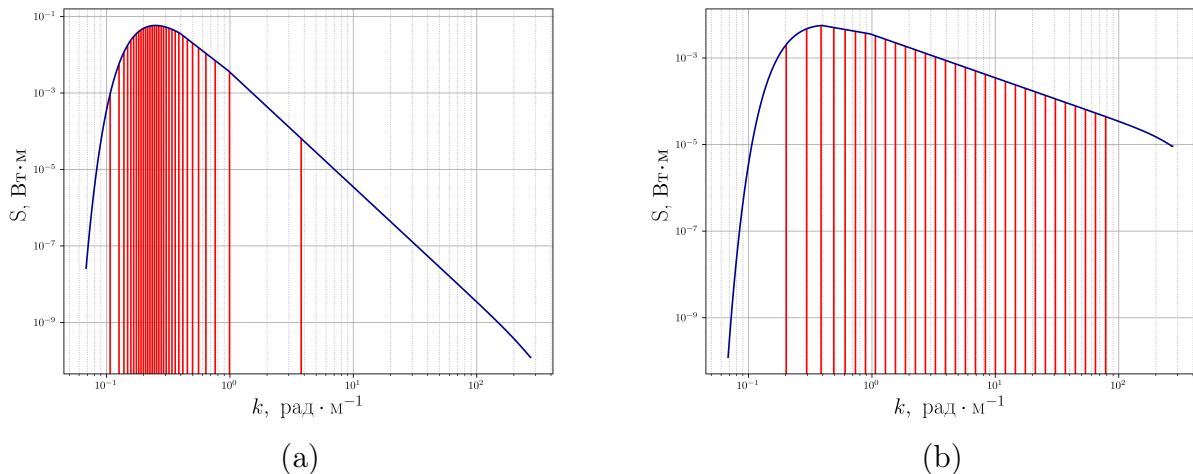


Рис. 13: Расположение гармоник для отбеливания (а) высот, (б) уклонов

Таким образом, двумерный вариант метода отбеливания является эффективным способом выбора расположения гармоник для численного моделирования морской поверхности, задаваемой моделью спектра.

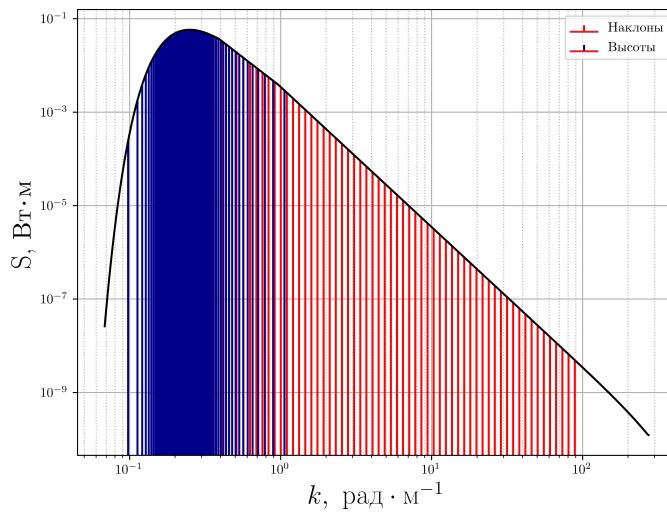


Рис. 14: Совмещение расположение гармоник для отбеливания

2.3.3 Аппаратное ускорение моделирования

В прошлом разделе мы обсуждали как можно ускорить без потери качества процесс моделирования за счет уменьшения количества гармоник в спектре волнения.

Когда с математической точки зрения все оптимизировано можно перейти к программной оптимизации: поскольку основное время моделирования приходится на суммирование в цикле по формуле (5), на который приходится $N \times M \times X \times Y$ итераций, где N – число гармоник в частотном спектре, M – число гармоник в азимутальном распределении, X – размер сетки вдоль оси x , Y – размер сетки вдоль оси y . Этот цикл требует больших затрат мощности и времени и именно его мы можем значительно ускорить благодаря его внутренней простоте.

Современные центральные процессоры (CPU) уже давно имеют в своём распоряжении несколько (обычно 4-8) вычислительных ядер, которые в нередко в несколько потоков могут производить вычисления.

Самое очевидное, что можно сделать – это выполнять программный код не в одном потоке процессора, а специальным образом разделить координатную

сетку на блоки и вычислять каждый блок в отдельном потоке.

Но ещё быстрее можно произвести вычисления на графическом процессоре (GPU). Современные GPU имеют порядка 1000 вычислительных ядер, что позволяет очень сильно ускорить процесс моделирования за счет распараллеливания вычислений между ядрами.

На рисунке рис. 15 приведено сравнение многопоточных вычислений на CPU и GPU из одной ценовой категории. Поверхность моделировалась при $N = 2048$, $M = 512$. При вычислении поверхности 256×256 точек центральному процессору для этого требуется 2030 секунд, в то время как графический процессор справится с этой задачей лишь за 71 секунду.

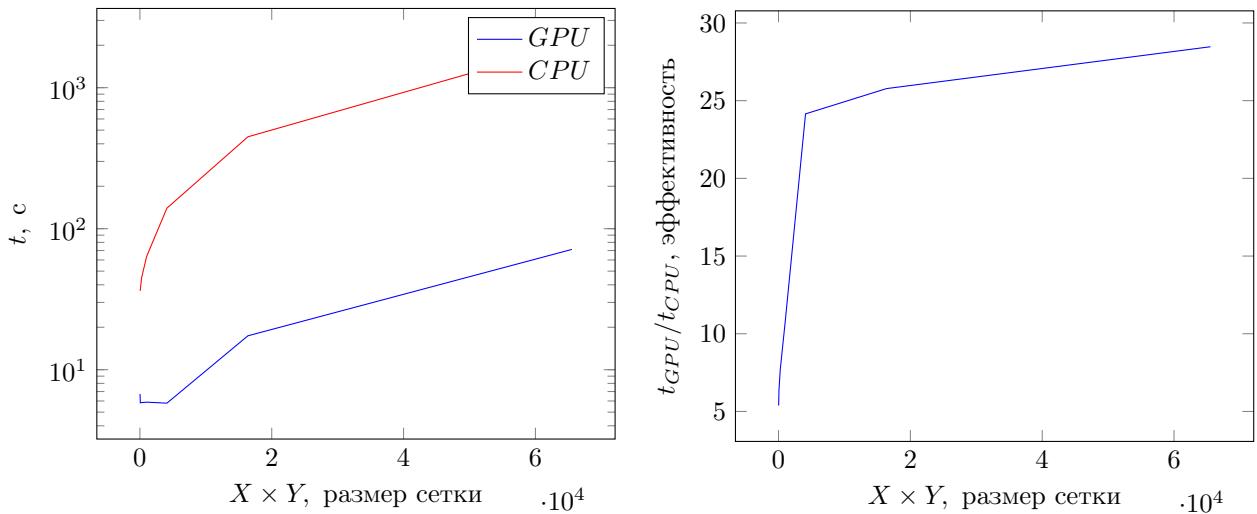


Рис. 15: Сравнение времени моделирования двумерной поверхности на CPU и GPU: (а) время вычисления поверхности на сетке размером $X \times Y$; (б) относительная скорость вычислений t_{GPU}/t_{CPU} ;

Программную реализацию вычислений на графическом процессоре можно посмотреть в приложении А в листинге 2.

2.4. Заостренная морская поверхность

Как отмечалось ранее, при моделировании морской поверхности синусоидами мы получаем нулевое среднее значение высот, что не позволяет смоде-

лировать поправки на состояние морской поверхности.

Ниже предлагается модель поверхности у которой средний уровень не равен нулю.

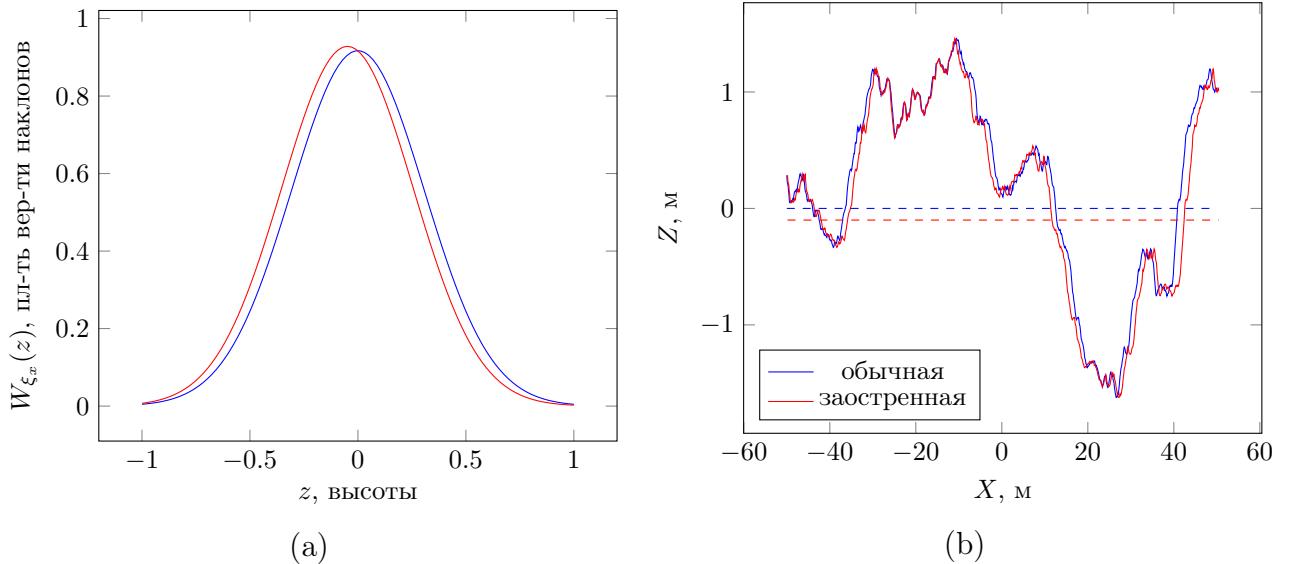


Рис. 16: (а) Плотность вероятности уклонов $W_{\xi_x}(z)$ для линейной поверхности (синяя кривая) и заостренной поверхности (красная кривая) в зависимости от высот z при скорости ветра 10 м/с
 (б) Срез поля высот морской поверхности для стандартного подхода (синяя кривая) и модели заостренной поверхности (красная кривая) при скорости ветра 10 м/с. Пунктиром показан средний уровень соответствующей поверхности

2.4.1 Двумерный случай

Рассмотрим для начала задачу моделирования двумерной поверхности суммой гармоник с детерминированными амплитудами и случайными фазами

$$z = \sum_{j=0}^N A_j \cos(k_j x + \psi_j)$$

Чтобы получить модель заостренной волны введем нелинейное преобразование координат

$$\{x, z(x)\} \longrightarrow \{x + D(x), z(x)\},$$

где $D(x)$ горизонтальное смещение

$$D(x) = \frac{i}{2\pi} \int_{-\infty}^{\infty} \hat{z} e^{ikx} dk,$$

а $S(k)$ – прямое Фурье преобразование исходной поверхности

$$S(k) = \int_{-\infty}^{\infty} z(x) e^{-ikx} dx$$

В нашем случае, функция $D(x)$ примет вид:

$$\begin{cases} x = x_0 - \underbrace{\sum_{j=0}^N A_j \sin(k_j x_0 + \psi_j)}_{D(x)} \\ z = \sum_{j=0}^N A_j \cos(k_j x_0 + \psi_j) \end{cases}$$

Иными словами мы будем моделировать волнение не суммой гармонических функций, а суммой трохоид.

