



IntvalPy — библиотека интервальных вычислений на языке Python

АНДРОСОВ АРТЕМ СТАНИСЛАВОВИЧ

МАГИСТРАНТ 2-ГО КУРСА ММФ НГУ

ШАРЫЙ СЕРГЕЙ ПЕТРОВИЧ

Д. Ф.-М. Н., ВЕДУЩИЙ НАУЧНЫЙ СОТРУДНИК ФИЦ ИВТ

Постулаты

- ❖ Соблюдение лицензий сторонних библиотек (numpy, matplotlib, mpmath);
- ❖ Библиотека проста в использовании, вся функциональность задокументирована;
- ❖ Библиотека должна удовлетворять общепринятым стандартам - IEEE 754-2008 для арифметики с плавающей точкой на ЭВМ и IEEE 1788-2015 для интервальных вычислений на ЭВМ;
- ❖ Высокая производительность;
- ❖ Кроссплатформенность;
- ❖ Возможность работы в разных интервальных арифметиках;
- ❖ Библиотека должна быть гибкой к изменениям и обеспечивать возможность расширения функциональности.

Архитектура

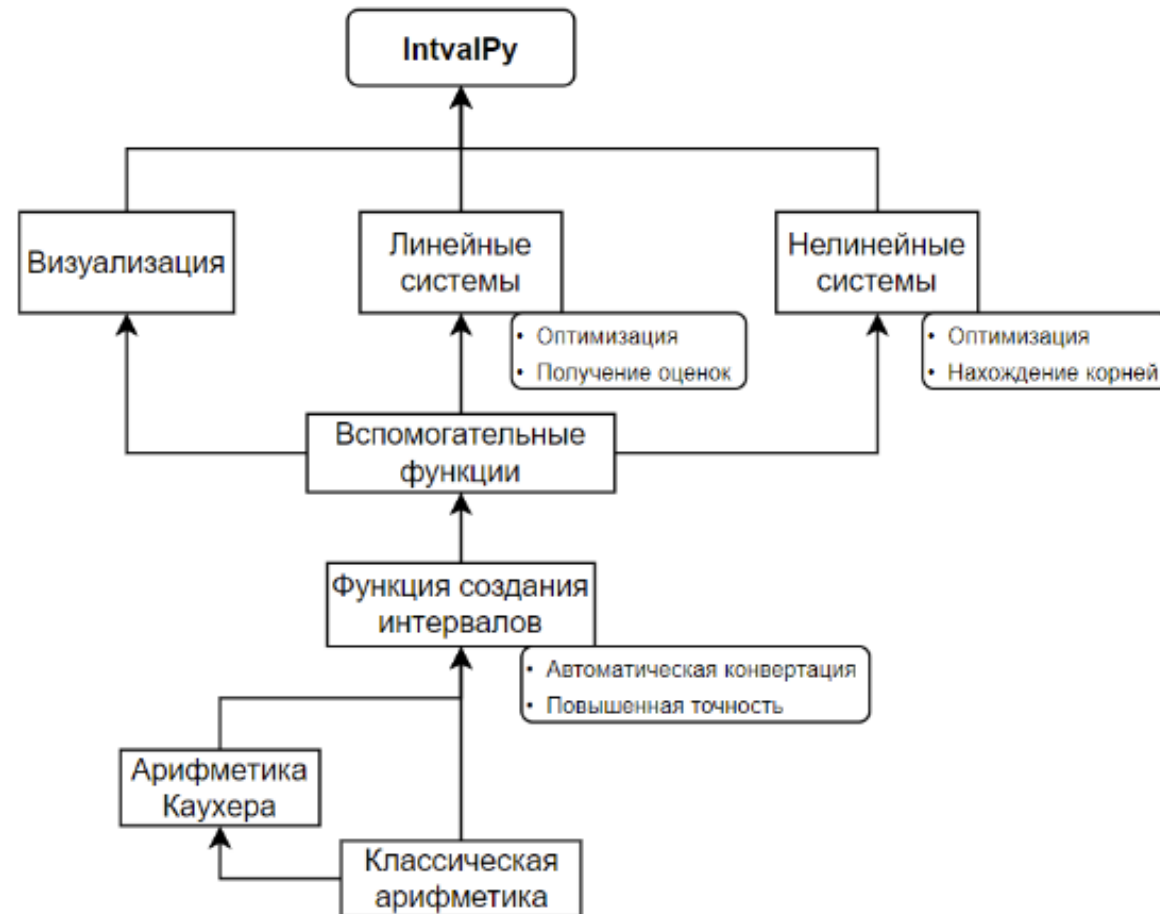


Рисунок 1 – Архитектура IntvalPy

Функциональность: Низкоуровневая

Таблица 2 – Доступные операции в IntvalPy

Операции	Описание
$+, -, *, /$ pow, abs, exp, log, sin, cos matmul, dot, transpose $<, \leq, =, \geq, >, \neq, \subset, \subseteq, \cap$ rad, wid, mid, mig, mag dual, pro, opp, inv	Арифметические операции Элементарные функции Матричные операции Бинарные отношения и операции Характеристики интервалов Преобразования интервалов

