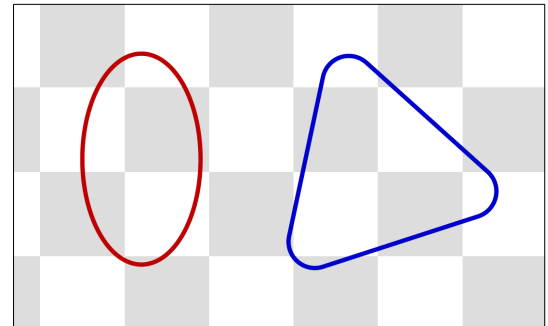


# Определение столкновений движущихся выпуклых объектов в двумерном пространстве (обзор задачи)

**Предисловие.** Определение столкновений движущихся трехмерных объектов – задача, решаемая при планировании движения роботов, в моделировании физических явлений и т.д. Чтобы подойти к решению этой задачи, рассмотрены аналогичные задачи в двумерном пространстве, оказывающиеся более простыми за счет пониженной размерности и меньшего числа переменных. Достигнут результат: разработан алгоритм определения столкновения движущихся выпуклых объектов в двумерном пространстве, в том числе один из двух объектов может быть нестрого выпуклым. Далее наработки будут использованы для решения задач в трехмерном пространстве.

**Применение алгоритма.** Предполагается, что при движении объектов, проводятся несложные для вычисления проверки (например, пересечение описанных окружностей). Если такие проверки не исключают столкновения, то начинает работу описываемый алгоритм.

## Определение выпуклого объекта (convex object).



Строго выпуклый (strictly convex) эллипс - слева и нестрого выпуклый (non-strictly convex) правильный N-угольник с закругленными краями.

## Способы задания геометрической фигуры в двумерном пространстве

В работе используется задание фигур уравнениями в параметрической форме. Такой способ позволяет выразить множество точек периметра фигуры как функцию 1 переменного.

Эллипс с длиной  $\frac{1}{2}$  горизонтальной оси  $a$  и длиной  $\frac{1}{2}$  вертикальной оси  $b$  задается уравнениями:

$$\begin{aligned} x &= a \cos \varphi, \\ y &= b \sin \varphi, \quad -\pi \leq \varphi \leq \pi. \end{aligned} \quad (1)$$

Параметр  $\varphi$  в общем случае не равен углу между положительным направлением оси  $X$  и лучом из начала координат через точку на периметре.

Потребуется направленный наружу вектор-перпендикуляр в точке, длиной 1:

$$\vec{N} = \left( \frac{\cos \varphi}{a}, \frac{\sin \varphi}{b} \right) / \sqrt{\frac{\cos^2 \varphi}{a^2} + \frac{\sin^2 \varphi}{b^2}}. \quad (2)$$

Также фигуры задаются в параметрической форме т.н. кусочно-заданными функциями (piecewise functions). Пример приведен в [Приложение А. Задание фигуры с использованием кусочно-заданной функции](#).

## Способы задания расположения объекта в двумерном пространстве

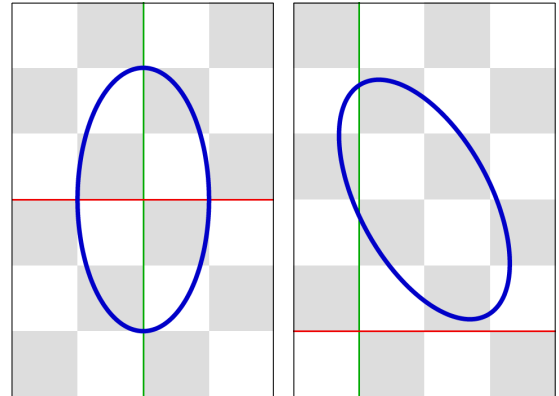
Задается точка - центр объекта  $C$ , и угол поворота объекта  $\text{angle}$ .

Существует способ задания поворота с использованием матрицы  $2 \times 2$ . Это, вероятно, сократит вычисления, поскольку вначале один раз вычисляются косинус и синус угла поворота.

### Поворот точки.

Формула поворота точки  $(x, y)$  на угол  $\theta$  вокруг начала координат:

$$\begin{aligned}\hat{x} &= x \cos \theta - y \sin \theta \\ \hat{y} &= y \cos \theta + x \sin \theta\end{aligned}\quad (3)$$

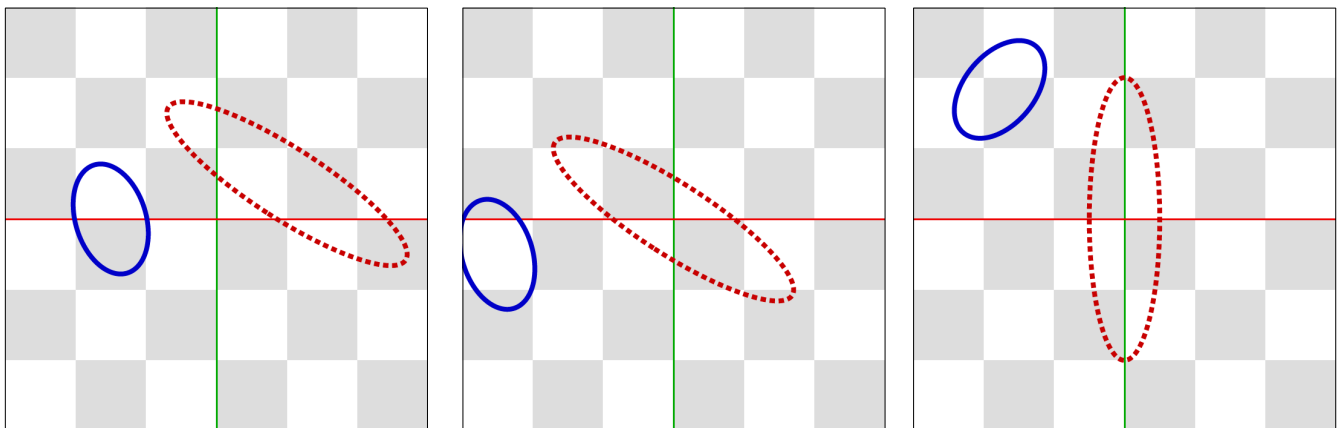


Слева: эллипс  $a=1, b=2$ .

Справа: тот же эллипс, сдвинутый относительно начала координат на  $C=(1, 2)$  и повернутый на  $\text{angle}=0.5$ .

Объекты будут обозначаться: первый объект `object1`, второй `object2`.

**Преобразование координат, такое, чтобы `object2` находился в начале координат не повернутым.**



Слева: `object1` сплошной линией (эллипс  $a_1=0.5, b_1=0.8$ ),  $C_1=(-1.5, 0)$ ,  $\text{angle}_1=0.3$ ; `object2` пунктирной линией (эллипс  $a_2=0.5, b_2=2$ ),  $C_2=(1, 0.5)$ ,  $\text{angle}_2=1$ .

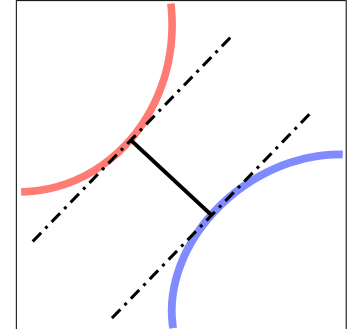
В центре: координаты смещены на  $C_1-C_2$ , чтобы `object2` находился в начале координат.

Справа: поворот на  $-1$  радиан, чтобы `object2` был не повернутым.

## Расстояние между двумя неподвижными объектами

Сначала будет рассмотрена функция расстояния между двумя неподвижными выпуклыми объектами. Эта же функция возникает при вычислении расстояния между движущимися объектами в любой фиксированный момент времени  $t$ .

Наименьшее расстояние между двумя кривыми будет между 2 точками на кривых, таких, что отрезок между этими 2 точками перпендикулярен кривым в точках. Производная расстояния при этом равна нулю.



