УДК 378:004
DOI: 10.25686/978-5-8158-2474-4-2025-61-70

Метод вычисления полиномиального разложения для высокоточной трассировки лучей в оптических системах

С. В. Ершов[1], В. В. Санжаров[2], В. А. Фролов[1,2], В. А. Галактионов[1]
[1]Институт прикладной математики им. М.В. Келдыша РАН, Москва, Россия
[2]Институт перспективных исследований проблем искусственного интеллекта и интеллектуальных систем МГУ им. М.В. Ломоносова, Москва, Россия

Аннотация. В работе рассматривается проблема точности суммирования полиномов, использующихся в оптике для моделирования асферических поверхностей. При увеличении порядка полиномиальных членов, а также при больших значениях коэффициентов могут возникать ошибки округления, связанные с представлением чисел с плавающей точкой. Накопление таких ошибок, в свою очередь, может приводить к сложностям при использовании численных методов, таких как, например, метод Ньютона в задаче трассировки лучей. Предлагается метод, использующий разложение в ряд Тейлора в заранее выбранных точках моделируемой поверхности. Разработанный метод позволяет значительно уменьшить ошибки округления при вычислении суммы полиномиальных членов и тем самым повысить точность расчетов точки пересечения луча и асферической поверхности.

***Ключевые слова*:** оптика, трассировка лучей, асферические поверхности

Polynomial expansion computation method for high-precision ray tracing in optical systems

S. V. Ershov[1], V. V. Sanzharov[2], V.A. Frolov[1,2], V.A. Galaktionov[1]
[1]Keldysh Institute of Applied Mathematics RAS, Moscow, Russia
[2]IAI Moscow State University, Moscow, Russia

Abstract. In this paper we consider the problem of accuracy of summation of polynomials used in optics for modeling aspherical surfaces. When the order of polynomial terms increases, as well as at large values of coefficients, rounding errors associated with the representation of floating-point numbers may occur. The accumulation of such errors, in turn, can lead to difficulties in using numerical methods such as Newton's method, for example, in case of ray tracing of such surfaces. We propose a method that uses Taylor series expansion at pre-selected points of the modeled surface. The developed method allows us to significantly reduce round-off errors when calculating the sum of polynomial terms and thereby improve the accuracy of calculations of the intersection point of the ray and the aspherical surface.

***Keywords*:** optics, ray tracing, aspheric surfaces

Введение

Один из распространенных способов представления асферических поверхностей в оптических системах представляет собой функцию, содержащую сумму полиномов [Sch19]. Например,

$$f(r^2) = \frac{cr^2}{1+\sqrt{1-(1+\kappa)c^2r^2}} + \sum_{m=1}^{N} a_m \left(\frac{r}{R}\right)^{2m} \quad (1)$$

содержит полиномиальную часть

$$p(\rho^2) = \sum_{m=1}^{N} a_m \cdot (\rho^2)^m, \quad (2)$$

где $\rho \equiv \frac{r}{R}$.

Максимальная степень полинома $2N$ обычно никак не ограничена, и программное обеспечение для проектирования оптических систем позволяет задавать её произвольным образом. Однако увеличение степени может вызывать ошибки вычислений, имеющие осциллирующий характер и сходные с так называемым феноменом Рунге – при интерполяции функции полиномом некоторой степени полином начинает расходится вне заданного интервала, при этом расхождение все больше с увеличением степени полинома.

Добавление членов высокого порядка может вызвать осцилляцию в вычисленных значениях, которая не пропадает при добавлении новых членов [Rayces06]. Эти ошибки суммирования во многом возникают из-за вычислений с конечной точностью и плохо обусловленной природы мономиальных базисов. Это ограничивает точность при прямом вычислении полиномов.

Значения коэффициентов a_m могут иметь высокий порядок, при этом значение всей суммы может быть невелико. На рисунке 1 представлен пример поверхности (первая поверхность слева) с коэффициентами a_m в таблице 1.

Таблица 1. **Пример значений коэффициентов полинома**

a_2	a_3	a_4	a_5	a_6	a_7	a_8
−74.856..	863.472..	−7975.92..	55309.02..	−284929.33..	1085886.42..	−3062225.10..
a_9	a_{10}	a_{11}	a_{12}	a_{13}	a_{14}	a_{15}
6386243.47..	−9791775.99..	10874632.94..	−8496288.36..	4422713.64..	−1375665.94..	193280.54..