Листинг 1 – Основные операции

```
1|| import intvalpy as ip
2|| import numpy as np
3||
4|| # Barth-Nuding system
5|| A = ip.Interval([
6||     [[2, 4], [-2, 1]],
7||     [[-1, 2], [2, 4]]
8|| ])
9|| b = ip.Interval([[ -2, 2], [-2, 2]])
10|| x = np.array([-0.2, 0.4])
11||
12|| ip.subset(A @ x, b)          # True
13|| # A @ x = Interval(['[-1.6, 0]', '[0.4, 1.8]'])
14||
15|| a = A[0, 1].opp              # Interval(['[2, -1]'])
16|| type(A[0, 1].opp)
17|| # intvalpy.RealInterval.KaucherArithmetic
```

Функциональность: Низкоуровневая

$$f(a, b) = 333.75 \cdot b^6 + a^2 \cdot (11 \cdot a^2 \cdot b^2 - b^6 - 121 \cdot b^4 - 2) + 5.5 \cdot b^8 + a/(2 \cdot b)$$

```
import pyibex as pb

f = lambda a, b: 333.75 * b**6 + a**2 * (11 * a**2 * b**2 - \
    b**6 - 121 * b**4 - 2) + 5.5 * b**8 + a/(2*b)

a = pb.Interval(77617, 77617)
b = pb.Interval(33096, 33096)
f(a, b)
```

```
[-4.72237e+21, 3.54177e+21]
```

```
import intervalpy as ip
```

```
ip.precision.extendedPrecisionQ = True
ip.precision.dps(40)
```

```
f = lambda a, b: 333.75 * b**6 + a**2 * (11 * a**2 * b**2 - \
    b**6 - 121 * b**4 - 2) + 5.5 * b**8 + a/(2*b)
```

```
a = ip.Interval(77617, 77617)
b = ip.Interval(33096, 33096)
f(a, b)
```

```
'[-0.827396, -0.827396]'
```

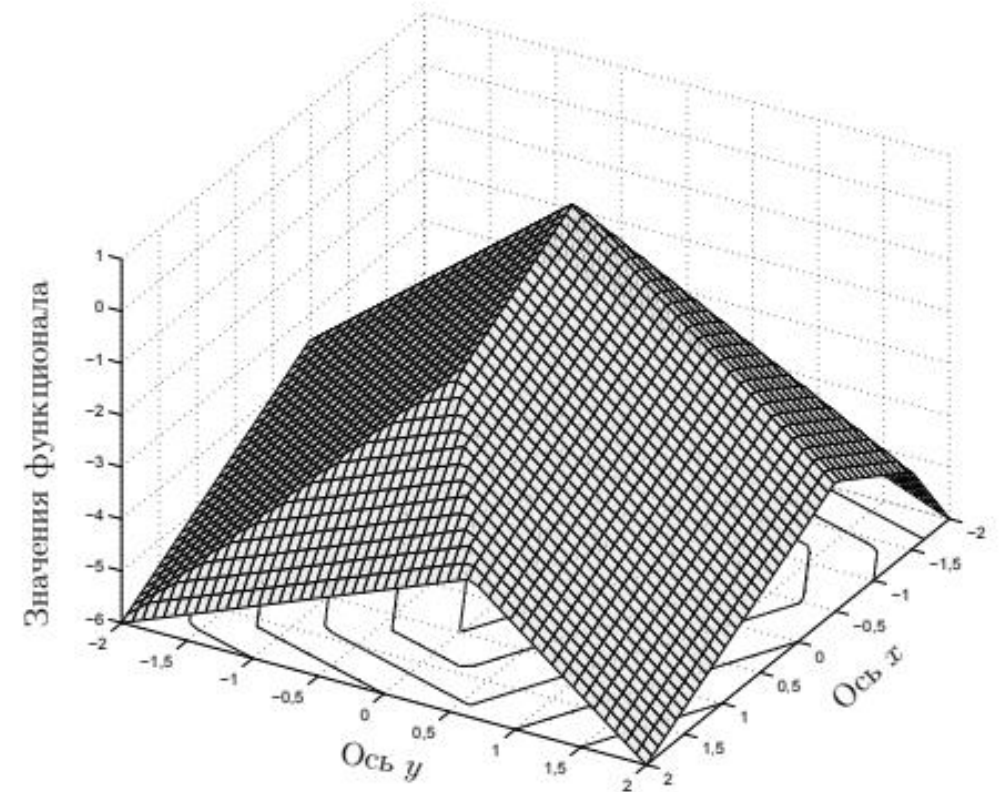
Функциональность: Верхнеуровневая

$$\text{Uni}(\beta, \mathbf{X}, \mathbf{y}) = \min_i \left\{ \text{rad } \mathbf{y}_i - \left\langle \text{mid } \mathbf{y}_i - \sum_{j=1}^n \mathbf{x}_{ij} \beta_j \right\rangle \right\}$$

$$\text{Tol}(\beta, \mathbf{X}, \mathbf{y}) = \min_i \left\{ \text{rad } \mathbf{y}_i - \left| \text{mid } \mathbf{y}_i - \sum_{j=1}^n \mathbf{x}_{ij} \beta_j \right| \right\}$$

Листинг 3 – Метод распознающих функционалов

```
1|| import intvalpy as ip
2||
3|| A, b = ip.Neumeier(3, 3.33, infb=-1, supb=2)
4|| x = np.array([-3, 7, 2])
5|| print('Tol: ', ip.linear.Tol(A, b, x=x))
6|| print('max Tol: ', ip.linear.Tol(A, b, maxQ=True))
```