Реализуется следующий подход: по параметру  $\varphi$  на объекте `object1` строится точка и перпендикуляр в точке, затем по перпендикуляру с обратным направлением находится точка на объекте `object2`. Затем вычисляется расстояние между двумя точками.

**Найти точку на периметре, в которой перпендикуляр длиной 1 будет равен данному вектору длиной 1 с обратным направлением.**

Для эллипса выведена формула:

$$(x, y) = (-a^2 N_x, -b^2 N_y) / \sqrt{a^2 N_x^2 + b^2 N_y^2} . \quad (4)$$

Для нестрого выпуклых фигур существует множество точек на периметре, в которых перпендикуляр длиной 1 будет равен данному вектору с обратным направлением, то есть взаимно-однозначное соответствие отсутствует. Поэтому описываемый подход применим только в случае, если `object2` строго выпуклый.

### Вычисление расстояния по заданному $\varphi$

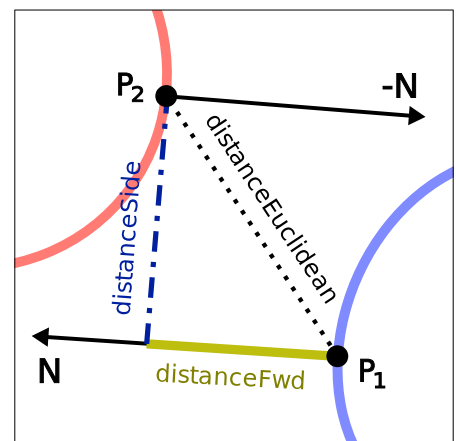
1. По  $\varphi$  найти точку  $P_1$  на периметре `object1` и перпендикуляр в точке  $N$ ;
2. Построить систему координат, в которой `object2` находится в начале координат не повернутым (локальная система координат `object2`);
3.  $P_1$  и  $N$  перевести в локальную систему координат `object2`;
4. По  $-N$  найти точку  $P_2$  на периметре `object2`;
5. Вычислить расстояние между двумя точками  $P_1$  и  $P_2$ .

При вычислении расстояния также вычисляются первая и вторая производные по  $\varphi$ .

Справа изображены три разные расстояния:

- “Обычное” расстояние по прямой на евклидовой плоскости – функция `distanceEuclidean( $\varphi$ )`
- Расстояние “вперед” из точки  $P_1$  вдоль перпендикуляра  $N$  – функция `distanceFwd( $\varphi$ )`
- “Боковое” расстояние между прямой, построенной из  $P_1$  и  $N$ , и точкой  $P_2$  – функция `distanceSide( $\varphi$ )`

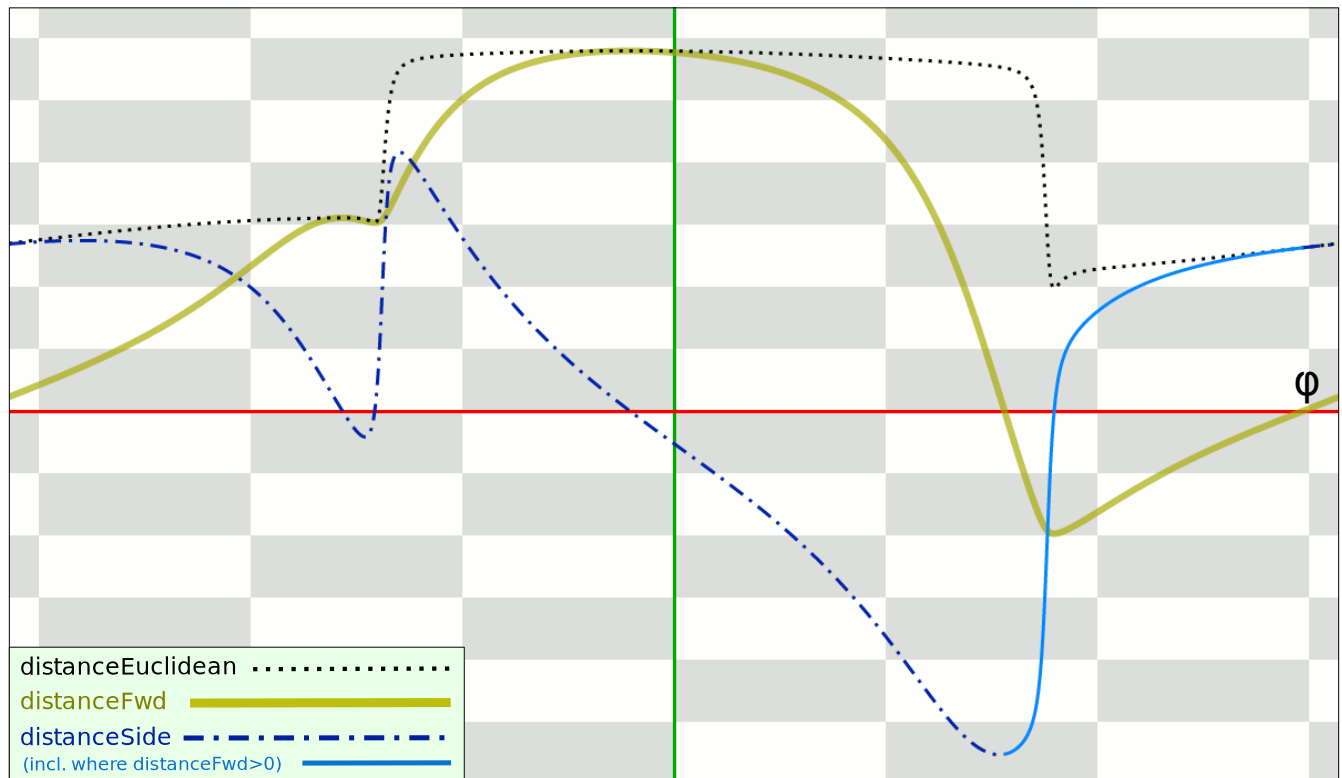
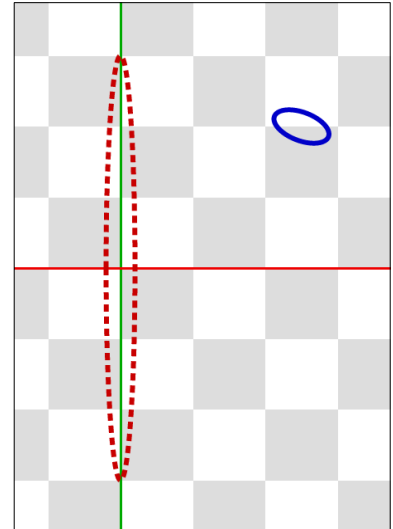
Основной объем действий при вычислениях этих функций один и тот же.



## Анализ функций расстояния

Справа изображены два объекта, ниже – функции расстояния в зависимости от  $\varphi \in [-\pi, \pi]$ . Значения функции  $\text{distanceFwd}(\varphi)$  выведены с обратным знаком для удобства сопоставления.

- Необходимо найти либо минимум  $\text{distanceEuclidean}(\varphi)$ , либо максимум  $\text{distanceFwd}(\varphi)$ , либо ноль  $\text{distanceSide}(\varphi)$ .
- У функций несколько локальных экстремумов. Глобальный экстремум отличается тем, что в нем  $\text{distanceFwd}(\varphi) > 0$ .
- У  $\text{distanceFwd}(\varphi)$  значительно шире интервалы сходимости, чем у  $\text{distanceEuclidean}(\varphi)$ , то есть интервалы, такие, что при начале работы численного метода в любой из точек которых будет найден локальный экстремум.
- Для нестрогих выпуклых объектов необходимо адресовать дополнительные вопросы. См. [Приложение В. Пример с одним нестрогим выпуклым объектом](#).



Реализовано нахождение глобального минимума  $\text{distanceEuclidean}(\varphi)$ . Алгоритм в значительной степени основан на методе Ньютона в оптимизации (Newton's method in optimization). При нахождении локального минимума, в котором  $\text{distanceFwd}(\varphi) < 0$ , продолжается поиск других минимумов методом бисекции на интервалах длиной более заданной, на которых поиск ранее не проводился.