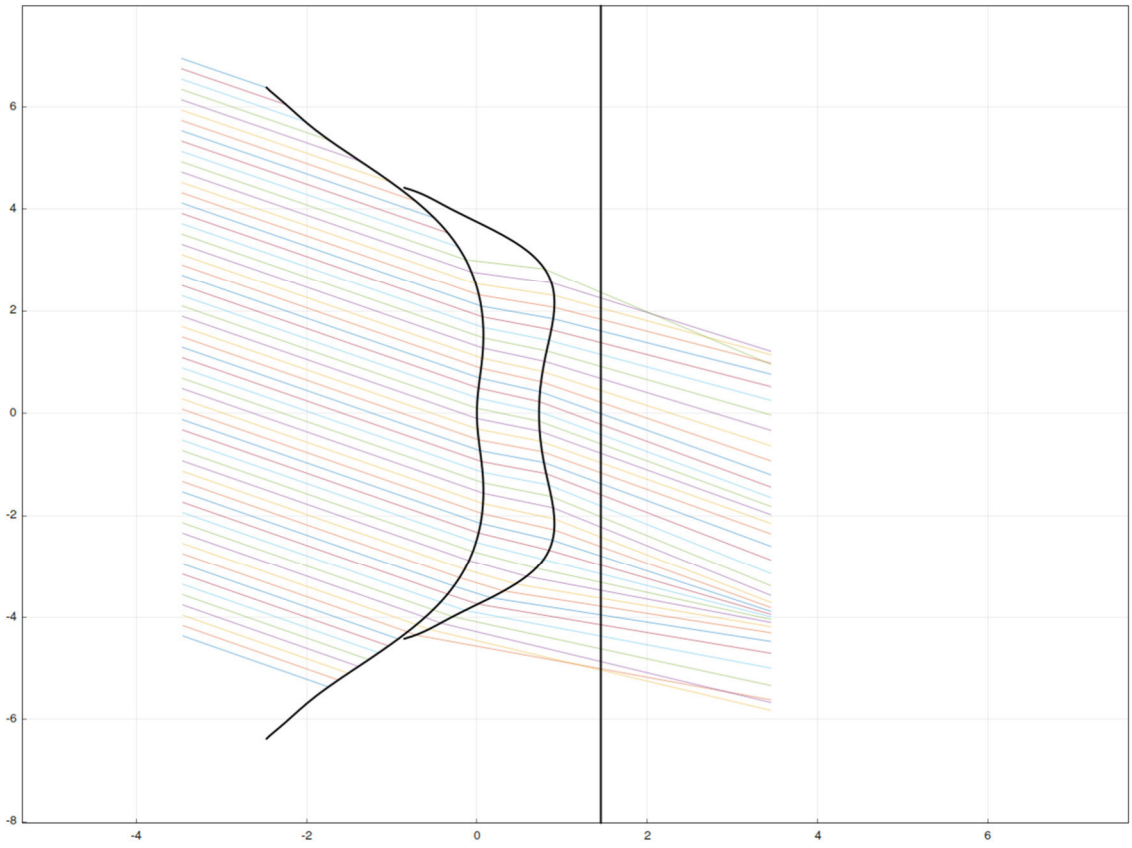

Рисунок 1. Пример линзы с асферической поверхностью с большими значениями коэффициентов полинома (таблица 1)

Вычисления с плавающей точкой двойной точности (т.е. с использованием типа double в языке C) дают точность порядка 10^{-15}. То есть для (2) можно записать

$$p_{CPU}(\rho^2) = \sum_{m=1}^{N} (1+\epsilon_m)\, a_m \cdot (\rho^2)^m = p(\rho^2) + \sum_{m=1}^{N} \epsilon_m\, a_m \cdot (\rho^2)^m,$$

где значение $\epsilon_m \sim 10^{-15}$. При этом значение ϵ_m может быстро и непредсказуемо изменяться для разных ρ and m. То есть значения ϵ_m можно рассматривать как условно «случайные», которые при этом умножаются на большие значения коэффициентов. Например, для $m = 9$ (табл. 1) a_m имеет порядок 10^7, что приводит к тому, что отклонение вычисленного значения суммы от истинного $p_{CPU} - p$ (рис. 2) будет иметь порядок $10^7 \cdot 10^{-15} = 10^{-8}$. С увеличением значения ρ это отклонение будет увеличиваться, т.е. ошибка вычислений ближе к краям оптической поверхности.

Вычисление пересечения луча с поверхностью, заданной выражением (1), производится с помощью метода Ньютона. Ошибка суммирования приводит к тому, что итерационный процесс при достижении точности решения порядка ошибки суммирования начинает осциллировать около корня. В зависимости от значений коэффициентов a_m для моделируемой оптической поверхности и от требуемой точности вычисления пересечений это может привести к тому, что решение не будет найдено.

