МЕТОДЫ ЭЛЛИПСОИДОВ ДЛЯ РАЗДЕЛЕНИЯ ДВУХ ВЫПУКЛЫХ МНОЖЕСТВ

Стецюк П.И.
stetsyukp@gmail.com

Институт кибернетики им. В.М. Глушкова, Киев

Семинар "Образный компьютер"
24 июня 2014 г., г. Киев

Содержание

Содержание

Метод эллипсоидов предложили

1976 **Юдин Д.Б. и Немировский А.С.** как метод последовательных отсечений [1].

1977 **Шор Н.З.** как вариант метода с растяжением пространства в направлении субградиента [2].

1. Юдин Д.Б., Немировский А.С. *Информационная сложность и эффективные методы решения выпуклых экстремальных задач* // Экономика и математические методы. – 1976. – Вып. 2. – С. 357–369.

2. Шор Н.З. *Метод отсечения с растяжением пространства для решения задач выпуклого программирования* // Кибернетика. – 1977. – № 1. – С. 94–95.

На основе метода эллипсоидов

1979 **Хачиян Л.** построил первый полиномиальный алгоритм решения задачи ЛП с рациональными коэффициентами

1981 **Grötchel M., Lóvasz L., Schrijver A.** разработали полиномиальные алгоритмы для ряда задач дискретной оптимизации

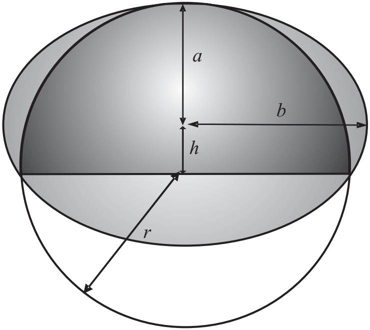
XI симпозиум по мат. программированию

Метод эллипсоидов и полученные на его основе результаты о сложности задач математического программирования были центральными на XI международном симпозиуме по математическому программированию (Бонн, ФРГ, август 1982).

3. Канторович Л.В., Михалевич В.С., Рубинштейн Г.Ш., Третьяков Н.В., Шор Н.З., Якимец В.Н. *XI Международный симпозиум по математическому программированию* // Техническая кибернетика. – М.: Изв. АН СССР. – 1983. – № 1. – С. 197–201.

Содержание

1-d эллипсоид и его свойства



Эллипсоид $\mathcal{E}_n$, содержащий полушар в E^n, имеет параметры

$$b = \left(\alpha + \frac{1}{\alpha}\right)\frac{r}{2}, \quad h = \left(1 - \frac{1}{\alpha^2}\right)\frac{r}{2},$$

где $\alpha = \frac{b}{a}$ и r – радиус шара S_n.

Если пространство „растянуть“ с коэффициентом α в направлении полуоси a, то $\mathcal{E}_n$ станет шаром в преобразованном пространстве.

Отношение объема эллипсоида $\mathcal{E}_n$ к объему шара S_n равно

$$q(n) = \frac{\mathrm{vol}(\mathcal{E}_n)}{\mathrm{vol}(S_n)} = \frac{1}{\alpha}\left(\frac{b}{r}\right)^n = \frac{1}{\alpha}\left(\frac{1}{2}\left(\alpha + \frac{1}{\alpha}\right)\right)^n.$$

Оператор растяжения пространства

Введен Н.З. Шором (1969) и имеет следующий вид

$$R_\alpha(\xi) = I_n + (\alpha - 1)\xi\xi^T, \quad \text{где} \;\; \alpha > 1.$$

Здесь: α – коэффициент растяжения пространства
в нормированном направлении $\xi \in E^n$, $\|\xi\| = 1$,
I_n – единичная матрица размером $n \times n$.

В методах используется обратный к нему оператор

$$R_\beta(\xi) = I_n + (\beta - 1)\xi\xi^T, \quad \text{где} \;\; \beta = \frac{1}{\alpha} < 1,$$

который означает "сжатие"пространства субградиентов.

Почему метод эллипсоидов сходится?