Для того, чтобы наше преобразование $D(x)$ имело физический смысл необходимо, чтобы для каждой j -ой гармоники выполнялось соотношение

$$A_j k_j \ll 1$$

Статистические моменты Запишем характеристическую функцию нового случайного процесса $z(x_0(x))$ по определению

$$\Theta(i\theta) = \langle e^{i\theta z(x_0(x))} \rangle \quad (10)$$

Поскольку процесс $z(x_0)$ стационарный, то от (10) можно перейти к

$$\Theta(i\theta) = \lim_{L \rightarrow \infty} \frac{1}{2L} \int_{-L}^L e^{i\theta z(x_0)} dx = \lim_{L \rightarrow \infty} \frac{1}{2L} \int_{-L}^L e^{i\theta z(x_0)} (1 + D'(x_0)) dx_0 \quad (11)$$

Поскольку $z(x_0)$ стационарный процесс, а $D'(x_0)$ стационарен по нашему определению, то (11) преобразуется к виду

$$\Theta(i\theta) = (1 - i\theta\sigma_1^2) \exp\left(-\frac{1}{2}\theta^2\sigma_0^2\right), \quad (12)$$

где $\sigma_n^2 = \int_{-\infty}^{\infty} k^n S(k) dk$ – момент n -го порядка спектра волнения.

Зная характеристическую функцию не сложно получить необходимые статистические моменты дифференцируя (12)

$$m_n = i^{-n} \frac{d^n \Theta(i\theta)}{d\theta^n} \Big|_{\theta=0}$$

Следовательно, среднее и дисперсия случайного процесса $z(x_0)$ будут равны

$$\begin{aligned} \langle z \rangle &= -\sigma_1^2, & \langle z^2 \rangle &= \sigma_0^2 \\ \langle z^2 \rangle - \langle z \rangle^2 &= \sigma_0^2 - \sigma_1^4 \end{aligned}$$

Также не сложно получить связь уклонов в смещенных координатах x с наклонами в несмешенных координатах x_0 пользуясь определением уклонов

$$z'(x) = \frac{dz(x)}{dx} = \frac{z'(x_0)}{1 + D'(x_0)}$$

2.4.2 Трехмерный случай

Для трехмерного случая Пирсон [7] представил решение линеаризованных уравнений движения для невязкой жидкости в лагранжевых координатах. Он показал, что в глубокой воде положение частиц на свободной поверхности задается следующими параметрическими уравнениями

$$\begin{cases} z(\vec{r}, t) = \sum_{n=1}^N \sum_{m=1}^M A_n(\kappa_n) \cdot F_m(\kappa_n, \varphi_m) \cos(\omega_n t + \vec{\kappa}_n \vec{r}_0 + \psi_{nm}), \\ x = x_0 - \sum_{n=1}^N \sum_{m=1}^M A_n(\kappa_n) \cdot F_m(\kappa_n, \varphi_m) \cos \varphi_m \sin(\omega_n t + \vec{\kappa}_n \vec{r}_0 + \psi_{nm}), \\ y = y_0 - \sum_{n=1}^N \sum_{m=1}^M A_n(\kappa_n) \cdot F_m(\kappa_n, \varphi_m) \sin \varphi_m \sin(\omega_n t + \vec{\kappa}_n \vec{r}_0 + \psi_{nm}), \end{cases}$$

где $\vec{\kappa}$ – двумерный волновой вектор, $\vec{r}_0 = (x_0, y_0)$, $\vec{r} = (x, y)$

Статистические моменты В трехмерном случае вычисления аналогичны двумерному случаю, но более громоздкие.

Введем смешанный $\sigma_{\alpha\beta\gamma}^2$ и начальный σ_n^2 моменты спектра волнения

$$\sigma_{\alpha\beta\gamma}^2 = \int \frac{\kappa_x^\alpha \kappa_y^\beta}{\kappa^\gamma} S(\vec{\kappa}) d\vec{\kappa}, \quad \sigma_n^2 = \int \kappa^n S(\vec{\kappa}) d\vec{\kappa}$$

можно получить следующую характеристическую функцию для трехмерного волнения

$$\Phi(\theta) = (1 - i\theta\sigma_1^2 + \theta^2\Sigma_1) \exp\left(-\frac{1}{2}\theta^2\sigma_0^2\right), \quad (13)$$

где $\Sigma_1 = \sigma_{111}^4 - \sigma_{201}^2\sigma_{021}^2$.

Из этой характеристической функции можно получить необходимые моменты процесса

$$\langle z \rangle = -\sigma_1^2, \quad \langle z^2 \rangle = \sigma_0^2 - 2\Sigma_1$$

На рис. 16а представлены срезы трехмерной морской поверхности для

стандартного подхода и метода заостренной волны. На рис. 16b изображена теоретическая плотность вероятности наклонов для двух подходов.

На рис. 17 представлена эволюция во времени гребня волны для двух методов.

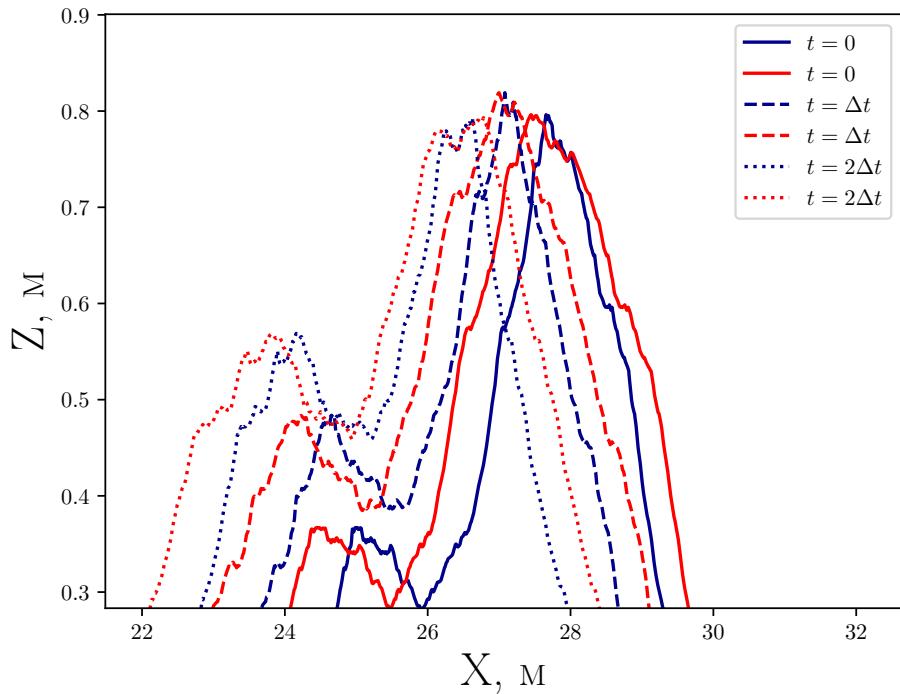


Рис. 17: Эволюция поверхности, построенной стандартным подходом в сравнении с моделью заостренной поверхности

На практике средний уровень морской поверхности не совпадает с тем, что может определить альтиметр. Этот эффект возникает из-за того, что площадь впадин на поверхности превышает площадь гребней, а значит во впадинах будет больше отражающих зеркальных точек. Из вида характеристической функции $\Phi(\theta)$ в формуле (13) мы можем найти связь плотности вероятности наклонов обычной поверхности с заостренной

$$\tilde{P}_{\xi_x}(z) = P_{\xi_x}(z) \left(1 + \frac{\Sigma_1}{\sigma_0^2} - \frac{\sigma_1^2}{\sigma_0^2} z - \frac{\Sigma_1}{\sigma_0^4} z^2 \right),$$

где $P_{\xi_x}(z)$ – гауссовая плотность вероятности наклонов линейной поверхности,

z – высоты морской поверхности.

На рис. 16 изображен график функции \tilde{P}_{ξ_x} в сравнении с функцией $P_{\xi_x}(z)$.

Можно заметить, что область нулевых наклонов, а значит и отражающих точек функции $\tilde{P}_{\xi_x}(z)$ смещается в сторону отрицательных высот, что приводит к изменению длительности импульса отраженного от такой поверхности импульса. Это приводит к изменению формы переднего фронта импульса, излучаемого радиолокатором. О значении этого эффекта речь пойдет следующих разделах.

3. Моделирование отраженного импульса

Важным преимуществом орбитального радиовысотомера по сравнению с другими радиолокаторами является то, что теоретические модели рассеяния хорошо описывают свойства радиолокационного сигнала, отраженного морской поверхностью. В результате алгоритмы обработки получены не с помощью регрессионного анализа, а основаны на аналитических формулах для формы отраженного импульса.

Если, например, говорить об определении скорости ветра по сечению обратного рассеяния для скаттерометра, то алгоритмы были получены благодаря применению регрессионного анализа массива данных, сформированного из контактных измерений скорости и направления ветра (морские буи) и сечения обратного рассеяния, измеренного радиолокатором. Погрешность оценки скорости ветра по сечению обратного рассеяния обусловлена неоднозначностью связи скорости ветра и сечения обратного рассеяния.

У радиовысотомера при определении с высоты значительного волнения происходит именно процесс измерения, т.к. существует однозначная связь формы переднего фронта отраженного импульса и высоты значительного волнения, которая выражается через известную формулу. В данном точность измерения ограничивается параметрами радиолокатора, в частности, длительностью излучаемого импульса и частотой дискретизации.

При измерении расстояния от радиолокатора до среднего уровня морской поверхности алгоритм также опирается аналитические формулы и модели, например, учитывает особенности распространения электромагнитного излучения в атмосфере и ионосфере, что позволяет обеспечить высокую точность.

Благодаря возможности достоверного теоретического описания рассеяния электромагнитного излучения взволнованной водной поверхностью, численное моделирование является эффективным инструментом для моделирова-

ния работы радиовысотомера и отладки алгоритмов обработки. С его помощью можно провести численный эксперимент и рассмотреть по отдельности и в комплексе влияние множества факторов, которые вносят вклад в точность измерений.

3.1. Схема измерения

Преимущество численного моделирования по сравнению с экспериментом состоит в том, что достаточно просто провести сравнение различных схем измерения и оценить их эффективность для решения конкретной задачи. Однако для этого необходимо подробно описать и перевести в числовую форму все важные для моделирования параметры схемы измерения. В результате это позволит провести полноценный «численный» эксперимент. Для описания схемы измерения необходимо задать угол зондирования (падения) θ , высоту орбиты H_0 , скорость и направление движения v_{rad} , и направление зондирования φ_{rad} .

Расстояние от радиолокатора до точки отражения на плоскости xy равно R_0 . Для определенности выберем направление движения радиовысотомера вдоль оси x .

Для плоской поверхности формирование отраженного импульса начинается при касании поверхности передним фронтом падающего импульса в точке непосредственно под радиовысотомером. Это кратчайшее расстояние от радиовысотомера до поверхности. На рис.18 показан пример изменения формы рассеивающей площадки и формы отраженного импульса в зависимости от времени.

Для нахождения отраженного импульса необходимо выполнить интегрирование по рассеивающей площадке для сферической волны с учетом длительности зондирующего импульса.