Для нахождения начального значения  $\varphi$ , создается система координат, в которой **object1** находится в начале координат не повернутым, создается вектор  $\vec{N}$  из начала координат в центр **object2** (то есть  $\vec{N}$  - это центр **object2** в локальной системе координат **object1**). В случае эллипса

$$\varphi = \text{atan2}(a N_y, b N_x). \quad (5)$$

Также см. [Приложение С. Эллипсы с соотношением сторон 1:100 и более](#).

## Расстояние между двумя движущимися объектами

В работе алгоритма используются функции зависимости расстояния от  $\varphi$  и  $t$ :  $\text{distanceFwd}(\varphi, t)$  и  $\text{distanceSide}(\varphi, t)$ . Для вычисления этих функций выполняются действия:

1. Вычисление положений `object1` и `object2` в момент времени  $t$ ;
2. Действия, перечисленные в “Вычисление расстояния по заданному  $\varphi$ ”;

Вычисляются как значения функций в точке, так и частные производные 1 и 2 порядка по  $\varphi$  и  $t$ :

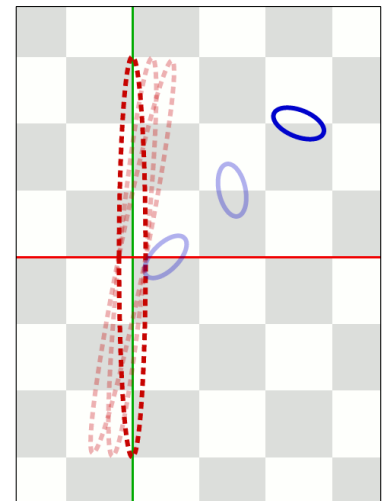
$$\nabla f = \begin{bmatrix} \frac{\partial f}{\partial \varphi} \\ \frac{\partial f}{\partial t} \end{bmatrix}; \quad \nabla^2 f = \begin{bmatrix} \frac{\partial^2 f}{\partial \varphi^2} & \frac{\partial^2 f}{\partial \varphi \partial t} \\ \frac{\partial^2 f}{\partial t \partial \varphi} & \frac{\partial^2 f}{\partial t^2} \end{bmatrix}. \quad (6)$$

Реализован следующий алгоритм:

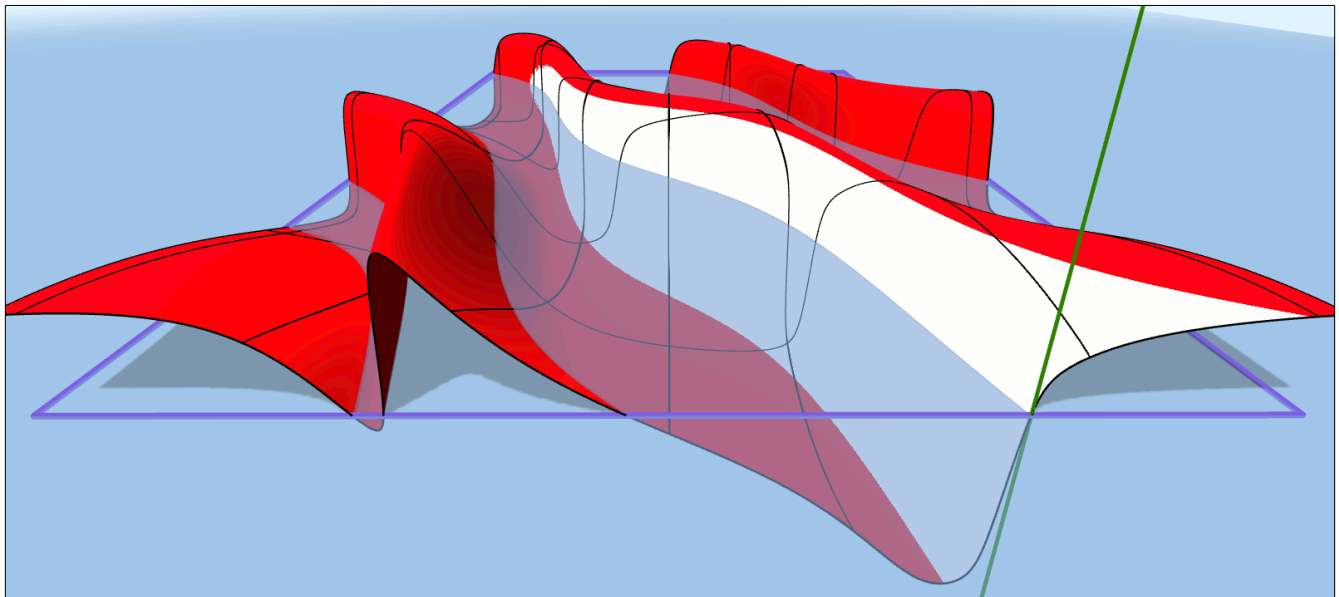
1. Определяется  $\varphi$  и расстояние между двумя объектами при  $t=0$ ;
2. Выполняются шаги алгоритма с увеличением  $t$ , при этом поддерживается и корректируется  $\varphi$ , соответствующее минимальному расстоянию.

Критерии прекращения работы алгоритма:

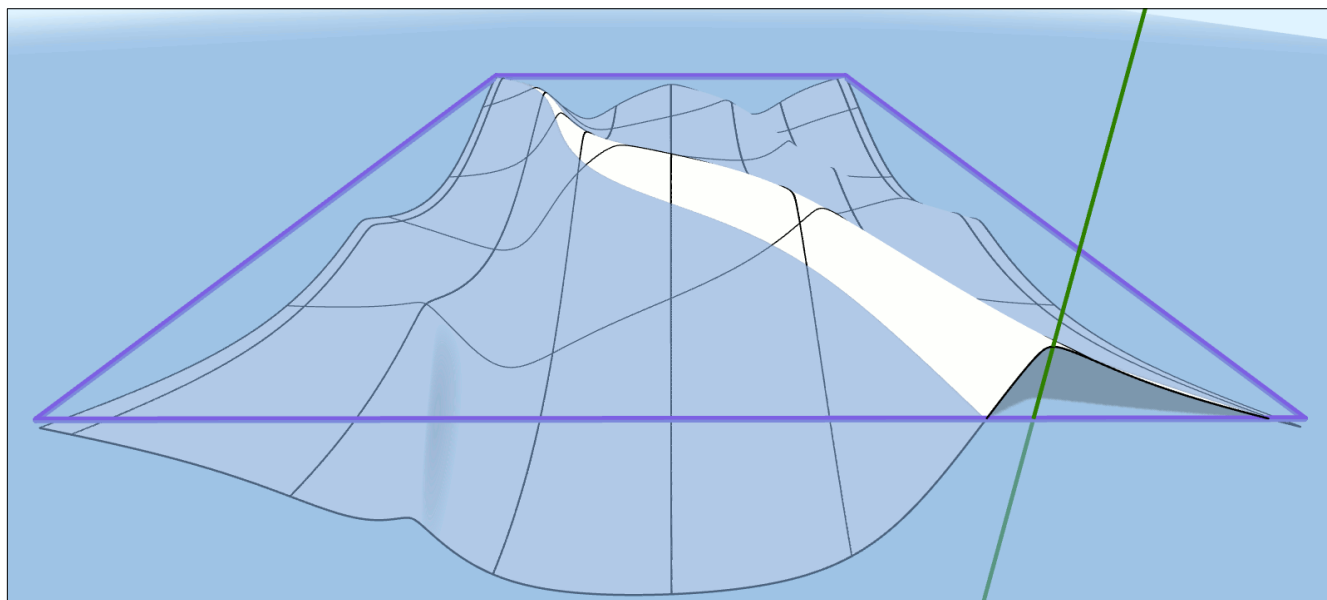
1. Столкновение объектов. Объекты считаются столкнувшимися при достижении заданного расстояния между ними (установлено  $(1.4) \cdot 10^{-6}$ ). Пересечения объектов не допускается.
2. Увеличение расстояния до такого, при котором столкновение точно невозможно, определяемое несложным для вычисления критерием (например, более суммы радиусов описанных окружностей).