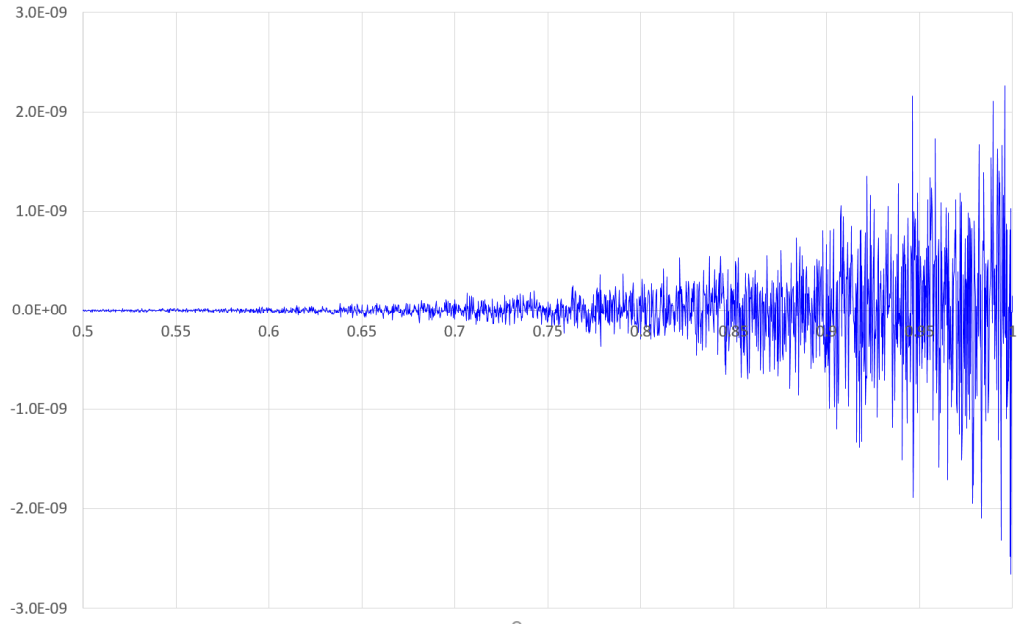

Рисунок 2. Разница между значением суммы (2), вычисленном с двойной точностью (тип double) и с высокой точностью (порядка 250 знаков после запятой) с помощью библиотеки qdpp [QDP]. По вертикальной оси показана ошибка суммирования, по горизонтальной оси отложено значение ρ^2

Рассмотрим существующие подходы к проблеме точности вычислений в трассировке лучей в оптических системах.

Существующие решения

Один из подходов состоит в том, чтобы изменить полиномиальный базис или схему вычислений. В работе [Forbes10] предложен подход с использованием рекуррентных формул для ортогональных полиномов, позволяющий выполнять стабильные вычисления для высоких степеней ортогональных полиномов. Такой прием позволяет значительно уменьшить проблемы, возникающие при наивном суммировании. За счет этого становится возможным использование «обычной» арифметики с одинарной и двойной точностью, имеющей аппаратную поддержку во многих вычислительных системах, для вычислений с большими степенями полиномиальных представлений оптических поверхностей. Недостатком данного подхода является то, что он применим только для ортогональных полиномов, а также увеличение сложности программной реализации по сравнению с обычным суммированием.

Также следует упомянуть об общем решении для проблем с точностью при суммировании на основе использования алгоритмов компенсационного суммирования [Kahan65], которые позволяют уменьшить накопление погрешности при суммировании чисел с плавающей запятой. Но компенсационное суммирование не поможет при суммировании слагаемых значительно различающихся порядков, что вполне возможно для оптических систем. Также такие алгоритмы обычно увеличивают количество арифметических операций, в частности [Kahan65] требует в 4 раза большего числа вычислений. В работе [Bakas24] предлагается использовать разложения в ряд Тейлора в комбинации с типами данных с высокой точностью (авторы рассматривают 2000-битные типы) для устранения вычислительных ошибок, в том числе подобных феномену Рунге. Однако задача трассировки лучей в оптических системах может требовать вычисления миллионов пересечений лучей с поверхностью. Выполнение таких расчетов для 2000-битных типов потребует очень больших вычислительных ресурсов.

Радикальным решением является переход к другому представлению поверхностей, в частности использование сплайнов и кусочной интерполяции. Например, в работе [Morita10] предлагается использовать сетку из квадратичных интерполянтов с вектором нормали, так называемых «патчей», для моделирования асферической поверхности по частям. Метод позволяет получить высокую точность и скорость трассировки лучей. Проблемой как этого подхода, так и других подобных решений, являются сложность построения сетки, трудности при моделировании сложных форм и

производстве полученных линз. Кроме того, альтернативные представления поверхности также требуют вычислений с плавающей точкой двойной точности. В работе [Kimura19] используется схожий подход на основе патчей, для которого приводится сравнение вычислений с плавающей точкой одинарной и двойной точности в оптических расчетах на графическом процессоре (GPU). Эксперименты авторов показали, что только вычисления с двойной точностью позволяют достичь достаточно малого уровня ошибок. Время трассировки при этом возросло более чем в два раза.

В работе [Wu23] предложена модель ошибок округления на каждом шаге трассировки лучей в оптических системах (пересечение с поверхностью и вычисление нового направления луча) на основе IEEE-754 арифметики. Из этой модели авторы доказывают верхние границы («границы ошибок») для трассировки лучей.

Основываясь на этом анализе, в следующей работе [Wu24] авторы предлагают приемы компенсации этих ошибок:

- аппроксимация каждой поверхности «виртуальной касательно плоскостью» при вычислении пересечения;
- ре-проекция лучей в новую систему координат после каждого пересечения;
- вычисление корней для квадратных уравнений через формулу из метода Мюллера;
- нормализация вектора направления после каждого шага;
- вычисления в локальных координатах относительно точки пересечения.

Эти приемы позволяют «сбросить» накопление ошибок на каждом пересечении. На ряде экспериментов в [Wu24] демонстрируется более высокая точность вычислений, чем в коммерческих программных продуктах оптического моделирования. Основным недостатком этой работы является то, что приведенные авторами математические выкладки ориентированы на сферические и конические поверхности. Расширение данного подхода на асферические поверхности и поверхности свободной формы потребует построение другой, ещё более сложной модели. Кроме того, предложенные в [Wu24] способы компенсации ошибок добавляют дополнительные вычисления на каждом шаге трассировки лучей, что может сказаться на производительности. Авторы никак не анализируют затраты на дополнительные вычисления.