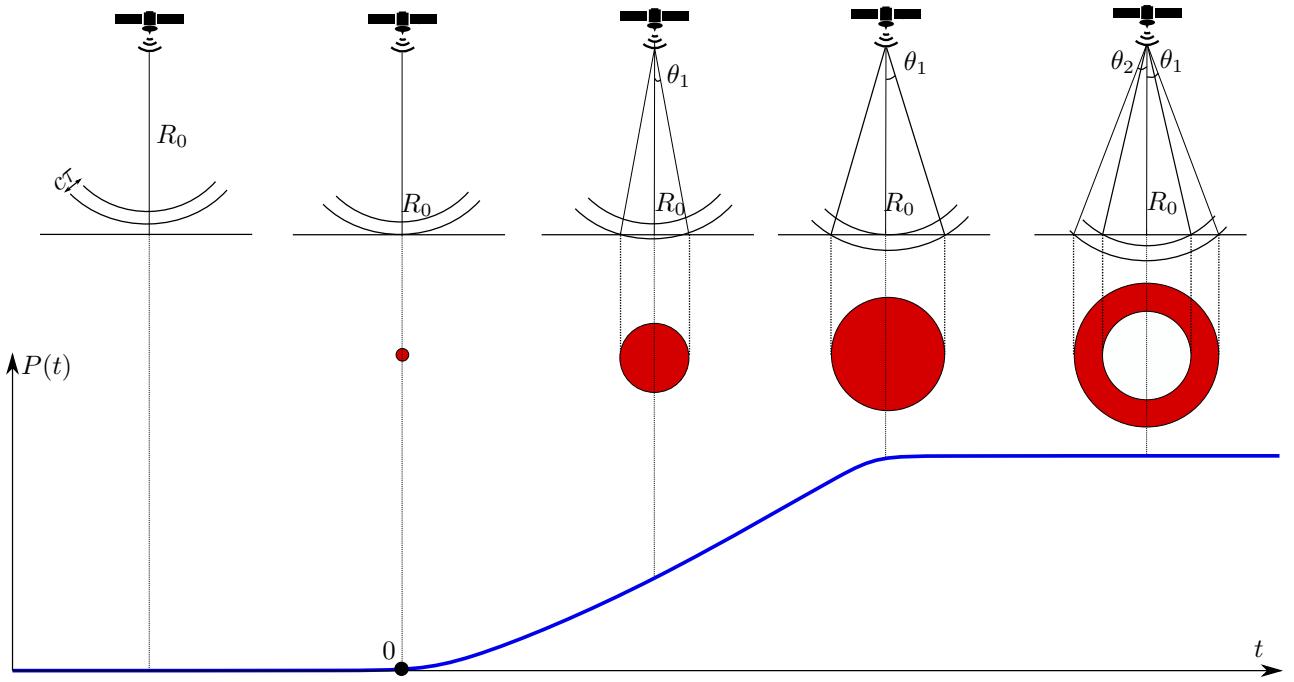


Рис. 18: Процесс формирования отраженного импульса в зависимости от времени

3.2. Влияние морского волнения на форму отраженного импульса

Если говорить о морской поверхности, то перед интегрированием по рассеивающей площадке необходимо выяснить какие участки поверхности будут вносить свой вклад в формирование отраженного импульса. При малых углах падения механизм обратного рассеяния является квазизеркальным и отражение происходит на участках волнового профиля, ориентированных перпендикулярно падающему излучению. Тогда в формировании отраженного сигнала будут участвовать только площадки, ориентированные нормально к излучению. Поэтому для моделирования рассеяния нам необходимо знать не только высоту в выбранной точке поверхности, но и уравнение касательной к ней плоскости, другими словами необходимо знать наклоны ζ_x и ζ_y в искомой точке.

Зная координаты радиолокатора (x_0, y_0, z_0) , координаты точки на поверх-

ности (x, y, z) и наклоны в этой точке $(\zeta_x, \zeta_y, 1)$, можем из геометрии (см. рис. 19) получить локальный угол падения волны на морскую поверхность θ_0 :

$$\cos \theta_0 = \frac{\vec{R}\vec{n}}{|\vec{R}||\vec{n}|}, \text{ где} \quad (14)$$

R – расстояние от радиолокатора до отражающей точки, \vec{n} – нормально касательной плоскости, проведенной к отражающей точке. В случае численного моделирования, когда мы хотим решить задачу нахождения формы отраженного импульса от известной морской поверхности, можем найти \vec{R} как

$$\vec{R} = \vec{R}_0 + \zeta_x \vec{n}_z,$$

Вероятность того, что угол θ_0 будет точно равен нулю и произойдет зеркальное отражение для случайной выбранной точки очень мала, поэтому имеет смысл рассматривать квазизеркальное отражение и вводить ограничение на максимально допустимый локальный угол отражения.

Нахождение всех зеркальных точек на характерном пятне радиолокатора $5 \times 5 \text{ км}^2$ представляет собой ресурсоемкую задачу. Но поскольку формирование импульса носит статистический характер, то мы можем ограничиться лишь выборкой зеркальных точек.

Теперь, для вычисления поля вблизи приемной антенны радиолокатора нам необходимо просуммировать отраженное от квазизеркальных точек поле.

Запишем скалярное поле, излучаемое антенной радиолокатора в зеркальную точку с радиус-вектором \vec{r}

$$E_s(\vec{r}, t) = \frac{E_0}{r} \cdot e^{i(\omega t - \vec{k}\vec{R})} U(t) G(\theta),$$

где $U(t)$ – некоторая функция, ограничивающая длительность импульса, $G(\theta)$

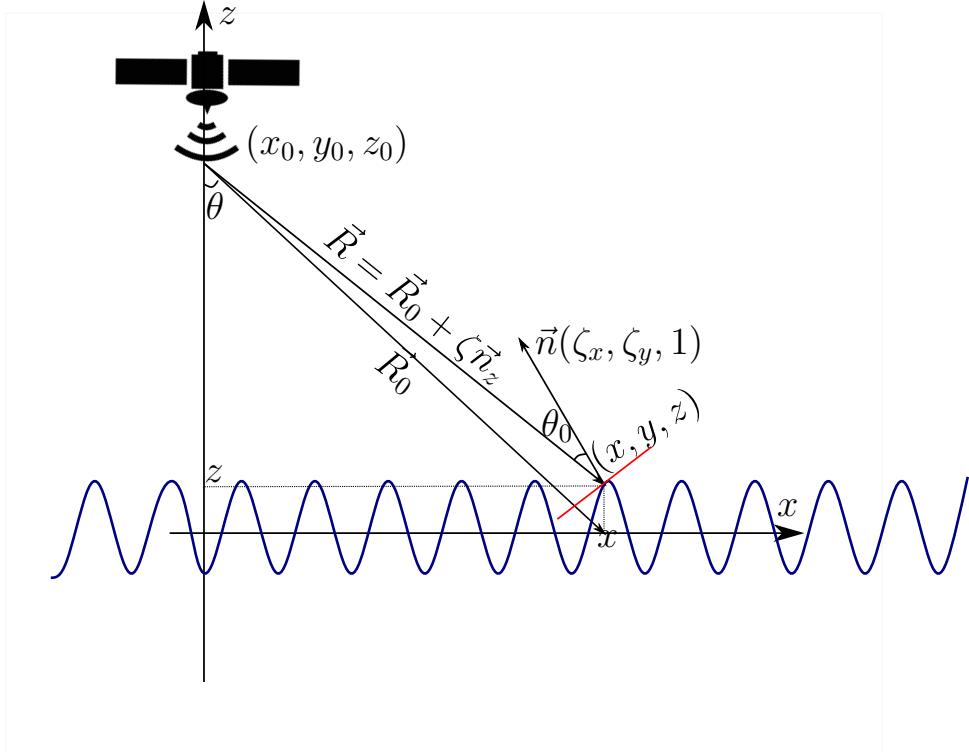


Рис. 19: Геометрия определения локального угла падения. Красной линией обозначена касательная плоскость к рассматриваемой отражающей точке (x, y, ζ)

– диаграмма направленности антенны.

Тогда вблизи приемной антенны амплитуду поля E можно записать как

$$E = \frac{E_0}{R^2} \exp\left\{-2i\vec{k}\vec{R}\right\} \sigma^o G^2(\theta), \text{ где} \quad (15)$$

σ^o – сечение обратного рассеяния площадки.

В численном моделировании сложно быстро оценить мощность каждой отдельно взятой площадки, но в соответствии с [4] при малых углах падения эффективно работает метод Кирхгофа и сечение обратного рассеяния σ^o можно найти по формуле

$$\sigma^o = \frac{|F(0)|^2}{2 \cos^4 \theta \sqrt{\sigma_{yy}^2 \sigma_{xx}^2}} \cdot \exp\left\{-\frac{-\tan^2 \theta}{2\sigma_{xx}^2}\right\},$$

где σ_{xx}^2 , σ_{yy}^2 – дисперсии наклонов вдоль оси x и y соответственно, вычисляемые по модельной реализации, $F(0)$ – коэффициент Френеля, θ – угол падения на рассеивающую площадку.

Остается только проинтегрировать уравнение (15) по всем отражающим точкам

$$E \sim \sum_{i=1}^M \frac{E_0}{R_i^2} \exp\{-2ikR_i\} G^2(x, y, \theta_0)$$

где M – количество точек, x_i, y_i – координаты i -ой отражающей точки, R_i – расстояние от спутника до i -ой точки.

Результирующая мощность импульса будет равна

$$P(t) = \frac{EE^*}{2}$$

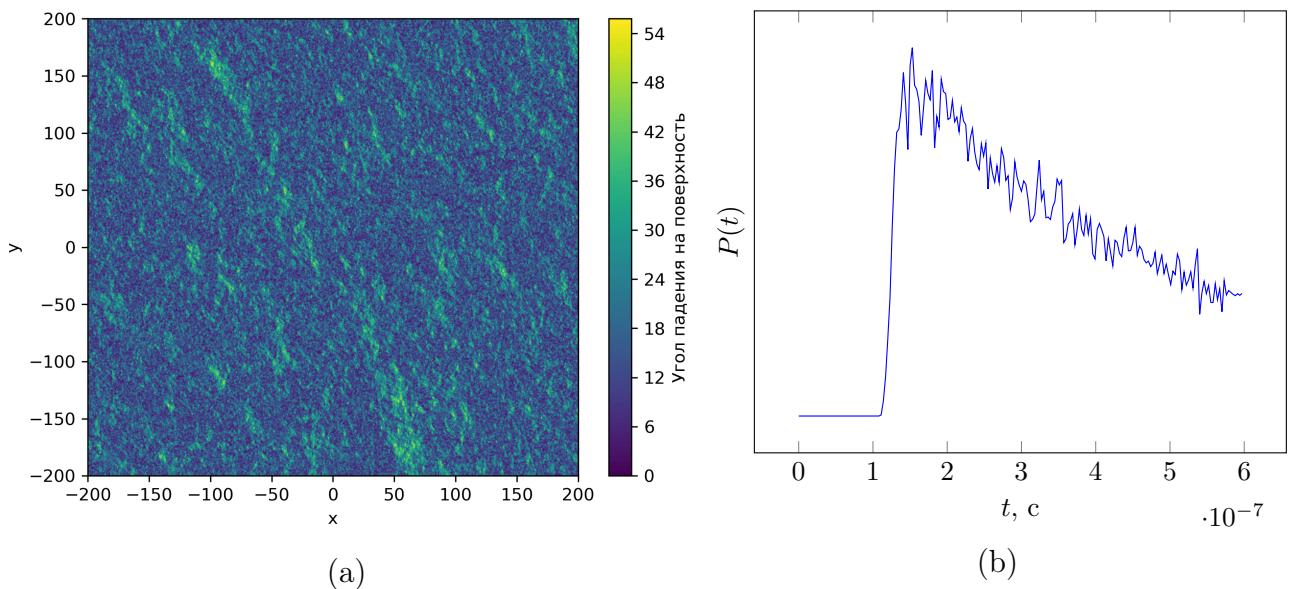


Рис. 20: (а) Вычисление локального угла падения (см.(14)) для радиовысотомера, находящегося в точке с координатами $(0, 0)$ на высоте 1000 км над уровнем моря. Точки, градусная мера которых меньше $\theta < 1^\circ$ в дальнейшем будут считаться зеркальными и они будут участвовать в формировании отраженного импульса. (б) Форма отраженного импульса в зависимости от времени.

На рис. 20 изображен модельный импульс, полученный суммированием отраженной мощности от выборки зеркальных точек на модельной поверхности.

Теперь нам необходимо обосновать теоретически полученный импульс и проверить как соотносится этот модельный импульс с импульсами, полученными реальными спутниками.

Об этом и пойдет речь в следующем разделе.