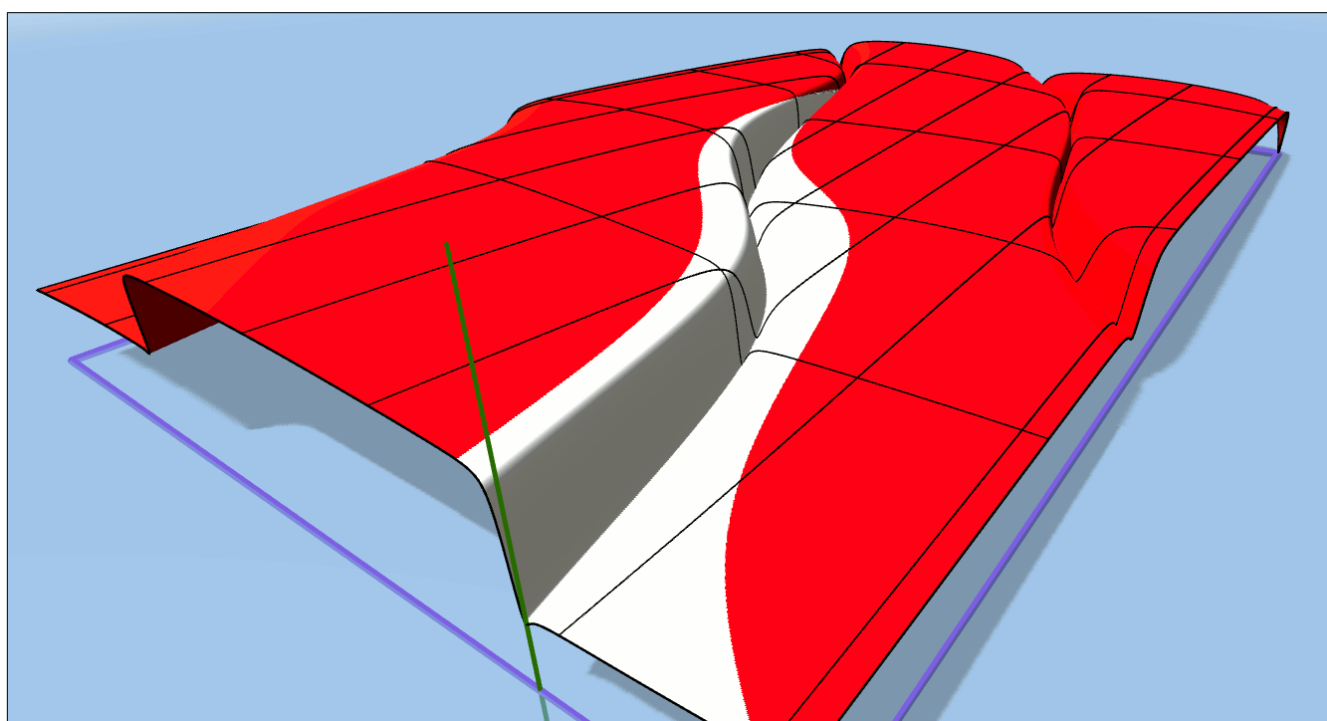
Объекты в моменты времени  $t=0$ ,  $t=1$ ,  $t=2$ .



Функция  $\text{distanceSide}(\varphi, t)$ .



Функция  $\text{distanceFwd}(\varphi, t)$ .



Функция  $\text{distanceEuclidean}(\varphi, t)$ . Работа с этой функцией оказывается менее результативной: численному методу в среднем требуется больше итераций (соответственно больше вычислений), в связи с меньшими интервалами конвергенции.

## Один шаг алгоритма

Эта тема продолжает находиться на этапе разработки, но уже понятны основные моменты.

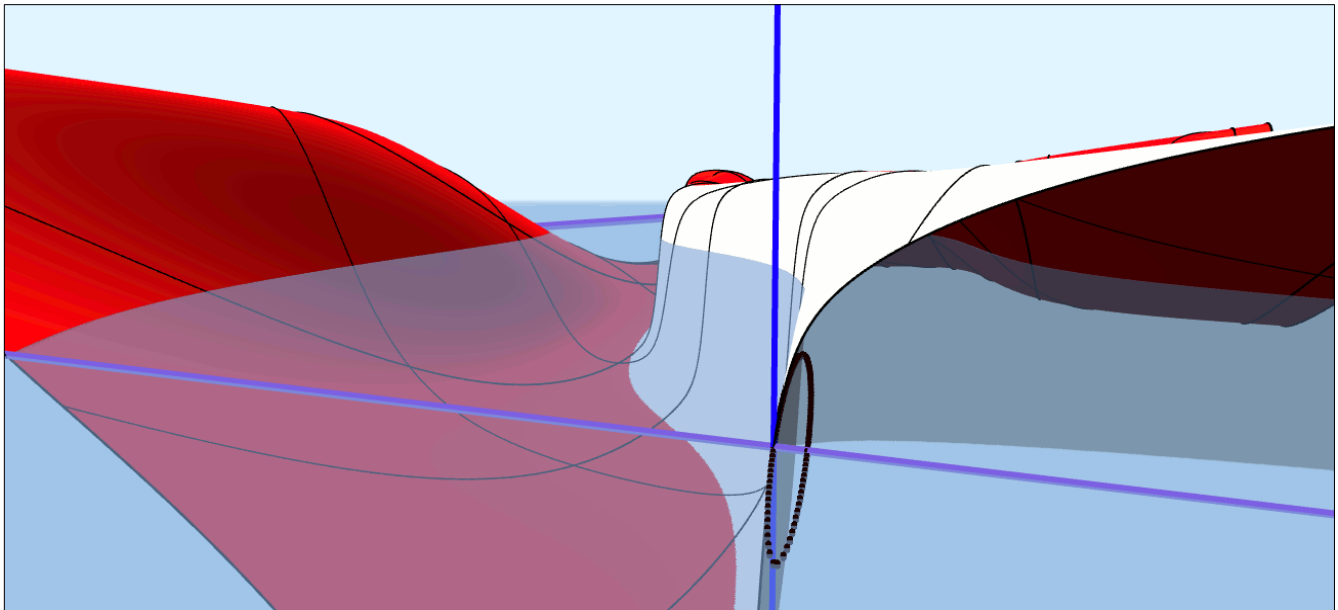
Алгоритм делает шаг вдоль нуля функции  $\text{distanceSide}(\varphi, t)$ . В отличие от движения вдоль экстремумов, это позволяет использовать дополнительно 1 порядок аппроксимации.

В окрестностях точки  $(\varphi_0, t_0)$  функция приближенно представляется многочленом Тэйлора 2 порядка:

$$P = \begin{bmatrix} \varphi - \varphi_0 \\ t - t_0 \end{bmatrix}; \quad f(\varphi, t) = f(\varphi_0, t_0) + P^T \nabla f(\varphi_0, t_0) + \frac{1}{2} P^T \nabla^2 f(\varphi_0, t_0) P. \quad (7)$$

Эквивалентная запись без матриц и операторов:

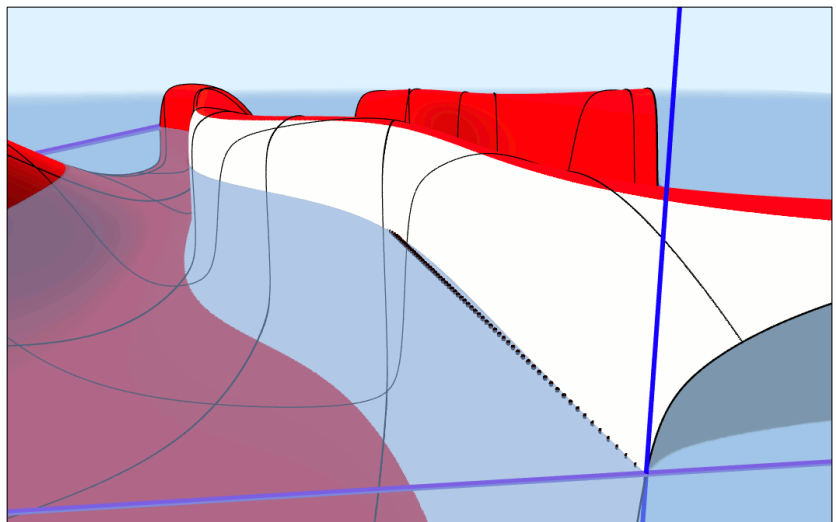
$$f(\varphi, t) = f(\varphi_0, t_0) + \frac{\partial f}{\partial \varphi}(\varphi - \varphi_0) + \frac{\partial f}{\partial t}(t - t_0) + \frac{1}{2} \frac{\partial^2 f}{\partial \varphi^2}(\varphi - \varphi_0)^2 + \frac{\partial^2 f}{\partial \varphi \partial t}(\varphi - \varphi_0)(t - t_0) + \frac{1}{2} \frac{\partial^2 f}{\partial t^2}(t - t_0)^2. \quad (8)$$



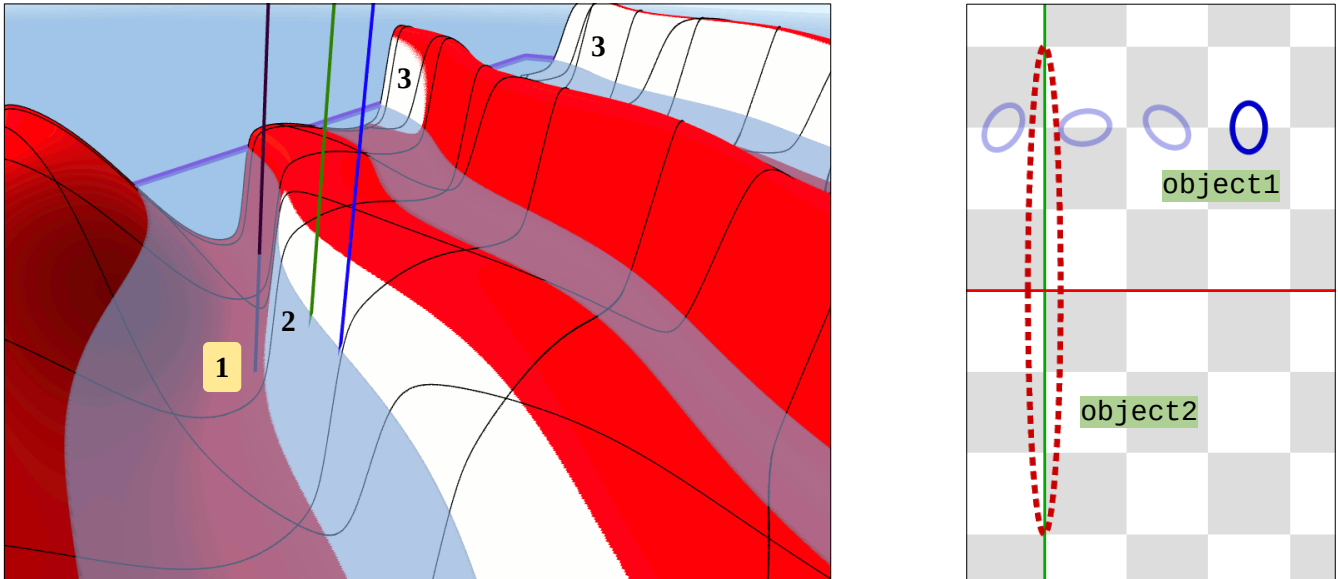
Получается алгебраическая кривая 2 порядка. Кривая, изгибаясь, может начинать поворачивать так, что движение вдоль кривой оказывается в направлении, противоположном увеличению  $t$ .

Это является одним из факторов, ограничивающих длину шага алгоритма.

Для сравнения, справа изображена аппроксимация прямой линией.



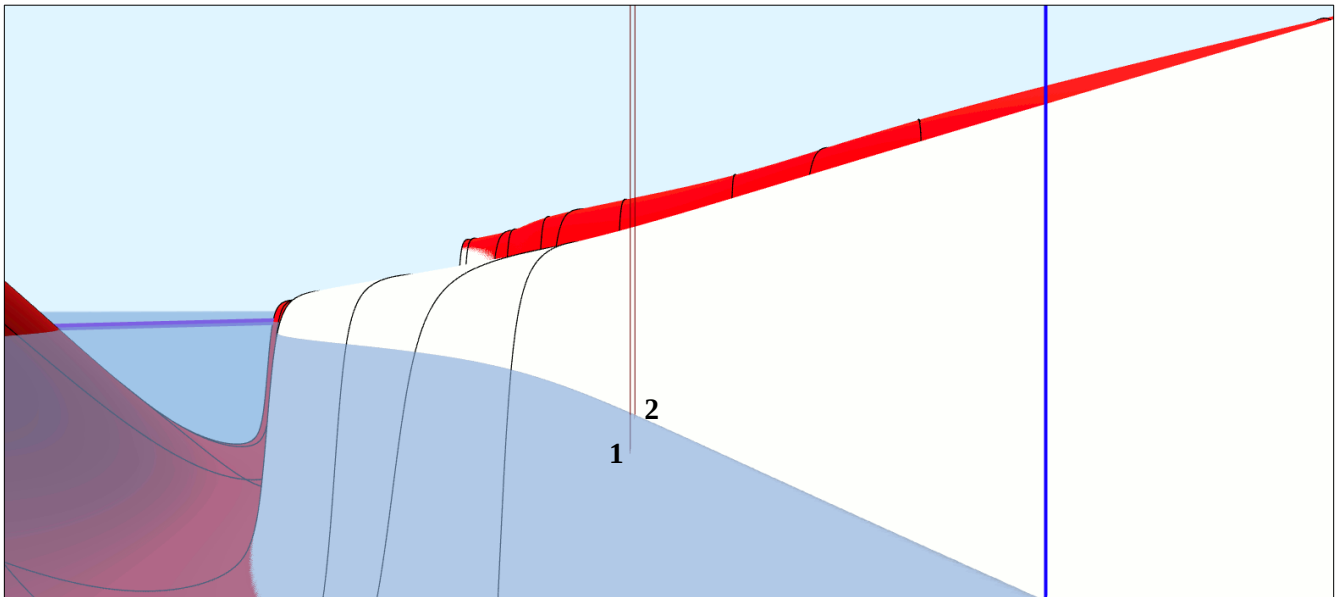
## Backtracking



1. Выполненный шаг оказывается в точке, не соответствующей критериям.
2. Выполнен backtracking.
3. Участок, движение по которому формально возможно, не связанный с участком, по которому идут шаги алгоритма. Он соответствует (см. изображение справа) с движением **object1** после прохождения насквозь через **object2**.

## Коррекция $\varphi$ для соответствия значения функции нулю

После выполнения шага алгоритма, значение функции часто оказывается не равным нулю.



1. Выполнен шаг алгоритма.  $\text{distanceSide}(\varphi, t) \approx -0.2$
2. Выполнена коррекция без изменения  $t$ .  $\text{distanceSide}(\varphi, t) \approx -0.0007$
3. Необходима еще коррекция (не показана). После коррекции  $\text{distanceSide}(\varphi, t) \approx -3 \cdot 10^{-11}$  и шаг алгоритма считается выполненным.

Для коррекции не требуется вычислять заново положения объектов и не нужны производные по  $t$ . Применяется метод Галлея (Halley's method).