В применении трассировки лучей в других областях также исследовалась проблема точности вычислений с плавающей точкой. В работе [Dammertz06] предлагает поиск пересечения в локальных координатах объекта в комбинации с подразбиением ограничивающего параллелепипеда объекта для поиска трехмерного интервала, в котором гарантированно содержится пересечение. Поскольку найденный интервал представлен в локальных координатах объекта, уменьшаются ошибки вычислений с плавающей точкой. В [PBRT3] предлагается использовать специальный тип данных EFloat, который явно хранит интервалы для накопленных ошибок вычислений.

Таким образом, существующие подходы ориентированы на ортогональные полиномы [Forbes10], патчи [Morita10, Kimura19], треугольные сетки [PBRT3], отличные от рассматриваемого представления в нашей работе на основе выражения (1). Работы [Wu23, Wu24] рассматривают более простые поверхности и для применения к асферикам требуют повторения математического вывода. Подходы на основе компенсационного суммирования [Kahan65] не работают, если слагаемые имеют сильно отличающийся порядок. Поэтому представляет интерес решение, которое бы позволило бороться с ошибками вычислений с плавающей точкой для асферических поверхностей, заданных полиномами, подобными (1).

Предлагаемое решение

1. Основная идея

Описанная проблема с суммированием полиномиальной части уравнения (1) возникает при независимом вычислении суммы для близких значений аргумента. Предположим, что после некоторого числа итераций метода Ньютона была достигнута точность порядка 10^{-5}. Мы можем заменить вычисление полинома на вычисление его разложения в ряд Тейлора в текущем значении точки ρ^2 на всех последующих итерациях. В этом случае слагаемые суммы будут достаточно малыми и не будут вызывать осциллирующего поведения. Ясно, что при чрезмерном упрощении, например линеаризации, точность будет недостаточно высокой, поэтому лучше использовать больше членов ряда Тейлора.

Для представленного на рисунке 1 и в таблице 1 примера задано 14 коэффициентов полинома, начиная с 4 степени. Выполним его разложение в ряд Тейлора также с 14 членами, используя биномиальную формулу для вычисления новых коэффициентов:

$$p = \sum_{k=1}^{14} b_k \cdot (\rho^2 - \rho_0^2)^k, \tag{3}$$

$$b_k = \sum_{m=k}^{14} \frac{m!}{k!\,(m-k)!} (\rho_0^2)^{m-k} a_m, \tag{4}$$

где ρ_0^2 – новая «базовая точка» или центр разложения.

При вычислении исходной суммы (2) около этой базовой точки, т.е. для ρ^2, достаточно близких к ρ_0^2, коэффициенты в формуле (3) будут быстро уменьшаться, т.е. по факту мы используем только несколько первых членов суммы. Это позволяет устранить осцилляции, вызываемые ошибками округления.

Важно отметить, что полином для близких значений аргумента ρ^2 должен вычисляться для тех же самых значений b_k, так как в формулу (4) входят исходные потенциально большие значения a_m и поэтому $b_k = b_k(\rho_0^2)$ также может осциллировать.

Алгоритм для вычисления коэффициентов b_k достаточно прост и отражает формулу (4).

Алгоритм 1. $Rebase(N, a, \rho_0)$, вычисление коэффициентов b_k

Вход:

N – число коэффициентов, a – массив исходных коэффициентов полинома, ρ_0 – центр разложения

Выход:

b – массив новых коэффициентов

```
1   k ← 0
2   while k < N do
3   |  sum ← a[k], prod ← 1, m ← 1
4   |  while m < N – k do
5   |  |  prod ← prod * (ρ0^2 * (m + k) / m)
10  |  |  sum ← sum + a[1 + k] * prod
    |  |  m ← m + 1
11  |  end
12  |  b[k] = sum
13  |  k ← k + 1
19  end
20  return b
```

Основная идея предлагаемого решения состоит в том, чтобы заменить непосредственное вычисление суммы полиномиальных коэффициентов (2) на вычисление разложения (3). То есть, например, при решении задачи трассировки лучей с помощью метода Ньютона, для вычисления значения функции поверхности, будет использоваться разложение (3) с заранее вычисленными по алгоритму 1 коэффициентами b_k для «базовой точки» ρ_0^2 , ближайшей к текущему аргументу ρ^2.

2. Множество базовых точек

Как было сказано, необходимо вычислять сумму (3) в окрестности базовой точки ρ_0^2. Рассматриваемая поверхность, описываемая уравнением (1), обладает радиальной симметрией относительно оптической оси, поэтому удобно заранее вычислить коэффициенты b_k для некоторого предопределенного множества базовых точек, например:

$$\rho_0^2 = \{0, 0.1, 0.2, \ldots, 0.9, 1.0\}$$

Тогда в процессе трассировки лучей, когда нам необходимо вычислить значение полинома в ρ^2, мы сначала находим ближайшее значение ρ_0^2. Затем берем соответствующие заранее рассчитанные значения коэффициентов b_k и подставляем в выражение (3). При небольших изменениях ρ^2 на последующих итерациях метода Ньютона ближайшее значение ρ_0^2 останется неизменным. Смена

коэффициентов b_kможет потребоваться только в случае, если ρ^2 перейдет среднее значение между двумя базовыми точками. Но если коэффициенты были рассчитаны с использованием арифметики с большой точностью (достаточно «четверной» точности, т.е. 128-битного типа данных, например, на основе библиотеки [QDP]), то это не приведет к каким-либо проблемам.