3.3. Аналитическая модель для формы формы отраженного импульса

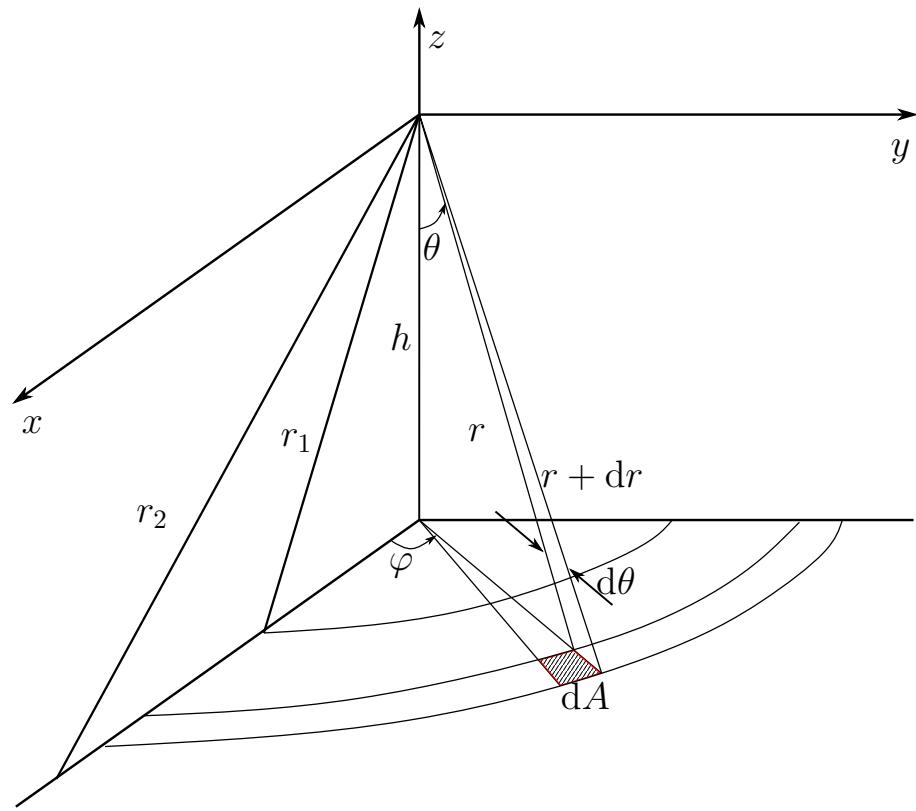


Рис. 21: Геометрия задачи вычисления отклика радиолокатора на плоскую поверхность с учетом отклонения антенны от надира

Посчитаем теоретически отклик плоской морской поверхности P_{FS} на сигнал с радиолокатора. Предполагаем известными диаграмму направленности $G(\theta)$, мощность излученной энергии как функцию времени $P(t)$ и длину волны излучения λ .

В рамках двухмасштабной модели морской поверхности отраженный сигнал при малых углах падения будем искать в приближении метода Кирхгофа [4]. Тогда для морской поверхности мощность отраженного сигнала от пло-

щадки m можно записать в следующем виде

$$P_{FS_m} = \frac{P(t - \frac{2r_m}{c})G_m}{4\pi r_m^2} \cdot \sigma_m A_m \cdot \frac{1}{4\pi r_m^2} \cdot \frac{G_m \lambda^2}{4\pi} = \frac{P(t - \frac{2r_m}{c})G_m^2 \lambda^2 \sigma_m}{(4\pi)^3 r_m^4}, \quad (16)$$

где r_m – расстояние от радара до рассеивающей площадки, σ_m – удельная эффективная площадь рассеяния площадки, G_m – диаграмма направленности антенны в направлении на рассеивающую площадку, A_m – площадь площадки.

Первый множитель в уравнении (16) соответствует плотности мощности излучаемого сигнала. Второй множитель характеризует энергию падающего излучения, которая излучается в направлении приемника, то есть эффективную площадь рассеяния. Третий множитель характеризует рассеяние в пространстве отраженной мощности из-за сферичности волны. Четвертый коэффициент это апертура антенны.

Для того, чтобы найти полную мощность переизлученного сигнала от интересующей нас поверхности, разобьем всю поверхность на элементарные площадки dA и проинтегрируем по ним

$$P_{FS}(t) = \frac{\lambda^2 \langle \sigma \rangle}{(4\pi)^3} \int \frac{P(t - \frac{2r}{c})G^2(r, \theta, \varphi)}{r^4} dA$$

Из геометрии задачи (см. рис. 21) задачи можно найти связь между азимутальным углом θ , полярными углами $\varphi, \tilde{\varphi}$ и отклонением антенны от положения надира

$$\cos \theta = \frac{\cos \xi + \frac{\rho}{h} \sin \xi \cos(\tilde{\varphi} - \varphi)}{\sqrt{1 + (\frac{\rho}{h})^2}} \quad (17)$$

Поскольку боковые лепестки по мощности гораздо меньше главного лепестка, то пренебрежем ими и положим диаграмму направленности равной следующей функции

$$G(\theta) = G_0 e^{-\frac{2}{\gamma} \sin^2 \theta}, \quad (18)$$

где G_0 – нормировочный множитель, γ – коэффициент, зависящий от ширины диаграммы направленности антенны.

Подставим (17) в (18), учтем, что элемент поверхности можно записать как $dA = \rho d\rho d\psi$ и тогда интеграл преобразуется к виду (учтем, что

$$r = \sqrt{h^2 + \rho^2}$$

$$P_{FS}(t) = \frac{G_0^2 \lambda^2}{(4\pi)^3 h^4} \int_0^\infty \int_0^{2\pi} \frac{P(t - \frac{2h}{c} \sqrt{1 + \varepsilon^2})}{(1 + \varepsilon^2)^2} \sigma^o(\psi) \cdot \exp \left\{ -\frac{4}{\gamma} \left[1 - \frac{\cos^2 \xi}{1 + \varepsilon^2} \right] + b + a \cos(\tilde{\varphi} - \varphi) - b \sin^2(\tilde{\varphi} - \varphi) \right\} d\varphi \rho d\rho,$$

$$\text{где } \varepsilon = \frac{\rho}{h}, \quad a = \frac{4\varepsilon}{\gamma} \frac{\sin 2\xi}{(1+\varepsilon^2)}, \quad b = \frac{4\varepsilon^2}{\gamma} \frac{\sin^2 \xi}{(1+\varepsilon^2)}.$$

Браун в своей работе [8] вычислил этот интеграл и показал, что он равен

$$P_{FS} = \frac{G_0^2 \lambda^2 c}{4(4\pi)^2 L_p h^3} \cdot \frac{\sigma^o(\psi)}{(\frac{ct}{2h})^3} \cdot \exp \left\{ -\frac{4}{\gamma} \left[\cos^2 \xi - \frac{\cos 2\xi}{(\frac{ct}{2h})^2} \right] \right\} \cdot (1 + \varepsilon^2)^2 \sum_{n=0}^{\infty} \frac{(-1)^n \Gamma(n + \frac{1}{2})}{\sqrt{\pi} \Gamma(n+1)} \left[\left(\frac{ct}{2h} \right)^2 - 1 \tan \xi \right]^n \cdot I_n \left(\frac{4}{\gamma} \sqrt{\frac{c\tau}{n}} \sin 2\xi \right), \text{ при } t \geq 2h/c$$

и $P_{FS} = 0$ при $t < 2h/c$

Это выражение можно упростить, переходя к новому времени $\tau = t - 2h/c$, где $2h/c$ – время задержки между излучением и приемом сигнала. Учитывая, что в масштабах спутниковой альtimетрии $\frac{c\tau}{h} \ll 1$, получим

$$P_{FS}(\tau) = \frac{G_0^2 \lambda^2 c \sigma^o(\psi_0)}{4(4\pi)^2 L_p h^3} \exp \left\{ -\frac{4}{\gamma} \sin^2 \xi - \frac{4c}{\gamma h} \tau \cos 2\xi \right\} \cdot \sum_{n=0}^{\infty} \frac{(-1)^n \Gamma(n + \frac{1}{2})}{\sqrt{\pi} \Gamma(n+1)} \left[\sqrt{\frac{c\tau}{h}} \tan \xi \right]^n I_n \left(\frac{4}{\gamma} \sqrt{\frac{c\tau}{h}} \sin 2\xi \right) \text{ при } \tau \geq 0 \quad (19)$$

и $P_{FS} = 0$, при $\tau < 0$

Рассмотрим теперь отдельно сумму из уравнения (19). Если переобозначить $Y = \frac{4}{\gamma} \sqrt{\frac{c\tau}{h}} \sin 2\xi$, то сумма примет вид

$$I_0(Y) \cdot \left\{ 1 + \sum_{n=1}^{\infty} \frac{(-1)^n \Gamma(n + \frac{1}{2})}{\sqrt{\pi} \Gamma(n+1)} \cdot \frac{I_n(Y)}{I_0(Y)} \left[\frac{\gamma Y}{8 \cos^2 \xi} \right]^n \right\}$$

Поскольку $Y \ll 1$ и $\xi \ll 1$, то множитель $\left[\frac{\gamma Y}{8 \cos^2 \xi} \right]^n$ будет быстро сходиться к нулю. Следовательно сумму n слагаемых мы можем приближенно заменить лишь одним слагаемым при $n = 0$.

$$P_{FS}(\tau) = \frac{G_0^2 \lambda^2 c \sigma^o(\psi_0)}{4(4\pi)^2 L_p h^3} \exp \left\{ -\frac{4}{\gamma} \sin^2 \xi - \frac{4c}{\gamma h} \tau \cos 2\xi \right\} \cdot I_0 \left(\frac{4}{\gamma} \sqrt{\frac{c\tau}{h}} \sin 2\xi \right) \text{ при } \tau \geq 0 \quad (20)$$

Можно прибегнуть к ещё одному упрощению и разложить в ряд функцию Бесселя в уравнении (20):

$$I_0(\zeta) = \sum_{n=0}^{\infty} \left(\frac{\zeta^2}{4} \right)^n \cdot \left(\frac{1}{n!} \right)^2, \text{ где } \zeta = \frac{4}{\gamma} \sqrt{\frac{c\tau}{h}} \sin 2\xi$$

поступим также, как Amarouche [9] в своей статье и оставим только два первых члена разложения, которые, в свою очередь, совпадают с разложением экспоненты

$$I_0(\zeta) \approx 1 + \frac{\zeta^2}{4} = e^{\frac{\zeta^2}{4}}$$

Тогда функция P_{FS} примет вид

$$P_{FS}(\tau) = A \exp \left\{ -\frac{4}{\gamma} \sin^2 \xi \right\} \exp \left\{ -\frac{4c}{\gamma h} (\cos 2\xi - \sin^2 2\xi) \tau \right\} \quad (21)$$

Зная отклик плоской поверхности на сигнал с радиовысотомера мы мо-

жем перейти к вычислению отклика на взволнованную морскую поверхность. С точки зрения физики, различия будут в том, что теперь не вся поверхность может отражать сигнал в нужном направлении, а только зеркально ориентированные площадки на поверхности. С радиотехнической точки зрения, взволнованная морская поверхность является линейным фильтром с импульсной переходной характеристикой $q(t)$. Тогда отклик взволнованной поверхности можно вычислить выполняя свертку

$$P_{RS}(t) = q(t) * P_{FS}(t) \quad (22)$$

Результирующая форму импульса будет сверткой функции отклика на плоскую поверхность P_{FS} и функции распределения зеркальных площадок q

$$P(\tau) = \int_{-\infty}^{\infty} q(\tau - \tilde{\tau}) P_{FS}(\tilde{\tau}) d\tilde{\tau}$$

Стоит заметить, что из-за малого значения параметра ξ и масштаба задачи $(c\tau)/h \ll 1$ функция P_{FS} изменяется гораздо медленнее функции плотности зеркальных точек q , а значит можно записать приближенное равенство

$$P \approx \begin{cases} P_{FS}(0) \int_0^{\infty} q(\tau - \tilde{\tau}) d\tilde{\tau}, & \text{при } \tau < 0 \\ P_{FS}(\tau) \int_0^{\infty} q(\tau - \tilde{\tau}) d\tilde{\tau}, & \text{при } \tau > 0 \end{cases}$$

Поскольку корреляция между соседними площадками мала, а количество площадок велико, то согласно центральной предельной теореме [3], мы можем

положить распределение q гауссовым

$$q(\tau) = \frac{1}{\sqrt{2\pi\left(\frac{2\sigma_s^2}{c}\right)}} \exp\left\{-\frac{\tau^2}{2\pi\left(\frac{2\sigma_s^2}{c}\right)^2}\right\} \quad (23)$$

Напоследок, следует учесть, что наш приемник (радиолокатор) тоже является линейной системой с некоторой импульсной характеристикой $P_T(t)$. Поэтому необходимо к уравнению (22) добавить ещё одну свертку. В работе [10] показано, что отклик радиолокатора имеет гауссовую форму

$$P_T(\tau) \sim \exp\left\{-\frac{\tau^2}{2\sigma_p^2}\right\}, \quad (24)$$

где $\sigma_p = \frac{1}{\sqrt{2\ln 2}}T \approx 0.425T$, T – временнбое разрешение антенны, по порядку величины обычно равное $T \sim 3$ нс.