## Приложение А. Задание фигуры с использованием кусочно-заданной функции

Фигура **shape1** на изображении справа определяется тремя параметрами  $r_1, r_2, r_3$ . Периметр фигуры непрерывен, функция перпендикуляра в точке в зависимости от  $\varphi$  также непрерывна. Периметр фигуры состоит из нескольких частей, задаваемых разными функциями:

- При  $|\varphi| > \pi/2$  фигура задается теми же уравнениями, что и эллипс при  $a=r_3, b=r_1+r_2$ .
- При  $|\varphi| \leq \pi/2 \vee |\varphi| \geq \arctan(r_2/r_1)$  периметр фигуры является частью окружности с центром в точке  $(0, \pm r_2)$  и радиусом  $r_1$ . Координаты точки на периметре задаются уравнениями:

$$\begin{aligned} d &= r_2 |\sin \varphi| + \sqrt{r_1^2 - r_2^2 \cos^2 \varphi}, \\ x &= d \cos \varphi, \\ y &= d \sin \varphi. \end{aligned} \quad (9)$$

Вектор – перпендикуляр в точке:

$$(N_x, N_y) = \left( \frac{x}{r_1}, \operatorname{sgn}(y) \frac{|y| - r_2}{r_1} \right) \quad (10)$$

Вектор нужен длиной 1. Для этого вычисляется длина вектора  $\text{length} = \sqrt{(N_x^2 + N_y^2)}$  и компоненты делятся на длину.

- При  $|\varphi| < \arctan(r_2/r_1)$  часть периметра является вертикальной прямой линией. Координаты точки:

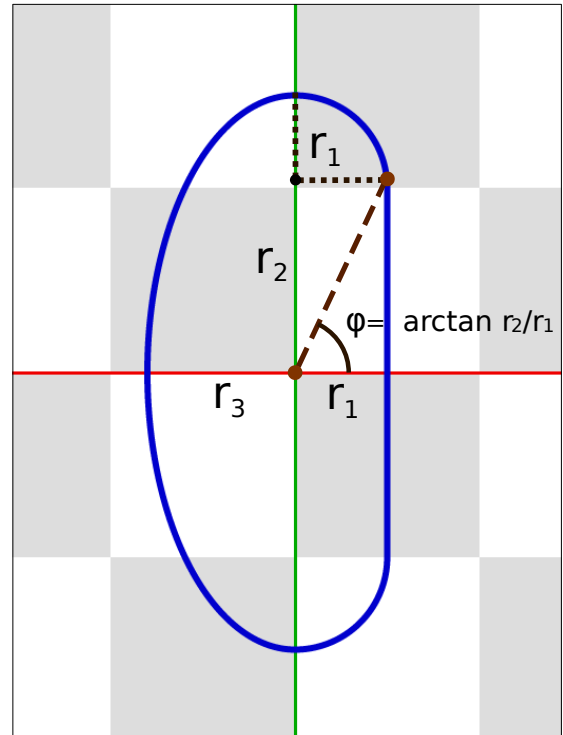
$$(x, y) = (r_1, r_1 \tan \varphi). \quad (11)$$

Вектор – перпендикуляр в точке на этом интервале не зависит от  $\varphi$  и равен:

$$\vec{N} = (1, 0). \quad (12)$$

Из-за того, что  $\vec{N}$  на интервале не зависит от  $\varphi$ , производные компонентов вектора

$\frac{d}{d\varphi} N_x(\varphi), \frac{d}{d\varphi} N_y(\varphi)$  на этом интервале равны нулю.



Фигура **shape1** с параметрами  $r_1=0.5, r_2=1, r_3=0.8$

Итого для последующих действий по определению столкновений с другими объектами нужны:

- Координаты точки в зависимости от  $\varphi$ ;
- Вектор-перпендикуляр длиной 1 в точке в зависимости от  $\varphi$ ;
- Первая и вторая производные вышеупомянутых 4 величин по  $\varphi$ .

Производные вычисляются вместе со значениями функций с применением методов автоматического дифференцирования (automatic differentiation).

На некоторых интервалах производные равны нулю, а также в точках  $\varphi = \pm \arctan(r_2/r_1)$ ,  $\varphi = \pm \pi/2$  часть производных не непрерывны, то есть некоторые из 4 функций не являются непрерывно дифференцируемыми в этих точках.

### Проверка с применением Computer Algebra System (CAS) “Wolfram Alpha”

Пусть  $r_1=0.5$ ,  $r_2=1$ ,  $r_3=0.8$ . Тогда  $\arctan(r_2/r_1) \approx 1.10715$ . Похоже, что в этой CAS нет возможности построить параметрическую кусочно-заданную функцию или одновременно несколько параметрически заданных функций.

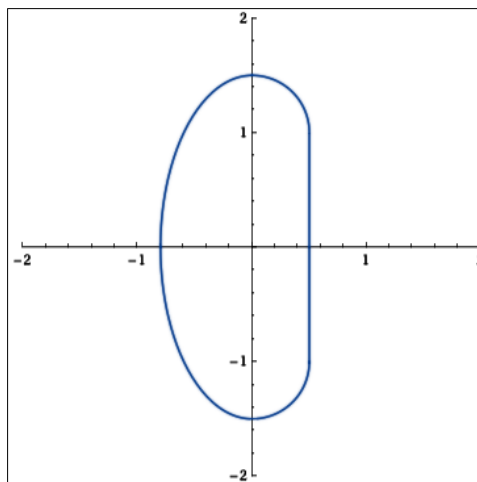
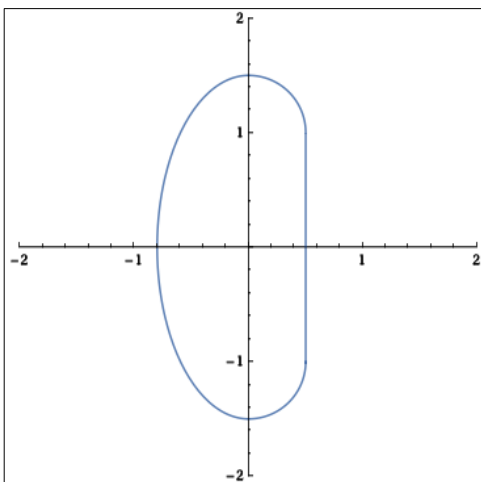
Проведено несколько построений с разными интервалами  $t$ , с указанием одинаковых диапазонов координат и отношения  $x/y=1$ :

```
parametric plot x=0.8 cos t, y=1.5 sin t, {t, -3/2 pi, -pi/2}, plotrange ((-2,2), (-2,2)), AspectRatio 1
```

```
parametric plot x=cos t(|sin t|+sqrt(0.25-cos^2 t)), y=sin t(|sin t|+sqrt(0.25-cos^2 t)), {t, 1.10715, pi/2}, plotrange ((-2,2), (-2,2)), AspectRatio 1
```

```
parametric plot x=cos t(|sin t|+sqrt(0.25-cos^2 t)), y=sin t(|sin t|+sqrt(0.25-cos^2 t)), {t, -pi/2, -1.10715}, plotrange ((-2,2), (-2,2)), AspectRatio 1
```

```
parametric plot x=0.5, y=0.5 tan(t), {t, -1.10715, 1.10715}, plotrange ((-2,2), (-2,2)), AspectRatio 1
```



Снимки экрана с несколькими построениями были скомбинированы с использованием gimp: наложены в несколько слоев, из верхних слоев убран фон при помощи Layer → Transparency → Color to Alpha.