На рисунке 3 показана разница между вычислением исходного полинома (2) с использованием арифметики большой точности и его разложения (3) для коэффициентов b_k, также вычисленных с использованием арифметики большой точности.

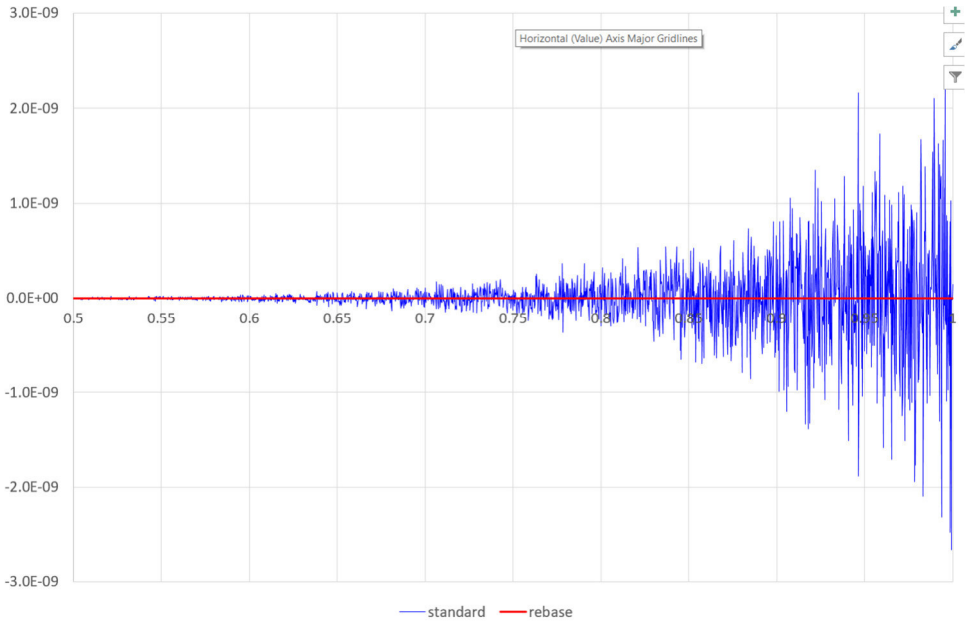

Рисунок 3. Разница между значениями исходного полинома (2), вычисленного со 128-битной арифметикой, и его разложения (3) с коэффициентами, также вычисленными со 128-битной арифметикой. Разница показана красной линией, синим показано значение исходного полинома (2), вычисленного с обычной двойной точностью

Если же коэффициенты b_k вычислить с обычной 64-битной арифметикой (т.е. с использованием типа double), то можно заметить скачки для разницы, которые происходят при значениях ρ_0^2, располагающихся между предрассчитанными значениями, т.е. для $\rho_0^2 = 0.05,0.15,0.25$ и т.д. (рис. 4).

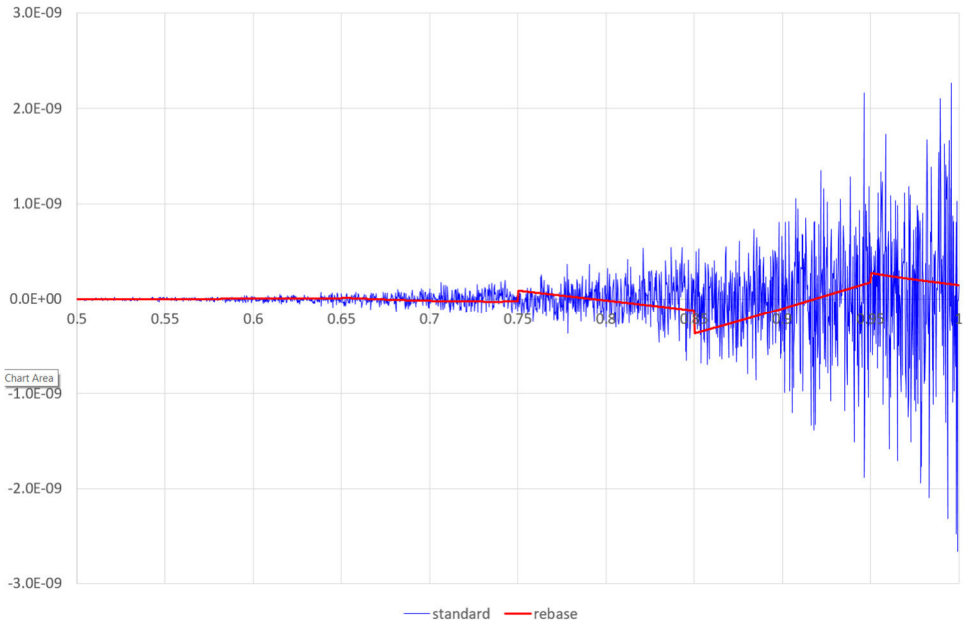

Рисунок 4. Разница между значениями исходного полинома (2), вычисленного со 128-битной арифметикой, и его разложения (3) с коэффициентами, также вычисленными с 64-битной арифметикой. Разница показана красной линией, синим показано значение исходного полинома (2), вычисленного с обычной двойной точностью

В данном примере линия разницы (рис. 4) состоит из сегментов прямых. Мы может вычислить величину ошибки на границах этих сегментов и выполнить вычитание линейной функции, соответствующей тому или иному сегменты. Но в этом случае потребуется вычислить эталонное значение (2) с высокой точностью, хоть и в небольшом числе точек.