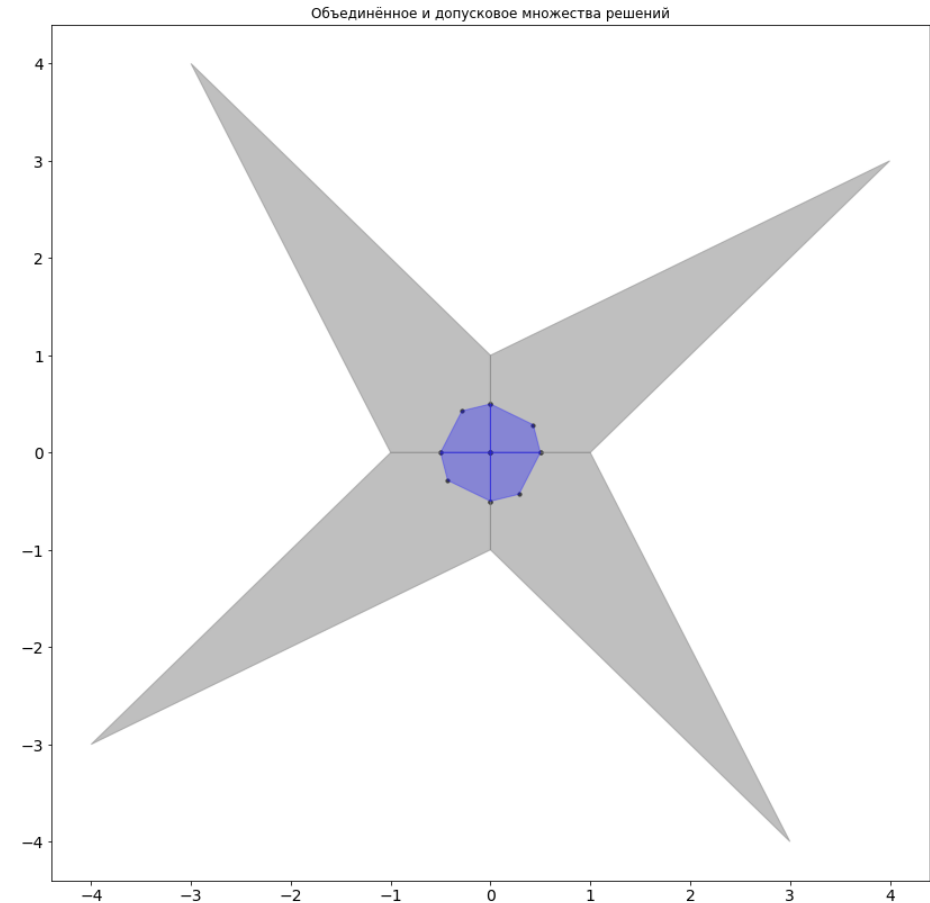


Функциональность: Верхнеуровневая

$$\begin{pmatrix} [2, 4] & [-2, 1] \\ [-1, 2] & [2, 4] \end{pmatrix} x = \begin{pmatrix} [-2, 2] \\ [-2, 2] \end{pmatrix}$$



```
1 import intervalpy as ip
2
3 A = ip.Interval([
4     [[2, 4], [-2, 1]],
5     [[-1, 2], [2, 4]]
6 ])
7 b = ip.Interval([[ -2, 2], [-2, 2]])
8
9 x = ip.linear.PSS(A, b)
10 print('x: ', x)
11 ip.IntLinIncR2(A, b, alpha=0.5, s=0, size=(11,11))
```