Отношение объема эллипсоида $\mathcal{E}_n$ к объему шара S_n равно

$$q(n) = \frac{\mathrm{vol}(\mathcal{E}_n)}{\mathrm{vol}(S_n)} = \frac{1}{\alpha}\left(\frac{1}{2}\left(\alpha + \frac{1}{\alpha}\right)\right)^n.$$

Если коэффициент α такой, что $\alpha + 1/\alpha < 2\sqrt[n]{\alpha}$, то отношение $q(n) < 1$ и объем эллипсоида, в котором локализуется искомая точка x^*, убывает со скоростью геометрической прогрессии со знаменателем $q(n)$.

О двух вариантах метода эллипсоидов

В методе эллипсоидов Юдина-Немировского-Шора

$$q(n) = 1 - \frac{1}{2n} \quad \text{и реализуется при} \quad \alpha = \sqrt{\frac{n+1}{n-1}}.$$

В приближенном методе эллипсоидов [4]

$$q(n) \approx 1 - \frac{1}{2n} \quad \text{и реализуется при} \quad \alpha = \sqrt{1 + \frac{1}{n^2}} + \frac{1}{n}.$$

Если $n = 1$, то $q(1) = 2 - \sqrt{2} \approx 0.5858$.

4. Стецюк П.И. *Приближенный метод эллипсоидов* // Кибернетика и системный анализ. – 2003. – № 3. – С. 141–146.

Содержание

Метод эллипсоидов предназначен

для решения такой задачи:

На E^n $(n \geq 1)$ определено векторное поле $g(x)$, $g(x) \in E^n$.

Требуется найти точку x^*, такую, что
$(g(x), x - x^*) \geq 0$ **для всех** $x \in E^n$**.**

Предполагается, что x^* существует и $g(x) \neq 0$ для $x \neq x^*$.

К ней сводятся задачи математического программирования:

- задача безусловной минимизации выпуклой функции,
- общая задача выпуклого программирования,
- задача о седловой точке выпукло-вогнутых функций.

Стартовые условия для МЭ

Задан:

коэффициент α такой, что $\alpha + 1/\alpha < 2\sqrt[n]{\alpha}$.

Инициализация:

1. Выберем стартовую точку $x_0 \in E^n$ и начальный радиус r_0, такой что $\|x_0 - x^*\| \leq r_0$.
2. Положим $B_0 := I_n$, где I_n – единичная $(n{\times}n)$-матрица.

Перейдем к очередной итерации со значениями x_0, r_0, B_0.

Собственно сам алгоритм

Пусть на k-й итерации найдены $x_k \in E^n$, r_k и B_k.
Для перехода к $(k+1)$-й итерации выполняем

Шаг 1. Вычислим $g(x_k)$. Если $g(x_k) = 0$, то ОСТАНОВ$(x^* = x_k)$. Иначе переходим к шагу 2.

Шаг 2. Вычислим очередную точку

$$x_{k+1} := x_k - h_k B_k \xi_k, \text{ где } h_k = \frac{1}{2}\left(1 - \frac{1}{\alpha^2}\right), \ \xi_k = \frac{B_k^T g(x_k)}{\|B_k^T g(x_k)\|}.$$

Шаг 3. Пересчитаем матрицу B_{k+1} и радиус r_{k+1}

$$B_{k+1} := B_k + (\beta - 1)(B_k \xi_k)\xi_k^T, \ \beta = 1/\alpha, \ r_{k+1} := (\alpha + \beta) \times r_k / 2.$$

Шаг 4. Переходим к $(k+1)$-й итерации с x_{k+1}, r_{k+1} и B_{k+1}.

О сходимости метода эллипсоидов

Теорема 1 (локализация x^* в эллипсоиде)

Генерируемая методом эллипсоидов последовательность точек $\{x_k\}_{k=0}^{\infty}$ удовлетворяет неравенствам

$$\|B_k^{-1}(x_k - x^*)\| \leq r_k, \quad k = 0, 1, 2, \ldots \qquad \text{(Shor77)}$$

Следствие:

Эллипсоид $\mathcal{E}_k = \{x : \|B_k^{-1}(x_k - x)\| \leq r_k\}$ содержит точку x^*.