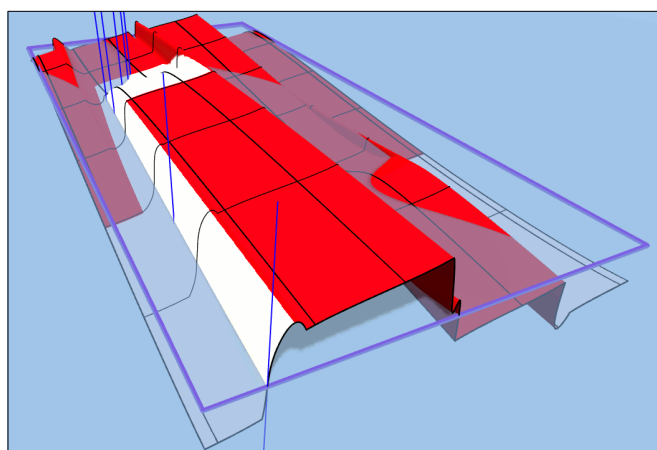
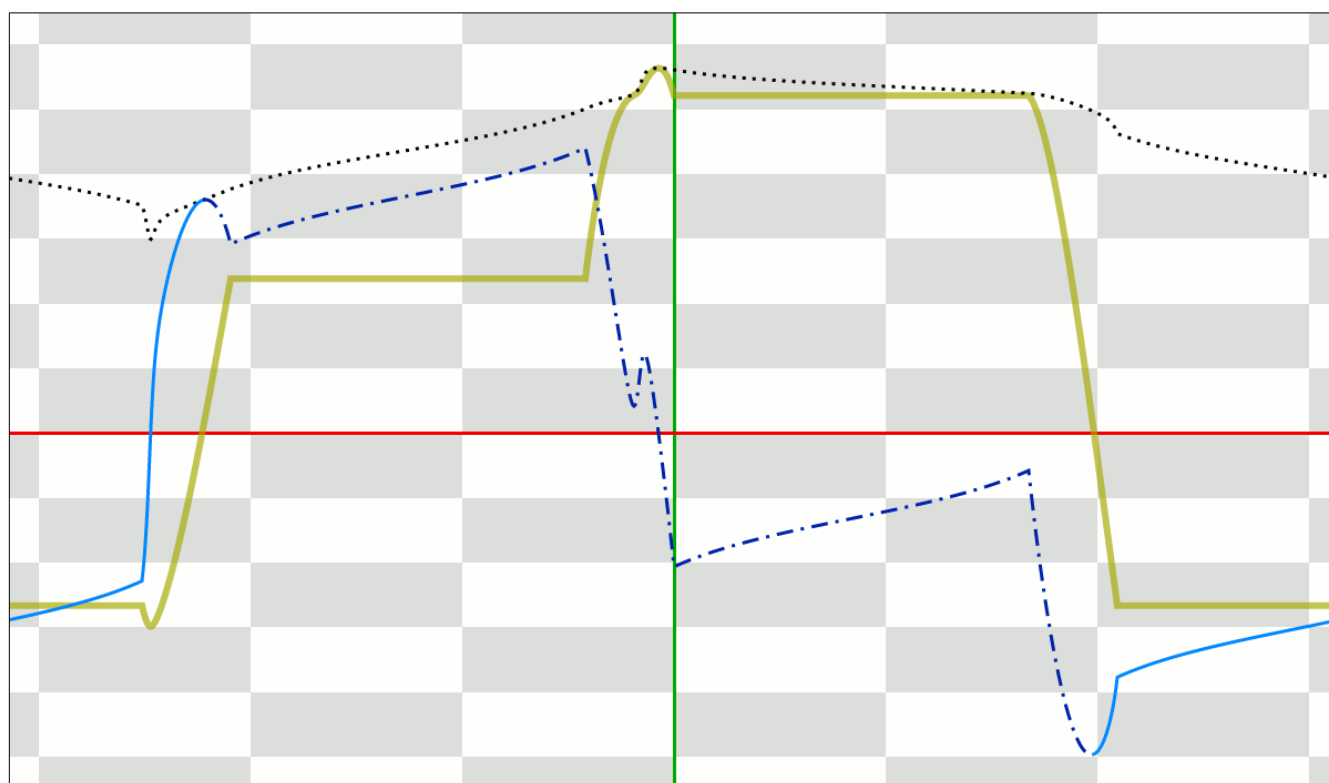
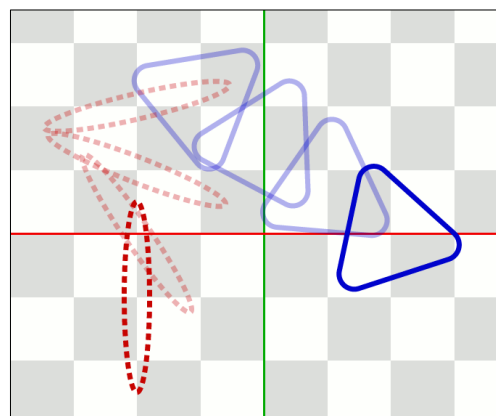
Результат (на изображении слева) соответствует ожидаемому.

Справа: дополнительная обработка в gimp.

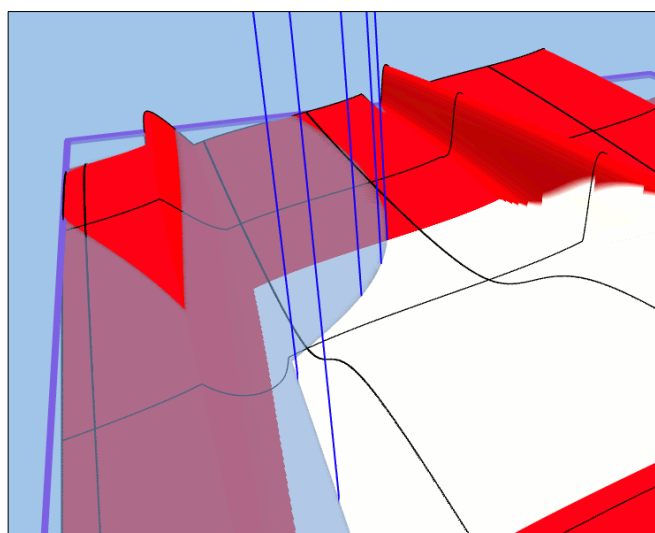
## Приложение В. Пример с одним нестрогим выпуклым объектом

На графике в начальный момент времени видны интервалы, в которых первая и вторая производные равны нулю. На этих интервалах метод Ньютона в оптимизации не будет работать.

Также видны точки, в которых первая производная не непрерывна.



$\text{distanceSide}(\varphi, t)$ .



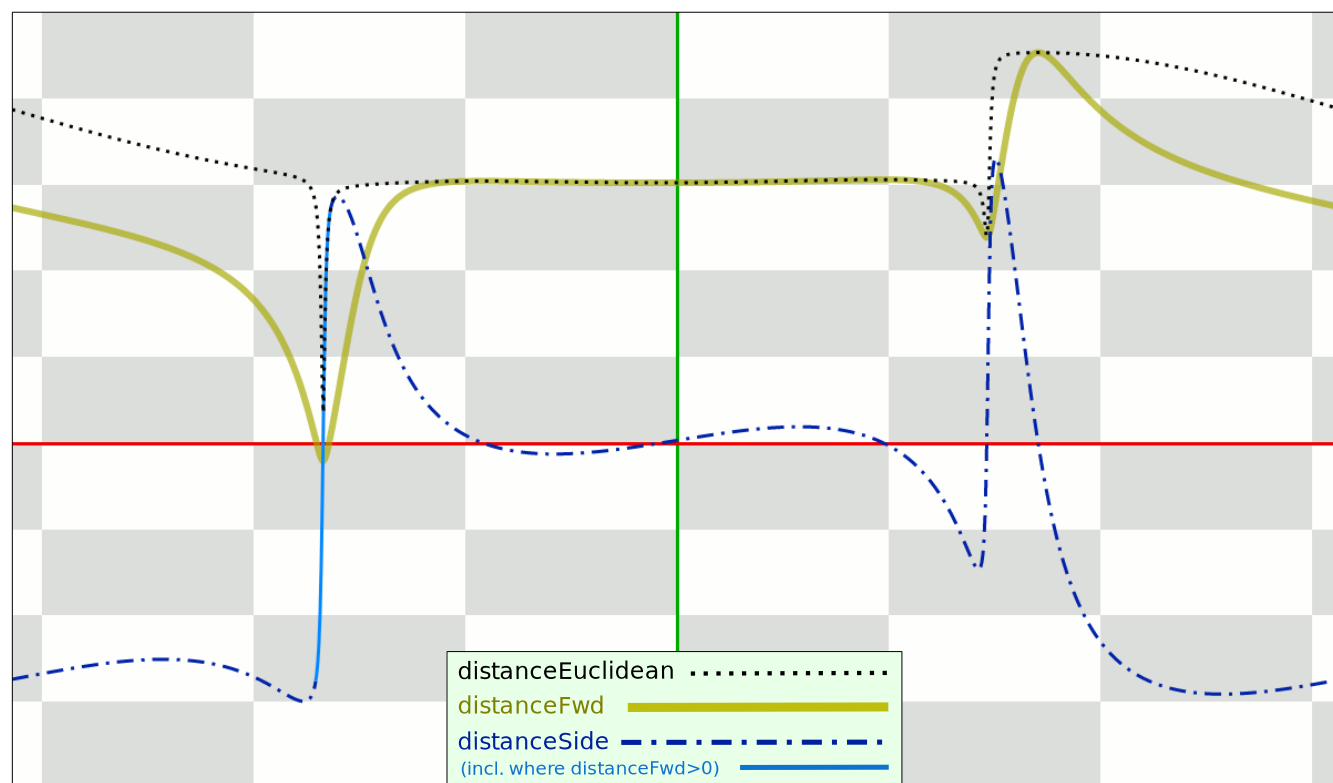
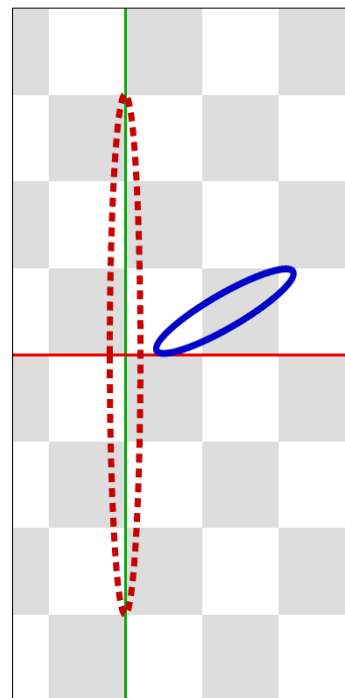
Справа: фрагмент 3D графика функции в окрестности точки столкновения. Точка находится на прямом участке периметра `object2`.