Другой вариант состоит в том, чтобы вычислить (3) для двух значений базовой точки ρ_0^2 на концах интервала, в который попадает текущее значение ρ^2 и выполнить линейную интерполяцию между этими значениями, чтобы сделать функцию ошибки гладкой (рис. 5)

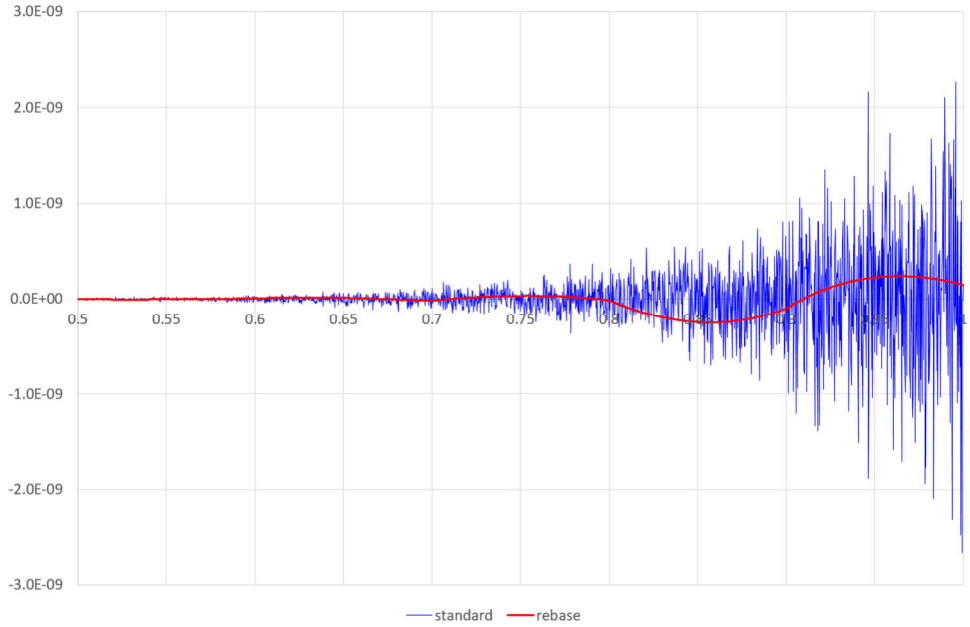

Рисунок 5. Разница между значениями исходного полинома (2), вычисленного со 128-битной арифметикой, и его разложения (3) с коэффициентами, также вычисленными с 64-битной арифметикой с интерполяцией между значениями базовых точек. Разница показана красной линией, синим показано значение исходного полинома (2), вычисленного с обычной двойной точностью

3. Предлагаемый метод

Основу предлагаемого метода составляют выражения (3), (4) и предварительный расчет коэффициентов для множества базовых точек. В экспериментах мы использовали значения для 11 точек $\rho_0^2 = 0{,}0.1{,}0.2{,}\ldots{,}0.9{,}1.0{,}1.1$. Принципиально метод может работать и для другого числа точек. Увеличение числа точек не приносит значимого улучшения результатов.

Таким образом, предлагаемый метод включает в себя два этапа:

1. Предварительный расчет при загрузке или обновлении поверхности:

a) задать множество базовых точек ρ_0^2;

b) вычислить значения коэффициентов b_k для каждой базовой точки из множества с помощью алгоритма 1;

2. Основной этап при поиске пересечения или других вычислений, требующих значений функции поверхности:

a) найти ближайшее значение базовой точки ρ_0^2 для требуемого значения ρ^2;

b) используя значения коэффициентов b_k для найденной базовой точки, вычислить значение полинома в базовой точке с помощью формулы (3).

На рисунке 6 для поверхности, задаваемой коэффициентами из таблицы 1, показаны её сечение (синяя кривая), базовые точки ρ_0^2 (синие маркеры), в которых выполнен предрасчет коэффициентов b_k. Для пучка лучей, идущего под углом в 30° к оптической оси, показаны найденные точки пересечения (красные треугольники).