О сходимости метода эллипсоидов

Теорема 2 (о скорости сходимости)

Для всех итераций метода эллипсоидов коэффициент уменьшения объема эллипсоида, локализующего x^*, есть величина постоянная и равная

$$q(n) = \frac{\mathrm{vol}(\mathcal{E}_{k+1})}{\mathrm{vol}(\mathcal{E}_k)} = \frac{1}{\alpha}\left(\frac{1}{2}\left(\alpha + \frac{1}{\alpha}\right)\right)^n < 1, \quad k = 0, 1, 2, \ldots$$

"Оптимальные" коэффициенты растяжения пространства:

1. $\alpha_1 = \sqrt{\frac{n+1}{n-1}} \quad \Rightarrow \quad q(n) = \sqrt{\frac{n-1}{n+1}}\left(\frac{n^2}{n^2-1}\right)^{n/2} \leq 1 - \frac{1}{2n}.$

2. $\alpha_2 = \sqrt{1+\frac{1}{n^2}} + \frac{1}{n} \quad \Rightarrow \quad q(n) = \left(1+\frac{1}{n^2}\right)^{n/2}\left(\sqrt{1+\frac{1}{n^2}} - \frac{1}{n}\right) \approx 1 - \frac{1}{2n}.$

Содержание

octave-функция emshor (ellipsoid method shor)

```
# octave-code for emshor (version 1.0, 13.09.2013)      #row00
# находит точку минимума выпуклой функции f(x)          #row00a
% Входные параметры:                                    #row00b
% calcfg - имя функции calcfg(x): вычисляет f и g(1:n) #row00c
% x  - начальная точка x0(1:n) (на выходе портится)     #row00d
% rad - радиус шара с центром в точке x0                #row00e
% epsf - точность останова по значению функции          #row00f
% maxitn - максимальное количество итераций             #row00g
% Выходные параметры:                                   #row00h
% xr -- точка минимума (рекордная)                      #row00i
% fr -- значение функции в точке xr                     #row00j
% ist -- код останова (1 = epsf, 4 = maxitn)            #row00k
function [xr,fr,ist]=emshor(calcfg,x,rad,epsf,maxitn); #row01
dn=float(length(x)); beta=sqrt((dn-1.d0)/(dn+1.d0));   #row02
radn=rad; B=eye(length(x)); fr=inf; ist=4;             #row03
for (itn = 0:maxitn)                                   #row04
   [f, g1] = calcfg(x); g=B'*g1; dg=norm(g);           #row05
   if (f < fr)  fr = f; xr = x; itr=itn; endif         #row06
   if(radn*dg < epsf) ist = 1;  return; endif          #row07
   xi=(1.d0/dg)*g; dx = B * xi; hs=radn/(dn+1.d0);     #row08
   x -= hs * dx;  B += (beta - 1) * B * xi * xi';      #row09
   radn=radn/sqrt(1.d0-1.d0/dn)/sqrt(1.d0+1.d0/dn);    #row10
   printf("itn %4d itr %4d f %16.8e fr %16.8e\n",      #row11
          itn,itr,f,fr);
endfor                                                 #row12
endfunction                                            #row13
```

Что находит программа emshor?

Пусть $f(x)$ – выпуклая функция, x^* – точка минимума, $f^* = f(x^*)$, x_0 – начальное приближение.

Теорема (о программе emshor)

Если x_0 такое, что $\|x_0 - x^*\| \leq rad$, то программа **emshor** заканчивает работу выполнением одного из условий:

1. найдена точка x_r – такая, что $f(x_r) - f^* \leq \varepsilon_f$ (ist=1),
2. **maxitn** итераций оказалось недостаточно (ist=4).

octave-функция emshor (комментарии)

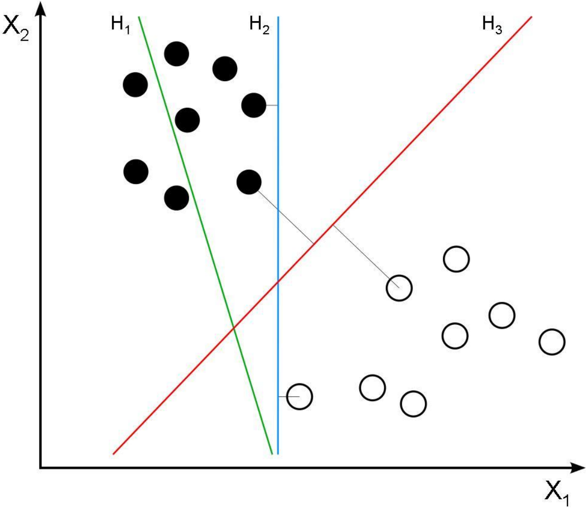
```
# octave-code for emshor (version 1.0, 13.09.2013)      #row00
# находит точку минимума выпуклой функции f(x)           #row00a
% Входные параметры:                                     #row00b
% calcfg - имя функции calcfg(x): вычисляет f и g(1:n) #row00c
% x   - начальная точка x0(1:n) (на выходе портится)     #row00d
% rad - радиус шара с центром в точке x0                 #row00e
% epsf - точность останова по значению функции           #row00f
% maxitn - максимальное количество итераций              #row00g
% Выходные параметры:                                    #row00h
% xr -- точка минимума (рекордная)                       #row00i
% fr -- значение функции в точке xr                      #row00j
% ist -- код останова (1 = epsf, 4 = maxitn)             #row00k
```