Тогда, результирующий импульс будет равен

$$P(t) = P_{FS}(t) * q(t) * P_T(t) \quad (25)$$

В свертке (25), с учетом (21), (24) и (23), каждый множитель представляет собой экспоненту. Свертку от трех экспонент не трудно посчитать аналитически.

Получаем окончательную формулу для сигнала на приемнике радиолокатора

$$P(t) = Ae^{-v}(1 + \operatorname{erf}(u)), \text{ где} \quad (26)$$

$$A = A_0 \exp \left\{ \frac{-4}{\gamma} \sin^2 \xi \right\},$$

$$u = \frac{t - \alpha \sigma_c^2}{\sqrt{2} \sigma_c},$$

$$v = \alpha \left(t - \frac{\alpha}{2} \sigma_c^2 \right),$$

в которых

$$\alpha = \delta - \frac{\beta^2}{4} = \frac{4}{\gamma} \cdot \frac{c}{h} \left(\cos 2\xi - \frac{\sin^2 2\xi}{\gamma} \right),$$

$$\gamma = \frac{\ln 2}{2} \sin^2 \theta_{-3dB},$$

$$\sigma_c^2 = \sigma_p^2 + \frac{\sigma^2}{c^2},$$

$\xi \ll 1$ – малое отклонение антенны от надира, θ_{-3dB} – ширина диаграммы направленности антенны на уровне $-3dB$, h – высота радиолокатора над поверхностью земли, c – скорость света в вакууме, σ^2 – дисперсия высот взволнованной морской поверхности.

График функции (26) изображен на рис. 22.

3.4. Восстановление параметров морской поверхности.

Зная зависимость принятого сигнала от параметров взволнованной морской поверхности, мы можем восстанавливать их по форме импульса. Это можно сделать, аппроксимируя форму импульса теоретической формулой и вычисляя из получившейся функции необходимые коэффициенты.

Однако решать подобную задачу для формулы (26) довольно сложно из-за сложной зависимости восстанавливаемых параметров и в их большом количестве. Это может приводить к большим вычислительным ошибкам даже при

большом соотношении сигнал-шум.

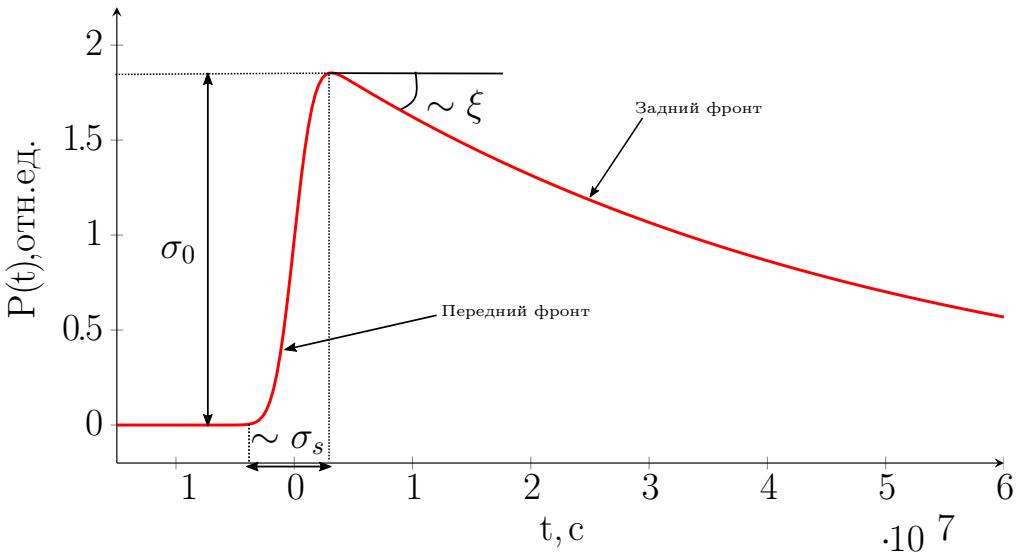


Рис. 22: Качественная форма импульса с обозначением основных параметров.

Поэтому для решения задачи ретрекинга предлагается, использовать менее физичную, но более наглядную запись формулы (26)

$$P(t) = A \exp\left\{S_T\left(t - \frac{\tau}{2}\right)\right\} \left(1 + \operatorname{erf} \frac{t - \tau}{\sigma_L}\right), \text{ где} \quad (27)$$

S_T – коэффициент наклона заднего фронта импульса, τ – эпоха, σ_L – ширина переднего фронта импульса,

Поиск наклона заднего фронта Формула (27), хороша тем, что можно найти некоторые коэффициенты, не прибегая к сложным методам оптимизации. После прохождения пика импульса, функция ошибок становится медленно меняющейся функцией и можно записать приближенное равенство

$$P(t) = 2A \exp\left\{S_T\left(t - \frac{\tau}{2}\right)\right\}, \text{ при } t > t_{max}, \quad (28)$$

где t_{max} – ордината пика импульса.

Логарифмируя (28)

$$\ln P(t) = \ln 2A + S_T(t - \frac{\tau}{2}) = S_T t + \text{const}$$

мы получаем линейную функцию времени. Значит, построив логарифм формы импульса при $t > t_{max}$ и найдя коэффициент наклона получившейся прямой мы можем найти наклон заднего фронта S_T . Подобная процедура проведена на рис.23

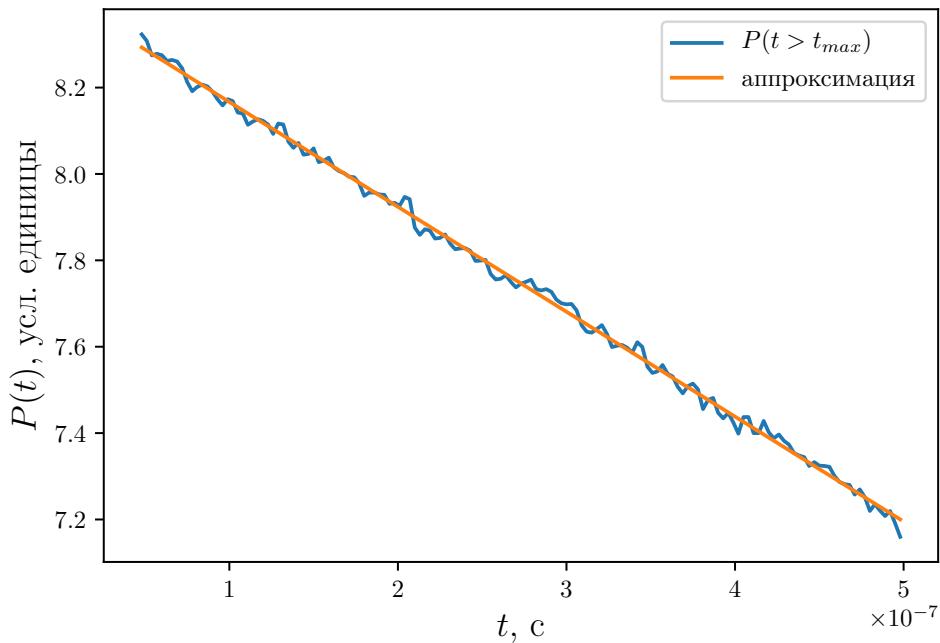


Рис. 23: Поиск наклона заднего фронта

Поиск ширины переднего фронта Как видно из рис.24, при $t < t_{max}$ функция ошибок $\text{erf}\left(\frac{t-\tau}{\sigma_L}\right)$ ведет себя быстрее экспоненты, а значит можно написать приближенное равенство

$$P(t) \approx A \left(1 + \text{erf} \frac{t - \tau}{\sigma_L} \right) \quad (29)$$

Аппроксимируя импульс при $t < t_{max}$ формулой (29) мы получим оценку коэффициентов A , τ , σ_L .

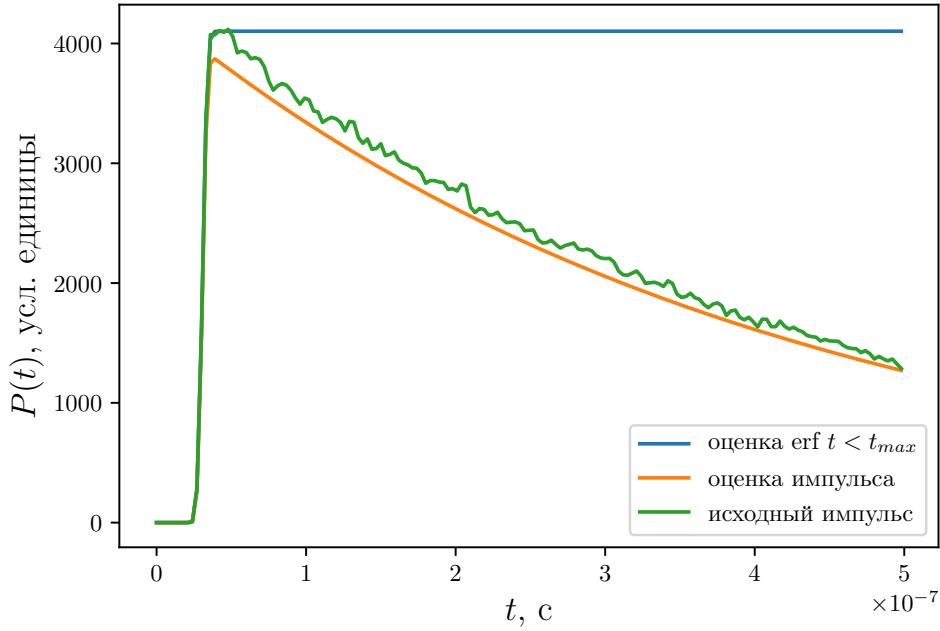


Рис. 24: Поиск ширины переднего фронта исходя из формы функции $\operatorname{erf}\left(\frac{t-\tau}{\sigma_L}\right)$

Имея оценки параметров аппроксимации по различным участкам функции $P(t)$ мы можем использовать формулу (27) для всего импульса

$$P(t) = A \exp\left\{S_T(t - \frac{\tau}{2})\right\} \left(1 + \operatorname{erf}\frac{t - \tau}{\sigma_L}\right).$$

с начальными условиями для параметров A, S_T, τ, σ_L , полученных на предыдущих этапах.

Ознакомиться с реализацией можно в приложении А в листинге ??.

Восстановление параметров поверхности Не сложно найти связь коэффициентов в формуле (26) и (27):

$$\begin{aligned} S_T &= -\alpha, \\ \sigma_L &= \sqrt{2}\sigma_c, \\ \sigma_c^2 &= \sigma_p^2 + \left(\frac{2\sigma_s}{c}\right)^2. \end{aligned} \tag{30}$$

Из соотношений (30) восстанавливается значение дисперсии высот (высоты значительного волнения). Из амплитуды импульса мы можем узнать сечение обратного рассеяния, которое с помощью различных регрессионных моделей позволяет оценить скорость приводного ветра.