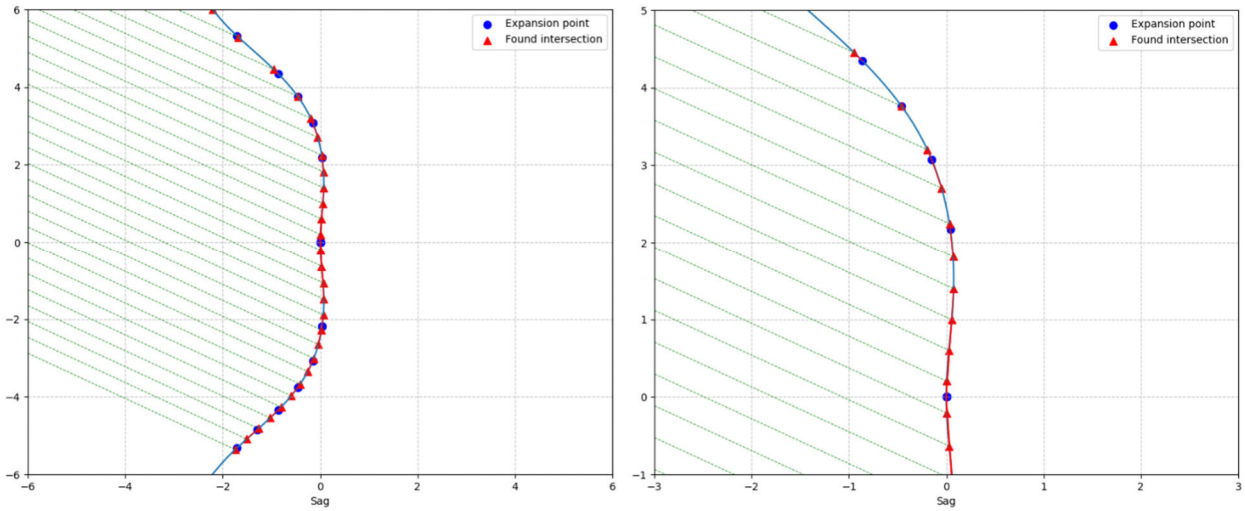

Рисунок 6. Визуализация поверхности, базовых точек разложения ρ_0^2 (синие маркеры) и найденных пересечений (красные треугольники). Справа показан увеличенный участок; найденные пересечения и точки разложения, которые использовались для их расчета, соединены красными линиями

Результаты экспериментов

Мы протестировали предложенный метод в задаче поиска пересечения с асферическими поверхностями вида (1). Для этого в методе Ньютона для вычисления значения функции поверхности мы использовали разложение (3) согласно описанию в предыдущем разделе.

В первом эксперименте были проведены расчеты для поверхности с большими значениями коэффициентов (рис. 1, табл. 1). Выполнялась трассировка пучка из ~1 миллиона лучей (1024 на 1024), в качестве целевого значения точности пересечения acc_{target} было задано 10^{-9} для суммирования полиномов в арифметике с двойной точностью (double) и 10^{-6} с «обычной» плавающей точкой (float). Значение точности пересечения оценивалось как разница между точкой на луче $\overline{p} = \overline{o} + \overline{d}t$ для найденного значения t и значением функции поверхности (1) для этой же точки $f(p.x * p.x + p.y * p.y)$.

Для лучей, у которых точка пересечения попадала внутрь апертуры, а точность пересечения acc лежала в интервале $acc_{target} < acc < 10^{-5}$ для арифметики double и $acc_{target} < acc < 10^{-4}$ для float, считалось, что пересечение найдено с низкой точностью. Для всех лучей, для которых были найдены пересечения, оценивалась средняя точность найденного пересечения. Также среднее число итераций метода Ньютона до достижения условия останова – достигнута требуемая точность или максимальное число итераций (равное 10). Результаты экспериментов приведены в таблицах 2 и 3. Замеры времени вычислений производились на CPU Ryzen 9 5950X.

Таблица 2. **Замеры для пучка 0°, 1024 на 1024 луча**

	Исходный полином		Предложенное разложение, 11 точек ρ_0^2		Предложенное разложение, 5 точек ρ_0^2		Предложенное разложение, 21 точка ρ_0^2	
	double	float	double	float	double	float	double	float
Число лучей с низкой точностью	4169	124752	0	0	0	782	0	0
Среднее значение ошибки для лучей с низкой точностью	1.26e-09	2.16e-05	-	-	-	1.05e-06	-	-
Среднее значение ошибки для всех лучей	5.24e-11	1.15e-05	2.62e-17	1.29e-07	2.96e-17	1.67e-07	2.56e-17	1.23e-07
Среднее число итераций метода Ньютона	2.91	4.96	2.87	2.61	2.87	2.61	2.87	2.59
Время вычислений, мс	59	64	59	55	57	55	59	56

Таблица 3. **Замеры для пучка 30°, 1024 на 1024 луча**

	Исходный полином		Предложенное разложение, 11 точек ρ_0^2		Предложенное разложение, 5 точек ρ_0^2		Предложенное разложение, 21 точка ρ_0^2	
	double	float	double	float	double	float	double	float
Число лучей с низкой точностью	3086	129831	0	12	0	210	0	0
Среднее значение ошибки для лучей с низкой точностью	1.27e-09	2.28e-05	-	1.02e-06	-	1.05e-06	-	-
Среднее значение ошибки для всех лучей	4.29e-11	1.14e-05	3.17e-16	1.18e-07	3.26e-16	1.56e-07	3.21e-16	1.19e-07
Среднее число итераций метода Ньютона	3.88	6.45	3.81	3.65	3.83	3.65	3.82	3.69
Время вычислений, мс	60	70	61	61	61	61	65	62

Из результатов видно, что предложенный метод позволяет достигнуть требуемой точности для всех лучей, для которых найдено пересечение. За счет этого среднее значение точности по всем лучам для двойной точности (double) при использовании предложенного метода оказывается значительно лучше, чем для исходного представления полинома, – на 6 порядков для пучка, параллельного оптической оси (табл. 2), и на 5 порядков для пучка под углом в 30° (табл. 3). За счет использования предложенного метода также быстрее достигается целевая точность и поэтому уменьшается среднее число итераций метода Ньютона. Время вычислений с помощью предложенного незначительно метода отличается от суммирования исходного полинома.