## Приложение С. Эллипсы с соотношением сторон 1:100 и более

Для иллюстрации возникающих сложностей при нахождении расстояния при  $t=0$  в случае большого отношения длин сторон, рассматриваются два эллипса (справа) с отношением сторон 1:5 и 1:15.

На графике  $\text{distanceEuclidean}(\varphi)$  три локальных минимума. У минимума при  $\varphi \approx 0$  большой радиус конвергенции.

У глобального минимума радиус конвергенции примерно 0.2 - 0.3. При увеличении соотношения длин сторон до 1:100 это значение уменьшается до примерно 0.01. На графике пик будет шириной менее 1px.

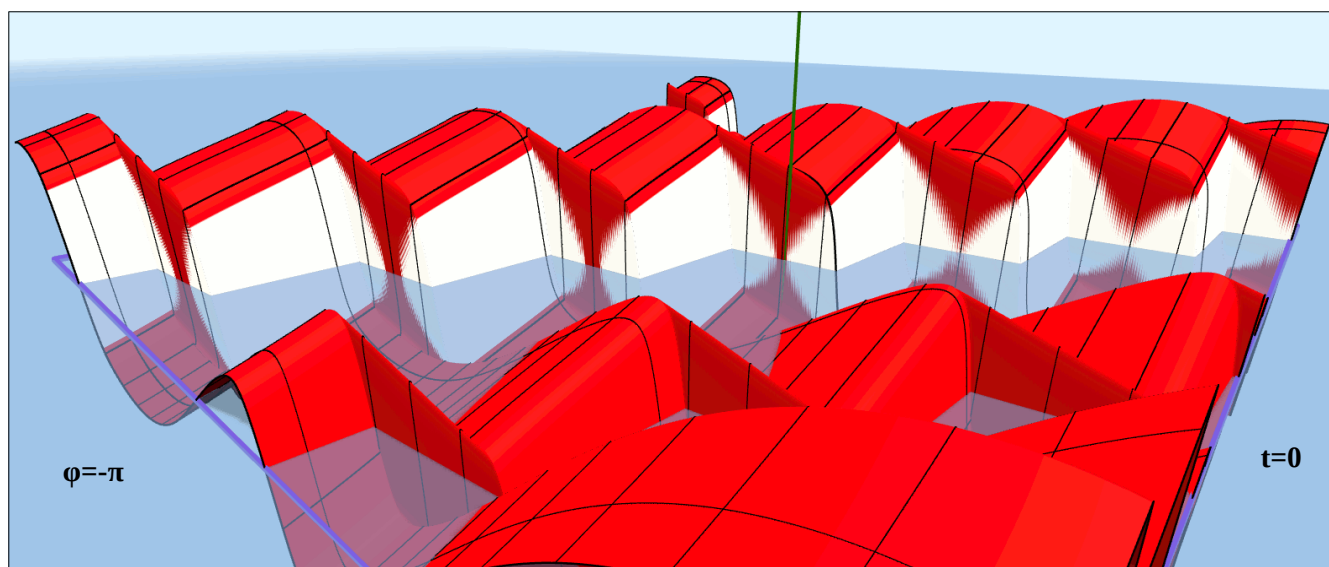
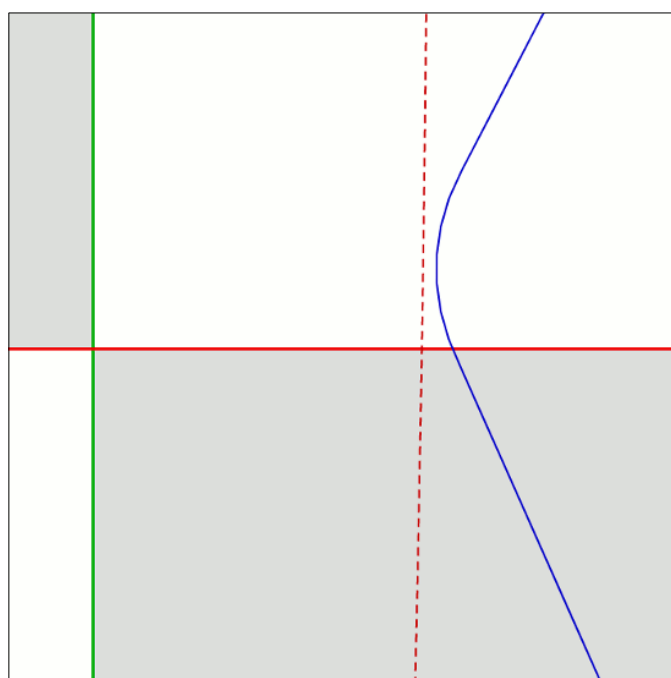
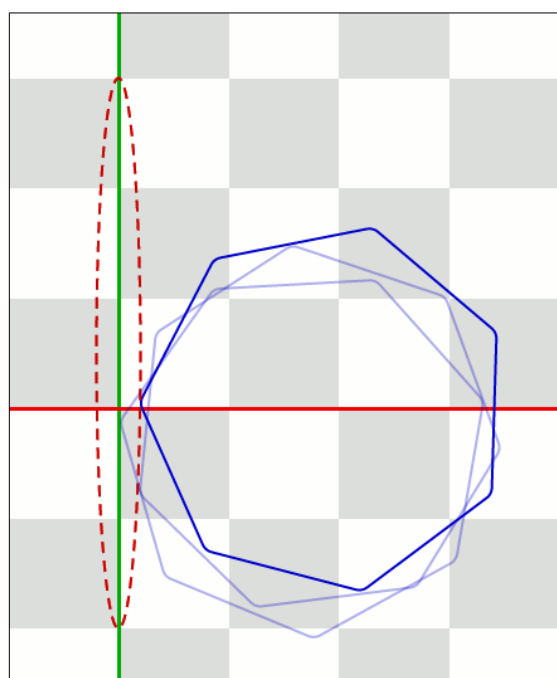


## Приложение D. Вращение объекта 30 радиан/с

В приложениях может оказаться необходимым обрабатывать и значительно более высокие скорости вращения.

**object1** - правильный гептагон с закругленными углами (regular heptagon with rounded corners) вращается и медленно движется. До момента столкновения с **object2** делает примерно пол-оборота.

Справа: фрагмент в окрестности начала координат, увеличенный в 20 раз.



$\text{distanceSide}(\varphi, t)$ .