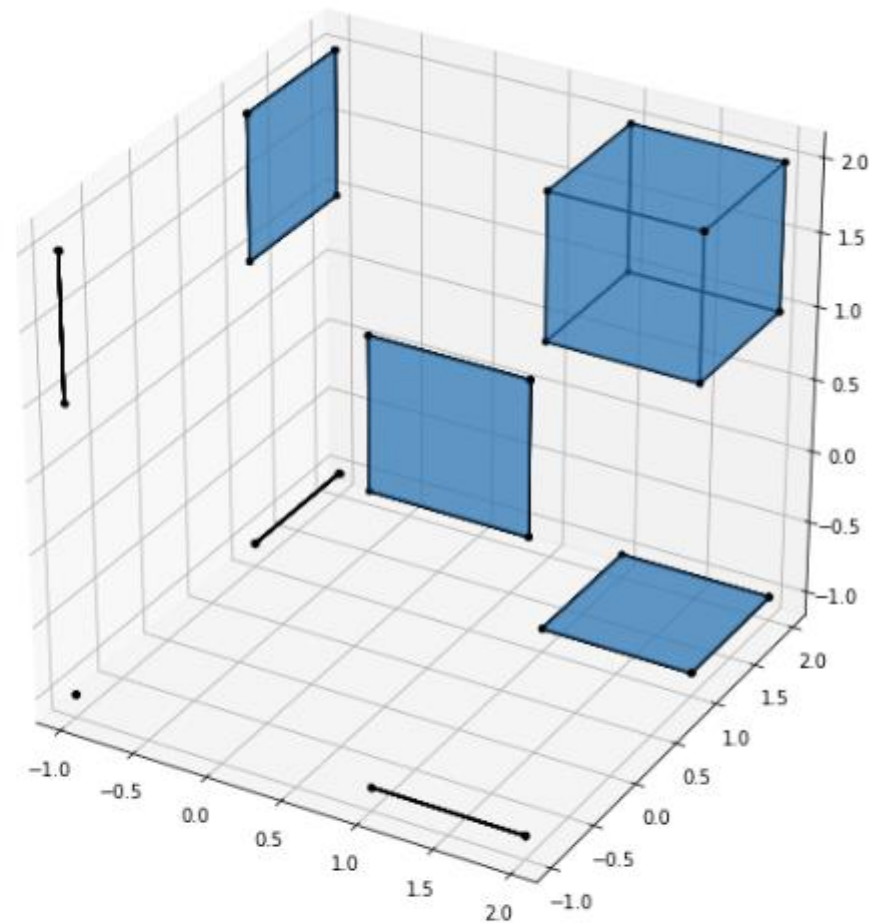
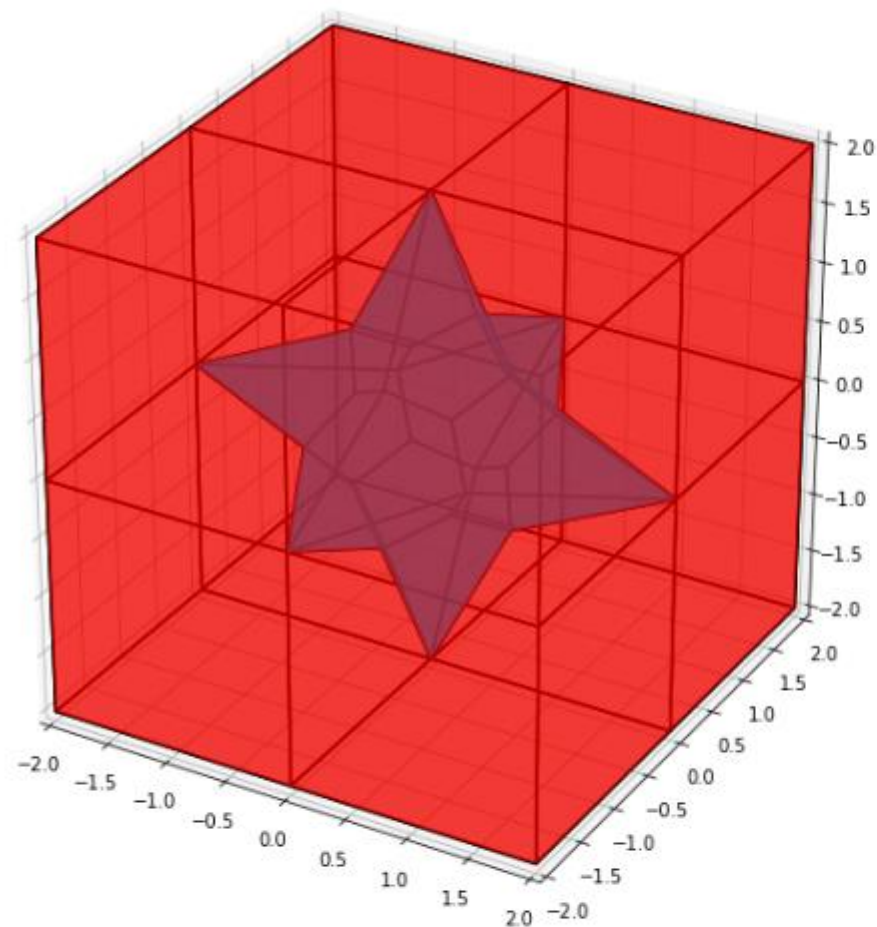
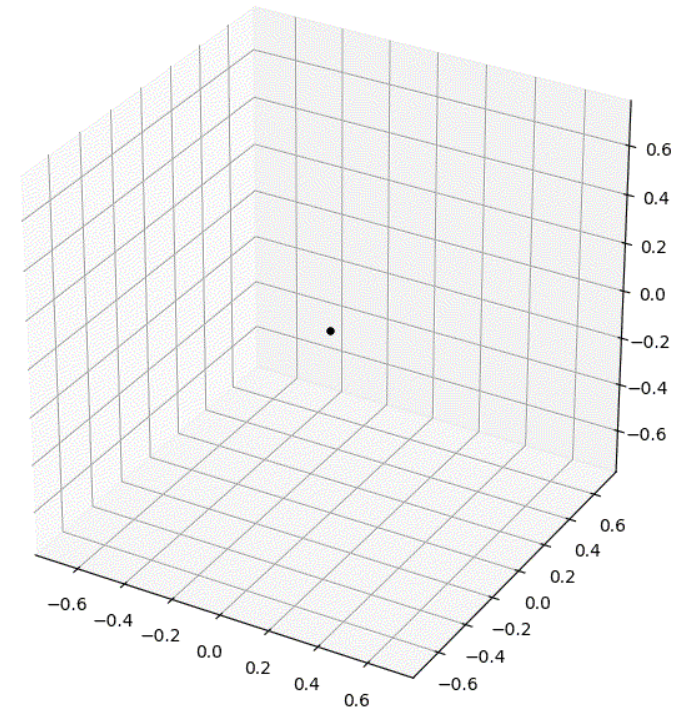
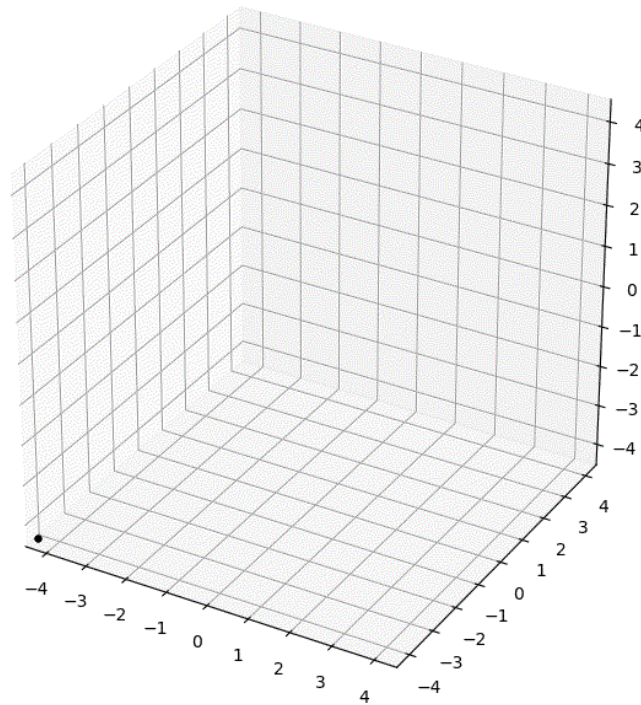