3.5. Восстановление параметров модельных поверхностей

Чтобы оценить насколько точно работает выбранный нами алгоритм восстановления параметров поверхности в реальных условиях, смоделируем морскую поверхность с известными параметрами такими как: скорость приводного ветра, дисперсия высот, дисперсия наклонов и применим алгоритм ретрекинга к этой поверхности. Так мы сможем получить необходимые сведения о стабильности и точности алгоритма восстановления и внести в него корректировки, если это будет необходимо.

В разделе 3 был описан процесс получения отраженного импульса от известной модельной поверхности. Программная реализация представлена в приложении А.

Для получения импульса с соотношением сигнал/шум таким же, как у качественного трека с радиовысотомера требуется просуммировать отраженную мощность от нескольких миллионов зеркальных точек, что требует длительного времени вычислений. В разделе 2.3.3 был описан способ более быстрых подсчетов благодаря вычислению на графическом процессоре, что позволяет посчитать отраженный импульс за конечное время.

На рис.25 представлены отраженные от модельных поверхностей импульсы при разных скоростях ветра. Для получения качественного отраженного импульса потребовалось около $8 \cdot 10^6$ зеркальных точек.

Применяя к импульсам на рис.25 алгоритм ретрекинга, получаем следу-

ющие восстановленные параметры:

Таблица 1: Восстановление высоты значительного волнения \tilde{h} для модельных поверхностей с известной высотой значительного волнения h для импульсов с рис. 25.

h , м	0.83	1.36	5.14
\tilde{h} , м	0.65	1.49	4.9

где h – высота значительного волнения, известная из моделирования поверхности, \tilde{h} – высота значительного волнения, полученная по форме отраженного импульса. Высота значительного волнения h вычисляется априорно исходя из дисперсии высот σ^2 моделируемой поверхности [11]

$$h = 4\sqrt{\sigma^2}$$

Сразу стоит оговориться, что в точность измерений высоты значительного волнения определяется в первую очередь временным разрешением приемной антенны радиолокатора. Это означает, что при малой длительности переднего фронта отраженного импульса (малой высоте значительного волнения) абсолютная ошибка может достигать 40 см, при высоте значительного волнения в 80 см.

С ростом высоты волнения длительность импульса нелинейно увеличивается и уже при высоте волнения в 1 м ошибка будет гораздо меньше и составлять 10 % от величины волнения.

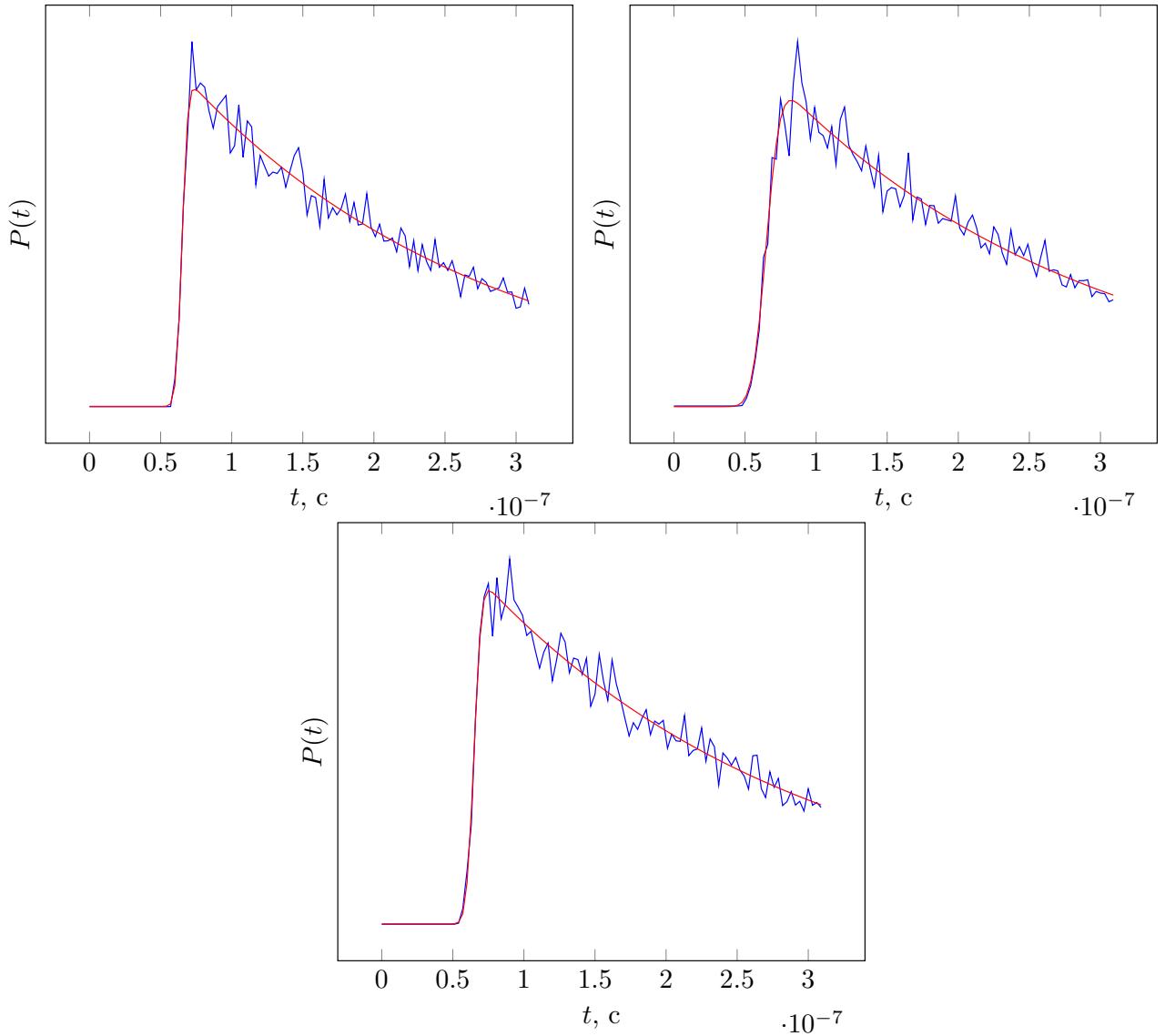


Рис. 25: Отраженный импульс от моделируемых морских поверхностей при разных скоростях ветра (a) $U_{10} = 5 \text{ м/с}$, (b) $U_{10} = 10 \text{ м/с}$, (c) $U_{10} = 15 \text{ м/с}$ для спутника находящегося на высоте в 1000 км над уровнем моря, длительности импульса $\tau = 3 \text{ нс}$ и частоты дискретизации антенны 1 нс

3.6. Восстановление параметров морской поверхности поверхности по данным радиовысотомера

Теперь применим алгоритм ретрекинга к реальным данным и восстановим высоту значительного волнения по данным радиовысотомера космической миссии Jason-3.

Данные находятся в открытом доступе на сайте Национального Центра

Океанографических данных (**NODC**).

Данные представлены как в необработанном виде, поэтому перед процедурой ретрекинга необходимо усреднить полученные импульсы, а также учесть траекторию спутника и не проводить вычисления для тех данных, где отсутствовала морская поверхность.

На рис. 26 представлены обработанные формы импульсов.

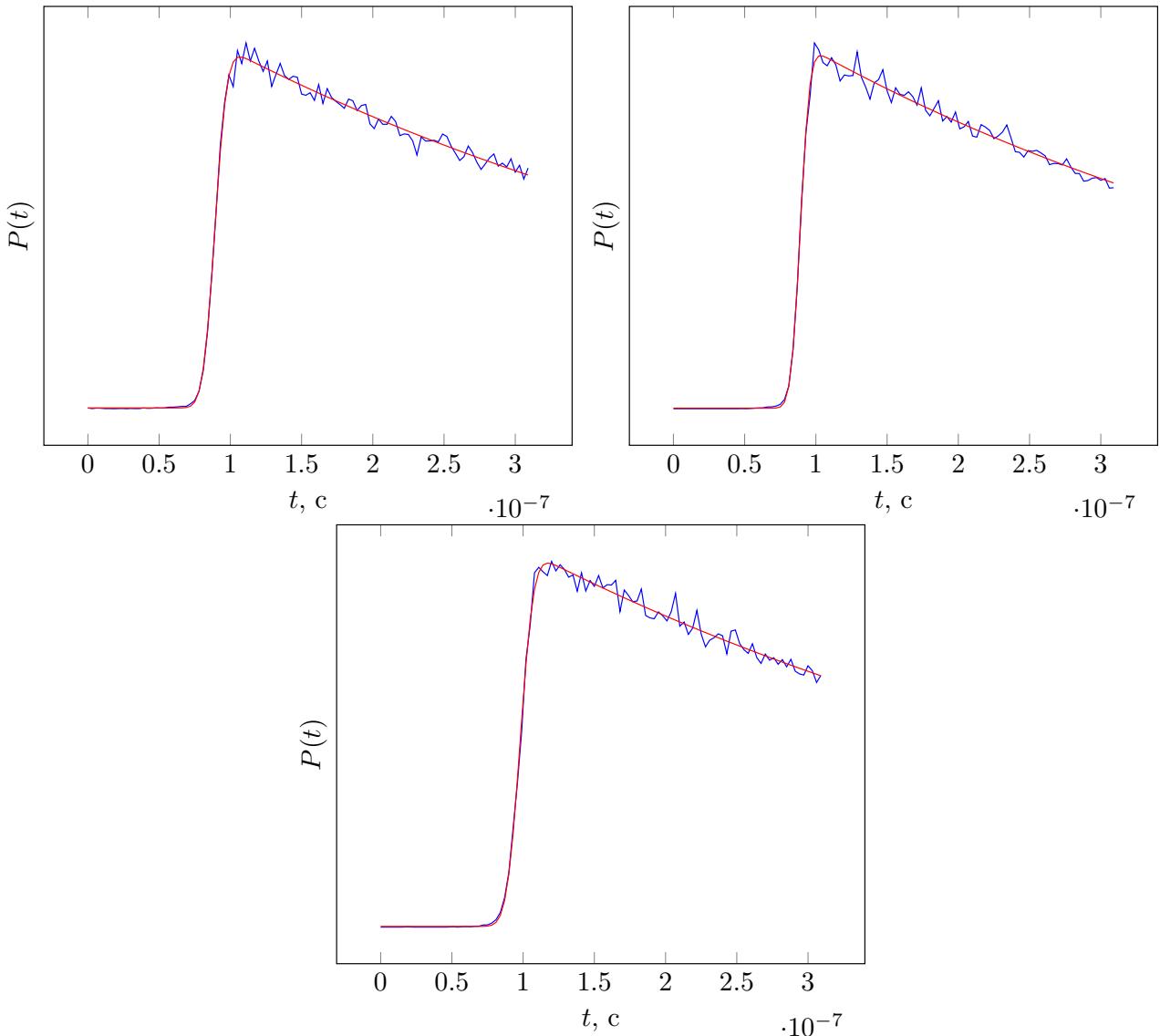


Рис. 26: Форма отраженного импульса в зависимости от времени, полученного с радиовысотомера космической миссии Jason-3.

По импульсам с рис. 26 мы можем восстановить значение высоты значительного волнения. В табл. 2 приведены результаты работы предложенного

алгоритма в сравнении с результатами работы штатного алгоритма MLE3. Для накопления статистических данных, один из треков космической лаборатории Jason-3 был обработан полностью и для каждой доступной координаты была вычислена высота значительного волнения.