octave-функция emshor (код программы)

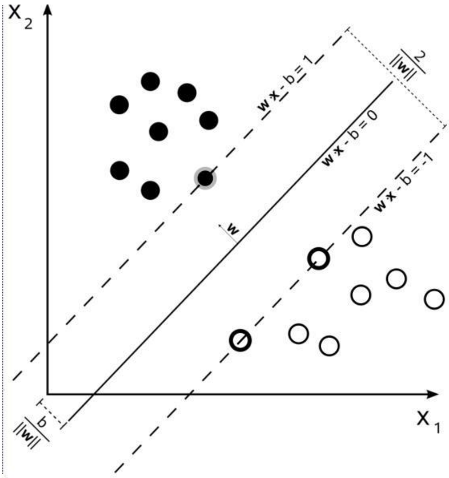
```
function [xr,fr,ist]=emshor(calcfg,x,rad,epsf,maxitn); #row01
dn=float(length(x)); beta=sqrt((dn-1.d0)/(dn+1.d0));   #row02
radn=rad; B=eye(length(x)); fr=inf; ist=4;             #row03
for (itn = 0:maxitn)                                   #row04
   [f, g1] = calcfg(x); g=B'*g1; dg=norm(g);           #row05
   if (f < fr)  fr = f; xr = x; itr=itn; endif         #row06
   if(radn*dg < epsf) ist = 1;  return; endif          #row07
   xi=(1.d0/dg)*g; dx = B * xi; hs=radn/(dn+1.d0);     #row08
   x -= hs * dx;  B += (beta - 1) * B * xi * xi';      #row09
   radn=radn/sqrt(1.d0-1.d0/dn)/sqrt(1.d0+1.d0/dn);    #row10
   printf("itn %4d itr %4d f %16.8e fr %16.8e\n",      #row11
          itn,itr,f,fr);
endfor                                                 #row12
endfunction                                            #row13
```

Содержание

Примеры линейных классификаторов H_2 и H_3



Пример наилучшего линейного классификатора



Наилучший линейный классификатор (НЛК)

Задача о наилучшем линейном классификаторе состоит в нахождении гиперплоскости с максимальным зазором, которая разделяет два множества точек $\hat{X} = \{\hat{x}_1, \ldots, \hat{x}_{m_1}\}$ и $\hat{Y} = \{\hat{y}_1, \ldots, \hat{y}_{m_2}\}$ евклидового пространства E^n.

Предполагается, что множества $\hat{X}$ и $\hat{Y}$ разделимы линейной гиперплоскостью $wx{-}b{=}0$, $w^T{\in}E^n$, $ww^T{=}1$.

Обозначения:

$2d^*$ – максимальный зазор (ширина полосы разделения),
$H^* \equiv w^*x - b^* = 0$ – наилучший линейный классификатор.

Содержание

Квадратичная задача для НЛК

найти

$$\left(\frac{1}{d^*}\right)^2 = \min_{(\tilde{w},\tilde{b})} \left\{\tilde{w}\tilde{w}^T = \|\tilde{w}\|^2\right\} \tag{1.1}$$

при ограничениях

$$\tilde{w}\hat{x}_i - \tilde{b} \geq 1, \quad i = 1, \ldots, m_1, \tag{1.2}$$

$$-\tilde{w}\hat{y}_j + \tilde{b} \geq 1, \quad j = 1, \ldots, m_2. \tag{1.3}$$

Здесь $2d^* = 2/\|\tilde{w}^*\|$ – максимальный зазор,
$w^* = \tilde{w}^*/\|\tilde{w}^*\|$, $b^* = \tilde{b}^*/\|\tilde{w}^*\|$ – параметры НЛК.