Рисунок 2 – Объединённые множества решений интер-
вальных линейных систем.

$$\begin{pmatrix} [3, 4] & [-0.77, 0.65] & [-0.77, 0.65] & [-0.77, 0.65] \\ [-0.77, 0.65] & [3, 4] & [-0.77, 0.65] & [-0.77, 0.65] \\ [-0.77, 0.65] & [-0.77, 0.65] & [3, 4] & [-0.77, 0.65] \\ [-0.77, 0.65] & [-0.77, 0.65] & [-0.77, 0.65] & [3, 4] \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} [-3, 3] \\ [-3, 3] \\ [-3, 3] \\ [-3, 3] \end{pmatrix}$$



Документация

Python
latest

Search docs

Установка
Реализация класса Interval
Общие функции
Линейные системы

Read the Docs
v: latest

Table of Contents

Edit on GitHub

intvalpy

Добро пожаловать в документацию intvalpy! Этот модуль Python реализует алгебраически замкнутую систему для работы с интервалами, решения интервальных систем как линейных, так и нелинейных уравнений, а также визуализирует множество решений.

Содержание

- Установка
 - Установка из источников
- Реализация класса Interval
 - Создание
 - Операции между интервалами
 - Методы класса Interval
 - Эмуляция коллекций
 - Согласование с другими библиотеками на Python
- Общие функции
 - Преобразование данных в интервальный тип
 - Интервальная диаграмма рассеяния
 - Пересечение интервалов
 - Метрика
 - Интервал из нулей
 - Тестовые системы
- Линейные системы
 - Вариабельность системы
 - Распознающие функционалы
 - Метод граничных интервалов
 - Методы для решения квадратных систем
 - Методы для решения переопределённых систем

Python
latest

Search docs

Установка
Реализация класса Interval
Общие функции

Линейные системы

- Вариабельность системы
- Распознающие функционалы
 - Распознающий функционал Uni
 - Распознающий функционал Tol
 - Список использованной литературы
- Метод граничных интервалов
- Методы для решения квадратных систем
- Методы для решения переопределённых систем

WRITE THE DOCS

Love Documentation? Write the Docs is for people like you! Join our virtual conferences or Slack.

Community Ad

Read the Docs
v: latest

Распознающие функционалы

Перед тем, как приступить к решению системы уравнений с интервальными данными необходимо понять, а разрешима ли она. Для этого рассматривается задача о распознавании разрешимости, т.е. непустоты множества решений. В случае интервальной линейной $(m \times n)$ -системы уравнений потребуется решить не более чем 2^n линейных неравенств размера $2m+n$. Это следует из факта о выпуклости и многогранности пересечения множеств решений интервальной системы линейных алгебраических уравнений (ИСЛАУ) с каждым из ортантов пространства $\mathbb{R}^n$. Уменьшение количества неравенств принципиально невозможно, что следует из факта труднорешаемости задачи, т.е. её NP-трудности. Ясно, что выше описанный метод применим лишь при малой размерности задачи, поэтому был предложен метод распознающего функционала.