Таблица 2: Сравнение работы штатного алгоритма для Jason-3 и предложенного в разделе 3.4 алгоритма для импульсов на рис. 26

h , м	0.937	0.699	1.075
\tilde{h} , м	0.931	0.703	1.081

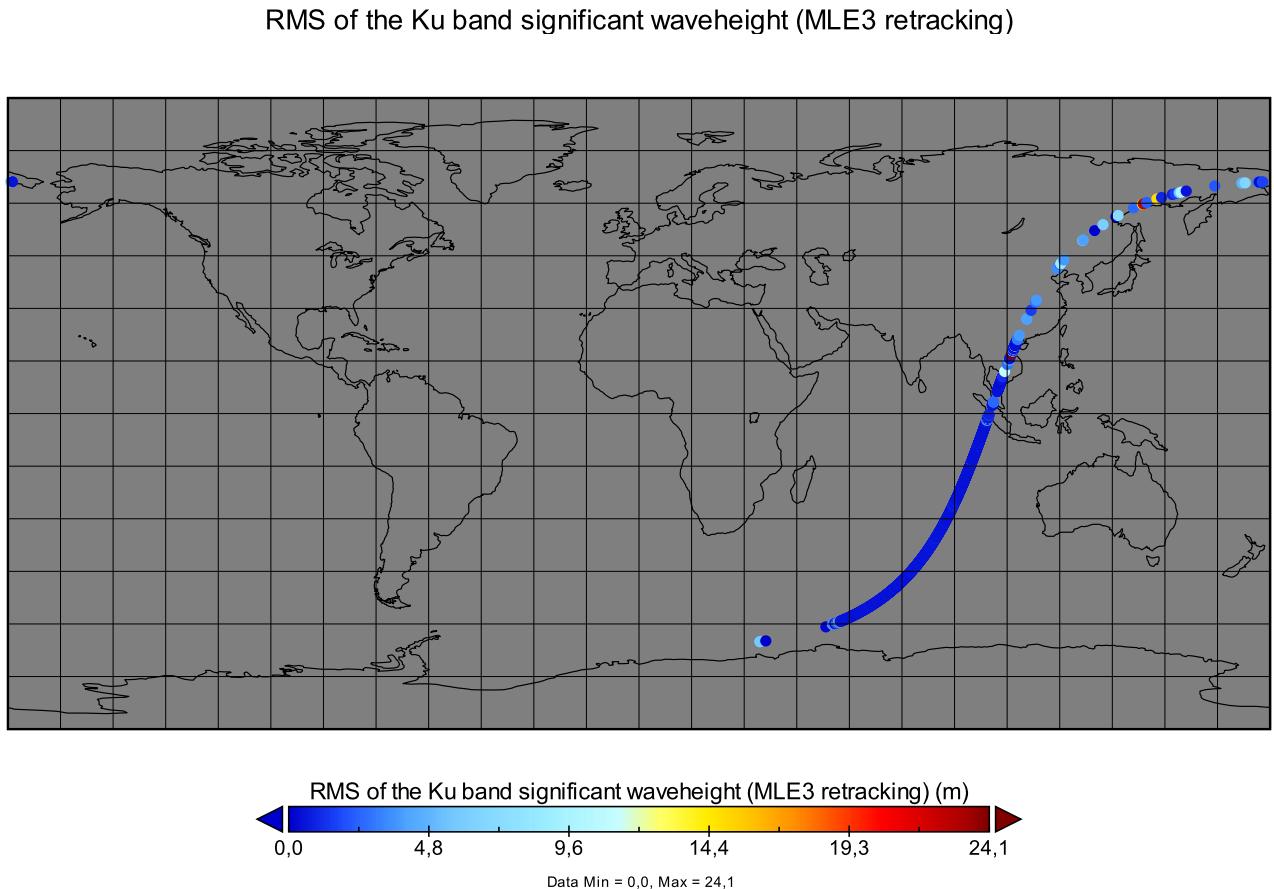


Рис. 27: Изображение траектории спутника Jason-3. Цветом обозначена восстановленная высота значительного волнения.

По оценкам, полученным из табл. 2 и табл. 1, а также обработки данных с рис. 27. можем сделать вывод о точности восстановленных данных:

модель представленной взволнованной морской поверхности хорошо симулирует реальные измерения, а значит может использоваться в проведении экспериментов по отладки радиоизмерительной аппаратуры и методов дистанционного зондирования. При этом в численном эксперимента мы можем с легкостью менять частоту дискретизации приемной антенны, длину волны, высоту полета спутника и его угол зондирования и таким образом находить оптимальную схему измерений для каждой конкретной задачи.

4. Заключение

В данной работе проводился численный эксперимент на взволнованной морской поверхности, в этот эксперимент входили следующие этапы:

1. моделирование морского волнения
2. моделирование отраженного с орбитального радиолокатора импульса на модельной поверхности
3. применение к отраженному импульсу алгоритма восстановления данных морского волнения и оценка точности используемого алгоритма восстановления

Так же был предложен метод по моделированию поверхности не гармоническими функциями, как это делают обычно, а троихоидальными функциями. Это метод позволяет учесть экспериментально известный факт, что у морской поверхности площадь впадин немногого превосходит площадь гребней. Подобный эффект важен в дистанционном зондировании, поскольку его учет повышает точность численного эксперимента. В разделе 2.4 получены основные статистические свойства, а также представлена связь характеристик заостренной поверхности с обычной.

На модельной поверхности был рассчитан отраженный с орбитального радиолокатора импульс и оценена точность алгоритмов восстановления данных морского волнения. Благодаря модели заостренной поверхности удалось увеличить точность моделирования и добиться относительной погрешности восстановления высоты значительного волнения порядка 12 % (см. предыдущий раздел), что делает модельный эксперимент по точности сравнимым с реальными измерениями, а значит делает пригодным для модельную поверхность для проведения численных экспериментов для отладки измерительной аппаратуры.

Список литературы

- [1] Лонг-Хиггинс М. С. Статистический анализ случайно движущейся поверхности. — Москва : Иностранная литература, 1962. — С. 112–230.
- [2] Караев К., Каневский М. Численное моделирование поверхностного волнения и дистанционное зондирование // Препринт. — № 552. Нижний Новгород : ИПФ РАН, 2000.
- [3] Тихонов В. И. Статистическая радиотехника. — 2 изд. — Москва : Радио и связь, 1982. — С. 293.
- [4] Басс Ф., Фукс И. Рассеяние волн на статически неровной морской поверхности. — Москва : Наука, 1972.
- [5] Fu L.-L., Cazenave A. A handbook of techniques and applications // Satellite altimetry and earth sciences. — Academic Press, 2001. — P. 463.
- [6] Пустовойтенко В., Запевалов А. Оперативная океанография: современное состояние перспективы и проблемы спутниковой альtimетрии. — Севастополь, 2012. — С. 218.
- [7] Pierson W. Perturbation analysis of the navier-stokes equations in lagrangian form with selected linear solutions // Geophys. Res. — 1962. — Vol. 67, no. 8. — P. 3151–3160.
- [8] Brown G. The Average Impulse Response of a Rough Surface and Its Applications // IEEE Transactions on Antennas and Propagation. — 1977. — Vol. 25, no. 1. — P. 67–74.
- [9] Amarouche L., Thibaut P., Zanife O. Improving the Jason-1 Ground Retracking to Better Account for Attitude Effects // Marine Geodesy. — 2004. — Vol. 27, no. 1-2. — P. 171–197.

- [10] Moore R., Williams C. Radar Terrain Return at Near-Vertical Incidence // Proceedings of the IRE. — 1957. — Vol. 45, no. 2. — P. 228–238.
- [11] Лопатухин Л. И., Бухановский А. В. Справочные данные по режиму ветра и волнения Баренцева, Охотского и Каспийского морей. — Санкт-Петербург, 2003.

A. Программный код

Обращаем внимание на то, что весь код в листингах лишен обработчика исключений и условных конструкций для экономии места. Листинги приведены в ознакомительных целях. Работоспособный программный код можно найти на [Github'e](#) автора.

```
1 import configparser
2 import numpy as np
3 from scipy import interpolate, integrate
4
5
6 class Spectrum:
7     def __init__(self, conf_file = None, U10 = None, x = None, band = None, KT = None):
8         # ускорение свободного падения.
9         self.g = 9.81
10
11     self.U10 = U10
12     # скорость ветра на высоте 10 м над уровнем моря.
13     # коэффициент gamma см(. спектр JONSWAP)
14     self.__gamma = self.Gamma(x)
15     # коэффициент alpha см(. спектр JONSWAP)
16     self.__alpha = self.Alpha(x)
17     # координата пика спектра по частоте
18     self.omega_m = self.Omega(x) * self.g / self.U10
19     # координата пика спектра по волновому числу
20     self.k_m = self.k_max( self.omega_m )
21     self.k_edge = {}
22
23     self.k_edge['Ku'] = (
24         68.13 + 72.9 * self.k_m + 12.9 * self.k_m ** 2 * np.log(self.k_m) -
25         -0.396 * np.log(self.k_m) / self.k_m - 0.42 / self.k_m
26     )
27     self.k_edge['C'] = (
28         2.74 - 2.26 * self.k_m + 15.498 * np.sqrt(self.k_m) + 1.7 / np.sqrt(self.k_m) -
29         0.00099 * np.log(self.k_m) / self.k_m ** 2
30     )
```

```

31     self.k_edge['P'] = ( 25.82 + 25.43*self.k_m - 16.43*self.k_m*np.log(self.k_m) + 1.983/np.sqrt(self.
32                                         k_m)
33                                         + 0.0996/self.k_m**1.5
34                                         )
35
36     # массив с границами моделируемого спектра.
37     self.KT = np.array([self.k_m/4, self.k_edge[self.band]])
38
39     # k0 -- густая сетка, нужна для интегрирования и интерполирования
40     self.k= np.logspace(np.log10(self.KT[0]), np.log10(self.KT[-1]), 10**4)

41
42     def get_spectrum(self):
43         # интерполируем смоделированный спектр
44         self.spectrum = self.interpolate()
45         self.sigma_sqr = integrate.quad(self.spectrum, self.KT[0],self.KT[-1])[0]
46         return self.spectrum

47
48     def find_decision(self,omega):
49         P = 9.8 * 1000.0/0.074
50         Q = -1000.0*omega**2/0.074
51         x1= -Q/2.0 + np.sqrt( (Q/2)**2 + (P/3)**3)
52         x2= -Q/2.0 - np.sqrt( (Q/2)**2 + (P/3)**3)
53         k=x1***(1/3)-(-x2)***(1/3)
54         return k

55
56     def det(self,k):
57         #     Функция возвращает Якобиан при переходе от частоты к
58         #     волновым числам по полному дисперсионному уравнению
59         det=(self.g + 3*k**2*0.074/1000 )/(2*np.sqrt(self.g*k+k**3*0.074/1000) )
60         return det

61
62     def k_max(self,omega_max):
63         # k_max -- координата пика спектра
64         k_max=omega_max**2/self.g
65         return k_max

66     def omega_k(self,k):
67         #     Пересчет волнового числа в частоту по полному дисперсионному
68         #     уравнению
69         omega_k=(self.g*k+0.074*k**3/1000)***(1/2)

```

```

69     return omega_k
70
71 def JONSWAP(self,k):
72     if k<=self.k_m:
73         sigma=0.074
74     else:
75         sigma=0.09
76     Sw=(
77         self.__alpha/2*k**(-3)*np.exp(-1.25*(self.k_m/k)**2 )*
78         self.__gamma**(np.exp(- ( np.sqrt(k/self.k_m)-1)**2 / (2*sigma**2) )))
79     )
80
81     return Sw
82
83 # Безразмерный коэффициент Gamma
84 def Gamma(self,x):
85     if x>=20170:
86         return 1
87     gamma = (
88         +5.253660929
89         +0.000107622*x
90         -0.03778776*np.sqrt(x)
91         -162.9834653/np.sqrt(x)
92         +253251.456472*x**(-3/2)
93     )
94     return gamma
95
96 # Безразмерный коэффициент Alpha
97 def Alpha(self,x):
98     if x >= 20170:
99         return 0.0081
100    alpha = np.array( [],dtype = 'float64' )
101    alpha = [
102        +0.0311937
103        -0.00232774 * np.log(x)
104        -8367.8678786/x**2
105        +4.5114599e+300*np.exp(-x)*1e+300*1e+17
106        #        +4.5114599e+17*exp(-x)
107    ]