При суммировании полинома в float-арифметике предложенный метод также позволяет достичь целевой точности 10^{-6} в отличие от суммирования исходного представления полинома (2). Средняя точность по всем лучам в этом случае в предложенном методе оказывается лучше на два порядка. Время вычислений для предложенного метода в float-арифметике несколько снижается за счет уменьшения числа итераций метода Ньютона в ~1.7-1.9 раз.

В float-арифметике заметно влияние числа базовых точек ρ_0^2. При пяти точках ещё остаются лучи низкой точности, при 11 точках такие лучи в небольшом количестве (12 лучей, около 0.1 %) остаются для пучка 30°, а при 21 точке нет ни одного луча низкой точности. С увеличением числа точек средняя ошибка по всем лучам также несколько падает, но остается в пределах одного порядка.

Для поверхностей, у которых значения коэффициентов полинома не такие большие, как в таблице 1, и находятся в пределах $\sim 10^4$, влияние предложенного метода не настолько выраженно, но все же присутствует. Были проведены замеры на 24 различных поверхностях с такими коэффициентами и оценена средняя ошибка для всех лучей, для которых были найдены пересечения. Результаты приведены в таблице 4.

Таблица 4. **Замеры для набора поверхностей с коэффициентами < 1e4 для пучка 30°, 1024 на 1024 луча**

	Исходный полином		Предложенное разложение	
	double	float	double	float
Среднее значение ошибки для всех лучей	4.4e-16	3.96e-08	4.2e-16	1.65e-08

Заключение

Предложенный метод разложения полиномов позволяет повысить точность расчетов при моделировании оптических систем, содержащих асферические поверхности вида (1). Для отдельных случаев, когда значения коэффициентов полиномов имеют большие величины, это позволяет избежать непредсказуемых изменений значения суммы, подобных феномену Рунге, при небольших изменениях

точки поверхности, в которой производится расчет. При этом точность расчетов в задаче поиска пересечения луча с такими поверхностями методом Ньютона возрастает на 5-6 порядков в сравнении с обычным суммированием полинома в double-арифметике и на 2 порядка в float-арифметике. Для производства оптических поверхностей такие значения точности (в double-арифметике) избыточны, но за счет увеличения точности расчетов также снижается и количество итераций метода Ньютона, необходимое для достижения целевых значений точности. Кроме того, повышение точности расчетов открывает возможности для разработки алгоритмов в смешенной точности (mixed precision), когда, например, все итерации метода Ньютона, кроме последней, вычисляются в float, а последняя – в double-арифметике.

Использование предложенного метода сохраняет скорость расчетов и требует лишь небольшого объема дополнительной памяти для хранения коэффициентов разложения в 11 точках (для радиально-симметричных поверхностей). Для double-арифметики потенциально можно использовать и меньше точек (до 5). Таким образом, предложенный метод может быть использован в системах оптического моделирования, использующих представление поверхностей с помощью неортогональных полиномов, подобных (1), для повышения точности и стабильности расчетов.

Список литературы

1. [Sch19] Schuhmann, Rainer. "Description of aspheric surfaces". *Advanced Optical Technologies* 8.3-4 (2019): 267-278.

2. [Rayces06] Rayces, Juan L. and Xuemin Cheng. "Numerical integration of an aspheric surface profile". *International Optical Design Conference 2006*. Vol. 6342. SPIE, 2006.

3. [QDP] Quad-double precision library. URL: https://github.com/lhprojects/qdpp

4. [Forbes10] Forbes, G. W. "Robust and fast computation for the polynomials of optics". *Optics express* 18.13 (2010): 13851-13862.

5. [Kahan65] Kahan, W. "Further remarks on reducing truncation errors". *Assoc. Comput. Mach* 8 (1965): 40.

6. [Bakas24] Bakas, Nikolaos. "Taylor Polynomials in a High Arithmetic Precision as Universal Approximators". *Computation* 12.3 (2024): 53.

7. [Kimura19] Hullin, Matthias B., Johannes Hanika and Wolfgang Heidrich. "Polynomial Optics: A construction kit for efficient ray-tracing of lens systems". *Computer Graphics Forum*. Vol. 31, no. 4. Oxford, UK: Blackwell Publishing Ltd, 2012.

8. [Wu23] Chang-Mao, Wu, et al. "High precision ray tracing method for space camera in optical design". *ACTA PHYSICA SINICA* 72.8 (2023).

9. [Wu24] Wu, Changmao, et al. "Mathematical modelling for high precision ray tracing in optical design". *Applied Mathematical Modelling* 128 (2024): 103-122.

10. [Dammertz06] Dammertz, Holger and Alexander Keller. "Improving ray tracing precision by object space intersection computation". *2006 IEEE Symposium on Interactive Ray Tracing*. IEEE, 2006.

11. [PBRT3] Pharr, Matt, Wenzel Jakob and Greg Humphreys. "Physically Based Rendering: From Theory to Implementation". 3d edition, Morgan Kaufmann, 2016.