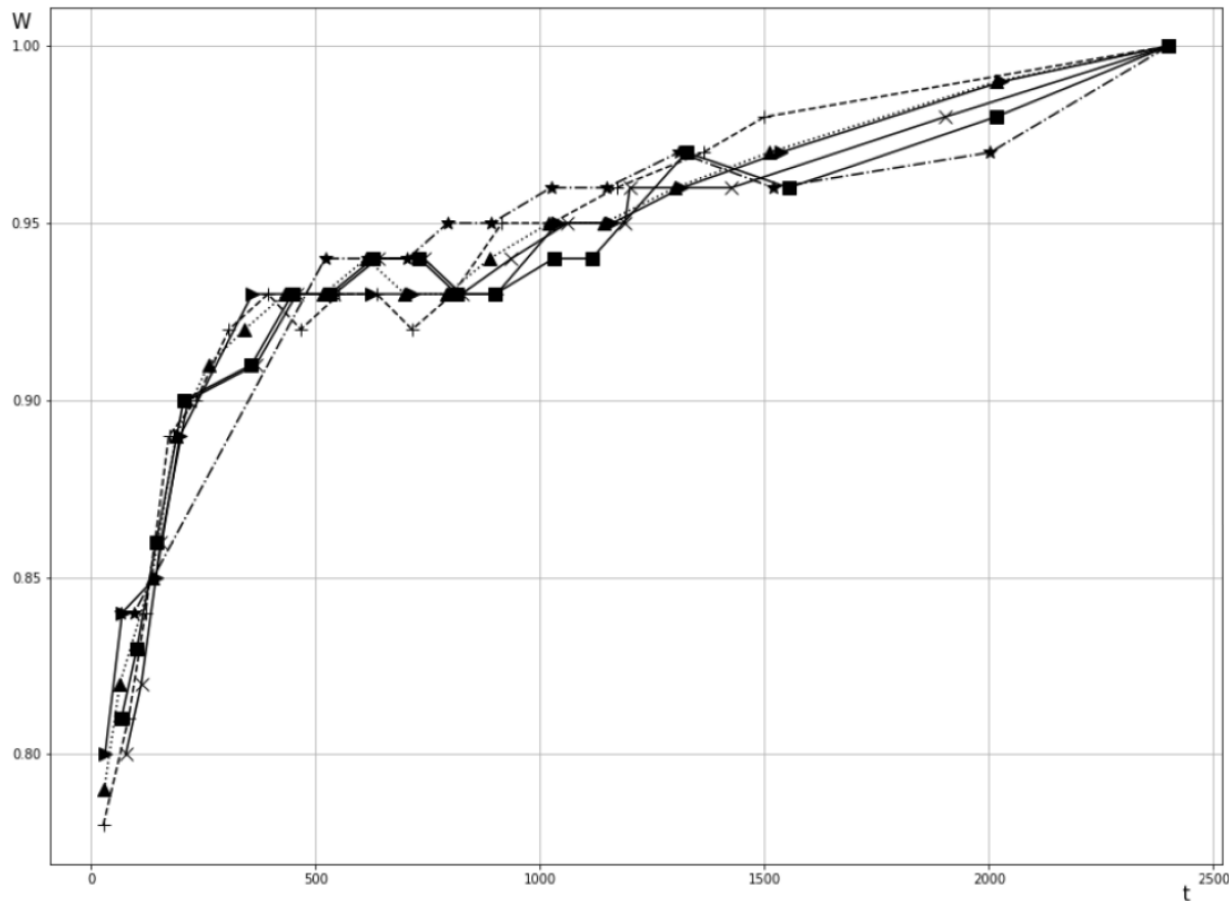
Распознающий функционал Uni

В случае, когда необходимо проверить интервальную систему линейных уравнений на её слабую разрешимость следует воспользоваться функционалом `Uni`. Для его оптимизации используется широко известный метод Нелдера-Мида, который не использует градиенты, поскольку в функции присутствует взятие абсолютного значения.

Parameters:

- A : Interval**
Входная интервальная матрица ИСЛАУ, которая может быть как квадратной, так и прямоугольной.
- b : Interval**
Интервальной вектор правой части ИСЛАУ.
- x : float, array_like, optional**
Точка в которой вычисляется распознающий функционал. По умолчанию x равен массиву из нулей.
- maxQ : bool, optional**
Если значение параметра равно True, то производится максимизация функционала.
- x0 : float, array_like, optional**
Первоначальная догадка для поиска глобального максимума.

Приложение



Для описания процесса формирования рыхлых осадков порошков металла была предложена следующая функция:

$$W(t) = \frac{D + t}{R + t}$$

D , R -параметры, измеряемые в секундах, на которые налагаются следующие условия:

$$R > D > 0$$

Приложение

Для обработки задано шесть выборок замеров

$$\{t_{n,m}, W_{n,m}\}, \quad t_{n,m} \in [0, 2400], \quad n = 1, \dots, 19, \quad m = 1, \dots, 6.$$

$$W(t) = \frac{D+t}{R+t} \quad \xrightarrow{\text{X}} \quad (1, -W) \begin{pmatrix} D \\ R \end{pmatrix} = t \cdot (W - 1)$$