```

```

108     return alpha[0]
109
110 #Вычисление безразмерной частоты Omega по безразмерному разгону x
111 def Omega(self,x):
112     if x>=20170:
113         return 0.835
114     omega_tilde=(0.61826357843576103
115             + 3.52883010586243843e-06*x
116             - 0.00197508032233982112*np.sqrt(x)
117             + 62.5540113059129759/np.sqrt(x)
118             - 290.214120684236224/x
119     )
120     return omega_tilde
121
122 def spectrum0(self,n,k,spectrum_type = 'Karaev'):
123     if spectrum_type == 'Karaev':
124         power = [0,4,5,2.7,5]
125         if n==0:
126             return self.JONSWAP(k)
127         else:
128             omega0 = self.omega_k(self.limit_k[n-1])
129             beta0 = self.spectrum0(n-1,self.limit_k[n-1]) * \
130                 omega0**power[n]/self.det(self.limit_k[n-1])
131             omega0 = self.omega_k(k)
132             return beta0/omega0**power[n]*self.det(k)
133
134
135 def full_spectrum(self,k,x=20170):
136     # Спектр JONSWAP.
137     # По совместительству, граница моделируемого спектра #0
138     # 0< omega < 1.2*omega_max
139     # См. функции spectrum_{1-4}(k).
140     # limit_{1-4} -- это соответствующие границы
141     self.limit_1 = 1.2
142     self.limit_2 =(
143         + 0.371347584096022408
144         + 0.290241610467870486 * self.U10
145         + 0.290178032985796564 / self.U10
146     )

```

```

147     self.limit_3 = self.omega_k(270.0)
148     self.limit_4 = self.omega_k(1020.0)
149     self.limit_k = np.zeros(4)
150     self.limit_k[0] = self.find_decision(self.limit_1 * self.omega_m)
151     self.limit_k[1] = self.find_decision(self.limit_2 * self.omega_m)
152     self.limit_k[2] = 270.0
153     self.limit_k[3] = 1020.0
154
155     try:
156         full_spectrum = np.zeros(len(k))
157     except:
158         full_spectrum = [0]
159         k = [k]
160
161     for i in range(len(k)):
162         if k[i] <= self.limit_k[0]:
163             full_spectrum[i] = self.spectrum0(0,k[i])
164         elif k[i] <= self.limit_k[1]:
165             full_spectrum[i] = self.spectrum0(1,k[i])
166         elif k[i] <= self.limit_k[2]:
167             full_spectrum[i] = self.spectrum0(2,k[i])
168         elif k[i] <= self.limit_k[3]:
169             full_spectrum[i] = self.spectrum0(3,k[i])
170         else:
171             full_spectrum[i] = self.spectrum0(4,k[i])
172
173
174     def interpolate(self):
175         # Интерполируем наш спектр.
176         # Позволяет не думать над различными размерами массивов при счете
177         full_spectrum = interpolate.interp1d(self.k0,
178                                              self.full_spectrum(self.k0))
179

```

Листинг 1: Вычисление спектра волнения как кусочно-аналитической функции

1

2 import numpy as np

```

3   from numpy import pi
4   from scipy import interpolate,integrate
5   from spectrum import Spectrum
6
7
8   class Surface(Spectrum):
9       def __init__(self,**kwargs):
10
11
12       Spectrum.__init__(self,conf_file, **kwargs)
13       self.spectrum = self.get_spectrum()
14       self.k = np.logspace(np.log10(self.k_m/4), np.log10(self.k_edge['Ku']), self.N + 1)
15       self.phi = np.linspace(0,2*np.pi,self.M + 1)
16
17       self.psi = np.array([ [ np.random.uniform(0,2*pi) for m in range(self.M)]
18                           for n in range(self.N) ])
19
20
21       print('Вычисление высот...')
22       self.A = self.amplitude(self.k)
23       self.F = self.angle(self.k,self.phi)
24
25       print('Вычисление полных наклонов...')
26       self.A_slopes = self.amplitude(self.k,method='s')
27       self.F_slopes = self.angle(self.k,self.phi,method='s')
28
29       print('Вычисление наклонов x...')
30       self.A_slopesxx = self.amplitude(self.k,method='xx')
31       self.F_slopesxx = self.angle(self.k,self.phi,method='xx')
32
33       print('Вычисление наклонов y...')
34       self.A_slopesyy = self.amplitude(self.k,method='yy')
35       self.F_slopesyy = self.angle(self.k,self.phi,method='yy')
36
37
38   def B(self,k):
39       def b(k):
40           b=(
41               -0.28+0.65*np.exp(-0.75*np.log(k/self.k_m))

```

```

42         +0.01*np.exp(-0.2+0.7*np.log10(k/self.k_m))
43     )
44     return b
45 B=10**b(k)
46     return B
47
48 def Phi(self,k,phi):
49     # Функция углового распределения
50     phi = phi -self.wind
51     normalization = lambda B: B/np.arctan(np.sinh(2* (pi)*B))
52     B0 = self.B(k)
53     A0 = normalization(B0)
54     Phi = A0/np.cosh(2*B0*(phi) )
55     return Phi
56
57
58 def angle(self,k,phi,method='h'):
59     M = self.M
60     N = self.N
61     if method =='h':
62         Phi = lambda phi,k: self.Phi(k,phi)
63     elif method == 'xx':
64         Phi = lambda phi,k: self.Phi(k,phi)*np.cos(phi)**2
65     elif method == 'yy':
66         Phi = lambda phi,k: self.Phi(k,phi)*np.sin(phi)**2
67     else:
68         Phi = lambda phi,k: self.Phi(k,phi)
69
70     integral = np.zeros((N,M))
71     for i in range(N):
72         for j in range(M):
73             integral[i][j] = np.trapz( Phi( phi[j:j+2],k[i] ), phi[j:j+2])
74     amplitude = np.sqrt(2 *integral )
75     return amplitude
76
77 def amplitude(self, k,method='h'):
78     N = len(k)
79     if method == 'h':
80         S = self.spectrum

```

```

81     else:
82         S = lambda k: self.spectrum(k) * k**2
83         integral = np.zeros(k.size-1)
84         for i in range(1,N):
85             integral[i-1] = integrate.quad(S,k[i-1],k[i])[0]
86             amplitude = np.sqrt(2 *integral )
87         return np.array(amplitude)

1
2 import numpy as np
3 import math
4 from numba import cuda
5 # Мои классы
6 from surface import Surface
7 from data import Data

8
9
10
11
12 TPB=16
13 # Ядро устройства для быстрых параллельных вычислений на GPU
14 # ans [0] -- поверхность высот
15 # ans [1] -- поверхность уклонов
16 # ans [2] -- поверхность уклонов по X
17 # ans [3] -- поверхность уклонов по Y
18 @cuda.jit
19 def kernel(ans, x, y, k, phi, A, F, psi):
20
21     i,j = cuda.grid(2)
22
23     if i >= x.size and j >= y.size:
24         return
25
26     for n in range(k.size):
27         for m in range(phi.size):
28             kr = k[n]*(x[i]*math.cos(phi[m]) + y[j]*math.sin(phi[m]))
29             tmp = math.cos(kr + psi[n][m]) * A[n] * F[n][m]
30             tmp1 = -math.sin(kr + psi[n][m]) * A[n] * F[n][m]
31             ans[0,j,i] += tmp

```

```

32     ans[1,j,i] += tmp1 * k[n]
33     ans[2,j,i] += tmp1 * k[n] * math.cos(phi[m])
34     ans[3,j,i] += tmp1 * k[n] * math.sin(phi[m])
35
36
37
38 N = 256
39 M = 128
40 wind = 30
41
42 x0 = np.linspace(-Xmax,Xmax, 1024)
43 y0 = np.linspace(-Xmax,Xmax, 1024)
44 x = cuda.to_device(x0)
45 y = cuda.to_device(y0)
46
47
48 # Параметры модели
49 data = Data(random_phases = 0, N = N, M=M, band='Ku',wind=wind, U10=5)
50 data_surface = data.surface()
51 data_spectrum = data.spectrum()
52 # Вычисление распределения по углу, по частоте и счет амплитуд гармоник
53 surface = Surface(data_surface, data_spectrum)
54
55 k = surface.k
56 phi = surface.phi
57 A = surface.A
58 F = surface.F
59 PHI_SIZE = F.shape[1]
60 psi = surface.psi
61
62 # Создание массива типа gpuarray
63 k = cuda.to_device(k)
64 phi = cuda.to_device(phi)
65 A = cuda.to_device(A)
66 F = cuda.to_device(F)
67 psi = cuda.to_device(psi)
68
69
70 # Количество потоков на блок

```

```

71 threadsperblock = (TPB, TPB)
72 # Вычисление необходимого количества блоков и создание двумерной сетки
73 blockspergrid_x = math.ceil(x.size / threadsperblock[0])
74 blockspergrid_y = math.ceil(y.size / threadsperblock[1])
75 blockspergrid = (blockspergrid_x, blockspergrid_y)
76
77
78 surfaces = np.zeros((4, x.size, y.size))
79
80 # Вызов ядра на видеокарте
81 kernel[blockspergrid, threadsperblock](surfaces, x, y, k, phi, A, F, psi)

```

Листинг 2: Реализации CUDA вычислений на языке Python

```

1 import numpy as np
2 class Pulse:
3     def __init__(self, r0 = [0,0,3e6],
4                  gane_width=np.deg2rad(1.5),
5                  c = 3e8):
6         self.c = c
7         self.r0 = r0
8         #gane_width ≡ θ3dB
9         gane_width = np.deg2rad(1.5) # Ширина диаграммы направленности в радианах
10        self.gamma = 2*np.sin(gane_width/2)**2/np.log(2)
11
12    def G(self,theta,G0=1):
13        # G -- диаграмма направленности
14        # theta -- угол падения
15        return G0*np.exp(-2/self.gamma * np.sin(theta)**2)
16
17    def R(self, R):
18        R = np.array(R)
19        if len(R.shape)>1:
20            Rabs = np.sqrt(np.sum(R**2, axis=0))
21        else:
22            Rabs = np.sqrt(np.sum(R**2))
23        return Rabs
24
25    def N(self,n):
26        N = np.sqrt(np.sum(n**2, axis=0))

```

```

27     return N
28
29 def theta_calc(self, r, r0):
30     r = np.array(r)
31     r0 = np.array(r0)
32     R = r - r0
33     theta = R[-1,:]/self.R(R)
34     return np.arccos(theta)
35
36 def theta0_calc(self,r,r0,n):
37     R = r - r0
38     Rabs = self.R(R)
39     theta0 = np.einsum('ij, ij -> j', R, n)
40     theta0 *= 1/Rabs/self.N(n)
41     return np.arccos(theta0)
42
43 def mirror_sort(self,r,r0,n,theta, err = 1):
44
45     index = np.where(theta < np.deg2rad(2))
46     r      = r[:, index]
47     r0     = r0[:, index]
48     n      = n[:, index]
49     theta = theta[index]
50
51     if r.size > 0:
52         is_mirror = r.size
53     else:
54         is_mirror = 0
55
56     return r, r0, n, theta, is_mirror
57
58 def power(self, t, omega ,timp ,R, theta,):
59
60     c = self.c
61     G = self.G
62     tau = R/c
63
64     index = [ i for i in range(tau.size) if 0 <= t - tau[i] <= timp ]
65     theta = theta[index]
       R = R[index]

```

```
66 tau = tau[index]
67 # Путь к поверхности
68 #  $\omega\tau \cos(\theta) = kR$ 
69 E0 = G(theta)/R
70 e0 = np.exp(1j*omega*(t - tau*np.cos(theta)) )
71 # Путь от поверхности
72 E0 = E0*G(theta)/R
73 e0 = np.exp(1j*omega*(tau+ tau*np.cos(theta)) )
74
75
76 E = np.sum(E0*e0)
77 return np.abs(E)**2/2
```