Негладкая задача для НЛК [5]

$$\textbf{найти} \quad -d^* = \min_{(w,b)} f(w,b) \tag{2.1}$$

$$\textbf{при ограничении} \quad \sum_{i=1}^{n} w_i^2 \leq 1, \tag{2.2}$$

где $f(w,b) = \max\{\max_{i=1,\ldots,m_1}\{-w\hat{x}_i + b\}, \max_{j=1,\ldots,m_2}\{w\hat{y}_j - b\}\}.$

5. Стецюк П.І., Березовський О.А., Журбенко М.Г., Кропотов Д.О. *Методи негладкої оптимізації в спеціальних задачах класифікації.* – Препр. / НАН України. Ін-т кібернетики ім. В.М. Глушкова; 2009-1. – 29 с.

Безусловная негладкая задача для НЛК

Задача (2.1),(2.2) сводится к задаче минимизации негладкой выпуклой функции

$$F_P(w,b) = f(w,b) + P\max\{0, \sum_{i=1}^{n} w_i^2 - 1\} \qquad (2.3)$$

для которой при $P > d^*/2$ [6] ее минимум совпадает с решеним задачи (2.1),(2.2).

6. Березовский О.А., Стецюк П.И. *Задачи негладкой безусловной оптимизации для линейного и квадратичного классификаторов с максимальным зазором* // V-a Міжнародна школа-семінар "Теорія прийняття рішень", Ужгород, 27 вересня - 1 жовтня 2010. Праці школи-семінару. – С. 15–16.

О методах решения задачи (2.3)

Для нахождения минимума функции $F_P(w, b)$ можно использовать алгоритмы минимизации негладких выпуклых функций.

Если $n = 2 \div 10$, то целесообразно использовать методы эллипсоидов. Их сходимость не зависит от количества точек $m = m_1 + m_2$.

Ниже опишем такой алгоритм и сделаем это для произвольного $\alpha > 1$ – коэффициента растяжения пространства. Для удобства неизвестные $(w, b)^T$ обозначим одним вектором $u \in E^{n+1}$.

Содержание

Установка начальных значений

Инициализация

1. Положим штраф $P = \min\limits_{i,j} \|\hat{x}_i - \hat{y}_j\|$.

2. Выберем стартовую точку $u_0 = 0$ и
начальный радиус $r_0 = \max\limits_{i,j}\{\|\hat{x}_i\|, \|\hat{y}_j\|\} + 1$.

3. Положим $B_0 = I_{n+1}$, где I_{n+1} – единичная матрица размером $(n+1) \times (n+1)$.

Перейдем к первой итерации со значениями u_0, r_0 и B_0.

Собственно сам алгоритм

Пусть на k-й итерации найдены $u_k \in E^{n+1}$, r_k и B_k.
Для перехода к $(k+1)$-й итерации выполняем

Шаг 1. Вычислим $F_P(u_k)$ и $g(u_k) = \partial F_P(u_k)$. Если $r_k \| B_k^T g(u_k) \| \leq \varepsilon$, то ОСТАНОВ $u^* = u_k$. Иначе переходим к шагу 2.

Шаг 2. Вычислим очередную точку

$$u_{k+1} = u_k - h_k B_k \xi_k, \text{ где } h_k = \frac{1}{2}\left(1 - \frac{1}{\alpha^2}\right), \; \xi_k = \frac{B_k^T g(u_k)}{\| B_k^T g(u_k) \|}.$$

Шаг 3. Пересчитаем матрицу B_{k+1} и радиус r_{k+1}

$$B_{k+1} = B_k + (\beta - 1)(B_k \xi_k) \xi_k^T, \; \beta = 1/\alpha, \;\; r_{k+1} = (\alpha + \beta) \times r_k / 2.$$

Шаг 4. Переходим к $(k+1)$-й итерации с u_{k+1}, r_{k+1} и B_{k+1}.

О сходимости алгоритма

Теорема

Если α – такое, что $\alpha + 1/\alpha < 2\sqrt[n+1]{\alpha}$, то алгоритм эллипсоидов сходится к F_P^* со скоростью геометрической прогресии с показателем

$$q = \frac{1}{2\sqrt[n+1]{\alpha}}\left(\alpha + \frac{1}{\alpha}\right) < 1.$$

Здесь

F_P^* – оптимальное значение штрафной функции (2.3).