$$\arg \max \text{Uni} \approx (170.69, 223.72) = (c_1, c_2)$$

$$\max \text{Uni} \approx 0.0018,$$

$$D = W \cdot (c_2 + t) - t = \gamma.$$

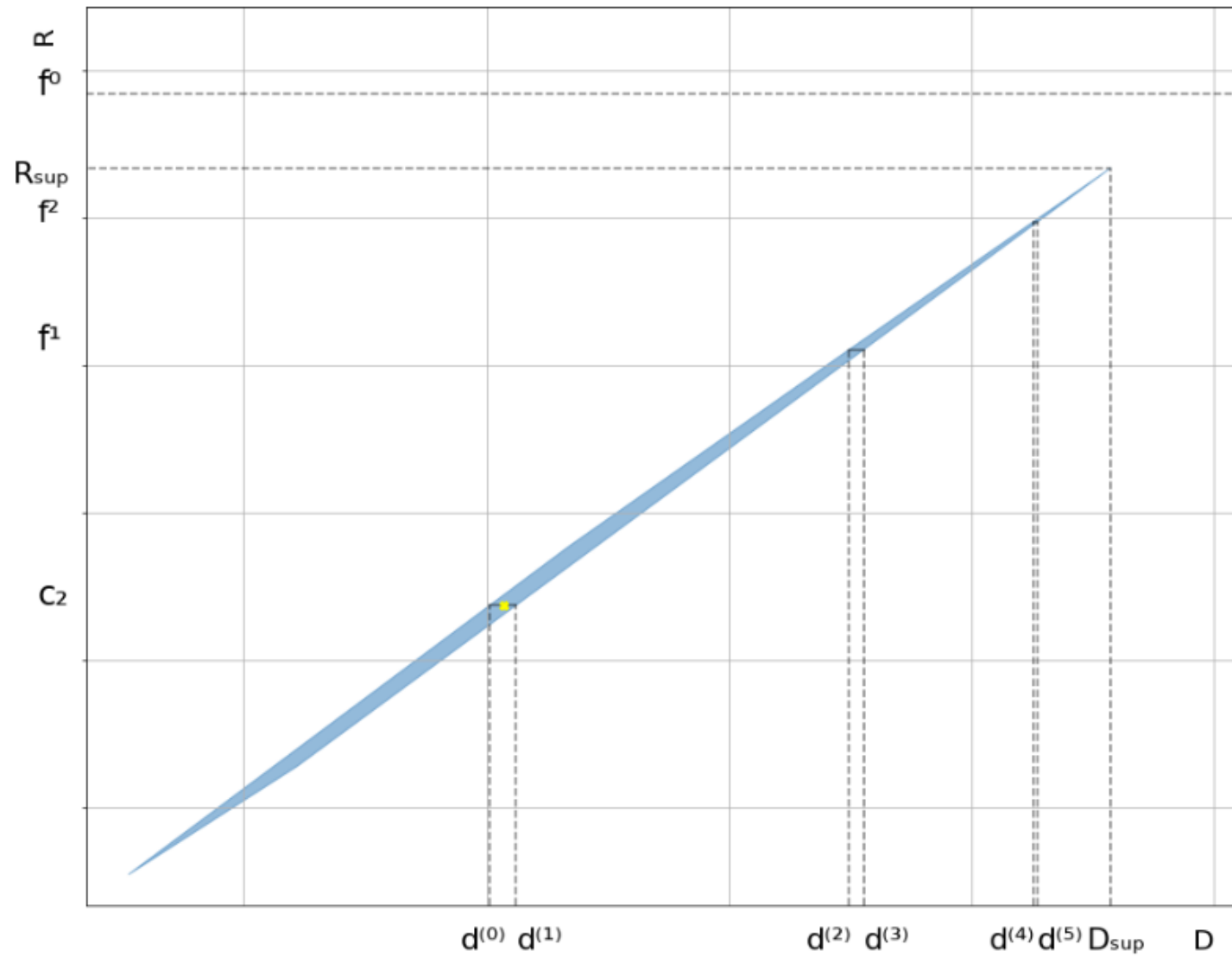
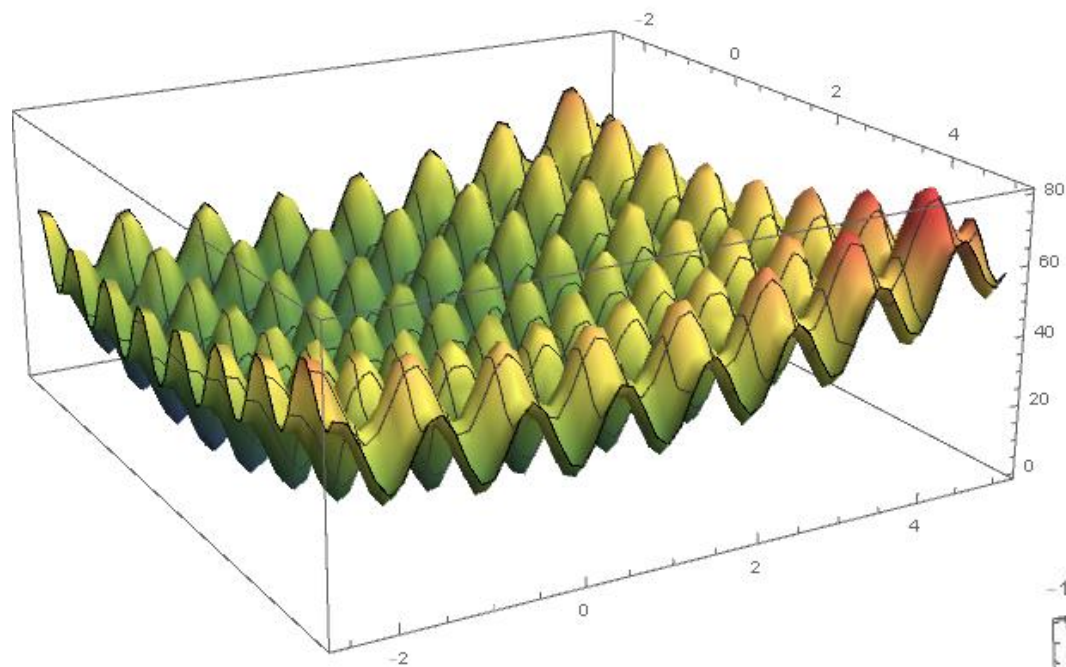


Рисунок 3 – Иллюстрация получения внешней оценки для значений первого аргумента.

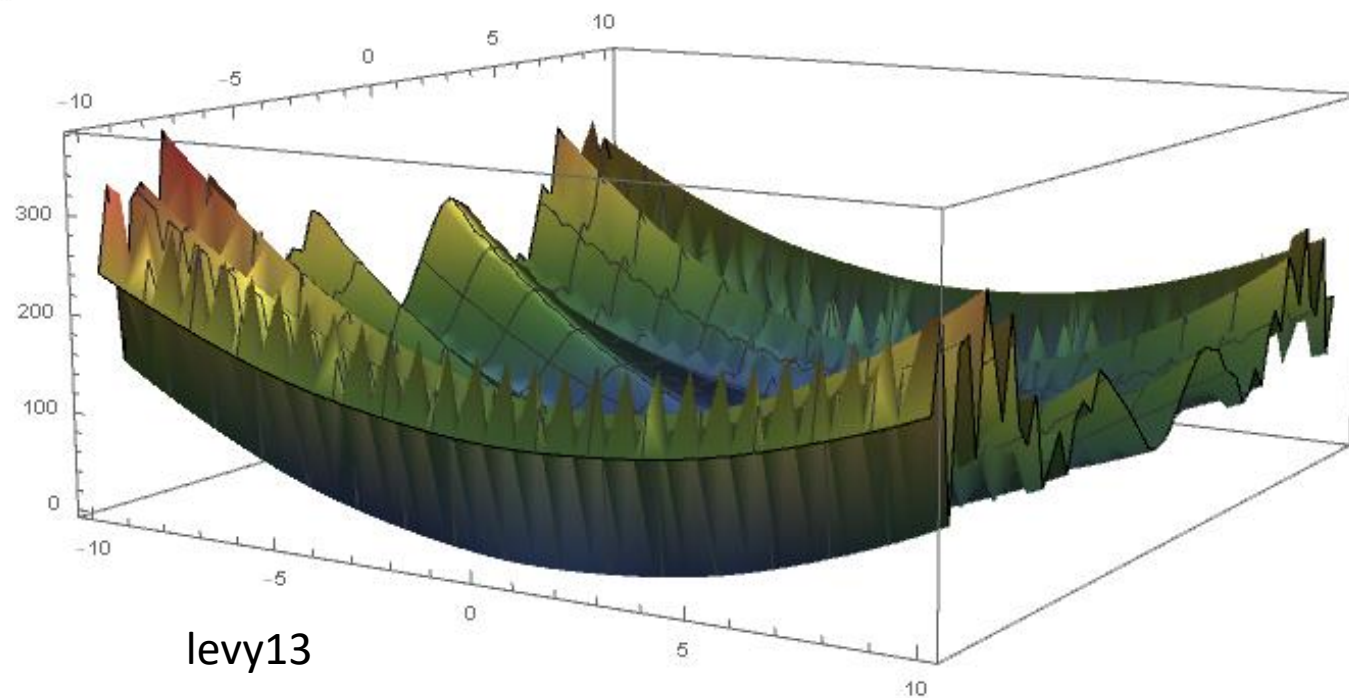
Сравнение с другими библиотеками

F	dim	PyInterval			PyIbex			IntvalPy			
		N_{it}	ϵ	t_{cpu}	N_{it}	ϵ	t_{cpu}	N_{it}	ϵ	t_{cpu}	
										double	ext.
Branin (RC)	2	160	7.1^{-15}	0.301	164	8.88^{-15}	0.047	142	8.9^{-15}	0.107	0.517
Bohachevsky (B_m)	2	66	6.66^{-15}	0.149	66	6.66^{-15}	0.016	66	6.44^{-15}	0.061	0.278
Beale (BL)	2	342	5.66^{-15}	0.833	342	5.66^{-15}	0.087	342	5.66^{-15}	0.373	1.637
Booth (BO)	2	146	7.9^{-15}	0.214	146	7.9^{-15}	0.035	146	7.9^{-15}	0.079	0.316
De Joung (DJ)	3	80	8.73^{-15}	0.07	80	8.73^{-15}	0.011	80	8.73^{-15}	0.029	0.157
Easom (ES)	2	63	8.1^{-15}	0.116	63	8.55^{-15}	0.016	70	4.44^{-15}	0.061	0.217
Eggholder	2	10000	1.06^{-5}	246.7	10000	1.06^{-5}	5.45	10000	1.06^{-5}	15.94	57.32
Rastrigin (RT_n)	4	10000	4.26^{-14}	42.73	10000	4.26^{-14}	78.03	129	7.11^{-15}	0.311	1.137
Schaffer № 4	2	10000	2.83^{-7}	27.3	10000	2.83^{-7}	3.897	10000	2.83^{-7}	11.57	68.18
Levy № 13	2	10000	1.7^{-14}	30.57	10000	3.4^{-14}	4.085	68	8.57^{-15}	0.097	0.468

F — целевая функция, dim — размерность задачи, N_{it} — ограниченное количество итераций для остановки оптимизации, ϵ — минимальная допустимая погрешность, t_{cpu} — усреднённое 50-ю запусками время выполнения задачи, double — двойная точность, extended — повышенная точность.



RT2



levy13

Сравнение с другими библиотеками

Таблица 4 – Качественное сравнение интервальных библиотек.

Библиотека	Арифм.	Мат. операции	Визуал.	Оценка парам.	Решение уравнений	Оптим.
IntvalPy	+	+	+	+	+	+
PyInterval	+	—	—	—	+	—
PyIbex	—	+	+	—	+	+
Octave ²	—	+	—	+	+	—
WM ³	+	+	—	—	—	—
JInterval	+	+	+	+	+	+
JI ⁴	—	+	+	—	+	+
IntLab	+	+	—	+	+	+