Содержание

Растояние между двумя полиэдрами

найти

$$\rho^* = \min_{x \in E^n, y \in E^n} \|x - y\| \tag{3.1}$$

при ограничениях

$$A_1 x \leq b_1, \qquad A_2 y \leq b_2, \tag{3.2}$$

где A_1 **–** $m_1 \times n$**-матрица,** b_1 **–** m_1**-мерный вектор,** A_2 **–** $m_2 \times n$**-матрица,** b_2 **–** m_2**-мерный вектор,**

Структура решения:

ρ^* – единственно, x^*, y^* – не обязательно единственны.

Растояние между двумя эллипсоидами

найти

$$\rho^* = \min_{x \in E^n, y \in E^n} \|x - y\| \tag{4.1}$$

при ограничениях

$$\|A_1(x - x_0)\| \leq r_1, \qquad \|A_2(y - y_0)\| \leq r_2, \tag{4.2}$$

где A_1 и A_2 – $n{\times}n$-матрицы.

Оптимальное решение ρ^*, x^*, y^* – единственное.

Выводы:

Теория:

Для задач разделения выпуклых множеств на основе метода эллипсоидов можно построить „короткие“ алгоритмы и гарантировать их сходимость со скоростью геометрической прогрессии.
Однако скорость сходимости будет уменьшаться с ростом количества переменных.

Практика:

Обеспечить эффективность методов эллипсоидов для задач больших размеров можно за счет „ускоренных“ вариантов методов эллипсоидов, которые используют не одно, а нескольких отсечений.

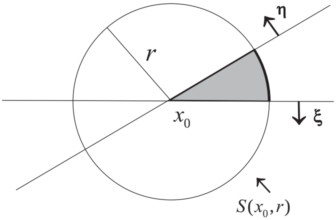
Благодарности

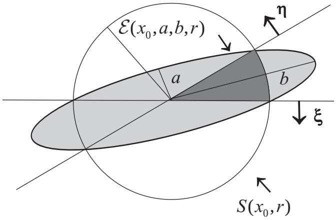
Работа выполнена при финансовой поддержке НАН Украины, проект 07-01-14 (У)

Вопросы?

СПАСИБО ЗА ВНИМАНИЕ!

BACK UP SLIDES: Тело W и 2d-эллипсоид





Тело W получено как пересечение шара и двух полупространств.

2d-эллипсоид содержит W и имеет минимальный объем.

Стецюк П.И. r-алгоритмы и эллипсоиды // Кибернетика и системный анализ. – 1996. – № 1. – С. 113–134.

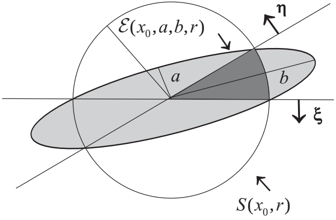
BACK UP SLIDES: Свойства 2d-эллипсоида

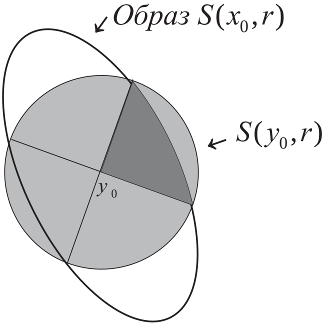
1. Если угол φ между векторами ξ и η тупой, то 2d-эллипсоид содержит тело W.

2. Объем 2d-эллипсоида меньше, чем объем шара, и это уменьшение равно $\sqrt{1-(\xi,\eta)^2} = \sqrt{1-(\cos\varphi)^2}$.

3. Преобразовать 2d-эллипсоид в шар можно растяжением пространства в направлении $\frac{\xi-\eta}{\|\xi-\eta\|}$ с коэффициентом $\alpha_1 = \frac{1}{\sqrt{1+(\xi,\eta)}}$ и сжатием пространства в ортогональном направлении $\frac{\xi+\eta}{\|\xi+\eta\|}$ с коэффициентом $\alpha_2 = \frac{1}{\sqrt{1-(\xi,\eta)}}$. Но экономнее это сделать с помощью ОЭО.

BACK UP SLIDES: 2d-эллипсоид до и после ...





2d-эллипсоид

в преобразованном пространстве становится шаром